

Chapter 7

Bootstrap Navbar

What is a 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.

Why Do We Add a Navbar?

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

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
Class / Attribute	Meaning / Function	
navbar	Enables Bootstrap navbar component styling	
navbar-expand-lg	Navbar expands on large 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	

Class	Navbar expands (shows full menu) at	Collapses (hamburger) at
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	
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)	

Example: Collapsible Navbar With Dropdown

```

<nav class="navbar navbar-expand-lg bg-dark navbar-dark">
<div class="container-fluid">
  <a class="navbar-brand" href="#">
     My Website
  </a>
  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent">
    <span class="navbar-toggler-icon"></span>
  </button>

```

```
<div class="collapse navbar-collapse" id="navbarSupportedContent">
```

```
<ul class="navbar-nav ms-auto mb-2">
```

```
<li class="nav-item">
```

```
<a class="nav-link active" href="#">Home</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="#">About</a>
```

```
</li>
```

```
<!--Dropdown -->
```

```
<li class="nav-item dropdown">
```

```
<a class="nav-link dropdown-toggle" href="#" data-bs-toggle="dropdown" >
```

```
Projects
```

```
</a>
```

```
<ul class="dropdown-menu">
```

```
<li><a class="dropdown-item" href="#">Staff</a></li>
```

```
<li><a class="dropdown-item" href="#">Company</a></li>
```

```
<li><hr class="dropdown-divider"></li>
```

```
<li><a class="dropdown-item" href="#">Services</a></li>
```

```
</ul>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link disabled">Contact</a>
```

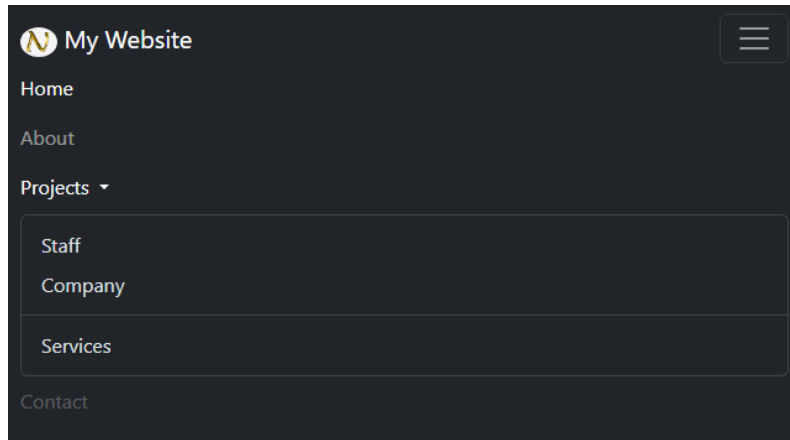
```
</li>
```

```
</ul>
```

```
</div>
```

```
</div>
```

```
</nav>
```



Difference 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 we have added **ms-auto**.

```
<ul class="navbar-nav ms-auto mb-2">
</ul>
```

Replace it with **me-auto** and you will get the below output.



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 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 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 sticky-top"></nav>
```

Scrollspy

- ✓ **Scrollspy** highlights the current section's link in the navigation while you scroll.
- ✓ Use it when your page has multiple sections and you want the menu to show where the user is on the page.

