Chapter 7

Bootstrap v5.x

Bootstrap 5 came into play back in May 2021. This release was launched after several iterations of alpha and beta releases and hence had a stack of major additions and new features. Some of the most talked about features are listed below.

- ✓ *Enhanced Grid System*
- ✓ *Enhanced Form elements*
- ✓ *Utilities API*
- ✓ *Support for IE chnaged*
- ✓ *Extended Colors*
- ✓ *Bootstrap Icons*

CSS

Copy-paste the stylesheet <link> into your <head> before all other stylesheets to load bootstrap CSS.

JS

- ✓ Many of bootstrap components require the use of JavaScript to function.
- ✓ Place **<script>** near the end of your pages, right before the closing </body> tag, to enable them.
- ✓ Both bootstrap.bundle.js and bootstrap.bundle.min.js include Popper for   tooltips   and popovers.

Bundle

```
<!—CSS Link -->
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" >

<!—bootstrap bundle js -->
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
```

Separate

If you decide to go with the separate scripts solution, Popper must come first (if you're using tooltips or popovers), and then JavaScript plugins.

```
<!—CSS Link -->
```

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" >


<!—popper js -->
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"></script>


<!—bootstrap js -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js"></script>
```

Bootstrap 5 vs. Bootstrap 4

This section talks about the upgrades that Bootstrap 5 provides over Bootstrap 4.

Enhanced Grid System

- ✓ The grid system is retained in Bootstrap 5.
- ✓ However, an extra grid tier **xxl** has been introduced to minimize the effort in making the pages responsive on higher resolution displays.
- ✓ Classes have been added to address vertical spacing.

Enhanced Form Elements

- ✓ The form elements in Bootstrap 4 have defaulted to the browser-provided view.
- ✓ However, in Bootstrap 5, the form elements have a custom design that enables them to have a consistent look and feel in all browsers.
- ✓ The new form controls are based on completely semantic, standard form controls. This helps developers to avoid adding extra markups for form controls.

Utilities API

- ✓ Unlike Bootstrap 4, Bootstrap 5 enables the developers to modify and create their own utilities.
- ✓ You can simply use sass to create your own utilities.
- ✓ You can use the state option to generate dummy class variations like hover and focus.

Support for IE

- ✓ Bootstrap 5 no longer supports Internet Explorer 10 and 11 like its predecessor, Bootstrap 4.

Extended Colors

- ✓ Bootstrap 4 had limited color options. But Bootstrap 5 has included many new color options to its color palette, enabling you to choose from the various shades available.

Bootstrap Icons

- ✓ Bootstrap 4 didn't have an SVG icon library. You had to use other third-Party libraries like Font Awesome to use icons in your applications.
- ✓ But Bootstrap 5 has taken care of this issue by introducing its own SVG library with 1000+ icons.

Floating Labels

- ✓ Bootstrap 5 adds support for floating labels in forms for the input fields. You can simply use the form-floating class to enable a floating label.
- ✓ When you enter some value into the input fields, they automatically adjust their position to their floated area.

RTL Support

- ✓ Bootstrap 5 has added support for RTL (Right-to-Left), which means you can develop content that needs writing from the right side of the page and continues to the left.
- ✓ As a result, websites in languages like Arabic, Sindhi, and Urdu can easily be developed.

Placeholder Components

- ✓ Bootstrap 5 allows loading placeholders in your pages.
- ✓ This means that you can utilize the space of the components by showing placeholders in their place while they're still loading the actual content.

jQuery Deprecation

- ✓ Bootstrap used jQuery from the very beginning as a dependency to offer dynamic features. With Bootstrap 5, jQuery is dropped, and Vanilla JS is introduced as its replacement.
- ✓ ***NOTE:*** *There are still JS dependencies that depend on Popper and Vanilla JS modules. However, jQuery is optional and can be added based on the requirement of the project.*

Other Significant Changes

- ✓ Bootstrap 5 introduces a new data attribute naming structure. **data-*** attribute has been renamed to **data-bs-*.**
- ✓ Bootstrap 5 now provides easier customizations on the pages and provides code snippets to build using Bootstrap's **Sass** files.

✓ Bootstrap 5 has added a new accordion component that includes **icons** that have a state and can be clicked. The **accordion-flush** class can be used to remove the default **background color, borders, or corners** to possibly render accordions edge-to-edge with the parent element.

✓ The **inline-block property is removed** in Bootstrap 5, and the **dropdown-menu-dark class now loads with a default black dropdown**.

✓ The support for **jumbotron** has been **terminated** in Bootstrap 5.

✓ **Font size is now set to rem instead of px**.

✓ Lastly, Bootstrap 5 also adds a new **offcanvas** component that enables you to create **hidden view panes** that would pop out when you interact with component tied to it -- e.g., side navigation bars, shopping carts, etc. Here is an example of the bottom offcanvas component.

✓ When you need a heading to stand out, consider using a **display heading**—a larger, slightly more opinionated heading style. Use **display-1 to display-6 in BS 5.**

Note:

✓ Bootstrap 4 used to do a great job providing all the flexibility and responsiveness for designing the web pages. However, Bootstrap 5 adds several new aspects that help to reduce the effort to do the same. But still, you need to keep certain parameters in mind before choosing either of these two.

✓ If you have an existing project that needs to support IE 10 and 11 or depends on jQuery, you should probably stick to Bootstrap 4. Bootstrap 5 can be overwhelming but might break your project structure if not migrated properly.

✓ But if you're starting a new project and want to provide a more immersive UI for the users, definitely opt for Bootstrap 5 to use its modern components, which this article discussed above.

BASIS OF	BOOTSTRAP 4	BOOTSTRAP 5
Grid System	It has 5 tier (xs, sm, md, lg, xl).	It has 6 tier (xs, sm, md, lg, xl, xxl).
Color	It has limited colors.	Extra colors added with the looks, A card improved color palette. there are various shades available to choose.

BASIS OF	BOOTSTRAP 4	BOOTSTRAP 5
Jquery	It has jquery and all related plugins.	Jquery is removed and switched to vanilla JS with some working plugins
Internet Explorer	Bootstrap 4 supports both IE 10 and 11.	Bootstrap 5 doesn't support IE 10 and 11.
Form elements	Radio buttons, checkboxes have different look in different OS and browsers. The form uses whatever default browsers provide.	The look of form elements will not change, on different OS or browser. The forms can be customized and form controls can be added, they would not depend on browser.
Utilities API	We cannot modify utilities in bootstrap 4	Bootstrap 5 gave freedom to modify and also create our own utility
Vertical Classes	Columns can be positioned relative	Columns cannot be positioned relative
Bootstrap Icons	Bootstrap 4 doesn't have its own SVG icons, we have to use font-awesome for icons.	Bootstrap 5 have its own SVG icons
Jumbotron	It supports.	It doesn't support jumbotron.
Navbar	We have inline-block property and we will get white dropdown as default for dropdown-menu-dark class.	Inline-block property is removed and we will get black dropdown as default for dropdown-menu-dark class.
flexbox grid	this makes easier to implement vertical designs, and the columns and rows can easily be implemented. the classes justify-	advanced grid system is made available , also columns don't have relative positions.

BASIS OF	BOOTSTRAP 4	BOOTSTRAP 5
	center-content can directly be used to align according to the requirement.	
RTL Support	It does not enable RTL(Right to Left) switching.	It enables RTL(Right to Left) switching.
Offcanvas Component	It does not support Offcanvas Component.	It supports Offcanvas Component(that is it is available now).

Navs

✓ Bootstrap provides an easy and quick way to create basic navigation as well as components like tabs and pills which are very flexible and elegant.
✓ All the Bootstrap's nav components, including tabs and pills, share the same base markup and styles through the base **.nav** class.

Creating Basic Nav with Bootstrap

You can use the Bootstrap .nav class to create a basic navigation menu, like this:

```
<ul class="nav">
 <li class="nav-item">
  <a class="nav-link active" href="#">Home</a>
 </li>
 <li class="nav-item">
  <a class="nav-link" href="#">Profile</a>
 </li>
 <li class="nav-item">
  <a class="nav-link" href="#">Messages</a>
 </li>
 <li class="nav-item">
  <a class="nav-link disabled" href="#">Reports</a>
 </li>
</ul>
```

Output

Home   Profile   Messages   Reports

Alignment of Nav Items

✓ By default, navs are left-aligned, but you can easily align them to **center** or **right** using flexbox utilities.
✓ The following example uses the class **.justify-content-center** to align nav items to **center**.

```
<ul class="nav justify-content-center">
    <li class="nav-item">
     <a class="nav-link" href="#">Home</a>
    </li>
    <li class="nav-item">
     <a class="nav-link" href="#">Profile</a>
```

```
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Messages</a>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled" href="#">Reports</a>
      </li>
    </ul>
```

Output

Home   Profile   Messages   Reports

Similarly, you can align nav items to **right** using the class **.justify-content-end**, like this:

```
<ul class="nav justify-content-end">
      <li class="nav-item">
        <a class="nav-link" href="#">Home</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Profile</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Messages</a>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled" href="#">Reports</a>
      </li>
</ul>
```

Output:

Home   Profile   Messages   Reports

Vertically stack nav items

✓ Moreover, you can even vertically stack your nav items by changing the flex item direction with the class **.flex-column**.

✓ Also, if you want to stack your nav vertically on some viewports but not on others, use it with **responsive** breakpoint (e.g., **.flex-sm-column**). If I add flex-sm-column then it will stack nav vertically till the small screen size(>= 576).

