

Web Design & Development

Lab-Manual

Department of Computer Science University of Baltistan, Skardu

---------------------------------------------------------------

Name: Basit Ali

Registration No: S23BSCS012

Session: 2023-2027

Section: A

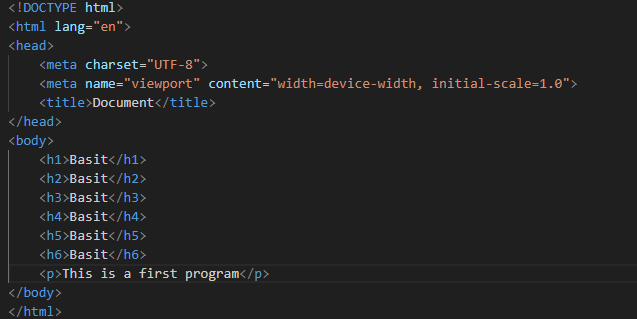
Submitted To: Sir Dilawar Abbas

**LAB#01: A basic HTML page showcasing structured content**

**Objective:** Learn the basics of HTML and structure a simple web page.  
**Task:**

1. Set up a basic HTML document.
2. Add elements like headings, paragraphs, images, and lists.
3. Practice with DOCTYPE, <html>, <head>, and <body> tags.

**Solution:**

****

**Output:**

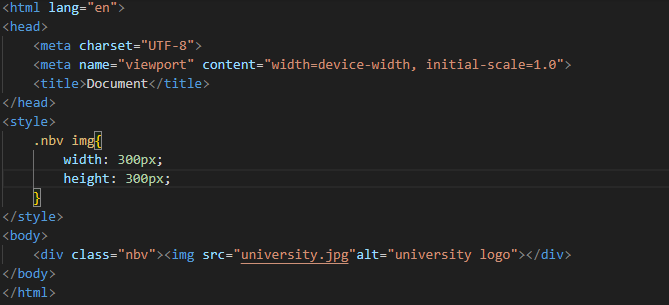
****

**LAB#02: A webpage with links and embedded images.**

**Objective:** Understand linking and embedding images.  
**Task:**

1. Add internal and external links.
2. Embed images using the <img> tag.
3. Experiment with attributes like alt, target, and href.

**Solution:**

****

**Output:**

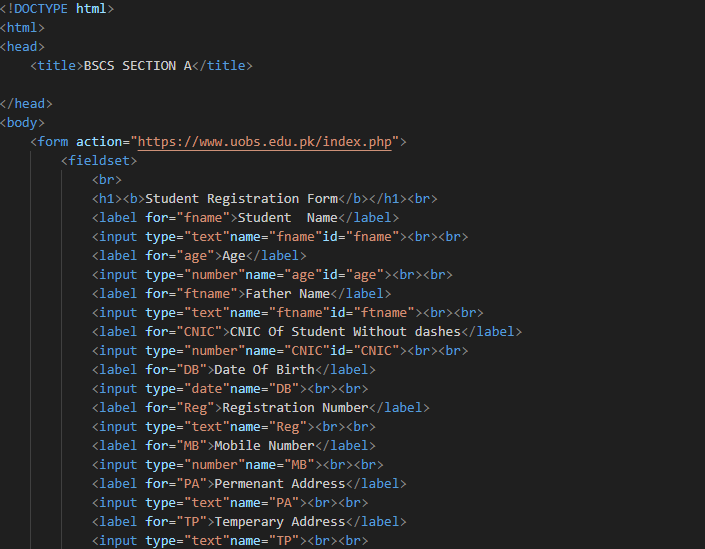
****

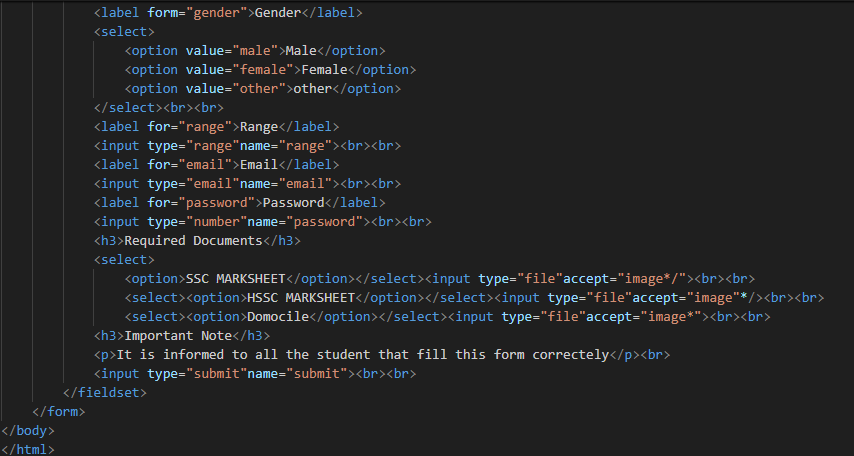
**LAB#03: A simple form for data entry.**

