Bootstrap 5 Notes

1.1 What is Bootstrap?

- Bootstrap is a free and open-source **CSS framework** used for developing responsive and mobile-first websites.
- It includes **predefined CSS classes, components, and JavaScript plugins** to simplify web development.
- Developed by **Twitter** and initially released in **2011**.
- Provides a **grid system, typography, forms, buttons, navigation, and other UI components**.

1.2 History and Evolution of Bootstrap

- **Bootstrap 1 (2011):** Introduced as "Twitter Bootstrap" with basic styling and grid.
- **Bootstrap 2 (2012):** Added responsive design features.
- **Bootstrap 3 (2013):** Made mobile-first approach default.
- **Bootstrap 4 (2018):** Introduced Flexbox, removed older components, improved customization.
- **Bootstrap 5 (2021):** Removed jQuery dependency, introduced a new grid system, and added more utility classes.

1.3 Differences Between Bootstrap 4 and Bootstrap 5

Feature	Bootstrap 4	Bootstrap 5
jQuery Dependency	Required	Removed (pure JS)
Grid System	Based on Flexbox	Improved with CSS Grid support
Utilities API	Limited customization	More utilities and configurable classes
Forms	Custom form controls	Redesigned, better validation
Navbar	Requires `navbar-light` and `navbar-dark`	Automatic color contrast detection

---

1.4 Setting Up Bootstrap (CDN vs. Local Installation)

CDN Method (Recommended for Quick Use)

- No need to download files; just include the following **Bootstrap CDN** links in your HTML file.
- Example:

```
<!-- Bootstrap 5 CSS -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

<!-- Bootstrap 5 JS -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
```

- Pros: **Fast, always updated, no setup required**
- Cons: **Requires an internet connection**

Local Installation (Recommended for Customization)

1. Download Bootstrap from the official website: https://getbootstrap.com/
2. Extract the files and link them in your project:

```
<link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">
<script src="bootstrap/js/bootstrap.bundle.min.js"></script>
```

- Pros: **Works offline, allows customization**
- Cons: **Requires manual updates**

1.5 Bootstrap Grid System Overview

- The **Grid System** in Bootstrap **divides the page into 12 columns**, making responsive layout design easy.
- Uses **rows and columns** to create flexible layouts.

- Example:

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

- **Key Grid Classes:**
  - `.container` - Defines a fixed-width or full-width container.
  - `.row` - Defines a row to hold columns.
  - `.col` - Automatically divides space equally.
  - `.col-sm-`, `.col-md-`, `.col-lg-`, `.col-xl-` - Responsive column sizes for different screen widths.

Bootstrap 5 Grid System – Responsive Column Sizes

Bootstrap uses a twelve-column grid system to create responsive layouts. The grid adjusts based on screen size using different classes.

Responsive Column Classes

1. col-sm-* for small screens with a minimum width of 576 pixels
2. col-md-* for medium screens with a minimum width of 768 pixels
3. col-lg-* for large screens with a minimum width of 992 pixels
4. col-xl-* for extra-large screens with a minimum width of 1200 pixels
5. col-xxl-* for extra-extra-large screens with a minimum width of 1400 pixels

Each class defines how many columns a section takes on different screen sizes.

How It Works

1. If a class like col-md-6 is used, it means the element will take six columns out of twelve when the screen width is 768 pixels or larger
2. On smaller screens, the columns stack automatically unless smaller column classes are specified

3. If no specific class is provided for a screen size, it follows the rules of the closest smaller size

Examples

col-sm-* for Small Screens and Up

Used for small devices with a minimum width of 576 pixels
On screens smaller than 576 pixels, columns stack

Example of two equal-width columns on small screens and above

```html
<div class="row">
  <div class="col-sm-6">Column 1</div>
  <div class="col-sm-6">Column 2</div>
</div>
```

On screens 576 pixels and above, both columns take 50 percent width
On screens smaller than 576 pixels, columns stack

col-md-* for Medium Screens and Up

Used for medium devices with a minimum width of 768 pixels
On smaller screens, columns stack automatically