```
<ul class="nav flex-column">
```

```
    <li class="nav-item">
      <a class="nav-link" href="#">Home</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Profile</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Messages</a>
    </li>
    <li class="nav-item">
      <a class="nav-link disabled" href="#">Reports</a>
    </li>
</ul>
```

Output

Home

Profile

Messages

Reports

Creating the Basic Tabs

Simply, add the class **.nav-tabs** to the basic nav to generate a tabbed navigation, like this:

```
<ul class="nav nav-tabs">
    <li class="nav-item">
      <a class="nav-link active" href="#">Home</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Profile</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Messages</a>
    </li>
    <li class="nav-item">
      <a class="nav-link disabled" href="#">Reports</a>
    </li>
</ul>
```

Output

| Home | Profile | Messages | Reports |

Creating the Pills Nav

Similarly, you can create pill based navigation by adding the class **.nav-pills** on the basic nav instead of class .nav-tabs, as shown in the following example:

```
<ul class="nav nav-pills">
    <li class="nav-item">
      <a class="nav-link active" href="#">Home</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Profile</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Messages</a>
    </li>
    <li class="nav-item">
      <a class="nav-link disabled" href="#">Reports</a>
    </li>
</ul>
```
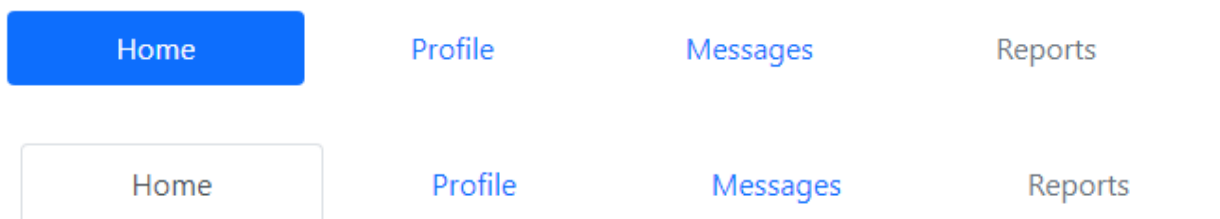
Output

Home   Profile   Messages   Reports

Justified Tabs/pills

Justify the tabs/pills with the .nav-justified class (equal width):

```
<ul class="nav nav-pills nav-justified">..</ul>
<ul class="nav nav-tabs nav-justified">..</ul>
```

Home   Profile   Messages   Reports

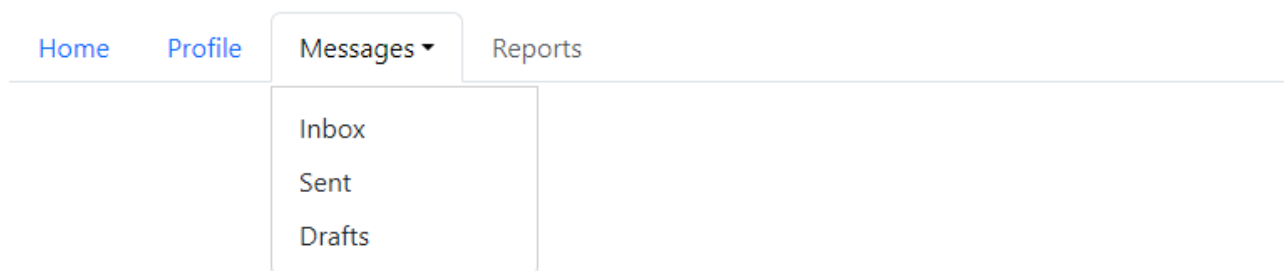Home   Profile   Messages   Reports

Bootstrap Nav with Dropdown Menus

✓ You can add dropdown menus to a link inside tabs and pills nav with a little extra markup.

✓ The four CSS classes **.dropdown, .dropdown-toggle, .dropdown-menu and .dropdown-item** are required in addition to the .nav, .nav-tabs or .nav-pills classes to create a simple dropdown menu inside tabs and pills nav without using any JavaScript code.

Creating Tabs with Dropdowns

The following example will show you how to add simple dropdown menu to a tab.

```
<ul class="nav nav-tabs">
  <li class="nav-item">
   <a class="nav-link" href="#">Home</a>
  </li>
  <li class="nav-item">
   <a class="nav-link" href="#">Profile</a>
  </li>
  <li class="nav-item dropdown">
   <a class="nav-link dropdown-toggle active" data-bs-toggle="dropdown"
href="#">Messages</a>
     <ul class="dropdown-menu">
       <li><a class="dropdown-item" href="#">Inbox</a></li>
       <li><a class="dropdown-item" href="#">Sent</a></li>
       <li><a class="dropdown-item" href="#">Drafts</a></li>
     </ul>
  </li>
  <li class="nav-item">
   <a class="nav-link disabled" href="#">Reports</a>
  </li>
</ul>
```

Output

Creating Pills with Dropdowns

The following example will show you how to add simple dropdown menu to a pill nav.

```html
<ul class="nav nav-pills">
  <li class="nav-item">
    <a class="nav-link active" href="#">Home</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Profile</a>
  </li>
  <li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" data-bs-toggle="dropdown" href="#">Messages</a>
    <ul class="dropdown-menu">
      <li><a class="dropdown-item" href="#">Inbox</a></li>
      <li><a class="dropdown-item" href="#">Sent</a></li>
      <li><a class="dropdown-item" href="#">Drafts</a></li>
    </ul>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Reports</a>
  </li>
</ul>
```
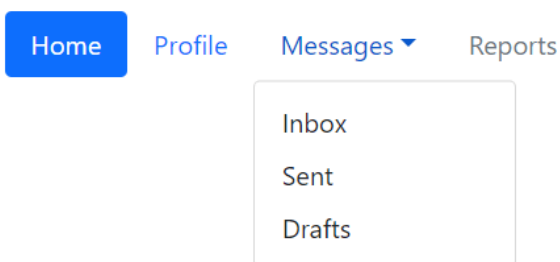
Output

Toggleable / Dynamic Tabs

✓ To make the tabs toggleable, add the **data-toggle="tab"** attribute to each link. Then add a **.tab-pane** class with a unique ID for every tab and wrap them inside a <div> element with class **.tab-content**.

✓ If you want the tabs to fade in and out when clicking on them, add the .fade class to .tab-pane:

```
<ul class="nav nav-tabs">
    <li class="nav-item">
     <a class="nav-link active" data-bs-toggle="tab" href="#home">Home</a>
    </li>
    <li class="nav-item">
     <a class="nav-link" data-bs-toggle="tab" href="#profile">Profile</a>
    </li>
    <li class="nav-item dropdown">
     <a class="nav-link" data-bs-toggle="tab" href="#msg">Messages</a>
    </li>
</ul>
<div class="tab-content">
  <div class="tab-pane container active" id="home">
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae cupiditate consequatur
voluptatem aspernatur consectetur necessitatibus recusandae maiores ex fugiat aliquid
repellat maxime dolor, asperiores distinctio cumque non magni cum assumenda!
  </div>
  <div class="tab-pane container fade" id="profile">
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae cupiditate consequatur
voluptatem aspernatur consectetur necessitatibus recusandae maiores ex fugiat aliquid
repellat maxime dolor, asperiores distinctio cumque non magni cum assumenda!
  </div>
  <div class="tab-pane container fade" id="msg">
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae cupiditate consequatur
voluptatem aspernatur consectetur necessitatibus recusandae maiores ex fugiat aliquid
repellat maxime dolor, asperiores distinctio cumque non magni cum assumenda!
  </div>
 </div>
```

| Home | Profile | Messages |

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae cupiditate consequatur voluptatem aspernatur consectetur necessitatibus recusandae maiores ex fugiat aliquid repellat maxime dolor, asperiores distinctio cumque non magni cum assumenda!

Toggleable / Dynamic Pills

The same code applies to pills; only change the data-toggle attribute to data-toggle="pill". Add nav-pills class.

```
<ul class="nav nav-pills">
     <li class="nav-item">
      <a class="nav-link active" data-bs-toggle="pill" href="#home">Home</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" data-bs-toggle="pill" href="#profile">Profile</a>
     </li>
     <li class="nav-item dropdown">
      <a class="nav-link" data-bs-toggle="pill" href="#msg">Messages</a>
     </li>
</ul>
<div class="tab-content">
  <div class="tab-pane container active" id="home">
     Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae cupiditate consequatur
voluptatem aspernatur consectetur necessitatibus recusandae maiores ex fugiat aliquid
repellat maxime dolor, asperiores distinctio cumque non magni cum assumenda!
  </div>
  <div class="tab-pane container fade" id="profile">
     Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae cupiditate consequatur
voluptatem aspernatur consectetur necessitatibus recusandae maiores ex fugiat aliquid
repellat maxime dolor, asperiores distinctio cumque non magni cum assumenda!
  </div>
  <div class="tab-pane container fade" id="msg">
     Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae cupiditate consequatur
voluptatem aspernatur consectetur necessitatibus recusandae maiores ex fugiat aliquid
repellat maxime dolor, asperiores distinctio cumque non magni cum assumenda!
  </div>
 </div>
```

Home   Profile   Messages

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae cupiditate consequatur voluptatem aspernatur consectetur necessitatibus recusandae maiores ex fugiat aliquid repellat maxime dolor, asperiores distinctio cumque non magni cum assumenda!

Navbar

You can use the Bootstrap navbar component to create responsive navigation header for your website or application.

✓ These responsive navbar will be collapsed on devices having small viewports like mobile phones but expand when user click the toggle button.
✓ However, it will be horizontal as normal on the medium and large devices such as laptop or desktop.
✓ You can also create different variations of the navbar such as navbars with dropdown menus and search boxes as well as fixed positioned navbar with much less effort.
✓ The following example will show you how to create a simple static navbar with navigation links.
✓ A navigation bar is a navigation header that is placed at the top of the page:

Basic Navbar

✓ With Bootstrap, a navigation bar can extend or collapse, depending on the screen size.
✓ A standard navigation bar is created with the .navbar class, followed by a responsive collapsing class: .navbar-expand-xxl|xl|lg|md|sm (stacks the navbar vertically on xxlarge, extra large, large, medium or small screens).
✓ To add links inside the navbar, use either an <ul> element (or a <div>) with class="navbar-nav". Then add <li> elements with a .nav-item class followed by an <a> element with a .nav-link class:

```
<!-- A grey horizontal navbar that becomes vertical on small screens -->
<nav class="navbar navbar-expand-sm bg-warning">
 <div class="container-fluid">
   <!-- Links -->
   <ul class="navbar-nav">
    <li class="nav-item">
     <a class="nav-link" href="#">Link 1</a>
    </li>
    <li class="nav-item">
     <a class="nav-link" href="#">Link 2</a>
    </li>
   </ul>
 </div>
</nav>
```

For screen size >= 576

> Link 1   Link 2

For screen size <576

> Link 1
>
> Link 2

Vertical Navbar

Remove the .navbar-expand-* class to create a navigation bar that will always be vertical:

```
<nav class="navbar bg-warning">
</nav>
```

> Link 1
>
> Link 2

Centered/Right aligned Navbar

✓ **Add the .justify-content-center** class to center the navigation bar:
✓ **Add the .justify-content-end** class to right align the navigation bar:

Colored Navbar

Use any of the .bg-color classes to change the background color of the navbar (.bg-primary, .bg-success, .bg-info, .bg-warning, .bg-danger, .bg-secondary, .bg-dark and .bg-light)

Note: Add a **white** text color to all links in the navbar with the **.navbar-dark** class, or use the .**navbar-light** class to add a **black** text color.

Active/disabled state:

Add the **.active** class to an <a> element to highlight the current link, or the **.disabled** class to indicate that the link is un-clickable.

Navbar Text

Use the .navbar-text class to vertical align any elements inside the navbar that are not links (ensures proper padding and text color).

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <div class="container-fluid">
    <span class="navbar-text">Navbar text</span>
  </div>
</nav>
```

Navbar text

Brand / Logo

The .navbar-brand class is used to highlight the brand/logo/project name of your page:

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Logo</a>
  </div>
</nav>
```

Adding Logo Images to Navbars

You can also place your logo image inside the navbar, instead of plain text. However, you need to set the logo height manually to fit it properly inside the navbar, as shown here:

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">
      <img src="logo.png" alt="Avatar Logo" style="width:40px;" class="rounded-pill">
    </a>
  </div>
</nav>
```

Collapsible Navbar

- ✓ Very often, especially on small screens, we want to hide the navigation links and replace them with a button that should reveal them when clicked on.
- ✓ To create a collapsible navigation bar, use a button with class="navbar-toggler", data-bs-toggle="collapse" and data-bs-target="#*thetarget*".
- ✓ Then wrap the navbar content (links, etc) inside a <div> element with class="collapse navbar-collapse", followed by an id that matches the data-bs-target of the button: "*thetarget*".

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
 <div class="container-fluid">
   <a class="navbar-brand" href="#">Logo</a>
< !—To add hamburger icon-->
   <button class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#nav1">
     <span class="navbar-toggler-icon"></span>
   </button>

<div class="collapse navbar-collapse" id="nav1">
    <ul class="navbar-nav">
     <li class="nav-item">
       <a class="nav-link" href="#">Link</a>
     </li>
     <li class="nav-item">
       <a class="nav-link" href="#">Link</a>
     </li>
     <li class="nav-item">
       <a class="nav-link" href="#">Link</a>
     </li>
    </ul>
   </div>
 </div>
</nav>
```

Collapsible Navbar With Dropdown

Navbars can also hold dropdown menus:

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
    <div class="container-fluid">
      <!—To add logo -- >
      <a class="navbar-brand" href="#"><img class="rounded-pill" width="50"
src="s.png"></a>
      <!—To add collapsible button -- >
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#nav1">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="nav1">
       <ul class="navbar-nav">
        <li class="nav-item">
         <a class="nav-link" href="#">Link 1</a>
        </li>
       <!—To add dropdown -- >
        <li class="nav-item dropdown">
         <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-
toggle="dropdown">Dropdown</a>
          <ul class="dropdown-menu">
           <li><a class="dropdown-item" href="#">Link 1</a></li>
           <li><a class="dropdown-item" href="#">link 2</a></li>
           <li><a class="dropdown-item" href="#">link 3</a></li>
          </ul>
        </li>
        <li class="nav-item">
         <a class="nav-link" href="#">Link 2</a>
        </li>
       </ul>
      </div>
    </div>
  </nav>
```
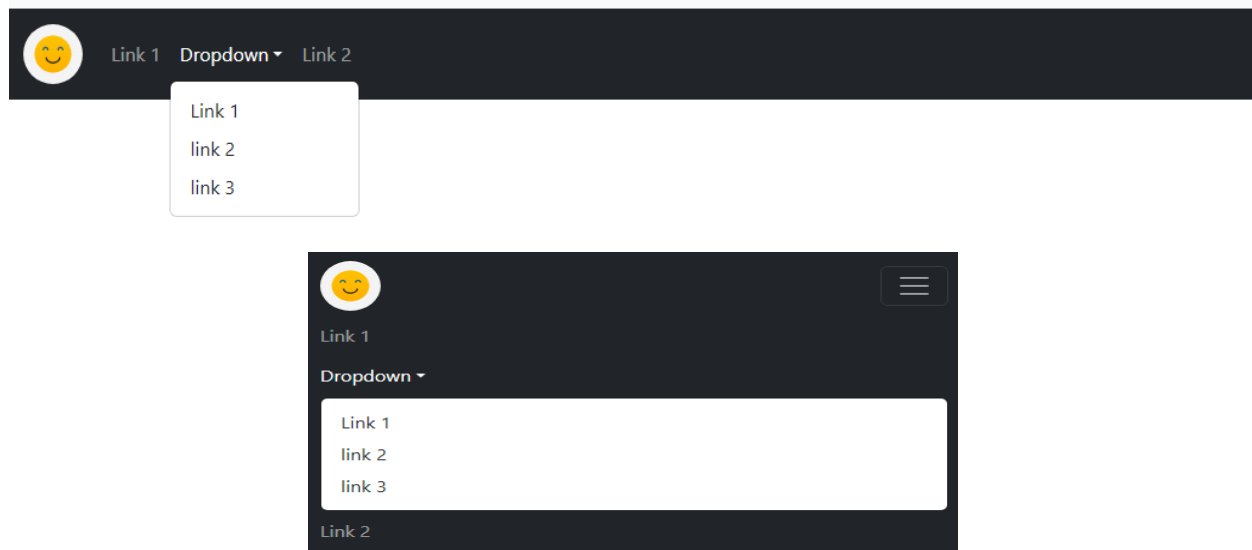
Fixed Navigation Bar

- ✓ The navigation bar can also be fixed at the top or at the bottom of the page.
- ✓ A fixed navigation bar stays visible in a fixed position (top or bottom) independent of the page scroll.
- ✓ The .fixed-top class makes the navigation bar fixed at the **top**:

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark fixed-top">
</nav>
```

Use the .fixed-bottom class to make the navbar stay at the **bottom** of the page:

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark fixed-bottom"> </nav>
```

Use the .sticky-top class to make the navbar fixed/stay at the **top** of the page when you scroll **past** it.

Note: This class does not work in IE11 and earlier (will treat it as position:relative).

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark sticky-top">
</nav>
```

Scrollspy

✓ Bootstrap 5 Scrollspy is an automatic navigation mechanism that automatically updates the scroll position.

✓ The Bootstrap scrollspy is a navigation mechanism that automatically highlights the nav links based on the scroll position to indicate the visitor where they are currently on the page. When we click on the link it automatically scrolls down to the currently active element.

✓ The scrollspy will make your web page more elegant and accessible, if you're using the bookmark links for directing the visitors to the different sections of a page that has a huge amount of content.

How it works:
• First, you need to add a ***data-bs-spy="scroll"*** attribute to the element that should be used for the scrollable area.
• Next, you need to add a ***data-bs-target*** attribute to the navigation menu, which should contain the ID of the element that should be used as the scrollable area.
• Finally, you need to add a ***href*** attribute to each menu item that corresponds to the ID of a section on the page.

Syntax:
```
<tag data-bs-spy="scroll" data-bs-target="class/id">
content...
</tag>
```

```
<body data-bs-spy="scroll" data-bs-target=".navbar" data-bs-offset="100">

  <nav class="navbar navbar-expand-sm bg-dark navbar-dark fixed-top">
   <div class="container-fluid">
    <ul class="navbar-nav">
     <li class="nav-item">
      <a class="nav-link" href="#section1">Section 1</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="#section2">Section 2</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="#section3">Section 3</a>
     </li>
    </ul>
```

```
    </div>
  </nav>

  <div id="section1" class="container-fluid bg-danger text-white" style="padding:250px
20px;">
    <h1>Section 1</h1>
    <p>Try to scroll this section and look at the navigation bar while scrolling! Try to scroll this
section and look at the navigation bar while scrolling!</p>
    <p>Try to scroll this section and look at the navigation bar while scrolling! Try to scroll this
section and look at the navigation bar while scrolling!</p>
  </div>

  <div id="section2" class="container-fluid bg-dark text-white" style="padding:250px 20px;">
    <h1>Section 2</h1>
    <p>Try to scroll this section and look at the navigation bar while scrolling! Try to scroll this
section and look at the navigation bar while scrolling!</p>
    <p>Try to scroll this section and look at the navigation bar while scrolling! Try to scroll this
section and look at the navigation bar while scrolling!</p>
  </div>