**Objective:** Learn to create interactive HTML forms for user input.  
**Task:**

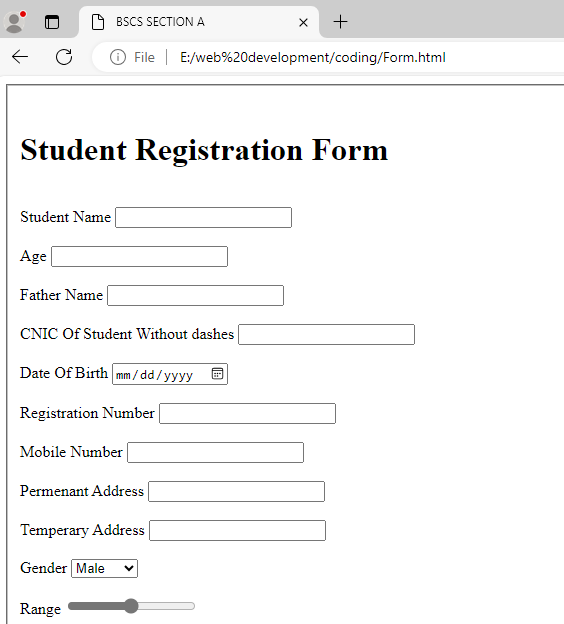
1. Create a form with input fields, checkboxes, radio buttons, and a submit button.
2. Explore form attributes like action, method, and placeholder.

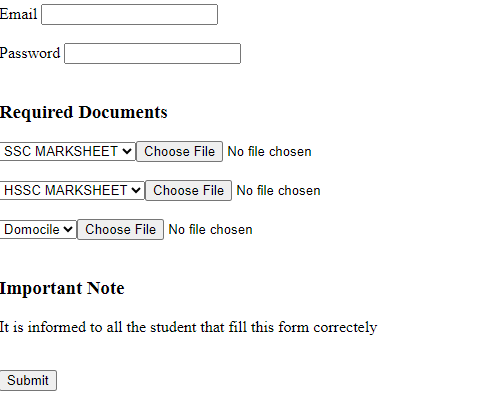
**Solution:**





**Output:**



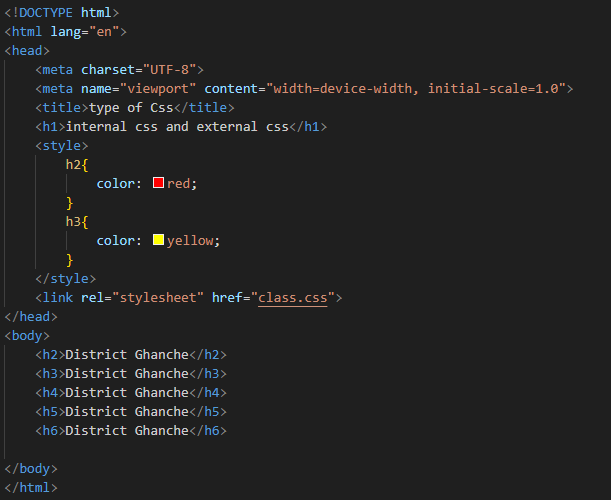


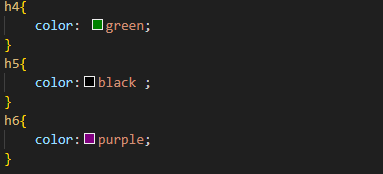
**LAB#04: A styled HTML page with CSS applied to various elements.**

**Objective:** Apply CSS to style HTML content.  
**Task:**

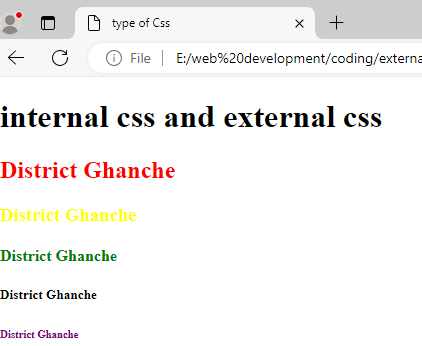
1. Apply basic CSS rules using inline, internal, and external styles.
2. Style text, colors, and backgrounds.

