BOOTSTRAP COMPONENTS: NAVIGATION SYSTEMS & FORM ENHANCEMENTS

ABSTRACT

This module introduces key Bootstrap 5.3 tools for building responsive interfaces. It covers navigation elements like Navbars, dropdowns, collapsible menus, Offcanvas sidebars, Carousels, Modals, and Scrollspy. It also includes essential form features such as Pagination, validation, and floating labels to create clean, user-friendly layouts.

Navbar

A **navbar** (navigation bar) in Bootstrap is a responsive header component that contains links, dropdowns, logos, and forms. It helps users navigate the website easily and adapts automatically to mobile screens.

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:
  o Provides clear navigation
  o Works on all screen sizes
  o Supports brand/logo section
  o Improves website structure and user experience

Basic Navbar

With Bootstrap, a navigation bar can extend or collapse, depending on the screen size.

Classes

> **.navbar** - A standard navigation bar is created with this class
> **.navbar-expand** - responsive collapsing classes xxl, xl, lg, md, sm

Bootstrap gives you **responsive navbars** that automatically collapse into a hamburger menu on smaller screens.

Class	Navbar expands (shows full menu) at	Collapses (hamburger) at
navbar-expand-sm	≥ 576px	< 576px
navbar-expand-md	≥ 768px	< 768px
navbar-expand-lg	≥ 992px	< 992px
navbar-expand-xl	≥ 1200px	< 1200px
navbar-expand-xxl	≥ 1400px	< 1400px
navbar-expand	Always expanded (never collapses)	Never collapses

Example: A grey horizontal navbar that becomes vertical on small screens

```html
<nav class="navbar navbar-expand-lg">
  <div class="container-fluid">
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link" href="#">Home</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">About</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Contact us</a>
      </li>
    </ul>
  </div>
</nav>
```

Output:

Home   About   Contact us

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

  For eg. <nav class="navbar">

- **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)

Tip: 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

- **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).

```html
<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

Prof. Khushbu Patel

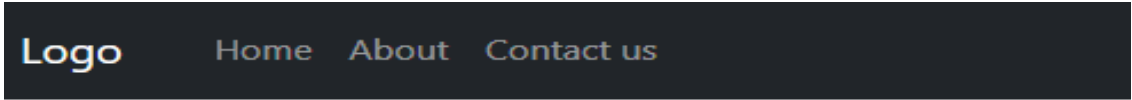- **Brand/Logo**

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

Example:

```
<nav class="navbar navbar-expand-lg bg-dark navbar-dark">
 <div class="container-fluid">
  <ul class="navbar-nav">
    <li class="navbar-brand">
     <a class="navbar-brand" href="#">Logo</a>
    </li>
    <li class="nav-item">
     <a class="nav-link" href="#">Home</a>
    </li>
    <li class="nav-item">
     <a class="nav-link" href="#">About</a>
    </li>
    <li class="nav-item">
     <a class="nav-link" href="#">Contact us</a>
    </li>
  </ul>
 </div>
</nav>
```
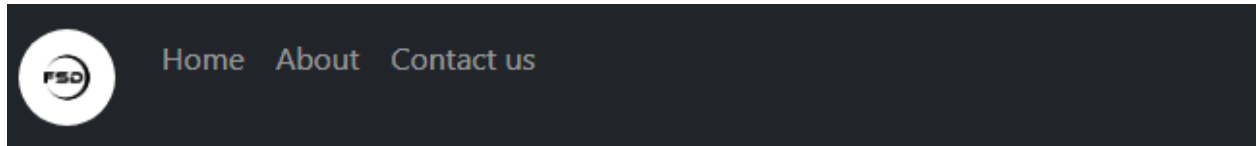
Output:

Example: Adding Logo Images to Navbars

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-lg bg-dark navbar-dark">
 <div class="container-fluid">
  <ul class="navbar-nav">
    <li class="navbar-brand">
        <img src="fsd.png" width="50px" class="rounded-pill"/>
    </li>
    <li class="nav-item"><a class="nav-link" href="#">Home</a></li>
    <li class="nav-item"><a class="nav-link" href="#">About</a></li>
    <li class="nav-item"><a class="nav-link" href="#">Contact us</a></li>
  </ul>
 </div>
</nav>
```
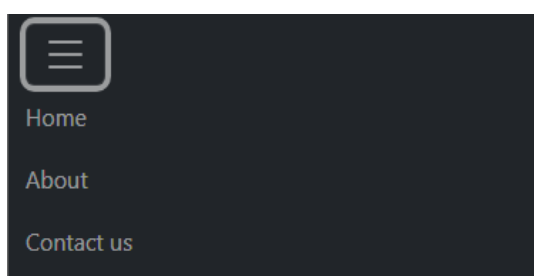
Output:

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="#link".**

> 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: "link"

Example

```
<nav class="navbar navbar-expand-lg bg-dark navbar-dark">
 <div class="container-fluid">
   <button class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#link">
       <span class="navbar-toggler-icon"></span>
   </button>
   <div class="collapse navbar-collapse" id="link">
       <ul class="navbar-nav">
          <li class="nav-item"><a class="nav-link" href="#">Home</a> </li>
          <li class="nav-item"> <a class="nav-link" href="#">About</a></li>
          <li class="nav-item"><a class="nav-link" href="#">Contact us</a></li>
       </ul>
   </div>
 </nav>
```
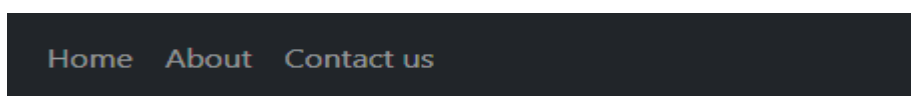
Output:

Collapsible Navbar with Dropdown

Example

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
<div class="container-fluid">
<!—To add logo -- >
<a class="navbar-brand" href="#">
<img src="p.png" class="rounded-circle" width="50px" >
 </a>
<!—To add collapsible button -- >
<button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#cn">
<span class="navbar-toggler-icon"></span>
</button>
 <div class="collapse navbar-collapse" id="cn">
 <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="#"  datta-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>
```
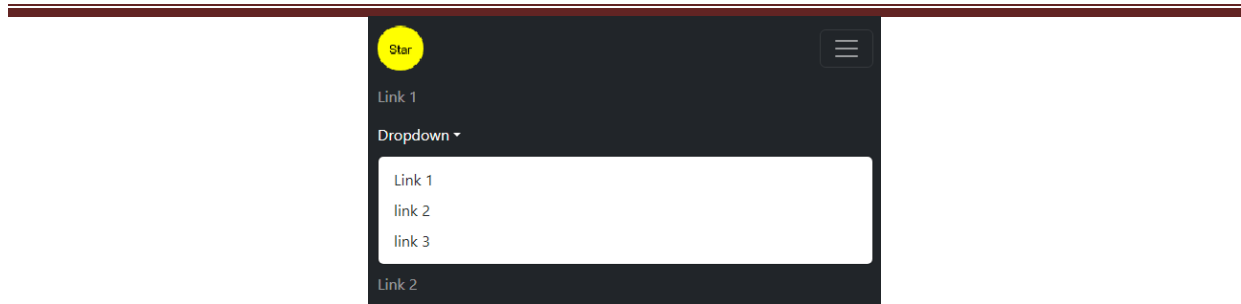
Output:

Impact of ms-auto and me-auto

Class	Does What?	Effect
ms-auto	Applies margin-left: auto	Pushes **this** element to the **right**
me-auto	Applies margin-right: auto	Pushes **next elements** to the **right**

In above example add ms-auto or me-auto at specific tag.

