

Lab Assignment-14

Task 1: 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.

Expected Output:

- A responsive web page with:
 - Header with navigation links
 - Main section with placeholder text/images
 - Footer with copyright or contact info
- Layout adapts correctly to desktop, tablet, and mobile screen sizes.

The screenshot shows a web development interface with three main panels: HTML, CSS, and JS. The HTML panel contains the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Responsive Web Page</title>
<link rel="stylesheet" href="style.css" />
</head>
<body>
```

The CSS panel contains the following styles:

```
/* Reset default browser styles */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

/* Basic page styles */
body {
    font-family: 'Poppins', sans-serif;
    line-height: 1.6;
    background-color: #f4f4f4;
```

The preview window shows a dark-themed website titled "My Website". The header includes a navigation bar with "Home", "About", "Services", and "Contact". The main content area displays the text "Welcome to My Page" and a placeholder image. The footer is visible at the bottom of the page.

Task 2: Interactive Button with JavaScript

Instructions:

- Create a **button** on a web page.
- Use AI to generate JavaScript code that displays an **alert message** when the button is clicked.
- Ensure the code is clean and well-commented.

Expected Output:

- A web page with a button.
- Clicking the button shows a pop-up alert message, e.g., “Button clicked!”.

The screenshot shows a code editor interface with two tabs: "HTML" and "CSS". The "HTML" tab contains the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Interactive Button</title>
<style>
/* Basic Page Styling */
body {
    font-family: Arial, sans-serif;
    background-color: #f9f9f9;
    text-align: center;
    padding-top: 100px;
}
```

The "CSS" tab is currently empty. Below the code editor, there is a blue button with the text "Click Me!".

Task 3: Form with Validation

Instructions:

- Design a **contact form** with fields: Name, Email, Message.
- Use AI to generate JavaScript code for **form validation** (e.g., non-empty fields, valid email format).
- Add **inline error messages** if input is invalid.

Expected Output:

- A functional contact form where:
 - Submitting with empty fields shows error messages.
 - Submitting with valid input displays a “Form submitted successfully” message.

```
150     successMessage.textContent = "Form  
151     submitted successfully!";  
152     form.reset(); // Clear the form  
153   }  
154 };  
155 </script>  
156 </body>  
157 </html>  
158 |
```

Save

Enter your name

Email

Enter your email

Message

Write your message...

Task 4: Dynamic Content Generation

Instructions:

- Create a list of items (e.g., product names) using **HTML**.
- Use AI-generated JavaScript to **dynamically add or remove items** from the list when a button is clicked.

Expected Output:

- A web page with a list and buttons:
 - “Add Item” adds a new item to the list dynamically.
 - “Remove Item” deletes the last item.
- Updates occur without reloading the page.

```
count--;
Decrease counter
} else {
    alert("No more items to remove!");
    // Alert if list is empty
}
});
</script>
</body>
</html>
```

Product List



Task 5: Styled Modal Popup

Instructions:

- Use AI to generate a modal popup that opens when a button is clicked.
- Style the modal using CSS with a semi-transparent overlay.
- Include a close button that hides the modal.

Expected Output:

- A web page with a button labeled “Open Modal”.
- Clicking the button displays the modal popup with content.
- Modal can be closed by clicking the “X” button or overlay area.

Untitled ·
VELPULA SRIKEERTHANA

HTML

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-
6 width, initial-scale=1.0" />
7   <title>Styled Modal Popup</title>
8   <style>
9    /* Basic Page Styling */
10  body {
11    font-family: 'Poppins', sans-serif;
12    background-color: #f4f4f4;
13    display: flex;
14    justify-content: center;
```

CSS

```
1
```

Open Modal