ASSIGNMENT-14.1

NAME:VITTAM VENKATESH.

HALLTICKET NUMBER:2403A52419

BATCH:15

SUBJECT:AI ASSISTANT CODING

TASK-1

QUESTION:

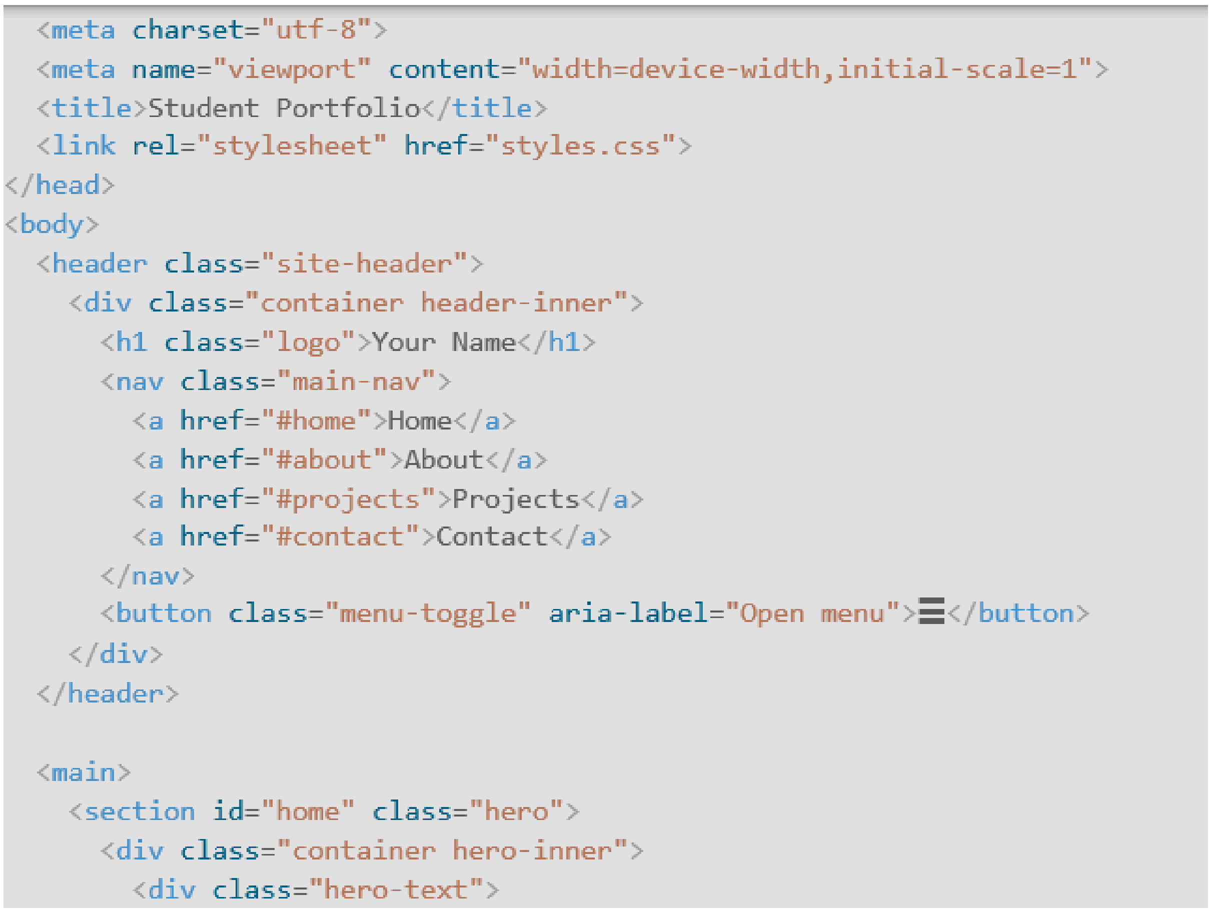
1: Create a Responsive Web Page Layout
Instructions:
• Design a basic web page layout with a header, main content area, and footer using HTML and CSS.
• Use AI to assist in generating responsive CSS for different screen sizes.
• Ensure the layout is clean and visually organized.
Expected Output:
• A 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

PROMPT:

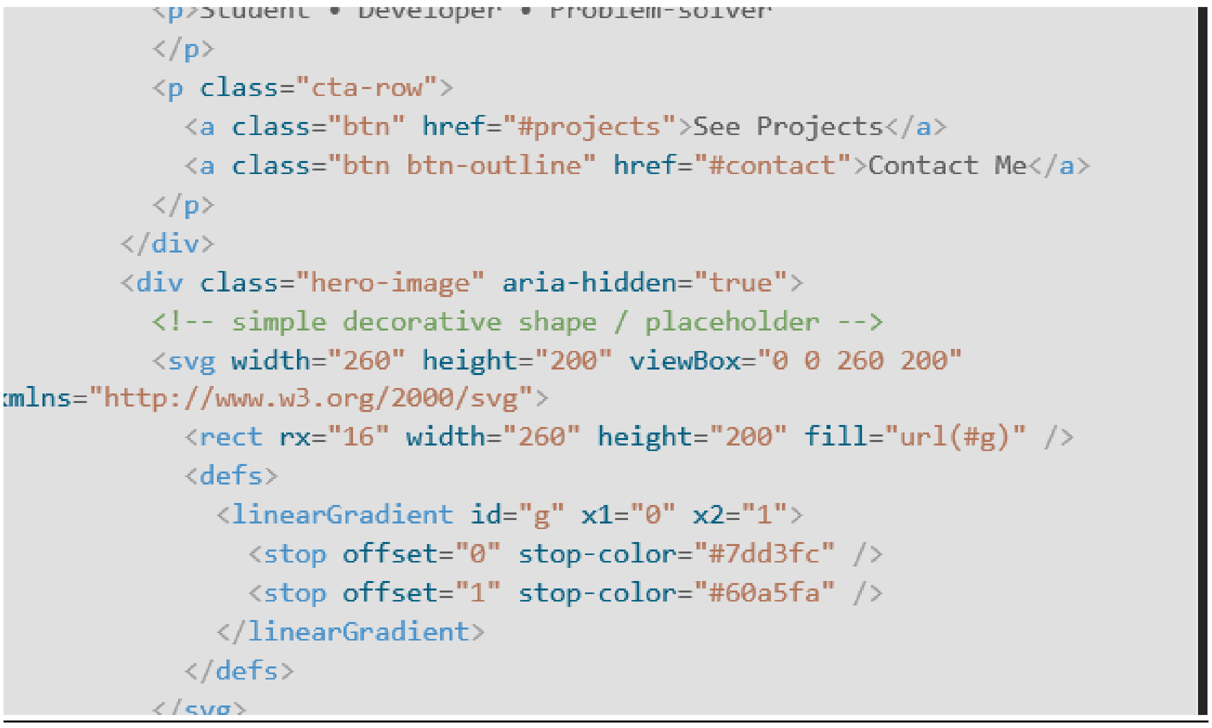
Design a responsive web page using HTML and CSS that includes a header, main content area, and footer.
Use AI assistance to generate CSS media queries for responsiveness on desktop, tablet, and mobile devices.
Ensure the layout looks clean and adapts smoothly to different screen sizes.

CODE:

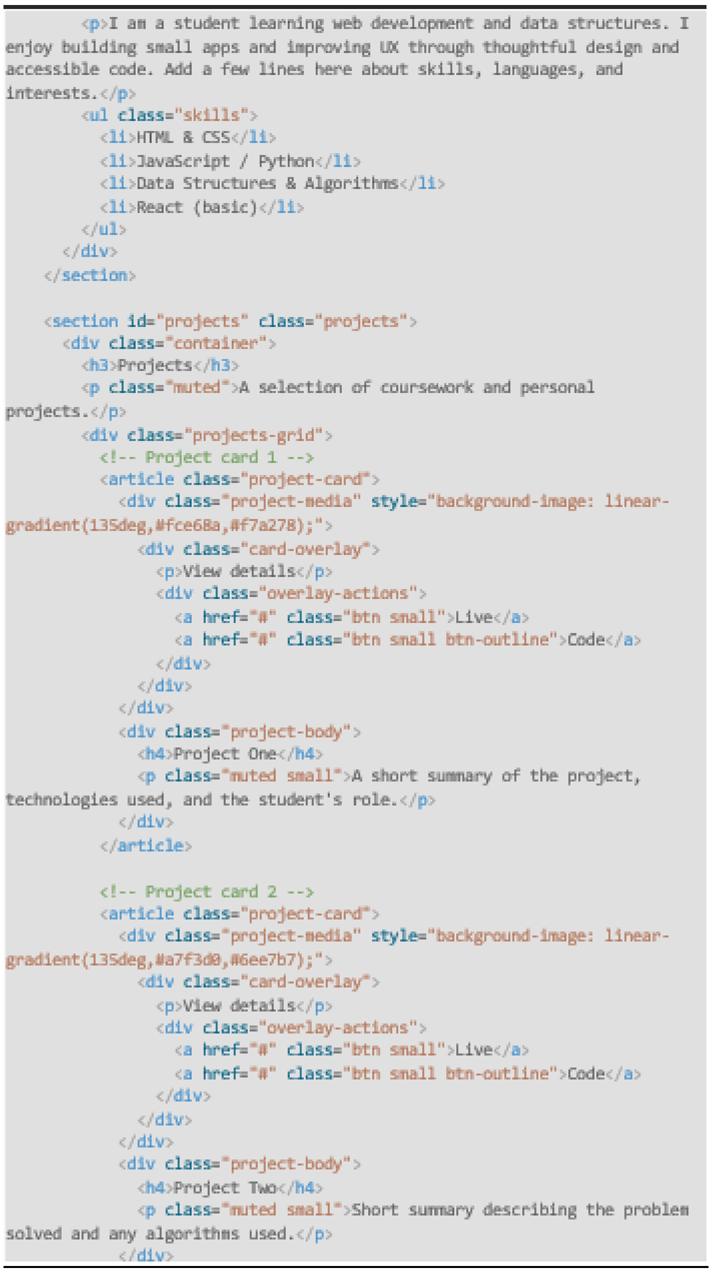
```
Student Portfolio
=================

This is a simple, responsive portfolio page generated to speed up the
process of creating a student portfolio. Files:

- `portfolio.html` — main HTML file. Sections: Home, About, Projects,
Contact.
- `styles.css` — responsive styles using grid and flexbox. Includes a hover
effect on project cards.

How to preview
---------------
1. Open `portfolio.html` in your browser (double-click the file or right-
click -> Open with...).
2. Resize the browser or open DevTools to test responsive breakpoints.

Customization tips
------------------
- Replace "Your Name" and the sample text with your real content.
- Add thumbnails or real images by changing the `background-image` in each
`.project-media` element.
- Link the Live / Code buttons to real URLs.

Next steps I can do for you
--------------------------
- Add more project cards dynamically from a JSON file.
```