  <div id="section3" class="container-fluid bg-secondary text-white" >
    <h1>Section 3</h1>
    <p>Try to scroll this section and look at the navigation bar while scrolling! Try to scroll this
section and look at the navigation bar while scrolling!</p>
    <p>Try to scroll this section and look at the navigation bar while scrolling! Try to scroll this
section and look at the navigation bar while scrolling!</p>
  </div>
  </body>
```
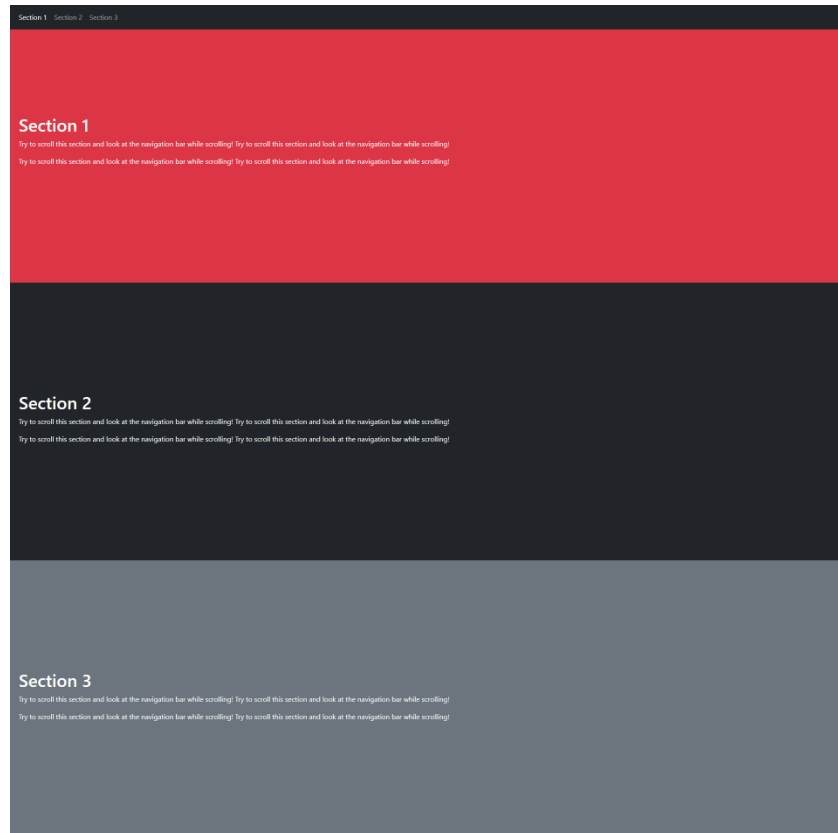
- ✓ Add data-bs-spy="scroll" to the element that should be used as the scrollable area (often this is the <body> element).
- ✓ Then add the data-bs-target attribute with a value of the id or the class name of the navigation bar (.navbar). This is to make sure that the navbar is connected with the scrollable area.
- ✓ Note that scrollable elements must match the ID of the links inside the navbar's list items (<div id="section1"> matches <a href="#section1">).
- ✓ The optional data-bs-offset attribute specifies the number of pixels to offset from top when calculating the position of scroll. This is useful when you feel that the links inside the navbar changes the active state too soon or too early when jumping to the scrollable elements. Default is 10 pixels.

Flex

- ✓ Designing flexible, responsive layout structures without utilizing float or positioning is made simpler with the **Flexible Box Layout Module**.
- ✓ The biggest difference between Bootstrap 3 and Bootstrap 4 & 5 is that Bootstrap 5 now uses flexbox, instead of floats, to handle the layout.
- ✓ The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.
  - ✓ Quickly control the layout,
  - ✓ alignment, and scaling of components,
  - ✓ grid columns,
  - ✓ navigation,
  - ✓ more using a comprehensive set of responsive flexbox utilities.

Basic Flex Function

The "d-flex" class create a basic flex box for handle layout. We can contain information, functions and other application data.

To create a flexbox container and to transform direct children into flex items, use the d-flex class:

```
<div class="d-flex p-3 bg-dark text-white">
   <div class="p-2 bg-info">Flex item 1</div>
   <div class="p-2 bg-warning">Flex item 2</div>
   <div class="p-2 bg-danger">Flex item 3</div>
 </div>
```


Inline Flex

To create an inline flexbox container, use the d-inline-flex class:

```
<div class="d-inline-flex p-3 bg-dark text-white">
   <div class="p-2 bg-info">Flex item 1</div>
   <div class="p-2 bg-warning">Flex item 2</div>
   <div class="p-2 bg-danger">Flex item 3</div>
 </div>
```

Horizontal Direction

- ✓ The "d-flex" class create a flex with the horizontal direction of the data. The "**flex-row**" class creates flex in horizontal format with its value.
- ✓ If we want to display the value in the horizontally reverse format, then use the "**flex-row-reverse**" class.
- ✓ Use .flex-row to display the flex items horizontally (side by side). This is default.

```
<div class="d-flex flex-row p-3 bg-dark text-white">
   <div class="p-2 bg-info">Flex item 1</div>
   <div class="p-2 bg-warning">Flex item 2</div>
   <div class="p-2 bg-danger">Flex item 3</div>
 </div>
```



```
<div class="d-flex flex-row-reverse p-3 bg-dark text-white">
   <div class="p-2 bg-info">Flex item 1</div>
   <div class="p-2 bg-warning">Flex item 2</div>
   <div class="p-2 bg-danger">Flex item 3</div>
 </div>
```


Vertical Direction

- ✓ The "**flex-column**" class creates a flex box in vertical format with value.
- ✓ If we want to display the value in the vertically reverse format, then use the "**flex-column-reverse**" class.

```
<div class="d-flex flex-column p-3 bg-dark text-white">
   <div class="p-2 bg-info">Flex item 1</div>
   <div class="p-2 bg-warning">Flex item 2</div>
   <div class="p-2 bg-danger">Flex item 3</div>
 </div>
<div class="d-flex flex-column-reverse p-3 bg-success text-white">
   <div class="p-2 bg-dark">Flex item 1</div>
   <div class="p-2 bg-warning">Flex item 2</div>
   <div class="p-2 bg-danger">Flex item 3</div>
```
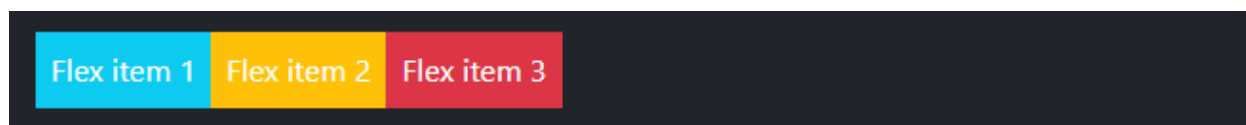
```
    </div>
```


Justify Content

Use the .justify-content-* classes to change the alignment of flex items.
Valid classes are start (default), end, center, between or around:

```
<div class="container mt-3">
  <Strong>justify-content-start</Strong>
  <div class="d-flex justify-content-start bg-warning mb-3">
   <div class="p-2 bg-dark text-white">Flex item 1</div>
   <div class="p-2 bg-success text-white">Flex item 2</div>
   <div class="p-2 bg-danger text-white">Flex item 3</div>
  </div>
  <Strong>justify-content-end</Strong>
  <div class="d-flex justify-content-end bg-warning mb-3">
    <div class="p-2 bg-dark text-white">Flex item 1</div>
    <div class="p-2 bg-success text-white">Flex item 2</div>
    <div class="p-2 bg-danger text-white">Flex item 3</div>
  </div>
  <Strong>justify-content-center</Strong>
  <div class="d-flex justify-content-center bg-warning mb-3">
    <div class="p-2 bg-dark text-white">Flex item 1</div>
    <div class="p-2 bg-success text-white">Flex item 2</div>
    <div class="p-2 bg-danger text-white">Flex item 3</div>
  </div>
  <Strong>justify-content-between</Strong>
  <div class="d-flex justify-content-between bg-warning mb-3">
```

```
      <div class="p-2 bg-dark text-white">Flex item 1</div>
      <div class="p-2 bg-success text-white">Flex item 2</div>
      <div class="p-2 bg-danger text-white">Flex item 3</div>
  </div>
  <Strong>justify-content-around</Strong>
  <div class="d-flex justify-content-around bg-warning mb-3">
      <div class="p-2 bg-dark text-white">Flex item 1</div>
      <div class="p-2 bg-success text-white">Flex item 2</div>
      <div class="p-2 bg-danger text-white">Flex item 3</div>
  </div>
 </div>
```
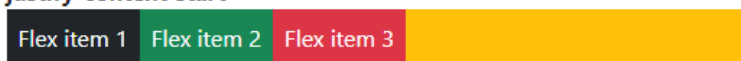
justify-content-start

| Flex item 1 | Flex item 2 | Flex item 3 | |

justify-content-end

| | Flex item 1 | Flex item 2 | Flex item 3 |

justify-content-center

| | Flex item 1 | Flex item 2 | Flex item 3 | |

justify-content-between

| Flex item 1 | Flex item 2 | Flex item 3 |

justify-content-around

| Flex item 1 | Flex item 2 | Flex item 3 |

Fill / Equal Widths

Use .flex-fill on flex items to force them into equal widths:

```
 <div class="d-flex">
   <div class="p-2 bg-danger flex-fill">Flex item 1</div>
   <div class="p-2 bg-warning flex-fill">Flex item 2</div>
   <div class="p-2 bg-primary flex-fill">Flex item 3</div>
 </div>
```

| Flex item 1 | Flex item 2 | Flex item 3 |

Grow

Use .flex-grow-1 on a flex item **to take up the rest of the space**.

In the example below, the first and last flex items take up their necessary space, while the **item 2 in the center** takes up the rest of the available space:

```
<div class="d-flex">
  <div class="p-2 bg-danger">Flex item 1</div>
  <div class="p-2 bg-warning flex-grow-1">Flex item 2</div>
  <div class="p-2 bg-primary">Flex item 3</div>
</div>
```

| Flex item 1 | Flex item 2 | | Flex item 3 |

Use .flex-shrink-1 on a flex item to make it shrink if necessary.

Order

✓ Change the visual order of a specific flex item(s) with the .order class.
✓ Valid classes are from **0 to 5**, where the lowest number has highest priority (order-1 is shown before order-2, etc..):

```
<div class="d-flex bg-dark">
  <div class="p-2 bg-danger order-3 text-white">Flex item 1</div>
  <div class="p-2 bg-warning order-1 text-white">Flex item 2</div>
  <div class="p-2 bg-primary order-2 text-white">Flex item 3</div>
</div>
```
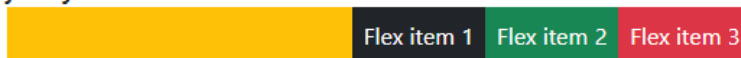
➤ Flex item 2 = order-1,

➤ Flex item 3 = order-2,

➤ Flex item 1 = order-3

| Flex item 2 | Flex item 3 | Flex item 1 | |

Auto Margins

Easily add auto margins to flex items with .ms-auto (push items **to the right**) and .me-auto (push items **to the left**):

```
<div class="d-flex bg-dark">
  <div class="p-2 bg-danger text-white me-auto">Flex item 1</div>
  <div class="p-2 bg-warning text-white">Flex item 2</div>
  <div class="p-2 bg-primary text-white">Flex item 3</div>
</div>
```

```
<div class="d-flex bg-dark">
   <div class="p-2 bg-danger text-white">Flex item 1</div>
   <div class="p-2 bg-warning text-white">Flex item 2</div>
   <div class="p-2 bg-primary text-white ms-auto">Flex item 3</div>
  </div>
```
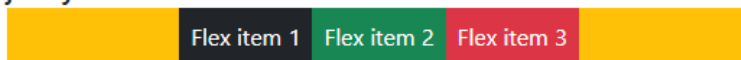

Wrap

Control how flex items wrap in a flex container with .flex-nowrap (default), .flex-wrap or .flex-wrap-reverse.

```
<p>.flex-wrap:</p>
 <div class="d-flex flex-wrap bg-light">
   <div class="p-4 border bg-danger text-white">Flex item 1</div>
   <div class="p-4 border bg-dark text-white">Flex item 1</div>
   <div class="p-4 border bg-danger text-white">Flex item 3</div>
   <div class="p-4 border bg-dark text-white">Flex item 4</div>
   <div class="p-4 border bg-danger text-white">Flex item 5</div>
   <div class="p-4 border bg-dark text-white">Flex item 6</div>
   <div class="p-4 border bg-danger text-white">Flex item 7</div>
   <div class="p-4 border bg-dark text-white">Flex item 8</div>
   <div class="p-4 border bg-danger text-white">Flex item 9</div>
   <div class="p-4 border bg-dark text-white">Flex item 10</div>
   <div class="p-4 border bg-danger text-white">Flex item 11</div>
   <div class="p-4 border bg-dark text-white">Flex item 12</div>
   <div class="p-4 border bg-danger text-white">Flex item 13</div>
   <div class="p-4 border bg-dark text-white">Flex item 14</div>
   <div class="p-4 border bg-danger text-white">Flex item 15</div>
   <div class="p-4 border bg-dark text-white">Flex item 16</div>

 </div>
 <br>
 <p>.flex-wrap-reverse:</p>
 <div class="d-flex flex-wrap-reverse bg-light">
   <div class="p-4 border bg-danger text-white">Flex item 1</div>
   <div class="p-4 border bg-dark text-white">Flex item 1</div>
```
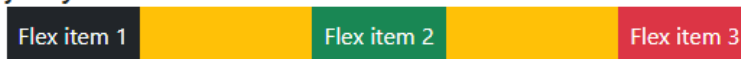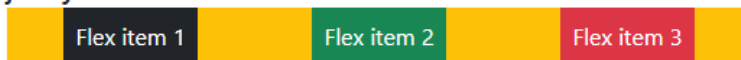
```
  <div class="p-4 border bg-danger text-white">Flex item 3</div>
  <div class="p-4 border bg-dark text-white">Flex item 4</div>
  <div class="p-4 border bg-danger text-white">Flex item 5</div>
  <div class="p-4 border bg-dark text-white">Flex item 6</div>
  <div class="p-4 border bg-danger text-white">Flex item 7</div>
  <div class="p-4 border bg-dark text-white">Flex item 8</div>
  <div class="p-4 border bg-danger text-white">Flex item 9</div>
  <div class="p-4 border bg-dark text-white">Flex item 10</div>
  <div class="p-4 border bg-danger text-white">Flex item 11</div>
  <div class="p-4 border bg-dark text-white">Flex item 12</div>
  <div class="p-4 border bg-danger text-white">Flex item 13</div>
  <div class="p-4 border bg-dark text-white">Flex item 14</div>
  <div class="p-4 border bg-danger text-white">Flex item 15</div>
  <div class="p-4 border bg-dark text-white">Flex item 16</div>
</div>
<br>
<p>.flex-nowrap:</p>
<div class="d-flex flex-nowrap bg-light">
  <div class="p-4 border bg-danger text-white">Flex item 1</div>
  <div class="p-4 border bg-dark text-white">Flex item 1</div>
  <div class="p-4 border bg-danger text-white">Flex item 3</div>
  <div class="p-4 border bg-dark text-white">Flex item 4</div>
  <div class="p-4 border bg-danger text-white">Flex item 5</div>
  <div class="p-4 border bg-dark text-white">Flex item 6</div>
  <div class="p-4 border bg-danger text-white">Flex item 7</div>
  <div class="p-4 border bg-dark text-white">Flex item 8</div>
  <div class="p-4 border bg-danger text-white">Flex item 9</div>
  <div class="p-4 border bg-dark text-white">Flex item 10</div>
  <div class="p-4 border bg-danger text-white">Flex item 11</div>
  <div class="p-4 border bg-dark text-white">Flex item 12</div>
  <div class="p-4 border bg-danger text-white">Flex item 13</div>
  <div class="p-4 border bg-dark text-white">Flex item 14</div>
  <div class="p-4 border bg-danger text-white">Flex item 15</div>
  <div class="p-4 border bg-dark text-white">Flex item 16</div>
</div>
```

.flex-wrap:

Flex item 1	Flex item 1	Flex item 3	Flex item 4	Flex item 5	Flex item 6	Flex item 7	Flex item 8	Flex item 9	Flex item 10	Flex item 11	Flex item 12	Flex item 13	Flex item 14

Flex item 15	Flex item 16

.flex-wrap-reverse:

Flex item 15	Flex item 16

Flex item 1	Flex item 1	Flex item 3	Flex item 4	Flex item 5	Flex item 6	Flex item 7	Flex item 8	Flex item 9	Flex item 10	Flex item 11	Flex item 12	Flex item 13	Flex item 14

.flex-nowrap:

Flex item 1	Flex item 1	Flex item 3	Flex item 4	Flex item 5	Flex item 6	Flex item 7	Flex item 8	Flex item 9	Flex item 10	Flex item 11	Flex item 12	Flex item 13	Flex item 14	Flex item 15	Flex item 16

Responsive Flex Classes

- ✓ All flex classes come with additional responsive classes, which makes it easy
- ✓ to set a specific flex class on a specific screen size.
- ✓ The * symbol can be replaced with sm, md, lg, xl or xxl, which represents small, medium, large, xlarge and xxlarge screens
- ✓ For Ex., d-sm-flex, d-md-flex, d-lg-flex, d-xl-flex, d-xxl-flex
  Some examples are as under.

  - o .flex-*-row
  - o .flex-*-row-reverse
  - o .flex-*-column
  - o .flex-*-column-reverse
  - o .justify-content-*-start
  - o .justify-content-*-end
  - o .justify-content-*-center
  - o .justify-content-*-between
  - o .justify-content-*-around

Carousel

✓ Carousel is a slideshow that cycles through various content using classes.
✓ It works with a collection of pictures, text, or unique markup elements.
✓ It supports previous and next indicators and buttons.
✓ The carousel also known as slideshow or image slider is some of the best way of showcasing the huge amount of contents within a small space on the web pages.
✓ It is a dynamic presentation of contents where text and images are made visible or accessible to the user by cycling through several items.

"carousel"	It creates a carousel function on the web page.
"carousel-indicators"	This class provides more carousel indicators. It indicates the number of slides in the carousel and which slide the user is currently showing or working on. The tiny dots on each slide's bottom represent the slide's indicator.
"carousel-inner"	It inserts slides in the carousel class.
"carousel-item"	It specifies the data or information on the carousel slide.
"carousel-control-prev"	It provides a previous or left button to the carousel. It allows the user to go to the previous slide of the carousel.
"carousel-control-next"	It provides a next or right button to the carousel. It allows the user to go next slide of the carousel.
"carousel-control-next-icon"	It comes with the carousel-control-next to create a "next or right side" button.
"carousel-control-prev-icon"	It comes with the "carousel-control-prev" class to create a "previous or left side" button.

"slide"	It provides a CSS style transition and animation effect to the carousel function. It slides from one item to the next. The carousel function removes the class if you do not want this effect.

Basic carousel

✓ The basic carousel uses different images in the slides with the next and previous buttons.
✓ This function uses on top of the web application.
✓ It attracts users' interaction and shows essential information.
✓ The following example will show you how to build a simple carousel or slideshow with **previous/next controls and indicators** using the Bootstrap carousel plugin.

