



Framework 7

tutorialspoint

S I M P L Y E A S Y L E A R N I N G

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

Framework7 is a free and open source framework for mobile HTML. It is used for developing hybrid mobile apps or web apps for iOS and Android devices.

This tutorial will teach you basic Framework7 and will also take you through some detailed concepts.

Audience

This tutorial has been prepared for the beginners to help them understand basic Framework7. After completing this tutorial, you will find yourself at a moderate level of expertise in Framework7 from where you can take yourself to next levels.

Prerequisites

Framework7 is based on HTML5 and JavaScript programming language, so if you have basic understanding on JavaScript programming and HTML5 then it will be a fun to learn.

Copyright & Disclaimer

© Copyright 2017 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial	i
Audience.....	i
Prerequisites.....	i
Copyright & Disclaimer	i
Table of Contents	ii
1. Framework7 – Overview	1
2. Framework7 – Environment.....	2
FRAMEWORK7 COMPONENTS	8
3. Framework7 – Layout	9
Framework7 – Static Layout	9
Framework7 – Fixed Layout	12
Framework7 – Through Layout	16
Framework7 – Mixed Layout.....	20
4. Framework7 – Navbars	25
Framework7 – Basic Navbar	26
Framework7 – Navbar with Links	28
Framework7 – Multiple Links	30
Framework7 – Links with Text & Icons.....	32
Framework7 – Links with Only Icons	34
Framework7 – Related App & View Methods	36
Framework7 – Hide Navbar Automatically	40
5. Framework7 – Toolbars	44
Framework7 – Toolbar Hiding	44
Framework7 – Toolbar Bottom	48
Methods of Toolbar	51
6. Framework7 – Search Bar	55
Search Bar Parameters	55
Search Bar Callbacks	56
Search Bar Properties.....	56
Search Bar Methods	57
Search Bar JavaScript Events	57
7. Framework7 – Status Bar	65
8. Framework7 – Side Panels	68
Framework7 – Open & Close Panels	69
Framework7 – Open & Close Panels From HTML.....	69
Framework7 – Open & Close Panels From JS	71
Framework7 – Panel Events	74
Framework7 – Open Panels With Swipe	77
Framework7 – Panel Is Opened?.....	80

9. Framework7 – Content Block	81
10. Framework7 – Layout Grid.....	84
11. Framework7 – Overlays	91
Framework7 – Overlay Modal	91
Framework7 – Alert Modal	92
Framework7 – Confirm Modal	95
Framework7 – Prompt Modal	99
Framework7 – Login and Password Modal	102
Framework7 – Nesting Modal	105
Framework7 – Preloader Modal.....	107
Framework7 – Indicator Modal	110
Custom Modals.....	112
Modal Events.....	113
Framework7 – Overlay Popup	114
Framework7 – Overlay Popup from HTML.....	114
Framework7 – Overlay Popup using JavaScript.....	117
Popup Events.....	120
Framework7 – Overlay Dynamic Popup	120
Framework7 – Popover	123
Framework7 – Open & Close Popover	124
Framework7 – Open & Close Popover From HTML.....	124
Framework7 – Open & Close Popover Using JavaScript.....	127
Framework7 – Dynamic Popover	130
Framework7 – Popover Events.....	134
Framework7 – Action Sheet	134
Framework7 – Create & Open Action Sheet	135
Framework7 – Convert Action Sheet To Popover	142
Framework7 – Close Action Sheet.....	144
Framework7 – Action Sheet Events.....	145
Framework7 – Action Sheet Template	145
Framework7 – Action Sheet To Popover Template.....	146
Framework7 – Overlay Login Screen	147
Framework7 – Login Screen from HTML	147
Framework7 – Login Screen from JavaScript	150
Login Screen Events.....	154
Framework7 – Start App With Login Screen	154
Framework7 – Embedded Login Screen	158
Framework7 – Overlay Picker Modal	161
Framework7 – Overlay Picker Modal from HTML	161
Framework7 – Overlay Picker Modal using JavaScript	164
Popup Events.....	167
Framework7 – Overlay Dynamic Picker Modal	167
12. Framework7 – Preloaders	171
13. Framework7 – Progress Bar	174
14. Framework7 – List Views	182
Framework7 – List Views.....	182
List View layout	183

Framework7 – List Block.....	183
Framework7 – Link Elements	187
Framework7 – List Divider.....	190
Framework7 – Grouped Lists.....	193
Framework7 – Inset List Block.....	199
Framework7 – List View Contacts List	202
Framework7 – Media List View	209
Framework7 – Swipeout.....	218
Swipeout Classes	218
Swipeout Types	218
Framework7 – Basic Swipeout	219
Framework7 – Swipe To Delete.....	224
Framework7 – Overswipe.....	229
Framework7 – Swipeout JavaScript API	234
Framework7 – Swipeout Events	239
Framework7 – Sortable List.....	243
Framework7 – Overlay Sortable List from HTML	243
Framework7 – Overlay Sortable List from JavaScript	247
Sortable Events.....	252
Framework7 – Virtual List.....	252
Virtual List Parameters	253
Virtual List Properties	254
Virtual List Methods	255
15. Framework7 – Accordion	263
16. Framework7 – Cards	272
Framework7 – Card HTML Layout.....	272
Framework7 – List View With Cards.....	280
17. Framework7 – Chips	283
18. Framework7 – Buttons.....	289
iOS Theme Buttons.....	289
Material Theme Buttons.....	295
Framework7 – Action Button	302
Framework7 – Floating Action Button Layout.....	302
Framework7 – Morph To Popover	305
Framework7 – Floating Action Button Speed Dial.....	309
19. Framework7 – Forms	314
Framework7 – Form Elements	315
Framework7 – Full Layout	316
Framework7 – Icons and Inputs	323
Framework7 – Labels and Inputs.....	328
Framework7 – Just Inputs	332
Framework7 – Inset.....	336
Framework7 – Checkboxes & Radios	340
Framework7 – Checkboxes Group.....	340
Framework7 – Radios Group iOS.....	343
Framework7 – Radios Group Material	346
Framework7 – With Media List View	349

Framework7 – Smart Select	355
Framework7 – Smart Select Layout.....	357
Framework7 – Smart Select Search Bar	360
Framework7 – Custom Page Title and Back Link Text	364
Framework7 – Open in Popup.....	368
Framework7 – Open in Picker	371
Framework7 – Custom icons, colors and images	374
Framework7 – Multiple Select and Optgroup	377
Framework7 – Multiple Select and Maxlength	380
Framework7 – Forms Disabled Elements	383
Framework7 – Form Data.....	386
Framework7 – Form To JSON	386
Framework7 – Form From JSON.....	390
Framework7 – Form Storage	393
Framework7 – Enable Form Storage	394
Framework7 – Form Storage JavaScript API.....	397
Ajax Form Submit	402
20. Framework7 – Tabs.....	404
Framework7 – Inline Tabs	406
Framework7 – Switch Tabs From Navbar	409
Framework7 – Switch Views From Tab Bar	412
Framework7 – Animated Tabs.....	416
Framework7 – Swipeable Tabs.....	419
Framework7 – Tabs JavaScript Events.....	422
Framework7 – Show Tab using JavaScript.....	425
21. Framework7 – Swiper Slider	430
Framework7 – Default Swiper with Pagination.....	432
Framework7 – Vertical Swiper	434
Framework7 – With Space Between Slides	436
Framework7 – Multiple Swipers.....	438
Framework7 – Nested Swipers.....	444
Framework7 – Custom Controls.....	446
Framework7 – Lazy Loading	448
22. Framework7 – Photo Browser.....	452
Framework7 – Create Photo Browser Instance.....	452
Framework7 – Photo Browser Parameters	453
Framework7 – Photo Browser Methods & Properties	456
Framework7 – Photos Array.....	457
Framework7 – Photo Browser Templates	462
23. Framework7 – Autocomplete	465
24. Framework7 – Picker	481
Framework7 – Picker With Single Value.....	486
Framework7 – Two Values and 3D-Rotate Effect.....	489
Framework7 – Dependent Values	491
Framework7 – Custom Toolbar	494
Framework7 – Inline Picker / Date-time	498

25. Framework7 – Calendar	503
Framework7 – Calendar Parameters	503
Calendar Methods & Properties	511
Access To Calendar's Instance	512
26. Framework7 – Refresh	519
27. Framework7 – Infinite Scroll	525
28. Framework7 – Messages.....	532
Initializing Messages with JavaScript.....	534
Messages Parameters.....	535
Messages Properties	535
Messages Methods.....	535
Single Message Parameters.....	537
Initializing Messages with HTML	537
29. Framework7 – Message Bar	542
Messagebar Parameter	543
Messagebar Properties.....	543
Messagebar Methods.....	543
Initialize Messagebar with HTML	544
Access to Messagebar's Instance	544
30. Framework7 – Notifications.....	545
Framework7 – Notifications JavaScript API	545
Framework7 – Notifications Layout	547
Framework7 – Example iOS.....	548
Framework7 – Example Material	551
31. Framework7 – Lazy Load.....	555
Framework7 – Lazy Load Usage	555
Framework7 – Init Lazy Load Manually	556
FRAMEWORK7 STYLING	561
32. Framework7 – Color Themes	562
Layout Themes	563
Framework7 – iOS Theme Colors	563
Framework7 – Material Theme Colors	568
33. Framework7 – Hairlines	576
FRAMEWORK7 TEMPLATES.....	577
34. Framework7 – Templates Overview.....	578
35. Framework7 – Auto Compilation	579

36. Framework7 – Template7 Pages 581
 Templates/Pages Data..... 581
 Page & URL Rules..... 584
 Pass Custom Context 584
 Load Templates Directly 585
 Framework7 – URL Query 586

FRAMEWORK7 FAST CLICKS 592

37. Framework 7 – Active State 593

38. Framework7 – Tap Hold Event 594

39. Framework7 – Touch Ripple..... 595



1. Framework7 – Overview

Framework7 is a free and open source framework for mobile HTML. It is used for developing hybrid mobile apps or web apps for iOS and Android devices.

Framework7 was introduced in the year 2014. The latest version 1.4.2 was released in February 2016 licensed under MIT.

Why Use Framework7?

- It is easier to develop apps for iOS and Android.
- The learning curve for Framework7 is very easy.
- It has many pre-styled widgets/components.
- It has built-in helper libraries.

Features

- Framework7 is an open source and free to use framework.
- Framework7 has easy and familiar jQuery syntax to get started without any delay.
- To control click delay for touch UI's, Framework7 has built-in *FastClick* library.
- Framework7 has built-in grid system layout for arranging your elements responsively.
- Framework7 dynamically loads pages from templates via *flexible router api*.

Advantages

- Framework7 is not dependent on any third party library even for DOM manipulation. Instead, it has its own custom DOM7.
- Framework7 can also be used with *Angular* and *React* frameworks.
- You can start creating apps once you know HTML, CSS and some basic JavaScript.
- It supports faster development via *Bower*.
- It is easy to develop apps for iOS and Android without learning it.

Disadvantages

- Framework7 only supports platforms like iOS and Android.
- The online community support for Framework7 framework is less compared to iOS and Android.

2. Framework7 – Environment

In this chapter, we will discuss about how to install and setup Framework7.

You can download the Framework7 in two ways:

Download from [Framework7 Github repository](#)

You can install the Framework7, using Bower as shown below:

```
bower install framework7
```

After successful installation of Framework7, you need to follow the below given steps to make use of Framework7 in your application:

Step 1: You need to install **gulp-cli** to build development and dist versions of Framework7 by using the following command:

```
npm install gulp-cli
```

The *cli* stands for Command Line Utility for Gulp.

Step 2: The Gulp must be installed globally by using the following command:

```
npm install --global gulp
```

Step 3: Next, install the NodeJS package manager, which installs the node programs that makes it easier to specify and link dependencies. The following command is used to install the npm:

```
npm install
```

Step 4: The development version of Framework7 can be built by using the following command:

```
npm build
```

Step 5: Once you build the development version of Framework7, start building the app from *dist/* folder by using the following command:

```
npm dist
```

Step 6: Keep your app folder in the server and run the following command for navigation between pages of your app:

```
gulp server
```

Download Framework7 Library from CDNs

A CDN or Content Delivery Network is a network of servers designed to serve files to users. If you use a CDN link in your web page, it moves the responsibility of hosting files from your own servers to a series of external ones. This also offers an advantage that if a visitor to your webpage has already downloaded a copy of Framework7 from the same CDN, it won't have to be re-downloaded. You can include the following CDN files into the HTML document.

The following CDNs are used in an **iOS App layout**:

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ios.min.css">
```

It is used to include Framework7 iOS CSS Library to your application.

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ios.colors.min.css">
```

It is used to include Framework7 iOS related color styles to your application.

The following CDNs are used in **Android/Material App Layout**:

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.js"></script>
```

It is used to include Framework7 JS library to your application.

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.material.min.css"></script>
```

It is used to include Framework7 Material styling to your application.

We are using the CDN versions of the library throughout this tutorial. We use AMPPS (AMPPS is a WAMP, MAMP and LAMP stack of Apache, MySQL, MongoDB, PHP, Perl & Python) server to execute all our examples.

Example

The following example demonstrates the use of simple application in the Framework7, which will display the alert box with the customized message when you click on the navigation bar.

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
```

```

<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style" content="black">
<title>My App</title>
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
  //you can control the background color of the Status bar
  <div class="statusbar-overlay"></div>
  <div class="panel-overlay"></div>
  <div class="panel panel-right panel-reveal">
    <div class="content-block">
      <p>Contents goes here...</p>
    </div>
  </div>
  <div class="views">
    <div class="view view-main">
      <div class="navbar">
        <div class="navbar-inner">
          <div class="center sliding">My App</div>
          <div class="right">
            <a href="#" class="link icon-only open-panel"><i
class="icon icon-bars"></i></a>
          </div>
        </div>
      </div>
      <div class="pages navbar-through toolbar-through">
        <div data-page="index" class="page">
          <div class="page-content">
            <p>This is simple application...</p>
            <div class="list-block">
              <ul>
                <li>
                  <a href="envirmnt_about.html" class="">
                    <div class="item-content">

```

```

        <div class="item-inner">
            <div class="item-title">Blog</div>
        </div>
    </div>
</a>
</li>
</ul>
</div>
</div>
</div>
</div>
<div class="toolbar">
    <div class="toolbar-inner">
        <a href="#" class="link">First Link</a>
        <a href="#" class="link">Second Link</a>
    </div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // here initialize the app
    var myApp = new Framework7();

    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;

    // Add the view
    var mainView = myApp.addView('.view-main', {
        // enable the dynamic navbar for this view:
        dynamicNavbar: true
    });

    //use the 'pageInit' event handler for all pages
    $$(<document>).on('pageInit', function (e) {
        //get page data from event data

```

```

        var page = e.detail.page;

        if (page.name === 'blog') {
            // you will get below message in alert box when page with data-
            // page attribute is equal to "about"
            myApp.alert('Here its your About page');
        }
    })
</script>
</body>
</html>

```

Next, create one more HTML page i.e. **envirmnt_about.html** as shown below:

envirmnt_about.html

```

<div class="navbar">
  <div class="navbar-inner">
    <div class="left">
      <a href="#" class="back link">
        <i class="icon icon-back"></i>
        <span>Back</span>
      </a>
    </div>
    <div class="center sliding">My Blog</div>
    <div class="right">
      <a href="#" class="link icon-only open-panel"><i class="icon icon-
      bars"></i></a>
    </div>
  </div>
</div>
<div class="pages">
  <div data-page="blog" class="page">
    <div class="page-content">
      <div class="content-block">
        <h2>My Blog</h2>
        <p>Lorem Ipsum is simply dummy text of the printing and typesetting
        industry. Lorem Ipsum has been the industry's standard dummy text ever since
        the 1500s, when an unknown printer took a galley of type and scrambled it to
        make a type specimen book.</p>
      </div>
    </div>
  </div>
</div>

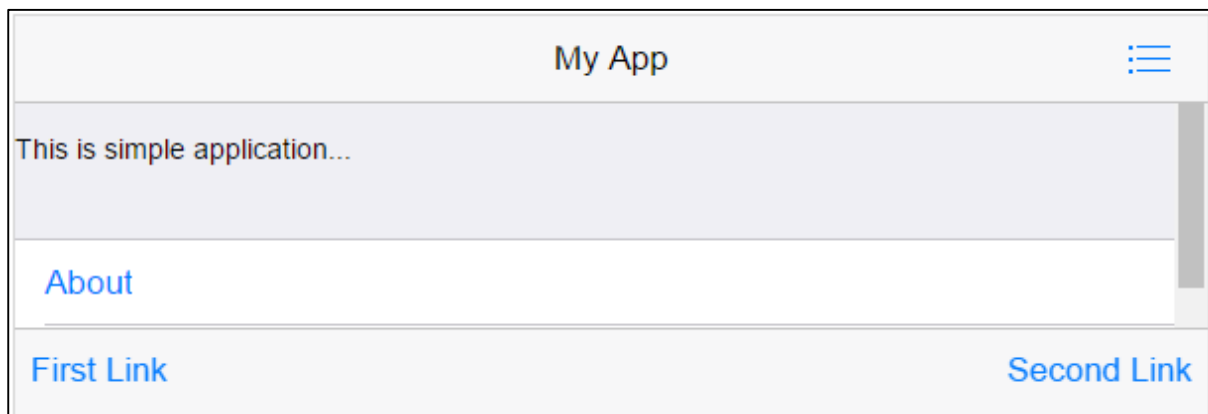
```

```
    </div>  
  </div>  
</div>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above HTML code as **framework7_environment.html** file in your server root folder.
- Open this HTML file as http://localhost/framework7_environment.html and output is displayed as shown below.
- When you click on the navigation bar, it will display the alert box with the customized message.



Framework7 Components

3. Framework7 – Layout

Description

Framework7 provides different types of layouts for your application. It supports three types of Navbar/Toolbar layouts:

S. No.	Layout Types & Description
1	Static Layout The static layout is most often used layout-type and includes navbar and toolbar, which can be a scrollable page content and each page contains its own navbar and toolbar.
2	Fixed Layout Fixed layout includes its own navbar and toolbar, which can be visible on screen and cannot be scrollable on page.
3	Through Layout In this layout, the navbar and the toolbar appears fixed for all pages within single view.
4	Mixed Layout You can mix the different types of layouts in the single view.

No Navbar/Toolbar

If you do not want to use navbar and toolbar, then do not include appropriate classes (*navbar-fixed, navbar-through, toolbar-fixed, toolbar-through*) to page/pages/view.

Framework7 – Static Layout

Description

The Framework7 static layout is rarely used layout type. It includes Navbar and Toolbar as part of scrollable page content.

Example

The following example demonstrates the use of static layout in the Framework7. The Navbar and Toolbar will disappear as you scroll and does not appear to be fixed.

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
```

```

<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style" content="black">
<title>Static Layout</title>
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
<div class="views">
<div class="view view-main">
<div class="pages">
<div data-page="index" class="page">
<div class="page-content">
<div class="navbar">
<div class="navbar-inner">
<div class="left"></div>
<div class="center">Static Navbar</div>
<div class="right"></div>
</div>
</div>
<div class="content-block">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
<p>Donec et nulla auctor massa pharetra adipiscing ut
sit amet sem. Suspendisse molestie velit vitae mattis tincidunt. Ut sit amet
quam mollis, vulputate turpis vel, sagittis felis. Vestibulum ornare ut eros
vitae adipiscing. Vestibulum volutpat justo enim, ullamcorper vulputate sapien
lacinia vel. Integer sed justo ultrices augue tincidunt dictum eu vel orci.
Mauris sodales auctor diam vel condimentum.</p>
<p>Praesent mauris purus, faucibus vel hendrerit at,
dapibus quis lorem. Sed placerat fermentum blandit. Suspendisse potenti. Cras
sollicitudin laoreet tellus, ut gravida leo eleifend convallis. Sed pharetra
nisl quis libero fermentum pharetra. Cras lacinia quam turpis, eget varius
risus interdum sit amet. Quisque laoreet tortor dui, vitae accumsan lacus
fringilla in. Quisque consequat placerat risus, non ornare felis scelerisque

```

```
quis. Sed adipiscing diam tellus, vel faucibus mauris rhoncus vel. Vestibulum eu ultrices tortor, non suscipit lorem. Mauris tellus nulla, volutpat quis lacus eu, scelerisque adipiscing dui. Nullam nec tempor sem, nec pulvinar sapien. Etiam blandit condimentum vehicula.</p>
```

```
<p>Praesent nec imperdiet diam. Maecenas vel lectus porttitor, consectetur magna nec, viverra sem. Aliquam sed risus dolor. Morbi tincidunt ut libero id sodales. Integer blandit varius nisi quis consectetur. Nulla pellentesque elementum ligula vitae porta. Nunc sollicitudin mi quis mi mattis cursus. Nulla diam felis, ullamcorper eget lacinia ac, auctor id velit. Fusce enim nunc, egestas a augue vitae, malesuada tincidunt risus. Nullam fringilla, enim nec porta iaculis, enim leo pharetra nunc, eget rutrum tortor dui et risus. Etiam sit amet molestie dolor. Curabitur ultrices justo ut augue ornare, vel pharetra libero adipiscing. Duis rhoncus a felis ac venenatis. Duis posuere non leo vitae tincidunt. Integer luctus arcu ut risus posuere, vel vehicula ipsum elementum. Duis et cursus sapien. </p>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu consectetur. Sed posuere a orci id imperdiet.</p>
```

```
</div>
```

```
<div class="toolbar">
```

```
<div class="toolbar-inner">
```

```
<a href="#" class="link">First Link</a>
```

```
<a href="#" class="link">Second Link</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.js"></script>
```

```
<script>
```

```
// here initialize the app
```

```
var myApp = new Framework7();
```

```
// If your using custom DOM library, then save it to $$ variable
```

```
var $$ = Dom7;
```

```
// Add the view
```

```
var mainView = myApp.addView('.view-main', {
```

```

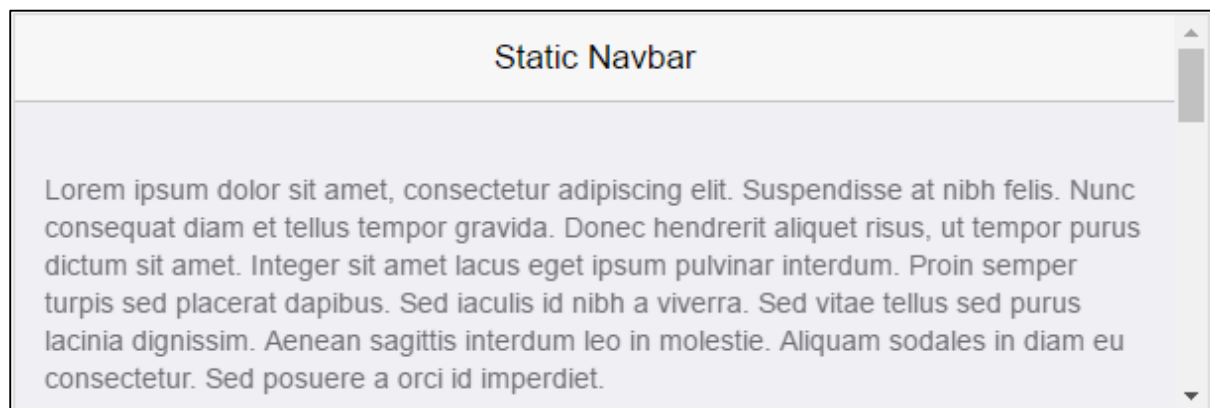
    // enable the dynamic navbar for this view:
    dynamicNavbar: true
  });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code as **static_layout.html** file in your server root folder.
- Open this HTML file as http://localhost/static_layout.html and the output is displayed as shown below.
- The static layout includes Navbar and Toolbar as parts of scrollable page content and each page has its own Navbar and Toolbar.



Framework7 – Fixed Layout

Description

In this layout, the Navbar and Toolbar will be fixed and cannot be scrollable on page. Each page can also have its own Navbar and Toolbar.

Example

The following example demonstrates the use of fixed layout in the Framework7. The Navbar and Toolbar in the following example appears to be fixed.

First, we will create our **fixed_layout.html** file.

```

<!DOCTYPE html>
<html>

```

```

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="apple-mobile-web-app-status-bar-style" content="black">
  <title>Fixed Layout</title>
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
  <div class="views">
    <div class="view view-main">
      <div class="pages">
        <div data-page="home" class="page navbar-fixed toolbar-fixed">
          <div class="navbar">
            <div class="navbar-inner">
              <div class="left"></div>
              <div class="center">Fixed Navbar</div>
              <div class="right"></div>
            </div>
          </div>
          <div class="page-content">
            <div class="content-block">
              <p> <a href="fixed_layout-services.html">Services
page</a></p>
              <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
              <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

```

```

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
        </div>
    </div>
    <div class="toolbar">
        <div class="toolbar-inner"> <a href="#" class="link">Link
1</a><a href="#" class="link">Link 1</a></div>
        </div>
    </div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var mainView = myApp.addView('.view');
</script>
</body>
</html>

```

Next, create the HTML page i.e. **fixed_layout-services.html** as shown below:

fixed_layout-services.html

```

<div class="views">
    <div class="view view-main">
        <div class="pages">
            <div data-page="services" class="page navbar-fixed toolbar-fixed">
                <div class="navbar">
                    <div class="navbar-inner">
                        <div class="center">Services Page</div>
                    </div>
                </div>
                <div class="page-content">
                    <div class="content-block">

```

```

        <p> <a href="bars-fixed-index.html" class="back">Go back to
Home page</a></p>

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec
hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit amet
lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat dapibus.
Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia dignissim.
Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec
hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit amet
lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat dapibus.
Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia dignissim.
Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec
hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit amet
lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat dapibus.
Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia dignissim.
Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec
hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit amet
lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat dapibus.
Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia dignissim.
Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

    </div>

</div>

<div class="toolbar">

    <div class="toolbar-inner"> <a href="#" class="link">Service
link 1</a><a href="#" class="link">Service link 2</a></div>

    </div>

</div>

</div>

</div>

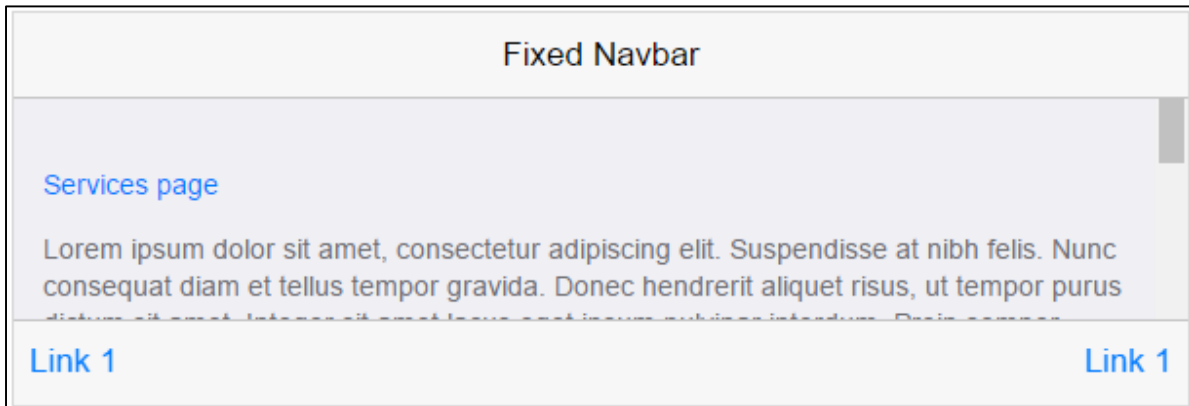
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code as **fixed_layout.html** file in your server root folder.
- Open this HTML file as http://localhost/fixed_layout.html and the output is displayed as shown below.

- When you scroll the page, Navbar and Toolbar will be fixed and they are visible on the screen.



You can create separate pages for all of the footer links.

Framework7 – Through Layout

Description

In this layout, the navbar and toolbar appears fixed for all pages within single view.

Example

The following example demonstrates the use of through layout in Framework7. The Navbar and Toolbar will be fixed and remains the same for all the pages.

First, we will create our **through_layout.html** file.

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Through Layout</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
```



```

<div class="views">
  <div class="view view-main">
    <div class="navbar">
      <div class="navbar-inner">
        <div class="left"></div>
        <div class="center sliding">Through Navbar</div>
        <div class="right"></div>
      </div>
    </div>
    <div class="pages navbar-through toolbar-through">
      <div data-page="home" class="page">
        <div class="page-content">
          <div class="content-block">
            <p> <a href="through-services.html">Services
page</a></p>
            <p>Praesent nec imperdiet diam. Maecenas vel lectus
porttitor, consectetur magna nec, viverra sem. Aliquam sed risus dolor. Morbi
tincidunt ut libero id sodales. Integer blandit varius nisi quis consectetur.
Nulla pellentesque elementum ligula vitae porta. Nunc sollicitudin mi quis mi
mattis cursus. Nulla diam felis, ullamcorper eget lacinia ac, auctor id velit.
Fusce enim nunc, egestas a augue vitae, malesuada tincidunt risus. Nullam
fringilla, enim nec porta iaculis, enim leo pharetra nunc, eget rutrum tortor
dui et risus. Etiam sit amet molestie dolor. Curabitur ultrices justo ut augue
ornare, vel pharetra libero adipiscing. Duis rhoncus a felis ac venenatis. Duis
posuere non leo vitae tincidunt. Integer luctus arcu ut risus posuere, vel
vehicula ipsum elementum. Duis et cursus sapien. </p>
            <p>Praesent mauris purus, faucibus vel hendrerit at,
dapibus quis lorem. Sed placerat fermentum blandit. Suspendisse potenti. Cras
sollicitudin laoreet tellus, ut gravida leo eleifend convallis. Sed pharetra
nisl quis libero fermentum pharetra. Cras lacinia quam turpis, eget varius
risus interdum sit amet. Quisque laoreet tortor dui, vitae accumsan lacus
fringilla in. Quisque consequat placerat risus, non ornare felis scelerisque
quis. Sed adipiscing diam tellus, vel faucibus mauris rhoncus vel. Vestibulum
eu ultrices tortor, non suscipit lorem. Mauris tellus nulla, volutpat quis
lacus eu, scelerisque adipiscing dui. Nullam nec tempor sem, nec pulvinar
sapien. Etiam blandit condimentum vehicula.</p>
            <p>Praesent nec imperdiet diam. Maecenas vel lectus
porttitor, consectetur magna nec, viverra sem. Aliquam sed risus dolor. Morbi
tincidunt ut libero id sodales. Integer blandit varius nisi quis consectetur.
Nulla pellentesque elementum ligula vitae porta. Nunc sollicitudin mi quis mi
mattis cursus. Nulla diam felis, ullamcorper eget lacinia ac, auctor id velit.
Fusce enim nunc, egestas a augue vitae, malesuada tincidunt risus. Nullam
fringilla, enim nec porta iaculis, enim leo pharetra nunc, eget rutrum tortor
dui et risus. Etiam sit amet molestie dolor. Curabitur ultrices justo ut augue
ornare, vel pharetra libero adipiscing. Duis rhoncus a felis ac venenatis. Duis
posuere non leo vitae tincidunt. Integer luctus arcu ut risus posuere, vel
vehicula ipsum elementum. Duis et cursus sapien. </p>
            <p>Praesent mauris purus, faucibus vel hendrerit at,
dapibus quis lorem. Sed placerat fermentum blandit. Suspendisse potenti. Cras

```

```
sollicitudin laoreet tellus, ut gravida leo eleifend convallis. Sed pharetra
nisl quis libero fermentum pharetra. Cras lacinia quam turpis, eget varius
risus interdum sit amet. Quisque laoreet tortor dui, vitae accumsan lacus
fringilla in. Quisque consequat placerat risus, non ornare felis scelerisque
quis. Sed adipiscing diam tellus, vel faucibus mauris rhoncus vel. Vestibulum
eu ultrices tortor, non suscipit lorem. Mauris tellus nulla, volutpat quis
lacus eu, scelerisque adipiscing dui. Nullam nec tempor sem, nec pulvinar
sapien. Etiam blandit condimentum vehicula.</p>
```

```
<p>Praesent nec imperdiet diam. Maecenas vel lectus
porttitor, consectetur magna nec, viverra sem. Aliquam sed risus dolor. Morbi
tincidunt ut libero id sodales. Integer blandit varius nisi quis consectetur.
Nulla pellentesque elementum ligula vitae porta. Nunc sollicitudin mi quis mi
mattis cursus. Nulla diam felis, ullamcorper eget lacinia ac, auctor id velit.
Fusce enim nunc, egestas a augue vitae, malesuada tincidunt risus. Nullam
fringilla, enim nec porta iaculis, enim leo pharetra nunc, eget rutrum tortor
dui et risus. Etiam sit amet molestie dolor. Curabitur ultrices justo ut augue
ornare, vel pharetra libero adipiscing. Duis rhoncus a felis ac venenatis. Duis
posuere non leo vitae tincidunt. Integer luctus arcu ut risus posuere, vel
vehicula ipsum elementum. Duis et cursus sapien. </p>
```

```
<p>Praesent mauris purus, faucibus vel hendrerit at,
dapibus quis lorem. Sed placerat fermentum blandit. Suspendisse potenti. Cras
sollicitudin laoreet tellus, ut gravida leo eleifend convallis. Sed pharetra
nisl quis libero fermentum pharetra. Cras lacinia quam turpis, eget varius
risus interdum sit amet. Quisque laoreet tortor dui, vitae accumsan lacus
fringilla in. Quisque consequat placerat risus, non ornare felis scelerisque
quis. Sed adipiscing diam tellus, vel faucibus mauris rhoncus vel. Vestibulum
eu ultrices tortor, non suscipit lorem. Mauris tellus nulla, volutpat quis
lacus eu, scelerisque adipiscing dui. Nullam nec tempor sem, nec pulvinar
sapien. Etiam blandit condimentum vehicula.</p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="toolbar">
```

```
<div class="toolbar-inner"> <a href="#" class="link">Link
1</a><a href="#" class="link">Link 2</a></div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
```

```
<script>
```

```
var myApp = new Framework7();
```

```
var mainView = myApp.addView('.view');
```

```
</script>
```

```
</body>
```

```
</html>
```

Next, create the HTML page i.e. ***through-services.html*** as shown below:

```
<div class="views">
  <div class="view view-main">
    <div class="pages">
      <div data-page="services" class="page navbar-through toolbar-through">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="center">Services Page</div>
          </div>
        </div>
        <div class="page-content">
          <div class="content-block">
            <p> <a href="through_layout.html" class="back">Go back to
Home page</a></p>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec
hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit amet
lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat dapibus.
Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia dignissim.
Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec
hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit amet
lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat dapibus.
Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia dignissim.
Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec
hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit amet
lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat dapibus.
Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia dignissim.
Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec
hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit amet
lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat dapibus.
Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia dignissim.
Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

```

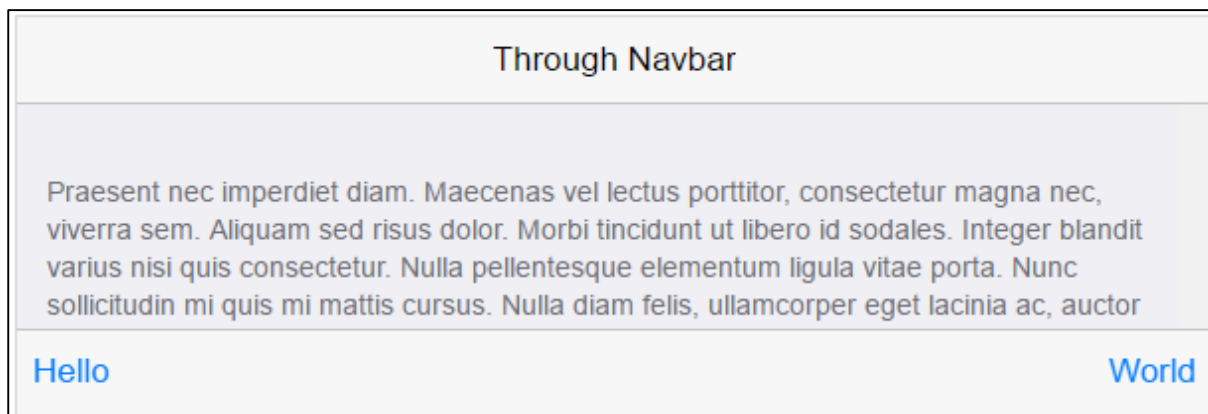
        <div class="toolbar">
            <div class="toolbar-inner"> <a href="#" class="link">Link
1</a><a href="#" class="link">Link 2</a></div>
            </div>
        </div>
    </div>
</div>
</div>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code as **through_layout.html** file in your server root folder.
- Open this HTML file as http://localhost/through_layout.html and the output is displayed as shown below.
- When you scroll the page, Navbar and Toolbar will be fixed and stay the same through all pages within single View.



Framework7 – Mixed Layout

Description

You can mix the different types of layouts in the single view.

Example

The following example demonstrates the use of mixed layout in the Framework7. The Navbar and Toolbars can have mixed layout like *fixed navbar* and *through toolbar* as shown below:

First, we will create our **mixed_layout.html** file.

```
<!DOCTYPE html>
```

```

<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>My App</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="center sliding">Mixed Navbar</div>
          </div>
        </div>
        <div class="pages navbar-fixed toolbar-through">
          <div data-page="index" class="page">
            <div class="page-content">
              <div class="list-block">
                <ul>
                  <li>
                    <a href="/framework7/src/mixed_layout-
services.html" class="">
                      <div class="item-content">
                        <div class="item-inner">
                          <div class="item-title">Go to Services
page</div>
                        </div>
                      </div>
                    </a>
                  </li>
                </ul>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>

```

```

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

```

```

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

```

```

    </div>

```

```

  </div>

```

```

</div>

```

```

<div class="toolbar">

```

```

  <div class="toolbar-inner">

```

```

    <a href="#" class="link">First Link</a>

```

```

    <a href="#" class="link">Second Link</a>

```

```

  </div>

```

```

</div>

```

```

</div>

```

```

  <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>

```

```

  <script>

```

```

    // here initialize the app

```

```

    var myApp = new Framework7();

```

```

    // If your using custom DOM library, then save it to $$ variable

```

```

    var $$ = Dom7;

```

```

    // Add the view

```

```

    var mainView = myApp.addView('.view-main', {

```

```

      // enable the dynamic navbar for this view:

```

```

      dynamicNavbar: true

```

```

    });

```

```

  </script>

```

```

</body>

```

```
</html>
```

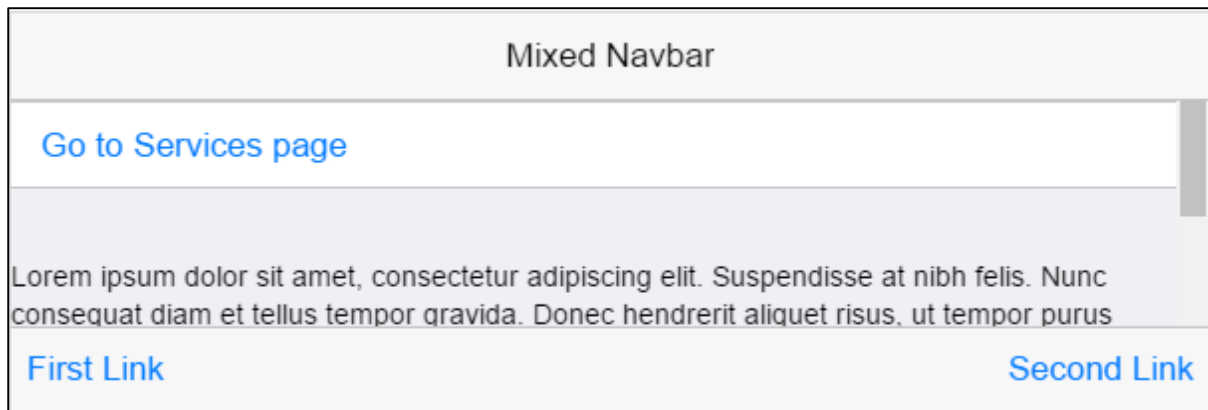
Next, create one more HTML page i.e. *mixed_layout-services.html* as shown below:

```
<div class="navbar">
  <div class="navbar-inner">
    <div class="center sliding">Services</div>
  </div>
</div>
<div class="pages">
  <div data-page="services" class="page">
    <div class="page-content">
      <div class="content-block">
        <br>
        <h2>Our Services</h2>
        <ul>
          <li>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed
do eiusmod tempor incididunt ut labore.</li>
          <li>Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris.</li>
          <li>Duis aute irure dolor in reprehenderit in voluptate velit
esse cillum dolore eu fugiat nulla pariatur.</li>
          <li>Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut
odit aut fugit.</li>
          <li>At vero eos et accusamus et iusto odio dignissimos ducimus
qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores
et.</li>
          <li>Nam libero tempore, cum soluta nobis est eligendi optio
cumque nihil impedit quo minus id.</li>
        </ul>
        <p>Go <a href="#" class="back">back</a> to home page.</p>
      </div>
    </div>
  </div>
</div>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code as **mixed_layout.html** file in your server root folder.
- Open this HTML file as `http://localhost/mixed_layout.html` and the output is displayed as shown below.
- You can see that the navbar is fixed and through layout is applied to toolbar.



4. Framework7 – Navbars

Description

In this chapter, let us study about *navbar*. It is a usually placed at top of a screen containing title of the page and navigation elements.

Navbar consists of three parts each of which may contain any HTML content, but it is suggested you use these in the way given below:

- **Left:** It is designed to place *back link* icons or single text link.
- **Center:** It is used to display title of the page or tab links.
- **Right:** This part is similar to the *left* part.

The following table demonstrates the use of *navbar* in detail:

S. No.	Navbar Types & Description
1	Basic navbar A basic navbar can be created by using the <i>navbar</i> , <i>navbar-inner</i> , <i>left</i> , <i>center</i> and <i>right</i> classes.
2	Navbar with links To use links in <i>left</i> and <i>right</i> part of your navbar, just add <code><a></code> tag with class <i>link</i> .
3	Multiple links To use multiple links, just add few more <code></code> to the part of your choice.
4	Links with text and icons The links can be provided with icons and texts by adding classes for icons and wrapping the link text with <code></code> element.
5	Links with only icons Navbar links can be provided with only icons by adding <i>icon-only</i> class to links.
6	Related app and view methods On initializing the <i>View</i> , framework7 allows you to use methods available for navbar.
7	Hide navbar automatically The navbar can be hidden/shown automatically for some Ajax loaded pages where navbar is not required.

Framework7 – Basic Navbar

Description

A basic navbar can be created by using the *navbar*, *navbar-inner*, *left*, *center* and *right* classes.

The *center* part has the lowest width priority, when screen size is reduced. If all the three elements (i.e., *center*, *left* and *right*) do not fit into the screen, then the *center* part will be removed.

Example

The following example demonstrates the use of navbar in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Navbar</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="left">Left</div>
            <div class="center">Center</div>
            <div class="right">Right</div>
          </div>
        </div>
        <div class="pages navbar-through ">
          <div data-page="index" class="page">
            <div class="page-content">
```

```

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

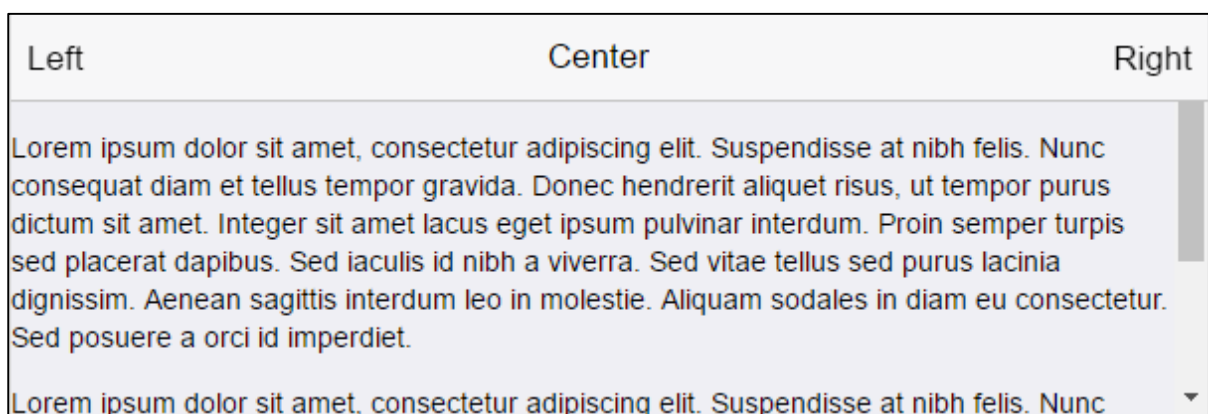
        </div>
    </div>
</div>
</div>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **navbar_basic.html** file in your server root folder.
- Open this HTML file as http://localhost/navbar_basic.html and the output is displayed as shown below.
- The basic navbar 3 parts such as left, Center and Right and each part includes any HTML content.



Framework7 – Navbar with Links

Description

To use links in *left* and *right* part of your navbar, just add `<a>` tag with class `link`.

Example

The following example displays the links at the top left and right corner sides in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Navbar With Links</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="left">
              <a href="#" class="link">Left Link</a>
            </div>
            <div class="center">Center</div>
            <div class="right">
              <a href="#" class="link">Right Link</a>
            </div>
          </div>
        </div>
      </div>
      <div class="pages navbar-through ">
        <div data-page="index" class="page">
```

```

        <div class="page-content">
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

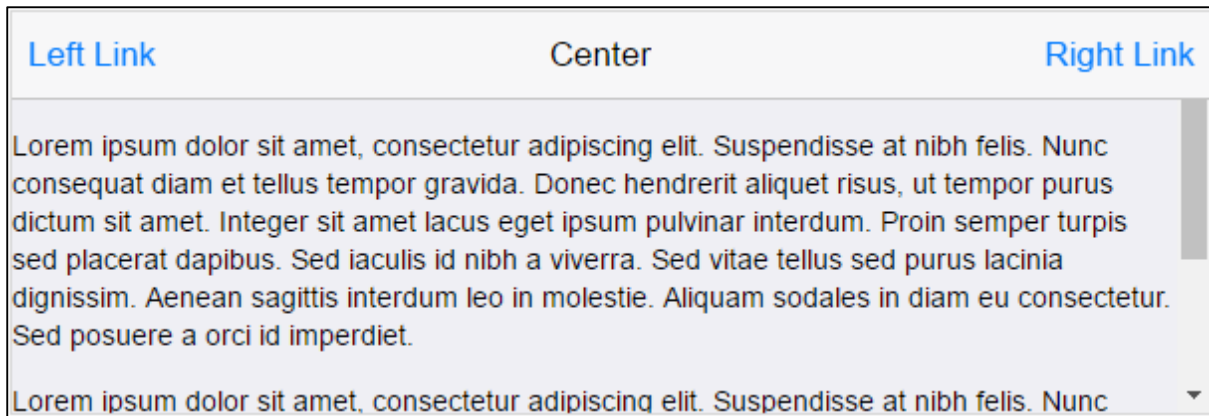
        </div>
    </div>
</div>
</div>
</div>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
    </body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code as **navbar_links.html** file in your server root folder.
- Open this HTML file as http://localhost/navbar_links.html and the output is displayed as shown below.
- The navbar provides links in the left and right part of the page.



Framework7 – Multiple Links

Description

To use multiple links, just add a few more `` to any part of your choice.

Example

The following example demonstrates the use of navbar with multiple links in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Navbar With Multiple Links</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="left">
              <a href="#" class="link">Left 1</a>
```

```

        </div>
        <div class="center">Center</div>
        <div class="right">
            <a href="#" class="link">Right 1</a>
            <a href="#" class="link">Right 2</a>
        </div>
    </div>
</div>

    <div class="pages navbar-through ">
        <div data-page="index" class="page">
            <div class="page-content">
                <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

                <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
            </div>
        </div>
    </div>
</div>
</div>
</div>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
</body>
</html>

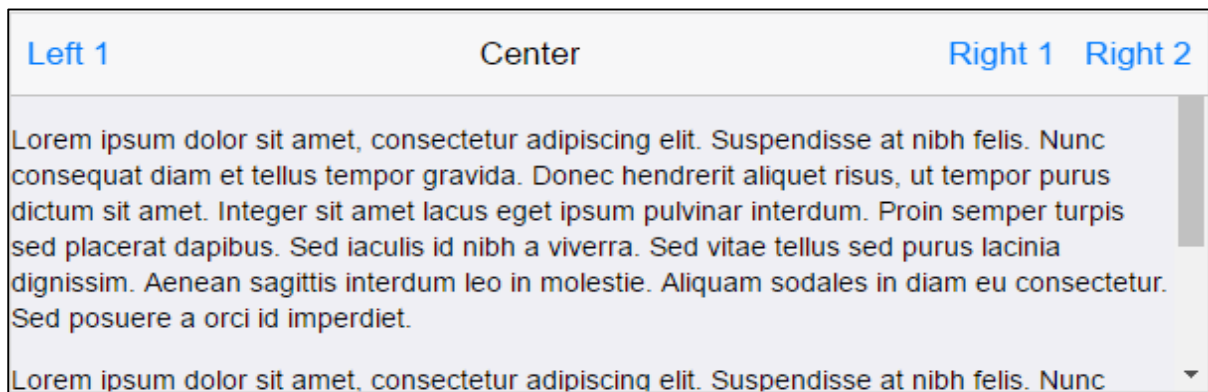
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code as **navbar_multi_links.html** file in your server root folder.
- Open this HTML file as http://localhost/navbar_multi_links.html and the output is displayed as shown below.

- The navbar provides multiple links either at the left part or right part of the page.



Framework7 – Links with Text & Icons

Description

The links can be provided with icons and text by adding classes for icons and wrapping the link text with `` element.

Example

The following example demonstrates the use of navbar links with text and icons in the Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Navbar Links With Text & Icons</title>
    <link
      href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css"
      rel="stylesheet"
    >
    <link
      href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css"
      rel="stylesheet"
    >
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
```



```

<div class="navbar-inner">
  <div class="left">
    <a href="#" class="link">
      <i class="icon icon-back"></i>
      <span>Previous</span>
    </a>
  </div>
  <div class="center">Center</div>
  <div class="right">
    <a href="#" class="link">
      <i class="icon icon-bars"></i>
      <span>Menu</span>
    </a>
  </div>
</div>
</div>
  <div class="pages navbar-through ">
    <div data-page="index" class="page">
      <div class="page-content">
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
      </div>
    </div>
  </div>
</div>
  <script
                                                                    type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
</body>

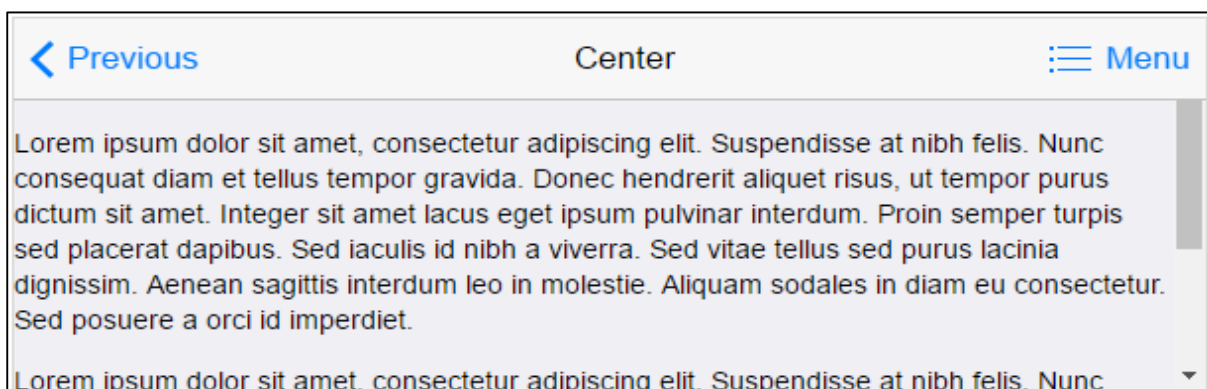
```

```
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code as **navbar_links_icons.html** file in your server root folder.
- Open this HTML file as http://localhost/navbar_links_icons.html and the output is displayed as shown below..
- The navbar provides links with the icons and text by wrapping the link's text with span element.



Framework7 – Links with Only Icons

Description

Navbar links can be provided with only icons by adding the *icon-only* class to links. It will make the link display in fixed 44x44 px size.

Example

The following example demonstrates the use of navbar links with only icons in the Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Navbar With Only Icons</title>
```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main">
            <div class="navbar">
                <div class="navbar-inner">
                    <div class="left">
                        <a href="#" class="link icon-only">
                            <i class="icon icon-back"></i>
                        </a>
                    </div>
                    <div class="center">Center Section</div>
                    <div class="right">
                        <a href="#" class="link icon-only">
                            <i class="icon icon-bars"></i>
                        </a>
                    </div>
                </div>
            </div>
            <div class="pages navbar-through ">
                <div data-page="index" class="page">
                    <div class="page-content">
                        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
                        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

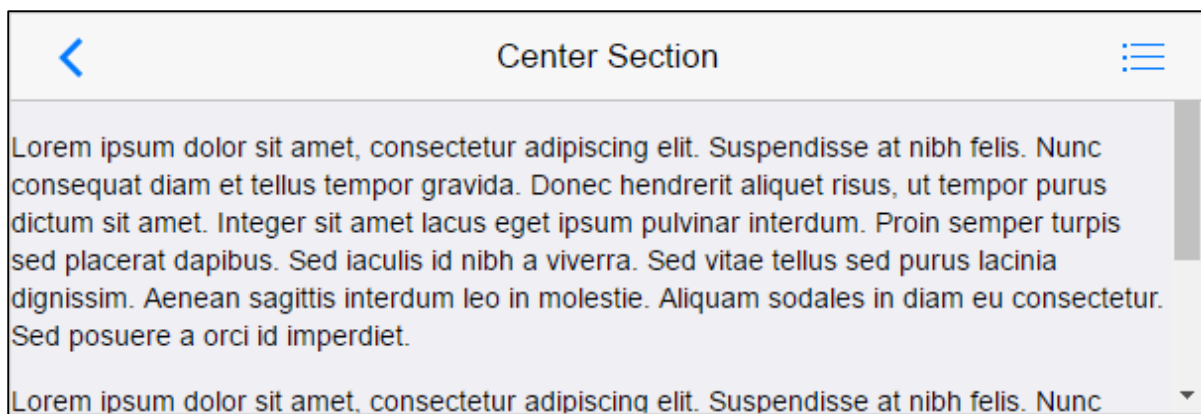
        </div>
    </div>
</div>
</div>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
    </body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **navbar_only_icons.html** file in your server root folder.
- Open this HTML file as http://localhost/navbar_only_icons.html and the output is displayed as shown below.
- The navbar provides links only with the icons without using the text.



Framework7 – Related App & View Methods

Description

The *View* (`<div class="view">`) is a disconnected visual section of app consisting of its own settings, navigations and history. Every view may have a different navbar and toolbar layouts.

Views is the wrapper container for all visual views except *modals* and *panels*. When *views* is set as a main wrapper for your app's body, only one *views* element is allowed. The following code shows the HTML structure for views:

```

<body>
    ...

```

```

<div class="panel panel-left panel-cover">
  <div class="view panel-view"> .... </div>
</div>
<!-- Views -->
<div class="views">
  <!-- Your main view -->
  <div class="view view-main">
    <!-- Navbar-->
    <!-- Pages -->
    <!-- Toolbar-->
  </div>
  <!-- Another view -->
  <div class="view another-view">
    <!-- Navbar-->
    <!-- Pages -->
    <!-- Toolbar-->
  </div>
</div>
<div class="popup">
  <div class="view popup-view"> .... </div>
</div>
....
</body>

```

On initializing the *View*, framework7 allows you to use the methods available for navbar as shown below:

S. No.	Methods & Description
1	myApp.hideNavbar(navbar) It is used to hide a particular navbar. The <i>navbar</i> can be HTML Element or string (with CSS selector) of required navbar and is a required parameter.
2	myApp.showNavbar(navbar) It is used to show a particular navbar. The <i>navbar</i> can be HTML Element or string (with CSS selector) of required navbar and is a required parameter.
3	view.hideNavbar() It hides navbar in this view.
4	view.showNavbar() It shows navbar in this view.

5	<p>myApp.sizeNavbars(viewContainer)</p> <p>It is used to recalculate positional styles for Navbar. The <i>viewContainer</i> can be CSS selector string or HTML element and is used only by iOS theme.</p>
---	--

Example

The following example demonstrates the use of view methods in Framework7, which hides the navbar when you click the *Hide Navbar* (uses *.hide-navbar* class) button and shows the navbar by clicking on the *Show Navbar* (uses *.show-navbar* class) button.

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Navbar View Methods</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> <a href="#" class="link"> <i
class="icon icon-back"></i><span>Previous</span></a></div>
                <div class="center">Center Part</div>
                <div class="right"> <a href="#" class="link icon-only">
<i class="icon icon-bars"></i></a></div>
              </div>
            </div>
            <div class="page-content">
              <div class="content-block">
                <p> <a href="#" class="button hide-navbar">Hide
Navbar</a></p>

```

```

        <p> <a href="#" class="button show-navbar">Show
Navbar</a></p>

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

    </div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var $ = Framework7.$;
    var mainView = myApp.addView('.view-main');
    $('<div>').on('click', function () {
        mainView.hideNavbar();
    });
    $('<div>').on('click', function () {

```

```

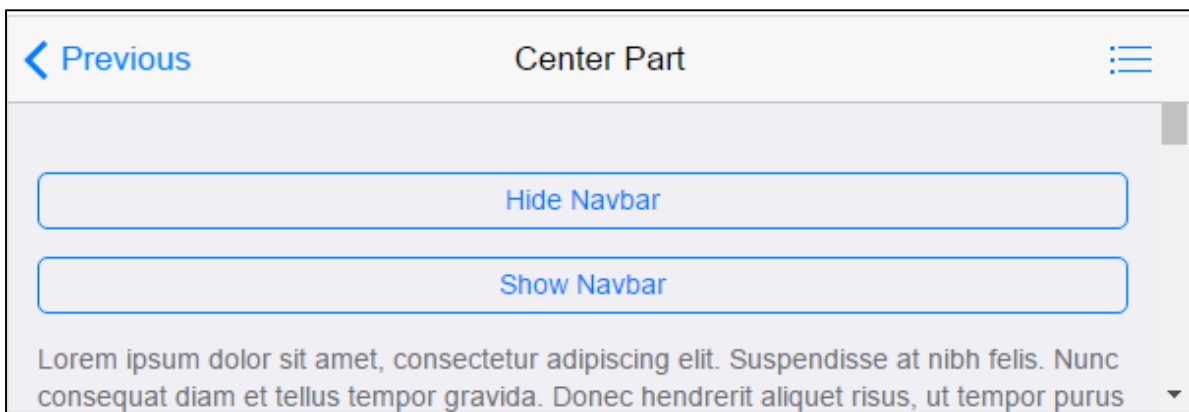
        mainView.showNavbar();
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code **navbar_app_view.html** file in your server root folder.
- Open this HTML file as http://localhost/navbar_app_view.html and the output as is displayed as shown below..
- When you run the code, you can hide the navbar by clicking on the Hide Navbar button and shows the navbar by clicking on the Show Navbar button.



Framework7 – Hide Navbar Automatically

Description

The navbar can be hidden/shown automatically for some Ajax loaded pages where navbar is not required. It is useful only whenever you use the through-type layout. You need to add *no-navbar* class to the loaded page and put an empty navbar.

Example

The following example hides the navbar automatically when you click on the link:

Create file `navbar_hide_auto_about.html`

navbar_hide_auto_about.html

```

<div class="navbar">
  <div class="navbar-inner"></div>

```



```

</div>
<div data-page="about" class="page no-navbar">
  <div class="page-content">
    <div class="content-block">
      <p> <a href="#" class="back button">Go Back</a></p>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec
hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit amet
lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat dapibus.
Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia dignissim.
Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec
hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit amet
lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat dapibus.
Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia dignissim.
Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec
hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit amet
lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat dapibus.
Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia dignissim.
Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec
hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit amet
lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat dapibus.
Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia dignissim.
Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
    </div>
  </div>
</div>

```

navbar_hide_auto.html

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Navbar View Methods</title>

```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main navbar-through">
            <div class="navbar">
                <div class="navbar-inner">
                    <div class="center">Center Part</div>
                    <div class="right"> <a href="#" class="link icon-only"> <i
class="icon icon-bars"></i></a></div>
                </div>
            </div>
            <div class="pages">
                <div data-page="home" class="page">
                    <div class="page-content">
                        <div class="content-block">
                            <p> <a href="navbar_hide_auto_about.html"
class="button">About</a></p>
                            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
                            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
                            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
                            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit

```

```

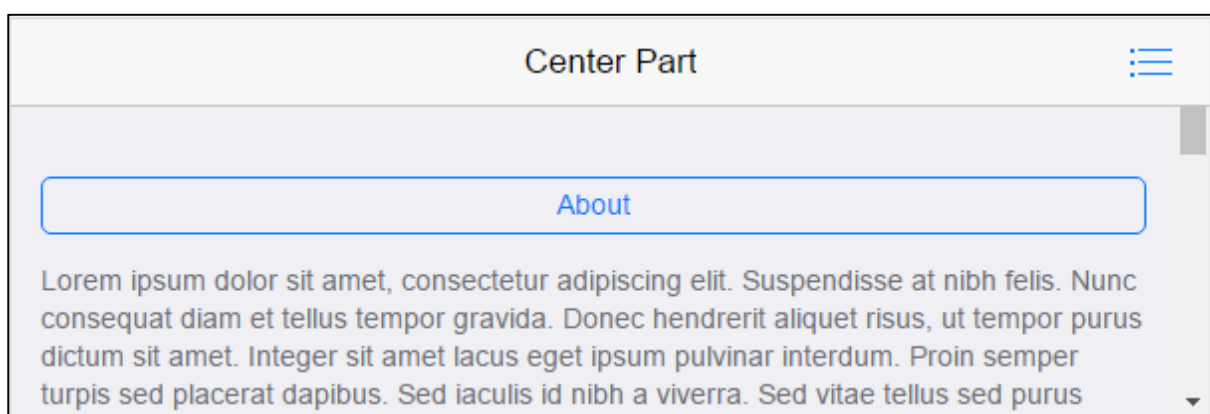
    amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
    dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
    dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
    consectetur. Sed posuere a orci id imperdiet.</p>
        </div>
    </div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var mainView = myApp.addView('.view-main', {
        dynamicNavbar: true
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code as **navbar_hide_auto.html** file in your server root folder.
- Open this HTML file as http://localhost/navbar_hide_auto.html and the output is displayed as shown below.
- When you run the code, you can hide or show the navbar automatically whenever navbar is not required.



5. Framework7 – Toolbars

Description

Toolbar provides easy access to other pages by using navigation elements at the bottom of the screen.

You can use toolbar in two ways as specified in the table:

S. No.	Toolbar types & Description
1	Hide Toolbar You can hide the toolbar automatically when you load the pages by using the no-toolbar class to loaded page.
2	Bottom Toolbar Place the toolbar at the bottom of the page by using the toolbar-bottom class.

Framework7 – Toolbar Hiding

Description

You can hide the toolbar automatically when you load the pages by using the **no-toolbar** class to loaded page.

Example

The following example demonstrates the use of toolbar hiding in Framework7:

First, we will create one HTML page called **toolbar_hiding.html**.

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Toolbar Hiding</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.i
s.colors.min.css">
</head>
<body>
  <div class="views">
    <div class="view view-main">
      <div class="navbar">
        <div class="navbar-inner">
          <div class="center sliding">Toolbar Hiding</div>
        </div>
      </div>
      <div class="pages navbar-through">
        <div data-page="index" class="page">
          <div class="page-content">
            <div class="list-block">
              <ul>
                <li>
                  <a href="/framework7/src/toolbar-services.html"
class="link">
                    <div class="item-content">
                      <div class="item-inner">
                        <div class="item-title">Go to Services
page</div>
                      </div>
                    </div>
                  </a>
                </li>
              </ul>
            </div>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Aenean sagittis
interdum leo in molestie. Aliquam sodales in diam eu consectetur. Sed posuere a
orci id imperdiet.</p>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit
amet lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat
dapibus. Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia
dignissim. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>
          </div>

```

```

        </div>
    </div>
    <div class="toolbar">
        <div class="toolbar-inner">
            <a href="#" class="link">Link 1</a>
            <a href="#" class="link">Link 2</a>
            <a href="#" class="link">Link 3</a>
        </div>
    </div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // here initialize the app
    var myApp = new Framework7();

    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;

    // Add the view
    var mainView = myApp.addView('.view-main', {
        // enable the dynamic navbar for this view
        dynamicNavbar: true
    });
</script>
</body>
</html>

```

Next, create one more HTML page i.e. **toolbar-services.html** as shown below:

```

<div class="navbar">
    <div class="navbar-inner">
        <div class="center sliding">Services</div>
    </div>
</div>
<div class="pages">

```

```

<div data-page="services" class="page no-toolbar">
  <div class="page-content">
    <div class="content-block">
      <br>
      <h2>Our Services</h2>
      <ul>
        <li>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed
do eiusmod tempor incididunt ut labore.</li>
        <li>Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris.</li>
        <li>Duis aute irure dolor in reprehenderit in voluptate velit
esse cillum dolore eu fugiat nulla pariatur.</li>
        <li>Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut
odit aut fugit.</li>
        <li>At vero eos et accusamus et iusto odio dignissimos ducimus
qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores
et.</li>
        <li>Nam libero tempore, cum soluta nobis est eligendi optio
cumque nihil impedit quo minus id.</li>
      </ul>
      <p>Go <a href="#" class="back">back</a> to home page.</p>
    </div>
  </div>
</div>
<div class="toolbar">
  <div class="toolbar-inner">
    <a href="#" class="link">Link 1</a>
    <a href="#" class="link">Link 2</a>
    <a href="#" class="link">Link 3</a>
  </div>
</div>

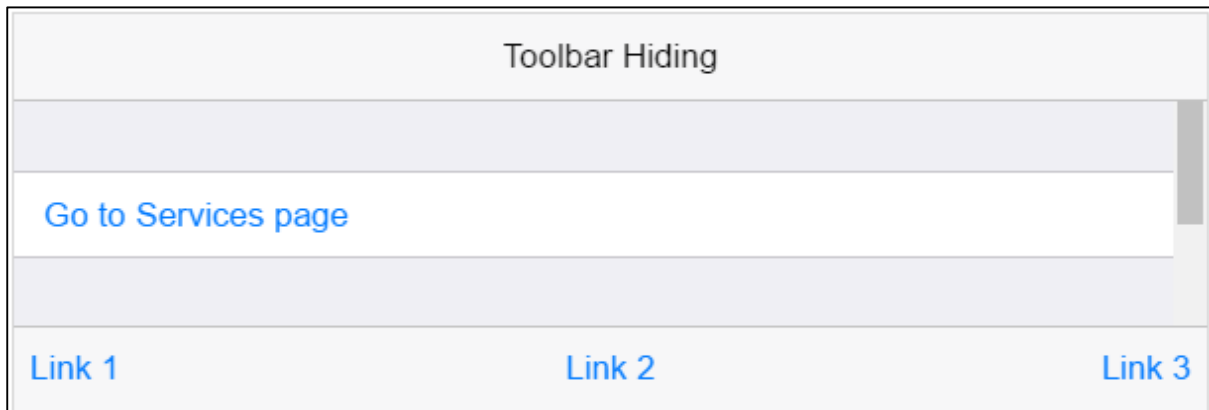
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code as **toolbar_hiding.html** file in your server root folder.

- Open this HTML file as http://localhost/toolbar_hiding.html and the output is displayed as shown below.
- When you run the code, click on the Services page link, which will hide the Toolbar automatically in the page.



Framework7 – Toolbar Bottom

Description

Place the toolbar at the bottom of the page by using the **toolbar-bottom** class. Bottom toolbar is only supported in Material theme and in iOS theme; it is placed at the bottom by default.

Example

The following example demonstrates the use of Toolbar bottom in Framework7:

First, we will create one HTML page called **toolbar_bottom.html**.

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Toolbar Hiding</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
```



```

<body>
  <div class="views">
    <div class="view view-main">
      <div class="navbar">
        <div class="navbar-inner">
          <div class="center sliding">Bottom Toolbar</div>
        </div>
      </div>
      <div class="pages navbar-through">
        <div data-page="index" class="page toolbar-fixed navbar-fixed">
          <div class="page-content">
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec
hendrerit aliquet risus, ut tempor purus dictum sit amet. Aenean sagittis
interdum leo in molestie. Aliquam sodales in diam eu consectetur. Sed posuere a
orci id imperdiet.</p>

            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec
hendrerit aliquet risus, ut tempor purus dictum sit amet. Integer sit amet
lacus eget ipsum pulvinar interdum. Proin semper turpis sed placerat dapibus.
Sed iaculis id nibh a viverra. Sed vitae tellus sed purus lacinia dignissim.
Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu
consectetur. Sed posuere a orci id imperdiet.</p>

          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="toolbar toolbar-bottom">
    <div class="toolbar-inner">
      <a href="#" class="link">Link 1</a>
      <a href="#" class="link">Link 2</a>
      <a href="#" class="link">Link 3</a>
    </div>
  </div>
  </div>
  </div>
  <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
  <script>
  // here initialize the app
  var myApp = new Framework7();
  // If your using custom DOM library, then save it to $$ variable
  var $$ = Dom7;
  // Add the view
  var mainView = myApp.addView('.view-main', {

```

```

    // enable the dynamic navbar for this view
    dynamicNavbar: true
  });
</script>
</body>
</html>

```

Next, create one more HTML page i.e. **toolbar-services.html** as shown below:

```

<div class="navbar">
  <div class="navbar-inner">
    <div class="center sliding">Services</div>
  </div>
</div>
<div class="pages">
  <div data-page="services" class="page no-toolbar">
    <div class="page-content">
      <div class="content-block">
        <br>
        <h2>Our Services</h2>
        <ul>
          <li>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed
do eiusmod tempor incididunt ut labore.</li>
          <li>Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris.</li>
          <li>Duis aute irure dolor in reprehenderit in voluptate velit
esse cillum dolore eu fugiat nulla pariatur.</li>
          <li>Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut
odit aut fugit.</li>
          <li>At vero eos et accusamus et iusto odio dignissimos ducimus
qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores
et.</li>
          <li>Nam libero tempore, cum soluta nobis est eligendi optio
cumque nihil impedit quo minus id.</li>
        </ul>
        <p>Go <a href="#" class="back">back</a> to home page.</p>
      </div>
    </div>
  </div>
</div>
<div class="toolbar">
  <div class="toolbar-inner">
    <a href="#" class="link">Link 1</a>

```

```

    <a href="#" class="link">Link 2</a>
    <a href="#" class="link">Link 3</a>
  </div>
</div>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **toolbar_bottom.html** file in your server root folder.
- Open this HTML file as http://localhost/toolbar_bottom.html and the output is displayed as shown below.
- In this example, place the toolbar at the bottom of the page.

Bottom Toolbar		
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu consectetur. Sed posuere a orci id imperdiet.</p>		
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse at nibh felis. Nunc</p>		
Link 1	Link 2	Link 3

Methods of Toolbar

The following available methods can be used with Toolbars:

S. No.	Toolbar Methods & Description
1	myApp.hideToolbar(toolbar) It hides the specified toolbar.
2	myApp.showToolbar(toolbar) It shows the specified toolbar.
3	view.hideToolbar() It hides the specified toolbar in the view.
4	view.showToolbar() It shows the specified toolbar in the view.

Example

The following example demonstrates the use of toolbar layout in the Framework7.

First, we will create one HTML page called **toolbar.html** as shown below:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Toolbar Layout</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="center sliding">Toolbar Layout</div>
          </div>
        </div>
        <div class="pages navbar-through">
          <div data-page="index" class="page with-subnavbar">
            <div class="page-content">
              <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Aenean sagittis
interdum leo in molestie. Aliquam sodales in diam eu consectetur. Sed posuere a
orci id imperdiet.</p>
              <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida.
Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Aenean sagittis
interdum leo in molestie. Aliquam sodales in diam eu consectetur. Sed posuere a
orci id imperdiet.</p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

        <div class="toolbar">
            <div class="toolbar-inner">
                <a href="#" class="link">Link 1</a>
                <a href="#" class="link">Link 2</a>
                <a href="#" class="link">Link 3</a>
            </div>
        </div>
    </div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // here initialize the app
    var myApp = new Framework7();

    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;

    // Add the view
    var mainView = myApp.addView('.view-main', {
        // enable the dynamic navbar for this view
        dynamicNavbar: true
    });
</script>
</body>
</html>

```

Now, initialize your app and views in the custom JS file **toolbar.js**.

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **toolbar.html** file in your server root folder.
- Open this HTML file as <http://localhost/toolbar.html> and the output is displayed as shown below.

Toolbar Layout		
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse at nibh felis. Nunc consequat diam et tellus tempor gravida. Donec hendrerit aliquet risus, ut tempor purus dictum sit amet. Aenean sagittis interdum leo in molestie. Aliquam sodales in diam eu consectetur. Sed posuere a orci id imperdiet.		
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse at nibh felis. Nunc		
Link 1	Link 2	Link 3

6. Framework7 – Search Bar

Description

Framework 7 allows searching the elements by using the *searchbar* class.

Search Bar Parameters

S. No.	Parameters & Description	Type	Default
1	searchList It searches the CSS selector or HTML element of the list.	string or HTML Element	-
2	searchIn You can search the list view elements of CSS selectors, also searches the elements by passing <i>.item-title</i> , <i>.item-text</i> classes.	string	' <i>.item-title</i> '
3	found It searches the CSS selector or HTML element using "found" element. Further, it uses <i>.searchbar-found</i> element if there is no element specified.	string or HTML Element	-
4	notfound It searches the CSS selector or HTML element using "not-found" element. Further, it uses <i>.searchbar-not-found</i> element if there is no element specified.	string or HTML Element	-
5	overlay It searches the CSS selector or HTML element using "searchbar overlay" element and uses <i>.searchbar-overlay</i> element, if there is no element specified.	string or HTML Element	-
6	ignore You can ignore the CSS selector for items by using the searchbar.	string	' <i>.searchbar-ignore</i> '
7	customSearch When it is enabled, the searchbar will not search through any of list blocks which is specified by <i>searchList</i> and you will be allowed to use custom search functionality.	boolean	false
8	removeDiacritics When searching for an element, remove the diacritics by enabling this parameter.	boolean	false

9	hideDividers This parameter will hide item dividers and group title, if there are no items.	boolean	true
10	hideGroups This parameter will hide the groups, if there are no items found in the list view groups.	boolean	true

Search Bar Callbacks

S. No.	Callbacks & Description	Type	Default
1	onSearch This method will fire callback function while doing search.	function (s)	-
2	onEnable This method will fire callback function when Search Bar becomes active.	function (s)	-
3	onDisable This method will fire callback function when Search Bar becomes inactive.	function (s)	-
4	onClear This method will fire callback function when you click on the "clear" element.	function (s)	-

Search Bar Properties

S. No.	Properties & Description
1	mySearchbar.params Represents the initialized parameters passed with object.
2	mySearchbar.query Searches the current query.
3	mySearchbar.searchList Defines the search list block.
4	mySearchbar.container Defines the search bar container with HTML element.
5	mySearchbar.input Defines the search bar input with HTML element.
6	mySearchbar.active It defines whether search bar is enabled or disabled.

Search Bar Methods

S. No.	Methods & Description
1	mySearchbar.search(query); This method searches the passed query.
2	mySearchbar.enable(); It enables the search bar.
3	mySearchbar.disable(); It disables the search bar.
4	mySearchbar.clear(); You can clear the query and search results.
5	mySearchbar.destroy(); It destroys the search bar instance.