Example of one full-width column on small screens, two columns on medium screens and above

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

On screens 768 pixels and above, columns are side by side with 50 percent width
On screens smaller than 768 pixels, columns stack

col-lg-* for Large Screens and Up

Used for large devices with a minimum width of 992 pixels
Ensures wider layouts on large screens

Example of three equal-width columns on large screens, stacked on smaller screens

```
<div class="row">
  <div class="col-lg-4">Column 1</div>
  <div class="col-lg-4">Column 2</div>
  <div class="col-lg-4">Column 3</div>
</div>
```

On screens 992 pixels and above, columns take 33 percent width
On screens smaller than 992 pixels, columns stack

col-xl-* for Extra-Large Screens and Up

Used for extra-large devices with a minimum width of 1200 pixels
Great for wide desktop screens

Example of four equal-width columns on extra-large screens, stacked on smaller screens

```
<div class="row">
  <div class="col-xl-3">Column 1</div>
  <div class="col-xl-3">Column 2</div>
  <div class="col-xl-3">Column 3</div>
  <div class="col-xl-3">Column 4</div>
</div>
```

On screens 1200 pixels and above, columns take 25 percent width
On screens smaller than 1200 pixels, columns stack

col-xxl-* for Extra-Extra-Large Screens and Up

Used for very large screens with a minimum width of 1400 pixels
Helps in designing layouts for ultra-wide monitors

Example of six equal-width columns on extra-extra-large screens, stacked on smaller screens

```
<div class="row">
  <div class="col-xxl-2">Column 1</div>
  <div class="col-xxl-2">Column 2</div>
  <div class="col-xxl-2">Column 3</div>
  <div class="col-xxl-2">Column 4</div>
  <div class="col-xxl-2">Column 5</div>
  <div class="col-xxl-2">Column 6</div>
</div>
```

On screens 1400 pixels and above, columns take 16.66 percent width
On screens smaller than 1400 pixels, columns adjust or stack

Key Points to Remember

1. The Bootstrap grid system divides layouts into twelve columns
2. col-sm- col-md- col-lg- col-xl- col-xxl- create responsive layouts
3. If no column class is provided for a screen size, it follows the rules of the closest smaller size
4. On smaller screens, Bootstrap automatically stacks columns unless specified otherwise

Bootstrap 5 Grid System Notes with Examples

2.1 Understanding the 12-Column Grid System

- Bootstrap uses a **12-column grid system** to create responsive layouts.
- A row is divided into **12 equal parts**, and elements can span multiple columns.
- Columns inside a row **must add up to 12** for a proper layout.

Example: Basic 12-Column Grid

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

Explanation:

- Three columns each take 4 spaces, summing up to 12.
- `col-4` means each div occupies **4 out of 12 columns**.

2.2 Container Types (.container, .container-fluid)

Container Type	Behavior
.container	Fixed width, adjusts at breakpoints
.container-fluid	Always spans full width of the viewport
.container-{breakpoint}	Adjusts at specific breakpoints

Example: Containers

```
<div class="container bg-light">
  <p>This is inside a fixed-width container.</p>
</div>

<div class="container-fluid bg-warning">
  <p>This is inside a full-width container.</p>
</div>
```

Explanation:

- `.container` keeps content centered with **fixed max-width**.
- `.container-fluid` makes it **full-screen width**.

2.3 Grid Classes (.col-, .col-sm-, .col-md-, .col-lg-, .col-xl-, .col-xxl-)

Class	Screen Size (Min Width)
`.col-`	Auto width on all screens
`.col-sm-*`	≥ 576px (Small)
`.col-md-*`	≥ 768px (Medium)
`.col-lg-*`	≥ 992px (Large)
`.col-xl-*`	≥ 1200px (Extra Large)
`.col-xxl-*`	≥ 1400px (Extra Extra Large)

Example: Responsive Columns

```
<div class="container">
  <div class="row">
```

```
    <div class="col-sm-6 col-md-4 col-lg-3 bg-danger text-white">Responsive Column</div>
    <div class="col-sm-6 col-md-4 col-lg-3 bg-info text-white">Responsive Column</div>
    <div class="col-sm-6 col-md-4 col-lg-3 bg-warning text-white">Responsive Column</div>
    <div class="col-sm-6 col-md-12 col-lg-3 bg-success text-white">Responsive Column</div>
  </div>
</div>
```

Explanation:

- On **small screens (≥576px)** → Each column takes **50% width**.
- On **medium screens (≥768px)** → Each column takes **33% width** except the last one (100%).
- On **large screens (≥992px)** → Each column takes **25% width**.

