Hands-on Lab: Responsive Web Design using Tailwind & Bootstrap Frameworks

Estimated Time: 30 minutes

Introduction:

Responsive web design ensures that websites adapt seamlessly to various screen sizes, offering optimal usability across devices such as mobile phones, tablets, and desktops. In this lab, you will implement responsiveness using **Tailwind CSS** and **Bootstrap** frameworks.

Objective:

By the end of this lab, learners will be able to:

1. Create a responsive webpage that adjusts for mobile, tablet, and desktop screens
2. Use **Tailwind CSS's responsive modifiers** to enhance responsiveness
3. Use **Bootstrap's grid system and navbar components** to add responsiveness and functionality
4. Understand the differences between Tailwind CSS and Bootstrap

---

Exercise 1: Understanding starter code

- You will be provided with a basic webpage layout consisting of the following sections:
    - Header
    - Navbar
    - Main Section
    - Aside
    - Article
    - Footer
- Each section will have distinct background colors to visualize the layout structure.
- Create a new HTML file and insert the following code into it:

Starter code

```
<!DOCTYPE html>
<html lang="en">
    <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Basic Layout</title>
    <style>
        body {
        margin: 0;
        font-family: Arial, sans-serif;
        }
        header, footer {
        background-color: red;
        text-align: center;
        padding: 10px;
        }
        nav {
        background-color: blue;
        padding: 10px;
        text-align: center;
        }
        section {
        background-color: white;
        padding: 10px;
        }
        aside {
        background-color: yellow;
        padding: 10px;
        }
        article {
        background-color: lightgreen;
        padding: 10px;
        }
        .container {
        display: flex;
        flex-direction: column;
        }
        .row {
        display: flex;
        }
    </style>
    </head>
    <body>
    <header>Header</header>
    <nav>Navigation</nav>
    <div class="container">
        <div class="row">
        <aside>Aside</aside>
        <section>Main Section</section>
        <article>Article</article>
        </div>
    </div>
    <footer>Footer</footer>
    </body>
</html>
```

Description:
This code creates a simple webpage layout with distinct background colors for visualizing the structure. The code is not fully responsive as it lacks media queries for adapting the layout to different screen sizes. The `.container` uses flexbox to stack elements vertically, but there's no flexibility in the layout to adjust for smaller screens. Additionally, the fixed padding and absence of width adjustments for the elements can cause overflow on smaller devices.

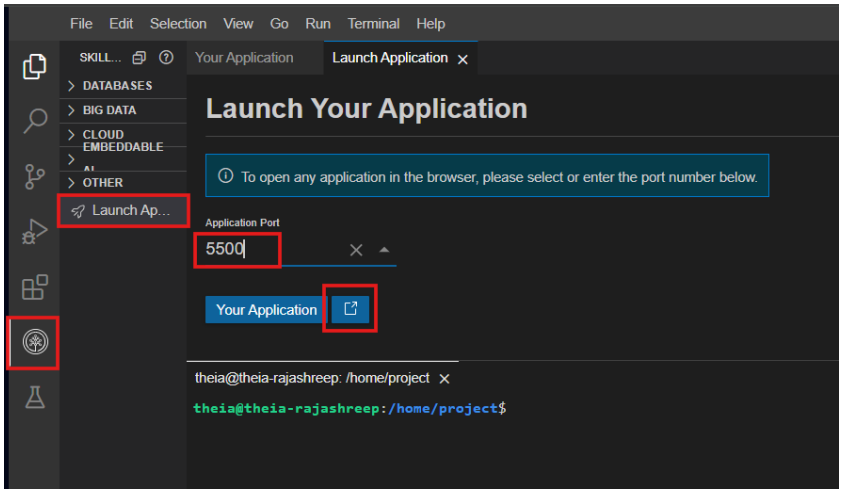
Preview and test your webpage's responsiveness

To preview your file, you can use the built-in Live Server extension by following the instructions below.

