LAB ASSIGNMENT – 14.1

Name : G.Sudeeksha

Roll No: 2403A510E0

Course : AI Assisted Coding

Batch : 05

Branch : CSE

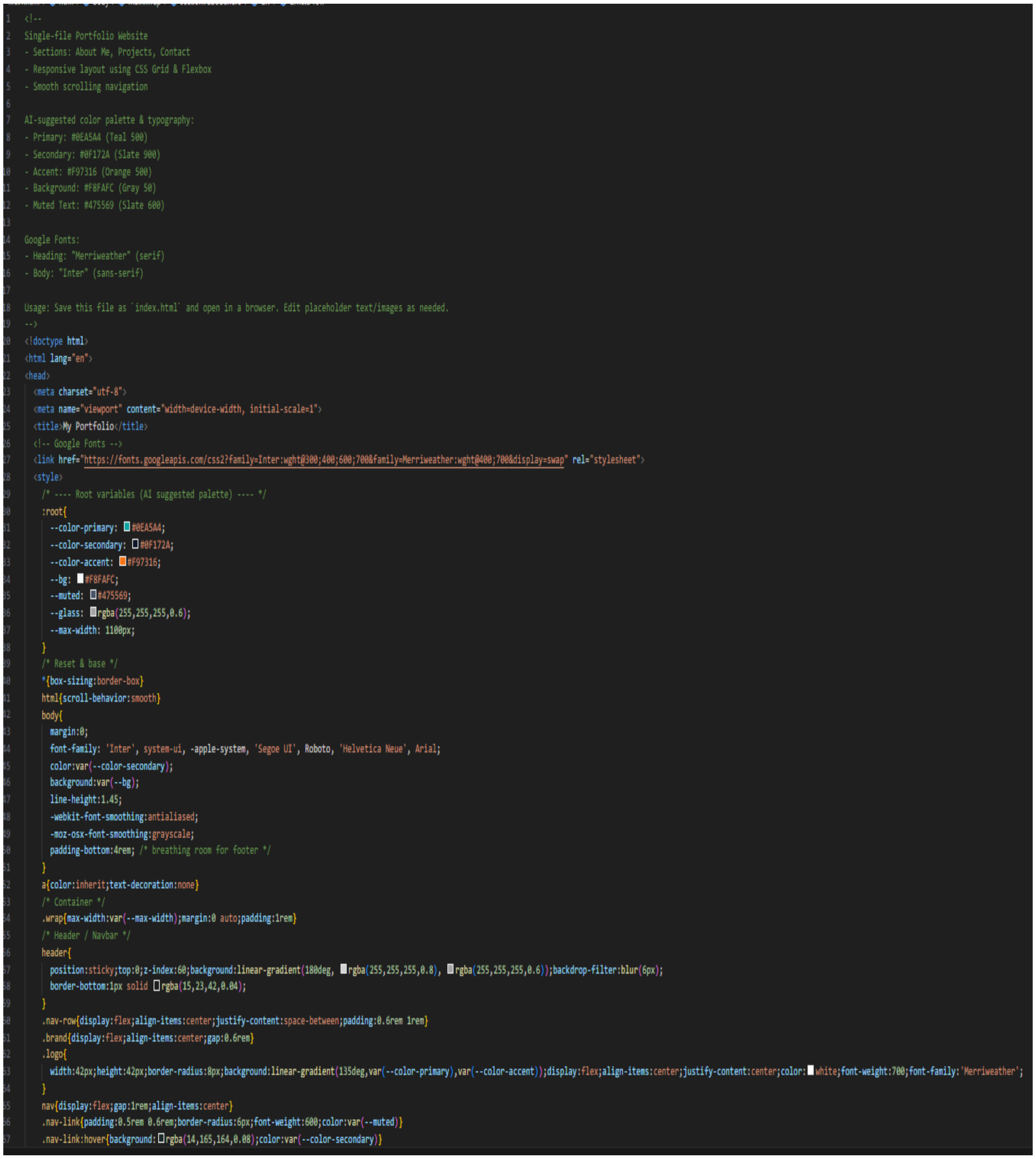
Task 1 – Portfolio Website Design

You are building a personal portfolio website to showcase your work.
Requirements:
• Create sections for About Me, Projects, and Contact.
• Use AI to:
o Suggest color palettes and typography.
o Create a responsive layout with Grid/Flexbox.
o Add smooth scrolling navigation.

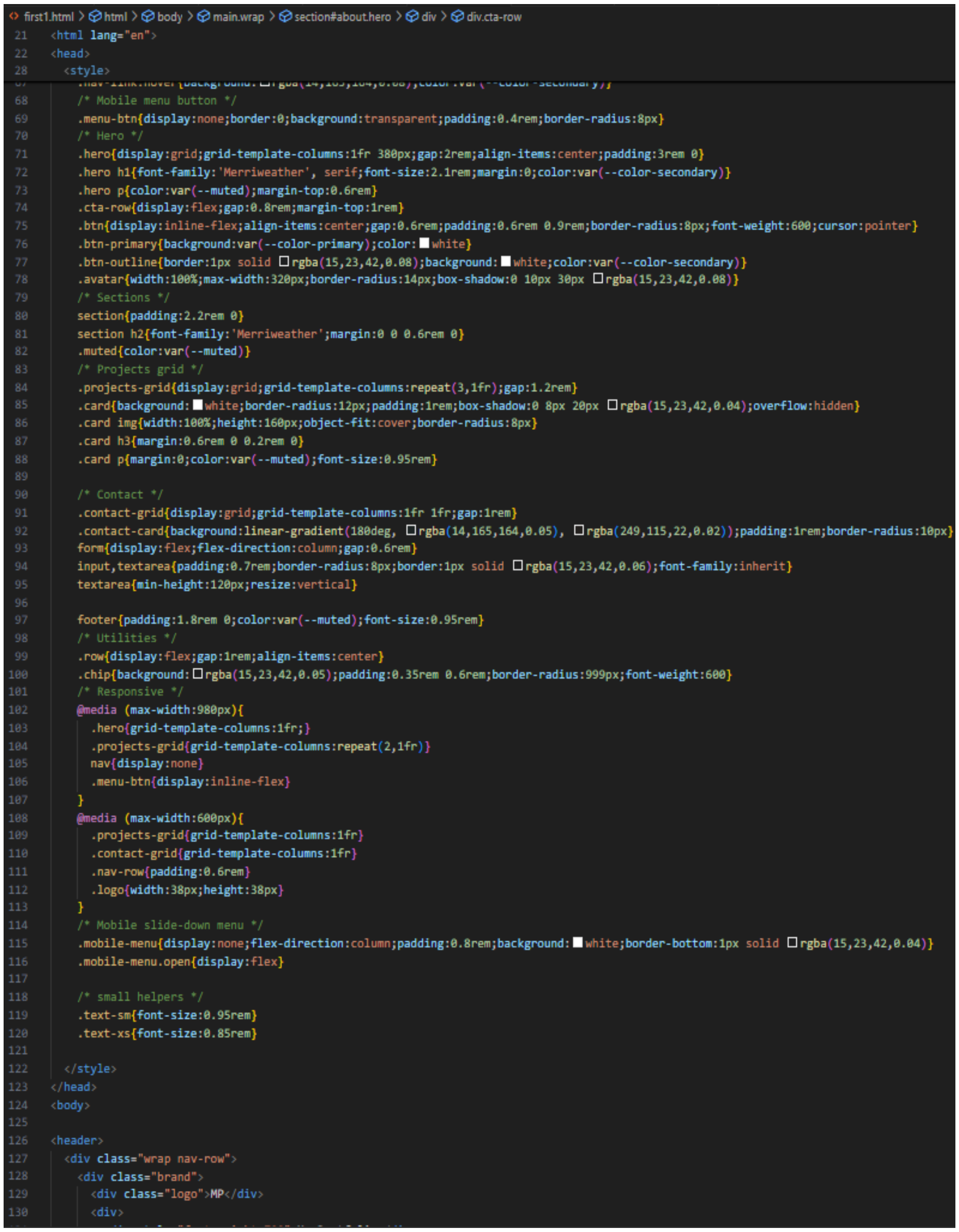
Prompt :

Create a responsive personal portfolio website with sections for About Me, Projects, and Contact using Grid/Flexbox, smooth scrolling, and modern design.

Code :

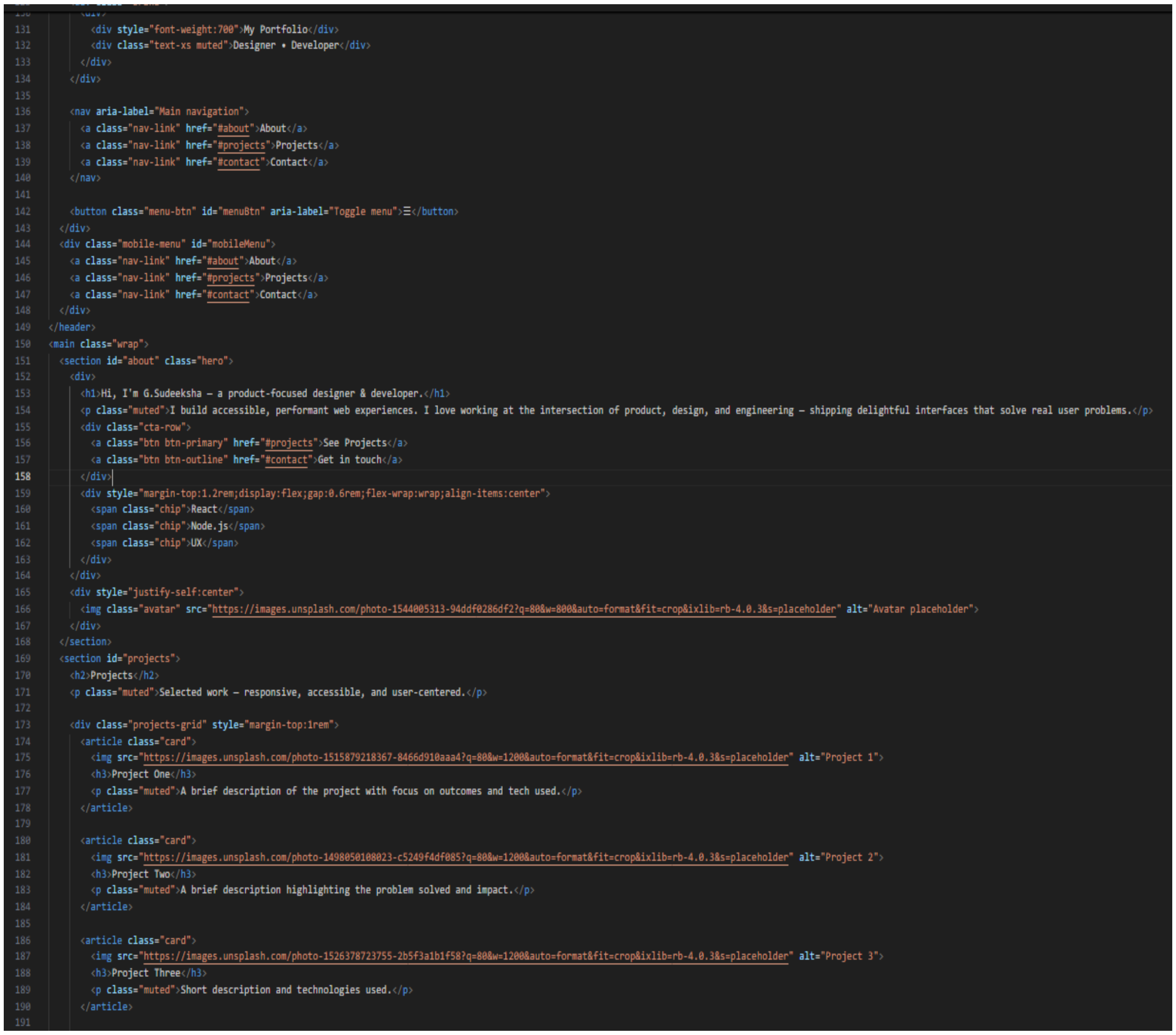


```html
<!--
Single-file Portfolio Website
- Sections: About Me, Projects, Contact
- Responsive layout using CSS Grid & Flexbox
- Smooth scrolling navigation

AI-suggested color palette & typography:
- Primary: #0EA5A4 (Teal 500)
- Secondary: #0F172A (Slate 900)
- Accent: #F97316 (Orange 500)
- Background: #F8FAFC (Gray 50)
- Muted Text: #475569 (Slate 600)

Google Fonts:
- Heading: "Merriweather" (serif)
- Body: "Inter" (sans-serif)

Usage: Save this file as `index.html` and open in a browser. Edit placeholder text/images as needed.
-->
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>My Portfolio</title>
  <!-- Google Fonts -->
  <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;600;700&family=Merriweather:wght@400;700&display=swap" rel="stylesheet">
  <style>
    /* ---- Root variables (AI suggested palette) ---- */
    :root{
      --color-primary: #0EA5A4;
      --color-secondary: #0F172A;
      --color-accent: #F97316;
      --bg: #F8FAFC;
      --muted: #475569;
      --glass: rgba(255,255,255,0.6);
      --max-width: 1100px;
    }
    /* Reset & base */
    *{box-sizing:border-box}
    html{scroll-behavior:smooth}
    body{
      margin:0;
      font-family: 'Inter', system-ui, -apple-system, 'Segoe UI', Roboto, 'Helvetica Neue', Arial;
      color:var(--color-secondary);
      background:var(--bg);
      line-height:1.45;
      -webkit-font-smoothing:antialiased;
      -moz-osx-font-smoothing:grayscale;
      padding-bottom:4rem; /* breathing room for footer */
    }
    a{color:inherit;text-decoration:none}
    /* Container */
    .wrap{max-width:var(--max-width);margin:0 auto;padding:1rem}
    /* Header / Navbar */
    header{
      position:sticky;top:0;z-index:60;background:linear-gradient(180deg, rgba(255,255,255,0.8), rgba(255,255,255,0.6));backdrop-filter:blur(6px);
      border-bottom:1px solid rgba(15,23,42,0.04);
    }
    .nav-row{display:flex;align-items:center;justify-content:space-between;padding:0.6rem 1rem}
    .brand{display:flex;align-items:center;gap:0.6rem}
    .logo{
      width:42px;height:42px;border-radius:8px;background:linear-gradient(135deg,var(--color-primary),var(--color-accent));display:flex;align-items:center;justify-content:center;color:white;font-weight:700;font-family:'Merriweather';
    }
    nav{display:flex;gap:1rem;align-items:center}
    .nav-link{padding:0.5rem 0.6rem;border-radius:6px;font-weight:600;color:var(--muted)}
    .nav-link:hover{background:rgba(14,165,164,0.08);color:var(--color-secondary)}
```

