AI-ASSISTED CODING

ASSIGNMENT-14.1

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

**NAME: CHIPPA BHUVAN PRAKASH**

**HALL-TICKET: 2403A510F3**

**BATCH NO: 06**

**TASK-1**:

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 personal portfolio website using HTML, CSS, and JavaScript. Include sections for About Me, Projects, and Contact. Use a responsive layout with Flexbox or Grid, add smooth scrolling navigation, and suggest a matching color palette and typography. include 3 sample projects and a space to upload my own profile image.

**CODE GENERATED:**

<!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 font (optional; remove if offline) -->

  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600;700&display=swap" rel="stylesheet">

  <style>

    :root{

      --primary:#1a1a2e;

      --secondary:#0f3460;

      --accent:#e94560;

      --bg:#f5f6fa;

      --text:#222;

      --header-h:72px; /\* used for offset when header is fixed \*/

    }

    \*{box-sizing:border-box}

    html,body{height:100%; margin:0; font-family:"Poppins",system-ui,-apple-system,Segoe UI,Roboto,"Helvetica Neue",Arial; color:var(--text); background:var(--bg); scroll-behavior: smooth;}

    /\* Fixed header \*/

    header{

      position:fixed; inset:0 0 auto 0; height:var(--header-h);

      display:flex; align-items:center; justify-content:center;

      background:var(--primary); color:#fff; z-index:1200;

      box-shadow:0 2px 8px rgba(0,0,0,0.12);

      padding:0 20px;

    }

    nav{display:flex; gap:20px; align-items:center;}

    nav a{color:#fff; text-decoration:none; font-weight:600; padding:8px 10px; border-radius:6px; transition:background .18s, color .18s}

    nav a:hover{background:rgba(255,255,255,0.05); color:var(--accent);}

    /\* ensure content isn't hidden behind fixed header \*/

    main{padding-top:calc(var(--header-h) + 28px); max-width:1100px; margin:0 auto;}

/\* sections \*/

    section{padding:48px 20px; margin-bottom:10px; background:transparent}

    h2{margin:0 0 18px; text-align:center; color:var(--secondary); font-size:1.9rem}

/\* ABOUT \*/

    #about{display:flex; flex-direction:column; align-items:center; text-align:center;}

    #about img{width:140px; height:140px; object-fit:cover; border-radius:50%; border:4px solid var(--accent); margin-bottom:18px}

    #about p{max-width:760px; line-height:1.6}

/\* PROJECTS \*/

    #projects .grid{display:grid; gap:22px; grid-template-columns:repeat(auto-fit,minmax(300px,1fr));}

    .card{background:#fff; border-radius:12px; padding:18px; box-shadow:0 6px 18px rgba(15,20,40,0.06); transition:transform .25s, box-shadow .25s}

    .card:hover{transform:translateY(-6px); box-shadow:0 14px 30px rgba(15,20,40,0.09)}

    .card h3{margin:6px 0 8px; color:var(--accent)}

    .card p{margin:0 0 12px; line-height:1.5}

    .card .meta{font-size:.9rem; color:#555}

    .project-actions{margin-top:10px; display:flex; gap:10px; align-items:center}

    .btn{display:inline-block; padding:8px 12px; border-radius:8px; text-decoration:none; font-weight:600; border:1px solid transparent}

    .btn-outline{background:transparent; color:var(--secondary); border-color:rgba(15,52,96,0.08)}

    .btn-solid{background:var(--accent); color:#fff}

/\* CONTACT \*/

    #contact form{max-width:520px; margin:0 auto; display:flex; flex-direction:column; gap:12px}

    input,textarea{padding:12px; border-radius:8px; border:1px solid #d7d7d7; font-size:1rem; width:100%}

    button{padding:12px 14px; border-radius:8px; border:none; background:var(--accent); color:#fff; font-weight:600; cursor:pointer}

    #form-msg{display:none; margin-top:8px; text-align:center; color:var(--secondary)}

/\* small screens \*/

    @media (max-width:640px){

      nav{gap:12px}

      h2{font-size:1.6rem}

    }

    /\* helpful: ensure anchors don't land under header on browsers without JS \*/

    section{scroll-margin-top:calc(var(--header-h) + 8px)}

  </style>

</head>

<body>

  <header>

    <nav>

      <a href="#about">About</a>

      <a href="#projects">Projects</a>

      <a href="#contact">Contact</a>

    </nav>

  </header>

  <main>

    <!-- ABOUT -->

    <section id="about" aria-labelledby="about-heading">

      <img src="myphoto.jpg" alt="My Photo">

      <h2 id="about-heading">About Me</h2>

      <p>Hello — I'm a web developer focused on clean, responsive interfaces and useful front-end features. I build projects with HTML, CSS, and JavaScript, and I love tiny details that make user experiences smoother.</p>