Part 1: Launch the webpage using Live Server

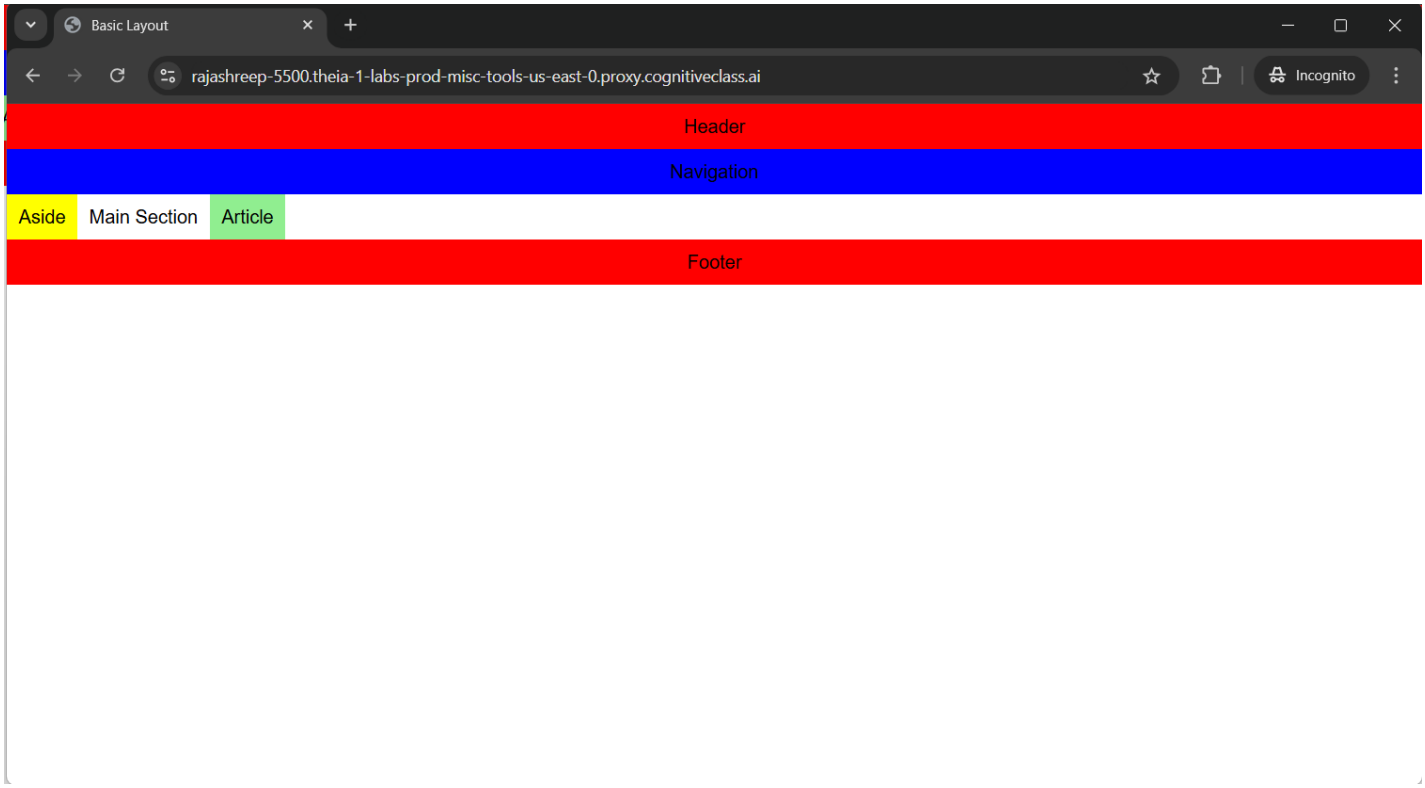
1. Open your file explorer and locate your HTML file
2. Right-click the file and select **Open with Live Server**
3. A notification will appear indicating that the server has started on port 5500
4. Click the button below the notification or use the **Skills Network Toolbox**, then navigate to **Other > Launch Application**, enter port `5500`, and click the launch button highlighted in the screenshot below. This will open the webpage in a new browser tab

