

Bootstrap

What is Bootstrap?

Bootstrap is an open-source front-end framework used for building responsive and visually appealing websites and web applications. It provides pre-designed CSS and JavaScript components like buttons, modals, navbars, forms, and many more that are easy to use and customize. Bootstrap enables developers to create flexible and mobile-first designs that work on different screen sizes and devices, without needing to start from scratch.

Why use Bootstrap?

- Responsive design: Bootstrap's grid system and components automatically adjust to different screen sizes and devices (desktops, tablets, mobile phones).
- Pre-designed components: It provides a wide variety of pre-built components that can be easily integrated into your projects (e.g., buttons, cards, forms).
- Consistency: Bootstrap ensures a consistent design throughout your website, regardless of the developer.
- Cross-browser compatibility: Websites built with Bootstrap work well on all modern browsers (Chrome, Firefox, Safari, etc.).
- Easy to use: Bootstrap simplifies the development process by providing a library of ready-to-use CSS classes and JavaScript components.

Setting up Bootstrap (CDN & Local Installation)

1. Using Bootstrap with CDN (Content Delivery Network)

- A CDN is an online server that hosts Bootstrap files, and you can include them directly in your project without downloading or hosting them locally.
- CDN Setup: You simply need to include the following `<link>` and `<script>` tags in the `<head>` and before the closing `</body>` tag of your HTML file.

`<!-- Bootstrap CSS (in <head>) -->`

```
<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
```

`<!-- Bootstrap JS (before </body>) -->`

```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.1/umd/popper.min.js"></script>
```

```
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
```

This method is simple and fast, but you depend on the availability of the internet and the external server.

2. Local Installation (Downloading Bootstrap)

If you want to use Bootstrap without relying on a CDN, you can download it and include it in your project files.

Steps:

1. Download the Bootstrap files from <https://getbootstrap.com> (choose the compiled version or the source).
2. Extract the files, and place the CSS and JS files in your project folder.
3. Link them in your HTML file:

```
<!-- Bootstrap CSS (in <head>) -->
<link href="css/bootstrap.min.css" rel="stylesheet">

<!-- Bootstrap JS (before </body>) -->
<script src="js/bootstrap.bundle.min.js"></script>
```

Bootstrap Grid System (12-column layout, container, row, column)

The Bootstrap Grid System is based on a 12-column layout. It helps developers create complex layouts by arranging content inside rows and columns, ensuring responsive designs that work well on any screen size.

1. Definitions:

Container: A wrapper that holds rows and columns. It ensures the content is aligned properly and remains within the limits of the screen. There are two types:

- .container: Fixed-width container that adjusts based on the screen size.
- .container-fluid: A full-width container that spans the entire width of the screen.

Row: A wrapper inside the container to hold columns. It's used to align columns horizontally.

Column: A section that holds content. Columns are placed inside rows and have flexible widths based on the screen size.

Syntax:

```
<div class="container">
  <div class="row">
    <div class="col">Column 1</div>
    <div class="col">Column 2</div>
    <div class="col">Column 3</div>
  </div>
</div>
```

Grid classes for different screen sizes:

Bootstrap provides different grid classes to define how many columns should be used based on the screen size:

- .col-xs-*: For extra small devices (portrait phones)
- .col-sm-*: For small devices (tablets)
- .col-md-*: For medium devices (laptops)
- .col-lg-*: For large devices (desktops)
- .col-xl-*: For extra-large devices (large desktops)

Each * represents the number of columns (from 1 to 12) that the content will occupy.

Sample Code (12-column layout):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Bootstrap Grid Example</title>

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container">

  <div class="row">

    <div class="col-12 col-sm-6 col-md-4 col-lg-3">

      <div class="border p-3">Column 1</div>

    </div>

    <div class="col-12 col-sm-6 col-md-4 col-lg-3">

      <div class="border p-3">Column 2</div>

    </div>

    <div class="col-12 col-sm-6 col-md-4 col-lg-3">

      <div class="border p-3">Column 3</div>

    </div>

    <div class="col-12 col-sm-6 col-md-4 col-lg-3">

      <div class="border p-3">Column 4</div>

    </div>

  </div>

</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.1/umd/popper.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>

</html>
```

In this example:

- The container ensures the content is centered.
- The row holds 4 columns, and each column takes up a certain portion of the screen based on the viewport size (responsive behavior).

Real-Time Use Case

- A common use case for the Bootstrap Grid system is creating a responsive layout for a portfolio website. Let's say you want to create a portfolio with four projects displayed in a grid on desktop, but stacking them vertically on mobile devices. Here's how you would implement that

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Responsive Portfolio</title>

  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container mt-5">

  <h2 class="text-center">My Portfolio</h2>

  <div class="row">

    <div class="col-12 col-sm-6 col-md-3">

      <div class="card">

        

        <div class="card-body">

          <h5 class="card-title">Project 1</h5>

          <p class="card-text">Description of project 1.</p>

        </div>

      </div>

    </div>

  </div>

  <div class="col-12 col-sm-6 col-md-3">

    <div class="card">

      

      <div class="card-body">

        <h5 class="card-title">Project 2</h5>

        <p class="card-text">Description of project 2.</p>

      </div>

    </div>

  </div>

  <div class="col-12 col-sm-6 col-md-3">

    <div class="card">

      

      <div class="card-body">

        <h5 class="card-title">Project 3</h5>

        <p class="card-text">Description of project 3.</p>

      </div>

    </div>

  </div>

</div>
```

```

    </div>
  </div>
</div>
<div class="col-12 col-sm-6 col-md-3">
  <div class="card">
    
    <div class="card-body">
      <h5 class="card-title">Project 4</h5>
      <p class="card-text">Description of project 4.</p>
    </div>
  </div>
</div>
</div>
</div>
</div>
</div>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.1/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>

```

In this case:

- On large devices (like desktops), the 4 projects will be displayed side-by-side.
- On medium devices (tablets), the columns will be split into two columns.
- On small devices (phones), the columns will stack vertically.

Layout and Components

1. Containers (.container, .container-fluid)

Definition:

A container is a wrapper element that ensures your content is properly aligned and spaced within the page. It acts as a boundary for the grid system.

.container: A fixed-width container that adjusts to the screen size.

.container-fluid: A full-width container that spans the entire width of the viewport, regardless of the screen size.