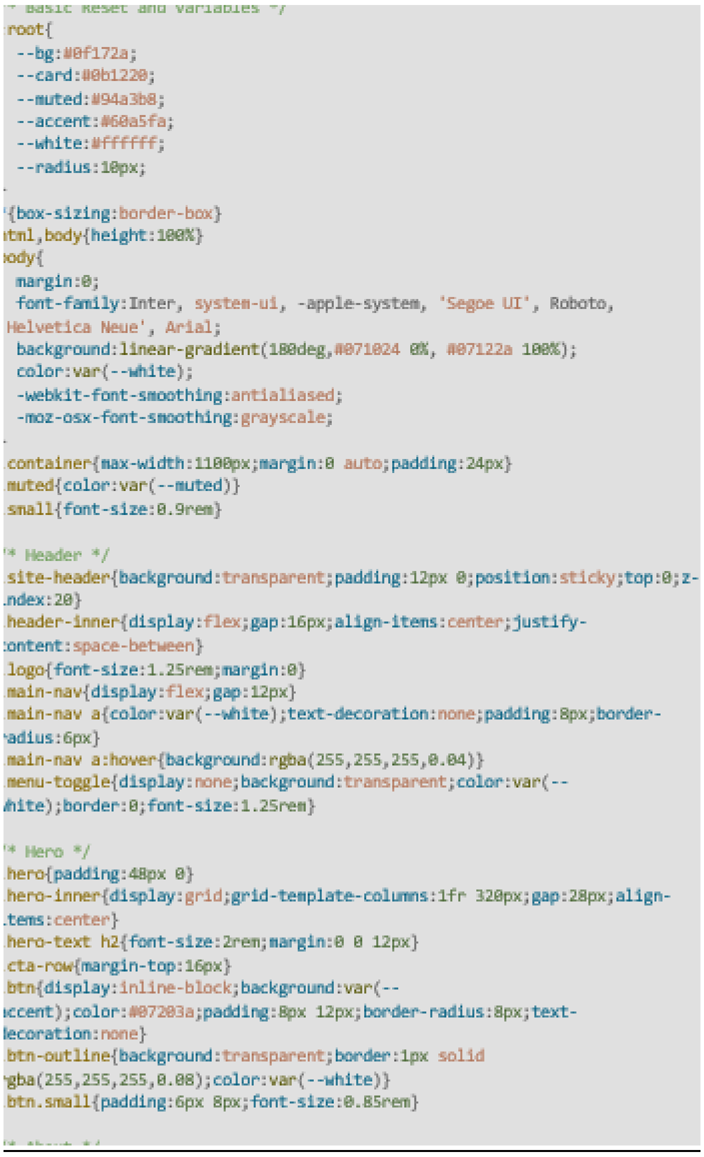
```html
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <title>Student Portfolio</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header class="site-header">
    <div class="container header-inner">
      <h1 class="logo">Your Name</h1>
      <nav class="main-nav">
        <a href="#home">Home</a>
        <a href="#about">About</a>
        <a href="#projects">Projects</a>
        <a href="#contact">Contact</a>
      </nav>
      <button class="menu-toggle" aria-label="Open menu">☰</button>
    </div>
  </header>

  <main>
    <section id="home" class="hero">
      <div class="container hero-inner">
        <div class="hero-text">
```



```html
          <p>Student • Developer • Problem-solver
          </p>
          <p class="cta-row">
            <a class="btn" href="#projects">See Projects</a>
            <a class="btn btn-outline" href="#contact">Contact Me</a>
          </p>
        </div>
        <div class="hero-image" aria-hidden="true">
          <!-- simple decorative shape / placeholder -->
          <svg width="260" height="200" viewBox="0 0 260 200"
xmlns="http://www.w3.org/2000/svg">
            <rect rx="16" width="260" height="200" fill="url(#g)" />
            <defs>
              <linearGradient id="g" x1="0" x2="1">
                <stop offset="0" stop-color="#7dd3fc" />
                <stop offset="1" stop-color="#60a5fa" />
              </linearGradient>
            </defs>
          </svg>
```



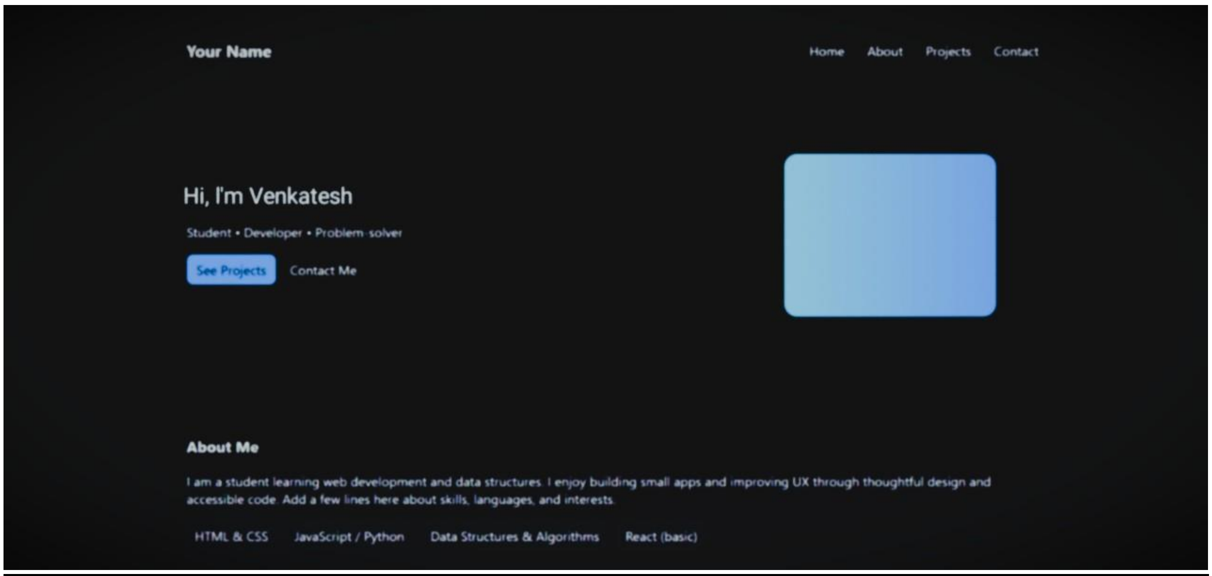
```
        <p>I am a student learning web development and data structures. I
enjoy building small apps and improving UX through thoughtful design and
accessible code. Add a few lines here about skills, languages, and
interests.</p>
        <ul class="skills">
          <li>HTML & CSS</li>
          <li>JavaScript / Python</li>
          <li>Data Structures & Algorithms</li>
          <li>React (basic)</li>
        </ul>
      </div>
    </section>

    <section id="projects" class="projects">
      <div class="container">
        <h3>Projects</h3>
        <p class="muted">A selection of coursework and personal
projects.</p>
        <div class="projects-grid">
          <!-- Project card 1 -->
          <article class="project-card">
            <div class="project-media" style="background-image: linear-
gradient(135deg,#fce68a,#f7a278);">
              <div class="card-overlay">
                <p>View details</p>
                <div class="overlay-actions">
                  <a href="#" class="btn small">Live</a>
                  <a href="#" class="btn small btn-outline">Code</a>
                </div>
              </div>
            </div>
            <div class="project-body">
              <h4>Project One</h4>
              <p class="muted small">A short summary of the project,
technologies used, and the student's role.</p>
            </div>
          </article>

          <!-- Project card 2 -->
          <article class="project-card">
            <div class="project-media" style="background-image: linear-
gradient(135deg,#a7f3d0,#6ee7b7);">
              <div class="card-overlay">
                <p>View details</p>
                <div class="overlay-actions">
                  <a href="#" class="btn small">Live</a>
                  <a href="#" class="btn small btn-outline">Code</a>
                </div>
              </div>
            </div>
            <div class="project-body">
              <h4>Project Two</h4>
              <p class="muted small">Short summary describing the problem
solved and any algorithms used.</p>
            </div>
```



```
'* Basic Reset and Variables */
root{
  --bg:#0f172a;
  --card:#0b1220;
  --muted:#94a3b8;
  --accent:#60a5fa;
  --white:#ffffff;
  --radius:10px;
.
'{box-sizing:border-box}
ıtml,body{height:100%}
ody{
  margin:0;
  font-family:Inter, system-ui, -apple-system, 'Segoe UI', Roboto,
 Helvetica Neue', Arial;
  background:linear-gradient(180deg,#071024 0%, #07122a 100%);
  color:var(--white);
  -webkit-font-smoothing:antialiased;
  -moz-osx-font-smoothing:grayscale;
.
.container{max-width:1100px;margin:0 auto;padding:24px}
.muted{color:var(--muted)}
.small{font-size:0.9rem}

'* Header */
.site-header{background:transparent;padding:12px 0;position:sticky;top:0;z-
.ndex:20}
.header-inner{display:flex;gap:16px;align-items:center;justify-
content:space-between}
.logo{font-size:1.25rem;margin:0}
.main-nav{display:flex;gap:12px}
.main-nav a{color:var(--white);text-decoration:none;padding:8px;border-
radius:6px}
.main-nav a:hover{background:rgba(255,255,255,0.04)}
.menu-toggle{display:none;background:transparent;color:var(--
white);border:0;font-size:1.25rem}

'* Hero */
.hero{padding:48px 0}
.hero-inner{display:grid;grid-template-columns:1fr 320px;gap:28px;align-
.tems:center}
.hero-text h2{font-size:2rem;margin:0 0 12px}
.cta-row{margin-top:16px}
.btn{display:inline-block;background:var(--
accent);color:#07203a;padding:8px 12px;border-radius:8px;text-
decoration:none}
.btn-outline{background:transparent;border:1px solid
rgba(255,255,255,0.08);color:var(--white)}
.btn.small{padding:6px 8px;font-size:0.85rem}

'* About */
```

```
.skills{display:flex;gap:12px;flex-wrap:wrap;list-
style:none;padding:0;margin:12px 0}
.skills li{background:rgba(255,255,255,0.03);padding:8px 10px;border-
radius:8px}

/* Projects Grid */
.projects{padding:28px 0}
.projects-grid{display:grid;grid-template-columns:repeat(auto-
fit,minmax(240px,1fr));gap:18px;margin-top:18px}
.project-card{background:linear-gradient(180deg, rgba(255,255,255,0.02),
rgba(255,255,255,0.01));border-radius:var(--radius);overflow:hidden;box-
shadow:0 6px 18px rgba(2,6,23,0.6);transition:transform 220ms ease, box-
shadow 220ms ease}
.project-media{height:160px;background-size:cover;background-
position:center;position:relative;display:flex;align-items:flex-end}
.project-body{padding:14px}
.project-body h4{margin:0 0 8px}
.project-body p{margin:0}

/* Hover effect: overlay fade + subtle lift */
.project-card:hover{transform:translateY(-8px) scale(1.01);box-shadow:0
18px 40px rgba(2,6,23,0.7)}
.card-overlay{position:absolute;inset:0;background:linear-gradient(180deg,
rgba(2,6,23,0.0), rgba(2,6,23,0.55));display:flex;flex-
direction:column;align-items:center;justify-
content:center;opacity:0;transform:translateY(6px);transition:opacity 220ms
ease, transform 220ms ease;padding:12px}
.project-card:hover .card-overlay{opacity:1;transform:translateY(0)}
.card-overlay p{margin:0 0 8px}
.overlay-actions{display:flex;gap:8px}

/* Contact */
.contact{padding:28px 0}
.contact-details a{color:var(--accent)}

/* Footer */
.site-footer{padding:18px 0;border-top:1px solid
rgba(255,255,255,0.03);margin-top:28px}

/* Responsive adjustments */
@media (max-width:880px){
  .hero-inner{grid-template-columns:1fr;}
  .menu-toggle{display:block}
  .main-
nav{position:fixed;right:16px;top:64px;background:rgba(2,6,23,0.9);padding:
12px;border-radius:10px;flex-direction:column;display:none}
  .main-nav.open{display:flex}
}

@media (max-width:480px){
  .project-media{height:140px}
  .hero-text h2{font-size:1.5rem}
}
```