Launch App

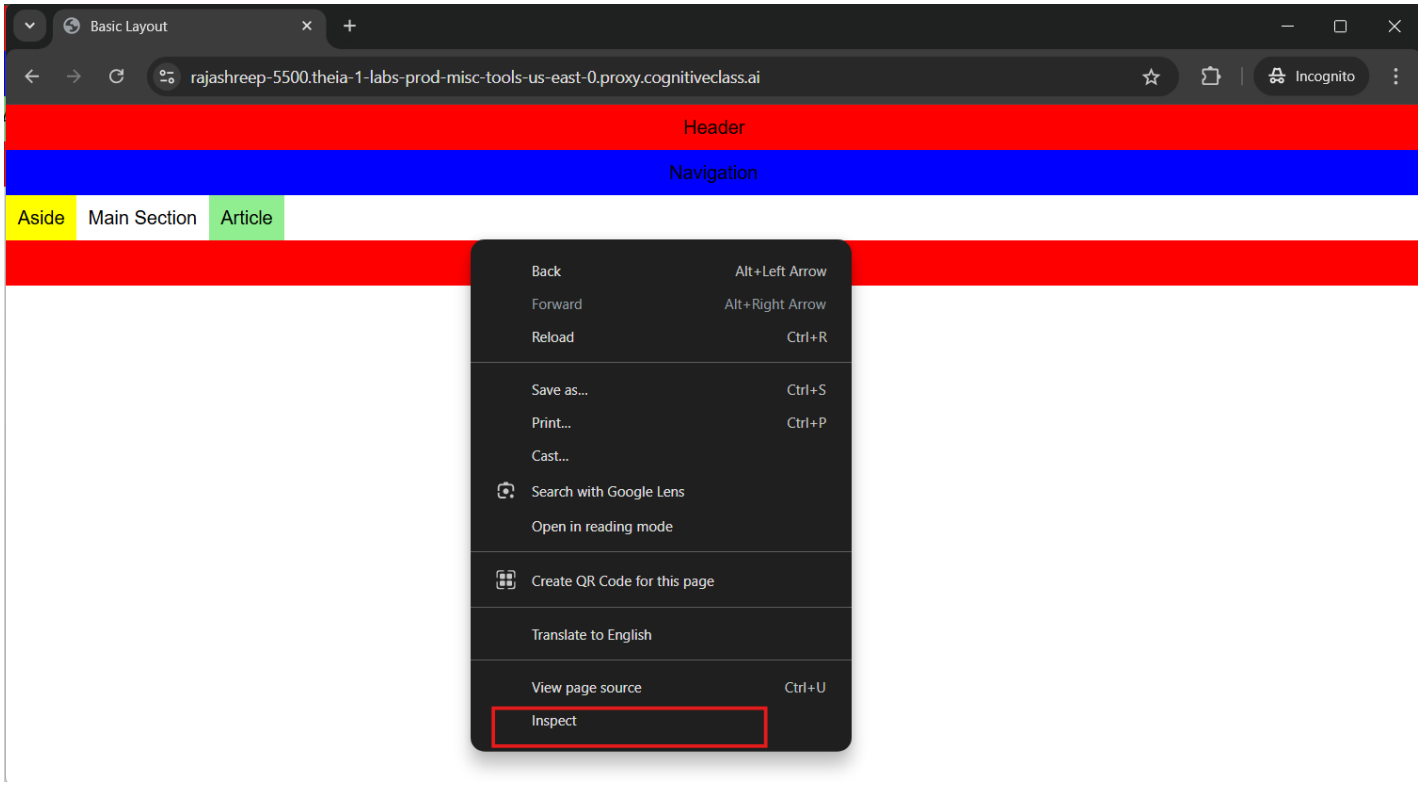


Part 2: Test responsiveness using Developer Tools

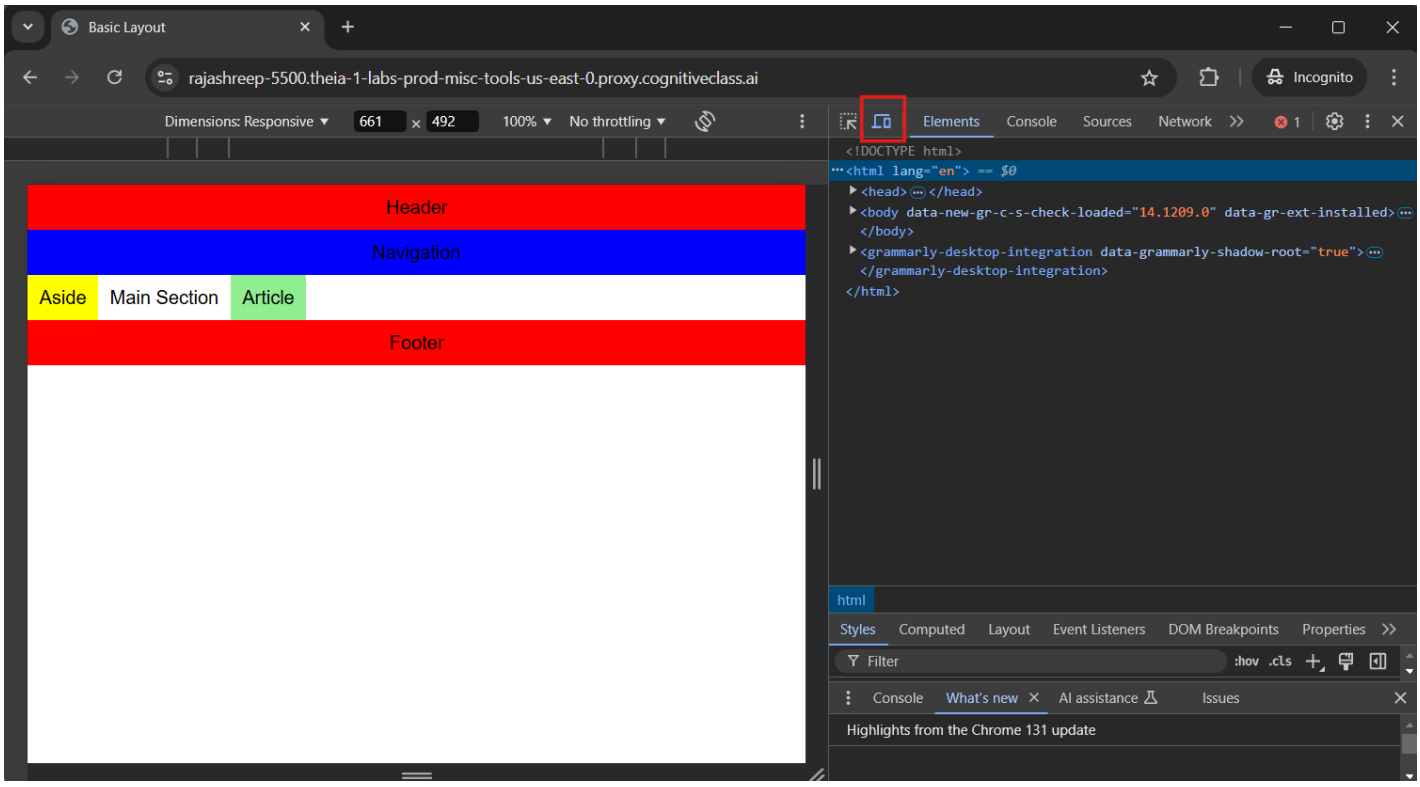
1. Make sure your webpage is opened in a new tab of the browser



