**Assignment-14**

**NAME:**V. VIVEK VARDHAN

**HT NO:**2403A52097 **BATCH**: 24BTCAIAIB05

**Task-1:**

Prompt:

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:

o Header with navigation links

o Main section with placeholder text/images

o Footer with copyright or contact info

• Layout adapts correctly to desktop, tablet, and mobile screen sizes.

Code and Output:

**<!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>**

**<style>**

**/\* Global styles \*/**

**\* {**

**box-sizing: border-box;**

**margin: 0;**

**padding: 0;**

**font-family: Arial, sans-serif;**

**}**

**body {**

**display: flex;**

**flex-direction: column;**

**min-height: 100vh;**

**}**

**a {**

**text-decoration: none;**

**color: white;**

**margin: 0 10px;**

**}**

**/\* Header styles \*/**

**header {**

**background-color: #0077cc;**

**color: white;**

**padding: 20px;**

**}**

**header .nav {**

**display: flex;**

**justify-content: space-between;**

**align-items: center;**

**flex-wrap: wrap;**

**}**

**header .nav .logo {**

**font-size: 1.5em;**

**font-weight: bold;**

**}**

**header .nav nav {**

**display: flex;**

**gap: 15px;**

**}**

**/\* Main content styles \*/**

**main {**

**flex: 1;**

**padding: 20px;**

**display: flex;**

**flex-wrap: wrap;**

**gap: 20px;**

**justify-content: center;**

**}**

**main .card {**

**background-color: #f4f4f4;**

**padding: 20px;**

**width: 300px;**

**text-align: center;**

**border-radius: 8px;**

**box-shadow: 0 0 5px rgba(0,0,0,0.1);**

**}**

**main .card img {**

**max-width: 100%;**

**border-radius: 8px;**

**}**

**/\* Footer styles \*/**

**footer {**

**background-color: #333;**

**color: white;**

**text-align: center;**

**padding: 15px;**

**}**

**/\* Responsive styles \*/**

**@media (max-width: 768px) {**

**header .nav {**

**flex-direction: column;**

**align-items: flex-start;**

**}**

**header .nav nav {**

**flex-direction: column;**

**width: 100%;**

**margin-top: 10px;**

**}**

**main {**

**flex-direction: column;**

**align-items: center;**

**}**

**}**

**@media (max-width: 480px) {**

**header {**

**padding: 15px;**

**}**

**main .card {**

**width: 90%;**

**}**

**}**

**</style>**

**</head>**

**<body>**

**<header>**

**<div class="nav">**

**<div class="logo">MyWebsite</div>**

**<nav>**

**<a href="#">Home</a>**

**<a href="#">About</a>**

**<a href="#">Services</a>**

**<a href="#">Contact</a>**

**</nav>**

**</div>**

**</header>**

**<main>**

**<div class="card">**

**<img src="https://images.unsplash.com/photo-1506744038136-46273834b3fb?auto=format&fit=crop&w=300&q=80" alt="Nature">**

**<h3>Nature</h3>**

**<p>Beautiful scenery of mountains and forests.</p>**

**</div>**

**<div class="card">**

**<img src="https://images.unsplash.com/photo-1494526585095-c41746248156?auto=format&fit=crop&w=300&q=80" alt="City">**

**<h3>City</h3>**

**<p>Modern city landscape with skyscrapers and streets.</p>**

**</div>**

**<div class="card">**

**<img src="https://images.unsplash.com/photo-1518779578993-ec3579fee39f?auto=format&fit=crop&w=300&q=80" alt="Technology">**