Syntax:

```
<div class="container">
  Content inside a fixed-width container
</div>

<div class="container-fluid">
  Content inside a full-width container
</div>
```

Sample Code

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Bootstrap Containers</title>

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

</head>

<body>

  <div class="container bg-light border">

    <h2>Fixed Container</h2>

    <p>This container has a fixed width and adjusts at breakpoints.</p>

  </div>

  <div class="container-fluid bg-secondary text-white mt-3 p-3">

    <h2>Fluid Container</h2>

    <p>This container stretches across the full width of the viewport. </p>

  </div>

</body>

</html>
```

Real-Time Use Case

- .container: Used for centralizing content on a webpage, ensuring proper spacing.
- .container-fluid: Used when you need a full-width layout, such as in dashboards or landing pages.

Grid System (Responsive Layouts, Breakpoints)

Definition

- The Bootstrap grid system helps create flexible and responsive layouts using rows and columns. It uses a 12-column layout and adjusts based on screen sizes (breakpoints).

Syntax

```
<div class="container">
  <div class="row">
    <div class="col-md-6">Column 1</div>
    <div class="col-md-6">Column 2</div>
  </div>
</div>
```

Sample Code

```
<div class="container">
  <div class="row">
    <div class="col-md-4 bg-primary text-white p-3">Column 1</div>
    <div class="col-md-4 bg-secondary text-white p-3">Column 2</div>
    <div class="col-md-4 bg-success text-white p-3">Column 3</div>
  </div>
</div>
```

Breakpoints

Breakpoint	Prefix	Max Width
xs	None	<576px
sm	col-sm-	≥576px
md	col-md-	≥768px
lg	col-lg-	≥992px
xl	col-xl-	≥1200px
xxl	col-xxl-	≥1400px

Real-Time Use Case

- Used for designing responsive webpages where content rearranges based on screen sizes, such as e-commerce product listings.

Bootstrap Utilities (Margin, Padding, Display, Flex, Colors)

Bootstrap provides utility classes for quick styling of elements, making it easier to manage spacing, alignment, and layout without writing custom CSS.

1. Margin (m)

Definition:

- The margin utility in Bootstrap allows you to control the outer spacing of an element.

Syntax:

```
m{side}-{size}
```

{side} → t (top), b (bottom), l (left), r (right), x (left & right), y (top & bottom), empty (all sides)

{size} → 0, 1, 2, 3, 4, 5, auto

Sample Code:

```
<div class="m-3">Margin on all sides</div>
<div class="mt-5">Top Margin</div>
<div class="mx-auto">Centered using Auto Margin</div>
```

Real-time Use Case:

Used to create spacing between elements, such as buttons and headings.

2. *Padding (p)*

Definition:

- Padding utility controls the inner spacing of an element.

Syntax:

```
p{side}-{size}
```

Sample Code:

```
<div class="p-4">Padding on all sides</div>
<div class="px-5">Horizontal Padding</div>
```

Real-time Use Case:

Used inside cards, buttons, and containers to provide consistent spacing.

3. *Display (d)*

Definition:

- The display utility changes the display property of elements.

Syntax:

```
d-{value}
```

- {value} → block, inline, inline-block, flex, grid, none

Sample Code:

```
<div class="d-block">Block Element</div>
<span class="d-inline">Inline Element</span>
<div class="d-none">This will be hidden</div>
```

Real-time Use Case:

Used to hide/show elements dynamically (e.g., mobile navigation menu).

4. Flexbox (*d-flex*)

Definition:

- Bootstrap provides a powerful d-flex utility for handling layouts using CSS Flexbox.

Syntax:

```
d-flex
```

Additional properties:

- justify-content-{value} (start, center, end, between, around)
- align-items-{value} (start, center, end, baseline, stretch)

Sample Code:

```
<div class="d-flex justify-content-between">  
  <div>Item 1</div>  
  <div>Item 2</div>  
</div>
```

Real-time Use Case:

- Used to align elements inside a navbar, cards, and form layouts.

5. Colors (*text-{color}* and *bg-{color}*)

Definition:

- Bootstrap provides predefined color classes for text and background.

Syntax:

```
text-{color}  
bg-{color}
```

Available colors: primary, secondary, success, danger, warning, info, light, dark

Sample Code:

```
<p class="text-primary">This is primary text</p>  
<div class="bg-warning p-3">Warning Background</div>
```

Real-time Use Case:

- Used to highlight alerts, buttons, and different sections of a webpage.

Typography in Bootstrap

Typography in Bootstrap includes predefined styles for headings, text alignment, and text utilities to enhance readability and consistency across devices.

Headings

- Bootstrap provides styles for headings (h1 to h6) with responsive font sizes.

Syntax:

```
<h1 class="h1">Heading 1</h1>
<h2 class="h2">Heading 2</h2>
<h3 class="h3">Heading 3</h3>
<h4 class="h4">Heading 4</h4>
<h5 class="h5">Heading 5</h5>
<h6 class="h6">Heading 6</h6>
```

Real-Time Use Case:

- Using Bootstrap heading classes ensures consistent typography across different pages of a website.

Text Alignment

- Bootstrap provides text alignment classes to control text placement.

Syntax:

```
<p class="text-start">Left aligned text.</p>
<p class="text-center">Center aligned text.</p>
<p class="text-end">Right aligned text.</p>
```

Responsive Alignment:

```
<p class="text-sm-start text-md-center text-lg-end">
```

This text is left-aligned on small screens, center-aligned on medium, and right-aligned on large screens.

```
</p>
```

Real-Time Use Case:

- Used in articles, blogs, or dashboards to align content dynamically based on the screen size.

Text Utilities

- Bootstrap provides various text utilities for styling.

Syntax:

Font Weight & Italic:

```
<p class="fw-bold">Bold text</p>
<p class="fw-light">Light text</p>
<p class="fst-italic">Italic text</p>
```

Text Colors

```
<p class="text-primary">Primary color</p>
<p class="text-success">Success color</p>
<p class="text-danger">Danger color</p>
```

Text Wrapping & Truncation

```
<p class="text-wrap">This text will wrap properly.</p>
<p class="text-nowrap">This text will not wrap.</p>
<p class="text-truncate" style="max-width: 200px; overflow: hidden;">
  This is a long text that will be truncated if it exceeds the container width.
</p>
```

Real-Time Use Case:

- Useful in dashboard designs, e-commerce platforms, and mobile-friendly websites to enhance text appearance.

Sample Code with Bootstrap Typography

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Typography</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body class="p-4">
<h1 class="h1 text-primary">Bootstrap Typography Example</h1>
<h2 class="h2 text-center">Text Alignment</h2>
  <p class="text-start">Left aligned text.</p>
  <p class="text-center">Center aligned text.</p>
  <p class="text-end">Right aligned text.</p>
<h2 class="h2 text-success">Font Weight & Italic</h2>
  <p class="fw-bold">This is bold text.</p>
  <p class="fw-light">This is light text.</p>
  <p class="fst-italic">This is italic text.</p>
<h2 class="h2 text-danger">Text Utilities</h2>
  <p class="text-wrap">This text will wrap properly inside its container.</p>
  <p class="text-nowrap">This text will not wrap and will overflow if needed.</p>
</body>
</html>
```