    </section>

    <!-- PROJECTS (only 3) -->

    <section id="projects" aria-labelledby="projects-heading">

      <h2 id="projects-heading">Projects</h2>

      <div class="grid">

        <article class="card" aria-labelledby="p1-title">

          <h3 id="p1-title">Personal Portfolio</h3>

          <p class="meta">HTML • CSS Grid • Responsive Design</p>

          <p>Self-hosted portfolio that showcases my work, contact form (demo), and modern layout built with Grid + Flexbox.</p>

          <div class="project-actions">

            <a class="btn btn-outline" href="#" aria-label="Open live demo for Portfolio">Live demo</a>

            <a class="btn btn-solid" href="#" aria-label="Open GitHub repo for Portfolio">Source</a>

          </div>

        </article>

        <article class="card" aria-labelledby="p2-title">

          <h3 id="p2-title">Weather App</h3>

          <p class="meta">JavaScript • API • Progressive</p>

          <p>Fetches real-time weather from OpenWeather (demo) with location search, icons, and mobile-first UI.</p>

          <div class="project-actions">

            <a class="btn btn-outline" href="#">Live demo</a>

            <a class="btn btn-solid" href="#">Source</a>

          </div>

        </article>

        <article class="card" aria-labelledby="p3-title">

          <h3 id="p3-title">Task Tracker</h3>

          <p class="meta">Vanilla JS • localStorage</p>

          <p>Small productivity app: add/edit/delete tasks, persist to localStorage, and filter completed items.</p>

          <div class="project-actions">

            <a class="btn btn-outline" href="#">Live demo</a>

            <a class="btn btn-solid" href="#">Source</a>

          </div>

        </article>

      </div>

    </section>

    <!-- CONTACT -->

    <section id="contact" aria-labelledby="contact-heading">

      <h2 id="contact-heading">Contact</h2>

      <form id="contact-form" novalidate>

        <input name="name" type="text" placeholder="Your name" required>

        <input name="email" type="email" placeholder="Your email" required>

        <textarea name="message" rows="5" placeholder="Your message"></textarea>

        <button type="submit">Send message</button>

        <div id="form-msg" role="status" aria-live="polite"></div>

      </form>

    </section>

  </main>

  <footer style="text-align:center;padding:20px 10px;color:#fff;background:var(--primary)">

    © 2025 My Portfolio — Built with ❤️ and AI

  </footer>

  <script>

    // DOM ready

    document.addEventListener('DOMContentLoaded', () => {

      const header = document.querySelector('header');

      const headerHeight = header.getBoundingClientRect().height;

      // Smooth scrolling for internal anchors — adjusts for fixed header

      document.querySelectorAll('a[href^="#"]').forEach(link => {

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

          const href = link.getAttribute('href');

          if (!href || href === '#') return;

          const target = document.querySelector(href);

          if (!target) return;

          e.preventDefault();

          const targetY = target.getBoundingClientRect().top + window.pageYOffset - headerHeight - 12;

          window.scrollTo({ top: Math.max(targetY, 0), behavior: 'smooth' });

          // update URL hash without jumping

          history.pushState(null, '', href);

        });

      });

      // Simple contact form demo handler (prevents real submission)

      const form = document.getElementById('contact-form');

      const msg = document.getElementById('form-msg');

      form.addEventListener('submit', (e) => {

        e.preventDefault();

        const data = new FormData(form);

        const name = data.get('name') || 'there';

        console.log('Contact form submitted (demo):', Object.fromEntries(data));

        msg.textContent = `Thanks ${name}! Message received (demo).`;

        msg.style.display = 'block';

        // reset input after brief delay to show message

        form.reset();

        setTimeout(() => msg.style.display = 'none', 4000);

      });

      // Optional: highlight current nav as you scroll (basic)

      const sections = document.querySelectorAll('main section[id]');

      const navLinks = Array.from(document.querySelectorAll('nav a'));

      const obs = new IntersectionObserver(entries => {

        entries.forEach(entry => {

          const id = entry.target.id;

          const link = document.querySelector(`nav a[href="#${id}"]`);

          if (!link) return;

          if (entry.isIntersecting) {

            navLinks.forEach(a => a.style.opacity = '0.6');

            link.style.opacity = '1';

          }

        });

      }, { root: null, threshold: 0.45 });

      sections.forEach(s => obs.observe(s));