**<h3>Technology</h3>**

**<p>Futuristic technology and innovation concept.</p>**

**</div>**

**</main>**

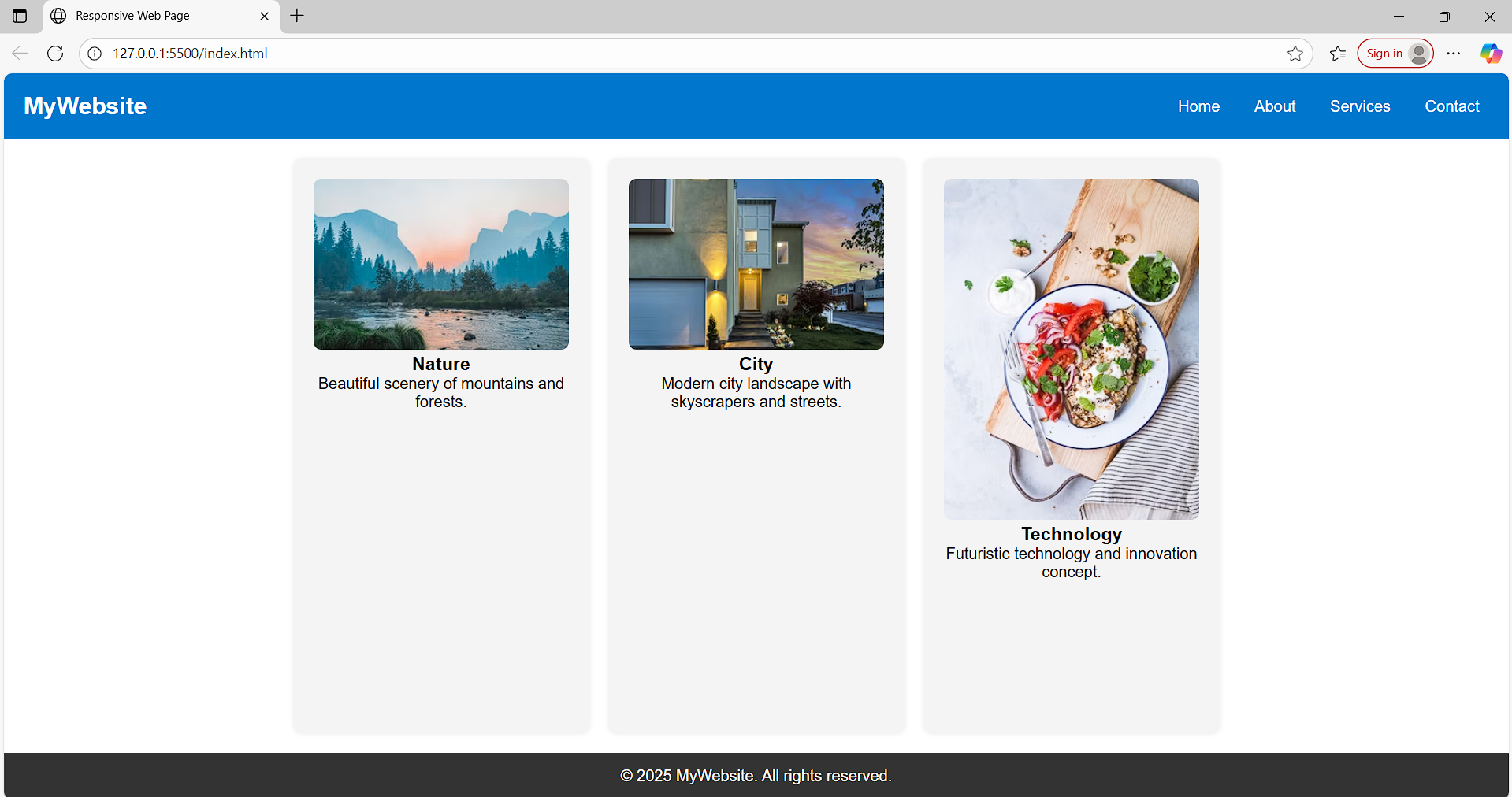
**<footer>**

**&copy; 2025 MyWebsite. All rights reserved.**

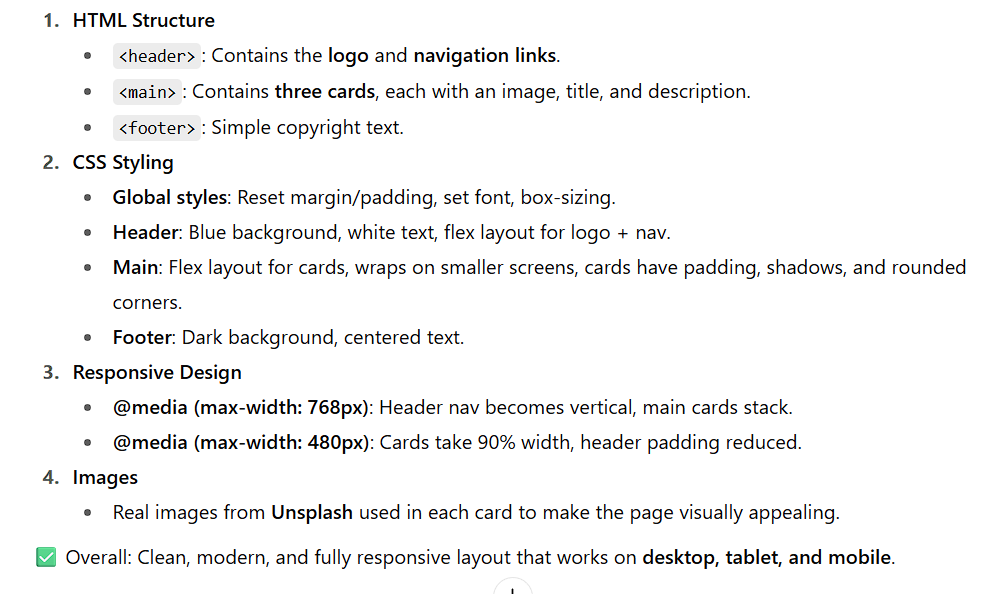
**</footer>**

**</body>**

**</html>**



Explanation:



**Task 2:**

Prompt:

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!”.

Code and Output:

