Assignment-14.5

Pasula Ajay Kumar

2303A51729

Batch-11

Lab 14: Web Design Application – AI-Assisted HTML/CSS/JS Generation

Lab Objectives

• Design functional, visually appealing web applications

Week7 -Friday

using HTML, CSS, and JavaScript with AI assistance.

• Apply responsive, accessible, and interactive design principles.

• Create practical UI components for real-world web applications.

• Use AI to optimize layout, UX, and performance

Task 1 – Image Gallery Website Create an image gallery web page.

Requirements:

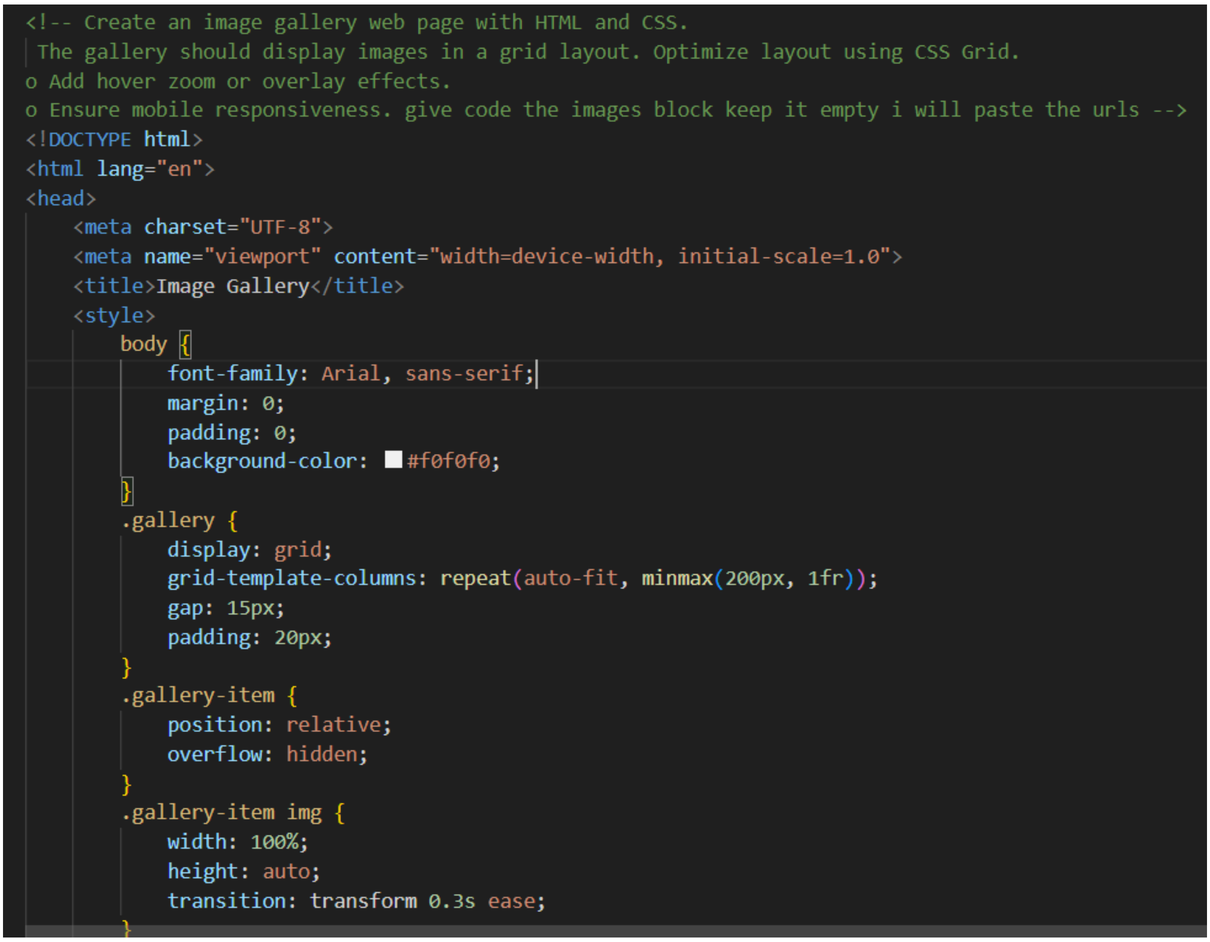
• Display images in a grid layout.

• Use AI to:

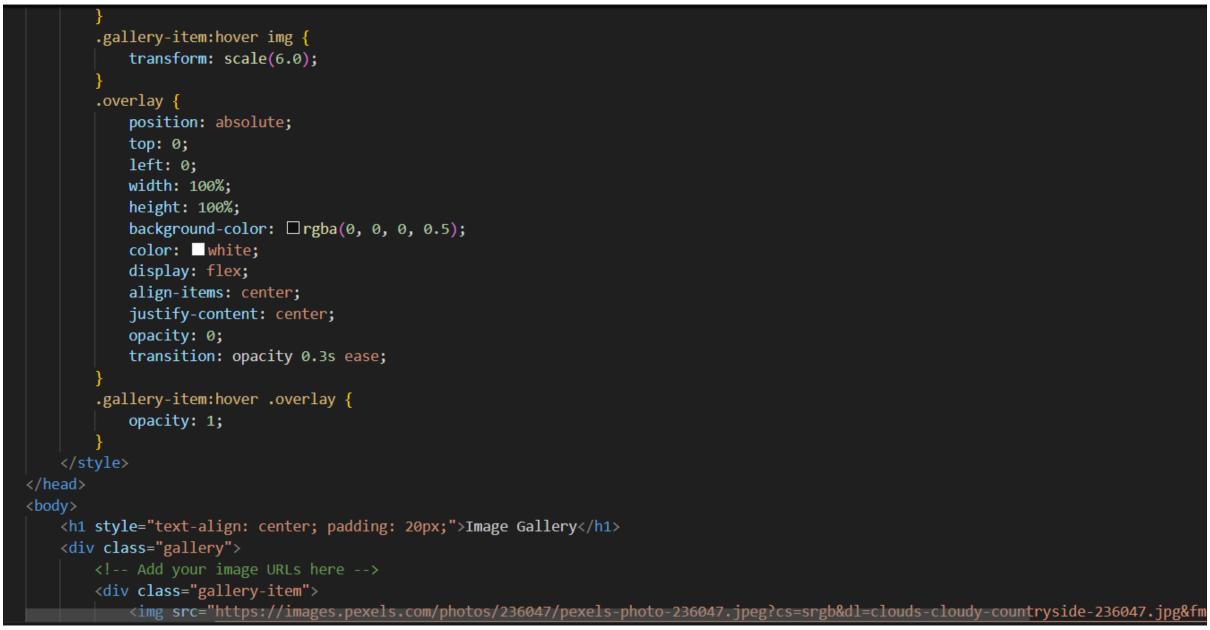
o Optimize layout using CSS Grid.

o Add hover zoom or overlay effects.

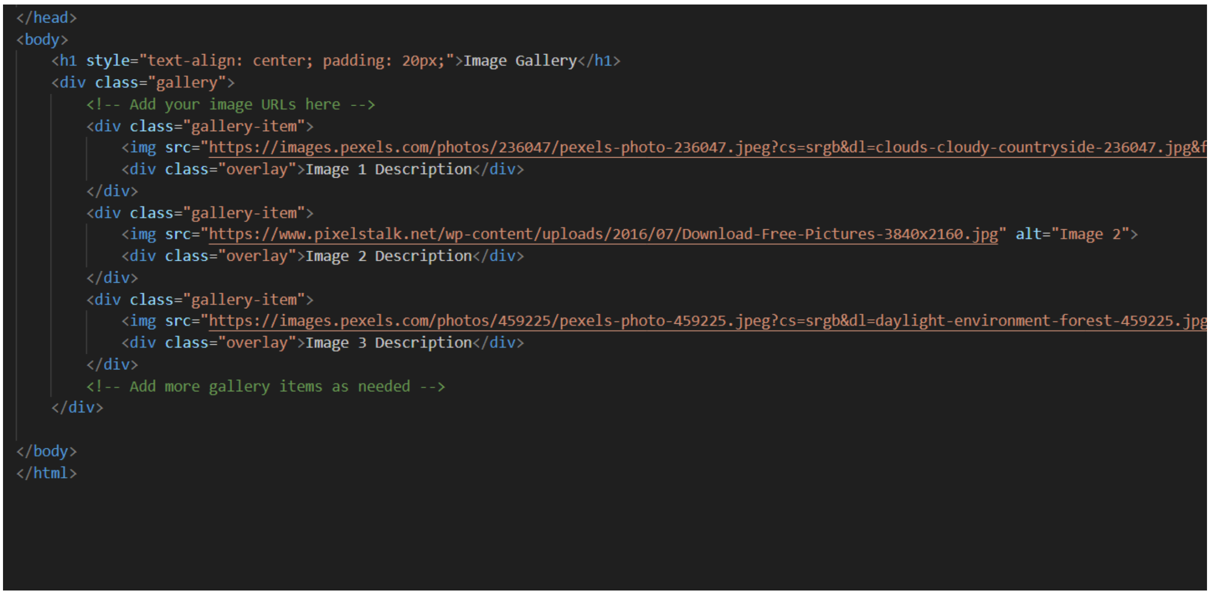
o Ensure mobile responsiveness.



```
<!-- Create an image gallery web page with HTML and CSS.
 The gallery should display images in a grid layout. Optimize layout using CSS Grid.
o Add hover zoom or overlay effects.
o Ensure mobile responsiveness. give code the images block keep it empty i will paste the urls -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Image Gallery</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f0f0f0;
        }
        .gallery {
            display: grid;
            grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
            gap: 15px;
            padding: 20px;
        }
        .gallery-item {
            position: relative;
            overflow: hidden;
        }
        .gallery-item img {
            width: 100%;
            height: auto;
            transition: transform 0.3s ease;
        }
```



```
        }
        .gallery-item:hover img {
            transform: scale(6.0);
        }
        .overlay {
            position: absolute;
            top: 0;
            left: 0;
            width: 100%;
            height: 100%;
            background-color: rgba(0, 0, 0, 0.5);
            color: white;
            display: flex;
            align-items: center;
            justify-content: center;
            opacity: 0;
            transition: opacity 0.3s ease;
        }
        .gallery-item:hover .overlay {
            opacity: 1;
        }
    </style>
</head>
<body>
    <h1 style="text-align: center; padding: 20px;">Image Gallery</h1>
    <div class="gallery">
        <!-- Add your image URLs here -->
        <div class="gallery-item">
            <img src="https://images.pexels.com/photos/236047/pexels-photo-236047.jpeg?cs=srgb&dl=clouds-cloudy-countryside-236047.jpg&fm
```



```
</head>
<body>
    <h1 style="text-align: center; padding: 20px;">Image Gallery</h1>
    <div class="gallery">
        <!-- Add your image URLs here -->
        <div class="gallery-item">
            <img src="https://images.pexels.com/photos/236047/pexels-photo-236047.jpeg?cs=srgb&dl=clouds-cloudy-countryside-236047.jpg&f
            <div class="overlay">Image 1 Description</div>
        </div>
        <div class="gallery-item">
            <img src="https://www.pixelstalk.net/wp-content/uploads/2016/07/Download-Free-Pictures-3840x2160.jpg" alt="Image 2">
            <div class="overlay">Image 2 Description</div>
        </div>
        <div class="gallery-item">
            <img src="https://images.pexels.com/photos/459225/pexels-photo-459225.jpeg?cs=srgb&dl=daylight-environment-forest-459225.jpg
            <div class="overlay">Image 3 Description</div>
        </div>
        <!-- Add more gallery items as needed -->
    </div>

</body>
</html>
```

Output

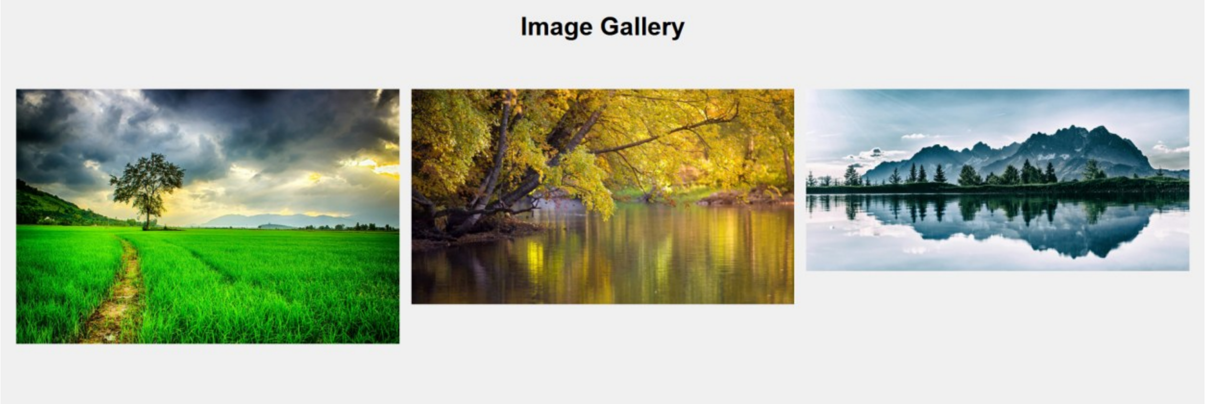


Image Gallery

Task 2 – Profile Card Generator

Create a profile card web page.