    });

  </script>

</body>

</html>

**OUTPUT:**

**A screenshot of a person's profile

AI-generated content may be incorrect.**

**A screenshot of a web page

AI-generated content may be incorrect.**

**A screenshot of a contact form

AI-generated content may be incorrect.**

**OBSERVATION :**

The personal portfolio website is clean, responsive, and easy to navigate. It includes well-organized sections for About Me, Projects, and Contact. Smooth scrolling and a simple color scheme make it visually appealing and user-friendly on both desktop and mobile devices

TASK -2:

Design a product display page for an online store.  
Requirements:  
• Display product image, title, price, and "Add to Cart" button.  
• Use AI to:  
o Style with BEM methodology.  
o Make layout responsive.  
o Add hover effects and "Add to Cart" alert

PROMPT:

Create a responsive product display page for an online store using HTML, CSS, and JavaScript.  
Display multiple products in a grid layout with image, title, price, and “Add to Cart” button.  
Use BEM methodology for CSS, add hover effects, and show an alert when a product is added to the cart**.**  
CODE GENERATED:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Online Store - Products</title>

  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600;700&display=swap" rel="stylesheet">

  <style>

    /\* === Base Styles === \*/

    \* { box-sizing: border-box; margin: 0; padding: 0; }

    body {

      font-family: 'Poppins', sans-serif;

      background-color: #f4f5f7;

      color: #333;

      padding: 40px 20px;

    }

    h1 { text-align: center; margin-bottom: 30px; color: #1a1a2e; }

    /\* === Product Grid === \*/

    .products {

      display: grid;

      grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));

      gap: 25px;

      max-width: 1200px;

      margin: auto;

    }

    /\* === Product Card BEM === \*/

    .product-card {

      background: #fff;

      border-radius: 12px;

      overflow: hidden;

      box-shadow: 0 4px 12px rgba(0,0,0,0.1);

      transition: transform 0.3s, box-shadow 0.3s;

      display: flex;

      flex-direction: column;

      align-items: center;

      text-align: center;

      padding-bottom: 15px;

    }

    .product-card:hover {

      transform: translateY(-8px);

      box-shadow: 0 10px 20px rgba(0,0,0,0.15);

    }

    .product-card\_\_image {

      width: 100%;

      height: 200px;

      object-fit: cover;

    }

    .product-card\_\_title {

      font-size: 1.1rem;

      font-weight: 600;

      margin: 12px 0 6px;

      color: #1a1a2e;

    }

    .product-card\_\_price {

      font-size: 1rem;

      color: #28a745;

      margin-bottom: 12px;

      font-weight: 500;

    }

    .product-card\_\_button {

      background-color: #007bff;

      color: white;

      border: none;

      padding: 10px 18px;

      border-radius: 8px;

      cursor: pointer;

      transition: background-color 0.3s;

      font-weight: 500;

    }

    .product-card\_\_button:hover {

      background-color: #0056b3;

    }

    /\* === Responsive Text === \*/

    @media (max-width: 480px) {

      .product-card\_\_title { font-size: 1rem; }

      .product-card\_\_price { font-size: 0.95rem; }

    }

  </style>

</head>

<body>

  <h1>Our Products</h1>

  <div class="products">

    <div class="product-card">

      <img src="https://via.placeholder.com/300x200?text=Headphones" alt="Wireless Headphones" class="product-card\_\_image">

      <h2 class="product-card\_\_title">Wireless Headphones</h2>

      <p class="product-card\_\_price">$59.99</p>

      <button class="product-card\_\_button" onclick="addToCart('Wireless Headphones')">Add to Cart</button>

    </div>

    <div class="product-card">

      <img src="https://via.placeholder.com/300x200?text=Smartwatch" alt="Smartwatch" class="product-card\_\_image">

      <h2 class="product-card\_\_title">Smartwatch</h2>

      <p class="product-card\_\_price">$129.99</p>

      <button class="product-card\_\_button" onclick="addToCart('Smartwatch')">Add to Cart</button>

    </div>

    <div class="product-card">

      <img src="https://via.placeholder.com/300x200?text=Camera" alt="Digital Camera" class="product-card\_\_image">

      <h2 class="product-card\_\_title">Digital Camera</h2>

      <p class="product-card\_\_price">$499.99</p>

      <button class="product-card\_\_button" onclick="addToCart('Digital Camera')">Add to Cart</button>

    </div>

    <div class="product-card">

      <img src="https://via.placeholder.com/300x200?text=Laptop" alt="Laptop" class="product-card\_\_image">

