LAB06 – Bootstrap Essentials

Exercise 1: Setting Up Bootstrap

Objective: Learn how to include Bootstrap in a project using a CDN.

Instructions:

1. Create a new file named index.html.
2. Add the basic HTML5 boilerplate structure (<!DOCTYPE html>, <html>, <head>, <body>).
3. In the <head> section, include Bootstrap 5 CSS by adding the following link:

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

4. In the <body> section, just before the closing </body> tag, include Bootstrap 5 JavaScript (including Popper.js) by adding:

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

5. Create a simple heading inside the <body> using Bootstrap's typography class: <h1 class="text-center">Welcome to Bootstrap</h1>.
6. Save and open the file in a browser to verify the heading is centered.

Expected Outcome: A centered heading styled with Bootstrap's default typography.

Exercise 2: Creating a Responsive Container

Objective: Understand Bootstrap's container classes.

Instructions:

1. Create a new index.html file and include the Bootstrap 5 CSS and JS CDN links as in Exercise 1.
2. Inside the <body>, add a <div> with the class container:

```
<div class="container">
    <h2>My Content</h2>
    <p>This is inside a container.</p>
```

```
</div>
```

3. Save and view in a browser.
4. Modify the <div> to use the container-fluid class instead of container.
5. Save and refresh the browser to observe the difference in width.
6. Experiment with responsive container classes like container-sm, container-md, etc., and resize the browser window to see the changes.

Expected Outcome: A container that adjusts its width based on the class used and screen size.

Exercise 3: Building a Responsive Grid

Objective: Learn Bootstrap's grid system.

Instructions:

1. Create a new index.html file with Bootstrap 5 CDN links.
2. Inside the <body>, add a container div.
3. Inside the container, create a row with three equal-width columns:

```
<div class="container">
    <div class="row">
        <div class="col">
            <h3>Column 1</h3>
            <p>Content here</p>
        </div>
        <div class="col">
            <h3>Column 2</h3>
            <p>Content here</p>
        </div>
        <div class="col">
            <h3>Column 3</h3>
            <p>Content here</p>
            </div>

        </div>

</div>
```

4. Save and view in a browser.
5. Modify the column classes to col-6 for the first two columns and col-12 for the third. Observe how the layout changes.
6. Test responsiveness by resizing the browser window.

Expected Outcome: A responsive grid layout that adjusts column widths based on screen size.

Exercise 4: Styling Buttons

Objective: Explore Bootstrap's button classes.

Instructions:

1. Create a new index.html file with Bootstrap 5 CDN links.
2. Inside the <body>, add a container div.
3. Inside the container, add buttons with different styles:

```
<div class="container">

    <button type="button" class="btn btn-
primary">Primary</button>

    <button type="button" class="btn btn-
secondary">Secondary</button>

    <button type="button" class="btn btn-
success">Success</button>

    <button type="button" class="btn btn-
danger">Danger</button>

</div>
```

4. Save and view in a browser.
5. Add button sizes by modifying one button to btn-lg and another to btn-sm.
6. Add an outline button by changing one button's class to btn-outline-primary.

7. Save and refresh to observe the changes.

Expected Outcome: A set of styled buttons with different colors, sizes, and outline variations.

Exercise 5: Creating a Navigation Bar

Objective: Build a responsive navigation bar.

Instructions:

1. Create a new index.html file with Bootstrap 5 CDN links.
2. Inside the <body>, add a navbar component:

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container-fluid">
        <a class="navbar-brand" href="#">MySite</a>
        <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <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="#">Contact</a>
                </li>

            </ul>

        </div>

    </div>

</nav>
```

3. Save and view in a browser.
4. Change the navbar background to bg-dark and text to navbar-dark.
5. Resize the browser to test the responsive toggle button.

Expected Outcome: A responsive navbar that collapses into a hamburger menu on smaller screens.

Exercise 6: Adding a Card Component

Objective: Create a card with an image, title, and text.

Instructions:

1. Create a new index.html file with Bootstrap 5 CDN links.
2. Inside the <body>, add a container div.
3. Inside the container, add a Bootstrap card:

```
<div class="container">

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

        <img src="https://via.placeholder.com/150"
class="card-img-top" alt="Placeholder">

        <div class="card-body">

            <h5 class="card-title">Card Title</h5>

            <p class="card-text">Some quick example text to
describe the card.</p>

            <a href="#" class="btn btn-primary">Go
somewhere</a>

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

4. Save and view in a browser.
5. Add a second card next to the first by wrapping both in a row and using col classes.
6. Adjust the card width using style="width: 20rem;" and observe the layout.

Expected Outcome: A card (or cards) with an image, title, text, and a button, arranged in a responsive layout.

Exercise 7: Building a Form

Objective: Create a styled form with Bootstrap.