**<!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>**

**/\* Center the button on the page \*/**

**body {**

**display: flex;**

**justify-content: center;**

**align-items: center;**

**height: 100vh;**

**font-family: Arial, sans-serif;**

**background-color: #f0f0f0;**

**}**

**/\* Button styling \*/**

**#myButton {**

**padding: 15px 30px;**

**font-size: 18px;**

**border: none;**

**border-radius: 8px;**

**background-color: #0077cc;**

**color: white;**

**cursor: pointer;**

**transition: background-color 0.3s;**

**}**

**/\* Hover effect \*/**

**#myButton:hover {**

**background-color: #005fa3;**

**}**

**</style>**

**</head>**

**<body>**

**<!-- Button element -->**

**<button id="myButton">Click Me!</button>**

**<script>**

**// Select the button using its ID**

**const button = document.getElementById('myButton');**

**// Add a click event listener to the button**

**button.addEventListener('click', function() {**

**// Display an alert message when the button is clicked**

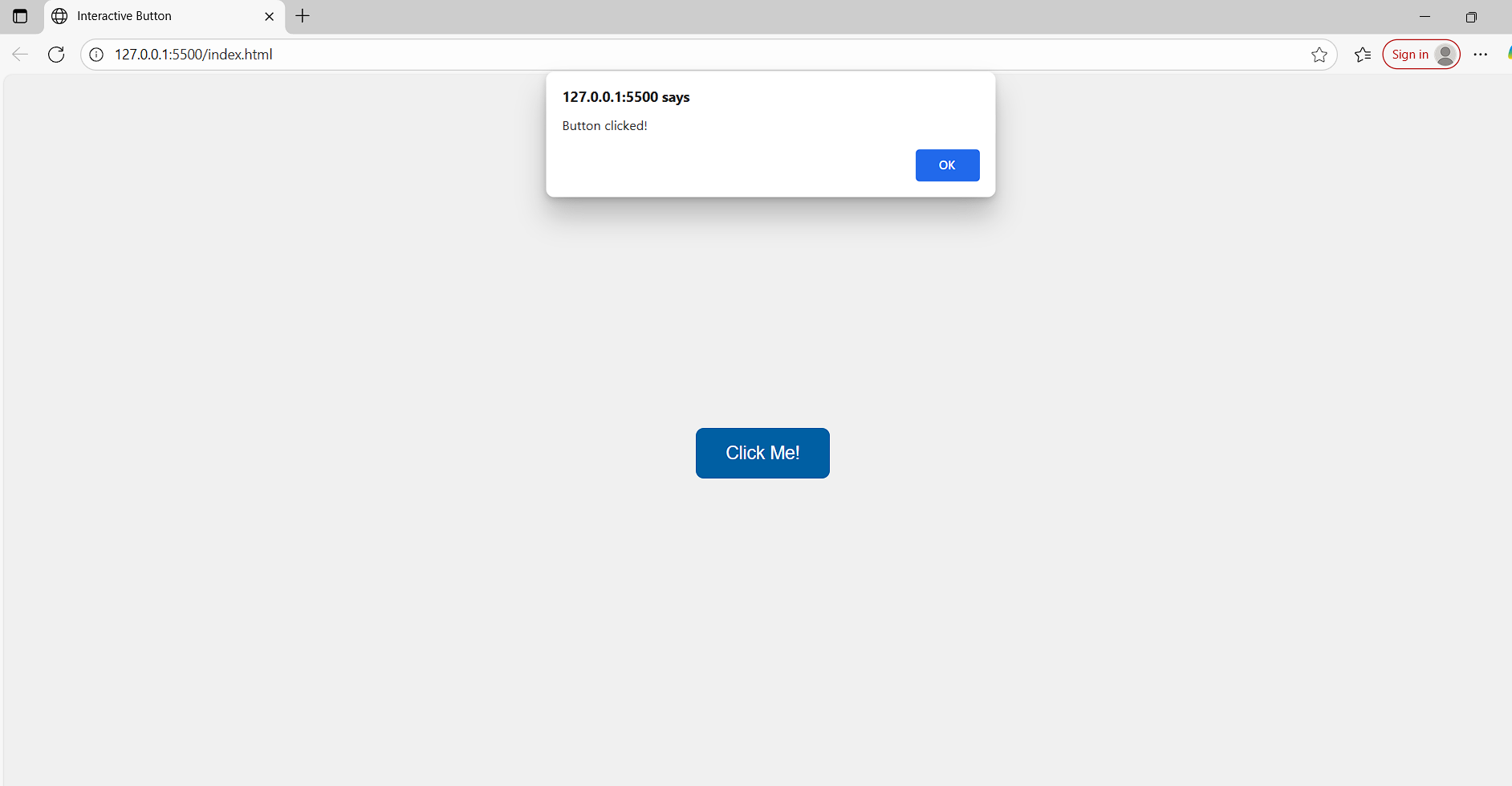
**alert('Button clicked!');**

**});**

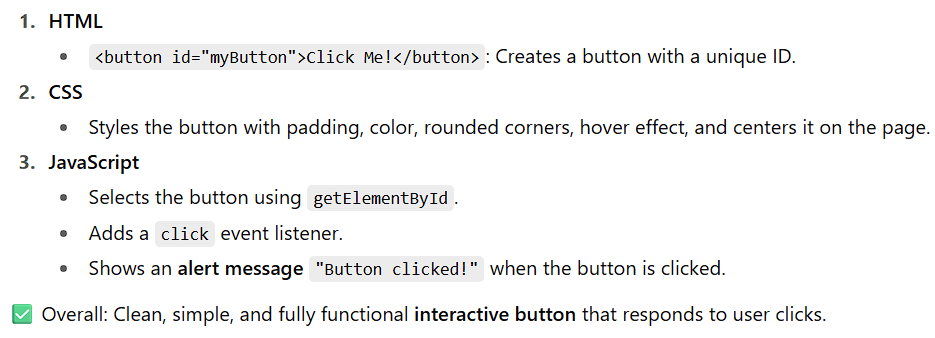
**</script>**

**</body>**

**</html>**

****

Explanation:



**Task 3**:

Prompt:

**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

Code and Output:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Contact Form with Validation</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #f4f4f4;

            display: flex;

            justify-content: center;

            align-items: center;

            height: 100vh;

        }

        form {

            background-color: white;

            padding: 30px;

            border-radius: 8px;

            box-shadow: 0 0 10px rgba(0,0,0,0.1);

            width: 100%;

            max-width: 400px;

        }

        h2 {

            text-align: center;

            margin-bottom: 20px;

        }

        label {

            display: block;

            margin-top: 15px;

            font-weight: bold;

        }

        input, textarea {

            width: 100%;

            padding: 10px;

            margin-top: 5px;

            border-radius: 5px;

            border: 1px solid #ccc;

            font-size: 16px;

