Hands-On Lab: Implementing Bootstrap Pre-built Components

Skills Network

Estimated time: 1 Hour

Introduction

Bootstrap is a widely used front-end framework that simplifies the development of responsive and visually appealing web applications. By providing a comprehensive set of pre-built components, Bootstrap enables developers to create professional-looking interfaces quickly, ensuring consistency and responsiveness across various devices. In this lab, you will implement various Bootstrap components and observe the differences in output when these components are applied to a webpage. This practical experience will help you understand the purpose and advantages of using Bootstrap pre-built components effectively.

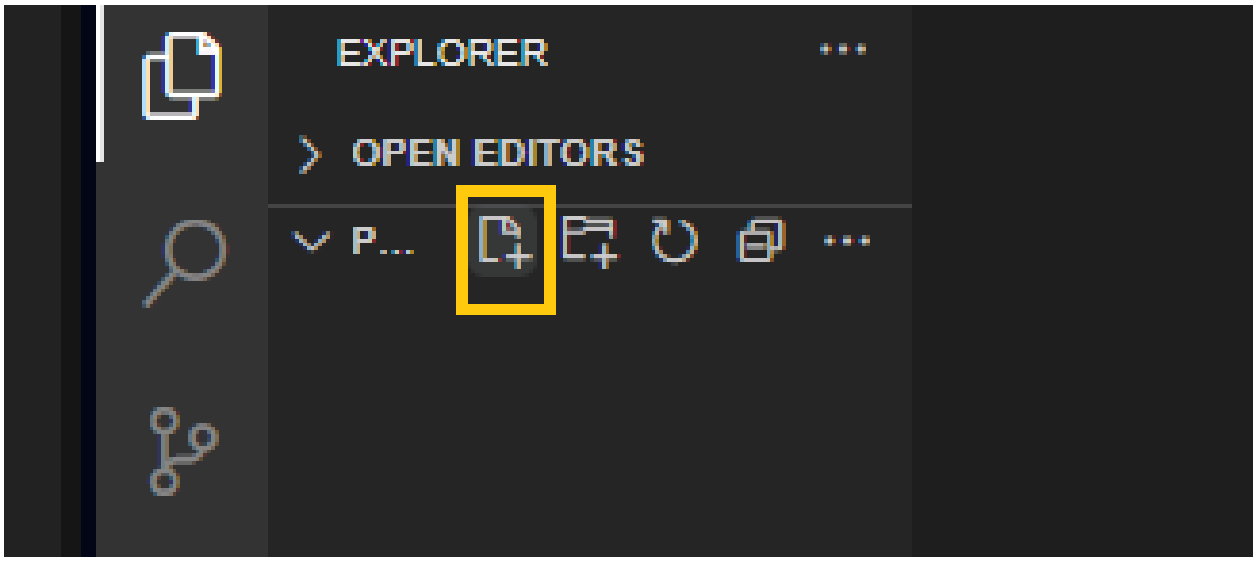
Objective

By the end of this lab, you will:

- Understand the purpose and advantages of using Bootstrap pre-built components.
- Implement various Bootstrap components, including navigation, forms, cards, alerts, modals, tooltips, and carousels.
- Observe the webpage before and after applying Bootstrap pre-built components.

Exercise 1: Setting Up Bootstrap

1. Create a new HTML file named `bootstrap_lab.html`.



2. Add the following basic structure in the html file:

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

</body>
</html>
```

This code creates a basic webpage with a title and a paragraph. Without Bootstrap, the styling is minimal, and the layout lacks responsiveness.

3. Modify the `<head>` section to include Bootstrap's CSS and JavaScript files, add the below code after <title>:

```
<!-- Bootstrap CSS CDN -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
        crossorigin="anonymous"></script>
```