Summary

Feature	Class
Headings	<code>.h1</code> , <code>.h2</code> , <code>.h3</code> , <code>.h4</code> , <code>.h5</code> , <code>.h6</code>
Text Alignment	<code>.text-start</code> , <code>.text-center</code> , <code>.text-end</code>
Font Weight	<code>.fw-bold</code> , <code>.fw-light</code> , <code>.fw-normal</code>
Font Style	<code>.fst-italic</code> , <code>.fst-normal</code>
Text Colors	<code>.text-primary</code> , <code>.text-success</code> , <code>.text-danger</code>
Text Wrapping	<code>.text-wrap</code> , <code>.text-nowrap</code> , <code>.text-truncate</code>

Forms and Tables

Bootstrap provides beautiful, responsive, and accessible form elements such as text inputs, checkboxes, radio buttons, select dropdowns, and validation styles.

Bootstrap Forms (.form-group, .form-control)

- Forms in Bootstrap are structured using `.form-group` (for grouping form elements) and `.form-control` (for styling inputs).

Syntax

```
<form>
  <div class="form-group">
    <label for="name">Name:</label>
    <input type="text" class="form-control" id="name" placeholder="Enter your name">
  </div>
</form>
```

Real-Time Use Case

- Used in registration, login, and contact forms to create visually appealing input fields.

Input Groups

- Bootstrap input groups allow adding icons, text, or buttons next to inputs.

Syntax

```
<div class="input-group mb-3">
  <span class="input-group-text">@</span>
  <input type="text" class="form-control" placeholder="Username">
</div>

<div class="input-group">
  <input type="text" class="form-control" placeholder="Recipient's username">
  <button class="btn btn-primary" type="button">Send</button>
</div>
```

Real-Time Use Case

- Used in search bars, email fields with "@" icon, and password inputs with show/hide buttons.

Checkboxes & Radio Buttons

- Bootstrap provides easy styling for checkboxes and radio buttons.

Syntax

```
<!-- Checkboxes -->
<div class="form-check">
  <input class="form-check-input" type="checkbox" id="check1">
  <label class="form-check-label" for="check1">Subscribe to newsletter</label>
</div>

<!-- Radio Buttons -->
<div class="form-check">
  <input class="form-check-input" type="radio" name="gender" id="male">
  <label class="form-check-label" for="male">Male</label>
</div>

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

Real-Time Use Case

- Used in surveys, preference selections, and forms with multiple-choice options.

Select Dropdowns

- Bootstrap provides a styled dropdown selection.

Syntax

```
<select class="form-select">
  <option selected>Select your country</option>
  <option value="1">USA</option>
  <option value="2">India</option>
  <option value="3">UK</option>
</select>
```

Real-Time Use Case

- Used in country selection, product filtering, and payment method selection.

Sample Project: Bootstrap Form with Validation & Input Groups

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Bootstrap Form Example</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

</head>

<body class="p-4">

  <div class="container">

    <h2 class="text-center text-primary">User Registration</h2>

    <form class="needs-validation" novalidate>

      <div class="mb-3">

        <label for="username" class="form-label">Username</label>

        <div class="input-group">

          <span class="input-group-text">@</span>

          <input type="text" class="form-control" id="username" required>

          <div class="invalid-feedback">Username is required.</div>

        </div>

      </div>

      <div class="mb-3">

        <label for="email" class="form-label">Email address</label>

        <input type="email" class="form-control" id="email" required>

        <div class="invalid-feedback">Please enter a valid email.</div>

      </div>

      <div class="mb-3">

        <label class="form-label">Gender</label><br>

        <div class="form-check form-check-inline">

          <input class="form-check-input" type="radio" name="gender" id="male" required>

          <label class="form-check-label" for="male">Male</label>

        </div>

        <div class="form-check form-check-inline">

          <input class="form-check-input" type="radio" name="gender" id="female" required>

          <label class="form-check-label" for="female">Female</label>

        </div>

      </div>

      <div class="mb-3">

        <label for="country" class="form-label">Country</label>

        <select class="form-select" id="country" required>
```

```
<option selected disabled>Select your country</option>
<option value="1">USA</option>
<option value="2">India</option>
<option value="3">UK</option>
</select>
<div class="invalid-feedback">Please select a country.</div>
</div>
<div class="mb-3 form-check">
  <input type="checkbox" class="form-check-input" id="agree" required>
  <label class="form-check-label" for="agree">I agree to the terms and conditions</label>
  <div class="invalid-feedback">You must agree before submitting.</div>
</div>

<button type="submit" class="btn btn-primary w-100">Register</button>
</form>
</div>
<script>
(function() {
  'use strict';
  var forms = document.querySelectorAll('.needs-validation');

  Array.prototype.slice.call(forms).forEach(function(form) {
    form.addEventListener('submit', function(event) {
      if (!form.checkValidity()) {
        event.preventDefault();
        event.stopPropagation();
      }
      form.classList.add('was-validated');
    }, false);
  });
})();
</script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Summary

Feature	Bootstrap Class
Form Grouping	<code>.form-group</code>
Styled Input	<code>.form-control</code>
Input Groups	<code>.input-group</code> , <code>.input-group-text</code>
Checkboxes	<code>.form-check</code> , <code>.form-check-input</code> , <code>.form-check-label</code>
Radio Buttons	<code>.form-check-input</code> , <code>.form-check-label</code>
Select Dropdown	<code>.form-select</code>
Validation	<code>.needs-validation</code> , <code>.was-validated</code> , <code>.invalid-feedback</code>

Bootstrap Tables

- Bootstrap provides pre-styled tables with additional features like striped rows, borders, and hover effects to improve readability and usability.

Basic Bootstrap Table (.table)

- Bootstrap applies basic styling to tables using the `.table` class.

Syntax

```
<table class="table">
  <thead>
    <tr>
      <th>#</th>
      <th>Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>John Doe</td>
      <td>john@example.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Jane Smith</td>
      <td>jane@example.com</td>
    </tr>
  </tbody>
</table>
```


Real-Time Use Case

- Used in admin dashboards, reports, and student or employee records.

Striped Table (.table-striped)

- Adds alternating row colors for better readability.

Syntax

```
<table class="table table-striped">
  <thead>
    <tr>
      <th>#</th>
      <th>Name</th>
      <th>City</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Emma Watson</td>
      <td>New York</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Chris Evans</td>
      <td>Los Angeles</td>
    </tr>
  </tbody>
</table>
```

Real-Time Use Case

- Used in large datasets and transaction logs to improve row differentiation.

Bordered Table (.table-bordered)

- Adds borders around all table cells.

Syntax