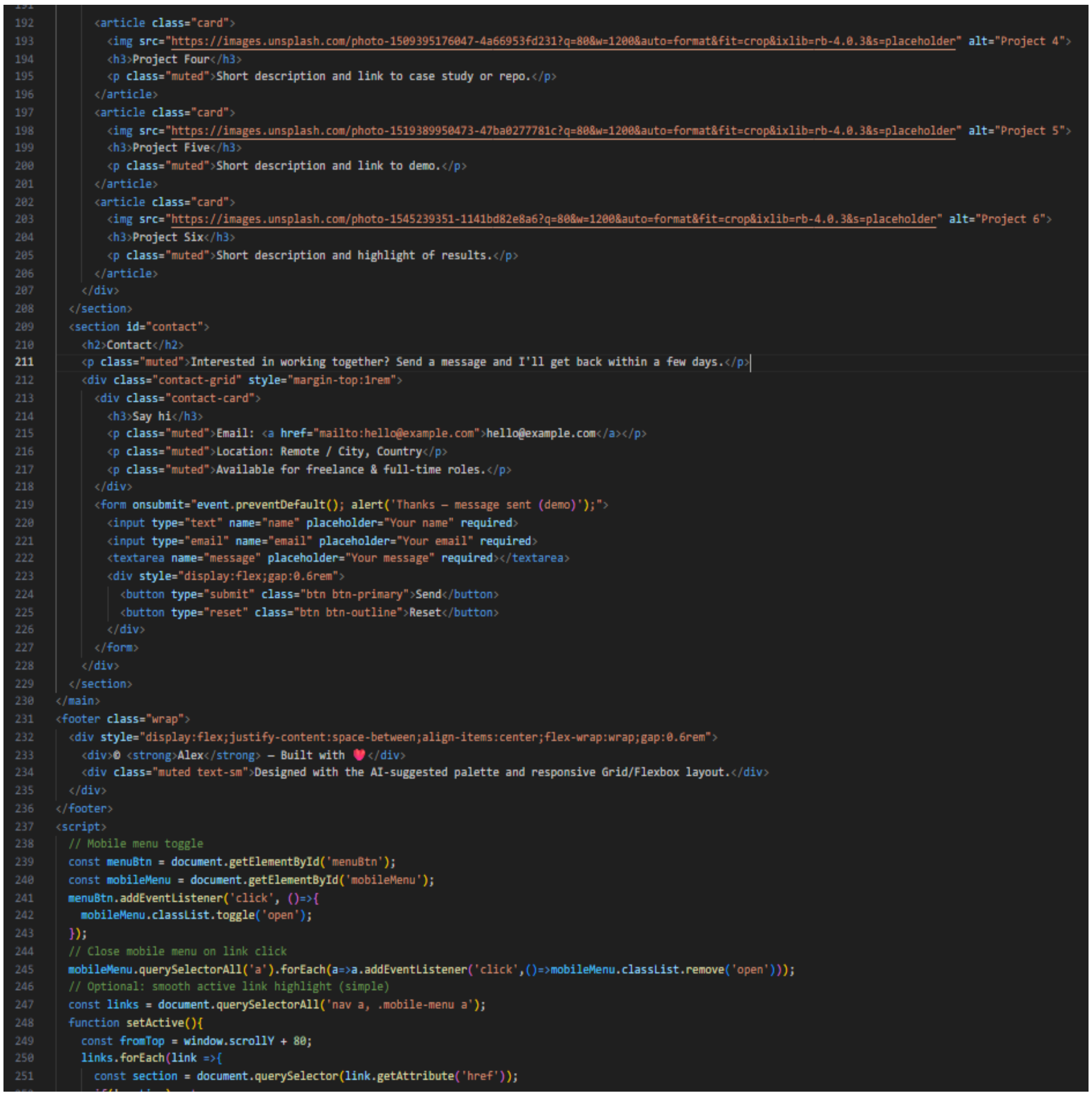


first1.html > html > body > main.wrap > section#about.hero > div > div.cta-row

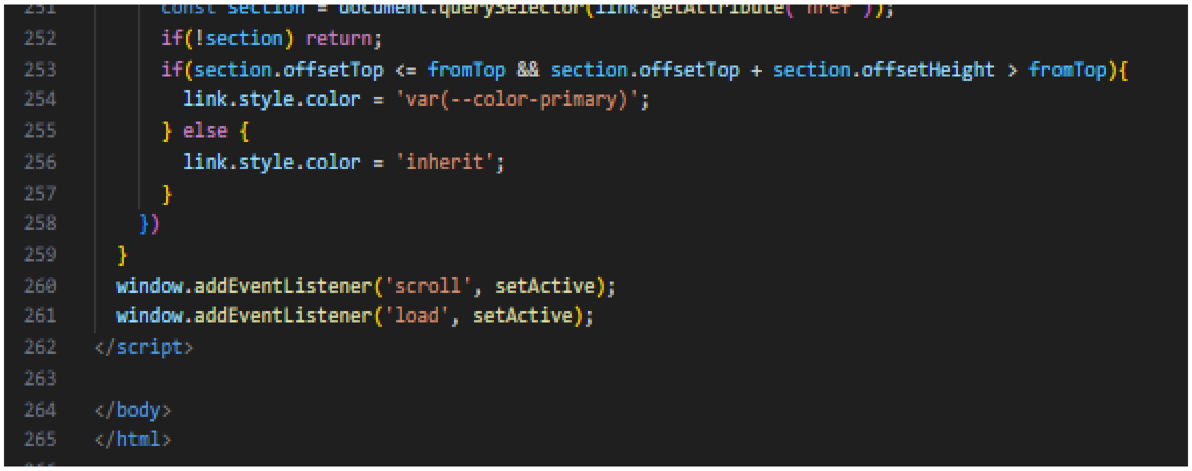
```html
<html lang="en">
<head>
  <style>
    /* Mobile menu button */
    .menu-btn{display:none;border:0;background:transparent;padding:0.4rem;border-radius:8px}
    /* Hero */
    .hero{display:grid;grid-template-columns:1fr 380px;gap:2rem;align-items:center;padding:3rem 0}
    .hero h1{font-family:'Merriweather', serif;font-size:2.1rem;margin:0;color:var(--color-secondary)}
    .hero p{color:var(--muted);margin-top:0.6rem}
    .cta-row{display:flex;gap:0.8rem;margin-top:1rem}
    .btn{display:inline-flex;align-items:center;gap:0.6rem;padding:0.6rem 0.9rem;border-radius:8px;font-weight:600;cursor:pointer}
    .btn-primary{background:var(--color-primary);color:white}
    .btn-outline{border:1px solid rgba(15,23,42,0.08);background:white;color:var(--color-secondary)}
    .avatar{width:100%;max-width:320px;border-radius:14px;box-shadow:0 10px 30px rgba(15,23,42,0.08)}
    /* Sections */
    section{padding:2.2rem 0}
    section h2{font-family:'Merriweather';margin:0 0 0.6rem 0}
    .muted{color:var(--muted)}
    /* Projects grid */
    .projects-grid{display:grid;grid-template-columns:repeat(3,1fr);gap:1.2rem}
    .card{background:white;border-radius:12px;padding:1rem;box-shadow:0 8px 20px rgba(15,23,42,0.04);overflow:hidden}
    .card img{width:100%;height:160px;object-fit:cover;border-radius:8px}
    .card h3{margin:0.6rem 0 0.2rem 0}
    .card p{margin:0;color:var(--muted);font-size:0.95rem}

    /* Contact */
    .contact-grid{display:grid;grid-template-columns:1fr 1fr;gap:1rem}
    .contact-card{background:linear-gradient(180deg, rgba(14,165,164,0.05), rgba(249,115,22,0.02));padding:1rem;border-radius:10px}
    form{display:flex;flex-direction:column;gap:0.6rem}
    input,textarea{padding:0.7rem;border-radius:8px;border:1px solid rgba(15,23,42,0.06);font-family:inherit}
    textarea{min-height:120px;resize:vertical}

    footer{padding:1.8rem 0;color:var(--muted);font-size:0.95rem}
    /* Utilities */
    .row{display:flex;gap:1rem;align-items:center}
    .chip{background:rgba(15,23,42,0.05);padding:0.35rem 0.6rem;border-radius:999px;font-weight:600}
    /* Responsive */
    @media (max-width:980px){
      .hero{grid-template-columns:1fr;}
      .projects-grid{grid-template-columns:repeat(2,1fr)}
      nav{display:none}
      .menu-btn{display:inline-flex}
    }
    @media (max-width:600px){
      .projects-grid{grid-template-columns:1fr}
      .contact-grid{grid-template-columns:1fr}
      .nav-row{padding:0.6rem}
      .logo{width:38px;height:38px}
    }
    /* Mobile slide-down menu */
    .mobile-menu{display:none;flex-direction:column;padding:0.8rem;background:white;border-bottom:1px solid rgba(15,23,42,0.04)}
    .mobile-menu.open{display:flex}

    /* small helpers */
    .text-sm{font-size:0.95rem}
    .text-xs{font-size:0.85rem}

  </style>
</head>
<body>

<header>
  <div class="wrap nav-row">
    <div class="brand">
      <div class="logo">MP</div>
      <div>
```



```html
        <div style="font-weight:700">My Portfolio</div>
        <div class="text-xs muted">Designer • Developer</div>
      </div>
    </div>

    <nav aria-label="Main navigation">
      <a class="nav-link" href="#about">About</a>
      <a class="nav-link" href="#projects">Projects</a>
      <a class="nav-link" href="#contact">Contact</a>
    </nav>

    <button class="menu-btn" id="menuBtn" aria-label="Toggle menu">☰</button>
  </div>
  <div class="mobile-menu" id="mobileMenu">
    <a class="nav-link" href="#about">About</a>
    <a class="nav-link" href="#projects">Projects</a>
    <a class="nav-link" href="#contact">Contact</a>
  </div>
</header>
<main class="wrap">
  <section id="about" class="hero">
    <div>
      <h1>Hi, I'm G.Sudeeksha — a product-focused designer & developer.</h1>
      <p class="muted">I build accessible, performant web experiences. I love working at the intersection of product, design, and engineering — shipping delightful interfaces that solve real user problems.</p>
      <div class="cta-row">
        <a class="btn btn-primary" href="#projects">See Projects</a>
        <a class="btn btn-outline" href="#contact">Get in touch</a>
      </div>
      <div style="margin-top:1.2rem;display:flex;gap:0.6rem;flex-wrap:wrap;align-items:center">
        <span class="chip">React</span>
        <span class="chip">Node.js</span>
        <span class="chip">UX</span>
      </div>
    </div>
    <div style="justify-self:center">
      <img class="avatar" src="https://images.unsplash.com/photo-1544005313-94ddf0286df2?q=80&w=800&auto=format&fit=crop&ixlib=rb-4.0.3&s=placeholder" alt="Avatar placeholder">
    </div>
  </section>
  <section id="projects">
    <h2>Projects</h2>
    <p class="muted">Selected work — responsive, accessible, and user-centered.</p>

    <div class="projects-grid" style="margin-top:1rem">
      <article class="card">
        <img src="https://images.unsplash.com/photo-1515879218367-8466d910aaa4?q=80&w=1200&auto=format&fit=crop&ixlib=rb-4.0.3&s=placeholder" alt="Project 1">
        <h3>Project One</h3>
        <p class="muted">A brief description of the project with focus on outcomes and tech used.</p>
      </article>

      <article class="card">
        <img src="https://images.unsplash.com/photo-1498050108023-c5249f4df085?q=80&w=1200&auto=format&fit=crop&ixlib=rb-4.0.3&s=placeholder" alt="Project 2">
        <h3>Project Two</h3>
        <p class="muted">A brief description highlighting the problem solved and impact.</p>
      </article>

      <article class="card">
        <img src="https://images.unsplash.com/photo-1526378723755-2b5f3a1b1f58?q=80&w=1200&auto=format&fit=crop&ixlib=rb-4.0.3&s=placeholder" alt="Project 3">
        <h3>Project Three</h3>
        <p class="muted">Short description and technologies used.</p>
      </article>