Search Bar JavaScript Events

S. No.	Event & Description	Target
1	search You can fire this event while searching elements.	<div class="list-block">
2	clearSearch This event will get fired when user clicks on the clearSearch element.	<div class="list-block">
3	enableSearch When Search Bar becomes enable, this event will get fired.	<div class="list-block">
4	disableSearch When Search Bar gets disabled, and user clicks on cancel button, or "search bar-overlay" element, this event will get fired.	<div class="list-block">

Example

The following example demonstrates the use of search bar on scroll in the Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
```

```

    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Search Bar Layout</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main">
            <div class="pages navbar-fixed">
                <div data-page="home" class="page">
                    <div class="navbar">
                        <div class="navbar-inner">
                            <div class="center sliding">Search Bar Layout</div>
                        </div>
                    </div>
                    <form data-search-list=".list-block-search" data-search-
in=".item-title" class="searchbar searchbar-init">
                        <div class="searchbar-input">
                            <input type="search" placeholder="Search"><a href="#"
class="searchbar-clear"></a>
                            </div>
                            <a href="#" class="searchbar-cancel">Cancel</a>
                        </form>
                    <div class="searchbar-overlay"></div>
                    <div class="page-content">
                        <div class="content-block searchbar-not-found">
                            <div class="content-block-inner">No element
found...</div>
                        </div>
                        <div class="list-block list-block-search searchbar-found">
                            <ul>
                                <li class="item-content">
                                    <div class="item-inner">
                                        <div class="item-title">India</div>
                                    </div>
                                </li>
                                <li class="item-content">

```

```
<div class="item-inner">
  <div class="item-title">Argentina</div>
</div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Belgium</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Brazil</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Canada</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Colombia</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Denmark</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Ecuador</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">France</div>
  </div>
</li>
```

```
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Germany</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Greece</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Haiti</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Hong Kong</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Iceland</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Ireland</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Jamaica</div>
        </div>
    </li>
    <li class="item-content">
```

```
<div class="item-inner">
  <div class="item-title">Japan</div>
</div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Kenya</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Kuwait</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Libya</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Liberia</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Malaysia</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Mauritius</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Mexico</div>
  </div>
</li>
```

```
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Namibia</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">New Zealand</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Oman</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Paraguay</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Philippines</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Russia</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Singapore</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">South Africa</div>
  </div>
</li>
```

```

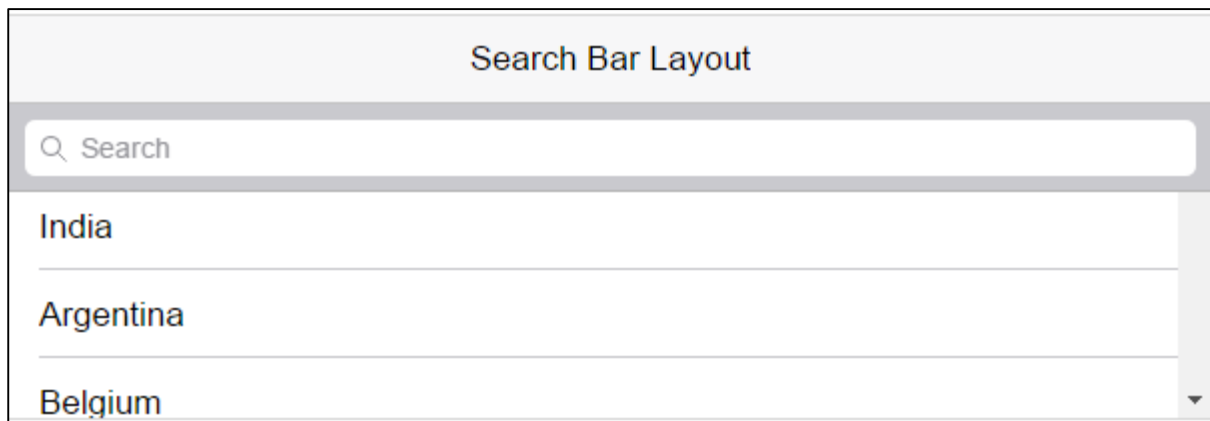
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Thailand</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">United Kingdom</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Vatican City</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Zimbabwe</div>
        </div>
    </li>
</ul>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var mainView = myApp.addView('.view-main');
</script>
</body>
</html>

```

Output

Let's carry out the following steps to see how above code works:

- Save the above given HTML code as **search_bar.html** file in your server root folder.
- Open this HTML file as http://localhost/search_bar.html and the output is displayed as shown below.
- If the element contained in the list is entered in the search bar, it displays that particular element from the list.
- If the element other than the elements contained in the list is entered, it displays no element found.



7. Framework7 – Status Bar

Description

The iOS 7+ allows you to build full screen apps, which could create an issue when your status bar overlaps your app. Framework7 solves this problem by detecting whether your app is in full screen mode or not. If your app is in full screen mode then, the Framework7 will automatically adds *with-statusbar-overlay* class to *<html>* (or removes if app is not in full screen mode) and you need to add *statusbar-overlay* class in *<body>* as shown in the following code:

```
<html class="with-statusbar-overlay">
...
<body>
  <div class="statusbar-overlay"></div>
  ...
```

By default, *<div>* will always be hidden and fixed on top of your screen. It will only be visible when the app is in full screen mode and *with-statusbar-overlay* class is added to *<html>*.

Example

The following example demonstrates the use of status bar in the Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>My App</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="statusbar-overlay"></div>
```

```

<div class="panel-overlay"></div>
<div class="panel panel-right panel-reveal">
  <div class="content-block">
    <p>Contents goes here...</p>
  </div>
</div>
<div class="views">
  <div class="view view-main">
    <div class="navbar">
      <div class="navbar-inner">
        <div class="center sliding">My App</div>
        <div class="right">
          <a href="#" class="link icon-only open-panel"><i
class="icon icon-bars"></i></a>
        </div>
      </div>
    </div>
    <div class="pages navbar-through toolbar-through">
      <div data-page="index" class="page">
        <div class="page-content">
          <p>This is simple application...</p>
          <p>page contents goes here!!!</p>
        </div>
      </div>
    </div>
    <div class="toolbar">
      <div class="toolbar-inner">
        <a href="#" class="link">First Link</a>
        <a href="#" class="link">Second Link</a>
      </div>
    </div>
  </div>
  <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
  <script>
    // here initialize the app

```

```

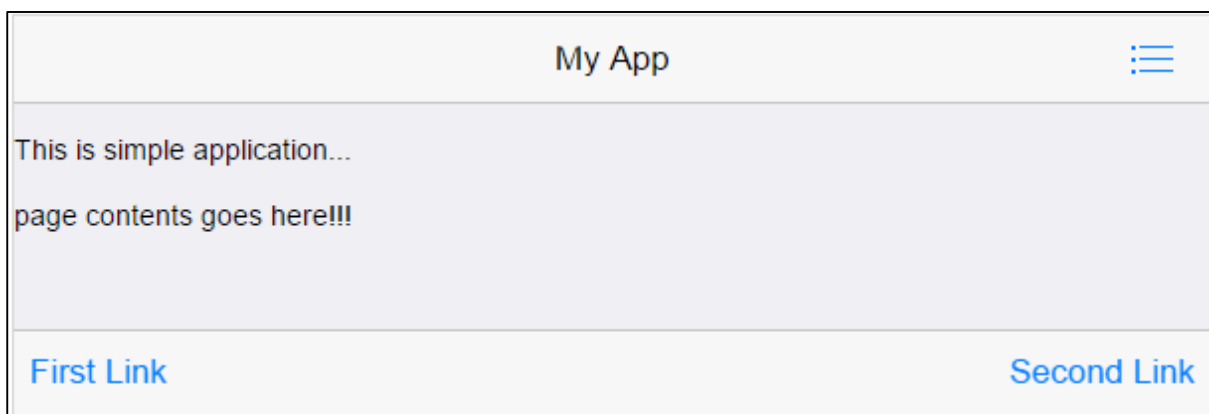
var myApp = new Framework7();
// If your using custom DOM library, then save it to $$ variable
var $$ = Dom7;
// Add the view
var mainView = myApp.addView('.view-main', {
  // enable the dynamic navbar for this view:
  dynamicNavbar: true
});
//use the 'pageInit' event handler for all pages
$$$(document).on('pageInit', function (e) {
  //get page data from event data
  var page = e.detail.page;
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how above given code works:

- Save the above given html code as **status_bar.html** file in your server root folder.
- Open this HTML file as http://localhost/status_bar.html and the output is displayed as shown below.



The example shows the use of the **statusbar-overlay**, which allows you to build full screen apps when your status bar overlaps the app.

8. Framework7 – Side Panels

Description

The *side panel* moves to the left or right side of the screen to display the content. Framework7 allows you to include up to 2 panels (right side panel and left side panel) to your app. You need to add panels in the beginning of the `<body>` and then choose the opening effect by applying the following listed classes:

- **panel-reveal** - This will make the whole app's content move out.
- **panel-cover** - This will make the panel to overlay on app's content.

For instance, the following code shows how to use the above classes:

```
<body>
  <!-- First add Panel's overlay which will overlays app while panel is
  opened -->
  <div class="panel-overlay"></div>
  <!-- Left panel -->
  <div class="panel panel-left panel-cover">
    panel's content
  </div>
  <!-- Right panel -->
  <div class="panel panel-right panel-reveal">
    panel's content
  </div>
</body>
```

The following table shows the panel types supported by Framework7:

S. No.	Type & Description
1	Open and Close Panels Once you add panel and effects, we need to add functionality to <i>open</i> and <i>close</i> the panels.
2	Panel Events To detect how a user interacts with the panel, you can use panel events.
3	Open Panels With Swipe Framework7 provides you the feature to open panel with <i>swipe gesture</i> .
4	Panel is Opened? We can determine whether panel is opened or not by using the with-panel-[position]-[effect] rule.

Framework7 – Open & Close Panels

Description

Once you add panel and effects, we need to add functionality to *open* and *close* the panels. The following table shows how can we *open* or *close* the panels:

S. No.	Type & Description
1	From HTML To open and close the panel, add <i>open-panel</i> and <i>close-panel</i> class respectively to any HTML element.
2	Using JavaScript Panels can be <i>opened</i> and <i>closed</i> with JavaScript by using the related app methods.

Framework7 – Open & Close Panels From HTML

Description

To open and close the panel, add *open-panel* and *close-panel* classes respectively to any HTML element. By default, the *left* panel will open/close when your app has two panels. You can specify which panel should open/close by adding the *data-panel="left"* attribute.

Example

The following example demonstrates the use of open and close panels in Framework7.

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Open and Close Panels</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="panel-overlay"></div>
```

```

<div class="panel panel-left panel-reveal">
  <div class="content-block">
    <p>Left Panel content</p>
    <p><a href="#" data-panel="right" class="open-panel">Open Right
Panel</a></p>
    <p><a href="#" class="close-panel">Close me</a></p>
  </div>
</div>
<div class="panel panel-right panel-cover">
  <div class="content-block">
    <p>Right Panel content</p>
    <p><a href="#" data-panel="left" class="open-panel">Open Left
Panel</a></p>
    <p><a href="#" class="close-panel">Close me</a></p>
  </div>
</div>
<div class="views">
  <div class="view view-main">
    <div class="pages">
      <div data-page="home" class="page navbar-fixed">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="left"> </div>
            <div class="center">Side Panels</div>
            <div class="right"> </div>
          </div>
        </div>
        <div class="page-content">
          <div class="content-block">
            <p> <a href="#" class="open-panel">Open Left Panel</a></p>
            <p> <a href="#" data-panel="right" class="open-
panel">Open Right Panel</a></p>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
</div>

```

```

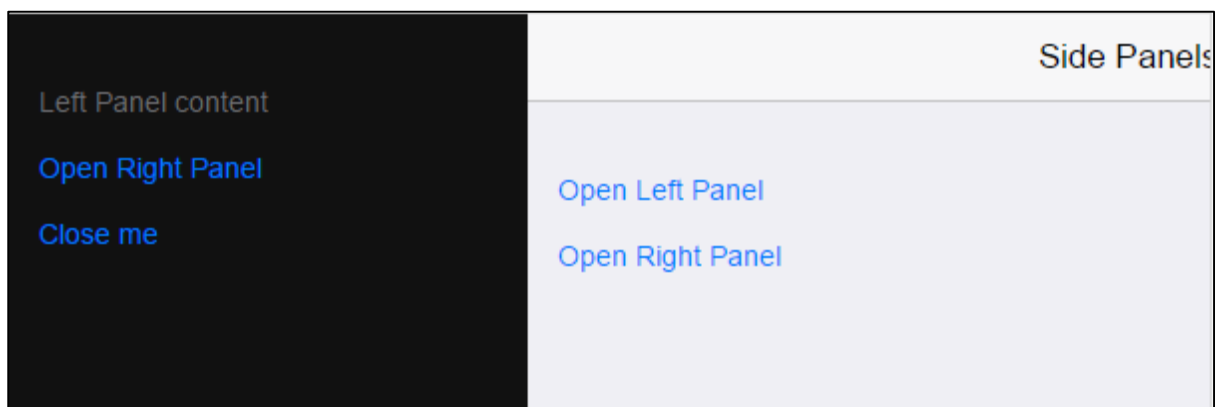
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
    <script>
        var myApp = new Framework7();
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **sidepanels_open_close_fromHTML.html** file in your server root folder.
- Open this HTML file as http://localhost/sidepanels_open_close_fromHTML.html and the output is displayed as shown below.
- This example displays the left panel when you click the *Open Left Panel* link and displays the right panel when you click the *Open Right Panel* link.



Framework7 – Open & Close Panels From JS

Description

Panels can be *opened* and *closed* with JavaScript by using the related app methods as shown below:

- **myApp.openPanel(position)** - It is used to open the panel which accepts the position of panel (left or right) as string.
- **myApp.closePanel()** - It closes the currently opened panel. The example displays the left panel when you click *Open Left Panel* and to display the right panel, click *Open Right Panel*.

Example

The following example demonstrates the use of open and close panels in Framework7:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Open and Close Panels</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="panel-overlay"></div>
    <div class="panel panel-left panel-reveal">
      <div class="content-block">
        <p>Left Panel content</p>
        <p><a href="#" class="open-right-panel">Open Right Panel</a></p>
        <p><a href="#" class="panel-close">Close me</a></p>
      </div>
    </div>
    <div class="panel panel-right panel-cover">
      <div class="content-block">
        <p>Right Panel content</p>
        <p><a href="#" class="open-left-panel">Open Left Panel</a></p>
        <p><a href="#" class="panel-close">Close me</a></p>
      </div>
    </div>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">

```



```

        <div class="left"> </div>
        <div class="center">Open Panels with JavaScript</div>
        <div class="right"> </div>
    </div>
</div>
<div class="page-content">
    <div class="content-block">
        <p> <a href="#" class="open-left-panel">Open Left
Panel</a></p>
        <p> <a href="#" class="open-right-panel">Open Right
Panel</a></p>
    </div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var $$ = Dom7;
    $$('.open-left-panel').on('click', function (e) {
        // 'left' position to open Left panel
        myApp.openPanel('left');
    });
    $$('.open-right-panel').on('click', function (e) {
        // 'right' position to open Right panel
        myApp.openPanel('right');
    });
    $$('.panel-close').on('click', function (e) {
        myApp.closePanel();
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **sidepanels_open_close_js.html** file in your server root folder.
- Open this HTML file as http://localhost/sidepanels_open_close_js.html and the output is displayed as shown below.
- This example displays the left panel when you click the *Open Left Panel* link and displays the right panel when you click the *Open Right Panel* link.



Framework7 – Panel Events

Description

To detect how a user interacts with the panel, you can use the panel events. It can also be used to do something in JS whenever panel is opened/closed.

The following table lists the events supported by Framework7:

S. No.	Event & Description	Target
1	open Whenever the panel begins its opening animation, this event will be triggered.	Panel Element <div class="panel">
2	opened Whenever the panel completes its opening animation, this event will be triggered.	Panel Element <div class="panel">
3	close Whenever the panel's closing animation is initiated, this event will be triggered.	Panel Element <div class="panel">
4	closed Whenever the panel's closing animation completes, this event will be triggered.	Panel Element <div class="panel">

Example

The following example demonstrates the panel events in the Framework7. The example detects the user interaction with the panel and displays the event messages accordingly.

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Panel Events</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="panel-overlay"></div>
    <div class="panel panel-left panel-reveal">
      <div class="content-block">
        <h2>Left Panel content</h2>
        <p><a href="#" data-panel="right" class="open-panel">Open Right
Panel</a></p>
        <p><a href="#" class="close-panel">Close me</a></p>
      </div>
    </div>
    <div class="panel panel-right panel-cover">
      <div class="content-block">
        <h2>Right Panel content</h2>
        <p><a href="#" data-panel="left" class="open-panel">Open Left
Panel</a></p>
        <p><a href="#" class="close-panel">Close me</a></p>
      </div>
    </div>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">

```

```

        <div class="navbar-inner">
            <div class="left"> </div>
            <div class="center">Side Panel Events</div>
            <div class="right"> </div>
        </div>
    </div>
    <div class="page-content">
        <div class="content-block">
            <p> <a href="#" class="open-panel">Open Left
Panel</a></p>
            <p> <a href="#" data-panel="right" class="open-
panel">Open Right Panel</a></p>
        </div>
    </div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();

    var $$ = Dom7;

    $$('.panel-left').on('opened', function () {
        myApp.alert('Left panel opened!!');
    });
    $$('.panel-left').on('close', function () {
        myApp.alert('Left panel is closing!!');
    });
    $$('.panel-right').on('open', function () {
        myApp.alert('Right panel is opening!!');
    });
    $$('.panel-right').on('closed', function () {
        myApp.alert('Right panel closed!!');
    });
</script>

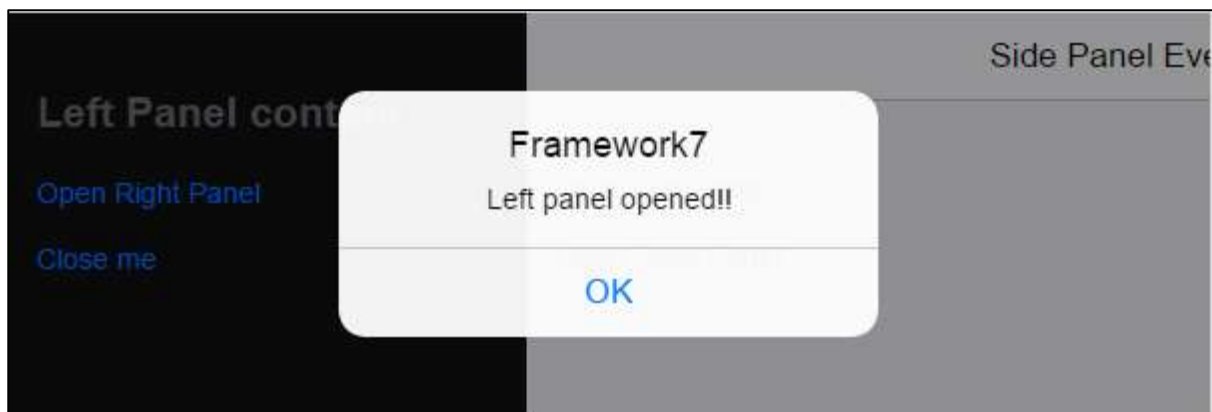
```

```
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **sidepanels_events.html** file in your server root folder.
- Open this HTML file as http://localhost/sidepanels_events.html and the output is displayed as shown below.
- This example displays the left panel when you click the *Open Left Panel* link and displays the right panel when you click the *Open Right Panel* link and the panel event will be triggered when the panel starts opening and closing animation.



Framework7 – Open Panels With Swipe

Description

Framework7 provides you the feature to open the panel with the *swipe gesture*. Framework7 limits you to use swipe with only one panel, that is, you can only apply a swipe to either the *left panel* or the *right panel*.

Example

The following example demonstrates the use of swipe panels in Framework7. The example allows you to open the left panel by swiping right.

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
```

```

    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Swipe Panel</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
    <div class="panel-overlay"></div>
    <div class="panel panel-left panel-reveal">
        <div class="content-block">
            <h2>Left Panel content</h2>
            <p><a href="#" class="close-panel">Close me</a></p>
            <p><a href="#" data-panel="right" class="open-panel">Open Right
Panel</a></p>
        </div>
    </div>
    <div class="panel panel-right panel-cover">
        <div class="content-block">
            <h2>Right Panel content</h2>
            <p><a href="#" class="close-panel">Close me</a></p>
            <p><a href="#" data-panel="left" class="open-panel">Open Left
Panel</a></p>
        </div>
    </div>
    <div class="views">
        <div class="view view-main">
            <div class="pages">
                <div data-page="home" class="page navbar-fixed">
                    <div class="navbar">
                        <div class="navbar-inner">
                            <div class="left"> </div>
                            <div class="center">Swipe Panel</div>
                            <div class="right"> </div>
                        </div>
                    </div>
                    <div class="page-content">

```

```

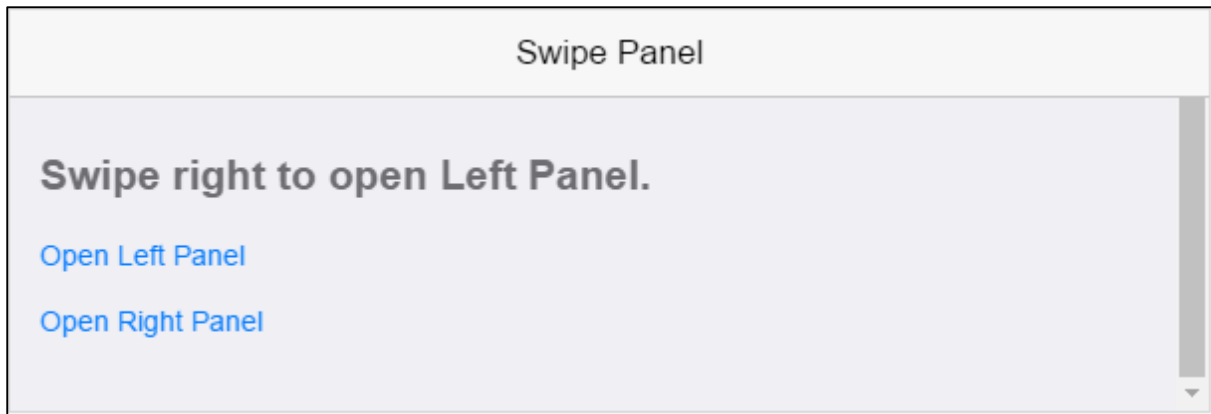
        <div class="content-block">
            <h2>Swipe right to open Left Panel.</h2>
            <p> <a href="#" class="open-panel">Open Left
Panel</a></p>
            <p> <a href="#" data-panel="right" class="open-
panel">Open Right Panel</a></p>
        </div>
    </div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7({
        swipePanel: 'left'
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **sidepanels_swipe.html** file in your server root folder.
- Open this HTML file as http://localhost/sidepanels_swipe.html and the output is displayed as shown below.
- This code allows you to open the left panel by swiping right using the swipe gesture.



Framework7 – Panel Is Opened?

Description

We can determine whether a panel is opened or not. Whenever some panel is opened, the `<body>` will have additional generated classes as shown below which uses the **with-panel-[position]-[effect]** rule.

- When you have left panel opened with cover effect, the *body* will have *with-panel-left-cover* class.
- When you have left panel opened with reveal effect, the *body* will have *with-panel-left-reveal* class.
- When you have right panel opened with cover effect, the *body* will have *with-panel-right-cover* class.
- When you have right panel opened with reveal effect, the *body* will have *with-panel-right-reveal* class.

You can use it in JavaScript or in CSS to detect an opened panel as shown below:

In JavaScript

```
if ($$('body').hasClass('with-panel-left-cover')) {
    console.log('Left Panel is opened')
}
```

In CSS

```
body.with-panel-left-cover .statusbar-overlay{
    background-color: #333;
}
```

This will change the status bar background when the panel is opened.

9. Framework7 – Content Block

Description

Content blocks can be used to add extra content with different format.

Example

The following example demonstrates the use of content block in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Content Block</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Content Block</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">
              <p>This is out side of content block!!!</p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

        <div class="content-block">
            <p>Sed ut perspiciatis unde omnis iste natus error sit
            voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa
            quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt
            explicabo.</p>
        </div>
        <div class="content-block">
            <div class="content-block-inner">This is another text
            block wrapped with "content-block-inner"</div>
        </div>
        <div class="content-block-title">Content Block Title</div>
        <div class="content-block">
            <p>Praesent nec imperdiet diam. Maecenas vel lectus
            porttitor, consectetur magna nec, viverra sem. Aliquam sed risus dolor. Morbi
            tincidunt ut libero id sodales. Integer blandit varius nisi quis
            consectetur.</p>
        </div>
        <div class="content-block-title">This is another long
        content block title</div>
        <div class="content-block">
            <div class="content-block-inner">
                <p>Sed ut perspiciatis unde omnis iste natus error
                sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa
                quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt
                explicabo.</p>
            </div>
        </div>
        <div class="content-block-title">Content Block Inset</div>
        <div class="content-block inset">
            <div class="content-block-inner">
                <p>Donec et nulla auctor massa pharetra adipiscing
                ut sit amet sem. Suspendisse molestie velit vitae mattis tincidunt. Ut sit amet
                quam mollis, vulputate turpis vel, sagittis felis. </p>
            </div>
        </div>
        <div class="content-block-title">Content Block Tablet
        Inset</div>
        <div class="content-block tablet-inset">
            <div class="content-block-inner">
                <p>Donec et nulla auctor massa pharetra adipiscing
                ut sit amet sem. Suspendisse molestie velit vitae mattis tincidunt. Ut sit amet
                quam mollis, vulputate turpis vel, sagittis felis. </p>
            </div>
        </div>

```

```

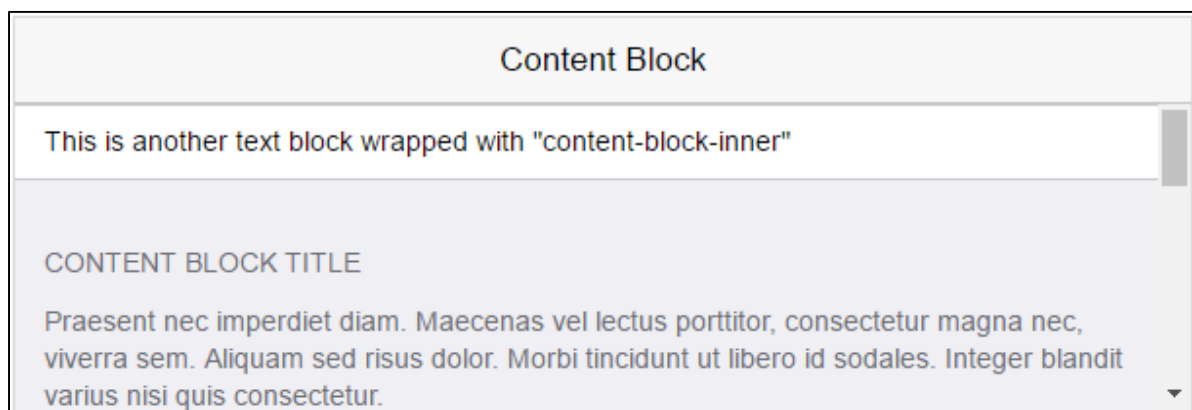
        </div>
    </div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var mainView = myApp.addView('.view-main');
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **content_block.html** file in your server root folder.
- Open this HTML file as http://localhost/content_block.html and the output is displayed as shown below.
- The code adds extra formatting and required spacing for the text content.



10. Framework7 – Layout Grid

Description

Framework7 provides different types of grid types to place the content as per user need.

The layout grid provides different types of columns size as described in the following table:

S. No.	Class	Table Class	Width
1	col-5	tablet-5	5%
2	col-10	tablet-10	10%
3	col-15	tablet-15	15%
4	col-20	tablet-20	20%
5	col-25	tablet-25	25%
6	col-30	tablet-30	30%
7	col-33	tablet-33	33.3%
8	col-35	tablet-35	35%
9	col-40	tablet-40	40%
10	col-45	tablet-45	45%
11	col-50	tablet-50	50%
12	col-55	tablet-55	55%
13	col-60	tablet-60	60%
14	col-65	tablet-65	65%
15	col-66	tablet-66	66.6%
16	col-70	tablet-70	70%
17	col-75	tablet-75	75%
18	col-80	tablet-80	80%
19	col-85	tablet-85	85%
20	col-90	tablet-90	90%

21	col-95	tablet-95	95%
21	col-100	tablet-100	100%
22	col-auto	tablet-auto	Equal width

Example

The following example provides the grid layout for placing your content as you need in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Layout Grid</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
    <style>
      div[class*="col-"] {
        background: #fff;
        text-align: center;
        color: #000;
        border: 1px solid #D8D8D8;
      }
      .row {
        margin-bottom: 10px;
      }
    </style>
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
```

```

        <div class="center sliding">Layout Grid</div>
    </div>
</div>
<div class="pages">
    <div data-page="index" class="page navbar-fixed">
        <div class="page-content">
            <div class="content-block-title">Columns with gutter</div>
            <div class="content-block">
                <div class="row">
                    <div class="col-50">col 1</div>
                    <div class="col-50">col 2</div>
                </div>
                <div class="row">
                    <div class="col-25">col 1</div>
                    <div class="col-25">col 2</div>
                    <div class="col-25">col 3</div>
                    <div class="col-25">col 4</div>
                </div>
                <div class="row">
                    <div class="col-33">col 1</div>
                    <div class="col-33">col 2</div>
                    <div class="col-33">col 3</div>
                </div>
            </div>
            <div class="content-block-title">Columns without gutter</div>
            <div class="content-block">
                <div class="row no-gutter">
                    <div class="col-50">col 1</div>
                    <div class="col-50">col 2</div>
                </div>
                <div class="row no-gutter">
                    <div class="col-25">col 1</div>
                    <div class="col-25">col 2</div>
                    <div class="col-25">col 3</div>
                    <div class="col-25">col 4</div>
                </div>
                <div class="row no-gutter">

```

```

        <div class="col-33">col 1</div>
        <div class="col-33">col 2</div>
        <div class="col-33">col 3</div>
    </div>
</div>
<div class="content-block-title">Nested Columns</div>
<div class="content-block">
    <div class="row">
        <div class="col-50">
            col 1
            <div class="row">
                <div class="col-40">col 2</div>
                <div class="col-60">col 3</div>
            </div>
        </div>
        <div class="col-50">
            col 1
            <div class="row">
                <div class="col-75">col 2</div>
                <div class="col-25">col 3</div>
            </div>
        </div>
    </div>
</div>
<div class="content-block-title">Columns With Different
Equal Width</div>
<div class="content-block">
    <div class="row">
        <div class="col-100 tablet-50">col 1</div>
        <div class="col-100 tablet-50">col 2</div>
    </div>
    <div class="row">
        <div class="col-50 tablet-25">col 1</div>
        <div class="col-50 tablet-25">col 2</div>
        <div class="col-50 tablet-25">col 3</div>
        <div class="col-50 tablet-25">col 4</div>
    </div>
    <div class="row">

```

```

                <div class="col-100 tablet-40">col 1</div>
                <div class="col-50 tablet-60">col 2</div>
                <div class="col-50 tablet-60">col 3</div>
                <div class="col-100 tablet-40">col 4</div>
            </div>
        </div>
<div class="content-block-title">Columns With Equal
Width</div>

<div class="content-block">
    <div class="row">
        <div class="col-auto">col-1</div>
        <div class="col-auto">col-2</div>
        <div class="col-auto">col-3</div>
        <div class="col-auto">col-4</div>
    </div>
    <div class="row no-gutter">
        <div class="col-auto">col-1</div>
        <div class="col-auto">col-2</div>
        <div class="col-auto">col-3</div>
        <div class="col-auto">col-4</div>
    </div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var mainView = myApp.addView('.view-main');
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **layout_grid.html** file in your server root folder.
- Open this HTML file as http://localhost/layout_grid.html the output is displayed as shown below.
- This code provides different types of grid types to place the content as per the user need.

Layout Grid

COLUMNS WITH GUTTER

col 1	col 2		
col 1	col 2	col 3	col 4
col 1	col 2	col 3	

COLUMNS WITHOUT GUTTER

COLUMNS WITHOUT GUTTER

col 1	col 2		
col 1	col 2		
col 1	col 2	col 3	col 4
col 1	col 2	col 3	

NESTED COLUMNS

col 1	col 1		
col 2	col 3	col 2	col 3

COLUMNS WITH DIFFERENT EQUAL WIDTH

col 1	
col 2	
col 1	col 2
col 3	col 4
col 1	
col 2	col 3
col 4	

COLUMNS WITH EQUAL WIDTH

COLUMNS WITH EQUAL WIDTH

col-1	col-2	col-3	col-4
col-1	col-2	col-3	col-4

11. Framework7 – Overlays

Description

Framework7 provides different types of overlays to work with the applications smoothly. The following table lists some of the Framework7 overlays:

S. No.	Overlay Type & Description
1	Modal Modal is a small window that display the content from separate sources without leaving the parent window.
2	Popup Popup is a popup box which displays the content when the user clicks on the element.
3	Popover To manage the presentation of temporary content, the popover component can be used.
4	Action Sheet The <i>Action Sheet</i> is used to present the user with a set of possibilities for how to handle a given task.
5	Login Screen Overlay login screen is used for displaying the login screen format which can be used in page or popup or as a standalone overlay.
6	Picker Modal Picker modal is used to pick some custom content which is similar to the calender picker.

Framework7 – Overlay Modal

Description

Modal is a small window that displays the content from a separate source without leaving the parent window.

There are different types of modals available in Framework7 as specified in the following table:

S. No.	Modal Type & Description
1	Alert It displays the alert modal.

2	Confirm Confirm modal is used to confirm some action for the displayed content.
3	Prompt Prompt modal allows the users to take some actions.
4	Login and Password This type of modal is used for authentication.
5	Nesting Modals You can use one modal inside another modal.
6	Preloader Modal Preloader modal defines the background action and stops the user actions during this action.
7	Indicator Indicator modal specifies small indication activity like preloader modal for the user actions.

Framework7 – Alert Modal

Description

You can display the alert modal by using the following methods:

```
myApp.alert(text, [title, callbackOk])
```

Or

```
myApp.alert(text, [callbackOk])
```

The above methods accept parameters, which are listed below:

- **text:** It displays the alert with the text.
- **title:** It is an optional method that displays alert with title.
- **callbackOk:** It is an optional method, which provides callback function that executes when the "OK" button is clicked by the user on alert modal.

Example

The following example demonstrates the use of alert modal in Framework7, which displays the alert box when you click on the links:

```
<!DOCTYPE html>
<html>
  <head>
```

```

    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Alert Modal</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main">
            <div class="navbar">
                <div class="navbar-inner">
                    <div class="center sliding">Alert Modal</div>
                </div>
            </div>
            <div class="pages">
                <div data-page="index" class="page navbar-fixed">
                    <div class="page-content">
                        <div class="content-block">
                            <p><a href="#" class="alert-text">Displays Alert Modal
with Text</a></p>
                            <p><a href="#" class="alert-text-title">Displays Alert
Modal With Text and Title</a></p>
                            <p><a href="#" class="alert-text-title-
callback">Displays Alert Modal With Text and Title and Callback</a></p>
                            <p><a href="#" class="alert-text-callback">Displays
Alert Modal With Text and Callback</a></p>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>

```

```

<script>
  // Here you can initialize the app
  var myApp = new Framework7();

  // If your using custom DOM library, then save it to $$ variable
  var $$ = Dom7;
  // Add the view
  var mainView = myApp.addView('.view-main', {
  // enable the dynamic navbar for this view:
  dynamicNavbar: true
  });
  $$('.alert-text').on('click', function () {
  myApp.alert('This is alert text!!!');
  });
  $$('.alert-text-title').on('click', function () {
  myApp.alert('This is alert text!!!', 'My Title!');
  });
  $$('.alert-text-title-callback').on('click', function () {
  myApp.alert('This is alert text!!!', 'My Title!', function () {
    myApp.alert('You have clicked the button!!!')
  });
  });
  $$('.alert-text-callback').on('click', function () {
  myApp.alert('This is alert text!!!', function () {
    myApp.alert('You have clicked the button!!!')
  });
  });
</script>
</body>
</html>

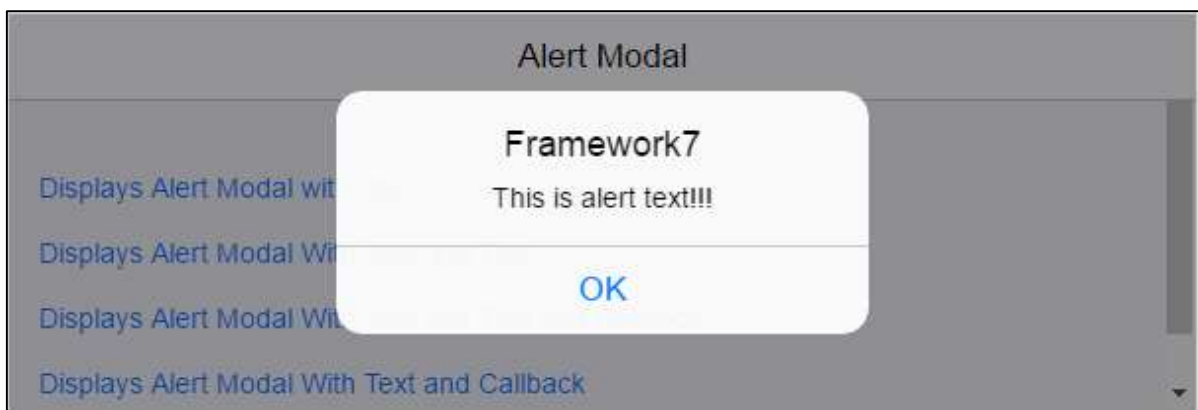
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **modal_alert.html** file in your server root folder.

- Open this HTML file as `http://localhost/modal_alert.html` and the output is displayed as shown below.
- When the user clicks on the first option, it displays the alert modal with the text.
- When the user clicks on the second option, it displays the alert modal along with the text and the title.
- When the user clicks on the third option, it displays the alert modal with text and title and when OK is clicked, it executes a callback function.
- When the user clicks on the last option, it displays the alert modal with text and when clicked OK, it executes a callback function.



Framework7 – Confirm Modal

Description

Confirm modal is used to confirm some action for the displayed content. The confirm modal uses the following methods:

```
myApp.confirm(text, [title, callbackOk, callbackCancel])
```

Or

```
myApp.confirm(text, [callbackOk, callbackCancel])
```

The above methods accept parameters which are listed below:

- **text:** It displays the confirm text.
- **title:** It is an optional method that displays the confirm modal with the title.
- **callbackOk:** It is an optional method, which provides callback function that executes when the "OK" button is clicked by the user on confirm modal.
- **callbackCancel:** It is an optional method, which provides the callback function that executes when the "Cancel" button is clicked by the user on confirm modal.

Example

The following example demonstrates the use of confirm modal in Framework7, which displays the confirm box when you click on the links to perform some actions:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Confirm Modal</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="center sliding">Confirm Modal</div>
          </div>
        </div>
        <div class="pages">
          <div data-page="index" class="page navbar-fixed">
            <div class="page-content">
              <div class="content-block">
                <p><a href="#" class="confirm-ok">Displays Confirm
Modal with Text and Ok callback</a></p>
                <p><a href="#" class="confirm-ok-cancel">Displays
Confirm Modal With Text, Ok and Cancel callbacks</a></p>
                <p><a href="#" class="confirm-title-ok">Displays
Confirm Modal With Text, Title and Ok callbacks</a></p>
                <p><a href="#" class="confirm-title-ok-cancel">Displays
Confirm Modal With Text, Title, Ok and Cancel callbacks</a></p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </body>
  </html>

```



```

        </div>
    </div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // Here you can initialize the app
    var myApp = new Framework7();

    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;
    // Add the view
    var mainView = myApp.addView('.view-main', {
    // enable the dynamic navbar for this view:
    dynamicNavbar: true
    });
    $$('.confirm-ok').on('click', function () {
        myApp.confirm('Are you ready to begin?', function () {
            myApp.alert('You have clicked the Ok button!!!');
        });
    });
    $$('.confirm-ok-cancel').on('click', function () {
    myApp.confirm('Are you ready to begin?',
    function () {
    myApp.alert('You have clicked the Ok button!!!');
    },
    function () {
    myApp.alert('You have clicked the Cancel button!!!');
    }
    );
    });
    $$('.confirm-title-ok').on('click', function () {
    myApp.confirm('Are you ready to begin?', 'My Title', function () {
    myApp.alert('You have clicked the Ok button!!!');
    });
    });
</script>

```

```

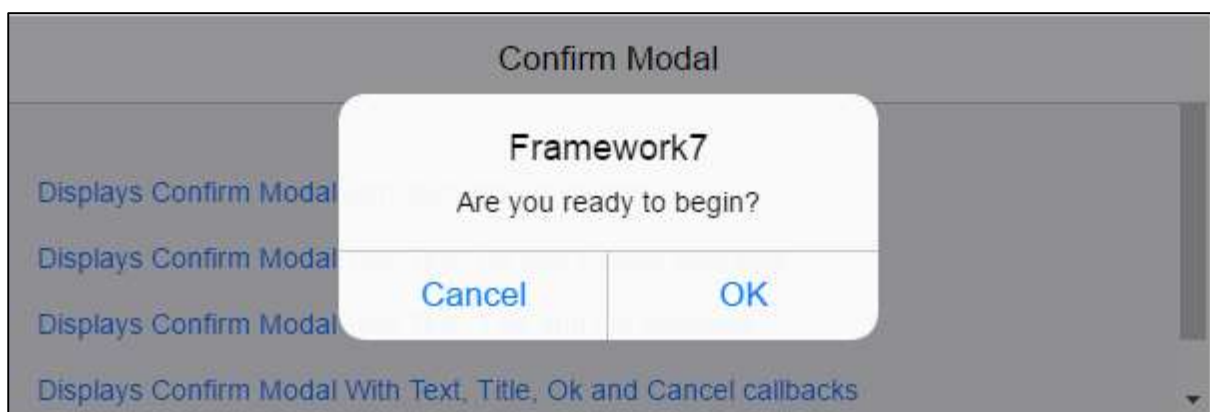
    $$('.confirm-title-ok-cancel').on('click', function () {
    myApp.confirm('Are you ready to begin?', 'My Title',
    function () {
    myApp.alert('You clicked Ok button!!!');
    },
    function () {
    myApp.alert('You have clicked the Cancel button!!!');
    }
    );
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **modal_confirm.html** file in your server root folder.
- Open this HTML file as http://localhost/modal_confirm.html and the output is displayed as shown below.
- When you click on the Displays Confirm Modal with Text and OK callback, it asks for confirmation. When you click OK, it displays the confirm text as a callback function.
- When you click on Displays Confirm Modal with Text, OK and Cancel callbacks; it displays a confirm text as a callback function when clicked OK and displays a callback function that executes cancel when the user clicks the Cancel button.



Framework7 – Prompt Modal

Description

The Prompt modal allows the users to take some actions on the displayed content. It uses the following methods:

```
myApp.prompt(text, [title, callbackOk, callbackCancel])
```

Or

```
myApp.prompt(text, [callbackOk, callbackCancel])
```

The above methods accept parameters that are listed below:

- **text:** It displays the prompt modal with the text.
- **title:** It is an optional method that displays the prompt modal with the title.
- **callbackOk:** It is an optional method, which provides the callback function that executes when the "OK" button is clicked by the user on prompt modal.
- **callbackCancel:** It is an optional method, which provides the callback function that executes when the "Cancel" button is clicked by the user on prompt modal.

Example

The following example demonstrates the use of prompt modal in the Framework7, which displays the prompt box when you click on the links to perform some actions:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Prompt Modal</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
```

```

<div class="view view-main">
  <div class="navbar">
    <div class="navbar-inner">
      <div class="center sliding">Prompt Modal</div>
    </div>
  </div>
  <div class="pages">
    <div data-page="index" class="page navbar-fixed">
      <div class="page-content">
        <div class="content-block">
          <p><a href="#" class="prompt-ok">Displays Prompt Modal
with Text and Ok callback</a></p>
          <p><a href="#" class="prompt-ok-cancel">Displays Prompt
Modal With Text, Ok and Cancel callbacks</a></p>
          <p><a href="#" class="prompt-title-ok">Displays Prompt
Modal With Text, Title and Ok callbacks</a></p>
          <p><a href="#" class="prompt-title-ok-cancel">Displays
Prompt Modal With Text, Title, Ok and Cancel callbacks</a></p>
        </div>
      </div>
    </div>
  </div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
  // Here you can initialize the app
  var myApp = new Framework7();
  // If your using custom DOM library, then save it to $$ variable
  var $$ = Dom7;
  // Add the view
  var mainView = myApp.addView('.view-main', {
  // enable the dynamic navbar for this view:
  dynamicNavbar: true
  });
  $$('.prompt-ok').on('click', function () {
    myApp.prompt('Enter your name?', function (value) {

```

```

        myApp.alert('Name is "' + value + '" and you have clicked Ok
button.');
```

```

        });
    });
    $$('.prompt-ok-cancel').on('click', function () {
        myApp.prompt('Enter your name?', function (value) {
            myApp.alert('Name is "' + value + '" and you have clicked Ok
button.');
```

```

        },
        function (value) {
            myApp.alert('Name is "' + value + '" and you have clicked
cancel button.');
```

```

        }
    });
    $$('.prompt-title-ok').on('click', function () {
        myApp.prompt('Enter your name?', 'My Title', function (value) {
            myApp.alert('Name is "' + value + '" and you have clicked Ok
button.');
```

```

        });
    });
    $$('.prompt-title-ok-cancel').on('click', function () {
        myApp.prompt('Enter your name?', 'My Title', function (value) {
            myApp.alert('Name is "' + value + '" and you have clicked Ok
button.');
```

```

        },
        function (value) {
            myApp.alert('Name is "' + value + '" and you have clicked
cancel button.');
```

```

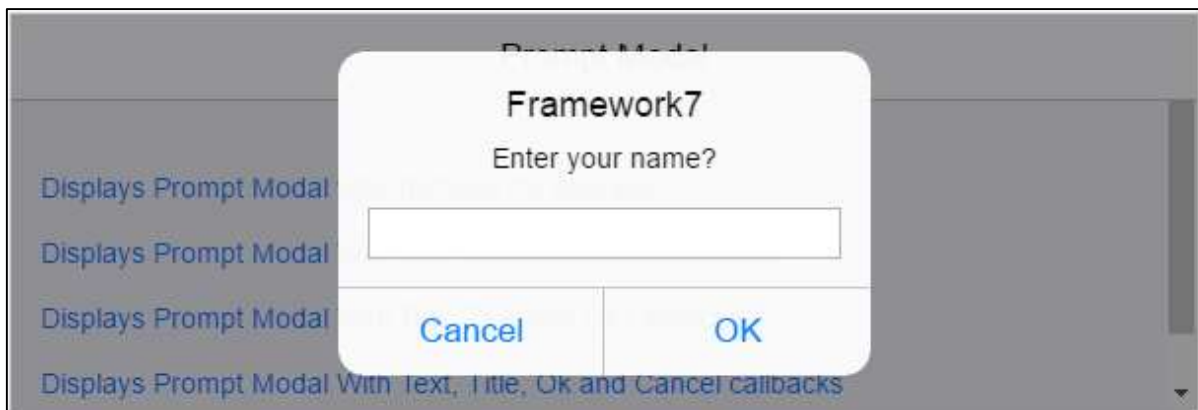
        }
    });
    </script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **modal_prompt.html** file in your server root folder.

- Open this HTML file as http://localhost/modal_prompt.html and the output is displayed as shown below.
- When the user clicks on the first option, it links to a popup window. When the user enters text into the box, it executes the callback function when OK is clicked.
- When the user clicks on the second option, it links to a popup window, where it executes the callback function when the user clicks on the Cancel button. It executes a callback function when the user enters the text into the box and clicks OK.
- When the user clicks on the third option, it links to a popup window with the text and the title. When the user enters text into the box, it executes the callback function when OK is clicked.
- When the user clicks on the last option, it links to a popup window with the text and the title and executes a callback function when the user clicks on cancel. It executes a callback function when the user enters the text and clicks OK.



Framework7 – Login and Password Modal

Description

You can use this type of modal for authentication purpose. It uses the following methods:

```
myApp.modalLogin(text, [title, callbackOk, callbackCancel])
```

Or

```
myApp.modalLogin(text, [callbackOk, callbackCancel])
```

The above methods accept the parameters, which are listed below:

- **text:** It displays the modal with text.
- **title:** It is an optional method that displays modal with title.
- **callbackOk:** It is an optional method, which provides the callback function that executes when the "OK" button is clicked by the user.

- **callbackCancel:** It is an optional method, which provides the callback function that executes when the "Cancel" button is clicked by the user.

Example

The following example demonstrates the use of login and password modal in the Framework7, which provides the modal box for entering username and password for authentication purpose:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Login and Password Modal</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="center sliding">Login and Password Modal</div>
          </div>
        </div>
        <div class="pages">
          <div data-page="index" class="page navbar-fixed">
            <div class="page-content">
              <div class="content-block">
                <p><a href="#" class="login-modal">Displays Login
Modal</a></p>
                <p><a href="#" class="password-modal">Displays Password
Modal</a></p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

        </div>
    </div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // Here you can initialize the app
    var myApp = new Framework7();

    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;

    // Add the view
    var mainView = myApp.addView('.view-main', {
    // enable the dynamic navbar for this view:
    dynamicNavbar: true
    });

    $$('.login-modal').on('click', function () {
        myApp.modalLogin('Enter your details:', function (uname, pwd) {
            myApp.alert('Username: ' + uname + ', Password: ' + pwd);
        });
    });

    $$('.password-modal').on('click', function () {
        myApp.modalPassword('Enter your password:', function (pwd) {
            myApp.alert('Your password is: ' + pwd);
        });
    });
</script>
</body>
</html>

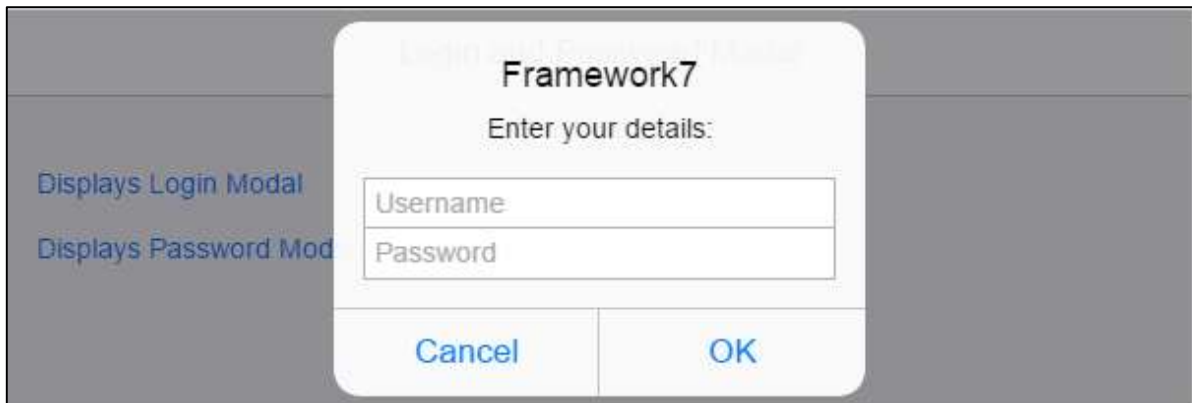
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **modal_login_password.html** file in your server root folder.

- Open this HTML file as http://localhost/modal_login_password.html and the output is displayed as shown below.
- When you click on the first option, you will get a popup window where you can enter the username and password. When OK is clicked, the callback function is executed displaying the entered credentials.
- When you click on the second option, you will get a popup window where you enter a password and when OK is clicked, the callback function is executed displaying the entered password.



Framework7 – Nesting Modal

Description

You can use one modal inside another modal.

Example

The following example demonstrates the use of nesting modal in the Framework7, which provides the modal inside the other modal:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Nesting Modals</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
  <div class="views">
    <div class="view view-main">
      <div class="navbar">
        <div class="navbar-inner">
          <div class="center sliding">Nesting Modals</div>
        </div>
      </div>
      <div class="pages">
        <div data-page="index" class="page navbar-fixed">
          <div class="page-content">
            <div class="content-block">
              <p><a href="#" class="nesting_modal">Enter your
name:</a></p>
            </div>
          </div>
        </div>
      </div>
    </div>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
    <script>
      // Here you can initialize the app
      var myApp = new Framework7();
      // If your using custom DOM library, then save it to $$ variable
      var $$ = Dom7;
      // Add the view
      var mainView = myApp.addView('.view-main', {
      // enable the dynamic navbar for this view:
      dynamicNavbar: true
      });
      $$('.nesting_modal').on('click', function () {
        myApp.prompt('Enter your name?', function (value) {

```

```

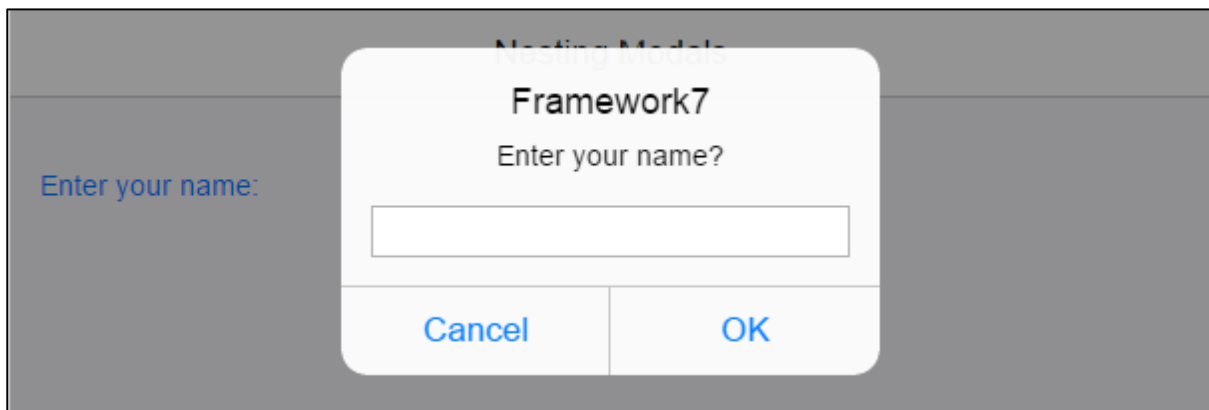
        myApp.confirm('Hey...your name is : ' + value + ', function
    () {
        myApp.alert('This is your name : "' + value + '!!!!');
    });
});
});
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **modal_nesting.html** file in your server root folder.
- Open this HTML file as http://localhost/modal_nesting.html and the output is displayed as shown below.
- When you click on the “Enter your name” option, you will get a popup wizard, which executes a function and displays the entered text. After the text is entered as input and OK is clicked, another function is executed when OK is clicked once again.



Framework7 – Preloader Modal

Description

The Preloader modal defines the background action and stops the user actions during this action. It uses the following method to open the preloader modal:

- `myApp.showPreloader([title])`
 - `title`: It is an optional method that displays the preloader modal with the title.

Example

The following example demonstrates the use of preloader modal in Framework7, which indicates some background activity while loading the modal:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Preloader Modal</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="center sliding">Preloader Modal</div>
          </div>
        </div>
        <div class="pages">
          <div data-page="index" class="page navbar-fixed">
            <div class="page-content">
              <div class="content-block">
                <p><a href="#" class="preloader_open">Display the
Preloader</a></p>
                <p><a href="#" class="preloader_title">Display the
Preloader with title</a></p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>

```

```

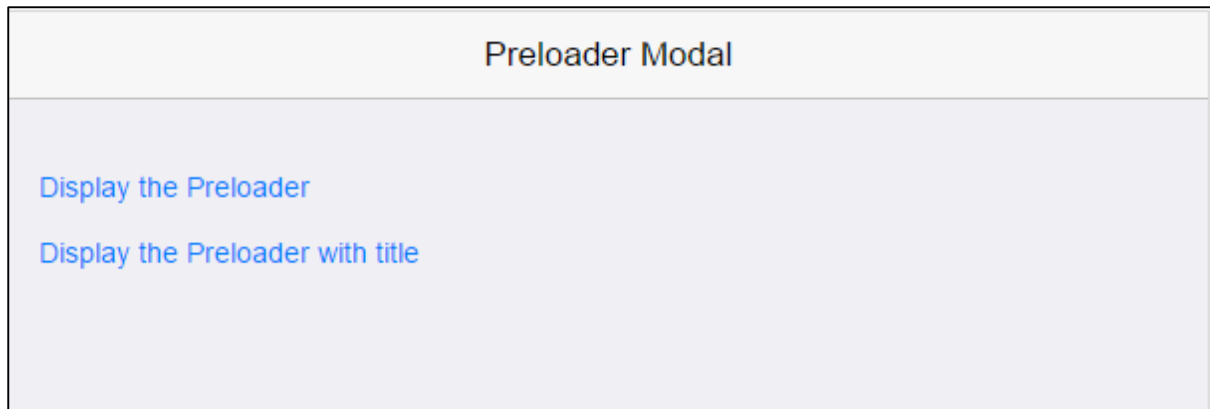
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
    <script>
        // Here you can initialize the app
        var myApp = new Framework7();
        // If your using custom DOM library, then save it to $$ variable
        var $$ = Dom7;
        // Add the view
        var mainView = myApp.addView('.view-main', {
        // enable the dynamic navbar for this view:
        dynamicNavbar: true
        });
        $$('.preloader_open').on('click', function () {
            myApp.showPreloader();
            setTimeout(function () {
                myApp.hidePreloader();
            }, 1500);
        });
        $$('.preloader_title').on('click', function () {
            myApp.showPreloader('My Title')
            setTimeout(function () {
                myApp.hidePreloader();
            }, 1500);
        });
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **modal_preloader.html** file in your server root folder.
- Open this HTML file as http://localhost/modal_preloader.html and the output is displayed as shown below.
- When the first option is clicked, it displays the preloader model.
- When the second option is clicked, it displays the preloader model along with the title.



Framework7 – Indicator Modal

Description

The Indicator modal specifies small indication activity like preloader modal for the user actions. It uses the following methods:

- **showIndicator:** It is used to show the modal with indicator.
- **hideIndicator:** It is used to hide the modal with indicator.

Example

The following example demonstrates the use of indicator modal in Framework7, which displays the small indication activity when you click on the link:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Indicator Modal</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
```

```

<div class="view view-main">
  <div class="navbar">
    <div class="navbar-inner">
      <div class="center sliding">Indicator Modal</div>
    </div>
  </div>
  <div class="pages">
    <div data-page="index" class="page navbar-fixed">
      <div class="page-content">
        <div class="content-block">
          <p><a href="#" class="indicator_modal">Display the
Indicator</a></p>
        </div>
      </div>
    </div>
  </div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
  // Here you can initialize the app
  var myApp = new Framework7();
  // If your using custom DOM library, then save it to $$ variable
  var $$ = Dom7;
  // Add the view
  var mainView = myApp.addView('.view-main', {
  // enable the dynamic navbar for this view:
  dynamicNavbar: true
  });
  $$('.indicator_modal').on('click', function () {
    myApp.showIndicator();
    setTimeout(function () {
      myApp.hideIndicator();
    }, 1500);
  });
</script>

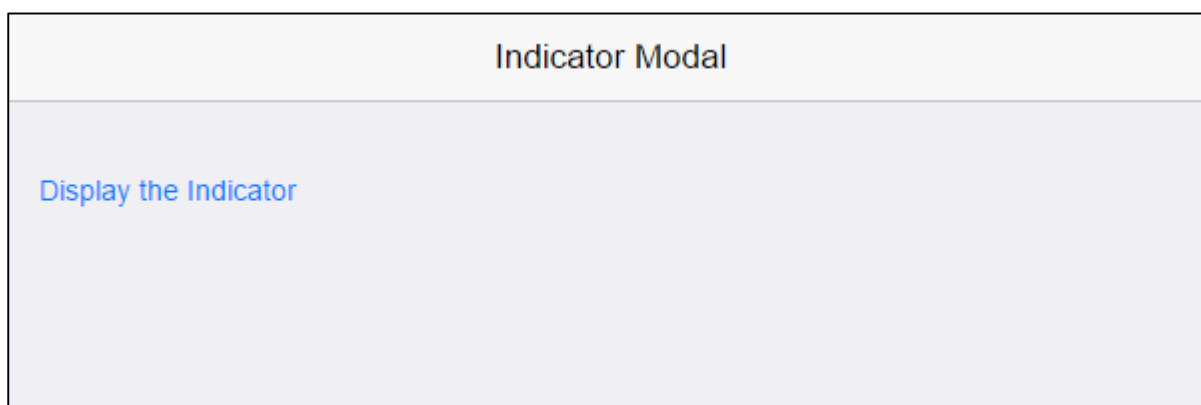
```

```
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **modal_indicator.html** file in your server root folder.
- Open this HTML file as `http://localhost/modal_indicator.html` and the output is displayed as shown below.
- When the user clicks on the "Display the Indicator" option, the indicator modal displays the indication activity like a preloader.



Custom Modals

The following table shows the list of Modal parameters:

S. No.	Parameter & Description	Type	Default
1	title Its an optional parameter that defines the string with the modal title.	string	-
2	text Its an optional parameter that defines the string with the modal text.	string	-
3	afterText Its an optional parameter that defines the string after the text.	string	
4	buttons Its an optional parameter that contains an array of buttons and each button is specified with the button parameter as object.	array	-

5	verticalButtons Its an optional parameter, which specifies the vertical buttons by setting it to true.	boolean	false
6	onClick Its an optional parameter that executes the callback function when the user clicks the modal button.	function	-

The following table lists the Button's parameters:

S. No.	Parameter & Description	Type	Default
1	text Its an optional parameter that defines the string with the button's text.	string	-
2	bold Its an optional parameter, which makes the text bold by setting it to true.	boolean	false
3	close Its an optional parameter, which will close the modal when you click on the button if it is set to true.	boolean	true
4	onClick Its an optional parameter that executes the callback function when the user clicks on the button.	function	-

Modal Events

The following table shows a list of the Modal events:

S. No.	Event	Target & Description
1	open	Modal Element When you open an animation, this event will get fired.
2	opened	Modal Element When opening of an animation is complete, this event will get fired.
3	close	Modal Element When you close an animation, this event will get fired.

4	closed	Modal Element When closing of an animation is complete, this event will get fired.
---	--------	--

Framework7 – Overlay Popup

Description

Popup is a popup box, which displays the content, when the user clicks on the element by using the *popup* class.

The size of the popup varies on different devices. On phones, it will display fullscreen and on tablets, it will display with 630px width and height. You can add *tablet-fullscreen* class to make fullscreen size on tablets.

You can open and close the popup in two ways as specified in the following table:

S. No.	Type
1	From HTML You can open and close the popup by using classes and data attributes.
2	Using JavaScript You can open and close the popup by using the JavaScript code.

Framework7 – Overlay Popup from HTML

Description

You can open and close the popup by using classes and data attributes as shown below:

- **open-popup:** It is used to open popup.
- **close-popup:** It is used to close popup.
- **data-popup=".my-popup":** Whenever more than one popups are used in your app, you need to specify the appropriate popup by using this attribute.

Example

The following example displays the popup from HTML in Framework7 by using the classes and attributes:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
```

```

<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style" content="black">
<title>Popup from HTML</title>
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
<div class="views">
<div class="view view-main">
<div class="navbar">
<div class="navbar-inner">
<div class="center sliding">Popup from HTML</div>
</div>
</div>
<div class="pages">
<div data-page="index" class="page navbar-fixed">
<div class="page-content">
<div class="content-block">
<p><a href="#" data-popup=".first_page" class="open-
popup">Open First Page</a></p>
<p><a href="#" data-popup=".second_page" class="open-
popup">Open Second Page</a></p>
</div>
</div>
</div>
</div>
</div>
<div class="popup first_page">
<div class="content-block">
<p>First Page</p>
<p> <a href="#" class="close-popup">Close popup</a></p>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque
ac diam ac quam euismod porta vel a nunc. Quisque sodales scelerisque est, at
porta justo cursus ac. Integer vitae quam a ante lobortis lobortis. Nam
vehicula sagittis quam, sit amet congue purus consequat sed. Maecenas eget
mattis lectus. Aliquam luctus luctus leo ac fringilla. Sed nec eros vel purus

```

```

tincidunt tincidunt in in orci. Sed tellus neque, pellentesque nec metus id,
congue elementum odio.</p>
    </div>
</div>
<div class="popup second_page">
    <div class="content-block">
        <p>Second Page</p>
        <p> <a href="#" class="close-popup">Close popup</a></p>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque
ac diam ac quam euismod porta vel a nunc. Quisque sodales scelerisque est, at
porta justo cursus ac. Integer vitae quam a ante lobortis lobortis. Nam
vehicula sagittis quam, sit amet congue purus consequat sed. Maecenas eget
mattis lectus. Aliquam luctus luctus leo ac fringilla. Sed nec eros vel purus
tincidunt tincidunt in in orci. Sed tellus neque, pellentesque nec metus id,
congue elementum odio.</p>
    </div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // Here you can initialize the app
    var myApp = new Framework7();
    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;
    // Add the view
    var mainView = myApp.addView('.view-main', {
    // enable the dynamic navbar for this view:
    dynamicNavbar: true
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **popup_from_html.html** file in your server root folder.
- Open this HTML file as http://localhost/popup_from_html.html and the output is displayed as shown below.

- When you click on the first option, a popup window from html is opened and the first page is displayed to you.
- Similarly, when you click on the second option, the popup window for the second page is displayed.
- You can click on the Close popup link to close the popup window.



Framework7 – Overlay Popup using JavaScript

Description

You can open and close the popup by using the JavaScript code. You can use the *popup(popup)* method to open the popup modal and *closeModal(popup)* method to close the popup modal.

Example

The following example displays the popup when you click on the links by using JavaScript in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Popup using JavaScript</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
```

```

<body>
  <div class="views">
    <div class="view view-main">
      <div class="navbar">
        <div class="navbar-inner">
          <div class="center sliding">Popup using JavaScript</div>
        </div>
      </div>
      <div class="pages">
        <div data-page="index" class="page navbar-fixed">
          <div class="page-content">
            <div class="content-block">
              <p><a href="#" class="first_page">Open First
Page</a></p>
              <p><a href="#" class="second_page">Open Second
Page</a></p>
            </div>
          </div>
        </div>
      </div>
      <div class="popup popup-first_page">
        <div class="content-block">
          <p>First Page</p>
          <p> <a href="#" class="close-popup">Close popup</a></p>
          <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque
ac diam ac quam euismod porta vel a nunc. Quisque sodales scelerisque est, at
porta justo cursus ac. Integer vitae quam a ante lobortis lobortis. Nam
vehicula sagittis quam, sit amet congue purus consequat sed. Maecenas eget
mattis lectus. Aliquam luctus luctus leo ac fringilla. Sed nec eros vel purus
tincidunt tincidunt in in orci. Sed tellus neque, pellentesque nec metus id,
congue elementum odio.</p>
        </div>
      </div>
      <div class="popup popup-second_page">
        <div class="content-block">
          <p>Second Page</p>
          <p> <a href="#" class="close-popup">Close popup</a></p>
        </div>
      </div>
    </div>
  </div>

```

```

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque
ac diam ac quam euismod porta vel a nunc. Quisque sodales scelerisque est, at
porta justo cursus ac. Integer vitae quam a ante lobortis lobortis. Nam
vehicula sagittis quam, sit amet congue purus consequat sed. Maecenas eget
mattis lectus. Aliquam luctus luctus leo ac fringilla. Sed nec eros vel purus
tincidunt tincidunt in in orci. Sed tellus neque, pellentesque nec metus id,
congue elementum odio.</p>

        </div>

    </div>

    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>

    <script>
        // Here you can initialize the app
        var myApp = new Framework7();

        // If your using custom DOM library, then save it to $$ variable
        var $$ = Dom7;

    // Add the view
    var mainView = myApp.addView('.view-main', {
        // enable the dynamic navbar for this view:
        dynamicNavbar: true
    });

    //It will open popup when you click the link with class 'first_page'
    $$('.first_page').on('click', function(){
        myApp.popup('.popup-first_page');
    });

    //It will open popup when you click the link with class 'second_page'
    $$('.second_page').on('click', function(){
        myApp.popup('.popup-second_page');
    });

    </script>
</body>
</html>

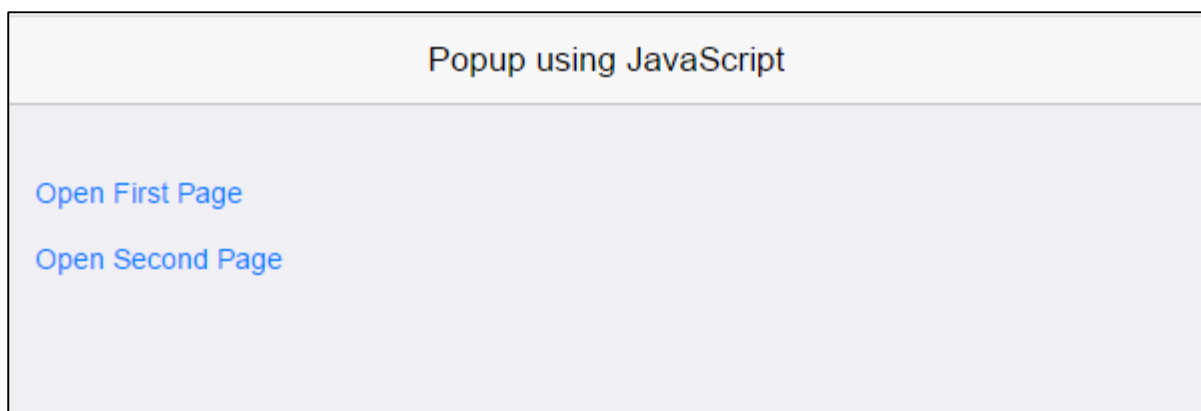
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **popup_using_javascript.html** file in your server root folder.
- Open this HTML file as http://localhost/popup_using_javascript.html and the output is displayed as shown below.

- When you click on the first option, a popup window is opened and the first page is displayed to you and a popup is opened and closed using javascript code.
- Similarly, when you click on the second option, the popup window for the secondpage is displayed.
- You can click on close popup link to close the popup window.



Popup Events

S. No.	Event	Target & Description
1	open	Popup Element When you open an animation, this event will get fired.
2	opened	Popup Element When the opening of an animation is completed, this event will get fired.
3	close	Popup Element When you close an animation, this event will get fired.
4	closed	Popup Element When the closing of an animation is completed, this event will get fired.

You can also create a dynamic popup by using its HTML to App methods. Here is an [example](#).

Framework7 – Overlay Dynamic Popup

Description

You can also create a dynamic popup by using its HTML to App methods. It uses two parameters:

- **popupHTML:** It contains string element of Popup content.

- **removeOnClose:** It includes boolean value, which will be removed from DOM when you close the Popup. By default, it includes the true value.

Example

The following example demonstrates the use of dynamic popup in Framework7:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Dynamic Popup</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="center sliding">Dynamic Popup</div>
          </div>
        </div>
        <div class="pages">
          <div data-page="index" class="page navbar-fixed">
            <div class="page-content">
              <div class="content-block">
                <p><a href="#" class="first_page">Open the
Popup</a></p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>

```

```

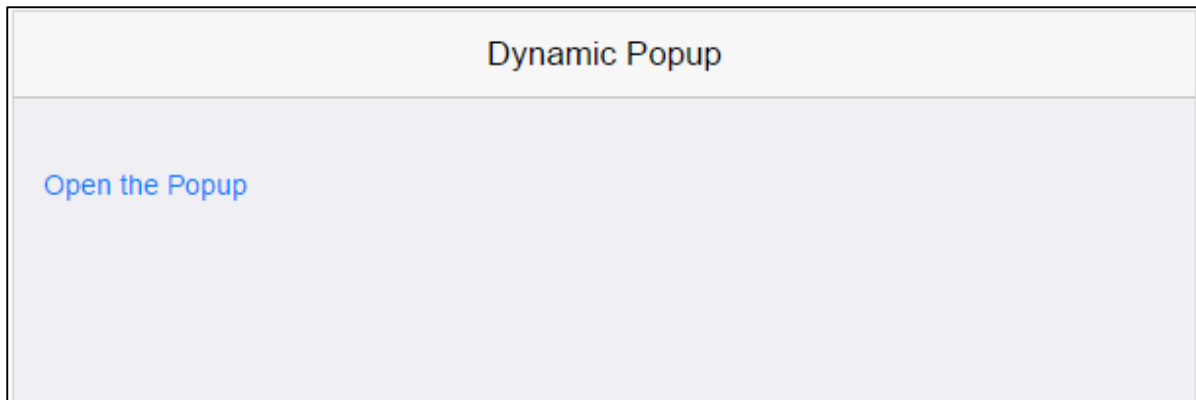
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
    <script>
        // Here you can initialize the app
        var myApp = new Framework7();
        // If your using custom DOM library, then save it to $$ variable
        var $$ = Dom7;
        // Add the view
        var mainView = myApp.addView('.view-main', {
        // enable the dynamic navbar for this view:
        dynamicNavbar: true
        });
        $$('.first_page').on('click', function () {
        var popupHTML = '<div class="popup">'+
        '<div class="content-block">'+
        '<p>You have created the Popup dynamically!!!</p>'+
        '<p><a href="#" class="close-popup">Close the Popup</a></p>'+
        '</div>'+
        '</div>'
        myApp.popup(popupHTML);
        });
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **popup_dynamic.html** file in your server root folder.
- Open this HTML file as http://localhost/popup_dynamic.html and the output is displayed as shown below.



Framework7 – Popover

Description

To manage the presentation of temporary content, popover component can be used. It will exist until you click outside of the popover window or you can dismiss it explicitly.

It is suggested not to use popover on small screens (iPhone). You should use [Action sheet](#) or [Action sheet to popover conversion](#) which are discussed in next chapter.

The popover layout is very simple and is added usually at the end of the `<body>` section as shown below:

```
<body>
  ....
  <div class="popover">
    <!-- Popover's angle arrow -->
    <div class="popover-angle"></div>
    <!-- Popover's content -->
    <div class="popover-inner">
      <!-- Any content of HTML here -->
    </div>
  </div>
</body>
```

Popover is highly customizable. You can place any element inside and event another view with navigation. The following table shows the popover types used in Framework7:

S. No.	Types & Description
1	Open and close Popover Opening and closing the popover can be achieved by using HTML and JavaScript.
2	Dynamic Popover The popover can be created dynamically by using related app methods.

3	<p>Popover Events</p> <p>To detect how the user interacts with the popover, you can use popover events.</p>
---	--

Framework7 – Open & Close Popover

Description

Opening and closing the popover can be achieved in the following two ways:

S. No.	Types & Description
1	<p>From HTML</p> <p>The required popover can be opened and closed using special classes and data attributes on links.</p>
2	<p>Using JavaScript</p> <p>Popover can be opened and closed with JavaScript by using the related app methods.</p>

Framework7 – Open & Close Popover From HTML

Description

The required popover can be opened and closed using special classes and data attributes on links. The following statements briefly describe how to open/close popover:

- To open popover, add *open-popover* class to any HTML element and you can use add *close-popover* class to close popover.
- When you have many popover in app, you need to specify *data-popover=".my-popover"* attribute to appropriate popover.
- When popover is opened using this method (from HTML), the popover will be automatically placed around element that we click to invoke this popover.

Example

The following example demonstrates the opening and the closing popover from HTML in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
```

```

<title>Open and close Popover From HTML</title>
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
<div class="views">
<div class="view view-main">
<div class="pages">
<div data-page="home" class="page navbar-fixed">
<div class="navbar">
<div class="navbar-inner">
<div class="left"> <a href="#" data-popover=".popover-
menus" class="link open-popover">Menus</a></div>
<div class="center">Open and close Popover From
HTML</div>
<div class="right"> <a href="#" data-popover=".popover-
about" class="link open-popover">About</a></div>
</div>
</div>
<div class="page-content">
<div class="content-block">
<p><a href="#" data-popover=".popover-about"
class="open-popover">Open About Popover</a></p>
<p><a href="#" data-popover=".popover-menu"
class="open-popover">Open Menu Popover</a></p>
</div>
</div>
</div>
</div>
</div>
<div class="popover popover-about">
<div class="popover-angle"></div>
<div class="popover-inner">
<div class="content-block">

```

```

        <p>About</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
        Quisque ac diam ac quam euismod porta vel a nunc.</p>
    </div>
</div>
</div>
<div class="popover popover-menus">
    <div class="popover-angle"></div>
    <div class="popover-inner">
        <div class="list-block">
            <ul>
                <li><a href="#" class="list-button item-link">Menu 1</a></li>
                <li><a href="#" class="list-button item-link">Menu 2</a></li>
                <li><a href="#" class="list-button item-link">Menu 3</a></li>
                <li><a href="#" class="list-button item-link">Menu 4</a></li>
                <li><a href="#" class="list-button item-link">Menu 5</a></li>
            </ul>
        </div>
    </div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var $$ = Dom7;
</script>
</body>
</html>

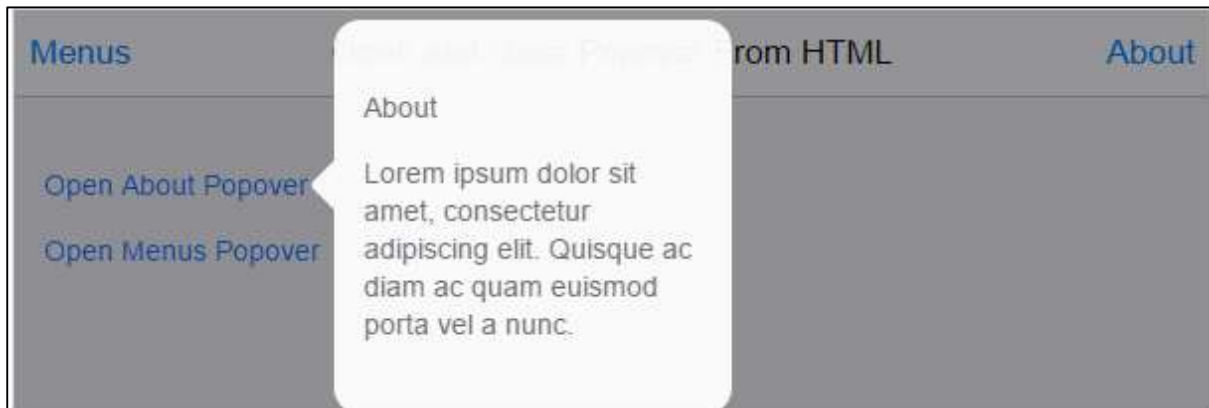
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **popover_open_close_html.html** file in your server root folder.
- Open this HTML file as http://localhost/popover_open_close_html.html and the output is displayed as shown below.

- When you click on the first link, the about popover window is opened. Similarly, when you click on the second link, the menu popover window is opened with several menu items.
- The menu and the about popover windows also are opened by clicking on the menu and about options on the top. The popover will be opened and closed from HTML.



When you click on the link, the popover will open using the `popover` class and if you click the mouse outside, the popover will get close.

Framework7 – Open & Close Popover Using JavaScript

Description

The popover can also be opened and closed with JavaScript by using the related app methods as shown below:

- **myApp.popover(popover, target)** - It is used to open the popover around the target element and it accepts the following arguments:
 - **popover** - It is a *required* argument, which is an *HTMLElement* or *string* (with *CSS Selector*) of popover to open.
 - **target** - It is a *required* argument, which is an *HTMLElement* or *string* (with *CSS Selector*) of target element used to set popover position around this element.
- **myApp.closeModal(popover)** - It is used to close the popover and accepts *popover* argument, which is an *HTMLElement* or *string* (with *CSS Selector*). It is an optional argument and if not specified, any of the opened popover will be closed.

If you open popover using JavaScript, the *target* element needs to be passed to set the popover around the target element.