```
<table class="table table-bordered">
  <thead>
    <tr>
      <th>ID</th>
      <th>Product</th>
      <th>Price</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Laptop</td>
      <td>$999</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Phone</td>
      <td>$499</td>
    </tr>
  </tbody>
</table>
```

Real-Time Use Case

- Used in invoice tables, pricing tables, and financial statements.

Hoverable Table (.table-hover)

- Highlights a row when the user hovers over it.

Syntax

```
<table class="table table-hover">
  <thead>
    <tr>
      <th>#</th>
      <th>Employee</th>
      <th>Department</th>
    </tr>
```

```
</thead>
<tbody>
  <tr>
    <td>1</td>
    <td>Michael Scott</td>
    <td>Sales</td>
  </tr>
  <tr>
    <td>2</td>
    <td>Pam Beesly</td>
    <td>Reception</td>
  </tr>
</tbody>
</table>
```

Real-Time Use Case

- Used in interactive dashboards, user lists, and product inventories.

Responsive Table (.table-responsive)

- Makes tables scrollable on small screens.

Syntax

```
<div class="table-responsive">
  <table class="table table-bordered">
    <thead>
      <tr>
        <th>#</th>
        <th>Name</th>
        <th>Address</th>
        <th>Phone</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>John Doe</td>
        <td>123 Main St</td>
```

```
<td>555-555-5555</td>
</tr>
<tr>
<td>2</td>
<td>Jane Doe</td>
<td>456 Elm St</td>
<td>555-555-1234</td>
</tr>
</tbody>
</table>
</div>
```

Real-Time Use Case

- Used in mobile-friendly web applications where tables contain large amounts of data.

Full Example: Employee List with Striped, Hover, and Bordered Table

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Table Example</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body class="p-4">
  <div class="container">
    <h2 class="text-center text-primary">Employee List</h2>
    <table class="table table-striped table-bordered table-hover">
      <thead class="table-dark">
        <tr>
          <th>ID</th>
          <th>Name</th>
          <th>Position</th>
          <th>Salary</th>
        </tr>
      </thead>
      <tbody>
```

```
<tr>

  <td>101</td>

  <td>John Doe</td>

  <td>Manager</td>

  <td>$80,000</td>

</tr>

<tr>

  <td>102</td>

  <td>Sarah Smith</td>

  <td>Developer</td>

  <td>$75,000</td>

</tr>

<tr>

  <td>103</td>

  <td>Mark Johnson</td>

  <td>Designer</td>

  <td>$65,000</td>

</tr>

</tbody>

</table>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>
```

Summary

Feature	Bootstrap Class
Basic Table	<code>.table</code>
Striped Rows	<code>.table-striped</code>
Bordered Table	<code>.table-bordered</code>
Hover Effect	<code>.table-hover</code>
Responsive Table	<code>.table-responsive</code>

Bootstrap Navbar (Fixed, Sticky, Responsive)

- The Bootstrap Navbar is a responsive navigation component that adapts to different screen sizes. It supports fixed positioning, sticky behavior, and full responsiveness.

Basic Responsive Navbar

- The .navbar class is used to create a responsive navigation bar, while .navbar-expand-* defines its collapse behavior on smaller screens.

Syntax

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Brand</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <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>
        <li class="nav-item"><a class="nav-link" href="#">Services</a></li>
        <li class="nav-item"><a class="nav-link" href="#">Contact</a></li>
      </ul>
    </div>
  </div>
</nav>
```

Real-Time Use Case

- Used in company websites, e-commerce sites, blogs, and web apps for navigation.

Fixed Navbar (.fixed-top)

- A fixed navbar stays at the top of the page even when scrolling.

Syntax

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
  <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="#navbarNav">
  <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNav">
  <ul class="navbar-nav ms-auto">
    <li class="nav-item"><a class="nav-link" href="#">Home</a></li>
    <li class="nav-item"><a class="nav-link" href="#">Services</a></li>
    <li class="nav-item"><a class="nav-link" href="#">Contact</a></li>
  </ul>
</div>
</div>
</nav>
```

Real-Time Use Case

- Used in SaaS dashboards, news websites, and online portals where navigation should always be accessible.

Sticky Navbar (.sticky-top)

- A sticky navbar sticks to the top only after scrolling past it.

Syntax

```
<nav class="navbar navbar-expand-lg navbar-primary bg-primary sticky-top">
  <div class="container-fluid">
    <a class="navbar-brand text-white" href="#">Sticky Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ms-auto">
        <li class="nav-item"><a class="nav-link text-white" href="#">Home</a></li>
        <li class="nav-item"><a class="nav-link text-white" href="#">About</a></li>
        <li class="nav-item"><a class="nav-link text-white" href="#">Contact</a></li>
      </ul>
    </div>
  </div>
</nav>
```

Real-Time Use Case

- Used in long scrolling pages like blogs, landing pages, and product pages.

Full Example: Fixed Navbar with Responsive Design

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Navbar</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body>
<!-- Fixed Navbar -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
  <div class="container">
    <a class="navbar-brand" href="#">Brand</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ms-auto">
        <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>
        <li class="nav-item"><a class="nav-link" href="#">Services</a></li>
        <li class="nav-item"><a class="nav-link" href="#">Contact</a></li>
      </ul>
    </div>
  </div>
</nav>
<!-- Page Content -->
<div class="container mt-5 pt-5">
  <h1>Welcome to My Website</h1>
  <p>This is an example of a Bootstrap fixed navbar.</p>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```


Summary

Feature	Bootstrap Class
Basic Navbar	<code>.navbar</code>
Fixed Navbar	<code>.fixed-top</code>
Sticky Navbar	<code>.sticky-top</code>
Responsive Navbar	<code>.navbar-expand-lg</code>

Bootstrap Buttons (.btn, Button Groups, Outline Buttons)

- Bootstrap provides various buttons for different use cases, including primary actions, grouped buttons, and outlined styles.

Bootstrap Buttons (.btn)

- The .btn class is used to style buttons, and different contextual classes define their appearance.

Syntax

```
<button class="btn btn-primary">Primary Button</button>
<button class="btn btn-secondary">Secondary Button</button>
<button class="btn btn-success">Success Button</button>
<button class="btn btn-danger">Danger Button</button>
<button class="btn btn-warning">Warning Button</button>
<button class="btn btn-info">Info Button</button>
<button class="btn btn-light">Light Button</button>
<button class="btn btn-dark">Dark Button</button>
```

Real-Time Use Case

- Used for call-to-action buttons, form submissions, and navigational links in websites and applications.