```

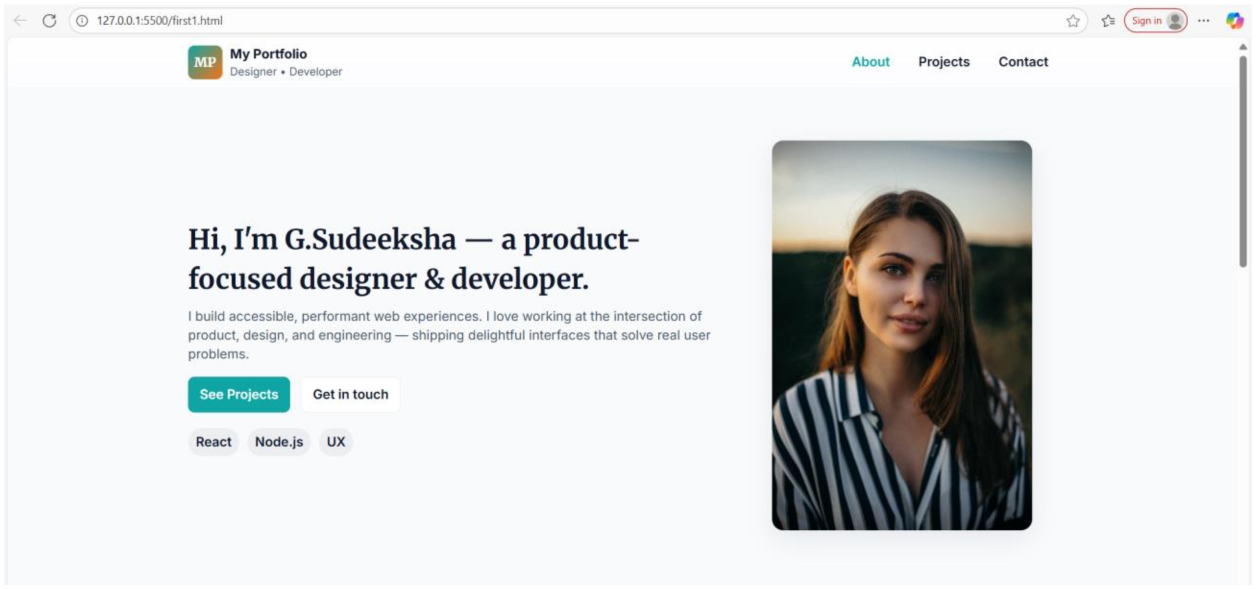


```html
      <article class="card">
        <img src="https://images.unsplash.com/photo-1509395176047-4a66953fd231?q=80&w=1200&auto=format&fit=crop&ixlib=rb-4.0.3&s=placeholder" alt="Project 4">
        <h3>Project Four</h3>
        <p class="muted">Short description and link to case study or repo.</p>
      </article>
      <article class="card">
        <img src="https://images.unsplash.com/photo-1519389950473-47ba0277781c?q=80&w=1200&auto=format&fit=crop&ixlib=rb-4.0.3&s=placeholder" alt="Project 5">
        <h3>Project Five</h3>
        <p class="muted">Short description and link to demo.</p>
      </article>
      <article class="card">
        <img src="https://images.unsplash.com/photo-1545239351-1141bd82e8a6?q=80&w=1200&auto=format&fit=crop&ixlib=rb-4.0.3&s=placeholder" alt="Project 6">
        <h3>Project Six</h3>
        <p class="muted">Short description and highlight of results.</p>
      </article>
    </div>
  </section>
  <section id="contact">
    <h2>Contact</h2>
    <p class="muted">Interested in working together? Send a message and I'll get back within a few days.</p>
    <div class="contact-grid" style="margin-top:1rem">
      <div class="contact-card">
        <h3>Say hi</h3>
        <p class="muted">Email: <a href="mailto:hello@example.com">hello@example.com</a></p>
        <p class="muted">Location: Remote / City, Country</p>
        <p class="muted">Available for freelance & full-time roles.</p>
      </div>
      <form onsubmit="event.preventDefault(); alert('Thanks — message sent (demo)');">
        <input type="text" name="name" placeholder="Your name" required>
        <input type="email" name="email" placeholder="Your email" required>
        <textarea name="message" placeholder="Your message" required></textarea>
        <div style="display:flex;gap:0.6rem">
          <button type="submit" class="btn btn-primary">Send</button>
          <button type="reset" class="btn btn-outline">Reset</button>
        </div>
      </form>
    </div>
  </section>
</main>
<footer class="wrap">
  <div style="display:flex;justify-content:space-between;align-items:center;flex-wrap:wrap;gap:0.6rem">
    <div>© <strong>Alex</strong> — Built with ❤️</div>
    <div class="muted text-sm">Designed with the AI-suggested palette and responsive Grid/Flexbox layout.</div>
  </div>
</footer>
<script>
  // Mobile menu toggle
  const menuBtn = document.getElementById('menuBtn');
  const mobileMenu = document.getElementById('mobileMenu');
  menuBtn.addEventListener('click', ()=>{
    mobileMenu.classList.toggle('open');
  });
  // Close mobile menu on link click
  mobileMenu.querySelectorAll('a').forEach(a=>a.addEventListener('click',()=>mobileMenu.classList.remove('open')));
  // Optional: smooth active link highlight (simple)
  const links = document.querySelectorAll('nav a, .mobile-menu a');
  function setActive(){
    const fromTop = window.scrollY + 80;
    links.forEach(link =>{
      const section = document.querySelector(link.getAttribute('href'));
```



```
   const section = document.querySelector(link.getAttribute('href'));
   if(!section) return;
   if(section.offsetTop <= fromTop && section.offsetTop + section.offsetHeight > fromTop){
    link.style.color = 'var(--color-primary)';
   } else {
    link.style.color = 'inherit';
   }
  })
 }
 window.addEventListener('scroll', setActive);
 window.addEventListener('load', setActive);
</script>

</body>
</html>
```

Output :



127.0.0.1:5500/first1.html

MP My Portfolio
Designer • Developer

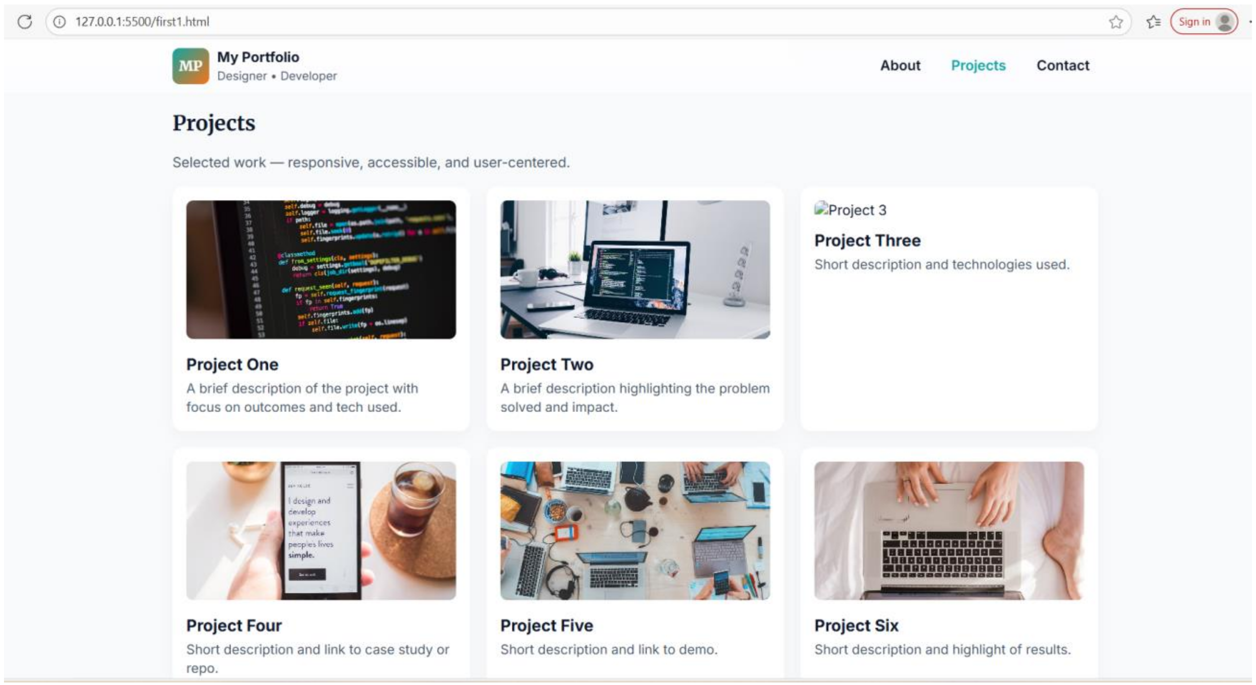
About Projects Contact

Hi, I'm G.Sudeeksha — a product-focused designer & developer.

I build accessible, performant web experiences. I love working at the intersection of product, design, and engineering — shipping delightful interfaces that solve real user problems.

See Projects Get in touch

React Node.js UX



127.0.0.1:5500/first1.html

MP **My Portfolio**
Designer • Developer

About Projects Contact

Projects

Selected work — responsive, accessible, and user-centered.

Project One
A brief description of the project with focus on outcomes and tech used.

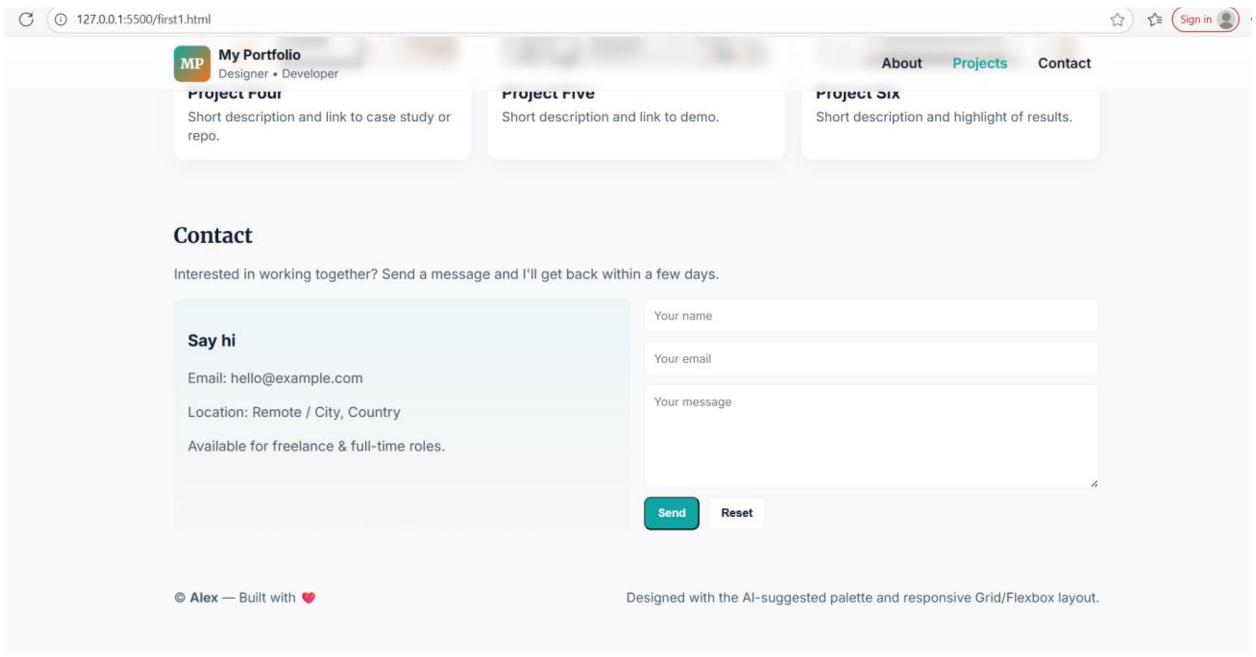
Project Two
A brief description highlighting the problem solved and impact.