```html
<div class="container">
  <!-- Carousel -->
  <div id="demo" class="carousel slide" data-bs-ride="carousel">

  <!-- Indicators/dots -->
  <div class="carousel-indicators">
    <button data-bs-target="#demo" data-bs-slide-to="0" class="active"></button>
    <button data-bs-target="#demo" data-bs-slide-to="1" ></button>
    <button data-bs-target="#demo" data-bs-slide-to="2" ></button>
  </div>

  <!-- The slideshow/carousel -->
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="a2.webp" alt="Los Angeles" class="d-block" style="width:100%;height: 500px;">
    </div>
    <div class="carousel-item">
      <img src="r.jpg" alt="Chicago" class="d-block" style=" width:100%;height: 500px;">
    </div>
    <div class="carousel-item">
      <img src="r.jpg" alt="New York" class="d-block" style=" width:100%;height: 500px;">
    </div>
  </div>

  <!-- Left and right controls/icons -->
  <button class="carousel-control-prev" data-bs-target="#demo" data-bs-slide="prev">
    <span class="carousel-control-prev-icon"></span>
```

```
      </button>
      <button class="carousel-control-next" data-bs-target="#demo" data-bs-slide="next">
        <span class="carousel-control-next-icon"></span>
      </button>
    </div>
</div>
```

Add Captions to Slides

```
  <div class="container">
<!-- Carousel -->
<div id="demo" class="carousel slide" data-bs-ride="carousel">

  <!-- Indicators/dots -->
  <div class="carousel-indicators">
    <button data-bs-target="#demo" data-bs-slide-to="0" class="active"></button>
    <button data-bs-target="#demo" data-bs-slide-to="1"></button>
  </div>

  <!-- The slideshow/carousel -->
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="r.jpg" alt="About" class="d-block w-100" style="height: 500px;">
      <!-- Caption -->
      <div class="carousel-caption">
        <h2>To know more about us</h2>
        <button class="btn btn-outline-light">Click here</button>
      </div>
    </div>
    <div class="carousel-item">
      <img src="r.jpg" alt="Hire" class="d-block w-100" style="height: 500px;">
      <!-- Caption -->
      <div class="carousel-caption bg-dark">
        <h2>We are hiring</h2>
        <button class="btn btn-outline-warning">Click here</button>
      </div>
    </div>
  </div>

  <!-- Left and right controls/icons -->
```
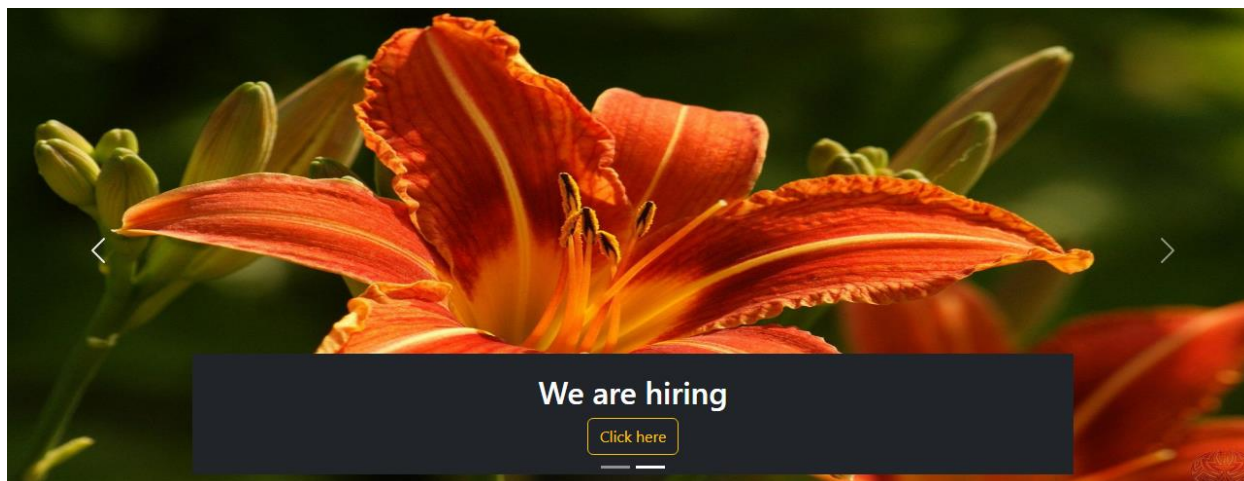
```
   <button class="carousel-control-prev" type="button" data-bs-target="#demo" data-bs-slide="prev"><span class="carousel-control-prev-icon"></span></button>

   <button class="carousel-control-next" type="button" data-bs-target="#demo" data-bs-slide="next"><span class="carousel-control-next-icon"></span></button>

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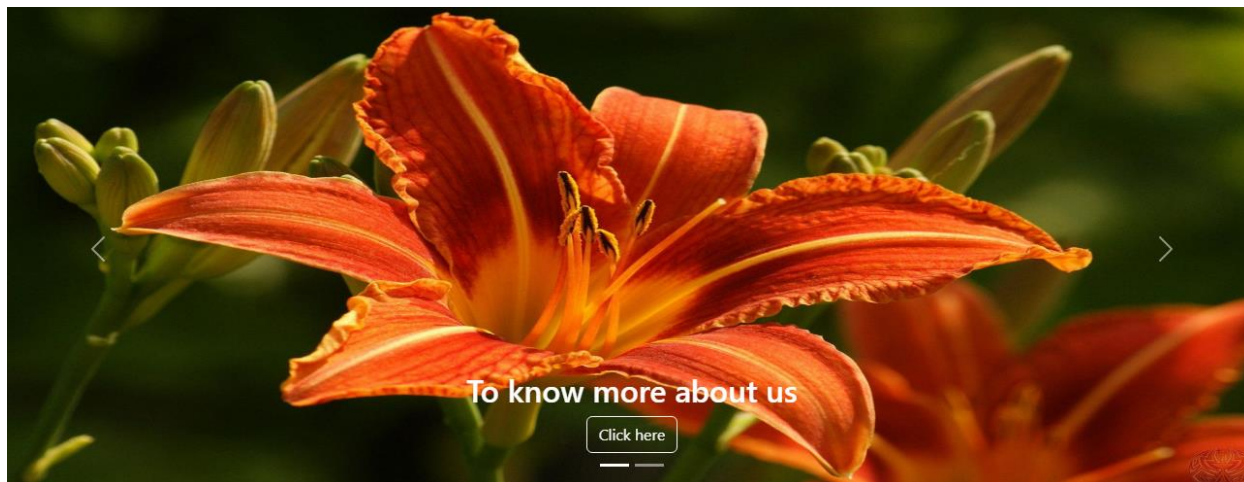
Modal

- ✓ A modal function in web applications to design the UI of the web application with information.
- ✓ The modal function displays a popup box using the button function.
- ✓ The website can have dialogues added for light boxes, user notifications, or unique content by using modules.

Basic modal function

- ✓ The basic modal function creates a single box with the web application's information.
- ✓ The modal function shows a popup box with unique user interactive data if we click the button.
- ✓ The close button helps to disclose the modal function.

```html
<!-- Button to Open the Modal -->
<button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#myModal">
  Open modal
</button>
<!-- The Modal -->
<div class="modal" id="myModal">
  <div class="modal-dialog">
    <div class="modal-content">
      <!-- Modal Header -->
      <div class="modal-header">
        <h4 class="modal-title">Modal Heading</h4>
        <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
      </div>
      <!-- Modal body -->
      <div class="modal-body">
        Modal body..
      </div>
      <!-- Modal footer -->
      <div class="modal-footer">
        <button type="button" class="btn btn-danger" data-bs-dismiss="modal">Close</button>
      </div>
    </div>
```

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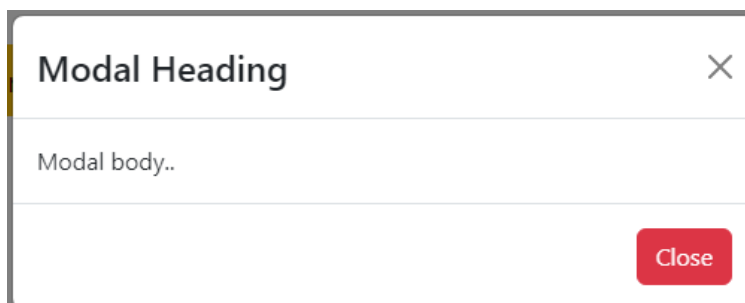

Add animation

Use the .**fade** class to add a fading effect when opening and closing the modal:

```
<!-- Fading modal -->
<div class="modal fade"></div>
<!-- Modal without animation -->
<div class="modal"></div>
```

Modal Size

Change the size of the modal by adding the **.modal-sm class for small modals**, **.modal-lg class for large modals** , or **.modal-xl for extra large modals**. **Default is 500px max-width**.

These sizes kick in at certain breakpoints to avoid horizontal scrollbars on narrower viewports. Modals have three optional sizes, available via modifier classes to be placed on a .modal-dialog.