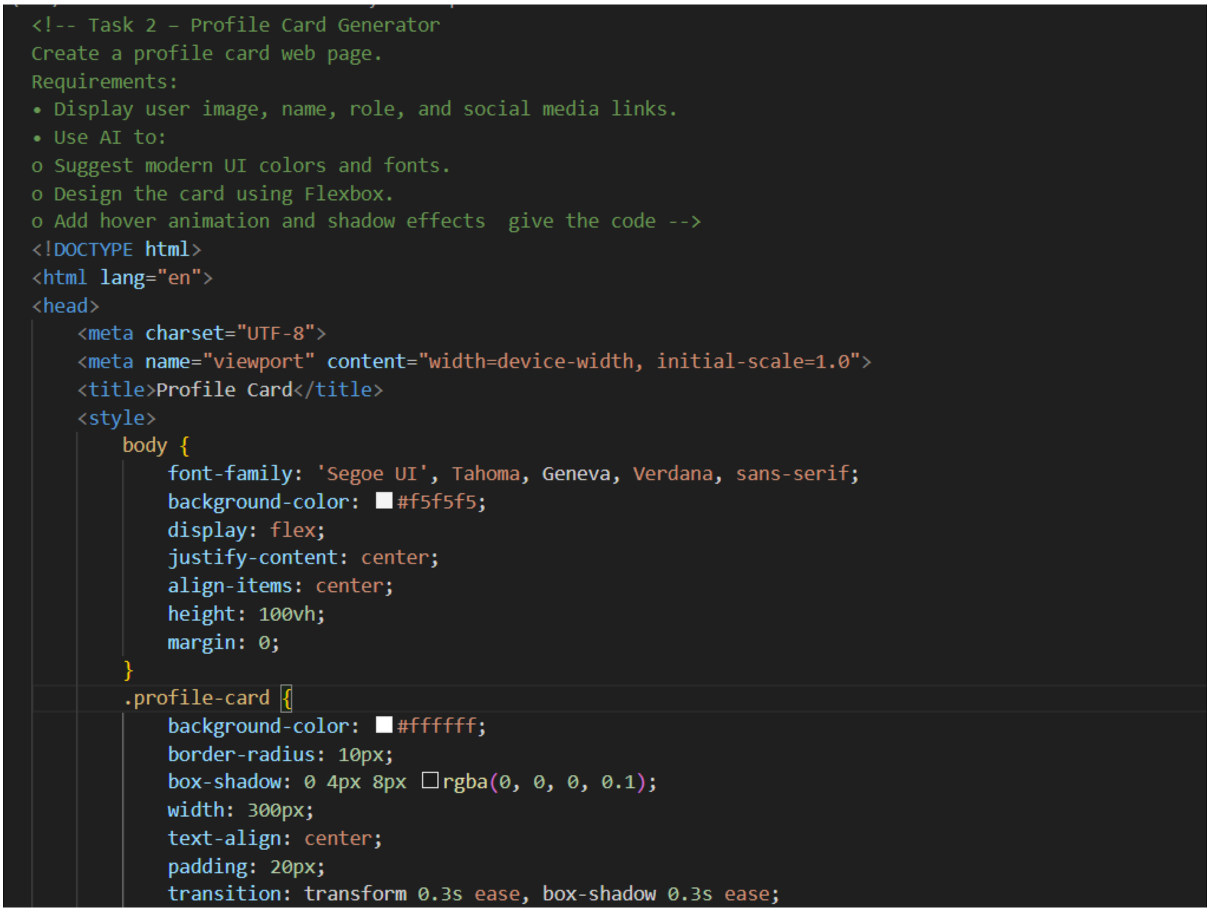
Requirements:

- Display user image, name, role, and social media links.
- Use AI to:

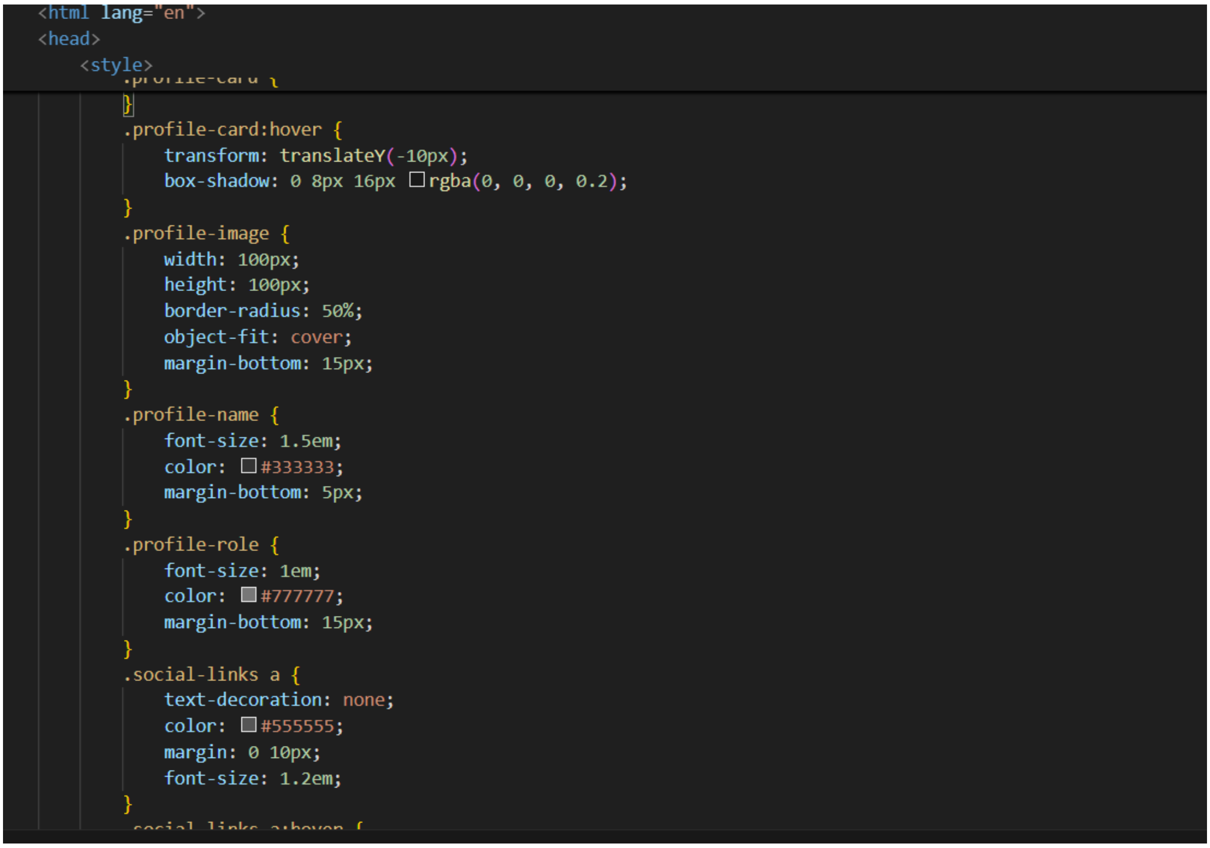
o Suggest modern UI colors and fonts.

o Design the card using Flexbox.

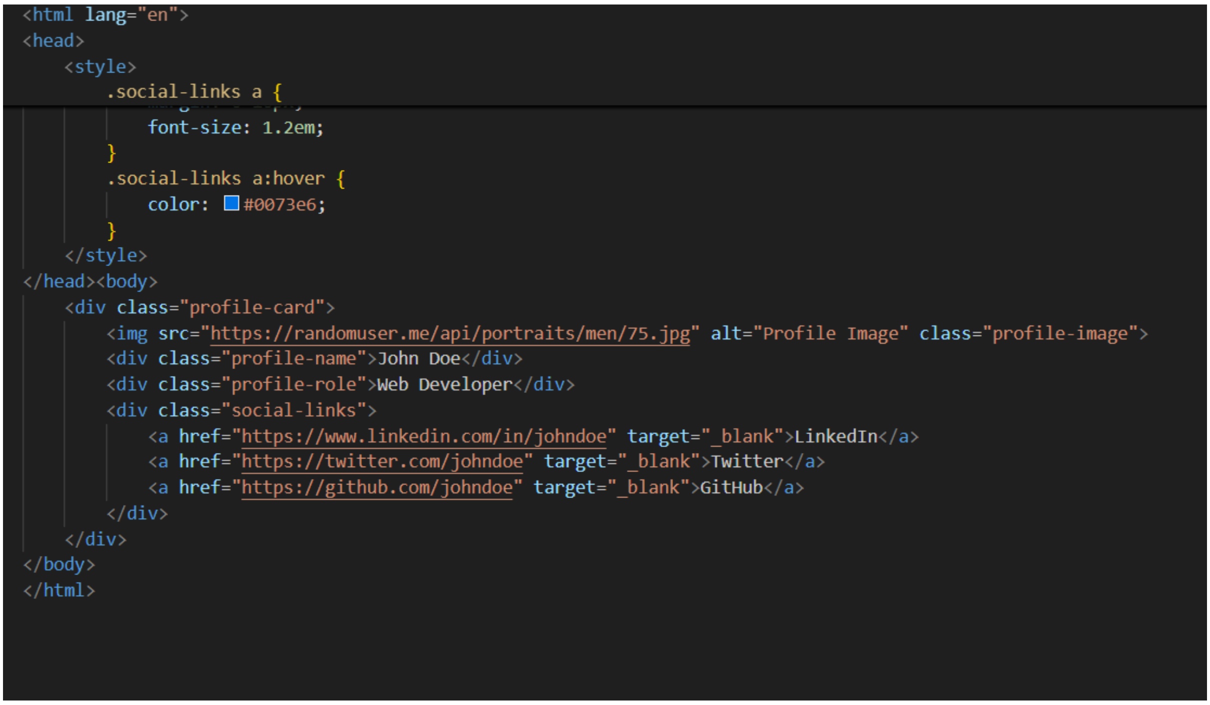
o Add hover animation and shadow effects



