ASSIGNMENT-14.1

Name : P.Reshmitha Reddy

Rollno : 2403A510A7

Batch : 05

Course : AI Assisted Coding

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 portfolio with About, Projects, Contact sections, color palette, typography, and smooth scrolling.

Code :

```
<!DOCTYPE html>

<html lang="en">

<head>
```

```
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width,
initial-scale=1" />
  <title>Portfolio - Your Name</title>
  <style>
    /* Color palette */
    :root {
      --primary-color: #2c3e50;   /* Dark Blue */
      --secondary-color: #18bc9c; /* Turquoise */
      --accent-color: #e67e22;    /* Orange */
      --bg-color: #ecf0f1;        /* Light Gray */
      --text-color: #34495e;      /* Medium Blue-gray */
      --font-primary: 'Montserrat', sans-serif;
      --font-secondary: 'Roboto', sans-serif;
    }
    * {
      box-sizing: border-box;
      scroll-behavior: smooth;
    }
    body {
      margin: 0;
```

```
  font-family: var(--font-primary);
  background-color: var(--bg-color);
  color: var(--text-color);
  line-height: 1.6;
}
a {
  color: var(--secondary-color);
  text-decoration: none;
  transition: color 0.3s ease;
}
a:hover {
  color: var(--accent-color);
}
header {
  position: fixed;
  width: 100%;
  top: 0;
  background: var(--primary-color);
  padding: 1rem 2rem;
  display: flex;
  justify-content: space-between;
```

```
  align-items: center;
  z-index: 1000;
  box-shadow: 0 2px 4px rgba(0,0,0,0.2);
}
header h1 {
  margin: 0;
  font-size: 1.5rem;
  color: #fff;
  font-weight: 700;
}
nav ul {
  list-style: none;
  display: flex;
  gap: 1.5rem;
  margin: 0;
  padding: 0;
}
nav li {
  font-weight: 500;
  font-size: 1rem;
}
```

```
main {
  padding-top: 80px; /* space for fixed header */
  display: flex;
  flex-direction: column;
  gap: 4rem;
  max-width: 960px;
  margin: 0 auto;
  padding-left: 1rem;
  padding-right: 1rem;
}
section {
  background: #fff;
  padding: 2rem;
  border-radius: 12px;
  box-shadow: 0 8px 16px rgba(0,0,0,0.1);
}
h2 {
  font-family: var(--font-secondary);
  font-size: 2rem;
  border-bottom: 3px solid var(--secondary-color);
  padding-bottom: 0.3rem;
```

```
    margin-bottom: 1.5rem;
    color: var(--primary-color);
  }
  #about p {
    font-size: 1.1rem;
    max-width: 680px;
  }
  #projects {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(280px,
1fr));
    gap: 1.8rem;
  }
  .project-card {
    border: 2px solid var(--secondary-color);
    border-radius: 10px;
    padding: 1.5rem;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
    transition: transform 0.3s ease, box-shadow 0.3s ease;
```

```
  background: #fafafa;
}
.project-card:hover {
  transform: translateY(-10px);
  box-shadow: 0 12px 24px rgba(24, 188, 156, 0.3);
}
.project-card h3 {
  margin: 0 0 1rem 0;
  color: var(--accent-color);
  font-family: var(--font-secondary);
}
.project-card p {
  flex-grow: 1;
  color: var(--text-color);
  margin-bottom: 1rem;
}
.project-card a {
  align-self: flex-start;
  font-weight: 600;
}
#contact form {
```

```
  display: flex;
  flex-direction: column;
  gap: 1rem;
  max-width: 480px;
}
#contact label {
  font-weight: 600;
}
#contact input, #contact textarea {
  padding: 0.8rem;
  border: 2px solid var(--secondary-color);
  border-radius: 6px;
  font-size: 1rem;
  font-family: var(--font-primary);