Example

The following example demonstrates the use of opening and closing popover using JavaScript in Framework7:

```
<!DOCTYPE html>
```

```

<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Open and close Popover Using JavaScript</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> <a href="#" class="link open-
menus">Menus</a></div>
                <div class="center">Open and close Popover Using
JavaScript</div>
                <div class="right"> <a href="#" class="link open-
about">About</a></div>
              </div>
            </div>
            <div class="page-content">
              <div class="content-block">
                <p><a href="#" class="open-menus">Open menus
Popover</a></p>
                <p><a href="#" class="open-about">Open About
Popover</a></p>
                <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
ex ea commodo consequat <a href="#" class="open-about">About</a> nibh iaculis
quis. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum
dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt mollit anim id est laborum <a
href="#" class="open-menus">Menus</a>.</p>

```



```

        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
<style>.popover{width:200px;}</style>
<div class="popover popover-about">
    <div class="popover-angle"></div>
    <div class="popover-inner">
        <div class="content-block">
            <p>About</p>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Quisque ac diam ac quam euismod porta vel a nunc.</p>
        </div>
    </div>
</div>
<div class="popover popover-menus">
    <div class="popover-angle"></div>
    <div class="popover-inner">
        <div class="list-block">
            <ul>
                <li><a href="#" class="list-button item-link">Menu 1</a></li>
                <li><a href="#" class="list-button item-link">Menu 2</a></li>
                <li><a href="#" class="list-button item-link">Menu 3</a></li>
                <li><a href="#" class="list-button item-link">Menu 4</a></li>
                <li><a href="#" class="list-button item-link">Menu 5</a></li>
            </ul>
        </div>
    </div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var $$ = Dom7;
    $$('.open-about').on('click', function () {
        var clickedLink = this;

```

```

        myApp.popover('.popover-about', clickedLink);
    });
    $$('.open-menus').on('click', function () {
        var clickedLink = this;
        myApp.popover('.popover-menu', clickedLink);
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **popover_open_close_js.html** file in your server root folder.
- Open this HTML file as http://localhost/popover_open_close_js.html and the output is displayed as shown below.
- When you click on the first link, the menu popover window opens with several menu items. Similarly, when you click on the second link, the about popover window opens.
- The menu and the about popover windows also open by clicking on the menu and about options. The popover will open and close around the target element using javascript.



Framework7 – Dynamic Popover

Description

The popover can be created dynamically using the related app methods as shown below:

- `myApp.popover(popoverHTML, target, removeOnClose)` - This method accepts the following arguments:

- **popoverHTML** - It is the HTML string of popover.
- **target** - It is an *HTMLElement* or *string (with CSS Selector)* of target element used to set popover position around. It is a *required* argument.
- **removeOnClose** - It is of *Boolean* type and is an optional argument. By default, it is set to *true*, which removes the popover from DOM when closed.

The dynamically created popover's *HTMLElement* will be returned by this method.

Example

The following example demonstrates the use of dynamic popover in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Dynamic Popover</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> <a href="#" class="link create-
menus">Menus</a></div>
                <div class="center">Dynamic Popover</div>
                <div class="right"> <a href="#" class="link create-
about">About</a></div>
              </div>
            </div>
            <div class="page-content">
```

```

        <div class="content-block">
            <p><a href="#" class="create-about">Create About
Popover</a></p>
            <p><a href="#" class="create-menus">Create Menus
Popover</a></p>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
ex ea commodo consequat <a href="#" class="create-about">About</a> nibh iaculis
quis. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum
dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt mollit anim id est laborum <a
href="#" class="create-menus">Menus</a>.</p>
        </div>
    </div>
</div>
</div>
</div>
</div>
<style>.popover{width:300px;}</style>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var $$ = Dom7;
    $$('.create-about').on('click', function () {
        var clickedLink = this;
        var popoverHTML = '<div class="popover">'+
            '<div class="popover-inner">'+
            '<div class="content-block">'+
            '<p>About Popover created dynamically.</p>'+
            '<p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Quisque ac diam ac quam euismod porta vel a nunc.</p>'+
            '</div>'+
            '</div>'+
            '</div>'
        myApp.popover(popoverHTML, clickedLink);
    });
    $$('.create-menus').on('click', function () {
        var clickedLink = this;
        var popoverHTML = '<div class="popover">'+

```

```

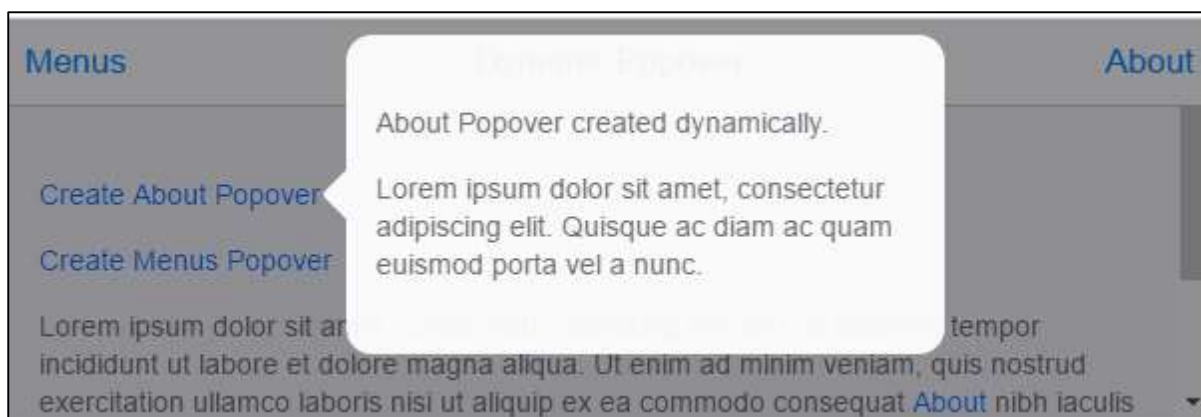
        '<div class="popover-inner">'+
        '<div class="list-block">'+
        '<ul>'+
        'button">Menus 1</li>'+
        '<li><a href="#" class="item-link list-
        button">Menus 2</li>'+
        '<li><a href="#" class="item-link list-
        button">Menus 3</li>'+
        '<li><a href="#" class="item-link list-
        button">Menus 4</li>'+
        '<li><a href="#" class="item-link list-
        button">Menus 5</li>'+
        '</ul>'+
        '</div>'+
        '</div>'+
        '</div>'
        myApp.popover(popoverHTML, clickedLink);
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **popover_dynamic.html** file in your server root folder.
- Open this HTML file as http://localhost/popover_dynamic.html and the output is displayed as shown below.
- You can create dynamic about popover by clicking on the first link. Similarly, to create dynamic menu popover click on the second link.
- When you click the links, a dynamic popover gets created and you can close the popover by clicking outside.



When you click on the links, the dynamic popover will open and click the mouse outside to close the popover.

Framework7 – Popover Events

Description

To detect how a user interacts with the popover, you can use the popover events as shown in the following table:

S. No.	Events & Description	Target
1	open When popover starts its opening animation, this event will trigger.	Popover Element <div class="popover">
2	opened After completing the popover opening animation, this event will be triggered.	Popover Element <div class="popover">
3	close When popover starts its closing animation, this event will be triggered.	Popover Element <div class="popover">
4	closed After completing the popover closing animation, this event will be triggered.	Popover Element <div class="popover">

Framework7 – Action Sheet

Description

The *Action Sheet* is used to present the user with a set of possibilities for how to handle a given task. It can also be used to prompt the user to handle potential actions, which are dangerous. The *Action Sheet* consists of an optional title and one or more buttons. Each button is dedicated to an action to take.

The *Action Sheet* is not suggested to use on large screens such as iPad, instead use *popover*.

The following table shows how to use *Action Sheet* in Framework7:

S. No.	Action Sheet types & Description
1	Create and open Action Sheet It is the dynamic element, which can be created and opened using JavaScript.
2	Convert Action Sheet to Popover You can use the extended <i>Action Sheet</i> syntax to automatically convert <i>Action Sheet</i> to popover.
3	Close Action Sheet The <i>Action Sheet</i> can be closed manually by using the appropriate app's method.
4	Action Sheet Events To detect how a user interacts with the <i>Action Sheet</i> , you can use <i>Action Sheet</i> events.
5	Action Sheet Template You can customize <i>Action Sheet</i> template by passing <i>modalActionsTemplate</i> parameter on App initialization.
6	Action Sheet To Popover Template Whenever you use <i>Action Sheet</i> conversion to popover, you can customize popover template by passing <i>modalActionsToPopoverTemplate</i> parameter on App initialization.

Framework7 – Create & Open Action Sheet

Description

Since *Action Sheet* is a dynamic element, it could be created and opened using JavaScript only. The related app's methods to create *Action Sheet* are listed below:

- **myApp.actions(groups)** - It is used to create and open *Action Sheet* with specified number of buttons groups.
- **myApp.actions(buttons)** - It is used to create and open *Action Sheet* with a group and defined number of buttons.
 - **groups** - It is an *array of groups* where each group containing *array of buttons*.
 - **buttons** - It is an *array of buttons* wherein there will be *one group*.

This method will return dynamically created HTML Element of the *Action Sheet*.

Every button in an array of buttons must be presented as *Object* with the button parameters as listed in the following table:

S. No.	Parameter & Description	Type	Default
--------	-------------------------	------	---------

1	text It is the String with Button's text.	string	
2	bold It is optional parameter, which makes the button's text <i>bolder</i> when set to <i>true</i> .	boolean	false
3	color It is an optional parameter and it is one of the 10 default button color.	string	
4	bg It is an optional parameter and it is one of the 10 default button background color.	string	
5	label It is an optional parameter and if is set to <i>true</i> then, it will be the title instead of button.	boolean	true
6	disabled It is an optional parameter and if you want to disable a button then, set this parameter to <i>true</i> .	boolean	false
7	onClick It is an optional parameter and is the callback function which will be executed whenever the user clicks this button.	function	

Example

The following example demonstrates the use of Action Sheet in Framework7, which displays one group and specified amount of buttons groups when you click on the links:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Create and open Action Sheet</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
```



```

<body>
  <div class="views">
    <div class="view view-main">
      <div class="pages">
        <div data-page="home" class="page navbar-fixed">
          <div class="navbar">
            <div class="navbar-inner">
              <div class="left"> </div>
              <div class="center">Action Sheet</div>
              <div class="right"> </div>
            </div>
          </div>
          <div class="page-content">
            <div class="content-block">
              <p><a href="#" class="ac-1"><b>One group</b> with
<b>three buttons</b></a></p>
              <p><a href="#" class="ac-2"><b>One group</b> with
<b>title</b> and <b>three buttons</b></a></p>
              <p><a href="#" class="ac-3"><b>Two groups</b></a></p>
              <p><a href="#" class="ac-4"><b>Three groups</b></a></p>
              <p><a href="#" class="ac-5">With <b>callbacks</b> on
click</a></p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
  <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
  <script>
    var myApp = new Framework7();
    var $$ = Dom7;
    $$('.ac-1').on('click', function () {
    var buttons = [
      {
        text: 'Button1',
        color: 'green'

```

```
    },
    {
      text: 'Button2',
      color: 'green'
    },
    {
      text: 'Cancel',
      color: 'red'
    },
  ];
myApp.actions(buttons);
});
$('.ac-2').on('click', function () {
var buttons = [
  {
    text: 'Some Text',
    label: true
  },
  {
    text: 'Button1',
    color: 'pink'
  },
  {
    text: 'Button2',
    color: 'pink'
  },
  {
    text: 'Cancel',
    color: 'green'
  },
];
myApp.actions(buttons);
});
$('.ac-3').on('click', function () {
var buttons1 = [
  {
    text: 'Some Text',
```

```
        label: true
    },
    {
        text: 'Button1',
        bold: true
    },
    {
text: 'Button2',
bold: true
    },
    {
text: 'Button3',
bold: true
    },
    {
        text: 'Button4',
    }
];
var buttons2 = [
    {
        text: 'Cancel',
        color: 'red'
    }
];
var groups = [buttons1, buttons2];
myApp.actions(groups);
});

$$('.ac-4').on('click', function () {
var buttons1 = [
    {
        text: 'Share',
        label: true
    },
    {
        text: 'Email',
    },

```

```
    {
      text: 'Message',
    }
  ];
  var buttons2 = [
    {
      text: 'Social Networks',
      label: true
    },
    {
      text: 'Facebook',
    },
    {
      text: 'YouTube',
    }
  ];
  var buttons3 = [
    {
      text: 'Cancel',
      color: 'red'
    }
  ];
  var groups = [buttons1, buttons2, buttons3];
  myApp.actions(groups);
});

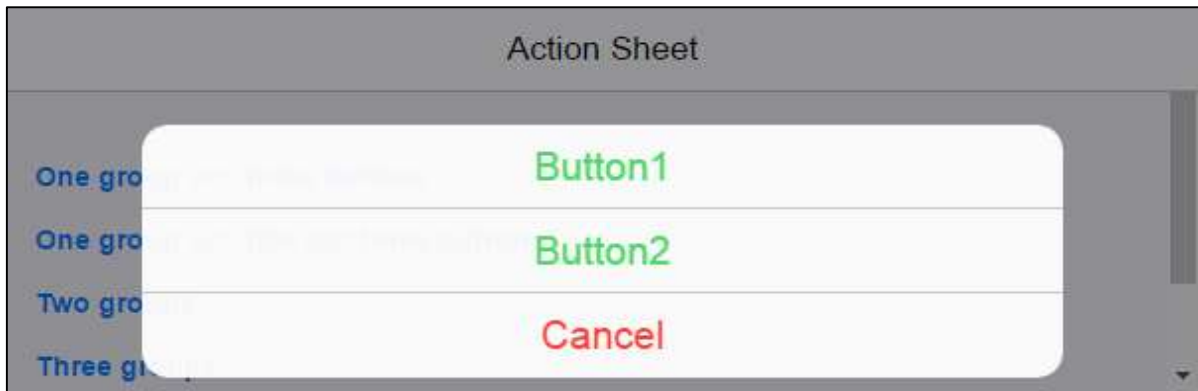
$('.ac-5').on('click', function () {
  var buttons = [
    {
      text: 'Callback Button1',
      onClick: function () {
        myApp.alert('Callback Button1 clicked');
      }
    },
    {
      text: 'Callback Button2',
      onClick: function () {
```

```
        myApp.alert('Callback Button2 clicked');
    }
},
{
    text: 'Cancel',
    color: 'red',
    onClick: function () {
        myApp.alert('Cancel clicked');
    }
},
];
myApp.actions(buttons);
});
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **actionsheet_create_open.html** file in your server root folder.
- Open this HTML file as http://localhost/actionsheet_create_open.html and the output is displayed as shown below.
- When you click on the first option, action sheet is created with three buttons in one group.
- When you click the second option, action sheet is created with three buttons along with a title in one group.
- When you click the third option, two groups are created and when you click the fourth option, three groups are created.
- In the last option, a callback function is executed on clicking the options.



Framework7 – Convert Action Sheet To Popover

Description

Popover could not be used on phones (iPhones) and Action Sheet on tablets, so you can use extended ActionSheet syntax, which will convert Action Sheet to popover automatically on tablets. You can make use of methods as shown below:

- **myApp.actions(target, groups)** - It is used to create and open Action Sheet (or popover on tablets) with defined number of buttons groups.

Or

- **myApp.actions(target, buttons)** - It is used to create and open Action Sheet (or popover on tablets) with a group and buttons.
 - **target** - It is a *HTMLElement* or *string (with CSS Selector)* of target element. It is a required parameter.
 - **groups** - It is an *array of groups*, where each group has an array of buttons.
 - **buttons** - It is an *array of buttons*, wherein there will be *one group*.

Example

The following example demonstrates how to convert Action Sheet to Popover in the Framework7, which converts the action sheet to popover when you click on the link:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Convert Action Sheet to Popover</title>
```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main">
            <div class="pages">
                <div data-page="home" class="page navbar-fixed">
                    <div class="navbar">
                        <div class="navbar-inner">
                            <div class="left"> </div>
                            <div class="center">Action Sheet</div>
                            <div class="right"> </div>
                        </div>
                    </div>
                    <div class="page-content">
                        <div class="content-block">
                            <p><a href="#" class="ac-1">Convert Action Sheet to
Popover</a></p>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
    <script>
        var myApp = new Framework7();
        var $$ = Dom7;
        $$('.ac-1').on('click', function (e) {
            var target = this;
            var buttons = [
                {
                    text: 'Button 1'

```

```

    },
    {
        text: 'Button 2'
    }
];
myApp.actions(target, buttons);
});

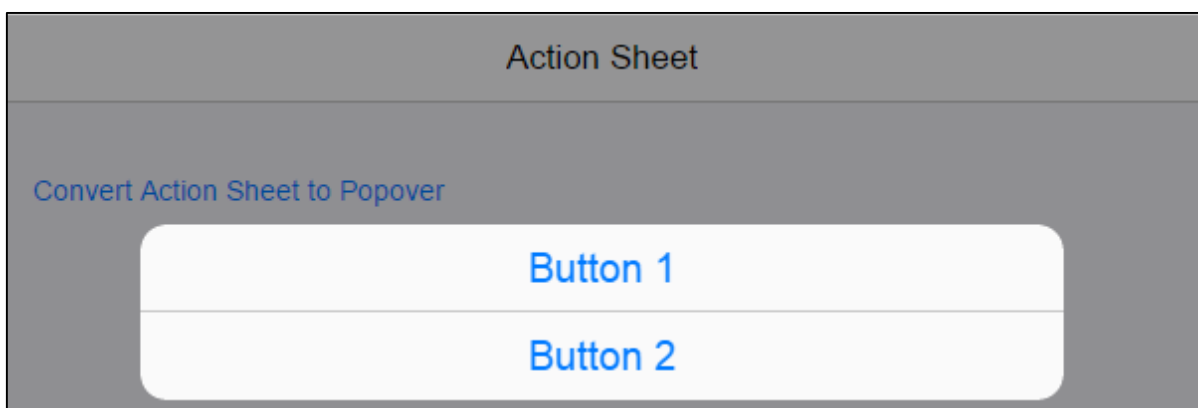
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **actionsheet_convert_to_popover.html** file in your server root folder.
- Open this HTML file as http://localhost/actionsheet_convert_to_popover.html and the output is displayed as shown below.
- Click on the option to convert the action sheet to popover automatically on tablets. When clicked, a window opens with two options as shown below.



Framework7 – Close Action Sheet

Description

By default, the *Action Sheet* will be closed automatically when the following condition occurs:

- When the *Action Sheet* buttons are clicked.
- When you click outside of the *Action Sheet's* overlay window.

The *Action Sheet* can be closed manually by using the appropriate app's method as shown below:

myApp.closeModal(actionSheet) - It is used to close popup. It takes *actionSheet* as optional parameter, which is an *HTML Element* or *string (with CSS Selector)* of the *Action Sheet* to be closed. If this parameter is not specified then any opened *Action Sheet* will be closed.

Framework7 – Action Sheet Events

Description

To detect how a user interacts with the *Action Sheet*, you can use *Action Sheet* events as shown in following table:

S. No.	Events & Description	Target
1	open When the <i>Action Sheet</i> begins its opening animation, this event will be triggered.	Action Sheet Element <div class="actions-model">
2	opened After the <i>Action Sheet</i> completes its opening animation, this event will be triggered.	Action Sheet Element <div class="actions-model">
3	close When the <i>Action Sheet</i> begins its closing animation, this event will be triggered.	Action Sheet Element <div class="actions-model">
4	closed After the <i>Action Sheet</i> completes its closing animation, this event will be triggered.	Action Sheet Element <div class="actions-model">

Framework7 – Action Sheet Template

Description

The *modalActionsTemplate* is a parameter, which can be passed on app initialization if you want to customize the *Action Sheet* template. You can customize *Action Sheet* template by passing *modalActionsTemplate* parameter on App initialization. The parameter will accept *Template7* formatted HTML string, which will be compiled by *Template7* with the *groups* context passed. The template will look as shown below:

```
<!-- This template will equal to default layout -->
<div class="actions-modal">
  <!-- this is a single group -->
  {{#each this}}
    <div class="actions-modal-group">
      <!-- this represents a single button -->
```

```

    {{#each this}}
      {{#if label}}
        <span class="actions-modal-label">{{text}}</span>
      {{else}}
        <div class="actions-modal-button {{#if color}}color-
{{color}}{/if}} {{#if bold}}actions-modal-button-bold{/if}}">{{text}}</div>
      {{/if}}
    {{/each}}
  </div>
{{/each}}
</div>

```

Framework7 – Action Sheet To Popover Template

Description

Whenever you use *Action Sheet* conversation to popover, you can customize the popover template by passing *modalActionsToPopoverTemplate* parameter on App initialization. The parameter will accept *Template7* formatted HTML string, which will be compiled by *Template7* with the *groups* context passed.

The default template will look like as shown below:

```

<div class="popover actions-popover">
  <div class="popover-inner">
    {{#each this}}
      <div class="list-block">
        <ul>
          {{#each this}}
            {{#if label}}
              <li class="actions-popover-label {{#if color}}color-{{color}}{/if}}
{{#if bold}}actions-popover-bold{/if}}">{{text}}</li>
            {{else}}
              <li><a href="#" class="item-link list-button {{#if color}}color-
{{color}}{/if}} {{#if bg}}bg-{{bg}}{/if}} {{#if bold}}actions-popover-
bold{/if}} {{#if disabled}}disabled{/if}}">{{ some text}}</a></li> {{/if}}
          {{/each}}
        </ul>
      </div>
    {{/each}}
  </div>

```

```
</div>
```

Framework7 – Overlay Login Screen

Description

Modal is a small window that displays the content from separate source without leaving the parent window.

There are two ways to open and close the login screen in Framework7 as specified in the following table:

S. No.	Type & Description
1	From HTML You can open and close the login screen using Framework7 classes.
2	Using JavaScript You can open and close the login screen using JavaScript app methods.

Framework7 – Login Screen from HTML

Description

You can open and close the login screen using *open-login-screen* and *close-login-screen* Framework7 classes.

Example

The following example displays the open and close login screens from HTML in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Login Screen from HTML</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
```

```

</head>
<body>
  <div class="views">
    <div class="view view-main">
      <div class="navbar">
        <div class="navbar-inner">
          <div class="center sliding">Login Screen from HTML</div>
        </div>
      </div>
      <div class="pages">
        <div data-page="index" class="page navbar-fixed">
          <div class="page-content">
            <div class="content-block">
              <p><a href="#" class="open-login-screen">Click here to
open Login Screen</a></p>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div class="login-screen">
      <div class="view">
        <div class="page">
          <div class="page-content login-screen-content">
            <div class="login-screen-title">Login Here</div>
            <form>
              <div class="list-block">
                <ul>
                  <li class="item-content">
                    <div class="item-inner">
                      <div class="item-title label">Username</div>
                      <div class="item-input">
                        <input type="text" name="username"
placeholder="Enter the username">
                      </div>
                    </div>
                  </li>
                </ul>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </body>

```

```

        <li class="item-content">
            <div class="item-inner">
                <div class="item-title label">Password</div>
                <div class="item-input">
                    <input type="password" name="password"
placeholder="Enter the password">
                </div>
            </div>
        </li>
    </ul>
</div>
<div class="list-block">
    <ul>
        <li><a href="#" class="item-link list-button">Log
In</a></li>
    </ul>
    <div class="list-block-label">
        <p><a href="#" class="close-login-screen">Click here
to close Login Screen</a></p>
    </div>
</div>
</form>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // Here you can initialize the app
    var myApp = new Framework7();
    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;
    // Add the view
    var mainView = myApp.addView('.view-main', {
    // enable the dynamic navbar for this view:
    dynamicNavbar: true
    });

```

```

    </script>
  </body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **login_screen_from_html.html** file in your server root folder.
- Open this HTML file as http://localhost/login_screen_from_html.html and the output is displayed as shown below.
- The output shows the use of `open-login-screen` and `close-login-screen` classes to open and close the login screen from HTML. When you click on the link, a login wizard gets displayed where you can enter username and password and click on the login button to login.
- Click on close login screen option to close the login screen.

Framework7 – Login Screen from JavaScript

Description

You can open and close the login screen using `loginScreen()` and `closeModal(loginScreen)` app methods in the Framework7.

Example

The following example displays the open and close login screens from JavaScript in Framework7:

```

<!DOCTYPE html>
<html>
  <head>

```

```

    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Login Screen from HTML</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main">
            <div class="navbar">
                <div class="navbar-inner">
                    <div class="center sliding">Login Screen using
JavaScript</div>
                </div>
            </div>
            <div class="pages">
                <div data-page="index" class="page navbar-fixed">
                    <div class="page-content">
                        <div class="content-block">
                            <p><a href="#" class="login_screen_js">Click here to
open Login Screen</a></p>
                        </div>
                    </div>
                </div>
            </div>
            <div class="login-screen">
                <div class="view">
                    <div class="page">
                        <div class="page-content login-screen-content">
                            <div class="login-screen-title">Login Here</div>
                            <form>
                                <div class="list-block">

```

```

        <ul>
            <li class="item-content">
                <div class="item-inner">
                    <div class="item-title label">Username</div>
                    <div class="item-input">
                        <input type="text" name="username"
placeholder="Enter the username">
                    </div>
                </div>
            </li>
            <li class="item-content">
                <div class="item-inner">
                    <div class="item-title label">Password</div>
                    <div class="item-input">
                        <input type="password" name="password"
placeholder="Enter the password">
                    </div>
                </div>
            </li>
        </ul>
    </div>
    <div class="list-block">
        <ul>
            <li><a href="#" class="item-link list-button">Log
In</a></li>
        </ul>
        <div class="list-block-label">
            <p><a href="#" class="close-login-screen">Click here
to close Login Screen</a></p>
        </div>
    </div>
</form>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>

```



```

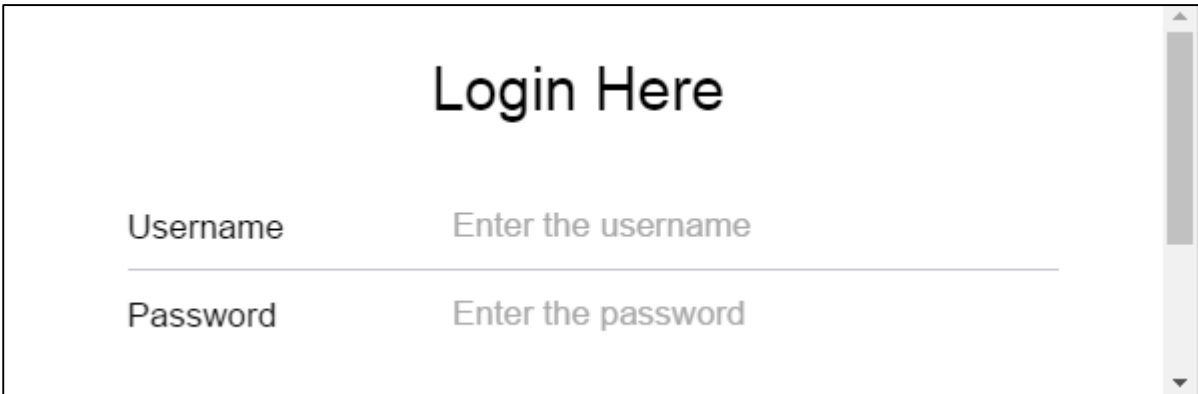
<script>
  // Here you can initialize the app
  var myApp = new Framework7();
  // If your using custom DOM library, then save it to $$ variable
  var $$ = Dom7;
  // Add the view
  var mainView = myApp.addView('.view-main', {
    // enable the dynamic navbar for this view:
    dynamicNavbar: true
  });
  $$('.login_screen_js').on('click', function () {
    myApp.loginScreen();
  });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **login_screen_from_javascript.html** file in your server root folder.
- Open this HTML file as http://localhost/login_screen_from_javascript.html and the output is displayed as shown below.
- The output shows the use of `loginScreen()` and `closeModal(loginScreen)` methods to open and close the login screen from javascript. When you click on the link, a login wizard gets displayed where you can enter username and password and click on the login button to login.
- Click on close login screen option to close the login screen.



The screenshot shows a login screen with the following elements:

- Title:** Login Here
- Username field:** Labeled "Username" with a placeholder "Enter the username".
- Password field:** Labeled "Password" with a placeholder "Enter the password".

Login Screen Events

The following table shows a list of Modal events:

S. No.	Event	Target & Description
1	open	Login Screen Element When you open an animation, this event will get fired.
2	opened	Login Screen Element When the opening of an animation is completed, this event will get fired.
3	close	Login Screen Element When you close an animation, this event will get fired.
4	closed	Login Screen Element When the closing of an animation is completed, this event will get fired.

You can create a login screen in two ways as specified in the table below:

S. No.	Type & Description
1	Start App With Login Screen You can create the login screen on applicaton's first screen by using the <i>modal-in</i> class.
2	Embedded Login Screen To embed a login screen, you can insert the login screen into the page.

Framework7 – Start App With Login Screen

Description

You can create the login screen on applicaton's first screen by using the *modal-in* class.

Example

The following example demonstrates the use of start app with login screen in the Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
```

```

    <title>Start App With Login Screen</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
  <div class="views">
    <div class="view view-main">
      <div class="navbar">
        <div class="navbar-inner">
          <div class="center sliding">Start App With Login Screen</div>
        </div>
      </div>
      <div class="pages">
        <div data-page="index" class="page navbar-fixed">
          <div class="page-content">
            <div class="content-block">
              <p><a href="#" class="open-login-screen">Click here to
open Login Screen</a></p>
            </div>
          </div>
        </div>
      </div>
    </div>
  <div class="login-screen modal-in">
    <div class="view">
      <div class="page">
        <div class="page-content login-screen-content">
          <div class="login-screen-title">Login Here</div>
          <form>
            <div class="list-block">
              <ul>
                <li class="item-content">
                  <div class="item-inner">
                    <div class="item-title label">Username</div>

```

```

        <div class="item-input">
            <input type="text" name="username"
placeholder="Enter the username">
        </div>
    </div>
</li>
<li class="item-content">
    <div class="item-inner">
        <div class="item-title label">Password</div>
        <div class="item-input">
            <input type="password" name="password"
placeholder="Enter the password">
        </div>
    </div>
</li>
</ul>
</div>
<div class="list-block">
    <ul>
        <li><a href="#" class="item-link list-button">Log
In</a></li>
    </ul>
    <div class="list-block-label">
        <p><a href="#" class="close-login-screen">Click here
to close Login Screen</a></p>
    </div>
</div>
</form>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // Here you can initialize the app
    var myApp = new Framework7();
    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;

```

```

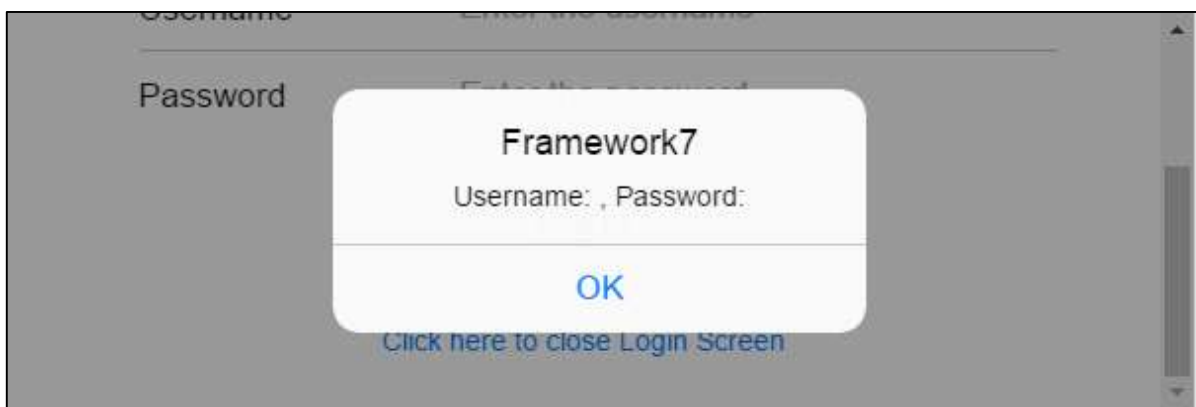
// Add the view
var mainView = myApp.addView('.view-main', {
// enable the dynamic navbar for this view:
  dynamicNavbar: true
});
$$('.login-screen .list-button').on('click', function () {
  var uname = $$('.login-screen input[name="username"]').val();
  var pwd = $$('.login-screen input[name="password"]').val();
  myApp.alert('Username: ' + uname + ', Password: ' + pwd, function
() {
    myApp.closeModal('.login-screen');
  });
});
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **login_screen_start_app.html** file in your server root folder.
- Open this HTML file as http://localhost/login_screen_start_app.html and the output is displayed as shown below.
- When you enter username, password and click on the "Log in" link, the login wizard is displayed with the entered username and password.
- You can close the login screen by clicking on the Close login link.



Framework7 – Embedded Login Screen

Description

To embed a login screen, you can insert the login screen into the page.

Example

The following example demonstrates the use of embedded login screen in the Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Embedded Login Screen</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="center sliding">Embedded Login Screen</div>
          </div>
        </div>
        <div class="pages">
          <div data-page="index" class="page navbar-fixed">
            <div class="page-content">
              <div class="content-block">
                <p><a href="/framework7/src/login_screen_page.html"
class="open-login-screen">Click here to open Login Screen</a></p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

        </div>
    </div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // Here you can initialize the app
    var myApp = new Framework7();
    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;
    // Add the view
    var mainView = myApp.addView('.view-main', {
        // enable the dynamic navbar for this view:
        dynamicNavbar: true
    });
    myApp.onPageInit('login-screen', function (page) {
        var myVal = $$ (page.container);
        myVal.find('.list-button').on('click', function () {
            var uname = myVal.find('input[name="username"]').val();
            var pwd = myVal.find('input[name="password"]').val();
            myApp.alert('Username: ' + uname + ', Password: ' + pwd,
function () {
                mainView.goBack();
            });
        });
    });
</script>
</body>
</html>

```

Let us create one more HTML page that creates login screen inside of page:

login_screen_page.html

```

<div data-page="login-screen" class="page no-navbar no-toolbar no-swipeback">
    <div class="page-content login-screen-content">
        <div class="login-screen-title">Login Here</div>
        <form>
            <div class="list-block">

```

```

        <ul>
            <li class="item-content">
                <div class="item-inner">
                    <div class="item-title label">Username</div>
                    <div class="item-input">
                        <input type="text" name="username" placeholder="Enter
the username">
                    </div>
                </div>
            </li>
            <li class="item-content">
                <div class="item-inner">
                    <div class="item-title label">Password</div>
                    <div class="item-input">
                        <input type="password" name="password"
placeholder="Enter the password">
                    </div>
                </div>
            </li>
        </ul>
    </div>
    <div class="list-block">
        <ul>
            <li><a href="#" class="item-link list-button">Log In</a></li>
        </ul>
    </div>
</form>
</div>
</div>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **login_screen_embedded.html** file in your server root folder.
- Open this HTML file as `http://localhost/login_screen_embedded.html` and the output is displayed as shown below.
- You can insert the login screen into the page where when clicked on the link, a login page opens where you can enter the username and password and click on the log in button. It displays a wizard with the entered username and password.

Login Here

Username

Password

Framework7 – Overlay Picker Modal

Description

Picker modal is used to pick some custom content, which is similar to a calendar picker. The picker modal can be created by using the *picker-modal* class.

You can open and close the picker modal in two ways as specified in the following table:

S. No.	Type & Description
1	From HTML You can open and close the picker modal by using classes and data attributes.
2	Using JavaScript You can open and close the picker modal by using the JavaScript code.

Framework7 – Overlay Picker Modal from HTML

Description

You can open and close the picker modal by using the classes and data attributes. Use the *open-picker* class to open the picker and *close-picker* class to close the picker to any HTML element.

Example

The following example displays the open and close the picker modal from HTML in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
```

```

    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Picker Modal from HTML</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main">
            <div class="pages">
                <div data-page="home" class="page navbar-fixed">
                    <div class="navbar">
                        <div class="navbar-inner">
                            <div class="left"> </div>
                            <div class="center">Picker Modal from HTML</div>
                            <div class="right"> <a href="#" data-picker=".my_page"
class="link open-picker">Picker</a></div>
                        </div>
                    </div>
                    <div class="page-content">
                        <div class="content-block">
                            <p><a href="#" data-picker=".my_page" class="open-
picker">Open the Page</a></p>
                            <p><a href="#" data-picker=".my_page" class="close-
picker">Close the Page</a></p>
                        </div>
                    </div>
                </div>
            </div>
            <div class="picker-modal my_page">
                <div class="toolbar">
                    <div class="toolbar-inner">
                        <div class="left"></div>
                        <div class="right"><a href="#" class="close-picker">Ok</a></div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

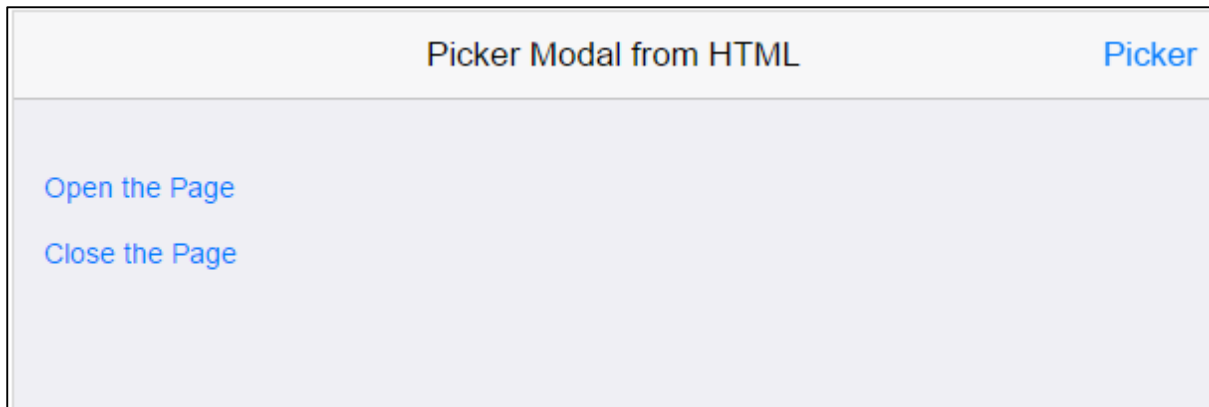
    <div class="picker-modal-inner">
      <div class="content-block">
        <h4>Welcome to Framework7!!!</h4>
        <p>Framework7 - is a free and open source mobile HTML framework
to develop hybrid mobile apps or web apps with iOS & Android native look and
feel. It is also an indispensable prototyping apps tool to show working app
prototype as soon as possible in case you need to. </p>
      </div>
    </div>
  </div>
  <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
  <script>
    // Here you can initialize the app
    var myApp = new Framework7();
    //If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;
  </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **picker_modal_from_html.html** file in your server root folder.
- Open this HTML file as http://localhost/picker_modal_from_html.html and the output is displayed as shown below.
- Click on 'Open the page' link to open the picker modal. The page can also be opened by clicking on link picker displayed on the top right.
- The picker modal can be closed by clicking on 'Close the Page' option. The picker modal is opened and closed to any HTML element.



Framework7 – Overlay Picker Modal using JavaScript

Description

You can open and close the picker modal by using the JavaScript code. You can use the *pickerModal(picker)* method to open the picker modal and *closeModal(picker)* method to close the picker modal.

Example

The following example displays the open and close the picker modal using JavaScript in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Picker Modal using JavaScript</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
```

```

        <div class="navbar">
            <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Picker Modal using JavaScript</div>
                <div class="right"> <a href="#" class="link open-
picker">Picker</a></div>
            </div>
        </div>
        <div class="page-content">
            <div class="content-block">
                <p><a href="#" class="open-picker">Open the
Page</a></p>
                <p><a href="#" class="close-picker">Close the
Page</a></p>
            </div>
        </div>
    </div>
</div>
<div class="picker-modal picker_val">
    <div class="toolbar">
        <div class="toolbar-inner">
            <div class="left"></div>
            <div class="right"><a href="#" class="close-picker">Ok</a></div>
        </div>
    </div>
    <div class="picker-modal-inner">
        <div class="content-block">
            <h4>Welcome to Framework7!!!</h4>
            <p>Framework7 - is a free and open source mobile HTML framework
to develop hybrid mobile apps or web apps with iOS & Android native look and
feel. It is also an indispensable prototyping apps tool to show working app
prototype as soon as possible in case you need to. </p>
        </div>
    </div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>

```

```

<script>
  // Here you can initialize the app
  var myApp = new Framework7();

  //If your using custom DOM library, then save it to $$ variable
  var $$ = Dom7;

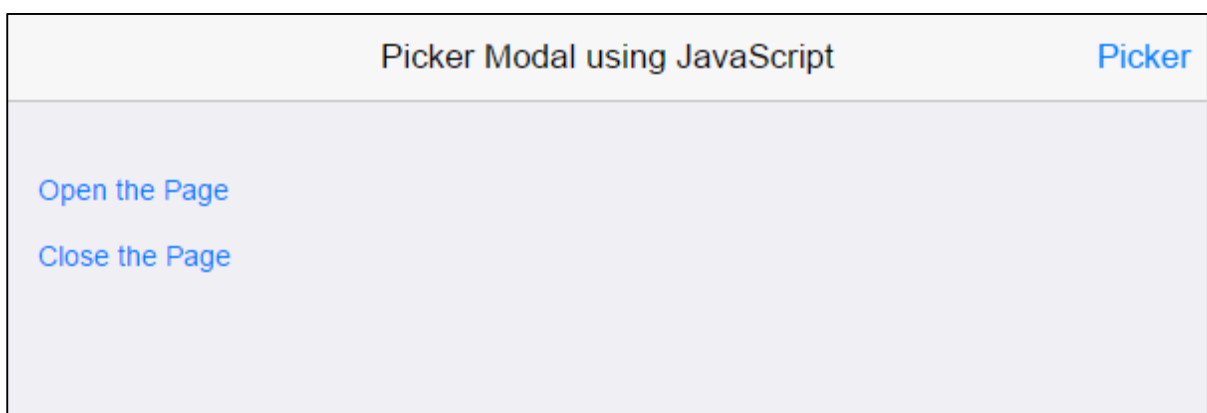
  $$('.open-picker').on('click', function () {
    myApp.pickerModal('.picker_val')
  });
  $$('.close-picker').on('click', function () {
    myApp.closeModal('.picker_val')
  });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **picker_modal_using_javascript.html** file in your server root folder.
- Open this HTML file as http://localhost/picker_modal_using_javascript.html and the output is displayed as shown below.
- Click on 'Open the Page' link to open the picker modal. The page can also be opened by clicking on the link picker displayed on the top right.
- The picker modal can be closed by clicking on the 'Close the Page' option. The picker modal is opened and closed using javascript code.



Popup Events

S. No.	Event	Target & Description
1	open	Picker Element When you open an animation, this event will get fired.
2	opened	Picker Element When the opening of an animation is completed, this event will get fired.
3	close	Picker Element When you close an animation, this event will get fired.
4	closed	Picker Element When the closing of an animation is completed, this event will get fired.

You can also create a dynamic picker modal by using it's HTML to App methods. Here is an [example](#).

Framework7 – Overlay Dynamic Picker Modal

Description

You can also create a dynamic picker modal by using it's HTML to App methods. It uses two parameters:

- **pickerHTML:** It contains string element of picker modal.
- **removeOnClose:** It includes boolean value, which will be removed from DOM when you close the Picker. By default, it includes the true value.

Example

The following example demonstrates the use of dynamic picker modal in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Dynamic Picker Modal</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ios.colors.min.css">
</head>
<body>
  <div class="views">
    <div class="view view-main">
      <div class="navbar">
        <div class="navbar-inner">
          <div class="center sliding">Dynamic Picker Modal</div>
        </div>
      </div>
      <div class="pages">
        <div data-page="index" class="page navbar-fixed">
          <div class="page-content">
            <div class="content-block">
              <p><a href="#" class="create-modal-picker">Open
Picker</a></p>
              <p><a href="#" class="close-picker">Close
Picker</a></p>
            </div>
          </div>
        </div>
      </div>
    </div>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.js"></script>
    <script>
      // Here you can initialize the app
      var myApp = new Framework7();
      // If your using custom DOM library, then save it to $$ variable
      var $$ = Dom7;
      // Add the view
      var mainView = myApp.addView('.view-main', {
      // enable the dynamic navbar for this view:
        dynamicNavbar: true
      });

```



```

    $$('.create-modal-picker').on('click', function () {
        //Here you can check whether we have already opened picker or not
        if ($$('.picker-modal.modal-in').length > 0) {
            myApp.closeModal('.picker-modal.modal-in');
        }
        myApp.pickerModal(
            '<div class="picker-modal">' +
            '<div class="toolbar">' +
            '<div class="toolbar-inner">' +
            '<div class="left"></div>' +
            '<div class="right"><a href="#" class="close-picker">Ok</a></div>' +
            '</div>' +
            '</div>' +
            '<div class="picker-modal-inner">' +
            '<div class="content-block">' +
            '<p>Framework7 - is a free and open source mobile HTML framework to
            develop hybrid mobile apps or web apps with iOS & Android native look and feel.
            It is also an indispensable prototyping apps tool to show working app prototype
            as soon as possible in case you need to.</p>' +
            '</div>' +
            '</div>' +
            '</div>'
        )
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **picker_modal_dynamic.html** file in your server root folder.
- Open this HTML file as http://localhost/picker_modal_dynamic.html and the output is displayed as shown below.

Dynamic Picker Modal

Open Picker

Close Picker

12. Framework7 – Preloaders

Description

Preloader in Framework7 is made with Scalable Vector Graphic (SVG) and animated with CSS, which makes it easily resizable. Preloader is available in two colors:

- the default is light background
- another one is dark background

You can use the *preloader* class in your HTML as shown below:

Example

The following example demonstrates the use of preloader in the Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Panel Events</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Framework7 Preloader</div>
                <div class="right"> </div>
```

```

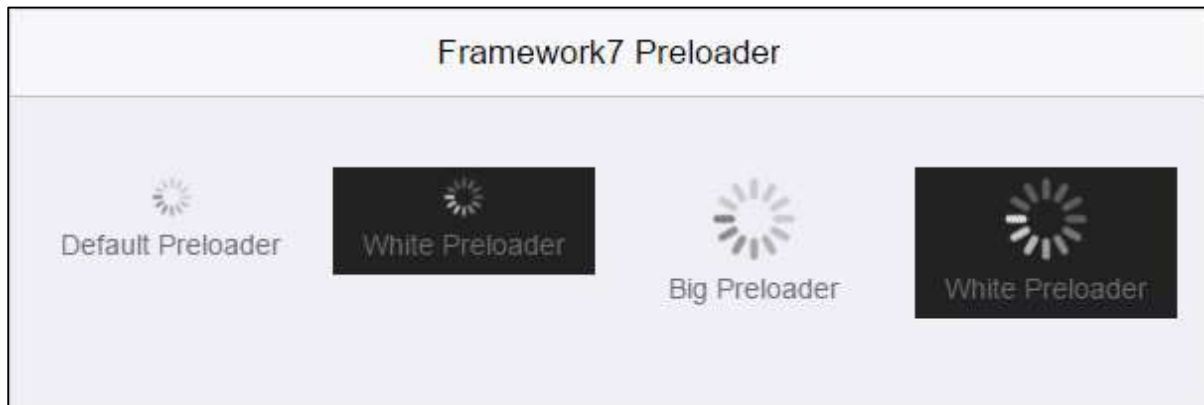
        </div>
    </div>
    <div class="page-content">
        <div class="content-block row">
            <div class="col-25"><span
class="preloader"></span><br>Default Preloader</div>
            <div class="col-25 col-dark"><span class="preloader
preloader-white"></span><br>White Preloader</div>
            <div class="col-25"><span style="width:42px;
height:42px" class="preloader"></span><br>Big Preloader</div>
            <div class="col-25 col-dark"><span style="width:42px;
height:42px" class="preloader preloader-white"></span><br>White Preloader</div>
        </div>
    </div>
</div>
</div>
</div>
<div>
    <style>.col-25{padding:5px;text-align:center;}.col-
dark{background:#222;}</style>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
    <script>
        var myApp = new Framework7();
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **preloader.html** file in your server root folder.
- Open this HTML file as <http://localhost/preloader.html> and the output is displayed as shown below.
- This code displays the preloader indicator, which is made with SVG and animated with CSS.



13. Framework7 – Progress Bar

Description

The progress bars can be used to show loading of assets or progress of a task to the users. You can specify the progress bar by using the *progressbar* class. When the user does not know how long the loading process will be there for the request, you can use *progressbar-infinite* class.

Progress Bar JavaScript API

The progress bar can be used along with the JavaScript API to specify *show*, *hide* and *progress* properties by using the following methods:

S. No.	Methods	Description & Parameters
1	myApp.setProgressbar(container, progress, speed)	<p>It sets the progress bar for the progress of a task.</p> <ul style="list-style-type: none">• container: It is a string or HTML element that defines the container of progress bar element.• progress: It represents in integer format and defines the progress of a task.• speed: It represents in integer format and specifies the duration of the progress of a task.
2	myApp.hideProgressbar(container)	<p>It hides the progress bar.</p> <ul style="list-style-type: none">• container: It is a string or HTML element that defines the container of progress bar element.
3	myApp.showProgressbar(container, progress, color)	<p>It displays the progress bar.</p> <ul style="list-style-type: none">• container: It is a string or HTML element that defines the

		<p>container of progress bar element.</p> <ul style="list-style-type: none"> • progress: It represents in integer format and defines the progress of a task. • speed: It represents in integer format and specifies the duration of the progress of a task.
--	--	---

Example

The following example displays an animated determinate and indeterminate progress bars to indicate activity in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Progress Bar</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
```

```

<div data-page="home" class="page navbar-fixed">
  <div class="navbar">
    <div class="navbar-inner">
      <div class="center">Progress Bar</div>
    </div>
  </div>
  <div class="page-content">
    <div class="content-block-title">Determinate Progress
Bar</div>
    <div class="content-block">
      <div class="content-block-inner">
        <p>Inline determinate progress bar:</p>
        <div class="progressbar-inline">
          <p><span data-progress="10"
class="progressbar"></span></p>
          <p class="buttons-row">
            <a href="#" data-progress="25" class="button
button-raised">25%</a><a href="#" data-progress="50" class="button button-
raised">50%</a><a href="#" data-progress="75" class="button button-
raised">75%</a><a href="#" data-progress="100" class="button button-
raised">100%</a>
          </p>
        </div>
        <p>Loads and hides the determinate progress bar:</p>
        <div class="progressbar-load-hide">
          <p><a href="#" class="button button-raised">Start
Loading</a></p>
        </div>
        <p>Displays the determinate progress bar on top:</p>
        <p class="progressbar-overlay"><a href="#"
class="button button-raised">Start Loading</a></p>
      </div>
    </div>
    <div class="content-block-title">Infinite Progress
Bar</div>
    <div class="content-block">
      <div class="content-block-inner">
        <p>Inline infinite progress bar:</p>
        <p><span class="progressbar-infinite"></span></p>
        <p>Displays the infinite progress bar in multiple
colors:</p>

```



```

        <p><span class="progressbar-infinite color-
multi"></span></p>
        <p>Displays the infinite progress bar on top:</p>
        <p class="progressbar-infinite-overlay"><a href="#"
class="button button-raised">Start Loading</a></p>
        <p>Displays the infinite progress bar in multiple
colors on top:</p>
        <p class="progressbar-infinite-multi-overlay"><a
href="#" class="button button-raised">Start Loading</a></p>
    </div>
</div>
<div class="content-block-title">Different types of
colored progress bars:</div>
<div class="list-block">
    <ul>
        <li class="item-content">
            <div class="item-inner">
                <div data-progress="5" class="progressbar
color-red"></div>
            </div>
        </li>
        <li class="item-content">
            <div class="item-inner">
                <div data-progress="10" class="progressbar
color-pink"></div>
            </div>
        </li>
        <li class="item-content">
            <div class="item-inner">
                <div data-progress="20" class="progressbar
color-deeppurple"></div>
            </div>
        </li>
        <li class="item-content">
            <div class="item-inner">
                <div data-progress="30" class="progressbar
color-blue"></div>
            </div>
        </li>
        <li class="item-content">

```

```

        <div class="item-inner">
            <div data-progress="40" class="progressbar
color-cyan"></div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div data-progress="50" class="progressbar
color-green"></div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div data-progress="60" class="progressbar
color-lime"></div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div data-progress="70" class="progressbar
color-amber"></div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div data-progress="80" class="progressbar
color-deepporange"></div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div data-progress="90" class="progressbar
color-gray"></div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div data-progress="100" class="progressbar
color-black"></div>
        </div>

```

```

        </li>
    </ul>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7({
        material: true
    });
    var $$ = Dom7;

    $$('.progressbar-inline .button').on('click', function () {
        var progress = $$('this').attr('data-progress');
        var progressbar = $$('.progressbar-inline .progressbar');
        myApp.setProgressbar(progressbar, progress);
    });
    $$('.progressbar-load-hide .button').on('click', function () {
        var container = $$('.progressbar-load-hide p:first-child');
        //it doesn't load if another progresbar is loading
        if (container.children('.progressbar').length) return;

        myApp.showProgressbar(container, 0);

        var progress = 0;
        function simulateLoading() {
            setTimeout(function () {
                var progressBefore = progress;
                progress += Math.random() * 20;
                myApp.setProgressbar(container, progress);
                if (progressBefore < 100) {
                    simulateLoading();
                }
            }

```

```

        else myApp.hideProgressbar(container);
    }, Math.random() * 200 + 200);
    }
    simulateLoading();
});
$$('.progressbar-overlay .button').on('click', function () {

    var container = $$('body');
    if (container.children('.progressbar, .progressbar-
infinite').length) return;

    myApp.showProgressbar(container, 0, 'orange');

    var progress = 0;
    function simulateLoading() {
        setTimeout(function () {
            var progressBefore = progress;
            progress += Math.random() * 20;
            myApp.setProgressbar(container, progress);
            if (progressBefore < 100) {
                simulateLoading();
            }
            //hides the progressbar
            else myApp.hideProgressbar(container);
        }, Math.random() * 200 + 200);
    }
    simulateLoading();
});
$$('.progressbar-infinite-overlay .button').on('click', function () {
    var container = $$('body');
    if (container.children('.progressbar, .progressbar-
infinite').length) return;
    myApp.showProgressbar(container, 'yellow');
    setTimeout(function () {
        myApp.hideProgressbar();
    }, 3000);
});

```

```

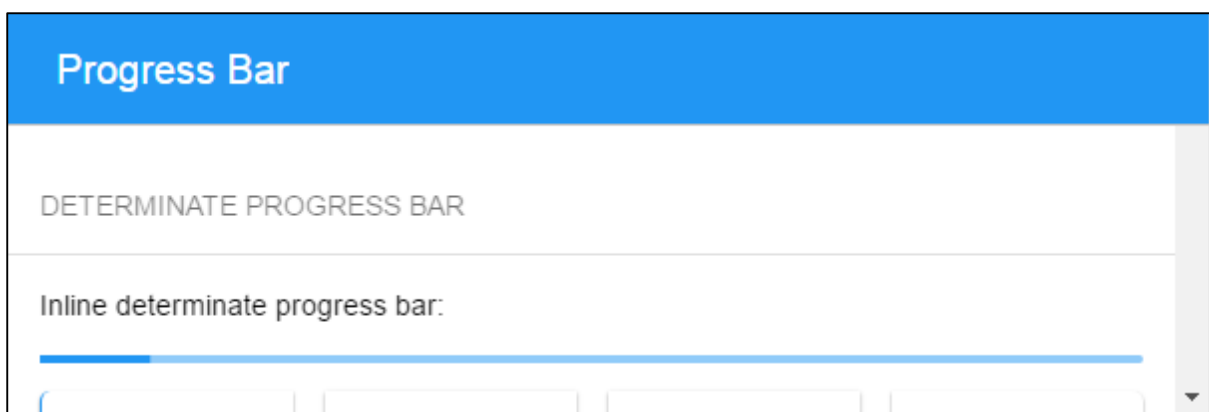
        $$('.progressbar-infinite-multi-overlay .button').on('click', function
    () {
        var container = $$('body');
        if (container.children('.progressbar, .progressbar-
infinite').length) return;
        myApp.showProgressbar(container, 'multi');
        setTimeout(function () {
            myApp.hideProgressbar();
        }, 3000);
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **progress_bar.html** file in your server root folder.
- Open this HTML file as http://localhost/progress_bar.html and the output is displayed as shown below.
- The example displays the progress bar, which indicates how long an operation will take to complete the process and displays the different types of progress bars to indicate activity.



14. Framework7 – List Views

Description

List views are powerful UI components that present data in a scrollable list of multiple rows. The Framework7 provides different types of *List View* to work with your application. The following table lists some of the *Framework7 List View*:

S. No.	Types & Description
1	List View List views are powerful user interface components, which present data in a scrollable list containing multiple rows.
2	Contacts List The contacts list is a type of list view, which can be used to display the list of people contacts.
3	Media List View The Media list view is used to display complicated data structures like products, services, users information.
4	Swipeout The <i>swipeout</i> allows you to reveal hidden menu actions by swiping over the list elements.
5	Sortable List The sortable list is a type of list view, which sorts the list view elements.
6	Virtual List Virtual list a type of list view, which includes lists of large number of data elements without dropping their performance

Framework7 – List Views

Description

List views are powerful user interface components, which presents data in a scrollable list containing multiple rows. List views has many uses as listed below:

- It allows users to link through orderly arranged data.
- It allows the user to select from an indexed list of items.
- It displays detailed information and controls in distinct groupings.
- It presents a list of selectable options.

List View layout

The following table shows common layout of list view:

S. No.	List View types & Description
1	List block It is your list view elements wrapper, which uses <i>list-block</i> class.
2	Link Elements The list view elements can be used as <i>links</i> to other pages and data.
3	List Divider It is a list element with title, which visually divide one list items from another.
4	Grouped Lists You can wrap your list view elements within single list block.
5	Inset List Block By adding <i>inset</i> class to <i>list-block</i> your list blocks can have <i>inset</i> styling.

Framework7 – List Block

Description

The *list block* is your list view elements wrapper, which uses *list-block* class. It is almost the similar as [content block](#) but for list elements. All your list view elements must be wrapped with *list-block* element as shown below:

```
<div class="list-block">
  <ul>
    ... list elements here ...
  </ul>
</div>
```

List block label

The *list block label* can be added at the end of list block as shown below:

```
<div class="list-block">
  <ul>
    ... list elements here ...
  </ul>
  <div class="list-block-label">List block label text</div>
</div>
```

List item/element

A flexible single list element layout is shown below:

```
<li>
  <div class="item-content"> <!-- It is Required and is the main flex
  wrapper. -->
    <div class="item-media"> <!-- Is optional element and is the container
  for media elements. -->
      <i class="icon my-icon"></i>
    </div>
    <div class="item-inner"> <!-- It is Required and is the main flex
  wrapper for item-title and item-after. -->
      <div class="item-title"> <!-- Required and is a single-line list
  item title -->
          List element title
      </div>
      <div class="item-after"> <!-- List item label. Can contain label
  text, badge, switch or button, etc.-->
          List element label
      </div>
    </div>
  </div>
</li>
```

Example

The following example specifies the list view elements wrapper by using the list block in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
  maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Swipe Panel</title>
    <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
  s.min.css">
    <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
  s.colors.min.css">
```



```

</head>
<body>
  <div class="views">
    <div class="view view-main">
      <div class="pages">
        <div data-page="home" class="page navbar-fixed">
          <div class="navbar">
            <div class="navbar-inner">
              <div class="left"> </div>
              <div class="center">List View</div>
              <div class="right"> </div>
            </div>
          </div>
          <div class="page-content">
            <div class="content-block-title">Full Layout</div>
            <div class="list-block">
              <ul>
                <li class="item-content">
                  <div class="item-media"><i class="icon icon-form-
name"></i></div>
                  <div class="item-inner">
                    <div class="item-title">Item title</div>
                    <div class="item-after">Label</div>
                  </div>
                </li>
                <li class="item-content">
                  <div class="item-media"><i class="icon icon-form-
name"></i></div>
                  <div class="item-inner">
                    <div class="item-title">Item with badge</div>
                    <div class="item-after"> <span
class="badge">78</span></div>
                  </div>
                </li>
                <li class="item-content">
                  <div class="item-media"><i class="icon icon-form-
name"></i></div>
                  <div class="item-inner">
                    <div class="item-title">Another item</div>

```

```

        <div class="item-after">Another label</div>
    </div>
</li>
</ul>
<div class="list-block-label">List block label text is
written here</div>
</div>
<div class="content-block-title">Only titles</div>
<div class="list-block">
    <ul>
        <li class="item-content">
            <div class="item-inner">
                <div class="item-title">Item title</div>
            </div>
        </li>
        <li class="item-content">
            <div class="item-inner">
                <div class="item-title">Item with badge</div>
            </div>
        </li>
        <li class="item-content">
            <div class="item-inner">
                <div class="item-title">Another item</div>
            </div>
        </li>
    </ul>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
</script>

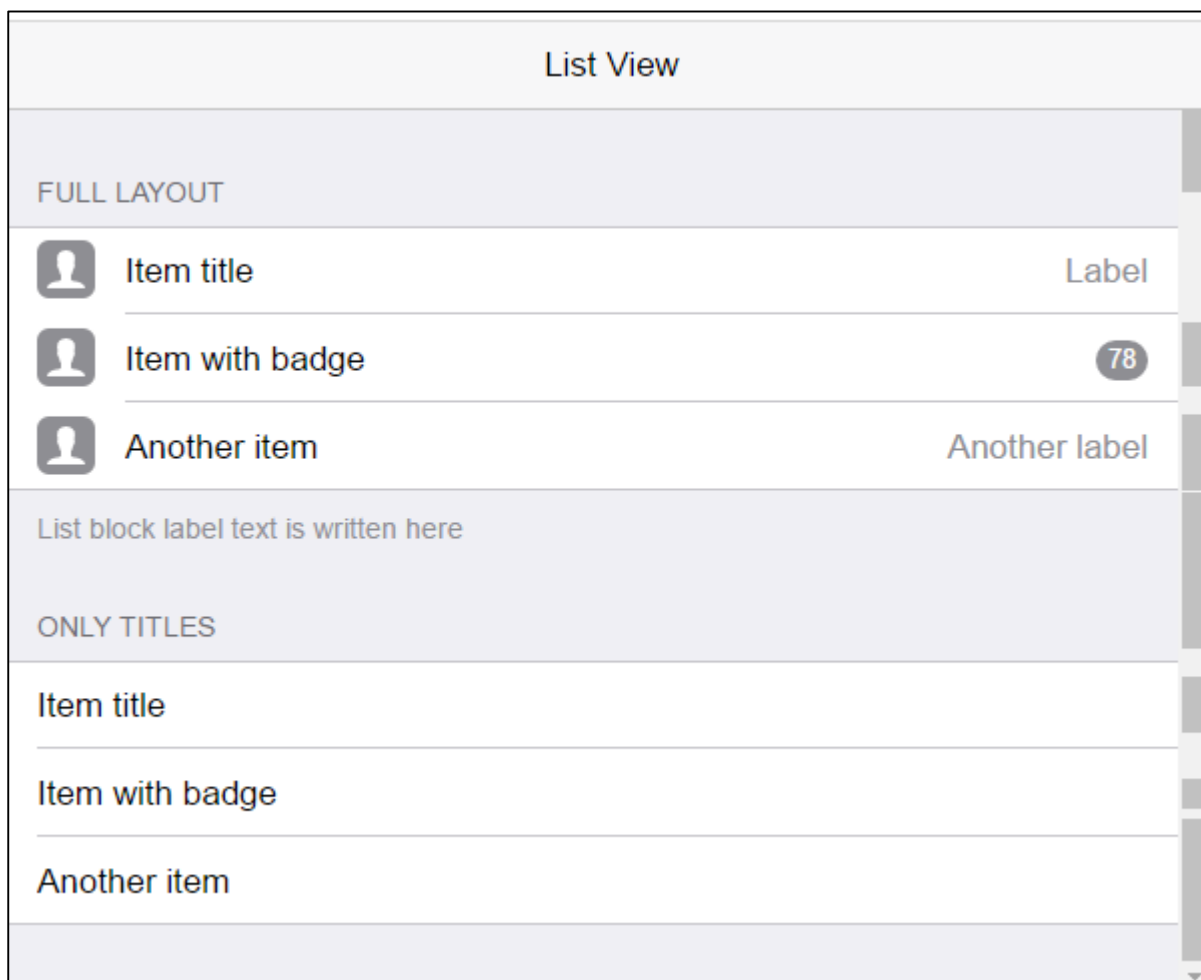
```

```
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **list_views_list_block.html** file in your server root folder.
- Open this HTML file as http://localhost/list_views_list_block.html and the output is displayed as shown below.
- The example displays the list view elements wrapper for list elements, which looks similar to the content block.



Framework7 – Link Elements

Description

The list view elements can be used as *links* to other pages and data. You need to wrap *item-content* with `` element.

Example

The following example demonstrates the use of link elements in Framework7 list view:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Link Elements</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Link Elements</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">
              <div class="content-block-title">Links</div>
              <div class="list-block">
                <ul>
                  <li>
                    <a href="#" class="item-link item-content">
                      <div class="item-media"><i class="icon icon-
form-name"></i></div>
                      <div class="item-inner">
                        <div class="item-title">Item title</div>

```

```

        <div class="item-after">Label</div>
    </div>
</a>
</li>
<li>
    <a href="#" class="item-link item-content">
        <div class="item-media"><i class="icon icon-
form-name"></i></div>
        <div class="item-inner">
            <div class="item-title">Item with
badge</div>
            <div class="item-after"> <span
class="badge">78</span></div>
        </div>
    </a>
</li>
<li>
    <a href="#" class="item-link item-content">
        <div class="item-media"><i class="icon icon-
form-name"></i></div>
        <div class="item-inner">
            <div class="item-title">Another item</div>
            <div class="item-after">Another label</div>
        </div>
    </a>
</li>
</ul>
<div class="list-block-label">List block label text
written here</div>
    </div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();

```

```

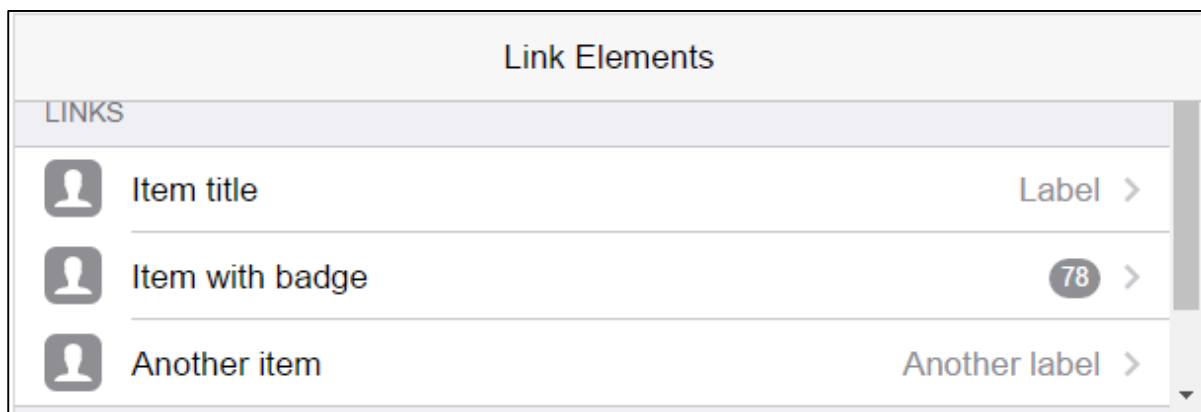
    </script>
  </body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **list_views_link_element.html** file in your server root folder.
- Open this HTML file as http://localhost/list_views_link_element.html and the output is displayed as shown below.
- You can use the list view elements as links () to other pages or data.



Framework7 – List Divider

Description

The list divider is a list element with title, which visually divide one list items from another.

Example

The following example demonstrates the use of list divider that divides list from other list with different items in Framework7 list view:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Link Elements</title>

```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main">
            <div class="pages">
                <div data-page="home" class="page navbar-fixed">
                    <div class="navbar">
                        <div class="navbar-inner">
                            <div class="left"> </div>
                            <div class="center">List Divider</div>
                            <div class="right"> </div>
                        </div>
                    </div>
                    <div class="page-content">
                        <div class="content-block-title">With Divider</div>
                        <div class="list-block">
                            <ul>
                                <li class="item-content">
                                    <div class="item-media"><i class="icon icon-form-
name"></i></div>
                                    <div class="item-inner">
                                        <div class="item-title">Item title</div>
                                        <div class="item-after">Label</div>
                                    </div>
                                </li>
                                <li class="item-content">
                                    <div class="item-media"><i class="icon icon-form-
name"></i></div>
                                    <div class="item-inner">
                                        <div class="item-title">Title</div>
                                    </div>
                                </li>
                                <li class="item-divider">Title of the divider
here</li>
                            </ul>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        <li class="item-content">
            <div class="item-media"><i class="icon icon-form-
name"></i></div>
            <div class="item-inner">
                <div class="item-title">Item with badge</div>
                <div class="item-after"> <span
class="badge">78</span></div>
            </div>
        </li>
        <li class="item-content">
            <div class="item-media"><i class="icon icon-form-
name"></i></div>
            <div class="item-inner">
                <div class="item-title">Another item</div>
                <div class="item-after">Another label</div>
            </div>
        </li>
    </ul>
    <div class="list-block-label">List block label text
written here</div>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
</script>
</body>
</html>

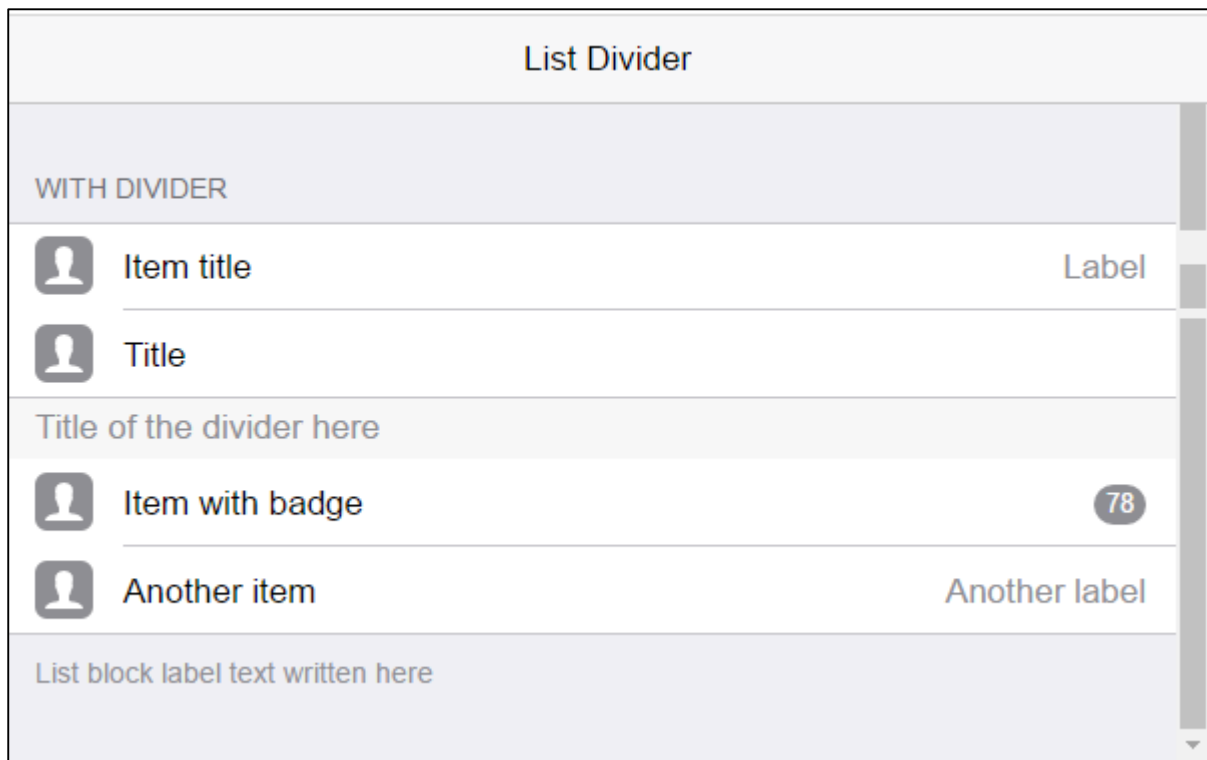
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **list_views_list_divider.html** file in your server root folder.

- Open this HTML file as http://localhost/list_views_list_divider.html and the output is displayed as shown below.
- This code divides one list items from another list items.



Framework7 – Grouped Lists

Description

You can wrap your list view elements within single list block.

Example

The following example demonstrates the use of grouped list in Framework7 list view:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Grouped Lists</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ios.colors.min.css">
</head>
<body>
  <div class="views">
    <div class="view view-main">
      <div class="pages">
        <div data-page="home" class="page navbar-fixed">
          <div class="navbar">
            <div class="navbar-inner">
              <div class="left"> </div>
              <div class="center">List View Groups</div>
              <div class="right"> </div>
            </div>
          </div>
          <div class="page-content">
            <div class="list-block">
              <div class="list-group">
                <ul>
                  <li class="list-group-title">First group</li>
                  <li class="item-content">
                    <div class="item-media"><i class="icon icon-
form-name"></i></div>
                    <div class="item-inner">
                      <div class="item-title">Item title 1</div>
                      <div class="item-after">Label</div>
                    </div>
                  </li>
                  <li class="item-content">
                    <div class="item-media"><i class="icon icon-
form-name"></i></div>
                    <div class="item-inner">
                      <div class="item-title">Item 1 with
badge</div>
                      <div class="item-after"> <span
class="badge">78</span></div>
                    </div>
                  </li>
                </ul>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        <li class="item-content">
            <div class="item-media"><i class="icon icon-
form-name"></i></div>
            <div class="item-inner">
                <div class="item-title">Item title 2</div>
                <div class="item-after">Label</div>
            </div>
        </li>
        <li class="item-content">
            <div class="item-media"><i class="icon icon-
form-name"></i></div>
            <div class="item-inner">
                <div class="item-title">Item 2 with
badge</div>
                <div class="item-after"> <span
class="badge">78</span></div>
            </div>
        </li>
        <li class="item-content">
            <div class="item-media"><i class="icon icon-
form-name"></i></div>
            <div class="item-inner">
                <div class="item-title">Item 3</div>
                <div class="item-after">Label</div>
            </div>
        </li>
    </ul>
</div>
<div class="list-group">
    <ul>
        <li class="list-group-title">Second group</li>
        <li class="item-content">
            <div class="item-media"><i class="icon icon-
form-name"></i></div>
            <div class="item-inner">
                <div class="item-title">Item title 1</div>
                <div class="item-after">Label</div>
            </div>
        </li>
    </ul>

```

```

        <li class="item-content">
            <div class="item-media"><i class="icon icon-
form-name"></i></div>
            <div class="item-inner">
                <div class="item-title">Item 1 with
badge</div>
                <div class="item-after"> <span
class="badge">78</span></div>
            </div>
        </li>
        <li class="item-content">
            <div class="item-media"><i class="icon icon-
form-name"></i></div>
            <div class="item-inner">
                <div class="item-title">Item title 2</div>
                <div class="item-after">Label</div>
            </div>
        </li>
        <li class="item-content">
            <div class="item-media"><i class="icon icon-
form-name"></i></div>
            <div class="item-inner">
                <div class="item-title">Item 2 with
badge</div>
                <div class="item-after"> <span
class="badge">78</span></div>
            </div>
        </li>
        <li class="item-content">
            <div class="item-media"><i class="icon icon-
form-name"></i></div>
            <div class="item-inner">
                <div class="item-title">Item 3</div>
                <div class="item-after">Label</div>
            </div>
        </li>
    </ul>
</div>
<div class="content-block">

```

```

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Cras condimentum molestie diam eu rhoncus. Aliquam a accumsan felis.
Morbi magna ipsum, pellentesque vitae diam mattis, cursus pellentesque lorem.
Curabitur orci tortor, feugiat nec erat non, iaculis iaculis nisi.</p>

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Cras condimentum molestie diam eu rhoncus. Aliquam a accumsan felis.
Morbi magna ipsum, pellentesque vitae diam mattis, cursus pellentesque lorem.
Curabitur orci tortor, feugiat nec erat non, iaculis iaculis nisi.</p>

        </div>
    </div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **list_views_grouped_lists.html** file in your server root folder.
- Open this HTML file as http://localhost/list_views_grouped_lists.html and the output is displayed as shown below.
- This code groups the list view elements inside the single list block.

List View Groups

First group

	Item title 1	Label
	Item 1 with badge	78
	Item title 2	Label
	Item 2 with badge	78
	Item 3	Label

Second group

	Item title 1	Label
	Item 1 with badge	78
	Item title 2	Label
	Item 2 with badge	78
	Item 3	Label

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras condimentum molestie diam eu rhoncus. Aliquam a accumsan felis. Morbi magna ipsum, pellentesque vitae diam mattis, cursus pellentesque lorem. Curabitur orci tortor, feugiat nec erat non, iaculis iaculis nisi.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras condimentum molestie diam eu rhoncus. Aliquam a accumsan felis. Morbi magna ipsum, pellentesque vitae diam mattis, cursus pellentesque lorem. Curabitur orci tortor, feugiat nec erat non, iaculis iaculis nisi.

Framework7 – Inset List Block

Description

By adding *inset* class to *list-block*, your list blocks can have *inset* styling.

Example

The following example demonstrates the use of inset i.e. to make the list block not full width styling in Framework7 list view:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Grouped Lists</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">List View Inset</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">
              <div class="content-block-title">Fullwidth block</div>
              <div class="list-block">
                <ul>
```

```

        <li class="item-content">
            <div class="item-inner">
                <div class="item-title">Item 1</div>
            </div>
        </li>
        <li class="item-content">
            <div class="item-inner">
                <div class="item-title">Item 2</div>
            </div>
        </li>
        <li class="item-content">
            <div class="item-inner">
                <div class="item-title">Item 3</div>
            </div>
        </li>
    </ul>
</div>
<div class="content-block-title">Inset list block</div>
<div class="list-block inset">
    <ul>
        <li class="item-content">
            <div class="item-media"><i class="icon icon-form-
name"></i></div>
            <div class="item-inner">
                <div class="item-title">Item 1</div>
                <div class="item-after">Label 1</div>
            </div>
        </li>
        <li class="item-content">
            <div class="item-media"><i class="icon icon-form-
name"></i></div>
            <div class="item-inner">
                <div class="item-title">Item 2 with
badge</div>
                <div class="item-after"> <span
class="badge">78</span></div>
            </div>
        </li>
        <li class="item-content">

```



```

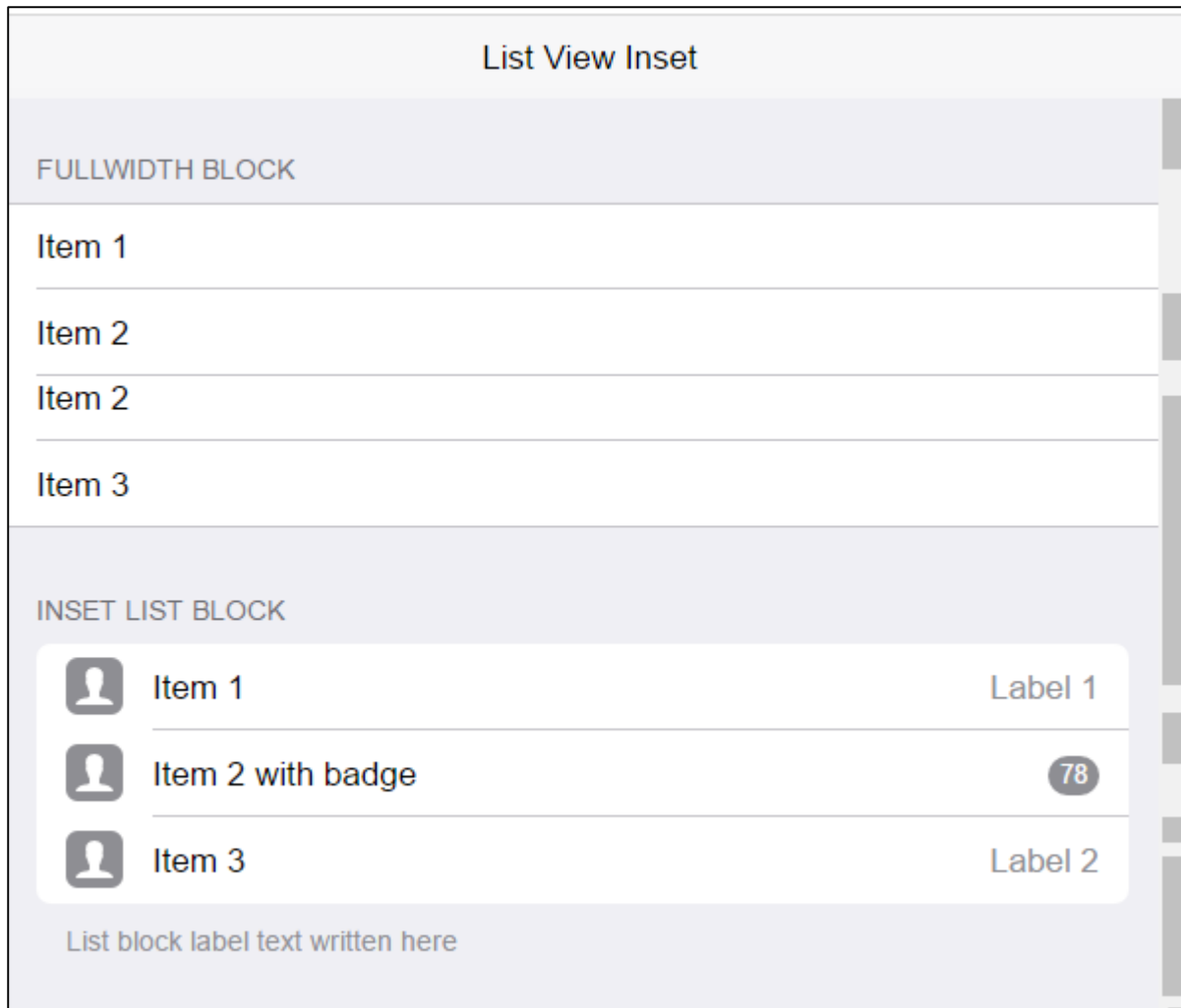
name"></i></div>
        <div class="item-media"><i class="icon icon-form-
        <div class="item-inner">
            <div class="item-title">Item 3</div>
            <div class="item-after">Label 2</div>
        </div>
        </li>
    </ul>
    <div class="list-block-label">List block label text
written here</div>
        </div>
    </div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **list_views_inset.html** file in your server root folder.
- Open this HTML file as http://localhost/list_views_inset.html and the output is displayed as shown below.
- This example removes the fullwidth styling of list block by using the inset class.



Framework7 – List View Contacts List

Description

The contacts list is a type of list view, which can be used to display the list of people contacts.