2.4 Responsive Layouts and Breakpoints

- **Breakpoints control layout changes at different screen sizes**.
- Bootstrap adjusts column sizes automatically based on screen width.

Example: Columns Changing at Breakpoints

```
<div class="container">
  <div class="row">
    <div class="col-12 col-md-6 col-lg-4 bg-primary text-white">Column</div>
    <div class="col-12 col-md-6 col-lg-4 bg-secondary text-white">Column</div>
    <div class="col-12 col-md-12 col-lg-4 bg-success text-white">Column</div>
  </div>
</div>
```

Explanation:

- On **small screens**, each column takes **full width** (`col-12`).
- On **medium screens**, the first two columns take **half width** (`col-md-6`), last one full.
- On **large screens**, all columns take **one-third width** (`col-lg-4`).

2.5 Row and Column Gutters

- **Gutters** are spaces **between columns**.
- The default gutter width is **1.5rem (24px total)**.
- Bootstrap provides `g-*` classes to **control gutter spacing**.

Example: Adjusting Gutters

```
<div class="container">
  <div class="row g-3">
    <div class="col-md-6 bg-danger text-white">Column with gutters</div>
    <div class="col-md-6 bg-info text-white">Column with gutters</div>
  </div>
</div>
```

Explanation:

- `g-3` increases spacing between columns.
- Using `g-0` removes gutters completely.

2.6 Nesting Grids and Alignment

- **Nesting grids** allow placing **rows inside columns**.
- `align-items-*` controls **vertical alignment**.
- `justify-content-*` controls **horizontal alignment**.

Example: Nested Grids

```
<div class="container">
  <div class="row">
    <div class="col-md-6 bg-warning">
```

```
      Parent Column
      <div class="row">
        <div class="col-6 bg-primary text-white">Nested Column</div>
        <div class="col-6 bg-success text-white">Nested Column</div>
      </div>
    </div>
    <div class="col-md-6 bg-secondary text-white">Parent Column</div>
  </div>
</div>
```

Explanation:

- A row is placed **inside another column** to create a **nested grid**.

Example: Column Alignment

```
<div class="container">
  <div class="row align-items-center justify-content-center" style="height: 300px;">
    <div class="col-4 bg-info text-white">Centered Column</div>
  </div>
</div>
```

Explanation:

- `align-items-center` aligns **vertically** in the row.
- `justify-content-center` aligns **horizontally** in the row.

2.7 Flexbox Utilities

- Bootstrap's **grid system is based on Flexbox**.
- **Flex utilities help in alignment and spacing**.

Example: Using Flexbox

```
<div class="d-flex justify-content-between bg-light p-3">
  <div class="p-2 bg-primary text-white">Flex Item 1</div>
  <div class="p-2 bg-secondary text-white">Flex Item 2</div>
  <div class="p-2 bg-success text-white">Flex Item 3</div>
</div>
```

Explanation:

- `d-flex` makes the container a **flexbox**.
- `justify-content-between` **spreads items evenly**.

Summary of Key Concepts

1. **Bootstrap uses a 12-column grid system** that adjusts for different screen sizes.
2. **Containers** control the width (`.container`, `.container-fluid`).
3. **Grid classes** (`.col-sm-*`, `.col-md-*`, etc.) define responsive layouts.
4. **Breakpoints** ensure layouts adjust based on screen size.
5. **Gutters** (`g-*`) manage spacing between columns.
6. **Nesting grids** allows creating complex layouts.
7. **Flexbox utilities** (`d-flex`, `justify-content-*`, etc.) make layouts more flexible.