```
<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>

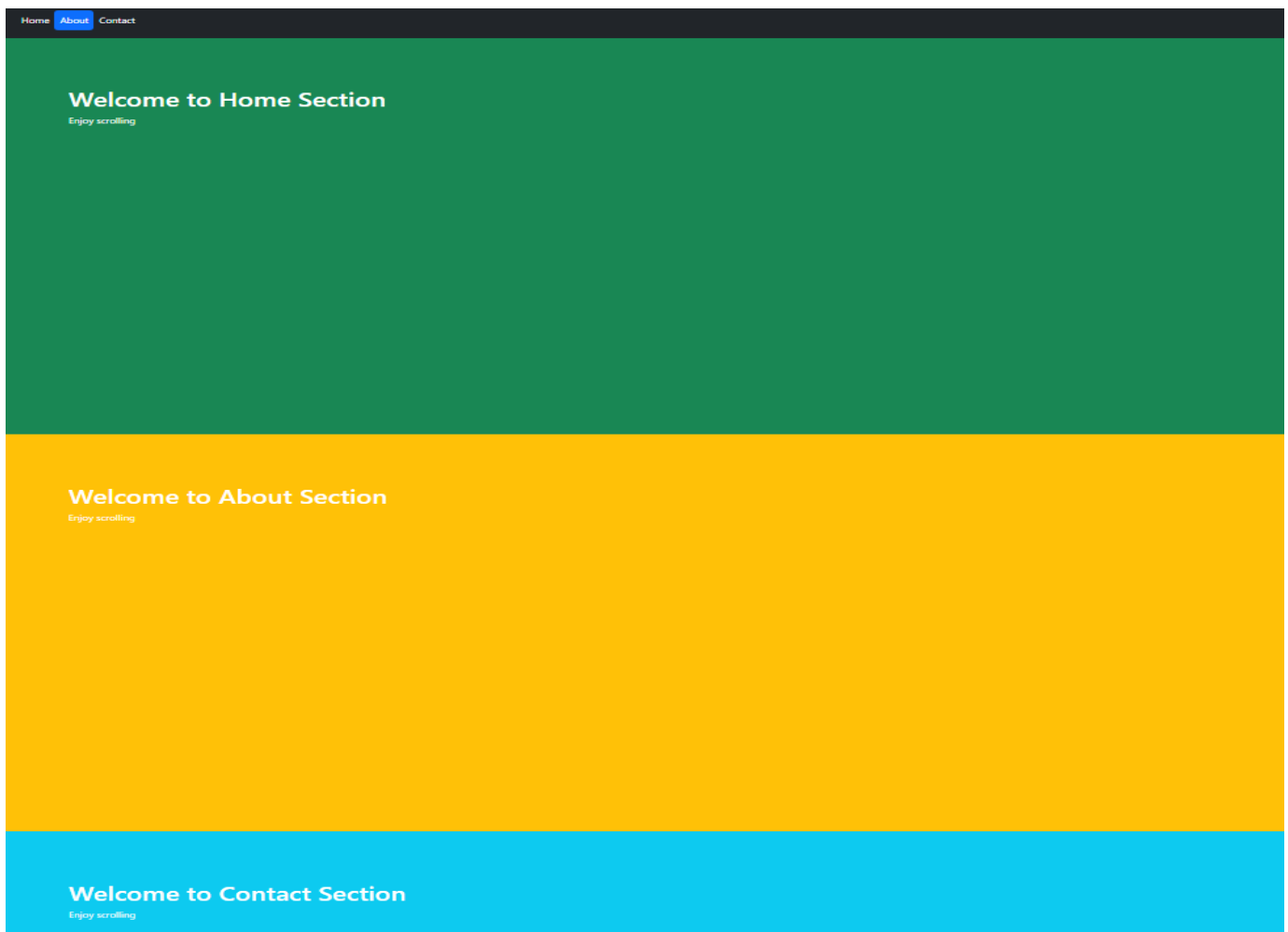
    <div id="section3" class="container-fluid bg-info text-white" style="height:800px;padding:
100px;">
      <h1>Welcome to Contact Section</h1>
```

```
<p>Enjoy scrolling</p>
</div>
```

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

nav-pills is added so that the active link becomes visible automatically

Without nav-pills, Bootstrap does **not apply any active background color**, so Scrollspy works but you won't see the active link change.



Pagination

Pagination divides **large content into multiple pages**, making it easier to view and faster to load. It helps users move to the **next, previous, or specific page** without overwhelming the screen.

Why use pagination?

- 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 Classes

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).

Example:

```
<nav>
  <ul class="pagination">
    <li class="page-item disabled">
      <a class="page-link">Previous</a>
    </li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item active">
      <a class="page-link" href="#">2</a>
    </li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item">
      <a class="page-link" href="#">Next</a>
    </li>
  </ul>
</nav>
```




Additional Pagination Alignment Classes

Use these on the `<ul class="pagination">` container:

Class	Effect
<code>justify-content-start</code>	Align pagination left (default).
<code>justify-content-center</code>	Center the pagination.
<code>justify-content-end</code>	Align pagination right .

Example:

```
<ul class="pagination justify-content-center">
```

Pagination Sizing Classes

Add these to the `<ul class="pagination">` to change button size:

Class	Size
<code>pagination-lg</code>	Large pagination buttons
<code>pagination-sm</code>	Small pagination buttons
<i>(no class)</i>	Default size

Example:

```
<ul class="pagination pagination-lg">
```

Carousel

A **carousel** is a slideshow component that allows rotating images, text, or custom content.

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



Example

```
<!-- Root carousel element: unique id used by controls/indicators -->
<div id="carouselExampleCaptions" class="carousel slide" data-bs-ride="carousel" data-bs-pause="hover">
  <!-- Indicators (small buttons that jump to a specific slide) -->
  <div class="carousel-indicators">
    <!--
      data-bs-target must match the carousel id.
      data-bs-slide-to is the zero-based slide index.
      'active' marks the initial visible indicator.
    -->
    <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">

  <!-- Each .carousel-item is one slide.
  Add .active to one item so the carousel shows something initially.
  data-bs-interval on a slide overrides the global interval for that slide. -->
  <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). -->
    
    <!-- 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="1000">
    
    <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>

  <!--third carousel-item -->
  <div class="carousel-item">
    
    <!-- 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. -->

```

<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">

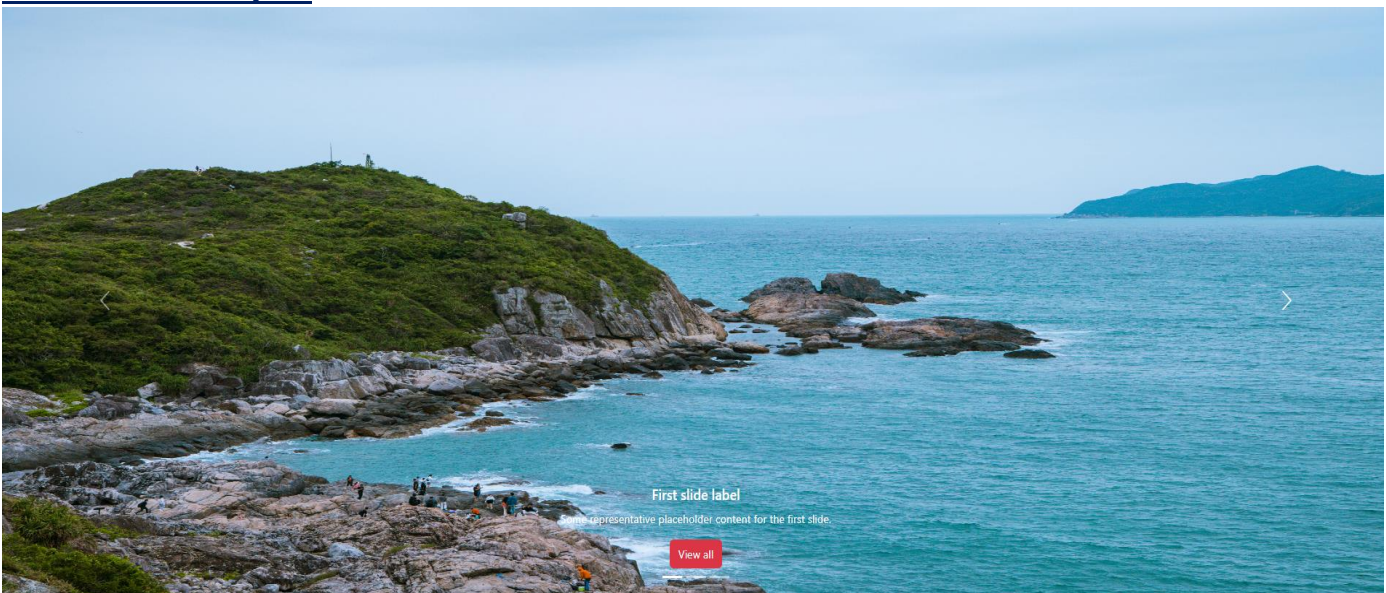
```

```

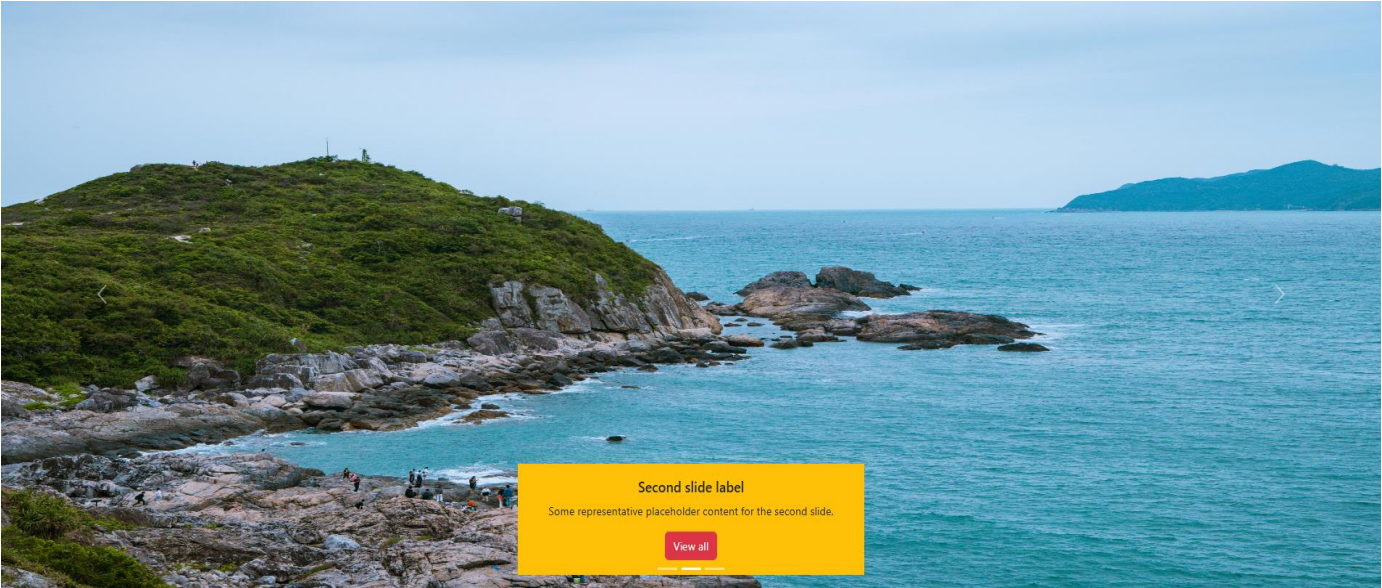
<span class="carousel-control-next-icon" ></span>
</button>
</div>