```
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"
        integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM8ODewa9r"
        crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js"
        integrity="sha384-0pUGZvbkm6XF6gxjEnlmuGrJXVbNuzT9qBBavbLwCsOGabYfZo0T0to5eqruptLy"
        crossorigin="anonymous"></script>
```

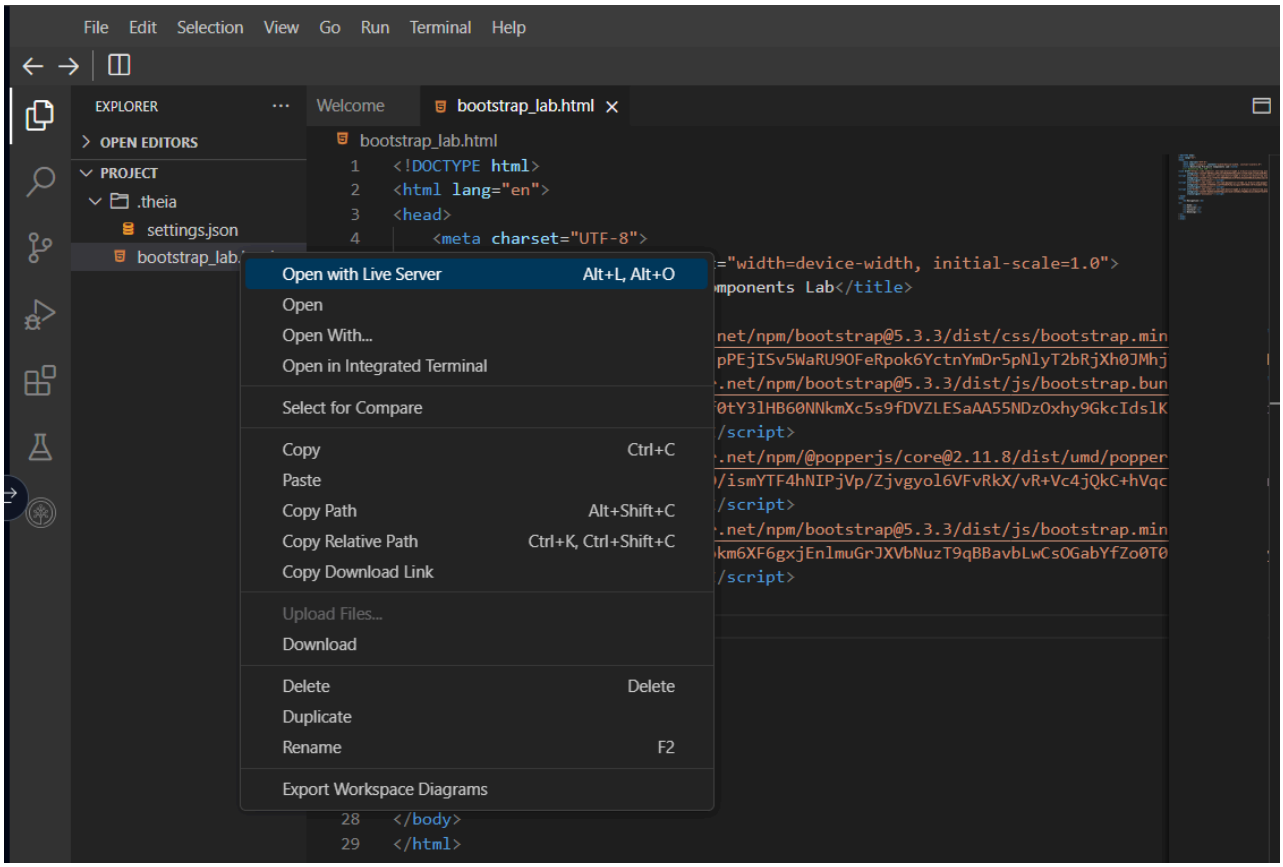
Exercise 2: Creating a Navigation Bar

A navbar (navigation bar) is a responsive and adaptable menu provided by Bootstrap that can be customized to display links to different sections of your website. The navbar will collapse into a toggleable menu on smaller screens, making it mobile-friendly.

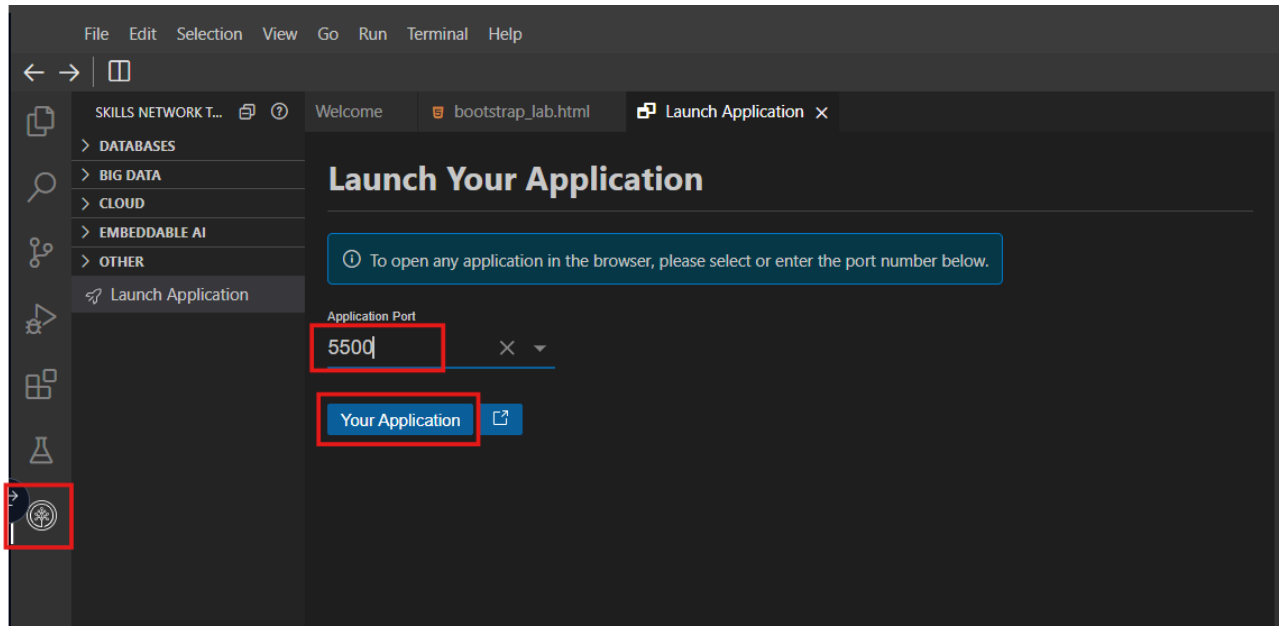
1. Add a basic navigation section to your HTML:

```
<h2>Navigation</h2>
<ul>
    <li>Home</li>
    <li>Services</li>
    <li>Contact</li>
    <li>Bookings</li>
</ul>
```

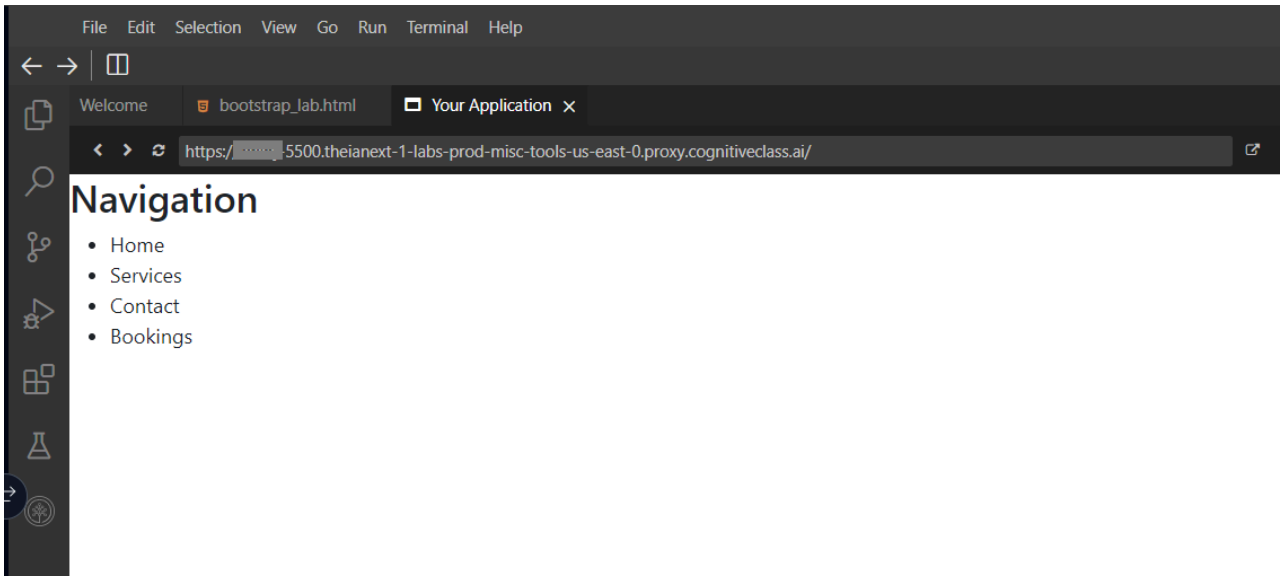
Right-click on the HTML file and choose `Open with Live Server` to view the webpage output.



Click on the `Skills Network Toolbar` icon, enter port 5500, and then click on the `Your Application` button.



Next, select the file `bootstrap_lab.html`. Then you can see the simple list functions as navigation but lacks styling and responsiveness.



2. To create a fully functional and responsive navbar, replace the above navigation code with the following Bootstrap navbar code:

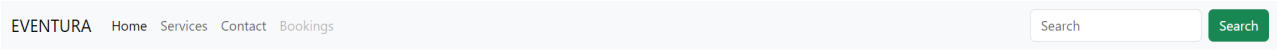
```
<!-- Navbar -->
<nav class="navbar navbar-expand-lg bg-body-tertiary">
    <div class="container-fluid">
        <a class="navbar-brand" href="#">EVENTURA</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
            aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                <li class="nav-item">
                    <a class="nav-link active" aria-current="page" 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>
                <li class="nav-item">
                    <a class="nav-link disabled" aria-disabled="true">Bookings</a>
                </li>
            </ul>
            <form class="d-flex" role="search">
                <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
                <button class="btn btn-outline-success" type="submit">Search</button>
            </form>
        </div>
    </div>
</nav>
```

You can modify the text content according to your preferences.

Here is the explanation of the Bootstrap navbar components used in the provided code:

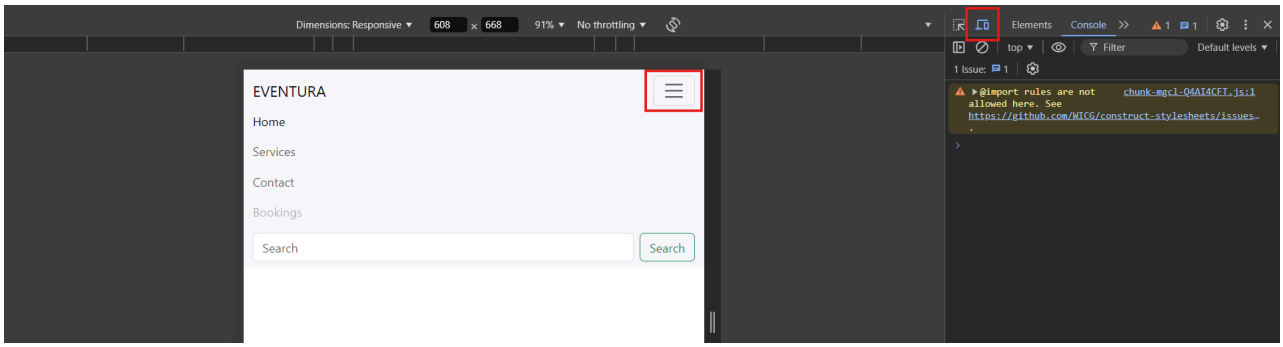
- **`navbar`**: The main class that creates the navbar container, providing basic styling such as spacing and alignment.
- **`navbar-expand-lg`**: Makes the navbar responsive, remaining collapsed on smaller screens and expanding into a horizontal menu at the large breakpoint (≥992px).
- **`bg-body-tertiary`**: Applies a background color to the navbar using Bootstrap's tertiary background scheme.
- **`navbar-toggler`**: The toggle button visible on smaller screens that collapses or expands the navbar content.
- **`navbar-toggler-icon`**: Applies a built-in icon to the toggle button, providing a visual indicator for expanding/collapsing the navbar.
- **`collapse navbar-collapse`**: Collapses the navbar content on smaller screens and expands it on larger screens, working with the `navbar-toggler`.
- **`navbar-brand`**: Applied to the branding element (e.g., logo or website name), ensuring proper spacing and styling.
- **`navbar-nav`**: Styles the navigation items inside the navbar, providing appropriate spacing and layout for the list of links.
- **`nav-item`**: Applied to each item in the navigation list, ensuring correct padding and alignment for individual links or dropdowns.
- **`nav-link`**: Styles each link inside the navbar, providing correct text color, padding, and hover effects.
- **`nav-link active`**: Highlights the current page's link, applying distinct styling to make it visually different.
- **`nav-link disabled`**: Disables the link, making it unclickable and visually distinct with lower opacity.

Output: Refresh your webpage to see the navigation bar styled with Bootstrap. Notice the improved layout, responsiveness, and mobile-friendly toggle functionality.



Responsive navbar output:

- Press Ctrl + Shift + I (Windows/Linux) or Cmd + Option + I (Mac).
- Then, click the device icon (which looks like a small phone/tablet) located in the top-left corner of the Developer Tools panel, as highlighted in the screenshot below.
- Interact with the navigation to observe how the website adjusts to various screen sizes.



Exercise 3: Adding a Carousel

Bootstrap's grid system allows you to create responsive layouts using a 12-column system. The `.container` class ensures your content is centered, while `.row` and `.col-` classes divide the content into columns.

1. Add a simple section for image slides without Bootstrap:

```
<h2>Image Gallery</h2>
<img src="https://via.placeholder.com/150" alt="Image 1">
<img src="https://via.placeholder.com/150" alt="Image 2">
<img src="https://via.placeholder.com/150" alt="Image 3">
```

This layout lacks organization and interactivity.

2. To create a functional and visually appealing image carousel that allows users to cycle through multiple slides easily, replace the above image section code with the following Bootstrap carousel code:

```
<!-- Carousel -->
<div id="carouselExample" class="carousel slide">
    <div class="carousel-inner">
        <div class="carousel-item active">
            <img src="https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/OM5-AjisaP143eE362lvFw/1.png" class="d-bloc
        </div>
        <div class="carousel-item">
            <img src="https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/lpYuxmA4zC9F4aG6RBkyrA/2.png" class="d-bloc
```

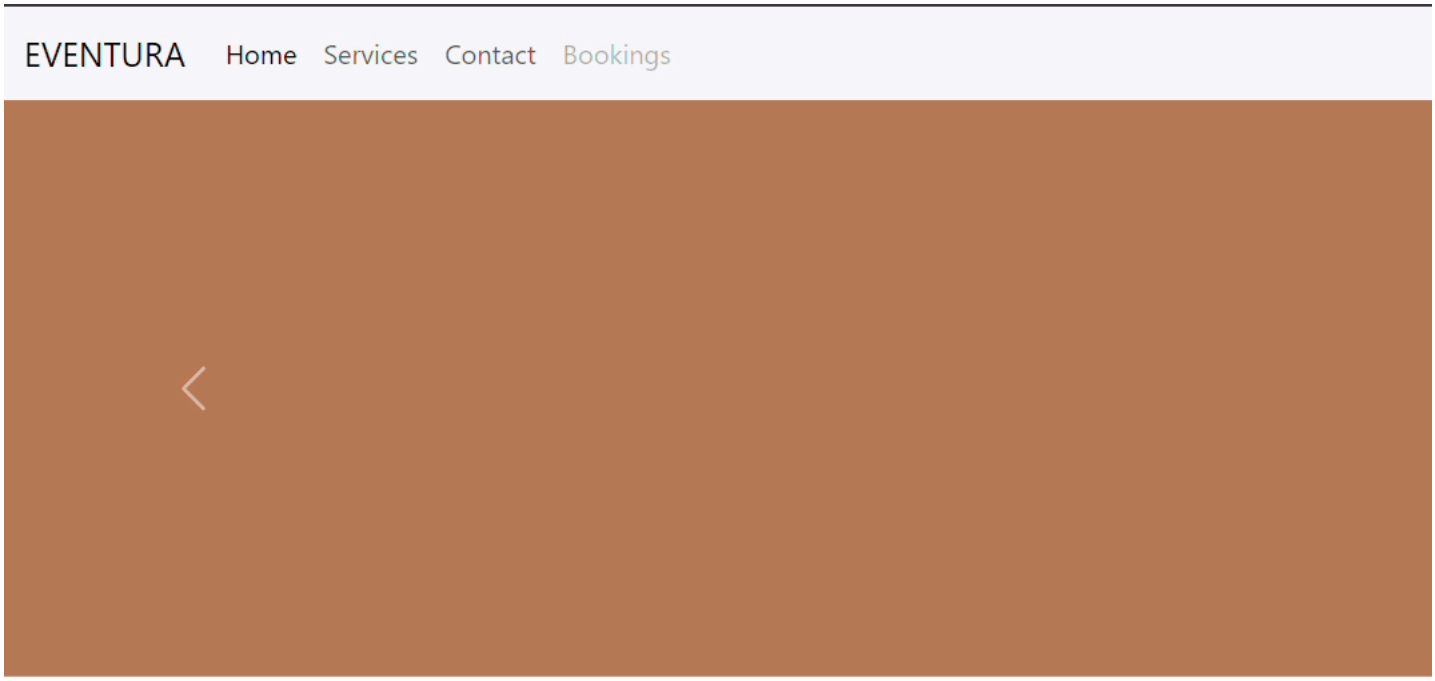
```
            </div>
            <div class="carousel-item">
                <img src="https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/MFCpEcnA6F-RQkrITCoxBA/3.png" class="d-bloc
            </div>
        </div>
        <button class="carousel-control-prev" type="button" data-bs-target="#carouselExample" 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="#carouselExample" data-bs-slide="next">
            <span class="carousel-control-next-icon" aria-hidden="true"></span>
            <span class="visually-hidden">Next</span>
        </button>
    </div>
```

You can modify the text content and images according to your preferences.

Here is the explanation of the Bootstrap carousel components used in the provided code:

- **`carousel`**: The main container for the carousel, establishing the structure necessary for creating a sliding image gallery.
- **`slide`**: Enables the sliding effect when transitioning between carousel items, adding styles for smooth animations.
- **`carousel-inner`**: Contains all the carousel items, wrapping the individual slides and managing their display and transition effects.
- **`carousel-item`**: Applied to each individual item (or slide) within the carousel, defining specific styles and behaviors for each slide. The first item should also have the `active` class.
- **`active`**: Indicates which carousel item is currently visible; only one item should have this class at any time to display as the active slide.
- **`d-block`**: Displays the image as a block-level element, allowing it to take up the full width of its parent container.
- **`w-100`**: Makes the image take up 100% of the width of the parent container, ensuring responsiveness and proper fitting within the carousel.
- **`carousel-control-prev`**: Applied to the previous button in the carousel, allowing users to navigate to the previous slide.
- **`data-bs-target="#carouselExample"`**: Specifies the ID of the carousel to which the button should apply, linking the button to the corresponding carousel.
- **`data-bs-slide="prev"`**: Indicates that the button should navigate to the previous slide when clicked.
- **`carousel-control-next`**: Applied to the next button in the carousel, allowing users to navigate to the next slide.
- **`data-bs-slide="next"`**: Indicates that the button should navigate to the next slide when clicked.
- **`carousel-control-prev-icon`** and **`carousel-control-next-icon`**: Used to apply the icons for the previous and next buttons, providing a visual cue for navigation.
- **`visually-hidden`**: Hides the text labels ("Previous" and "Next") from sighted users while keeping them accessible to screen readers, enhancing accessibility.

Output: Refresh your browser to see the interactive carousel that allows users to navigate between images seamlessly.



Exercise 4: Displaying Cards for Services

Cards are versatile containers that display different types of content like text, images, and links in a structured format. This step will create a card layout for displaying various services.

1. Below is the code to create a basic list of services:

```
<h2>Our Services</h2>
<ul>
    <li>Service 1</li>
    <li>Service 2</li>
    <li>Service 3</li>
</ul>
```

This list lacks organization and visual appeal.

2. To create a structured and visually appealing card layout, suitable for displaying content such as services, products, or features in a concise manner, replace the basic service list code with Bootstrap cards:

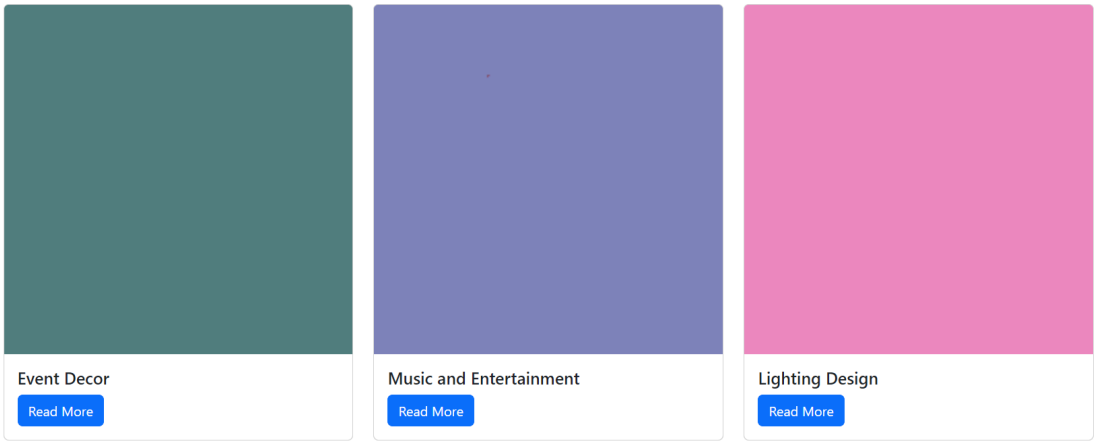
```
<!-- Card Section -->
<div class="container mt-5">
    <div class="row">
        <div class="col-md-4">
            <div class="card">
                <img src="https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/zdabgP316w685n0T_wPqGg/C1.png" class="c
                <div class="card-body">
                    <h5 class="card-title">Event Decor</h5>
                    <a href="#" class="btn btn-primary">Read More</a>
                </div>
            </div>
        </div>
        <div class="col-md-4">
            <div class="card">
                <img src="https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/Yq7tYF1aOJse6YgMs1PeBg/C2.png" class="c
                <div class="card-body">
                    <h5 class="card-title">Music and Entertainment</h5>
                    <a href="#" class="btn btn-primary">Read More</a>
                </div>
            </div>
        </div>
        <div class="col-md-4">
            <div class="card">
                <img src="https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/3vSyMU5LZPiCf7hJtNLuDw/C3.png" class="c
                <div class="card-body">
                    <h5 class="card-title">Lighting Design</h5>
                    <a href="#" class="btn btn-primary">Read More</a>
                </div>
            </div>
        </div>
    </div>
</div>
```

   You can modify the text content images to according to your preferences.

Here is the explanation of the Bootstrap card components used in the provided code:

- **`card`**: This class creates a flexible and extensible content container. It acts as the main wrapper for the card component, providing styling and structure for the card's content.
- **`card-img-top`**: This class is applied to the image within the card. It styles the image to be displayed at the top of the card and ensures it is responsive, adapting to the card's width.
- **`card-body`**: This class wraps the main content of the card, including the title, text, and buttons. It provides appropriate padding and styling to ensure the content is well-organized and visually appealing.
- **`card-title`**: This class styles the title of the card, typically used for headings. It applies specific font styles and sizes to make the title stand out.
- **`card-text`**: This class styles the text content of the card, providing default styling for paragraphs or descriptions, ensuring consistency and readability.

Output: Refresh your browser to see how the card layout organizes service information attractively. The cards are responsive, adapting to different screen sizes.



Exercise 5: Implementing a Modal

Modals are dialog boxes that pop up over the current page content. They are ideal for displaying forms, alerts, or other content that needs user interaction.

1. Add a basic button for opening a modal without Bootstrap functionality:

```
<button>Open Modal</button>
```

The button does not perform any action.

2. To define the structure and functionality of the modal within the Bootstrap framework, replace the button code with the following modal implementation:

```
<!-- Button trigger modal -->
<div class="d-flex justify-content-center" style="margin: 20px">
    <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#exampleModal">
        Contact Us
    </button>
</div>
<!-- Modal -->
<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">
                <h1 class="modal-title fs-5" id="exampleModalLabel">Contact Us</h1>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <form id="contactForm">
                    <div class="mb-3">
                        <label for="recipient-name" class="col-form-label">Recipient:</label>
                        <input type="text" class="form-control" id="recipient-name">
                    </div>
                    <div class="mb-3">
                        <label for="message-text" class="col-form-label">Message:</label>
                        <textarea class="form-control" id="message-text"></textarea>
                    </div>
                </form>
                <div id="alertMessage" class="alert d-none" role="alert"></div>
            </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" id="sendButton">Send</button>
            </div>
        </div>
    </div>
</div>
```

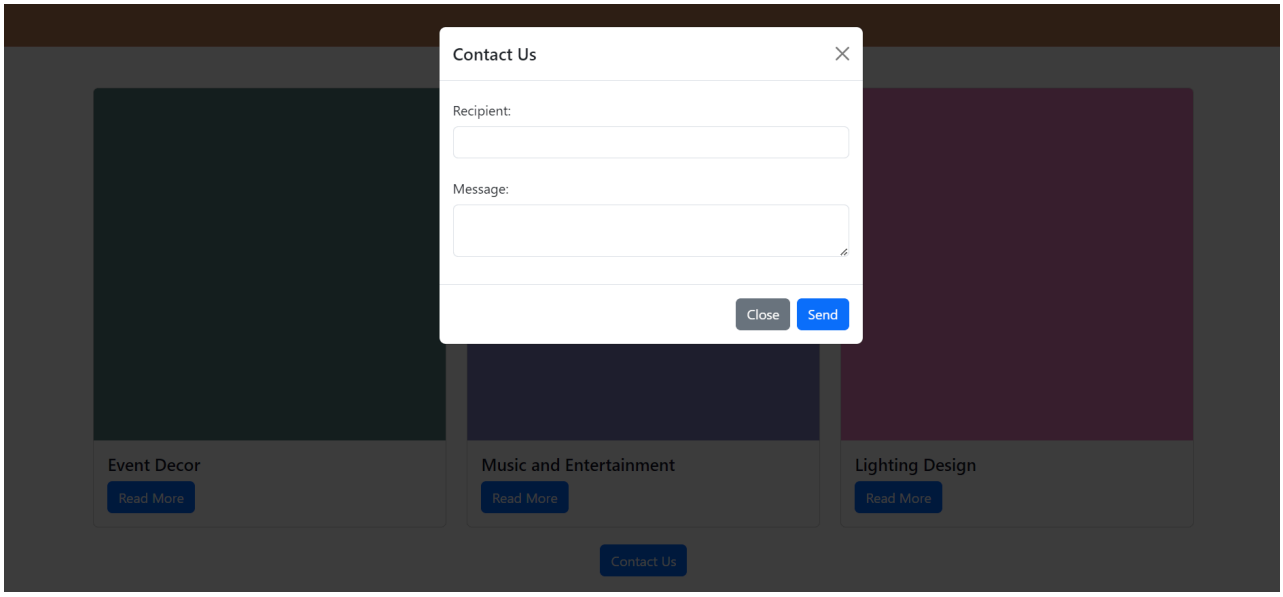
   You can modify the text content according to your preferences.

Here is the explanation of the Bootstrap modal components used in the provided code:

- **`modal`**: The main container for the modal component, creating a dialog box that overlays the main content.
- **`fade`**: Adds a fading effect when the modal is shown or hidden.
- **`id="exampleModal"`**: A unique identifier for the modal, referenced by the button to trigger its display.
- **`tabindex="-1"`**: Allows the modal to be focusable and receive keyboard events, enhancing accessibility.
- **`aria-labelledby="exampleModalLabel"`**: Associates the modal with a label for screen readers.
- **`aria-hidden="true"`**: Indicates that the modal is hidden when not displayed.

- **`modal-dialog`**: A wrapper for the modal's content, providing styling and positioning.
- **`modal-content`**: Wraps the content of the modal, including the header, body, and footer.
- **`modal-header`**: Defines the header section of the modal, typically containing the title and close button.
- **`modal-title`**: Styles the title of the modal.
- **`btn-close`**: Styles the close button in the modal header, allowing users to dismiss the modal.
- **`modal-body`**: Contains the main content of the modal, where forms or text can be placed.
- **`modal-footer`**: Defines the footer section of the modal, typically containing action buttons.

Output: Refresh your browser and click the "Open Modal" button to see the modal in action. The modal overlays the current content, providing a focused view for interactions. The modal example used here is a contact form.



Exercise 6: Displaying Alerts and Toasts

Alerts and Toasts are used to provide feedback messages to users. Alerts are static messages, while Toasts are temporary messages that appear and fade away.

Add the below code after the modal code to display alert messages:

```
<script>
    // Adds click event listener to the send button
    document.getElementById('sendButton').addEventListener('click', function () {
        // Gets the trimmed value of the recipient's name input field
        const recipientName = document.getElementById('recipient-name').value.trim();
        // Gets the trimmed value of the message text input field
        const messageText = document.getElementById('message-text').value.trim();
        // References the alert message element
        const alertMessage = document.getElementById('alertMessage');
        // Checks if both fields are filled
        if (recipientName && messageText) {
            // Sets alert message for success
            alertMessage.className = 'alert alert-success';
            alertMessage.innerText = 'Message sent successfully!';
        } else {
            // Sets alert message for error
            alertMessage.className = 'alert alert-danger';
            alertMessage.innerText = 'Please fill in all fields.';
        }