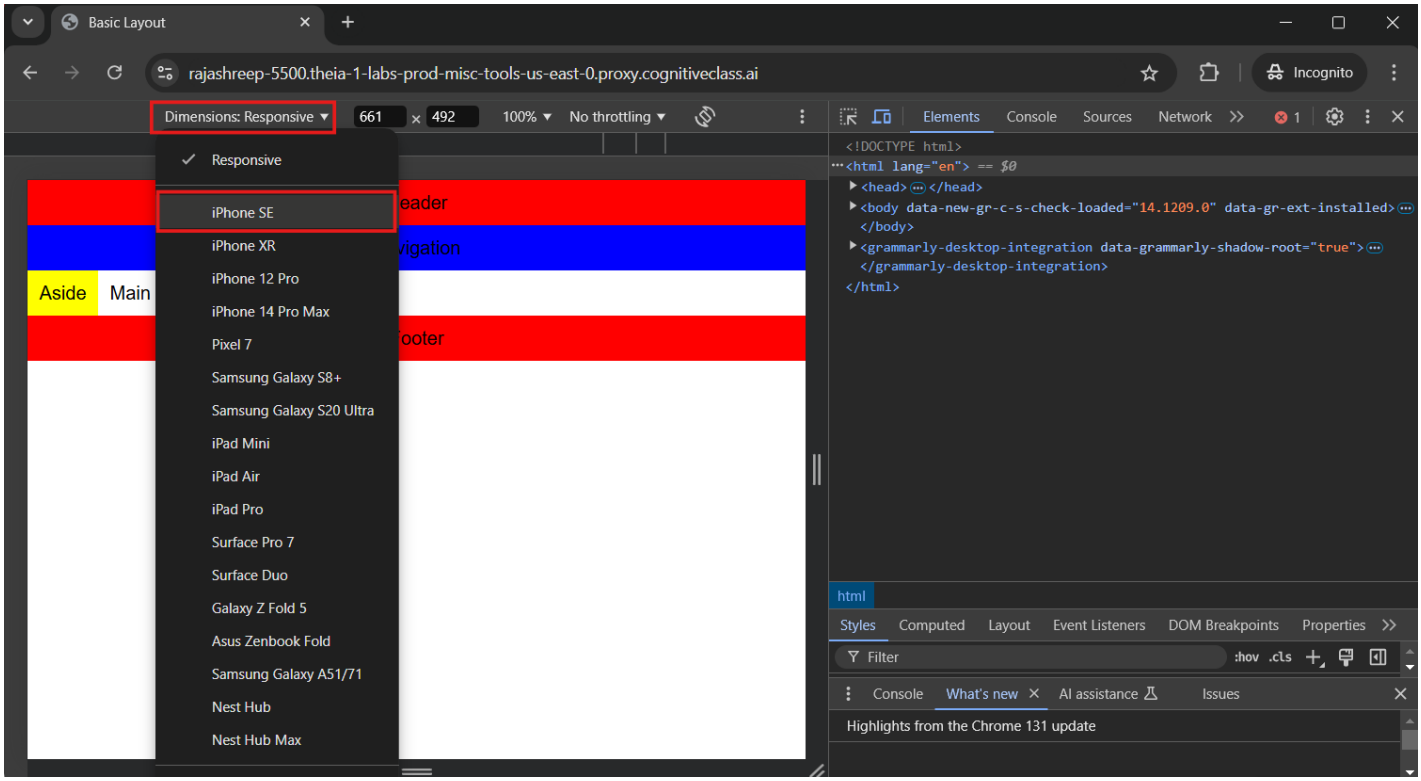
2. Right-click the webpage and select **Inspect** or **Inspect Element** (varies by browser)
   Alternatively, use the shortcut:
   - `Ctrl + Shift + I` (Windows/Linux)
   - `Cmd + Option + I` (Mac)