  transition: border-color 0.3s ease;
}
#contact input:focus, #contact textarea:focus {
  outline: none;
  border-color: var(--accent-color);
  background: #fff;
}
```

```
#contact button {
  padding: 1rem;
  font-size: 1.1rem;
  border: none;
  border-radius: 8px;
  background: var(--secondary-color);
  color: #fff;
  font-weight: 700;
  cursor: pointer;
  transition: background-color 0.3s ease;
}
#contact button:hover {
  background: var(--accent-color);
}
@media (max-width: 600px) {
  nav ul {
    flex-direction: column;
    background: var(--primary-color);
    position: fixed;
    top: 60px;
    right: -200px;
```

```
    width: 180px;
    transition: right 0.3s ease;
    border-radius: 8px 0 0 8px;
    padding: 1rem;
  }
  nav ul.open {
    right: 0;
  }
  nav li {
    margin-bottom: 1rem;
    font-size: 1.2rem;
  }
  .menu-toggle {
    display: block;
    cursor: pointer;
    color: #fff;
    font-size: 1.4rem;
    user-select: none;
  }
}
.menu-toggle {
```

```
      display: none;
    }
  </style>
</head>
<body>
  <header>
    <h1>D. Sravika Reddy </h1>
    <nav>
      <span class="menu-toggle" id="menuToggle">☰</span>
      <ul id="navList">
        <li><a href="#about">About Me</a></li>
        <li><a href="#projects">Projects</a></li>
        <li><a href="#contact">Contact</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <section id="about">
      <h2>About Me</h2>
      <p>Hello! I’m P.Reshmitha Reddy a passionate developer/designer dedicated to creating beautiful and functional digital experiences. With experience in web
```

```
development, UI/UX design, and problem-solving, I enjoy turning ideas into reality using modern technologies.</p>
  </section>
  <section id="projects">
    <h2>Projects</h2>
    <div class="project-card">
      <h3>Project One</h3>
      <p>A responsive web app built with React that helps users track habits and stay productive.</p>
      <a href="#" target="_blank">View Project</a>
    </div>
    <div class="project-card">
      <h3>Project Two</h3>
      <p>An e-commerce website designed with user experience and accessibility in mind.</p>
      <a href="#" target="_blank">View Project</a>
    </div>
    <div class="project-card">
      <h3>Project Three</h3>
      <p>A Python data analytics tool that visualizes social media sentiment trends.</p>
      <a href="#" target="_blank">View Project</a>
```

```
    </div>
  </section>
  <section id="contact">
    <h2>Contact</h2>
    <form>
      <label for="name">Name</label>
      <input type="text" id="name" placeholder="Your name"
required/>


      <label for="email">Email</label>
      <input type="email" id="email"
placeholder="your.email@example.com" required/>
      <label for="message">Message</label>
      <textarea id="message" rows="5" placeholder="Your
message here" required></textarea>
      <button type="submit">Send</button>
    </form>
  </section>
 </main>
 <script>
  const menuToggle =
document.getElementById('menuToggle');
```

```
    const navList = document.getElementById('navList');
    menuToggle.addEventListener('click', () => {
      navList.classList.toggle('open');
    });
    navList.querySelectorAll('a').forEach(link =>
      link.addEventListener('click', () =>
navList.classList.remove('open'))
    );
  </script>
</body>
</html>
```