**Solution:**





**Output:**

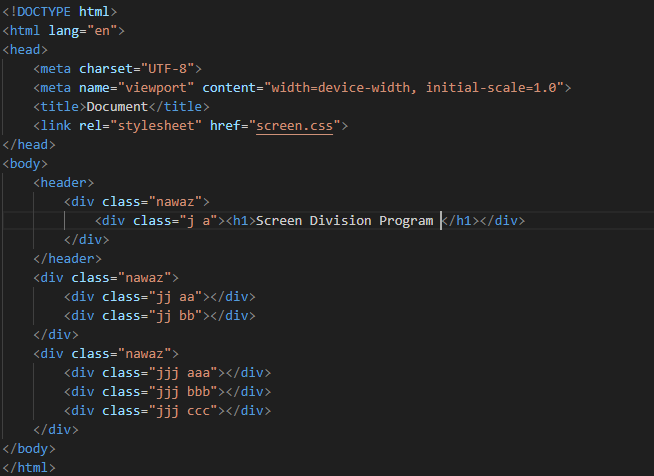


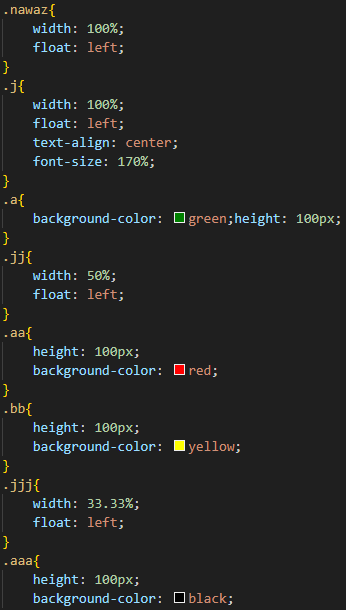
**LAB#05: A webpage that uses the Box Model to layout content neatly.**

**Objective:** Understand and utilize the CSS Box Model.  
**Task:**

1. Experiment with padding, margin, and border.
2. Adjust width and height for layout control.

**Solution:**

****

****

**Output:**

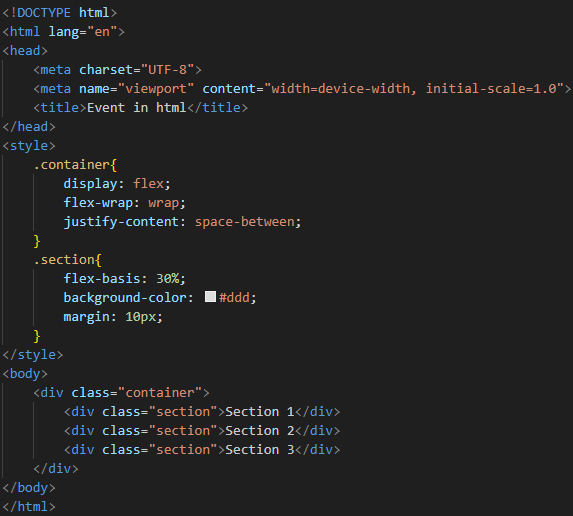
****

**LAB#06: A responsive layout using Flexbox.**

**Objective:** Implement CSS Flexbox for responsive layouts.  
**Task:**

1. Use Flexbox properties for organizing content in rows and columns.
2. Apply alignment properties like justify-content and align-items.

