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.

CODE:

HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Savika's Portfolio</title>
  <link rel="stylesheet" href="styles.css" />
</head>

<body>
  <header>
    <img src="savika.jpg" alt="Savika's Profile Photo" />
    <h1>Savika Chidara</h1>
    <nav>
      <ul>
        <li><a href="#about">About Me</a></li>
        <li><a href="#skills">Skills</a></li>
        <li><a href="#projects">Projects</a></li>
        <li><a href="#contact">Contact</a></li>
      </ul>
    </nav>
  </header>

  <main>
    <section id="about">
      <h2>About Me</h2>
      <p>Hello! I am Savika, a passionate web developer.<br>
        I specialize in creating websites with a background in computer
science.<br>
```

```
        I bring aesthetic appeal to my work and enjoy learning new
technologies in web development.<br>
        I am always exploring new ways to improve my work.</p>
    </section>

    <section id="skills">
      <h2>Technical Skills</h2>
      <ol>
        <li>HTML</li>
        <li>C Language</li>
        <li>Python</li>
        <li>JavaScript</li>
        <li>SQL</li>
      </ol>
    </section>

    <section id="projects">
      <h2>Projects</h2>
      <ol>
        <li>
          <h3>Portfolio</h3>
          <p>A personal portfolio built with HTML5 to showcase my skills,
projects, and contact information.</p>
        </li>
        <li>
          <h3>Task Manager App</h3>
          <p>A full-stack task management application using React and
Node.js.</p>
        </li>
      </ol>
    </section>

    <section id="contact">
      <h2>Contact</h2>
      <address>
        <p>Email: <a href="mailto:xyz@gmail.com">xyz@gmail.com</a></p>
        <p>Phone: 87121xxxxx</p>
        <p>LinkedIn: <a href="https://www.linkedin.com/in/savika-
chidara">savika-chidara</a></p>
        <p>GitHub: <a href="https://github.com/savika-chidara">savika-
chidara</a></p>
      </address>
    </section>
  </main>
  <footer>
    <p>&copy; 2025 Savika Chidara. All rights reserved.</p>
  </footer>

  <script src="script.js"></script>
</body>
</html>
```

CSS CODE:

```
/* Basic Reset */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: Arial, sans-serif;
  line-height: 1.6;
  background-color: #f4f4f4;
  color: #333;
  padding: 20px;
  max-width: 900px;
  margin: auto;
}

/* Image styling */
img {
  max-width: 150px;
  border-radius: 50%;
  display: block;
  margin: 20px auto;
}

/* Header */
header {
  text-align: center;
  margin-bottom: 30px;
}

header h1 {
  font-size: 2.8rem;
  margin-bottom: 10px;
  color: #222;
}
```

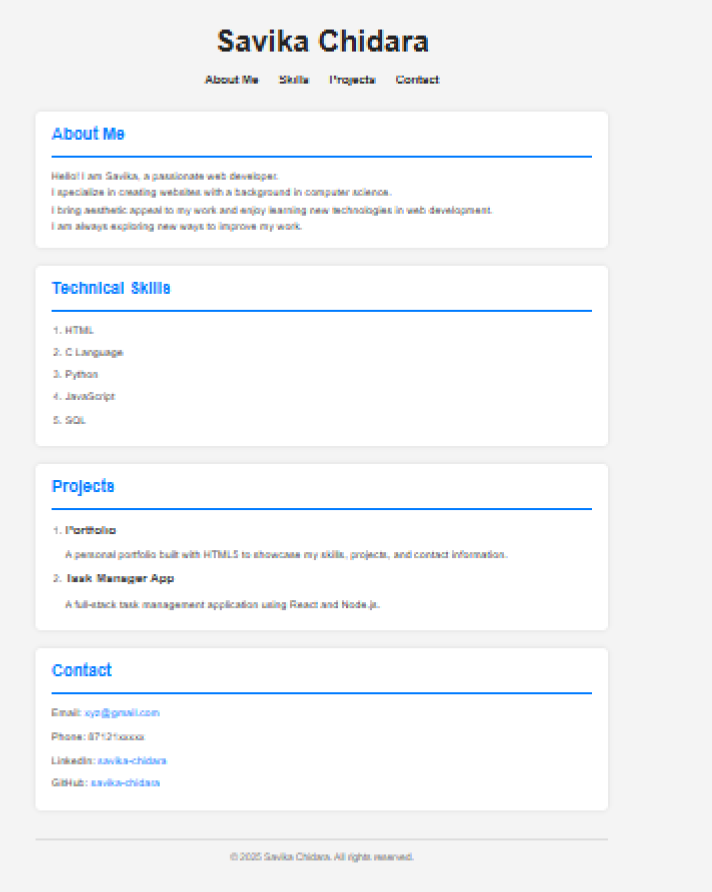
```
/* Navigation */
nav ul {
  list-style: none;
  display: flex;
  justify-content: center;
  gap: 30px;
}