        // Removes 'd-none' class to show the alert message
        alertMessage.classList.remove('d-none');
    });
</script>
```

You can modify the text content according to your preferences.

This script adds a click event listener to a button with the ID sendButton. When the button is clicked, it retrieves the values from input fields for the recipient's name and message text, trimming any extra whitespace. If both fields are filled, it displays a success message. If either field is empty, it shows an error message prompting the user to fill in all fields. The alert message visibility is managed by removing a class that hides it.

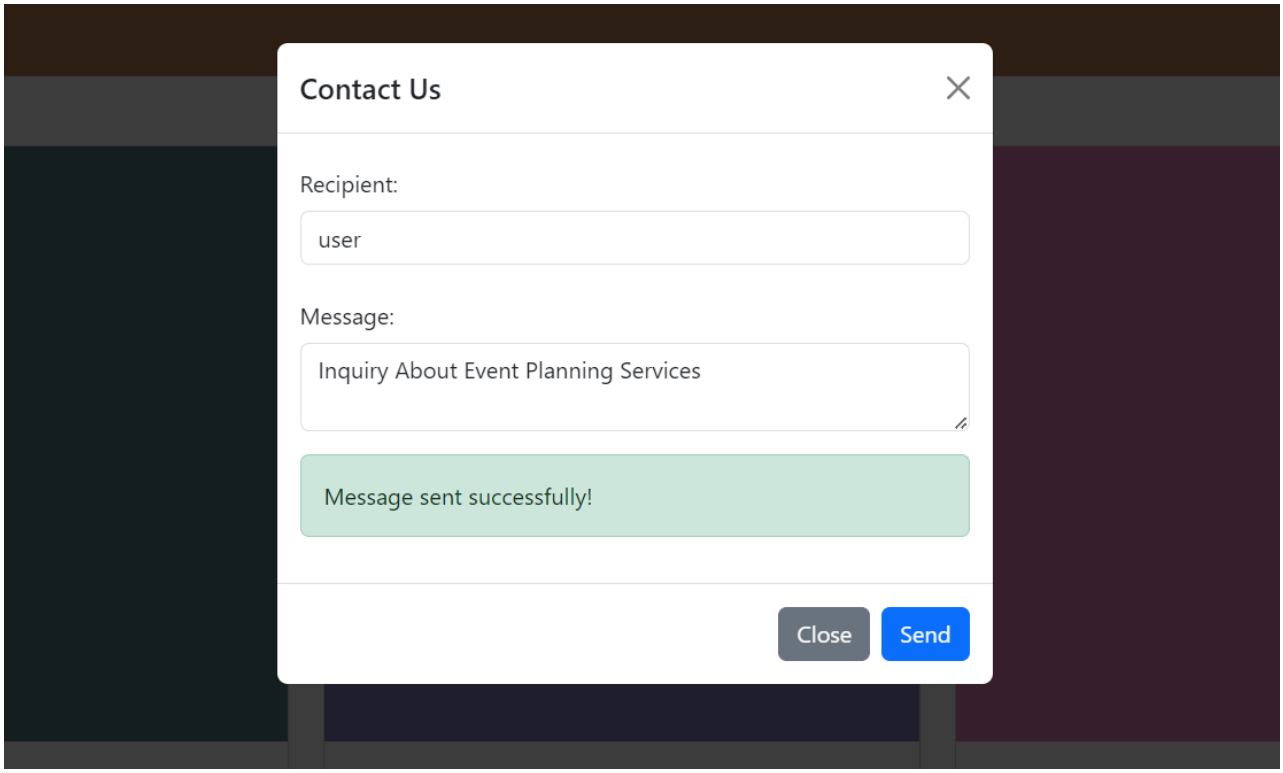
Here is the explanation of the Bootstrap alert components used in the provided code:

- **`alert`**: This class is the base class for creating alert messages. It provides the necessary styling and padding to distinguish the alert from other content on the page.

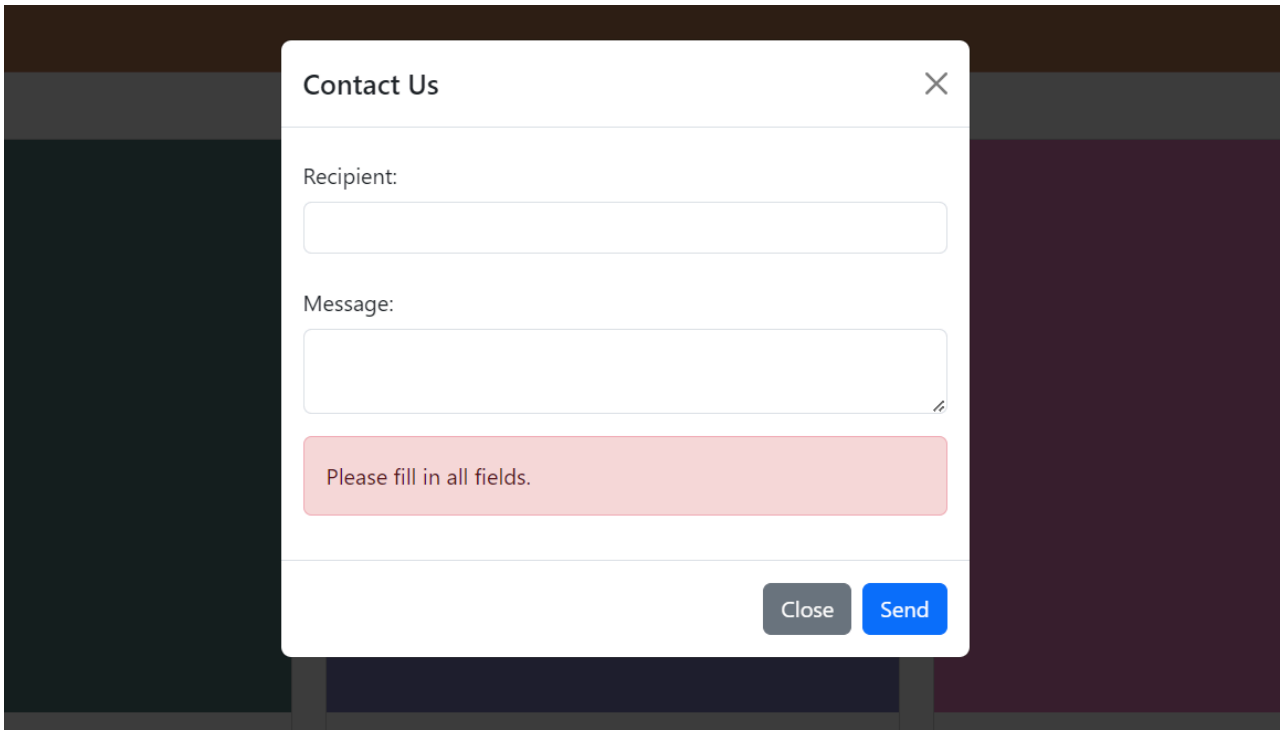
- **`alert-success`**: This class applies a success style to the alert, typically indicating that an operation was completed successfully. It usually includes a green background color and contrasting text to signal a positive message.
- **`alert-warning`**: This class applies a warning style to the alert, indicating that there is caution needed. It typically features a yellow or orange background, signaling the user to pay attention to the message.
- **`alert-danger`**: This class applies a danger style to the alert, indicating that there is an error or critical issue that needs immediate attention. It typically features a red background color and contrasting text to signal a negative message, alerting the user to potential problems or the need for corrective action.
- **`role="alert"`**: This attribute enhances accessibility by indicating to assistive technologies that the content within the div is an alert. It helps users understand that the message is important and may require their attention.

These components together create visually distinct alert messages that communicate important information or feedback to users, allowing them to quickly identify the status of actions or notifications.