OUTPUT:



Your Name

Home About Projects Contact

Hi, I'm Venkatesh

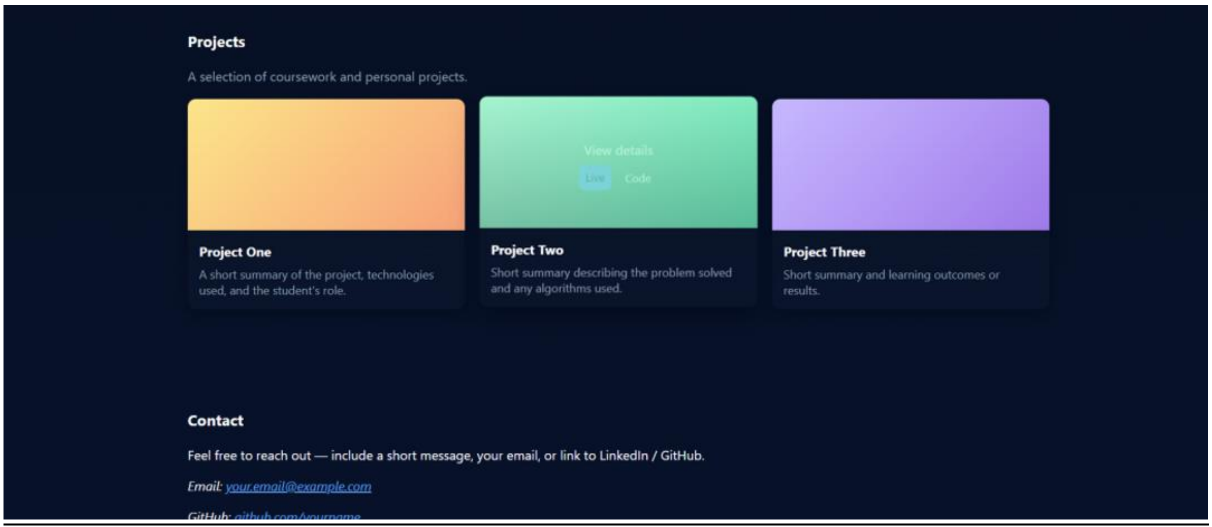
Student • Developer • Problem-solver

See Projects Contact Me

About Me

I am a student learning web development and data structures. I enjoy building small apps and improving UX through thoughtful design and accessible code. Add a few lines here about skills, languages, and interests.

HTML & CSS JavaScript / Python Data Structures & Algorithms React (basic)



Projects

A selection of coursework and personal projects.

View details

Live Code

Project One

A short summary of the project, technologies used, and the student's role.

Project Two

Short summary describing the problem solved and any algorithms used.

Project Three

Short summary and learning outcomes or results.

Contact

Feel free to reach out — include a short message, your email, or link to LinkedIn / GitHub.

Email: your.email@example.com

GitHub: github.com/yourname

OBSERVATION:

The web page layout was successfully created with a responsive design.
The header contained navigation links, the main section included placeholder text and images, and the footer displayed copyright information.
When viewed on different screen sizes (desktop, tablet, and mobile), the layout adjusted automatically — navigation stacked vertically on smaller screens, content resized properly, and no horizontal scrolling was needed.

TASK-2

QUESTION:

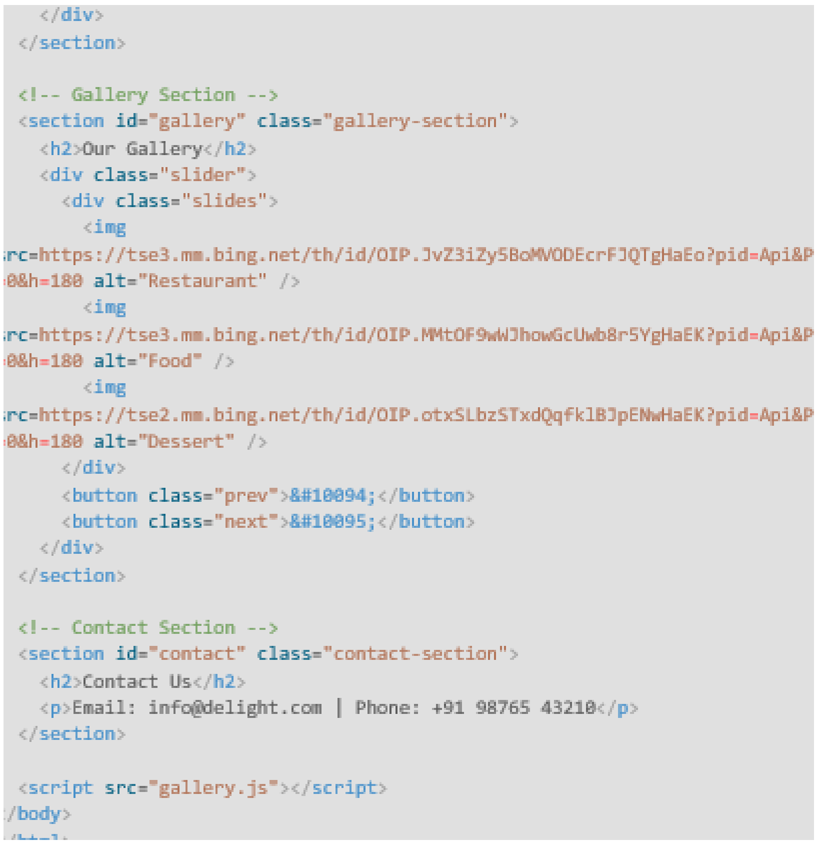
2: Interactive Button with JavaScript
Instructions:
• Create a button on a web page.
• Use AI to generate JavaScript code that displays an alert
message when the button is clicked.
• Ensure the code is clean and well-commented.
Expected Output:
• A web page with a button.
• Clicking the button shows a pop-up alert message, e.g., "Button
clicked!".

PROMPT:

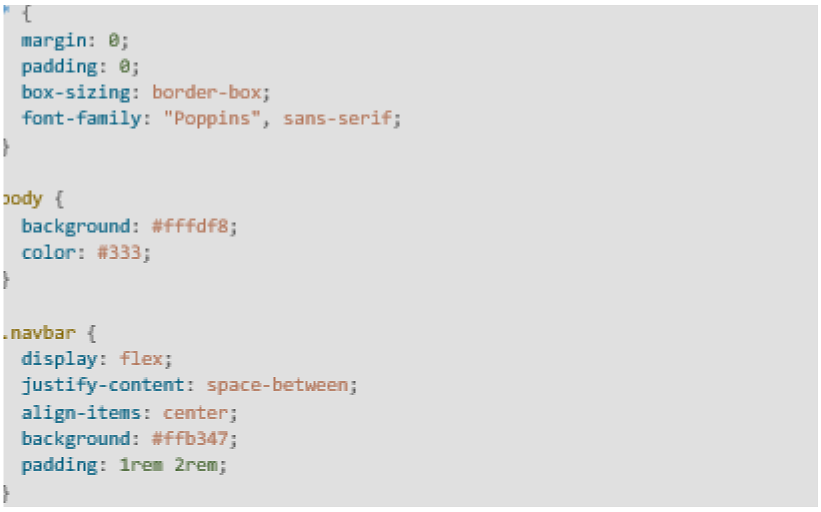
Create a web page containing a button.
Use JavaScript to generate code that displays an alert message when the button is clicked.
Ensure the script is clean, simple, and properly commented.