Button Sizes (.btn-lg, .btn-sm)

- Bootstrap provides different button sizes for various UI needs.

Syntax

```
<button class="btn btn-primary btn-lg">Large Button</button>
<button class="btn btn-secondary">Default Button</button>
<button class="btn btn-success btn-sm">Small Button</button>
```

Real-Time Use Case

- Used to highlight primary actions with larger buttons or reduce prominence with smaller ones.

Outline Buttons (.btn-outline-)*

- Outline buttons have a transparent background and colored borders.

Syntax

```
<button class="btn btn-outline-primary">Primary</button>
<button class="btn btn-outline-secondary">Secondary</button>
<button class="btn btn-outline-success">Success</button>
<button class="btn btn-outline-danger">Danger</button>
<button class="btn btn-outline-warning">Warning</button>
<button class="btn btn-outline-info">Info</button>
<button class="btn btn-outline-dark">Dark</button>
```

Real-Time Use Case

- Used in minimal UI designs, filter buttons, and secondary actions.

Button Groups (.btn-group)

- Button groups are used to group multiple buttons together.

Syntax

```
<div class="btn-group">
  <button class="btn btn-primary">Left</button>
  <button class="btn btn-primary">Middle</button>
  <button class="btn btn-primary">Right</button>
</div>
```

Real-Time Use Case

- Used in toolbar actions, grouped controls, and pagination buttons.

Full Example: Bootstrap Buttons with Groups

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Buttons</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body class="p-5">
<h2>Bootstrap Buttons</h2>
  <button class="btn btn-primary">Primary</button>
  <button class="btn btn-danger btn-lg">Large Danger</button>
  <button class="btn btn-outline-success">Outline Success</button>
```

```
<h2 class="mt-4">Button Groups</h2>

<div class="btn-group">
  <button class="btn btn-dark">Left</button>
  <button class="btn btn-dark">Middle</button>
  <button class="btn btn-dark">Right</button>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Summary Table

Feature	Bootstrap Class
Basic Button	<code>.btn</code>
Primary Button	<code>.btn-primary</code>
Outline Button	<code>.btn-outline-*</code>
Large Button	<code>.btn-lg</code>
Small Button	<code>.btn-sm</code>
Button Group	<code>.btn-group</code>

Bootstrap Alerts & Badges

Bootstrap provides alerts for displaying messages and badges for adding small notifications or labels. These components help in creating user-friendly notifications, warnings, and labels.

Bootstrap Alerts (.alert)

- Alerts are used to display feedback messages like success, errors, warnings, or informational messages.

Syntax

```
<div class="alert alert-primary">This is a primary alert!</div>
<div class="alert alert-success">Success! Your action was completed.</div>
<div class="alert alert-danger">Error! Something went wrong.</div>
<div class="alert alert-warning">Warning! Check your input.</div>
<div class="alert alert-info">Info! Here is some important information.</div>
```

Real-Time Use Case

- Used for form validation messages, system notifications, and error messages in applications.

Alerts with Close Button

- Bootstrap alerts can be dismissed by adding the `.alert-dismissible` and `data-bs-dismiss="alert"` attributes.

```
<div class="alert alert-warning alert-dismissible fade show" role="alert">
  Warning! Check your input.
  <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
```

Real-Time Use Case

- Used for temporary notifications where users can dismiss the message manually.

Bootstrap Badges (.badge)

- Badges are used to display numerical indicators or labels for elements like notifications, messages, and categories.

Syntax

```
<h1>Notifications <span class="badge bg-danger">5</span></h1>
<button class="btn btn-primary">
  Messages <span class="badge bg-light text-dark">3</span>
</button>
```

Real-Time Use Case

- Used in notification counters, unread messages, status indicators, and labels for categories.

Full Example: Bootstrap Alerts & Badges

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Alerts & Badges</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body class="p-5">
  <h2>Bootstrap Alerts</h2>
  <div class="alert alert-success">Success! Your action was completed.</div>
  <div class="alert alert-danger">Error! Something went wrong.</div>
  <div class="alert alert-info alert-dismissible fade show" role="alert">
    Info Alert! Click close button to dismiss.
    <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
  </div>
```

```
<h2 class="mt-4">Bootstrap Badges</h2>

<p>Unread Messages: <span class="badge bg-danger">8</span></p>

<button class="btn btn-primary">Inbox <span class="badge bg-light text-dark">4</span></button>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>
```

Summary Table

Feature	Bootstrap Class
Basic Alert	<code>.alert</code>
Success Alert	<code>.alert-success</code>
Dismissible Alert	<code>.alert-dismissible</code>
Badge	<code>.badge</code>
Badge Colors	<code>.bg-primary</code> , <code>.bg-danger</code> , <code>.bg-success</code>

Bootstrap Cards

Bootstrap cards are flexible content containers that include titles, text, images, links, and buttons. They are commonly used for displaying articles, product details, user profiles, and dashboards.

Bootstrap Basic Card (.card)

- A simple card consists of a header, body, and footer.

Syntax

```
<div class="card" style="width: 18rem;">
  <div class="card-body">
    <h5 class="card-title">Card Title</h5>
    <p class="card-text">This is a simple card with text and a button.</p>
    <a href="#" class="btn btn-primary">Read More</a>
  </div>
</div>
```

Real-Time Use Case

- Used for displaying user information, product descriptions, and blog previews.

Card with Image (.card-img-top)

- You can add images to the top of the card.

Syntax

```
<div class="card" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Card with Image</h5>
    <p class="card-text">This card includes an image at the top.</p>
    <a href="#" class="btn btn-success">Learn More</a>
  </div>
</div>
```

Real-Time Use Case

- Used in e-commerce websites to showcase products with images.

Card Layout (.card-deck, .card-group, .row)

- Cards can be displayed in rows and columns for layout purposes.

Syntax

```
<div class="row">
  <div class="col-md-4">
    <div class="card">
      
      <div class="card-body">
        <h5 class="card-title">Card 1</h5>
        <p class="card-text">Description of card 1.</p>
      </div>
    </div>
  </div>
</div>
<div class="col-md-4">
  <div class="card">
    
    <div class="card-body">
      <h5 class="card-title">Card 2</h5>
      <p class="card-text">Description of card 2.</p>
    </div>
  </div>
</div>
</div>
```

Real-Time Use Case

- Used for blog layouts, team member sections, or dashboard cards.

Card with Image Overlay (.card-img-overlay)

- An image overlay places text on top of an image.

Syntax