Bootstrap 5 Typography, Colors, and Icons – Detailed Notes

3.1 Default Typography (h1 to h6, .lead, .display-*)

Bootstrap provides **built-in typography styles** to enhance text appearance.

Headings (h1 to h6)

Bootstrap supports **six levels of headings** with predefined font sizes.

Tag	Default Size	Example
h1	2.5rem (40px)	`<h1>Heading 1</h1>`
h2	2rem (32px)	`<h2>Heading 2</h2>`
h3	1.75rem (28px)	`<h3>Heading 3</h3>`
h4	1.5rem (24px)	`<h4>Heading 4</h4>`
h5	1.25rem (20px)	`<h5>Heading 5</h5>`
h6	1rem (16px)	`<h6>Heading 6</h6>`

Example: Headings in Bootstrap

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

Lead Text (.lead)

- The `.lead` class is used to make paragraphs stand out.
- It increases font size and adjusts line spacing.

Example: Lead Text

```
<p class="lead">This is an important paragraph with a larger font and better readability.</p>
```

Display Headings (.display-*)

- `.display-*` classes create **larger, more impactful headings**.
- Available in sizes `display-1` to `display-6`.

Example: Display Headings

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

3.2 Text Alignment, Text Wrapping, and Text Colors

Text Alignment Classes

- `text-start` → Aligns text to the left (default).
- `text-center` → Centers the text.
- `text-end` → Aligns text to the right.

Example: Text Alignment

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

Text Wrapping and Overflow

- **`text-wrap`** → Allows text to wrap within its container.
- **`text-nowrap`** → Prevents text from wrapping.
- **`text-truncate`** → Cuts off long text with an ellipsis (...).

Example: Text Wrapping and Truncation

```
<div class="text-nowrap">This text will not wrap.</div>
<div class="text-truncate" style="width: 200px; overflow: hidden;">This text is too long and will be
truncated.</div>
```

Text Colors

Bootstrap provides **text color utility classes**.

Class	Color
`text-primary`	Blue
`text-secondary`	Gray
`text-success`	Green
`text-danger`	Red
`text-warning`	Yellow
`text-info`	Cyan

Class	Color
`text-light`	White (on dark backgrounds)
`text-dark`	Black (on light backgrounds)

Example: Text Colors

```
<p class="text-primary">This is a primary colored text.</p>
<p class="text-danger">This is a danger (red) colored text.</p>
<p class="text-success">This is a success (green) colored text.</p>
```

3.3 Background Colors and Utility Classes

Bootstrap provides **background color classes** that can be applied to any element.

Background Color Classes

Class	Color
`bg-primary`	Blue
`bg-secondary`	Gray
`bg-success`	Green
`bg-danger`	Red
`bg-warning`	Yellow
`bg-info`	Cyan

Class	Color
`bg-light`	Light Gray
`bg-dark`	Black

Example: Background Colors

```
<div class="bg-primary text-white p-3">Primary Background</div>
<div class="bg-danger text-white p-3">Danger Background</div>
<div class="bg-success text-white p-3">Success Background</div>
```

Background Gradient

- Bootstrap provides gradient backgrounds using `bg-gradient`.

Example: Background Gradient

```
<div class="bg-primary bg-gradient text-white p-3">Primary Gradient Background</div>
<div class="bg-success bg-gradient text-white p-3">Success Gradient Background</div>
```

Opacity Classes

- Bootstrap allows setting background transparency using `opacity-*`.

Class	Opacity Level
`opacity-100`	Fully visible

Class	Opacity Level
opacity-75	75% visible
opacity-50	50% visible
opacity-25	25% visible
opacity-0	Fully transparent

Example: Opacity

```
<div class="bg-dark text-white opacity-50 p-3">50% Opacity</div>
<div class="bg-dark text-white opacity-25 p-3">25% Opacity</div>
```

3.4 Bootstrap Icons

Bootstrap provides a **built-in icon library** with over **1,500+ icons**.

- Icons are **SVG-based** and require adding the bi-* class.
- The Bootstrap Icons CDN or local installation is needed.