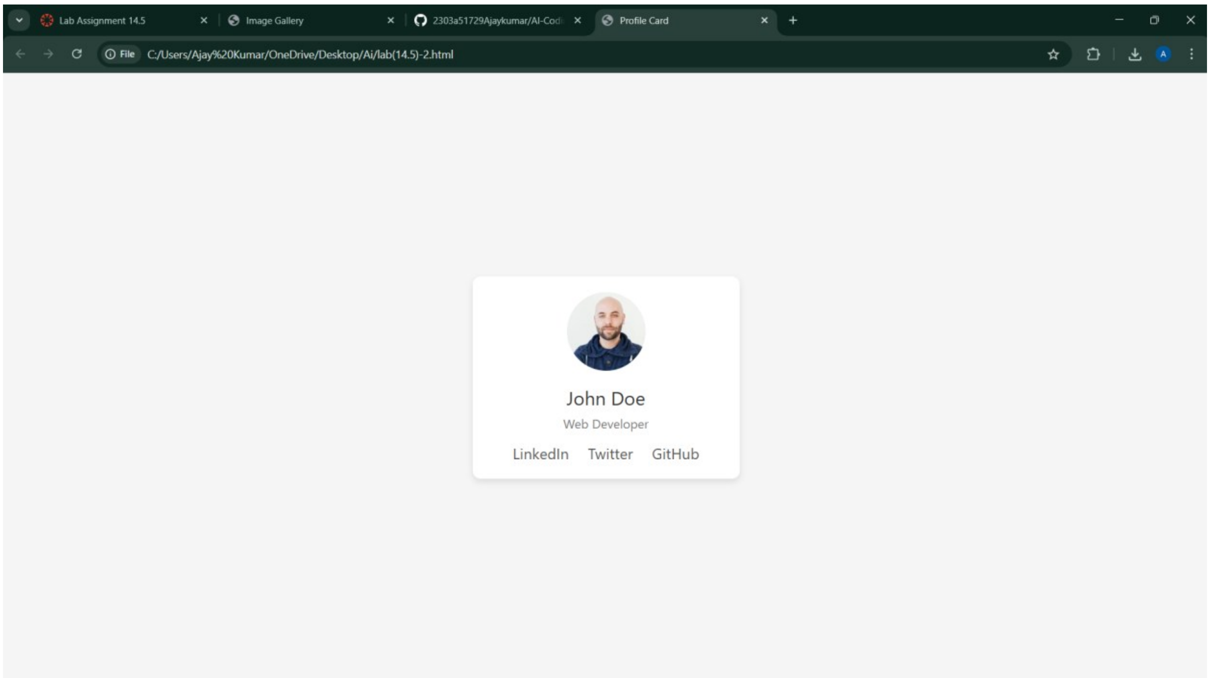
```
<!-- Task 2 – Profile Card Generator
Create a profile card web page.
Requirements:
• Display user image, name, role, and social media links.
• Use AI to:
o Suggest modern UI colors and fonts.
o Design the card using Flexbox.
o Add hover animation and shadow effects  give the code -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Profile Card</title>
    <style>
        body {
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
            background-color: #f5f5f5;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
        }
        .profile-card {
            background-color: #ffffff;
            border-radius: 10px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            width: 300px;
            text-align: center;
            padding: 20px;
            transition: transform 0.3s ease, box-shadow 0.3s ease;
```



```
<html lang="en">
<head>
    <style>
        .profile-card {
        }
        .profile-card:hover {
            transform: translateY(-10px);
            box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
        }
        .profile-image {
            width: 100px;
            height: 100px;
            border-radius: 50%;
            object-fit: cover;
            margin-bottom: 15px;
        }
        .profile-name {
            font-size: 1.5em;
            color: #333333;
            margin-bottom: 5px;
        }
        .profile-role {
            font-size: 1em;
            color: #777777;
            margin-bottom: 15px;
        }
        .social-links a {
            text-decoration: none;
            color: #555555;
            margin: 0 10px;
            font-size: 1.2em;
        }
        .social-links a:hover {
```



```
<html lang="en">
<head>
    <style>
        .social-links a {
            font-size: 1.2em;
        }
        .social-links a:hover {
            color: #0073e6;
        }
    </style>
</head><body>
    <div class="profile-card">
        <img src="https://randomuser.me/api/portraits/men/75.jpg" alt="Profile Image" class="profile-image">
        <div class="profile-name">John Doe</div>
        <div class="profile-role">Web Developer</div>
        <div class="social-links">
            <a href="https://www.linkedin.com/in/johndoe" target="_blank">LinkedIn</a>
            <a href="https://twitter.com/johndoe" target="_blank">Twitter</a>
            <a href="https://github.com/johndoe" target="_blank">GitHub</a>
        </div>
    </div>
</body>
</html>
```



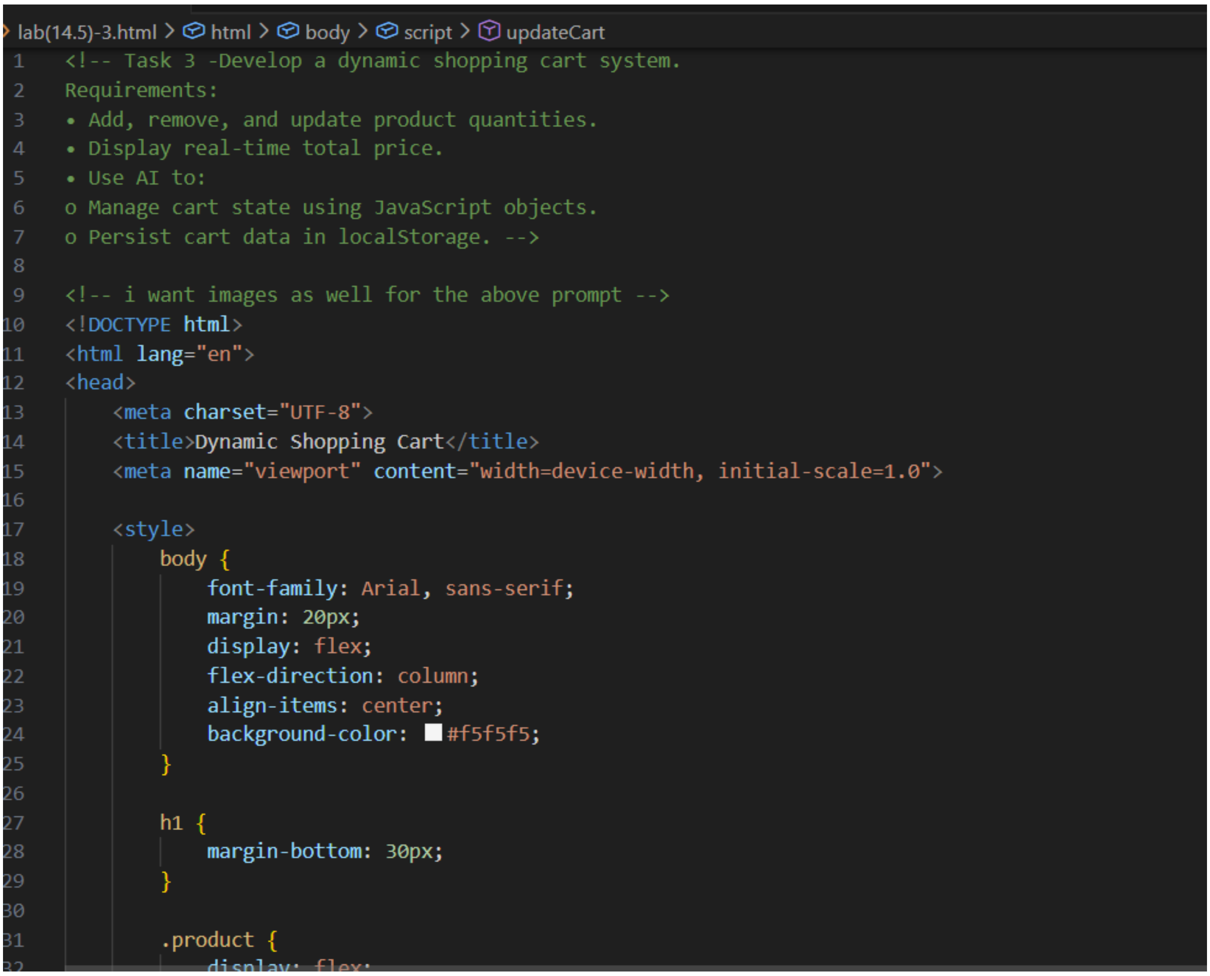
Task 3 -Develop a dynamic shopping cart system.

Requirements:

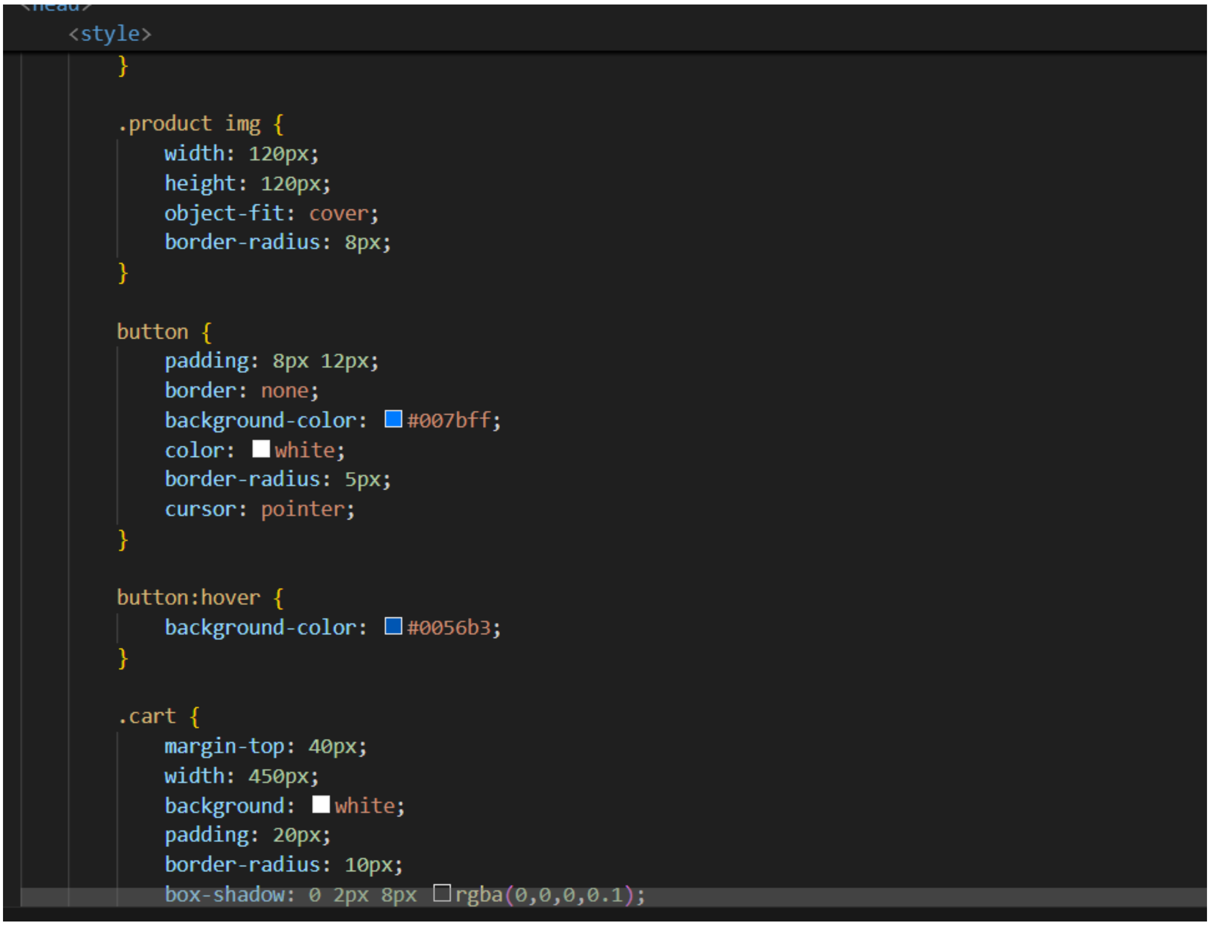
- Add, remove, and update product quantities.
- Display real-time total price.
- Use AI to:

o Manage cart state using JavaScript objects.