nav ul li a {
  text-decoration: none;
  color: #333;
  font-weight: 600;
  font-size: 1.1rem;
  transition: color 0.3s ease;
}

nav ul li a:hover {
  color: #007BFF;
}

/* Sections */
section {
  background: white;
  padding: 20px 25px;
  margin-bottom: 25px;
  border-radius: 8px;
  box-shadow: 0 0 8px rgba(0,0,0,0.1); }

section h2 {
  margin-bottom: 15px;
  border-bottom: 2px solid #007BFF;
  padding-bottom: 8px;
  color: #007BFF;
}

section p, section ol {
  font-size: 1rem;
  margin-top: 10px;
}

section ol {
  padding-left: 20px;
}
```

```css
section ol li {
  margin-bottom: 8px;
}

/* Project titles */
section ol li h3 {
  margin-bottom: 5px;
  color: #444;
}

/* Contact links */
address p {
  margin-bottom: 8px;
  font-style: normal;
}

address a {
  color: #007BFF;
  text-decoration: none;
}

address a:hover {
  text-decoration: underline;
}

/* Footer */
footer {
  text-align: center;
  padding: 15px 0;
  font-size: 0.9rem;
  color: #777;
  border-top: 1px solid #ddd;
  margin-top: 40px;
}

/* Responsive Design */
@media (max-width: 600px) {
  nav ul {
    flex-direction: column;
    gap: 15px;
  }
  img {
    max-width: 120px;
  }
}
```

JS CODE:

```
// Select all nav links
const navLinks = document.querySelectorAll('nav ul li a');

navLinks.forEach(link => {
  link.addEventListener('click', (e) => {
    e.preventDefault();
    const targetID = link.getAttribute('href').slice(1);
    const targetSection = document.getElementById(targetID);
    if (targetSection) {
      targetSection.scrollIntoView({ behavior: 'smooth' });
    }
  });
});
```

OUTPUT:



Savika Chidara

About Me

Hello! I am Savika, a passionate web developer.
I specialize in creating websites with a background in computer science.
I bring aesthetic appeal to my work and enjoy learning new technologies in web development.
I am always exploring new ways to improve my work.

Technical Skills

1. HTML
2. C Language
3. Python
4. JavaScript
5. SQL

Projects

1. **Portfolio**
   A personal portfolio built with HTML5 to showcase my skills, projects, and contact information.
2. **Task Manager App**
   A full-stack task management application using React and Node.js.

Contact

Email: xyz@gmail.com
Phone: 87121xxxxx
LinkedIn: savika-chidara
GitHub: savika-chidara

© 2025 Savika Chidara. All rights reserved.

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.

Code:

HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Interactive Button</title>
  <!-- Link to external CSS file -->
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <!-- Button Element -->
  <button id="myButton">Click Me!</button>

  <!-- Link to external JavaScript file -->
  <script src="script.js"></script>
</body>
</html>
```

CSS CODE:

```
/* Styling for the button */
button {
  padding: 10px 20px;
  font-size: 16px;
  background-color: #3498db;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

/* Add hover effect */
```