```
    <ul class="navbar-nav ms-auto mb-2">
|
|
    </ul>
                              Or
<img src="p.png" class="rounded-circle" width="50px"  class="d-inline-block me-auto">
</a>
```


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.

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

Prof. Khushbu Patel

Navbar cheetsheet

Class / Attribute	Meaning / Function
navbar	Enables Bootstrap navbar component styling
navbar-expand	Always expanded (never collapses) Never collapses
navbar-expand-sm/md/lg/xl/xxl	Navbar expands on mentioned screens and collapses on smaller screens
bg-dark	Applies a dark background color
navbar-dark	navbar-dark is a Bootstrap class used on a navbar to make the **text, links, and toggler button turn light-colored** (usually white). It is meant to be used when the navbar has a **dark background**.
container-fluid	Full-width responsive container
navbar-brand	Styles website logo/title area
d-inline-block	Displays logo image inline with text
rounded-circle	Makes the image perfectly circular
navbar-toggler	Creates the mobile "hamburger" button
data-bs-toggle="collapse"	Tells the button to open/close collapsible menu
data-bs-target="#navbarSupportedContent"	Connects button to the collapsible menu div
navbar-toggler-icon	Default Bootstrap hamburger icon
collapse	Makes the menu collapsible on small screens
navbar-collapse	Ensures collapse is styled correctly for navbars
navbar-nav	Styles as a horizontal navbar menu
ms-auto	**Pushes nav items to the right side**
me-auto	Pushes items to the left
mb-2	Margin-bottom on small screens;
nav-item	Styles each list item inside navbar
nav-link	Styles hyperlinks inside navbar
active	Highlights the current page link
dropdown	Enables dropdown behavior
dropdown-toggle	Turns link into dropdown trigger
data-bs-toggle="dropdown"	Enables dropdown toggle functionality
dropdown-menu	Container for dropdown items
dropdown-item	Styles individual dropdown links
dropdown-divider	Horizontal line separator inside dropdown
disabled	Disables a nav link (unclickable)
fixed-top	Fixes navbar at the top of the page
fixed-bottom	Fixes navbar at the bottom of the page
sticky-top	Navbar sticks to top when scrolling past it (not supported in IE11)

Scrollspy

- Scrollspy is used to automatically update navigation links based on the user's scroll position.
- In Bootstrap, Scrollspy highlights the corresponding nav link as the user scrolls, showing which section of the page is currently active.
- When a navigation link is clicked, the page smoothly scrolls to the targeted section.
- Scrollspy makes large-content pages easier to navigate, especially when bookmark-style links are used to move through long sections.

Steps:

- Add data-bs-spy="scroll" to the element that should act as the scrollable area (commonly the <body> or a specific scrollable container).

- Add the data-bs-target attribute and set it to the ID or class of the navigation component (e.g., .navbar).
  This connects Scrollspy to the navigation so active link highlighting works correctly.
  (In modern browsers, this may work without the attribute, but it is still recommended.)

- Ensure that section IDs match the navbar link targets:
  Example: <div id="section1"> must match <a href="#section1">.

- The optional data-bs-offset attribute controls how many pixels from the top should be used when calculating the active section.
  This is helpful when the active link changes too early or too late — for example, when a fixed navbar overlaps content.The default offset is 10px.

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

```
<body style="padding-top:70px ;">
  <nav class="navbar navbar-expand-lg bg-dark fixed-top" id="scrollpage">
    <div class="container-fluid">
      <ul class="navbar-nav nav-pills m-2">
        <li class="nav-item ">
        <a class="nav-link active text-white" href="#section1">Home</a>
        </li>
        <li class="nav-item">
        <a class="nav-link text-white" href="#section2">About</a>
        </li>
        <li class="nav-item">
        <a href="#section3" class="nav-link text-white">Contact</a>
        </li>
      </ul>
    </div>
  </nav>
```

```
<div data-bs-spy="scroll" data-bs-target="#scrollpage">

    <div id="section1" class="container-fluid bg-success text-white" style="height:800px;
padding: 100px;">
    <h1>Welcome to Home Section</h1>
    <p>Enjoy scrolling</p>
    </div>

    <div id="section2" class="container-fluid bg-warning text-white"
style="height:800px;padding: 100px;">
    <h1>Welcome to About Section</h1>
    <p>Enjoy scrolling</p>
    </div>

    <div id="section3" class="container-fluid bg-info text-white"
style="height:800px;padding: 100px;">
    <h1>Welcome to Contact Section</h1>
    <p>Enjoy scrolling</p>
    </div>
  </div>
</body>
```
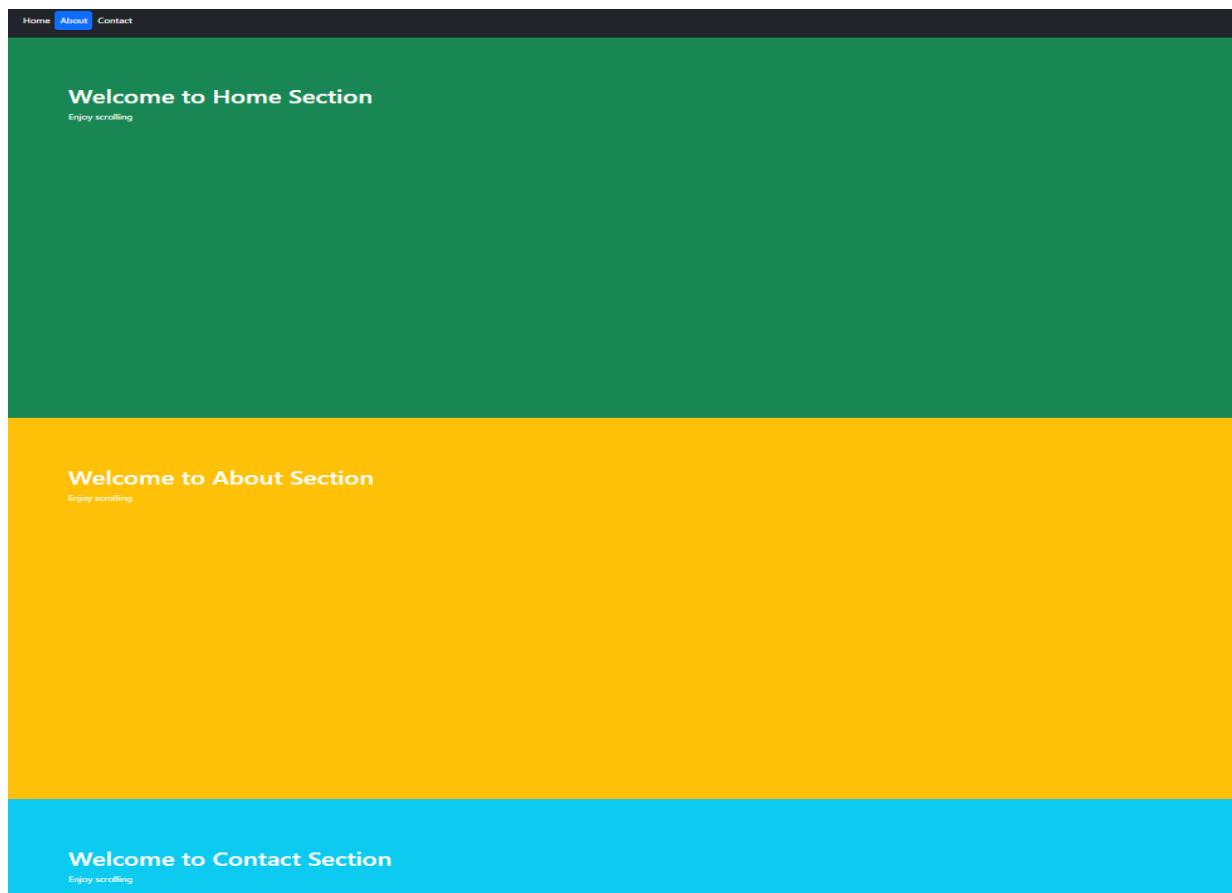
scrollspy cheetsheet

Code / Class / Attribute	Purpose
`<nav class="navbar navbar-expand-lg bg-dark fixed-top" id="scrollpage">`	Creates a dark navbar that stays fixed at the top. Id="scrollpage" allows Scrollspy to update this navbar. **Navbar stays at the top during scroll.**
`<ul class="navbar-nav nav-pills m-2">`	Container for navigation links.
Navbar-nav	Styles list items as navbar links.
Nav-pills	**Enables Bootstrap's built-in active link highlighting (blue active tab). Needed to make active link visible.**
`<a class="nav-link text-white" href="#section1">`	Each link points to a section ID. Scrollspy will highlight these.
Active	Indicates the starting active link (Home). Scrollspy updates this dynamically.
`<div data-bs-spy="scroll">`	Activates Scrollspy functionality.
Data-bs-target="#scrollpage"	Tells Scrollspy which navbar to update.
`<div id="section1">`	Section anchor. Must match navbar link (href="#section1").

Note (*Ref):

Use the following only when the default Scrollspy behaviour is not fully supported by the browser:

```
<div data-bs-spy="scroll" data-bs-target="#scrollpage"  data-bs-offset="70" tabindex="0"
style="position: relative; overflow-y:auto; height:100vh;">
```

Attribute	Why It's Needed
data-bs-offset="70"	Adjusts when the active link changes, especially when using a fixed navbar (prevents early/late highlighting).
tabindex="0"	Makes the `<div>` focusable so Scrollspy can detect scrolling inside it.
position: relative	Gives Scrollspy a positioning context to calculate section boundaries properly.
overflow-y: auto	Makes the `<div>` scrollable instead of the whole page.
height: 100vh	Sets full viewport height so the container can scroll independently.

Pagination

Pagination is used to sort the web pages of your website in an organized manner. It becomes very necessary if your website has a lot of web pages by making it easier to view and faster to load.

It helps users move to the next, previous, or specific page without overwhelming the screen.

> ➢ Shows content in small, manageable parts
> ➢ Improves performance (loads fewer items at a time)
> ➢ Makes navigation easy and organized
> ➢ Prevents very long, cluttered pages

Pagination Class:

Class	Where Used	Meaning / Purpose
pagination		Main Bootstrap pagination component container.
page-item		Wrapper for each pagination item/button.
page-link	<a>	The clickable pagination link/button.
active		Indicates the current/selected page.
disabled		Disables a link (not clickable).

Basic Pagination:

> ✓ In order to create a basic pagination, the user has to include **the .pagination** class to an <u>\
</u> element.
> ✓ After that, the user will have to add the **.page-item** to each <li> element and a **.page-link** class to each link<a> inside <u>\
</u>.

```
<div class="container">
  <ul class="pagination">
   <li class="page-item"><a class="page-link" href="#">Previous</a></li>
   <li class="page-item"><a class="page-link" href="#">Page 1</a></li>
   <li class="page-item"><a class="page-link" href="#">Page 2</a></li>
   <li class="page-item"><a class="page-link" href="#">Page 3</a></li>
   <li class="page-item"><a class="page-link" href="#">Page 4</a></li>
   <li class="page-item"><a class="page-link" href="#">Next</a></li>
  </ul>
 </div>
```

| Previous | Page 1 | Page 2 | Page 3 | Page 4 | Next |

Active and disabled State Pagination:

- In order to make the active state pagination, the **.active** class can be used to highlight the current page.
- The disabled state is used for the purpose of creating an unclickable links. The class used in this case is **.disabled** class.
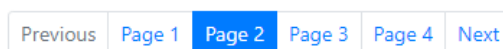
```
<div class="container m-4">
  <ul class="pagination">
  <li class="page-item disabled"><a class="page-link" href="#">Previous</a></li>
  <li class="page-item"><a class="page-link" href="#">Page 1</a></li>
  <li class="page-item active"><a class="page-link " href="#">Page 2</a></li>
  <li class="page-item"><a class="page-link" href="#">Page 3</a></li>
  <li class="page-item"><a class="page-link" href="#">Page 4</a></li>
  <li class="page-item"><a class="page-link" href="#">Next</a></li>
  </ul>
 </div>
```
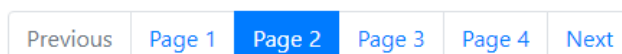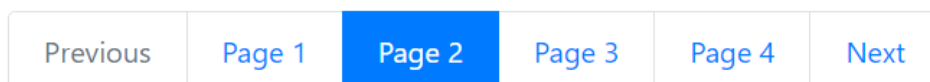
| Previous | Page 1 | Page 2 | Page 3 | Page 4 | Next |

Sizing in Pagination:

- The sizes of pagination can be changed. In order words, pagination blocks can also be sized to a larger or a smaller size.
- Add class **.pagination-lg** for larger blocks
- Add class **.pagination-sm** for smaller blocks

```
<!—large size -->
<ul class="pagination pagination-lg" style="margin:20px 0">
 <li class="page-item">...</li>
</ul>
<!—default size -->
<ul class="pagination" style="margin:20px 0">
 <li class="page-item">...</li>
</ul>
<!—Small size -->
<ul class="pagination pagination-sm" style="margin:20px 0">
 <li class="page-item">...</li>
</ul>
```

| Previous | Page 1 | Page 2 | Page 3 | Page 4 | Next |

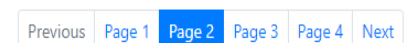| Previous | Page 1 | Page 2 | Page 3 | Page 4 | Next |

| Previous | Page 1 | Page 2 | Page 3 | Page 4 | Next |

Alignment in Pagination:

- ✓ The alignment in pagination uses the utility classes to change the alignment of the pagination.
- ✓ Deafult is **justify-content-start.**
- ✓ Add class **justify-content-center** to align pagination in **center**
- ✓ Add class **justify-content-end** to align pagination in **right** side of the page

```
<!-- Default (left-aligned) -->
<ul class="pagination" style="margin:20px 0">
  <li class="page-item">...</li>
</ul>

<!-- Center-aligned -->
<ul class="pagination justify-content-center" style="margin:20px 0">
  <li class="page-item">...</li>
</ul>

<!-- Right-aligned -->
<ul class="pagination justify-content-end" style="margin:20px 0">
  <li class="page-item">...</li>
</ul>
```
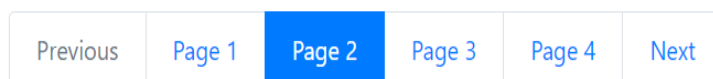
| Previous | Page 1 | Page 2 | Page 3 | Page 4 | Next |

| Previous | Page 1 | Page 2 | Page 3 | Page 4 | Next |

| Previous | Page 1 | Page 2 | Page 3 | Page 4 | Next |

Carousel

Bootstrap 5 carousel is a slideshow that cycles through various content using classes. It works with a collection of pictures, text elements, etc. Additionally, it supports previous and next indicators and buttons.

Purpose of a Carousel

- Display multiple images or slides in limited space
- Add visual appeal to a website
- Highlight important content (banners, promotions, events)
- Provide automatic or manual slide controls
- Works on all screen sizes
- Includes indicators, captions, previous/next navigation

Classes used in carousel

Class	Description
"carousel"	It creates a carousel function on the web page.
"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.
"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-caption	To create caption area. Set position (bottom-center by default). Can add custom classes inside
"carousel-item"	It specifies the data or information on the carousel slide.
"active"	Marks initial visible slider/indicator. Add only in one item and to appropriate indicator.
"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.

Attributes used in carousel

Item	Meaning / Effect	Possible values / notes
id="cs"	Unique identifier used by indicators/controls to target this carousel	Any unique id; required to link controls/indicators
data-bs-ride="carousel"	Autostart the carousel on page load (autoplay)	omit if you want manual init; alternative: initialize via JS new bootstrap.Carousel
data-bs-pause="hover"	Now, when the mouse is over the carousel, it stops sliding.	hover
button data-bs-slide-to="n"	Jump to slide n (0-based)	n = 0,1,2,... must match number of slides
data-bs-interval="ms" on .carousel-item	Per-slide duration in milliseconds	e.g. 1000, 10000; overrides global interval 5000 ms(default).
data-bs-target="#id" on controls	Targets the carousel id	Must match carousel id
data-bs-slide="prev" / "next"	Tells control direction	Only prev or next

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.

Example