Output: Refresh your browser to see the alerts styled with Bootstrap, enhancing their visibility and clarity.
Check the alerts: a success alert appears if both fields are filled.

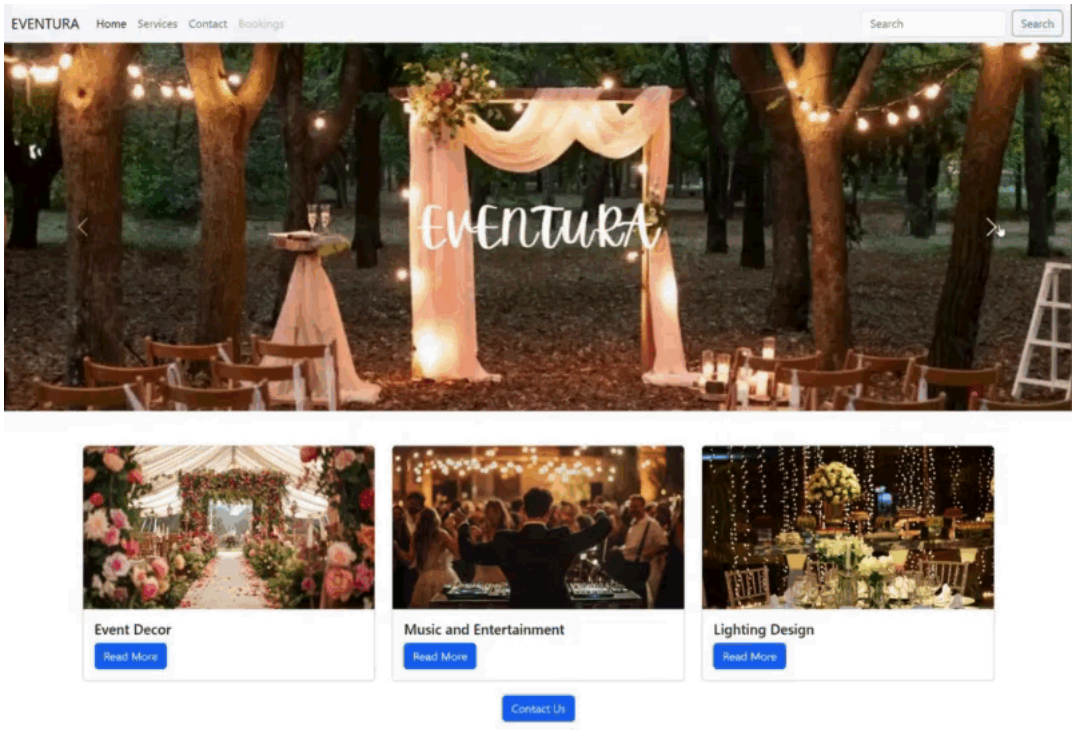


And an error alert prompts if either of the fields are not filled.



▶ Click here to view the complete code:

Here is an sample! of the completed webpage after adding images:



> *DISCLAIMER: It's important to understand that due to copyright constraints, we cannot provide the original images used in this practice project, as seen in the provided screenshots. However, we have provided the image source below for your reference. Alternatively, you are welcome to download and use images of your preference from the internet. Please be aware that the appearance of the website may vary depending on the images you choose to use. Additionally, you can also generate your images using the free AI tool provided below.*

Source: ImageSource

AI Tool: Free AI Image Editor

Conclusion

In this lab, you successfully implemented various Bootstrap pre-built components, including navigation bars, cards, forms, alerts, modals, and carousels. You observed how these components transformed a basic webpage into a responsive, visually appealing interface. Understanding

how to utilize Bootstrap's pre-built components is essential for developing efficient and consistent web applications. Continue to experiment with these components and explore further customization to enhance your web development skills.

Note

This lab includes Bootstrap 5 CSS and JavaScript via the CDN provided by jsDelivr.

Important: If a new version of Bootstrap is released, it may introduce breaking changes or new features. Ensure to review the Bootstrap documentation for updates, as your project may require adjustments to maintain compatibility and take advantage of new functionalities.

Congratulations!

You've successfully implemented various pre-built Bootstrap components, creating a responsive and functional webpage.

Author

Rajashree Patil

© IBM Corporation. All rights reserved.