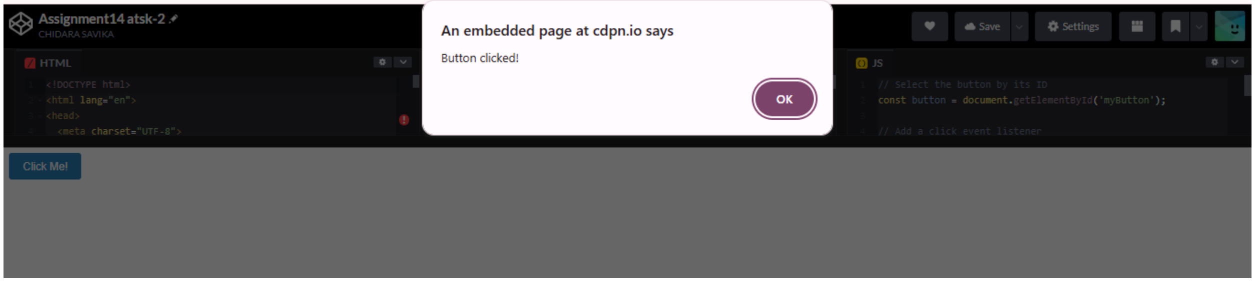
```
button:hover {
  background-color: #2980b9;
}
```

JS CODE:

```
// Select the button by its ID
const button = document.getElementById('myButton');

// Add a click event listener
button.addEventListener('click', function() {
  // Show an alert when the button is clicked
  alert('Button clicked!');
});
```

OUTPUT:

When the button was clicked it shows like an alert box.



Assignment14 atsk-2

CHIDARA SAVIKA

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```
// Select the button by its ID
const button = document.getElementById('myButton');

// Add a click event listener
```

An embedded page at cdpn.io says

Button clicked!

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

CODE:

HTML CODE:

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contact Form with Validation</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <div class="container">
    <h2>Contact Us</h2>

    <!-- Success Message -->
    <div id="successMessage" class="success-message"></div>

    <!-- Contact Form -->
    <form id="contactForm" novalidate>
      <div class="form-group">
        <label for="name">Name:</label>
        <input type="text" id="name" />
        <span class="error-message" id="nameError"></span>
      </div>

      <div class="form-group">
        <label for="email">Email:</label>
        <input type="text" id="email" />
        <span class="error-message" id="emailError"></span>
      </div>

      <div class="form-group">
        <label for="message">Message:</label>
        <textarea id="message"></textarea>
        <span class="error-message" id="messageError"></span>
      </div>

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

  <script src="script.js"></script>
</body>
</html>
```