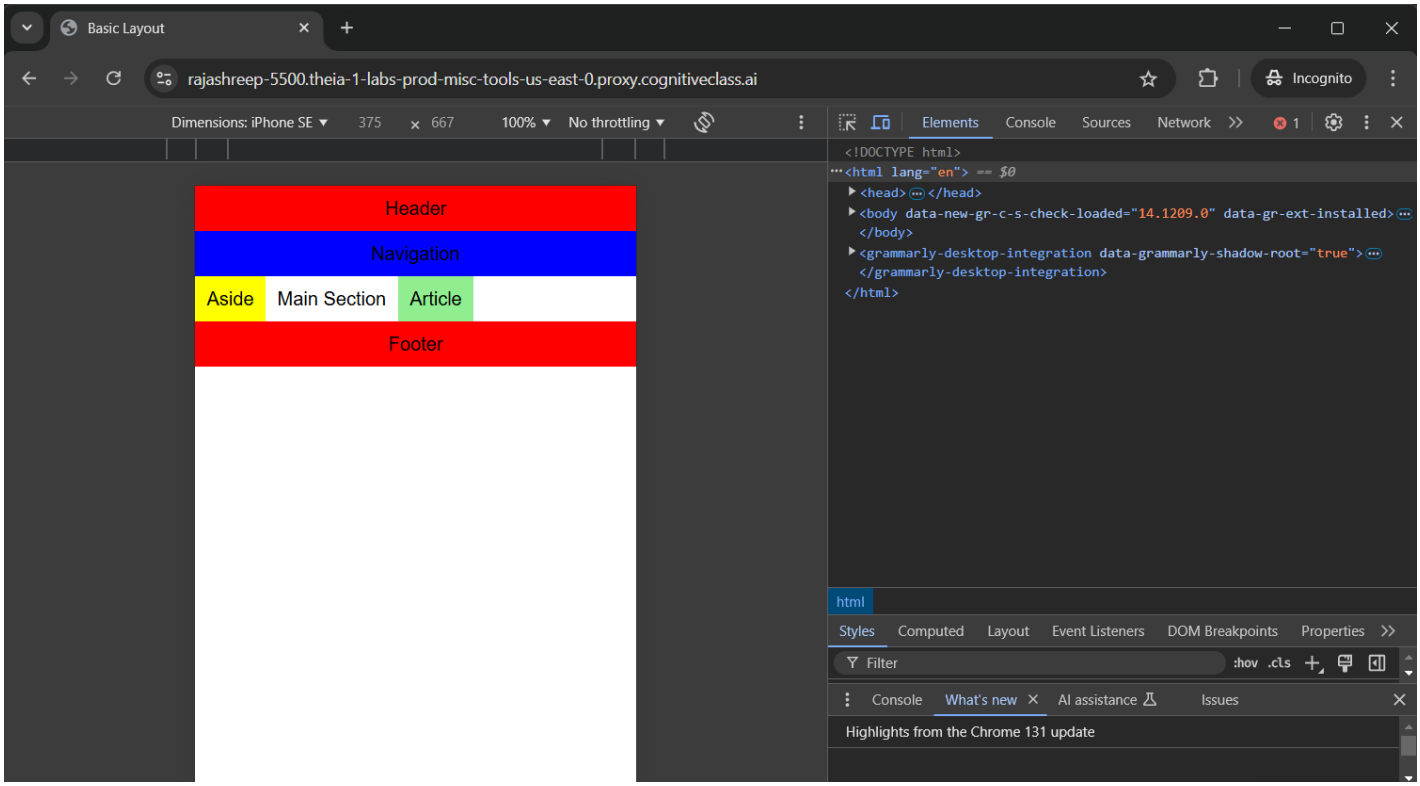
3. In the Developer Tools panel, click the **Device Toggle Toolbar** icon (phone/tablet icon) to switch to mobile view
   - Shortcut:
     - `Ctrl + Shift + M` (Windows/Linux)
     - `Cmd + Shift + M` (Mac)



4. Use the dropdown in the toolbar to choose preset devices (e.g., iPhone, iPad) or set custom dimensions to simulate different screen sizes



5. Observe that the layout has not readjusted according to the device size because the page is not yet responsive



6. To return to desktop view click the **Device Toggle Toolbar** icon again or use the shortcut (`Ctrl + Shift + M` or `Cmd + Shift + M`)

Exercise 2: Adding responsiveness with Tailwind CSS

Introduction to Tailwind CSS

Tailwind CSS is a utility-first CSS framework that enables developers to design directly in HTML. It provides responsive design utilities using **breakpoints** (`sm`, `md`, `lg`, and `xl`). These breakpoints correspond to different screen sizes, allowing the layout to adapt to various devices, from small mobile screens to large desktop monitors.

