AI ASSISTED CODING

LAB TEST-14.4

NAME: T. SHIRISHA

ENROLLNO:2403A52077

BATCH:04

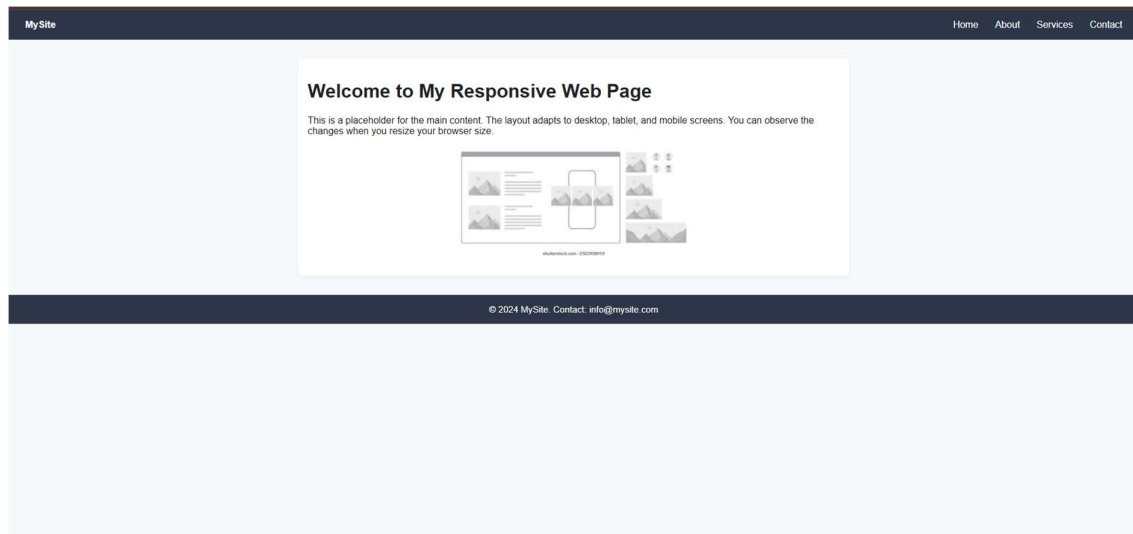**TASK-01:**

Create a Responsive Web Page Layout

PROMPT:

Generate a basic webpage layout which include header, main content ,footer also add CSS to make it more responsive in different screen sizes.

CODE:

```html
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Responsive Web Page Layout</title>
7     <style>
8       body {
9         margin: 0;
10        font-family: Arial, sans-serif;
11        background: #f8f9fa;
12        color: #222;
13      }
14      header {
15        background: #2d3748;
16        color: #fff;
17        padding: 1rem 2rem;
18        display: flex;
19        justify-content: space-between;
20        align-items: center;
21      }
22      nav a {
23        color: #fff;
24        text-decoration: none;
25        margin-left: 1.5rem;
26        font-weight: 500;
27        transition: color 0.2s;
28      }
29      nav a:hover {
30        color: #63b3ed;
31      }
32      main {
33        max-width: 900px;
34        margin: 2rem auto;
35        padding: 1rem;
36        background: #fff;
37        border-radius: 8px;
38        box-shadow: 0 2px 8px rgba(0,0,0,0.05);
39      }
40      .placeholder-img {
41        width: 100%;
42        max-width: 400px;
43        display: block;
44        margin: 1rem auto;
45        border-radius: 6px;
46      }
47      footer {
48        background: #2d3748;
```

```html
 2  <html lang="en">
 3  <head>
 7    <style>
47      footer {
48        background: ⬜#2d3748;
49        color: ⬜#fff;
50        text-align: center;
51        padding: 1rem 0;
52        font-size: 0.95rem;
53        margin-top: 2rem;
54      }
55      /* Responsive Styles */
56      @media (max-width: 900px) {
57        main {
58          max-width: 95%;
59        }
60      }
61      @media (max-width: 600px) {
62        header {
63          flex-direction: column;
64          align-items: flex-start;
65          padding: 1rem;
66        }
67        nav {
68          margin-top: 0.5rem;
69        }
70        nav a {
71          margin-left: 0;
72          margin-right: 1rem;
73          display: inline-block;
74        }
75        main {
76          padding: 0.5rem;
77        }
78        .placeholder-img {
79          max-width: 100%;
80        }
81      }
82    </style>
83  </head>
84  <body>
85    <header>
86      <div class="logo"><strong>MySite</strong></div>
87      <nav>
88        <a href="#">Home</a>
89        <a href="#">About</a>
90        <a href="#">Services</a>
91        <a href="#">Contact</a>
```

```html
 89        <a href="#">About</a>
 90        <a href="#">Services</a>
 91        <a href="#">Contact</a>
 92      </nav>
 93    </header>
 94    <main>
 95      <h1>Welcome to My Responsive Web Page</h1>
 96      <p>
 97        This is a placeholder for the main content. The layout adapts to desktop, tablet, and mobile screens.
 98        You can observe the changes when you resize your browser size.
 99      </p>
100      <img src="https://via.placeholder.com/400x200?text=Placeholder+Image" alt="Rectangular placeholder graphic displaying the words Placeholder Image centered on a light gray background, set within a clean
      and minimal web page layout" class="placeholder-img">
101    </main>
102    <footer>
103      &copy; 2024 MySite. Contact: info@mysite.com
104    </footer>
105  </body>
106  </html>
```