Project 3
Project Three
Short description and technologies used.

Project Four
Short description and link to case study or repo.

Project Five
Short description and link to demo.

Project Six
Short description and highlight of results.



127.0.0.1:5500/first1.html

MP **My Portfolio**
Designer • Developer

About Projects Contact

Project Four
Short description and link to case study or repo.

Project Five
Short description and link to demo.

Project Six
Short description and highlight of results.

Contact

Interested in working together? Send a message and I'll get back within a few days.

Say hi

Email: hello@example.com

Location: Remote / City, Country

Available for freelance & full-time roles.

Your name
Your email
Your message

Send Reset

© **Alex** — Built with ❤

Designed with the AI-suggested palette and responsive Grid/Flexbox layout.

Observation :

While creating the responsive web page layout, I observed that using **HTML5 semantic elements** like `<header>`, `<main>`, and `<footer>` helped organize the structure clearly. Applying **responsive CSS** with **media queries** allowed the layout to automatically adjust across different screen sizes such as desktop, tablet, and mobile. On larger screens, elements were aligned horizontally for better visibility, while on smaller screens they stacked vertically for readability. The use of **flexbox and relative units (%, em, rem)** improved flexibility and consistency. Overall, I noticed that a clean and simple design with proper spacing and adaptive elements enhances user experience across all devices.

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.

Output:

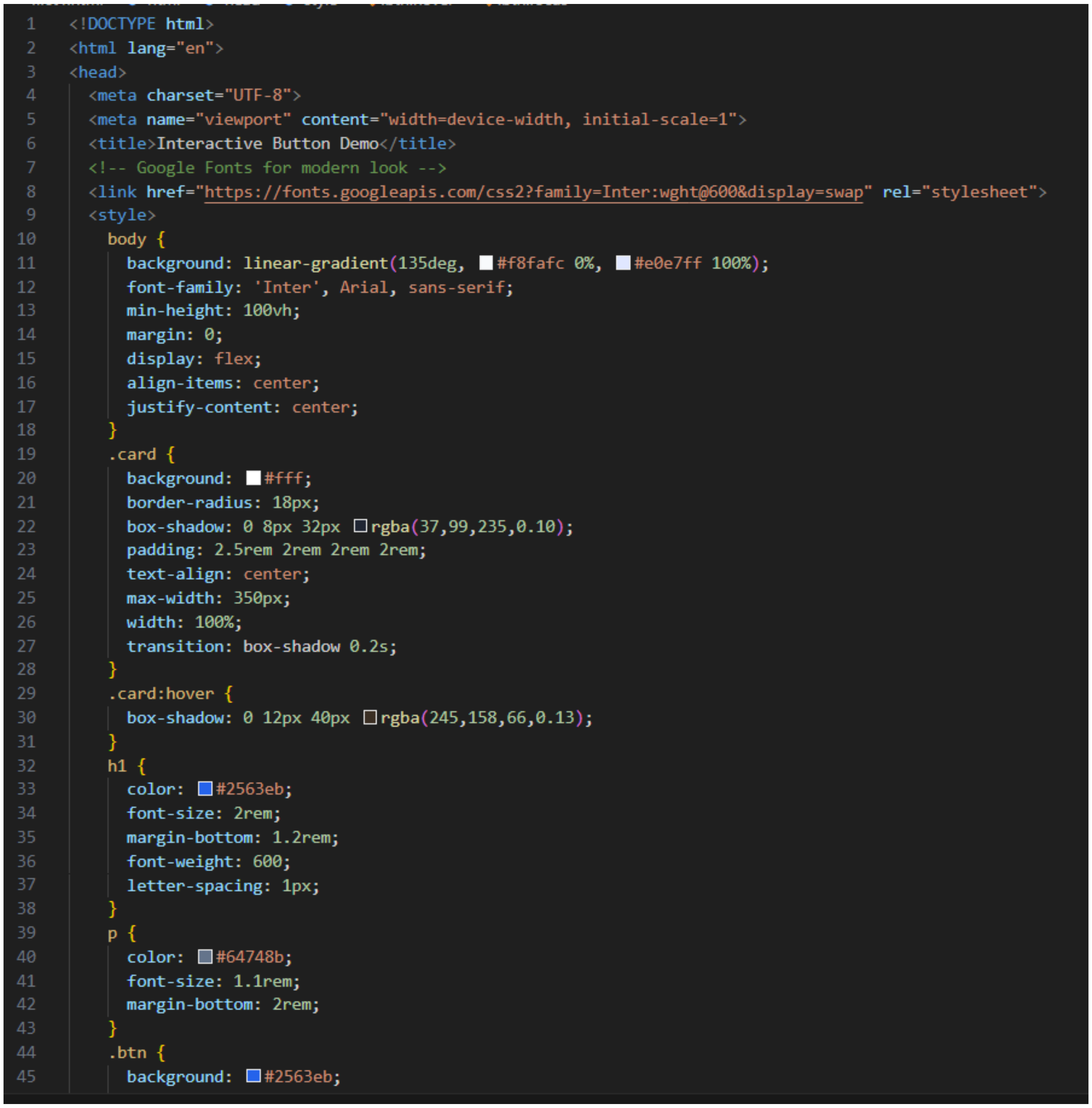
• A web page with a button.

• Clicking the button shows a pop-up alert message, e.g., “Button clicked!”

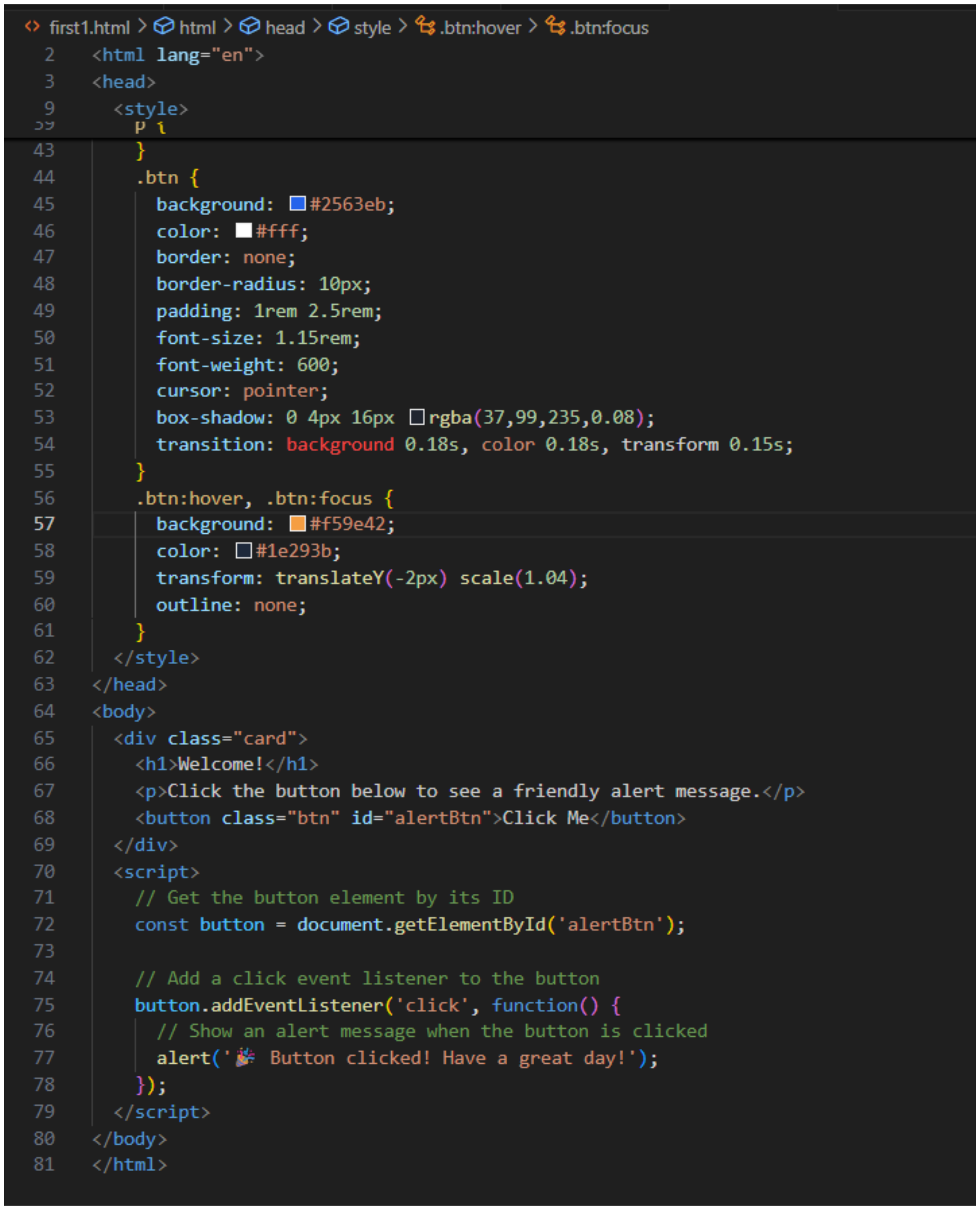
Prompt :

Create a basic HTML page that has one button. When I click the button, it should display an alert message saying “Button clicked!”. Make the code simple and add comments for clarity.

Code :



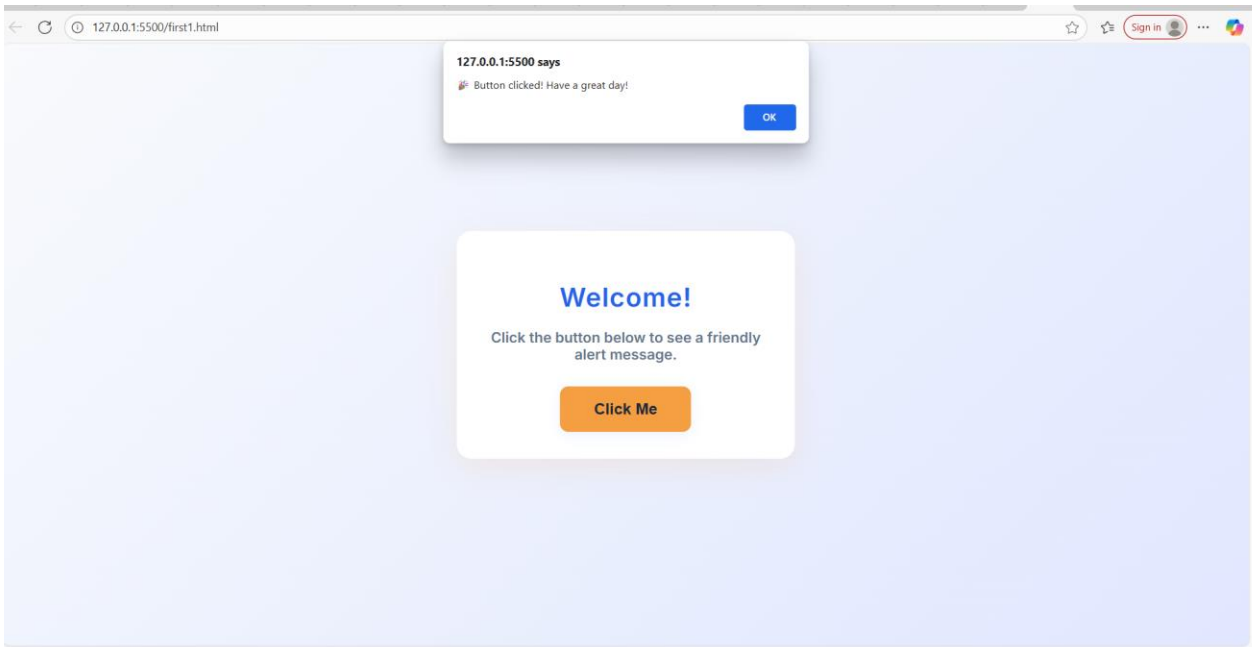
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Interactive Button Demo</title>
  <!-- Google Fonts for modern look -->
  <link href="https://fonts.googleapis.com/css2?family=Inter:wght@600&display=swap" rel="stylesheet">
  <style>
    body {
      background: linear-gradient(135deg, #f8fafc 0%, #e0e7ff 100%);
      font-family: 'Inter', Arial, sans-serif;
      min-height: 100vh;
      margin: 0;
      display: flex;
      align-items: center;
      justify-content: center;
    }
    .card {
      background: #fff;
      border-radius: 18px;
      box-shadow: 0 8px 32px rgba(37,99,235,0.10);
      padding: 2.5rem 2rem 2rem 2rem;
      text-align: center;
      max-width: 350px;
      width: 100%;
      transition: box-shadow 0.2s;
    }
    .card:hover {
      box-shadow: 0 12px 40px rgba(245,158,66,0.13);
    }
    h1 {
      color: #2563eb;
      font-size: 2rem;
      margin-bottom: 1.2rem;
      font-weight: 600;
      letter-spacing: 1px;
    }
    p {
      color: #64748b;
      font-size: 1.1rem;
      margin-bottom: 2rem;
    }
    .btn {
      background: #2563eb;
```



first1.html > html > head > style > .btn:hover > .btn:focus