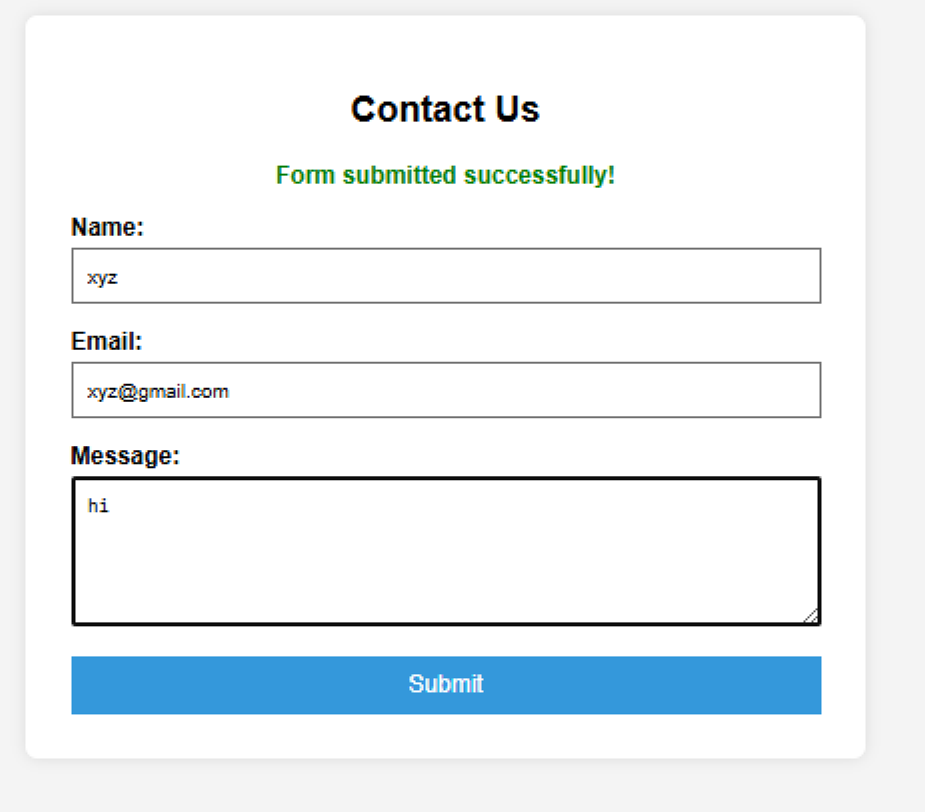
CSS CODE:

```
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  padding: 20px;
}

.container {
  max-width: 500px;
  margin: auto;
  background: white;
  padding: 30px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0,0,0,0.1);
}

h2 {
  text-align: center;
}

.form-group {
  margin-bottom: 15px;
}

label {
  display: block;
  font-weight: bold;
}

input, textarea {
  width: 100%;
  padding: 10px;
  margin-top: 5px;
  box-sizing: border-box;
}

textarea {
  resize: vertical;
  height: 100px;
}

button {
  padding: 10px 20px;
  background-color: #3498db;
  color: white;
  border: none;
  cursor: pointer;
  width: 100%;
  font-size: 16px;
}

button:hover {
  background-color: #2980b9;
}

.error-message {
  color: red;
  font-size: 14px;
}

.success-message {
  color: green;
  font-weight: bold;
  text-align: center;
  margin-bottom: 15px;
}
```

JS CODE:

```
// Wait until the DOM is fully loaded
document.addEventListener('DOMContentLoaded', function () {
  const form = document.getElementById('contactForm');

  form.addEventListener('submit', function (event) {
    event.preventDefault(); // Prevent actual form submission

    // Clear previous messages
    document.getElementById('successMessage').textContent = '';
    document.getElementById('nameError').textContent = '';
    document.getElementById('emailError').textContent = '';
    document.getElementById('messageError').textContent = '';

    // Get input values
    const name = document.getElementById('name').value.trim();
    const email = document.getElementById('email').value.trim();
    const message = document.getElementById('message').value.trim();

    let isValid = true;

    // Validate name
    if (name === '') {
      document.getElementById('nameError').textContent = 'Name is required.';
      isValid = false;
    }

    // Validate email
    if (email === '') {
      document.getElementById('emailError').textContent = 'Email is required.';
      isValid = false;
    } else if (!isValidEmail(email)) {
      document.getElementById('emailError').textContent = 'Please enter a valid email.';
      isValid = false;
    }

    // Validate message
    if (message === '') {
      document.getElementById('messageError').textContent = 'Message is required.';
      isValid = false;
    }

    // If all fields are valid
    if (isValid) {
      document.getElementById('successMessage').textContent = 'Form submitted successfully!';
      form.reset(); // Clear form fields
    }
  });

  // Email format validation using regex
  function isValidEmail(email) {
    const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    return emailPattern.test(email);
  }
});
```

OUTPUT:



Contact Us

Form submitted successfully!

Name:

xyz

Email:

xyz@gmail.com

Message:

hi

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.