**Solution:**

****

**Output:**

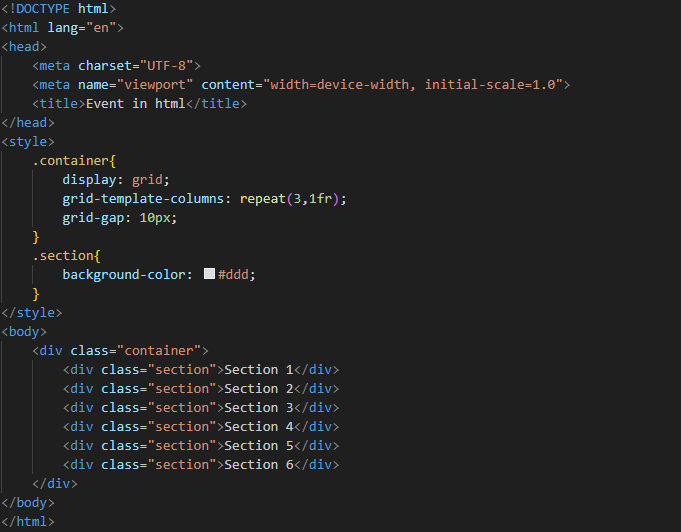
****

**LAB#07: A structured page layout using CSS Grid.**

**Objective:** Create complex layouts using CSS Grid.  
**Task:**

1. Define grid containers and items.
2. Experiment with grid-template-rows, grid-template-columns, and gap.

**Solution:**

****

**Output:**

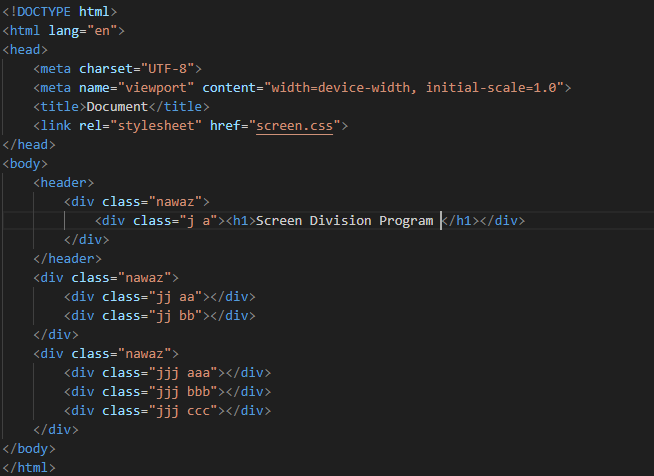
****

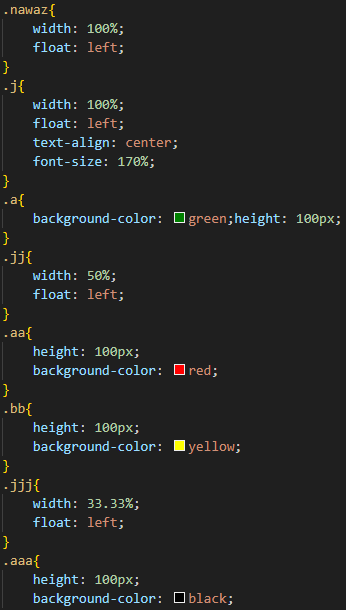
**LAB#08: A responsive webpage that adapts to various screen sizes.**

**Objective:** Apply media queries for responsive design.  
**Task:**

1. Create breakpoints for different screen sizes.
2. Adjust layouts and fonts for mobile, tablet, and desktop.

**Solution:**

****

****

**Output:**

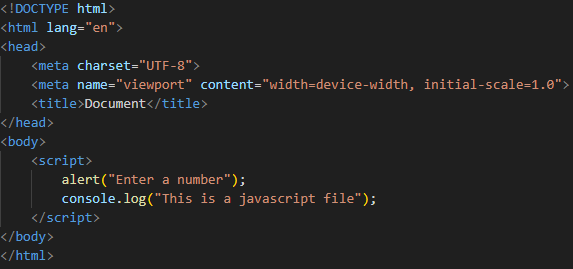
****

**LAB#09: A page with simple JavaScript interactivity.**

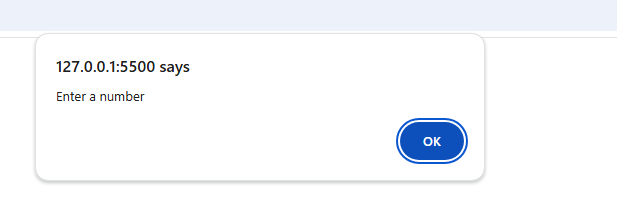
**Objective:** Learn basic JavaScript and apply it to web pages.  
**Task:**