Example

The following example demonstrates the use of contacts list in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
```

```

<title>List View Contacts List</title>
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
<div class="views">
<div class="view view-main">
<div class="navbar">
<div class="navbar-inner">
<div class="center sliding">List View Contacts List</div>
</div>
</div>
<div class="pages navbar-through">
<div data-page="index" class="page navbar-fixed">
<div class="page-content contacts-content">
<div class="list-block contacts-block">
<div class="list-group">
<ul>
<li class="list-group-title">A</li>
<li>
<div class="item-content">
<div class="item-inner">
<div class="item-title">Aadi</div>
</div>
</div>
</li>
<li>
<div class="item-content">
<div class="item-inner">
<div class="item-title">Aakash</div>
</div>
</div>
</li>
<li>
<div class="item-content">

```

```

        <div class="item-inner">
            <div class="item-title">Abhay</div>
        </div>
    </li>
    <li>
        <div class="item-content">
            <div class="item-inner">
                <div class="item-title">Abhinav</div>
            </div>
        </li>
    <li>
        <div class="item-content">
            <div class="item-inner">
                <div class="item-title">Aditya</div>
            </div>
        </li>
    </ul>
</div>
<div class="list-group">
    <ul>
        <li class="list-group-title">B</li>
        <li>
            <div class="item-content">
                <div class="item-inner">
                    <div class="item-title">Bhairav</div>
                </div>
            </div>
        </li>
        <li>
            <div class="item-content">
                <div class="item-inner">
                    <div class="item-title">Bhargava</div>
                </div>
            </div>
        </li>
    </ul>
</div>

```

```

        </li>
        <li>
            <div class="item-content">
                <div class="item-inner">
                    <div class="item-title">Bhooshan</div>
                </div>
            </div>
        </li>
        <li>
            <div class="item-content">
                <div class="item-inner">
                    <div class="item-title">Brijesh</div>
                </div>
            </div>
        </li>
    </ul>
</div>
<div class="list-group">
    <ul>
        <li class="list-group-title">C</li>
        <li>
            <div class="item-content">
                <div class="item-inner">
                    <div class="item-title">Chaitan</div>
                </div>
            </div>
        </li>
        <li>
            <div class="item-content">
                <div class="item-inner">
                    <div class="item-
title">Chandrakumar</div>
                </div>
            </div>
        </li>
        <li>
            <div class="item-content">
                <div class="item-inner">

```

```

        <div class="item-title">Chinmay</div>
    </div>
</div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-title">Chirag</div>
        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-
title">Chiranjeevi</div>
        </div>
    </div>
</li>
</ul>
</div>
<div class="list-group">
    <ul>
        <li class="list-group-title">D</li>
        <li>
            <div class="item-content">
                <div class="item-inner">
                    <div class="item-title">Danish</div>
                </div>
            </div>
        </li>
        <li>
            <div class="item-content">
                <div class="item-inner">
                    <div class="item-title">Darshan</div>
                </div>
            </div>
        </li>
    </ul>

```


List View Contacts List	
A	
Aadi	
Aakash	
Abhay	
Abhinav	
Aditya	
Aditya	
B	
B	
Bhairav	
Bhairav	
Bhargava	
Bhooshan	
Brijesh	
C	
Chaitan	
Chandrakumar	
Chinmay	
Chirag	
Chiranjeevi	
D	
D	
Danish	
Darshan	

Framework7 – Media List View

Description

- The Media list view is used to display complicated data structures like products, services and user information.
- You can use media list view by using the *item-content* class, which is main wrapper for the *item-media* (used for the media elements) and *item-inner* (includes title, sub title, text elements) classes.

Example

The following example displays the list elements along with the images in the Framework7 list view:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Media List View</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="center sliding">Media List View</div>
          </div>
        </div>
        <div class="pages navbar-through">
          <div data-page="index" class="page navbar-fixed">
            <div class="page-content">
              <div class="content-block-title">Countries</div>
            </div>
          </div>
        </div>
      </div>
    </body>
</html>
```

```

<div class="list-block media-list">
  <ul>
    <li>
      <a href="#" class="item-link item-content">
        <div class="item-media"></div>
        <div class="item-inner">
          <div class="item-title-row">
            <div class="item-title">India</div>
            <div class="item-after">Rupees</div>
          </div>
          <div class="item-subtitle">Capital -
Delhi</div>
          <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt. Cras dolor metus, ultrices
condimentum sodales sit amet, pharetra sodales eros. Phasellus vel felis
tellus. Mauris rutrum ligula nec dapibus feugiat. In vel dui laoreet, commodo
augue id, pulvinar lacus.</div>
        </div>
      </a>
    </li>
    <li>
      <a href="#" class="item-link item-content">
        <div class="item-media"></div>
        <div class="item-inner">
          <div class="item-title-row">
            <div class="item-title">Argentina</div>
            <div class="item-after">Argentine
Pesos</div>
          </div>
          <div class="item-subtitle">Capital - Buenos
Aires</div>
          <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt. Cras dolor metus, ultrices
condimentum sodales sit amet, pharetra sodales eros. Phasellus vel felis
tellus. Mauris rutrum ligula nec dapibus feugiat. In vel dui laoreet, commodo
augue id, pulvinar lacus.</div>
        </div>
      </a>
    </li>
  </ul>
</div>

```

```

        </li>
        <li>
            <a href="#" class="item-link item-content">
                <div class="item-media"></div>
                <div class="item-inner">
                    <div class="item-title-row">
                        <div class="item-title">Belgium</div>
                        <div class="item-after">Euro</div>
                    </div>
                    <div class="item-subtitle">Capital -
Brussels</div>
                    <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt. Cras dolor metus, ultrices
condimentum sodales sit amet, pharetra sodales eros. Phasellus vel felis
tellus. Mauris rutrum ligula nec dapibus feugiat. In vel dui laoreet, commodo
augue id, pulvinar lacus.</div>
                </div>
            </a>
        </li>
    </ul>
</div>
<div class="content-block-title">Cricket Teams Ranking in
T20</div>
<div class="list-block media-list">
    <ul>
        <li>
            <a href="#" class="item-link item-content">
                <div class="item-inner">
                    <div class="item-title-row">
                        <div class="item-title">India</div>
                    </div>
                    <div class="item-subtitle">Position :
1</div>
                    <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt. </div>
                </div>
            </a>
        </li>
    </ul>

```

```

        <li>
            <a href="#" class="item-link item-content">
                <div class="item-inner">
                    <div class="item-title-row">
                        <div class="item-title">West
Indies</div>

                    </div>
                    <div class="item-subtitle">Position :
2</div>

                    <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt.</div>
                    </div>
                </a>
            </li>
            <li>
                <a href="#" class="item-link item-content">
                    <div class="item-inner">
                        <div class="item-title-row">
                            <div class="item-title">New
Zealand</div>

                        </div>
                        <div class="item-subtitle">Position :
3</div>

                        <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt.</div>
                        </div>
                    </a>
                </li>
                <li>
                    <a href="#" class="item-link item-content">
                        <div class="item-inner">
                            <div class="item-title-row">
                                <div class="item-title">England</div>
                            </div>
                            <div class="item-subtitle">Position :
4</div>

                            <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt.</div>
                    </a>
                </li>
            </li>
        </li>
    </div>

```

```

        </div>
    </a>
</li>
</ul>
</div>
<div class="content-block-title">Fruits</div>
<div class="list-block media-list">
    <ul>
        <li>
            <div class="item-content">
                <div class="item-media"></div>
                <div class="item-inner">
                    <div class="item-title-row">
                        <div class="item-title">Apple</div>
                    </div>
                    <div class="item-subtitle">Contains list of
antioxidants and essential nutrients.</div>
                </div>
            </div>
</li>
<li>
            <a href="#" class="item-link item-content">
                <div class="item-media"></div>
                <div class="item-inner">
                    <div class="item-title-row">
                        <div class="item-title">Orange</div>
                    </div>
                    <div class="item-subtitle">It is low in
calories, contains no saturated fats or cholesterol.</div>
                </div>
            </a>
</li>
<li>
            <div class="item-content">
                <div class="item-media"></div>
                <div class="item-inner">

```

```

        <div class="item-title-row">
            <div class="item-title">Watermelon</div>
        </div>
        <div class="item-subtitle">It is a source
of Vitamin-A and powerful natural anti-oxidant.</div>
        </div>
    </div>
</li>
</ul>
</div>
<div class="content-block-title">Sports</div>
<div class="list-block media-list inset">
    <ul>
        <li>
            <a href="#" class="item-link item-content">
                <div class="item-media"></div>
                <div class="item-inner">
                    <div class="item-title-row">
                        <div class="item-title">India</div>
                    </div>
                    <div class="item-subtitle">Hockey</div>
                </div>
            </a>
        </li>
        <li>
            <a href="#" class="item-link item-content">
                <div class="item-media"></div>
                <div class="item-inner">
                    <div class="item-title-row">
                        <div class="item-title">England</div>
                    </div>
                    <div class="item-subtitle">Cricket</div>
                </div>
            </a>
        </li>
        <li>

```



Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **media_list_view.html** file in your server root folder.
- Open this HTML file as http://localhost/media_list_view.html and the output is displayed as shown below.
- The example displays the list elements along with the images.

Media List View


COUNTRIES



India Rupees >

Capital - Delhi


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis condimentum, ut dignissim lacus tincidunt. Cras dolor...



Argentina Argentina Peso >

Capital - Buenos Aires

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis condimentum, ut dignissim lacus tincidunt. Cras dolor...




Belgium Euro >

Capital - Brussels

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis condimentum, ut dignissim lacus tincidunt. Cras dolor...

CRICKET TEAMS RANKING IN T20

CRICKET TEAMS RANKING IN T20




India >

Position : 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis condimentum, ut dignissim lacus tincidunt.


condimentum, ut dignissim lacus tincidunt.



West Indies >

Position : 2


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis condimentum, ut dignissim lacus tincidunt.



New Zealand >

Position : 3

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis condimentum, ut dignissim lacus tincidunt.




England >

Position : 4


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis condimentum, ut dignissim lacus tincidunt.

FRUITS




Apple

Contains list of antioxidants and essential nutrients.



Orange >


It is low in calories, contains no saturated fats or cholesterol.



Watermelon


It is a source of Vitamin-A and powerful natural anti-oxidant.

SPORTS




India >

Hockey



England >

Cricket



United States >

Baseball

Framework7 – Swipeout

Description

The *swipeout* allows you to reveal hidden menu actions by swiping over the list elements. *Swipeout* is an extension of *list view*.

Swipeout Classes

The following table shows the list of classes used in *swipeout*:

S. No.	Class & Description
1	swipeout-content It is the wrapper for your list element. It will move when you swipe over the list item.
2	swipeout-actions-left It is the container consisting of left swipeout action buttons/links.
3	swipeout-actions-right It is the container consisting of right swipeout action buttons/links.
4	swipeout-close It is an additional class on swipeout link. It is used to close swipeout element whenever you click this link.

The *swipeout-content* and *swipeout-actions-left/right* should be direct children of ``

Swipeout Types

The following table shows the *swipeout* types used in Framework7:

S. No.	Type & Description
1	Swipeout Basic A basic swipeout contains <i>swipeout-content</i> wrapper and action buttons/links.
2	Swipe To Delete To delete the swipeout action buttons, you can use <i>swipeout-delete</i> class.
3	Overswipe The <i>overswipe</i> actions will be automatically triggered when you swipe actions too much.
4	Swipeout JavaScript API To control the swipeout elements, swipeout provide you with JavaScript API.
5	Swipeout events To detect how a user interacts with the swipeout, you can use swipeout events.

Framework7 – Basic Swipeout

Description

A basic swipeout contains *swipeout-content* wrapper and action buttons/links.

Example

The following example demonstrates the use of swipeout in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Swipeout</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Swipeout</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">
              <div class="content-block-title">Swipe on right side</div>
              <div class="list-block">
                <ul>
                  <li class="swipeout">
```

```

        <div class="swipeout-content item-content">
            <div class="item-media"><i class="icon icon-
form-name"></i></div>
            <div class="item-inner">
                <div class="item-title">Item title</div>
                <div class="item-after">Label</div>
            </div>
        </div>
        <div class="swipeout-actions-right"><a href="#"
class="task1 bg-red">Task 1</a></div>
    </li>
    <li class="swipeout">
        <div class="swipeout-content item-content">
            <div class="item-media"><i class="icon icon-
form-name"></i></div>
            <div class="item-inner">
                <div class="item-title">Item with
badge</div>
                <div class="item-after"> <span class="badge
bg-blue">78</span></div>
            </div>
        </div>
        <div class="swipeout-actions-right"><a href="#"
class="task1 bg-red">Task 1</a><a href="#" class="task2 bg-green">Task
2</a></div>
    </li>
    <li class="swipeout">
        <div class="swipeout-content item-content">
            <div class="item-media"><i class="icon icon-
form-name"></i></div>
            <div class="item-inner">
                <div class="item-title">Another item</div>
                <div class="item-after">Another label</div>
            </div>
        </div>
        <div class="swipeout-actions-right"><a href="#"
class="task1 bg-red">Task 1</a><a href="#" class="task2 bg-green">Task
2</a></div>
    </li>
</ul>
</div>

```

```

<div class="content-block-title">Swipe on left side</div>
<div class="list-block">
  <ul>
    <li class="swipeout">
      <div class="swipeout-content item-content">
        <div class="item-media"><i class="icon icon-
form-name"></i></div>
        <div class="item-inner">
          <div class="item-title">Item title</div>
          <div class="item-after">Label</div>
        </div>
      </div>
      <div class="swipeout-actions-left"><a href="#"
class="task1 bg-red">Task 1</a></div>
    </li>
    <li class="swipeout">
      <div class="swipeout-content item-content">
        <div class="item-media"><i class="icon icon-
form-name"></i></div>
        <div class="item-inner">
          <div class="item-title">Item with
badge</div>
          <div class="item-after"> <span class="badge
bg-blue">78</span></div>
        </div>
      </div>
      <div class="swipeout-actions-left"><a href="#"
class="task1 bg-red">Task 1</a><a href="#" class="task2 bg-green">Task
2</a></div>
    </li>
    <li class="swipeout">
      <div class="swipeout-content item-content">
        <div class="item-media"><i class="icon icon-
form-name"></i></div>
        <div class="item-inner">
          <div class="item-title">Another item</div>
          <div class="item-after">Another label</div>
        </div>
      </div>
    </li>
  </ul>
</div>

```

```

                <div class="swipeout-actions-left"><a href="#"
class="task1 bg-red">Task 1</a><a href="#" class="task2 bg-green">Task
2</a></div>
            </li>
        </ul>
    </div>
    <div class="content-block-title">Swipe on both sides</div>
    <div class="list-block">
        <ul>
            <li class="swipeout">
                <div class="swipeout-content item-content">
                    <div class="item-media"><i class="icon icon-
form-name"></i></div>
                    <div class="item-inner">
                        <div class="item-title">Item title</div>
                        <div class="item-after">Label</div>
                    </div>
                </div>
                <div class="swipeout-actions-left"><a href="#"
class="task1 bg-red">Task L1</a></div>
                <div class="swipeout-actions-right"><a href="#"
class="task1 bg-pink">Task R1</a></div>
            </li>
            <li class="swipeout">
                <div class="swipeout-content item-content">
                    <div class="item-media"><i class="icon icon-
form-name"></i></div>
                    <div class="item-inner">
                        <div class="item-title">Item with
badge</div>
                        <div class="item-after"> <span class="badge
bg-blue">78</span></div>
                    </div>
                </div>
                <div class="swipeout-actions-left"><a href="#"
class="task1 bg-red">Task L1</a><a href="#" class="task2">Task L2</a></div>
                <div class="swipeout-actions-right"><a href="#"
class="task1 bg-red">Task R1</a><a href="#" class="task2">Task R2</a></div>
            </li>
        </ul>
    </div>

```

```

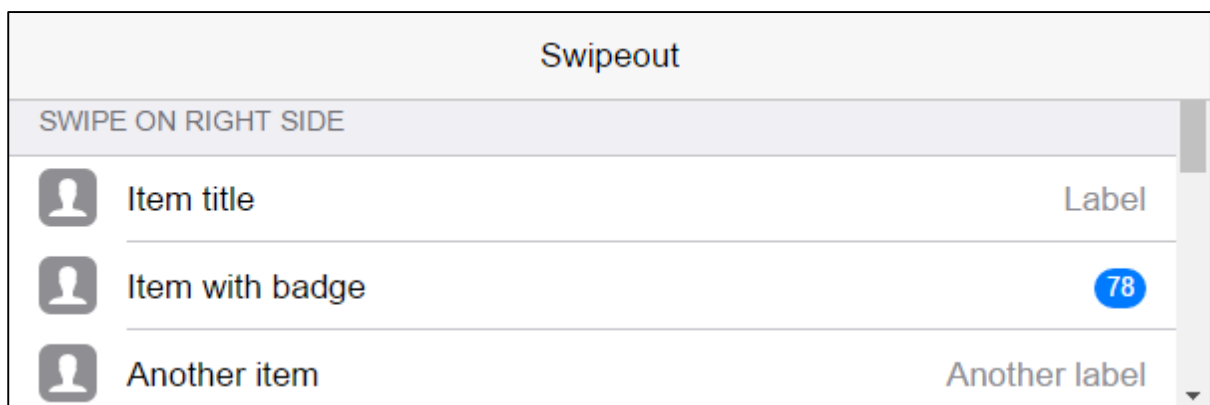
        </div>
    </div>
    </div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var $$ = Dom7;
    $$('.task1').on('click', function () {
        myApp.alert('Task 1 Clicked!!');
    });
    $$('.task2').on('click', function () {
        myApp.alert('Task 2 Clicked!!');
    });
</script>
</body>
</html>

```

Output

Let's carry out the following steps to see how above code works:

- Save above HTML code as **list_views_swipe_basic.html** file in your server root folder.
- Open this HTML file as http://localhost/list_views_swipe_basic.html and output as below gets displayed.
- It allows you to swipe over list elements using the `swipeout-content` wrapper.



This example is best viewed and executed on a mobile.

Framework7 – Swipe To Delete

Description

To delete the swipeout action buttons, follow the instructions:

- You can use `` link.
- Next, you can make delete on confirmation by adding `data-confirm` and `data-confirm-title` attributes to the delete link.
- To cancel the confirm model, you need to use `data-close-on-cancel` attribute.

Example

The following example demonstrates the use of swipeout to delete the action buttons in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Swipe To Delete</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Swipe To Delete</div>
                <div class="right"> </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```



```

    </div>
    <div class="page-content">
      <div class="list-block">
        <ul>
          <li class="swipeout">
            <div class="swipeout-content item-content">
              <div class="item-media"><i class="icon icon-
form-name"></i></div>
              <div class="item-inner">
                <div class="item-title">Item title</div>
                <div class="item-after">Label</div>
              </div>
            </div>
            <div class="swipeout-actions-right"><a href="#"
class="task1 bg-green">Task 1</a><a href="#" class="swipeout-delete bg-
yellow">Delete</a></div>
          </li>
          <li class="swipeout">
            <div class="swipeout-content item-content">
              <div class="item-media"><i class="icon icon-
form-name"></i></div>
              <div class="item-inner">
                <div class="item-title">Item with
badge</div>
                <div class="item-after"> <span class="badge
bg-lightblue">78</span></div>
              </div>
            </div>
            <div class="swipeout-actions-right"><a href="#"
class="task1 bg-green">Task 1</a><a href="#" class="swipeout-delete bg-
yellow">Delete</a></div>
          </li>
        </ul>
      </div>
      <div class="content-block-title">Delete only after
confirmation</div>
      <div class="list-block">
        <ul>
          <li class="swipeout">
            <div class="swipeout-content item-content">

```

```

        <div class="item-media"><i class="icon icon-
form-name"></i></div>
        <div class="item-inner">
            <div class="item-title">Item title</div>
            <div class="item-after">Label</div>
        </div>
    </div>
    <div class="swipeout-actions-right"><a href="#"
class="task1 bg-green">Task 1</a><a href="#" data-confirm="Are you sure to
delete this item?" class="swipeout-delete bg-yellow">Delete</a></div>
</li>
<li class="swipeout">
    <div class="swipeout-content item-content">
        <div class="item-media"><i class="icon icon-
form-name"></i></div>
        <div class="item-inner">
            <div class="item-title">With
confirmation</div>
            <div class="item-after"> <span class="badge
bg-lightblue">78</span></div>
        </div>
    </div>
    <div class="swipeout-actions-right"><a href="#"
class="task1 bg-green">Task 1</a><a href="#" data-confirm="Are you sure to
delete this item?" data-confirm-title="Delete?" class="swipeout-delete bg-
yellow">Delete</a></div>
</li>
</ul>
</div>
<div class="content-block-title">With auto close on
cancel</div>
<div class="list-block">
    <ul>
        <li class="swipeout">
            <div class="swipeout-content item-content">
                <div class="item-media"><i class="icon icon-
form-name"></i></div>
                <div class="item-inner">
                    <div class="item-title">Item title</div>
                    <div class="item-after">Label</div>
                </div>
            </div>
        </li>
    </ul>
</div>

```

```
        </div>
        <div class="swipeout-actions-right"><a href="#"
class="task1 bg-green">Task 1</a><a href="#" data-confirm="Are you sure to
delete this item?" data-close-on-cancel="true" class="swipeout-delete bg-
yellow">Delete</a></div>
    </li>
    <li class="swipeout">
        <div class="swipeout-content item-content">
            <div class="item-media"><i class="icon icon-
form-name"></i></div>
            <div class="item-inner">
                <div class="item-title">With
confirmation</div>
                <div class="item-after"> <span class="badge
bg-lightblue">78</span></div>
            </div>
        </div>
        <div class="swipeout-actions-right"><a href="#"
class="task1 bg-green">Task 1</a><a href="#" data-confirm="Are you sure to
delete this item?" data-confirm-title="Delete?" data-close-on-cancel="true"
class="swipeout-delete bg-yellow">Delete</a></div>
    </li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();

    var $$ = Dom7;

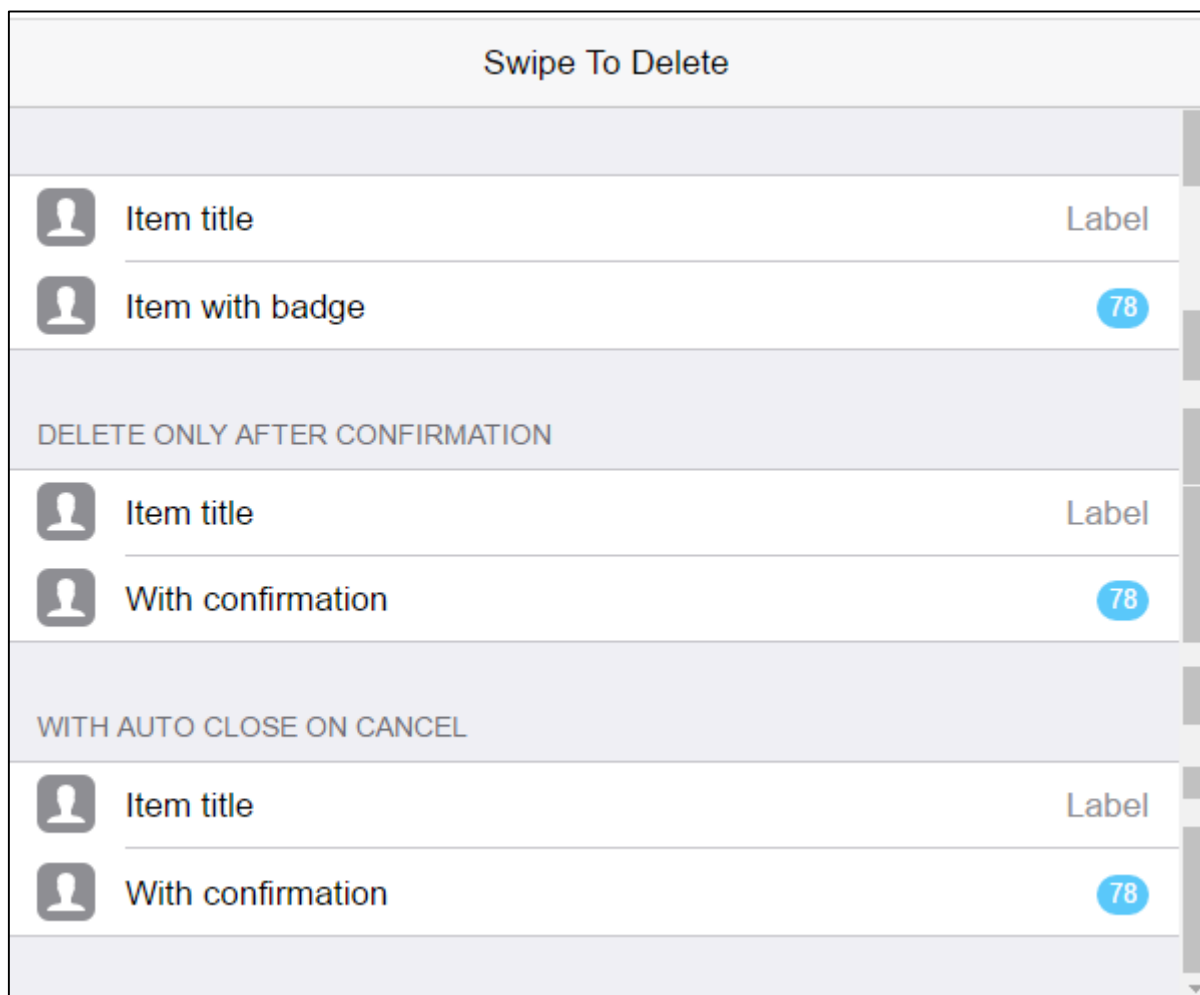
    $$('.task1').on('click', function () {
        myApp.alert('Task 1 Clicked !!');
    });
</script>
```

```
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **list_views_swipe_delete.html** file in your server root folder.
- Open this HTML file as http://localhost/list_views_swipe_delete.html and the output is displayed as shown below.
- The example allows you to swipe over list elements and delete the element on the confirmation model.



Framework7 – Overswipe

Description

The *overswipe* action is triggered automatically when you swipe too much. The following list describes a few more points on an *overswipe*.

- The *overswipe* can be applied to an action button by using the *swipeout-overswipe* class.
- It can only be used on last button in right swipeout and only on first button in left swipeout actions.
- You need to add appropriate event listener to overswipe button.
- During overswipe, the *swipeout-overswipe-active* class will be added to overswipe button, which can be used for extra styling.

Example

The following example displays the items if there are too many swipe actions by using the overswipe in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Overswipe</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
```

```

        <div class="center">Swipe To Delete</div>
        <div class="right"> </div>
    </div>
</div>
<div class="page-content">
    <div class="list-block media-list">
        <ul>
            <li class="swipeout">
                <div class="swipeout-content">
                    <a href="#" class="item-content item-link">
                        <div class="item-inner">
                            <div class="item-title-row">
                                <div class="item-title">Amayon</div>
                                <div class="item-after">7:10</div>
                            </div>
                            <div class="item-subtitle">Your order
has been shipped</div>
                                <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt.</div>
                            </div>
                        </a>
                    </div>
                    <div class="swipeout-actions-left"><a href="#"
class="swipeout-overswipe bg-pink reply">Reply</a><a href="#" class="bg-blue
forward">Forward</a></div>
                    <div class="swipeout-actions-right"><a href="#"
class="mark bg-lightblue">Spam</a><a href="#" class="swipeout-delete swipeout-
overswipe">Delete</a></div>
                </li>
            <li class="swipeout">
                <div class="swipeout-content">
                    <a href="#" class="item-content item-link">
                        <div class="item-inner">
                            <div class="item-title-row">
                                <div class="item-
title">VodafoneZone</div>
                                    <div class="item-after">10:15</div>
                            </div>
                        </a>
                    </div>
                </li>
            </ul>
        </div>
    </div>

```

```

                <div class="item-subtitle">Bill
Payments</div>

                <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt.</div>

                </div>
            </a>
        </div>
        <div class="swipeout-actions-left"><a href="#"
class="swipeout-overswipe bg-pink reply">Reply</a><a href="#" class="bg-blue
forward">Forward</a></div>
        <div class="swipeout-actions-right"><a href="#"
class="mark bg-lightblue">Spam</a><a href="#" class="swipeout-delete swipeout-
overswipe">Delete</a></div>
    </li>
    <li class="swipeout">
        <div class="swipeout-content">
            <a href="#" class="item-content item-link">
                <div class="item-inner">
                    <div class="item-title-row">
                        <div class="item-title">Popular in
your network</div>
                        <div class="item-after">19:14</div>
                    </div>
                    <div class="item-subtitle">New messages
from John Doe</div>
                    <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt.</div>
                </div>
            </a>
        </div>
        <div class="swipeout-actions-left"><a href="#"
class="swipeout-overswipe bg-pink reply">Reply</a><a href="#" class="bg-blue
forward">Forward</a></div>
        <div class="swipeout-actions-right"><a href="#"
class="mark bg-lightblue">Spam</a><a href="#" class="swipeout-delete swipeout-
overswipe">Delete</a></div>
    </li>
    <li class="swipeout">
        <div class="swipeout-content">
            <a href="#" class="item-content item-link">

```

```

                <div class="item-inner">
                    <div class="item-title-row">
                        <div class="item-title">Adam
WillSmith</div>

                        <div class="item-after">22:44</div>
                    </div>
                    <div class="item-subtitle">Car Insurance
renewal</div>

                    <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt.</div>
                </div>
            </a>
        </div>
        <div class="swipeout-actions-left"><a href="#"
class="swipeout-overswipe bg-pink reply">Reply</a><a href="#" class="bg-blue
forward">Forward</a></div>
        <div class="swipeout-actions-right"><a href="#"
class="mark bg-lightblue">Spam</a><a href="#" class="swipeout-delete swipeout-
overswipe">Delete</a></div>
    </li>
    <li class="swipeout">
        <div class="swipeout-content">
            <a href="#" class="item-content item-link">
                <div class="item-inner">
                    <div class="item-title-row">
                        <div class="item-title">Twitter</div>
                        <div class="item-after">23:22</div>
                    </div>
                    <div class="item-subtitle">Follow
Vivekananda Fans and Bill Gates</div>

                    <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt.</div>
                </div>
            </a>
        </div>
        <div class="swipeout-actions-left"><a href="#"
class="swipeout-overswipe bg-pink reply">Reply</a><a href="#" class="bg-blue
forward">Forward</a></div>
        <div class="swipeout-actions-right"><a href="#"
class="mark bg-lightblue">Spam</a><a href="#" class="swipeout-delete swipeout-
overswipe">Delete</a></div>

```



```

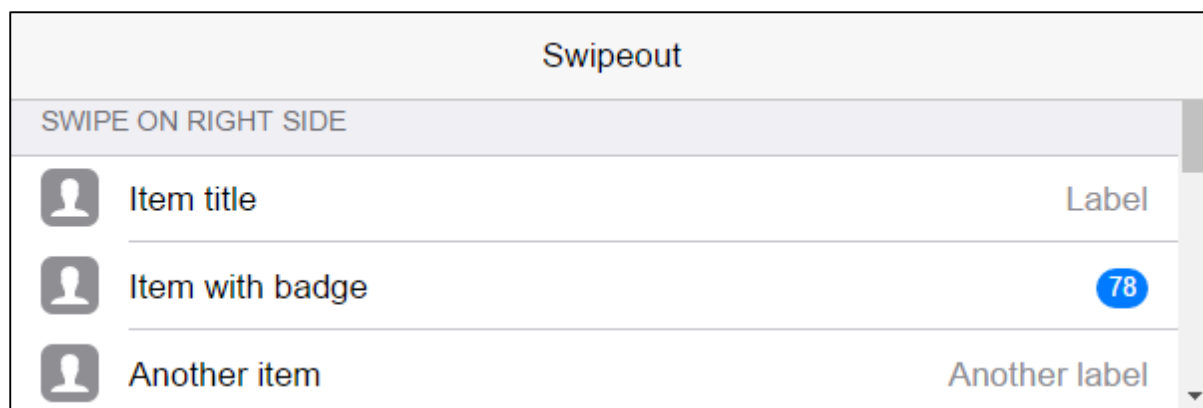
        </li>
    </ul>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var $$ = Dom7;
    $$('.mark').on('click', function () {
        myApp.alert('Spam');
    });
    $$('.reply').on('click', function () {
        myApp.alert('Reply');
    });
    $$('.forward').on('click', function () {
        myApp.alert('Forward');
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **list_views_swipe_basic.html** file in your server root folder.
- Open this HTML file as http://localhost/list_views_swipe_basic.html and the output is displayed as shown below.
- This example triggers an overswipe action automatically when you swipe too much and the script will automatically trigger the "click" event on overswipe button.



Framework7 – Swipeout JavaScript API

Description

The swipeout provides you with JavaScript API to control its elements. The appropriate app's properties and methods are listed below:

S. No.	Methods	Description & Parameters
1	myApp.swipeoutOpen(el, direction, callback)	<p>It is used to reveal swipeout actions on element which is specified.</p> <ul style="list-style-type: none"> • el - It is <i>HTMLElement</i> or <i>string</i> (with <i>CSS Selector</i>) of list element. • direction - It is a <i>string</i> (either <i>left</i> or <i>right</i>) that specifies swipeout action to open. • callback - This <i>function</i> will be executed after the swipeout completes its opening animation.
2	myApp.swipeoutClose(el, callback)	<p>It is used to close swipeout actions on element which is specified.</p> <ul style="list-style-type: none"> • el - It is <i>HTMLElement</i> or <i>string</i> (with <i>CSS Selector</i>) of list element. • callback - This <i>function</i> will be executed after the swipeout completes its opening animation.
3	myApp.swipeoutDelete (el, callback)	It is used to delete specified element of swipeout.

		<ul style="list-style-type: none"> • el - It is <i>HTMLElement</i> or <i>string</i> (with <i>CSS Selector</i>) of list element. • callback - This <i>function</i> will be executed after the swipeout completes its opening animation.
4	myApp.swipeoutOpenedEl	This property contains link to swipeout <i>HTMLElement</i> that is opened currently.

Example

The following example allows swiping out elements by using the use of JavaScript API in Framework7:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Swipeout JavaScript API</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Swipeout JavaScript API</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">

```

```

        <div class="list-block">
            <ul>
                <li class="swipeout">
                    <div class="swipeout-content item-content">
                        <div class="item-media"><i class="icon icon-
form-name"></i></div>
                        <div class="item-inner">
                            <div class="item-title">Item title</div>
                            <div class="item-after">Label 1</div>
                        </div>
                        <div class="swipeout-actions-right"><a href="#"
class="action1 bg-red">Task</a></div>
                    </li>
                    <li class="swipeout">
                        <div class="swipeout-content item-content">
                            <div class="item-media"><i class="icon icon-
form-name"></i></div>
                            <div class="item-inner">
                                <div class="item-title">Item with
badge</div>
                                <div class="item-after"> <span class="badge
bg-lightblue">78</span></div>
                            </div>
                            <div class="swipeout-actions-right"><a href="#"
class="action1 bg-red">Task</a></div>
                        </li>
                        <li class="swipeout">
                            <div class="swipeout-content item-content">
                                <div class="item-media"><i class="icon icon-
form-name"></i></div>
                                <div class="item-inner">
                                    <div class="item-title">Another item</div>
                                    <div class="item-after">Label 2</div>
                                </div>
                                <div class="swipeout-actions-right"><a href="#"
class="action1 bg-red">Task</a></div>
                            </li>

```

```

        <li class="swipeout">
            <div class="swipeout-content item-content">
                <div class="item-media"><i class="icon icon-
form-name"></i></div>
                <div class="item-inner">
                    <div class="item-title">Item title</div>
                    <div class="item-after">Label 3</div>
                </div>
            </div>
            <div class="swipeout-actions-right"><a href="#"
class="action1 bg-red">Task</a></div>
        </li>
        <li class="swipeout">
            <div class="swipeout-content item-content">
                <div class="item-media"><i class="icon icon-
form-name"></i></div>
                <div class="item-inner">
                    <div class="item-title">Item with
badge</div>
                    <div class="item-after"> <span class="badge
bg-lightblue">78</span></div>
                </div>
            </div>
            <div class="swipeout-actions-right"><a href="#"
class="action1 bg-red">Task</a></div>
        </li>
    </ul>
</div>
<div class="content-block">
    <p><a href="#" class="open-first">Open first
item</a></p>
    <p><a href="#" class="open-second">Open second item and
close after 1s</a></p>
    <p><a href="#" class="delete-last">Delete last
item</a></p>
</div>
</div>
</div>
</div>
</div>

```

```

</div>

<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>

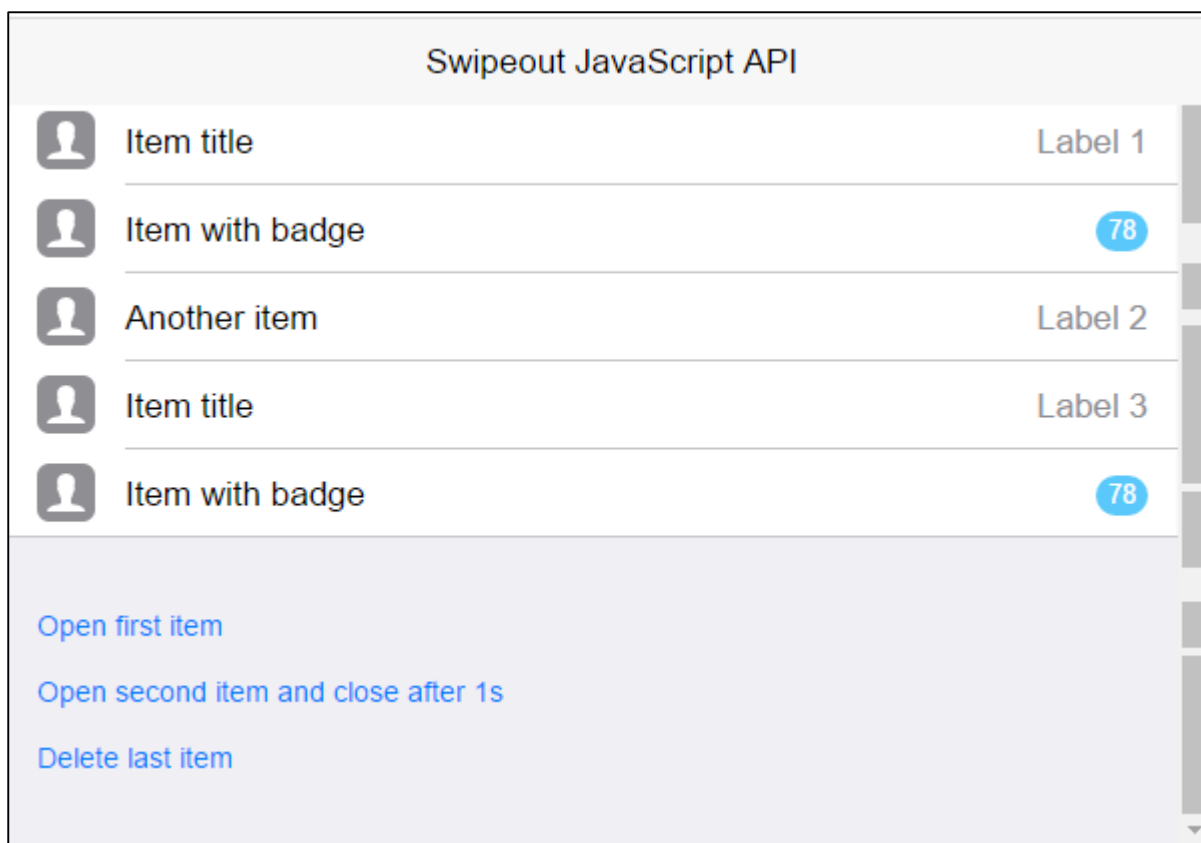
<script>
    var myApp = new Framework7();
    var $$ = Dom7;
    $$('.open-first').on('click', function(){
        myApp.swipeoutOpen('li.swipeout:first-child');
    });
    $$('.open-second').on('click', function(){
        myApp.swipeoutOpen($$('.li.swipeout').eq(1));
        setTimeout(function () {
            myApp.swipeoutClose($$('.li.swipeout').eq(1));
        }, 1000);
    });
    $$('.delete-last').on('click', function(){
        myApp.swipeoutDelete('li.swipeout:last-child');
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **list_views_swipe_jsapis.html** file in your server root folder.
- Open this HTML file as http://localhost/list_views_swipe_jsapis.html and the output is displayed as shown below.
- The example allows you to open the element, deleting the element by clicking on the links.



Framework7 – Swipeout Events

Description

To detect how a user interacts with the swipeout, you can use swipeout events. The following table lists the events supported by the Framework7:

S. No.	Event & Description	Target
1	swipeout This event will be triggered when you move swipeout element.	Swipeout Element <li class="swipeout">
2	open Whenever the swipeout begins its opening animation, this event will be triggered.	Swipeout Element <li class="swipeout">
3	opened Whenever the swipeout completes its opening animation, this event will be triggered.	Swipeout Element <li class="swipeout">
4	close Whenever the swipeout closing animation is initiated, this event will be triggered.	Swipeout Element <li class="swipeout">

5	closed Whenever the swipeout closing animation completes, this event will be triggered.	Swipeout Element <li class="swipeout">
6	delete Whenever the swipeout element begins its delete animation, this event will be triggered.	Swipeout Element <li class="swipeout">
7	deleted This event will be triggered after the element finishes its delete animation right before it will be removed from DOM.	Swipeout Element <li class="swipeout">

Example

The following example demonstrates the use of swipeout events in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Swipeout events</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Swipeout Events</div>
                <div class="right"> </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```



```

<div class="page-content">
  <div class="list-block">
    <ul>
      <li class="swipeout">
        <div class="swipeout-content item-content">
          <div class="item-media"><i class="icon icon-
form-name"></i></div>
          <div class="item-inner">
            <div class="item-title">Item title</div>
            <div class="item-after">Label</div>
          </div>
          </div>
          <div class="swipeout-actions-right"><a href="#"
class="action1">Task</a><a href="#" data-confirm="Are you sure to delete this
item?" class="swipeout-delete">Delete</a></div>
        </li>
      <li class="swipeout">
        <div class="swipeout-content item-content">
          <div class="item-media"><i class="icon icon-
form-name"></i></div>
          <div class="item-inner">
            <div class="item-title">With confirm
title</div>
            <div class="item-after"> <span
class="badge">78</span></div>
          </div>
          </div>
          <div class="swipeout-actions-right"><a href="#"
class="action1">Task</a><a href="#" data-confirm="Are you sure to delete this
item?" data-confirm-title="Delete?" class="swipeout-delete">Delete</a></div>
        </li>
      </ul>
    </div>
  </div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>

```

```

<script>
  var myApp = new Framework7();

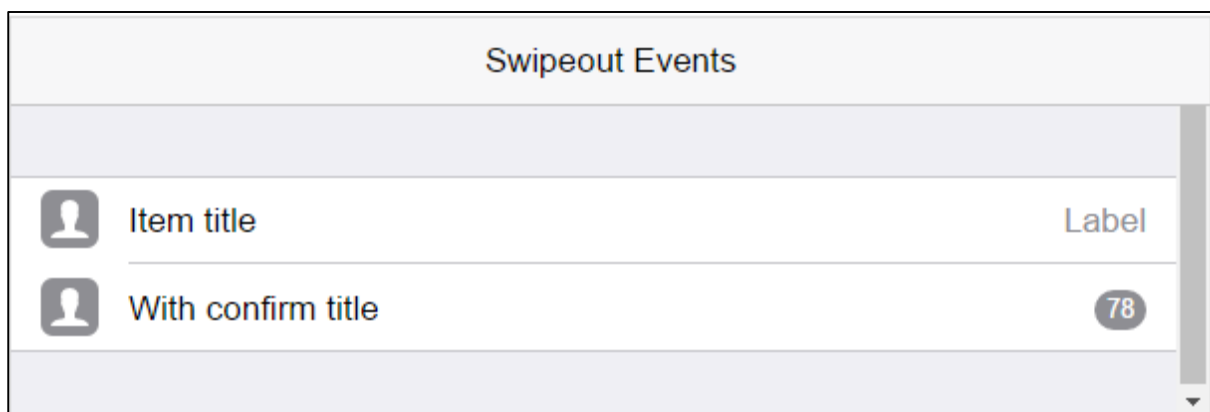
  var $$ = Dom7;
  $$('.swipeout').on('deleted', function () {
    myApp.alert('Item removed');
  });
  $$('.swipeout').on('opened', function () {
    myApp.alert('Item opened');
  });
  $$('.swipeout').on('closed', function () {
    myApp.alert('Item closed');
  });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **list_views_swipe_events.html** file in your server root folder.
- Open this HTML file as http://localhost/list_views_swipe_events.html and the output is displayed as shown below.
- The example provides user interaction with the swipeout by using the swipeout events and these events will be triggered when you move, open, close and delete the swipeout element.



Framework7 – Sortable List

Description

The sortable list is a type of list view, which sorts the list view elements.

You can enable and disable the Sortable in two ways as specified in the following table:

S. No.	Type & Description
1	From HTML You can enable and disable sortable by using the classes and data attributes.
2	Using JavaScript You can enable and disable sortable by using the JavaScript code.

Framework7 – Overlay Sortable List from HTML

Description

You can enable and disable sortable by using the classes and data attributes as shown below:

S. No.	Type & Description
1	open-sortable It is required to enable sortable mode to any HTML element.
2	close-sortable It is required to disable sortable mode.
3	toggle-sortable It is required to toggle sortable mode.
4	data-sortable=".sortable" Whenever your app has more than one sortable list in app, you need to specify appropriate sortable list by using this attribute as additional attribute on HTML element.

Example

The following example specifies enabling, disabling and toggling of the sorting of the HTML element on links in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
```

```

<meta name="apple-mobile-web-app-status-bar-style" content="black">
<title>Sortable from HTML</title>

<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">

<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
<div class="views">
<div class="view view-main">
<div class="navbar">
<div class="navbar-inner">
<div class="center sliding">Sortable from HTML</div>
</div>
</div>
<div class="pages navbar-through">
<div data-page="index" class="page navbar-fixed">
<div class="page-content">
<div class="content-block">
<p><a href="#" data-sortable=".sortable" class="open-
sortable">Click here to enable sortable</a></p>
<p><a href="#" data-sortable=".sortable" class="close-
sortable">Click here to disable sortable</a></p>
<p><a href="#" data-sortable=".sortable" class="toggle-
sortable">Click here to toggle sortable</a></p>
</div>
<div class="list-block sortable">
<ul>
<li>
<div class="item-content">
<div class="item-media"><i class="icon icon-
form-name"></i></div>
<div class="item-inner">
<div class="item-title">Sachin</div>
<div class="item-after">India</div>
</div>
</div>
<div class="sortable-handler"></div>

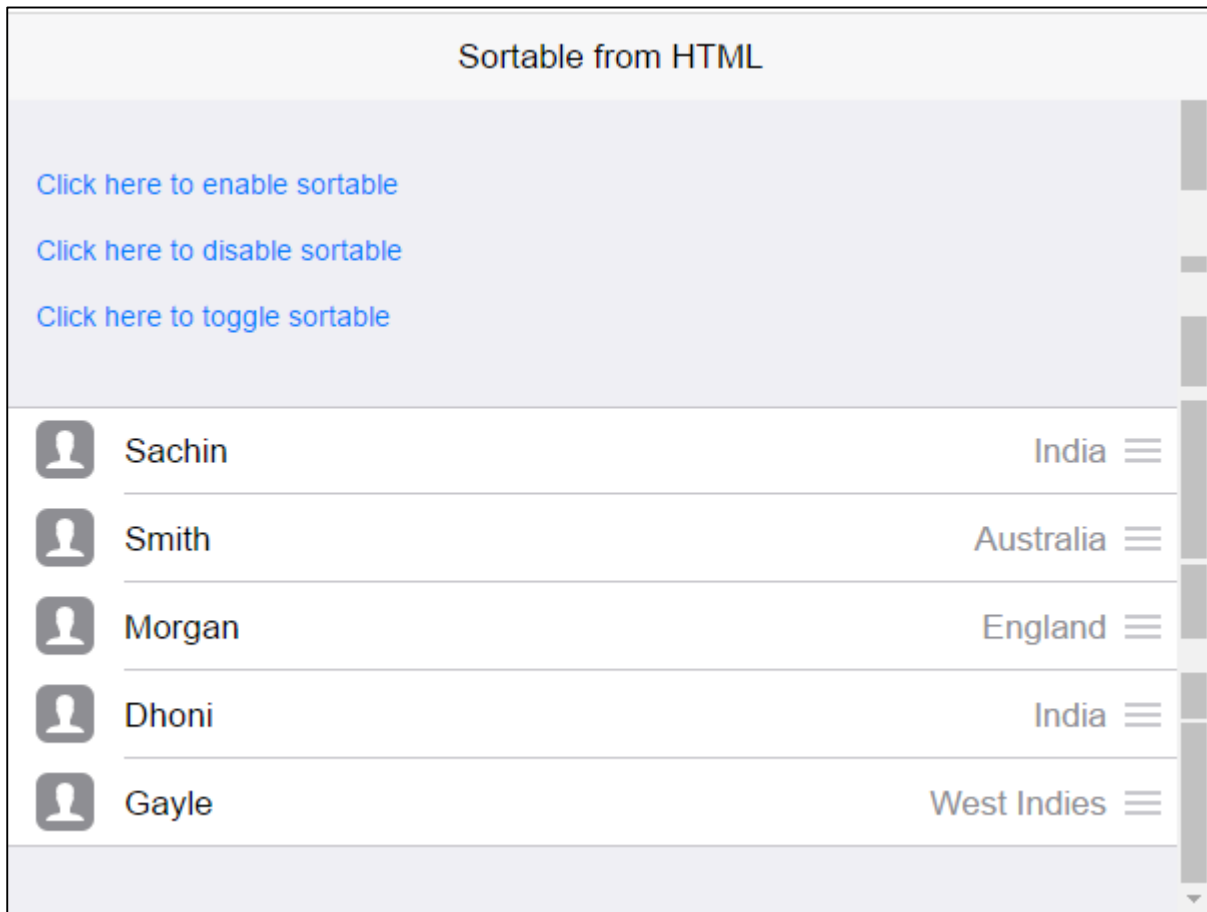
```

```

        </li>
        <li>
            <div class="item-content">
                <div class="item-media"><i class="icon icon-
form-name"></i></div>
                <div class="item-inner">
                    <div class="item-title">Smith</div>
                    <div class="item-after">Australia</div>
                </div>
            </div>
            <div class="sortable-handler"></div>
        </li>
        <li>
            <div class="item-content">
                <div class="item-media"><i class="icon icon-
form-name"></i></div>
                <div class="item-inner">
                    <div class="item-title">Morgan</div>
                    <div class="item-after">England</div>
                </div>
            </div>
            <div class="sortable-handler"></div>
        </li>
        <li>
            <div class="item-content">
                <div class="item-media"><i class="icon icon-
form-name"></i></div>
                <div class="item-inner">
                    <div class="item-title">Dhoni</div>
                    <div class="item-after">India</div>
                </div>
            </div>
            <div class="sortable-handler"></div>
        </li>
        <li>
            <div class="item-content">
                <div class="item-media"><i class="icon icon-
form-name"></i></div>
                <div class="item-inner">

```


- Open this HTML file as http://localhost/enable_disable_sortable_from_html.html and the output is displayed as shown below.
- The example specifies enabling, disabling and toggling of the sorting of the HTML element on links.



Framework7 – Overlay Sortable List from JavaScript

Description

You can enable and disable sortable by using the JavaScript App methods as shown below:

- **myApp.sortableOpen(sortableContainer)** - It is used to enable sorting mode on specified sortable container.
- **myApp.sortableClose(sortableContainer)** - It is used to disable sorting mode on specified sortable container.
- **myApp.sortableToggle(sortableContainer)** - It is used to toggle sorting mode on specified sortable container.

The above methods accept **sortableContainer** parameter, which is optional *HTML Element* or *string* of sortable container.

Example

The following example specifies the sorting of the HTML element using JavaScript in Framework7:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Sortable using JavaScript</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="center sliding">Sortable using JavaScript</div>
          </div>
        </div>
        <div class="pages navbar-through">
          <div data-page="index" class="page navbar-fixed">
            <div class="page-content">
              <div class="content-block">
                <p><a href="#" class="open-sortable">Enable sortable</a></p>
                <p><a href="#" class="close-sortable">Disable sortable</a></p>
                <p><a href="#" class="toggle">Toggle sortable</a></p>
              </div>
              <div class="list-block sortable">
                <ul>
                  <li>
                    <div class="item-content">
                      <div class="item-media"><i class="icon icon-
form-name"></i></div>

```



```

        <div class="item-inner">
            <div class="item-title">Sachin</div>
            <div class="item-after">India</div>
        </div>
    </div>
    <div class="sortable-handler"></div>
</li>
<li>
    <div class="item-content">
        <div class="item-media"><i class="icon icon-
form-name"></i></div>
        <div class="item-inner">
            <div class="item-title">Smith</div>
            <div class="item-after">Australia</div>
        </div>
    </div>
    <div class="sortable-handler"></div>
</li>
<li>
    <div class="item-content">
        <div class="item-media"><i class="icon icon-
form-name"></i></div>
        <div class="item-inner">
            <div class="item-title">Morgan</div>
            <div class="item-after">England</div>
        </div>
    </div>
    <div class="sortable-handler"></div>
</li>
<li>
    <div class="item-content">
        <div class="item-media"><i class="icon icon-
form-name"></i></div>
        <div class="item-inner">
            <div class="item-title">Dhoni</div>
            <div class="item-after">India</div>
        </div>
    </div>
</li>

```

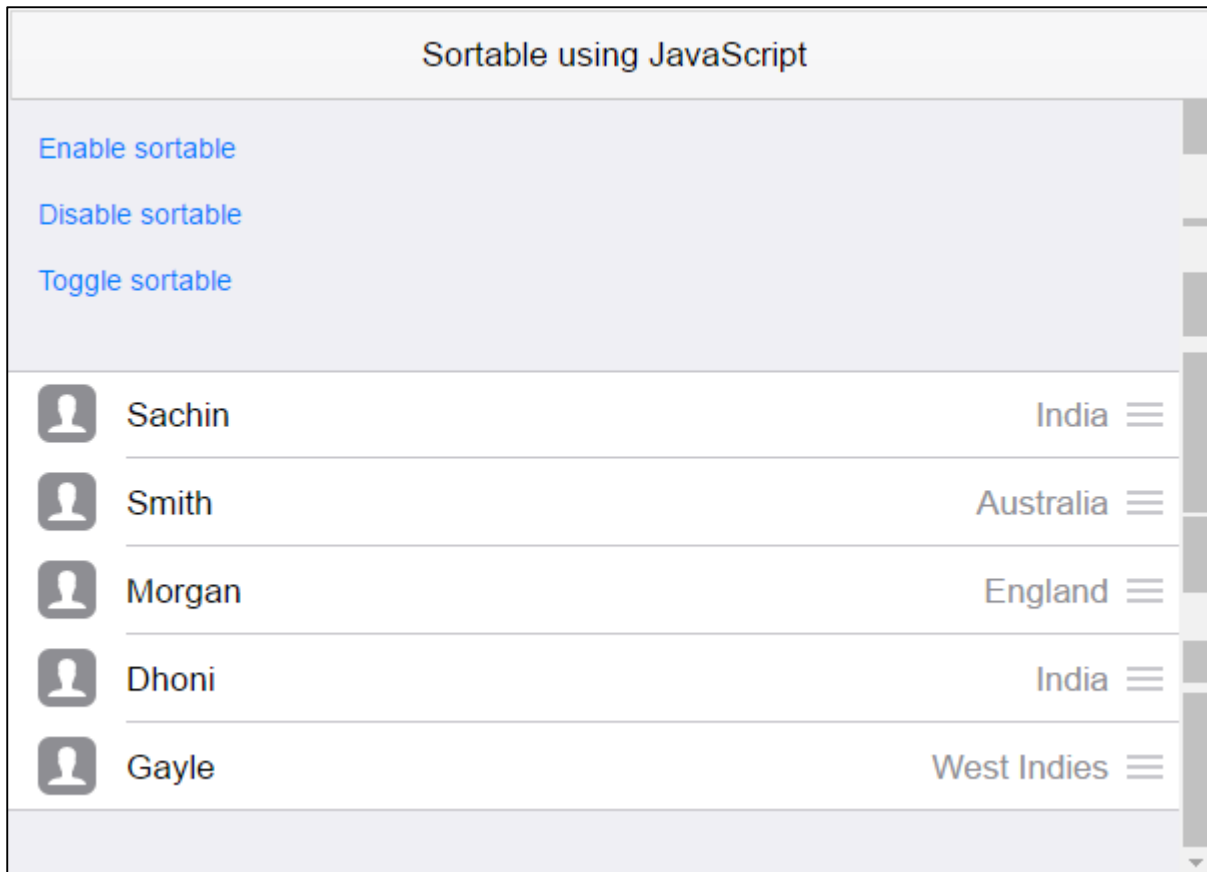


```
    $$('.close-sortable').on('click', function () {
        myApp.sortableClose('.sortable');
    });
    $$('.toggle').on('click', function () {
        myApp.sortableToggle('.sortable');
    });
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **enable_disable_sortable_using_javascript.html** file in your server root folder.
- Open this HTML file as http://localhost/enable_disable_sortable_using_javascript.html and the output is displayed as shown below.
- The example specifies enabling, disabling and toggling of the sorting of the HTML element on links.



Sortable Events

The following table shows a list of events used in *sortable list*:

S. No.	Event	Target & Description
1	open	Sortable Container When you enable the sortable, this event will get fired.
2	close	Sortable Container When you disable the sortable, this event will get fired.
3	sort	List view Element When you place the sorted element in new position, this event will get fired.

Framework7 – Virtual List

Description

Virtual list is a type of list view, which includes large number of data elements without dropping their performance. You can create HTML layout of virtual list by using the *virtual-list* class along with the *list-block* class.

Initialize Virtual List

You can initialize the virtual list by using the following method:

```
myApp.virtualList(listBlockContainer, parameters)
```

The method contains the following parameters:

- **listBlockContainer:** It is a required HTML or string element for the list block container.
- **parameters:** It is a required object, which includes virtual list parameters.

Virtual List Parameters

The following table provides a list of virtual parameters:

S. No.	Parameter & Description	Type	Default
1	items Provides a list of array items.	array	-
2	rowsBefore It defines the number of rows to be displayed before scrolling the screen position.	number	-
3	rowsAfter It defines the number of rows to be displayed after scrolling the screen position.	number	-
4	cols It specifies the number of items per row.	number	1
5	height This parameter returns item height in px.	number or function(item)	44
6	template It represents the single item.	string or function	-
7	renderItem It uses custom function to display the item HTML.	function(index, item)	-
8	dynamicHeightBufferSize It specifies the buffer size on virtual list along with the dynamic height.	number	1
9	cache You can enable or disable the DOM cache for list of items.	boolean	true
10	updatableScroll It updates the device and manipulates the scroll events when you scroll the page.	boolean	-

11	showFilteredItemsOnly It displays the filtered items by using the filter() method.	boolean	false
12	searchByItem It is used to search the item by using search bar and uses search query, item index and item itself as parameters.	function(query, index, item)	-
13	searchAll It is used to search all items by using search bar and uses search query and array of items as parameters.	function(query, items)	-
14	onItemBeforeInsert It executes the callback function before inserting item to virtual document fragment.	function(list, item)	
15	onBeforeClear It executes the callback function before removing DOM list and replaced with new document fragment.	function(list, item)	
16	onItemsBeforeInsert It executes the callback function after removing DOM list and before inserting new document fragment.	function(list, item)	
17	onItemsAfterInsert It executes the callback function after inserting the items with new document fragment.	function(list, item)	

Virtual List Properties

S. No.	Property & Description
1	myList.items It displays the array with items.
2	myList.filteredItems It displays the array with filtered items.
3	myList.domCache It represents the items with DOM cache.
4	myList.params It displays the parameters that are passed on initialization.
5	myList.listBlock It specifies the list block container of DOM7 instance.

6	myList.pageContent It specifies the page content container of DOM7 instance.
7	myList.currentFromIndex It displays the first rendered item.
8	myList.currentToIndex It displays the last rendered item.
9	myList.reachEnd It displays the last item of all specified items if it is true.

Virtual List Methods

S. No.	Method & Description
1	myList.filterItems(indexes); You can filter the items by using array with indexes.
2	myList.resetFilter(); Displays all the items by preventing the filter.
3	myList.appendItem(item); It appends the items to a virtual list.
4	myList.appendItems(items); It appends the array of items to a virtual list.
5	myList.prependItem(item); It prepends the items to a virtual list.
6	myList.prependItems(items); It prepends the array of items to a virtual list.
7	myList.replaceItem(index, items); It replaces the item with new item at the specified index.
8	myList.replaceAllItems(items); It replaces all the items with the new items.
9	myList.moveItem(oldIndex, newIndex); It transfers items from old to new index.
10	myList.insertItemBefore(index, item); You can insert the item before the specified index.
11	myList.deleteItem(index); You can delete the item at the specified index.
12	myList.deleteItems(indexes); You can delete the items at the specified array of indexes.

13	myList.deleteAllItems(); It deletes all the items.
14	myList.clearCache(); It is used to clear the cache of DOM elements.
15	myList.destroy(); It destroys the virtual list and its events.
16	myList.update(); It updates the virtual list and re-renders the virtual list.
17	myList.scrollToItem(index); You can scroll the virtual list to item by using the index number.

Templating

Sometimes you need to filter or load the items from JSON data using some logic. To do this, you can pass array with data objects using *items* parameter and *template* parameter or using *renderItem* parameter.

You can use Framework7 *template* parameter as follows:

```
var myVal = myApp.virtualList('.list-block.virtual-list', {
  //List of array items
  items: [
    {
      name: 'Element 1',
      image: '/path/image1.jpg'
    },
    {
      name: 'Element 2',
      image: '/path/image2.jpg'
    },
    // ...
    {
      name: 'Element 50',
      image: '/path/image50.jpg'
    },
  ],
  // Display the each item using Template7 template parameter
  template: '<li class="item-content">' +
    '<div class="item-media"></div>' +
    '<div class="item-inner">' +
```



```

        '<div class="item-title">{{name}}</div>' +
        '</div>' +
        '</li>'
    });

```

You can also use custom render function as shown below:

```

var myVal = myApp.virtualList('.list-block.virtual-list', {
    //List of array items
    items: [
        {
            name: 'Element 1',
            image: '/path/image1.jpg'
        },
        {
            name: 'Element 2',
            image: '/path/image2.jpg'
        },
        // ...
        {
            name: 'Element 50',
            image: '/path/image50.jpg'
        },
    ],
    // Display the each item using custom render function
    renderItem: '<li class="item-content">' +
        '<div class="item-media"></div>' +
        '<div class="item-inner">' +
            '<div class="item-title">{{name}}</div>' +
            '</div>' +
        '</li>'
    });

```

Using With Search Bar

You can use *searchAll* or *searchByItem* parameters in order to use the search bar with virtual list that supply the search function in parameters.

```

var myVal = myApp.virtualList('.list-block.virtual-list', {
    //List of array items

```

```

    items: [
      {
        name: 'Element 1',
        image: '/path/image1.jpg'
      },
      {
        name: 'Element 2',
        image: '/path/image2.jpg'
      },
      // ...
      {
        name: 'Element 50',
        image: '/path/image50.jpg'
      },
    ],
    //Here you can searches all items in array and send back array with matched
    items
    searchAll: function (query, items) {
      var myItems = [];
      for (var i = 0; i < items.length; i++) {
        // Here check if name contains query string
        if (items[i].name.indexOf(query.trim()) >= 0) myItems.push(i);
      }
      // Returns array with indexes of matched items
      return myItems;
    }
  });

```

Using *searchByItem* parameter:

```

var myVal = myApp.virtualList('.list-block.virtual-list', {
  //List of array items
  items: [
    {
      name: 'Element 1',
      image: '/path/image1.jpg'
    },
    {
      name: 'Element 2',

```

```

        image: '/path/image2.jpg'
    },
    // ...
    {
        name: 'Element 50',
        image: '/path/image50.jpg'
    },
],
//Here you can searches all items in array and send back array with matched
items
searchByItem: function (query, index, items) {
    // Here check if name contains query string
    if (items[i].name.indexOf(query.trim()) >= 0){
        return true;
    }
    {
        else{
            return false;
        }
    }
}
});

```

Dynamic Height

You can use dynamic height for the items by using the *height* parameter instead of a number.

```

var myVal = myApp.virtualList('.list-block.virtual-list', {
    //List of array items
    items: [
        {
            name: 'Element 1',
            image: '/path/image1.jpg'
        },
        {
            name: 'Element 2',
            image: '/path/image2.jpg'
        },
        // ...
    ]
});

```

```

    {
      name: 'Element 50',
      image: '/path/image50.jpg'
    },
  ],
  //template parameter
  template: '...',

  //using height function
  height: function (item) {
    if (item.image) return 120; //display the image with 100px height
    else return 50; //display the image with 50px height
  }
});

```

API Methods

You can use the API methods to add, remove, replace or move virtual list items.

```

//Here you can initialize the list
var myVal = myApp.virtualList('.list-block.virtual-list', {
  //List of array items
  items: [
    {
      name: 'Element 1',
      image: '/path/image1.jpg'
    },
    {
      name: 'Element 2',
      image: '/path/image2.jpg'
    },
    // ...
    {
      name: 'Element 50',
      image: '/path/image50.jpg'
    },
  ],
  //template parameter
  template: '...',

```

```
});

//Here append your item
myVal.appendItem({
  image: 'Element 55'
});

// You can append multiple items when clicking on button
$('.button.append-items').on('click', function () {
  //You can append multiple items by passing array with items
  myVal.appendItem([
    {
      image: 'Element 60'
    },
    {
      image: 'Element 61'
    },
    {
      image: 'Element 62'
    }
  ]);
});

//replace the first item
myList.replaceItem(0, {
  name: 'Element 4'
});

//you can display first 10 items when clicking on button
$('.button.show-first-10').on('click', function () {
  //Passing array with indexes to show items
  myList.filter([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]);
});

//Reset the filter
```

```
$('.button.reset-filter').on('click', function () {  
    myList.resetFilter();  
});  
//You can insert the item before 4th and 5th item  
myList.insertItemBefore(1, {  
    name: 'Element 4.5'  
});
```

15. Framework7 – Accordion

Description

The accordion is a graphical control element displayed as a stacked list of items. Each accordion can be *expanded* or *stretched* to reveal the content associated with that accordion.

Collapsible Layout

When you are using single separate collapsible element, you need to omit the *accordion-list* wrapper element.

Following is a structure of collapsible layout:

```
<!-- Single collapsible element ----->
<div class="accordion-item">
  <div class="accordion-item-toggle"></div>
  <div class="accordion-item-content"></div>
</div>
<!-- Separate collapsible element -->
<div class="accordion-item">
  <div class="accordion-item-toggle"></div>
  <div class="accordion-item-content"></div>
</div>
</div>
```

The following classes are used for accordion in Framework7:

S. No.	Classes & Description
1	accordion-list It is an optional class, which contains a group of accordion items list.
2	accordion-item It is a required class for single accordion item.
3	accordion-item-toggle It is a required class used to expand accordion item content.
4	accordion-item-content It is a required class used for hidden accordion item content.
5	accordion-item-expanded It is a single expanded accordion item.

Accordion JavaScript API

JavaScript API methods are used to open and close the accordion programmatically.

It contains the following JavaScript API methods:

- **myApp.accordionOpen(item):** Used to open accordion.
- **myApp.accordionClose(item):** Used to close accordion.
- **myApp.accordionToggle(item):** Used to toggle accordion.

All methods contains parameter called *item*, which is HTML or string element of accordion-item.

Accordion Events

Accordion contains four events as listed in the following table:

S. No.	Event	Target & Description
1	open	Accordion item When you open an animation, this event will get fired.
2	opened	Accordion item When the opening of an animation is completed, this event will get fired.
3	close	Accordion item When you close an animation, this event will get fired.
4	closed	Accordion item When the closing of an animation is completed, this event will get fired.

Accordion List View

In accordion list view, you can use *item-link* element instead of *accordion-toggle*.

```
<div class="list-block accordion-list">
  <ul>
    <li class="accordion-item">
      <a href="" class="item-link item-content">
        <div class="item-inner">
          <div class="item-title">1st Item</div>
        </div>
      </a>
      <div class="accordion-item-content">Content for 1st Item...</div>
    </li>
    <li class="accordion-item">
```



```

    <a href="" class="item-link item-content">
      <div class="item-inner">
        <div class="item-title">2nd Item</div>
      </div>
    </a>
    <div class="accordion-item-content">Content for 2nd Item...</div>
  </li>
</ul>
</div>

```

Example

The following example demonstrates the use of accordion in the Framework7:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Accordion</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Accordion</div>
                <div class="right"> </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>

```

```

<div class="page-content">
  <div class="content-block-title">List of Programming
Lagauges</div>
  <div class="list-block accordion-list">
    <ul>
      <li class="accordion-item">
        <a href="#" class="item-content item-link">
          <div class="item-inner">
            <div class="item-title">C</div>
          </div>
        </a>
        <div class="accordion-item-content">
          <div class="content-block">
            <p>C is a general-purpose, procedural,
imperative computer programming language developed in 1972 by Dennis M. Ritchie
at the Bell Telephone Laboratories to develop the UNIX operating system. C is
the most widely used computer language. It keeps fluctuating at number one
scale of popularity along with Java programming language, which is also equally
popular and most widely used among modern software programmers.</p>
          </div>
        </div>
      </li>
      <li class="accordion-item">
        <a href="#" class="item-content item-link">
          <div class="item-inner">
            <div class="item-title">C++</div>
          </div>
        </a>
        <div class="accordion-item-content">
          <div class="content-block">
            <p>C++ is a middle-level programming
language developed by Bjarne Stroustrup starting in 1979 at Bell Labs. C++ runs
on a variety of platforms, such as Windows, Mac OS, and the various versions of
UNIX.

            This reference will take you through
simple and practical approach while learning C++ Programming language.
          </p>
          </div>
        </div>
      </li>
      <li class="accordion-item">

```

```

    <a href="#" class="item-content item-link">
      <div class="item-inner">
        <div class="item-title">Java</div>
      </div>
    </a>
    <div class="accordion-item-content">
      <div class="content-block">
        <p>Java is a high-level programming
language originally developed by Sun Microsystems and released in 1995. Java
runs on a variety of platforms, such as Windows, Mac OS, and the various
versions of UNIX. This tutorial gives a complete understanding of Java.

        This reference will take you through
simple and practical approach while learning Java Programming language.

        </p>
      </div>
    </div>
  </li>
</ul>
</div>
<div class="content-block-title">Separate
Collapsibles</div>
<div class="list-block">
  <ul>
    <li class="accordion-item">
      <a href="#" class="item-content item-link">
        <div class="item-inner">
          <div class="item-title">C</div>
        </div>
      </a>
      <div class="accordion-item-content">
        <div class="content-block">
          <p >C is a general-purpose, procedural,
imperative computer programming language developed in 1972 by Dennis M. Ritchie
at the Bell Telephone Laboratories to develop the UNIX operating system. C is
the most widely used computer language. It keeps fluctuating at number one
scale of popularity along with Java programming language, which is also equally
popular and most widely used among modern software programmers.</p>
        </div>
      </div>
    </li>
    <li class="accordion-item">

```

```

<a href="#" class="item-content item-link">
  <div class="item-inner">
    <div class="item-title">C++</div>
  </div>
</a>
<div class="accordion-item-content">
  <div class="content-block">
    <p>C++ is a middle-level programming
language developed by Bjarne Stroustrup starting in 1979 at Bell Labs. C++ runs
on a variety of platforms, such as Windows, Mac OS, and the various versions of
UNIX.

    This reference will take you through
simple and practical approach while learning C++ Programming language.
    </p>
  </div>
</div>
</li>
<li class="accordion-item">
  <a href="#" class="item-content item-link">
    <div class="item-inner">
      <div class="item-title">Java</div>
    </div>
  </a>
  <div class="accordion-item-content">
    <div class="content-block">
      <p>Java is a high-level programming
language originally developed by Sun Microsystems and released in 1995. Java
runs on a variety of platforms, such as Windows, Mac OS, and the various
versions of UNIX. This tutorial gives a complete understanding of Java.

      This reference will take you through
simple and practical approach while learning Java Programming language.
      </p>
    </div>
  </div>
</li>
</ul>
</div>
<div class="content-block-title">Custom Accordion</div>
<div class="content-block accordion-list custom-
accordion">
  <div class="accordion-item">

```

```

        <div class="accordion-item-toggle"><i class="icon
icon-plus">+</i><i class="icon icon-minus">-</i><span>C</span></div>
        <div class="accordion-item-content">
            <p>C is a general-purpose, procedural, imperative
computer programming language developed in 1972 by Dennis M. Ritchie at the
Bell Telephone Laboratories to develop the UNIX operating system. C is the most
widely used computer language. It keeps fluctuating at number one scale of
popularity along with Java programming language, which is also equally popular
and most widely used among modern software programmers.</p>
        </div>
    </div>
    <div class="accordion-item">
        <div class="accordion-item-toggle"><i class="icon
icon-plus">+</i><i class="icon icon-minus">-</i><span>C++</span></div>
        <div class="accordion-item-content">
            <p>C++ is a middle-level programming language
developed by Bjarne Stroustrup starting in 1979 at Bell Labs. C++ runs on a
variety of platforms, such as Windows, Mac OS, and the various versions of
UNIX.

            This reference will take you through simple
and practical approach while learning C++ Programming language.
        </p>
        </div>
    </div>
    <div class="accordion-item">
        <div class="accordion-item-toggle"><i class="icon
icon-plus">+</i><i class="icon icon-minus">-</i><span>Java</span></div>
        <div class="accordion-item-content">
            <p>Java is a high-level programming language
originally developed by Sun Microsystems and released in 1995. Java runs on a
variety of platforms, such as Windows, Mac OS, and the various versions of
UNIX. This tutorial gives a complete understanding of Java.

            This reference will take you through simple
and practical approach while learning Java Programming language.
        </p>
        </div>
    </div>
</div>

```

```

    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>

    <style>.custom-accordion{padding-left:0;padding-right:0;}.custom-
accordion .accordion-item-toggle{padding:0px 15px;height:44px;line-
height:44px;font-size:17px;color:#000;border-bottom:1px solid
rgba(0,0,0,0.15);cursor:pointer;}.custom-accordion .accordion-item-
toggle.active{background:rgba(0,0,0,0.15);}.custom-accordion .accordion-item-
toggle span{display:inline-block;margin-left:15px;}.custom-accordion
.accordion-item:last-child .accordion-item-toggle{border-bottom:none;}.custom-
accordion .icon-plus,.custom-accordion .icon-minus{display:inline-
block;width:22px;height:22px;border:1px solid #000;border-radius:100%;line-
height:20px;text-align:center;}.custom-accordion .icon-
minus{display:none;}.custom-accordion .accordion-item-expanded .icon-
minus{display:inline-block;}.custom-accordion .accordion-item-expanded .icon-
plus{display:none;}.custom-accordion .accordion-item-content{padding:0px
15px;}</style>

    <script>
        // here initialize the app
        var myApp = new Framework7();

        // If your using custom DOM library, then save it to $$ variable
        var $$ = Dom7;

        // Add the view
        var mainView = myApp.addView('.view-main', {
            // enable the dynamic navbar for this view:
            dynamicNavbar: true
        });
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **accordion.html** file in your server root folder.
- Open this HTML file as <http://localhost/accordion.html> and the output is displayed as shown below
- The example provides collapsible layout, which can be expanded to display the content associated with the accordion.

Accordion

LIST OF PROGRAMMING LANGUAGES

- C >
- C++ >
- C++ >
- Java >

SEPARATE COLLAPSIBLES

- C >
- C++ >
- Java >

CUSTOM ACCORDION

- + C
- + C++
- + Java

16. Framework7 – Cards

Description

Cards contain organized information related to a single subject like a photo, link, and text. The following table shows Framework7 card types:

S. No.	Types & Description
1	Card HTML Layout The basic card HTML layout uses card classes to arrange its items.
2	List View With Cards You can use <i>cards</i> as list view elements by adding <i>cards-list</i> class to <code><div class="list-block"></code> .

Framework7 – Card HTML Layout

Description

The card HTML layout contains many classes, which are listed below:

S. No.	Classes & Description
1	cards It is the card container.
2	card-header It is the optional card header that is used to display card title.
3	card-footer It is optional and used for specifying additional information or custom links.
4	card-content It is the main container for content of the card and is required.
5	card-content-inner It is the optional additional inner wrapper that is used for additional padding to content.

Both the *card-header* and the *card-footer* has *flexbox layout* where items have center vertical alignment. If you want to order your items to top or to bottom of header/footer, then you can use *valign* attribute.

Example

The following example demonstrates the use of cards HTML layouts in Framework7:

```

<!DOCTYPE html>
<html class="with-statusbar-overlay">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Card HTML Layout</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
    <link rel="stylesheet" href="custom.css">
  </head>
  <body>
    <style>.demo-card-header-pic .card-header{height:40vw;background-
size:cover;background-position:center;}.facebook-card .card-
header{display:block;padding:10px;}.facebook-card .facebook-
avatar{float:left;}.facebook-card .facebook-name{margin-left:44px;font-
size:14px;font-weight:500;}.facebook-card .facebook-date{margin-left:44px;font-
size:13px;color:#8e8e93;}.facebook-card .card-
footer{background:#fafafa;}.facebook-card .card-footer a{color:#81848b;font-
weight:500;}.facebook-card .card-content img{display:block;}.facebook-card
.card-content-inner{padding:15px 10px;}</style>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Card HTML Layout</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">
              <div class="content-block-title">Simple Cards</div>
              <div class="card">

```

```

        <div class="card-content">
            <div class="card-content-inner">Lorem ipsum dolor
sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua.</div>
        </div>
    </div>
    <div class="card">
        <div class="card-header">Card header</div>
        <div class="card-content">
            <div class="card-content-inner">Lorem ipsum dolor
sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua.</div>
        </div>
        <div class="card-footer">Card Footer</div>
    </div>
    <div class="card">
        <div class="card-content">
            <div class="card-content-inner">Another card. Lorem
ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incidunt ut labore et dolore magna aliqua. </div>
        </div>
    </div>
    <div class="content-block-title">Styled Cards</div>
    <div class="card demo-card-header-pic">
        <div style="background-
image:url(/framework7/images/nature.jpg)" valign="bottom" class="card-header
color-white no-border">Beautiful Mountains</div>
        <div class="card-content">
            <div class="card-content-inner">
                <p class="color-gray">Posted on May 20, 2015</p>
                <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.</p>
            </div>
        </div>
        <div class="card-footer"><a href="#"
class="link">Like</a><a href="#" class="link">Comment</a></div>
    </div>
    <div class="card facebook-card">
        <div class="card-header">
            <div class="facebook-avatar"></div>

```

```

        <div class="facebook-name">John Doe</div>
        <div class="facebook-date">Monday at 2:15 PM</div>
    </div>
    <div class="card-content">
        <div class="card-content-inner">
            <p>What a nice view!!</p>
            
            <p class="color-gray">Likes: 112    Comments:
43</p>
        </div>
    </div>
    <div class="card-footer"><a href="#"
class="link">Like</a><a href="#" class="link">Comment</a><a href="#"
class="link">Share</a></div>
</div>
<div class="content-block-title">Cards With List
View</div>
<div class="card">
    <div class="card-content">
        <div class="list-block">
            <ul>
                <li>
                    <a href="#" class="item-link item-content">
                        <div class="item-media"><i class="icon
icon-form-name"></i></div>
                        <div class="item-inner">
                            <div class="item-title">Link 1</div>
                        </div>
                    </a>
                </li>
                <li>
                    <a href="#" class="item-link item-content">
                        <div class="item-media"><i class="icon
icon-form-name"></i></div>
                        <div class="item-inner">
                            <div class="item-title">Link 2</div>
                        </div>
                    </a>
                </li>
            </ul>
        </div>
    </div>
</div>

```

```

        <li>
            <a href="#" class="item-link item-content">
                <div class="item-media"><i class="icon
icon-form-name"></i></div>
                <div class="item-inner">
                    <div class="item-title">Link 3</div>
                </div>
            </a>
        </li>
        <li>
            <a href="#" class="item-link item-content">
                <div class="item-media"><i class="icon
icon-form-name"></i></div>
                <div class="item-inner">
                    <div class="item-title">Link 4</div>
                </div>
            </a>
        </li>
        <li>
            <a href="#" class="item-link item-content">
                <div class="item-media"><i class="icon
icon-form-name"></i></div>
                <div class="item-inner">
                    <div class="item-title">Link 5</div>
                </div>
            </a>
        </li>
    </ul>
</div>
</div>
</div>
<div class="card">
    <div class="card-header">Albums:</div>
    <div class="card-content">
        <div class="list-block media-list">
            <ul>
                <li class="item-content">
                    <div class="item-media"></div>

```

```

                <div class="item-inner">
                    <div class="item-title-row">
                        <div class="item-title">Cast
away</div>
                    </div>
                    <div class="item-subtitle">James</div>
                </div>
            </li>
            <li class="item-content">
                <div class="item-media"></div>
                <div class="item-inner">
                    <div class="item-title-row">
                        <div class="item-title">Don't Follow
me</div>
                    </div>
                    <div class="item-subtitle">Rema
Taylor</div>
                </div>
            </li>
            <li class="item-content">
                <div class="item-media"></div>
                <div class="item-inner">
                    <div class="item-title-row">
                        <div class="item-title">Billion
words</div>
                    </div>
                    <div class="item-subtitle">Kaveen
sharma</div>
                </div>
            </li>
        </ul>
    </div>
    <div class="card-footer"> <span>May 1,
2016</span><span>45 comments</span></div>
</div>
</div>
</div>

```

```
        </div>
    </div>

</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>var myApp = new Framework7();</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **cards_html_layout.html** file in your server root folder.
- Open this HTML file as http://localhost/cards_html_layout.html and the output is displayed as shown below.
- The example defines the card layout, which contains unique related data, such as photo, text, link.

Card HTML Layout

SIMPLE CARDS

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor


Card header

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua

Card Footer

Another card. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua

STYLED CARDS




Beautiful Mountains


Posted on May 20, 2015

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua

[Like](#) [Comment](#)

 **John Doe**
Monday at 2:15 PM





What a nice view!





Likes: 112 · Comments: 43

[Like](#) [Comment](#) [Share](#)

CARDS WITH LIST VIEW

-  [Link 1](#) >
-  [Link 2](#) >
-  [Link 3](#) >
-  [Link 5](#) >

Albums:

-  **Cast away**
James
-  **Don't Follow me**
Rema Taylor
-  **Billion words**
Kaveen sharma

May 1, 2015 43 comments

Framework7 – List View With Cards

Description

You can use *cards* as list view elements by adding *cards-list* class to `<div class="list-block">`.

Example

The following example demonstrates the use of list view with cards in Framework7:

```
<!DOCTYPE html>
<html class="with-statusbar-overlay">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>List View With Cards</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
    <link rel="stylesheet" href="custom.css">
  </head>
  <body>
    <body>
      <div class="views">
        <div class="view view-main">
          <div class="pages">
            <div data-page="home" class="page navbar-fixed">
              <div class="navbar">
                <div class="navbar-inner">
                  <div class="left"> </div>
                  <div class="center">List View With Cards</div>
                  <div class="right"> </div>
                </div>
              </div>
              <div class="page-content">
                <div class="list-block cards-list">
                  <ul>
```



```

        <li class="card">
            <div class="card-header">Card Header</div>
            <div class="card-content">
                <div class="card-content-
inner">Content</div>

            </div>
            <div class="card-footer">Card footer</div>
        </li>
        <li class="card">
            <div class="card-header">Card Header</div>
            <div class="card-content">
                <div class="card-content-
inner">Content</div>

            </div>
            <div class="card-footer">Card footer</div>
        </li>
        <li class="card">
            <div class="card-header">Card Header</div>
            <div class="card-content">
                <div class="card-content-
inner">Content</div>

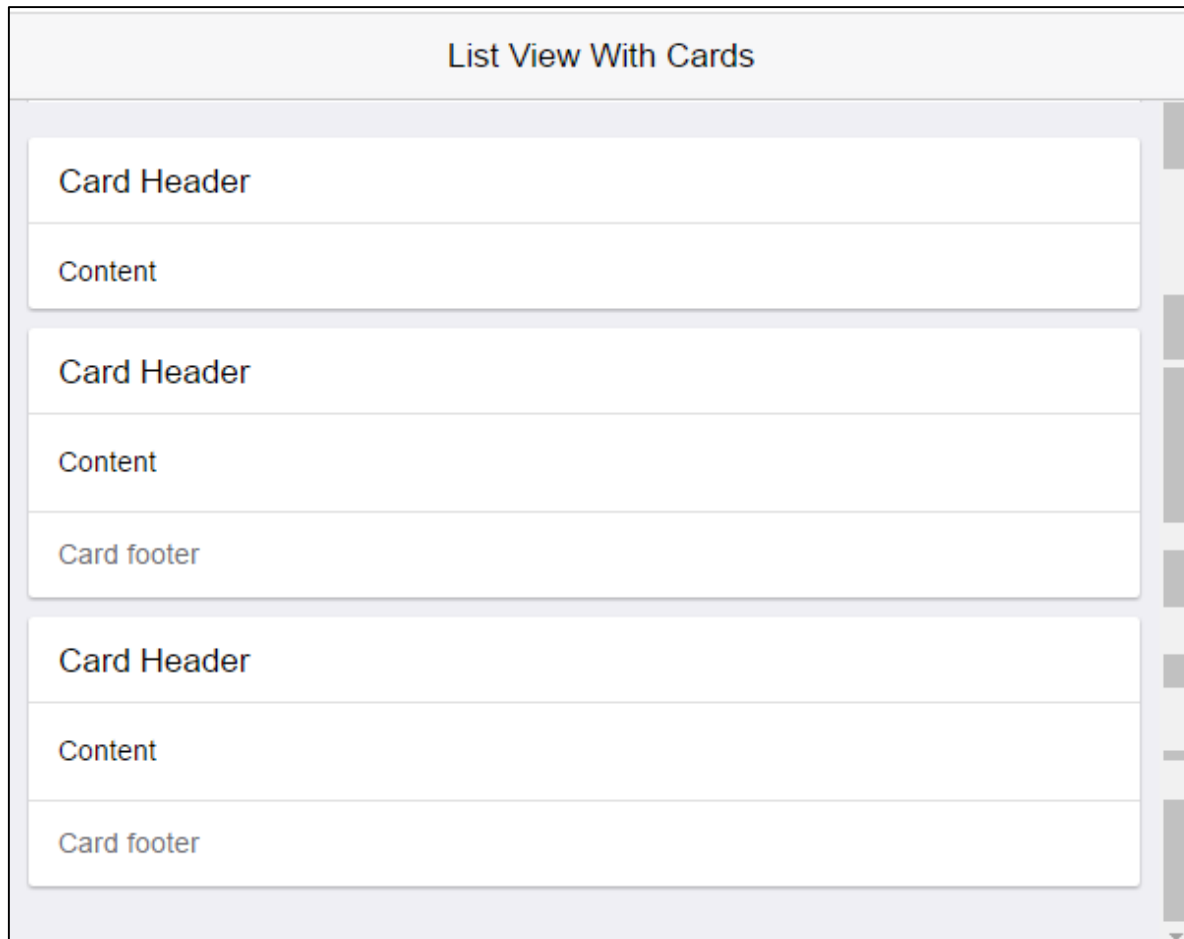
            </div>
            <div class="card-footer">Card footer</div>
        </li>
    </ul>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>var myApp = new Framework7();</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **cards_list_view.html** file in your server root folder.
- Open this HTML file as http://localhost/cards_list_view.html and the output is displayed as shown below.
- The example provides the cards as elements of list view along with the card header and the card footer.