      <h2 class="product-card\_\_title">Laptop</h2>

      <p class="product-card\_\_price">$899.99</p>

      <button class="product-card\_\_button" onclick="addToCart('Laptop')">Add to Cart</button>

    </div>

    <div class="product-card">

      <img src="https://via.placeholder.com/300x200?text=Shoes" alt="Sports Shoes" class="product-card\_\_image">

      <h2 class="product-card\_\_title">Sports Shoes</h2>

      <p class="product-card\_\_price">$79.99</p>

      <button class="product-card\_\_button" onclick="addToCart('Sports Shoes')">Add to Cart</button>

    </div>

    <div class="product-card">

      <img src="https://via.placeholder.com/300x200?text=Backpack" alt="Travel Backpack" class="product-card\_\_image">

      <h2 class="product-card\_\_title">Travel Backpack</h2>

      <p class="product-card\_\_price">$49.99</p>

      <button class="product-card\_\_button" onclick="addToCart('Travel Backpack')">Add to Cart</button>

    </div>

  </div>

  <script>

    function addToCart(productName) {

      alert(`✅ ${productName} added to cart successfully!`);

    }

  </script>

</body>

</html>

OUTPUT:  
A screenshot of a phone

AI-generated content may be incorrect.

OBSERVATION:

The product display page is clean, modern, and responsive. Products are shown in a grid with images, titles, prices, and “Add to Cart” buttons. Hover effects and alert messages enhance interactivity, and the layout works well on both desktop and mobile devices.

TASK – 3:

Build an event registration form for a conference.  
Requirements:  
• Collect name, email, phone number, and session selection.  
• Use AI to:  
o Add form validation with JavaScript.  
o Make the form accessible with labels and ARIA.  
o Style with a professional look.

PROMPT:

Build an event registration form for a conference using HTML, CSS, and JavaScript.  
Collect name, email, phone number, and session selection.  
Add form validation, accessible labels and ARIA attributes, and style it with a professional design.

CODE GENERATED:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Conference Registration Form</title>

  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600;700&display=swap" rel="stylesheet">

  <style>

    /\* === Base Styles === \*/

    body {

      font-family: 'Poppins', sans-serif;

      background-color: #f4f5f7;

      display: flex;

      justify-content: center;

      align-items: center;

      min-height: 100vh;

      margin: 0;

      padding: 20px;

    }

    .registration-form {

      background: #fff;

      padding: 30px 25px;

      border-radius: 12px;

      box-shadow: 0 8px 20px rgba(0,0,0,0.12);

      width: 100%;

      max-width: 450px;

    }

    .registration-form h2 {

      text-align: center;

      margin-bottom: 25px;

      color: #1a1a2e;

    }

    .form-group {

      margin-bottom: 18px;

      display: flex;

      flex-direction: column;

    }

    label {

      font-weight: 500;

      margin-bottom: 6px;

    }

    input, select {

      padding: 10px 12px;

      border-radius: 6px;

      border: 1px solid #ccc;

      font-size: 1rem;

    }

    input:focus, select:focus {

      outline: none;

      border-color: #007bff;

      box-shadow: 0 0 5px rgba(0,123,255,0.3);

    }

    button {

      width: 100%;

      padding: 12px;

      border: none;

      border-radius: 8px;

      background-color: #007bff;

      color: #fff;

      font-weight: 600;

      font-size: 1rem;

      cursor: pointer;

      transition: background 0.3s;

    }

    button:hover {

      background-color: #0056b3;

    }

    .error-message {

      color: #e94560;

      font-size: 0.9rem;

      margin-top: 4px;

      display: none;

    }

    @media (max-width: 480px) {

      .registration-form { padding: 20px; }

    }

  </style>

</head>

<body>

  <form class="registration-form" id="registrationForm" aria-labelledby="form-title" novalidate>

    <h2 id="form-title">Conference Registration</h2>

    <div class="form-group">

      <label for="name">Full Name <span aria-hidden="true">\*</span></label>

      <input type="text" id="name" name="name" aria-required="true">

      <span class="error-message" id="nameError">Please enter your name.</span>

    </div>

    <div class="form-group">

      <label for="email">Email <span aria-hidden="true">\*</span></label>

      <input type="email" id="email" name="email" aria-required="true">

      <span class="error-message" id="emailError">Please enter a valid email.</span>

    </div>

    <div class="form-group">

      <label for="phone">Phone Number <span aria-hidden="true">\*</span></label>

      <input type="tel" id="phone" name="phone" aria-required="true" placeholder="e.g., +1234567890">