Steps to implement Tailwind CSS

1. Include the Tailwind CSS CDN in the `<head>` of your HTML:

```
<link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
```

2. Update the starter code to use Tailwind classes for responsiveness

Updated code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Responsive Layout - Tailwind</title>
  <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
</head>
<body class="bg-gray-100">
  <header class="bg-red-500 text-white text-center p-4">Header</header>
  <nav class="bg-red-500 text-white text-center p-4">Navigation</nav>
  <div class="flex flex-col md:flex-row">
    <aside class="bg-yellow-300 p-4 w-full md:w-1/4">Aside</aside>
    <section class="bg-white p-4 flex-grow">Main Section</section>
    <article class="bg-green-300 p-4 w-full md:w-1/4">Article</article>
  </div>
  <footer class="bg-blue-500 text-white text-center p-4">Footer</footer>
</body>
</html>
```

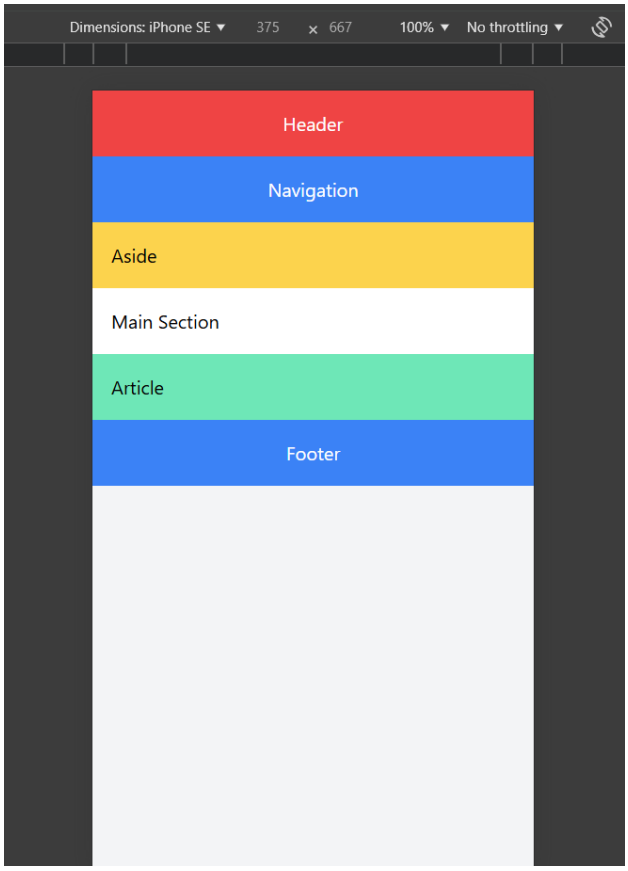
Key Tailwind classes used

1. **`flex` and `flex-col`**
   - **`flex`**: Activates flexbox layout, allowing easy alignment and distribution of child elements (the elements inside it) along a row or column
   - **`flex-col`**: By default, arranges child elements vertically in a column layout, stacking them one below the other
2. **`md:flex-row`**
   - **`md:`**: A responsive prefix that targets medium screens (768px and larger)
   - **`flex-row`**: Changes the flex container's layout from a column to a row on medium and larger screens, arranging child elements horizontally side by side
3. **`w-full` and `md:w-1/4`**
   - **`w-full`**: Sets the element's width to 100% of its parent container, making it full width on small screens
   - **`md:w-1/4`**: On medium and larger screens, sets the element's width to 25% of the container, making it more compact and allowing for a horizontal layout
4. **`p-4`**
   - **`p-4`**: Adds padding of 1rem (16px) to all sides of the element, providing consistent spacing around the content and improving visual appeal
5. **`min-h-screen`**
   - Ensures the element's height is at least as tall as the viewport height, so the layout fills the screen even if the content inside is minimal
6. **`bg-*`**
   - **`bg-*`** classes are used to set background colors, such as `bg-gray-100`, which help distinguish different sections and improve the overall visual clarity of the layout

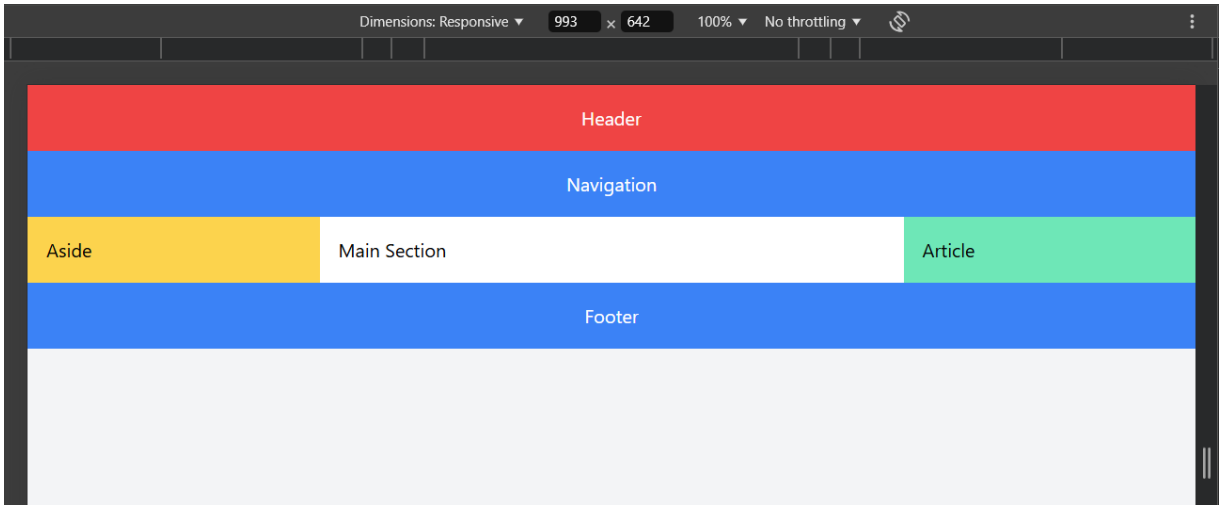
Test responsiveness:

Resize the browser window manually or select a preset device (e.g., iPhone, iPad) to simulate different screen sizes. Refer to the **Preview and test your webpage's responsiveness** section.

- On **mobile**, the sections stack vertically. The sections (aside, main, and article) will stack one below the other.



- On **tablet/desktop**, the layout will adjust to a more compact and horizontal design. The sections will align side by side (aside, main, and article), with the main section taking the central focus, and the navigation bar will be expanded with links shown directly.



Exercise 3: Adding responsiveness with Bootstrap

Introduction to Bootstrap

Bootstrap is a popular CSS framework with pre-designed components and a powerful grid system. It simplifies responsiveness with classes such as `col-*` and built-in navigation components.

Steps to implement Bootstrap CSS

1. Include the Bootstrap CSS and JavaScript CDN in the <head> of your HTML:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>
```