Adding Bootstrap Icons CDN

Add the following in the `<head>` of your HTML file:

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons/font/bootstrap-icons.css">
```

Using Icons in HTML

Bootstrap icons are added using `<i>` or `<span>` elements.

Example: Using Bootstrap Icons

```
<i class="bi bi-alarm"></i> Alarm Icon
<i class="bi bi-bell"></i> Bell Icon
<i class="bi bi-check-circle text-success"></i> Check Circle
```

Changing Icon Size

- `fs-*` classes (`fs-1` to `fs-6`) adjust icon size.

Example: Large Icons

```
<i class="bi bi-heart fs-1 text-danger"></i> Large Heart Icon
<i class="bi bi-star fs-2 text-warning"></i> Medium Star Icon
```

Using Icons in Buttons

Bootstrap icons can be used inside buttons.

Example: Button with Icons

```
<button class="btn btn-primary">
  <i class="bi bi-download"></i> Download
</button>
```

Summary of Key Concepts

1. **Typography** includes headings (`h1-h6`), `.lead` for important text, and `.display-*` for larger headings.

2. **Text alignment** (`text-start`, `text-center`, `text-end`) controls text positioning.
3. **Text wrapping and colors** allow truncation (`text-truncate`) and predefined color classes.
4. **Background colors** (`bg-*`) and gradients (`bg-gradient`) enhance UI design.
5. **Bootstrap Icons** provide a rich set of vector icons for UI components.

Bootstrap 5 Components – Detailed Notes

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

Buttons in Bootstrap are styled using the `.btn` class with contextual color classes.

Basic Buttons

Class	Button Color
`.btn-primary`	Blue
`.btn-secondary`	Gray
`.btn-success`	Green
`.btn-danger`	Red
`.btn-warning`	Yellow

Class	Button Color
`.btn-info`	Cyan
`.btn-light`	Light Gray
`.btn-dark`	Black

Example: Basic Buttons

```
<button class="btn btn-primary">Primary Button</button>
<button class="btn btn-danger">Danger Button</button>
<button class="btn btn-success">Success Button</button>
```

Outline Buttons

Outline buttons have only borders and text color without a solid background.

Example: Outline Buttons

```
<button class="btn btn-outline-primary">Primary Outline</button>
<button class="btn btn-outline-danger">Danger Outline</button>
<button class="btn btn-outline-success">Success Outline</button>
```

Button Groups

Button groups are used to group multiple buttons together.

Example: Button Group

```
<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>
```

4.2 Alerts and Notifications

Bootstrap provides **alert components** for notifications.

Example: Basic Alerts

```
<div class="alert alert-success">This is a success alert!</div>
<div class="alert alert-danger">This is a danger alert!</div>
<div class="alert alert-warning">This is a warning alert!</div>
```

Dismissing Alerts

Alerts can be dismissed using JavaScript.

```
<div class="alert alert-info alert-dismissible fade show">
  This is a dismissible alert!
  <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
```

4.3 Badges and Labels

Badges are used to display notifications or highlight text.

Example: Basic Badges

```
<h1>Messages <span class="badge bg-danger">4</span></h1>
```

```
<button class="btn btn-primary">Notifications <span class="badge bg-light text-dark">10</span></button>
```

4.4 Cards (Layouts, Image Overlays, Groups)

Cards provide a flexible and extensible content container.

Example: Basic Card

```
<div class="card" style="width: 18rem;">
  <img src="image.jpg" class="card-img-top" alt="Image">
  <div class="card-body">
    <h5 class="card-title">Card Title</h5>
    <p class="card-text">This is a simple card example.</p>
    <a href="#" class="btn btn-primary">Read More</a>
  </div>
</div>
```

4.5 Modals (Usage, Customization)

Modals are used for popups and dialogs.

Example: Basic Modal

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

<div class="modal fade" id="myModal">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Modal Title</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
      </div>
      <div class="modal-body">
        Modal body content goes here.
      </div>
    </div>
  </div>
</div>
```

4.6 Navs and Tabs (Navigation, Pills, Dropdowns)

Bootstrap provides navigation components like **tabs, pills, and dropdowns**.