CODE:

```
Restaurant Landing Page
=======================

Files created:

- `restaurant.html` — landing page with navigation (Home, Menu, Gallery, Contact).
- `restaurant.css` — styles including menu cards and gallery layout.
- `gallery.js` — JavaScript slider with auto-play, next/prev buttons, and responsive resizing.

How to preview
---------------
1. Open `restaurant.html` in your browser (double-click the file or use "Open with").
2. The gallery uses images from Unsplash (random images based on keywords)
you'll need an internet connection for images to load.

Notes
-----
- The slider is intentionally simple: slides resize to the viewport width and the track is translated to show the current slide. It auto-plays and pauses on hover.
- Menu cards use placeholder images — replace the `img` src attributes in `restaurant.html` with local images if you prefer.

Next improvements I can add
--------------------------
- Add keyboard accessibility (left/right arrow to navigate slider).
- Add thumbnails or pagination dots below the slider.
- Wire the Contact section to a contact form endpoint (Formspree, Netlify Forms, or a small server).
```



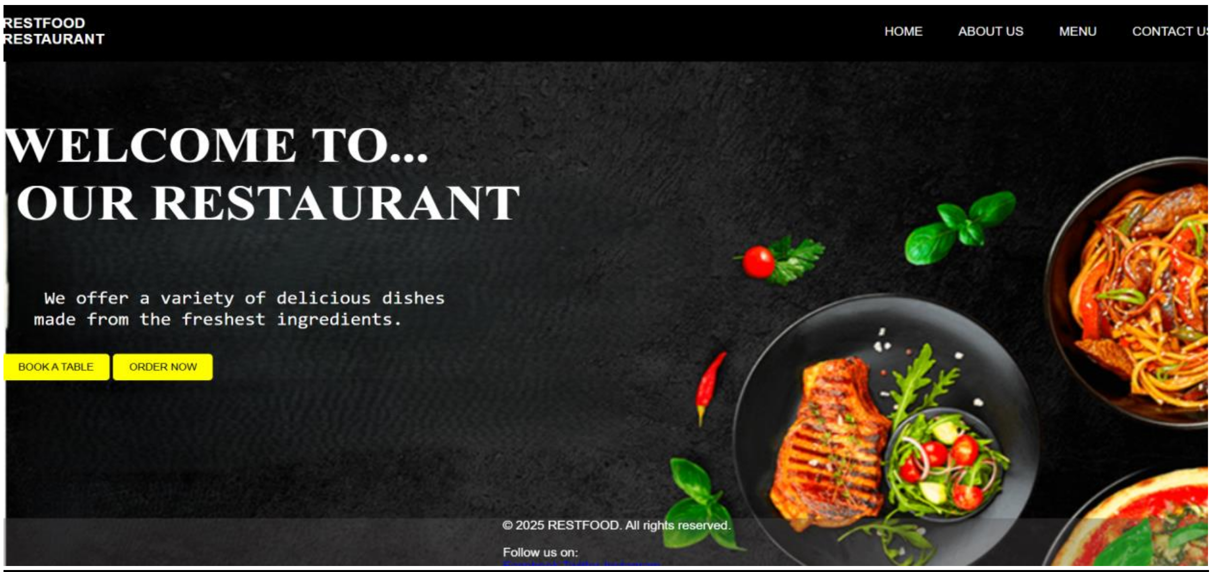
```
    </div>
  </section>

  <!-- Gallery Section -->
  <section id="gallery" class="gallery-section">
    <h2>Our Gallery</h2>
    <div class="slider">
      <div class="slides">
        <img
rc=https://tse3.mm.bing.net/th/id/OIP.JvZ3iZy5BoMVODEcrFJQTgHaEo?pid=Api&P
0&h=180 alt="Restaurant" />
        <img
rc=https://tse3.mm.bing.net/th/id/OIP.MMtOF9wWJhowGcUwb8r5YgHaEK?pid=Api&P
0&h=180 alt="Food" />
        <img
rc=https://tse2.mm.bing.net/th/id/OIP.otxSLbzSTxdQqfklBJpENwHaEK?pid=Api&P
0&h=180 alt="Dessert" />
      </div>
      <button class="prev">&#10094;</button>
      <button class="next">&#10095;</button>
    </div>
  </section>

  <!-- Contact Section -->
  <section id="contact" class="contact-section">
    <h2>Contact Us</h2>
    <p>Email: info@delight.com | Phone: +91 98765 43210</p>
  </section>

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



```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
}

ody {
  background: #fffdf8;
  color: #333;
}

.navbar {
  display: flex;
  justify-content: space-between;
  align-items: center;
  background: #ffb347;
  padding: 1rem 2rem;
}
```

```
  padding: 2rem;
  text-align: center;
  background: #fafafa;
}

.slider {
  position: relative;
  width: 80%;
  margin: 1rem auto;
  overflow: hidden;
  border-radius: 10px;
}

.slides {
  display: flex;
  transition: transform 0.5s ease;
}

.slides img {
  width: 100%;
  border-radius: 10px;
}

.prev, .next {
  position: absolute;
  top: 50%;
  transform: translateY(-50%);
  background: rgba(0,0,0,0.5);
  color: white;
  border: none;
  padding: 0.5rem 1rem;
  cursor: pointer;
  border-radius: 50%;
}

.prev { left: 10px; }
.next { right: 10px; }

.contact-section {
  text-align: center;
  padding: 2rem;
  background: #ffb347;
  color: white;
}
```

<u>OUTPUT:</u>

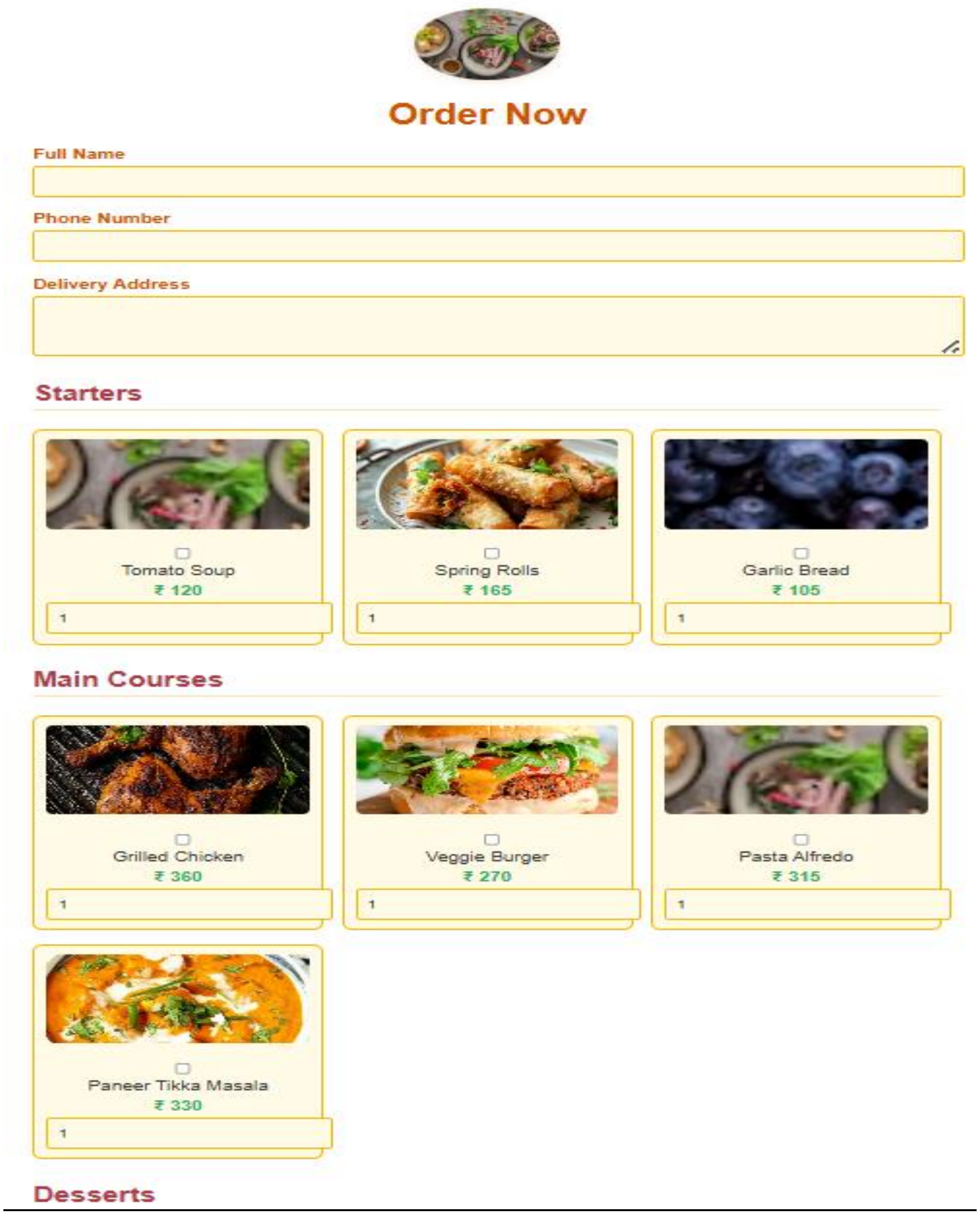


WELCOME TO...
OUR RESTAURANT

We offer a variety of delicious dishes made from the freshest ingredients.

© 2025 RESTFOOD. All rights reserved.

Follow us on:



Order Now

Full Name

Phone Number

Delivery Address

Starters

Tomato Soup
₹ 120
1

Spring Rolls
₹ 165
1

Garlic Bread
₹ 105
1

Main Courses

Grilled Chicken
₹ 360
1

Veggie Burger
₹ 270
1

Pasta Alfredo
₹ 315
1

Paneer Tikka Masala
₹ 330
1

Desserts

OBSERVATION:

The web page was successfully created with a clickable button.
When the button was clicked, a pop-up alert message appeared displaying “Button clicked!”.
The JavaScript code worked correctly and demonstrated the use of event handling in web development.
This experiment shows how JavaScript adds interactivity to a web page.

TASK-3

QUESTION:

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:
o Submitting with empty fields shows error messages.

o Submitting with valid input displays a "Form submitted
successfully" message.

PROMPT:

Design a contact form using html with three are Name, Email, and Message.
Use JavaScript to validate the form inputs all fields must be filled, and the email must be in a valid format.
If any field is invalid, show an inline error message below that field.
When all inputs are valid, display an alert saying Form submitted successfully

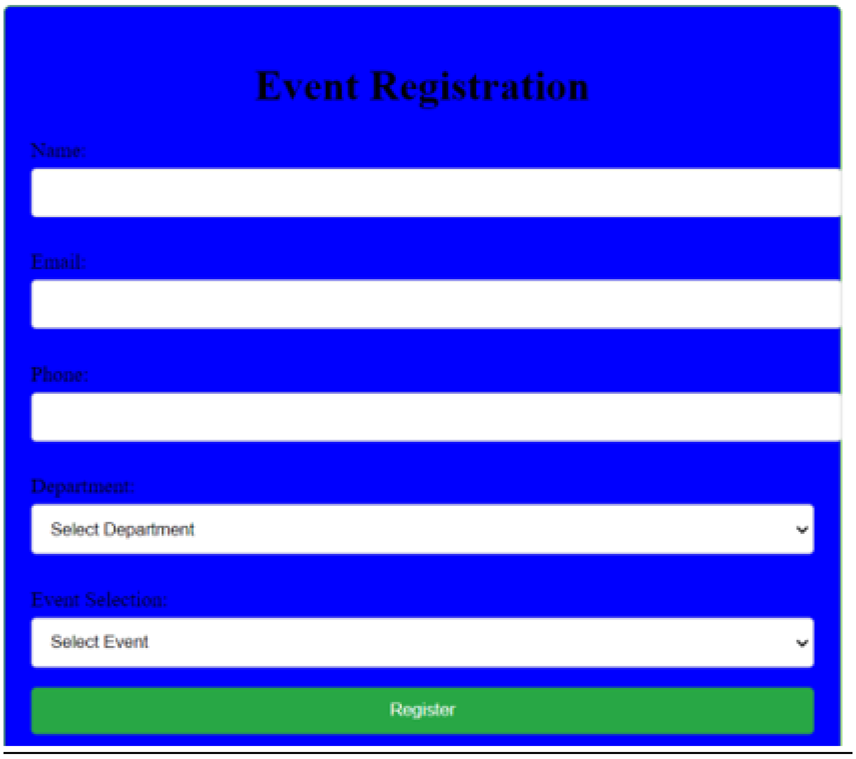
CODE:

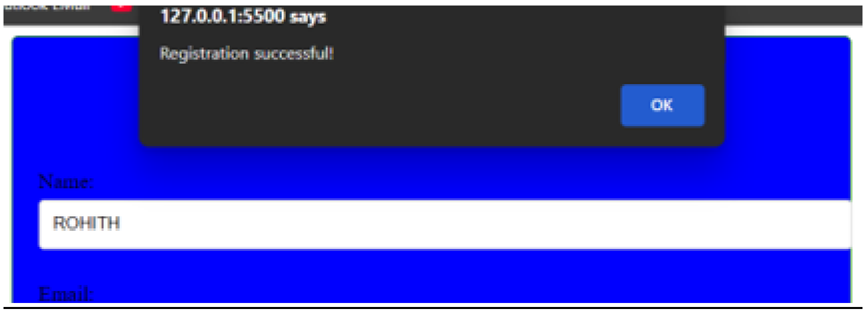
```
!DOCTYPE html>
tml lang="en">
ead>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" href="registration.css">
   <title>Event Registration Form</title>
/head>
ody>
   <div class="container">
       <h1>Event Registration</h1>
       <form id="registrationForm">
           <label for="name">Name:</label>
           <input type="text" id="name" name="name" required>