Instructions:

1. Create a new index.html file with Bootstrap 5 CDN links.
2. Inside the <body>, add a container div.
3. Inside the container, add a form:

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

    </form>

</div>
```

4. Save and view in a browser.
5. Add a textarea input with the class form-control inside another mb-3 div.
6. Save and refresh to verify the form layout.

Expected Outcome: A styled form with text inputs, a textarea, and a submit button.

Exercise 8: Using Alerts

Objective: Display different types of alerts.

Instructions:

1. Create a new index.html file with Bootstrap 5 CDN links.
2. Inside the <body>, add a container div.
3. Inside the container, add alerts with different contextual classes:

```
<div class="container">

    <div class="alert alert-success" role="alert">

        Success! Your action was completed.

    </div>

    <div class="alert alert-warning" role="alert">

        Warning! Something might be wrong.

    </div>

    <div class="alert alert-danger" role="alert">

        Error! Action failed.

    </div>

</div>
```

4. Save and view in a browser.

5. Add a dismissible alert by including a close button:

```
<div class="alert alert-info alert-dismissible fade show"
role="alert">

    Info! This alert can be closed.

    <button type="button" class="btn-close" data-bs-
dismiss="alert" aria-label="Close"></button>

</div>
```

6. Save and test the dismissible alert by clicking the close button.

Expected Outcome: A set of styled alerts, including one that can be dismissed.

Exercise 9: Creating a Modal

Objective: Build a modal dialog.

Instructions:

1. Create a new index.html file with Bootstrap 5 CDN links.
2. Inside the <body>, add a container div.
3. Inside the container, add a button to trigger a modal:

```
<div class="container">

    <button type="button" class="btn btn-primary" data-bs-
toggle="modal" data-bs-target="#myModal">

        Open Modal

    </button>

</div>
```

4. Below the button, add the modal structure:

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

    <div class="modal-dialog">

        <div class="modal-content">

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

```
                <h5 class="modal-title"
id="myModalLabel">Modal Title</h5>
                <button type="button" class="btn-close" data-
bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                This is the modal content.
            </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>
```

5. Save and view in a browser. Click the button to open the modal.
6. Modify the modal size by adding modal-lg to the modal-dialog class.

Expected Outcome: A functional modal that appears when the button is clicked and can be closed.

Exercise 10: Implementing a Carousel

Objective: Create a sliding carousel.

Instructions:

1. Create a new index.html file with Bootstrap 5 CDN links.
2. Inside the <body>, add a container div.
3. Inside the container, add a carousel:

```
<div class="container">

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

        <div class="carousel-inner">

            <div class="carousel-item active">

                <img
src="https://via.placeholder.com/800x400?text=Slide+1"
class="d-block w-100" alt="Slide 1">

            </div>

            <div class="carousel-item">

                <img
src="https://via.placeholder.com/800x400?text=Slide+2"
class="d-block w-100" alt="Slide 2">

            </div>

            <div class="carousel-item">

                <img
src="https://via.placeholder.com/800x400?text=Slide+3"
class="d-block w-100" alt="Slide 3">

            </div>

        </div>

        <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>

            <span class="visually-hidden">Previous</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>

            <span class="visually-hidden">Next</span>
            </button>

        </div>

</div>
```

4. Save and view in a browser. Test the carousel navigation.
5. Add carousel indicators by including:

```
<div class="carousel-indicators">

    <button type="button" data-bs-target="#myCarousel" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>

    <button type="button" data-bs-target="#myCarousel" data-bs-slide-to="1" aria-label="Slide 2"></button>

    <button type="button" data-bs-target="#myCarousel" data-bs-slide-to="2" aria-label="Slide 3"></button>

</div>
```

Place this inside the carousel div, before carousel-inner.

6. Save and test the indicators.

Expected Outcome: A functional carousel with navigation buttons and indicators.

Exercise 11: Using Typography and Text Utilities

Objective: Style text using Bootstrap's typography classes.

Instructions:

1. Create a new index.html file with Bootstrap 5 CDN links.
2. Inside the <body>, add a container div.
3. Inside the container, add various text elements:

```
<div class="container">

    <h1 class="display-1">Display Heading</h1>

    <p class="lead">This is a lead paragraph.</p>

    <p class="text-muted">This text is muted.</p>
```

```
        <p class="text-uppercase">This text is uppercase.</p>
        <p class="fw-bold">This text is bold.</p>
</div>
```

4. Save and view in a browser.
5. Experiment with other text utilities like text-center, text-primary, or fs-1 for font size.
6. Save and refresh to observe the changes.

Expected Outcome: A variety of styled text elements using Bootstrap's typography and utility classes.

Exercise 12: Creating a Responsive Table

Objective: Build a responsive, styled table.

Instructions:

1. Create a new index.html file with Bootstrap 5 CDN links.
2. Inside the <body>, add a container div.
3. Inside the container, add a table:

```
<div class="container">
    <table class="table table-striped table-bordered">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Name</th>
                <th scope="col">Email</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <th scope="row">1</th>
```

```
                    <td>John Doe</td>
                    <td>john@example.com</td>
                </tr>
                <tr>
                    <th scope="row">2</th>
                    <td>Jane Smith</td>
                    <td>jane@example.com</td>
                </tr>
            </tbody>
        </table>
</div>
```

4. Save and view in a browser.
5. Make the table responsive by wrapping it in a div with class table-responsive.
6. Add the table-dark class to the table and observe the style change.

Expected Outcome: A styled, responsive table with striped rows and borders.

Exercise 13: Using Spacing Utilities

Objective: Apply Bootstrap's spacing utilities (margin and padding).

Instructions:

1. Create a new index.html file with Bootstrap 5 CDN links.
2. Inside the <body>, add a container div.
3. Inside the container, add elements with spacing utilities:

```
<div class="container">
    <div class="bg-primary p-3 m-2">Box with padding and
margin</div>
    <div class="bg-success p-4 mx-auto" style="width:
200px;">Centered box</div>
```

```
    <div class="bg-warning mt-5 p-2">Box with top margin</div>
</div>
```

4. Save and view in a browser.
5. Experiment with other spacing classes like ms-3, py-4, or mb-5.
6. Save and refresh to observe the changes.

Expected Outcome: Elements with customized margins and padding using Bootstrap's utility classes.

Exercise 14: Creating a Jumbotron

Objective: Build a jumbotron-style hero section.

Instructions:

1. Create a new index.html file with Bootstrap 5 CDN links.
2. Inside the <body>, add a jumbotron-like section using Bootstrap utilities:

```
<div class="bg-light p-5 rounded-3 text-center">
    <div class="container">
        <h1 class="display-4">Welcome to My Site</h1>
        <p class="lead">This is a simple hero unit, a
jumbotron-style component.</p>
        <a class="btn btn-primary btn-lg" href="#"
role="button">Learn more</a>
    </div>
</div>
```

3. Save and view in a browser.
4. Change the background to bg-primary and text to text-white for contrast.
5. Add margin utilities like my-4 to the outer div and test the layout.

Expected Outcome: A prominent hero section with a heading, text, and a call-to-action button.

Exercise 15: Building a Complete Landing Page

Objective: Combine multiple Bootstrap components to create a landing page.

Instructions:

1. Create a new index.html file with Bootstrap 5 CDN links.
2. Build a landing page with a navbar, hero section, card grid, and footer:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Landing Page</title>

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

</head>

<body>

    <!-- Navbar -->

    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">

        <div class="container-fluid">

            <a class="navbar-brand" href="#">MySite</a>

            <button class="navbar-toggler" type="button" data-
bs-toggle="collapse" data-bs-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">

                <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="#">Services</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="#">Contact</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <!-- Hero Section -->
    <div class="bg-primary text-white text-center py-5">
        <div class="container">
            <h1 class="display-4">Welcome to MySite</h1>
            <p class="lead">Discover our amazing services.</p>
            <a class="btn btn-light btn-lg" href="#"
role="button">Get Started</a>
        </div>
    </div>

    <!-- Card Grid -->
    <div class="container my-5">
        <div class="row">
            <div class="col-md-4">
                <div class="card">
                    <img src="https://via.placeholder.com/150"
class="card-img-top" alt="Service 1">
                    <div class="card-body">
                        <h5 class="card-title">Service 1</h5>
                        <p class="card-text">Description of
service 1.</p>
                        <a href="#" class="btn btn-
primary">Learn More</a>
                    </div>
                </div>
            </div>
            <div class="col-md-4">
                <div class="card">
                    <img src="https://via.placeholder.com/150"
class="card-img-top" alt="Service 2">
                    <div class="card-body">
                        <h5 class="card-title">Service 2</h5>
                        <p class="card-text">Description of
service 2.</p>
                        <a href="#" class="btn btn-
primary">Learn More</a>
                    </div>
                </div>
            </div>
            <div class="col-md-4">
```

```
                <div class="card">
                    <img src="https://via.placeholder.com/150"
class="card-img-top" alt="Service 3">
                    <div class="card-body">
                        <h5 class="card-title">Service 3</h5>
                        <p class="card-text">Description of
service 3.</p>
                        <a href="#" class="btn btn-
primary">Learn More</a>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <!-- Footer -->
    <footer class="bg-dark text-white text-center py-3">
        <div class="container">
            <p>&copy; 2025 MySite. All rights reserved.</p>
        </div>
    </footer>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/boot
strap.bundle.min.js"></script>
</body>
</html>
```

3. Save and view in a browser.
4. Test responsiveness by resizing the browser window.

5. Customize the hero section’s background color or add a form in the footer.

Expected Outcome: A fully responsive landing page with a navbar, hero section, card grid, and footer.

THE END