```
<div class="card text-white">
  
  <div class="card-img-overlay">
    <h5 class="card-title">Image Overlay</h5>
    <p class="card-text">Text appears over the image.</p>
  </div>
</div>
```

Real-Time Use Case

- Used for advertisements, banners, and hero sections.

Full Example: Bootstrap Cards

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Cards</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body class="p-5">
  <h2>Basic Card</h2>
  <div class="card" style="width: 18rem;">
    <div class="card-body">
      <h5 class="card-title">Card Example</h5>
      <p class="card-text">This is a simple Bootstrap card.</p>
      <a href="#" class="btn btn-primary">More Info</a>
    </div>
  </div>

  <h2 class="mt-4">Card with Image</h2>
  <div class="card" style="width: 18rem;">
    
    <div class="card-body">
      <h5 class="card-title">Card with Image</h5>
```

```
<p class="card-text">This card includes an image.</p>
<a href="#" class="btn btn-success">Details</a>
</div>
</div>
```

<h2 class="mt-4">Card Layout</h2>

```
<div class="row">
  <div class="col-md-4">
    <div class="card">
      
      <div class="card-body">
        <h5 class="card-title">Card 1</h5>
        <p class="card-text">Description of card 1.</p>
      </div>
    </div>
  </div>
  <div class="col-md-4">
    <div class="card">
      
      <div class="card-body">
        <h5 class="card-title">Card 2</h5>
        <p class="card-text">Description of card 2.</p>
      </div>
    </div>
  </div>
</div>
```

<h2 class="mt-4">Image Overlay Card</h2>

```
<div class="card text-white">
  
  <div class="card-img-overlay">
    <h5 class="card-title">Image Overlay</h5>
    <p class="card-text">Text appears over the image.</p>
  </div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```


Summary Table

Feature	Bootstrap Class
Basic Card	<code>.card</code>
Card with Image	<code>.card-img-top</code>
Card Layout	<code>.row .col-md-*</code>
Image Overlay	<code>.card-img-overlay</code>
Card Body	<code>.card-body</code>

Card Hover Effect (Scale on Hover)

- This effect makes the card slightly enlarge when hovered.

Syntax

```
<style>
  .card-hover:hover {
    transform: scale(1.05);
    transition: 0.3s ease-in-out;
  }
</style>

<div class="card card-hover" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Hover Effect</h5>
    <p class="card-text">This card scales up slightly when hovered.</p>
  </div>
</div>
```

Real-Time Use Case

- Used in product showcases, portfolio cards, and blog previews.

Card with Shadow Effect on Hover

- This effect adds a shadow when hovered, giving a lifting effect.

Syntax

```
<style>
  .card-shadow:hover {
    box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
    transition: 0.3s ease-in-out;
  }
</style>

<div class="card card-shadow" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Shadow on Hover</h5>
    <p class="card-text">A shadow appears when you hover.</p>
  </div>
</div>
```

Real-Time Use Case

- Used in e-commerce websites for product listings.

Flip Animation on Hover

- This makes the card flip when hovered.

Syntax

```
<style>
```

```
.flip-card {  
  perspective: 1000px;  
}
```

```
.flip-card-inner {  
  position: relative;  
  width: 100%;  
  height: 100%;  
  transform-style: preserve-3d;  
  transition: transform 0.6s;  
}
```

```
.flip-card:hover .flip-card-inner {  
  transform: rotateY(180deg);  
}
```

```
.flip-card-front, .flip-card-back {  
  position: absolute;  
  width: 100%;  
  height: 100%;  
  backface-visibility: hidden;  
}
```

```
.flip-card-back {  
  background: #007bff;  
  color: white;  
  transform: rotateY(180deg);  
}
```

```
</style>
```

```
<div class="card flip-card" style="width: 18rem; height: 200px;">
```

```
<div class="flip-card-inner">
```

```
<div class="card-body flip-card-front">
```

```
<h5 class="card-title">Hover to Flip</h5>
```

```
<p class="card-text">This card flips when hovered.</p>
</div>
<div class="card-body flip-card-back">
  <h5 class="card-title">Back Side</h5>
  <p class="card-text">This is the hidden content.</p>
</div>
</div>
</div>
```

Real-Time Use Case

- Used for business cards, quiz flashcards, and testimonials.

Full Example: Cards with Hover Effects and Animations

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Cards with Hover Effects</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
  <style>
    /* Scale on hover */
    .card-hover:hover {
      transform: scale(1.05);
      transition: 0.3s ease-in-out;
    }

    /* Shadow on hover */
    .card-shadow:hover {
      box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
      transition: 0.3s ease-in-out;
    }

    /* Flip animation */
    .flip-card {
      perspective: 1000px;
    }

    .flip-card-inner {
```

```
position: relative;
width: 100%;
height: 100%;
transform-style: preserve-3d;
transition: transform 0.6s;
}
.flip-card:hover .flip-card-inner {
  transform: rotateY(180deg);
}
.flip-card-front, .flip-card-back {
  position: absolute;
width: 100%;
height: 100%;
backface-visibility: hidden;
}
.flip-card-back {
  background: #007bff;
  color: white;
  transform: rotateY(180deg);
}
```

</style>

</head>

<body class="p-5">

<h2>Card Hover Effect</h2>

<div class="card card-hover" style="width: 18rem;">

<div class="card-body">

<h5 class="card-title">Hover Effect</h5>

<p class="card-text">This card scales up slightly when hovered.</p>

</div>

</div>

<h2 class="mt-4">Card with Shadow on Hover</h2>

<div class="card card-shadow" style="width: 18rem;">

<div class="card-body">

<h5 class="card-title">Shadow on Hover</h5>

<p class="card-text">A shadow appears when you hover.</p>

</div>

```
</div>

<h2 class="mt-4">Flip Animation Card</h2>
<div class="card flip-card" style="width: 18rem; height: 200px;">
  <div class="flip-card-inner">
    <div class="card-body flip-card-front">
      <h5 class="card-title">Hover to Flip</h5>
      <p class="card-text">This card flips when hovered.</p>
    </div>
    <div class="card-body flip-card-back">
      <h5 class="card-title">Back Side</h5>
      <p class="card-text">This is the hidden content.</p>
    </div>
  </div>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Summary Table

Animation	CSS Class / Effect
Scale on Hover	<code>transform: scale(1.05); transition: 0.3s;</code>
Shadow on Hover	<code>box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);</code>
Flip Animation	<code>transform: rotateY(180deg);</code>

Bootstrap Modals (.modal)

A modal in Bootstrap is a pop-up dialog box that appears over the main content. It is commonly used for displaying forms, alerts, confirmation messages, or additional information without navigating away from the page.

Syntax & Basic Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Modal Example</title>
```

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body>

  <!-- Button to trigger modal -->
  <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#exampleModal">
    Open Modal
  </button>

  <!-- Modal Structure -->
  <div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title" id="exampleModalLabel">Modal Title</h5>
          <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
        </div>
        <div class="modal-body">
          This is a Bootstrap modal. You can place content here!
        </div>
        <div class="modal-footer">
          <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
          <button type="button" class="btn btn-primary">Save changes</button>
        </div>
      </div>
    </div>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Real-Time Use Case

- Used for login/signup forms, delete confirmation dialogs, and pop-up notifications.