           <label for="email">Email:</label>
           <input type="email" id="email" name="email" required>

           <label for="phone">Phone:</label>
           <input type="tel" id="phone" name="phone" required>
```

```css
.container {
    max-width: 600px;
    margin: auto;
    padding: 20px;
    border: 1px solid green;
    border-radius: 5px;
    background-color: blue;
}

h1 {
    text-align: center;
}

label {
    display: block;
    margin: 10px 0 5px;
}

input, select {
    width: 100%;
    padding: 10px;
    margin-bottom: 15px;
    border: 1px solid #ccc;
    border-radius: 4px;
}

button {
    width: 100%;
    padding: 10px;
    background-color: #28a745;
```

```javascript
document.getElementById('registrationForm').addEventListener('submit',
function(event) {
    event.preventDefault(); // Prevent form submission

    // Validate email format
    const email = document.getElementById('email').value;
    const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!emailPattern.test(email)) {
        alert('Please enter a valid email address.');
        return;
    }

    // Validate phone number length
    const phone = document.getElementById('phone').value;
    if (phone.length < 10) {
        alert('Phone number must be at least 10 digits long.');
        return;
    }

    // If all validations pass, submit the form
    alert('Registration successful!');
    // Here you can add code to actually submit the form data to the server
});
```

OUTPUT:





127.0.0.1:5500 says

Registration successful!

OBSERVATION:

The contact form successfully validates user inputs.
Error messages are displayed when fields are empty or the email format is incorrect.
When all inputs are valid, the form shows a Form submitted successfully alert message, indicating proper validation and functionality. The contact form successfully validates user inputs.

TASK-4

QUESTION:

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:
o “Add Item” adds a new item to the list dynamically.
o “Remove Item” deletes the last item.
• Updates occur without reloading the page.

PROMPT:

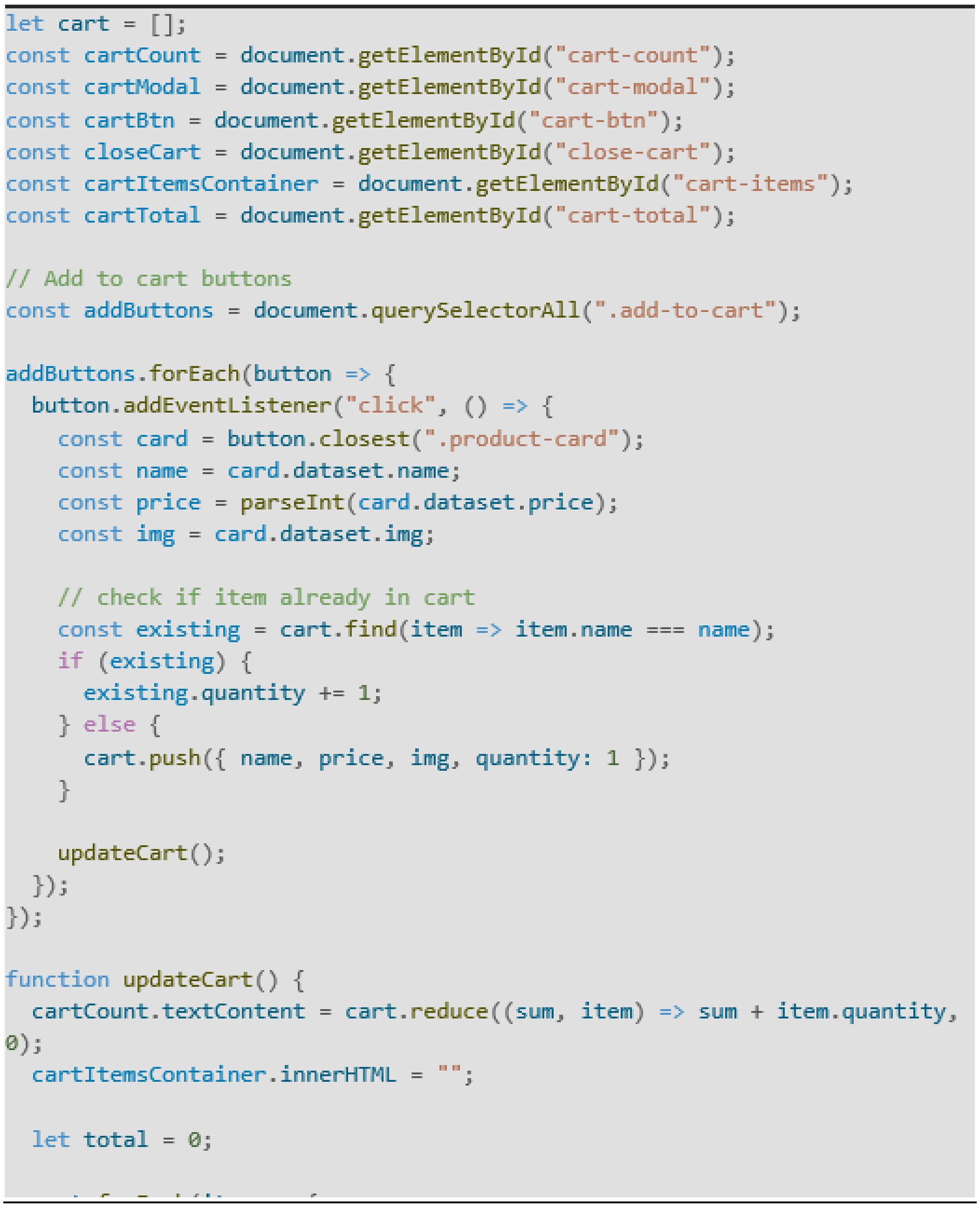
Create a web page with an HTML list of items .
Use JavaScript to dynamically add a new item when the “Add Item” button is clicked and remove the last item when the “Remove Item” button is clicked.
The list should update instantly without reloading the page.

CODE:

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>ShopEase - Products</title>
  <link rel="stylesheet" href="product.css" />
</head>
<body>
  <!-- Header -->
  <header class="header">
    <h1>ShopEase</h1>
    <div class="cart-icon" id="cart-btn">
      🛒 <span id="cart-count">0</span>
    </div>
  </header>

  <!-- Product Grid -->
  <main class="product-grid">
    <div class="product-card" data-name="Wireless Headphones" data-
price="1999" data-img="https://source.unsplash.com/300x200/?headphones">
```

```
  </div>

  <div class="product-card" data-name="Keyboard Combo" data-price="1299
ata-img="https://source.unsplash.com/300x200/?keyboard">
    <img
rc=https://tse3.mm.bing.net/th/id/OIP.k__xMGWza7Mg8oY6Bs6L7wHaFB?pid=Api
0&h=180 alt="Keyboard" />
    <h2>Keyboard Combo</h2>
    <p>₹1299</p>
    <button class="add-to-cart">Add to Cart</button>
  </div>
</main>

<!-- Cart Modal -->
<div class="cart-modal" id="cart-modal">
  <div class="cart-content">
    <span id="close-cart" class="close-btn">&times;</span>
    <h2>Your Cart</h2>
    <div id="cart-items"></div>
    <h3>Total: ₹<span id="cart-total">0</span></h3>

    <h4>Choose Payment Method:</h4>
    <div class="payment-methods">
      <label><input type="radio" name="payment" checked /> UPI</label>
      <label><input type="radio" name="payment" /> Credit/Debit
ard</label>
      <label><input type="radio" name="payment" /> Cash on
elivery</label>
    </div>

    <button id="checkout-btn">Proceed to Pay</button>
  </div>
</div>

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

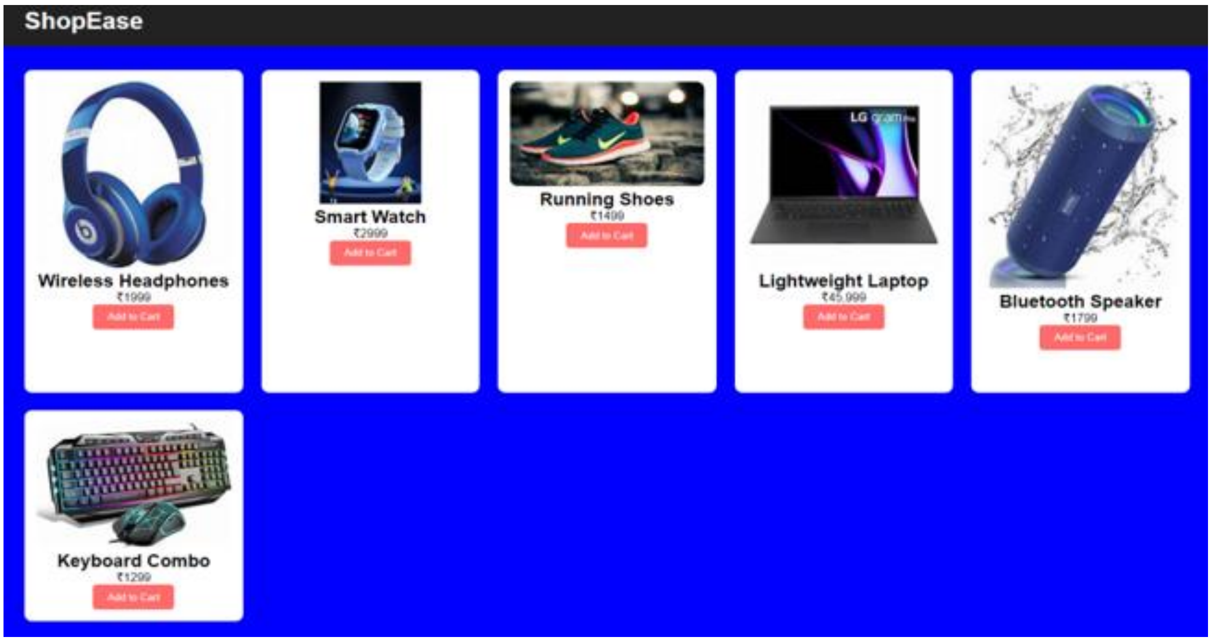
```css
  align-items: center;
  background: #222;
  color: white;
  padding: 1rem 2rem;
}