        }

        textarea {

            resize: vertical;

        }

        .error {

            color: red;

            font-size: 14px;

            margin-top: 5px;

            display: none; /\* Hidden by default \*/

        }

        button {

            margin-top: 20px;

            padding: 12px;

            width: 100%;

            font-size: 16px;

            border: none;

            border-radius: 8px;

            background-color: #0077cc;

            color: white;

            cursor: pointer;

            transition: background-color 0.3s;

        }

        button:hover {

            background-color: #005fa3;

        }

    </style>

</head>

<body>

    <form id="contactForm">

        <h2>Contact Us</h2>

        <label for="name">Name</label>

        <input type="text" id="name" name="name">

        <div class="error" id="nameError">Please enter your name</div>

        <label for="email">Email</label>

        <input type="text" id="email" name="email">

        <div class="error" id="emailError">Please enter a valid email</div>

        <label for="message">Message</label>

        <textarea id="message" name="message" rows="5"></textarea>

        <div class="error" id="messageError">Please enter your message</div>

        <button type="submit">Submit</button>

    </form>

    <script>

        // Select the form and input fields

        const form = document.getElementById('contactForm');

        const nameInput = document.getElementById('name');

        const emailInput = document.getElementById('email');

        const messageInput = document.getElementById('message');

        // Select error divs

        const nameError = document.getElementById('nameError');

        const emailError = document.getElementById('emailError');

const messageError = document.getElementById('messageError');

  // Function to validate email format

        function isValidEmail(email) {

            const pattern = /^[^ ]+@[^ ]+\.[a-z]{2,3}$/;

            return pattern.test(email);

        }

// Form submit event

        form.addEventListener('submit', function(event) {

            event.preventDefault(); // Prevent form submission

            let valid = true;

            // Name validation

            if (nameInput.value.trim() === '') {

                nameError.style.display = 'block';

                valid = false;

            } else {

                nameError.style.display = 'none';

            }

// Email validation

            if (emailInput.value.trim() === '' || !isValidEmail(emailInput.value.trim())) {

                emailError.style.display = 'block';

                valid = false;

            } else {

                emailError.style.display = 'none';

            }

  // Message validation

            if (messageInput.value.trim() === '') {

                messageError.style.display = 'block';

                valid = false;

            } else {

                messageError.style.display = 'none';

            }

            // If all fields are valid

            if (valid) {

                alert('Form submitted successfully!');