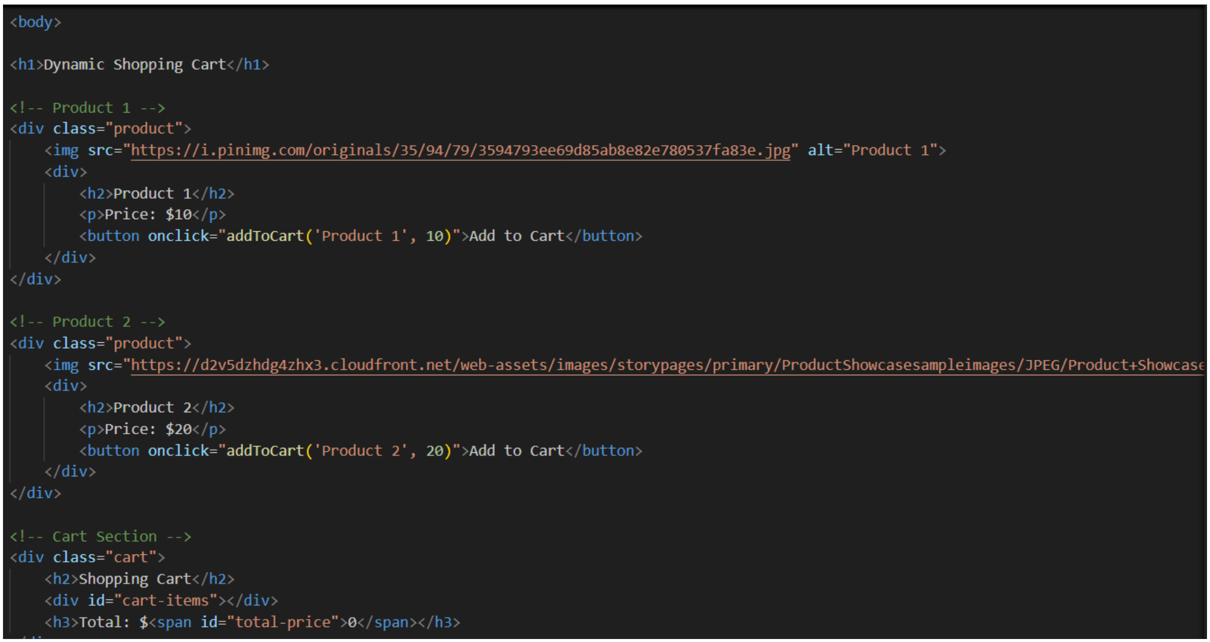
o Persist cart data in localStorage.



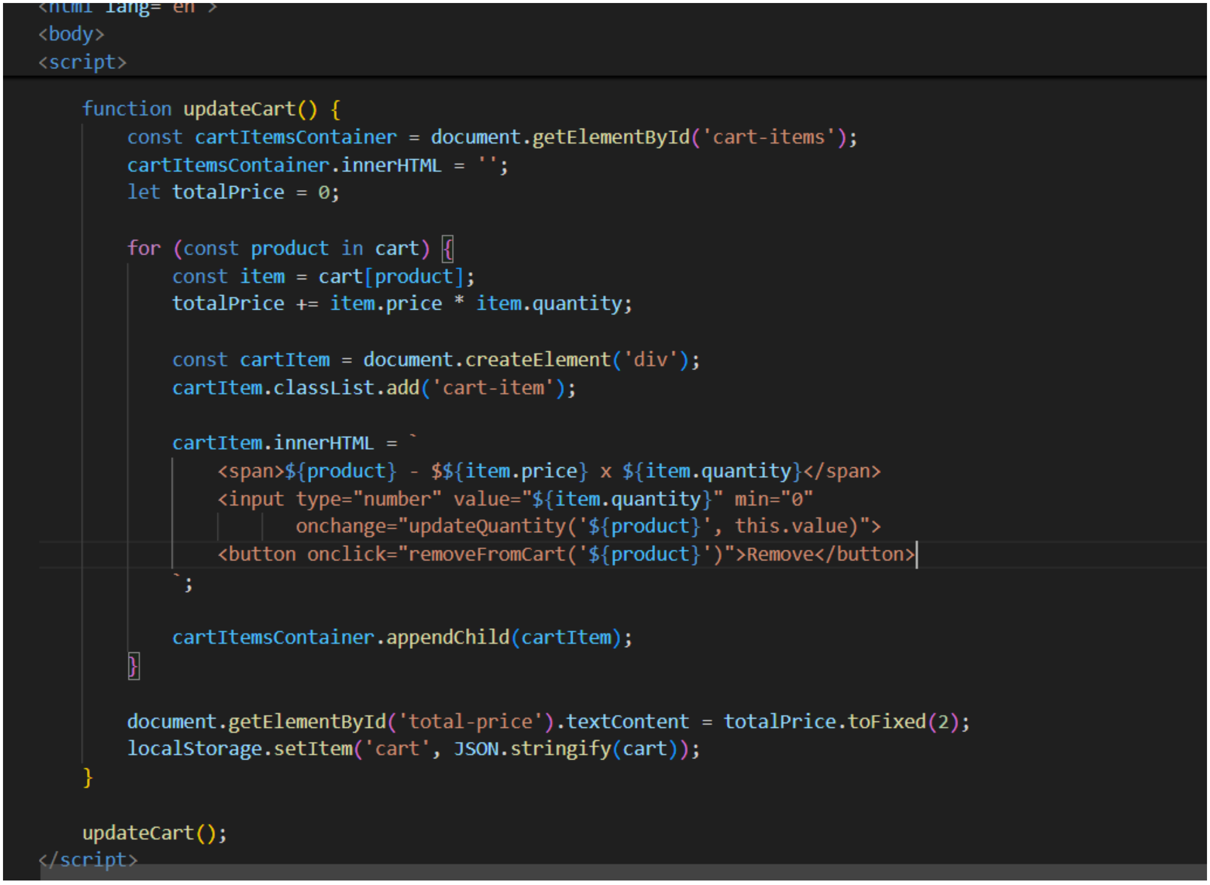
```
<!-- Task 3 -Develop a dynamic shopping cart system.
Requirements:
• Add, remove, and update product quantities.
• Display real-time total price.
• Use AI to:
o Manage cart state using JavaScript objects.
o Persist cart data in localStorage. -->

<!-- i want images as well for the above prompt -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Dynamic Shopping Cart</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
            display: flex;
            flex-direction: column;
            align-items: center;
            background-color: #f5f5f5;
        }

        h1 {
            margin-bottom: 30px;
        }

        .product {
            display: flex;
```



```
<head>
    <style>
        }

        .product img {
            width: 120px;
            height: 120px;
            object-fit: cover;
            border-radius: 8px;
        }

        button {
            padding: 8px 12px;
            border: none;
            background-color: #007bff;
            color: white;
            border-radius: 5px;
            cursor: pointer;
        }

        button:hover {
            background-color: #0056b3;
        }

        .cart {
            margin-top: 40px;
            width: 450px;
            background: white;
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0 2px 8px rgba(0,0,0,0.1);
```

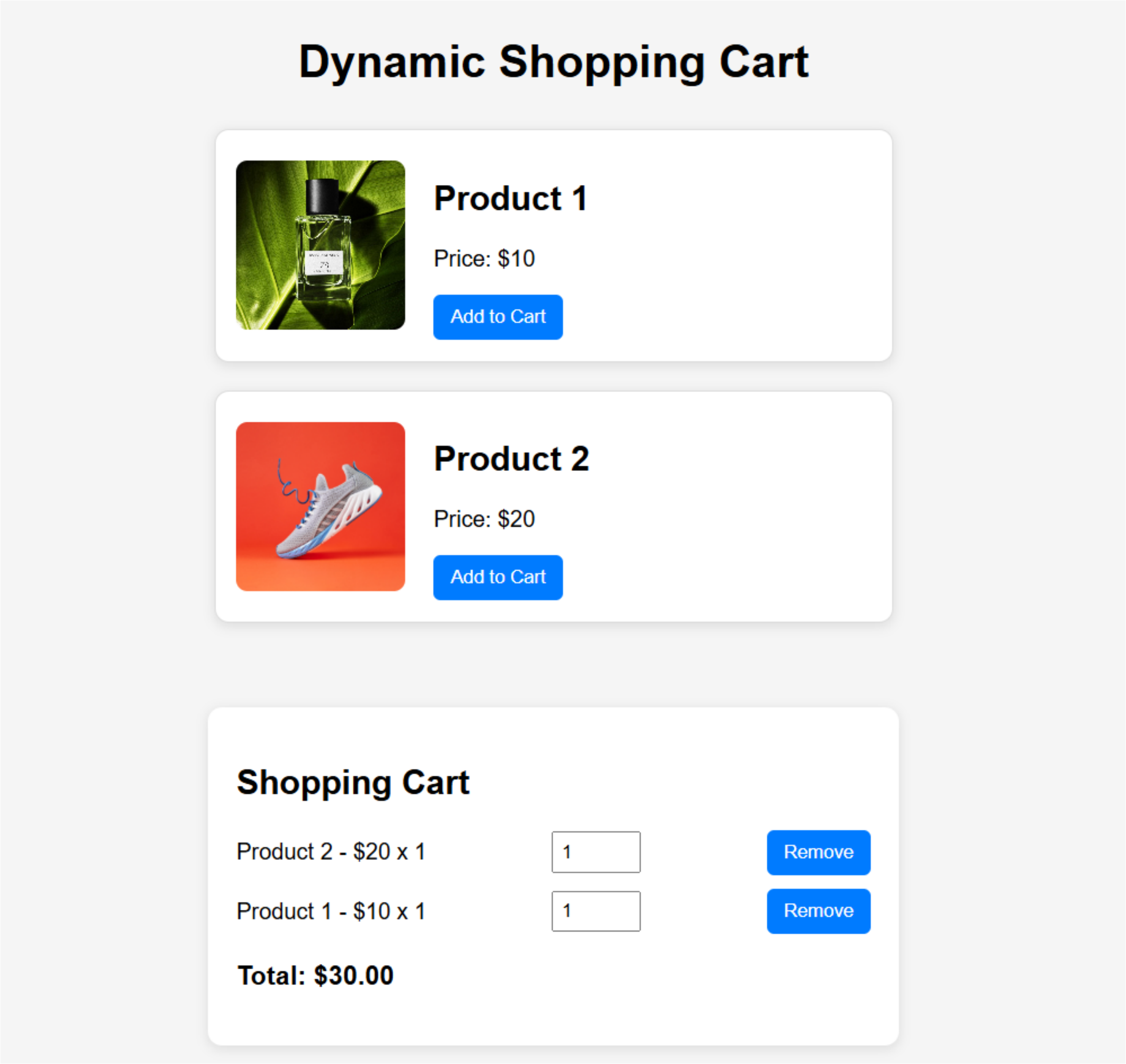


```
<body>

<h1>Dynamic Shopping Cart</h1>

<!-- Product 1 -->
<div class="product">
    <img src="https://i.pinimg.com/originals/35/94/79/3594793ee69d85ab8e82e780537fa83e.jpg" alt="Product 1">
    <div>
        <h2>Product 1</h2>
        <p>Price: $10</p>
        <button onclick="addToCart('Product 1', 10)">Add to Cart</button>
    </div>
</div>

<!-- Product 2 -->
<div class="product">
    <img src="https://d2v5dzhdg4zhx3.cloudfront.net/web-assets/images/storypages/primary/ProductShowcasesampleimages/JPEG/Product+Showcase
    <div>
        <h2>Product 2</h2>
        <p>Price: $20</p>
        <button onclick="addToCart('Product 2', 20)">Add to Cart</button>
    </div>
</div>

<!-- Cart Section -->
<div class="cart">
    <h2>Shopping Cart</h2>
    <div id="cart-items"></div>
    <h3>Total: $<span id="total-price">0</span></h3>
```



```
<html lang="en">
<body>
<script>

    function updateCart() {
        const cartItemsContainer = document.getElementById('cart-items');
        cartItemsContainer.innerHTML = '';
        let totalPrice = 0;

        for (const product in cart) {
            const item = cart[product];
            totalPrice += item.price * item.quantity;

            const cartItem = document.createElement('div');
            cartItem.classList.add('cart-item');

            cartItem.innerHTML = `
                <span>${product} - $${item.price} x ${item.quantity}</span>
                <input type="number" value="${item.quantity}" min="0"
                       onchange="updateQuantity('${product}', this.value)">
                <button onclick="removeFromCart('${product}')">Remove</button>
            `;

            cartItemsContainer.appendChild(cartItem);
        }

        document.getElementById('total-price').textContent = totalPrice.toFixed(2);
        localStorage.setItem('cart', JSON.stringify(cart));
    }

    updateCart();
</script>
```