.cart-icon {
  cursor: pointer;
  font-size: 1.4rem;
  position: relative;
}

#cart-count {
  background: crimson;
  color: white;
  border-radius: 50%;
  padding: 3px 8px;
  font-size: 0.8rem;
  position: absolute;
  top: -10px;
  right: -12px;
}

/* Product grid */
.product-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 1.5rem;
  padding: 2rem;
}

.product-card {
  background: white;
  border-radius: 10px;
  text-align: center;
  box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);
  padding: 1rem;
  transition: transform 0.3s;
}

.product-card:hover {
  transform: scale(1.05);
}

.product-card img {
  width: 100%;
  border-radius: 10px;
}

.add-to-cart {
  background: #ff6b6b;
  border: none;
```



```
let cart = [];
const cartCount = document.getElementById("cart-count");
const cartModal = document.getElementById("cart-modal");
const cartBtn = document.getElementById("cart-btn");
const closeCart = document.getElementById("close-cart");
const cartItemsContainer = document.getElementById("cart-items");
const cartTotal = document.getElementById("cart-total");

// Add to cart buttons
const addButtons = document.querySelectorAll(".add-to-cart");

addButtons.forEach(button => {
  button.addEventListener("click", () => {
    const card = button.closest(".product-card");
    const name = card.dataset.name;
    const price = parseInt(card.dataset.price);
    const img = card.dataset.img;

    // check if item already in cart
    const existing = cart.find(item => item.name === name);
    if (existing) {
      existing.quantity += 1;
    } else {
      cart.push({ name, price, img, quantity: 1 });
    }

    updateCart();
  });
});

function updateCart() {
  cartCount.textContent = cart.reduce((sum, item) => sum + item.quantity,
0);
  cartItemsContainer.innerHTML = "";

  let total = 0;
```

```
    const itemDiv = document.createElement("div");
    itemDiv.classList.add("cart-item");
    itemDiv.innerHTML = `
      <img src="${item.img}" alt="${item.name}">
      <div style="flex:1;">
        <strong>${item.name}</strong><br>
        ₹${item.price} × ${item.quantity}
      </div>
      <button class="remove-btn">✖</button>
    `;
    cartItemsContainer.appendChild(itemDiv);

    itemDiv.querySelector(".remove-btn").addEventListener("click", () => {
      removeItem(item.name);
    });

    total += item.price * item.quantity;
  });

  cartTotal.textContent = total;


unction removeItem(name) {
  cart = cart.filter(item => item.name !== name);
  updateCart();


/ Modal open/close
artBtn.addEventListener("click", () => {
  cartModal.style.display = "flex";
);
loseCart.addEventListener("click", () => {
  cartModal.style.display = "none";
);