CODE:

HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Dynamic List</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
```

```
  <div class="container">
    <h2>Product List</h2>

    <!-- Buttons -->
    <div class="buttons">
      <button id="addItemBtn">Add Item</button>
      <button id="removeItemBtn">Remove Item</button>
    </div>

    <!-- Item List -->
    <ul id="itemList">
      <li>Product 1</li>
      <li>Product 2</li>
      <li>Product 3</li>
    </ul>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

CSS CODE:

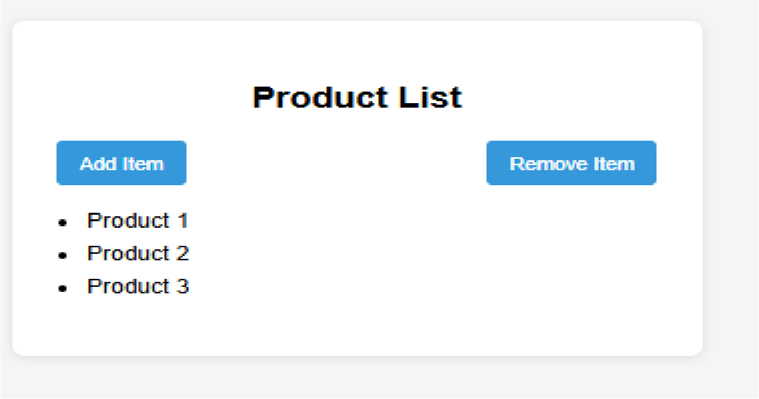
```
body {
  font-family: Arial, sans-serif;
  background: #f5f5f5;
  padding: 40px;
}

.container {
  max-width: 400px;
  margin: auto;
  background: white;
  padding: 30px;
  border-radius: 8px;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
}

h2 {
  text-align: center;
}

ul {
  list-style-type: disc;
  padding-left: 20px;
}
```

```
li {
  margin: 8px 0;
  font-size: 16px;
}

.buttons {
  display: flex;
  justify-content: space-between;
  margin-bottom: 20px;
}

button {
  padding: 10px 15px;
  font-size: 14px;
  background-color: #3498db;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

button:hover {
  background-color: #2980b9;
}
```

JS CODE:

```
// Wait until the DOM is fully loaded
document.addEventListener('DOMContentLoaded', function () {
  const itemList = document.getElementById('itemList');
  const addItemBtn = document.getElementById('addItemBtn');
  const removeItemBtn = document.getElementById('removeItemBtn');

  let itemCount = itemList.children.length; // Start with existing items

  // Add item when button is clicked
  addItemBtn.addEventListener('click', function () {
    itemCount++;
    const newItem = document.createElement('li');
    newItem.textContent = `Product ${itemCount}`;
    itemList.appendChild(newItem);
  });

  // Remove last item when button is clicked
  removeItemBtn.addEventListener('click', function () {
    if (itemList.children.length > 0) {
      itemList.removeChild(itemList.lastElementChild);
      itemCount--;
    }
  });
});
```

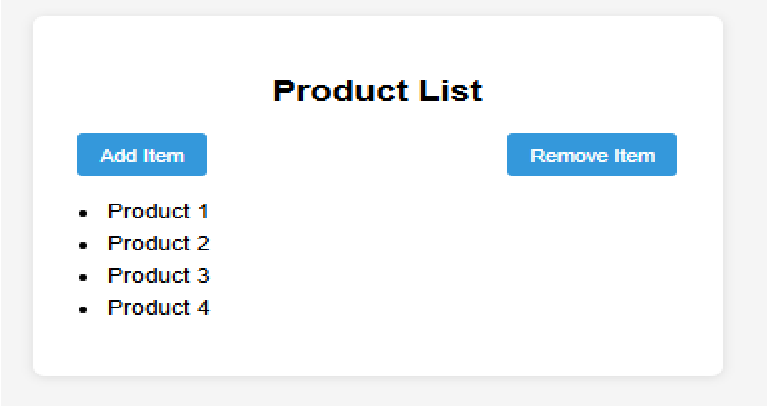
OUTPUT:



Product List

- Product 1
- Product 2
- Product 3

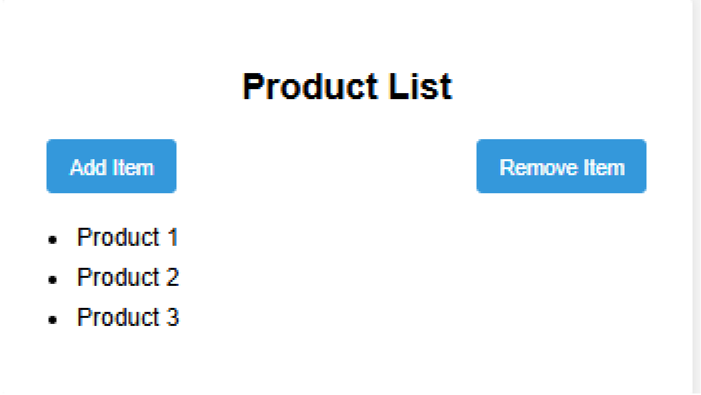
When you click on add item it will added to list.