1. Add a script to display alerts and modify content.
2. Use console.log() for debugging.

**Solution:**

****

**Output:**

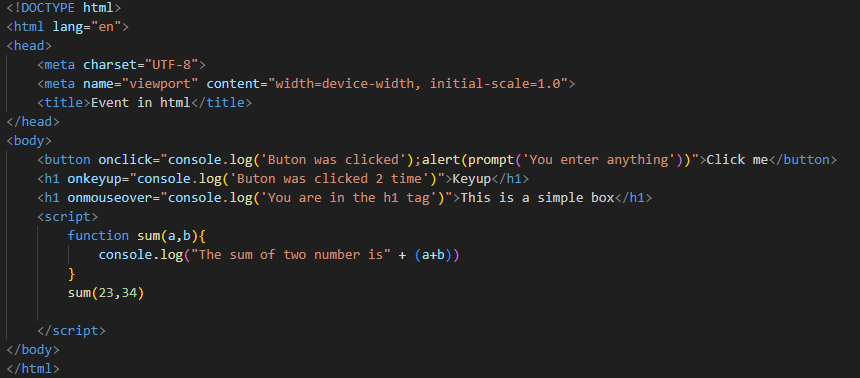
****

**LAB#10: Webpage with interactive buttons and form elements.**

**Objective:** Understand JavaScript functions and event handling.  
**Task:**

1. Write functions for basic calculations.
2. Use onclick, onmouseover, and onkeyup events.

**Solution:**

****

**Output:**

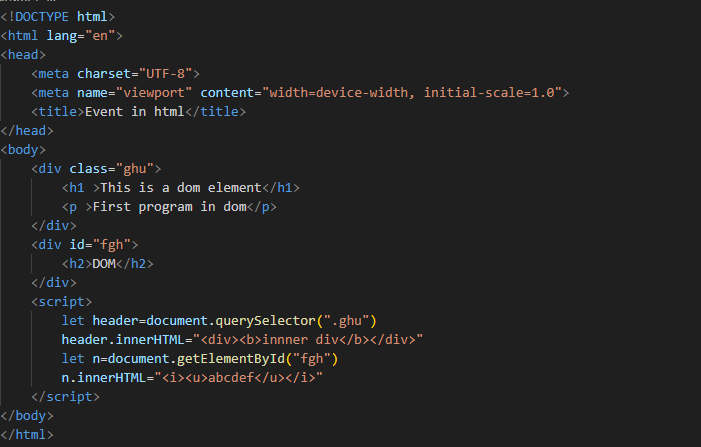
****

**LAB#11: Interactive page that changes content or style based on user actions.**

**Objective:** Modify HTML content dynamically using the DOM.  
**Task:**

1. Access and modify elements using getElementById, querySelector, etc.
2. Change HTML and CSS properties using JavaScript.

**Solution:**

****

**Output:**

****

**LAB#12: A form with client-side validation.**

**Objective:** Add validation to HTML forms using JavaScript.  
**Task:**

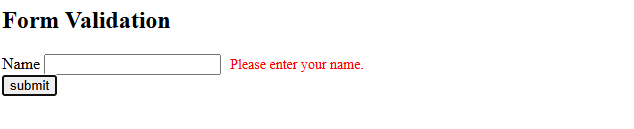
1. Write JavaScript to check form inputs.
2. Display error messages for invalid inputs.