2. Create a new HTML file and insert the following code into it to use Bootstrap's grid system and include a hamburger menu

Updated code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Responsive Layout - Bootstrap</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
  <header class="bg-danger text-white text-center p-3">Header</header>
  <nav class="bg-primary 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" 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>
  <div class="container my-3">
    <div class="row">
      <aside class="col-12 col-md-3 bg-warning p-3">Aside</aside>
      <section class="col-12 col-md-6 bg-white p-3">Main Section</section>
      <article class="col-12 col-md-3 bg-success p-3">Article</article>
    </div>
    <div class="row">
      <footer class="col-12 bg-primary text-white text-center p-3">Footer</footer>
    </div>
  </div>
</body>
</html>
```

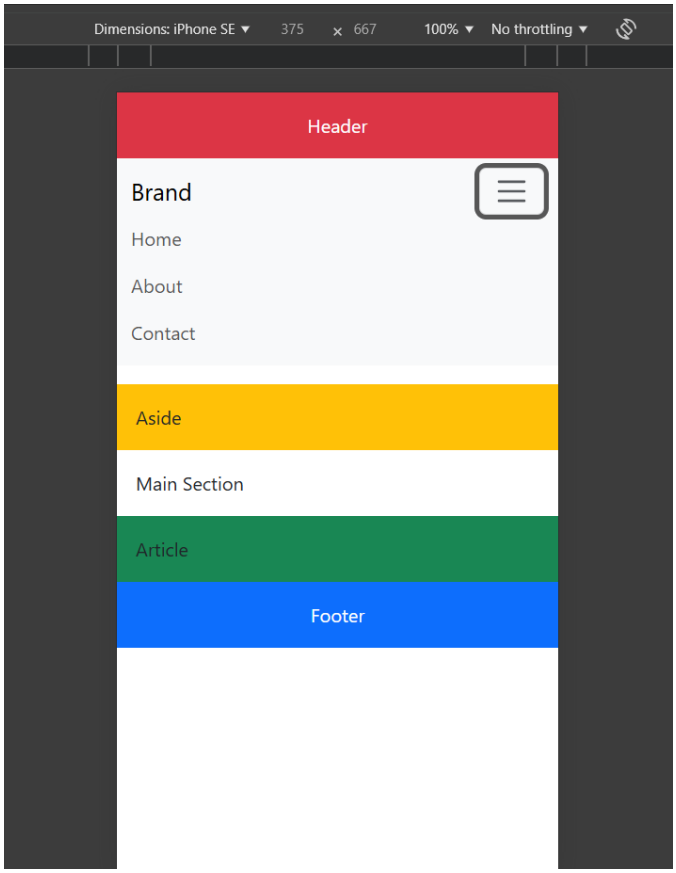
Key Bootstrap classes used

1. **`navbar` and `navbar-expand-lg`**
   - **`navbar`**: Creates a navigation bar with built-in features such as text alignment, background color, and dropdown menu support
   - **`navbar-expand-lg`**: Makes the navigation bar expand horizontally on large screens. On smaller screens, it collapses into a hamburger menu (three horizontal lines) that reveals the menu when clicked.
   - **Together**: The combination of `navbar` and `navbar-expand-lg` creates a responsive navigation bar that adapts to different screen sizes, expanding on large screens and collapsing on smaller ones.
2. **`col-*` classes**
   - **`col-12`**: In Bootstrap's grid system, `col-12` makes the element take up 100% width on extra small and small screens, stacking elements vertically on mobile devices.
   - **`col-md-3`**: For medium screens (768px and larger), `col-md-3` sets the element's width to 25%, allowing it to sit next to other elements in a horizontal layout.
   - **`col-md-6`**: On medium and larger screens, `col-md-6` makes the element take up 50% width, ensuring a balanced layout.
   - **Together**: Combining `col-12` for small screens and `col-md-*` for medium and larger screens ensures a responsive layout that adjusts dynamically.
3. **`p-3`**
   - **`p-3`**: Adds 1rem (16px) of padding to all sides of an element, ensuring there's space inside, preventing content from touching the edges, and improving the layout's visual appeal.