```
<body>
  <!-- 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 bg-warning">
    </button>
    <button data-bs-target="#demo" data-bs-slide-to="1" class="bg-warning"></button>
    <button data-bs-target="#demo" data-bs-slide-to="2" class="bg-warning"></button>
  </div>
  <!-- The slideshow/carousel -->
  <div class="carousel-inner">
    <div class="carousel-item active">
        <img src="img1.jfif" class="d-block w-100"  height="300px" >
    </div>
    <div class="carousel-item">
        <img src="img2.jfif"  class="d-block w-100" height="300px">
     </div>
    <div class="carousel-item">
        <img src="img3.jfif"  class="d-block w-100" height="300px">
    </div>
  </div>
<!-- Left and right controls/icons  Remove bg-success only for visulization-->
    <button class="carousel-control-prev bg-success" data-bs-target="#demo"
      data-bs-slide="prev">
        <span class="carousel-control-prev-icon"></span>
    </button>
    <button class="carousel-control-next bg-success" data-bs-target="#demo"
      data-bs-slide="next">
        <span class="carousel-control-next-icon"></span>
    </button>
</div>
</body>
```

Carousel Slideshow

Carousel with caption

Add <div class= "carousel-caption">…..</div> inside each carousel-item

```
<body>
<div id="demo" class="carousel slide" data-bs-ride="carousel">
 <!-- Indicators/dots -->
   <div class="carousel-indicators">
     <button type="button" data-bs-target="#demo" data-bs-slide-to="0"
class="active"></button>
     <button type="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="ahm.jfif" alt="AHMEDABAD" height="300px" width="100%">
      <div class="carousel-caption">
        <h3>AHMEDABAD</h3>
        <p>We had such a great time at historical places</p>
      </div>
     </div>
     <div class="carousel-item">
       <img src="gnr.jfif" alt="Gandhinagar" height="300px" width="100%">
      <div class="carousel-caption">
        <h3>Gandhinagar</h3>
        <p>AKSHARDHAM TEMPLE</p>
      </div>
     </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></body>
```
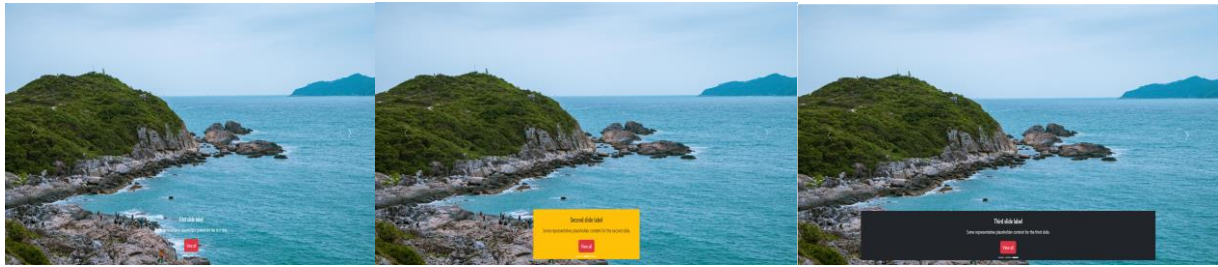
Output:

Example:

Create carousel of three slide as per image. All slide must change after interval of 1000ms.

Example

```
<div id="cc" class="carousel slide" data-bs-ride="carousel" data-bs-pause="hover">
  <!-- Indicators (small buttons that jump to a specific slide) -->
  <div class="carousel-indicators">
    <button  data-bs-target="#carouselExampleCaptions" data-bs-slide-to="0"
class="active"></button>
    <button  data-bs-target="#carouselExampleCaptions" data-bs-slide-to="1"></button>
    <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-
to="2"></button>
  </div>

  <!-- Container for the slides -->
  <div class="carousel-inner">

    <div class="carousel-item active" data-bs-interval="1000">
      <!-- .d-block makes the image display:block; .w-100 makes it full width of the
carousel. h-75 is a utility that sets height: 75% (use cautiously — better to control image
aspect with CSS). -->
      <img src="1.jpg" class="d-block w-100 h-75" alt="First slide image description">
      <!-- Caption area (positioned over the image). Use for headings/short descriptions. -->
<div class="carousel-caption">
        <h5>First slide label</h5>
        <p>Some representative placeholder content for the first slide.</p>
        <a href="#" class="btn btn-danger">View all</a>
      </div>
    </div>
    <!—second carousel-item -->
    <div class="carousel-item" data-bs-interval="10000">
     <img src="1.jpg" class="d-block w-100" alt="Second slide image description">
 <div class="carousel-caption bg-warning text-dark w-25 mx-auto">
        <h5>Second slide label</h5>
        <p>Some representative placeholder content for the second slide.</p>
        <a href="#" class="btn btn-danger">View all</a>
      </div>
    </div>
```

```html
<!—third carousel-item -->
<div class="carousel-item">
<img src="1.jpg" class="d-block w-100" alt="Third slide image description">
<!-- You added bg-dark on caption — that places a dark background behind caption text. -->
<div class="carousel-caption bg-dark">
<h5>Third slide label</h5>
<p>Some representative placeholder content for the third slide.</p>
<a href="#" class="btn btn-danger">View all</a>
</div>
</div>

</div>

<!-- Prev control: moves to previous slide -->
<button class="carousel-control-prev" data-bs-target="#carouselExampleCaptions" data-bs-slide="prev">
<!-- The inner span is the visual icon. aria-hidden or visually-hidden are for accessibility. -->
<span class="carousel-control-prev-icon"></span>
</button>

<!-- Next control: moves to next slide -->
<button class="carousel-control-next" data-bs-target="#carouselExampleCaptions" data-bs-slide="next">
<span class="carousel-control-next-icon" ></span>
</button>
</div>
```

➢ **One slide must have .active** — otherwise carousel is invisible initially.
➢ **Unique id**: indicators & controls must use data-bs-target matching the carousel id.
➢ **Autoplay**: data-bs-ride="carousel" will initialize automatically; if absent, call new bootstrap.Carousel(myEl) to enable touch/swipe events. Don't double-initialize.
➢ **Per-slide timing**: data-bs-interval on .carousel-item overrides the global interval for that slide.
➢ **Accessibility**: include alt text on images, and consider pause-on-hover or a visible pause button
➢ **Do not nest carousels** — nested carousels are unsupported.

Modal

Bootstrap 5 Modal is a dialog box or popup window that allows you to display important information to users or request necessary actions.

Modals are built with HTML, CSS, and JavaScript components that create lightboxes, user notifications, or custom content.

Modals typically consist of a header, message body, and footer containing action buttons. They are commonly used

- ➤ To display important messages or alerts
- ➤ To show forms (login, signup, feedback)
- ➤ To confirm actions (delete, submit, save)
- ➤ To display extra information (images, details) without switching pages.
- ➤ To keep the user in the same location while interacting with content.

Class	Purpose
Modal Container Classes	
modal	Base class for modal
fade	Adds fade-in animation
Modal Dialog Classes	
modal-dialog	Main wrapper for dialog box
modal-sm	Small modal (≈300px)
modal-lg	Large modal (≈800px)
modal-xl	Extra-large modal (≈1140px)
modal-fullscreen	Fullscreen modal on all sizes
modal-dialog-centered	Vertically centers the modal
modal-dialog-scrollable	Makes body scrollable; header/footer fixed
Modal Content Structure	
modal-content	Wraps the entire content inside dialog
modal-header	Header section (title + close button)
modal-title	Title style
modal-body	Main content area
modal-footer	Footer (buttons like Save / Close)
Modal Attributes	
data-bs-toggle="modal"	Opens a modal
data-bs-target="#id"	Specifies which modal to open
data-bs-dismiss="modal"	Closes the modal (used in buttons)

Example:

```html
<!-- Button trigger modal -->
<button class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#mb1">
 Click Me
</button>

<!-- Modal -->
<div class="modal fade " id="mb1" >
  <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                 <h1 class="modal-title fs-5">Modal title</h1>
                 <button type="button" class="btn-close" data-bs-dismiss="modal" ></button>
             </div>

             <div class="modal-body ">
                Modal data
             </div>

             <div class="modal-footer">
                <button class="btn btn-secondary" data-bs-dismiss="modal" >Close</button>
                <button class="btn btn-primary">Save changes</button>
            </div>
        </div>
  </div>
</div>
```
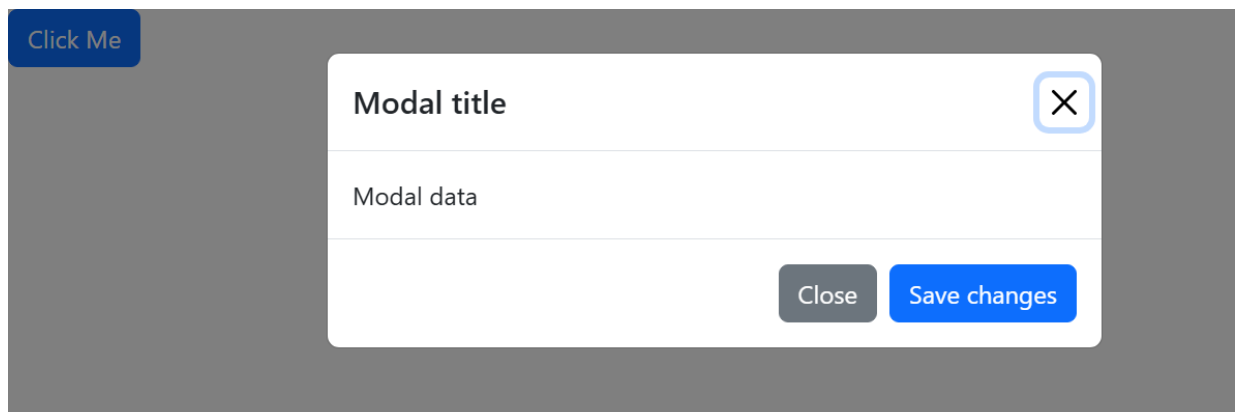

Animation in Modal

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

Eg.   `<div class="modal fade" id="ex">`

Modal Size

Change the size of the modal by adding the **.modal-sm** class for small modals (max-width 300px), **.modal-lg** class for large modals (max-width 800px), or **.modal-xl** for extra large modals (max-width 1140px). Default is 500px max-width. There is no class for xxl.

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

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

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

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

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

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">
```

Combine Scroll + Center + Size

```
<div class="modal fade" id="mb1">
  <div class="modal-dialog modal-dialog-centered modal-dialog-scrollable modal-sm">
    <div class="modal-content">
      Add long content here
    </div>
  </div>
</div>
```

Prof. Khushbu Patel

Toggle Between Modals

- You can switch from one modal to another by using data-bs-toggle="modal" and data-bs-target="#modalID" on a button.
- Bootstrap does **not** allow two modals to be open at the same time.
  So when you open the second modal, Bootstrap **automatically closes** the first one.
- This creates a smooth toggle effect between two modals (example: open "Forgot Password" from inside "Login" or Accept Terms & Conditions etc).

```html
<button class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#mb1">
 Click Me </button>
<!—Modal Box -1 -->
<div class="modal fade " id="mb1" >
  <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
             <h1 class="modal-title fs-5">Our Terms</h1>
         </div>
         <div class="modal-body ">
           Modal data
        </div>
        <div class="modal-footer">
          <button class="btn btn-secondary" data-bs-dismiss="modal" >Accept</button>
          <button class="btn btn-primary" data-bs-target="#mb2"  data-bs-toggle="modal">
            Cancel</button>
        </div>
     </div>
  </div>
</div>
<!—Modal Box -2 -->
<div class="modal fade " id="mb2" >
  <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
             <h1 class="modal-title fs-5">Our Terms</h1>
        </div>
         <div class="modal-body ">
           Clcik on Accept to continue
        </div>
        <div class="modal-footer">
          <button class="btn btn-secondary" data-bs-target="#mb1" data-bs-toggle="modal"
> Ok</button>
        </div>
     </div>
  </div>
</div>
```

Offcanvas Component

An **offcanvas** in Bootstrap is a hidden sidebar panel that slides into view from the **left, right, top, or bottom** of the screen. It is used to show menus, navigation, images, lists, or any content without leaving the current page.

- ➢ Saves screen space
- ➢ Works well on mobile
- ➢ Ideal for menus, carts, profile panels
- ➢ Slides in smoothly
- ➢ Easy to control using buttons or links
- ➢ Better alternative to large modals for navigation

Example:

```
<a class="btn btn-primary" data-bs-toggle="offcanvas" href="#OE">
 Link with href
</a>
<button class="btn btn-danger" data-bs-toggle="offcanvas" data-bs-target="#OE" >
 Button with data-bs-target
</button>

<div class="offcanvas offcanvas-start" id="OE">
   <div class="offcanvas-header">
     <h5 class="offcanvas-title" >Offcanvas</h5>
     <button type="button" class="btn-close" data-bs-dismiss="offcanvas"></button>
   </div>
   <div class="offcanvas-body">
        <div>
        Some text as placeholder. In real life you can have the elements you have chosen. Like,
        text, images, lists, etc.
        </div>
        <img src="1.jpg" class="w-100 h-auto" alt="test">
   </div>
</div>
```

Classes and Attributes

Class / Attribute	Meaning / Function	Possible Values / Notes
offcanvas	Enables Bootstrap Offcanvas component	Must be the main wrapper div
offcanvas-start	Offcanvas slides in from **left**	offcanvas-start, offcanvas-end, offcanvas-top, offcanvas-bottom
id="offcanvasExample"	Unique ID for linking trigger buttons	Any valid ID
data-bs-toggle="offcanvas"	Activates offcanvas behavior	Used on link/button
href="#offcanvasExample"	For <a> triggers; opens the selected Offcanvas	Must match Offcanvas ID
data-bs-target="#offcanvasExample"	Used on <button> triggers	Same effect as href
offcanvas-header	Header area of offcanvas	Contains title + close button
offcanvas-title	Styles the title inside offcanvas header	Any text inside it
btn-close	Bootstrap close button	Used with data-bs-dismiss="offcanvas"
data-bs-dismiss="offcanvas"	Closes the offcanvas when clicked	Works on close button
offcanvas-body	Scrollable content area	Holds text, images, links, etc.
w-100	Image width 100%	Any width utility like w-50, w-auto
h-auto	Keeps natural image height	Alternatives: h-25, h-50, h-100
btn btn-primary / btn-danger	Styled Bootstrap buttons	btn-success, btn-dark, btn-secondary, etc.