Product List

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

When you click on remove item it will remove from list.



Product List

- Product 1
- Product 2
- Product 3

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.

CODE:

HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Styled Modal Popup</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <!-- Button to open the modal -->
  <button id="openModalBtn">Open Modal</button>

  <!-- Modal overlay and content -->
  <div id="modalOverlay" class="modal-overlay">
    <div class="modal-box">
      <span id="closeModalBtn" class="close-button">&times;</span>
      <h2>Modal Title</h2>
      <p>This is some content inside the modal popup.</p>
    </div>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

CSS CODE:

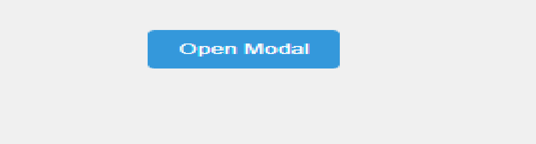
```
/* General page styles */
body {
  font-family: Arial, sans-serif;
  background: #f0f0f0;
  padding: 50px;
```

```css
  text-align: center;
}

/* Button styles */
button {
  padding: 12px 20px;
  font-size: 16px;
  background-color: #3498db;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #2980b9;
}
/* Modal overlay */
.modal-overlay {
  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;
  z-index: 1000;
}
/* Modal box */
.modal-box {
  background: white;
  padding: 30px;
  border-radius: 8px;
  width: 400px;
  max-width: 90%;
  position: relative;
  box-shadow: 0 4px 12px rgba(0,0,0,0.3);
}
/* Close button (X) */
.close-button {
  position: absolute;
  top: 10px;
  right: 15px;
  font-size: 24px;
  color: #888;
  cursor: pointer;
}
.close-button:hover {
  color: #333;
}
```

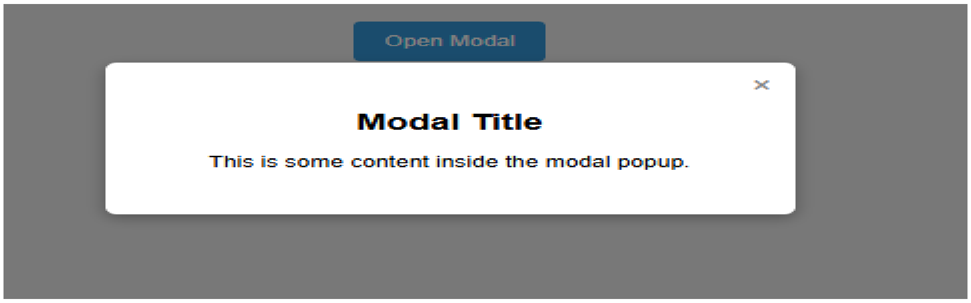
JS CODE:

```
document.addEventListener('DOMContentLoaded', function () {
  const openModalBtn = document.getElementById('openModalBtn');
  const modalOverlay = document.getElementById('modalOverlay');
  const closeModalBtn = document.getElementById('closeModalBtn');

  // Show the modal
  openModalBtn.addEventListener('click', function () {
    modalOverlay.style.display = 'flex';
  });
  // Close the modal when "X" is clicked
  closeModalBtn.addEventListener('click', function () {
    modalOverlay.style.display = 'none';
  });
  // Close the modal if clicking outside the modal box
  modalOverlay.addEventListener('click', function (e) {
    if (e.target === modalOverlay) {
      modalOverlay.style.display = 'none';
    }
  });
});
```

OUTPUT:



When you click on “open modal” it will show output like this.



Modal Title

This is some content inside the modal popup.