                form.reset(); // Clear the form

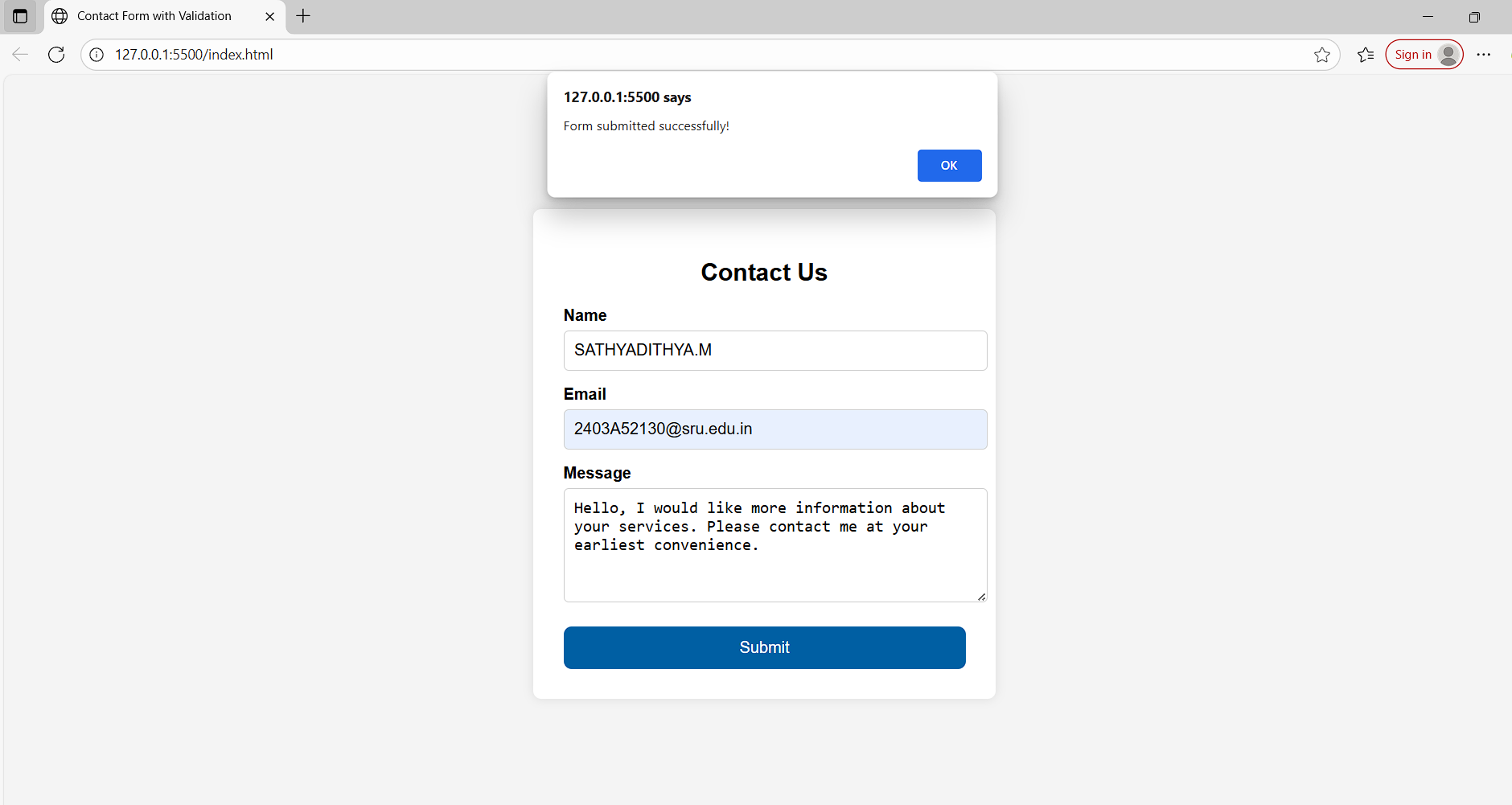
            }

        });

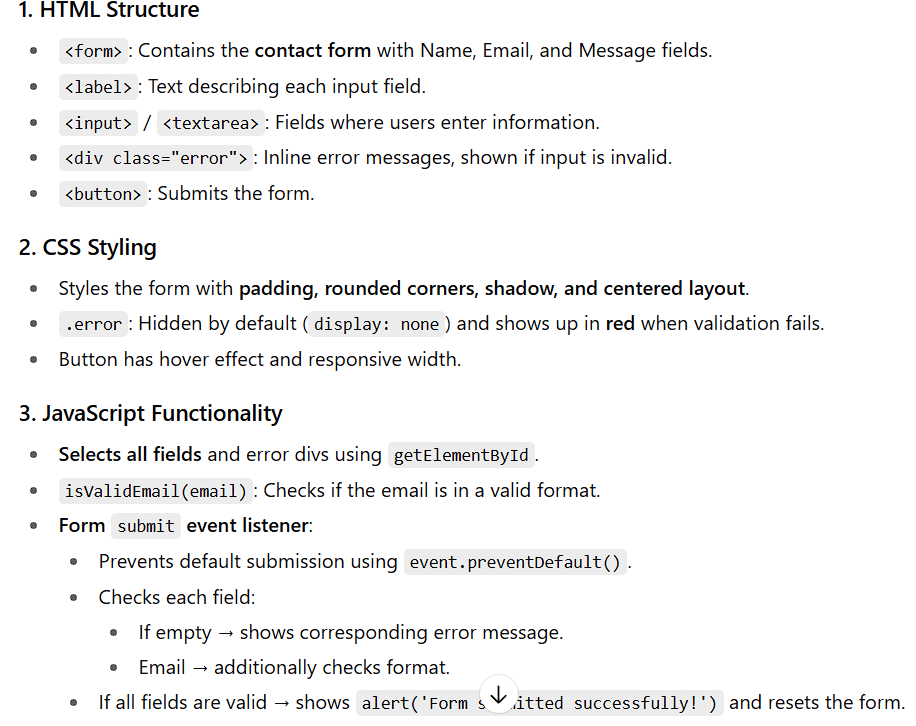
    </script>

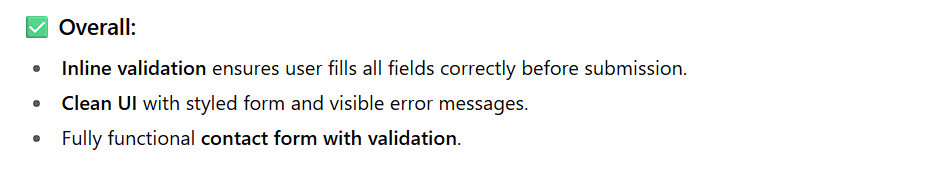
</body>

</html>



Explanation:





**Task 4:**

Prompt:

**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.

Code & Output:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Dynamic List</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #f4f4f4;

            display: flex;

            flex-direction: column;

            align-items: center;

            padding: 50px;

        }

        h2 {

            margin-bottom: 20px;

        }

        ul {

            list-style: none;

            padding: 0;

            width: 300px;

            margin-bottom: 20px;

        }

        li {

            background-color: #fff;

            margin-bottom: 10px;

            padding: 10px;

            border-radius: 6px;

            box-shadow: 0 0 5px rgba(0,0,0,0.1);

        }

        button {

            padding: 10px 20px;

            margin: 5px;

            font-size: 16px;

            border: none;

            border-radius: 6px;

            background-color: #0077cc;

            color: white;

            cursor: pointer;

            transition: background-color 0.3s;

        }

        button:hover {

            background-color: #005fa3;

        }

    </style>

</head>

<body>

    <h2>Product List</h2>

    <ul id="productList">

        <li>Product 1</li>

        <li>Product 2</li>

        <li>Product 3</li>

    </ul>

    <button id="addBtn">Add Item</button>

    <button id="removeBtn">Remove Item</button>

    <script>

        // Select the list and buttons

        const productList = document.getElementById('productList');

        const addBtn = document.getElementById('addBtn');

        const removeBtn = document.getElementById('removeBtn');

        let itemCount = 3; // Start with 3 items

        // Add Item button click event

        addBtn.addEventListener('click', function() {

            itemCount++;

            const li = document.createElement('li'); // Create new list item

            li.textContent = `Product ${itemCount}`;

            productList.appendChild(li); // Add to the list

        });

        // Remove Item button click event

        removeBtn.addEventListener('click', function() {

            if (productList.lastElementChild) {

                productList.removeChild(productList.lastElementChild); // Remove last item

                itemCount--;