17. Framework7 – Chips

Description

Chip is a small block of entity, which can contain a photo, small string of title, and short information.

Chips HTML Layout

The following code shows the basic chip HTML layout used in Framework7:

```
<div class="chip">
  <div class="chip-media">
    
  </div>
  <div class="chip-label">Jane Doe</div>
  <a href="#" class="chip-delete"></a>
</div>
```

The above HTML layout contains many classes as listed below:

- **chips** - It is the chip container.
- **chip-media** - This is the chip media element that can contain images, avatar or icon. It is optional.
- **card-label** - It is the chip text label.
- **card-delete** - It is the optional delete icon link of a chip.

Example

The following example represents the entities such as albums, card elements etc. along with a photo and brief information:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Chips HTML Layout</title>
```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.min.css">

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.colors.min.css">
</head>
<body>
  <div class="views">
    <div class="view view-main">
      <div class="pages">
        <div class="page navbar-fixed">
          <div class="navbar">
            <div class="navbar-inner">
              <div class="center">Chips HTML Layout</div>
            </div>
          </div>
          <div class="page-content">
            <div class="content-block-title">Chips With Text</div>
            <div class="content-block">
              <div class="chip">
                <div class="chip-label">Chip one</div>
              </div>
              <div class="chip">
                <div class="chip-label">Chip two</div>
              </div>
            </div>
            <div class="content-block-title">Chips with icons</div>
            <div class="content-block">
              <div class="chip">
                <div class="chip-media bg-blue"><i class="icon icon-
form-calendar"></i></div>
                <div class="chip-label">Set Date</div>
              </div>
              <div class="chip">

```

```

                <div class="chip-media bg-purple"><i class="icon
icon-form-email"></i></div>
                <div class="chip-label">Sent Mail</div>
            </div>
        </div>
        <div class="content-block-title">Contact Chips</div>
        <div class="content-block">
            <div class="chip">
                <div class="chip-media"></div>
                <div class="chip-label">James Willsmith</div>
            </div>
            <div class="chip">
                <div class="chip-media"></div>
                <div class="chip-label">Sunil Narayan</div>
            </div>
            <div class="chip">
                <div class="chip-media bg-pink">R</div>
                <div class="chip-label">Raghav</div>
            </div>
            <div class="chip">
                <div class="chip-media bg-teal">S</div>
                <div class="chip-label">Sharma</div>
            </div>
            <div class="chip">
                <div class="chip-media bg-red">Z</div>
                <div class="chip-label">Zien</div>
            </div>
        </div>
        <div class="content-block-title">Deletable Chips</div>
        <div class="content-block">
            <div class="chip">
                <div class="chip-label">Chip one</div>
                <a href="#" class="chip-delete"></a>
            </div>
            <div class="chip">
                <div class="chip-media bg-teal">S</div>
                <div class="chip-label">Sharma</div>
            </div>
        </div>
    </div>

```

```

        <a href="#" class="chip-delete"></a>
    </div>
    <div class="chip">
        <div class="chip-media bg-purple"><i class="icon
icon-form-email"></i></div>
        <div class="chip-label">Sent</div>
        <a href="#" class="chip-delete"></a>
    </div>
    <div class="chip">
        <div class="chip-media"></div>
        <div class="chip-label">James Willsmith</div>
        <a href="#" class="chip-delete"></a>
    </div>
    <div class="chip">
        <div class="chip-label">Chip two</div>
        <a href="#" class="chip-delete"></a>
    </div>
    <div class="chip">
        <div class="chip-media bg-green">R</div>
        <div class="chip-label">Raghav</div>
        <a href="#" class="chip-delete"></a>
    </div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<style>
</style>
<script>
    var myApp = new Framework7({
        material: true
    });
    var $$ = Dom7;

```

```
    $$('.chip-delete').on('click', function (e) {
        e.preventDefault();
        var chip = $$(this).parents('.chip');
        myApp.confirm('Do you want to delete this Chip?', function () {
            chip.remove();
        });
    });
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:



- Save above HTML code as **cards_html_layout.html** file in your server root folder.
- Open this HTML file as http://localhost/cards_html_layout.html and the output is displayed as shown below.
- The example represents the complex entities in small blocks such as albums, card elements, posted image, which contains photo, title string and brief information.

Chips HTML Layout




CHIPS WITH TEXT



Chip one Chip two

CHIPS WITH ICONS






 Set Date  Sent Mail






CONTACT CHIPS

 James Willsmith  Sunil Narayan  R Raghav

 S Sharma  Z Zien

DELETABLE CHIPS

Chip one   S Sharma   Sent 

 James Willsmith  Chip two   R Raghav 

18. Framework7 – Buttons

Description

Framework7 provides a group of ready to use buttons by just adding appropriate classes to links or input buttons.

S. No.	Types & Description
1	iOS Theme Buttons Framework7 provides many iOS theme buttons, which can be used by applying appropriate classes.
2	Material Theme Buttons The material theme provides many buttons to use in your application by using appropriate classes.

iOS Theme Buttons

Description

Framework7 provides many iOS theme buttons, which can be used by applying appropriate classes. The following table shows the iOS theme buttons:

S. No.	Types & Description
1	Usual Buttons It can be used by using the <i>button</i> class to any link or input button.
2	Active State You can make the button active by adding class <i>active</i> to your button.
3	Round Buttons The <i>button-round</i> class is used to add round buttons.
4	Big Buttons You can use the big button by adding <i>button-big</i> class to your input button.
5	Buttons Row / Segmented Control To get segmented buttons row, wrap the buttons with <i>buttons-row</i> class. Each button will be of equal width.
6	Fill Buttons It looks same as active button but intended to be used as a single button.
7	Color Buttons You can use 10 default color themes to color your buttons.

8	List Block Buttons It is possible to use buttons with <i>list views</i> .
9	Using Grid You can also place your buttons in grid.

Example

The following example demonstrates the use of buttons in Framework7:

```
<!DOCTYPE html>
<html class="with-statusbar-overlay">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>iOS Theme Buttons</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
    <link rel="stylesheet" href="custom.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Buttons</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">
              <div class="content-block-title">Usual Buttons</div>
              <div class="content-block">
                <p> <a href="#" class="button">Usual Button</a></p>

```

```

        </div>
        <div class="content-block-title">Active Button</div>
        <div class="content-block">
            <p> <a href="#" class="button active">Active
Button</a></p>
        </div>
        <div class="content-block-title">Round Buttons</div>
        <div class="content-block">
            <p> <a href="#" class="button button-round">Round
Button</a></p>
            <p> <a href="#" class="button button-round
active">Round Button With Active</a></p>
        </div>
        <div class="content-block-title">Big Buttons</div>
        <div class="content-block">
            <p> <a href="#" class="button button-big">Big Button
</a></p>
            <p> <a href="#" class="button button-big button-
round">Rounded Big Button </a></p>
            <p> <a href="#" class="button button-big button-round
active">Rounded Big Button and Active</a></p>
        </div>
        <div class="content-block-title">Buttons Row / Segmented
Controller</div>
        <div class="content-block">
            <p class="buttons-row"> <a href="#"
class="button">Button 1</a><a href="#" class="button">Button 2</a><a href="#"
class="button">Button 3</a></p>
            <p class="buttons-row"> <a href="#" class="button
button-round">Button 1</a><a href="#" class="button button-round">Button
2</a></p>
        </div>
        <div class="content-block-title">Fill buttons </div>
        <div class="content-block">
            <p><a href="#" class="button button-fill">Fill
Button</a></p>
            <p><a href="#" class="button button-big button-
fill">Big Fill Button </a></p>
        </div>
        <div class="content-block-title">Color buttons</div>
        <div class="content-block">
            <p class="buttons-row theme-blue">

```

```

        <a href="#" class="button active">Button 1</a><a
href="#" class="button">Button 2</a><a href="#" class="button">Button 3</a>
    </p>
    <p class="buttons-row theme-yellow">
        <a href="#" class="button">Button 1</a><a href="#"
class="button active">Button 2</a><a href="#" class="button">Button 3</a>
    </p>
    <p><a href="#" class="button button-fill color-
green">Green</a></p>
    <p><a href="#" class="button button-big button-fill
color-pink">Big Pink</a></p>
    <p><a href="#" class="button button-fill button-big
color-gray">Big Gray Button</a></p>
</div>
<div class="content-block-title">List block buttons</div>
<div class="list-block">
    <ul>
        <li><a href="#" class="item-link list-button">List
Button 1</a></li>
        <li><a href="#" class="item-link list-button">List
Button 2</a></li>
        <li><a href="#" class="item-link list-button">List
Button 3</a></li>
        <li><a href="#" class="item-link list-button">List
Button 4</a></li>
    </ul>
</div>
<div class="content-block-title">Inset list block
buttons</div>
<div class="list-block inset">
    <ul>
        <li><a href="#" class="item-link list-button">List
Button 1</a></li>
        <li><a href="#" class="item-link list-button">List
Button 2</a></li>
        <li><a href="#" class="item-link list-button">List
Button 3</a></li>
        <li><a href="#" class="item-link list-button">List
Button 4</a></li>
    </ul>
</div>
<div class="content-block-title">Using Grid</div>
<div class="content-block">

```

```

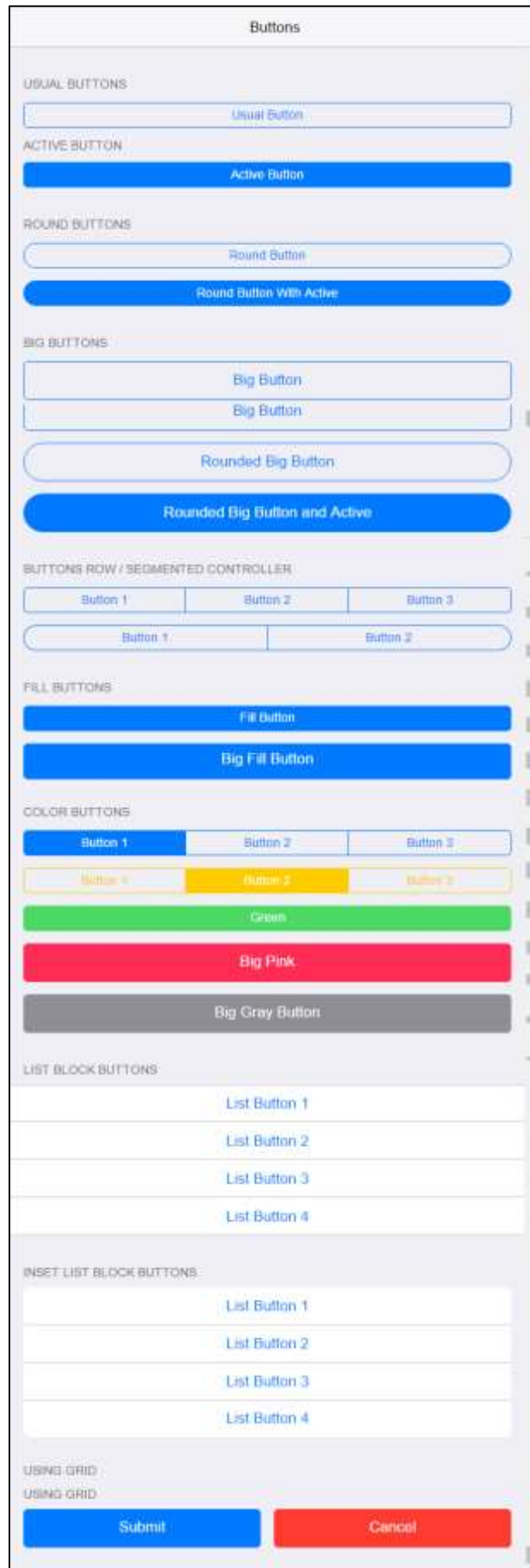
        <div class="row">
            <div class="col-50"><a href="#" class="button
button-big button-fill color-blue">Submit</a></div>
            <div class="col-50"><a href="#" class="button
button-big button-fill color-red">Cancel</a></div>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
    <script>var myApp = new Framework7();</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **buttons_iostheme.html** file in your server root folder.
- Open this HTML file as http://localhost/buttons_iostheme.html and the output is displayed as shown below.
- The example provides different types of iOS theme buttons such as round buttons, big buttons, row buttons, color buttons etc.



Material Theme Buttons

Description

The material theme provides many buttons to use in your application by using appropriate classes. The following table lists material theme buttons used in Framework7:

S. No.	Types & Description
1	Usual/Flat Buttons To use flat buttons use <i>button</i> class to your input button.
2	Buttons Row To use this, wrap buttons with element with class <i>buttons-row</i> .
3	Raised Buttons Add <i>button-raised</i> class to get raised button styled buttons.
4	Raised Fill Buttons You can add <i>button-fill</i> class to your input button.
5	Color Buttons You can use 20 default material color themes to color your buttons.
6	Color Fill Buttons You can use color fill buttons by applying <i>button-fill</i> and color classes.
7	Color Raised Fill Buttons You can use color raised fill buttons by applying <i>button-raised</i> , <i>button-fill</i> and color classes.
8	Color Ripple Buttons Different color ripple effect can be added to your button by using <i>ripple-[color]</i> class.
9	Big Buttons Button can be made as big button by adding <i>button-big</i> class.
10	List Block Buttons It is possible to use buttons with <i>list views</i> .

Example

The following example displays the different types of buttons such as color buttons, flat buttons, color fill buttons, big buttons etc. by using the material theme buttons in Framework7:

```
<!DOCTYPE html>
<html class="with-statusbar-overlay">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
```

```

<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style" content="black">
<title>iOS Theme Buttons</title>
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.min.css">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.colors.min.css">
</head>
<body>
<div class="views">
<div class="view view-main">
<div class="pages">
<div data-page="home" class="page navbar-fixed">
<div class="navbar">
<div class="navbar-inner">
<div class="center">Buttons</div>
</div>
</div>
<div class="page-content">
<div class="content-block-title">Usual/Flat Buttons</div>
<div class="content-block">
<p><a href="#" class="button">Button</a></p>
</div>
<div class="content-block-title">Buttons Row</div>
<div class="content-block">
<p class="buttons-row">
<a href="#" class="button">Button</a>
<a href="#" class="button">Button</a>
</p>
</div>
<div class="content-block-title">Raised Buttons</div>
<div class="content-block">
<p class="buttons-row">
<a href="#" class="button button-raised">Button</a>
<a href="#" class="button button-raised">Button</a>
<a href="#" class="button button-raised">Button</a>
<a href="#" class="button button-raised">Button</a>

```



```

        </p>
    </div>
<div class="content-block-title">Raised Fill Buttons</div>
<div class="content-block">
    <p class="buttons-row">
        <a href="#" class="button button-fill button-
raised">Button</a>
        <a href="#" class="button button-fill button-
raised">Button</a>
        <a href="#" class="button button-fill button-
raised">Button</a>
        <a href="#" class="button button-fill button-
raised">Button</a>
    </p>
</div>
<div class="content-block-title">Color Buttons</div>
<div class="content-block">
    <p class="buttons-row">
        <a href="#" class="button color-pink">Pink</a>
        <a href="#" class="button color-indigo">Indigo</a>
        <a href="#" class="button color-blue">Blue</a>
    </p>
    <p class="buttons-row">
        <a href="#" class="button color-cyan">Cyan</a>
        <a href="#" class="button color-green">Green</a>
        <a href="#" class="button color-purple">Purple</a>
    </p>
    <p class="buttons-row">
        <a href="#" class="button color-orange">Orange</a>
        <a href="#" class="button color-teal">Teal</a>
        <a href="#" class="button color-red">Red</a>
    </p>
</div>
<div class="content-block-title">Color Fill Buttons</div>
<div class="content-block">
    <p class="buttons-row">
        <a href="#" class="button button-fill color-
pink">Pink</a>
        <a href="#" class="button button-fill color-
indigo">Indigo</a>

```

```

        <a href="#" class="button button-fill color-
blue">Blue</a>
    </p>
    <p class="buttons-row">
        <a href="#" class="button button-fill color-
cyan">Cyan</a>
        <a href="#" class="button button-fill color-
green">Green</a>
        <a href="#" class="button button-fill color-
purple">Purple</a>
    </p>
    <p class="buttons-row">
        <a href="#" class="button button-fill color-
orange">Orange</a>
        <a href="#" class="button button-fill color-
teal">Teal</a>
        <a href="#" class="button button-fill color-
red">Red</a>
    </p>
</div>
<div class="content-block-title">Color Raised Fill
Buttons</div>
<div class="content-block">
    <p class="buttons-row">
        <a href="#" class="button button-fill button-raised
color-pink">Pink</a>
        <a href="#" class="button button-fill button-raised
color-indigo">Indigo</a>
        <a href="#" class="button button-fill button-raised
color-blue">Blue</a>
    </p>
    <p class="buttons-row">
        <a href="#" class="button button-fill button-raised
color-cyan">Cyan</a>
        <a href="#" class="button button-fill button-raised
color-green">Green</a>
        <a href="#" class="button button-fill button-raised
color-purple">Purple</a>
    </p>
    <p class="buttons-row">
        <a href="#" class="button button-fill button-raised
color-orange">Orange</a>
        <a href="#" class="button button-fill button-raised
color-teal">Teal</a>
        <a href="#" class="button button-fill button-raised
color-red">Red</a>

```

```

        </p>
    </div>
    <div class="content-block-title">Color Ripple
Buttons</div>

    <div class="content-block">
        <p class="buttons-row">
            <a href="#" class="button ripple-blue">Button</a>
            <a href="#" class="button color-cyan ripple-
green">Button with green ripple</a>
            <a href="#" class="button color-red ripple-
gray">Button with gray ripple</a>
        </p>
        <p class="buttons-row">
            <a href="#" class="button button-raised button-fill
color-yellow ripple-green">Button</a>
            <a href="#" class="button button-raised button-fill
color-purple ripple-teal">Button</a>
            <a href="#" class="button button-raised button-fill
color-pink ripple-orange">Button</a>
        </p>
    </div>
    <div class="content-block-title">Big Buttons</div>
    <div class="content-block">
        <p class="buttons-row">
            <a href="#" class="button button-big color-
pink">Button</a>
            <a href="#" class="button button-big color-
blue">Button</a>
        </p>
        <p class="buttons-row">
            <a href="#" class="button button-big button-fill
button-raised color-green">Button</a>
            <a href="#" class="button button-big button-fill
button-raised color-gray">Button</a>
        </p>
    </div>
    <div class="content-block-title">List Block Buttons</div>
    <div class="card">
        <div class="list-block">
            <ul>
                <li><a href="#" class="list-button item-
link">List Button 1</a></li>

```




Framework7 – Action Button

Description

Framework7 provides *floating action button* for a promoted action. They appear as a floating circled icon over the UI. It has the motion behaviors that contain *morphing*, *launching*, and *transferring an anchor point*.

Floating action button is only supported in Material theme.

The following table lists the *action buttons* types used in Framework7:

S. No.	Types & Description
1	Floating Action Button Layout It is very simple; you just need to place it as a child of <i>page</i> or <i>view</i> .
2	Morph To Popover If you want to open popover on clicking the floating action button then you can use <i>floating-button-to-popover</i> class.
3	Speed Dial You can invoke related actions upon clicking the floating action button by using <i>speed dial</i> .

Framework7 – Floating Action Button Layout

Description

Floating action button layout is very simple, you need to just place it as a child of *page* or *view*. To use the floating action button in your application, you need to use *floating-button* class.

Example

The following example demonstrates the use of floating action button in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Create and open Action Sheet</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.min.css">

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main">
            <div class="pages">
                <div class="page navbar-fixed">
                    <div class="navbar">
                        <div class="navbar-inner">
                            <div class="center">Floating Action Button Layout</div>
                        </div>
                    </div>
                    <a href="#" class="floating-button color-teal"><i class="icon
icon-plus"></i></a>
                    <div class="page-content">
                        <div class="content-block">
                            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
ex ea commodo consequat.</p>
                            <p> Duis aute irure dolor in reprehenderit in voluptate
velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est
laborum.</p>
                            <p>Sed ut perspiciatis unde omnis iste natus error sit
voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa
quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt
explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut
fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi
nesciunt.</p>
                            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
ex ea commodo consequat.</p>
                            <p> Duis aute irure dolor in reprehenderit in voluptate
velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est
laborum.</p>

```

```

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
ex ea commodo consequat.Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
ex ea commodo consequat.Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
ex ea commodo consequat.</p>

```

```

        <p> Duis aute irure dolor in reprehenderit in voluptate
velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est
laborum.</p>

```

```

        <p>Sed ut perspiciatis unde omnis iste natus error sit
voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa
quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt
explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut
fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi
nesciunt.Sed ut perspiciatis unde omnis iste natus error sit voluptatem
accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo
inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo
enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia
consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.Sed ut
perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque
laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et
quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem
quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni
dolores eos qui ratione voluptatem sequi nesciunt.</p>

```

```

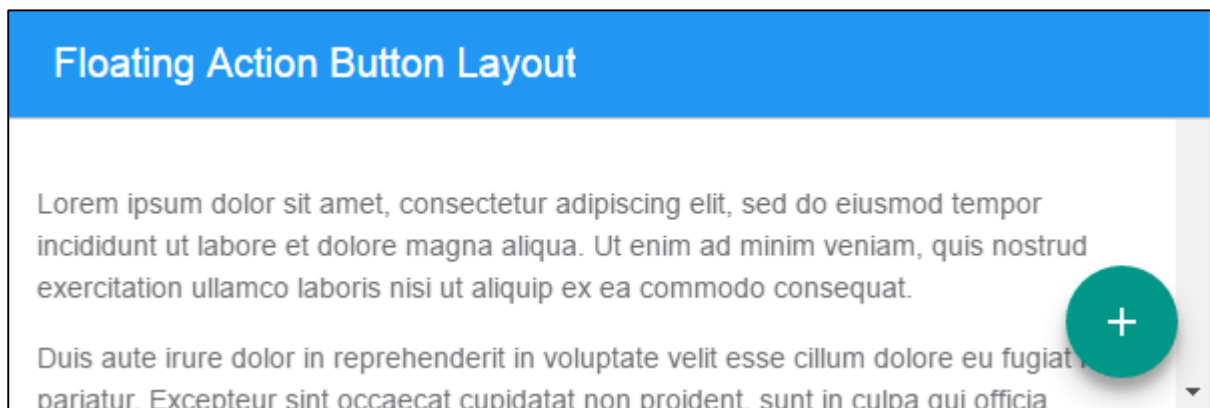
                </div>
            </div>
        </div>
    </div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7({
        material: true
    });
</script>
</body>
</html>

```


Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **action_button_layout.html** file in your server root folder.
- Open this HTML file as http://localhost/action_button_layout.html and the output is displayed as shown below.
- The example provides floating action button floats on UI in a circular shape with action attached to it and displays some related information when you click on it. Here, the code shows how you can add floating action button on UI.



Framework7 – Morph To Popover

Description

If you want to open popover on clicking the floating action button then you can use *floating-button-to-popover* class.

Example

The following example specifies the presentation of the content using morph transition (creates appearance of content with smooth animations) in the popover:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Morph To Popover</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.min.css">

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main">
            <div class="pages">
                <div class="page navbar-fixed">
                    <div class="navbar">
                        <div class="navbar-inner">
                            <div class="center">Morph To Popover</div>
                        </div>
                    </div>
                    <a href="#" class="floating-button floating-button-to-popover
open-popover color-orange"><i class="icon icon-form-email"></i></a>
                    <div class="page-content">
                        <div class="content-block">
                            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
ex ea commodo consequat.</p>
                            <p> Duis aute irure dolor in reprehenderit in voluptate
velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est
laborum.</p>
                            <p>Sed ut perspiciatis unde omnis iste natus error sit
voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa
quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt
explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut
fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi
nesciunt.</p>
                            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
ex ea commodo consequat.</p>
                            <p> Duis aute irure dolor in reprehenderit in voluptate
velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est
laborum.</p>

```

```

    <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
ex ea commodo consequat.Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
ex ea commodo consequat.Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
ex ea commodo consequat.</p>

```

```

    <p> Duis aute irure dolor in reprehenderit in voluptate
velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est
laborum.</p>

```

```

    <p>Sed ut perspiciatis unde omnis iste natus error sit
voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa
quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt
explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut
fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi
nesciunt.Sed ut perspiciatis unde omnis iste natus error sit voluptatem
accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo
inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo
enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia
consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.Sed ut
perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque
laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et
quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem
quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni
dolores eos qui ratione voluptatem sequi nesciunt.</p>

```

```

    </div>

```

```

    </div>

```

```

    </div>

```

```

    </div>

```

```

    </div>

```

```

</div>

```

```

<div style="width:250px%" class="popover demo-popover">

```

```

    <div class="popover-inner">

```

```

        <div class="list-block">

```

```

            <ul>

```

```

                <li>

```

```

                    <a href="#" class="item-content item-link">

```

```

                        <div class="item-inner">

```

```

                            <div class="item-title">Mail 1</div>

```

```

                        </div>

```

```

                    </a>

```

```

                </li>

```

```

            <li>

```

```

        <a href="#" class="item-content item-link">
            <div class="item-inner">
                <div class="item-title">Mail 2</div>
            </div>
        </a>
    </li>
    <li>
        <a href="#" class="item-content item-link">
            <div class="item-inner">
                <div class="item-title">Mail 3</div>
            </div>
        </a>
    </li>
</ul>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7({
        material: true
    });
</script>
</body>
</html>

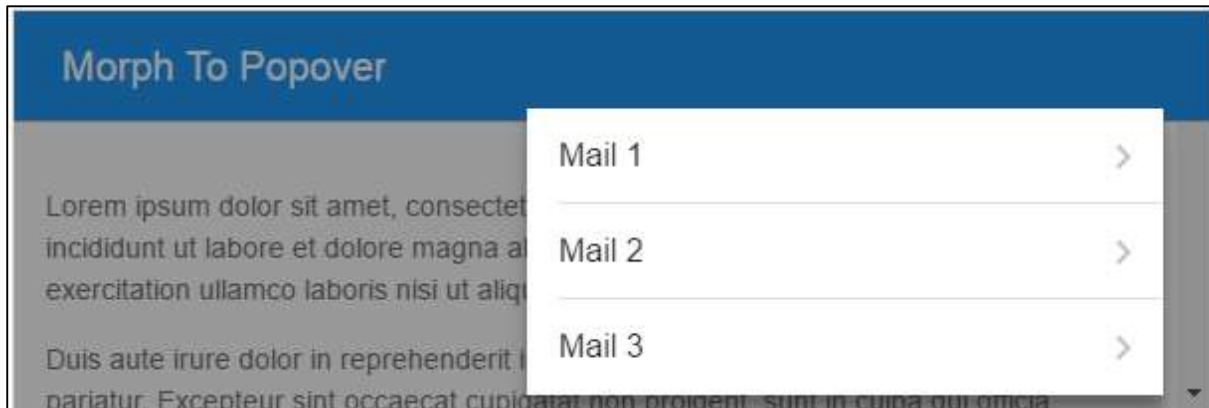
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **action_button_morph.html** file in your server root folder.

- Open this HTML file as `http://localhost/action_button_morph.html` and the output is displayed as shown below.
- When the floating action button is clicked, the morph to popover opens different types of links and displays some related information when you click on it. Here, the code shows how you can open the popover by tapping on floating action button on UI which adds more transition.



Framework7 – Floating Action Button Speed Dial

Description

You can invoke related actions upon clicking the floating action button by using *speed dial*. Minimum of 3 and maximum of 6 actions can be used in speed dial actions.

Example

The following example specifies different types of speed dial actions upon button press:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Floating Action Button Speed Dial</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.min.css">
```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.colors.min.css">
</head>
<body>
  <div class="views">
    <div class="view view-main">
      <div class="pages">
        <div class="page navbar-fixed">
          <div class="navbar">
            <div class="navbar-inner">
              <div class="center">Speed Dial</div>
            </div>
          </div>
          <div class="speed-dial">
            <a href="#" class="floating-button color-orange"><i
class="icon icon-plus"></i><i class="icon icon-close"></i></a>
            <div class="speed-dial-buttons">
              <a href="#" class="color-orange"><i class="icon icon-
form-email"></i></a><a href="#" class="color-orange"><i class="icon icon-form-
calendar"></i></a><a href="#" class="color-orange"><i class="icon icon-form-
comment"></i></a>
            </div>
          </div>
        </div>
        <div class="page-content">
          <div class="content-block">
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
ex ea commodo consequat.</p>
            <p> Duis aute irure dolor in reprehenderit in voluptate
velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est
laborum.</p>
            <p>Sed ut perspiciatis unde omnis iste natus error sit
voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa
quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt
explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut
fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi
nesciunt.</p>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
ex ea commodo consequat.</p>

```

```
<p> Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
```

```
<p> Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
```

```
<p>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.</p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div style="width:250px%" class="popover demo-popover">
```

```
<div class="popover-inner">
```

```
<div class="list-block">
```

```
<ul>
```

```
<li>
```

```
<a href="#" class="item-content item-link">
```

```
<div class="item-inner">
```

```
<div class="item-title">Mail 1</div>
```

```
</div>
```

```

        </a>
    </li>
    <li>
        <a href="#" class="item-content item-link">
            <div class="item-inner">
                <div class="item-title">Mail 2</div>
            </div>
        </a>
    </li>
    <li>
        <a href="#" class="item-content item-link">
            <div class="item-inner">
                <div class="item-title">Mail 3</div>
            </div>
        </a>
    </li>
</ul>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7({
        material: true
    });
</script>
</body>
</html>

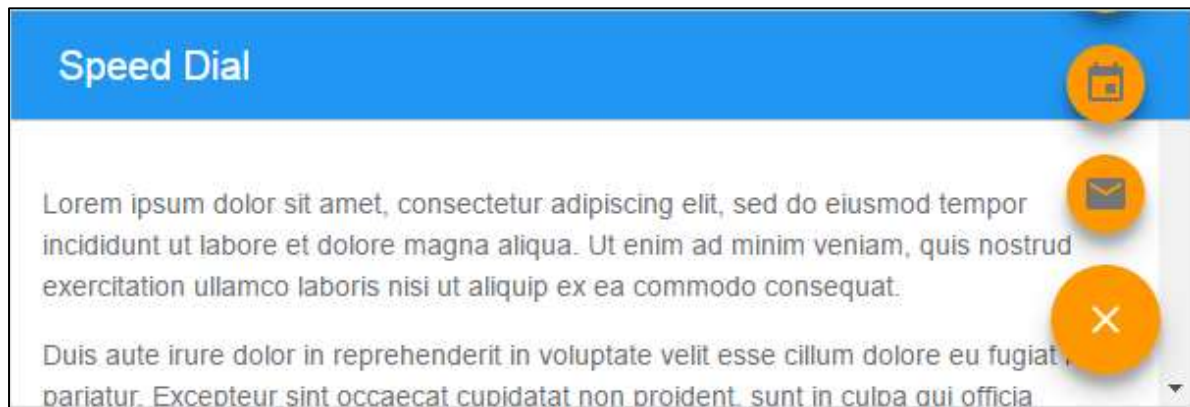
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **action_button_speed_dial.html** file in your server root folder.

- Open this HTML file as http://localhost/action_button_speed_dial.html and the output is displayed as shown below.
- When you click floating action button, the different types of speed dial actions are called as shown and displays some related information when you click on it. Here, the code shows how you can create speed dial actions on UI by using the floating action button.



19. Framework7 – Forms

Description

Forms are used for interacting with the users and collect data from the web users using text fields, checkboxes, radio buttons etc.

Framework7 provides different types of form elements to work with the applications smoothly as specified in the table below:

S. No.	Overlay Type & Description
1	Form Elements Form elements are used to make good looking form layouts.
2	Checkboxes and Radios The <i>list view</i> extension allows you to create checkbox and radio inputs in Framework7.
3	Smart Select Smart select is an easy way of changing form selects to dynamic pages by using groups of checkboxes and radio inputs.
4	Disabled Elements The elements can be disabled by applying the <i>disabled</i> class to the element or adding <i>disabled</i> attributes for form element.
5	Form Data Framework7 has very few useful methods, which makes working with forms easy.
6	Form Storage It is easy with the form storage to store and parse forms data automatically on Ajax loaded pages.
7	Ajax Form Submit Framework7 allows you to send data automatically using Ajax.

Framework7 – Form Elements

Description

Form elements are used to make good looking form layouts.

Framework7 provides some form elements as specified in the table below:

S. No.	Form Element	Structure
1	All text inputs	<pre><div class="item-input"> <input type="password"> </div></pre>
2	Select	<pre><div class="item-input"> <select>...</select> </div></pre>
3	Textarea	<pre><div class="item-input"> <textarea></textarea> </div></pre>
4	Resizable Textarea	<pre><div class="item-input"> <textarea class="resizable"></textarea> </div></pre>
5	Switch (Checkbox)	<pre><label class="label-switch"> <input type="checkbox"> <div class="checkbox"></div> </label></pre>
6	Slider (Range input)	<pre><div class="item-input"> <div class="range-slider"> <input type="range" min="10" max="50" step="0.1"> </div> </div></pre>

We can use form elements in various types as specified in the table below:

S. No.	Form Element Type & Description
1	Full Layout The full layout type is used to display the whole form element.
2	Icons and Inputs You can use icons along with the input fields in the form layout.
3	Labels and Inputs You can use labels along with the input fields in the form layout.
4	Just Inputs Framework7 allows you to use only input fields with the form elements.
5	Inset You can make the form elements inset by using the <i>inset</i> class.

Framework7 – Full Layout

Description

The full layout type is used to display the whole form element.

Example

The following example demonstrates the full layout of the form including all fields in the Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Full Layout</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
```

```

<div class="pages">
  <div data-page="home" class="page navbar-fixed">
    <div class="navbar">
      <div class="navbar-inner">
        <div class="left"> </div>
        <div class="center">Form Elements</div>
        <div class="right"> </div>
      </div>
    </div>
    <div class="page-content">
      <div class="content-block-title">Full Layout</div>
      <div class="list-block">
        <ul>
          <li>
            <div class="item-content">
              <div class="item-media"><i class="icon icon-
form-name"></i></div>
              <div class="item-inner">
                <div class="item-title label">Name</div>
                <div class="item-input">
                  <input type="text" placeholder="Enter
your name">
                </div>
              </div>
            </div>
          </li>
          <li>
            <div class="item-content">
              <div class="item-media"><i class="icon icon-
form-email"></i></div>
              <div class="item-inner">
                <div class="item-title label">E-mail</div>
                <div class="item-input">
                  <input type="email" placeholder="Enter
your e-mail">
                </div>
              </div>
            </div>
          </li>
        </ul>
      </div>
    </div>
  </div>
</div>

```

```

        <li>
            <div class="item-content">
                <div class="item-media"><i class="icon icon-
form-password"></i></div>
                <div class="item-inner">
                    <div class="item-title
label">Password</div>
                    <div class="item-input">
                        <input type="password"
placeholder="Enter your password">
                    </div>
                </div>
            </div>
        </li>
        <li>
            <div class="item-content">
                <div class="item-media"><i class="icon icon-
form-tel"></i></div>
                <div class="item-inner">
                    <div class="item-title label">Phone</div>
                    <div class="item-input">
                        <input type="tel" placeholder="Enter
your phone number">
                    </div>
                </div>
            </div>
        </li>
        <li>
            <div class="item-content">
                <div class="item-media"><i class="icon icon-
form-gender"></i></div>
                <div class="item-inner">
                    <div class="item-title label">Gender</div>
                    <div class="item-input">
                        <select>
                            <option>Male</option>
                            <option>Female</option>
                        </select>
                    </div>
                </div>
            </div>
        </li>
    </ul>

```

```

        </div>
    </li>
    <li>
        <div class="item-content">
            <div class="item-media"><i class="icon icon-
form-calendar"></i></div>
            <div class="item-inner">
                <div class="item-title label">Birth
date</div>
                <div class="item-input">
                    <input type="date" placeholder="Birth
day" value="2014-04-30">
                </div>
            </div>
        </div>
    </li>
    <li>
        <div class="item-content">
            <div class="item-media"><i class="icon icon-
form-calendar"></i></div>
            <div class="item-inner">
                <div class="item-title label">Date
time</div>
                <div class="item-input">
                    <input type="datetime-local">
                </div>
            </div>
        </div>
    </li>
    <li>
        <div class="item-content">
            <div class="item-media"><i class="icon icon-
form-toggle"></i></div>
            <div class="item-inner">
                <div class="item-title label">Switch</div>
                <div class="item-input">
                    <label class="label-switch">
                        <input type="checkbox">
                        <div class="checkbox"></div>
                    </label>
                </div>
            </div>
        </div>
    </li>

```



```
<script>
  // here initialize the app
  var myApp = new Framework7();
  // If your using custom DOM library, then save it to $$ variable
  var $$ = Dom7;
  // Add the view
  var mainView = myApp.addView('.view-main', {
    // enable the dynamic navbar for this view
    dynamicNavbar: true
  });
</script>
</body>
</html>
```













Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **form_element_full_layout.html** file in your server root folder.
- Open this HTML file as http://localhost/form_element_full_layout.html and the output is displayed as shown below.
- The user can enter all the details in the form to display the whole form element.
- The user can select any one of the two options for gender. The values can be set for birth date and date time by clicking on the arrows and can use a switch to toggle to on and off.

Form Elements

FULL LAYOUT

	Name	Enter your name
	E-mail	Enter your e-mail
	Password	Enter your password
	Password	Enter your password
	Phone	Enter your phone number
	Phone	Enter your phone number
	Gender	Male
	Birth date	04/30/2014
	Date time	mm/dd/yyyy --:-- --
	Switch	<input type="checkbox"/>
	Textarea	
	Resizable	

Framework7 – Icons and Inputs

Description

You can use icons along with the input fields in the form layout.

Example

The following example demonstrates the use of icons and inputs in the Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Icons and Inputs</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Form Elements</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">
              <div class="content-block-title">Icons and Inputs</div>
              <div class="list-block">
                <ul>
                  <li>
```

```

        <div class="item-content">
            <div class="item-media"><i class="icon icon-
form-name"></i></div>

            <div class="item-inner">
                <div class="item-input">
                    <input type="text" placeholder="Enter
your name">

                </div>
            </div>
        </li>
        <li>
            <div class="item-content">
                <div class="item-media"><i class="icon icon-
form-email"></i></div>

                <div class="item-inner">
                    <div class="item-input">
                        <input type="email" placeholder="Enter
your e-mail">

                    </div>
                </div>
            </li>
            <li>
                <div class="item-content">
                    <div class="item-media"><i class="icon icon-
form-password"></i></div>

                    <div class="item-inner">
                        <div class="item-input">
                            <input type="password"
placeholder="Enter your password">

                        </div>
                    </div>
                </li>
                <li>
                    <div class="item-content">
                        <div class="item-media"><i class="icon icon-
form-tel"></i></div>

                        <div class="item-inner">

```

```

        <div class="item-input">
            <input type="tel" placeholder="Enter
your phone number">
        </div>
    </div>
</div>
</li>
<li>
    <div class="item-content">
        <div class="item-media"><i class="icon icon-
form-gender"></i></div>
        <div class="item-inner">
            <div class="item-input">
                <select>
                    <option>Male</option>
                    <option>Female</option>
                </select>
            </div>
        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-media"><i class="icon icon-
form-calendar"></i></div>
        <div class="item-inner">
            <div class="item-input">
                <input type="date" placeholder="Birth
day" value="2014-04-30">
            </div>
        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-media"><i class="icon icon-
form-calendar"></i></div>
        <div class="item-inner">
            <div class="item-input">

```

```

        <input type="datetime-local">
      </div>
    </div>
  </div>
</li>
<li>
  <div class="item-content">
    <div class="item-media"><i class="icon icon-
form-toggle"></i></div>
    <div class="item-inner">
      <div class="item-input">
        <label class="label-switch">
          <input type="checkbox">
          <div class="checkbox"></div>
        </label>
      </div>
    </div>
  </li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
  // here initialize the app
  var myApp = new Framework7();
  // If your using custom DOM library, then save it to $$ variable
  var $$ = Dom7;
  // Add the view
  var mainView = myApp.addView('.view-main', {
  // enable the dynamic navbar for this view
  dynamicNavbar: true
  });

```

```
    </script>
  </body>
</html>
```









Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **form_element_icons_inputs.html** file in your server root folder.
- Open this HTML file as http://localhost/form_element_icons_inputs.html and the output is displayed as shown below.
- The user can enter the details in the form layout, which uses icons with the input fields.
- The user can select any one of the two options for gender. The values can be set for birth date and date time by clicking on the arrows and can use a switch to toggle to on and off.

Form Elements

ICONS AND INPUTS

	Enter your name
	Enter your e-mail
	Enter your password
	Enter your phone number
	Male
	04/30/2014
	mm/dd/yyyy --:-- --
	<input type="checkbox"/>

Framework7 – Labels and Inputs

Description

You can use labels along with the input fields in the form layout.

Example

The following example demonstrates the use of labels and inputs in the Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Labels and Input</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Form Elements</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">
              <div class="content-block-title">Labels and Input</div>
              <div class="list-block">
                <ul>
                  <li>
                    <div class="item-content">
```



```

        <div class="item-inner">
            <div class="item-title label">Name</div>
            <div class="item-input">
                <input type="text" placeholder="Enter
your name">
            </div>
        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-title label">E-mail</div>
            <div class="item-input">
                <input type="email" placeholder="Enter
your e-mail">
            </div>
        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-title
label">Password</div>
            <div class="item-input">
                <input type="password"
placeholder="Enter your password">
            </div>
        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-title label">Phone</div>
            <div class="item-input">
                <input type="tel" placeholder="Enter
your phone number">
            </div>
        </div>
    </div>

```

```

        </div>
    </li>

    <li>
        <div class="item-content">
            <div class="item-inner">
                <div class="item-title label">Gender</div>
                <div class="item-input">
                    <select>
                        <option>Male</option>
                        <option>Female</option>
                    </select>
                </div>
            </div>
        </div>
    </li>
    <li>
        <div class="item-content">
            <div class="item-inner">
                <div class="item-title label">Birth
date</div>
                <div class="item-input">
                    <input type="date" placeholder="Birth
day" value="2014-04-30">
                </div>
            </div>
        </div>
    </li>
    <li>
        <div class="item-content">
            <div class="item-inner">
                <div class="item-title label">Date
time</div>
                <div class="item-input">
                    <input type="datetime-local">
                </div>
            </div>
        </div>
    </li>
    <li>

```


- Save the above given HTML code as **form_element_labels_inputs.html** file in your server root folder.
- Open this HTML file as http://localhost/form_element_labels_inputs.html and the output is displayed as shown below.
- The user can enter the details in the form layout, which labels along with the input fields.
- The user can select any one of the two options for gender. The values can be set for birth date and date time by clicking on the arrows and can use a switch to toggle to on and off.

Form Elements

LABELS AND INPUT

Name	Enter your name
E-mail	Enter your e-mail
Password	Enter your password
Phone	Enter your phone number
Gender	Male
Birth date	04/30/2014
Date time	mm/dd/yyyy --:-- --
Switch	<input type="checkbox"/>

Framework7 – Just Inputs

Description

Framework7 allows using only input fields with the form elements.

Example

The following example demonstrates the use of just inputs in the Framework7:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Just Inputs</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Form Elements</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">
              <div class="content-block-title">Just Inputs</div>
              <div class="list-block">
                <ul>
                  <li>
                    <div class="item-content">
                      <div class="item-inner">
                        <div class="item-input">
                          <input type="text" placeholder="Enter
your name">

```

```

        </div>
    </div>

    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-input">
                <input type="email" placeholder="Enter
your e-mail">
            </div>
        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-input">
                <input type="password"
placeholder="Enter your password">
            </div>
        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-input">
                <select>
                    <option>Male</option>
                    <option>Female</option>
                </select>
            </div>
        </div>
    </div>
</li>
<li>
    <div class="item-content">

```

```

        <div class="item-inner">
            <div class="item-input">

                <input type="datetime-local">
            </div>
        </div>
    </li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // here initialize the app
    var myApp = new Framework7();

    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;

    // Add the view
    var mainView = myApp.addView('.view-main', {
        // enable the dynamic navbar for this view
        dynamicNavbar: true
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **form_element_just_inputs.html** file in your server root folder.

- Open this HTML file as http://localhost/form_element_just_inputs.html and the output is displayed as shown below.
- The user can enter the details in the form layout, which allows using only input fields with form elements.
- The user can select any one of the two options for gender. The values can be set for birth date and date time by clicking on the arrows.

Form Elements

JUST INPUTS

Enter your name

Enter your e-mail

Enter your password

Male

mm/dd/yyyy --:-- --

Framework7 – Inset

Description

You can make the form elements inset by using the *inset* class.

Example

The following example makes the form inset i.e. not full width styling of the from elements in the Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Inset</title>
```



```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">

```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">

```

```
</head>
```

```
<body>
```

```
  <div class="views">
```

```
    <div class="view view-main">
```

```
      <div class="pages">
```

```
        <div data-page="home" class="page navbar-fixed">
```

```
          <div class="navbar">
```

```
            <div class="navbar-inner">
```

```
              <div class="left"> </div>
```

```
              <div class="center">Form Elements</div>
```

```
              <div class="right"> </div>
```

```
            </div>
```

```
          </div>
```

```
        <div class="page-content">
```

```
          <div class="content-block-title">Inset</div>
```

```
          <div class="list-block inset">
```

```
            <ul>
```

```
              <li>
```

```
                <div class="item-content">
```

```
                  <div class="item-inner">
```

```
                    <div class="item-input">
```

```
                      <input type="text" placeholder="Enter
your name">
```

```
                    </div>
```

```
                  </div>
```

```
                </div>
```

```
              </li>
```

```
              <li>
```

```
                <div class="item-content">
```

```
                  <div class="item-inner">
```

```
                    <div class="item-input">
```

```
                      <input type="email" placeholder="Enter
your e-mail">
```

```
                    </div>
```

```

        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-input">
                <input type="password"
placeholder="Enter your password">
            </div>
        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-input">
                <select>
                    <option>Male</option>
                    <option>Female</option>
                </select>
            </div>
        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-input">
                <input type="datetime-local">
            </div>
        </div>
    </div>
</li>
</ul>
</div>
</div>
</div>

```

```

        </div>
    </div>

</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // here initialize the app
    var myApp = new Framework7();

    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;

    // Add the view
    var mainView = myApp.addView('.view-main', {
    // enable the dynamic navbar for this view
    dynamicNavbar: true
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **form_element_inset.html** file in your server root folder.
- Open this HTML file as http://localhost/form_element_inset.html and the output is displayed as shown below.
- The user can make the form elements inset by using the inset class as shown.
- The user can select any one of the two options for gender. The values can be set for birth date and date time by clicking on the arrows.

Form Elements

INSET

Enter your name

Enter your e-mail

Framework7 – Checkboxes & Radios

Description

The *list view* extention allows you to create checkbox and radio inputs in Framework7.

The following table shows the checkboxes and radio inputs used in Framework7:

S. No.	Types & Description
1	<p>Checkboxes group</p> <p>It allows the user to select a list of specified options by checking them.</p>
2	<p>Radios group iOS</p> <p>It allows the user to select only one option from the list of options.</p>
3	<p>Radios group Material</p> <p>Radios group is also supported by the Material theme and you have to add additional icons for radios.</p>
4	<p>With media list view</p> <p>It is possible to use both checkboxes and radios groups with media list view.</p>

Framework7 – Checkboxes Group

Description

Checkbox group allows the user to select list of specified options by checking them. The *item-content* must be a label element with *label-checkbox* class and the checkbox icon should be in *item-media*.

Example

The following example demonstrates the use of checkbox group in Framework7:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Checkboxes group</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Checkboxes Group</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">
              <div class="content-block-title">Checkboxes group</div>
              <div class="list-block">
                <ul>
                  <li>
                    <label class="label-checkbox item-content">
                      <input type="checkbox" name="ks-checkbox"
value="Pants" checked>
                      <div class="item-media"><i class="icon icon-
form-checkbox"></i></div>
                      <div class="item-inner">
                        <div class="item-title">Pants</div>
                      </div>
                    </label>

```

```

        </li>
        <li>
            <label class="label-checkbox item-content">
                <input type="checkbox" name="ks-checkbox"
value="Shirts">
                <div class="item-media"><i class="icon icon-
form-checkbox"></i></div>
                <div class="item-inner">
                    <div class="item-title">Shirts</div>
                </div>
            </label>
        </li>
        <li>
            <label class="label-checkbox item-content">
                <input type="checkbox" name="ks-checkbox"
value="Sarees">
                <div class="item-media"><i class="icon icon-
form-checkbox"></i></div>
                <div class="item-inner">
                    <div class="item-title">Sarees</div>
                </div>
            </label>
        </li>
        <li>
            <label class="label-checkbox item-content">
                <input type="checkbox" name="ks-checkbox"
value="Jeans">
                <div class="item-media"><i class="icon icon-
form-checkbox"></i></div>
                <div class="item-inner">
                    <div class="item-title">Jeans</div>
                </div>
            </label>
        </li>
    </ul>
</div>
</div>
</div>
</div>

```

```

    </div>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
    <script>
        var myApp = new Framework7();
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **forms_checkboxes.html** file in your server root folder.
- Open this HTML file as http://localhost/forms_checkboxes.html and the output is displayed as shown below.
- The form layout shows the checkbox group, which allows you to select the options by checking them where you can select one or more options.

The screenshot displays a mobile application interface for a 'Checkboxes Group'. The title 'CHECKBOXES GROUP' is centered at the top. Below the title, there are two options: 'Pants' with a checked blue checkbox and 'Shirts' with an unchecked white checkbox. At the bottom of the interface, there are three navigation buttons: 'Previous Page' with a left arrow, 'Print' with a printer icon, and 'Next Page' with a right arrow.

Framework7 – Radios Group iOS

Description

It allows the user to select only one option from the list of options. The *item-content* must be a label element with *label-radio* class and radio input should be the first child of *item-content*.

Example

The following example demonstrates the use of radios group in Framework7:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Checkboxes group</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Radios Group</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">
              <div class="content-block-title">Radios group</div>
              <div class="list-block">
                <ul>
                  <li>
                    <label class="label-radio item-content">
                      <input type="radio" name="ks-radio"
value="Pants" checked>
                    <div class="item-inner">
                      <div class="item-title">Pants</div>
                    </div>
                  </li>
                </ul>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>

```



```

        </label>
    </li>
    <li>
        <label class="label-radio item-content">
            <input type="radio" name="ks-radio"
value="Shirts">
            <div class="item-inner">
                <div class="item-title">Shirts</div>
            </div>
        </label>
    </li>
    <li>
        <label class="label-radio item-content">
            <input type="radio" name="ks-radio"
value="Jeans">
            <div class="item-inner">
                <div class="item-title">Jeans</div>
            </div>
        </label>
    </li>
    <li>
        <label class="label-radio item-content">
            <input type="radio" name="ks-radio"
value="Sarees">
            <div class="item-inner">
                <div class="item-title">Sarees</div>
            </div>
        </label>
    </li>
</ul>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>

```

```

        var myApp = new Framework7();
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **forms_radio.html** file in your server root folder.
- Open this HTML file as http://localhost/forms_radio.html and the output is displayed as shown below.
- The form layout shows the list of options, which allows you to check only one option from the list.

Framework7 – Radios Group Material

Description

The Radios group is also supported by the Material theme and you have to add additional icons for radios.

Example

The following example demonstrates the use of radios group material in Framework7:

```

<!DOCTYPE html>
<html>

```

```

<head>
  <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="apple-mobile-web-app-status-bar-style" content="black">
  <title>Radios group Material</title>
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.min.css">
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.colors.min.css">
</head>
<body>
  <div class="views">
    <div class="view view-main">
      <div class="pages">
        <div data-page="home" class="page navbar-fixed">
          <div class="navbar">
            <div class="navbar-inner">
              <div class="center">Radios Group</div>
            </div>
          </div>
          <div class="page-content">
            <div class="content-block-title">Radios group</div>
            <div class="list-block">
              <ul>
                <li>
                  <label class="label-radio item-content">
                    <input type="radio" name="ks-radio"
value="Pants" checked>
                    <div class="item-media"><i class="icon icon-
form-radio"></i></div>
                    <div class="item-inner">
                      <div class="item-title">Pants</div>
                    </div>
                  </label>
                </li>
                <li>
                  <label class="label-radio item-content">

```

```

value="Shirts">
form-radio"></i></div>
    <input type="radio" name="ks-radio"
    <div class="item-media"><i class="icon icon-
    <div class="item-inner">
        <div class="item-title">Shirts</div>
    </div>
    </label>
</li>
<li>
    <label class="label-radio item-content">
        <input type="radio" name="ks-radio"
value="Jeans">
form-radio"></i></div>
    <div class="item-media"><i class="icon icon-
    <div class="item-inner">
        <div class="item-title">Jeans</div>
    </div>
    </label>
</li>
<li>
    <label class="label-radio item-content">
        <input type="radio" name="ks-radio"
value="Sarees">
form-radio"></i></div>
    <div class="item-media"><i class="icon icon-
    <div class="item-inner">
        <div class="item-title">Sarees</div>
    </div>
    </label>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>

```

```

    <script>
        var myApp = new Framework7();
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **forms_radio_material.html** file in your server root folder.
- Open this HTML file as http://localhost/forms_radio_material.html and the output is displayed as shown below.
- The form layout shows the radio group with the list of options, which allows you to check only one option from the list.

Framework7 – With Media List View

Description

It is possible to use both checkboxes and radios groups with media list view.

Example

The following example demonstrates the use of checkbox and radios with media list:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">

```

```

    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Checkboxes group</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main">
            <div class="pages">
                <div data-page="home" class="page navbar-fixed">
                    <div class="navbar">
                        <div class="navbar-inner">
                            <div class="left"> </div>
                            <div class="center">Checkboxes and Radios</div>
                            <div class="right"> </div>
                        </div>
                    </div>
                    <div class="page-content">
                        <div class="content-block-title">Select Your Message</div>
                        <div class="list-block media-list">
                            <ul>
                                <li>
                                    <label class="label-checkbox item-content">
                                        <input type="checkbox" name="my-checkbox"
value="1">
                                        <div class="item-media"><i class="icon icon-
form-checkbox"></i></div>
                                    <div class="item-inner">
                                        <div class="item-title-row">
                                            <div class="item-title">Amayon</div>
                                            <div class="item-after">7:10</div>
                                        </div>
                                        <div class="item-subtitle">Your order has
been shipped</div>
                                        <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt.</div>

```

```

        </div>
    </label>
</li>
<li>
    <label class="label-checkbox item-content">
        <input type="checkbox" name="my-checkbox"
value="2">
        <div class="item-media"><i class="icon icon-
form-checkbox"></i></div>
        <div class="item-inner">
            <div class="item-title-row">
                <div class="item-
title">VodafoneZone</div>
                <div class="item-after">10:15</div>
            </div>
            <div class="item-subtitle">Bill
Payments</div>
            <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt.</div>
        </div>
    </label>
</li>
<li>
    <label class="label-checkbox item-content">
        <input type="checkbox" name="my-checkbox"
value="3">
        <div class="item-media"><i class="icon icon-
form-checkbox"></i></div>
        <div class="item-inner">
            <div class="item-title-row">
                <div class="item-title">Popular in your
network</div>
                <div class="item-after">19:14</div>
            </div>
            <div class="item-subtitle">New messages
from John Doe</div>
            <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt.</div>
        </div>
    </label>
</li>

```

```

        </li>
        <li>
            <label class="label-checkbox item-content">
                <input type="checkbox" name="my-checkbox"
value="4">
                <div class="item-media"><i class="icon icon-
form-checkbox"></i></div>
                <div class="item-inner">
                    <div class="item-title-row">
                        <div class="item-title">Adam
WillSmith</div>
                        <div class="item-after">22:44</div>
                    </div>
                    <div class="item-subtitle">Car Insurance
renewal</div>
                    <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt.</div>
                </div>
            </label>
        </li>
    </ul>
</div>
<div class="content-block-title">Which is your favourite
Magazine?</div>
<div class="list-block media-list">
    <ul>
        <li>
            <label class="label-radio item-content">
                <input type="radio" name="my-radio" value="1"
checked>
                <div class="item-media"></div>
                <div class="item-inner">
                    <div class="item-title-row">
                        <div class="item-title">Tarangi</div>
                        <div class="item-after">$10</div>
                    </div>
                    <div class="item-subtitle">Monthly</div>
                    <div class="item-text">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt. Cras dolor metus, ultrices

```



```
condimentum sodales sit amet, pharetra sodales eros. Phasellus vel felis
tellus. Mauris rutrum ligula nec dapibus feugiat. In vel dui laoreet, commodo
augue id, pulvinar lacus.</div>
```

```
</div>
```

```
</label>
```

```
</li>
```

```
<li>
```

```
<label class="label-radio item-content">
```

```
<input type="radio" name="my-radio" value="2">
```

```
<div class="item-media"></div>
```

```
<div class="item-inner">
```

```
<div class="item-title-row">
```

```
<div class="item-title">Business
```

```
Today</div>
```

```
<div class="item-after">$20</div>
```

```
</div>
```

```
<div class="item-subtitle">Monthly</div>
```

```
<div class="item-text">Lorem ipsum dolor
```

```
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt. Cras dolor metus, ultrices
condimentum sodales sit amet, pharetra sodales eros. Phasellus vel felis
tellus. Mauris rutrum ligula nec dapibus feugiat. In vel dui laoreet, commodo
augue id, pulvinar lacus.</div>
```

```
</div>
```

```
</label>
```

```
</li>
```

```
<li>
```

```
<label class="label-radio item-content">
```

```
<input type="radio" name="my-radio" value="3">
```

```
<div class="item-media"></div>
```

```
<div class="item-inner">
```

```
<div class="item-title-row">
```

```
<div class="item-title">Frontier</div>
```

```
<div class="item-after">$15</div>
```

```
</div>
```

```
<div class="item-subtitle">Weakly</div>
```

```
<div class="item-text">Lorem ipsum dolor
```

```
sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis
condimentum, ut dignissim lacus tincidunt. Cras dolor metus, ultrices
condimentum sodales sit amet, pharetra sodales eros. Phasellus vel felis
tellus. Mauris rutrum ligula nec dapibus feugiat. In vel dui laoreet, commodo
augue id, pulvinar lacus.</div>
```

```
        </div>
        </label>
    </li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **forms_radio_with_media.html** file in your server root folder.
- Open this HTML file as http://localhost/forms_radio_with_media.html and the output is displayed as shown below.
- The form layout shows the use of both checkboxes and the radio groups with media list view which allows you to select only one option from radio groups and check one or more options from checkboxes.

Checkboxes and Radios

SELECT YOUR MESSAGE (CHECKBOX EXAMPLE)

Amayon 7:10

Your order has been shipped

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sagittis tellus ut turpis condimentum, ut dignissim lacus tincidunt.

Framework7 – Smart Select

Description

Smart select is an easy way of changing form selects to dynamic pages by using groups of checkboxes and radio inputs.

We can use smart select in various types as specified in the table below:

S. No.	Type & Description
1	<p>Smart Select Layout</p> <p>Smart select layout defines a list view inside the <i>select</i> element using <i>smart-select</i> class.</p>
2	<p>Smart Select with Search Bar</p> <p>Smart select searches the elements by using <i>search bar</i> and enables it by setting the <i>data-searchbar</i> class to true.</p>
3	<p>Custom Page Title and Back Link Text</p> <p>You can set the custom page title and back link for smart select using <i>data-page-title</i> and <i>data-back-text</i> attributes.</p>
4	<p>Open in Popup</p> <p>Smart select can be open as popup by using the <i>data-open-in</i> attribute to <i>popup</i>.</p>
5	<p>Open in Picker</p> <p>Smart select can be displayed as picker modal by setting the <i>data-open-in</i> attribute to <i>picker</i>.</p>
6	<p>Custom Icons, Colors and Images</p> <p>You can define custom icon, color or image on the smart select by using the <i>data-option-icon</i>, <i>data-option-color</i> and <i>data-option-image</i> attributes respectively.</p>
7	<p>Multiple Select and Optgroup</p> <p>Smart select allows to use multiple select and group options by using the <i>multiple</i> and <i>optgroup</i> attributes.</p>

8	<p>Multiple Select and Maxlength Smart select allows to select limited number items by using the <i>maxlength</i> attribute.</p>
---	---

You can have some more types of smart select, which can be used in different scenario as listed in the table below:

S. No.	Type & Description	Attribute
1	<p>Close Smart Select On User Select You can close the smart select when the user selects any option.</p>	data-back-on-select="true"
2	<p>Smart Select With Virtual List Virtual list can used with smart select if you have lots of options.</p>	data-virtual-list="true"
3	<p>Smart Select Color Themes You can specify the color themes for form and navbar on smart select page.</p>	data-form-theme="color" data-navbar-theme="color"
4	<p>Set Smart Select Value By Option Text Value for smart select can be set by using the option value.</p>	smart-select-value
5	<p>Open Smart Select Using JavaScript You can open the smart select using JavaScript method.</p>	<p>myApp.smartSelectOpen(smartSelect)</p> <ul style="list-style-type: none"> • smartSelect: It is HTMLElement or string parameter used to open specified smart select.
6	<p>Adding Options Dynamically You can add options dynamically to smart select, even if they are already opened.</p>	<p>myApp.smartSelectAddOption(select, optionHTML, index)</p> <ul style="list-style-type: none"> • smartSelect: It is HTMLElement or string parameter used to open specified smart select. • optionHTML: It is required string element which can be used to add HTML option.

		<ul style="list-style-type: none"> • index: It specifies the index number for the new option.
--	--	---

Framework7 – Smart Select Layout

Description

Smart select layout defines a list view inside the *select* element using *smart-select* class.

Example

The following example demonstrates the use of smart select layout in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Smart Select Layout</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="left"> </div>
            <div class="center sliding">Smart Select Layout</div>
            <div class="right"> </div>
          </div>
        </div>
        <div class="pages navbar-through">
          <div data-page="home" class="page">
            <div class="page-content">
              <div class="list-block">
```

```

        <ul>
            <li>
                <a href="#" class="item-link smart-select">
                    <select name="countries">
                        <option value="India"
selected>India</option>
                        <option value="Newz=Zealnad">New
Zealand</option>
                        <option value="Srilanka">Srilanka</option>
                        <option value="WestIndies">West
Indies</option>
                        <option value="China">China</option>
                    </select>
                    <div class="item-content">
                        <div class="item-inner">
                            <div class="item-title">Countries</div>
                        </div>
                    </div>
                </a>
            </li>
            <li>
                <a href="#" class="item-link smart-select">
                    <select name="sports">
                        <option value="Cricket">Cricket</option>
                        <option value="Hockey">Hockey</option>
                        <option value="Football">Football</option>
                        <option value="Volleyball"
selected>Volleyball</option>
                        <option value="Rugby">Rugby</option>
                    </select>
                    <div class="item-content">
                        <div class="item-inner">
                            <div class="item-title">Sports</div>
                        </div>
                    </div>
                </a>
            </li>
            <li>
                <a href="#" class="item-link smart-select">

```

```

                                <select name="books">
                                <option value="EyesDragon" selected>Eyes of
the Dragon</option>
                                <option value="Winter">Winter
Wedding</option>
                                <option value="DarkWood" selected>In a
Dark, Dark Wood</option>
                                <option value="FarAway" selected>Far and
Away</option>
                                <option value="WasteLands">The Waste
Lands</option>
                                </select>
                                <div class="item-content">
                                <div class="item-inner">
                                <div class="item-title">Books</div>
                                </div>
                                </div>
                                </a>
                                </li>
                                </ul>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
                                <script>
                                // here initialize the app
                                var myApp = new Framework7({
                                animateNavBackIcon:true
                                });
                                // If your using custom DOM library, then save it to $$ variable
                                var $$ = Dom7;
                                // Add the view
                                var mainView = myApp.addView('.view-main', {
                                // enable the dynamic navbar for this view
                                dynamicNavbar: true
                                });

```

```

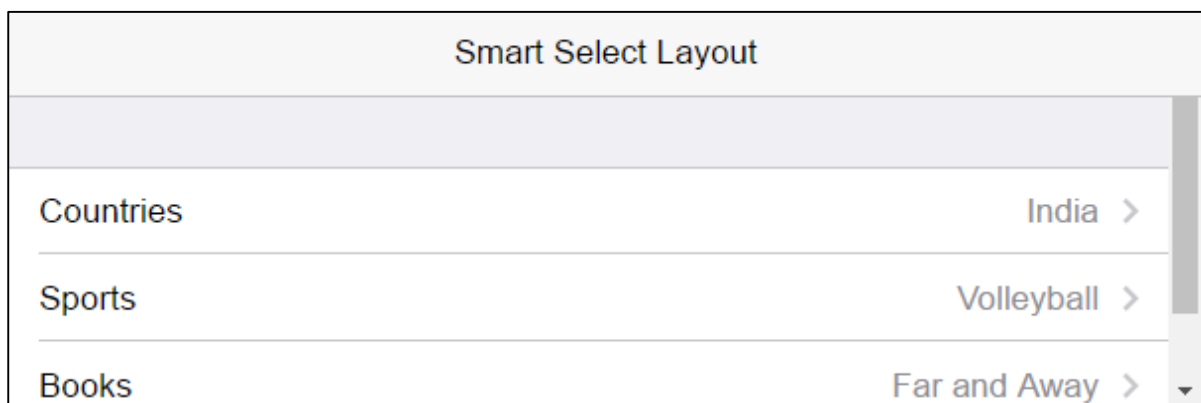
    </script>
  </body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **smart_select_layout.html** file in your server root folder.
- Open this HTML file as `http://localhost/smart_select_layout.html` and the output is displayed as shown below.
- The output shows the use smart-select class to display the smart select layout to view a list inside the select element.
- When you click on any particular element from the list, you can view another list inside that selected element where you can select only one option.



Framework7 – Smart Select Search Bar

Description

Smart select searches the elements by using *search bar* and enable it by setting the *data-searchbar* class to true.

Example

The following example allows to select the item from the list using select search bar in Framework7:

```

<!DOCTYPE html>
<html>
  <head>

```



```

    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Smart Select Search Bar</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main">
            <div class="navbar">
                <div class="navbar-inner">
                    <div class="left"> </div>
                    <div class="center sliding">Smart Select with Search Bar</div>
                    <div class="right"> </div>
                </div>
            </div>
            <div class="pages navbar-through">
                <div data-page="home" class="page">
                    <div class="page-content">
                        <div class="list-block">
                            <ul>
                                <li>
                                    <a href="#" data-searchbar="true" data-searchbar-
placeholder="Search Countries" class="item-link smart-select">
                                        <select name="countries">
                                            <option value="India"
selected>India</option>
                                            <option value="Newz=Zealnad">New
Zealand</option>
                                            <option value="Srilanka">Srilanka</option>
                                            <option value="WestIndies">West
Indies</option>
                                            <option value="China">China</option>
                                        </select>
                                        <div class="item-content">

```

```

        <div class="item-inner">
            <div class="item-title">Countries</div>
        </div>
    </div>
</a>
</li>
<li>
    <a href="#" data-searchbar="true" data-searchbar-
placeholder="Search Sports" class="item-link smart-select">
        <select name="sports">
            <option value="Cricket">Cricket</option>
            <option value="Hockey">Hockey</option>
            <option value="Football">Football</option>
            <option value="Volleyball"
selected>Volleyball</option>
            <option value="Rugby">Rugby</option>
        </select>
        <div class="item-content">
            <div class="item-inner">
                <div class="item-title">Sports</div>
            </div>
        </div>
    </a>
</li>
<li>
    <a href="#" data-searchbar="true" data-searchbar-
placeholder="Search Books" class="item-link smart-select">
        <select name="books">
            <option value="EyesDragon" selected>Eyes of
the Dragon</option>
            <option value="Winter">Winter
Wedding</option>
            <option value="DarkWood" selected>In a
Dark, Dark Wood</option>
            <option value="FarAway" selected>Far and
Away</option>
            <option value="WasteLands">The Waste
Lands</option>
        </select>
        <div class="item-content">
            <div class="item-inner">

```

```

        <div class="item-title">Books</div>
    </div>
</div>
</a>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // here initialize the app
    var myApp = new Framework7({
        animateNavBackIcon:true
    });

    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;

    // Add the view
    var mainView = myApp.addView('.view-main', {
        // enable the dynamic navbar for this view
        dynamicNavbar: true
    });
</script>
</body>
</html>

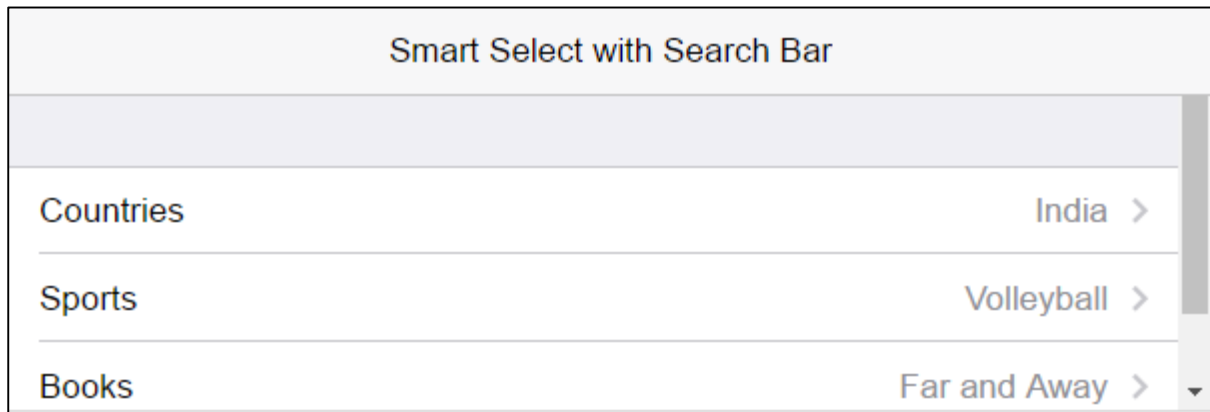
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **smart_select_with_search_bar.html** file in your server root folder.
- Open this HTML file as http://localhost/smart_select_with_search_bar.html and the output is displayed as shown below.

- The output shows the use smart-select class to display the smart select layout to view a list inside the select element along with a search bar.
- When you click on any particular element from the list, you can view another list inside that selected element along with the search bar where you can select only one option and if you want to search any particular element from the list, you can enter element name in the search bar.



Framework7 – Custom Page Title and Back Link Text

Description

You can set custom page title and back link for smart select using *data-page-title* and *data-back-text* attributes.

Example

The following example demonstrates the use of custom page title and back link for smart select in the Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Custom Page Title and Back Link Text</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
```

```

<body>
  <div class="views">
    <div class="view view-main">
      <div class="navbar">
        <div class="navbar-inner">
          <div class="left"> </div>
          <div class="center sliding">Custom Page Title and Back Link
Text</div>
          <div class="right"> </div>
        </div>
      </div>
      <div class="pages navbar-through">
        <div data-page="home" class="page">
          <div class="page-content">
            <div class="list-block">
              <ul>
                <li>
                  <a href="#" data-page-title="Countries List"
data-back-text="Click here to go back" class="item-link smart-select">
                    <select name="countries">
                      <option value="India"
selected>India</option>
                      <option value="Newz=Zealnad">New
Zealand</option>
                      <option value="Srilanka">Srilanka</option>
                      <option value="WestIndies">West
Indies</option>
                      <option value="China">China</option>
                    </select>
                    <div class="item-content">
                      <div class="item-inner">
                        <div class="item-title">Countries</div>
                      </div>
                    </div>
                  </a>
                </li>
                <li>
                  <a href="#" data-page-title="Sports List" data-
back-text="Click here to go back" class="item-link smart-select">
                    <select name="sports">

```

```

        <option value="Cricket">Cricket</option>
        <option value="Hockey">Hockey</option>
        <option value="Football">Football</option>
        <option value="Volleyball"
selected>Volleyball</option>
        <option value="Rugby">Rugby</option>
    </select>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-title">Sports</div>
        </div>
    </div>
</a>
</li>
<li>
    <a href="#" data-page-title="Books List" data-
back-text="Click here to go back" class="item-link smart-select">
        <select name="books">
            <option value="EyesDragon" selected>Eyes of
the Dragon</option>
            <option value="Winter">Winter
Wedding</option>
            <option value="DarkWood" selected>In a
Dark, Dark Wood</option>
            <option value="FarAway" selected>Far and
Away</option>
            <option value="WasteLands">The Waste
Lands</option>
        </select>
        <div class="item-content">
            <div class="item-inner">
                <div class="item-title">Books</div>
            </div>
        </div>
    </a>
</li>
</ul>
</div>
</div>
</div>
</div>

```

```

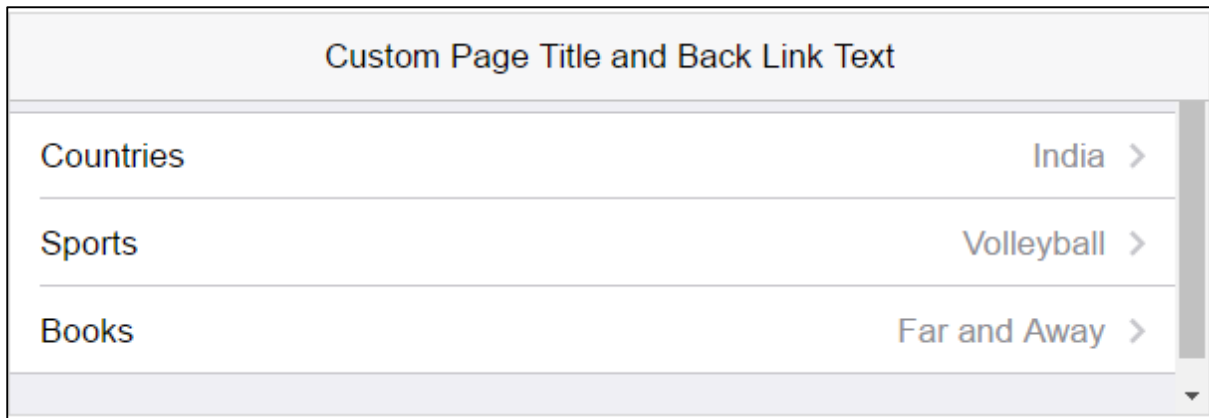
        </div>
    </div>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
    <script>
        // here initialize the app
        var myApp = new Framework7({
            animateNavBackIcon:true
        });
        // If your using custom DOM library, then save it to $$ variable
        var $$ = Dom7;
        // Add the view
        var mainView = myApp.addView('.view-main', {
            // enable the dynamic navbar for this view
            dynamicNavbar: true
        });
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **smart_select_page_title_back_link.html** file in your server root folder.
- Open this HTML file as http://localhost/smart_select_page_title_back_link.html and the output is displayed as shown below.
- The output shows the use of data-page-title and data-back-text attributes to set the custom page title and back link for smart select where the page title gets displays when you click on any one option from the list.
- You can go back to the previous list by clicking on the back link.



Framework7 – Open in Popup

Description

Smart select can be open as popup by using the *data-open-in* attribute to the *popup*.

Example

The following example demonstrates the use of smart select open in popup in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Open in Popup</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="left"> </div>
            <div class="center sliding">Smart Select Open in Popup</div>
            <div class="right"> </div>
```



```

        </div>
    </div>
    <div class="pages navbar-through">
        <div data-page="home" class="page">
            <div class="page-content">
                <div class="list-block">
                    <ul>
                        <li>
                            <a href="#" data-open-in="popup" class="item-link
smart-select">
                                <select name="countries">
                                    <option value="India"
selected>India</option>
                                    <option value="Newz=Zealnad">New
Zealand</option>
                                    <option value="Srilanka">Srilanka</option>
                                    <option value="WestIndies">West
Indies</option>
                                    <option value="China">China</option>
                                </select>
                                <div class="item-content">
                                    <div class="item-inner">
                                        <div class="item-title">Countries</div>
                                    </div>
                                </div>
                            </a>
                        </li>
                        <li>
                            <a href="#" data-open-in="popup" data-popup-
close-text="Close the List" class="item-link smart-select">
                                <select name="sports">
                                    <option value="Cricket">Cricket</option>
                                    <option value="Hockey">Hockey</option>
                                    <option value="Football">Football</option>
                                    <option value="Volleyball"
selected>Volleyball</option>
                                    <option value="Rugby">Rugby</option>
                                </select>
                                <div class="item-content">
                                    <div class="item-inner">

```

```

        <div class="item-title">Sports</div>
    </div>
</div>
</a>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // here initialize the app
    var myApp = new Framework7({
    animateNavBackIcon:true
    });
    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;
    // Add the view
    var mainView = myApp.addView('.view-main', {
    // enable the dynamic navbar for this view
    dynamicNavbar: true
    });
</script>
</body>
</html>

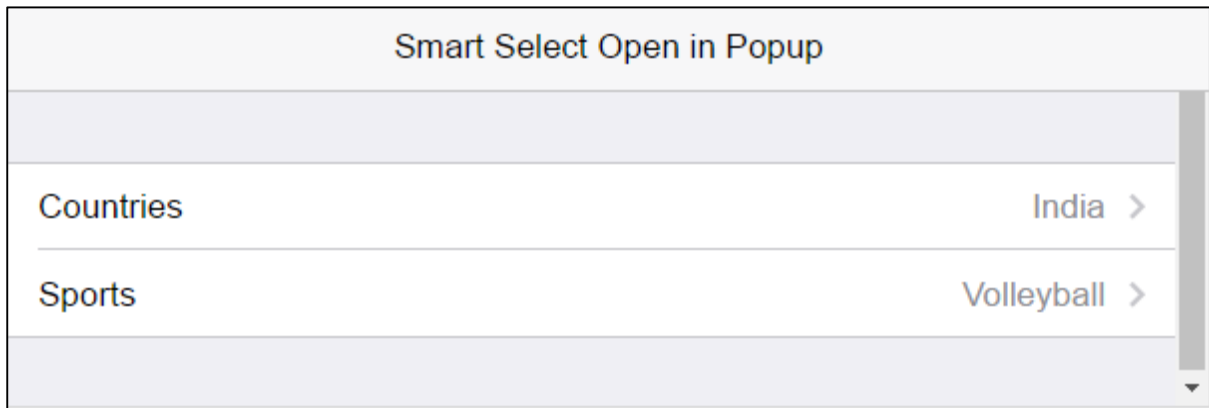
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **smart_select_popup.html** file in your server root folder.
- Open this HTML file as http://localhost/smart_select_popup.html and the output is displayed as shown below.

