**AI ASSISTED CODING**

**LAB-14: Web Design Application – AI-Assisted HTML/CSS/JS Generation**

Roll no: 2503A51L40

Name: E. Harini

Batch: 25BTCAICSB20

**Task -1 Description: Create a Responsive Web Page Layout**

**Instructions:**

**• Design a basic web page layout with a header, main content area, and footer using HTML and CSS.**

**• Use AI to assist in generating responsive CSS for different**

**screen sizes.**

**• Ensure the layout is clean and visually organized**

**1.Prompt:**

Assist in generating responsive CSS for different screen sizes. Ensure the layout is clean and visually organized.

**2.Code Generated:**

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

**3.Output:**

(Responsive website- Changes its size automatically based on device)

A screenshot of a computer

AI-generated content may be incorrect.

**4.Observation:**

The responsive web page layout designed using HTML and AI-assisted CSS works efficiently across different screen sizes such as mobile, tablet, and desktop. The structure includes a well-defined header, main content area, and footer, maintaining a neat and balanced appearance. The AI-generated CSS code ensures flexibility and visual consistency when resizing. Overall, the layout demonstrates effective use of responsive design principles and clean styling.

**Task-2 Description: Interactive Button with JavaScript**

**1.Prompt:**

Generate JavaScript code that displays an alert message when the button is clicked, also ensure the code is clean and well-commented.

**2.Code Generated :**

A computer screen shot of a program code

AI-generated content may be incorrect.

**3.Output:** (When button is clicked)

A screenshot of a computer

AI-generated content may be incorrect.

**4.Observation:**

The interactive button functions properly by displaying an alert message when clicked, as per the task requirement. The JavaScript code generated by AI is clean, easy to understand, and includes meaningful comments that enhance clarity. The implementation successfully demonstrates user interaction on the webpage. This task highlights how AI assistance can simplify JavaScript coding for beginners while ensuring functional accuracy.

**Task-3 Description: Form with Validation**

**1.Prompt:** Generate JavaScript code for form validation (e.g.,non-empty fields, valid email format).Add inline error messages if input is invalid.

**2.Code Generated:**

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen shot of text

AI-generated content may be incorrect.

**4.Observation:**

The contact form includes fields for Name, Email, and Message, and effectively validates user input before submission. The AI-generated JavaScript detects empty fields and invalid email formats, displaying appropriate inline error messages. This enhances user experience and ensures data accuracy. The validation logic is simple yet efficient, making the form interactive, reliable, and user-friendly.

**Task 4: AI-Assisted E-Commerce Product Page**

**Scenario:**

**A startup wants a basic e-commerce product page to display products**

**with prices and an “Add to Cart” button.**

**• Use Copilot to generate a grid-based product catalog in**

**HTML/CSS.**

**• Implement a JavaScript “Add to Cart” functionality with**

**Copilot’s guidance.**

**• Modify Copilot’s suggestions to include a cart counter at the**

**top-right corner of the page.**

**1.Prompt:**

A startup wants a basic e-commerce product page to display products

with prices and an “Add to Cart” button.

**2.Code Generated:**

A computer screen shot of a program

AI-generated content may be incorrect. A screen shot of a computer

AI-generated content may be incorrect.

**3.Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**4.Observation:**

The code creates a basic e-commerce product page using HTML, CSS, and JavaScript in a single file. It displays products in a responsive grid layout with images, names, prices, and an “Add to Cart” button. A cart counter is placed at the top-right corner inside the header and dynamically updates whenever the user clicks “Add to Cart.” The implementation is simple, lightweight, and serves as a starting point for building a full shopping cart system.