**Solution:**

****

****

**Output:**

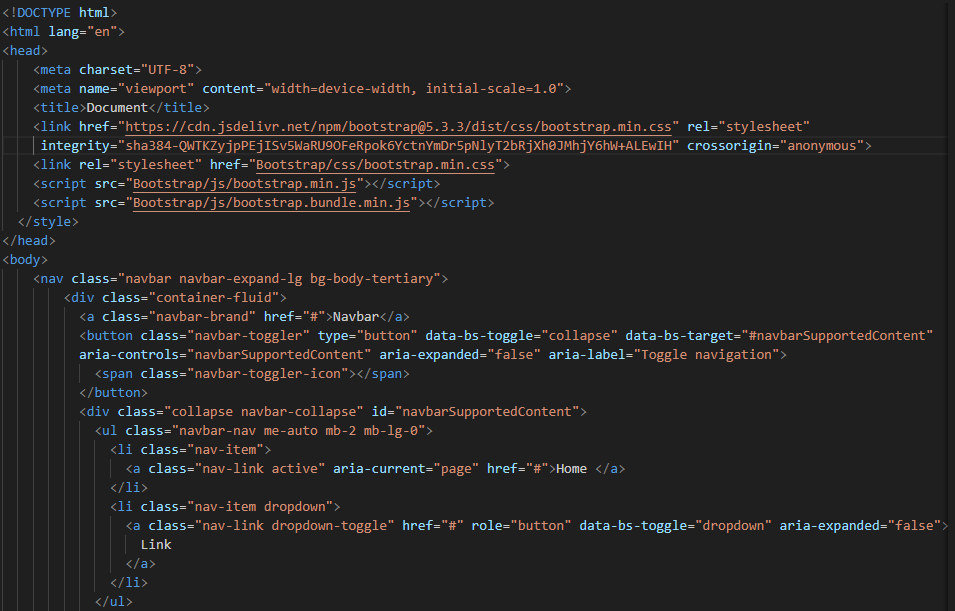
****

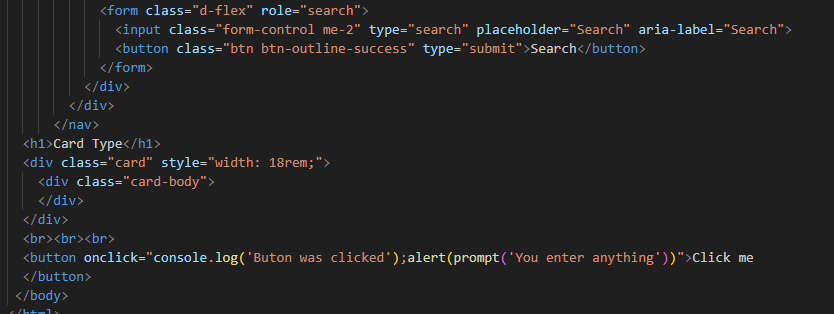
**LAB#13: A webpage styled and structured with Bootstrap.**

**Objective:** Use Bootstrap for responsive design and styling.  
**Task:**

1. Link Bootstrap CSS and JavaScript.
2. Add responsive components like Navbar, Cards, and Buttons.

**Solution:**

****

****

**Output:**

****

**LAB#14: A page that changes content or style when the user interacts with it.**

**Objective:** Use JavaScript to dynamically manipulate HTML elements.  
**Tasks:**

1. Access HTML elements by id and class using document.getElementById and document.querySelector.
2. Change text, style, and attributes of HTML elements based on user actions.

**Solution:**

****

**Output:**

****

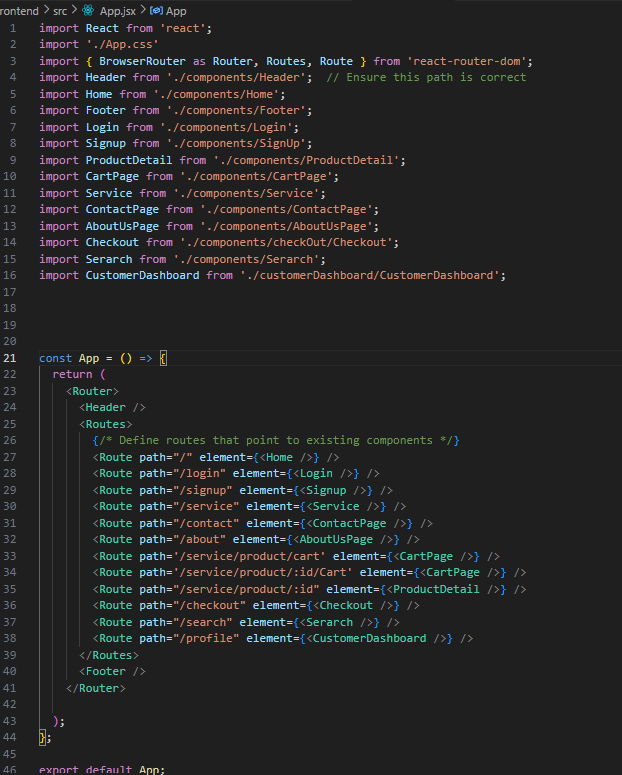
**LAB#15: A complete website that incorporates the learned concepts and skills.**