Output :

About Me

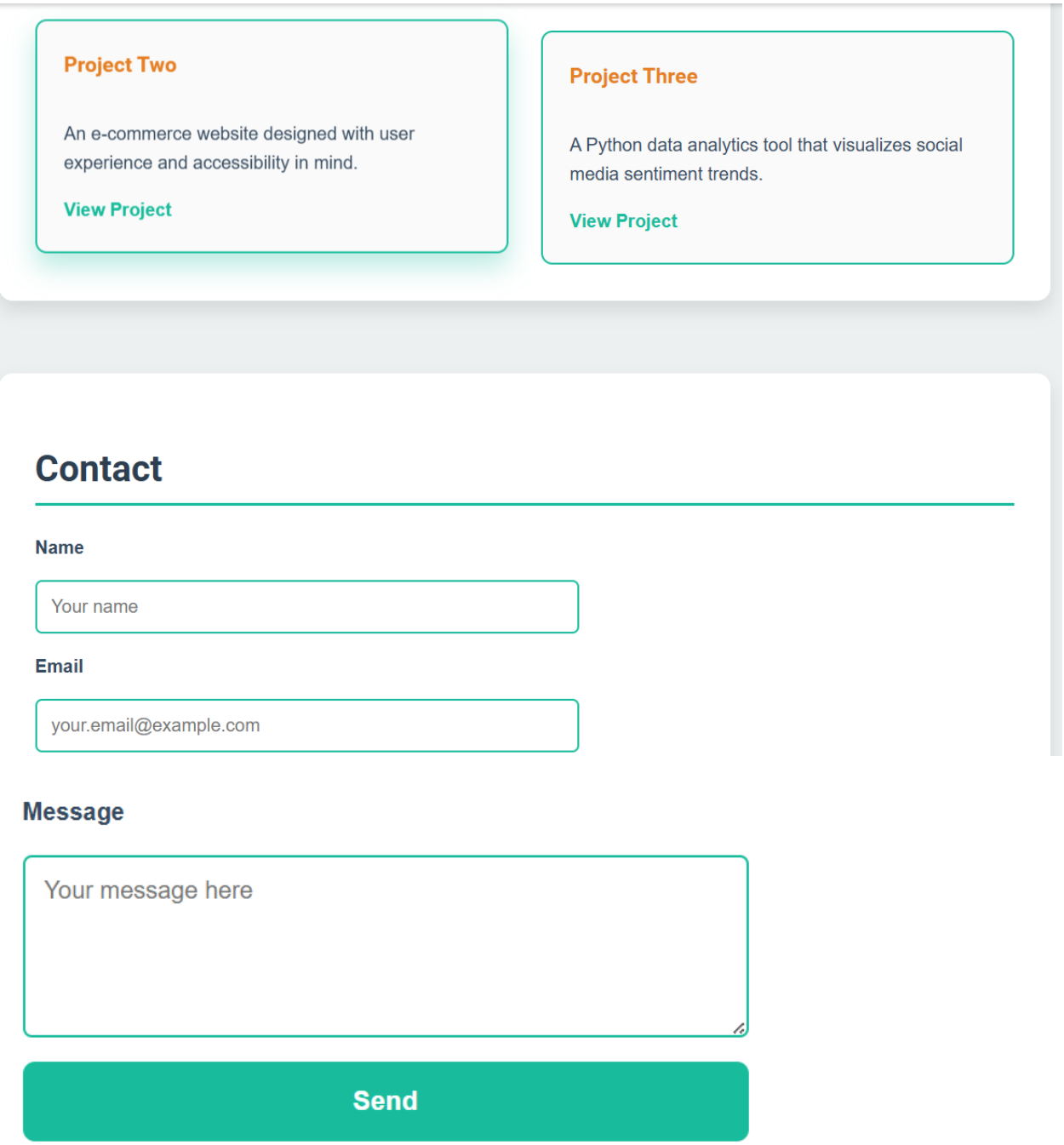
Hello! I'm P.Reshmitha Reddy a passionate developer/designer dedicated to creating beautiful and functional digital experiences. With experience in web development, UI/UX design, and problem-solving, I enjoy turning ideas into reality using modern technologies.

Projects

Project One

A responsive web app built with React that helps users track habits and stay productive.

View Project



Observation :

The web page is cleanly organized into sections: About Me, Projects, Products, and Contact, with a header and footer for navigation and information.The navigation bar includes links

to all major sections, including the newly added Products section.

The Products section uses cards to display product details, prices, and descriptions, matching the style of the Projects section for visual consistency.Responsive design is achieved using CSS Grid, Flexbox, and media queries, ensuring the layout adapts well to desktop, tablet, and mobile screens.The color palette and typography are modern and visually appealing, enhancing readability and user experience.The code is well-structured, easy to maintain, and demonstrates good practices for HTML and CSS layout.

Task 2 – Online Store Product Page

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 page with image, title, price, and 'Add to Cart' button using BEM, hover effects, and alert on add.