- The output shows the use of data-open-in attribute to open the smart select as a popup.
- When you click on any one option from the list, it links to a popup containing another list along with the page title where you can select only one option from the list.



Framework7 – Open in Picker

Description

Smart select can be display as picker modal by setting the *data-open-in* attribute to *picker*.

Example

The following example demonstrates the use of smart select open in picker in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Open in Picker</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
```

```

<div class="navbar">
  <div class="navbar-inner">
    <div class="left"> </div>
    <div class="center sliding">Smart Select Open in Picker</div>
    <div class="right"> </div>
  </div>
</div>
<div class="pages navbar-through">
  <div data-page="home" class="page">
    <div class="page-content">
      <div class="list-block">
        <ul>
          <li>
            <a href="#" data-open-in="picker" class="item-
link smart-select">
              <select name="countries">
                <option value="India"
selected>India</option>
                <option value="Newz=Zealnad">New
Zealand</option>
                <option value="Srilanka">Srilanka</option>
                <option value="WestIndies">West
Indies</option>
                <option value="China">China</option>
              </select>
              <div class="item-content">
                <div class="item-inner">
                  <div class="item-title">Countries</div>
                </div>
              </div>
            </a>
          </li>
          <li>
            <a href="#" data-open-in="picker" data-picker-
close-text="Close the List" class="item-link smart-select">
              <select name="sports">
                <option value="Cricket">Cricket</option>
                <option value="Hockey">Hockey</option>
                <option value="Football">Football</option>

```

```

selected>Volleyball</option>
        <option value="Volleyball"
        <option value="Rugby">Rugby</option>
    </select>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-title">Sports</div>
        </div>
    </div>
</a>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // here initialize the app
    var myApp = new Framework7({
    animateNavBackIcon:true
    });

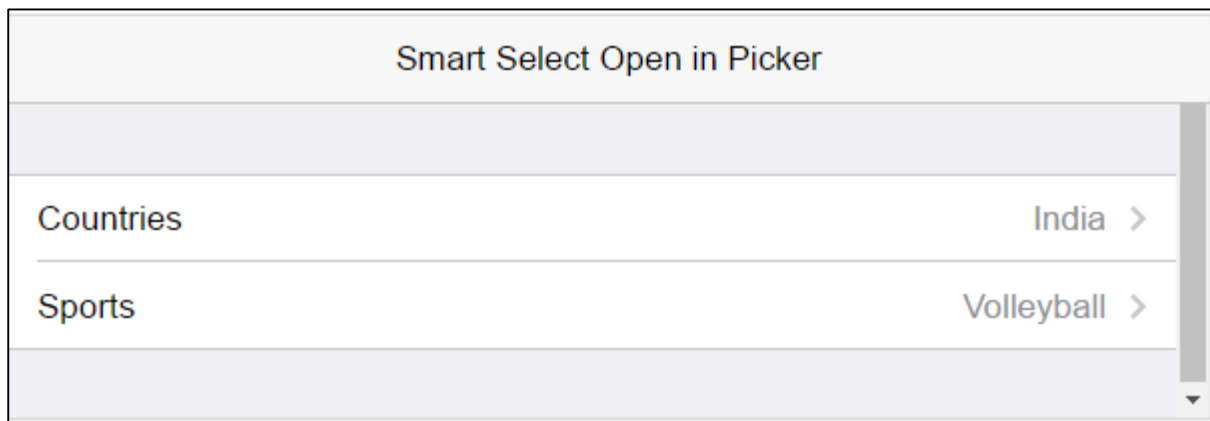
    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;
    // Add the view
    var mainView = myApp.addView('.view-main', {
    // enable the dynamic navbar for this view
    dynamicNavbar: true
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **smart_select_picker.html** file in your server root folder.
- Open this HTML file as http://localhost/smart_select_picker.html and the output is displayed as shown below.
- The output shows the smart select that displays as a picker modal by setting attribute `data-open-in` to `picker`.
- When you click on any one option, it displays as a picker modal, which contains a list with options where you can select any one option.



Framework7 – Custom icons, colors and images

Description

You can define custom icon, color or image on the smart select by using the `data-option-icon`, `data-option-color` and `data-option-image` attributes respectively.

Example

The following example demonstrates the use of smart select in custom icons, colors and images in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
```

```

    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Custom icons, colors and images</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main">
            <div class="navbar">
                <div class="navbar-inner">
                    <div class="left"> </div>
                    <div class="center sliding">Smart Select Custom Icons, Colors
and Images</div>
                    <div class="right"> </div>
                </div>
            </div>
            <div class="pages navbar-through">
                <div data-page="home" class="page">
                    <div class="page-content">
                        <div class="list-block">
                            <ul>
                                <li>
                                    <a href="#" class="item-link smart-select">
                                        <select name="countries">
                                            <option value="India" selected data-option-
image="images/ind.gif">India</option>
                                            <option value="Newz=Zealnad" data-option-
image="images/arg.gif">Argentina</option>
                                            <option value="Srilanka" data-option-
color="red" data-option-image="images/bgm.gif">Belgium</option>
                                        </select>
                                        <div class="item-content">
                                            <div class="item-inner">
                                                <div class="item-title">Countries</div>
                                            </div>
                                        </div>
                                    </a>
                                </li>
                            </ul>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </body>

```

```

        </li>
    </ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // here initialize the app
    var myApp = new Framework7({
    animateNavBackIcon:true
    });
    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;
    // Add the view
    var mainView = myApp.addView('.view-main', {
    // enable the dynamic navbar for this view
    dynamicNavbar: true
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **smart_select_icons_colors_images.html** file in your server root folder.
- Open this HTML file as http://localhost/smart_select_icons_colors_images.html and the output is displayed as shown below.
- The output shows the use of data-option-icon, data-option-color and data-option-image to define the custom icon, color or image for smart select.
- When you click on the option named as countries, a list of different countries is displayed with icon, color or image which allows you to select any one option.

Smart Select Custom Icons, Colors and Images	
Countries	India >

Framework7 – Multiple Select and Optgroup

Description

Smart select allows to use multiple select and group options by using the *multiple* and *optgroup* attributes.

Example

The following example demonstrates the use of multiple select and optgroup in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Multiple Select and Optgroup</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="left"> </div>
```

```

        <div class="center sliding">Multiple Select and
Optgroup</div>
        <div class="right"> </div>
    </div>
</div>
<div class="pages navbar-through">
    <div data-page="home" class="page">
        <div class="page-content">
            <div class="list-block">
                <ul>
                    <li>
                        <a href="#" class="item-link smart-select">
                            <select name="car" multiple>
                                <optgroup label="India">
                                    <option value="delhi"
selected>Delhi</option>
                                    <option value="mumbai">Mumbai</option>
                                    <option
value="bangalore">Bangalore</option>
                                </optgroup>
                                <optgroup label="United Kingdom">
                                    <option value="london">London</option>
                                    <option value="belfast"
selected>Belfast</option>
                                    <option value="wells">Wells</option>
                                </optgroup>
                                <optgroup label="Australia">
                                    <option value="sydney">Sydney</option>
                                    <option value="perth"
selected>Perth</option>
                                    <option
value="melbourne">Melbourne</option>
                                </optgroup>
                            </select>
                            <div class="item-content">
                                <div class="item-inner">
                                    <div class="item-title">Cities</div>
                                </div>
                            </div>
                        </a>
                    </li>
                </ul>
            </div>
        </div>
    </div>
</div>

```

```

        </li>
    </ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // here initialize the app
    var myApp = new Framework7({
    animateNavBackIcon:true
    });
    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;
    // Add the view
    var mainView = myApp.addView('.view-main', {
    // enable the dynamic navbar for this view
    dynamicNavbar: true
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **smart_select_multiple_optgroup.html** file in your server root folder.
- Open this HTML file as http://localhost/smart_select_multiple_optgroup.html and the output is displayed as shown below.
- When you click on the option 'Cities', you can see different options for each city, which allows you to select multiple options for each city and you can go back by clicking on the back link.

Multiple Select and Optgroup	
Cities	Delhi, Belfast, Perth >

Framework7 – Multiple Select and Maxlength

Description

Smart select allows to select limited number items by using the *maxlength* attribute.

Example

The following example demonstrates the selection of items with limit number of possible selected items in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Multiple Select and Maxlength</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="left"> </div>
            <div class="center sliding">Multiple Select and
Maxlength</div>
```

```

        <div class="right"> </div>
    </div>
</div>
<div class="pages navbar-through">
    <div data-page="home" class="page">
        <div class="page-content">
            <div class="list-block">
                <ul>
                    <li>
                        <a href="#" class="item-link smart-select">
                            <select name="car" multiple maxlength="4">
                                <optgroup label="India">
                                    <option value="delhi"
selected>Delhi</option>
                                    <option value="mumbai">Mumbai</option>
                                    <option
value="bangalore">Bangalore</option>
                                </optgroup>
                                <optgroup label="United Kingdom">
                                    <option value="london">London</option>
                                    <option value="belfast">Belfast</option>
                                    <option value="wells">Wells</option>
                                </optgroup>
                                <optgroup label="Australia">
                                    <option value="sydney">Sydney</option>
                                    <option value="perth">Perth</option>
                                    <option
value="melbourne">Melbourne</option>
                                </optgroup>
                            </select>
                            <div class="item-content">
                                <div class="item-inner">
                                    <div class="item-title">Cities</div>
                                </div>
                            </div>
                        </a>
                    </li>
                </ul>
            </div>
        </div>
    </div>

```

```

        </div>
    </div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // here initialize the app
    var myApp = new Framework7({
    animateNavBackIcon:true
    });
    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;
    // Add the view
    var mainView = myApp.addView('.view-main', {
    // enable the dynamic navbar for this view
    dynamicNavbar: true
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **smart_select_multiple_maxlength.html** file in your server root folder.
- Open this HTML file as http://localhost/smart_select_multiple_maxlength.html and the output is displayed as shown below.
- Here the maximum length is set to 4. When you click on the Cities option, you can see different options for each city where you can select only upto four options.

Multiple Select and Maxlength	
Cities	Delhi >

Framework7 – Forms Disabled Elements

Description

There are two ways, which can be used for disabling the form elements:

- **disabled class:** It can be applied to any elements of HTML.
- **disabled attribute:** It can only be applied for form elements like *input*, *textarea* and *select*.

Example

The following example demonstrates the use of disabled elements in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Checkboxes group</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
```

```

<div class="navbar">
  <div class="navbar-inner">
    <div class="left"> </div>
    <div class="center">Disabled elements</div>
    <div class="right"> </div>
  </div>
</div>
<div class="page-content">
  <div class="list-block">
    <ul>
      <li class="item-content">
        <div class="item-inner">
          <div class="item-title">Item 1</div>
        </div>
      </li>
      <!-- This element will be disabled -->
      <li class="item-content disabled">
        <div class="item-inner">
          <div class="item-title">Item 2 with
<i>disabled</i> class applied</div>
        </div>
      </li>
      <li class="item-content">
        <div class="item-inner">
          <div class="item-title">Item 3</div>
        </div>
      </li>
      <li class="item-content">
        <div class="item-inner">
          <div class="item-title label">This is input
text</div>
          <div class="item-input">
            <!-- Disabled input -->
            <input type="text" name="name"
placeholder="Enter your name">
          </div>
        </div>
      </li>
      <!-- Element with disabled input -->

```



```
        <li class="item-content">
            <div class="item-inner">
                <div class="item-title label">This is disabled
input text</div>

                <div class="item-input">
                    <!-- Disabled input -->
                    <input type="text" name="name"
placeholder="Enter your name" disabled>
                </div>
            </div>
        </li>
    </ul>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **forms_disabled_element.html** file in your server root folder.
- Open this HTML file as http://localhost/forms_disabled_element.html and the output is displayed as shown below.
- The output shows the form where the class and input element are disabled.
- As you can see that the user is not allowed to enter the text in the input field because the input element is disabled.

Disabled elements
Item 1
Item 2 with <i>disabled</i> class applied
Item 3

Framework7 – Form Data

Description

Framework7 has very few useful methods, which makes working with forms easy. The following table shows form data methods used in Framework7:

S. No.	Methods & Description
1	Form to JSON You can use <i>formToJSON()</i> method to convert all form fields values to JSON.
2	Form from JSON You can use <i>formFromJSON</i> method to fill up form according to JSON data.

Framework7 – Form To JSON

Description

Framework7 allows you to convert all form field values to JSON by using the following method:

- **myApp.formToJSON(form)** - This method accepts *form* as parameter, which is a *HTMLElement* or *string* of form that must be converted to JSON. This parameter is required and this method returns JSON data.

Each input must have the *name* attribute, otherwise its value will not appear in JSON. Multiple selects and checkboxes will appear as arrays in JSON.

Example

The following example demonstrates the use of form to JSON in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
```

```

    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Checkboxes group</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main">
            <div class="pages">
                <div data-page="home" class="page navbar-fixed">
                    <div class="navbar">
                        <div class="navbar-inner">
                            <div class="left"> </div>
                            <div class="center">Form To JSON</div>
                            <div class="right"> </div>
                        </div>
                    </div>
                    <div class="page-content">
                        <form id="my-form" class="list-block">
                            <ul>
                                <li>
                                    <div class="item-content">
                                        <div class="item-inner">
                                            <div class="item-title label">Name</div>
                                            <div class="item-input">
                                                <input type="text" name="name"
placeholder="Enter your name">
                                            </div>
                                        </div>
                                    </div>
                                </li>
                                <li>
                                    <div class="item-content">

```

```

        <div class="item-inner">
            <div class="item-title label">E-mail</div>
            <div class="item-input">
                <input type="email" name="email"
placeholder="Enter your e-mail">
            </div>
        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-title label">Gender</div>
            <div class="item-input">
                <select name="gender">
                    <option value="male"
selected>Male</option>
                    <option
value="female">Female</option>
                </select>
            </div>
        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-title label">Switch</div>
            <div class="item-input">
                <label class="label-switch">
                    <input type="checkbox" name="switch"
value="yes">
                    <div class="checkbox"></div>
                </label>
            </div>
        </div>
    </div>
</li>
</ul>

```

```

        </form>
        <div class="content-block"><a href="#" class="button
button-fill button-round color-red form-to-json">Get Form Data</a></div>
    </div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var $$ = Dom7;
    $$('.form-to-json').on('click', function(){
        var formData = myApp.formToJSON('#my-form');
        alert(JSON.stringify(formData));
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **forms_to_json.html** file in your server root folder.
- Open this HTML file as http://localhost/forms_to_json.html and the output is displayed as shown below.
- When you enter your details in the form and click on 'Get form data' button, all your field values will be converted to JSON format and is displayed to you.

Form To JSON	
Name	Enter your name
E-mail	Enter your e-mail
Gender	Male
Switch	<input type="checkbox"/>

Framework7 – Form From JSON

Description

Framework7 allows you fill up form according to JSON by using the following method:

- **myApp.formFromJSON(form, formData)** - It accepts two parameters as listed below:
 - **form** - It is a *HTMLElement* or *string* of form that must be converted to JSON. This is a required parameter.
 - **formData** - This is the *object* with form data in JSON format. This is a required parameter.

Example

The following example demonstrates the use filling form from JSON data in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Checkboxes group</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Form To JSON</div>
                <div class="right"> </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

<div class="page-content">
  <form id="my-form" class="list-block">
    <ul>
      <li>
        <div class="item-content">
          <div class="item-inner">
            <div class="item-title label">Name</div>
            <div class="item-input">
              <input type="text" name="name"
placeholder="Enter your name">
            </div>
          </div>
        </div>
      </li>
      <li>
        <div class="item-content">
          <div class="item-inner">
            <div class="item-title label">E-mail</div>
            <div class="item-input">
              <input type="email" name="email"
placeholder="Enter your e-mail">
            </div>
          </div>
        </div>
      </li>
      <li>
        <div class="item-content">
          <div class="item-inner">
            <div class="item-title label">Gender</div>
            <div class="item-input">
              <select name="gender">
                <option value="male"
selected>Male</option>
                <option
value="female">Female</option>
              </select>
            </div>
          </div>
        </div>
      </li>
    </ul>
  </form>

```

```

        </li>
        <li>
            <div class="item-content">
                <div class="item-inner">
                    <div class="item-title label">Switch</div>
                    <div class="item-input">
                        <label class="label-switch">
                            <input type="checkbox" name="switch"
value="yes">
                            <div class="checkbox"></div>
                        </label>
                    </div>
                </div>
            </div>
        </li>
    </ul>
</form>
    <div class="content-block"><a href="#" class="button
button-fill button-round color-red form-from-json">Fill Up Form</a></div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var $$ = Dom7;
    $$('.form-from-json').on('click', function(){
        var formData = {
            'name': 'William Smith',
            'email': 'william_smith@tutorialspoint.com',
            'gender': 'male',
            'switch': ['yes'],
        }
        myApp.formFromJSON('#my-form', formData);
    });

```



```

    </script>
  </body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **forms_from_json.html** file in your server root folder.
- Open this HTML file as http://localhost/forms_from_json.html and the output is displayed as shown below.
- Here, the method allows to fill up the form easily according to the JSON data when the switch is ON and 'Fill up form' button is clicked.

Form To JSON

Name	Enter your name
E-mail	Enter your e-mail
Gender	Male
Switch	<input type="checkbox"/>

Framework7 – Form Storage

Description

Form storage stores and parses forms data automatically on Ajax loaded pages. When the page gets loaded again, the Framework7 will parse this data and automatically fill the form.

S. No.	Form storage types & Description
1	Enable form storage To enable form storage, you need to add <i>store-data</i> class and <i>id</i> attribute to form.
2	Form storage JavaScript API Form storage can be achieved by using JavaScript API.

Form storage JavaScript Events

The following table shows the form storage JavaScript events used in Framework7:

S. No.	Events & Description	Target
1	formFromJSON This will be triggered after <i>formFromJSON</i> method invoked on form to fill the form fields.	Form Element
2	formToJSON This will be triggered after <i>formToJSON</i> method invoked on form to convert the form fields to JSON.	Form Element

Framework7 – Enable Form Storage

Description

To enable form storage, you need to add *store-data* class and *id* attribute to form. Ensure that all the inputs have *name* attribute, otherwise they will be ignored.

Example

The following example displays the form storage, which stores the data entered in the form and parse the form data automatically when you reload the page:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Enable form storage</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
```

```

        <div class="center">Enable form storage</div>
        <div class="right"> </div>
    </div>
</div>
<div class="page-content">
    <div class="content-block">Just fill the form and reload
the page, your form fields will keep your data.</div>
    <form id="my-form" class="list-block store-data">
        <ul>
            <li>
                <div class="item-content">
                    <div class="item-inner">
                        <div class="item-title label">Name</div>
                        <div class="item-input">
                            <input type="text" name="name"
placeholder="Enter your name">
                        </div>
                    </div>
                </div>
            </li>
            <li>
                <div class="item-content">
                    <div class="item-inner">
                        <div class="item-title label">E-mail</div>
                        <div class="item-input">
                            <input type="email" name="email"
placeholder="Enter your e-mail">
                        </div>
                    </div>
                </div>
            </li>
            <li>
                <div class="item-content">
                    <div class="item-inner">
                        <div class="item-title label">Gender</div>
                        <div class="item-input">
                            <select name="gender">
                                <option value="male"
selected>Male</option>

```

```

        <option
value="female">Female</option>
        </select>
    </div>
</div>
</div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-title label">Switch</div>
            <div class="item-input">
                <label class="label-switch">
                    <input type="checkbox" name="switch"
value="yes">
                </label>
                <div class="checkbox"></div>
            </div>
        </div>
    </li>
</ul>
</form>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **forms_enable_storage.html** file in your server root folder.
- Open this HTML file as `http://localhost/forms_enable_storage.html` and the output is displayed as shown below.
- When you enter the data in the form and reload the page, the form storage is enabled and you can see that the data which you entered in the fields is stored.

Enable form storage

Just fill the form and reload the page, your form fields will keep your data.

Name	<input type="text" value="Enter your name"/>
E-mail	<input type="text" value="Enter your e-mail"/>

Framework7 – Form Storage JavaScript API

Description

Form storage can be achieved by using JavaScript API. Here the Framework7 invokes *formToJSON* on any input change and *formFromJSON* on *pageInit* event. All forms data are stored in local storage with keys. The App's methods to manage these local storage keys with form data are listed below:

S. No.	Classes & Description	Parameters
1	myApp.formGetData(formId) When you open an animation, this event will get fired.	<ul style="list-style-type: none"> • formId: It is the id attribute of required form of type string.
2	myApp.formDeleteData(formId) When the opening of an animation is completed, this event will get fired.	<ul style="list-style-type: none"> • formId: It is the id attribute of required form of type string.
3	myApp.formStoreData(formId, formJSON) When you close an animation, this event will get fired.	<ul style="list-style-type: none"> • formId: It is the id attribute of required form of type string. • formJSON: It is an object of JSON data to store.

Example

The following example demonstrates the use of form storage JavaScript API which stores data in local storage in Framework7:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Form storage JavaScript API</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Form Storage API</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">
              <form id="myform" class="list-block store-data">
                <ul>
                  <li>
                    <div class="item-content">
                      <div class="item-inner">
                        <div class="item-title label">Name</div>
                        <div class="item-input">
                          <input type="text" name="name"
placeholder="Enter your name">
                        </div>
                      </div>
                    </div>
                  </li>
                </ul>
              </form>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>

```

```

        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-title label">E-mail</div>
            <div class="item-input">
                <input type="email" name="email"
placeholder="Enter your e-mail">
            </div>
        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-title label">Gender</div>
            <div class="item-input">
                <select name="gender">
                    <option value="male"
selected>Male</option>
                    <option
value="female">Female</option>
                </select>
            </div>
        </div>
    </div>
</li>
<li>
    <div class="item-content">
        <div class="item-inner">
            <div class="item-title label">Switch</div>
            <div class="item-input">
                <label class="label-switch">
                    <input type="checkbox" name="switch"
value="yes">
                    <div class="checkbox"></div>
                </label>
            </div>
        </div>
    </div>
</li>

```

```

        </div>
    </div>
</div>
</li>
</ul>
</form>
<div class="content-block">
    <p><a href="#" class="button button-fill button-round
color-blue get-storage-data">Get Data</a></p>
    <p><a href="#" class="button button-fill button-round
color-red delete-storage-data">Delete Data</a></p>
    <p><a href="#" class="button button-fill button-round
color-green save-storage-data">Save Data</a></p>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var $$ = Dom7;
    $$('.get-storage-data').on('click', function() {
        var storedData = myApp.formGetData('myform');
        if(storedData) {
            alert(JSON.stringify(storedData));
        }
        else {
            alert('Yet there is no stored data for this form. Please try to
change any field')
        }
    });

    $$('.delete-storage-data').on('click', function() {
        var storedData = myApp.formDeleteData('myform');
        alert('Form data is deleted')
    });

```



```
    $$('.save-storage-data').on('click', function() {
        var storedData = myApp.formStoreData('myform', {
            'name': 'William Smith',
            'email': 'williamsmith@tutorialspoint.com',
            'gender': 'male',
            'switch': ['yes'],
        });
        alert('Form data is replaced, refresh the browser to reflect the
changes')
    });
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **forms_storage_javascript.html** file in your server root folder.
- Open this HTML file as http://localhost/forms_storage_javascript.html and the output is displayed as shown below.
- When you enter the details in the form and click on the 'Get data' button, all your field values will be converted to JSON format and is displayed to you.
- When the 'Save data' button is clicked, the forms data is saved in local storage.
- When the 'Delete data' button is clicked, all the forms data gets deleted.

Form Storage API

Name	Enter your name
E-mail	Enter your e-mail
Gender	Male
Switch	<input type="checkbox"/>

Get Data

Delete Data

Save Data

Ajax Form Submit

Description

Framework7 allows you to automatically send data using Ajax by using the following two ways:

- When the user submits the form or *submit* event is triggered on form programmatically.
- When the user modifies any form field or *change* event triggered on form programmatically.

Send form data on submit

To enable Ajax form and send the form data on clicking Submit, automatically, you need to add the *ajax-submit* class to form. After the user submits the form, Ajax will automatically send the form data using the following rules:

- Form data will be sent to the file or the URL mentioned in *action* attribute of the form.
- Request method will be similar as mentioned in the *method* attribute of the form.
- Content type will be the same as mentioned in the *enctype* attribute of the form. If not mentioned by default, it is *application/x-www-form-urlencoded*.

Send form data on input change

We can submit the form data when the user makes any changes in the form fields by using *ajax-submit-onchange* class as shown below:

```
<form action="send.html" method="GET" class="ajax-submit-onchange">
    ...
</form>
```

When the user modifies any form field, form data will be sent automatically using Ajax with same rules as discussed above.

Ajax submit event

To get actual XHR response from the file/URL where you sent data, you can use special events as shown below:

S. No.	Event & Description	target
1	submitted This event will be triggered after successful Ajax request.	Form Element <form class="ajax-submit">
2	beforeSubmit This event will be triggered right before the Ajax request.	Form Element <form class="ajax-submit">
3	submitError This event will be triggered on Ajax request error.	Form Element <form class="ajax-submit">

20. Framework7 – Tabs

Description

Tabs are sets of logically grouped content that allow us to quickly flip between them to and saves the space like accordions.

Tabs Layout

The following code defines the layout for the tabs:

```
<!-- Tabs wrapper, should have "tabs" class. It is a required element -->
<div class="tabs">
  <!-- The tab, should have "tab" class and unique id attribute -->
  <!-- The tab is active by default - "active" class -->
  <div class="tab active" id="tab1">
    ... The content for Tab 1 goes here ...
  </div>
  <!-- The second tab, should have "tab" class and unique id attribute -->
  <div class="tab" id="tab2">
    ... The content for Tab 2 goes here ...
  </div>
</div>
```

Where:

- **<div class="tabs">**: It is a required wrapper for all tabs. If we miss this, tabs will not work at all.
- **<div class="tab">**: It is a single tab, which should have **unique id** attribute.
- **<div class="tab active">**: It is a single active tab, which uses additional *active* class to make tab visible (active).

Switching Between Tabs

You can use some controller in tabs layout, so that the user can switch between them.

For this, you need to create links (<a> tags) with *tab-link* class and *href* attribute equal to the *id* attribute of target tab:

```
<!-- Here the link is used to activate 1st tab, has the same href attribute
(#tab1) as the id attribute of 1st tab (tab1) -->
<a href="#tab1" class="tab-link active">Tab 1</a>
```

```

<!-- Here the link is used to activates 2st tab, has the same href attribute
(#tab2) as the id attribute of 2st tab (tab2) -->
<a href="#tab2" class="tab-link">Tab 2</a>
<a href="#tab3" class="tab-link">Tab 3</a>

```

Switch Multiple Tabs

If you are using single tab link to switch between multiple tabs, then you can use classes instead of using *ID's* and *data-tab* attribute.

```

<!-- For Top Tabs -->
<div class="tabs tabs-top">
  <div class="tab tab1 active">...</div>
  <div class="tab tab2">...</div>
  <div class="tab tab3">...</div>
</div>
<!-- For Bottom Tabs -->
<div class="tabs tabs-bottom">
  <div class="tab tab1 active">...</div>
  <div class="tab tab2">...</div>
  <div class="tab tab3">...</div>
</div>
<!-- For Tabs links -->
<div class="tab-links">
  <!-- Links are switch top and bottom tabs to .tab1 -->
  <a href="#" class="tab-link" data-tab=".tab1">Tab 1</a>
  <!-- Links are switch top and bottom tabs to .tab2 -->
  <a href="#" class="tab-link" data-tab=".tab2">Tab 2</a>
  <!-- Links are switch top and bottom tabs to .tab3 -->
  <a href="#" class="tab-link" data-tab=".tab3">Tab 3</a>
</div>

```

The *data-tab* attribute of tab-link contains CSS selector of target tab/tabs

We can use different ways of tabs, these are specified in the following table:

S. No.	Tabs Types & Description
1	Inline Tabs Inline tabs are sets of grouped in inline that allow you to quickly flip between them.
2	Switch Tabs From Navbar We can place tabs in navigation bar that allow you to flip between them.

3	Switch Views From Tab Bar Single tab can be used to switch between views with its own navigation and layout.
4	Animated Tabs You can use simple transition (animation) to switch tabs.
5	Swipeable Tabs You can create swipeable tabs with simple transition by using the <i>tabs-swipeable-wrap</i> class for the tabs.
6	Tabs JavaScript Events JavaScript events can be used when you are working with JavaScript code for the tabs.
7	Show Tab Using JavaScript You can switch or show the tab using JavaScript methods.

Framework7 – Inline Tabs

Description

Inline tabs are sets of grouped in inline that allow us to quickly flip between them.

Example:

The following example demonstrates the switching between different content using inline tabs in Framework7:

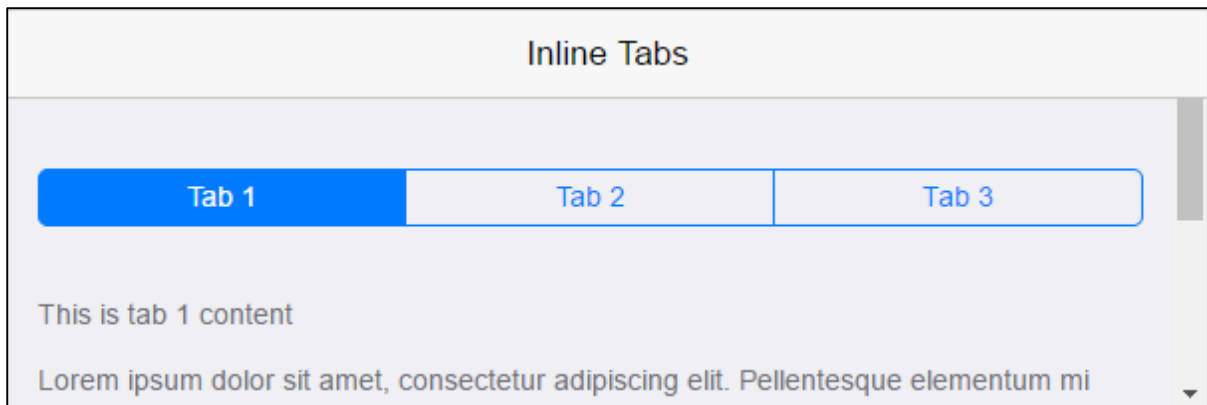
```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Inline Tabs</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
```

```

<div class="pages">
  <div data-page="home" class="page navbar-fixed">
    <div class="navbar">
      <div class="navbar-inner">
        <div class="left"> </div>
        <div class="center">Inline Tabs</div>
        <div class="right"> </div>
      </div>
    </div>
    <div class="page-content">
      <div class="content-block">
        <div class="buttons-row"><a href="#tab1" class="tab-link active button">Tab 1</a><a href="#tab2" class="tab-link button">Tab 2</a><a href="#tab3" class="tab-link button">Tab 3</a></div>
      </div>
      <div class="tabs">
        <div id="tab1" class="tab active">
          <div class="content-block">
            <p>This is tab 1 content</p>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque elementum mi quis felis scelerisque faucibus. Aliquam ut commodo justo. Mauris vitae pharetra arcu. Sed tincidunt dui et nibh auctor pretium. Nam accumsan fermentum sem. Suspendisse potenti. Nulla sed orci malesuada, pellentesque elit vitae, cursus lorem. Praesent et vehicula sapien, ut rhoncus quam. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In vitae mi nec lorem aliquet venenatis quis nec nibh. Aenean sit amet leo ligula. Fusce in massa et nisl dictum ultricies et vitae dui. Sed sagittis quis diam sed lobortis. Donec in massa pharetra, tristique purus vitae, consequat mauris. Aliquam tellus ante, pharetra in mattis ut, dictum quis erat.</p>
          </div>
        </div>
        <div id="tab2" class="tab">
          <div class="content-block">
            <p>This is tab 2 content</p>
            <p>Ut ac lobortis lacus, non pellentesque arcu. Quisque sodales sapien malesuada, condimentum nunc at, viverra lacus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vivamus eu pulvinar turpis, id tristique quam. Aenean venenatis molestie diam, sit amet condimentum nisl pretium id. Donec diam tortor, mollis in vehicula id, vehicula consectetur nulla. Quisque posuere rutrum mauris, eu rutrum turpis blandit at. Proin volutpat tortor sit amet metus porttitor accumsan. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Ut dapibus posuere dictum.</p>
          </div>
        </div>
      </div>
    </div>
  </div>

```


- Save the above given HTML code as **inline_tabs.html** file in your server root folder.
- Open this HTML file as http://localhost/inline_tabs.html and the output is displayed as shown below.
- The code displays the three inline tabs which allow us to quickly flip between them.



Framework7 – Switch Tabs From Navbar

Description

You can place the tabs in navigation bar that allows you to flip between them and looks like navigation bar.

Example

The following example demonstrates the use of switch tabs from navbar in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Switch tabs from Navbar </title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
```

```

<div class="view view-main">
  <div class="pages">
    <div data-page="home" class="page navbar-fixed">
      <div class="navbar">
        <div class="navbar-inner">
          <div class="left"> </div>
          <div class="center">
            <div class="buttons-row"><a href="#tab1" class="tab-link active button">Tab 1</a><a href="#tab2" class="tab-link button">Tab 2</a><a href="#tab3" class="tab-link button">Tab 3</a></div>
          </div>
          <div class="right"> </div>
        </div>
      </div>
      <div class="page-content tabs">
        <div id="tab1" class="tab active">
          <div class="content-block">
            <p>This is tab 1 content</p>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque elementum mi quis felis scelerisque faucibus. Aliquam ut commodo justo. Mauris vitae pharetra arcu. Sed tincidunt dui et nibh auctor pretium. Nam accumsan fermentum sem. Suspendisse potenti. Nulla sed orci malesuada, pellentesque elit vitae, cursus lorem. Praesent et vehicula sapien, ut rhoncus quam. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In vitae mi nec lorem aliquet venenatis quis nec nibh. Aenean sit amet leo ligula. Fusce in massa et nisl dictum ultricies et vitae dui. Sed sagittis quis diam sed lobortis. Donec in massa pharetra, tristique purus vitae, consequat mauris. Aliquam tellus ante, pharetra in mattis ut, dictum quis erat.</p>
          </div>
        </div>
        <div id="tab2" class="tab">
          <div class="content-block">
            <p>This is tab 2 content</p>
            <p>Ut ac lobortis lacus, non pellentesque arcu. Quisque sodales sapien malesuada, condimentum nunc at, viverra lacus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vivamus eu pulvinar turpis, id tristique quam. Aenean venenatis molestie diam, sit amet condimentum nisl pretium id. Donec diam tortor, mollis in vehicula id, vehicula consectetur nulla. Quisque posuere rutrum mauris, eu rutrum turpis blandit at. Proin volutpat tortor sit amet metus porttitor accumsan. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Ut dapibus posuere dictum.</p>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        <div id="tab3" class="tab">
            <div class="content-block">
                <p>This is tab 3 content</p>
                <p>Nulla gravida libero eget lobortis iaculis. In
sed elit eu nibh adipiscing faucibus. Sed ac accumsan lacus. In ut diam quis
turpis fringilla volutpat. In ultrices dignissim consequat. Cras pretium tortor
et lorem condimentum posuere. Nulla facilisi. Suspendisse pretium egestas lacus
ac laoreet. Mauris rhoncus quis ipsum quis tristique. Vivamus ultricies urna
quis nunc egestas, in euismod turpis fringilla. Nam tellus massa, vehicula eu
sapien non, dapibus tempor lorem. Fusce placerat orci arcu, eu dignissim enim
porttitor vel. Nullam porttitor vel dolor sed feugiat. Suspendisse potenti.
Maecenas ac mattis odio. Sed vel ultricies lacus, sed posuere libero.</p>
            </div>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // here initialize the app
    var myApp = new Framework7();
    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;
    // Add the view
    var mainView = myApp.addView('.view-main', {
        // enable the dynamic navbar for this view:
        dynamicNavbar: true
    });
</script>
</body>
</html>

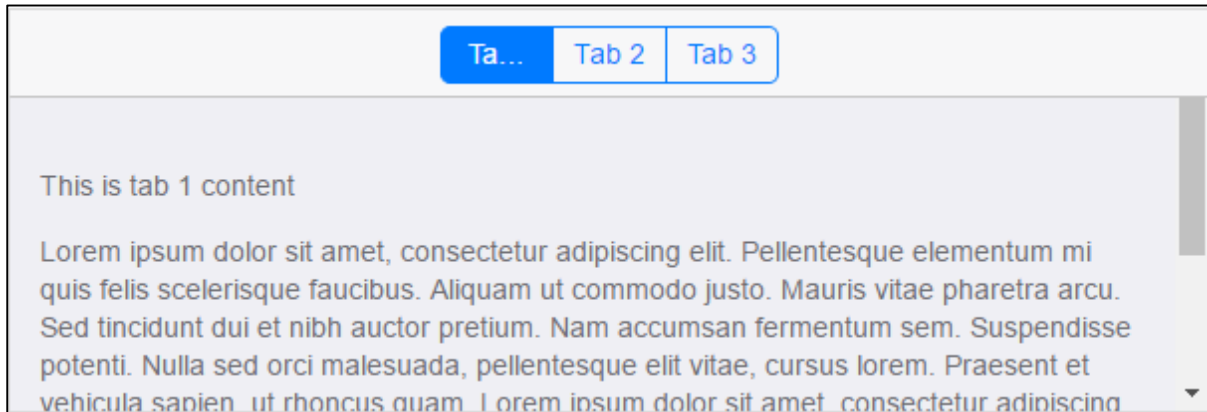
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **switch_tabs_from_navbar.html** file in your server root folder.

- Open this HTML file as http://localhost/switch_tabs_from_navbar.html and the output is displayed as shown below.
- This example places the tabs in navigation bar that allows you to flip between them and looks like navigation bar.



Framework7 – Switch Views From Tab Bar

Description

Single tab can be used to switch between views with its own navigation and layout.

Example

The following example demonstrates the use of switch views from tab bar in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Switch views from Tab bar </title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <style>i.demo-icon-1{width:30px;height:30px;background-
image:url("data:image/svg+xml;charset=utf-8,<svg
xmlns='http://www.w3.org/2000/svg' viewBox='0 0 30 30'
fill='%23929292'><g><circle cx='15.2' cy='8.8' r='1.5'/></g></polygon
```

```

points='16.7,12.3 16.7,11.7 16.7,11.3 13.7,11.3 13.7,12.3 14.7,12.3 14.7,20.3
13.7,20.3 13.7,21.3 17.7,21.3 17.7,20.3 16.7,20.3' /><path
d='M15.2,2.3C8.3,2.3,2.7,7.9,2.7,14.8s5.6,12.5,12.5,12.5c6.9,0,12.5-5.6,12.5-
12.5S22.1,2.3,15.2,2.3z M15.2,26.3 c-6.3,0-11.5-5.2-11.5-
11.5S8.8,3.3,15.2,3.3s11.5,5.2,11.5,11.5S21.5,26.3,15.2,26.3z' /></g></svg>");}.
active i.demo-icon-1{background-image:url("data:image/svg+xml;charset=utf-
8,<svg xmlns='http://www.w3.org/2000/svg' x='0px' y='0px' viewBox='0 0 30 30'
fill='%23007aff'><path
d='M15,2.6C8.1,2.6,2.5,8.2,2.5,15.1C2.5,22,8.1,27.6,15,27.6S27.5,22,27.5,15.1C2
7.5,8.2,21.9,2.6,15,2.6z M15,7.6 c0.8,0,1.5,0.7,1.5,1.5c0,0.8-0.7,1.5-1.5,1.5c-
0.8,0-1.5-0.7-1.5-1.5C13.5,8.3,14.2,7.6,15,7.6z M17.5,21.6h-4v-1h1v-8h-1v-
1h3V12 v0.6v8h1V21.6z' /></svg>");}i.demo-icon-
2{width:25px;height:30px;background-image:url("data:image/svg+xml;charset=utf-
8,<svg xmlns='http://www.w3.org/2000/svg' x='0px' y='0px' viewBox='0 0 25 17'
fill='%23929292'><path d='M19.5,0h-14L0,6v11h25V6L19.5,0z
M5.9,1h13.2l4.5,5H16c0,1.9-1.6,3.4-3.5,3.4C10.6,9.4,9,7.9,9,6H1.4L5.9,1z
M24,16H1V7h7.1 c0.5,1.9,2.3,3.4,4.4,3.4s3.9-1.5,4.4-
3.4H24V16z' /></svg>");}.active i.demo-icon-2{background-
image:url("data:image/svg+xml;charset=utf-8,<svg
xmlns='http://www.w3.org/2000/svg' x='0px' y='0px' viewBox='0 0 25 17'
fill='%23007aff'><path d='M19.5,0h-14L0,6v11h25V6L19.5,0z
M5.9,1h13.2l4.5,5H16c0,1.9-1.6,3.4-
3.5,3.4C10.6,9.4,9,7.9,9,6H1.3L5.9,1z' /></svg>");}i.demo-icon-
3{width:30px;height:30px;background-image:url("data:image/svg+xml;charset=utf-
8,<svg xmlns='http://www.w3.org/2000/svg' x='0px' y='0px' viewBox='0 0 30 19'
fill='%23929292'><path
d='M16.8,1c3.7,0,6.8,3,6.8,6.8l0,0.1c0,0,0.1,0,0.110,111,0C27,9,29,11,29,13.5
c0,2.5-2,4.5-4.5,4.5H6c-2.8,0-5-2.2-5-5 c0-2.1,1.4-4,3.4-4.7L5,8.110.1-
0.6C5.3,6,6.5,5,8,5c0.4,0,0.9,0.1,1.3,0.3l0.9,0.5l0.4-
0.9C11.8,2.5,14.2,1,16.8,1 M16.8,0 c-3.1,0-5.7,1.8-
7,4.4C9.2,4.2,8.6,4,8,4C6,4,4.4,5.4,4.1,7.3C1.7,8.1,0,10.4,0,13c0,3.3,2.7,6,6,6
h18.5v0c3,0,5.5-2.5,5.5-5.5 c0-3-2.5-5.5-5.5-5.5c0-0.1,0-0.2,0-
0.2C24.5,3.5,21.1,0,16.8,0L16.8,0z' /></svg>");}.active i.demo-icon-
3{background-image:url("data:image/svg+xml;charset=utf-8,<svg
xmlns='http://www.w3.org/2000/svg' x='0px' y='0px' viewBox='0 0 30 19'
fill='%23007aff'><path d='M24.5,8c0-0.1,0-0.2,0-0.2C24.5,3.5,21,0,16.8,0c-
3.1,0-5.7,1.8-7,4.4C9.2,4.2,8.6,4,8,4C6,4,4.4,5.4,4,7.3
C1.7,8.1,0,10.4,0,13c0,3.3,2.7,6,6,6h18.5v0c3,0,5.5-2.5,5.5-
5.5C30,10.5,27.5,8,24.5,8z' /></svg>");}</style>

<div class="views tabs toolbar-fixed">
  <div id="tab1" class="view tab active">
    <div class="navbar">
      <div class="navbar-inner">
        <div class="center">View 1</div>
      </div>
    </div>
    <div class="pages navbar-fixed">
      <div data-page="home-1" class="page">
        <div class="page-content">
          <div class="content-block">
            <p>This is view 1</p>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```
        </div>
      </div>
    </div>
  </div>
</div>
<div id="tab2" class="view tab">
  <div class="navbar">
    <div class="navbar-inner">
      <div class="center">View 2</div>
    </div>
  </div>
  <div class="pages navbar-fixed">
    <div data-page="home-2" class="page">
      <div class="page-content">
        <div class="content-block">
          <p>This is view 2</p>
        </div>
      </div>
    </div>
  </div>
</div>
<div id="tab3" class="view tab">
  <div class="navbar">
    <div class="navbar-inner">
      <div class="center">View 3</div>
    </div>
  </div>
  <div class="pages navbar-fixed">
    <div data-page="home-3" class="page">
      <div class="page-content">
        <div class="content-block">
          <p>This is view 3</p>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="toolbar tabbar">
```

```

        <div class="toolbar-inner"><a href="#tab1" class="tab-link active">
<i class="icon demo-icon-1"></i></a><a href="#tab2" class="tab-link"><i
class="icon demo-icon-2"></i></a><a href="#tab3" class="tab-link"><i
class="icon demo-icon-3"></i></a></div>

    </div>

</div>

    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>

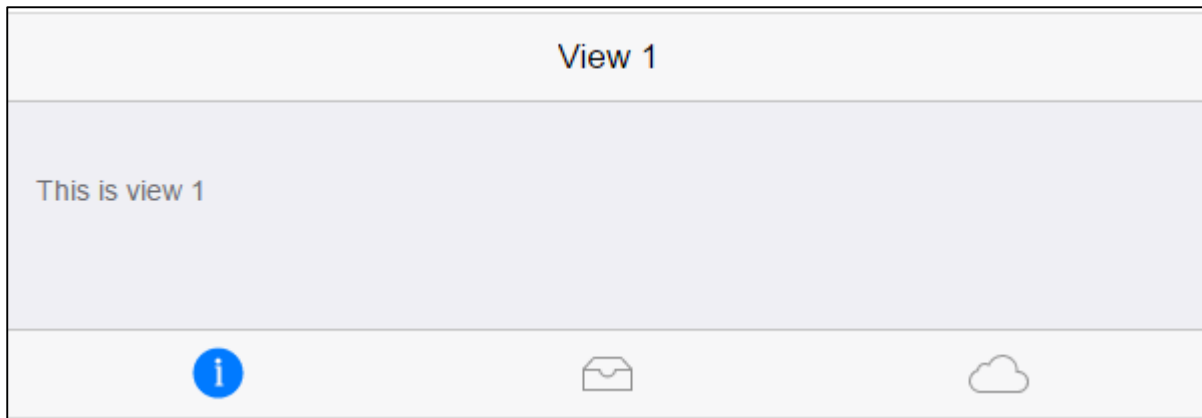
    <script>
        // here initialize the app
        var myApp = new Framework7();
        // If your using custom DOM library, then save it to $$ variable
        var $$ = Dom7;
        // Add the view
        var mainView = myApp.addView('.view-main', {
        // enable the dynamic navbar for this view:
        dynamicNavbar: true
        });
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **switch_views_from_tab_bar.html** file in your server root folder.
- Open this HTML file as http://localhost/switch_views_from_tab_bar.html and the output is displayed as shown below.
- The example provides a single tab that can be used to switch between views with its own navigation and layout.



Framework7 – Animated Tabs

Description

You can use simple transition (animation) to switch tabs and use additional class *tabs-animated-wrap* to wrapper for *tabs*.

Example

The following example demonstrates the use of animated tabs in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Animated Tabs </title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
```

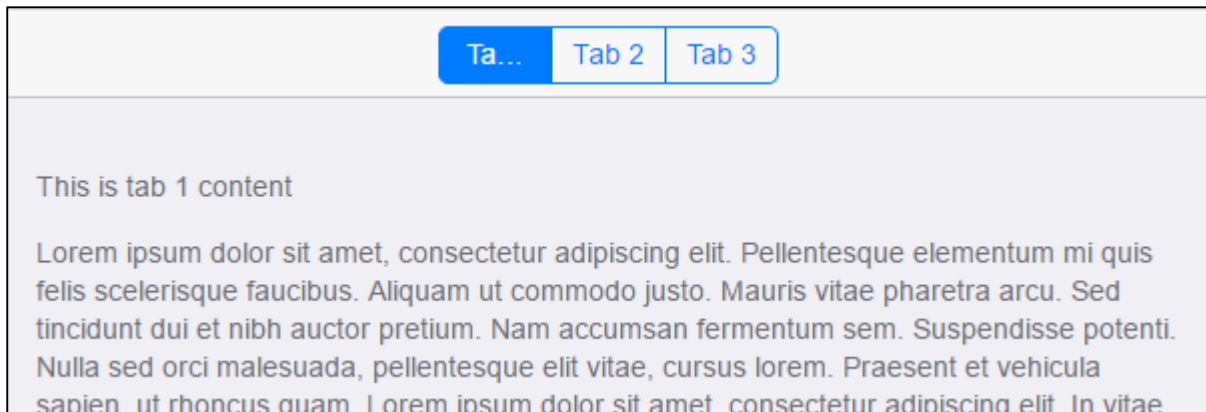


```

        <div class="left"> </div>
        <div class="center">
            <div class="buttons-row"><a href="#tab1" class="tab-
link active button">Tab 1</a><a href="#tab2" class="tab-link button">Tab
2</a><a href="#tab3" class="tab-link button">Tab 3</a></div>
        </div>
        <div class="right"> </div>
    </div>
</div>
<div class="page-content">
    <div class="tabs-animated-wrap">
        <div class="tabs">
            <div id="tab1" class="tab active">
                <div class="content-block">
                    <p>This is tab 1 content</p>
                    <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Pellentesque elementum mi quis felis scelerisque faucibus.
Aliquam ut commodo justo. Mauris vitae pharetra arcu. Sed tincidunt dui et nibh
auctor pretium. Nam accumsan fermentum sem. Suspendisse potenti. Nulla sed orci
malesuada, pellentesque elit vitae, cursus lorem. Praesent et vehicula sapien,
ut rhoncus quam. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In
vitae mi nec lorem aliquet venenatis quis nec nibh. Aenean sit amet leo ligula.
Fusce in massa et nisl dictum ultricies et vitae dui. Sed sagittis quis diam
sed lobortis. Donec in massa pharetra, tristique purus vitae, consequat mauris.
Aliquam tellus ante, pharetra in mattis ut, dictum quis erat.</p>
                </div>
            </div>
            <div id="tab2" class="tab">
                <div class="content-block">
                    <p>This is tab 2 content</p>
                    <p>Ut ac lobortis lacus, non pellentesque
arcu. Quisque sodales sapien malesuada, condimentum nunc at, viverra lacus.
Class aptent taciti sociosqu ad litora torquent per conubia nostra, per
inceptos himenaeos. Vivamus eu pulvinar turpis, id tristique quam. Aenean
venenatis molestie diam, sit amet condimentum nisl pretium id. Donec diam
tortor, mollis in vehicula id, vehicula consectetur nulla. Quisque posuere
rutrum mauris, eu rutrum turpis blandit at. Proin volutpat tortor sit amet
metus porttitor accumsan. Class aptent taciti sociosqu ad litora torquent per
conubia nostra, per inceptos himenaeos. Ut dapibus posuere dictum.</p>
                </div>
            </div>
            <div id="tab3" class="tab">
                <div class="content-block">
                    <p>This is tab 3 content</p>

```


- This code allows switching between tabs with simple transition.



Framework7 – Swipeable Tabs

Description

Swipeable tabs are required component, which can switch the tabs with simple transition by using additional *tabs-swipeable-wrap* wrapper class for the *tabs*.

Example

The following example demonstrates the swiping of tabs with simple transition in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Swipeable Tabs </title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages navbar-fixed">
```

```

<div data-page="home" class="page with-subnavbar">
  <div class="navbar">
    <div class="navbar-inner">
      <div class="center">Swipeable Tabs</div>
      <div class="subnavbar">
        <div class="buttons-row"><a href="#tab1"
class="button active tab-link">Tab 1</a><a href="#tab2" class="button tab-
link">Tab 2</a><a href="#tab3" class="button tab-link">Tab 3</a></div>
        </div>
      </div>
    </div>
    <div class="tabs-swipeable-wrap">
      <div class="tabs">
        <div id="tab1" class="page-content with-subnavbar tab
active">
          <div class="content-block">
            <p>This is tab 1 content</p>
            <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Pellentesque elementum mi quis felis scelerisque faucibus.
Aliquam ut commodo justo. Mauris vitae pharetra arcu. Sed tincidunt dui et nibh
auctor pretium. Nam accumsan fermentum sem. Suspendisse potenti. Nulla sed orci
malesuada, pellentesque elit vitae, cursus lorem. Praesent et vehicula sapien,
ut rhoncus quam. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In
vitae mi nec lorem aliquet venenatis quis nec nibh. Aenean sit amet leo ligula.
Fusce in massa et nisl dictum ultricies et vitae dui. Sed sagittis quis diam
sed lobortis. Donec in massa pharetra, tristique purus vitae, consequat mauris.
Aliquam tellus ante, pharetra in mattis ut, dictum quis erat.</p>
          </div>
        </div>
        <div id="tab2" class="page-content tab">
          <div class="content-block">
            <p>This is tab 2 content</p>
            <p>Ut ac lobortis lacus, non pellentesque arcu.
Quisque sodales sapien malesuada, condimentum nunc at, viverra lacus. Class
aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos
himenaeos. Vivamus eu pulvinar turpis, id tristique quam. Aenean venenatis
molestie diam, sit amet condimentum nisl pretium id. Donec diam tortor, mollis
in vehicula id, vehicula consectetur nulla. Quisque posuere rutrum mauris, eu
rutrum turpis blandit at. Proin volutpat tortor sit amet metus porttitor
accumsan. Class aptent taciti sociosqu ad litora torquent per conubia nostra,
per inceptos himenaeos. Ut dapibus posuere dictum.</p>
          </div>
        </div>
        <div id="tab3" class="page-content tab">
          <div class="content-block">

```

```

        <p>This is tab 3 content</p>
        <p>Nulla gravida libero eget lobortis iaculis. In
sed elit eu nibh adipiscing faucibus. Sed ac accumsan lacus. In ut diam quis
turpis fringilla volutpat. In ultrices dignissim consequat. Cras pretium tortor
et lorem condimentum posuere. Nulla facilisi. Suspendisse pretium egestas lacus
ac laoreet. Mauris rhoncus quis ipsum quis tristique. Vivamus ultricies urna
quis nunc egestas, in euismod turpis fringilla. Nam tellus massa, vehicula eu
sapien non, dapibus tempor lorem. Fusce placerat orci arcu, eu dignissim enim
porttitor vel. Nullam porttitor vel dolor sed feugiat. Suspendisse potenti.
Maecenas ac mattis odio. Sed vel ultricies lacus, sed posuere libero.</p>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    // here initialize the app
    var myApp = new Framework7();
    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;
    // Add the view
    var mainView = myApp.addView('.view-main', {
    // enable the dynamic navbar for this view:
    dynamicNavbar: true
    });
</script>
</body>
</html>

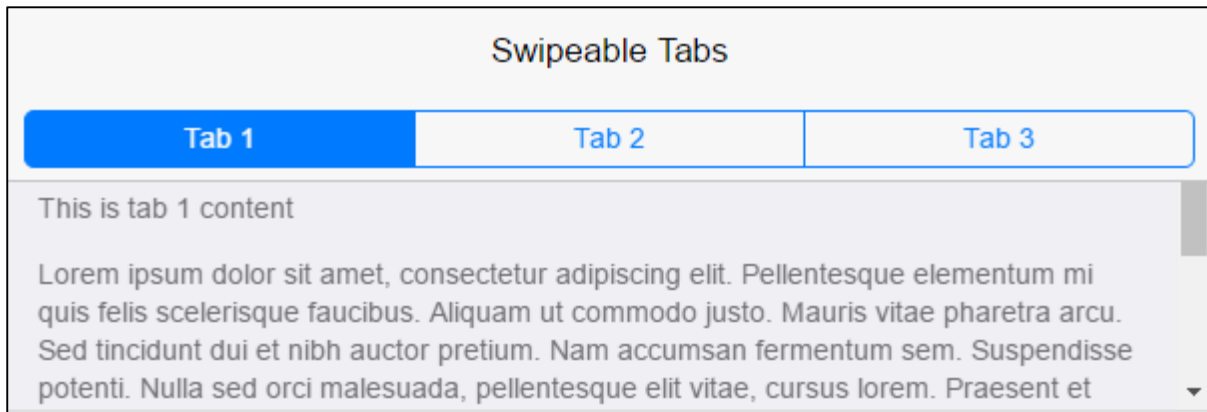
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **swipeable_tabs.html** file in your server root folder.
- Open this HTML file as http://localhost/swipeable_tabs.html and the output is displayed as shown below.

- The example can switch between the tabs by swiping the tabs.



Framework7 – Tabs JavaScript Events

Description

JavaScript events can be used when you are working with JavaScript code for the tabs.

Example

The following example demonstrates the use of JavaScript events for tabs in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Tabs Events </title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
```

```

        <div class="left"> </div>
        <div class="center">Tabs Events</div>
        <div class="right"> </div>
    </div>
</div>
<div class="page-content">
    <div class="content-block">
        <div class="buttons-row"><a href="#tab1" class="tab-
link active button">Tab 1</a><a href="#tab2" class="tab-link button">Tab
2</a><a href="#tab3" class="tab-link button">Tab 3</a></div>
    </div>
    <div class="tabs">
        <div id="tab1" class="tab active">
            <div class="content-block">
                <p>This is tab 1 content</p>
                <p>Ut ac lobortis lacus, non pellentesque arcu.
Quisque sodales sapien malesuada, condimentum nunc at, viverra lacus. Class
aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos
himenaeos. Vivamus eu pulvinar turpis, id tristique quam. Aenean venenatis
molestie diam, sit amet condimentum nisl pretium id. Donec diam tortor, mollis
in vehicula id, vehicula consectetur nulla. Quisque posuere rutrum mauris, eu
rutrum turpis blandit at. Proin volutpat tortor sit amet metus porttitor
accumsan. Class aptent taciti sociosqu ad litora torquent per conubia nostra,
per inceptos himenaeos. Ut dapibus posuere dictum.</p>
            </div>
        </div>
        <div id="tab2" class="tab">
            <div class="content-block">
                <p>This is tab 2 content</p>
                <p>Ut ac lobortis lacus, non pellentesque arcu.
Quisque sodales sapien malesuada, condimentum nunc at, viverra lacus. Class
aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos
himenaeos. Vivamus eu pulvinar turpis, id tristique quam. Aenean venenatis
molestie diam, sit amet condimentum nisl pretium id. Donec diam tortor, mollis
in vehicula id, vehicula consectetur nulla. Quisque posuere rutrum mauris, eu
rutrum turpis blandit at. Proin volutpat tortor sit amet metus porttitor
accumsan. Class aptent taciti sociosqu ad litora torquent per conubia nostra,
per inceptos himenaeos. Ut dapibus posuere dictum.</p>
            </div>
        </div>
        <div id="tab3" class="tab">
            <div class="content-block">
                <p>This is tab 3 content</p>
                <p>Nulla gravida libero eget lobortis iaculis. In
sed elit eu nibh adipiscing faucibus. Sed ac accumsan lacus. In ut diam quis

```

turpis fringilla volutpat. In ultrices dignissim consequat. Cras pretium tortor et lorem condimentum posuere. Nulla facilisi. Suspendisse pretium egestas lacus ac laoreet. Mauris rhoncus quis ipsum quis tristique. Vivamus ultricies urna quis nunc egestas, in euismod turpis fringilla. Nam tellus massa, vehicula eu sapien non, dapibus tempor lorem. Fusce placerat orci arcu, eu dignissim enim porttitor vel. Nullam porttitor vel dolor sed feugiat. Suspendisse potenti. Maecenas ac mattis odio. Sed vel ultricies lacus, sed posuere libero.</p>

```

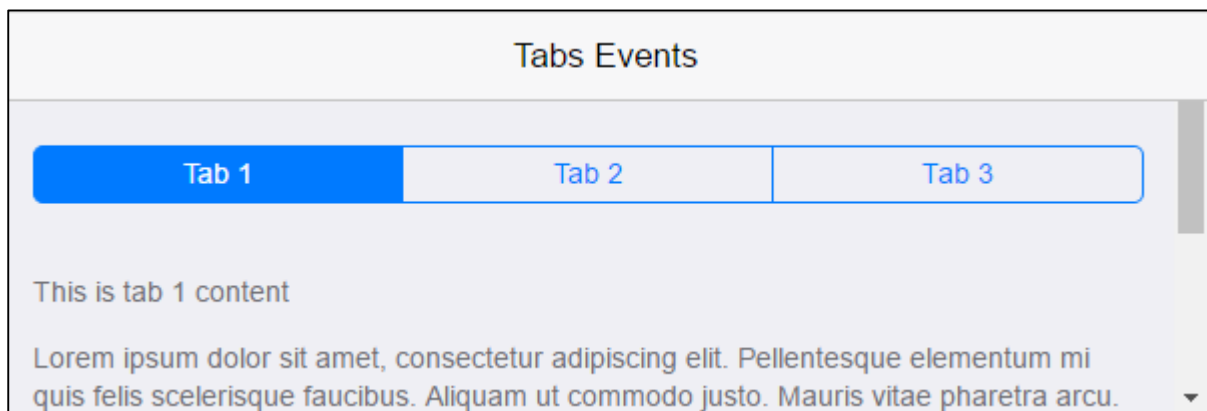
                </div>
            </div>
        </div>
    </div>
</div>
<script>
    var myApp = new Framework7();
    var $$ = Dom7;
    $('#tab1').on('show', function () {
        myApp.alert('Tab 1 is visible');
    });
    $('#tab2').on('show', function () {
        myApp.alert('Tab 2 is visible');
    });
    $('#tab3').on('show', function () {
        myApp.alert('Tab 3 is visible');
    });
</script>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.js"></script>
<script>
    // here initialize the app
    var myApp = new Framework7();
    // If your using custom DOM library, then save it to $$ variable
    var $$ = Dom7;
    // Add the view
    var mainView = myApp.addView('.view-main', {
    // enable the dynamic navbar for this view:
    dynamicNavbar: true
    });
</script>
</body>
</html>

```


Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **tabs_js_events.html** file in your server root folder.
- Open this HTML file as `http://localhost/tabs_js_events.html` and the output is displayed as shown below.
- The example allows switching between tabs and the alert event occurs when the tab becomes visible.



Framework7 – Show Tab using JavaScript

Description

You can switch or show the tab using the following JavaScript method.

```
myApp.showTab(tab)
```

- *tab*- It's a required HTML element or string used to open the tab.

This method returns **true** if the tab has been changed, otherwise it returns **false**.

Example

The following example demonstrates the use of the show tab using JavaScript in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
```

```

    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Show Tabs Events </title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main">
            <div class="pages">
                <div data-page="home" class="page navbar-fixed">
                    <div class="navbar">
                        <div class="navbar-inner">
                            <div class="left"> </div>
                            <div class="center">Tabs Events</div>
                            <div class="right"> </div>
                        </div>
                    </div>
                    <div class="page-content">
                        <div class="content-block">
                            <div class="buttons-row"><a href="#tab1" class="tab-
link active button">Tab 1</a><a href="#tab2" class="tab-link button">Tab
2</a><a href="#tab3" class="tab-link button">Tab 3</a></div>
                            </div>
                            <div class="tabs">
                                <div id="tab1" class="tab active">
                                    <div class="content-block">
                                        <p>This is tab 1 content</p>
                                        <p> <a href="#" class="show-tab-2">Show Tab
2</a></p>
                                        <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Pellentesque elementum mi quis felis scelerisque faucibus.
Aliquam ut commodo justo. Mauris vitae pharetra arcu. Sed tincidunt dui et nibh
auctor pretium. Nam accumsan fermentum sem. Suspendisse potenti. Nulla sed orci
malesuada, pellentesque elit vitae, cursus lorem. Praesent et vehicula sapien,
ut rhoncus quam. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In
vitae mi nec lorem aliquet venenatis quis nec nibh. Aenean sit amet leo ligula.

```

```
Fusce in massa et nisl dictum ultricies et vitae dui. Sed sagittis quis diam
sed lobortis. Donec in massa pharetra, tristique purus vitae, consequat mauris.
Aliquam tellus ante, pharetra in mattis ut, dictum quis erat.</p>
```

```
</div>
```

```
</div>
```

```
<div id="tab2" class="tab">
```

```
<div class="content-block">
```

```
<p>This is tab 2 content</p>
```

```
<p> <a href="#" class="show-tab-3">Show Tab
```

```
3</a></p>
```

```
<p>Ut ac lobortis lacus, non pellentesque arcu.
```

```
Quisque sodales sapien malesuada, condimentum nunc at, viverra lacus. Class
aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos
himenaeos. Vivamus eu pulvinar turpis, id tristique quam. Aenean venenatis
molestie diam, sit amet condimentum nisl pretium id. Donec diam tortor, mollis
in vehicula id, vehicula consectetur nulla. Quisque posuere rutrum mauris, eu
rutrum turpis blandit at. Proin volutpat tortor sit amet metus porttitor
accumsan. Class aptent taciti sociosqu ad litora torquent per conubia nostra,
per inceptos himenaeos. Ut dapibus posuere dictum.</p>
```

```
</div>
```

```
</div>
```

```
<div id="tab3" class="tab">
```

```
<div class="content-block">
```

```
<p>This is tab 3 content</p>
```

```
<p> <a href="#" class="show-tab-1">Show Tab
```

```
1</a></p>
```

```
<p>Nulla gravida libero eget lobortis iaculis. In
```

```
sed elit eu nibh adipiscing faucibus. Sed ac accumsan lacus. In ut diam quis
turpis fringilla volutpat. In ultrices dignissim consequat. Cras pretium tortor
et lorem condimentum posuere. Nulla facilisi. Suspendisse pretium egestas lacus
ac laoreet. Mauris rhoncus quis ipsum quis tristique. Vivamus ultricies urna
quis nunc egestas, in euismod turpis fringilla. Nam tellus massa, vehicula eu
sapien non, dapibus tempor lorem. Fusce placerat orci arcu, eu dignissim enim
porttitor vel. Nullam porttitor vel dolor sed feugiat. Suspendisse potenti.
Maecenas ac mattis odio. Sed vel ultricies lacus, sed posuere libero.</p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<script type="text/javascript"
```

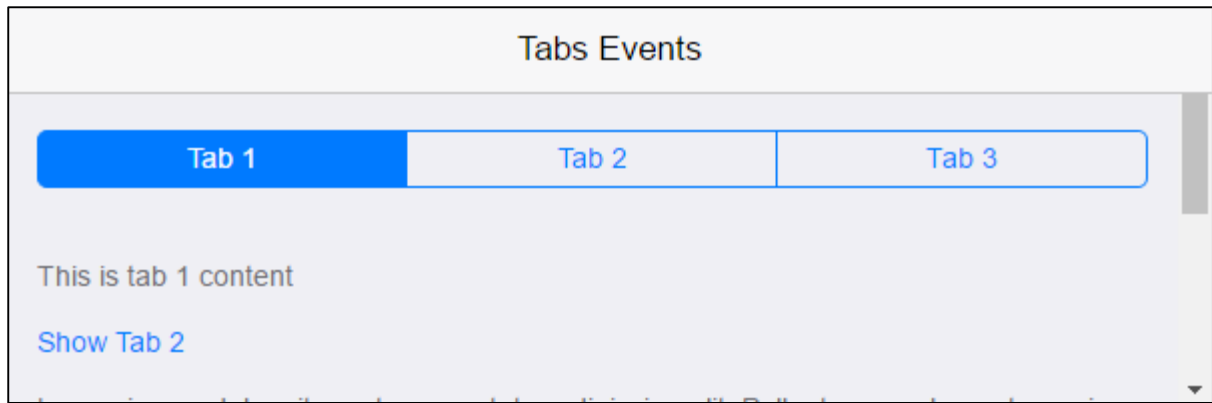
```
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
```

```
<script>
  // here initialize the app
  var myApp = new Framework7();
  // If your using custom DOM library, then save it to $$ variable
  var $$ = Dom7;
  // Add the view
  var mainView = myApp.addView('.view-main', {
    // enable the dynamic navbar for this view:
    dynamicNavbar: true
  });
  $$('.show-tab-1').on('click', function () {
    myApp.showTab('#tab1');
  });
  $$('.show-tab-2').on('click', function () {
    myApp.showTab('#tab2');
  });
  $$('.show-tab-3').on('click', function () {
    myApp.showTab('#tab3');
  });
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **show_tab_using_js.html** file in your server root folder.
- Open this HTML file as http://localhost/show_tab_using_js.html and the output is displayed as shown below.
- This example can switch or show the content of another tab inside the tab.



21. Framework7 – Swiper Slider

Description

The swiper slider is the most powerful and modern touch slider and comes into Framework7 with lots of features.

The following HTML layout shows the swiper slider:

```
<!-- Container class for slider -->
<div class="swiper-container">
  <!-- Wrapper class for slider -->
  <div class="swiper-wrapper">
    <!-- Slides -->
    <div class="swiper-slide">Slide 1</div>
    <div class="swiper-slide">Slide 2</div>
    <div class="swiper-slide">Slide 3</div>
    ... other slides ...
  </div>
  <!-- Define pagination, if it is required -->
  <div class="swiper-pagination"></div>
</div>
```

The following classes are used for swiper slider:

- **swiper-container:** It is a required element for main slider container and it is used for slides and paginations.
- **Swiper-wrapper:** It is a required element for additional wrapper slides.
- **swiper-slide:** It is a single slide element and it could contain any HTML inside.
- **swiper-pagination:** It is optional for pagination bullets and those are created automatically.

You can initialize the swiper with JavaScript using its related methods:

```
myApp.swiper(swiperContainer,parameters)
```

OR

```
new Swiper(swiperContainer, parameters)
```

Both the methods are used to initialize the slider with options.

The above given methods contain the following parameters:

- **swiperContainer:** It is *HTMLElement* or *string* of a swiper container and it is required.
- **parameters:** These are optional elements containing object with swiper parameters.

For example:

```
var mySwiper = app.swiper('.swiper-container', {
  speed: 400,
  spaceBetween: 100
});
```

It is possible to access a swiper's instance and the *swiper* property of the slider's container has the following HTML element:

```
var mySwiper = $$('.swiper-container')[0].swiper;
// Here you can use all slider methods like:
mySwiper.slideNext();
```

You can see the different ways and types of swiper in the following table:

S. No.	Swiper Types & Description
1	Default Swiper With Pagination It is a modern touch slider and the swiper swipes horizontally, by default.
2	Vertical Swiper This one also works as a default swiper but it swipes vertically.
3	With Space Between Slides This swiper is used to give space between two slides.
4	Multiple Swipers This swiper uses more than one swipers on a single page.
5	Nested Swipers It is the combination of vertical and horizontal slides.
6	Custom Controls It is used for custom controls to choose or swipe any slide.
7	Lazy Loading It can be used for multimedia file, which takes time to load.

Framework7 – Default Swiper with Pagination

Description

The swiper slider is the most powerful and modern touch slider and it swipes horizontally, by default.

Example

The following example demonstrates the use of default swiper with pagination in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Default Swiper with Pagination</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Default Swiper</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">
              <div class="swiper-container">
                <div class="swiper-wrapper">
                  <div class="swiper-slide"><span>Slide 1</span></div>
```



```

        <div class="swiper-slide"><span>Slide 2</span></div>
        <div class="swiper-slide"><span>Slide 3</span></div>
        <div class="swiper-slide"><span>Slide 4</span></div>
    </div>
    <div class="swiper-pagination"></div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<style>.swiper-container{height:100%;}.swiper-
slide{background:#fff;}.swiper-slide span{text-
align:center;display:block;margin:20px;font-size:21px;}</style>
<script>
    var myApp = new Framework7();

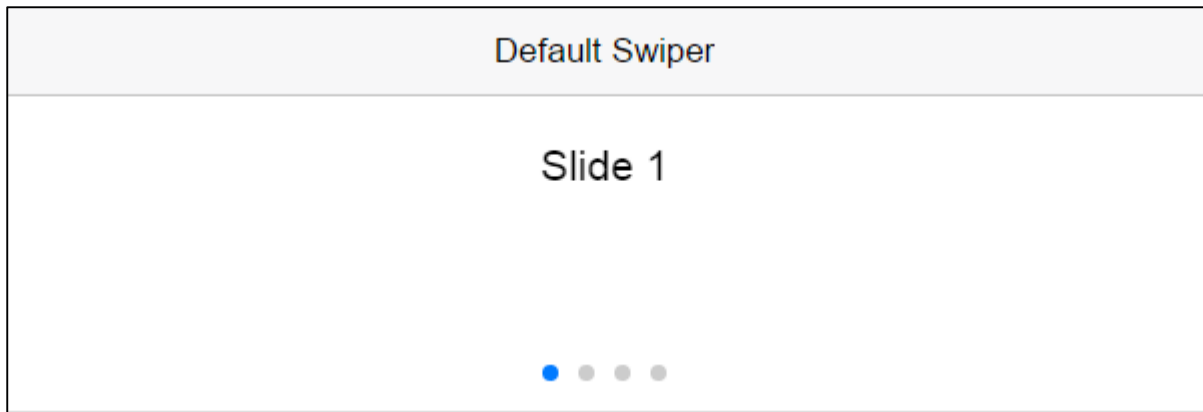
    var mySwiper = myApp.swiper('.swiper-container', {
        pagination:'.swiper-pagination'
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **default_swiper_pagination.html** file in your server root folder.
- Open this HTML file as http://localhost/default_swiper_pagination.html and the output is displayed as shown below.
- When you run the code, it will provide the slider by swiping it horizontally.



Framework7 – Vertical Swiper

Description

It is used to swipe vertically and it is a powerful and modern touch slider.

Example

The following example demonstrates the use of vertical swiper in the Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Vertical Swiper</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
```

```

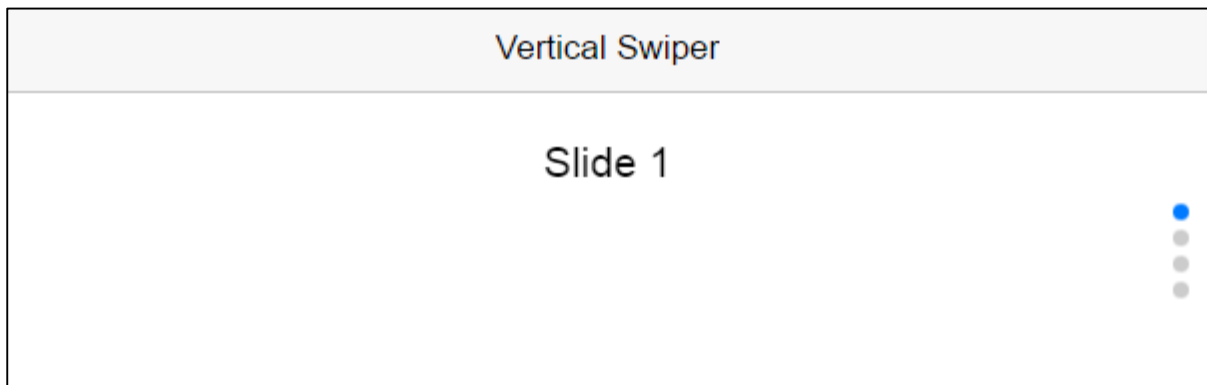
        <div class="center">Vertical Swiper</div>
        <div class="right"> </div>
    </div>
</div>
<div class="page-content">
    <div class="swiper-container">
        <div class="swiper-wrapper">
            <div class="swiper-slide"><span>Slide 1</span></div>
            <div class="swiper-slide"><span>Slide 2</span></div>
            <div class="swiper-slide"><span>Slide 3</span></div>
            <div class="swiper-slide"><span>Slide 4</span></div>
        </div>
        <div class="swiper-pagination"></div>
    </div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<style>.swiper-container{height:100%;}.swiper-
slide{background:#fff;}.swiper-slide span{text-
align:center;display:block;margin:20px;font-size:21px;}</style>
<script>
    var myApp = new Framework7();
    var mySwiper = myApp.swiper('.swiper-container', {
        pagination:'.swiper-pagination',
        direction:'vertical',
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **vertical_swiper.html** file in your server root folder.
- Open this HTML file as http://localhost/vertical_swiper.html and the output is displayed as shown below.
- When you run the code, it will provide the slider by swiping it vertically.



Framework7 – With Space Between Slides

Description

It gives space between two slides.

Example

The following example demonstrates the use of space between slides in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>With space between slides</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
```

```

<div class="view view-main">
  <div class="pages">
    <div data-page="home" class="page navbar-fixed">
      <div class="navbar">
        <div class="navbar-inner">
          <div class="left"> </div>
          <div class="center">Space Between Slides</div>
          <div class="right"> </div>
        </div>
      </div>
      <div class="page-content">
        <div class="swiper-container">
          <div class="swiper-wrapper">
            <div class="swiper-slide"><span>Slide 1</span></div>
            <div class="swiper-slide"><span>Slide 2</span></div>
            <div class="swiper-slide"><span>Slide 3</span></div>
            <div class="swiper-slide"><span>Slide 4</span></div>
          </div>
          <div class="swiper-pagination"></div>
        </div>
      </div>
    </div>
  </div>
  <div>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
    <style>.swiper-container{height:100%;}.swiper-
slide{background:#fff;}.swiper-slide span{text-
align:center;display:block;margin:20px;font-size:21px;}</style>
    <script>
      var myApp = new Framework7();

      var mySwiper = myApp.swiper('.swiper-container', {
        pagination: '.swiper-pagination',
        spaceBetween: 100
      });
    </script>
  </div>

```

```
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **with_space_between_slides.html** file in your server root folder.
- Open this HTML file as http://localhost/with_space_between_slides.html and the output is displayed as shown below.
- When you run the code, it will provide the slider by keeping space between slides.



Framework7 – Multiple Swipers

Description

You can use more than one swipers on a single page.

Example

The following example demonstrates the use of multiple swipers in the Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Multiple Swipers</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ios.colors.min.css">
</head>
<body>
  <div class="views">
    <div class="view view-main">
      <div class="pages">
        <div data-page="home" class="page navbar-fixed">
          <div class="navbar">
            <div class="navbar-inner">
              <div class="left"> </div>
              <div class="center">Multiple Swipers</div>
              <div class="right"> </div>
            </div>
          </div>
          <div class="page-content">
            <div class="content-block-title">1 Slide Per View, 50px
Between</div>
            <div class="swiper-container swiper-1">
              <div class="swiper-pagination"></div>
              <div class="swiper-wrapper">
                <div class="swiper-slide"><span>Slide 1</span></div>
                <div class="swiper-slide"><span>Slide 2</span></div>
                <div class="swiper-slide"><span>Slide 3</span></div>
                <div class="swiper-slide"><span>Slide 4</span></div>
                <div class="swiper-slide"><span>Slide 5</span></div>
                <div class="swiper-slide"><span>Slide 6</span></div>
                <div class="swiper-slide"><span>Slide 7</span></div>
                <div class="swiper-slide"><span>Slide 8</span></div>
                <div class="swiper-slide"><span>Slide 9</span></div>
                <div class="swiper-slide"><span>Slide
10</span></div>
              </div>
            </div>
            <div class="content-block-title">2 Slides Per View, 20px
Between</div>
            <div class="swiper-container swiper-2">
              <div class="swiper-pagination"></div>
              <div class="swiper-wrapper">

```

```

<div class="swiper-slide"><span>Slide 1</span></div>
<div class="swiper-slide"><span>Slide 2</span></div>
<div class="swiper-slide"><span>Slide 3</span></div>
<div class="swiper-slide"><span>Slide 4</span></div>
<div class="swiper-slide"><span>Slide 5</span></div>
<div class="swiper-slide"><span>Slide 6</span></div>
<div class="swiper-slide"><span>Slide 7</span></div>
<div class="swiper-slide"><span>Slide 8</span></div>
<div class="swiper-slide"><span>Slide 9</span></div>
<div class="swiper-slide"><span>Slide
10</span></div>
</div>
</div>
<div class="content-block-title">3 Slides Per View, 10px
Between</div>
<div class="swiper-container swiper-3">
  <div class="swiper-pagination"></div>
  <div class="swiper-wrapper">
    <div class="swiper-slide"><span>Slide 1</span></div>
    <div class="swiper-slide"><span>Slide 2</span></div>
    <div class="swiper-slide"><span>Slide 3</span></div>
    <div class="swiper-slide"><span>Slide 4</span></div>
    <div class="swiper-slide"><span>Slide 5</span></div>
    <div class="swiper-slide"><span>Slide 6</span></div>
    <div class="swiper-slide"><span>Slide 7</span></div>
    <div class="swiper-slide"><span>Slide 8</span></div>
    <div class="swiper-slide"><span>Slide 9</span></div>
    <div class="swiper-slide"><span>Slide
10</span></div>
  </div>
</div>
<div class="content-block-title">Vertical, 0px
Between</div>
<div class="swiper-container swiper-vertical">
  <div class="swiper-pagination"></div>
  <div class="swiper-wrapper">
    <div class="swiper-slide"><span>Slide 1</span></div>
    <div class="swiper-slide"><span>Slide 2</span></div>
    <div class="swiper-slide"><span>Slide 3</span></div>

```



```

        <div class="swiper-slide"><span>Slide 4</span></div>
        <div class="swiper-slide"><span>Slide 5</span></div>
    </div>
</div>
<div class="content-block-title">Slow speed</div>
<div class="swiper-container swiper-slow">
    <div class="swiper-pagination"></div>
    <div class="swiper-wrapper">
        <div class="swiper-slide"><span>Slide 1</span></div>
        <div class="swiper-slide"><span>Slide 2</span></div>
        <div class="swiper-slide"><span>Slide 3</span></div>
        <div class="swiper-slide"><span>Slide 4</span></div>
        <div class="swiper-slide"><span>Slide 5</span></div>
        <div class="swiper-slide"><span>Slide 6</span></div>
        <div class="swiper-slide"><span>Slide 7</span></div>
        <div class="swiper-slide"><span>Slide 8</span></div>
        <div class="swiper-slide"><span>Slide 9</span></div>
        <div class="swiper-slide"><span>Slide
10</span></div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<style>.swiper-slide{background:#fff;box-sizing:border-box;border:1px
solid #ccc;line-height:120px;text-align:center;}.swiper-slide span{font-
size:17px;}.swiper-container{height:120px;margin:0px 0 35px;}</style>
<script>
    var myApp = new Framework7();
    // 1 Slide Per View, 50px Between
    var mySwiper1 = myApp.swiper('.swiper-1', {
        pagination:'.swiper-1 .swiper-pagination',
        spaceBetween: 50
    });
    // 2 Slides Per View, 20px Between

```

```
var mySwiper2 = myApp.swiper('.swiper-2', {
  pagination:'.swiper-2 .swiper-pagination',
  spaceBetween: 20,
  slidesPerView: 2
});
// 3 Slides Per View, 10px Between
var mySwiper3 = myApp.swiper('.swiper-3', {
  pagination:'.swiper-3 .swiper-pagination',
  spaceBetween: 10,
  slidesPerView: 3
});
// Vertical, 0px Between
var mySwiperVertical = myApp.swiper('.swiper-vertical', {
  pagination:'.swiper-vertical .swiper-pagination',
  direction: 'vertical'
});
// Slow speed
var mySwiperSlow = myApp.swiper('.swiper-slow', {
  pagination:'.swiper-slow .swiper-pagination',
  speed: 600
});
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **multiple_swipers.html** file in your server root folder.
- Open this HTML file as http://localhost/multiple_swipers.html and the output is displayed as shown below.
- The example allows multiple sliders on a single page.