4. **`bg-danger`, `bg-primary`, `bg-warning`, and `bg-success`**
   - **`bg-*`**: These classes apply specific background colors to elements, helping distinguish sections and enhancing visual clarity.
5. **`navbar-toggler` and `navbar-toggler-icon`**
   - **`navbar-toggler`**: A button used to toggle the visibility of menu items in the navigation bar on smaller screens, showing or hiding the links.
   - **`navbar-toggler-icon`**: Displays the "hamburger menu" icon (three horizontal lines) inside the button, indicating the collapsible menu.

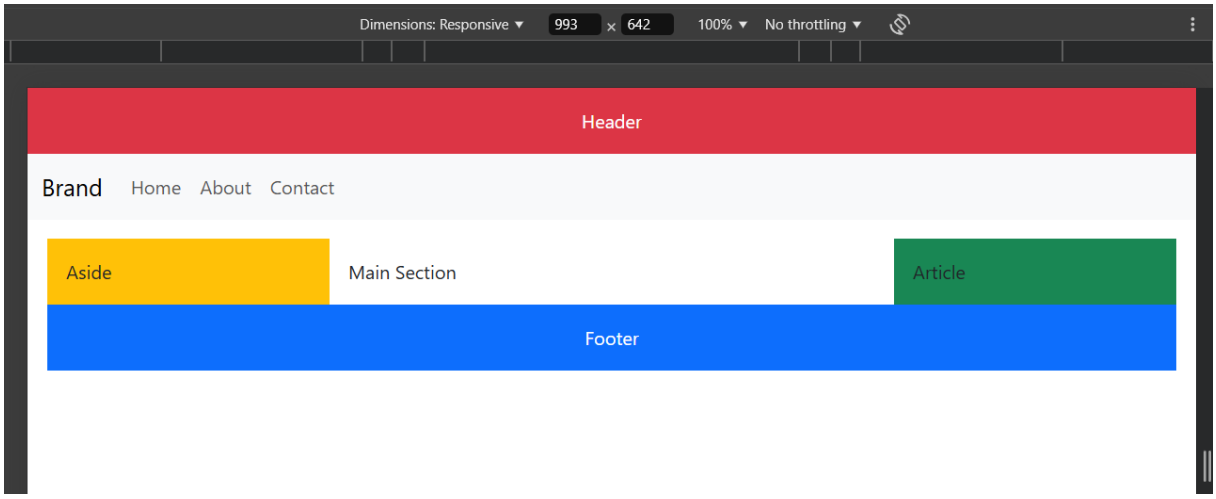
Test responsiveness:

Resize the browser window manually or select a preset device (e.g., iPhone, iPad) to simulate different screen sizes. Refer to the **Preview and test your webpage's responsiveness** section.

- On **mobile**, the layout should stack vertically. The navigation bar will collapse into a hamburger menu, and the sections (aside, main, and article) will stack one below the other.



- On **tablet/desktop**, the layout will adjust to a more compact and horizontal design. The sections will align side by side (aside, main, and article), with the main section taking the central focus, and the navigation bar will be expanded with links shown directly.



Exercise 4: Comparison between Tailwind CSS and Bootstrap

Feature	Tailwind CSS	Bootstrap
Approach	Utility-first, allows for full customization	Predefined components and styles
Customization	High level of customization and flexibility	Limited flexibility, relies on pre-designed components
Learning curve	Steeper, requires understanding of CSS	Easier, built-in classes and components
File size	Smaller, as unused styles can be purged	Larger, as it includes all components by default
Design control	Full control over design and layout	Limited control without overriding default styles
Grid system	Flexbox-based grid system	12-column grid system with predefined widths
Components	No pre-designed components	Offers a variety of pre-designed components such as buttons, modals, forms

Conclusion

In this lab, you created responsive layouts using **Tailwind CSS** and **Bootstrap**. You learned about the key features of both frameworks and implemented a responsive webpage that adapts to different screen sizes. With these skills, you can now build modern, mobile-friendly websites.

Author

Rajashree Patil