/ Checkout
locument.getElementById("checkout-btn").addEventListener("click", () => {
  if (cart.length === 0) {
    alert("Your cart is empty!");
  } else {
    alert("Payment Successful! Thank you for shopping with ShopEase ❤️");
    cart = [];
    updateCart();
    cartModal.style.display = "none";
  }
);
```

OUTPUT:



ShopEase

Wireless Headphones
₹1999
Add to Cart

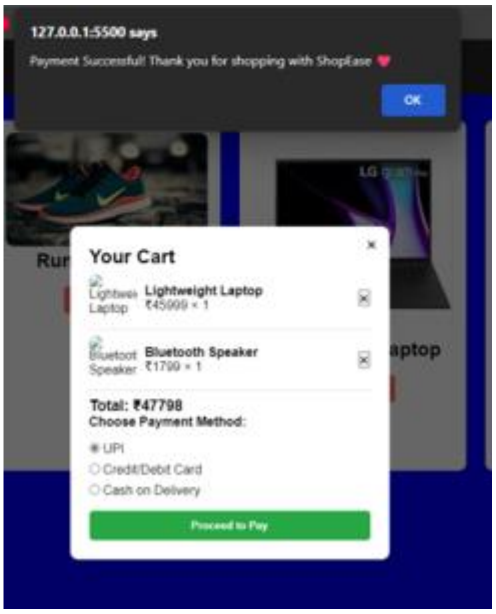
Smart Watch
₹2999
Add to Cart

Running Shoes
₹1499
Add to Cart

Lightweight Laptop
₹45,999
Add to Cart

Bluetooth Speaker
₹1799
Add to Cart

Keyboard Combo
₹1299
Add to Cart



127.0.0.1:5500 says
Payment Successful! Thank you for shopping with ShopEase 💗
OK

Your Cart ×

Lightweight Laptop
₹45999 × 1

Bluetooth Speaker
₹1799 × 1

Total: ₹47798
Choose Payment Method:
[✓ UPI]
Credit/Debit Card
Cash on Delivery

Proceed to Pay

OBSERVATION:

The web page successfully adds and removes list items dynamically.
Clicking Add Item adds a new item to the list, while Remove Item deletes the last one.
All updates happen smoothly without refreshing the page, showing proper use of JavaScript for dynamic content manipulation.

TASK-5

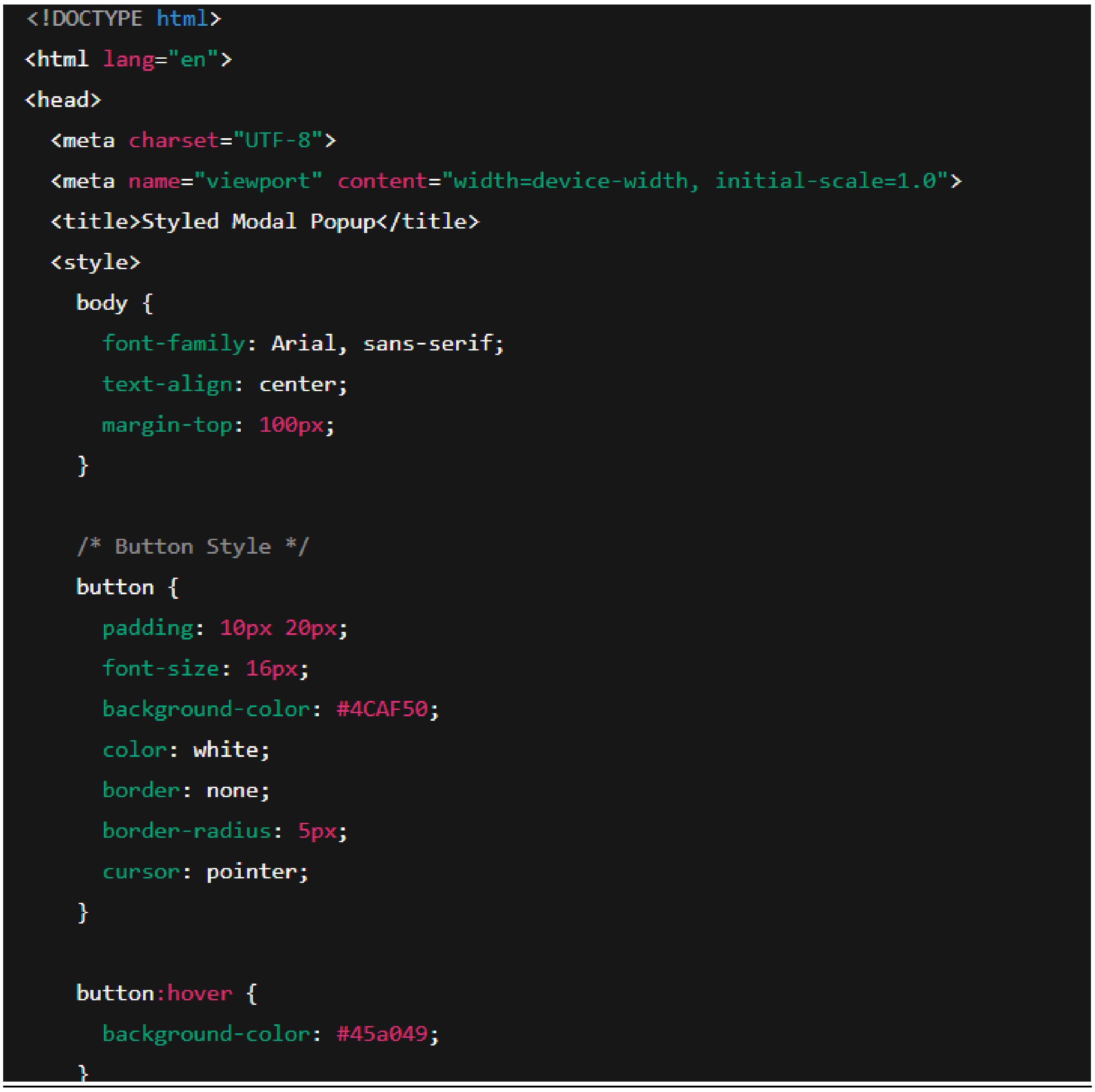
QUESTION:

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.

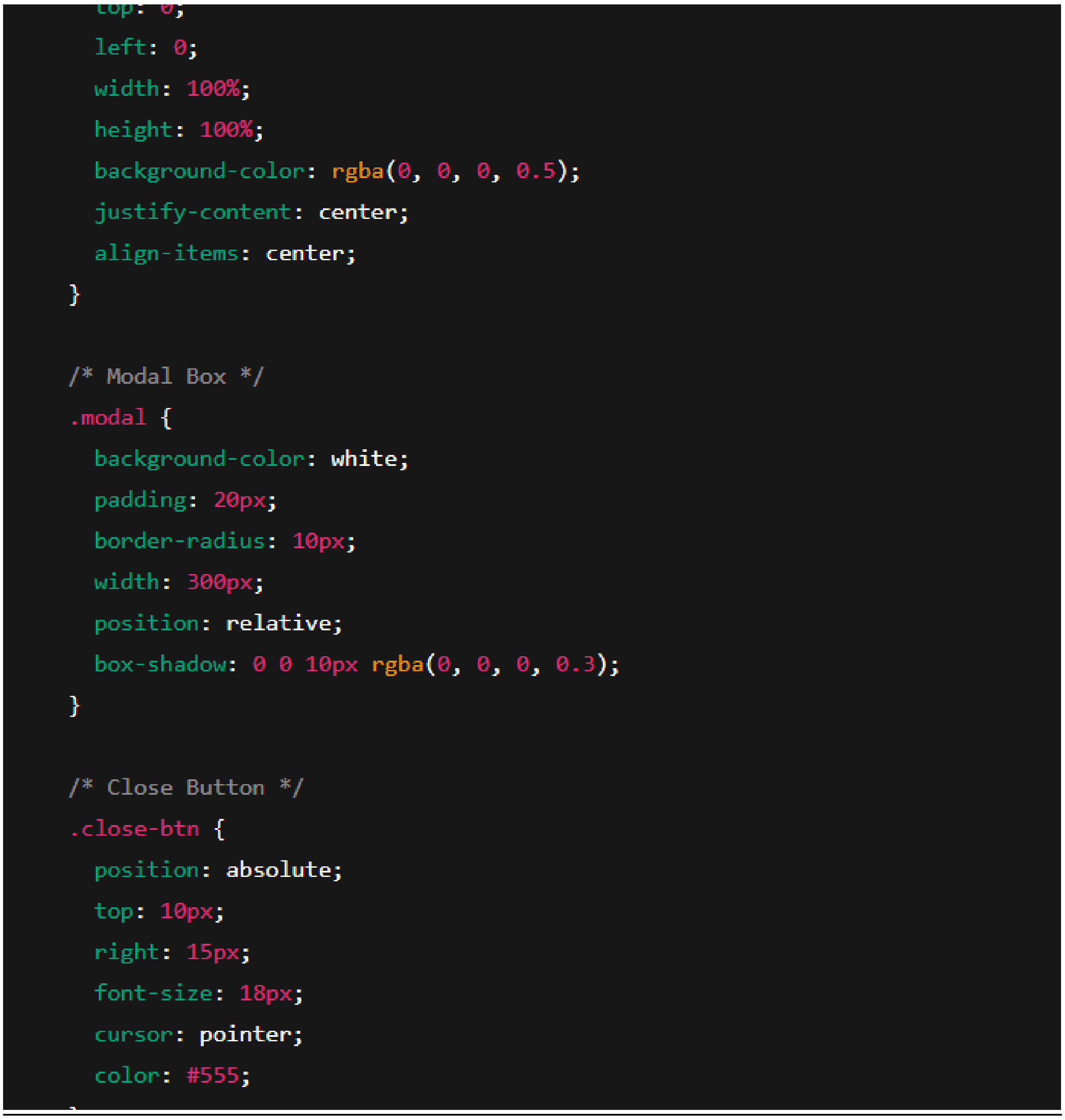
PROMPT:

Create a web page with a button labeled Open Modal.
When the button is clicked, a modal popup should appear with some text content and a close button.
Style the modal using CSS to have a centered box and a semi-transparent background overlay.
The modal should close when the user clicks the button or anywhere outside the modal area.

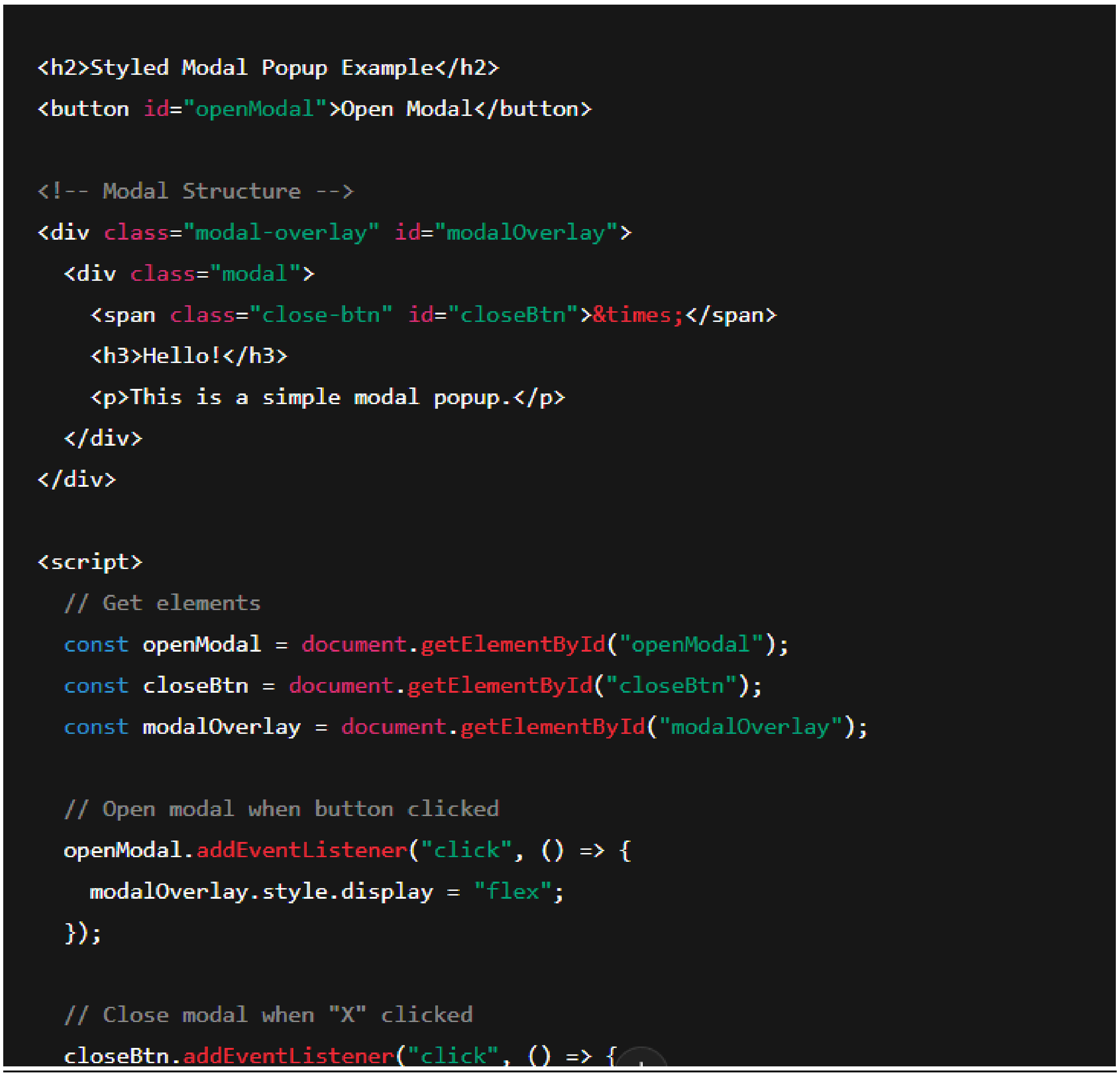
CODE:



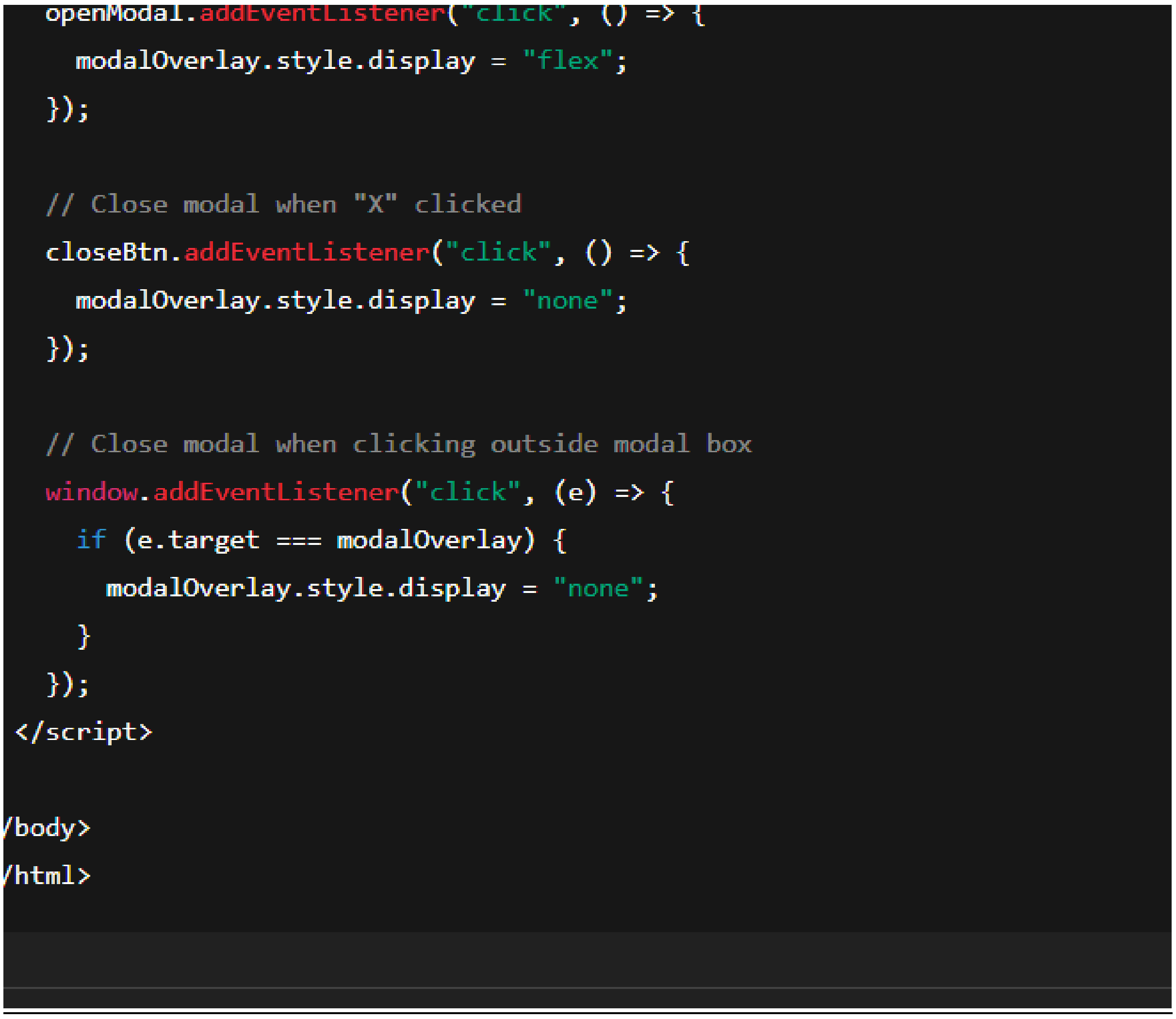
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Styled Modal Popup</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      margin-top: 100px;
    }

    /* Button Style */
    button {
      padding: 10px 20px;
      font-size: 16px;
      background-color: #4CAF50;
      color: white;
      border: none;
      border-radius: 5px;
      cursor: pointer;
    }

    button:hover {
      background-color: #45a049;
    }
```



```css
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-color: rgba(0, 0, 0, 0.5);
  justify-content: center;
  align-items: center;
}

/* Modal Box */
.modal {
  background-color: white;
  padding: 20px;
  border-radius: 10px;
  width: 300px;
  position: relative;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
}

/* Close Button */
.close-btn {
  position: absolute;
  top: 10px;
  right: 15px;
  font-size: 18px;
  cursor: pointer;
  color: #555;
```