            } else {

                alert('No more items to remove!');

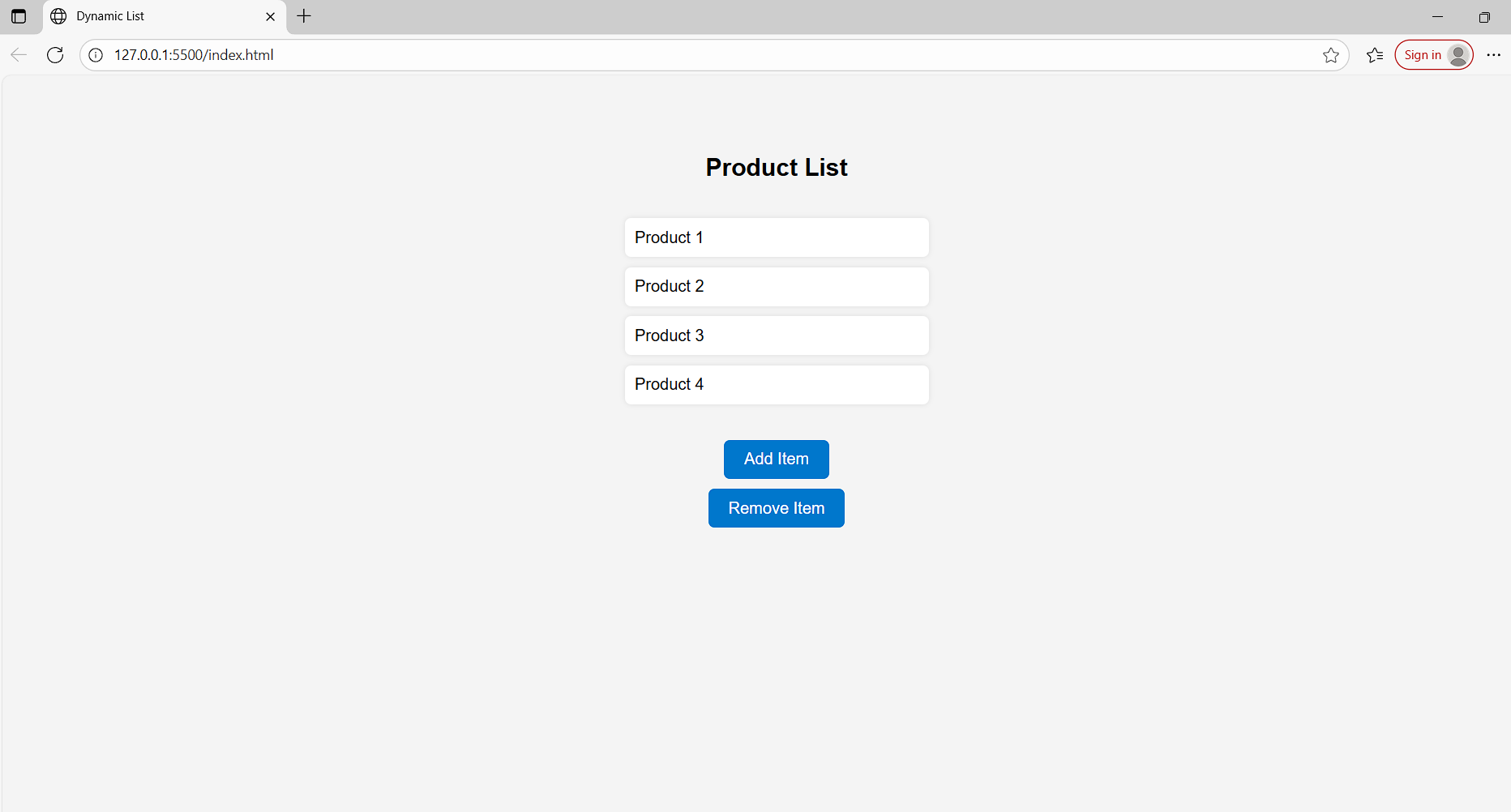
            }

        });