Output



Dynamic Shopping Cart

Product 1

Price: $10

Product 2

Price: $20

Shopping Cart

Product 2 - $20 x 1 1

Product 1 - $10 x 1 1

Total: $30.00

Task 4 – Role-Based Access Control Application

Create a role-based web application.

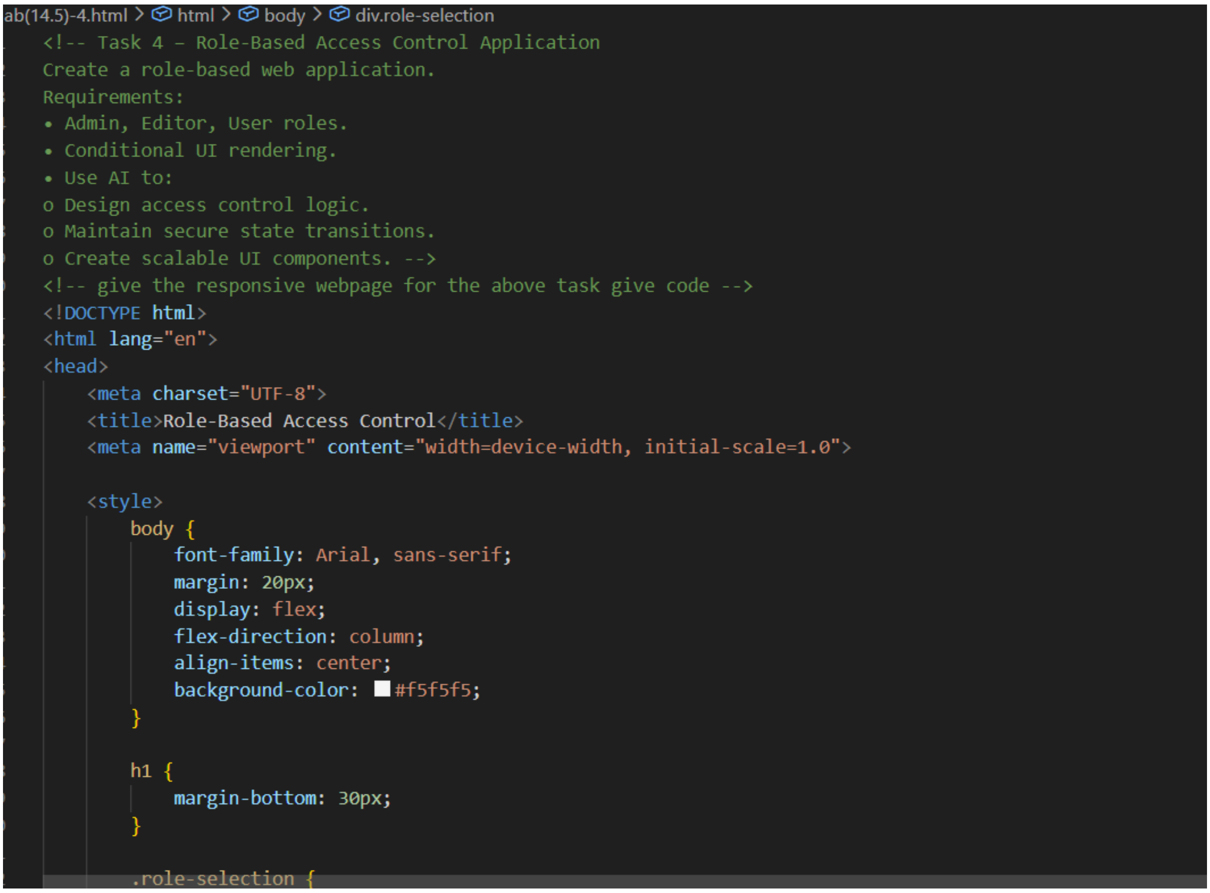
Requirements:

- Admin, Editor, User roles.
- Conditional UI rendering.
- Use AI to:

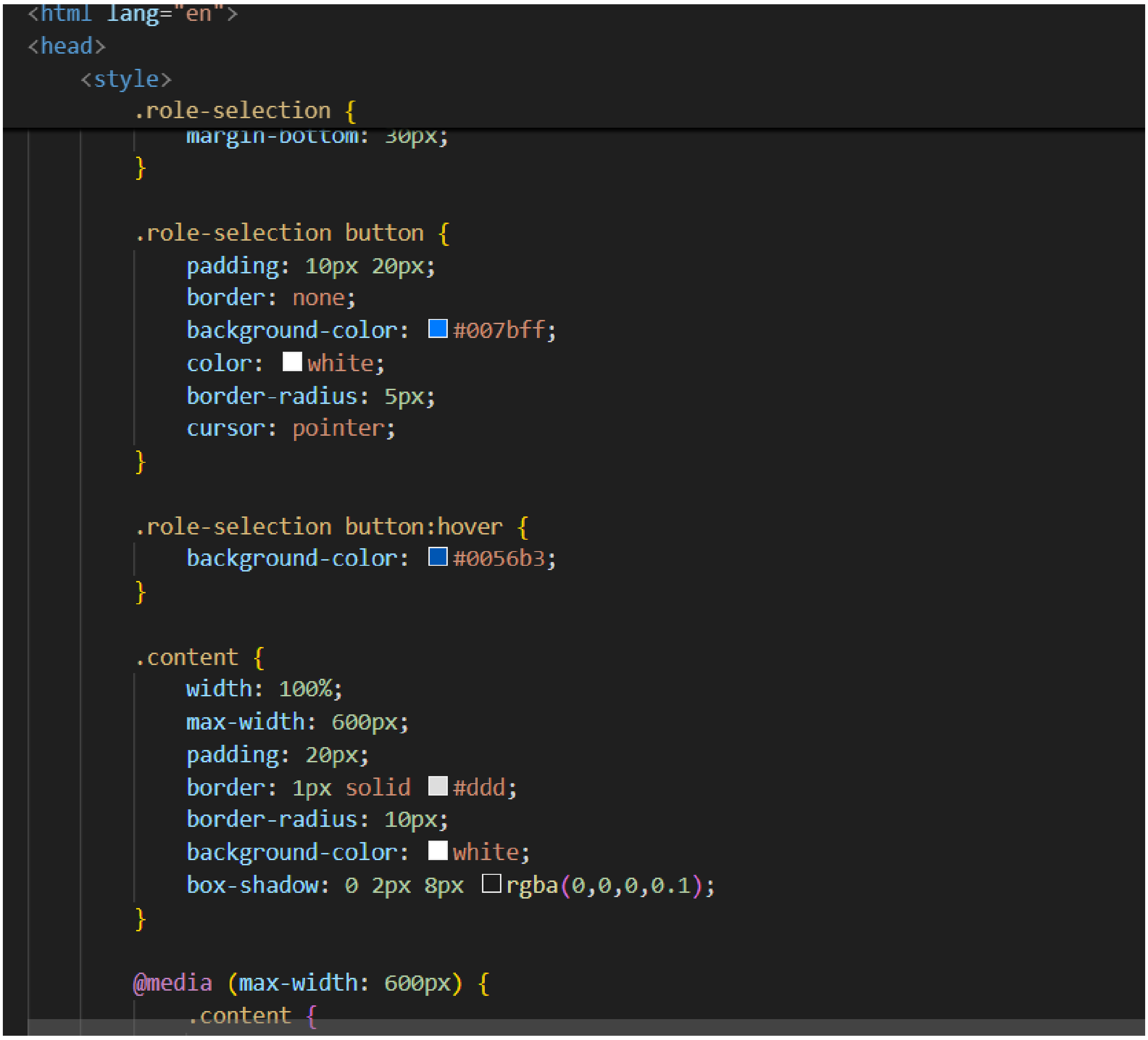
o Design access control logic.

o Maintain secure state transitions.

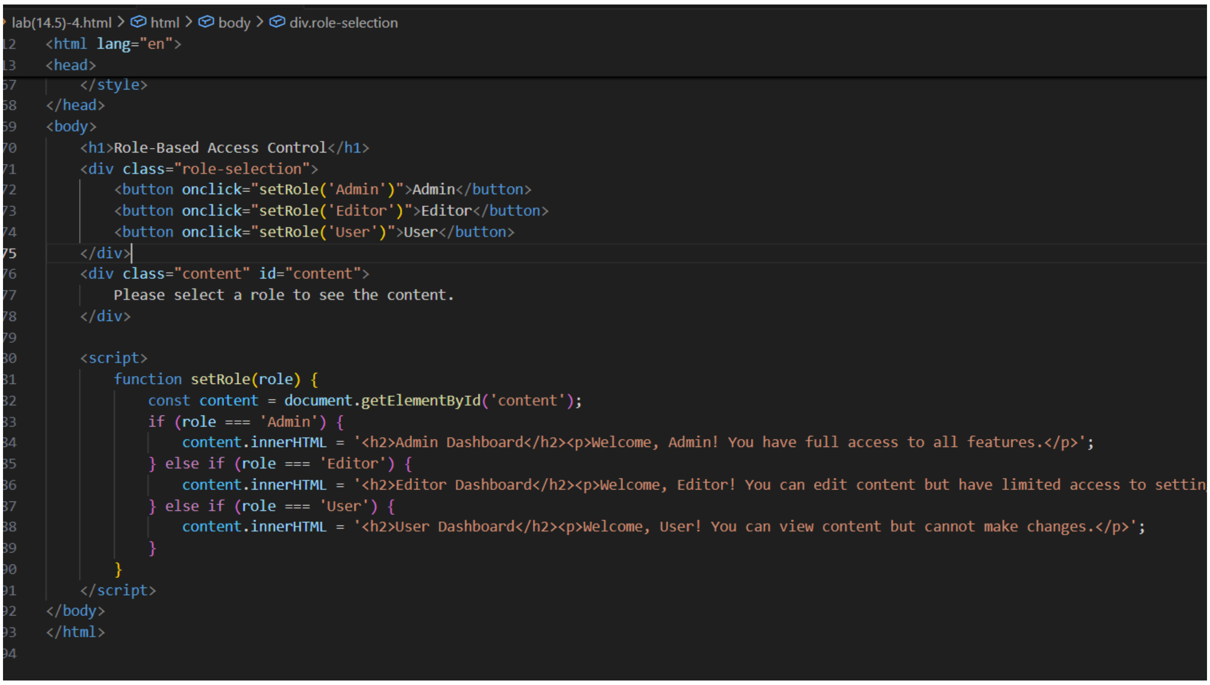
o Create scalable UI components.



```
<!-- Task 4 – Role-Based Access Control Application
Create a role-based web application.
Requirements:
• Admin, Editor, User roles.
• Conditional UI rendering.
• Use AI to:
o Design access control logic.
o Maintain secure state transitions.
o Create scalable UI components. -->
<!-- give the responsive webpage for the above task give code -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Role-Based Access Control</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
            display: flex;
            flex-direction: column;
            align-items: center;
            background-color: #f5f5f5;
        }

        h1 {
            margin-bottom: 30px;
        }

        .role-selection {
```