```

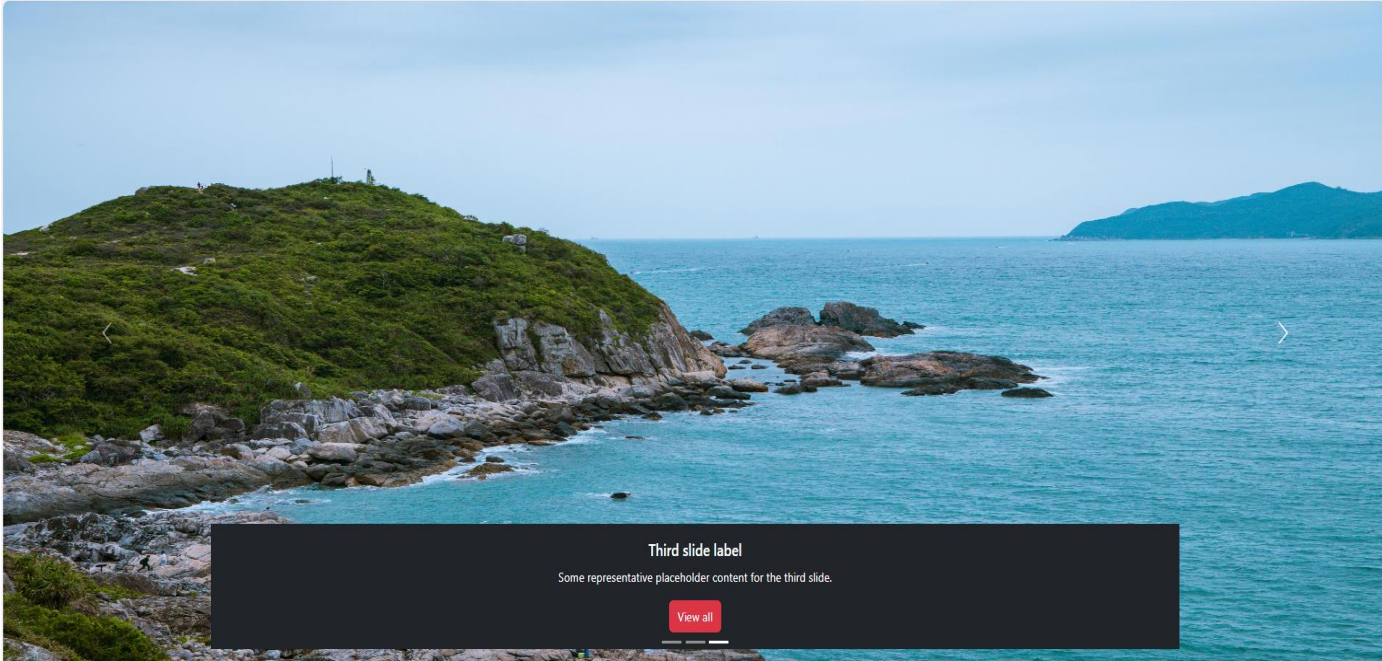
First slide output



Second slide output



Third slide output



Classes and Attributes

Item	Meaning / Effect	Possible values / notes
id="carouselExampleCaptions"	Unique identifier used by indicators/controls to target this carousel	Any unique id; required to link controls/indicators
class="carousel"	Base Bootstrap carousel styling	required for component
class="slide"	Enables sliding animation (CSS/JS)	omit to disable slide animation
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
.carousel-indicators	Wrapper for indicator buttons (small dots)	optional . you can omit indicators
button data-bs-slide-to="n"	Jump to slide n (0-based)	n = 0,1,2,... must match number of slides
class="active" on indicator/item	Marks initial visible slide/indicator	Only on one slide/indicator
.carousel-inner	Slide container	required
.carousel-item	One slide wrapper	required for each slide
data-bs-interval="ms" on .carousel-item	Per-slide duration in milliseconds	e.g. 1000, 10000; overrides global interval 5000 ms(default).
	d-block (display:block); w-100 (width:100%)	Replace w-100 with custom sizing or use object-fit for cropping
.carousel-caption	Positions caption area (bottom-center by default)	You can add text-start or text-end, or custom classes
.carousel-control-prev / .carousel-control-next	Prev/next control wrappers	Buttons recommended; can use <a> with role="button"

Item	Meaning / Effect	Possible values / notes
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
.carousel-control-* -icon	Built-in icon visuals (uses background-image)	You can replace with custom icons (SVG/font icon)

- ✓ **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

- ✓ A **modal** is a popup window that appears on top of a webpage. It is used to show important information or interact with the user **without leaving the current page**.
- ✓ When a modal opens, the rest of the page becomes inactive until the modal is closed. This helps the user focus on a specific task such as filling a form, confirming an action, or viewing additional details.

Why Modals Are 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

Modal Container Classes

Class	Purpose
modal	Base class for modal
fade	Adds fade-in animation

Modal Dialog Classes

Class	Purpose
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

Class	Purpose
modal-dialog-scrollable	Makes body scrollable; header/footer fixed

Modal Content Structure

Class	Purpose
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 Triggers

Attribute	Purpose
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:

```

<!-- Button trigger modal -->
<button class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#exampleModal">
  Launch demo modal
</button>

<!-- Modal -->
<div class="modal fade " id="exampleModal" >
  <div class="modal-dialog">

    <div class="modal-content">

```

```
<div class="modal-header">
  <h1 class="modal-title fs-5" id="exampleModalLabel">Modal title</h1>
  <button type="button" class="btn-close" data-bs-dismiss="modal" ></button>
</div>
```

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

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ratione reiciendis, perspiciatis quaerat tenetur rem placeat? Velit mollitia reiciendis officiis sunt asperiores? Alias, cumque tempora! A corrupti mollitia ad perferendis nobis.