```html
<html lang="en">
<head>
  <style>
    p {
    }
    .btn {
      background: #2563eb;
      color: #fff;
      border: none;
      border-radius: 10px;
      padding: 1rem 2.5rem;
      font-size: 1.15rem;
      font-weight: 600;
      cursor: pointer;
      box-shadow: 0 4px 16px rgba(37,99,235,0.08);
      transition: background 0.18s, color 0.18s, transform 0.15s;
    }
    .btn:hover, .btn:focus {
      background: #f59e42;
      color: #1e293b;
      transform: translateY(-2px) scale(1.04);
      outline: none;
    }
  </style>
</head>
<body>
  <div class="card">
    <h1>Welcome!</h1>
    <p>Click the button below to see a friendly alert message.</p>
    <button class="btn" id="alertBtn">Click Me</button>
  </div>
  <script>
    // Get the button element by its ID
    const button = document.getElementById('alertBtn');

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

Output :


127.0.0.1:5500 says

🎉 Button clicked! Have a great day!

127.0.0.1:5500/first1.html
OK
Welcome!
Click the button below to see a friendly alert message.
Click Me

Observation :

The provided code creates an attractive and user-friendly web page featuring a centered card with a welcoming message and a prominent interactive button. The use of a modern Google Font, a soft gradient background, and subtle card shadows gives the page a clean and contemporary appearance. The button includes smooth hover effects, enhancing interactivity and visual feedback. When clicked, the button triggers a cheerful alert message, demonstrating effective use of JavaScript for user engagement. Overall, the design is responsive, visually appealing, and provides a pleasant user experience with clear, well-structured code.

Task 3: Form with Validation

Instructions:

• Design a contact form with fields: Name, Email, Message.

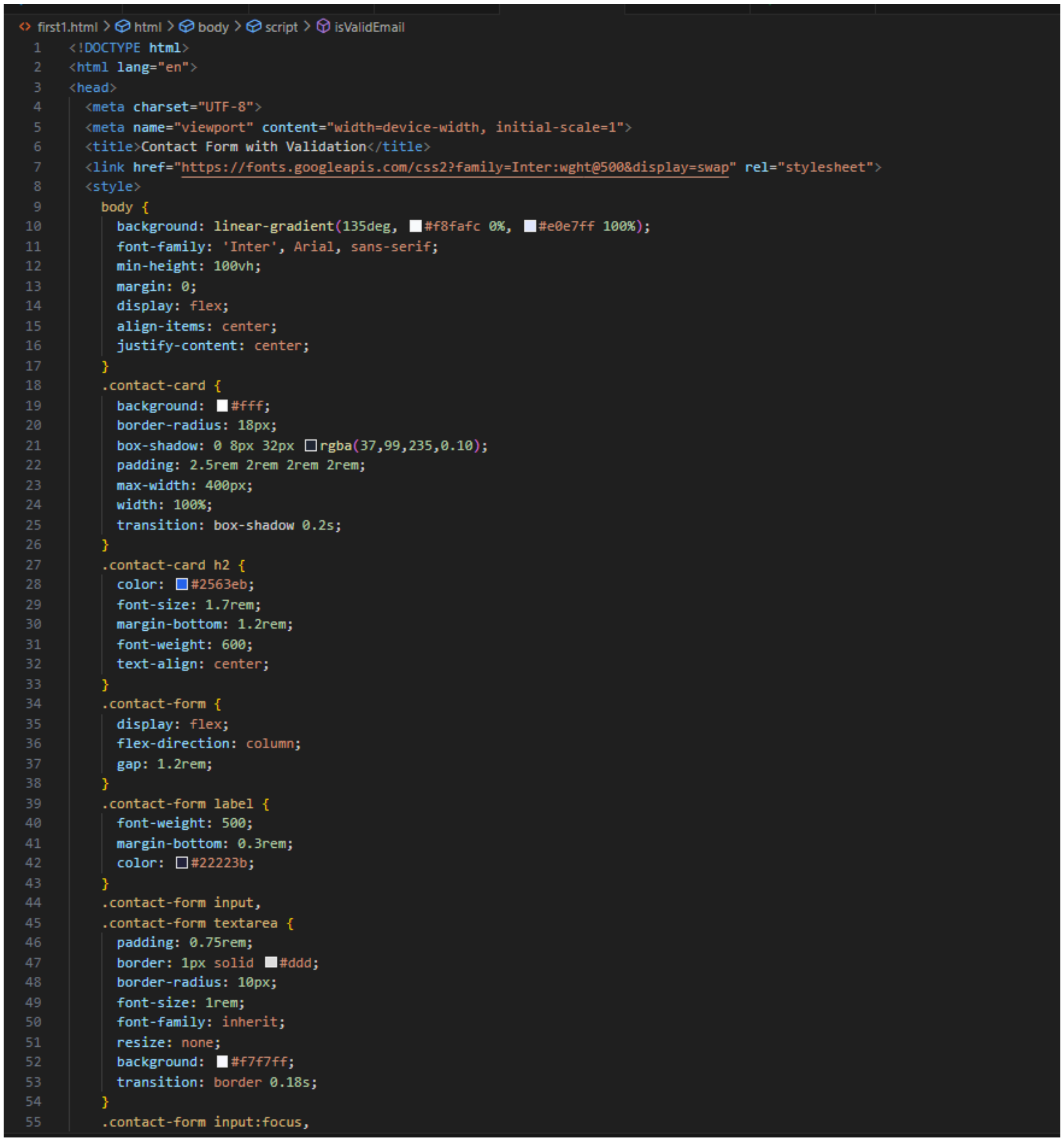
• Use AI to generate JavaScript code for form validation (e.g., non-empty fields, valid email format).

• Add inline error messages if input is invalid.
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.

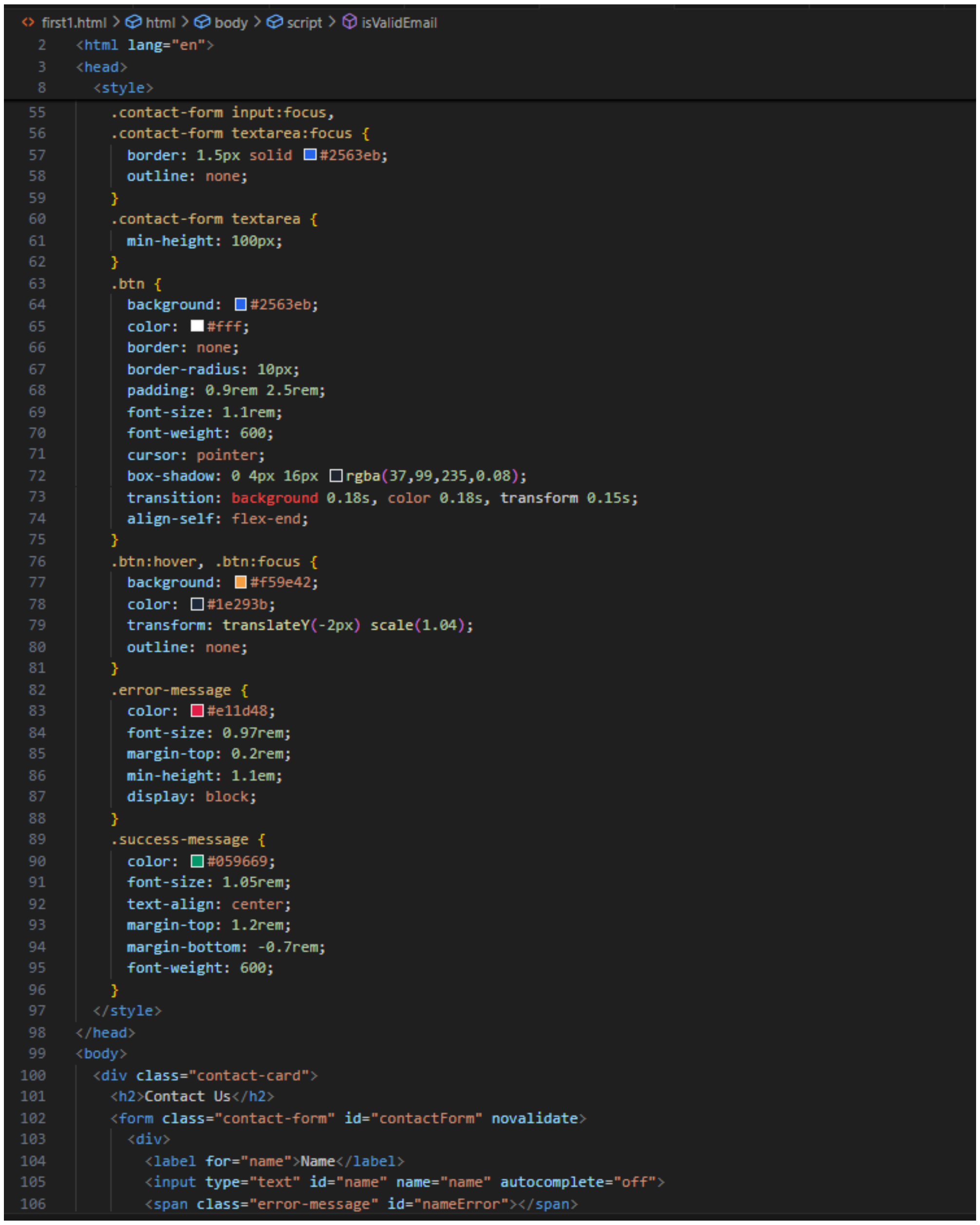
Prompt :

Create a contact form with Name, Email, and Message fields. Add JavaScript validation so that empty fields or invalid email show error messages, and display “Form submitted successfully” when all inputs are valid. Make the code clean and well-commented.