**Objective:** Create a multi-page website as a final project.  
**Task:**

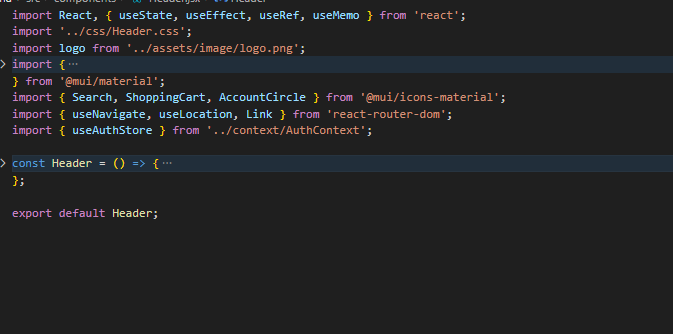
1. Design a website with 3-5 pages, including a homepage, about, and contact page.
2. Use HTML, CSS, JavaScript, to style and structure the site.
3. Implement form validation and interactive elements.

**Solution:**

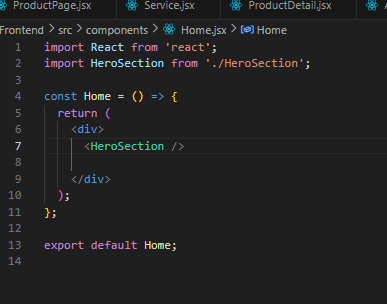
**App.js**

****

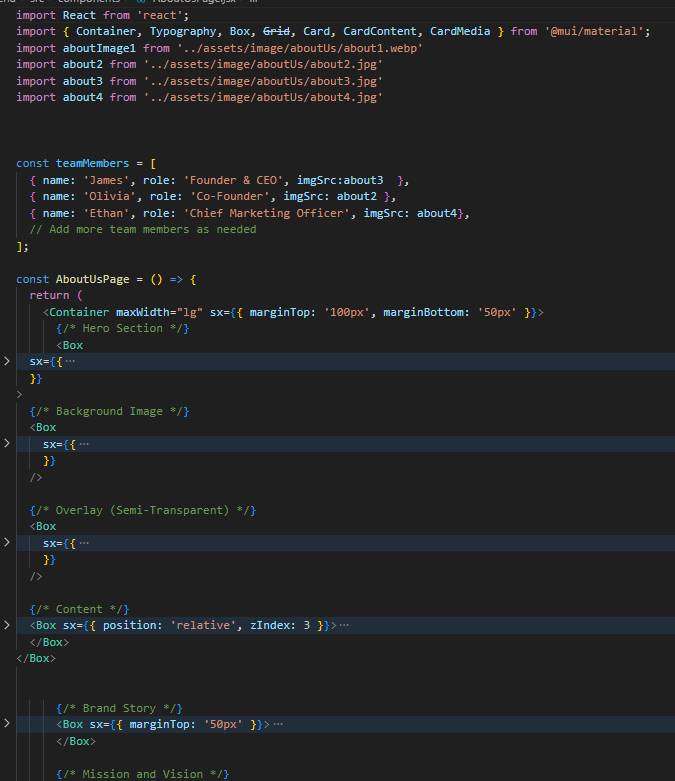
**Navbar.js**

****

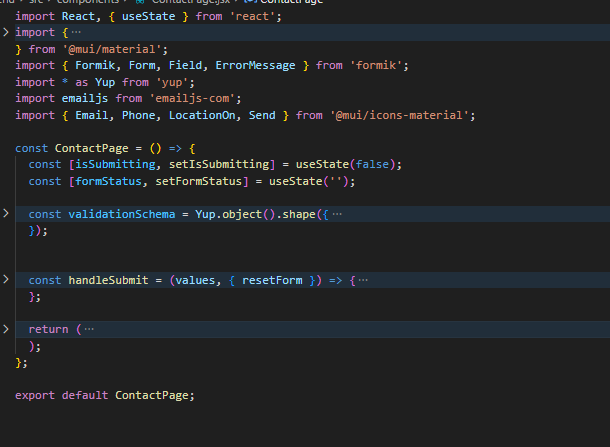
**Home.js**

****

**About.js**

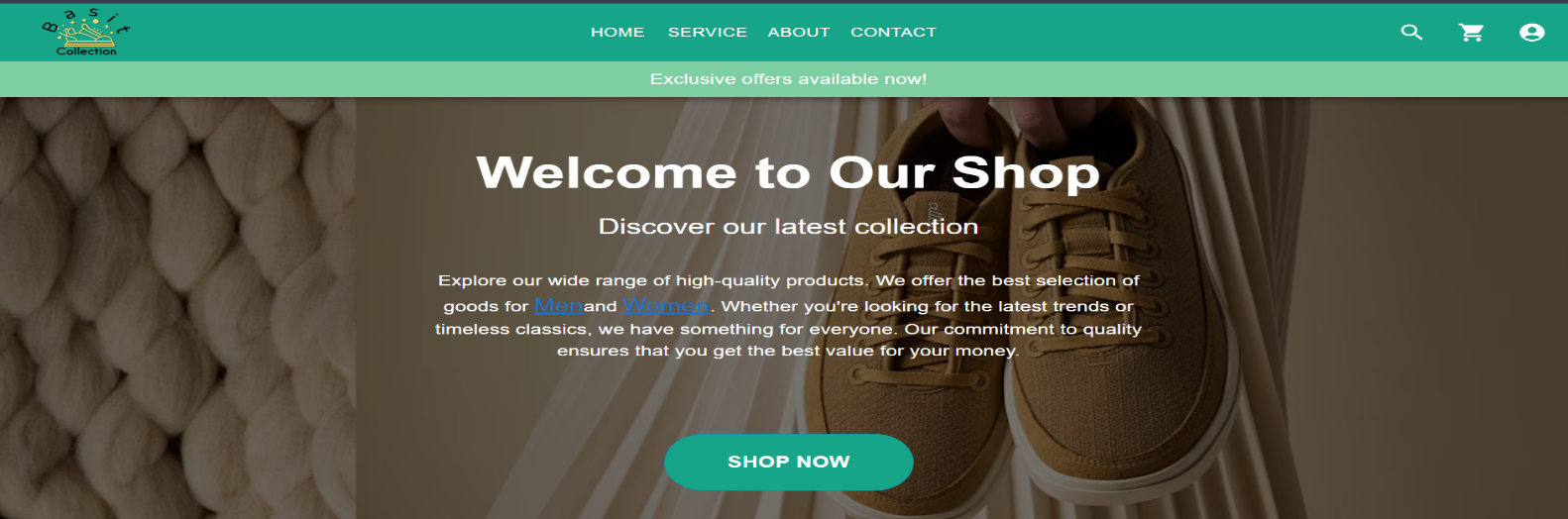
****

**Contact.js**

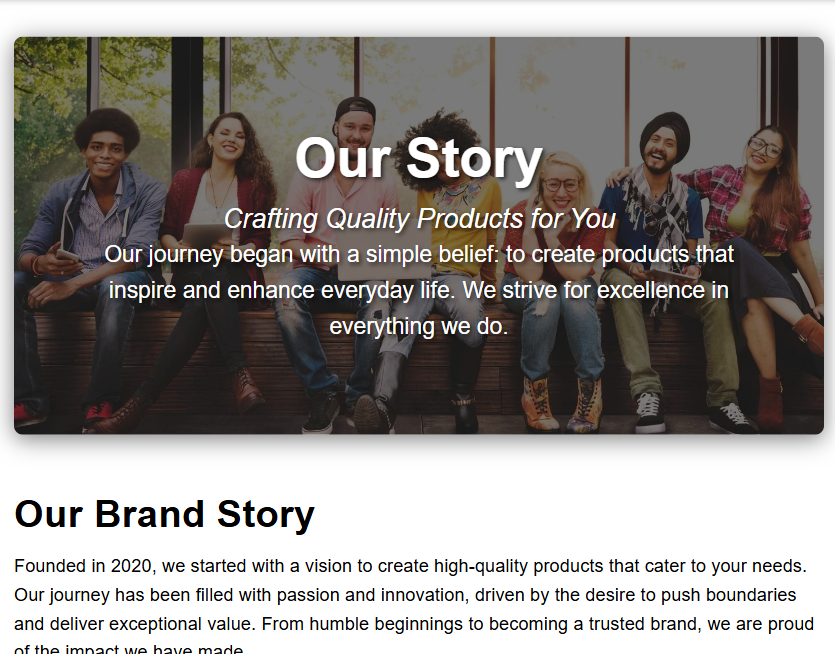
****

**Output:**

**Navbar and home page**



About page:



Contact page :