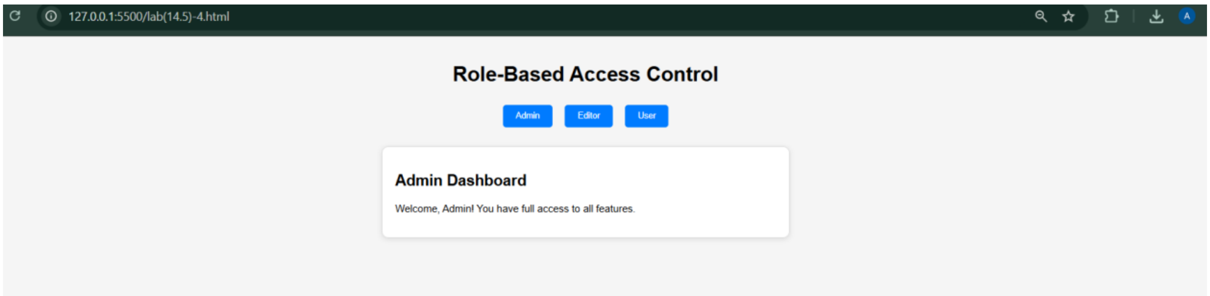


```
<html lang="en">
<head>
    <style>
        .role-selection {
            margin-bottom: 30px;
        }

        .role-selection button {
            padding: 10px 20px;
            border: none;
            background-color: #007bff;
            color: white;
            border-radius: 5px;
            cursor: pointer;
        }

        .role-selection button:hover {
            background-color: #0056b3;
        }

        .content {
            width: 100%;
            max-width: 600px;
            padding: 20px;
            border: 1px solid #ddd;
            border-radius: 10px;
            background-color: white;
            box-shadow: 0 2px 8px rgba(0,0,0,0.1);
        }

        @media (max-width: 600px) {
            .content {
```



```
<html lang="en">
<head>
    </style>
</head>
<body>
    <h1>Role-Based Access Control</h1>
    <div class="role-selection">
        <button onclick="setRole('Admin')">Admin</button>
        <button onclick="setRole('Editor')">Editor</button>
        <button onclick="setRole('User')">User</button>
    </div>
    <div class="content" id="content">
        Please select a role to see the content.
    </div>

    <script>
        function setRole(role) {
            const content = document.getElementById('content');
            if (role === 'Admin') {
                content.innerHTML = '<h2>Admin Dashboard</h2><p>Welcome, Admin! You have full access to all features.</p>';
            } else if (role === 'Editor') {
                content.innerHTML = '<h2>Editor Dashboard</h2><p>Welcome, Editor! You can edit content but have limited access to settin
            } else if (role === 'User') {
                content.innerHTML = '<h2>User Dashboard</h2><p>Welcome, User! You can view content but cannot make changes.</p>';
            }
        }
    </script>
</body>
</html>
```

Output:



Role-Based Access Control

Admin Dashboard

Welcome, Admin! You have full access to all features.

Task 5 – Multi-Language Website

Design a multilingual website.

Requirements:

- Switch languages dynamically.

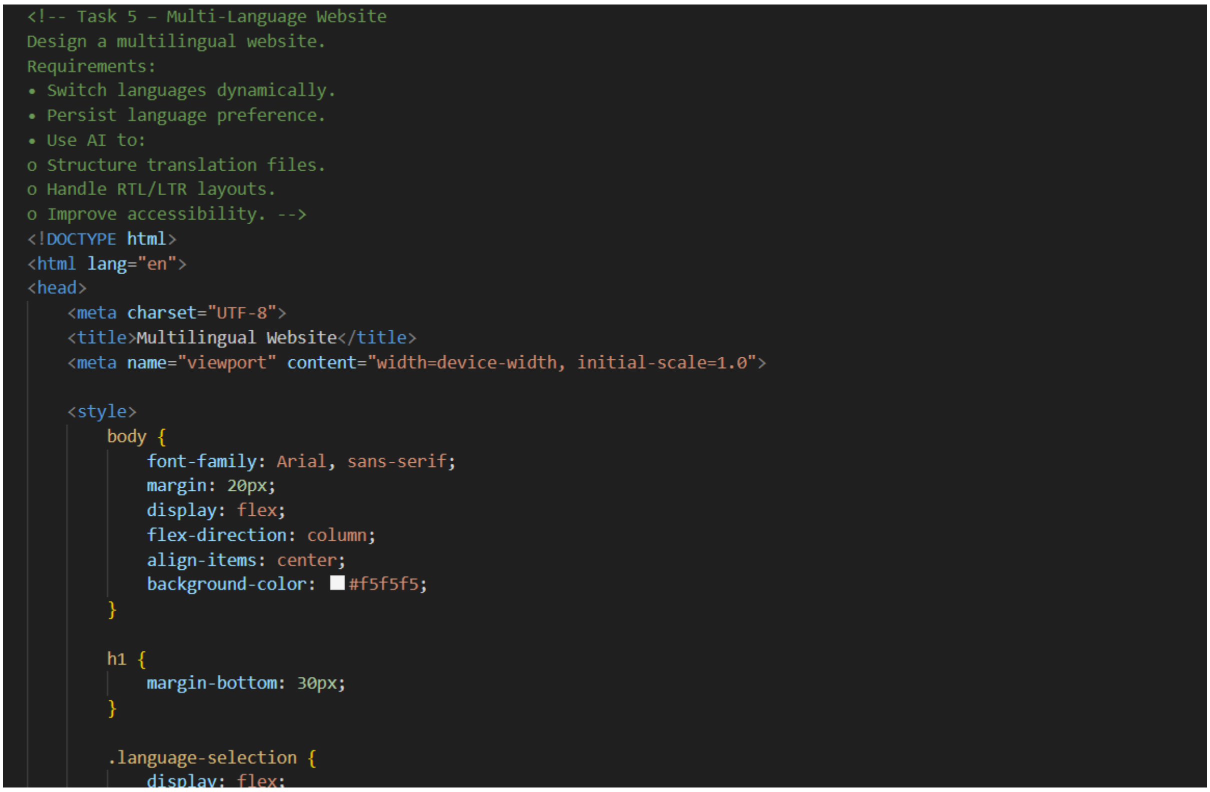
- Persist language preference.

- Use AI to:

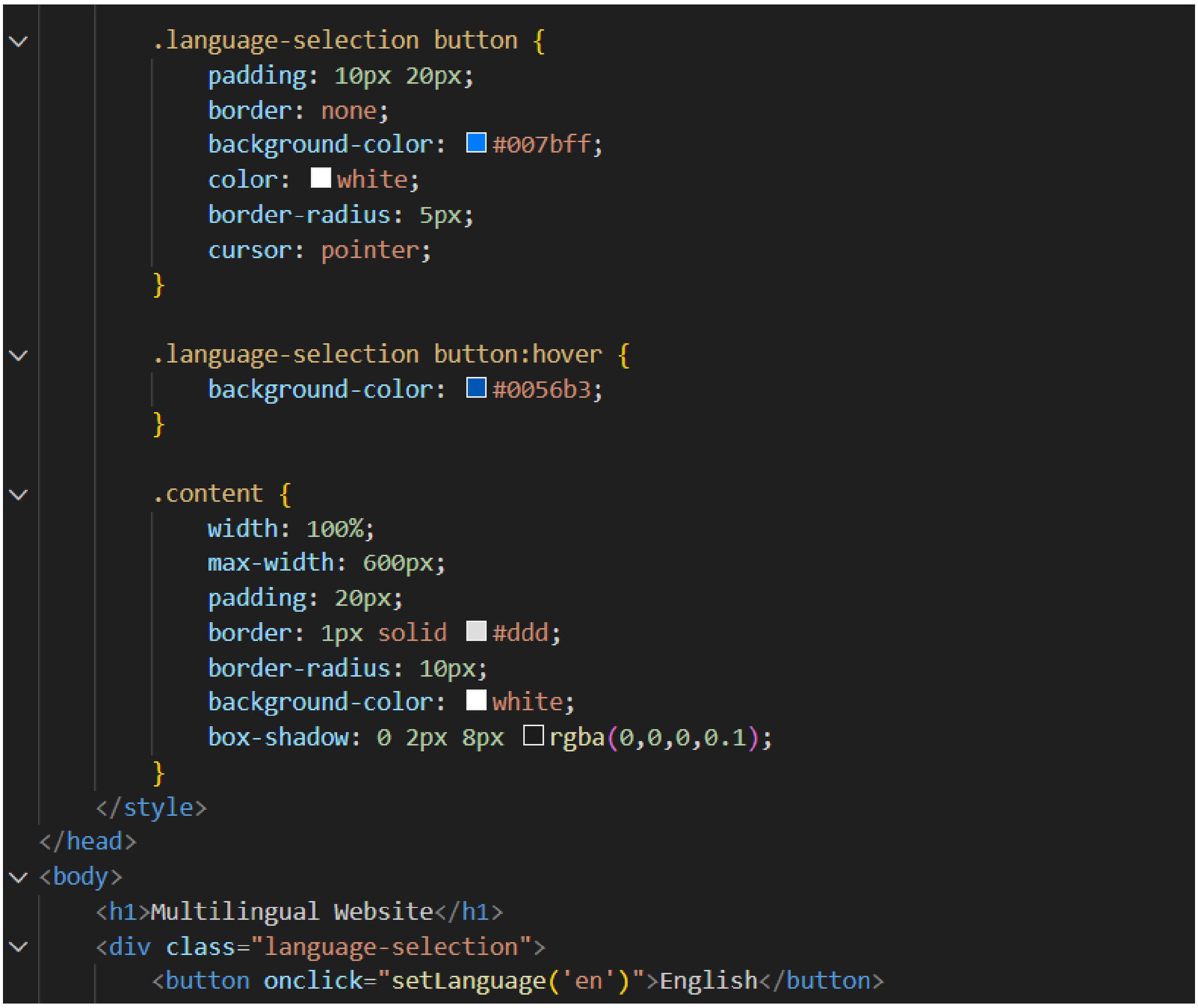
o Structure translation files.

o Handle RTL/LTR layouts.