Code :

```
<!DOCTYPE html>

<html lang="en">
```

```
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Product Gallery - Cool Online Store</title>
  <style>
    :root {
      --primary-color: #1e88e5;
      --secondary-color: #e53935;
      --bg-color: #fafafa;
      --text-color: #333;
      --font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    }

    body {
      margin: 0;
      font-family: var(--font-family);
      background-color: var(--bg-color);
      color: var(--text-color);
      padding: 2rem;
      display: flex;
```

```
  justify-content: center;
}
.products {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));
  gap: 2rem;
  max-width: 1080px;
  width: 100%;
}
.product-card {
  background: #fff;
  border-radius: 12px;
  box-shadow: 0 10px 20px rgba(30, 136, 229, 0.15);
  overflow: hidden;
  display: flex;
  flex-direction: column;
  transition: transform 0.3s ease, box-shadow 0.3s ease;
}
.product-card:hover {
  transform: translateY(-8px);
  box-shadow: 0 20px 40px rgba(30, 136, 229, 0.3);
```

```
}
.product-card__image {
  width: 100%;
  aspect-ratio: 4 / 3;
  object-fit: cover;
  border-bottom: 1px solid #ddd;
}
.product-card__info {
  padding: 1.2rem 1.5rem;
  flex-grow: 1;
}

.product-card__title {
  font-size: 1.3rem;
  font-weight: 700;
  margin: 0 0 0.5rem 0;
  color: var(--primary-color);
}
.product-card__price {
  font-size: 1.1rem;
  font-weight: 600;
```

```
  margin: 0 0 1.2rem 0;
  color: var(--secondary-color);
}
.product-card__button {
  background-color: var(--primary-color);
  color: white;
  border: none;
  border-radius: 8px;
  padding: 0.8rem;
  width: 100%;
  font-size: 1.1rem;
  font-weight: 600;
  cursor: pointer;
  transition: background-color 0.3s ease;
}
.product-card__button:hover {
  background-color: #1565c0;
}
@media (max-width: 400px) {
  body {
    padding: 1rem;
```