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Fugit, obcaecati aliquid ratione saepe aspernatur nisi adipisci laudantium velit suscipit aperiam odio autem in laborum inventore possimus veniam eum quaerat officia.

```
</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>
```

Example of where to add size classes

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

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

Combine Scroll + Center + Size

```
<div class="modal fade" id="exampleModal">
```

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

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

Add long content here

```
</div>
```

```
</div>
```

```
</div>
```

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”).

```
<button class="btn btn-primary" data-bs-target="#exampleModalToggle" data-bs-  
toggle="modal">Open first modal</button>
```

```
<div class="modal fade" id="exampleModalToggle">  
  <div class="modal-dialog modal-dialog-centered">  
    <div class="modal-content">  
      <div class="modal-header">  
        <h1 class="modal-title fs-5" >Modal 1</h1>  
        <button type="button" class="btn-close" data-bs-dismiss="modal" ></button>  
      </div>  
      <div class="modal-body">  
        Show a second modal and hide this one with the button below.  
      </div>  
      <div class="modal-footer">  
        <button class="btn btn-primary" data-bs-target="#exampleModalToggle2" data-bs-  
toggle="modal">Open second modal</button>  
      </div>  
    </div>  
  </div>  
</div>
```

```
<div class="modal fade" id="exampleModalToggle2" >  
  <div class="modal-dialog modal-dialog-centered">  
    <div class="modal-content">  
      <div class="modal-header">  
        <h1 class="modal-title fs-5" >Modal 2</h1>  
        <button type="button" class="btn-close" data-bs-dismiss="modal" ></button>  
      </div>  
      <div class="modal-body">  
        Hide this modal and show the first with the button below.
```

```
</div>
<div class="modal-footer">
  <button class="btn btn-primary" data-bs-target="#exampleModalToggle" data-bs-
toggle="modal">Back to first</button>
</div>
</div>
</div>
</div>
```

Offcanvas Component

What is an Offcanvas?

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.

Why Use Offcanvas?

- 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="#offcanvasExample">
```

Link with href

```
</a>
```

```
<button class="btn btn-danger" data-bs-toggle="offcanvas" data-bs-target="#offcanvasExample" >
```

Button with data-bs-target

```
</button>
```

```
<div class="offcanvas offcanvas-start" id="offcanvasExample">
```

```
<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>
```

```

```

```
</div>
```


Link with href

Button with data-bs-target

Offcanvas

×

Some text as placeholder. In real life you can have the elements you have chosen. Like, text, images, lists, etc.



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

Class / Attribute	Meaning / Function	Possible Values / Notes
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.

Placement Classes

Placement Class	Where Offcanvas Appears
offcanvas-start	Slides in from left
offcanvas-end	Slides in from right
offcanvas-top	Slides in from top
offcanvas-bottom	Slides in from bottom

To see the effect of different placements, replace **offcanvas-start** with: **offcanvas-end**, **offcanvas-top**, or **offcanvas-bottom**.

Bootstrap Forms

Bootstrap provides a set of form controls, layouts, and components that ensure **consistent, responsive, and visually polished** forms across all browsers. It enhances default HTML form elements using utility classes and custom styles while maintaining accessibility.

1. Form Control

Used to style standard input fields to ensure consistent appearance across different browsers. Helps create uniform text fields, date inputs, and textareas with support for placeholders, disabled/readonly states, and validation styling.

2. Select

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

3. Checks & Radios

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

4. Range

Styles slider inputs to make them visually consistent across platforms. Improves the appearance of the slider thumb and track for a modern UI.

5. 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.

6. 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.

7. Layout

Helps organize form elements using inline, horizontal, or grid-based arrangements. Ensures your forms are responsive and properly aligned using spacing and grid utilities.

8. Validation

Adds visual feedback for valid or invalid inputs. Helps guide users during form submission by highlighting errors and successes while relying on HTML5 validation rules.

Bootstrap Form Classes

Category	Class / Attribute	Description
Form Control	form-control	Styles inputs & textareas
	form-control-sm / form-control-lg	Small / large inputs
	form-control-plaintext	Displays read-only plain text
Disabled Attribute	disabled (<i>HTML attribute</i>)	Disables the field, gives a gray appearance, removes pointer events, prevents focus & submission
Readonly Attribute	readonly (<i>HTML attribute</i>)	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 items inline
	form-switch	Turns checkbox into a switch
Range	form-range	Styles range sliders
Input Group	input-group	Wraps inputs with addons
	input-group-text	Text or icon addon
Floating Labels	form-floating	Creates floating label layout
Layout Helpers	row, col-*	Grid layout classes
Validation	is-valid	Valid input styling

Category	Class / Attribute	Description
	is-invalid	Invalid input styling
	was-validated	Enables validation feedback
	valid-feedback / invalid-feedback	Messages for validation

What Is Layout in Bootstrap Forms?

- ✓ **Layout** refers to *how your form fields are arranged on the page* vertically, horizontally, or inline.
- ✓ Bootstrap provides layout systems to control **spacing, alignment, and responsiveness** using its grid (row, col-*) and utility classes.

Vertical Layout

- Stacked fields
- Simple, mobile-friendly

<!-- Vertical Form -->