Multiple Swipers

1 SLIDE PER VIEW, 50PX BETWEEN

Slide 1

2 SLIDES PER VIEW, 20PX BETWEEN

Slide 1
Slide 1

Slide 2
Slide 2

3 SLIDES PER VIEW, 10PX BETWEEN

Slide 1

Slide 2

Slide 3

VERTICAL, 0PX BETWEEN

Slide 1

SLOW SPEED

Slide 1

Framework7 – Nested Swipers

Description

The nested swiper is a combination of vertical and horizontal swipers.

Example

The following example demonstrates the use of nested swipers in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Nested Swipers</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Nested Sliders</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">
              <div data-pagination=".swiper-pagination-h" class="swiper-
container swiper-init">
                <div class="swiper-pagination swiper-pagination-
h"></div>
                <div class="swiper-wrapper">
```

```

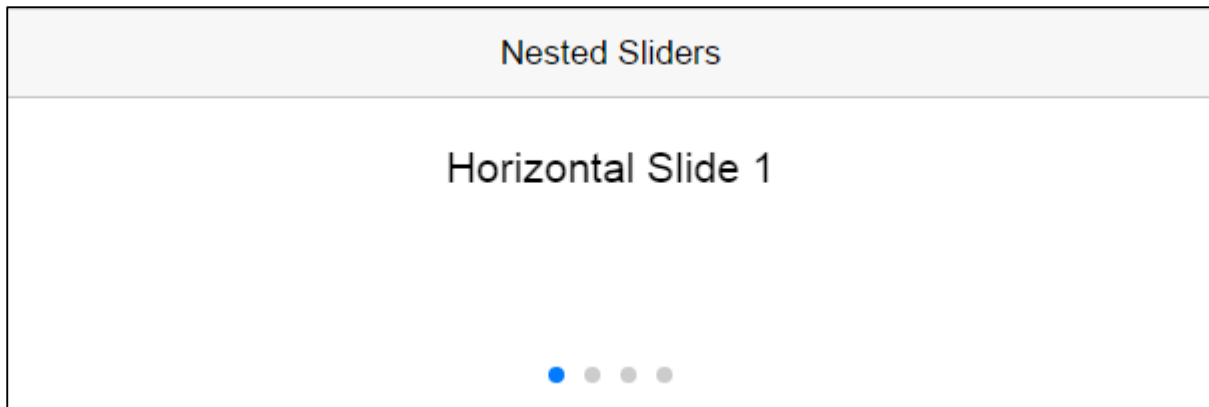
1</span></div>
    <div class="swiper-slide"><span>Horizontal Slide
    <div class="swiper-slide">
        <div data-pagination=".swiper-pagination-v" data-
direction="vertical" class="swiper-container swiper-init">
            <div class="swiper-pagination swiper-
pagination-v"></div>
                <div class="swiper-wrapper">
                    <div class="swiper-slide"><span>Vertical
Slide 1</span></div>
                    <div class="swiper-slide"><span>Vertical
Slide 2</span></div>
                    <div class="swiper-slide"><span>Vertical
Slide 3</span></div>
                </div>
            </div>
        </div>
    <div class="swiper-slide"><span>Horizontal Slide
3</span></div>
    <div class="swiper-slide"><span>Horizontal Slide
4</span></div>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
    <style>.swiper-container{height:100%;}.swiper-
slide{background:#fff;}.swiper-slide span{text-
align:center;display:block;margin:20px;font-size:21px;}</style>
    <script>var myApp = new Framework7(); </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **nested_swipers.html** file in your server root folder.
- Open this HTML file as http://localhost/nested_swipers.html and the output is displayed as shown below.
- The example provides the nested swiper with a combination of vertical and horizontal swipers.



Framework7 – Custom Controls

Description

This one is used for custom controls to choose or swipe between any slides of a page.

Example

The following example demonstrates the use of custom controls in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Custom Controls</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
```

```

        <div class="center">Custom Controls</div>
        <div class="right"> </div>
    </div>
</div>
<div class="page-content">
    <div class="swiper-custom">
        <div class="swiper-container">
            <div class="swiper-pagination"></div>
            <div class="swiper-wrapper">
                <div class="swiper-slide"><span>Slide
1</span></div>
                <div class="swiper-slide"><span>Slide
2</span></div>
                <div class="swiper-slide"><span>Slide
3</span></div>
                <div class="swiper-slide"><span>Slide
4</span></div>
                <div class="swiper-slide"><span>Slide
5</span></div>
                <div class="swiper-slide"><span>Slide
6</span></div>
                <div class="swiper-slide"><span>Slide
7</span></div>
            </div>
        </div>
        <div class="swiper-button-prev"> </div>
        <div class="swiper-button-next"></div>
    </div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>

<style>.swiper-custom{position:relative;height:100%;}.swiper-
container{height:100%;}.swiper-slide{background:#fff;}.swiper-slide span{text-
align:center;display:block;margin:20px;font-size:21px;}.swiper-pagination-
bullet{width:10px;height:10px;background:none;border:1px solid #aaa;border-
radius:0 0 0 0;opacity:1;cursor:pointer;}.swiper-pagination-bullet-
active{background:#007aff;border-color:#007aff;}</style>

<script>
    var myApp = new Framework7();
    var mySwiper = myApp.swiper('.swiper-container', {
        pagination: '.swiper-pagination',
        paginationHide: false,
        paginationClickable: true,
        nextButton: '.swiper-button-next',

```

```

        prevButton: '.swiper-button-prev',
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **custom_controls.html** file in your server root folder.
- Open this HTML file as http://localhost/custom_controls.html and the output is displayed as shown below.
- The example provides custom controls to swipe between any slides of a page.



Framework7 – Lazy Loading

Description

The lazy loading, when used for large size multimedia files, takes time to load such as pictures, videos etc.

Example

The following example demonstrates the use of lazy loading, which loads the images by triggering 'lazy' event in Framework7:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">

```



```

    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Lazy Loading</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
    <div class="views">
        <div class="view view-main">
            <div class="pages">
                <div data-page="home" class="page navbar-fixed">
                    <div class="navbar">
                        <div class="navbar-inner">
                            <div class="left"> </div>
                            <div class="center">Lazy Loading</div>
                            <div class="right"> </div>
                        </div>
                    </div>
                    <div class="page-content">
                        <div class="swiper-container">
                            <div class="swiper-wrapper">
                                <div class="swiper-slide">
                                    
                                    <div class="preloader"></div>
                                </div>
                                <div class="swiper-slide">
                                    
                                    <div class="preloader"></div>
                                </div>
                                <div class="swiper-slide">
                                    
                                    <div class="preloader"></div>
                                </div>
                                <div class="swiper-slide">

```

```

        
        <div class="preloader"></div>
    </div>
    <div class="swiper-slide">
        
        <div class="preloader"></div>
    </div>
</div>
<div class="swiper-pagination color-white"></div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<style>.swiper-container{height:100%;}.swiper-
slide{background:#000;position:relative;height:100%;}.swiper-slide
img{position:absolute;left:50%;top:50%;max-width:100%;max-
height:100%;width:auto;height:auto;-webkit-transform:translate(-50%,-
50%);transform:translate(-50%,-50%);}</style>
<script>
    var myApp = new Framework7();
    var mySwiper = new Swiper('.swiper-container', {
        preloadImages: false,
        lazyLoading: true,
        pagination: '.swiper-pagination'
    })
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **lazy_loading.html** file in your server root folder.
- Open this HTML file as http://localhost/lazy_loading.html and output as below gets displayed.
- The example demonstrates the lazy loading when you are using the large size multimedia files such as images, videos etc.



22. Framework7 – Photo Browser

Description

The Photo browser is similar to iOS photo browser component to display group of images, which can be zoomed and panned. To slide between images, photo browser uses [Swiper Slider](#).

The following table shows the photo browser types used in framework7:

S. No.	Photo browser types & Description
1	Create Photo Browser Instance Photo browser can be created and initialized using JavaScript only.
2	Photo Browser Parameters Framework7 provides a list of parameters, which are used with photo browser.
3	Photo Browser Methods & Properties You will get an initialized instance variable to use photo browser methods and properties once you initialize photo browser.
4	Photos Array During initializing photo browser, you need to pass array with photos/videos in <i>photos</i> parameter.
5	Photo Browser Templates Framework7 provides you many photo browser templates, which you can pass on photo browser initialization.

Framework7 – Create Photo Browser Instance

Description

Photo browser can be created and initialized using JavaScript only. You need to use related the App's method as shown below:

- **myApp.photoBrowser(parameters)** - It is used to initialize photo browser with parameters and it also returns initialized photo browser instance.
 - **parameters** - It is a required parameter, which is an *object* with photo browser parameters.

The following code shows how to use the above *photoBrowser()* method:

```
var myPhotoBrowser = app.photoBrowser({
  zoom: 400,
  photos: ['image1.jpg', 'image2.jpg']
});
```

```
});  
myPhotoBrowser.open(); // It opens photo browser
```

Framework7 – Photo Browser Parameters

Description

Framework7 provides a list of parameters, which are used with photo browser and are listed in the following table:

S. No.	Parameter & Description	Type	Default
1	photos It is an array with URLs of images or an array of objects with URL and caption properties.	array	[]
2	initialSlide It is an initial slide's index number.	number	0
3	spaceBetween It represents the distance between photo slides in pixels.	number	20
4	speed It represents the time duration of transition between photo slides.	number	300
5	zoom It is used to enable/disable ability to zoom and pan photos.	boolean	true
6	maxZoom It is the maximum zoom ratio.	number	3
7	minZoom It is the minimum zoom ratio.	number	1
8	exposition It enables/disables the exposition mode when clicking on photo browser.	boolean	true
9	expositionHideCaptions If this is set to true then captions in exposition mode is hidden.	boolean	false
10	swipeToClose If this is enabled, you can close the photo browser with swipe up/down.	boolean	true
11	view It is the link to initialized view instance.	View instance	undefined

12	type It defines how the photo browser should be opened.	string	standalone
13	loop When set to <i>true</i> , it enables continuous loop mode.	boolean	false
14	theme The photo browser color theme can be <i>light</i> or <i>dark</i> .	string	light
15	captionsTheme It can be of two color <i>dark</i> or <i>light</i> .	string	-
16	navbar It is set to <i>false</i> to remove the photo browser's navbar.	boolean	true
17	toolbar It is set to <i>false</i> to remove the photo browser's toolbar.	boolean	true
18	backLinkText It is the text on the link at the back in navbar.	string	close
19	ofText It is the text of in photos counter. For eg 3 of 5.	string	of
Lazy Loading			
1	lazyLoading It is set to <i>true</i> to enable lazy loading of images.	boolean	false
2	lazyLoadingInPrevNext It is set to <i>true</i> to enable lazy loading for the closest photos.	boolean	false
3	lazyLoadingOnTransitionStart After the transition to photo, the photo browser will load lazy photos by default. If you enable this parameter, you can load new photo in the beginning of transition.	boolean	false
Templates			
1	navbarTemplate It is the navbar HTML template.	string	-
2	toolbarTemplate It is the toolbar HTML template.	string	-
3	photoTemplate It is the single photo element HTML template.	string	-
4	lazyPhotoTemplate It is the single lazy loading photo element HTML template.	string	-

5	objectTemplate It is the single object element HTML template.	string	-
6	captionTemplate It is the single caption element HTML template.	string	-
Callbacks			
1	onOpen(photobrowser) It is the callback function, which will be executed when photo browser is opened.	function	-
2	onClose(photobrowser) It is the callback function, which will be executed when user closes the photo browser.	function	-
3	onSwipeToClose(photobrowser) It is the callback function, which will be executed when user closes the photo browser by swiping up or down.	function	-
Swiper Callbacks			
1	onSlideChangeStart(swiper) This callback function will be executed in the beginning of animation to other slide. It takes swiper instance as an argument.	function	-
2	onSlideChangeEnd(swiper) This callback function will be executed after animation to other slide. It takes swiper instance as an argument.	function	-
3	onTransitionStart(swiper) This callback function will be executed at the beginning of transition. It takes swiper instance as an argument.	function	-
4	onTransitionEnd(swiper) This callback function will be executed after the transition. It takes swiper instance as an argument.	function	-
5	onClick(swiper,event) This callback function will run when the user clicks on slider after 300ms delay. It accepts swiper instance and event as a parameter.	function	-
6	onDoubleTap(swiper, event) This callback function will run when the user double clicks on slider container.	function	-
7	onLazyImageLoad(swiper, slide, image) This callback function will be executed when lazy load begins loading photo.	function	-

8	onLazyImageReady(swiper, slide, image) This function will be executed after lazy loading photo will be loaded.	function	-
---	--	----------	---

Framework7 – Photo Browser Methods & Properties

Description

You will get an initialized instance variable to use the photo browser methods and properties once you initialize the photo browser.

Photo Browser Properties

The following table shows the photo browser properties used in Framework7:

S. No.	Properties & Description
1	myPhotoBrowser.swiper It holds initialized swiper instances with every available swiper methods and properties.
2	myPhotoBrowser.container It is the DOM7 element with photo browser container HTML element.
3	myPhotoBrowser.exposed It contains <i>true</i> , if photo browser is in exposition mode.
4	myPhotoBrowser.activeIndex It is the index number of the currently active slide.
5	myPhotoBrowser.params It is the object with passed initialization parameters.

Photo Browser Methods

The following table shows the photo browser methods used in Framework7:

S. No.	Methods & Description
1	myPhotoBrowser.open(index); It is used to open the photo browser on photo with index number. If index is not specified then, last closed photo will be opened.
2	myPhotoBrowser.close(); It is used to close the photo browser.
3	myPhotoBrowser.toggleZoom(); It toggles the zoom of currently active slide.
4	myPhotoBrowser.toggleExposition(); It is used to toggle the exposition mode.

5	myPhotoBrowser.enableExposition(); It is used to enable the exposition mode.
6	myPhotoBrowser.disableExposition(); It is used when you want to disable the exposition mode.

Framework7 – Photos Array

Description

During initialization of the photo browser, you need to pass array with photos/videos in *photos* parameter.

Example

The following example demonstrates the use of photo browser in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Photos Array</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <body>
      <div class="views">
        <div class="view view-main">
          <div class="navbar">
            <div class="navbar-inner">
              <div class="left"> </div>
              <div class="center sliding">Photo Browser</div>
              <div class="right"> </div>
            </div>
          </div>
          <div class="pages navbar-through">
```

```

    <div data-page="home" class="page">
      <div class="page-content">
        <div class="content-block-title">Light Theme</div>
        <div class="content-block row">
          <div class="col-33"><a href="#" class="button color-
pink button-fill pb-standalone">Standalone</a></div>
          <div class="col-33"><a href="#" class="button color-
green pb-popup">Popup</a></div>
          <div class="col-33"><a href="#" class="button color-
blue button-fill pb-page">Page</a></div>
        </div>
        <div class="content-block-title">Dark Theme</div>
        <div class="content-block row">
          <div class="col-33"><a href="#" class="button color-
blue button-fill pb-standalone-dark">Standalone</a></div>
          <div class="col-33"><a href="#" class="button color-
pink button-fill pb-popup-dark">Popup</a></div>
          <div class="col-33"><a href="#" class="button color-
green pb-standalone-captions">Captions</a></div>
        </div>
        <div class="content-block"><a href="#" class="button
color-orange button-round pb-standalone-video">With Video</a></div>
      </div>
    </div>
  </div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
  var myApp = new Framework7();
  var $$ = Dom7;
  var mainView = myApp.addView('.view-main', {
    dynamicNavbar: true
  });
  /* Default standalone */
  var myPhotoBrowserStandalone = myApp.photoBrowser({
    photos : [
      '/framework7/images/background.jpg',
      '/framework7/images/birds.jpg',

```

```

        '/framework7/images/nature.jpg',
    ]
});
// Open the photo browser on click
$$('.pb-standalone').on('click', function () {
    myPhotoBrowserStandalone.open();
});
/* Popup */
var myPhotoBrowserPopup = myApp.photoBrowser({
    photos : [
        '/framework7/images/background.jpg',
        '/framework7/images/birds.jpg',
        '/framework7/images/nature.jpg',
    ],
    type: 'popup'
});
$$('.pb-popup').on('click', function () {
    myPhotoBrowserPopup.open();
});
/* As Page */
var myPhotoBrowserPage = myApp.photoBrowser({
    photos : [
        '/framework7/images/background.jpg',
        '/framework7/images/birds.jpg',
        '/framework7/images/nature.jpg',
    ],
    type: 'page',
    backLinkText: 'Back'
});
$$('.pb-page').on('click', function () {
    myPhotoBrowserPage.open();
});
/* Standalone Dark */
var myPhotoBrowserDark = myApp.photoBrowser({
    photos : [
        '/framework7/images/background.jpg',
        '/framework7/images/birds.jpg',
        '/framework7/images/nature.jpg',

```

```

    ],
    theme: 'dark'
  });
  $$('.pb-standalone-dark').on('click', function () {
    myPhotoBrowserDark.open();
  });
  /* Popup Dark */
  var myPhotoBrowserPopupDark = myApp.photoBrowser({
    photos : [
      '/framework7/images/background.jpg',
      '/framework7/images/birds.jpg',
      '/framework7/images/nature.jpg',
    ],
    theme: 'dark',
    type: 'popup'
  });
  $$('.pb-popup-dark').on('click', function () {
    myPhotoBrowserPopupDark.open();
  });
  /* With Captions */
  var myPhotoBrowserCaptions = myApp.photoBrowser({
    photos : [
      {
        url: '/framework7/images/background.jpg',
        caption: 'This is caption 1 text'
      },
      {
        url: '/framework7/images/birds.jpg',
        caption: 'This is second caption text'
      },
      // This one without caption
      {
        url: '/framework7/images/nature.jpg',
      },
    ],
    theme: 'dark',
    type: 'standalone'
  });

```

```

    $$('.pb-standalone-captions').on('click', function () {
        myPhotoBrowserCaptions.open();
    });

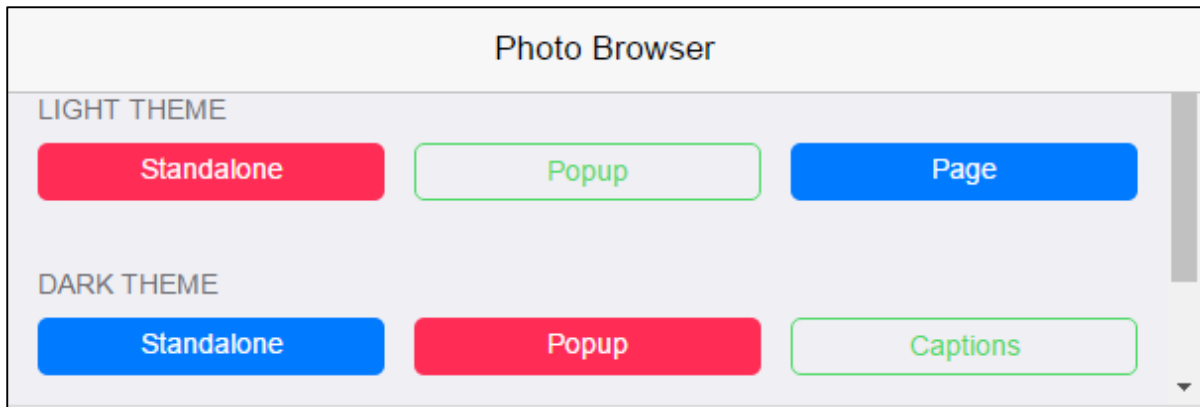
    /* With Video */
    var myPhotoBrowserVideo = myApp.photoBrowser({
        photos : [
            {
                html: '<iframe src="/framework7/images/video.mp4"
frameborder="0" allowfullscreen></iframe>',
                caption: 'My Video'
            },
            {
                url: '/framework7/images/birds.jpg',
                caption: 'Second Caption Text'
            },
            {
                url: '/framework7/images/nature.jpg',
            },
        ],
        theme: 'dark',
        type: 'standalone'
    });
    $$('.pb-standalone-video').on('click', function () {
        myPhotoBrowserVideo.open();
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **photo_browser_photos_array.html** file in your server root folder.
- Open this HTML file as http://localhost/photo_browser_photos_array.html and the output is displayed as shown below.
- The example gives the photo browser array, which takes array of photos or videos from the *photos* parameter.



Framework7 – Photo Browser Templates

Description

Framework7 provides you many photo browser templates, which you can pass on to the photo browser initialization.

Navbar Template

The following code shows an example of navbar template, which you can pass in *navbarTemplate* parameter:

```
<div class="navbar">
  <div class="navbar-inner">
    <div class="left sliding">
      <a href="#" class="link close-popup photo-browser-close-link
{{#unless backLinkText}}icon-only{{/unless}} {{js "this.type === \'page\' ?
\'back\' : \'\'"}}">
        <i class="icon icon-back {{iconsColorClass}}"></i>
        {{#if backLinkText}}<span>{{backLinkText}}</span>{{/if}}
      </a>
    </div>
    <div class="center sliding">
      <span class="photo-browser-current"></span>
      <span class="photo-browser-of">{{ofText}}</span>
      <span class="photo-browser-total"></span>
    </div>
    <div class="right"></div>
  </div>
</div>
```

The above template uses classes, which are listed below:

- **** - This link will close the photo browser on click. Not only does it close the popup or the page, but it also detaches all events listeners.
- **** - The photo browser will put the index number currently active slide into element with class *photo-browser-current*.
- **** - The photo browser will put the total number of images into the element with class *photo-browser-total*.

Toolbar Template

The following code shows an example of the toolbar template, which you can pass in *toolbarTemplate* parameter:

```
<div class="toolbar tabbar">
  <div class="toolbar-inner">
    <a href="#" class="link photo-browser-prev">
      <i class="icon icon-prev {{iconsColorClass}}"></i>
    </a>
    <a href="#" class="link photo-browser-next">
      <i class="icon icon-next {{iconsColorClass}}"></i>
    </a>
  </div>
</div>
```

The above given template uses classes which are listed below:

- **** - It works like the Next button.
- **** - It works like the Previous button.

Photo Element Template

The following code shows the template example of single photo slide element, which you can pass in *photoTemplate* parameter:

```
<div class="photo-browser-slide swiper-slide">
  <span class="photo-browser-zoom-container">
    
  </span>
</div>
```

Lazy Photo Element Template

The following code shows the template example of single photo slide element, which you can pass in *lazyPhotoTemplate* parameter:

```

<div class="photo-browser-slide photo-browser-slide-lazy swiper-slide">
  <div class="preloader {{@root.preloaderColorClass}}">{{#if
@root.material}}{{@root.materialPreloaderSvg}}{}/if}}</div>
  <span class="photo-browser-zoom-container">
    
  </span>
</div>

```

Object Element Template

The following code shows the template example of single photo slide element, which you can pass in *objectTemplate* parameter:

```

<div class="photo-browser-slide photo-browser-object-slide swiper-slide">{{js
"this.html || this"}}</div>

```

Single Caption Template

The following code shows the template example of single caption element, which you can pass in *captionTemplate* parameter:

```

<div class="photo-browser-caption" data-caption-index="{{@index}}">
  {{caption}}
</div>

```


23. Framework7 – Autocomplete

Description

Autocomplete is a Framework7's mobile friendly and touch optimized component, which can be as dropdown or in standalone way. You can create and initialize Autocomplete instance by using the JavaScript method:

```
myApp.autocomplete(parameters)
```

Where *parameters* are required objects used to initialize the Autocomplete instance.

Autocomplete Parameters

The following table lists the available Autocomplete parameters in Framework7:

S. No.	Parameters & Description	Type	Default
1	openIn It defines how to open an Autocomplete which could be used as dropdown, popup or page.	string	page
2	source It uses autocomplete instance, search query and renders function to pass matched items with array.	function (autocomplete, query, render)	-
3	valueProperty It specifies the item value of matched item object's key.	string	id
4	limit It displays the limited number of items in autocomplete per query.	number	-
5	preloader Preloader can be used to specify autocomplete layout by setting it to true.	boolean	false
6	preloaderColor It specifies the preloader color. By default, the color is "black".	string	-
7	value Defines the array with default selected values.	array	-
8	textProperty It specifies the item value of matched item	string	text

	object's key, which can be used as a title of displayed options.		
--	--	--	--

Standalone Autocomplete Parameters

The following table lists the available Standalone Autocomplete parameters in Framework7:

S. No.	Parameters & Description	Type	Default
1	opener It is string or HTML element parameter, which will open the standalone autocomplete page.	string or HTML element	-
2	popupCloseText It is used to close the autocomplete popup.	string	'Close'
3	backText It provides the back link when autocomplete is opened as page.	string	'Back'
4	pageTitle It specifies the autocomplete page title.	string	-
5	searchbarPlaceholderText It specifies the search bar placeholder text.	string	'Search'
6	searchbarCancelText It defines the search bar cancel button text.	string	'cancel'
7	notFoundText It displays the text when there is no matched element found.	string	'Nothing found'
8	multiple It allows to select multiple selection by setting it to true.	boolean	false
9	navbarTheme It specifies the color theme for navbar.	string	-
10	backOnSelect When the user picks value, the autocomplete will be closed by setting it to true.	boolean	false
11	formTheme It specifies the color theme for form.	string	-

Dropdown Autocomplete Parameters

The following table lists the available Dropdown Autocomplete parameters in Framework7:

S. No.	Parameters & Description	Type	Default
1	input It is string or HTML element used for text input.	string or HTML Element	-
2	dropdownPlaceholderText It specifies the dropdown placeholder text.	string	-
3	updateInputValueOnSelect You can update the input value on select by setting it to true.	boolean	true
4	expandInput You can expand the text input in List View to make full screen wide during dropdown visible by setting <i>item-input</i> it to true.	boolean	false

Autocomplete Callbacks Functions

The below table lists available Dropdown Autocomplete parameters in Framework7:

S. No.	Parameters & Description	Type	Default
1	onChange Whenever the autocomplete value is changed, this callback function will be executed.	function (autocomplete, value)	-
2	onOpen Whenever autocomplete is opened, this callback function will be executed.	function (autocomplete)	-
3	onClose Whenever autocomplete is closed, this callback function will be executed.	function (autocomplete)	-

Autocomplete Templates

The following table lists the available Dropdown Autocomplete parameters in Framework7:

S. No.	Parameters & Description	Type	Default
1	navbarTemplate It is standalone autocomplete navbar template.	string	-
2	itemTemplate It is standalone template7 form item.	string	-

3	dropdownTemplate It is template7 dropdown template.	string	-
4	dropdownItemTemplate It is template7 dropdown list item.	string	-
5	dropdownPlaceholderTemplate It is template7 dropdown placeholder item.	string	-

Default templates

Following are the default template code snippets for the above defined templates parameters:

navbarTemplate

```
<div class="navbar {{#if navbarTheme}}theme-{{navbarTheme}}{/if}">
  <div class="navbar-inner">
    <div class="left sliding">
      {{#if material}}
      <a href="#" class="link {{#if inPopup}}close-
popup{{else}}back{/if}} icon-only">
        <i class="icon icon-back"></i>
      </a>
      {{else}}
      <a href="#" class="link {{#if inPopup}}close-
popup{{else}}back{/if}">
        <i class="icon icon-back"></i>
        {{#if inPopup}}
        <span>{{popupCloseText}}</span>
        {{else}}
        <span>{{backText}}</span>
        {{/if}}
      </a>
      {{/if}}
    </div>
    <div class="center sliding">{{pageTitle}}</div>
    {{#if preloader}}
    <div class="right">
      <div class="autocomplete-preloader preloader {{#if
preloaderColor}}preloader-{{preloaderColor}}{/if}"></div>
    </div>
    {{/if}}
  </div>
```

```

    </div>
</div>

```

itemTemplate

```

<li>
  <label class="label-{{inputType}} item-content">
    <input type="{{inputType}}" name="{{inputName}}" value="{{value}}"
    {{#if selected}}checked{{/if}}>
    {{#if material}}
      <div class="item-media">
        <i class="icon icon-form-{{inputType}}"></i>
      </div>
      <div class="item-inner">
        <div class="item-title">{{text}}</div>
      </div>
    {{else}}
      {{#if checkbox}}
      <div class="item-media">
        <i class="icon icon-form-checkbox"></i>
      </div>
      {{/if}}
      <div class="item-inner">
        <div class="item-title">{{text}}</div>
      </div>
    {{/if}}
  </label>
</li>

```

dropdownTemplate

```

<div class="autocomplete-dropdown">
  <div class="autocomplete-dropdown-inner">
    <div class="list-block">
      <ul></ul>
    </div>
  </div>
  {{#if preloader}}
  <div class="autocomplete-preloader preloader {{#if
  preloaderColor}}preloader-{{preloaderColor}}>

```

```

    {{#if material}}{{materialPreloaderHtml}}{{/if}}
  </div>
  {{/if}}
</div>

```

dropdownItemTemplate

```

<li>
  <label class="{{#unless placeholder}}label-radio{{/unless}} item-content"
  data-value="{{value}}">
    <div class="item-inner">
      <div class="item-title">{{text}}</div>
    </div>
  </label>
</li>

```

dropdownPlaceholderTemplate

```

<li class="autocomplete-dropdown-placeholder">
  <div class="item-content">
    <div class="item-inner">
      <div class="item-title">{{text}}</div>
    </div>
  </label>
</li>

```

Autocomplete Methods

The following table specifies Autocomplete methods available in Framework7:

S. No.	Methods & Description
1	myAutocomplete.params Defines the initialization parameters that are passes with object.
2	myAutocomplete.value It defines the array with selected values.
3	myAutocomplete.opened It opens the Autocomplete if it is set to true.
4	myAutocomplete.dropdown It specifies an instance of Autocomplete dropdown.

5	myAutocomplete.popup It specifies an instance of Autocomplete popup.
6	myAutocomplete.page It specifies an instance of Autocomplete page.
7	myAutocomplete.pageData It defines Autocomplete page data.
8	myAutocomplete.searchbar It defines Autocomplete searchbar instance.

Autocomplete Properties

The following table specifies Autocomplete methods available in Framework7:

S. No.	Properties & Description
1	myAutocomplete.open() It opens the Autocomplete, which could be used as dropdown, popup or page.
2	myAutocomplete.close() It closes the Autocomplete.
3	myAutocomplete.showPreloader() It shows the Autocomplete preloader.
4	myAutocomplete.hidePreloader() It hides the Autocomplete preloader.
5	myAutocomplete.destroy() It ruins the Autocomplete preloader instance and removes all events.

Example

The following example demonstrates the use of autocomplete parameters hiding in the Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Autocomplete</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ios.colors.min.css">
</head>
<body>
  <div class="views">
    <div class="view view-main">
      <div class="pages">
        <div data-page="home" class="page navbar-fixed">
          <div class="navbar">
            <div class="navbar-inner">
              <div class="left"> </div>
              <div class="center">Autocomplete</div>
              <div class="right"> </div>
            </div>
          </div>
          <div class="page-content">
            <div class="content-block-title">Simple Dropdown
Autocomplete</div>
            <div class="list-block">
              <ul>
                <li class="item-content">
                  <div class="item-title label">Country</div>
                  <div class="item-input">
                    <input type="text" placeholder="Country"
id="autocomplete-dropdown">
                  </div>
                </li>
              </ul>
            </div>
            <div class="content-block-title">Dropdown With Input
Expand</div>
            <div class="list-block">
              <ul>
                <li class="item-content">
                  <div class="item-title label">Country</div>
                  <div class="item-input">
                    <input type="text" placeholder="Country"
id="autocomplete-dropdown-expand">
                  </div>
                </li>
              </ul>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```



```

        </li>
    </ul>
</div>
<div class="content-block-title">Dropdown With All
Values</div>
<div class="list-block">
    <ul>
        <li class="item-content">
            <div class="item-title label">Country</div>
            <div class="item-input">
                <input type="text" placeholder="Country"
id="autocomplete-dropdown-all">
            </div>
        </li>
    </ul>
</div>
<div class="content-block-title">Dropdown With
Placeholder</div>
<div class="list-block">
    <ul>
        <li class="item-content">
            <div class="item-title label">Country</div>
            <div class="item-input">
                <input type="text" placeholder="Country"
id="autocomplete-dropdown-placeholder">
            </div>
        </li>
    </ul>
</div>
<div class="content-block-title">Simple Standalone
Autocomplete</div>
<div class="list-block">
    <ul>
        <li>
            <a href="#" id="autocomplete-standalone"
class="item-link item-content autocomplete-opener">
                <input type="hidden">
                <div class="item-inner">
                    <div class="item-title">Favorite Country</div>
                    <div class="item-after"></div>

```

```

        </div>
    </a>
</li>
</ul>
</div>
<div class="content-block-title">Popup Standalone
Autocomplete</div>
<div class="list-block">
    <ul>
        <li>
            <a href="#" id="autocomplete-standalone-popup"
class="item-link item-content autocomplete-opener">
                <input type="hidden">
                <div class="item-inner">
                    <div class="item-title">Favorite Country</div>
                    <div class="item-after"></div>
                </div>
            </a>
        </li>
    </ul>
</div>
<div class="content-block-title">Multiple Values Standalone
Autocomplete</div>
<div class="list-block">
    <ul>
        <li>
            <a href="#" id="autocomplete-standalone-multiple"
class="item-link item-content autocomplete-opener">
                <input type="hidden">
                <div class="item-inner">
                    <div class="item-title">Favorite
Countries</div>
                    <div class="item-after"></div>
                </div>
            </a>
        </li>
    </ul>
</div>
</div>
</div>

```

```

    </div>
  </div>
  <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
  <script>
    var myApp = new Framework7();
    var $$ = Dom7;
    var mainView = myApp.addView('.view-main');
    // Countries data array
    var countries = ('India Africa Australia NewZealand England
WestIndies Scotland Zimbabwe Srilanka Bangladesh').split(' ');
    // Simple Dropdown
    var autocompleteDropdownSimple = myApp.autocomplete({
      input: '#autocomplete-dropdown',
      openIn: 'dropdown',
      source: function (autocomplete, query, render) {
        var results = [];
        if (query.length === 0) {
          render(results);
          return;
        }
        // You can find matched items
        for (var i = 0; i < countries.length; i++) {
          if
(countries[i].toLowerCase().indexOf(query.toLowerCase()) >= 0)
results.push(countries[i]);
        }
        // Display the items by passing array with result items
        render(results);
      }
    });
    // Dropdown with input expand
    var autocompleteDropdownExpand = myApp.autocomplete({
      input: '#autocomplete-dropdown-expand',
      openIn: 'dropdown',
      expandInput: true, // expandInput used as item-input in List
View will be expanded to full screen wide //during dropdown
      source: function (autocomplete, query, render) {
        var results = [];

```

```

        if (query.length === 0) {
            render(results);
            return;
        }
        // Find matched items
        for (var i = 0; i < countries.length; i++) {
            if
(countries[i].toLowerCase().indexOf(query.toLowerCase()) >= 0)
results.push(countries[i]);
        }
        // Display the items by passing array with result items
        render(results);
    }
});
// Dropdown with all values
var autocompleteDropdownAll = myApp.autocomplete({
    input: '#autocomplete-dropdown-all',
    openIn: 'dropdown',
    source: function (autocomplete, query, render) {
        var results = [];
        // You can find matched items
        for (var i = 0; i < countries.length; i++) {
            if
(countries[i].toLowerCase().indexOf(query.toLowerCase()) >= 0)
results.push(countries[i]);
        }
        // Display the items by passing array with result items
        render(results);
    }
});
// Dropdown with placeholder
var autocompleteDropdownPlaceholder = myApp.autocomplete({
    input: '#autocomplete-dropdown-placeholder',
    openIn: 'dropdown',
    dropdownPlaceholderText: 'Type as "India"',
    source: function (autocomplete, query, render) {
        var results = [];
        if (query.length === 0) {
            render(results);
        }
    }
});

```

```

        return;
    }
    // You can find matched items
    for (var i = 0; i < countries.length; i++) {
        if
(countries[i].toLowerCase().indexOf(query.toLowerCase()) >= 0)
results.push(countries[i]);
    }
    // Display the items by passing array with result items
    render(results);
}
});
// Simple Standalone
var autocompleteStandaloneSimple = myApp.autocomplete({
    openIn: 'page', //open in page
    opener: $('#autocomplete-standalone'), //link that opens
autocomplete
    backOnSelect: true, //go back after we select something
    source: function (autocomplete, query, render) {
        var results = [];
        if (query.length === 0) {
            render(results);
            return;
        }
        // You can find matched items
        for (var i = 0; i < countries.length; i++) {
            if
(countries[i].toLowerCase().indexOf(query.toLowerCase()) >= 0)
results.push(countries[i]);
        }
        // Display the items by passing array with result items
        render(results);
    },
    onChange: function (autocomplete, value) {
        // Here add the item text value to item-after
        $('#autocomplete-standalone').find('.item-
after').text(value[0]);
        // You can add item value to input value
        $('#autocomplete-standalone').find('input').val(value[0]);
    }
}

```

```

    });
    // Standalone Popup
    var autocompleteStandalonePopup = myApp.autocomplete({
        openIn: 'popup', // Opens the Autocomplete in page
        opener: $('#autocomplete-standalone-popup'), // It will open
standalone autocomplete popup
        backOnSelect: true, //After selecting item, then go back to
page
        source: function (autocomplete, query, render) {
            var results = [];
            if (query.length === 0) {
                render(results);
                return;
            }
            // You can find matched items
            for (var i = 0; i < countries.length; i++) {
                if
(countries[i].toLowerCase().indexOf(query.toLowerCase()) >= 0)
results.push(countries[i]);
            }
            // Display the items by passing array with result items
            render(results);
        },
        onChange: function (autocomplete, value) {
            // Here add the item text value to item-after
            $('#autocomplete-standalone-popup').find('.item-
after').text(value[0]);
            // You can add item value to input value
            $('#autocomplete-standalone-
popup').find('input').val(value[0]);
        }
    });
    // Multiple Standalone
    var autocompleteStandaloneMultiple = myApp.autocomplete({
        openIn: 'page', //Opens the Autocomplete in page
        opener: $('#autocomplete-standalone-multiple'), //link that
opens autocomplete
        multiple: true, //Allow multiple values
        source: function (autocomplete, query, render) {
            var results = [];
            if (query.length === 0) {

```

```

        render(results);
        return;
    }
    // You can find matched items
    for (var i = 0; i < countries.length; i++) {
        if
(countries[i].toLowerCase().indexOf(query.toLowerCase()) >= 0)
results.push(countries[i]);
    }
    // Display the items by passing array with result items
    render(results);
},
onChange: function (autocomplete, value) {
    // Here add the item text value to item-after
    $('#autocomplete-standalone-multiple').find('.item-
after').text(value.join(', '));
    // You can add item value to input value
    $('#autocomplete-standalone-
multiple').find('input').val(value.join(', '));
}
});
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **autocomplete.html** file in your server root folder.
- Open this HTML file as <http://localhost/autocomplete.html> and the output is displayed as shown below.
- The example provides autocomplete of values in simple dropdown, dropdown with all values, dropdown with placeholder, standalone autocomplete etc.

Autocomplete	
SIMPLE DROPDOWN AUTOCOMPLETE	
Country	Country
DROPDOWN WITH INPUT EXPAND	
Country	Country
DROPDOWN WITH ALL VALUES	
Country	Country
DROPDOWN WITH PLACEHOLDER	
Country	Country
SIMPLE STANDALONE AUTOCOMPLETE	
Favorite Country	>
POPUK STANDALONE AUTOCOMPLETE	
Favorite Country	>
MULTIPLE VALUES STANDALONE AUTOCOMPLETE	
Favorite Countries	>

24. Framework7 – Picker

Description

Picker looks like iOS native picker and it is a powerful component that allows you to pick any values from list and also used to create custom overlay pickers. You can use Picker as inline component or as overlay. The overlay picker will be automatically converted to Popover on tablets (iPad).

Using the following App's method, you can create and initialize the JavaScript method:

```
myApp.picker(parameters)
```

Where *parameters* are required objects, used to initialize the picker instance and it is a required method.

Picker Parameters

The following table specifies the available parameters in the pickers:

S. No.	Parameters & Description	Type	Default
1	container String with CSS selector or HTML element used to generate Picker HTML for an inline pickers.	string or HTML element	-
2	input The related input element placed with the string with CSS selector or HTML element.	string or HTML element	-
3	scrollToInput It is used to scroll viewport (page-content) of input, whenever picker is opened.	boolean	true
4	inputReadOnly Used to set the "readonly" attribute on specified input.	boolean	true
5	convertToPopover It is used to convert the picker modal to Popover on large screens like iPad.	boolean	true
6	onlyOnPopover You can open the picker in Popover by enabling it.	boolean	true
7	cssClass To picker modal, you can use additional CSS class name.	string	-

8	closeByOutsideClick You can close the picker by clicking outside of picker or input element by enabling the method.	boolean	false
9	toolbar It is used to enable the picker modal toolbar.	boolean	true
10	toolbarCloseText Used for Done/Close toolbar button.	string	'Done'
11	toolbarTemplate It is toolbar HTML Template, by default it is HTML string with the following template: <pre><div class="toolbar"> <div class="toolbar-inner"> <div class="left"></div> <div class="right"> {{closeText}} </div> </div> </div></pre>	string	-

Specific Picker Parameters

The following table lists the available specific picker parameters:

S. No.	Parameters & Description	Type	Default
1	rotateEffect It enables 3D rotate effect in picker.	boolean	false
2	momentumRatio When you release picker after fast touch and move, it produces more momentum.	number	7
3	updateValuesOnMomentum Used to update pickers and input values during momentum.	boolean	false
4	updateValuesOnTouchmove Used to update pickers and input values during touch move.	boolean	true
5	value The array has its initial values and also each array item represents the value of related column.	array	-

6	formatValue The function is used to format the input value, and it should return new/formatted string value. The <i>values</i> and <i>displayValues</i> are arrays of related column.	function (p, values, displayValues)	-
7	cols Every array item represents an object with column parameters.	array	-

Picker Parameter Callbacks

The following table shows the lists of callback functions available in pickers:

S. No.	Callbacks & Description	Type	Default
1	onChange The callback function will be executed whenever the picker value changed and the <i>values</i> and <i>displayValues</i> are arrays of related column.	function (p, values, displayValues)	-
2	onOpen The callback function will be executed whenever picker is opened.	function (p)	-
3	onClose The callback function will be executed whenever picker is closed.	function (p)	-

Column Parameters

At the moment of configuration of Picker, we need to pass *cols* parameter. It is represented as array, where each item is object with column parameters:

S. No.	Parameters & Description	Type	Default
1	values You can specify the string columns values with an array.	array	-
2	displayValues It has array with string columns values and it will display value from <i>values</i> parameter, when it not specified.	array	-
3	cssClass The <i>CSS class</i> name used to set on column HTML container.	string	-
4	textAlign It is used to set text alignment of column values and it could be " <i>left</i> ", " <i>center</i> " or " <i>right</i> ".	string	-

5	width Width is calculated automatically, by default. If you need to fix column widths in picker with dependent columns that should be in <i>px</i> .	number	-
6	divider It is used for column that should be visual divider, it doesn't have any values.	boolean	false
7	content It is used to specified divider-column (<i>divider:true</i>) with content of the column.	string	-

Column Callbacks Parameter

S. No.	Callbacks & Description	Type	Default
1	onChange When the column value will change, at that time callback function will execute. The <i>value</i> and <i>displayValue</i> represent the current column <i>value</i> and <i>displayValue</i> .	function (p, value, displayValue)	-

Picker Properties

The Picker variable has many properties those are given in the following table:

S. No.	Properties & Description
1	myPicker.params You can initialize passed parameters with object.
2	myPicker.value The selected value for each column is represented by an array of item.
3	myPicker.displayValue The selected display value for each column is represented by an array of item.
4	myPicker.opened When picker is opened, it sets to <i>true</i> value.
5	myPicker.inline When picker is inline, it sets to <i>true</i> value.
6	myPicker.cols The Picker columns has its own methods and properties.
7	myPicker.container The Dom7 instance is used for HTML container.

Picker Methods

The picker variable has helpful methods, which are given in the following table:

S. No.	Methods & Description
1	myPicker.setValue(values, duration) Use to set new picker value. The values are in array where each item represents <i>value</i> for each column. <i>duration</i> - It is transition duration in ms.
2	myPicker.open() Use to open Picker.
3	myPicker.close() Use to close Picker.
4	myPicker.destroy() Use to destroy Picker instance and remove all events.

Column Properties

Each column in *myPicker.cols* array also has its own useful properties, which are given in the following table:

```
//Get first column
var col = myPicker.cols[0];
```

S. No.	Properties & Description
1	col.container You can create an instance with column HTML container.
2	col.items You can create an instance with column items HTML elements.
3	col.value Used to select the current column value.
4	col.displayValue Used to select the current column value of display.
5	col.activeIndex Give the current index number, which is selected/active item.

Column Methods

The useful column methods are given in the following table:

S. No.	Methods & Description
1	col.setValue(value, duration) Used to set a new value for the current column. The <i>value</i> must be a new value, and <i>duration</i> is transition duration given in ms.
2	col.replaceValues(values, displayValues) Used to replace the column values and displayValues with new ones.

It is used to access Picker's instance from its HTML container, whenever you initialize Picker as inline Picker.

```
var myPicker = $$('.picker-inline')[0].f7Picker;
```

There are different types of Pickers as specified in the following table:

S. No.	Tabs Types & Description
1	Picker With Single Value It is a powerful component that allows you to pick any values from list.
2	Two Values and 3D-Rotate Effect In the picker you can use for 3D rotate effect.
3	Dependent Values Values are dependent on each other for specified element.
4	Custom Toolbar You can use one or more picker on single page for customize.
5	Inline Picker / Date-time You can select number of values in the inline picker. Like date has <i>date</i> , <i>month</i> , <i>year</i> and time has <i>hours</i> , <i>minutes</i> , <i>seconds</i> .

Framework7 – Picker With Single Value

Description

It is a powerful component that allows you to pick any values from the list.

Example

The following example demonstrates the use of *Picker with Single Value* in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
```

```

    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Picker with single value</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<div class="views">
  <div class="view view-main">
    <div class="pages">
      <div data-page="home" class="page navbar-fixed">
        <div class="navbar">
          <div class="navbar-inner">
            <div class="left"> </div>
            <div class="center">Picker</div>
            <div class="right"> </div>
          </div>
        </div>
        <div class="page-content">
          <div class="content-block-title">Picker with single
value</div>
          <div class="list-block">
            <ul>
              <li>
                <div class="item-content">
                  <div class="item-inner">
                    <div class="item-input">
                      <input type="text" placeholder="Your iOS
device" readonly id="picker-device">
                    </div>
                  </div>
                </div>
              </li>
            </ul>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

        </div>
    </div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<style>.swiper-slide{background:#fff;box-sizing:border-box;border:1px solid
#ccc;line-height:120px;text-align:center;}.swiper-slide span{font-
size:17px;}.swiper-container{height:120px;margin:0px 0 35px;}</style>
<style>#picker-date-container .picker-item{color:#999;}#picker-date-
container .picker-selected{color:#000;}@media (max-width: 767px) {#picker-date-
container .picker-items{font-size:21px;}#picker-date-container .picker-
item{height:36px;line-height:36px;padding:0 6px;}}</style>
<script>
    var myApp = new Framework7();

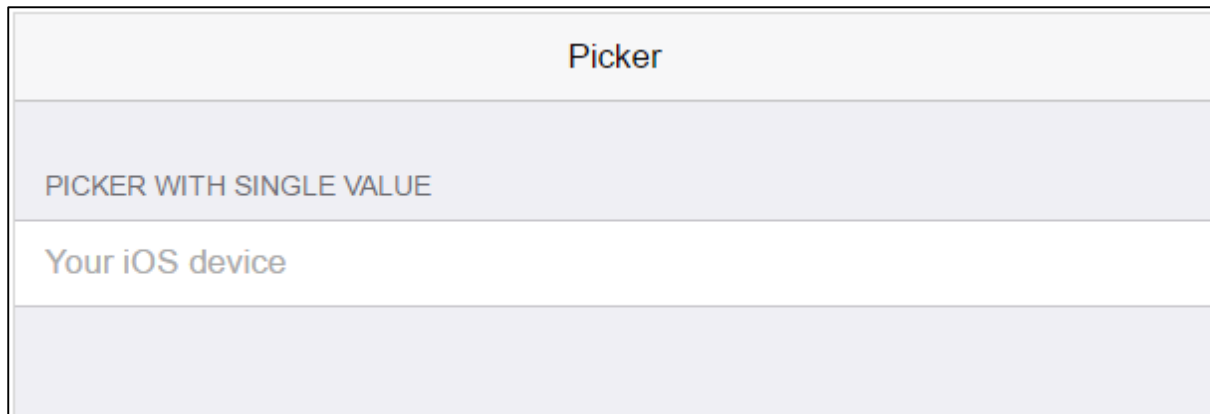
    var pickerDevice = myApp.picker({
        input: '#picker-device',
        cols: [
            {
                textAlign: 'center',
                values: ['iPhone 4', 'iPhone 4S', 'iPhone 5', 'iPhone 5S',
'iPhone 6', 'iPhone 6 Plus', 'iPad 2', 'iPad Retina', 'iPad Air', 'iPad mini',
'iPad mini 2', 'iPad mini 3']
            }
        ]
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **picker_with_single_value.html** file in your server root folder.
- Open this HTML file as http://localhost/picker_with_single_value.html and the output is displayed as shown below.
- The example allows you to pick up single value from the list.



Framework7 – Two Values and 3D-Rotate Effect

Description

This picker is the same as the picker; except, in this picker, you can use it for 3D rotate effect.

Example

The following example demonstrates the use of 2 values and 3d-rotate effect in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>2 values and 3d-rotate effect</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <div class="views">
    <div class="view view-main">
      <div class="pages">
        <div data-page="home" class="page navbar-fixed">
          <div class="navbar">
            <div class="navbar-inner">
```

```

        <div class="left"> </div>
        <div class="center">Picker</div>
        <div class="right"> </div>
    </div>
</div>
<div class="page-content">
    <div class="content-block-title">2 values and 3d-rotate
effect</div>
    <div class="list-block">
        <ul>
            <li>
                <div class="item-content">
                    <div class="item-inner">
                        <div class="item-input">
                            <input type="text" placeholder="Describe
yourself" readonly id="picker-describe">
                        </div>
                    </div>
                </div>
            </li>
        </ul>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<style>.swiper-slide{background:#fff;box-sizing:border-box;border:1px solid
#ccc;line-height:120px;text-align:center;}.swiper-slide span{font-
size:17px;}.swiper-container{height:120px;margin:0px 0 35px;}</style>
<style>#picker-date-container .picker-item{color:#999;}#picker-date-
container .picker-selected{color:#000;}@media (max-width: 767px) {#picker-date-
container .picker-items{font-size:21px;}#picker-date-container .picker-
item{height:36px;line-height:36px;padding:0 6px;}}</style>
<script>
    var myApp = new Framework7();
    var pickerDescribe = myApp.picker({
        input: '#picker-describe',

```

```

        rotateEffect: true,
        cols: [
            {
                textAlign: 'left',
                values: ('Super Lex Amazing Bat Iron Rocket Lex Cool
Beautiful Wonderful Raining Happy Amazing Funny Cool Hot').split(' ')
            },
            {
                values: ('Man Luthor Woman Boy Girl Person Cutie Babe
Raccoon').split(' ')
            },
        ]
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **two_values_3D_effect.html** file in your server root folder.
- Open this HTML file as http://localhost/two_values_3D_effect.html and the output is displayed as shown below.
- The example allows you to pick up two values from the list with 3D rotate effect.

Picker
2 VALUES AND 3D-ROTATE EFFECT
Describe yourself

Framework7 – Dependent Values

Description

Values are dependent on each other for specified element.

Example

The following example displays the dependent values of a particular element in Framework7:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Dependent values</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <div class="views">
    <div class="view view-main">
      <div class="pages">
        <div data-page="home" class="page navbar-fixed">
          <div class="navbar">
            <div class="navbar-inner">
              <div class="left"> </div>
              <div class="center">Picker</div>
              <div class="right"> </div>
            </div>
          </div>
          <div class="page-content">
            <div class="content-block-title">Dependent values</div>
            <div class="list-block">
              <ul>
                <li>
                  <div class="item-content">
                    <div class="item-inner">
                      <div class="item-input">
                        <input type="text" placeholder="Your car"
readonly id="picker-dependent">
                      </div>
                    </div>
                  </div>
                </li>
              </ul>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        </div>
    </div>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<style>.swiper-slide{background:#fff;box-sizing:border-box;border:1px solid
#ccc;line-height:120px;text-align:center;}.swiper-slide span{font-
size:17px;}.swiper-container{height:120px;margin:0px 0 35px;}</style>
<style>#picker-date-container .picker-item{color:#999;}#picker-date-
container .picker-selected{color:#000;}@media (max-width: 767px) {#picker-date-
container .picker-items{font-size:21px;}#picker-date-container .picker-
item{height:36px;line-height:36px;padding:0 6px;}}</style>
<script>
var myApp = new Framework7();
// Dependent values
var carVendors = {
    Japanese : ['Honda', 'Lexus', 'Mazda', 'Nissan', 'Toyota'],
    German : ['Audi', 'BMW', 'Mercedes', 'Volkswagen', 'Volvo'],
    American : ['Cadillac', 'Chrysler', 'Dodge', 'Ford']
};
var pickerDependent = myApp.picker({
    input: '#picker-dependent',
    rotateEffect: true,
    formatValue: function (picker, values) {
        return values[1];
    },
    cols: [
        {
            textAlign: 'left',
            values: ['Japanese', 'German', 'American'],
            onChange: function (picker, country) {
                if(picker.cols[1].replaceValues){
                    picker.cols[1].replaceValues(carVendors[country]);

```

```

        }
    },
    {
        values: carVendors.Japanese,
        width: 160,
    },
]
});
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **dependent_values.html** file in your server root folder.
- Open this HTML file as http://localhost/dependent_values.html and the output is displayed as shown below.
- The example allows you to select dependent values for the specified element. When you rotate the element, you will get the dependent value of it.

Picker
DEPENDENT VALUES
Your car

Framework7 – Custom Toolbar

Description

You can use one or more picker on a single page to customize.

Example

The following example allows you to select different values for different elements in Framework7:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Custom toolbar</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <div class="views">
    <div class="view view-main">
      <div class="pages">
        <div data-page="home" class="page navbar-fixed">
          <div class="navbar">
            <div class="navbar-inner">
              <div class="left"> </div>
              <div class="center">Picker</div>
              <div class="right"> </div>
            </div>
          </div>
          <div class="page-content">
            <div class="content-block-title">Custom toolbar</div>
            <div class="list-block">
              <ul>
                <li>
                  <div class="item-content">
                    <div class="item-inner">
                      <div class="item-input">
                        <input type="text" placeholder="Describe
yourself" readonly id="picker-custom-toolbar">
                      </div>
                    </div>
                  </div>
                </li>
              </ul>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        </div>
    </li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>

<style>.swiper-slide{background:#fff;box-sizing:border-box;border:1px solid
#ccc;line-height:120px;text-align:center;}.swiper-slide span{font-
size:17px;}.swiper-container{height:120px;margin:0px 0 35px;}</style>

<style>#picker-date-container .picker-item{color:#999;}#picker-date-
container .picker-selected{color:#000;}@media (max-width: 767px) {#picker-date-
container .picker-items{font-size:21px;}#picker-date-container .picker-
item{height:36px;line-height:36px;padding:0 6px;}}</style>

<script>
    var myApp = new Framework7();
    // Custom Toolbar
    var pickerCustomToolbar = myApp.picker({
        input: '#picker-custom-toolbar',
        rotateEffect: true,
        toolbarTemplate:
            '<div class="toolbar">' +
                '<div class="toolbar-inner">' +
                    '<div class="left">' +
                        '<a href="#" class="link toolbar-randomize-
link">Randomize</a>' +
                    '</div>' +
                    '<div class="right">' +
                        '<a href="#" class="link close-picker">That\'s me</a>' +
                    '</div>' +
                '</div>' +
            '</div>',
        cols: [
            {
                values: ['Mr', 'Ms'],
            },

```



```

        {
            textAlign: 'left',
            values: ('Super Lex Amazing Bat Iron Rocket Lex Cool
Beautiful Wonderful Raining Happy Amazing Funny Cool Hot').split(' ')
        },
        {
            values: ('Man Luthor Woman Boy Girl Person Cutie Babe
Raccoon').split(' ')
        },
    ],
    onOpen: function (picker) {
        picker.container.find('.toolbar-randomize-link').on('click',
function () {
            var col0Values = picker.cols[0].values;
            var col0Random = col0Values[Math.floor(Math.random() *
col0Values.length)];

            var col1Values = picker.cols[1].values;
            var col1Random = col1Values[Math.floor(Math.random() *
col1Values.length)];

            var col2Values = picker.cols[2].values;
            var col2Random = col2Values[Math.floor(Math.random() *
col2Values.length)];

            picker.setValue([col0Random, col1Random, col2Random]);
        });
    }
});
</script>
</body>
</html>

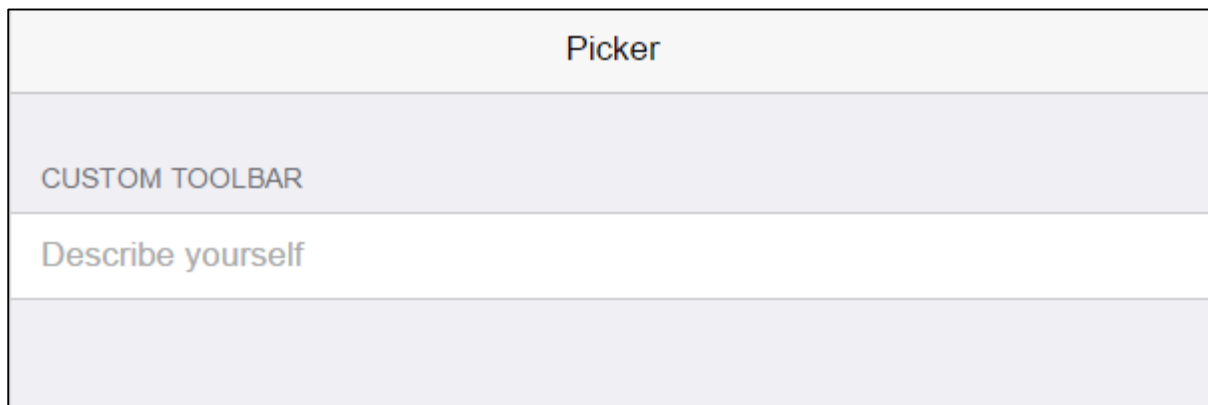
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **custom_toolbar.html** file in your server root folder.
- Open this HTML file as http://localhost/custom_toolbar.html and the output is displayed as shown below.

- The example allows you to select values from the list by using one or more picker to customize.



Framework7 – Inline Picker / Date-time

Description

You can select the number of values in the inline picker. Like date has *date*, *month*, *year* and time has *hours*, *minutes*, *seconds*.

Example

The following example demonstrates the use of *Inline Picker / Date-time* in Framework7:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Inline Picker / Date-time</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <div class="views">
    <div class="view view-main">
      <div class="pages">
        <div data-page="home" class="page navbar-fixed">
          <div class="navbar">
```

```

        <div class="navbar-inner">
            <div class="left"> </div>
            <div class="center">Picker</div>
            <div class="right"> </div>
        </div>
    </div>
    <div class="page-content">
        <div class="content-block-title">Inline Picker / Date-
time</div>
        <div class="content-block">
            <div style="padding:0; margin-right:-15px; width:auto"
class="content-block-inner">
                <div style="margin:0" class="list-block">
                    <ul style="border-top:none">
                        <li>
                            <div class="item-content">
                                <div class="item-inner">
                                    <div class="item-input">
                                        <input type="text" placeholder="Date
Time" readonly id="picker-date">
                                    </div>
                                </div>
                            </div>
                        </li>
                    </ul>
                </div>
                <div id="picker-date-container"></div>
            </div>
        </div>
    </div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<style>.swiper-slide{background:#fff;box-sizing:border-box;border:1px solid
#ccc;line-height:120px;text-align:center;}.swiper-slide span{font-
size:17px;}.swiper-container{height:120px;margin:0px 0 35px;}</style>

```

```

<style>#picker-date-container .picker-item{color:#999;}#picker-date-
container .picker-selected{color:#000;}@media (max-width: 767px) {#picker-date-
container .picker-items{font-size:21px;}#picker-date-container .picker-
item{height:36px;line-height:36px;padding:0 6px;}}</style>

<script>
    var myApp = new Framework7();
    // Inline date-time
    var today = new Date();
    var pickerInline = myApp.picker({
        input: '#picker-date',
        container: '#picker-date-container',
        toolbar: false,
        rotateEffect: true,
        value: [today.getMonth(), today.getDate(), today.getFullYear(),
today.getHours(), (today.getMinutes() < 10 ? '0' + today.getMinutes() :
today.getMinutes())],
        onChange: function (picker, values, displayValues) {
            var daysInMonth = new Date(picker.value[2], picker.value[0]*1 +
1, 0).getDate();
            if (values[1] > daysInMonth) {
                picker.cols[1].setValue(daysInMonth);
            }
        },
        formatValue: function (p, values, displayValues) {
            return displayValues[0] + ' ' + values[1] + ', ' + values[2] + '
' + values[3] + ':' + values[4];
        },
        cols: [
            // Months
            {
                values: ('0 1 2 3 4 5 6 7 8 9 10 11').split(' '),
                displayValues: ('January February March April May June July
August September October November December').split(' '),
                textAlign: 'left'
            },
            // Days
            {
                values:
[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,
30,31],
            },
            // Years

```

```

        {
            values: (function () {
                var arr = [];
                for (var i = 1950; i <= 2030; i++) { arr.push(i); }
                return arr;
            })(),
        },
        // Space divider
        {
            divider: true,
            content: ' '
        },
        // Hours
        {
            values: (function () {
                var arr = [];
                for (var i = 0; i <= 23; i++) { arr.push(i); }
                return arr;
            })(),
        },
        // Divider
        {
            divider: true,
            content: ':'
        },
        // Minutes
        {
            values: (function () {
                var arr = [];
                for (var i = 0; i <= 59; i++) { arr.push(i < 10 ? '0' + i
: i); }

                return arr;
            })(),
        }
    ]
});
</script>
</body>

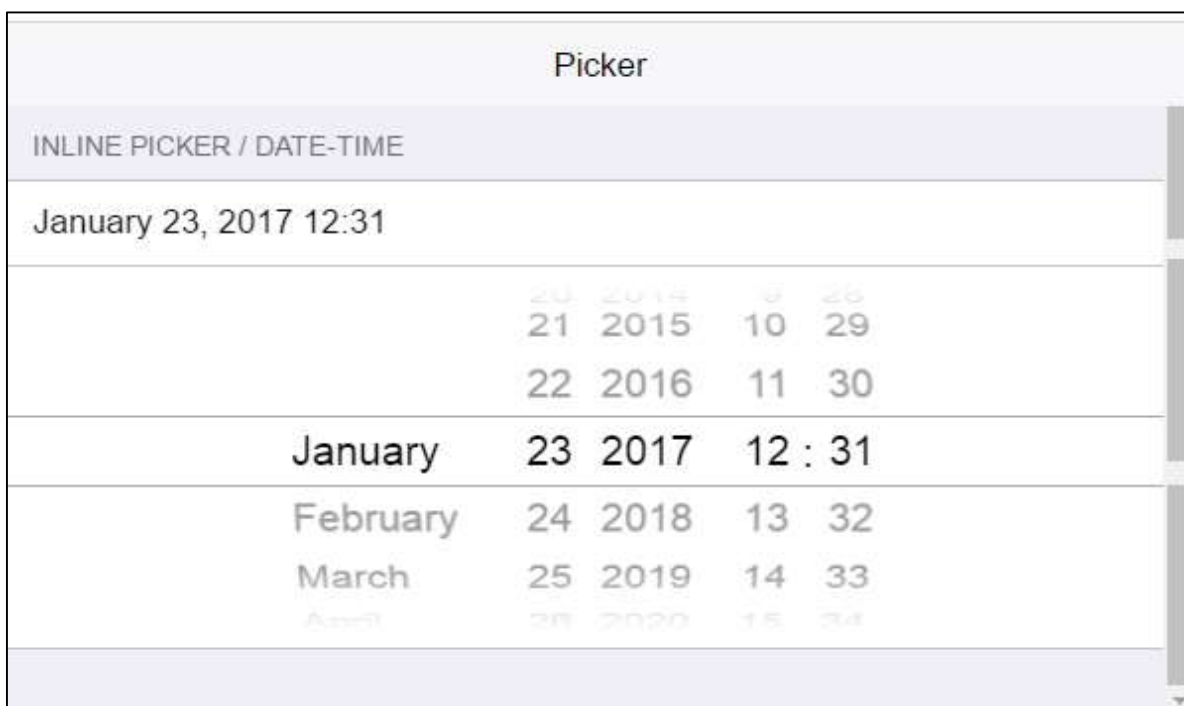
```

```
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **inline_pickerDate_time.html** file in your server root folder.
- Open this HTML file as http://localhost/inline_pickerDate_time.html and the output is displayed as shown below.
- The code can be used to select date, month, year for date and hours, minutes, seconds for time.



25. Framework7 – Calendar

Description

The Calendar component allows you to handle dates easily and can be used as an inline or as an overlay component. The overlay calendar will be converted to popover on tablets automatically.

Calendar can be created and initialized only by using JavaScript. You need to use the related App's method as shown below:

- **myApp.calendar(parameters)** - This method returns initialized calendar instance. It accepts an *object* as paramaters.

The following table shows the calendar usage in Framework7:

S. No.	Calendar Usage & Description
1	Calendar Parameters Framework7 provides a list of parameters, which are used with calendar.
2	Calendar Methods & Properties You will get an initialized instance variable to use the calendar methods and properties, once you initialize the calendar.
3	Access to Calendar's Instance It is possible to access the calendar instance from its HTML container when you initialize the calendar as inline.

Framework7 – Calendar Parameters

Description

Framework7 provides a list of parameters, which are used with calendar and are listed in the following table:

S. No.	Parameter & Description	Type	Default
Common Picker Modal Component Parameters			
1	container This is the place where the generated calendar HTML is placed. It is used only for inline pickers.	string or HTMLElement	-
2	input It is the string or HTMLElement with related input element.	string or HTMLElement	-

3	scrollToInput It is the scroll viewport to input on opening calendar.	boolean	true
4	inputReadOnly It is used to set <i>readonly</i> attribute on certain input.	boolean	true
5	convertToPopover It is used to convert calendar modal to popover on big screen.	boolean	true
6	onlyOnPopover If it is enabled, the calendar is opened in popover always.	boolean	false
7	cssClass It is the additional CSS to be set on calendar modal.	string	-
8	closeByOutsideClick If this is enabled, the picker will be closed when you click outside of picker.	boolean	true
9	toolbar It enables calendar modal toolbar.	boolean	true
10	toolbarCloseText It is the text for close button of toolbar.	string	Done
11	toolbarTemplate It is the toolbar HTML template. It is HTML string by default, with below template: <pre><div class="toolbar"> <div class="toolbar-inner"> {{monthPicker}} {{yearPicker}} </div> </div></pre>	string	-
Specific Calendar Parameters			
1	value It is the array with initial selected dates	array	-
2	disabled It is the additional disabled dates.	Date Range	-
3	events It is the dates with events and will be marked with <i>dot</i> on calendar day.	Date Range	-

4	rangesClasses It is used to add custom CSS class for additional styling.	array	-
5	formatValue It is the function to format input value and returns new/formatted string value. The <i>values</i> is array consisting of item representing selected date.	function (p, values)	-
6	monthNames It is the array with full months.	array	['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']
7	monthNamesShort It is the array with short form for month names.	array	['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
8	dayNames It is the array with week days.	array	['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday']
9	dayNamesShort It is the array with short form for week day names.	array	['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat']
10	updateValuesOnTouchmove It updates picker and input values when touch move.	boolean	true
11	firstDay It is the first day of the week. By default, it is 1 - monday.	number	1
12	weekendDays It is the an array with index number of weekend days. By default, it is saturday and sunday.	array	[0, 6]
13	dateFormat It is the default date format and below are the available expressions: <ul style="list-style-type: none"> 'yyyy' - It is the 4 digits year. 'yy' - It is the 2 digits year. 'mm' - It is the 2 digits month number i.e. from 01 to 12. 	string	'yyyy-mm-dd'

	<ul style="list-style-type: none"> • 'm' - It is the month number from 1 to 12. • 'MM' - It is the full month name. • 'M' - It is the short form of month name. • 'dd' - It is the 2 digit day number i.e. from 01 to 31. • 'd' - It is the day number from 1 to 31. • 'DD' - It is the full week day name. • 'D' - It is the short form for week day name. 		
14	multiple It allows you to select multiple dates/values.	boolean	false
15	rangePicker Enable this to enable the range picker.	boolean	false
16	direction It is the month layout direction which can be <i>horizontal</i> or <i>vertical</i> .	string	horizontal
17	minDate It is the minimum allowed date.	Date	null
18	maxDate It is the maximum allowed date.	Date	null
19	touchmove If this is enabled then calendar months slides follow fingers during touch move.	boolean	true
20	animate It enables transition between months.	boolean	true
21	closeOnSelect If it is enabled then, calendar will be closed when user pick a date.	boolean	false
22	weekHeader It represents the week header with short form of week days.	boolean	true
23	monthPicker If enabled then, display the month picker in toolbar.	boolean	true
24	monthPickerTemplate It is the month picker HTML template. By default it is as follows:	string	-

	<pre> <div class="picker-calendar-month- picker"> <i class="icon icon- prev"></i> <i class="icon icon- next"></i> </div> </pre>		
25	<p>yearPicker If it is enabled then, displays year picker in toolbar.</p>	boolean	true
26	<p>yearPickerTemplate It is the year picker HTML template. By default, it is as follows:</p> <pre> <div class="picker-calendar-year- picker"> <i class="icon icon- prev"></i> <i class="icon icon- next"></i> </div> </pre>	string	-
Callbacks			
1	<p>onChange It is the callback function which will get run when picker value changes. It accepts <i>values</i> and <i>displayValues</i> arrays</p>	function (p, values, displayValues)	-

	as parameters where each item represents <i>value/display value</i> of related column.		
2	onMonthAdd It is the callback functions which will get run when newly generated month HTML element will be added to calendar.	function (p, monthContainer)	-
3	onDayClick This callback function will be executed if user clicks any date.	function (p, dayContainer, year, month, day)	-
4	onMonthYearChangeStart This callback function will be executed during the beginning of transition to another month/year.	function (p, year, month)	-
5	onMonthYearChangeEnd This callback function will be executed in the end of transition to another month/year.	function (p, year, month)	-
6	onOpen This callback function will be executed when picker is opened.	function (p)	-
7	onClose This callback function will be executed when picker is closed.	function (p)	-

Date Range

Calendar parameters such as *disabled*, *events* and *rangesClasses* accept the so called Date Range. It is easy to specify and capture all the possible combinations of the dates. The following code shows how to use array with dates:

```
var myCalendar = myApp.calendar({
  ...
  //Disabled 1st December 2016 and 10th December 2016:
  disabled: [new Date(2016, 12 , 1), new Date(2016, 12, 10)],
  ...
});
```

You can use the custom function where you need to return *true* or *false* as shown in the following code:

```
var myCalendar = myApp.calendar({
  ...
  //Disabled all dates in December 2016
  disabled: function (date) {
    if (date.getFullYear() === 2016 && date.getMonth() === 12) {
      return true;
    }
  }
});
```

```

        else {
            return false;
        }
    },
    ...
});

```

You can also use any of the following listed ways:

- Object with *from* and *to* properties.
- Just *from* or *to* properties.
- Array with mixed dates and objects.

```

var myCalendar = myApp.calendar({
    ...
    //Disable all dates between 1st December 2016 and 10th December 2016
    disabled: {
        from: new Date(2016, 12, 1),
        to: new Date(2016, 12, 10)
    },
    ...
});

```

Or

```

var myCalendar = myApp.calendar({
    ...
    //Disable everyting since December 2015
    disabled: {
        from: new Date(2016, 12, 1)
    },
    ...
});

```

Or

```

var myCalendar = myApp.calendar({
    ...
    events: [
        new Date(2016, 12, 1),
        new Date(2016, 12, 10),
    ],
});

```

```

    {
      from: new Date(2016, 12, 15),
      to: new Date(2016, 12, 20)
    },
    {
      from: new Date(2016, 12, 25),
      to: new Date(2016, 12, 30)
    }
  ],
  ...
});

```

rangesClasses

It takes array of objects with Date Range and class names as shown in the following code:

```

var myCalendar = myApp.calendar({
  ...
  //Add classes for november and december
  rangesClasses: [
    //Add day-november class for all november dates
    {
      // string CSS class name for this range in cssClass property
      cssClass: 'day-november ', //string CSS class
      // Date Range in range property
      range: function (date) {
        return date.getMonth() === 11
      }
    },
    //Add day-holiday class for 1-10th December 2016
    {
      cssClass: 'day-holiday',
      range: {
        from: new Date(2016, 12, 1),
        to: new Date(2016, 12, 10)
      }
    }
  ],
  ...
});

```

```
});
```

Calendar Methods & Properties

Description

You will get an initialized instance variable to use the calendar methods and properties, once you initialize the calendar.

The following table shows the calendar methods and properties used in Framework7:

S. No.	Methods & Properties with Description
Properties	
1	myCalendar.params It is the object with passed initialization parameters.
2	myCalendar.value It is an array where each item holds the selected date.
3	myCalendar.opened It is <i>true</i> if calendar is currently opened.
4	myCalendar.inline It is <i>true</i> if calendar is an inline calendar.
5	myCalendar.cols It is an array with specified columns where each column also has its own method and properties.
6	myCalendar.container It is the DOM7 instance with calendar HTML container.
Methods	
7	myCalendar.setValue(values) It sets new selected dates. It accepts values, which is an array, where each item represents the selected date.
8	myCalendar.nextMonth(duration) It is the calendar transition to the next month in ms.
9	myCalendar.prevMonth(duration) It is the calendar transition to the previous month in ms.
10	myCalendar.nextYear() It is the calendar transition to the next year.
11	myCalendar.prevYear() It is the calendar transition to the previous year.

12	myCalendar.setYearMonth(year, month, duration) It is the calendar transition to aspecified year and month for specified duration in ms.
13	myCalendar.open() It is used to open the calendar.
14	myCalendar.close() It is used to close the calendar.
15	myCalendar.destroy() It is used to destroy calendar instance and removes all the events.

Access To Calendar's Instance

Description

It is possible to access the calendar instance from its HTML container when you initialize the calendar as inline as shown below:

```
var myCalendar = $$('.calenadr-inline')[0].f7Calendar;
```

Example

The following example demonstrates the use of calendar in Framework7:

```
<!DOCTYPE html>
<html class="with-statusbar-overlay">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Calendar</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
    <link rel="stylesheet" href="custom.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
```



```

<div data-page="home" class="page navbar-fixed">
  <div class="navbar">
    <div class="navbar-inner">
      <div class="left"> </div>
      <div class="center">Calendar</div>
      <div class="right"> </div>
    </div>
  </div>
  <div class="page-content">
    <div class="content-block-title">Default calendar
setup</div>
    <div class="list-block">
      <ul>
        <li>
          <div class="item-content">
            <div class="item-inner">
              <div class="item-input">
                <input type="text" placeholder="Select a
date" readonly id="calendar-default">
              </div>
            </div>
          </div>
        </li>
      </ul>
    </div>
    <div class="content-block-title">Custom date format
calendar</div>
    <div class="list-block">
      <ul>
        <li>
          <div class="item-content">
            <div class="item-inner">
              <div class="item-input">
                <input type="text" placeholder="Select a
date" readonly id="calendar-date-format">
              </div>
            </div>
          </div>
        </li>
      </ul>
    </div>
  </div>
</div>

```

```

        </ul>
    </div>
    <div class="content-block-title">Multiple values
calendar</div>

    <div class="list-block">
        <ul>
            <li>
                <div class="item-content">
                    <div class="item-inner">
                        <div class="item-input">
                            <input type="text" placeholder="Select
multiple dates" readonly id="calendar-multiple">
                        </div>
                    </div>
                </div>
            </li>
        </ul>
    </div>
    <div class="content-block-title">Range picker
calendar</div>

    <div class="list-block">
        <ul>
            <li>
                <div class="item-content">
                    <div class="item-inner">
                        <div class="item-input">
                            <input type="text" placeholder="Select
date range" readonly id="calendar-range">
                        </div>
                    </div>
                </div>
            </li>
        </ul>
    </div>
    <div class="content-block-title">Calendar with events
(Additional dot is marked on calendar day from today to weeklater.)</div>
    <div class="list-block">
        <ul>
            <li>
                <div class="item-content">

```

```

        <div class="item-inner">
            <div class="item-input">
                <input type="text" placeholder="Select
date" readonly id="calendar-events">
            </div>
        </div>
    </div>
</li>
</ul>
</div>
<div class="content-block-title">Disabled dates (Today to
weeklater)</div>
<div class="list-block">
    <ul>
        <li>
            <div class="item-content">
                <div class="item-inner">
                    <div class="item-input">
                        <input type="text" placeholder="Select
date" readonly id="calendar-disabled">
                    </div>
                </div>
            </div>
        </li>
    </ul>
</div>
<div class="content-block-title">Inline with custom
toolbar</div>
<div class="content-block">
    <div style="padding:0; margin-right:-15px; width:auto"
class="content-block-inner">
        <div id="calendar-inline-container"></div>
    </div>
</div>
</div>
</div>
</div>
</div>

```

```
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    // Default Calendar
    var calendarDefault = myApp.calendar({
        input: '#calendar-default',
    });
    // With custom date format
    var calendarDateFormat = myApp.calendar({
        input: '#calendar-date-format',
        dateFormat: 'DD, dd MM, yyyy'
    });
    // With multiple values
    var calendarMultiple = myApp.calendar({
        input: '#calendar-multiple',
        dateFormat: ' DD dd M yyyy',
        multiple: true
    });
    // Range Picker
    var calendarRange = myApp.calendar({
        input: '#calendar-range',
        dateFormat: 'dd M yyyy',
        rangePicker: true
    });
    // With events
    var today = new Date();
    var weekLater = new Date().setDate(today.getDate() + 7);
    var calendarEvents = myApp.calendar({
        input: '#calendar-events',
        dateFormat: 'M dd yyyy',
        events: {
            from: today,
            to: weekLater
        }
    });
    // Disabled
    var today = new Date();
```

```

var weekLater = new Date().setDate(today.getDate() + 7);
var calendarDisabled = myApp.calendar({
  input: '#calendar-disabled',
  dateFormat: 'M dd yyyy',
  disabled: {
    from: today,
    to: weekLater
  }
});
// Inline with custom toolbar
var $$ = Dom7;
var monthNames = ['January', 'February', 'March', 'April', 'May',
'June', 'July', 'August', 'September', 'October', 'November', 'December'];
var calendarInline = myApp.calendar({
  container: '#calendar-inline-container',
  value: [new Date()],
  weekHeader: false,
  toolbarTemplate:
    '<div class="toolbar calendar-custom-toolbar">' +
      '<div class="toolbar-inner">' +
        '<div class="left">' +
          '<a href="#" class="link icon-only"><i class="icon
icon-back"></i></a>' +
          '</div>' +
          '<div class="center"></div>' +
          '<div class="right">' +
            '<a href="#" class="link icon-only"><i class="icon
icon-forward"></i></a>' +
            '</div>' +
          '</div>'+
        '</div>',
  onOpen: function (p) {
    $$('.calendar-custom-toolbar
.center').text(monthNames[p.currentMonth] +', ' + p.currentYear);
    $$('.calendar-custom-toolbar .left .link').on('click',
function () {
      calendarInline.prevMonth();
    });
    $$('.calendar-custom-toolbar .right .link').on('click',
function () {

```

```

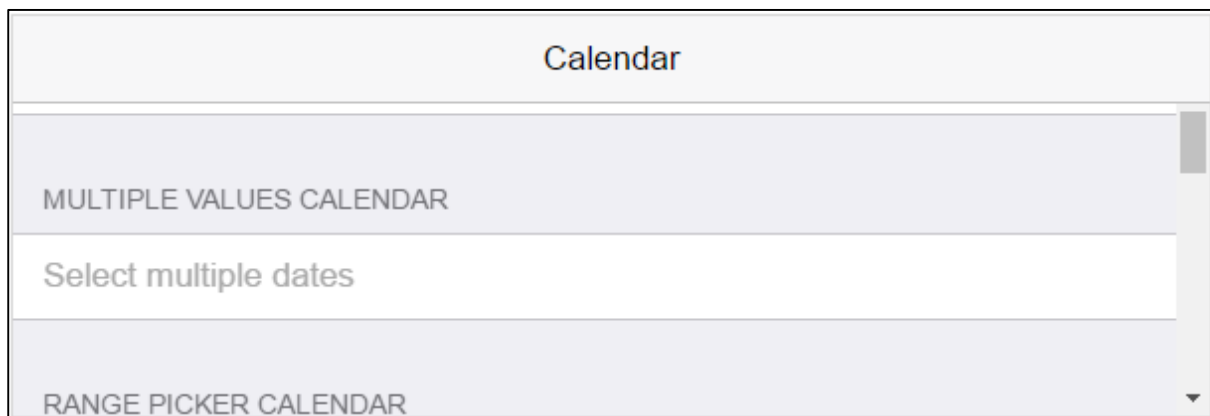
        calendarInline.nextMonth();
    });
},
onMonthYearChangeStart: function (p) {
    $$('.calendar-custom-toolbar
.center').text(monthNames[p.currentMonth] + ', ' + p.currentYear);
}
});
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **calendar_access_instance.html** file in your server root folder.
- Open this HTML file as http://localhost/calendar_access_instance.html and the output is displayed as shown below.
- Here, when you initialize the calendar, you can access the calendar instance from its HTML container.