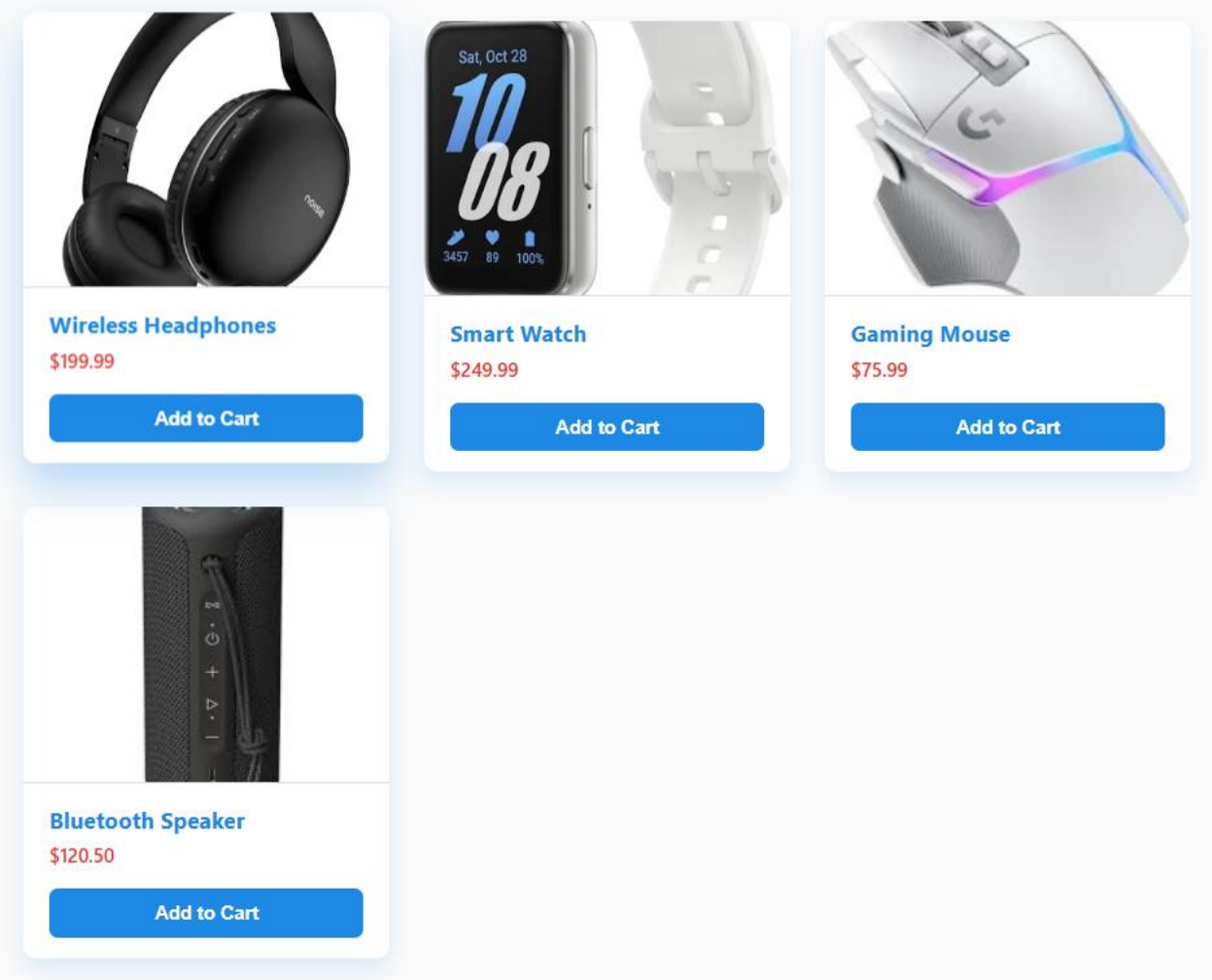
```
    }
    .products {
      grid-template-columns: 1fr;
      gap: 1.5rem;
    }
  }
 </style>
</head>
<body>
 <section class="products">
  <article class="product-card">
    <img src="https://images.unsplash.com/photo-1606813909317-a654bf4c568f?auto=format&fit=crop&w=600&q=80" alt="Wireless Headphones" class="product-card__image" />
    <div class="product-card__info">
      <h2 class="product-card__title">Wireless Headphones</h2>
      <p class="product-card__price">$199.99</p>
      <button class="product-card__button" data-name="Wireless Headphones">Add to Cart</button>
    </div>
  </article>
  <article class="product-card">
```

```
    <img src="https://images.unsplash.com/photo-1503602642458-232111445657?auto=format&fit=crop&w=600&q=80" alt="Smart Watch" class="product-card__image" />
    <div class="product-card__info">
      <h2 class="product-card__title">Smart Watch</h2>
      <p class="product-card__price">$249.99</p>
      <button class="product-card__button" data-name="Smart Watch">Add to Cart</button>
    </div>
  </article>

  <article class="product-card">
    <img src="https://images.unsplash.com/photo-1618224501159-1386e2d0e145?auto=format&fit=crop&w=600&q=80" alt="Gaming Mouse" class="product-card__image" />
    <div class="product-card__info">
      <h2 class="product-card__title">Gaming Mouse</h2>
      <p class="product-card__price">$75.99</p>
      <button class="product-card__button" data-name="Gaming Mouse">Add to Cart</button>
    </div>
  </article>
  <article class="product-card">
```

```
      <img src="https://images.unsplash.com/photo-1560448204-0c1bc7de7a7a?auto=format&fit=crop&w=600&q=80" alt="Bluetooth Speaker" class="product-card__image" />
      <div class="product-card__info">
        <h2 class="product-card__title">Bluetooth Speaker</h2>
        <p class="product-card__price">$120.50</p>
        <button class="product-card__button" data-name="Bluetooth Speaker">Add to Cart</button>
      </div>
    </article>
  </section>
  <script>
    const buttons = document.querySelectorAll('.product-card__button');
    buttons.forEach(button => {
      button.addEventListener('click', () => {
        const productName = button.getAttribute('data-name');
        alert(`Added "${productName}" to your cart!`);
      });
    });
  </script>
</body>
```

</html>

Output :



Observation :

The HTML page displays a visually appealing product gallery for an online store, using a modern color palette and clean typography.Products are presented as cards in a responsive grid layout, adapting smoothly to different screen sizes (desktop, tablet, mobile) via CSS Grid and media queries.Each product card includes an image, title, price, and an "Add to Cart" button.The JavaScript adds interactivity: clicking any

"Add to Cart" button triggers an alert showing the product name, confirming the action to the user.

Task 3 – Event Registration Form
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 :# Create an event registration form with name, email, phone, session fields — add JS validation, ARIA accessibility, and professional styling.