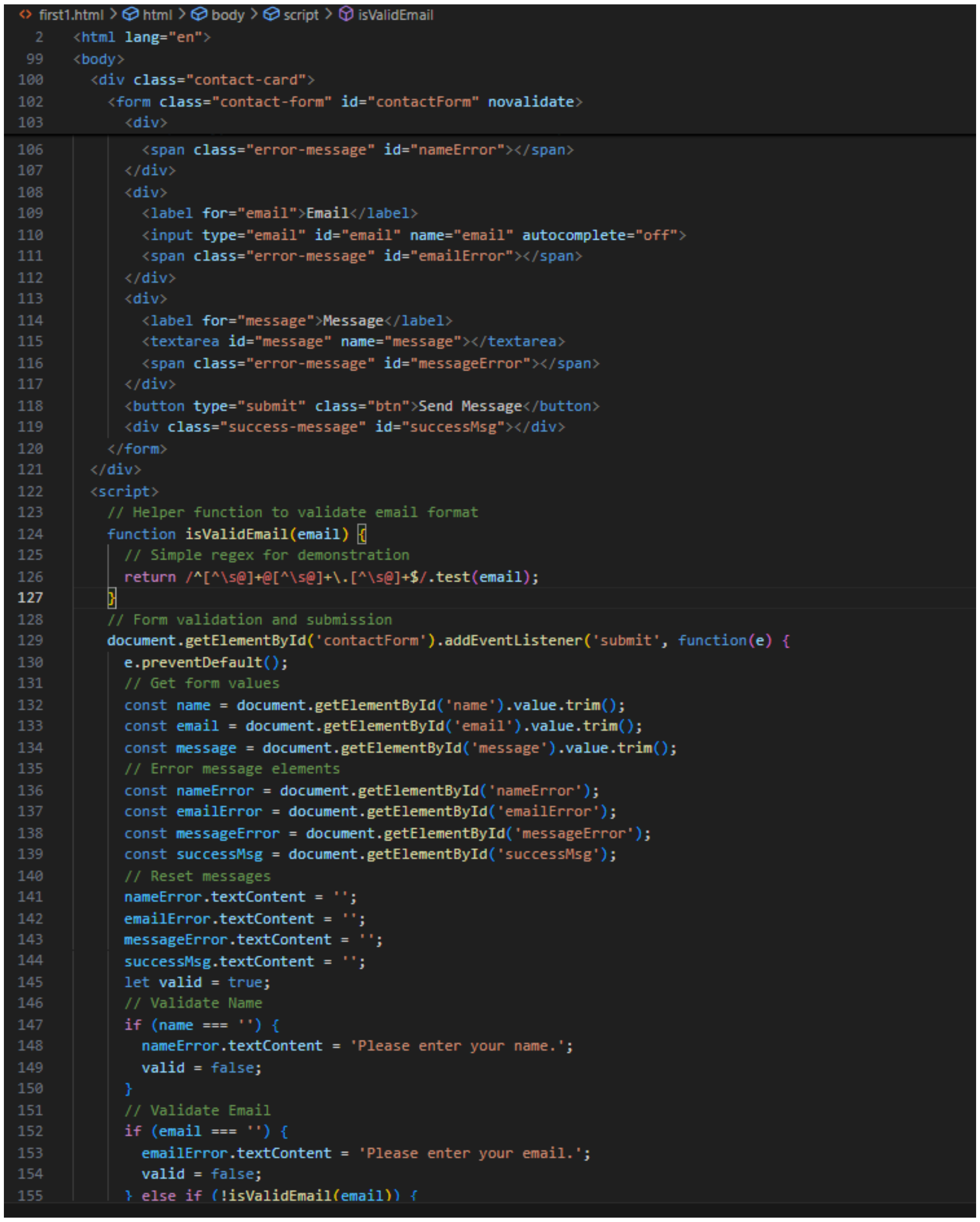
Code :



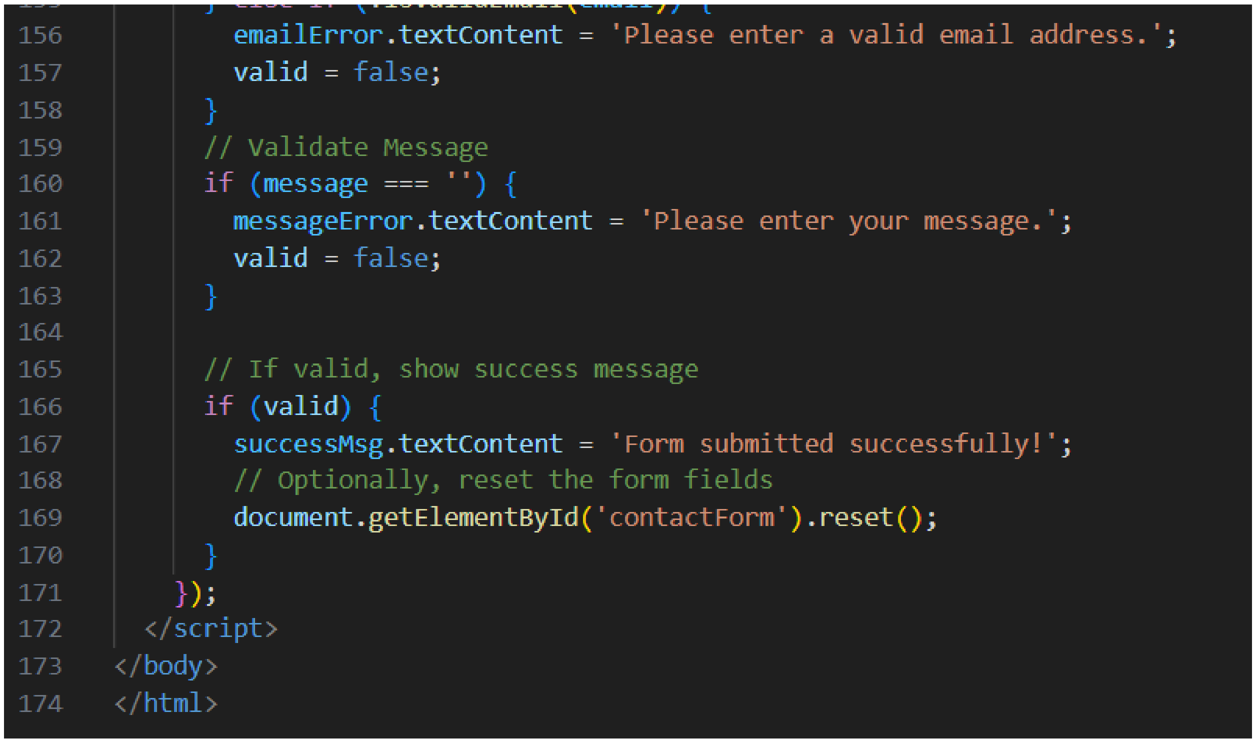
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Contact Form with Validation</title>
  <link href="https://fonts.googleapis.com/css2?family=Inter:wght@500&display=swap" rel="stylesheet">
  <style>
    body {
      background: linear-gradient(135deg, #f8fafc 0%, #e0e7ff 100%);
      font-family: 'Inter', Arial, sans-serif;
      min-height: 100vh;
      margin: 0;
      display: flex;
      align-items: center;
      justify-content: center;
    }
    .contact-card {
      background: #fff;
      border-radius: 18px;
      box-shadow: 0 8px 32px rgba(37,99,235,0.10);
      padding: 2.5rem 2rem 2rem 2rem;
      max-width: 400px;
      width: 100%;
      transition: box-shadow 0.2s;
    }
    .contact-card h2 {
      color: #2563eb;
      font-size: 1.7rem;
      margin-bottom: 1.2rem;
      font-weight: 600;
      text-align: center;
    }
    .contact-form {
      display: flex;
      flex-direction: column;
      gap: 1.2rem;
    }
    .contact-form label {
      font-weight: 500;
      margin-bottom: 0.3rem;
      color: #22223b;
    }
    .contact-form input,
    .contact-form textarea {
      padding: 0.75rem;
      border: 1px solid #ddd;
      border-radius: 10px;
      font-size: 1rem;
      font-family: inherit;
      resize: none;
      background: #f7f7ff;
      transition: border 0.18s;
    }
    .contact-form input:focus,
```



```
<html lang="en">
<head>
  <style>
    .contact-form input:focus,
    .contact-form textarea:focus {
      border: 1.5px solid #2563eb;
      outline: none;
    }
    .contact-form textarea {
      min-height: 100px;
    }
    .btn {
      background: #2563eb;
      color: #fff;
      border: none;
      border-radius: 10px;
      padding: 0.9rem 2.5rem;
      font-size: 1.1rem;
      font-weight: 600;
      cursor: pointer;
      box-shadow: 0 4px 16px rgba(37,99,235,0.08);
      transition: background 0.18s, color 0.18s, transform 0.15s;
      align-self: flex-end;
    }
    .btn:hover, .btn:focus {
      background: #f59e42;
      color: #1e293b;
      transform: translateY(-2px) scale(1.04);
      outline: none;
    }
    .error-message {
      color: #e11d48;
      font-size: 0.97rem;
      margin-top: 0.2rem;
      min-height: 1.1em;
      display: block;
    }
    .success-message {
      color: #059669;
      font-size: 1.05rem;
      text-align: center;
      margin-top: 1.2rem;
      margin-bottom: -0.7rem;
      font-weight: 600;
    }
  </style>
</head>
<body>
  <div class="contact-card">
    <h2>Contact Us</h2>
    <form class="contact-form" id="contactForm" novalidate>
      <div>
        <label for="name">Name</label>
        <input type="text" id="name" name="name" autocomplete="off">
        <span class="error-message" id="nameError"></span>
```

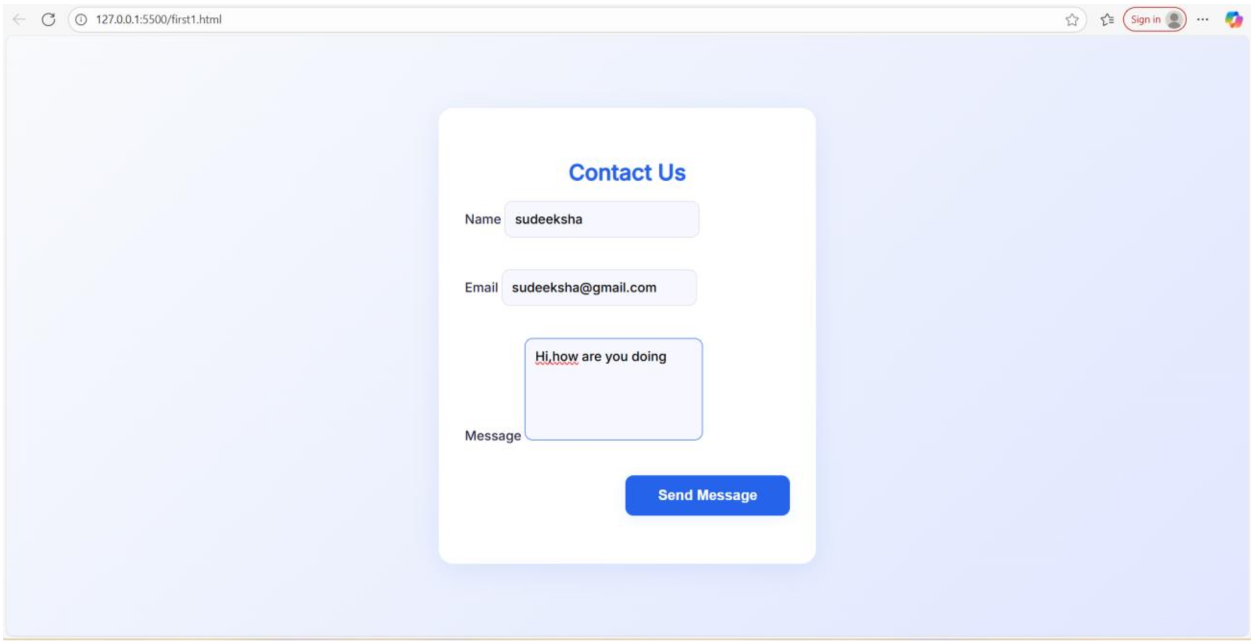


```
<html lang="en">
<body>
  <div class="contact-card">
    <form class="contact-form" id="contactForm" novalidate>
      <div>
        <span class="error-message" id="nameError"></span>
      </div>
      <div>
        <label for="email">Email</label>
        <input type="email" id="email" name="email" autocomplete="off">
        <span class="error-message" id="emailError"></span>
      </div>
      <div>
        <label for="message">Message</label>
        <textarea id="message" name="message"></textarea>
        <span class="error-message" id="messageError"></span>
      </div>
      <button type="submit" class="btn">Send Message</button>
      <div class="success-message" id="successMsg"></div>
    </form>
  </div>
  <script>
    // Helper function to validate email format
    function isValidEmail(email) {
      // Simple regex for demonstration
      return /^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email);
    }
    // Form validation and submission
    document.getElementById('contactForm').addEventListener('submit', function(e) {
      e.preventDefault();
      // Get form values
      const name = document.getElementById('name').value.trim();
      const email = document.getElementById('email').value.trim();
      const message = document.getElementById('message').value.trim();
      // Error message elements
      const nameError = document.getElementById('nameError');
      const emailError = document.getElementById('emailError');
      const messageError = document.getElementById('messageError');
      const successMsg = document.getElementById('successMsg');
      // Reset messages
      nameError.textContent = '';
      emailError.textContent = '';
      messageError.textContent = '';
      successMsg.textContent = '';
      let valid = true;
      // Validate Name
      if (name === '') {
        nameError.textContent = 'Please enter your name.';
        valid = false;
      }
      // Validate Email
      if (email === '') {
        emailError.textContent = 'Please enter your email.';
        valid = false;
      } else if (!isValidEmail(email)) {
```



```
        emailError.textContent = 'Please enter a valid email address.';
        valid = false;
      }
      // Validate Message
      if (message === '') {
        messageError.textContent = 'Please enter your message.';
        valid = false;
      }