Example: Nav Tabs

```
<ul class="nav nav-tabs">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active Tab</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Another Tab</a>
  </li>
</ul>
```

4.7 Navbar (Fixed, Sticky, Responsive)

The navbar is used for site navigation.

Example: Responsive Navbar

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <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>
    </ul>
  </div>
</nav>
```

4.8 Pagination (Basic, Advanced)

Pagination is used for dividing content into multiple pages.

Example: Basic Pagination

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

4.9 Progress Bars and Spinners

Progress bars indicate loading status.

Example: Progress Bar

```
<div class="progress">
  <div class="progress-bar bg-success" style="width: 75%;">75%</div>
</div>
```

Example: Spinners

```
<div class="spinner-border text-primary"></div>
<div class="spinner-grow text-danger"></div>
```

4.10 Tooltips and Popovers

Tooltips show extra information when hovering over elements.

Example: Tooltip

```
<button class="btn btn-secondary" data-bs-toggle="tooltip" title="Tooltip text">Hover Me</button>
```

Popovers show additional content on click.

Example: Popover

```
<button class="btn btn-warning" data-bs-toggle="popover" title="Popover Title" data-bs-content="Popover body
content">Click Me</button>
```

4.11 Toasts (Notifications)

Toasts provide lightweight notifications.

Example: Toast Notification

```
<div class="toast show">
  <div class="toast-header">
    <strong class="me-auto">Bootstrap</strong>
    <button type="button" class="btn-close" data-bs-dismiss="toast"></button>
  </div>
  <div class="toast-body">
    This is a toast notification!
  </div>
</div>
```

Summary of Key Concepts

1. **Buttons** support various styles including outline and grouped buttons.
2. **Alerts** provide messages that can be dismissed.
3. **Badges** highlight notifications or labels.
4. **Cards** offer flexible content layouts with images.
5. **Modals** create pop-up windows for user interaction.
6. **Navs and Tabs** manage different sections in a structured way.
7. **Navbar** is useful for site navigation and supports responsiveness.
8. **Pagination** manages multi-page content.
9. **Progress Bars and Spinners** indicate loading or completion.
10. **Tooltips and Popovers** provide extra information on hover or click.
11. **Toasts** serve as lightweight notifications.

Bootstrap 5 Forms

5.1 Form Controls and Input Fields Bootstrap provides various form controls such as text fields, password inputs, and textarea fields.

Example:

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

5.2 Select Dropdowns, Checkboxes, and Radio Buttons Dropdowns allow users to select an option, while checkboxes and radio buttons are used for multiple selections and single selections respectively.

Example:

```
<select class="form-select">
  <option selected>Select an option</option>
  <option value="1">Option 1</option>
  <option value="2">Option 2</option>
</select>
```

5.3 Floating Labels and Input Groups Floating labels enhance the usability of forms by placing the label inside the input field.

Example:

```
<div class="form-floating">
  <input type="email" class="form-control" id="email" placeholder="name@example.com">
  <label for="email">Email address</label>
</div>
```

5.4 Form Validation and Feedback Messages Bootstrap provides built-in validation styles for form inputs.

Example:

```
<input type="text" class="form-control is-invalid" placeholder="Enter name">
<div class="invalid-feedback">Please enter a valid name.</div>
```

5.5 Form Layouts (Inline Forms, Grid Forms) Forms can be structured in inline or grid-based layouts.

Example:

```
<form class="row g-3">
  <div class="col-auto">
    <input type="text" class="form-control" placeholder="Enter text">
  </div>
  <div class="col-auto">
    <button type="submit" class="btn btn-primary">Submit</button>
  </div>
</form>
```

5.6 Custom Forms (Range, Switches, File Uploads) Bootstrap supports custom input types such as range sliders, switches, and file uploads.

Example:

```
<input type="range" class="form-range">
<div class="form-check form-switch">
  <input class="form-check-input" type="checkbox">
  <label class="form-check-label">Toggle Switch</label>
</div>
```