```
<div class="mb-4">
  <form >
    <div class="mb-3">
      <label class="form-label">Email</label>
      <input type="email" class="form-control" placeholder="Enter email">
    </div>
    <div class="mb-3">
      <label class="form-label">Password</label>
      <input type="password" class="form-control" placeholder="Enter password">
    </div>
    <button type="submit" class="btn btn-primary">Go</button>
  </form>
</div>
```

Email

Password

Horizontal Layout

- Label and input aligned in one row
- Uses grid (row, col-*)
- Best for structured, long forms

<!-- Horizontal Form -->

```
<form>
<div class="row mb-3">
  <div class="col-sm-10">
    <label class="col-form-label">Email</label>
  </div>
  <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">Go</button>
</form>
```

Email

Password

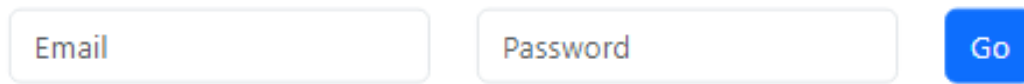
Inline Layout

- Multiple fields on one line
- Compact, fast input
- Uses grid (row, col-*)

<!-- Inline Form -->

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

A visual representation of the HTML code above. It shows a form with three elements: a text input field with the placeholder 'Email', a password input field with the placeholder 'Password', and a blue 'Go' button. The inputs are styled with rounded corners and light gray borders.

Floating Labels

Wrap input + label inside **.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/file).
- Textarea needs **manual height**. (for effectiveness of floating label).

Form fields and classes

1. Email field with floating label

```
<div class="form-floating">
  <input type="email" class="form-control" placeholder="Email">
  <label class="fw-bold">Email Address</label>
</div>
<div class="form-text">We'll never share your email with anyone else.</div>
```

Email Address

We'll never share your email with anyone else.

Email Address

We'll never share your email with anyone else.

2. Disabled Text field with Floating Label

```
<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>
```

Username

This field is disabled

3. Readonly Plaintext

```
<label class="form-label fw-bold">User ID (Auto-assigned)</label>
<input type="text" class="form-control-plaintext" value="IN-001" readonly>
```

User ID (Auto-assigned)

IN-001

4. Dropdown with floating label

```
<div class="form-floating">
  <select class="form-select">
    <option value="">Select your city</option>
    <option>Ahmedabad</option>
    <option>Rajkot</option>
    <option>Bhavnagar</option>
    <option>Surat</option>
  </select>
  <label>City</label>
</div>
```

City
Select your city



5. Radio Buttons

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

Gender

☐ Male
☒ Female

6. Checkboxes (Inline)

```
<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" checked>
  <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>
```

Hobbies ☐ Reading ☒ Traveling ☐ Photography

7. Switch

```
<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>
```

☒ Subscribe to newsletter

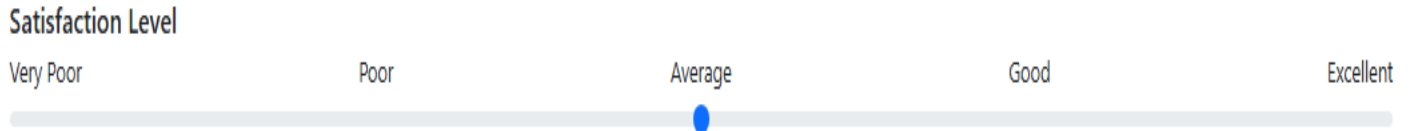
8. Range Slider

```
<h5 class="mb-1">Satisfaction Level</h5>
```

```
<div class="d-flex justify-content-between mb-1">
  <span>Very Poor</span>
  <span>Poor</span>
</div>
```

```
<span>Average</span>
<span>Good</span>
<span>Excellent</span>
</div>

<input type="range" class="form-range" min="1" max="5" step="1">
```



9. Input Group

```
<div class="input-group">
  <span class="input-group-text">Rs</span>
  <input type="text" class="form-control" placeholder="Amount">
</div>
<br>
<div class="input-group">
  <input type="text" class="form-control" placeholder="username">
  <span class="input-group-text">@</span>
  <input type="text" class="form-control" placeholder="domain">
</div>
```

Rs

@

10. Textarea with floating label

```
<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>
```


Comments

11. Submit Button

```
<button type="submit" class="btn btn-primary w-100">Submit Form</button>
```

Validations

needs-validation

Purpose:

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."**
- Almost always used together with **novalidate** to disable default browser popups.
- Does **not** show validation immediately styles appear **only after submission**.
- Requires JavaScript to check validity and apply **.was-validated**.