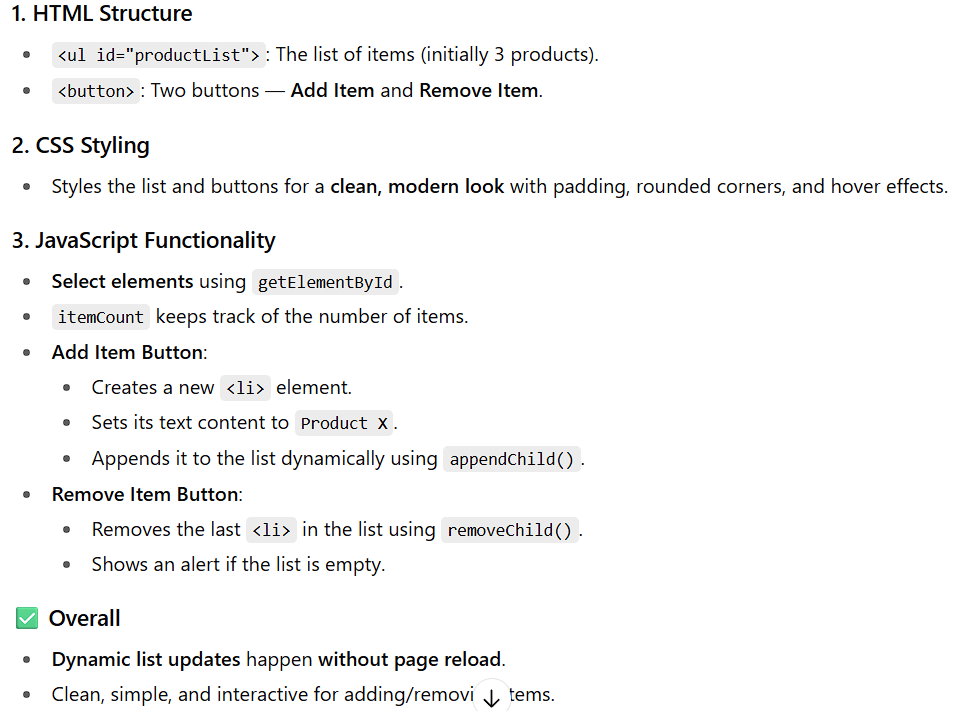
    </script>

</body>

</html>



Explanation:



**Task 5:**

Prompt:

Generate a complete HTML, CSS, and JavaScript code for a styled modal popup. The modal should open when a button labeled ‘Open Modal’ is clicked, display a semi-transparent overlay, and include a close (×) button. The modal should close when clicking the close button or anywhere outside the modal. Ensure the design is clean, centered, and responsive

Code & Output:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Styled Modal Popup</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

/\* Open Modal Button \*/

#openModalBtn {

padding: 15px 30px;

font-size: 16px;

border: none;

border-radius: 8px;

background-color: #0077cc;

color: white;

cursor: pointer;

transition: background-color 0.3s;

}

#openModalBtn:hover {

background-color: #005fa3;

}

/\* Modal overlay \*/

.modal {

display: none; /\* Hidden by default \*/

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(0,0,0,0.5); /\* Semi-transparent overlay \*/

justify-content: center;

align-items: center;

}

/\* Modal content box \*/

.modal-content {

background-color: white;

padding: 30px;

border-radius: 8px;

width: 90%;

max-width: 500px;

position: relative;

box-shadow: 0 5px 15px rgba(0,0,0,0.3);

text-align: center;

}

/\* Close button \*/

.close-btn {

position: absolute;

top: 10px;

right: 15px;

font-size: 20px;

font-weight: bold;

color: #333;

cursor: pointer;

}

.close-btn:hover {

color: #0077cc;

}

</style>

</head>

<body>

<!-- Open Modal Button -->

<button id="openModalBtn">Open Modal</button>

<!-- Modal Structure -->

<div class="modal" id="myModal">

<div class="modal-content">

<span class="close-btn" id="closeModal">&times;</span>

<h2>Modal Popup</h2>

<p>This is a styled modal popup. You can put any content here!</p>

</div>

</div>

<script>

// Select elements

const openBtn = document.getElementById('openModalBtn');

const modal = document.getElementById('myModal');

const closeBtn = document.getElementById('closeModal');

// Open modal on button click

openBtn.addEventListener('click', () => {

modal.style.display = 'flex';

});

// Close modal on 'X' click

closeBtn.addEventListener('click', () => {

modal.style.display = 'none';

});

// Close modal on overlay click

window.addEventListener('click', (e) => {

if (e.target === modal) {

modal.style.display = 'none';