Modal with a Form (Login Modal)

<!-- Button to Open the Modal -->

```
<button type="button" class="btn btn-success" data-bs-toggle="modal" data-bs-target="#loginModal">
  Open Login Modal
</button>
```

<!-- Modal Structure -->

```
<div class="modal fade" id="loginModal" tabindex="-1" aria-labelledby="loginModalLabel" aria-
hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="loginModalLabel">Login</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <form>
          <div class="mb-3">
            <label for="email" class="form-label">Email address</label>
            <input type="email" class="form-control" id="email" placeholder="Enter email">
          </div>
          <div class="mb-3">
            <label for="password" class="form-label">Password</label>
            <input type="password" class="form-control" id="password" placeholder="Enter
password">
          </div>
          <button type="submit" class="btn btn-primary w-100">Login</button>
        </form>
      </div>
    </div>
  </div>
</div>
```

Real-Time Use Case

- Used for user login or signup pop-ups.

Centered Modal

- To center the modal vertically, add the .modal-dialog-centered class.

```
<div class="modal fade" id="centeredModal" tabindex="-1" aria-labelledby="centeredModalLabel" aria-
hidden="true">
```

```
<div class="modal-dialog modal-dialog-centered">
  <div class="modal-content">
    <div class="modal-header">
      <h5 class="modal-title" id="centeredModalLabel">Centered Modal</h5>
      <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
    </div>
    <div class="modal-body">
      This modal is vertically centered!
    </div>
  </div>
</div>
</div>
```

Real-Time Use Case

✔ Used when a modal should be focused in the middle of the screen for better user attention.

Modal with Animation Effects

Bootstrap modals fade in by default, but you can remove the .fade class to disable the effect.

For custom animations, use CSS:

```
<style>
  .modal.fade .modal-dialog {
    transform: scale(0.8);
    transition: transform 0.3s ease-in-out;
  }
  .modal.show .modal-dialog {
    transform: scale(1);
  }
</style>
```

```
<div class="modal fade" id="animatedModal" tabindex="-1" aria-labelledby="animatedModalLabel" aria-
hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="animatedModalLabel">Animated Modal</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        This modal has a custom animation effect.
      </div>
    </div>
  </div>
```



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

Real-Time Use Case

✔ Used for notifications, pop-ups, or interactive dialogs.

Summary Table

Feature	Class / Attribute Used
Basic Modal	<code>.modal</code> , <code>.modal-dialog</code> , <code>.modal-content</code>
Trigger Modal	<code>data-bs-toggle="modal" data-bs-target="#id"</code>
Close Button	<code>data-bs-dismiss="modal"</code>
Vertically Centered	<code>.modal-dialog-centered</code>
Animated Modal	<code>.fade</code> (default) or custom CSS animations
Show Modal via JS	<code>new bootstrap.Modal(element).show()</code>

Bootstrap Carousel (.carousel)

Definition

- The Bootstrap Carousel component is used to create a slideshow that cycles through different images or content. It supports automatic sliding, navigation controls, and indicators to move between slides.

Real-Time Use Case

The Bootstrap Carousel is widely used in:

- ✔ Landing pages – Display promotional banners and featured content.
- ✔ E-commerce websites – Showcase top-selling or new products.
- ✔ Portfolio websites – Present project images in a visually appealing way.
- ✔ Event websites – Display event highlights or upcoming activities.

Sample Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Carousel Example</title>

  <!-- Bootstrap CSS CDN -->
```

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

<style>
  /* Custom Styles for Carousel */
  .carousel-inner img {
    height: 400px; /* Adjust image height */
    object-fit: cover;
    border-radius: 10px;
  }
  .carousel-container {
    max-width: 800px;
    margin: auto;
    padding-top: 20px;
  }
</style>
</head>
<body>

<div class="carousel-container">
  <div id="myCarousel" class="carousel slide" data-bs-ride="carousel" data-bs-interval="3000">

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

    <!-- Carousel Items -->
    <div class="carousel-inner">
      <div class="carousel-item active">
        
      </div>
      <div class="carousel-item">
        
      </div>
    </div>
  </div>
</div>
```

```

        </div>
        <div class="carousel-item">
            
        </div>
    </div>

    <!-- Navigation Buttons -->
    <button class="carousel-control-prev" type="button" data-bs-target="#myCarousel" data-bs-slide="prev">
        <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    </button>
    <button class="carousel-control-next" type="button" data-bs-target="#myCarousel" data-bs-slide="next">
        <span class="carousel-control-next-icon" aria-hidden="true"></span>
    </button>

</div>
</div>

<!-- Bootstrap JS CDN -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

Explanation:

- ◆ Bootstrap CDN is included to load Bootstrap without installation.
- ◆ Custom CSS styles the carousel and sets a fixed height for images.
- ◆ Auto-slide is enabled with data-bs-ride="carousel" and data-bs-interval="3000".
- ◆ Navigation controls (Prev & Next buttons) allow users to switch slides.
- ◆ Three slides are added with sample placeholder images.

Bootstrap Collapse & Accordion

1. Collapse Component

- The collapse component is used to show and hide content using buttons or links. It is often used for toggling sections of a page without reloading.

Syntax for Collapse

```
<button class="btn btn-primary" type="button" data-bs-toggle="collapse" data-bs-target="#exampleCollapse">
```

Toggle Content

```
</button>
```

```
<div id="exampleCollapse" class="collapse">
```

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

This is a collapsible content section.

```
</div>
```

```
</div>
```

Real-time Use Case:

✓ FAQs sections – Show/hide answers.

✓ Read more buttons – Expand additional text.

2. Accordion Component

- The accordion component is an enhanced version of collapse, where only one section can be open at a time.

Syntax for Accordion

```
<div class="accordion" id="exampleAccordion">
```

```
<!-- Accordion Item 1 -->
```

```
<div class="accordion-item">
```

```
<h2 class="accordion-header">
```

```
<button class="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#collapseOne">
```

Section 1

```
</button>
```

```
</h2>
```

```
<div id="collapseOne" class="accordion-collapse collapse show" data-bs-parent="#exampleAccordion">
```

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

Content for Section 1.

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- Accordion Item 2 -->
```

```
<div class="accordion-item">
```

```

    <h2 class="accordion-header">
      <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-
target="#collapseTwo">
        Section 2
      </button>
    </h2>
    <div id="collapseTwo" class="accordion-collapse collapse" data-bs-parent="#exampleAccordion">
      <div class="accordion-body">
        Content for Section 2.
      </div>
    </div>
  </div>
</div>

```

Real-time Use Case:

- ✓ Multi-step Forms – Show/hide steps dynamically.
- ✓ Service Details Sections – Display more info on demand.