OUTPUT:

OBSERVATION:

The AI generated the webpage in an efficient way and it also included all the elements what we have instructed to it. It included the header section , body section and also the footer section.

TASK-02:

Interactive Button with Java script.

PROMPT:

Create a button on the webpage then use java script to display the popup alert message "Button clicked" when the button is clicked.
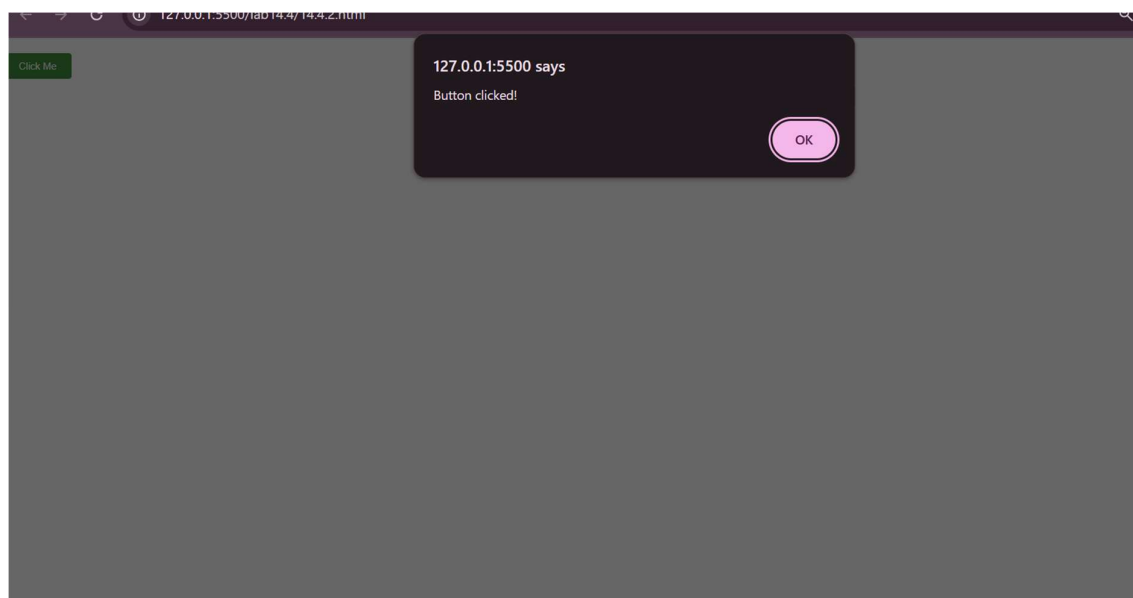
CODE:

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Button Alert Example</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }
    button {
      background-color: #4CAF50;
      color: white;
      padding: 10px 20px;
      border: none;
      border-radius: 4px;
      cursor: pointer;
    }
    button:hover {
      background-color: #45a049;
    }
    #alertButton {
      margin-top: 20px;
    }
    #alertButton:hover {
      background-color: #3e8e3c;
    }

  </style>
</head>
<body>
  <!-- Button that triggers the alert -->
  <button id="alertButton">Click Me</button>

  <script>
    // Get the button element by its ID
    const button = document.getElementById('alertButton');

    // Add a click event listener to the button
    button.addEventListener('click', function() {
      // Show an alert message when the button is clicked
      alert('Button clicked!');
    });
  </script>
</body>
</html>
```

OUTPUT:

OBSERVATION:

The code generated by the AI which shows the popup message when the button is clicked.

TASK-03:

Form with Validation.

PROMPT:

Generate a form which accepts the details like name, email ,message use java script to validate the correct format and display the error messages if the required field is empty after completion of filling the data then it should display the message "Form submitted Successfully" on clicking the submit button.

CODE:

```html
lab14.4 > <> 14.4.3.html > @ html > @ body > @ form#contactForm
 1  <!DOCTYPE html>
 2  <html lang="en">
 3  <head>
 4    <meta charset="UTF-8">
 5    <title>Contact Form</title>
 6    <style>
 7      body { font-family: Arial, sans-serif; margin: 40px; }
 8      .form-group { margin-bottom: 15px; }
 9      label { display: block; margin-bottom: 5px; }
10      input, textarea { width: 100%; padding: 8px; box-sizing: border-box; }
11      .error { color: ▮red; font-size: 0.95em; margin-top: 3px; }
12      .success { color: ▮green; font-size: 1em; margin-top: 10px; }
13      button { padding: 8px 16px; }
14    </style>
15  </head>
16  <body>
17    <h2>Contact Us</h2>
18    <form id="contactForm" novalidate>
19      <div class="form-group">
20        <label for="name">Name:</label>
21        <input type="text" id="name" name="name">
22        <div class="error" id="nameError"></div>
23      </div>
24      <div class="form-group">
25        <label for="email">Email:</label>
26        <input type="email" id="email" name="email">
27        <div class="error" id="emailError"></div>
28      </div>
29      <div class="form-group">
30        <label for="message">Message:</label>
31        <textarea id="message" name="message" rows="4"></textarea>
32        <div class="error" id="messageError"></div>
33      </div>
34      <button type="submit">Submit</button>
35      <div class="success" id="successMsg"></div>
36    </form>
37    <script>
38      const form = document.getElementById('contactForm');
39      const nameInput = document.getElementById('name');
40      const emailInput = document.getElementById('email');
41      const messageInput = document.getElementById('message');
42      const nameError = document.getElementById('nameError');
43      const emailError = document.getElementById('emailError');
44      const messageError = document.getElementById('messageError');
45      const successMsg = document.getElementById('successMsg');
46
47      function validateEmail(email) {
48        // Simple email regex
```

```
37    <script>
47      function validateEmail(email) {
49        return /^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email);
50      }
51
52      form.addEventListener('submit', function(e) {
53        e.preventDefault();
54        let valid = true;
55        nameError.textContent = '';
56        emailError.textContent = '';
57        messageError.textContent = '';
58        successMsg.textContent = '';
59
60        if (nameInput.value.trim() === '') {
61          nameError.textContent = 'Name is required.';
62          valid = false;
63        }
64        if (emailInput.value.trim() === '') {
65          emailError.textContent = 'Email is required.';
66          valid = false;
67        } else if (!validateEmail(emailInput.value.trim())) {
68          emailError.textContent = 'Please enter a valid email address.';
69          valid = false;
70        }
71        if (messageInput.value.trim() === '') {
72          messageError.textContent = 'Message is required.';
73          valid = false;
74        }
75
76        if (valid) {
77          successMsg.textContent = 'Form submitted successfully';
78          form.reset();
79        }
80      });
81    </script>
82  </body>
83  </html>
```

OUTPUT:

Contact Us

Name:

chandragiri ram charan

Email:

ramcharanch06@gmail.com

Message:

Hi everyone this is ramcharan from computer science and engineering.

Submit

OBSERVATION:

From the code given by the AI we observed that it accepts the details such as name , email and a message if the required field is not filled it displays the error message.

TASK-04:

Dynamic content Generation

PROMPT:

Generate the html code which consists of list of items and the buttons add item and remove item then by using java script make it dynamic to add the item when clicked on add item and remove the item if remove item is clicked.

CODE:

```html
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>Dynamic Product List</title>
6     <style>
7       body { font-family: Arial, sans-serif; margin: 30px; }
8       ul { margin-bottom: 20px; }
9       button { margin-right: 10px; }
10    </style>
11  </head>
12  <body>
13    <h2>Product List</h2>
14    <ul id="productList">
15      <li>Product 1</li>
16      <li>Product 2</li>
17      <li>Product 3</li>
18    </ul>
19    <button id="addBtn">Add Item</button>
20    <button id="removeBtn">Remove Item</button>
21
22    <script>
23      const productList = document.getElementById('productList');
24      const addBtn = document.getElementById('addBtn');
25      const removeBtn = document.getElementById('removeBtn');
26      let productCount = productList.children.length;
27
28      addBtn.onclick = function() {
29        productCount++;
30        const li = document.createElement('li');
31        li.textContent = `Product ${productCount}`;
32        productList.appendChild(li);
33      };
34
35      removeBtn.onclick = function() {
36        if (productList.children.length > 0) {
37          productList.removeChild(productList.lastElementChild);
38          productCount = productList.children.length;
39        }
40      };
41    </script>
42  </body>
43  </html>
```

OUTPUT:

Product List

- Product 1
- Product 2
- Product 3
- Product 4

[Add Item] [Remove Item]

OBSERVATION:

The AI generated the code which contains the list of product names and it has the buttons add item and remove item which works in the dynamic way.

TASK-05:

Styled Modal Popup.

PROMPT:

Generate a modal popup which opens when we click the button which is present on the web page use CSS to style it in an impressive way also include the hide button which hides the modal popup.

CODE:

```html
1    <!DOCTYPE html>
2    <html lang="en">
3    <head>
4      <meta charset="UTF-8">
5      <title>Modal Popup Example</title>
6      <style>
7        body {
8          font-family: Arial, sans-serif;
9        }
10       /* Overlay */
11       .modal-overlay {
12         display: none;
13         position: fixed;
14         top: 0; left: 0;
15         width: 100vw; height: 100vh;
16         background: □rgba(0,0,0,0.5);
17         z-index: 1000;
18         justify-content: center;
19         align-items: center;
20       }
21       /* Modal */
22       .modal {
23         background: ■#fff;
24         padding: 2em;
25         border-radius: 8px;
26         position: relative;
27         min-width: 300px;
28         box-shadow: 0 2px 10px □rgba(0,0,0,0.2);
29       }
30       /* Close button */
31       .close-btn {
32         position: absolute;
33         top: 10px; right: 10px;
34         background: none;
35         border: none;
36         font-size: 1.2em;
37         cursor: pointer;
38       }
39       .modal-overlay.show {
40         display: flex;
41         animation: fadeIn 0.3s ease-out;
42       }
43       @keyframes fadeIn {
44         from { opacity: 0; } to { opacity: 1; }
45       }
46     </style>
47   </head>
```

```html
2    <html lang="en">
48   <body>
49     <button id="openModalBtn">Open Modal</button>
50
51     <div class="modal-overlay" id="modalOverlay">
52       <div class="modal">
53         <button class="close-btn" id="closeModalBtn">&times;</button>
54         <h2>Modal Popup</h2>
55         <p>This is the modal content.</p>
56       </div>
57     </div>
58
59     <script>
60       const openBtn = document.getElementById('openModalBtn');
61       const overlay = document.getElementById('modalOverlay');
62       const closeBtn = document.getElementById('closeModalBtn');
63
64       function openModal() {
65         overlay.classList.add('show');
66       }
67
68       function closeModal() {
69         // We can add a fade-out animation here in the future if desired
70         overlay.classList.remove('show');
71       }
72
73       openBtn.onclick = openModal;
74
75       closeBtn.onclick = closeModal;
76
77       // Close modal when clicking on the overlay area
78       overlay.onclick = (e) => {
79         if (e.target === overlay) {
80           closeModal();
81         }
82       };
83
84     </script>
85   </body>
86   </html>
```
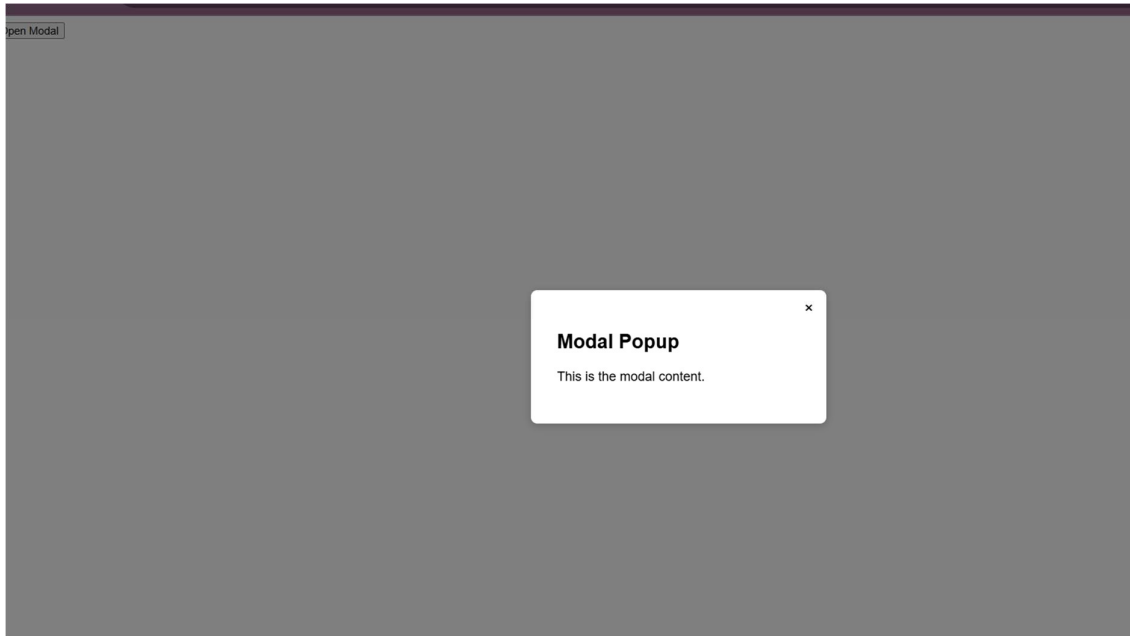
OUTPUT:

OBSERVATION:

The code generated by AI contains a button which displays a modal popup by clicking on it . the AI styled the modal popup in an impressive way.