Add different Placement Classes to serve specific purpose

<div class="offcanvas **offcanvas-start**" **id="OE"**>

Placement Class	Effect	Purpose
offcanvas-start	Slides in from **left**	**main navigation menus**, category lists, or app-style sidebars.
offcanvas-end	Slides in from **right**	**shopping carts**, **filters**, chat panels, or user settings.
offcanvas-top	Slides in from **top**	**search bars**, notification panels, announcements, or quick actions.
offcanvas-bottom	Slides in from **bottom**	**mobile actions**, media controls, form snippets, or short detail sheets (like YouTube's "more info" drawer).

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.

Three different types of form layouts:

➢ Vertical/Stacked Form (default form layout)
➢ Horizontal Form (Label and input aligned in one row)
➢ Inline Form (All fields on one line label replaced by placeholders)

➢ 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.

➢ Club class="form-control form-control-color" to add effect in <input type="color"…>

➢ Add **"form-control-lg"** class in <input> for large form control.

➢ Add **"form-control-sm"** class in <input> for small form control

➢ Use the "disabled" and/or "readonly" **attributes** to disable the input field.

➢ Input Group: Used to combine an input with add-ons like text, icons, or buttons (e.g., currency symbols, search buttons, email suffixes) using the **.input-group** wrapper for compact connected controls.

➢ Form Floating: Creates modern floating label inputs where the label sits inside the field and moves up on typing, giving a clean space-saving UI using the **.form-floating** class.

➢ Form Range: A styled slider control for numeric input, created with **.form-range**, allowing users to select values visually instead of typing numbers.

Stacked form

1. Input Fields

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. All controls will be placed one after other.

Example:

```
<div class="container mt-3">
 <h2>Stacked form</h2>
 <form action="" method="get">
 <div class="mt-3">
    <label for="email" class="form-label">Email:</label>
    <input type="email" class="form-control" id="email" placeholder="Enter email">
 </div>
 <div class="mt-3">
    <label for="pwd" class="form-label">Password:</label>
    <input type="password" class="form-control" id="pwd" placeholder="Enter
      password">
 </div>
 <div class="form-check mt-3">
   <label>
 <input class="form-check-input" type="checkbox"> Remember me
</label>
 </div>
 <button type="submit" class="btn btn-primary">Submit</button>
 </form>
</div>
```

Output:

Stacked form

Email:

Enter email

Password:

Enter password

☐ Remember me

Submit

❖ **Add "form-control-lg" class for large form control.**
❖ **Add "form-control-sm" class for small form control.**

2. Dropdown (Select Menu)

Enhances the look and usability of dropdown menus. Provides better styling, size options, and accessible behavior compared to default browser selects.

Classes: .form-group, .form-select   ⬇

Example:

```
<body>
<div class="container mt-3">
   <form action="">
     <div class="form-group">
       <label>Select Country </label>
       <select class="form-select">
         <optgroup label="Asia">
           <option>India</option>
           <option>China</option>
         </optgroup>
       </select>
     </div>
   </form>
 </div>
 </body>
```
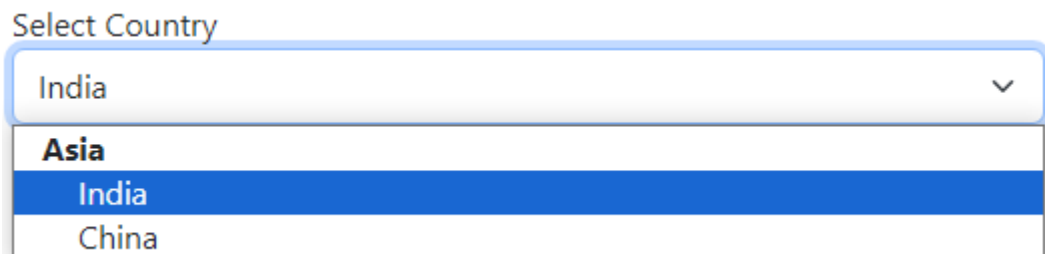
Output:

Select Country

India ⌄

Asia
India
China

3. Textarea

By default, textarea renders full width, so there is no significance of giving cols attribute in <textarea> tag.

Example:

```
<body>
   <div class="container mt-3">
     <h2>Textarea</h2>
     <form action="">
       <div class="mt-3">
```

```
        <label for="comment">Comments:</label>
        <textarea class="form-control" rows="5" id="comment" name="text"></textarea>
      </div>
      <button type="submit" class="btn btn-primary mt-3">Submit</button>
    </form>
  </div>
  </body>
```

Output:

Textarea

Comments:

Submit

4. Radio button & Checkbox

Used to create clean, uniform checkboxes, radio buttons, and switch toggles. Improves layout alignment, accessibility, and allows better control over appearance and validation feedback.

Classes:
Form-check: To style checkboxes, use a wrapper element with class="form-check" to ensure proper margins for labels and checkboxes.
Form-check-input: to style checkboxes properly inside the .form-check container.
Form-check-label: to label elements.

Example:

```
  <div class="container mt-3">
  <h2>Radio Button</h2>
  <form>
    <div class="form-check">
      <input type="radio" class="form-check-input" name="optradio" checked>
      <label class="form-check-label">Option 1</label>
    </div>
    <div class="form-check">
      <input type="radio" class="form-check-input" name="optradio">
      <label class="form-check-label">Option 2</label>
    </div>
    <div class="form-check">
      <input type="radio" class="form-check-input" name="optradio" disabled>
      <label class="form-check-label">Option 3</label>
    </div></form></div>
```

Output:

Radio Button

- ● Option 1
- ○ Option 2
- ○ Option 3

Checkbox

```
<body>
  <div class="container mt-3">
  <h2>Checkbox</h2>
  <form>
    <div class="form-check">
      <input type="checkbox" class="form-check-input" name="optcheck" checked>
      <label class="form-check-label">Option 1</label>
    </div>
    <div class="form-check">
      <input type="checkbox" class="form-check-input" name="optcheck">
      <label class="form-check-label">Option 2</label>
    </div>
    <div class="form-check">
      <input type="checkbox" class="form-check-input" name="optcheck" disabled>
      <label class="form-check-label">Option 3</label>
    </div>
  </form></div></body>
```

Output:

Checkbox

- ☑ Option 1
- ☐ Option 2
- ☐ Option 3

Checkbox (inline)

Add form-check-inline class with form-check

```
<div class="form-check form-check-inline">
    <input type="checkbox" class="form-check-input" name="optcheck" checked>
    <label class="form-check-label">Option 1</label>
  </div>
  <div class="form-check form-check-inline ">
    <input type="checkbox" class="form-check-input" name="optcheck">
    <label class="form-check-label">Option 2</label>
```

```
    </div>
    <div class="form-check form-check-inline ">
      <input type="checkbox" class="form-check-input" name="optcheck" disabled>
      <label class="form-check-label">Option 3</label>
    </div>
```

Output:

☑ Option 1   ☐ Option 2   ☐ Option 3

5. Toggle switch

If you want your checkbox to be styled as a toggle switch, use the **.form-switch** class together with the .form-check container.