```
<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 "novalidate" class with "needs-validation" to disable default browser validation messages.

was-validated

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
is-valid	Marks a field as valid (green border)
is-invalid	Marks a field as invalid (red border)
valid-feedback	Normal message shown when input is valid
invalid-feedback	Normal message shown when input is invalid

Example: feedback valid and invalid message example



City

Ahmedabad 

City looks good!

Submit

State

Choose...  

Please choose a state.

```
<div class="container py-4">
```

```
<form class="was-validated" >
```

```
<!-- ===== -->
```

```
<!-- Normal Feedback Style -->
```

```
<!-- ===== -->
```

```

<div class="row">
  <!-- City (Feedback) -->
  <div class="col-md-6 mt-5">
    <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 (Feedback) -->
  <div class="col-md-6 mt-5">
    <label class="form-label">State</label>
    <select class="form-select" required>
      <option value="">Choose...</option>
      <option>Gujarat</option>
      <option>Maharashtra</option>
      <option>Punjab</option>
      <option>Karnataka</option>
    </select>
    <div class="valid-feedback">State selected.</div>
    <div class="invalid-feedback">Please choose a state.</div>
  </div>
</div>
<div class="row">
  <!-- Submit -->
  <div class="col">
    <button class="btn btn-primary" type="submit">Submit</button>
  </div>
</div>
</form>
</div>

```

If you use was-validated, then is-valid and is-invalid are NOT required.
 Bootstrap will apply them automatically based on validity.

Example

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

User Information

Username

This field is disabled

Email Address

We'll never share your email with anyone else.

User ID (Auto-assigned)

IN-001

Location

City

Select your city

Personal Details

Gender

☐ Male

☐ Female

☐ Other

Hobbies

☐ Reading

☐ Traveling

☐ Photography

Preferences

☒ Subscribe to newsletter

Satisfaction Level

Very Poor

Poor

Average

Good

Excellent

Donation

Rs

Amount

First and last name

Comments

Submit Form

```
<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>
```

```
</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>
  </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>

```

```

<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>
<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>

```



```

    </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>

```

1. Layout & Grid Classes

Class	Purpose
container	Creates a centered, responsive layout wrapper
row	Creates a horizontal group of columns
col, col-md-4, col-sm-6, col-md-8, col-md-12	Controls column width on different screen sizes
p-5	Adds padding (all sides = 5 units)
mt-4, mb-3, mb-4, mb-5	Adds vertical margin spacing

Class	Purpose
w-100	Sets element width to 100%

2. Form Control Classes

Class	Purpose
form-control	Styles text inputs, email fields, passwords and textareas
form-control-plaintext	Displays read-only text aligned like normal inputs
form-label	Styles labels above inputs
form-select	Styles dropdown/select elements
form-text	Displays helper text under inputs

3. Floating Label Classes

Class	Purpose
form-floating	Enables floating label design for inputs/selects/textareas
placeholder=""	Required for floating labels to work properly (attribute, not a class)

4. Radio Buttons & Checkboxes

Class	Purpose
form-check	Wrapper for radio/checkbox groups
form-check-input	Styles radio/checkbox elements
form-check-label	Label for radio/checkbox
form-check-inline	Places radio/checkbox elements side-by-side

5. Switch Control

Class	Purpose
form-switch	Converts checkbox into an on/off toggle switch

6. Input Group Classes

Class	Purpose
input-group	Groups inputs with prepend/append elements
input-group-text	Prefix or suffix text attached to an input (e.g., Rs, First name)

7. Range Slider

Class	Purpose
form-range	Styles a slider (range input)

8 Important HTML Attributes (not classes)

Attribute	Purpose
disabled	Makes an input uneditable
readonly	Shows data but prevents editing
placeholder	Required in floating labels