      <span class="error-message" id="phoneError">Please enter a valid phone number.</span>

    </div>

    <div class="form-group">

      <label for="session">Select Session <span aria-hidden="true">\*</span></label>

      <select id="session" name="session" aria-required="true">

        <option value="">-- Please choose a session --</option>

        <option value="morning">Morning Session (9 AM - 12 PM)</option>

        <option value="afternoon">Afternoon Session (1 PM - 4 PM)</option>

        <option value="evening">Evening Session (5 PM - 8 PM)</option>

      </select>

      <span class="error-message" id="sessionError">Please select a session.</span>

    </div>

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

  </form>

  <script>

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

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

      e.preventDefault();

      // Clear previous errors

      document.querySelectorAll('.error-message').forEach(el => el.style.display = 'none');

      let valid = true;

      // Name validation

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

      if(name.value.trim() === "") {

        document.getElementById('nameError').style.display = 'block';

        valid = false;

      }

      // Email validation

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

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

      if(!emailPattern.test(email.value.trim())) {

        document.getElementById('emailError').style.display = 'block';

        valid = false;

      }

      // Phone validation

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

      const phonePattern = /^\+?\d{7,15}$/;

      if(!phonePattern.test(phone.value.trim())) {

        document.getElementById('phoneError').style.display = 'block';

        valid = false;

      }

      // Session validation

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

      if(session.value === "") {

        document.getElementById('sessionError').style.display = 'block';

        valid = false;

      }

      if(valid) {

        alert(`✅ Registration successful!\nName: ${name.value}\nSession: ${session.value}`);

        form.reset();

      }

    });

  </script>

</body>

</html>

OUTPUT:

A screen shot of a computer screen

AI-generated content may be incorrect.

OBSERVATION:

The event registration form is clean, professional, and easy to use. It collects name, email, phone number, and session selection, and provides real-time validation to ensure correct input. Labels and ARIA attributes make it accessible, and the responsive design ensures it works well on both desktop and mobile devices.

TASK – 4:

ask: Fetch JSON from an API and render items to the DOM  
with loading and error UI.  
• Instructions:  
o Ask AI to write fetch() logic, create DOM nodes safely,  
and add skeleton/loading text.

PROMPT:

Write a JavaScript program that fetches JSON data from an API, safely creates DOM nodes to display items, and shows a loading indicator while fetching. Include error handling to display a message if the request fails.

CODE GENERATED:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Fetch API Items</title>

  <style>

    body {

      font-family: 'Poppins', sans-serif;

      padding: 20px;

      background: #f4f5f7;

      color: #333;

    }

    h1 {

      text-align: center;

      margin-bottom: 20px;

      color: #1a1a2e;

    }

    #items {

      display: grid;

      grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));

      gap: 20px;

      max-width: 1000px;

      margin: auto;

    }

    .item-card {

      background: #fff;

      padding: 15px;

      border-radius: 10px;

      box-shadow: 0 4px 12px rgba(0,0,0,0.08);

      transition: transform 0.3s;

    }

    .item-card:hover {

      transform: translateY(-5px);

    }

    .loading, .error {

      text-align: center;

      font-size: 1.1rem;

      color: #555;

      margin-top: 40px;

    }

    .error { color: #e94560; }

  </style>

</head>

<body>

  <h1>API Items</h1>

  <div id="loading" class="loading">Loading items...</div>

  <div id="items"></div>

  <div id="error" class="error" style="display:none;"></div>

  <script>

    const apiURL = 'https://jsonplaceholder.typicode.com/posts?\_limit=10'; // Sample API

    const itemsContainer = document.getElementById('items');

    const loadingEl = document.getElementById('loading');

    const errorEl = document.getElementById('error');

    // Fetch items from API

    fetch(apiURL)

      .then(response => {

        if(!response.ok) throw new Error('Network response was not ok');

        return response.json();

      })

      .then(data => {

        loadingEl.style.display = 'none'; // hide loading

        renderItems(data);

      })

      .catch(err => {

        loadingEl.style.display = 'none';

        errorEl.textContent = `Error fetching data: ${err.message}`;

        errorEl.style.display = 'block';

      });

    // Render items safely

    function renderItems(items) {

      items.forEach(item => {

        const card = document.createElement('div');

        card.className = 'item-card';

        // Safely set text

        const title = document.createElement('h3');

        title.textContent = item.title;

        const body = document.createElement('p');

        body.textContent = item.body;

        card.appendChild(title);

        card.appendChild(body);

        itemsContainer.appendChild(card);

      });

    }

  </script>

</body>

</html>

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.