Size	Class	Modal max-width
Small	.modal-sm	300px
Default	None	500px
Large	.modal-lg	800px
Extra large	.modal-xl	1140px

Add the size class to the <div> element with class .modal-dialog:

Small Modal

<div class="modal-dialog **modal-sm**">

Large Modal

<div class="modal-dialog **modal-lg**">

Extra Large Modal

<div class="modal-dialog **modal-xl**">

Fullscreen Modals

If you want the modal to span the whole width and height of the page, use the .modal-fullscreen class:

<div class="modal-dialog **modal-fullscreen**">

Centered Modal

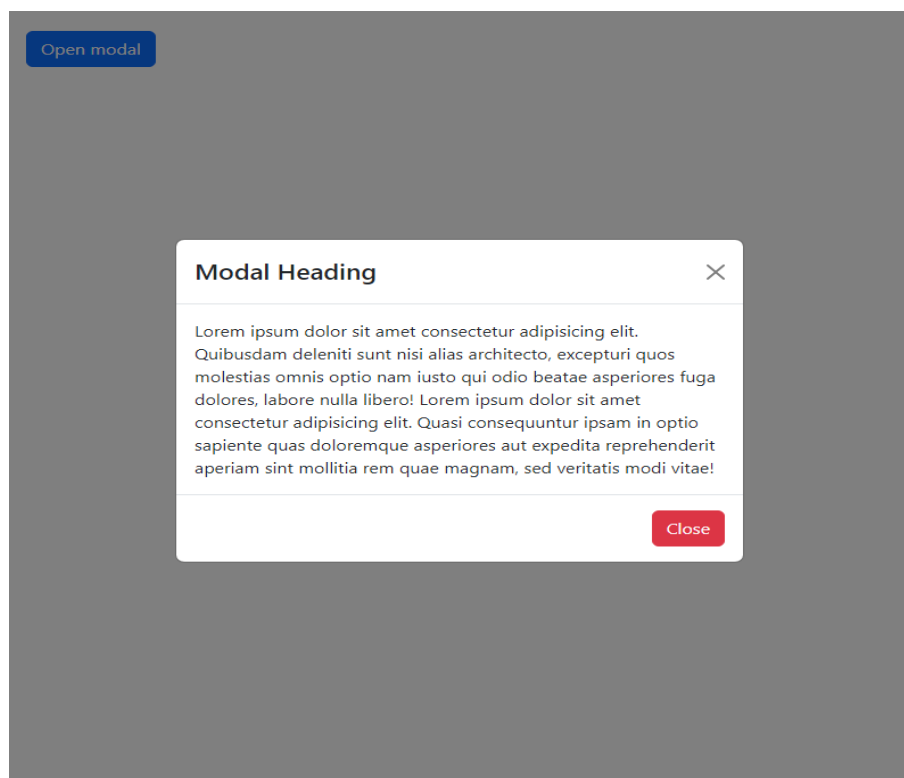Center the modal vertically and horizontally within the page, with the .modal-dialog-centered class:

<div class="modal-dialog **modal-dialog-centered**">
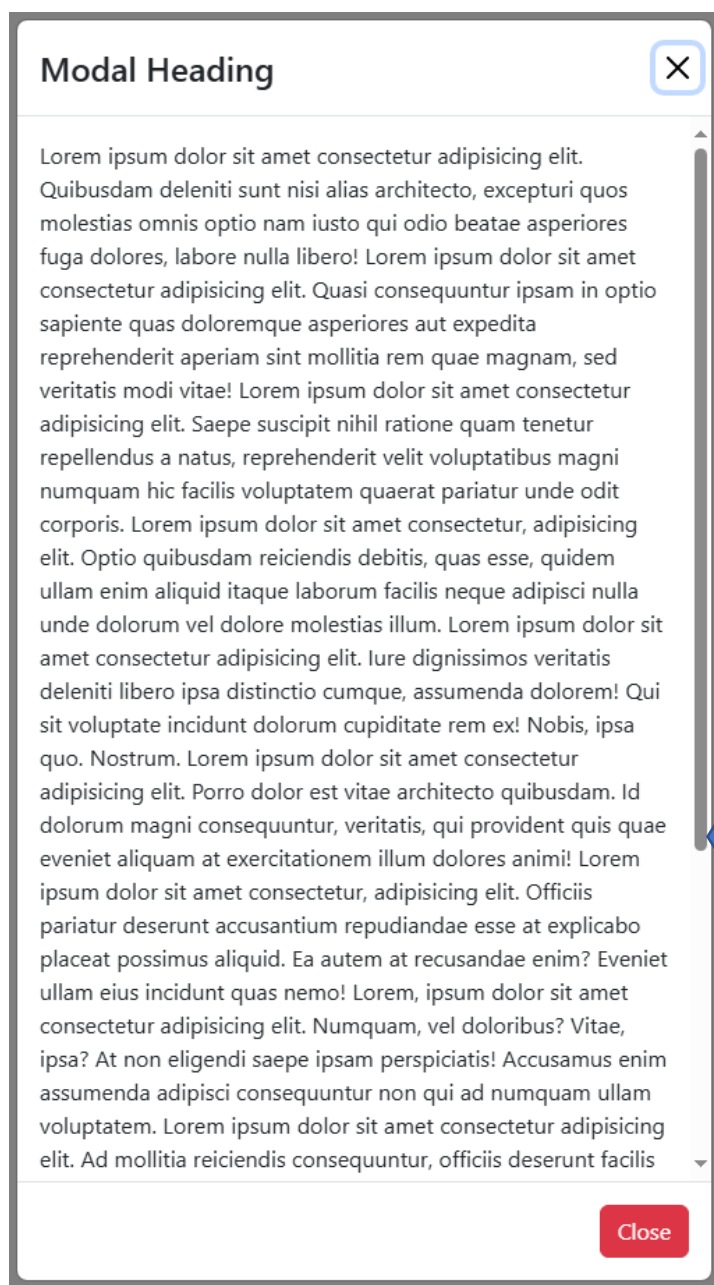
Scrolling Modal

When you have a lot of content inside the modal, a scrollbar is added to the page. See the examples below to understand it:

```
<div class="modal-dialog">
```

However, it is possible to only scroll inside the modal, instead of the page itself, by adding .modal-dialog-scrollable to .modal-dialog:

```
<div class="modal-dialog modal-dialog-scrollable">
```

Modal Heading   ✕

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quibusdam deleniti sunt nisi alias architecto, excepturi quos molestias omnis optio nam iusto qui odio beatae asperiores fuga dolores, labore nulla libero! Lorem ipsum dolor sit amet consectetur adipisicing elit. Quasi consequuntur ipsam in optio sapiente quas doloremque asperiores aut expedita reprehenderit aperiam sint mollitia rem quae magnam, sed veritatis modi vitae! Lorem ipsum dolor sit amet consectetur adipisicing elit. Saepe suscipit nihil ratione quam tenetur repellendus a natus, reprehenderit velit voluptatibus magni numquam hic facilis voluptatem quaerat pariatur unde odit corporis. Lorem ipsum dolor sit amet consectetur, adipisicing elit. Optio quibusdam reiciendis debitis, quas esse, quidem ullam enim aliquid itaque laborum facilis neque adipisci nulla unde dolorum vel dolore molestias illum. Lorem ipsum dolor sit amet consectetur adipisicing elit. Iure dignissimos veritatis deleniti libero ipsa distinctio cumque, assumenda dolorem! Qui sit voluptate incidunt dolorum cupiditate rem ex! Nobis, ipsa quo. Nostrum. Lorem ipsum dolor sit amet consectetur adipisicing elit. Porro dolor est vitae architecto quibusdam. Id dolorum magni consequuntur, veritatis, qui provident quis quae eveniet aliquam at exercitationem illum dolores animi! Lorem ipsum dolor sit amet consectetur, adipisicing elit. Officiis pariatur deserunt accusantium repudiandae esse at explicabo placeat possimus aliquid. Ea autem at recusandae enim? Eveniet ullam eius incidunt quas nemo! Lorem, ipsum dolor sit amet consectetur adipisicing elit. Numquam, vel doloribus? Vitae, ipsa? At non eligendi saepe ipsam perspiciatis! Accusamus enim assumenda adipisci consequuntur non qui ad numquam ullam voluptatem. Lorem ipsum dolor sit amet consectetur adipisicing elit. Ad mollitia reiciendis consequuntur, officiis deserunt facilis

Close

The **modal-dialog-scrollable** class is used to make a modal's content scrollable when the content exceeds the viewport height. This ensures that the modal's header and footer remain fixed, while only the body of the modal scrolls.

Forms

Creating Forms with Bootstrap

✓ HTML forms are an integral part of the web pages and applications, but creating the form layouts or styling the form controls manually one by one using CSS are often boring and tedious.

✓ Bootstrap greatly simplifies the process of styling and alignment of form controls like labels, input fields, selectboxes, textareas, buttons, radio buttons, checkbox etc. through predefined set of classes.