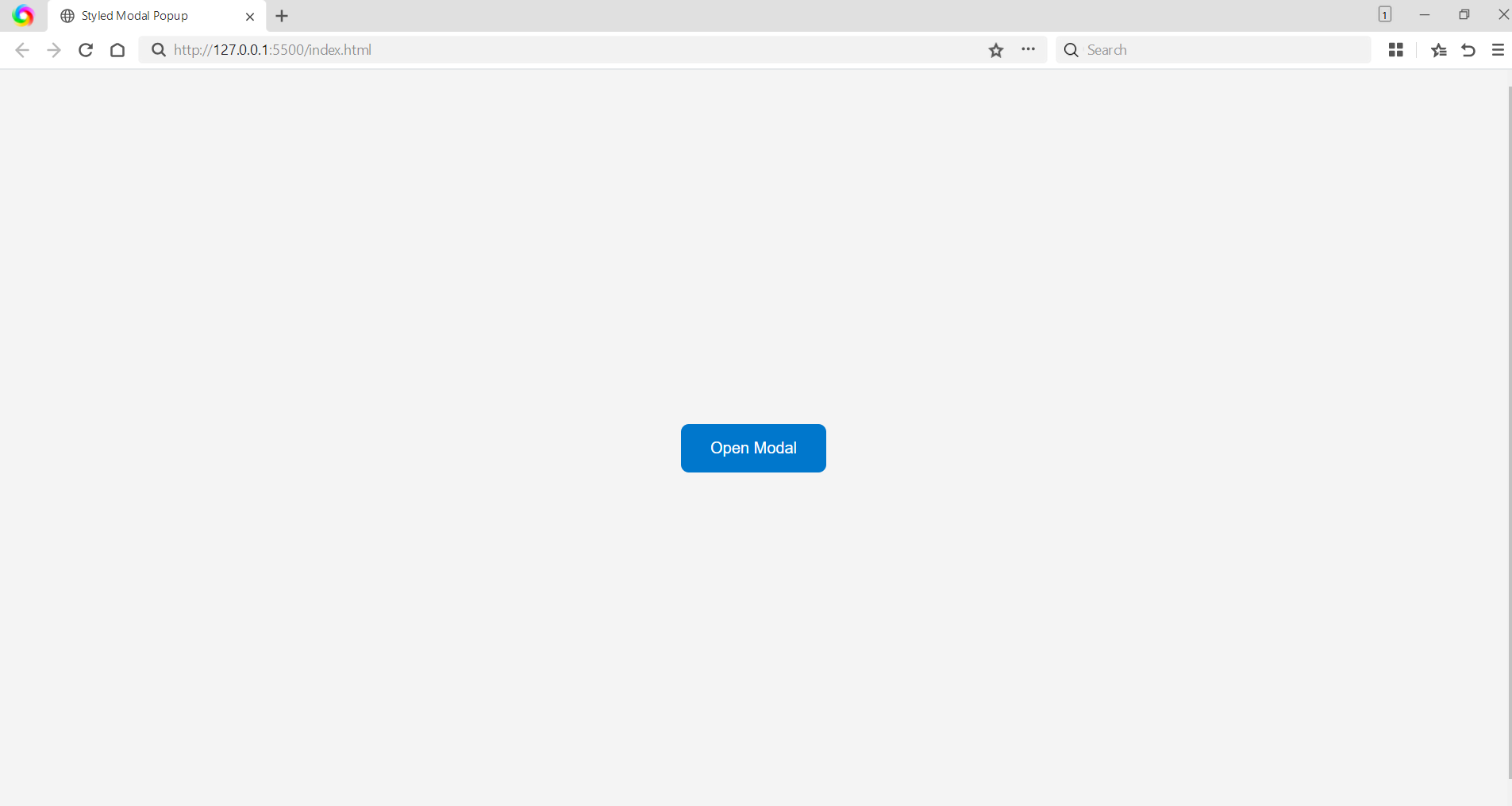
}

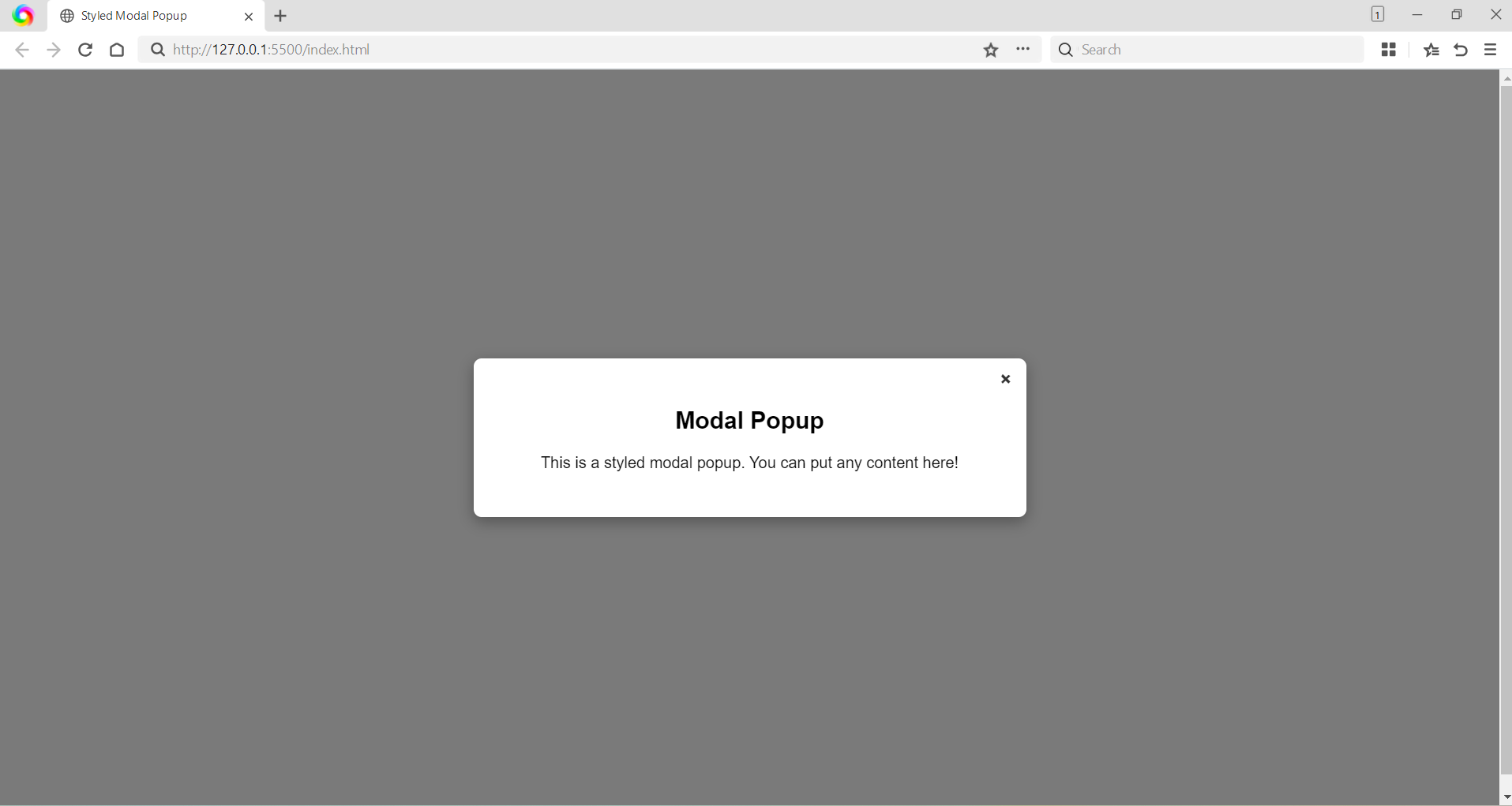
});

</script>

</body>

</html>





Explanation:

