Day: 1
Topic: HTML Basics – Personal Bio Webpage

Objective

The objective of Day 1 was to understand the basic structure of HTML and create a simple personal bio webpage using standard and semantic HTML tags.

Tools Used

- Visual Studio Code (VS Code)
- Web Browser (Chrome/Edge)

Work Done

On Day 1, I learned the fundamentals of HTML, including document structure, basic tags, and semantic elements. I created a new project folder and an index.html file using VS Code. I implemented a complete HTML5 document structure using <!DOCTYPE html>, <html>, <head>, and <body> tags.

I designed a personal bio webpage that includes:

- A header section with my name and role
- An “About Me” section describing my interest in web development
- An education section
- A skills list using unordered lists
- A contact section with email and location
- A footer section

Semantic tags such as <header>, <section>, and <footer> were used to improve structure and readability.

Output

A fully functional personal bio webpage was successfully created and displayed correctly in the browser.

Conclusion

Day 1 helped me understand the basics of HTML and how to structure a simple webpage. This practical session built a strong foundation for further learning in CSS and JavaScript.

Day: 2
Topic: CSS Basics – Styling Personal Bio Page

Objective

The objective of Day 2 was to learn basic CSS syntax and apply styling to the HTML personal bio page created on Day 1.

Tools Used

- Notepad (for editing files)
- Web Browser (Chrome/Edge)

Work Done

On Day 2, I created an external CSS file named style.css and linked it to my existing index.html file using the <link> tag inside the <head> section.
I learned the basic structure of CSS, which includes selectors, properties, and values.

I applied styles to different HTML elements such as:

- body to set font, background color, margin, and padding
- header and footer to add background color, text color, alignment, and spacing
- section to create white content boxes with margin, padding, and rounded corners
- h2 to change heading color
- ul to adjust list spacing

Output

The personal bio webpage now appears visually improved with better colors, spacing, alignment, and readability.

Conclusion

Day 2 helped me understand how CSS controls the appearance of a webpage. I learned how to separate structure (HTML) from design (CSS), which is an important concept in web development.

Day: 3
Topic: JavaScript Basics – Adding Interactivity

Objective

The objective of Day 3 was to learn the basics of JavaScript and add simple interactivity to the personal bio webpage.

Tools Used

- Notepad (for editing files)
- Web Browser (Chrome/Edge)

Work Done

On Day 3, I created a JavaScript file named script.js and linked it to the HTML file using the <script> tag before the closing </body> tag.
I added a button in the “About Me” section of the webpage and assigned it an ID.

Using JavaScript, I:

- Selected the button using getElementById
- Added a click event using addEventListener
- Changed the text of a paragraph using innerHTML when the button was clicked

This demonstrated how JavaScript can respond to user actions and modify webpage content dynamically.

Output

When the button is clicked, a message appears on the webpage saying that JavaScript is working, proving that the script is functioning correctly.

Conclusion

Day 3 helped me understand how JavaScript adds interactivity to a webpage. I learned how events and DOM manipulation work together to create dynamic web pages.

Day: 4
Topic: Responsive Design Using Bootstrap

Objective

The objective of Day 4 was to make the personal bio webpage responsive so that it works properly on different screen sizes such as mobile, tablet, and desktop.

Tools Used

- Notepad (for editing files)
- Web Browser (Chrome/Edge)
- Bootstrap Framework

Work Done

On Day 4, I added the Bootstrap CDN link inside the <head> section of the HTML file. I wrapped the main content inside a Bootstrap container to center and space the layout properly.

I used Bootstrap's grid system by applying:

- row and col classes
- Responsive column classes like col-12 and col-md-8
- Utility classes like bg-white, p-3, rounded, and mb-3

I also styled the button using Bootstrap class btn btn-primary.

These changes made the webpage automatically adjust its layout based on screen size.

Output

The webpage now displays correctly on all screen sizes. On small screens, content stacks vertically, and on larger screens it stays centered with proper spacing.

Conclusion

Day 4 helped me understand responsive design and how frameworks like Bootstrap make layout creation faster and easier. I learned how to use the grid system and utility classes to build mobile-friendly web pages.

Day: 5
Topic: DOM Manipulation Using JavaScript

Objective

The objective of Day 5 was to learn how to manipulate the DOM (Document Object Model) using JavaScript to dynamically change webpage content and styles based on user actions.

Tools Used

- Notepad (for editing files)
- Web Browser (Chrome/Edge)

Work Done

On Day 5, I updated the HTML file by adding two new buttons: “Change Background” and “Reset.” These buttons were given unique IDs so they could be accessed using JavaScript.

In the JavaScript file:

- I selected HTML elements using getElementById
- I added event listeners to multiple buttons
- I changed text using innerHTML
- I changed page background color using style.backgroundColor
- I reset the changes using a reset function

This showed how JavaScript can control webpage content and styles in real time.

Output

The webpage now responds to button clicks by changing text, changing background color, and resetting everything back to normal.

Conclusion

Day 5 helped me understand how the DOM works and how JavaScript can be used to control and modify webpage elements dynamically. This is an important step toward building interactive web applications.

Day: 6
Topic: APIs and Asynchronous JavaScript

Objective

The objective of Day 6 was to understand what an API is and how to fetch data from an external source using JavaScript with asynchronous programming (fetch, async, and await).

Tools Used

- Notepad (for editing files)
- Web Browser (Chrome/Edge)
- Public API (JSONPlaceholder)

Work Done

On Day 6, I added a new button and a display area in my HTML file to show data fetched from an API.
I updated the JavaScript file to:

- Use the fetch() function to request data from a public API
- Convert the response into JSON format
- Use async and await to handle asynchronous code
- Display the fetched data (name, email, and city) on the webpage
- Handle errors using try...catch

This showed how a webpage can get live data from external sources.

Output

When the “Load API Data” button is clicked, user information is fetched from the API and displayed on the webpage. If there is an error, an error message is shown.

Conclusion

Day 6 helped me understand how APIs work and how JavaScript can communicate with external servers. I learned the basics of asynchronous programming and how to use fetch, async, and await to build dynamic web applications.

Day: 7
Topic: Capstone Project – Personal Portfolio Website

Objective

The objective of Day 7 was to combine all the concepts learned during the course and build a complete personal portfolio website.

Tools Used

- Notepad (for editing files)
- Web Browser (Chrome/Edge)
- HTML, CSS, JavaScript, Bootstrap

Work Done

On Day 7, I created a single-page personal portfolio website using HTML for structure, CSS and Bootstrap for styling and responsiveness, and JavaScript for interactivity.

The website includes:

- A header section with my name and role
- An About section with a welcome button
- A Skills section listing my technical skills
- A Projects section showing sample projects
- A Contact section with email and location
- A Footer section

Using JavaScript, I added a button that displays a welcome message when clicked.

Bootstrap classes were used to make the layout responsive on different screen sizes.

Output

The portfolio website works correctly, looks clean on all screen sizes, and shows interactive behavior when the button is clicked.

Conclusion

Day 7 helped me apply all the skills learned during the course into one complete project. I now understand how HTML, CSS, JavaScript, Bootstrap, DOM, and APIs work together to build real websites.