Complete Working Example

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Collapse & Accordion</title>

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

  <div class="container mt-4">
    <h2>Bootstrap Collapse Example</h2>

    <!-- Collapse Example -->
    <button class="btn btn-primary" type="button" data-bs-toggle="collapse" data-bs-
target="#collapseExample">

```

Toggle Collapse

</button>

<div class="collapse mt-2" id="collapseExample">

<div class="card card-body">

This is a collapsible content section.

</div>

</div>

<h2 class="mt-4">Bootstrap Accordion Example</h2>

<!-- Accordion Example -->

<div class="accordion" id="accordionExample">

<div class="accordion-item">

<h2 class="accordion-header">

<button class="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#collapseOne">

Accordion Item #1

</button>

</h2>

<div id="collapseOne" class="accordion-collapse collapse show" data-bs-parent="#accordionExample">

<div class="accordion-body">

This is the content for the first accordion item.

</div>

</div>

</div>

<div class="accordion-item">

<h2 class="accordion-header">

<button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#collapseTwo">

Accordion Item #2

</button>

</h2>

<div id="collapseTwo" class="accordion-collapse collapse" data-bs-parent="#accordionExample">

<div class="accordion-body">

This is the content for the second accordion item.

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

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

```
<!-- Bootstrap JS -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
```

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

This example demonstrates both Collapse and Accordion in action!

Bootstrap Tooltips & Popovers

1. Tooltips

Definition

- Tooltips are small, interactive pop-up messages that appear when the user hovers over an element. They provide additional information without taking up space on the UI.

Syntax for Tooltips

```
<button type="button" class="btn btn-secondary" data-bs-toggle="tooltip" title="This is a tooltip!">
    Hover over me
</button>
```

Real-time Use Case:

- ✓ Form fields – Show hints for inputs (e.g., password requirements).
- ✓ Icons & buttons – Display extra details without cluttering the UI.

2. Popovers

◆ Definition

- Popovers are like tooltips but more interactive. They support titles and rich content. Popovers require explicit user interaction (click or hover).

Syntax for Popovers

```
<button type="button" class="btn btn-info" data-bs-toggle="popover" title="Popover Title" data-bs-
content="This is a popover message!">
    Click me
</button>
```

Real-time Use Case:

- ✓ Product descriptions – Show details when clicking on an item.
- ✓ User profiles – Display extra info when clicking on a user's name.

Complete Working Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Tooltips & Popovers</title>

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

  <div class="container mt-5">
    <h2>Bootstrap Tooltips</h2>
    <button type="button" class="btn btn-secondary" data-bs-toggle="tooltip" title="This is a tooltip!">
      Hover over me
    </button>

    <h2 class="mt-4">Bootstrap Popovers</h2>
    <button type="button" class="btn btn-info" data-bs-toggle="popover" title="Popover Title" data-bs-
content="This is a popover message!">
      Click me
    </button>
  </div>

  <!-- Bootstrap JS -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

  <!-- Initialize Tooltips & Popovers -->
  <script>
    document.addEventListener("DOMContentLoaded", function() {
      var tooltipTriggerList = [].slice.call(document.querySelectorAll('[data-bs-toggle="tooltip"]'));

```



```

        tooltipTriggerList.map(function(tooltipTriggerEl) {
            return new bootstrap.Tooltip(tooltipTriggerEl);
        });

        var popoverTriggerList = [].slice.call(document.querySelectorAll('[data-bs-toggle="popover"]'));
        popoverTriggerList.map(function(popoverTriggerEl) {
            return new bootstrap.Popover(popoverTriggerEl);
        });
    });
</script>

```

```
</body>
```

```
</html>
```

Bootstrap Scrollspy

Definition

- Scrollspy is a Bootstrap feature that automatically updates navigation links based on the scroll position of the page. It highlights the active section in a navigation bar when a user scrolls through a long page.

Syntax for Scrollspy

- Add data-bs-spy="scroll" and data-bs-target="#navbar-example" to the scrollable container

```

<div data-bs-spy="scroll" data-bs-target="#navbar-example" data-bs-offset="0" class="scrollspy-example">
  <!-- Scrollable content -->
</div>

```

- Ensure the Navigation has id="navbar-example" and links contain href="#sectionID"

```

<nav id="navbar-example" class="navbar navbar-light bg-light fixed-top">
  <ul class="nav nav-pills">
    <li class="nav-item"><a class="nav-link" href="#section1">Section 1</a></li>
    <li class="nav-item"><a class="nav-link" href="#section2">Section 2</a></li>
    <li class="nav-item"><a class="nav-link" href="#section3">Section 3</a></li>
  </ul>
</nav>

```

Real-Time Use Cases

- ✓ Landing pages – Highlight active sections as users scroll.
- ✓ Documentation websites – Auto-highlight the section being read.
- ✓ One-page websites – Keep navigation linked to scrolling content.

Full Working Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Scrollspy Example</title>

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

  <style>
    body {
      position: relative;
    }
    .scrollspy-example {
      height: 400px;
      overflow-y: scroll;
      padding: 10px;
      margin-top: 60px;
      border: 1px solid #ddd;
    }
    section {
      height: 300px;
      padding: 20px;
    }
  </style>
</head>
<body data-bs-spy="scroll" data-bs-target="#navbar-example">

  <!-- Navigation -->
```

```
<nav id="navbar-example" class="navbar navbar-light bg-light fixed-top">
  <ul class="nav nav-pills">
    <li class="nav-item"><a class="nav-link active" href="#section1">Section 1</a></li>
    <li class="nav-item"><a class="nav-link" href="#section2">Section 2</a></li>
    <li class="nav-item"><a class="nav-link" href="#section3">Section 3</a></li>
  </ul>
</nav>

<!-- Scrollable Content -->
<div class="scrollspy-example">
  <section id="section1" class="bg-primary text-white">Section 1 Content</section>
  <section id="section2" class="bg-success text-white">Section 2 Content</section>
  <section id="section3" class="bg-danger text-white">Section 3 Content</section>
</div>

<!-- Bootstrap JS -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>
</html>
```

Key Takeaways

- ✓ Automatically highlights the active section in the navbar.
 - ✓ Works great for long, scrolling pages.
 - ✓ Requires a fixed navigation bar to work properly.
-