Example:

```
<body>
  <div class="container mt-3">
  <form>
   <div class="form-check form-switch">
     <input class="form-check-input" type="checkbox" id="mySwitch" checked>
     <label class="form-check-label" for="mySwitch">ON/OFF</label>
   </div>
  </form>
  </div>
  </body>
```

Output:

🔵 ON/OFF

6. Disabled/readonly/Size (Ref*)

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

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

| Normal input |
| Disabled input |
| Readonly input using form-control |

Readonly input using form-control-plaintext

7. Range

Styles slider inputs to make them visually consistent across platforms. Improves the appearance of the slider thumb and track for a modern UI. The .form-range class is used to create a styled slider control that lets users select numeric values visually instead of typing them.

```
<body>
   <div class="container mt-3">
   <form>
      <div class="mt-3">
      <label for="myRange" class="form-label">Select Level (1–5)</label>
      <input type="range" class="form-range" id="myRange" min="1" max="5" step="1">
    </div>

      <div class="form-text"> Select difficulty level [Text style assigned by ".form-text"
class]</div>


   </form>
   </div>
   </body>
```

Output:

Select Level (1–5)

Select difficulty level [Text style assigned by ".form-text" class]

```
<h5>Difficulty Level</h5>
 <div class="d-flex justify-content-between">
 <span>Easy</span>
 <span>Medium</span>
  <span>Hard</span>
</div>
 <input type="range" class="form-range" min="1" max="3" step="1">
```

Output:

Difficulty Level

Easy   Medium   Hard

8. Input Group

Allows adding text, icons, or buttons before or after an input field. Useful for prefixes (like currency symbols), search icons, or combining buttons with inputs for better interaction.

```
<div class="input-group mb-3">
 <span class="input-group-text">©</span>
 <input type="text" class="form-control" placeholder="Enter author name">
</div>
```

Output

```
©  Enter author name
```

Example -2

```
<div class="mb-3">
 <label class="form-label">Website URL</label>
 <div class="input-group">
  <span class="input-group-text">https://</span>
  <input type="text" class="form-control" placeholder="example">
  <span class="input-group-text">.com</span>
 </div>
</div>
```
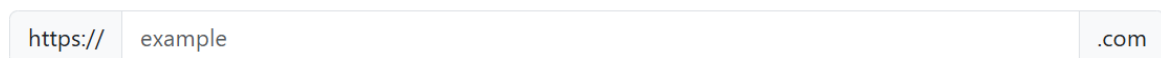
Output

Website URL

```
https://  example                                    .com
```

Horizontal Layout

To make form elements appear side by side, use Bootstrap's **.row and .col** classes**(Grid).** Group form controls in a .row, and place each control inside its own .col to align them horizontally. This is useful when you want compact, clean, horizontal form alignment.

```
<h5>Horizontal Form</h5>
 <form>
<div class="row mb-3">
  <label class="col-sm-2 col-form-label">Email</label>
  <div class="col-sm-10">
    <input type="email" class="form-control" placeholder="Enter email">
  </div>
</div>
<div class="row mb-3">
  <label class="col-sm-2 col-form-label">Password</label>
  <div class="col-sm-10">
    <input type="password" class="form-control" placeholder="Enter password">
  </div>
</div>
<button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Output

Horizontal Form

Email   Enter email

Password   Enter password

Submit

class= col-form-label — Label Used Inside Grid Columns for Line-to-line effect.

Form 1 (with *.col-form-label*)

Example label   Example input ✓

Another label ✓   Another input

Another label

Form 2 (without *.col-form-label*)

Example label   Example input

Another label   Another input

Another label

Prof. Khushbu Patel

Inline Form Layout

Inline forms can be made more compact by removing the visible labels and using meaningful placeholders inside the inputs, allowing all form controls to align neatly in a single horizontal row.

Example:

```
<h5>Inline Form</h5>
<form class="row">
  <div class="col-sm-4">
    <input type="email" class="form-control" placeholder="Email">
  </div>
  <div class="col-sm-4">
    <input type="password" class="form-control" placeholder="Password">
  </div>
  <div class="col-sm-4">
    <button type="submit" class="btn btn-primary">Go</button>
  </div>
</form>
```

Output:

Inline Form

| Email | Password | Submit |

Floating Labels

Creates minimal, space-saving input fields where labels float above the input when typing. Enhances readability, saves space, and provides a clean modern form design.

Wrap input + label inside <div class= "form-floating">

➢ Input comes first, **label comes after.**
➢ **Input** must have a **placeholder= "" (required for floating effect).**
➢ **Works with** text inputs, textarea, select **(not radios/checkboxes).**
➢ **Textarea needs** manual height.**(for effectiveness of floating label)**

Example : Stacked view

```
<form class="mt-3">

 <!-- Full Name  -->
 <div class="form-floating mb-3">
  <input type="text" class="form-control" id="fullName" placeholder="Enter your name">
  <label for="fullName">Full Name</label>
 </div>

 <!-- Textarea -->
 <div class="form-floating mb-3">
  <textarea class="form-control" placeholder="Write your message here" id="message"
style="height: 120px"></textarea>
  <label for="message">Message</label>
 </div>

 <!-- Select -->
 <div class="form-floating mb-3">
  <select class="form-select" id="country" aria-label="Floating label select">
   <option selected disabled value="">Select Country</option>
   <option value="1">India</option>
   <option value="2">USA</option>
   </select>
  <label for="country">Choose Country</label>
 </div>
</form>
```

Full Name

Message

Choose Country
Select Country

Example: Disabled Text field with Floating Label

```
 <div class="form-floating">
  <input type="text" class="form-control" placeholder="Enter username" disabled>
  <label class="fw-bold">I am Disable label</label>
</div>
<div class="form-text">This field is disabled only label display</div>
```

I am Disable label

This field is disabled only label display

Example : Create following horizontal view layout which *will automatically stack* below the medium (md) breakpoint.

Quantity   Quantity

Select Date   Select Date
                  dd-mm-yyyy

```
<div class="container m-3">

  <div class="row">
    <label class="col-md-3 col-form-label">Quantity</label>
    <div class="col-md-9">
      <div class="form-floating">
        <input type="number" class="form-control" id="qty" placeholder="0">
        <label for="qty">Quantity</label>
      </div>
    </div>
  </div>

  <div class="row">
    <label class="col-md-3 col-form-label">Select Date</label>
    <div class="col-md-9">
      <div class="form-floating">
        <input type="date" class="form-control" id="date" placeholder=" ">
        <label for="date">Select Date</label>
      </div>
    </div>
  </div>

</div>
```

Bootstrap Form Classes

Category	Class / Attribute	Description
Form Control	form-control	Styles text inputs, email fields, and textareas
	form-control-sm / form-control-lg	Small / large inputs
	form-control-plaintext	Displays read-only text aligned like normal inputs
Disabled Attribute	disabled *(HTML attribute)*	Disables the field, gives a gray appearance, removes pointer events, prevents focus & submission
Readonly Attribute	readonly *(HTML attribute)*	Makes input uneditable but still focusable; value is submitted
Labels & Help Text	form-label	Styles labels
	form-text	Helper text shown under an input
Select	form-select	Styles <select> dropdowns
	form-select-sm / form-select-lg	Small / large dropdown sizes
Checks & Radios	form-check	Wrapper for checkbox/radio
	form-check-input	Checkbox or radio input
	form-check-label	Label for checkbox/radio
	form-check-inline	Display checkbox/radio items inline
	form-switch	Turns checkbox into a switch add with form-check
Range	form-range	Styles range sliders
Input Group	input-group	Wraps inputs with addons
	input-group-text	Text or icon addon field Prefix or suffix text attached to an input
Floating Labels	form-floating	Creates floating label layout inputs/selects/textareas
Layout Helpers	row, col-*	Grid layout classes
Validation	is-valid	Valid input styling
	is-invalid	Invalid input styling
	was-validated	Enables validation feedback
	valid-feedback / invalid-feedback	Messages for validation

Prof. Khushbu Patel

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.

Class= "needs-validation"

Marks a form to use Bootstrap's custom validation styles.

Key Points:
• Added on the **form tag**.
• Tells Bootstrap: **"This form should use Bootstrap validation styles."**
• Does **not** show validation immediately styles appear **only after submission**.