      // If valid, show success message
      if (valid) {
        successMsg.textContent = 'Form submitted successfully!';
        // Optionally, reset the form fields
        document.getElementById('contactForm').reset();
      }
    });
  </script>
</body>
</html>
```

Output :



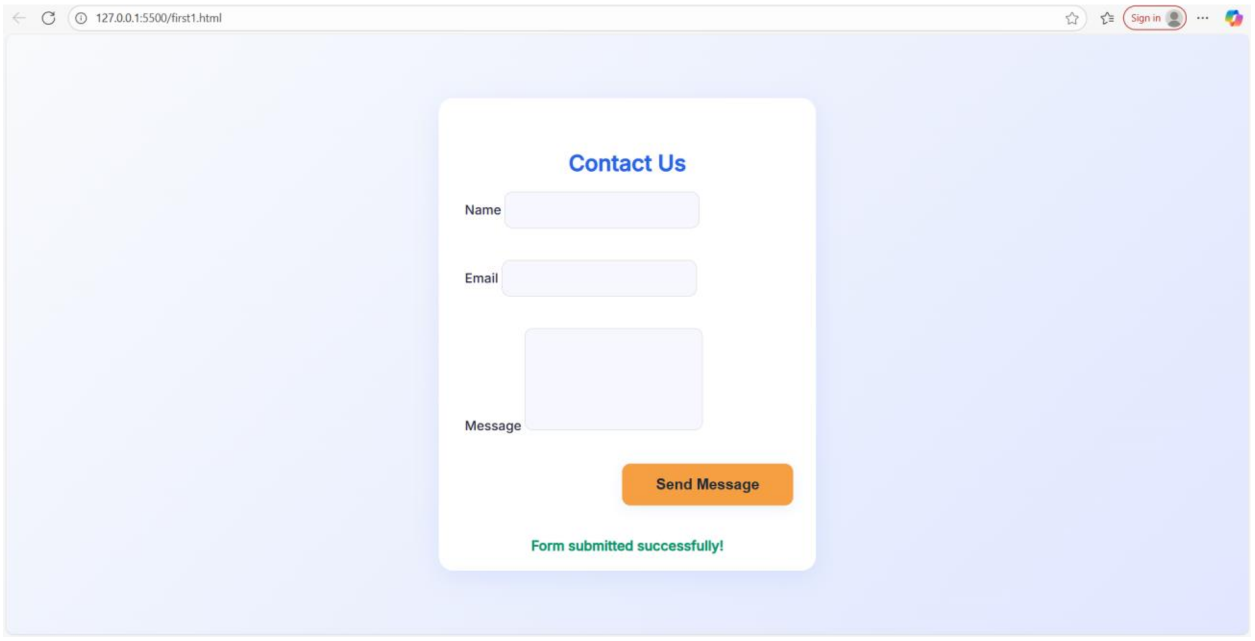
127.0.0.1:5500/first1.html

Contact Us

Name: sudeeksha

Email: sudeeksha@gmail.com

Message: Hi,how are you doing



Form submitted successfully!

Observation :

The provided contact form demonstrates effective use of modern web design and client-side validation. The form is visually appealing, featuring a clean layout, soft color palette, and responsive design that ensures usability across devices. JavaScript validation checks for non-empty fields and a valid email format, providing immediate, inline error messages to guide the user. Upon successful submission, a clear success message is displayed, enhancing user experience. Overall, the implementation is user-friendly, accessible, and robust, ensuring both functionality and a polished appearance.

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.

Output:

• A web page with a list and buttons:

o “Add Item” adds a new item to the list dynamically.

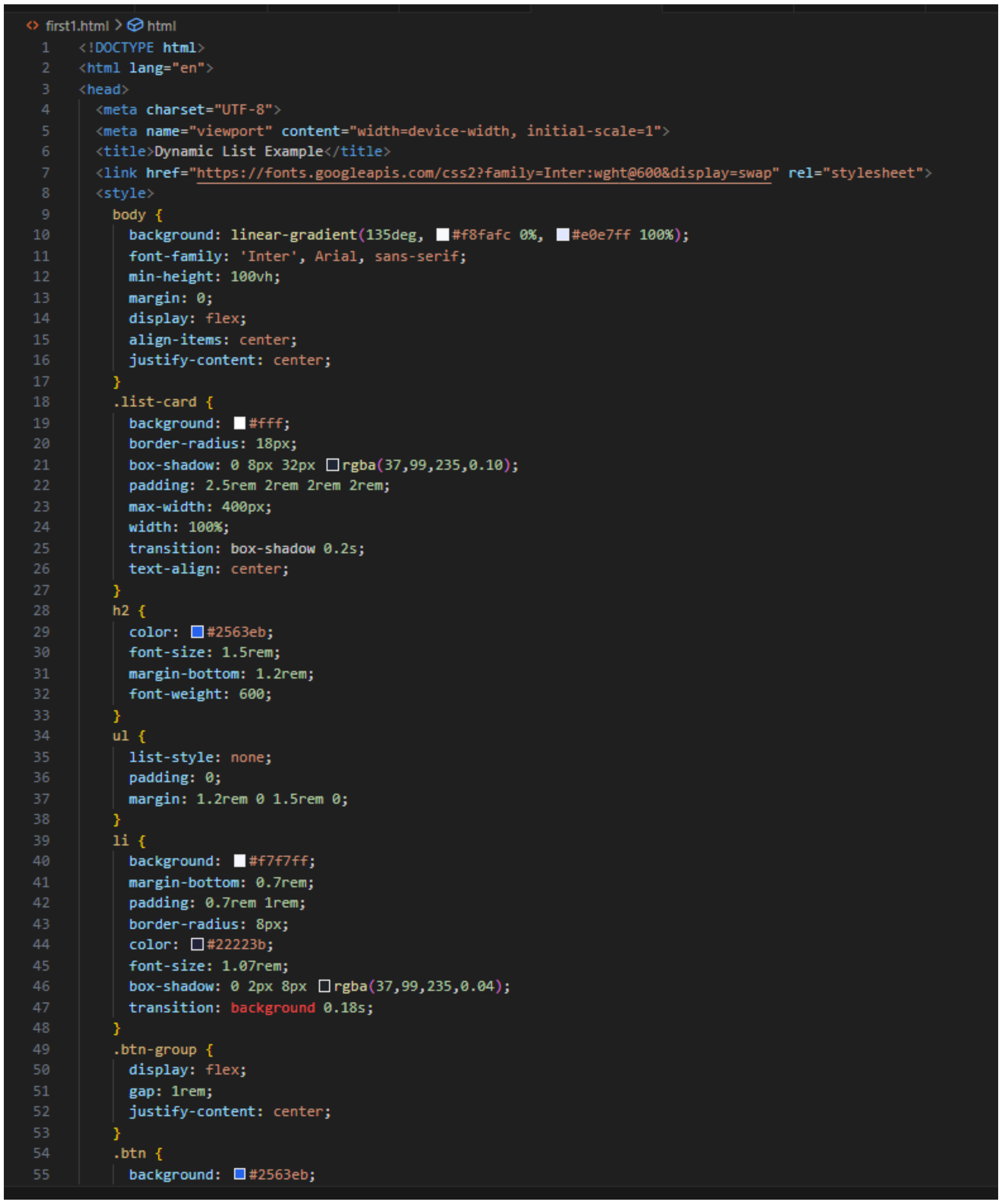
o “Remove Item” deletes the last item.

• Updates occur without reloading the page

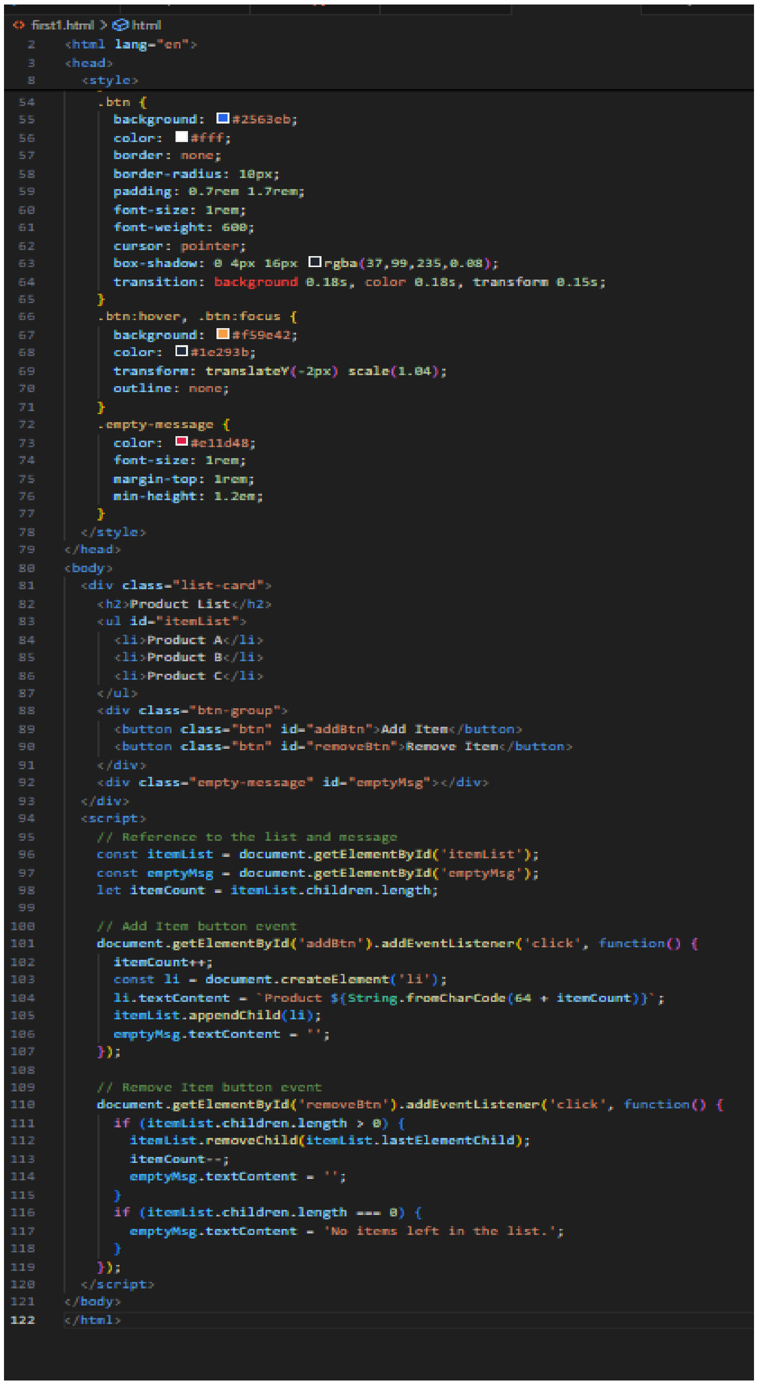
Prompt :

Create an HTML page with a list of items and two buttons — “Add Item” and “Remove Item.” Write JavaScript code to dynamically add a new item or remove the last item from the list when the buttons are clicked, without reloading the page. Add comments for clarity.

Code :

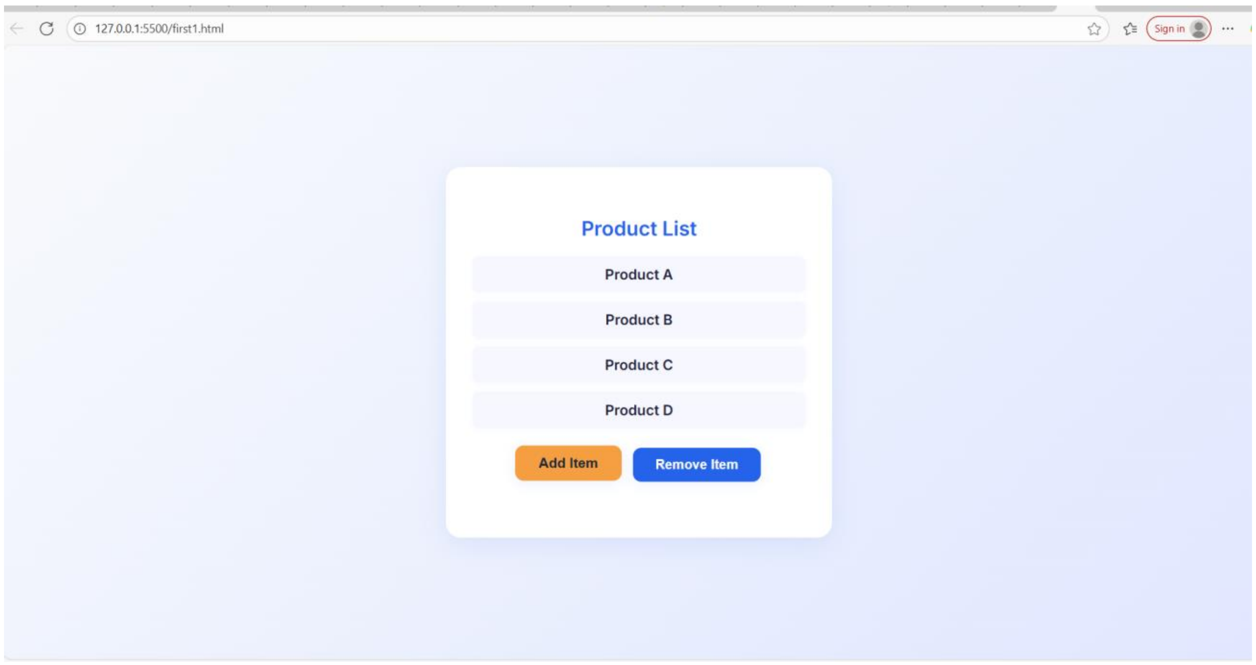


```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Dynamic List Example</title>
  <link href="https://fonts.googleapis.com/css2?family=Inter:wght@600&display=swap" rel="stylesheet">
  <style>
    body {
      background: linear-gradient(135deg, #f8fafc 0%, #e0e7ff 100%);
      font-family: 'Inter', Arial, sans-serif;
      min-height: 100vh;
      margin: 0;
      display: flex;
      align-items: center;
      justify-content: center;
    }
    .list-card {
      background: #fff;
      border-radius: 18px;
      box-shadow: 0 8px 32px rgba(37,99,235,0.10);
      padding: 2.5rem 2rem 2rem 2rem;
      max-width: 400px;
      width: 100%;
      transition: box-shadow 0.2s;
      text-align: center;
    }
    h2 {
      color: #2563eb;
      font-size: 1.5rem;
      margin-bottom: 1.2rem;
      font-weight: 600;
    }
    ul {
      list-style: none;
      padding: 0;
      margin: 1.2rem 0 1.5rem 0;
    }
    li {
      background: #f7f7ff;
      margin-bottom: 0.7rem;
      padding: 0.7rem 1rem;
      border-radius: 8px;
      color: #22223b;
      font-size: 1.07rem;
      box-shadow: 0 2px 8px rgba(37,99,235,0.04);
      transition: background 0.18s;
    }
    .btn-group {
      display: flex;
      gap: 1rem;
      justify-content: center;
    }
    .btn {
      background: #2563eb;
```



```html
<html lang="en">
<head>
  <style>
    .btn {
      background: #2563eb;
      color: #fff;
      border: none;
      border-radius: 10px;
      padding: 0.7rem 1.7rem;
      font-size: 1rem;
      font-weight: 600;
      cursor: pointer;
      box-shadow: 0 4px 16px rgba(37,99,235,0.08);
      transition: background 0.18s, color 0.18s, transform 0.15s;
    }
    .btn:hover, .btn:focus {
      background: #f59e42;
      color: #1e293b;
      transform: translateY(-2px) scale(1.04);
      outline: none;
    }
    .empty-message {
      color: #e11d48;
      font-size: 1rem;
      margin-top: 1rem;
      min-height: 1.2em;
    }
  </style>
</head>
<body>
  <div class="list-card">
    <h2>Product List</h2>
    <ul id="itemList">
      <li>Product A</li>
      <li>Product B</li>
      <li>Product C</li>
    </ul>
    <div class="btn-group">
      <button class="btn" id="addBtn">Add Item</button>
      <button class="btn" id="removeBtn">Remove Item</button>
    </div>
    <div class="empty-message" id="emptyMsg"></div>
  </div>
  <script>
    // Reference to the list and message
    const itemList = document.getElementById('itemList');
    const emptyMsg = document.getElementById('emptyMsg');
    let itemCount = itemList.children.length;

    // Add Item button event
    document.getElementById('addBtn').addEventListener('click', function() {
      itemCount++;
      const li = document.createElement('li');
      li.textContent = `Product ${String.fromCharCode(64 + itemCount)}`;
      itemList.appendChild(li);
      emptyMsg.textContent = '';
    });

    // Remove Item button event
    document.getElementById('removeBtn').addEventListener('click', function() {
      if (itemList.children.length > 0) {
        itemList.removeChild(itemList.lastElementChild);
        itemCount--;
        emptyMsg.textContent = '';
      }
      if (itemList.children.length === 0) {
        emptyMsg.textContent = 'No items left in the list.';
      }
    });
  </script>
</body>
</html>
```

Output :



Observation :

The provided code effectively demonstrates dynamic content generation on a web page using JavaScript. The user interface is clean and visually appealing, with a styled card layout and responsive design. Users can easily add new items to the product list or remove the last item with dedicated buttons, and all changes occur instantly without reloading the page. The script also provides helpful feedback by displaying a message when the list becomes empty, enhancing the user experience. Overall, the implementation is intuitive, interactive, and showcases best practices for manipulating the DOM in a modern, user-friendly way.

Task 5: Styled Modal Popup

Instructions:

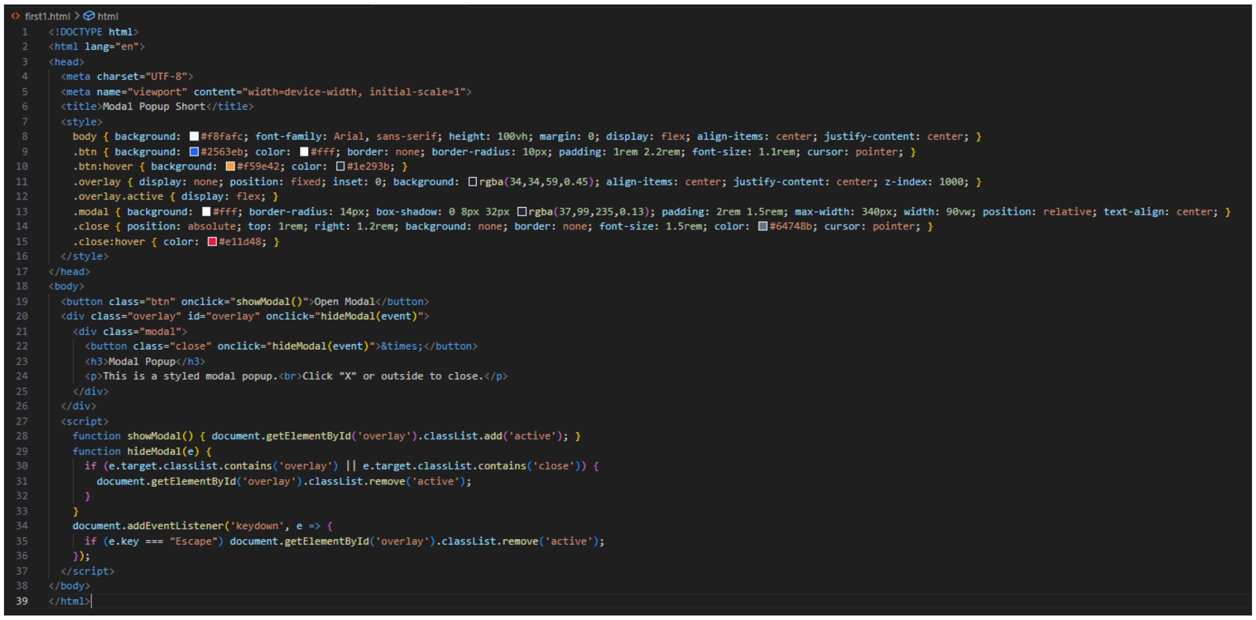
• Use AI to generate a modal popup that opens when a button is clicked.

• Style the modal using CSS with a semi-transparent overlay.
• Include a close button that hides the modal.
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 :

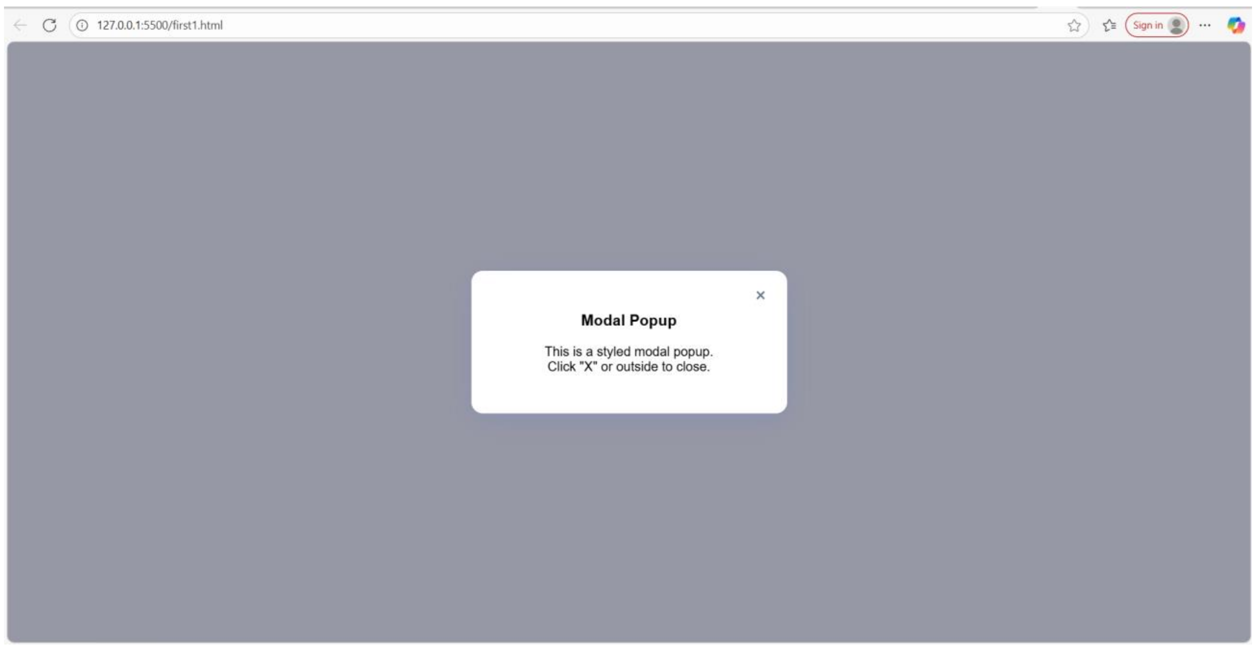
Create an HTML page with a button labeled “Open Modal.” When clicked, a styled modal popup with a semi-transparent overlay should appear, containing some text and a close (“X”) button. The modal should close when the “X” button or the overlay is clicked. Add clean CSS and JavaScript with comments.

Code :



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Modal Popup Short</title>
  <style>
    body { background: #f8fafc; font-family: Arial, sans-serif; height: 100vh; margin: 0; display: flex; align-items: center; justify-content: center; }
    .btn { background: #2563eb; color: #fff; border: none; border-radius: 10px; padding: 1rem 2.2rem; font-size: 1.1rem; cursor: pointer; }
    .btn:hover { background: #f59e42; color: #1e293b; }
    .overlay { display: none; position: fixed; inset: 0; background: rgba(34,34,59,0.45); align-items: center; justify-content: center; z-index: 1000; }
    .overlay.active { display: flex; }
    .modal { background: #fff; border-radius: 14px; box-shadow: 0 8px 32px rgba(37,99,235,0.13); padding: 2rem 1.5rem; max-width: 340px; width: 90vw; position: relative; text-align: center; }
    .close { position: absolute; top: 1rem; right: 1.2rem; background: none; border: none; font-size: 1.5rem; color: #64748b; cursor: pointer; }
    .close:hover { color: #e11d48; }
  </style>
</head>
<body>
  <button class="btn" onclick="showModal()">Open Modal</button>
  <div class="overlay" id="overlay" onclick="hideModal(event)">
    <div class="modal">
      <button class="close" onclick="hideModal(event)">&times;</button>
      <h3>Modal Popup</h3>
      <p>This is a styled modal popup.<br>Click "X" or outside to close.</p>
    </div>
  </div>
  <script>
    function showModal() { document.getElementById('overlay').classList.add('active'); }
    function hideModal(e) {
      if (e.target.classList.contains('overlay') || e.target.classList.contains('close')) {
        document.getElementById('overlay').classList.remove('active');
      }
    }
    document.addEventListener('keydown', e => {
      if (e.key === "Escape") document.getElementById('overlay').classList.remove('active');
    });
  </script>
</body>
</html>
```

Output :


Modal Popup

This is a styled modal popup.
Click "X" or outside to close.

Observation :

The code in first1.html provides a concise and effective implementation of a modal popup using HTML, CSS, and JavaScript. The design features a clean, centered "Open Modal" button that, when clicked, displays a modal overlay with a semi-transparent background, drawing the user's focus to the popup content. The modal can be closed intuitively by clicking the "X" button, clicking outside the modal area, or pressing the Escape key, enhancing usability and accessibility. The styling is modern and visually appealing, with smooth transitions and clear visual cues for interactive elements. Overall, the solution is user-friendly, responsive, and demonstrates best practices for creating modal dialogs in web development.