Two different types of form layouts:

✓ **Vertical/Stacked Form (default form layout)**
✓ **Inline Form**

✓ All textual form controls, such as **<input> type text,email, password,number ,<textarea>** etc require the class **.form-control** for general styling.
✓ **<select>** requires the class **.form-select** for general styling.
✓ **<input>** type **checkbox** and **radio** require the classes **.form-check, form-check-input, form-check-label** for general styling. Use the **checked** attribute if you want the checkbox to be checked by default.
✓ These classes also make the forms controls 100% wide.
✓ To change their width or use them inline, you can utilize the Bootstrap's predefined grid classes.
✓ If you want your **checkbox** to be styled as a **toggle switch**, use the **.form-**
✓ **switch** class together with the .form-check container:

Stacked forms

✓ The bootstrap 5 stacked form generates a full-width form with mandatory input fields such as a text field, an email, a password, a submit button etc.

```
<div class = "container mt-2 border border-dark">
  <h5> Contact form </h5>
  <form action = "#">
      <!—To add textbox -- >
    <div>
      <label class = "form-label"> User Name: </label>
```

```html
        <input type = "text" class = "form-control" placeholder = "Enter user_name" name = "
uid " required>
    </div>
      <!—To add textbox -- >
    <div>
       <label class = "form-label"> Email Id: </label>
       <input type = "email" class = "form-control" placeholder = "Enter email_id" name = "
emid " required>
    </div>
      <!—To add textarea -- >
    <div>
       <label class = "form-label"> Message: </label>
       <textarea class = "form-control" placeholder = "Enter message" name = "txt"
required></textarea>
    </div>
     <!—To add radio buttons -- >
    <div class="d-flex">
     <label class = "form-label m-2"> Gender: </label>

      <div class = "form-check m-2">
        <input class = "form-check-input" value="male" type = "radio" name = "gender"
required>
        <label class = "form-check-label">Male</label>
      </div>

      <div class = "form-check m-2">
        <input class = "form-check-input" value="female" type = "radio" name = "gender"
required>
        <label class = "form-check-label">Female</label>
      </div>
    </div>


   <!—To add dropdown -- >
   <div>
     <label>Select City</label>
      <select class="form-select">
       <optgroup label="Gujarat">
          <option>Ahmedabad</option>
          <option>Gandhinagar</option>
       </optgroup>
       <optgroup label="Rajasthan">
```

```
            <option>Jaipur</option>
        </optgroup>
      </select>
   </div>
   <!—To add checkbox -- >
   <div class = "form-check">
       <input class = "form-check-input" type = "checkbox" id = "Checked" name =
"remembers" required>
       <label class = "form-check-label" for = "Checked"> I agree</label>
   </div>


       <!—To add button -- >
      <button type = "submit" class = "btn btn-danger text-light d-block mx-auto my-4"> Submit
</button>
    </form>
  </div>
```

Note: Kindly note that it is not compulsory to add **name, id, value** etc attributes in *input tag* in exam. Also, **for attribute** is also not compulsory to add in *label* tag.

Contact form

User Name:

> Enter user_name

Email Id:

> Enter email_id

Message:

> Enter message

Gender: ○ Male ○ Female

Select City

> Ahmedabad ⌄

☐ I agree

Submit

✓ Add "**form-control-lg**" class for <mark>large</mark> form control.

✓ Add "**form-control-sm**" class for <mark>small</mark> form control.

```
<!—To add textbox -- >

    <div class = "userid">
        <label for = "uid" class = "form-label"> User Name: </label>
        <input type = "text" class = "form-control-lg" id = " uid " placeholder = "Enter
user_name" name = " uid " required>
    </div>
 <!—To add textbox -- >
    <div class = "emailed m-2">
        <label for = "emid" class = "form-label"> Email Id: </label>
        <input type = "email" class = "form-control-sm" id = " emid " placeholder = "Enter
email_id" name = " emid " required>
    </div>
```

User Name: | Enter user_name |

Email Id: | Enter email_id |

Disabled/readonly

Use the disabled and/or readonly attributes to disable the input field:

```
<input type="text" class="form-control" placeholder="Normal input">
<input type="text" class="form-control" placeholder="Disabled input" disabled>
<input type="text" class="form-control" placeholder="Readonly input" readonly>
```

| Normal input |

| Disabled input |

| Readonly input |

Creating Inline Form Layout

If you want your form elements to appear side by side, use .row and .col:

Sometimes you may want to display a series of form controls, and buttons in a single horizontal row to compact the layout. You can do this easily by using the **Bootstrap's grid classes**.

Let's take a look at following example and see how it actually works:

```
<div class = "container mt-2 border border-dark">
   <h5> Contact form </h5>
   <form action = "#">

<div class="row">
 <!—To add textbox -- >
 <div class = "col">
    <label class = "form-label"> User Name: </label>
    <input type = "text" class = "form-control" id = " uid " placeholder = "Enter user_name"
name = " uid " required>
 </div>
<!—To add textbox -- >
 <div class = "col">
    <label for = "emid" class = "form-label"> Email Id: </label>
    <input type = "email" class = "form-control" id = " emid " placeholder = "Enter email_id"
name = " emid " required>
 </div>
</div>

 <!—To add textarea -- >
    <div>
       <label for = "txt" class = "form-label"> Message: </label>
       <textarea class = "form-control" id = "txt" placeholder = "Enter message" name = "txt"
required></textarea>
    </div>

    <div class="row">
    <!—start of Responsive grids -- >
    <!—To add radio buttons -- >
      <div class = "col-md-4 d-flex">
      <label for = "gender" class = "form-label m-2"> Gender: </label>
      <div class = "form-check m-2">
        <input class = "form-check-input" value="male" type = "radio" name = "gender"
required>
        <label class = "form-check-label">Male</label>
      </div>
      <div class = "form-check m-2">
        <input class = "form-check-input" value="female" type = "radio" name = "gender"
required>
```

```
            <label class = "form-check-label">Female</label>
        </div>
    </div>

        <!—To add dropdown -- >
    <div class = "col-md-6">
        <label>Select City</label>
          <select class="form-select">
            <optgroup label="Gujarat">
              <option>Ahmedabad</option>
              <option>Gandhinagar</option>
            </optgroup>
            <optgroup label="Rajasthan">
              <option>Jaipur</option>
            </optgroup>
          </select>
      </div>

        <!—To add checkbox -- >
    <div class = "col-md-2">
        <div class = "form-check">
            <input class = "form-check-input" type = "checkbox" id = "Checked" name =
"remembers" required>
            <label class = "form-check-label" for = "Checked"> I agree</label>
        </div>
    </div>
      <!—end of Responsive grids -- >
      </div>
      <button type = "submit" class = "btn btn-danger d-block mx-auto my-4"> Submit
</button>
    </form>
  </div>
```

Contact form

User Name: Email Id:

[Enter user_name   ]   [Enter email_id   ]

Message:

[Enter message   ]

Gender: ○ Male   ○ Female   Select City   ☐ I agree

[Ahmedabad   ⌄]

[Submit]

Form Validations

✓ Form validation is "the technical process by which a web form verifies the accuracy of the user's input." The form will either notify the user of their error or require them to correct it. It will allow a user to show validation in advance.

✓ Initially, the form validation function verifies that all required fields contain data. It only takes a bootstrap 5 form validation classes to traverse each form field and check for data.

✓ Second, the entered data must be validated for correct form and value.

You can use different validation classes to provide valuable feedback to users.

✓ Add either **.was-validated or .needs-validation** to the <form> element, depending on whether you want to provide validation feedback before or after submitting the form.

Example using .was-validated class

✓ The input fields will have a green (valid) or red (invalid) border to indicate what's missing in the form.

✓ You can also add a **.valid-feedback** or **.invalid-feedback** message to tell the user explicitly what's missing, or needs to be done before submitting the form.

```
<div class = "container mt-2" >
    <h5> Bootstrap 5 Registration form validations </h5>
    <form action = "#" class = "was-validated">
        <!—Username -- >
        <div>
        <label class = "form-label"> User Name: </label>
        <input type = "text" class = "form-control" placeholder = "Enter user_name"
required>
```

```
            <div class = "valid-feedback"> Valid data. </div>
            <div class = "invalid-feedback"> Please fill the user name. </div>
            </div>
        <!—Email id -- >
            <div>
            <label class = "form-label"> Email Id: </label>
            <input type = "email" class = "form-control" placeholder = "Enter email_id" required>
            <div class = "valid-feedback"> Valid data. </div>
            <div class = "invalid-feedback"> Please fill the email id. </div>
            </div>
        <!—Message textarea -- >
            <div>
            <label class = "form-label"> Message: </label>
            <textarea class = "form-control" placeholder = "Enter message" name = "txt"
required></textarea>
            <div class = "valid-feedback"> Valid Message. </div>
            <div class = "invalid-feedback"> Please fill the message. </div>
            </div>
        <!—Checkbox -- >
         <div class = "form-check">
            <input class = "form-check-input" type = "checkbox" name = "remembers" required>
            <label class = "form-check-label" for = "Checked"> I agree</label>
            <div class = "valid-feedback"></div>
            <div class = "invalid-feedback"></div>
         </div>

     <button type = "submit" class = "btn btn-warning d-block mx-auto my-4"> Submit
</button>
     </form>
     </div>
```

Bootstrap 5 Registration form validations

User Name:

[Enter user_name   ⊘]

Please fill the user name.

Email Id:

[test   ⊘]

Please include an '@' in the email address. 'test' is missing an '@'.

Please fill the email id.

Message:

[Hi   ✓]

Valid Meddage.

☐ I agree

Submit

Example using .needs-validation class

```
<div class = "container mt-2" >
    <form action = "#" class = "needs-validation">

        <div class = "userid">
            <label for = "uid" class = "form-label"> User Name: </label>
            <input type = "text" class = "form-control" id = " uid " placeholder = "Enter
user_name" name = " uid " required>
        </div>

        <div class = "emailid">
            <label for = "emid" class = "form-label"> Email Id: </label>
            <input type = "email" class = "form-control" id = " emid " placeholder = "Enter
email_id" name = " emid " required>
        </div>

        <div class = "txt">
            <label for = "txt" class = "form-label"> Message: </label>
            <textarea class = "form-control" id = "txt" placeholder = "Enter message" name = "txt"
></textarea>
        </div>

    <button type = "submit" class = "btn btn-warning d-block mx-auto my-4"> Submit
</button>
    </form>
    </div>
```

User Name:

Enter user_name

Email Id:

asdf@dfgdf

! Please fill out this field.

Message:

Enter message

Submit

Grid Classes

The Bootstrap 5 grid system has six classes:

- ✓ **.col- (**extra small devices - screen width **less than 576px**)
- ✓ **.col-sm-** (small devices - screen width **equal to or greater than 576px**)
- ✓ **.col-md-** (medium devices - screen width **equal to or greater than 768px**)
- ✓ **.col-lg-** (large devices - screen width **equal to or greater than 992px**)
- ✓ **.col-xl-** (xlarge devices - screen width **equal to or greater than 1200px**)
- ✓ **.col-xxl-** (xxlarge devices - screen width **equal to or greater than 1400px**)