```
<form class="needs-validation">
  <div class="mb-3">
    <label for="name" class="form-label">Name</label>
    <input type="text" class="form-control" id="name" required>
  </div>
  <div class="mb-3">
    <label class="form-label">Email</label>
    <input type="email" class="form-control" required>
  </div>
  <button class="btn btn-primary" type="submit">Submit</button>
</form>
```

Name

Email

⚠ Please fill out this field.

Submit

Note: Add the novalidate attribute together with the needs-validation class on the form to disable the browser's default validation messages and enable Bootstrap's custom validation. JavaScript is then required to activate the validation behavior.

Prof. Khushbu Patel

Class= "was-validated"

➢ 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.

➢ If you use was-validated, then is-valid and is-invalid are NOT required. Bootstrap will apply them automatically based on validity. is-valid and is-invalid are used for manual color control of form fields (green for valid, red for invalid), while was-validated applies these classes automatically based on validity after form submission.

Purpose:
Displays Bootstrap validation results after form submission.

Key Points:
• Added **automatically by JavaScript** after a submit event.
• Stays on the form so validation state remains visible.

Class Table

Class	Purpose
was-validated	Shows validation results after submit
valid-feedback	Normal message shown when input is valid
invalid-feedback	Normal message shown when input is invalid

Example: Provide validation for below using valid and invalid feedback message.

Username:

    Enter username   ⊙
Please fill out this field.

Password:

    •••••••••   ✓
Valid.

City
    Ahmedabad   ✓

State
    Choose...   ⊙ ⌄

City looks good!

Please choose a state.

☐ I agree on terms and conditions
    Check this checkbox to continue.

Submit

```html
<form class="was-validated">
    <div class="mt-3">
    <label class="form-label">Username:</label>
    <input type="text" class="form-control" placeholder="Enter username"
        name="uname" required>
     <div class="valid-feedback">Valid.</div>
     <div class="invalid-feedback">Please fill out this field.</div>
    </div>
    <div class="mt-3">
    <label class="form-label">Password:</label>
    <input type="password" class="form-control" placeholder="Enter password" required>
    <div class="valid-feedback">Valid.</div>
    <div class="invalid-feedback">Please fill out this field.</div>
    </div>
    <!-- City & State (Row)   -->
    <div class="row">
     <!-- City -->
    <div class="col-md-6 mt-4">
      <label class="form-label">City</label>
      <input type="text" class="form-control" placeholder="Enter city" required>
      <div class="valid-feedback">City looks good!</div>
      <div class="invalid-feedback">Please enter a valid city.</div>
     </div>
     <!-- State -->
     <div class="col-md-6 mt-4">
      <label class="form-label">State</label>
      <select class="form-select" required>
       <option value="">Choose...</option>
       <option>Gujarat</option>
       <option>Maharashtra</option>
      </select>
      <div class="valid-feedback">State selected.</div>
      <div class="invalid-feedback">Please choose a state.</div>
     </div>
    </div>
    <!-- Terms Checkbox       -->
    <div class="form-check mt-4">
     <input class="form-check-input" type="checkbox" required>
     <label class="form-check-label">I agree on terms and conditions</label>
     <div class="valid-feedback">Valid.</div>
     <div class="invalid-feedback">Check this checkbox to continue.</div>
    </div>
    <!-- Submit Button  -->
    <button type="submit" class="btn btn-primary mt-4">Submit</button>
  </form>
```

Example:

Design a form as shown below using existing Bootstrap layout classes, floating labels, and form classes.


```
<div class="container bg-light p-5">
<form class="mt-4">

 <!-- ========================= -->
 <!-- USER INFO (3 Columns)  -->
 <!-- ========================= -->
 <h4>User Information</h4>
 <div class="row">
   <div class="col-md-4">
     <div class="form-floating">
      <input type="text" class="form-control"  placeholder="Enter username" disabled>
      <label class="fw-bold">Username</label>
     </div>
     <div class="form-text">This field is disabled</div>
   </div>
   <div class="col-md-4">
     <div class="form-floating">
      <input type="email" class="form-control"  placeholder="te">
      <label class="fw-bold">Email Address</label>
```

```html
    </div>
    <div class="form-text">We'll never share your email with anyone else.</div>
  </div>
  <div class="col-md-4">
    <label class="form-label" class="fw-bold">User ID (Auto-assigned)</label>
    <input type="text" class="form-control-plaintext" value="IN-001" readonly>
  </div>
</div>

<!-- ========================= -->
<!--  LOCATION            -->
<!-- ========================= -->
<h4 class="mt-4">Location</h4>
<div class="row">
  <div class="col">
    <div class="form-floating">
      <select class="form-select" >
        <option value="" class="fw-bold">Select your city</option>
        <option>Ahmedabad</option>
        <option>Rajkot</option>
        <option>Bhavnagar</option>
        <option>Surat</option>
      </select>
      <label >City</label>
    </div>
  </div>
</div>
<!-- ========================= -->
<!-- PERSONAL DETAILS     -->
<!-- ========================= -->
<h4 class="mt-4">Personal Details</h4>
<!-- Gender (cannot use floating label) -->

<div class="row">
  <div class="col-sm-6">
    <label class="form-label fw-bold">Gender</label>
    <div class="form-check">
      <input class="form-check-input" type="radio" name="gender" value="male">
      <label class="form-check-label">Male</label>
    </div>

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

    <div class="form-check">
      <input class="form-check-input" type="radio" name="gender" value="other">
      <label class="form-check-label">Other</label>
    </div>
```

```html
      </div>

      <!-- Hobbies (cannot use floating label) -->

      <div class="col-sm-6">
        <label class="form-label fw-bold">Hobbies</label>
        <div class="form-check form-check-inline">
          <input class="form-check-input" type="checkbox" value="h1">
          <label class="form-check-label">Reading</label>
        </div>

        <div class="form-check form-check-inline">
          <input class="form-check-input" type="checkbox" value="h2">
          <label class="form-check-label">Traveling</label>
        </div>

        <div class="form-check form-check-inline">
          <input class="form-check-input" type="checkbox" value="h3">
          <label class="form-check-label">Photography</label>
        </div>
      </div>
    </div>
    <!-- ========================= -->
    <!--  PREFERENCES        -->
    <!-- ========================= -->
    <h4>Preferences</h4>
    <div class="row">
      <div class="col-md-6 mb-4">
        <div class="form-check form-switch">
          <input class="form-check-input" type="checkbox" value="newsletter" checked>
          <label class="form-check-label" >Subscribe to newsletter</label>
        </div>
      </div>
      <div class="col-md-6 mb-4">
        <h5 class="mb-1">Satisfaction Level</h5>
        <div class="d-flex justify-content-between mb-1">
          <span>Very Poor</span>
          <span>Poor</span>
          <span>Average</span>
          <span>Good</span>
          <span>Excellent</span>
        </div>
        <input type="range" class="form-range" min="1" max="5" step="1">
      </div>
    </div>


    <!-- ========================= -->
    <!-- DONATION        -->
    <!-- ========================= -->
    <h4>Donation</h4>
```

```html
  <div class="row">
    <div class="col-md-4 mb-3">
      <div class="input-group">
        <span class="input-group-text">Rs</span>
        <input type="text" class="form-control" placeholder="Amount">
      </div>
    </div>
    <div class="col-md-8 mb-5">
      <div class="input-group">
        <span class="input-group-text">First and last name</span>
        <input type="text"  class="form-control">
        <input type="text"  class="form-control">
      </div>
    </div>
  </div>
  <!-- ======================== -->
  <!--comments (textarea)-->
  <!-- ======================== -->
  <div class="row mb-4">
  <div class="col-md-12">
    <div class="form-floating">
      <textarea class="form-control" id="comments" placeholder="Leave your comments here" style="height: 100px;"></textarea>
      <label for="comments">Comments</label>
    </div>
  </div>
</div>
<button type="submit" class="btn btn-primary w-100">Submit Form</button>

</form>
</div>
```