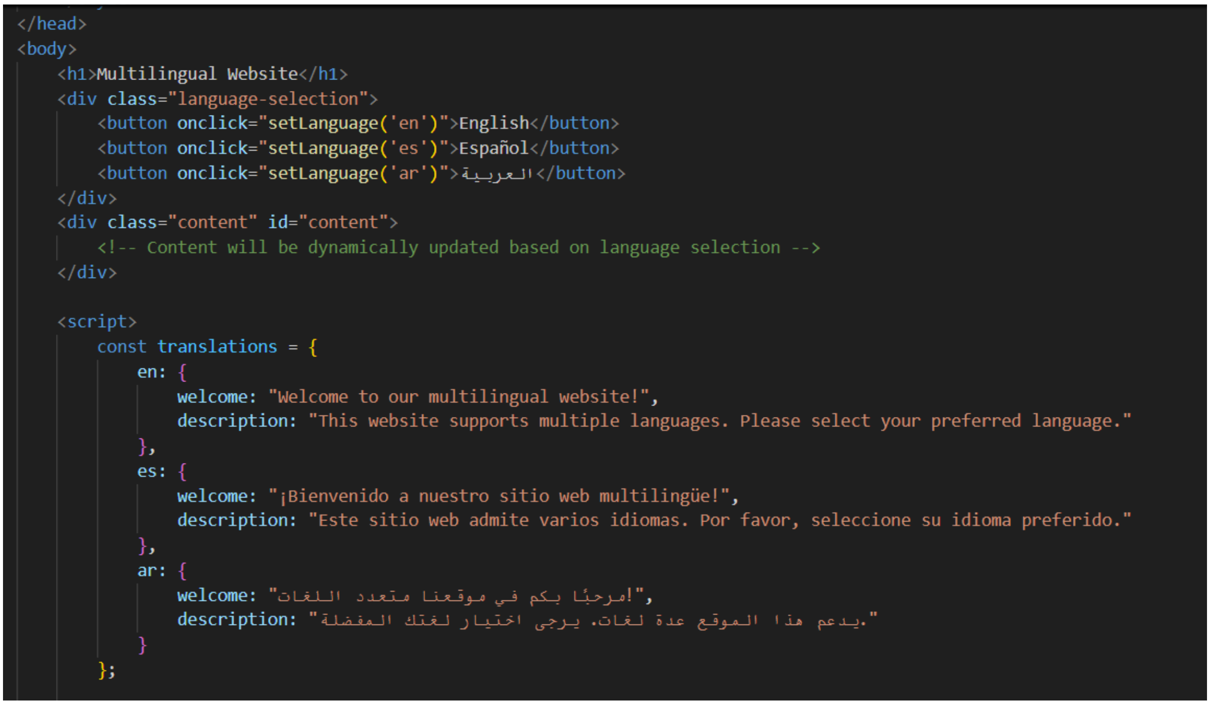
o Improve accessibility.



```
<!-- Task 5 – Multi-Language Website
Design a multilingual website.
Requirements:
• Switch languages dynamically.
• Persist language preference.
• Use AI to:
o Structure translation files.
o Handle RTL/LTR layouts.
o Improve accessibility. -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Multilingual Website</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
            display: flex;
            flex-direction: column;
            align-items: center;
            background-color: #f5f5f5;
        }

        h1 {
            margin-bottom: 30px;
        }

        .language-selection {
            display: flex;
```

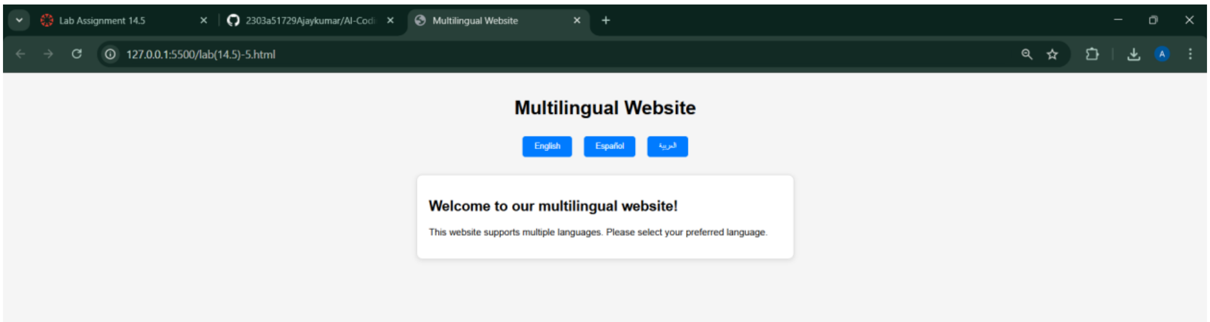


```
        .language-selection button {
            padding: 10px 20px;
            border: none;
            background-color: #007bff;
            color: white;
            border-radius: 5px;
            cursor: pointer;
        }

        .language-selection button:hover {
            background-color: #0056b3;
        }

        .content {
            width: 100%;
            max-width: 600px;
            padding: 20px;
            border: 1px solid #ddd;
            border-radius: 10px;
            background-color: white;
            box-shadow: 0 2px 8px rgba(0,0,0,0.1);
        }
    </style>
</head>
<body>
    <h1>Multilingual Website</h1>
    <div class="language-selection">
        <button onclick="setLanguage('en')">English</button>
```



```
</head>
<body>
    <h1>Multilingual Website</h1>
    <div class="language-selection">
        <button onclick="setLanguage('en')">English</button>
        <button onclick="setLanguage('es')">Español</button>
        <button onclick="setLanguage('ar')">العربية</button>
    </div>
    <div class="content" id="content">
        <!-- Content will be dynamically updated based on language selection -->
    </div>

    <script>
        const translations = {
            en: {
                welcome: "Welcome to our multilingual website!",
                description: "This website supports multiple languages. Please select your preferred language."
            },
            es: {
                welcome: "¡Bienvenido a nuestro sitio web multilingüe!",
                description: "Este sitio web admite varios idiomas. Por favor, seleccione su idioma preferido."
            },
            ar: {
                welcome: "مرحبًا بكم في موقعنا متعدد اللغات!",
                description: "يدعم هذا الموقع عدة لغات. يرجى اختيار لغتك المفضلة."
            }
        };
```

Output:



Multilingual Website

Welcome to our multilingual website!

This website supports multiple languages. Please select your preferred language.

Task 6 – Online Examination System

Create an online exam interface.

Requirements:

• Timer, questions, and submission.

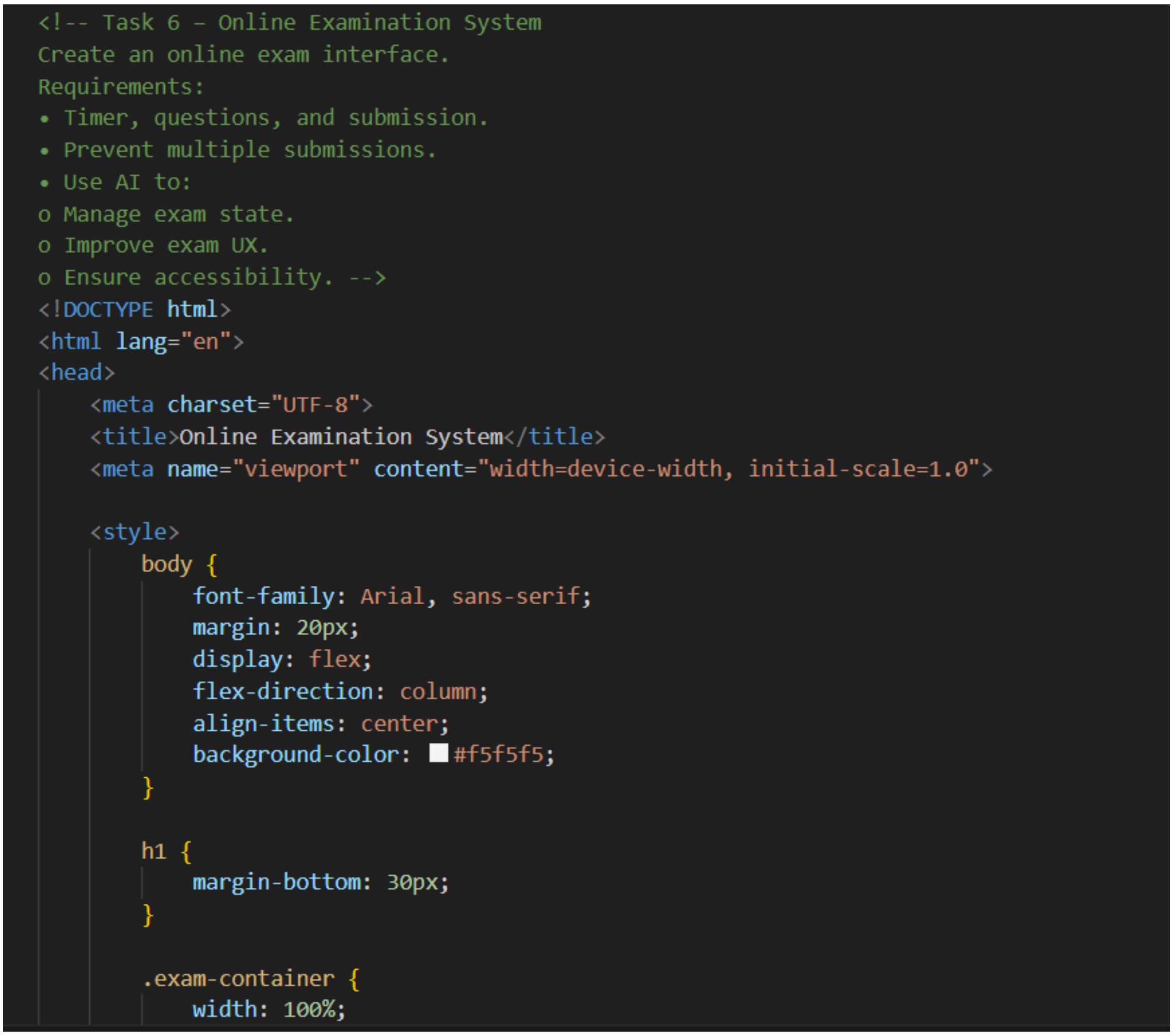
• Prevent multiple submissions.

• Use AI to:

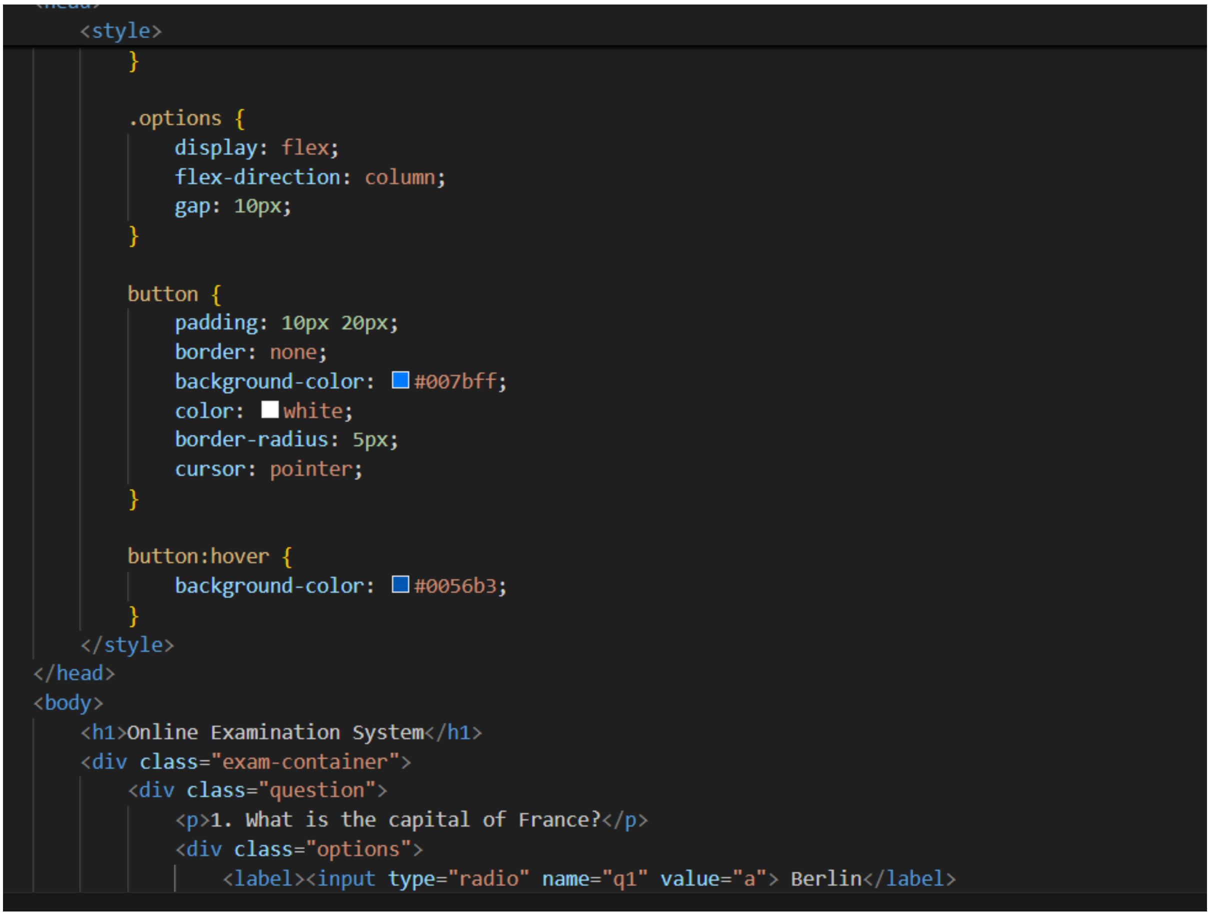
o Manage exam state.

o Improve exam UX.