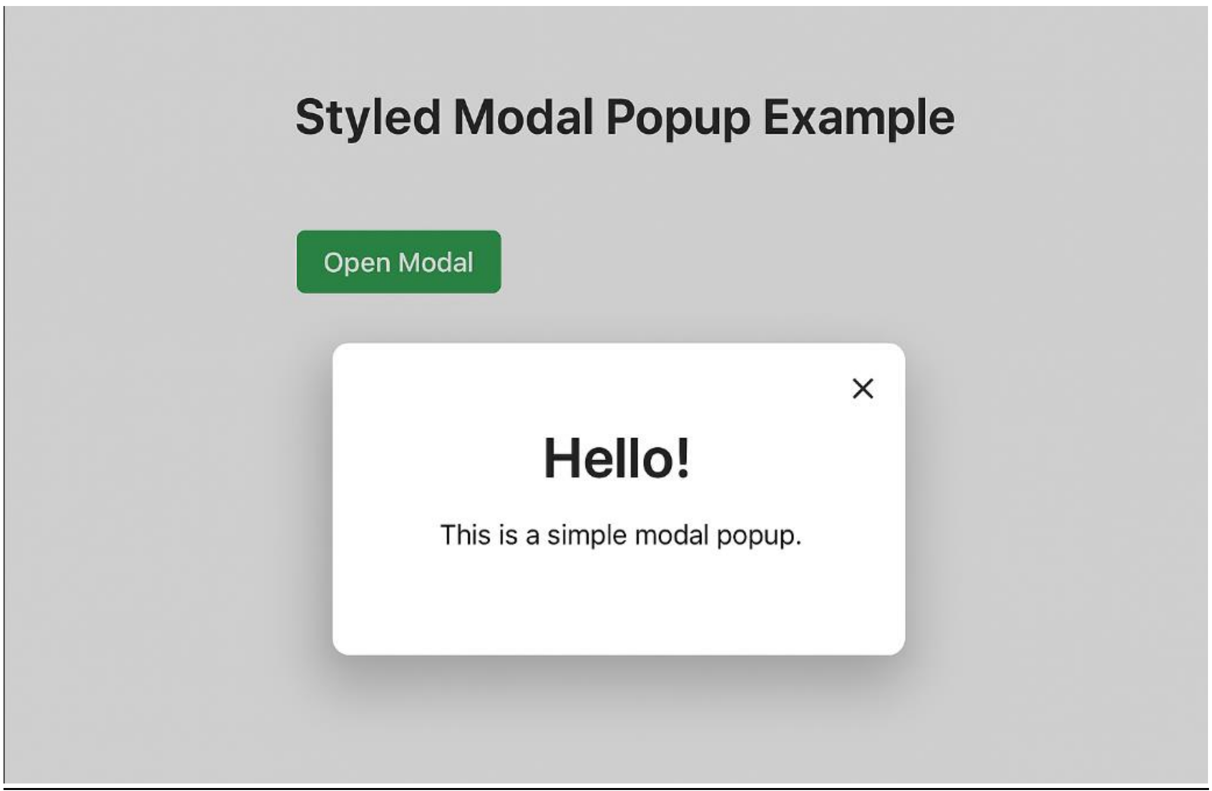


```html
<h2>Styled Modal Popup Example</h2>
<button id="openModal">Open Modal</button>

<!-- Modal Structure -->
<div class="modal-overlay" id="modalOverlay">
  <div class="modal">
    <span class="close-btn" id="closeBtn">&times;</span>
    <h3>Hello!</h3>
    <p>This is a simple modal popup.</p>
  </div>
</div>

<script>
  // Get elements
  const openModal = document.getElementById("openModal");
  const closeBtn = document.getElementById("closeBtn");
  const modalOverlay = document.getElementById("modalOverlay");

  // Open modal when button clicked
  openModal.addEventListener("click", () => {
    modalOverlay.style.display = "flex";
  });

  // Close modal when "X" clicked
  closeBtn.addEventListener("click", () => {
```



```
  openModal.addEventListener("click", () => {
    modalOverlay.style.display = "flex";
  });

  // Close modal when "X" clicked
  closeBtn.addEventListener("click", () => {
    modalOverlay.style.display = "none";
  });

  // Close modal when clicking outside modal box
  window.addEventListener("click", (e) => {
    if (e.target === modalOverlay) {
      modalOverlay.style.display = "none";
    }
  });
</script>

/body>
/html>
```

OUTPUT:


Styled Modal Popup Example

Hello!

This is a simple modal popup.

OBSERVATION:

The web page successfully displays a styled modal popup.
Clicking Open Modal shows the popup with content and a semi-transparent overlay.
The modal closes smoothly when the X button or the overlay is clicked, showing correct implementation of modal functionality using html, css, and javaScript.