26. Framework7 – Refresh

Description

It is a special component used to refresh (reload) the page contents by pulling it.

The following code shows how to refresh the page content:

```
<div class="page">
  <!-- Page content should have additional "pull-to-refresh-content" class -->
  <div class="page-content pull-to-refresh-content" data-ptr-distance="55">
    <!-- Default pull to refresh layer-->
    <div class="pull-to-refresh-layer">
      <div class="preloader"></div>
      <div class="pull-to-refresh-arrow"></div>
    </div>

    <!-- usual content below -->
    <div class="list-block">
      ...
    </div>
  </div>
</div>
```

The following classes are used in refresh:

- **page-content:** It has an additional *pull-to-refresh-content* class and its required to enable pull to refresh.
- **pull-to-refresh-layer:** It is a hidden layer, which is used to *pull to refresh* visual element and it is just a preloader and an arrow.
- **data-ptr-distance="55":** This is additional attribute that allows you to set custom 'pull to refresh' trigger distance and its default value is 44px.

Pull To Refresh Events

In 'Pull to Refresh' there are some JavaScript events, which are given in the following table:

S. No.	Event & Description	Target
1	pullstart It will be triggered whenever you start pulling to refresh content.	Pull To Refresh content. <pre><div class="pull-to-refresh-content"></pre>
2	pullmove It is triggered when you are pulling to refresh content.	Pull To Refresh content. <pre><div class="pull-to-refresh-content"></pre>
3	pullend It will be triggered whenever you release pull to refresh content.	Pull To Refresh content. <pre><div class="pull-to-refresh-content"></pre>
4	refresh This event will be triggered when the pull to refresh enters in the "refreshing" state.	Pull To Refresh content. <pre><div class="pull-to-refresh-content"></pre>
5	refreshdone It will be triggered after it is refreshed and it goes back to the initial state. This will be done after calling <i>pullToRefreshDone</i> method.	Pull To Refresh content. <pre><div class="pull-to-refresh-content"></pre>

There are App's methods that can be used with Pull to Refresh.

S. No.	App's Methods & Description
1	myApp.pullToRefreshDone(ptrContent) It is used to reset <i>pull-to-refresh</i> content.
2	myApp.pullToRefreshTrigger(ptrContent) It is used to trigger to refresh on specified <i>pull to refresh</i> content.
3	myApp.destroyPullToRefresh(ptrContent) It is used to <i>destroy/disable</i> pull to refresh on specified <i>pull to refresh</i> content.

4	<p>myApp.initPullToRefresh(ptrContent) It is used to <i>initialize/enable pull to refresh content</i>.</p>
---	--

Where *ptrContent* is used to *HTMLElement* or *string* to pull-to-refresh-content to reset/trigger or disable/enable.

Example

The following example demonstrates the use of refresh component that initiates the refreshing of a page contents:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Pull To Refresh</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <div class="views">
    <div class="view view-main">
      <div class="pages">
        <div data-page="home" class="page navbar-fixed">
          <div class="navbar">
            <div class="navbar-inner">
              <div class="left"> </div>
              <div class="center">Pull To Refresh</div>
              <div class="right"> </div>
            </div>
          </div>
          <div class="page-content pull-to-refresh-content">
            <div class="pull-to-refresh-layer">
              <div class="preloader"></div>
              <div class="pull-to-refresh-arrow"></div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

    <div class="list-block media-list">
      <ul>
        <li class="item-content">
          <div class="item-media"></div>
          <div class="item-inner">
            <div class="item-title-row">
              <div class="item-title">apple</div>
            </div>
          </div>
        </li>
        <li class="item-content">
          <div class="item-media"></div>
          <div class="item-inner">
            <div class="item-title-row">
              <div class="item-title">strawberry</div>
            </div>
          </div>
        </li>
        <li class="item-content">
          <div class="item-media"></div>
          <div class="item-inner">
            <div class="item-title-row">
              <div class="item-title">Mango</div>
            </div>
          </div>
        </li>
      </ul>
      <div class="list-block-label">
        <p>Just pull page down to let the magic happen.</p>
      </div>
    </div>
  </div>
</div>
</div>
</div>
</div>

```

```

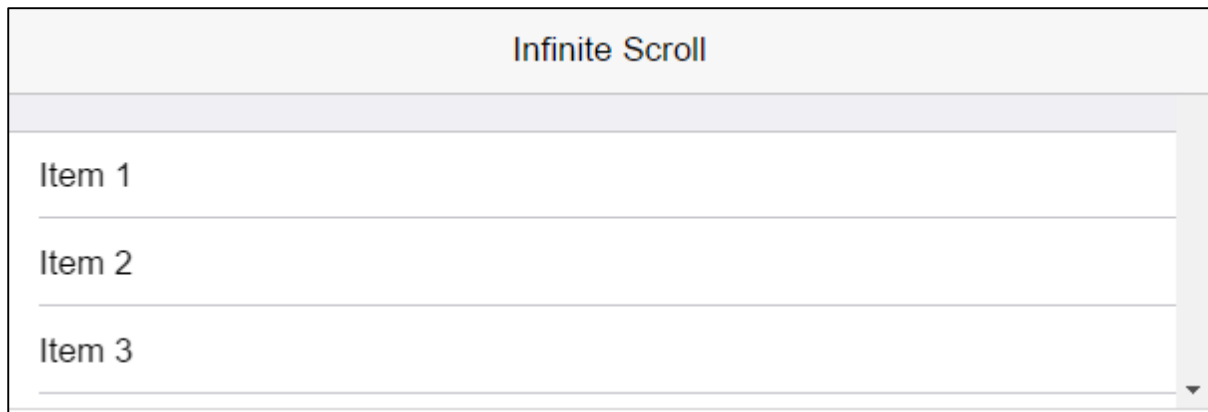
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
  var myApp = new Framework7();
  var $$ = Dom7;
  // Dummy Content
  var fruits = ['mango', 'orange', 'watermelon', 'banana'];
  // Pull to refresh content
  var ptrContent = $$('.pull-to-refresh-content');
  // Add 'refresh' listener on it
  ptrContent.on('refresh', function (e) {
    // Emulate 2s loading
    setTimeout(function () {
      var picURL = 'images/Fruit.jpg' + Math.round(Math.random() * 100);
      var fruit = fruits[Math.floor(Math.random() * fruits.length)];
      var itemHTML = '<li class="item-content">' +
        '<div class="item-media"></div>' +
        '<div class="item-inner">' +
        '<div class="item-title-row">' +
        '<div class="item-title">' + fruit + '</div>' +
        '</div>' +
        '</div>' +
        '</li>';
      // Prepend new list element
      ptrContent.find('ul').prepend(itemHTML);
      // When loading done, we need to reset it
      myApp.pullToRefreshDone();
    }, 2000);
  });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **pull_to_refresh.html** file in your server root folder.
- Open this HTML file as http://localhost/pull_to_refresh.html and the output is displayed as shown below.
- When the user pulls down, the page will be refreshed with the contents.



27. Framework7 – Infinite Scroll

Description

The Infinite Scroll allows you to load additional content and performs the required actions when the page is near to the bottom.

The following HTML layout shows the infinite scroll:

```
<div class="page">
  <div class="page-content infinite-scroll" data-distance="100">
    ...
  </div>
</div>
```

The above layout contains the following classes:

- **page-content infinite-scroll:** It is used for infinite scroll container.
- **data-distance:** This attribute allows you to configure distance from the bottom of page (in px) to trigger infinite scroll event and its default value is **50px**.

If you want to use infinite scroll on top the page, you need to add additional *"infinite-scroll-top"* class to *"page-content"*:

```
<div class="page">
  <div class="page-content infinite-scroll infinite-scroll-top">
    ...
  </div>
</div>
```

Infinite Scroll Events

infinite: It is used to trigger when page scroll reaches specified distance to the bottom. It will be target by *div class="page-content infinite-scroll"*.

There are two App's methods that can be used with infinite scroll container:

To add infinite scroll event listener to the specified HTML container, use the following method:

```
myApp.attachInfiniteScroll(container)
```

You can remove the infinite scroll event listener from the specified HTML container by using the following method:

```
myApp.detachInfiniteScroll(container)
```

Where *parameters* are required options used as *HTMLElement* or *string* for infinite scroll container.

Example

The following example demonstrates the infinite scroll that loads the additional content when the page scroll is near to the bottom:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>infinite_scroll</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <div class="views">
    <div class="view view-main">
      <div class="navbar">
        <div class="navbar-inner">
          <div class="left"> </div>
          <div class="center sliding">Infinite Scroll</div>
          <div class="right"> </div>
        </div>
      </div>
      <div class="pages navbar-through">
        <div data-page="home" class="page">
          <div class="page-content infinite-scroll">
            <div class="list-block">
              <ul>
                <li class="item-content">
                  <div class="item-inner">
                    <div class="item-title">Item 1</div>
                  </div>
                </li>
              </ul>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
```

```
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Item 2</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Item 3</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Item 4</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Item 5</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Item 6</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Item 7</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Item 8</div>
  </div>
</li>
<li class="item-content">
  <div class="item-inner">
    <div class="item-title">Item 9</div>
  </div>
</li>
```

```
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Item 10</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Item 11</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Item 12</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Item 13</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Item 14</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Item 15</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Item 16</div>
        </div>
    </li>
    <li class="item-content">
```



```

        <div class="item-inner">
            <div class="item-title">Item 17</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Item 18</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Item 19</div>
        </div>
    </li>
    <li class="item-content">
        <div class="item-inner">
            <div class="item-title">Item 20</div>
        </div>
    </li>
</ul>
</div>
<div class="infinite-scroll-preloader">
    <div class="preloader"></div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<style>.infinite-scroll-preloader{margin-top:-20px;margin-bottom:10px;text-
align:center;}.infinite-scroll-preloader
.preloader{width:34px;height:34px;}</style>
<script>
    var myApp = new Framework7();
    var $$ = Dom7;
    // Loading flag

```

```

var loading = false;
// Last loaded index
var lastIndex = $$('.list-block li').length;
// Max items to load
var maxItems = 60;
// Append items per load
var itemsPerLoad = 20;
// Attach 'infinite' event handler
$$('.infinite-scroll').on('infinite', function () {
  // Exit, if loading in progress
  if (loading) return;
  // Set loading flag
  loading = true;
  // Emulate 1s loading
  setTimeout(function () {
    // Reset loading flag
    loading = false;
    if (lastIndex >= maxItems) {
      // Nothing more to load, detach infinite scroll events to prevent
      // unnecessary loadings
      myApp.detachInfiniteScroll($$('.infinite-scroll'));
      // Remove preloader
      $$('.infinite-scroll-preloader').remove();
      return;
    }
    // Generate new items HTML
    var html = '';
    for (var i = lastIndex + 1; i <= lastIndex + itemsPerLoad; i++) {
      html += '<li class="item-content"><div class="item-inner"><div
class="item-title">Item ' + i + '</div></div></li>';
    }
    // Append new items
    $$('.list-block ul').append(html);
    // Update last loaded index
    lastIndex = $$('.list-block li').length;
  }, 1000);
});
</script>
</body>

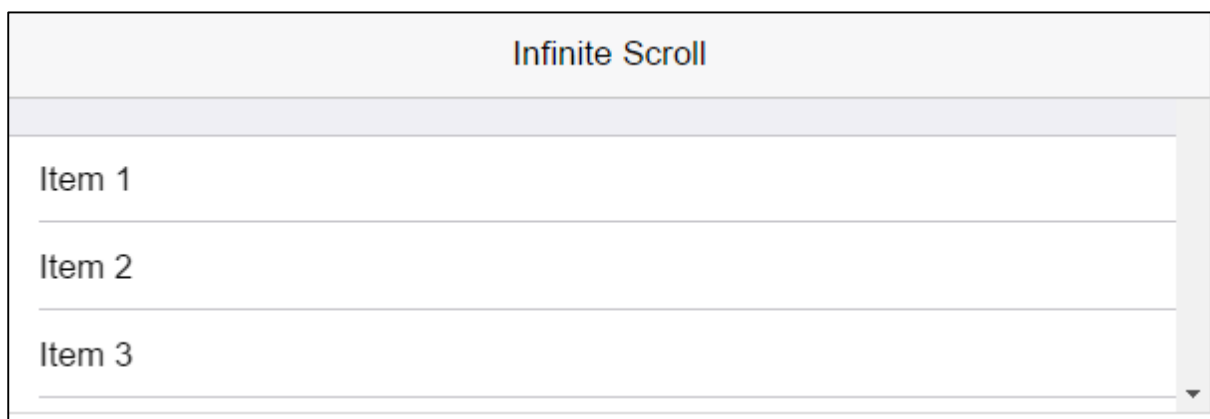
```

```
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **infinite_scroll.html** file in your server root folder.
- Open this HTML file as http://localhost/infinite_scroll.html and the output is displayed as shown below.
- The example allows loading of additional content when your page scroll reaches specified distance to the bottom.



28. Framework7 – Messages

Description

Messages are component of Framework7, which gives visualization of comments and messaging system in the application.

Messages Layout

The framework7 has the following messages layout structure:

```
<div class="page">
<div class="page-content messages-content">
  <div class="messages">
    <!-- Defines the date stamp -->
    <div class="messages-date">Monday, Apr 26 <span>10:30</span></div>
    <!-- Displays the sent message and by default, it will be in green color
on right side -->
    <div class="message message-sent">
      <!-- Define the text with bubble type -->
      <div class="message-text">Hello</div>
    </div>
    <!-- Displays the another sent message -->
    <div class="message message-sent">
      <!-- Define the text with bubble type -->
      <div class="message-text">How are you?</div>
    </div>
    <!-- Displays the received message and by default, it will be in grey
color on left side -->
    <div class="message message-with-avatar message-received">
      <!-- Provides sender name -->
      <div class="message-name">Smith</div>
      <!-- Define the text with bubble type -->
      <div class="message-text">I am fine, thanks</div>
      <!-- Defines the another date stamp -->
      <div class="messages-date">Tuesday, April 28 <span>11:16</span></div>
    </div>
  </div>
</div>
```

The layout contains the following classes in different areas:

Messages Page Layout

The following table shows the classes of messages page layout with description.

S. No.	Classes & Description
1	messages-content It is a required additional class added to "page-content" and used for messages wrapper.
2	messages It is a required element for messages bubbles.
3	messages-date It uses date stamp container to display date and time of message sent or received.
4	message It is a single message to be displayed.

Single Message Inner Parts

The following table shows the classes of simple message inner parts with description.

S. No.	Classes & Description
1	message-name It provides the sender name.
2	message-text Define the text with bubble type.
3	message-avatar It specifies the sender avatar.
4	message-label It specifies the text label below the bubble.

Additional classes for Single Message Container

The following table shows additional classes for single message container description.

S. No.	Classes & Description
1	message-sent It specifies that message was sent by the user and is displayed with green background color on the right side.
2	message-received It is used for displaying the single message indicating that, the message was received by user and stays on the left side with grey background color.

3	message-pic It is an additional class for displaying image with a single message.
4	message-with-avatar It is an additional class for displaying a single message (received or sent) with avatar.
5	message-with-tail You can add abubble tail for single message (received or sent).

Additional Available Classes for Single Message

The following table shows the available classes for a single message with description.

S. No.	Classes & Description
1	message-hide-name It is an additional class for hiding message name for a single message (received or sent).
2	message-hide-avatar It is an additional class for hiding message avatar for a single message (received or sent).
3	message-hide-label It is an additional class for hiding message label for a single message (received or sent).
4	message-last You can use this class to indicate the last received or sent message in current conversation by one sender for a single message (received or sent).
5	message-first You can use this class to indicate first received or first sent message in current conversation by one sender for a single message (received or sent).

Initializing Messages with JavaScript

You can initialize the messages with JavaScript by using the following method:

```
myApp.messages(messagesContainer, parameters)
```

The method takes two options:

- **messagesContainer:** It is a required HTML element or string that includes messages container HTML element.
- **parameters:** It specifies an object with messages parameters.

Messages Parameters

The following table shows the parameters of messages with description.

S. No.	Parameter & Description	Type	Default
1	autoLayout It adds additional required classes to each message by enabling it.	boolean	true
2	newMessagesFirst You can display message on top instead of displaying on bottom by enabling it.	boolean	false
3	messages It displays an array of messages in which each message should be represented as object with message parameters.	array	-
4	messageTemplate It displays the a single message template.	string	-

Messages Properties

The following table shows the properties of messages with description.

S. No.	Property & Description
1	myMessages.params You can initialize the passed parameters with object.
2	myMessages.container Defines the DOM7 element with a message bar HTML container.

Messages Methods

The following table shows the methods of messages with description.

S. No.	Method & Description
1	<p>myMessages.addMessage(messageParameters, method, animate); The message can be added to the beginning or to the end by using the method parameter.</p> <p>It has the following parameters:</p> <ul style="list-style-type: none"> • messageParameters: It provides message parameters to add. • method: It is a string type that adds message to the beginning or to the end of messages container.

	<ul style="list-style-type: none"> • animate: It is a boolean type, which adds a message without any transition or scrolling animation by setting it to false. By default, it will be true.
2	myMessages.appendMessage(messageParameters, animate); It adds a message to the end of message container.
3	myMessages.prependMessage(messageParameters, animate); It adds a message to the beginning of message container.
4	myMessages.addMessages(messages, method, animate); You can add multiple messages at one time. It has the following parameter: <ul style="list-style-type: none"> • messages: It provides an array of messages to add, which should be represented as object with message parameters.
5	myMessages.removeMessage(message); It is used to remove the message. It has the following parameter: <ul style="list-style-type: none"> • message: It is a required HTML element or string that removes the message element.
6	myMessages.removeMessages(messages); You can remove the multiple messages. It has the following parameter: <ul style="list-style-type: none"> • messages: It is a required array with HTML elements or string that removes the messages.
7	myMessages.scrollMessages(); You can scroll messages from top to bottom and vice versa depending on the first parameter of new message.
8	myMessages.layout(); Auto layout can be applied to the messages.
9	myMessages.clean(); It is used to clean the messages.
10	myMessages.destroy(); It is used to destroy the messages.

Single Message Parameters

The following table shows the parameters for a single message with description.

S. No.	Parameter & Description	Type	Default
1	text It defines the message text, which could be a HTML string.	string	-
2	name It specifies the sender name.	string	-
3	avatar It specifies the sender avatar URL string.	string	-
4	time It specifies the time string of the message like '9:45 AM', '18:35'.	string	-
5	type It provides type of message whether it could be sent or recieved message.	string	sent
6	label It defines the label of the message.	string	-
7	day It gives the day string of the message like 'sunday', 'monday', 'yesterday' etc.	string	-

Initializing Messages with HTML

You can initialize the messages with HTML without JavaScript by using the additional *messages-init* class to *messages* and use the *data-* attributes to pass the required parameters as shown in the code snippet given below:

```

...
<div class="page-content messages-content">
  <!-- Initialize the messages using additional "messages-init" class-->
  <div class="messages messages-init" data-auto-layout="true" data-new-
messages-first="false">
    ...
  </div>
</div>
...

```

Example

The following example demonstrates the use of messages in the Framework7:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Messages</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed toolbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Messages</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="toolbar messagebar">
              <div class="toolbar-inner">
                <textarea placeholder="Message"></textarea><a href="#"
class="link">Send</a>
              </div>
            </div>
            <div class="page-content messages-content">
              <div class="messages">
                <div class="messages-date">Friday, Apr 26
<span>10:30</span></div>
                <div class="message message-sent">

```

```

        <div class="message-text">Hello</div>
    </div>
    <div class="message message-sent">
        <div class="message-text">Happy Birthday</div>
    </div>
    <div class="message message-received">
        <div class="message-name">Smith</div>
        <div class="message-text">Thank you</div>
        <div style="background-
image:url(/framework7/images/person.png)" class="message-avatar"></div>
    </div>
    <div class="messages-date">Saturday, Apr 27
<span>9:30</span></div>
    <div class="message message-sent">
        <div class="message-text">Good Morning...</div>
    </div>
    <div class="message message-sent">
        <div class="message-text"></div>
        <div class="message-label">Delivered</div>
    </div>
    <div class="message message-received">
        <div class="message-name">Smith</div>
        <div class="message-text">Very Good Morning...</div>
        <div style="background-
image:url(/framework7/images/person.png)" class="message-avatar"></div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var $$ = Dom7;
    // It indicates conversation flag

```

```

var conversationStarted = false;
// Here initialize the messages
var myMessages = myApp.messages('.messages', {
  autoLayout:true
});
// Initialize the messagebar
var myMessagebar = myApp.messagebar('.messagebar');
// Displays the text after clicking the button
$$('.messagebar .link').on('click', function () {
  // specifies the message text
  var messageText = myMessagebar.value().trim();
  // If there is no message, then exit from there
  if (messageText.length === 0) return;
  // Specifies the empty messagebar
  myMessagebar.clear()
  // Defines the random message type
  var messageType = (['sent',
'received'])[Math.round(Math.random())];
  // Provides the avatar and name for the received message
  var avatar, name;
  if(messageType === 'received') {
    name = 'Smith';
  }
  // It adds the message
  myMessages.addMessage({
    // It provides the message text
    text: messageText,
    // It displays the random message type
    type: messageType,
    // Specifies the avatar and name of the sender
    avatar: avatar,
    name: name,
    // Displays the day, date and time of the message
    day: !conversationStarted ? 'Today' : false,
    time: !conversationStarted ? (new Date()).getHours() + ':' +
(new Date()).getMinutes() : false
  })
  // Here you can update the conversation flag
  conversationStarted = true;

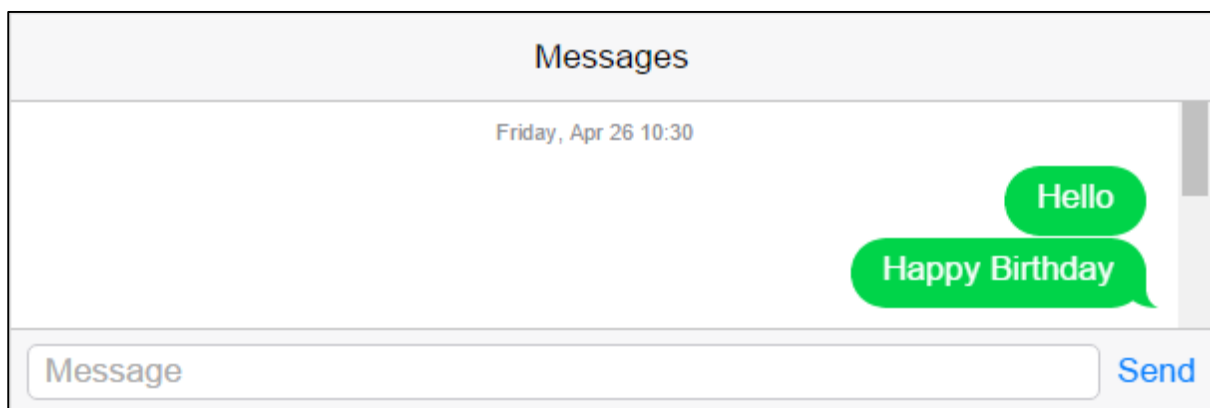
```

```
});  
</script>  
</body>  
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **messages.html** file in your server root folder.
- Open this HTML file as <http://localhost/messages.html> and the output is displayed as shown below.
- When you type a message into the message box and click the Send button, it specifies that your message has been sent and is displayed with green background color on the right side.
- The message which you receive, appears on the left side with the grey background along with the sender name.



29. Framework7 – Message Bar

Description

Framework7 provides special resizable toolbar to work with the messaging system in the application.

The following code shows the message bar layout:

```
<div class="toolbar messagebar">
  <div class="toolbar-inner">
    <textarea placeholder="Message"></textarea><a href="#"
class="link">Send</a>
  </div>
</div>
```

In the message bar, the inside of the "page" is very important and is at the right of "messages-content":

```
<div class="page toolbar-fixed">
  <!-- messagebar -->
  <div class="toolbar messagebar">
    <div class="toolbar-inner">
      <textarea placeholder="Message"></textarea><a href="#"
class="link">Send</a>
    </div>
  </div>
  <!-- messages-content -->
  <div class="page-content messages-content">
    <div class="messages">
      ... messages
    </div>
  </div>
</div>
```

You can use the following method for initializing the message bar with JavaScript:

```
myApp.messagesbar(messagesbarContainer, parameters)
```

The method has two options:

- **messagebarContainer:** It is a required HTML element or string that includes messagebar container HTML element.
- **parameters:** It specifies an object with messagebar parameters.

For example:

```
var myMessagebar = app.messagebar('.messagebar', {
    maxHeight: 200
});
```

Messagebar Parameter

maxHeight: It is used to set maximum height of textarea and will be resized depending on the amount of its text. The type of parameter is *number* and the default value is *null*.

Messagebar Properties

The following table shows the messagebar properties:

S. No.	Properties & Description
1	myMessagebar.params You can specify object with passed initialization parameters.
2	myMessagebar.container You can specify dom7 element with messagebar container HTML element.
3	myMessagebar.textarea You can specify dom7 element with messagebar textarea HTML element.

Messagebar Methods

The following table shows the messagebar methods:

S. No.	Methods & Description
1	myMessagebar.value(newValue); It sets messagebar textarea value/text and returns messagebar textarea value, if newValue is not specified.
2	myMessagebar.clear(); It clears the textarea and updates/resets the size.
3	myMessagebar.destroy(); It destroy messagebar instance.

Initialize Messagebar with HTML

You can initialize the messagebar using HTML without JavaScript methods and properties by adding the *messagebar-init* class to the *.messagebar* and you can pass the required parameters using *data-* attributes.

The following code specifies the initialization of messagebar with HTML:

```
<div class="toolbar messagebar messagebar-init" data-max-height="200">
  <div class="toolbar-inner">
    <textarea placeholder="Message"></textarea><a href="#"
class="link">Send</a>
  </div>
</div>
```

Access to Messagebar's Instance

It is possible to access the message bar instance, if you initialize it by using HTML; it is achieved by using the *f7 Message bar* property of its container element.

```
var myMessagebar = $$('.messagebar')[0].f7Messagebar;
// Now you can use it
myMessagebar.value('Hello world');
```

You can see the example of Messagebar, which is explained in this [link](#)

30. Framework7 – Notifications

Description

Notifications are used to show the required messages, which appear like Push (or Local) iOS notifications.

The following table demonstrates the use of notifications in details:

S. No.	Notifications usage & Description
1	Notifications JavaScript API The notifications can also be added or closed with JavaScript by using the related app methods.
2	Notifications Layout Framework7 notifications will be added by using JavaScript.
3	Example iOS Framework7 allows you to use different types of notifications in your iOS layout.
4	Example Material Framework7 notifications can also be used in material layout.

Framework7 – Notifications JavaScript API

Description

The notifications can also be added and closed with JavaScript by using the related app methods as shown below:

- **myApp.addNotification(parameters)** - It is used to add/show notification, which accepts required *object* with notification parameters. This method returns *HTMLElement* with notification element created dynamically.
- **myApp.closeNotification(notificationElement)** - It is used to close specified notification, which accepts *notificationElement* as mandatory parameter and it is a *HTMLElement* or *string* of required notification.

The following table shows the notification parameters required when new notification is added:

S. No.	Parameter & Description	Type	Default
1	title It represents the notification title and is equal to <i>notificationTitle</i> app's parameter by default. It is used with iOS theme only.	string	

2	<p>subtitle</p> <p>It represents the notification subtitle and is equal to <i>notificationSubtitle</i> app's parameter, by default. It is used with iOS theme only.</p>	string	
3	<p>media</p> <p>It is the notification media element, which is an HTML string with icon or image. It is equal to <i>notificationMedia</i> app's parameter by default. It is used with iOS theme only.</p>	string	
4	<p>hold</p> <p>It is used to close notification automatically after the specified timeout. It is equal to <i>notificationHold</i> app's parameter by default. It is used with iOS theme only.</p>	number	
5	<p>closeIcon</p> <p>It is disabled to remove close icon from notification. It is equal to <i>notificationCloseIcon</i> app's parameter by default.</p>	boolean	true
6	<p>button</p> <p>It is the material notification button, which accepts 3 properties namely <i>text</i>, <i>color</i> and <i>close</i>. It is used with material theme only.</p>	object	
7	<p>closeOnClick</p> <p>If this is set to <i>true</i> then, notification will be closed after clicking on it. It is equal to <i>notificationCloseOnClick</i> app's parameter by default.</p>	boolean	false
8	<p>additionalClass</p> <p>It is used to add additionally specified CSS class to notification element.</p>	string	
9	<p>custom</p> <p>When you want to use notification with custom HTML layout, this parameter is used. The <i>title</i>, <i>subtitle</i>, <i>media</i> and <i>closeIcon</i> parameters will be ignored while using this parameter.</p>	string	
10	<p>onClick</p> <p>This callback function will be executed after clicking the notification item.</p>	function	
11	<p>onClose</p> <p>This callback function will be executed when notification is closed.</p>	function	

Framework7 – Notifications Layout

Description

Although, notifications will be added using JavaScript, it is necessary to understand them for custom styling. The Framework7 will put special *notifications* div to the *body* along with the list block, when you add notification. Notification layout for this case is shown below:

```
<body>
...
<div class="notifications list-block media-list">
  <ul>
    <!-- Single notification item -->
    <li class="notification-item">
      <div class="item-content">
        <div class="item-media">
          <!-- Notification media -->
        </div>
        <div class="item-inner">
          <div class="item-title-row">
            <div class="item-title">
              <!-- Notification title goes here -->
            </div>
            <div class="item-after">
              <!-- Notification close icon -->
              <a href="#" class="close-notification"></a>
            </div>
          </div>
          <div class="item-subtitle">
            <!-- Notification subtitle goes here -->
          </div>
          <div class="item-text">
            <!-- Notification message goes here -->
          </div>
        </div>
      </div>
    </li>
  </ul>
</div>
</body>
```

The custom notification will have the following layout:

```
<body>
  ...
  <div class="notifications list-block media-list">
    <ul>
      <!-- Single notification item -->
      <li class="notification-item">
        <!-- Custom notification content goes here -->
      </li>
    </ul>
  </div>
</body>
```

Framework7 – Example iOS

Description

Framework7 allows you to use different types of notifications in your iOS layout.

Example

The following example demonstrates the use of iOS notifications in Framework7:

```
<!DOCTYPE html>
<html class="with-statusbar-overlay">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Notifications</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
    <link rel="stylesheet" href="custom.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
```

```

<div class="pages navbar-fixed">
  <div data-page="home" class="page">
    <div class="navbar">
      <div class="navbar-inner">
        <div class="center sliding">Notifications</div>
      </div>
    </div>
    <div class="page-content">
      <div class="content-block">
        <p><a href="#" class="button notification-
default">Default</a></p>
        <p><a href="#" class="button notification-full">Full-
layout notification</a></p>
        <p><a href="#" class="button notification-
custom">Custom image</a></p>
        <p><a href="#" class="button notification-
callback">Callback on close</a></p>
      </div>
    </div>
  </div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
  var myApp = new Framework7();
  var mainView = myApp.addView('.view-main');
  var $$ = Dom7;
  $$('.notification-default').on('click', function () {
    myApp.addNotification({
      title: 'Default notification',
      message: 'This is default notification with title and message'
    });
  });
  $$('.notification-full').on('click', function () {
    myApp.addNotification({
      title: 'Full-layout',
      subtitle: 'Notification subtitle',

```

```

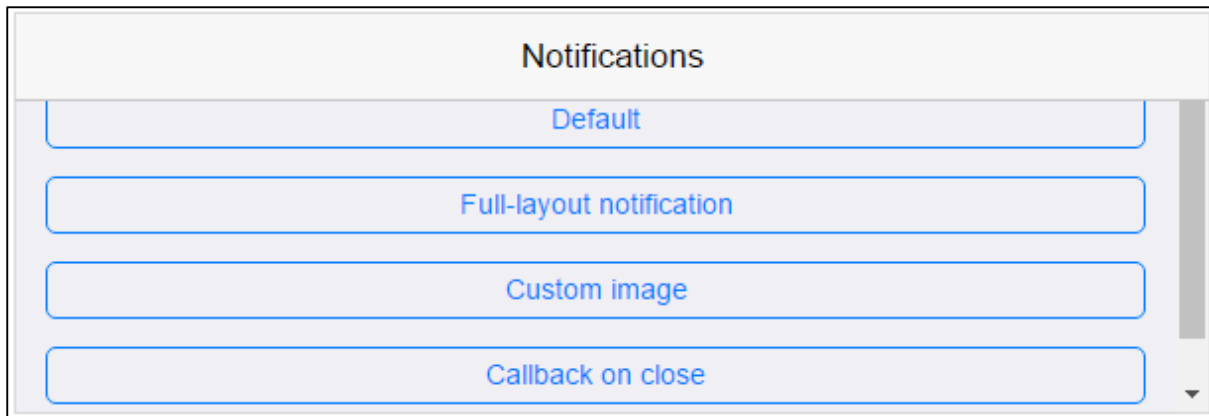
        message: 'You have a new mail',
        media: '<i class="icon icon-form-email"></i>'
    });
});
$$('.notification-custom').on('click', function () {
    myApp.addNotification({
        title: 'Event',
        subtitle: 'You missed an event at your place..',
        message: 'Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Aenean ut posuere erat. Aliquam sed molestie risus, quis tincidunt dui.',
        media: ''
    });
});
$$('.notification-callback').on('click', function () {
    myApp.addNotification({
        title: 'Album',
        subtitle: 'A Wonderful Myth',
        message: 'Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Aliquam sed molestie risus, quis tincidunt dui.',
        media: '',
        onClose: function () {
            myApp.alert('Callback activated. Notification closed');
        }
    });
});
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **notifications_iOS.html** file in your server root folder.
- Open this HTML file as http://localhost/notifications_iOS.html and the output is displayed as shown below.
- This example provides different types of notifications such as default notification, full layout notification, notification with custom message etc in the iOS layout.



Framework7 – Example Material

Description

Framework7 notifications can also be used in material layout.

Example

The following example demonstrates the use of material layout notifications in Framework7:

```
<!DOCTYPE html>
<html class="with-statusbar-overlay">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Notifications</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages navbar-fixed">
          <div data-page="home" class="page">
            <div class="navbar">
              <div class="navbar-inner">
```

```

        <div class="center">Notifications</div>
    </div>
</div>
<div class="page-content">
    <div class="content-block">
        <p><a href="#" class="button button-raised
notification-single">Single-line notification</a></p>
        <p><a href="#" class="button button-raised
notification-multi">Multi-line notification</a></p>
        <p><a href="#" class="button button-raised
notification-custom">With custom button</a></p>
        <p><a href="#" class="button button-raised
notification-callback">With callback on close</a></p>
    </div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7({
        material: true
    });
    var mainView = myApp.addView('.view-main');

    var $$ = Dom7;
    $$('.notification-single').on('click', function () {
        myApp.addNotification({
            message: 'Battery remaining only 20%'
        });
    });
    $$('.notification-multi').on('click', function () {
        myApp.addNotification({
            message: 'Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.',
        });
    });
    $$('.notification-custom').on('click', function () {

```

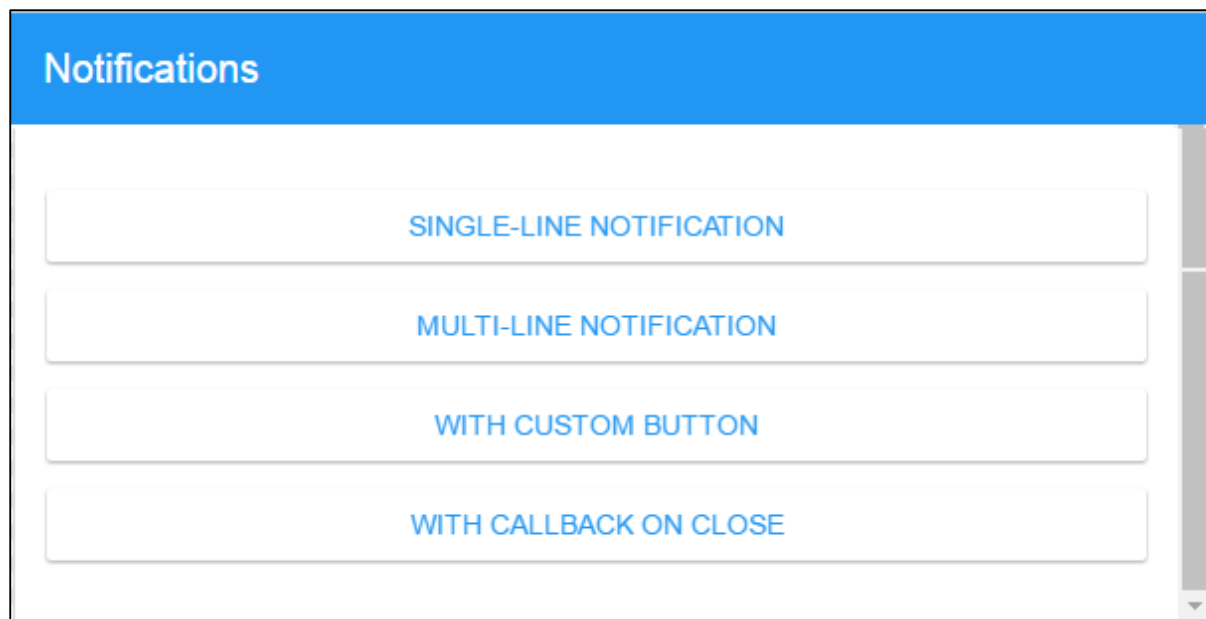


```
        myApp.addNotification({
            message: 'Nice teal color button',
            button: {
                text: 'Click me',
                color: 'teal'
            }
        });
    });
    $$('.notification-callback').on('click', function () {
        myApp.addNotification({
            message: 'Close this notification to see an alert',
            button: {
                text: 'Close Me',
                color: 'red'
            },
            onClose: function () {
                myApp.alert('Callback made.. Notification closed');
            }
        });
    });
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **notifications_material.html** file in your server root folder.
- Open this HTML file as http://localhost/notifications_material.html and the output is displayed as shown below.
- This example provides different types of notifications such as single line notification, multiline notification, notification with custom button etc. in the material layout.



31. Framework7 – Lazy Load

Description

Lazy loading delays your image loading process on a given page. Lazy loading improves scrolling performance, speeds page load and saves traffic.

Lazy load elements and images must be inside of scrollable `<div class="page-content">` to work properly.

The following table demonstrates the use of lazy load:

S. No.	Lazy load usage & Description
1	Usage The lazy load can be applied to images, background images and with fade-in effect.
2	Init Lazy Load Manually After initializing a page, if you add lazy load images manually then, lazy load will not work and you need to use methods to initialize it.

Framework7 – Lazy Load Usage

Description

The lazy load can be applied to images, background images and with fade-in effect as described below:

For images ``

To make use of lazy load on images follow the given steps:

- Use `data-src` attribute instead of `src` attribute to specify image source.
- Add class `lazy` to image.

```
<div class="page-content">
  ...
  
  ...
</div>
```

For Background Images

Lazy load can also be used for background images, for this case follow the given steps:

- Specify background image source into *data-background*.
- Add class *lazy* to image.

```
<div class="page-content">
  ...
  <div data-background="image_path.jpg" class="lazy">
    ...
  </div>
  ...
</div>
```

With Fade-in Effect

You can add fade-in effect to your images by using the *lazy-fadein* class to image/element.

```
<div class="page-content">
  ...
  
  <div data-background="image_path.jpg" class="lazy lazy-fadein">
    ...
  </div>
  ...
</div>
```

Framework7 – Init Lazy Load Manually

Description

After initializing a page, if you add lazy load images manually then, lazy load will not work and you need to use the following methods to initialize it:

- **myApp.initImagesLazyLoad(pageContainer)** - It is used to initialize the lazy loading on page, which accepts a required parameter as listed below:
 - **pageContainer** - It is a HTML Element or string of page containing lazy load images.
- **myApp.destroyImagesLazyLoad(pageContainer)** - It is used to destroy lazy loading on page.
 - **pageContainer** - It is an HTML Element or string of page containing lazy load images.

It is possible to trigger image loading manually by using *lazy* event on lazy image/element as shown below:

```
$$('img.lazy').trigger('lazy');
$$('div.lazy').trigger('lazy');
```

Example

The following example demonstrates the use of lazy loading in Framework7:

```
<!DOCTYPE html>
<html class="with-statusbar-overlay">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Lazy Load</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> </div>
                <div class="center">Lazy Load</div>
                <div class="right"> </div>
              </div>
            </div>
            <div class="page-content">
              <div class="content-block">
                <div class="content-block-inner">
                  <p> </p>
```

```
<p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi
ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in
voluptate velit esse cillum dolore eu fugiat nulla pariatur.</p>
```

```
<p> </p>
```

```
<p>Sed ut perspiciatis unde omnis iste natus error
sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa
quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt
explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut
fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi
nesciunt.</p>
```

```
<p> </p>
```

```
<p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi
ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in
voluptate velit esse cillum dolore eu fugiat nulla pariatur.</p>
```

```
<p> </p>
```

```
<p>Sed ut perspiciatis unde omnis iste natus error
sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa
quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt
explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut
fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi
nesciunt.</p>
```

```
<p> </p>
```

```
<p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi
ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in
voluptate velit esse cillum dolore eu fugiat nulla pariatur.</p>
```

```
<p> </p>
```

```
<p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Praesent laoreet nisl eget neque blandit lobortis. Sed
sagittis risus id vestibulum finibus. Cras vestibulum sem et massa hendrerit
maximus. Vestibulum suscipit tristique iaculis. Nam vitae risus non eros auctor
tincidunt quis vel nulla. Sed volutpat, libero ac blandit vehicula, est sem
gravida lectus, sed imperdiet sapien risus ut neque.</p>
```

```
<p><b>Using as background image:</b></p>
```

```
<div data-background="/framework7/images/pic7.jpg"
style="background: #aaa; height:60vw; background-size:cover" class="lazy lazy-
fadeIn"></div>
```

```
<p>Sed ut perspiciatis unde omnis iste natus error
sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa
quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt
```

```
explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut
fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi
nesciunt.</p>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
    <script>var myApp = new Framework7();</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **framework7_lazy_load.html** file in your server root folder.
- Open this HTML file as http://localhost/framework7_lazy_load.html and the output is displayed as shown below.
- The example specifies the lazy load of images where images will get loaded on the page when you scroll them down.



Framework7 Styling

32. Framework7 – Color Themes

Description

Framework7 provides different color themes for your applications.

A color theme provides different types of theme colors used to work with the applications smoothly as specified in the table below:

S. No.	Theme Type & Description
1	iOS Theme Colors You can use 10 different default iOS color themes for the application.
2	Material Theme Colors Framework7 provides 22 different default material color themes for the application.

Applying Color Themes

Framework7 allows you to apply color themes on different elements such as page, list-block, navbar, buttons-row etc by using the *theme-[color]* class to the parent element.

Example

```
...
</body>

<div class="page theme-gray">
  ...
</div>

<div class="list-block theme-blue">
  ...
</div>

<div class="navbar theme-green">
  ...
</div>

<div class="buttons-row theme-red">
  ...
</div>
```

Layout Themes

You can use default layout theme for your application by using two themes *white* and *dark*. The themes can be applied by using the *layout-[theme]* class to the parent element.

Example

```

    ...
</body>

<div class="navbar layout-white">
    ...
</div>

<div class="buttons-row layout-dark">
    ...
</div>

```

Helper Classes

Framework7 provides additional helper classes, which can be used outside or without themes as listed below:

- **color-[color]:** It can be used to change text color of block or color of button, link, icon etc.
- **bg-[color]:** It sets the predefined background color on the block or element.
- **border-[color]:** It sets the predefined border color on the block or element.

Framework7 – iOS Theme Colors

Description

You can use 10 different default iOS color themes for the application.

Example

The following example demonstrates the use of iOS color themes in Framework7:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Color Themes</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">

```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
  <div class="views">
    <div class="view view-main">
      <div class="pages">
        <div data-page="home" class="page navbar-fixed toolbar-fixed">
          <div class="navbar">
            <div class="navbar-inner">
              <div class="left"> <a href="#" class="link"><i
class="icon icon-back"></i><span>Back</span></a></div>
              <div class="center">Color Themes</div>
              <div class="right">
                <a href="#" class="link icon-only"><i class="icon
icon-bars"></i></a>
              </div>
            </div>
          </div>
          <div class="toolbar">
            <div class="toolbar-inner"><a href="#"
class="link">Left</a><a href="#" class="link">Right</a></div>
          </div>
          <div class="page-content">
            <div class="content-block">Lorem ipsum dolor sit amet, <a
href="#">consectetur</a> adipiscing elit, sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo <a
href="#">consequat</a>.</div>
            <div class="content-block">
              <p class="buttons-row">
                <a href="#tab1" class="tab-link active button">Tab
One</a><a href="#tab2" class="tab-link button">Tab Two</a><a href="#tab3"
class="tab-link button">Tab Three</a>
              </p>
            </div>
            <div class="content-block-title">Select Your Theme
Color</div>
            <div class="list-block">
              <ul>
                <li>

```

```

        <label class="label-radio item-content">
            <input type="radio" name="color-radio"
value="blue" checked>
                <div class="item-inner">
                    <div class="item-title">Blue</div>
                </div>
            </label>
        </li>
        <li>
            <label class="label-radio item-content">
                <input type="radio" name="color-radio"
value="gray">
                    <div class="item-inner">
                        <div class="item-title">Gray</div>
                    </div>
            </label>
        </li>
        <li>
            <label class="label-radio item-content">
                <input type="radio" name="color-radio"
value="lightblue">
                    <div class="item-inner">
                        <div class="item-title">Light Blue</div>
                    </div>
            </label>
        </li>
        <li>
            <label class="label-radio item-content">
                <input type="radio" name="color-radio"
value="orange">
                    <div class="item-inner">
                        <div class="item-title">Orange</div>
                    </div>
            </label>
        </li>
        <li>
            <label class="label-radio item-content">
                <input type="radio" name="color-radio"
value="pink">
                    <div class="item-inner">

```

```

        <div class="item-title">Pink</div>
    </div>
</label>
</li>
<li>
    <label class="label-radio item-content">
        <input type="radio" name="color-radio"
value="green">
        <div class="item-inner">
            <div class="item-title">Green</div>
        </div>
    </label>
</li>
<li>
    <label class="label-radio item-content">
        <input type="radio" name="color-radio"
value="red">
        <div class="item-inner">
            <div class="item-title">Red</div>
        </div>
    </label>
</li>
</ul>
</div>
<div class="content-block-title">Select Your Default
Layout Theme</div>
<div class="list-block">
    <ul>
        <li>
            <label class="label-radio item-content">
                <input type="radio" name="layout-radio"
value="" checked>
                <div class="item-inner">
                    <div class="item-title">Default</div>
                </div>
            </label>
        </li>
        <li>
            <label class="label-radio item-content">

```

```

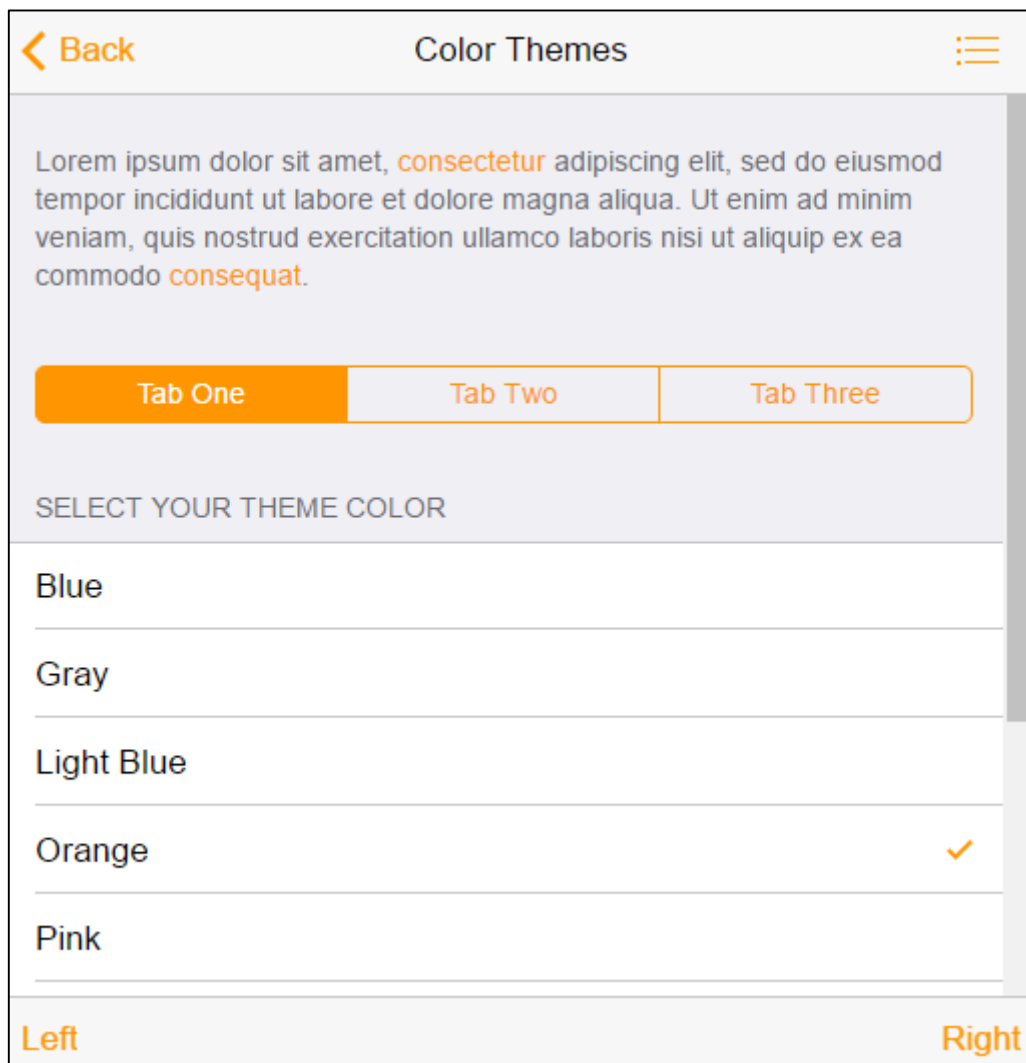
value="layout-dark">
    <input type="radio" name="layout-radio"
    <div class="item-inner">
        <div class="item-title">Dark</div>
    </div>
</label>
</li>
<li>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7();
    var $$ = Dom7;
    $$('input[name="color-radio"]').on('change', function () {
        if (this.checked) {
            $$('.view').removeClass('theme-pink theme-blue theme-red theme-gray
theme-orange theme-lightblue theme-green');
            $$('.view').addClass('theme-' + $(this).val());
        }
    });
    $$('input[name="layout-radio"]').on('change', function () {
        if (this.checked) {
            $$('.view').removeClass('layout-dark layout-white');
            $$('.view').addClass(this.value);
        }
    });
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **ios_theme_colors.html** file in your server root folder.
- Open this HTML file as http://localhost/ios_theme_colors.html and the output is displayed as shown below.
- The example provides 10 iOS color themes for your application. The default color is blue.



Framework7 – Material Theme Colors

Description

Framework7 provides 22 different material color themes, by default, for the application.

Example

The following example demonstrates the use of material color themes in Framework7:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Color Themes</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.ma
terial.colors.min.css">
  </head>
  <body>
    <div class="views">
      <div class="view view-main">
        <div class="pages">
          <div data-page="home" class="page navbar-fixed">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="left"> <a href="#" class="link icon-
only"><i class="icon icon-back"></i></a></div>
                <div class="center">Color Themes</div>
                <div class="right"> <a href="#" class="link icon-
only"><i class="icon icon-bars"></i></a></div>
              </div>
            </div>
            <a href="#" class="floating-button"><i class="icon icon-
plus"></i></a>
            <div class="page-content">
              <div class="content-block">Lorem ipsum dolor sit amet, <a
href="#">consectetur</a> adipiscing elit, sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo <a
href="#">consequat</a>.</div>
              <div class="content-block">
                <p class="buttons-row">
                  <a href="#tab1" class="tab-link active button">Tab
One</a><a href="#tab2" class="tab-link button">Tab Two</a><a href="#tab3"
class="tab-link button">Tab Three</a>

```

```

        </p>
        <p class="buttons-row">
            <a href="#" class="button button-raised
active">Button One</a><a href="#" class="button button-raised">Button Two</a><a
href="#" class="button button-raised">Button Three</a>
        </p>
        <p class="buttons-row">
            <a href="#" class="button button-raised button-fill
active">Button One</a><a href="#" class="button button-raised button-
fill">Button Two</a><a href="#" class="button button-raised button-fill">Button
Three</a>
        </p>
    </div>
    <div class="content-block-title">Select Your Theme
Color</div>
    <div class="list-block">
        <ul>
            <li>
                <label class="label-radio item-content">
                    <input type="radio" name="color-radio"
value="red">
                    <div class="item-inner">
                        <div class="item-title">Red</div>
                    </div>
                </label>
            </li>
            <li>
                <label class="label-radio item-content">
                    <input type="radio" name="color-radio"
value="pink">
                    <div class="item-inner">
                        <div class="item-title">Pink</div>
                    </div>
                </label>
            </li>
            <li>
                <label class="label-radio item-content">
                    <input type="radio" name="color-radio"
value="blue">
                    <div class="item-inner">
                        <div class="item-title">Blue</div>

```

```

        </div>
    </label>
</li>
<li>
    <label class="label-radio item-content">
        <input type="radio" name="color-radio"
value="lightblue">
        <div class="item-inner">
            <div class="item-title">Light Blue</div>
        </div>
    </label>
</li>
<li>
    <label class="label-radio item-content">
        <input type="radio" name="color-radio"
value="cyan">
        <div class="item-inner">
            <div class="item-title">Cyan</div>
        </div>
    </label>
</li>
<li>
    <label class="label-radio item-content">
        <input type="radio" name="color-radio"
value="green">
        <div class="item-inner">
            <div class="item-title">Green</div>
        </div>
    </label>
</li>
<li>
    <label class="label-radio item-content">
        <input type="radio" name="color-radio"
value="lime">
        <div class="item-inner">
            <div class="item-title">Lime</div>
        </div>
    </label>
</li>

```

```

        <li>
            <label class="label-radio item-content">
                <input type="radio" name="color-radio"
value="amber">
                <div class="item-inner">
                    <div class="item-title">Amber</div>
                </div>
            </label>
        </li>
        <li>
            <label class="label-radio item-content">
                <input type="radio" name="color-radio"
value="orange">
                <div class="item-inner">
                    <div class="item-title">Orange</div>
                </div>
            </label>
        </li>
        <li>
            <label class="label-radio item-content">
                <input type="radio" name="color-radio"
value="brown">
                <div class="item-inner">
                    <div class="item-title">Brown</div>
                </div>
            </label>
        </li>
        <li>
            <label class="label-radio item-content">
                <input type="radio" name="color-radio"
value="gray">
                <div class="item-inner">
                    <div class="item-title">Gray</div>
                </div>
            </label>
        </li>
        <li>
            <label class="label-radio item-content">
                <input type="radio" name="color-radio"
value="bluegray">

```

```

        <div class="item-inner">
            <div class="item-title">Blue Gray</div>
        </div>
    </label>
</li>
<li>
    <label class="label-radio item-content">
        <input type="radio" name="color-radio"
value="black">
        <div class="item-inner">
            <div class="item-title">Black</div>
        </div>
    </label>
</li>
</ul>
</div>
<div class="content-block-title">Select Your Default
Layout Theme</div>
<div class="list-block">
    <ul>
        <li>
            <label class="label-radio item-content">
                <input type="radio" name="layout-radio"
value="" checked>
                <div class="item-inner">
                    <div class="item-title">Default</div>
                </div>
            </label>
        </li>
        <li>
            <label class="label-radio item-content">
                <input type="radio" name="layout-radio"
value="layout-dark">
                <div class="item-inner">
                    <div class="item-title">Dark</div>
                </div>
            </label>
        </li>
    </ul>

```

```

        </div>
    </div>
</div>
</div>
</div>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script>
    var myApp = new Framework7({
        material: true
    });
    var $$ = Dom7;
    $$('input[name="color-radio"]').on('change', function () {
        if (this.checked) {
            var colors = 'theme- ' + 'red pink blue lightblue cyan green lime
amber orange brown gray bluegray black'.split(' ').join(' theme-');
            $$('.view').removeClass(colors);
            $$('.view').addClass('theme- ' + $(this).val());
        }
    });
    $$('input[name="layout-radio"]').on('change', function () {
        if (this.checked) {
            $$('.view').removeClass('layout-dark layout-white');
            $$('.view').addClass(this.value);
        }
    });
</script>
</body>
</html>

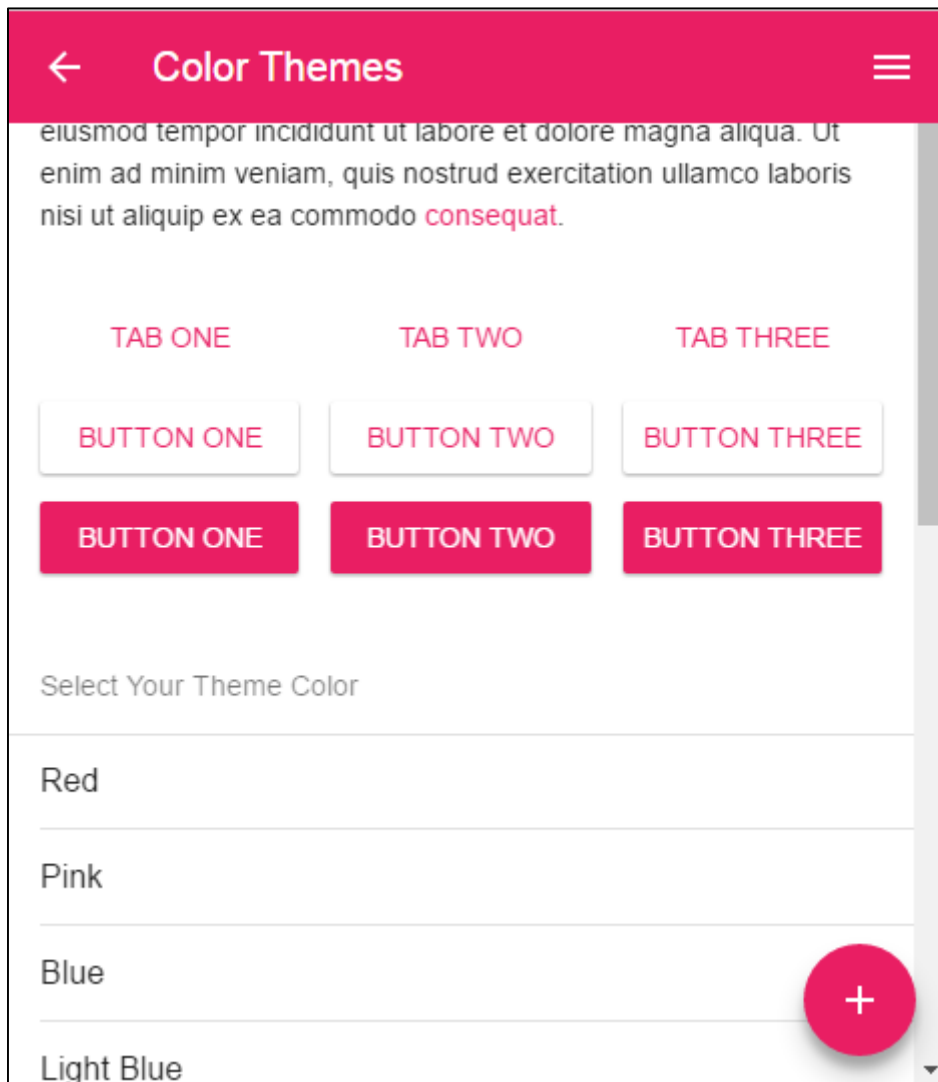
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **material_theme_colors.html** file in your server root folder.
- Open this HTML file as http://localhost/material_theme_colors.html and the output is displayed as shown below.

- The example provides 22 Material color themes for your application. Here we have used only some of the colors to show how they work in the material theme layout. The default color is blue.



33. Framework7 – Hairlines

Description

Hairline is a class that adds 1px border around the images by using the border class. With the release of 1.x, hairlines revised the working with *:after* and *:before* pseudo elements instead of using CSS borders.

Hairlines contains the following rules:

- **:after:** This pseudo element is used for *bottom* and *right* hairlines.
- **:before:** This pseudo element is used for *top* and *left* hairlines.

The following code snippet shows the use of *:after* element.

```
.navbar:after {  
  background-color: red;  
}
```

The following code snippet removes the bottom hairline toolbar:

```
.navbar:after {  
  display:none;  
}  
  
.toolbar:before {  
  display:none;  
}
```

"no-border" class

The hairline will be removed by using *no-border* class and it is supported on *Navbar*, *toolbar*, *card* and its elements.

The following code is used to remove hairline from navbar:

```
<div class="navbar no-border">  
  ...  
</div>
```


Framework7 Templates

34. Framework7 – Templates Overview

Description

Template7 is a lightweight, mobile-first JavaScript engine, which represents Ajax and dynamic pages as Template7 templates with specified context and does not require any additional scripts. Template7 is associated with Framework7 as a default, lightweight template engine, which works faster for applications.

Performance

The process of compiling string to JS function is the slowest segment of template7. Hence, you do not need to compile the template multiple times, only once is sufficient.

```
//Here initialize the app
var myApp = new Framework7();
// After initializing compile templates on app
var searchTemplate = $('script#search-template').html();
var compiledSearchTemplate = Template7.compile(searchTemplate);
var listTemplate = $('script#list-template').html();
var compiledListTemplate = Template7.compile(listTemplate);
// Execute the compiled templates with required context using onPageInit()
method
myApp.onPageInit('search', function (page) {
    // Execute the compiled templates with required content
    var html = compiledSearchTemplate({/*...some data...*/});
    // Do stuff with html
});
```

Template7 is a lightweight template engine used as a standalone library without Framework7. The Template7 files can be installed using two ways:

- You can download from Template7 [github repository](#).

Or

- You can install it using the following command via Bower:

```
bower install template7
```

35. Framework7 – Auto Compilation

Description

In Template7 you can compile your templates automatically by specifying special attributes in a `<script>` tag.

The following code shows the auto compilation layout:

```
<script type="text/template7" id="myTemplate">
  <p>Hello, my name is {{name}} and i am {{age}} years old</p>
</script>
```

You can use the following attributes for initializing the auto compilation:

- **type="text/template7"**: It is used to tell Template7 to auto compile and it is a required script type.
- **id="myTemplate"**: Template is accessible through the id and it is a required template id.

For automatic compilation, you need to enable app initialization by passing the following parameter:

```
var myApp = new Framework7({
  //It is used to compile templates on app init in Framework7
  precompileTemplates: true,
});
```

Template7.templates / myApp.templates

The automatically compiled templates can be accessed as properties of `Template7.templates` after initializing the app. It is also known as `myApp.templates` where `myApp` is an initialized instance of the app.

You can use the following templates in our index.html file:

```
<script type="text/template7" id="personTemplate">
  <p>Hello, my name is {{name}} and i am {{age}} years old</p>
  <p>I work as {{position}} at {{company}}</p>
</script>
<script type="text/template7" id="carTemplate">
  <p>I have a great car, it is {{vendor}} {{model}}, made in {{year}}
year.</p>
  <p>It has {{power}} hp engine with {{speed}} km/h maximum speed.</p>
```

```
</script>
```

You can also access templates in JavaScript *after* app initialization:

```
var myApp = new Framework7({
  //Tell Framework7 to compile templates on app init
  precompileTemplates: true
});
// Render person template to HTML, its template is already compiled and
// accessible as Template7.templates.personTemplate
var personHTML = Template7.templates.personTemplate({
  name: 'King Amit',
  age: 27,
  position: 'Developer',
  company: 'AngularJs'
});
// Compile car template to HTML, its template is already compiled and
// accessible as Template7.templates.carTemplate
var carHTML = Template7.templates.carTemplate({
  vendor: 'Honda',
  model: 'city',
  power: 1200hp,
  speed: 300
});
```

36. Framework7 – Template7 Pages

Description

Template7 is a mobile-first JavaScript template engine with [handlebars.js](#) like syntax. It is ultra lightweight and blazing-fast default template engine in Framework7.

First, we need to pass the following parameter on app initialization that renders all Ajax and dynamic pages as Template7 template:

```
var myApp = new Framework7({
  template7Pages: true // enable Template7 rendering for Ajax and Dynamic
  pages
});
```

S. No.	Template7 Pages - Usage & Description
1	Templates/Pages Data You can pass the required data/context for specific pages by specifying all pages data in <i>template7Data</i> parameter, sent on initializing an App.
2	Pass Custom Context Framework7 allows you to pass custom context to any dynamic page or any loaded Ajax.
3	Load Templates Directly You can render and load templates on fly as dynamic pages.
4	URL Query If you are using Template7 for rendering Ajax pages, its context always will be extended with special property <i>url_query</i> .

Templates/Pages Data

Description

You can pass the required data/context for specific pages by specifying all pages data in *template7Data* parameter sent on initializing an App as shown below:

```
// Initialize your app
var myApp = new Framework7({
  animateNavBackIcon: true,
  // Enable templates auto precompilation
  precompileTemplates: true,
  // Enabled rendering pages using Template7
```

```
template7Pages: true,
// Specify Template7 data for pages
template7Data: {
  'url:likes.html': {
    likes: [
      {
        title: 'Nelson Mandela',
        description: 'Champion of Freedom'
      },
      {
        title: 'Srinivasa Ramanujan',
        description: 'The Man Who Knew Infinity'
      },
      {
        title: 'James Cameron',
        description: 'Famous Filmmaker'
      }
    ]
  },
  // Will be applied for page/template with data-page="contacts"
  'page:contacts': {
    tel: '+1-541-754-3010',
    email: 'william@root.com',
    country: 'Canada',
  },
  // Just plain data object which can be passed for other pages using
  data-context-name attribute
  cars: [
    {
      brand: 'Audi',
      model: 'Q73',
      fuel_type: 'Diesel',
      top_speed: 234,
      seating: 7,
      color: 'white',
      price: 54800,
    },
    {
```

```
        brand: 'Porsche',
        model: 'Cayen',
        fuel_type: 'Diesel',
        top_speed: 230,
        seating: 5,
        color: 'black',
        price: 84800,
    },
    {
        brand: 'BMW',
        model: 'X6',
        fuel_type: 'Diesel',
        top_speed: 270,
        seating: 5,
        color: 'blue',
        price: 54400,
    },
    {
        brand: 'Toyota',
        model: 's3',
        fuel_type: 'Diesel',
        top_speed: 204,
        seating: 7,
        color: 'white',
        price: 45800,
    },
],
about: {
    firstname: 'William ',
    lastname: 'Root',
    age: 27,
    position: 'Developer',
    company: 'TechShell',
}
}
});
// Add main View
var mainView = myApp.addView('.view-main', {
```

```
// Enable dynamic Navbar
dynamicNavbar: true
});
```

You can access and modify any *template7Data* data in any moment. The *template7Data* is accessible in global object *Template7.data* or in its alias *myApp.template7Data*.

Page & URL Rules

The *page* and *url* rules are explained briefly below:

- When you specify the property name starting from **url:** then, this context will be used for the page.
- When you specify the property name starting from **page:** then, this context will be used with specified name of the page.

It is suggested to use **url:** as it provides accurate detection. However, in some cases, it is impossible to use **url:**, for instance, in pages with URLs having distinct GET parameters (like `contact.html?id=1` and `contact.html?id=2`) or for dynamic pages. In such a case use the **page:** rule.

Plain Data Objects

The plain data objects provide a way to handle your pages data in more complex and customizable manner. These objects allow you to pass custom context to loaded page by using links and *data-context-name* attribute.

Pass Custom Context

Description

Framework7 allows you to pass custom context to any dynamic page or any loaded Ajax.

The *data-context* attribute is used to pass custom context where you need to put valid stringified JSON as shown below:

```
<a href="interests.html" class="item-link item-content" data-context='{ "sport": "Baseball", "place": "Scotland", "movies": "Science Fiction", "actor": "Tom Cruise" }'>Interests</a>
```

You can also pass custom context using View's router *.load()* method and context property in JavaScript as shown in the following instance:

```
mainView.router.load({
  url: 'interests.html',
  context: {
    sport: 'Baseball',
    place: 'Scotland',
    movies: 'Science Fiction',
```



```

        actor: 'Tom Cruise'
    }
})

```

Load Templates Directly

Description

You can also render and load templates on fly as dynamic pages. The following code demonstrates the use of templates as dynamic pages:

index.html

```

<script type="text/template7" id="about">
  <div class="pages">
    <div data-page="about" class="page">
      <div class="page-content">
        <div class="content-block">
          <div class="content-block-inner">
            <p>Hello, i am <b>{{firstname}} {{lastname}}</b>, <b>{{age}}</b>
years old and working as <b>{{position}}</b> at <b>{{company}}</b>.</p>
          </div>
        </div>
      </div>
    </div>
  </div>
</script>

```

The required data is fetched from *template7Data*. To specify the required template, you need to use *data-template* attribute with template id.

You can use JavaScript View's *.load ()* method and *template* property to pass custom data as shown in the following instance:

```

mainView.router.load({
  template: Template7.templates.aboutTemplate,
  context: {
    firstname: 'William',
    age: 27
  }
})

```

Framework7 – URL Query

Description

If you are using Template7 for rendering Ajax pages, its context always will be extended with special property `url_query`, which contains URL GET arguments as shown below:

index.html

```
<a href="projects.html?pname=XYZ&duration=12months&cost=500000" class="item-link item-content">Project</a>
```

projects.html

```
<div class="page" data-page="person">
  <div class="page-content">
    <div class="content-block">
      <div class="content-block-inner">
        <p><b>Project working on :</b> {{url_query.pname}}</p>
        <p><b>Duration :</b> {{url_query.duration}}</p>
        <p><b>Cost :</b> {{url_query.cost}}</p>
      </div>
    </div>
  </div>
</div>
```

Example

The following example provides the links, which displays the user information such as user details, user likes, etc. when you click on those links.

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <title>Framework7</title>
```

```

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.min.css">

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/css/framework7.io
s.colors.min.css">
</head>
<body>
  <div class="views">
    <div class="view view-main">
      <div class="navbar">
        <div class="navbar-inner">
          <div class="center sliding">Template7 Pages</div>
        </div>
      </div>
      <div class="pages navbar-through toolbar-through">
        <div data-page="index" class="page">
          <div class="page-content">
            <div class="list-block">
              <ul>
                <li>
                  //plain data objects allow to pass custom context
to loaded page using 'data-context-name' attribute
                  <a href="#" data-template="about" data-context-
name="about" class="item-link item-content">
                    <div class="item-inner">
                      //provides link as 'About'
                      <div class="item-title">About</div>
                    </div>
                  </a>
                </li>
                <li>
                  //a context name for this link we pass context
path from template7Data root
                  <a href="/framework7/src/likes.html" class="item-
link item-content">
                    <div class="item-inner">
                      //provides link as 'Likes'
                      <div class="item-title">Likes</div>
                    </div>
                  </a>

```

```

        </li>
    </ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/template7" id="about">
    <div class="navbar">
        <div class="navbar-inner">
            <div class="left sliding"><a href="#" class="back link"> <i
class="icon icon-back"></i><span>Back</span></a></div>
            <div class="center sliding">About Me</div>
            <div class="right"><a href="#" class="link icon-only open-panel">
<i class="icon icon-bars"></i></a></div>
        </div>
    </div>
    <div class="pages">
        <div data-page="about" class="page">
            <div class="page-content">
                <div class="content-block">
                    <div class="content-block-inner">
                        //displays the details of the user by using the 'my-app.js'
file
                        <p>Hello, i am <b>{{firstname}} {{lastname}}</b>,
<b>{{age}}</b> years old and working as <b>{{position}}</b> at
<b>{{company}}</b>.</p>
                    </div>
                </div>
            </div>
        </div>
    </div>
</script>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/framework7/1.4.2/js/framework7.min.
js"></script>
<script type="text/javascript" src="/framework7/src/js/my-
app.js"></script>
</body>

```

```
</html>
```

my-app.js

```
// Initialize your app
var myApp = new Framework7({
  animateNavBackIcon: true,
  // Enable templates auto precompilation
  precompileTemplates: true,
  // Enabled rendering pages using Template7
  template7Pages: true,
  // Specify Template7 data for pages
  template7Data: {
    //provides the url for different page with data-page="likes"
    'url:likes.html': {
      likes: [
        {
          title: 'Nelson Mandela',
          description: 'Champion of Freedom'
        },
        {
          title: 'Srinivasa Ramanujan',
          description: 'The Man Who Knew Infinity'
        },
        {
          title: 'James Cameron',
          description: 'Famous Filmmaker'
        }
      ]
    },
    about: {
      firstname: 'William ',
      lastname: 'Root',
      age: 27,
      position: 'Developer',
      company: 'TechShell',
    }
  }
});

// Add main View
```

```

var mainView = myApp.addView('.view-main', {
  // Enable dynamic Navbar
  dynamicNavbar: true
});

```

likes.html

```

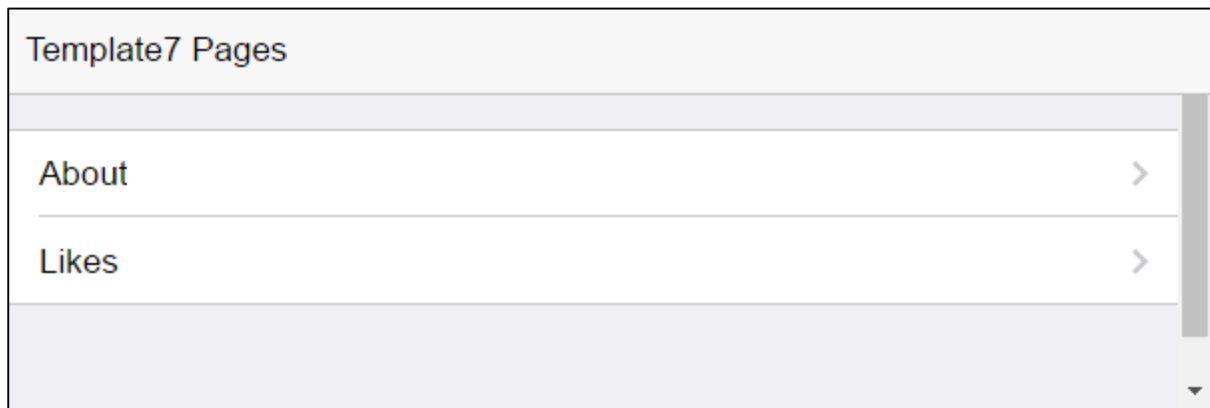
<div class="navbar">
  <div class="navbar-inner">
    <div class="left sliding"><a href="#" class="back link"> <i class="icon
icon-back"></i><span>Back</span></a></div>
    <div class="center sliding">Likes</div>
    <div class="right"><a href="#" class="link icon-only open-panel"> <i
class="icon icon-bars"></i></a></div>
  </div>
</div>
<div class="pages">
  <div data-page="likes" class="page">
    <div class="page-content">
      <div class="content-block-title">My Likes</div>
      <div class="list-block media-list">
        //iterate through likes
        <ul>{{#each likes}}
          <li class="item-content">
            <div class="item-inner">
              <div class="item-title-row">
                //displays the title and description by using the 'my-app.js' file
                <div class="item-title">{{title}}</div>
              </div>
              <div class="item-subtitle">{{description}}</div>
            </div>
          </li>{{/each}}
        </ul>
      </div>
    </div>
  </div>
</div>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given HTML code as **index.html** file in your server root folder.
- Open this HTML file as <http://localhost/index.html> and the output is displayed as shown below.
- The example provides the links, which displays the user information such as user details, user likes when you click on those links.



Framework7 Fast Clicks

37. Framework 7 – Active State

Description

When you tap the links and buttons, they are highlighted. This is done by *active state* in Framework7

- It behaves like a native app, not like a web app.
- It has a built-in *Fast clicks* library and it should be enabled.
- The *active-state* class is the same as the CSS *:active* selector.
- Active state is enabled by adding *watch-active-state* class to *<html>* element.

The following code is used for active state in CSS style:

```
/* Usual state */
.my-button {
    color: red;
}
/* Active/tapped state */
.my-button.active-state {
    color: blue;
}
```

The following code shows the fallback compatibility, when Active state or Fast clicks is disabled:

```
/* Usual state */
.my-button {
    color: red;
}
/* Active/tapped state */
.my-button.active-state {
    color: blue;
}
/* Fallback, when active state is disabled */
html:not(.watch-active-state) .my-button:active {
    color: blue;
}
```

38. Framework7 – Tap Hold Event

Description

The Tap hold event is used to trigger (enable) after a sustained and complete the touch event so only, it is called *tap hold* event. The *Tap Hold* is a built-in **Fast Clicks** library.

The following parameters are used to disable or enable and configured by default:

S. No.	Parameter & Description	Type	Default
1	tapHold To enable tap hold events when it is set to true.	<i>boolean</i>	<i>false</i>
2	tapHoldDelay It specifies the duration of holding the tap before triggering taphold event on the target element.	<i>number</i>	<i>750</i>
3	tapHoldPreventClicks The tap hold event will not be fired after clicking the event.	<i>boolean</i>	<i>true</i>

The following code is used for enable tap hold events:

```
var myApp = new Framework7({
  tapHold: true //enable tap hold events
});
var $$ = Dom7;
$$('.some-link').on('taphold', function () {
  myApp.alert('Tap hold fired!');
});
```

39. Framework7 – Touch Ripple

Description

Touch Ripple is an effect that is supported only in Framework7 material theme. By default, it is enabled in material theme and you can disable it by using the *materialRipple:false* parameter.

Ripple Elements

You can enable the ripple elements to match some CSS selectors such as:

- ripple
- a.link
- a.item-link
- .button
- .tab-link
- .label-radio
- .label-checkbox etc.

These are specified in the *materialRippleElements* parameter. You need to enable the *touch* ripple, add the element's selector to *materialRippleElements* parameter to make use of ripple effect, or just use the *ripple* class.

Ripple Wave Color

The color of the ripple can be changed on the element by adding the *ripple-[color]* class to the element.

For instance:

```
<a href="#" class="button ripple-orange">Ripple Button</a>
```

or you can define the ripple wave color by using the CSS as shown below:

```
.button .ripple-wave {  
    background-color: #FFFF00;  
}
```

Disable Ripple Elements

You can disable the ripple for some specified elements by adding the *no-ripple* class to those elements. For instance:

```
<a href="#" class="button no-ripple">Ripple Button</a>
```