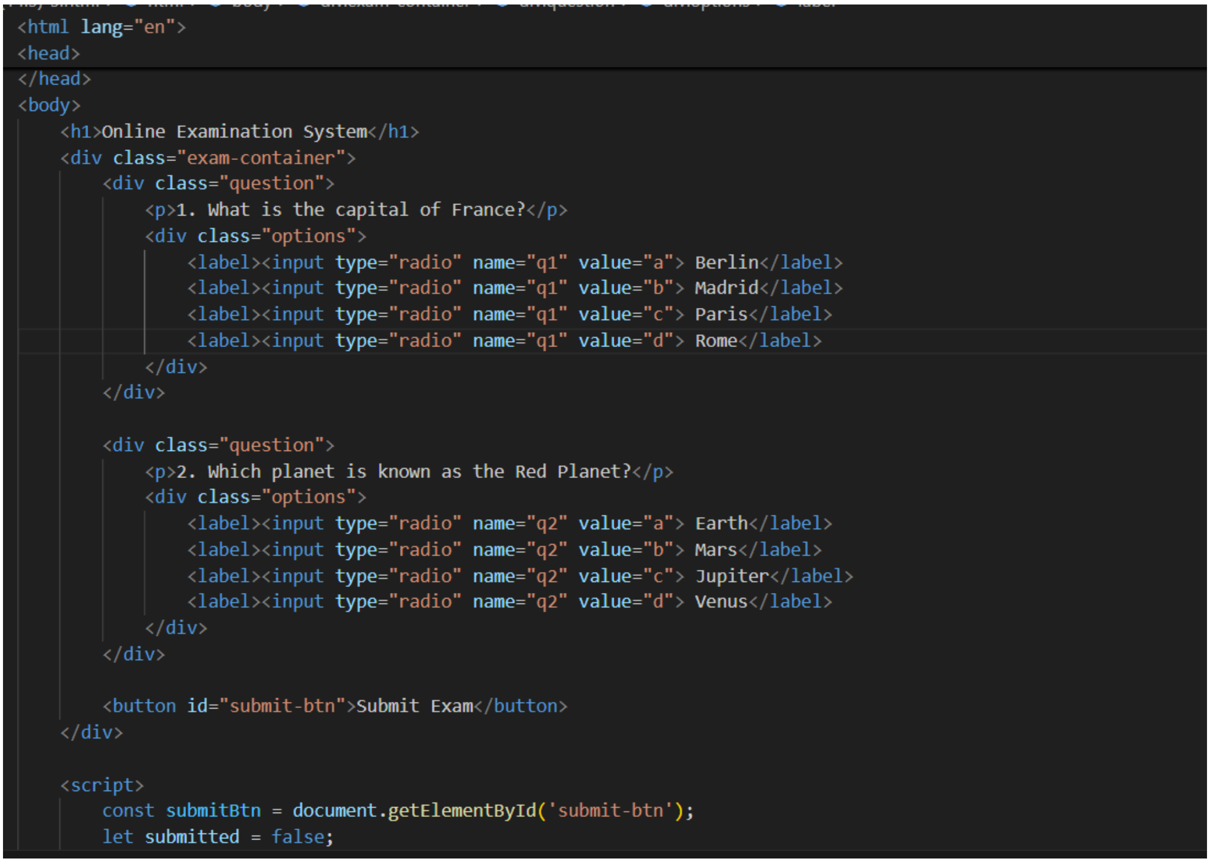
o Ensure accessibility.



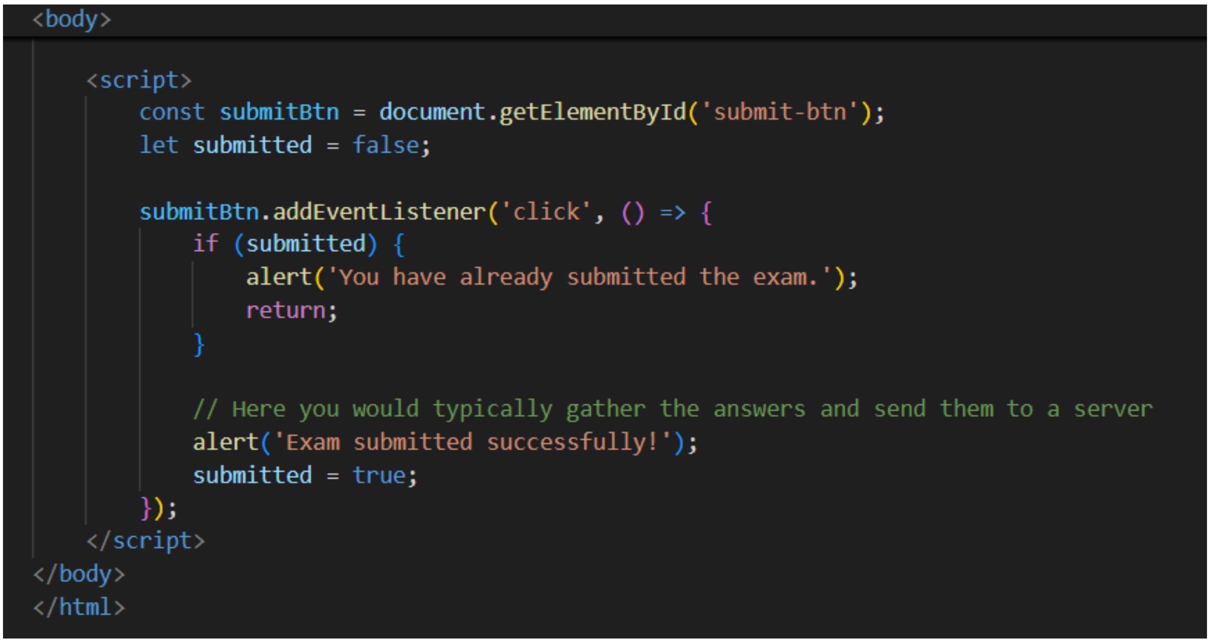
```html
<!-- Task 6 – Online Examination System
Create an online exam interface.
Requirements:
• Timer, questions, and submission.
• Prevent multiple submissions.
• Use AI to:
o Manage exam state.
o Improve exam UX.
o Ensure accessibility. -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Online Examination System</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
            display: flex;
            flex-direction: column;
            align-items: center;
            background-color: #f5f5f5;
        }

        h1 {
            margin-bottom: 30px;
        }

        .exam-container {
            width: 100%;
```



```
    <style>
        }

        .options {
            display: flex;
            flex-direction: column;
            gap: 10px;
        }

        button {
            padding: 10px 20px;
            border: none;
            background-color: #007bff;
            color: white;
            border-radius: 5px;
            cursor: pointer;
        }

        button:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
    <h1>Online Examination System</h1>
    <div class="exam-container">
        <div class="question">
            <p>1. What is the capital of France?</p>
            <div class="options">
                <label><input type="radio" name="q1" value="a"> Berlin</label>
```

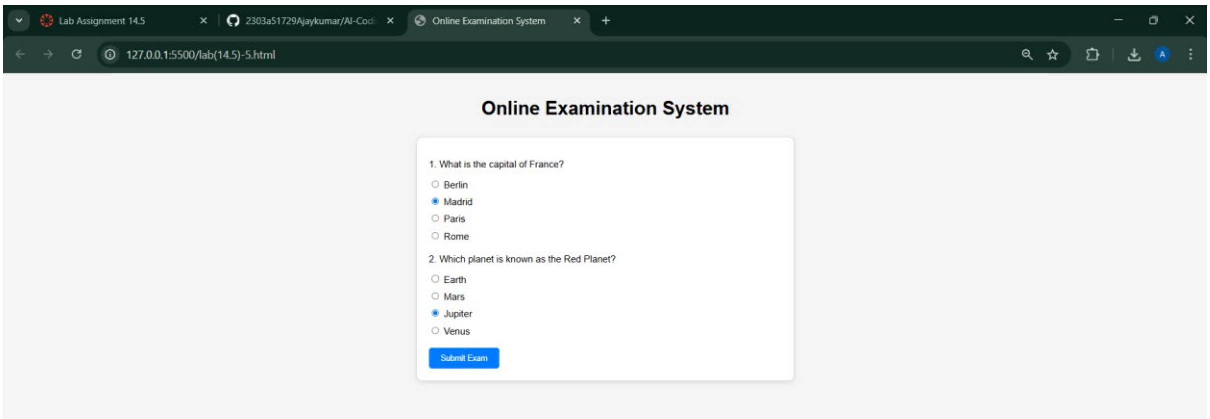


```
<html lang="en">
<head>
</head>
<body>
    <h1>Online Examination System</h1>
    <div class="exam-container">
        <div class="question">
            <p>1. What is the capital of France?</p>
            <div class="options">
                <label><input type="radio" name="q1" value="a"> Berlin</label>
                <label><input type="radio" name="q1" value="b"> Madrid</label>
                <label><input type="radio" name="q1" value="c"> Paris</label>
                <label><input type="radio" name="q1" value="d"> Rome</label>
            </div>
        </div>

        <div class="question">
            <p>2. Which planet is known as the Red Planet?</p>
            <div class="options">
                <label><input type="radio" name="q2" value="a"> Earth</label>
                <label><input type="radio" name="q2" value="b"> Mars</label>
                <label><input type="radio" name="q2" value="c"> Jupiter</label>
                <label><input type="radio" name="q2" value="d"> Venus</label>
            </div>
        </div>

        <button id="submit-btn">Submit Exam</button>
    </div>

    <script>
        const submitBtn = document.getElementById('submit-btn');
        let submitted = false;
```



```
<body>

    <script>
        const submitBtn = document.getElementById('submit-btn');
        let submitted = false;

        submitBtn.addEventListener('click', () => {
            if (submitted) {
                alert('You have already submitted the exam.');
                return;
            }

            // Here you would typically gather the answers and send them to a server
            alert('Exam submitted successfully!');
            submitted = true;
        });
    </script>
</body>
</html>
```

Output:


Lab Assignment 14.5
2303a51729Ajaykumar/AI-Codi
Online Examination System
127.0.0.1:5500/lab(14.5)-5.html

Online Examination System

1. What is the capital of France?

○ Berlin
● Madrid
○ Paris
○ Rome

2. Which planet is known as the Red Planet?

○ Earth
○ Mars
● Jupiter
○ Venus

Submit Exam