Code :

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <title>Conference Registration</title>

  <style>
```

```
    @import
url('https://fonts.googleapis.com/css2?family=Roboto:wght@400;7
00&display=swap');
    :root {
      --primary-color: #0073e6;
      --secondary-color: #005bb5;
      --error-color: #e63946;
      --bg-color: #f7f9fc;
      --text-color: #1a1a1a;
      --font-family: 'Roboto', sans-serif;
      --border-radius: 8px;
      --transition: 0.3s ease;
    }

    body {
      font-family: var(--font-family);
      background-color: var(--bg-color);
      color: var(--text-color);
      margin: 0;
      display: flex;
      justify-content: center;
```

```
  align-items: center;
  min-height: 100vh;
  padding: 20px;
}
.registration-form {
  background: #fff;
  padding: 2.5rem 3rem;
  border-radius: var(--border-radius);
  box-shadow: 0 8px 20px rgba(0,0,0,0.1);
  width: 100%;
  max-width: 450px;
  box-sizing: border-box;
}
.registration-form h1 {
  margin-top: 0;
  font-weight: 700;
  font-size: 1.8rem;
  margin-bottom: 1.5rem;
  text-align: center;
  color: var(--primary-color);
}
```

```
label {
  display: block;
  font-weight: 600;
  margin-bottom: 0.5rem;
  margin-top: 1rem;
}
input[type="text"],
input[type="email"],
input[type="tel"],
select {
  width: 100%;
  padding: 0.75rem 1rem;
  font-size: 1rem;
  border: 2px solid #ccc;
  border-radius: var(--border-radius);
  transition: border-color var(--transition);
}
input[type="text"]:focus,
input[type="email"]:focus,
input[type="tel"]:focus,
select:focus {
```

```
  outline: none;
  border-color: var(--primary-color);
  box-shadow: 0 0 8px rgba(0, 115, 230, 0.25);
}
.error-message {
  color: var(--error-color);
  font-size: 0.85rem;
  margin-top: 0.25rem;
  display: none;
}
button {
  margin-top: 2rem;
  width: 100%;
  background-color: var(--primary-color);
  border: none;
  color: white;
  padding: 0.9rem;
  font-size: 1.1rem;
  font-weight: 700;
  border-radius: var(--border-radius);
  cursor: pointer;
```

```
    transition: background-color var(--transition);
  }
  button:hover {
    background-color: var(--secondary-color);
  }
  /* Accessibility focus styles */
  input:focus-visible,
  select:focus-visible,
  button:focus-visible {
    outline: 3px solid var(--primary-color);
    outline-offset: 2px;
  }
 </style>
</head>
<body>

 <form class="registration-form" id="regForm" aria-labelledby="formTitle" novalidate>
  <h1 id="formTitle">Conference Registration</h1>
  <label for="name">Full Name <span aria-hidden="true" style="color: var(--error-color);">*</span></label>
```

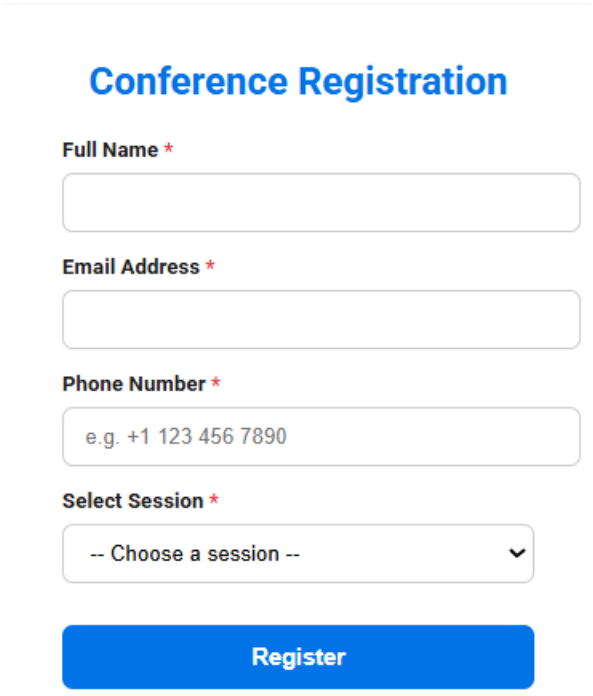
```
    <input type="text" id="name" name="name" aria-required="true" aria-describedby="nameError" required minlength="3" />

    <div class="error-message" id="nameError">Please enter at least 3 characters.</div>

    <label for="email">Email Address <span aria-hidden="true" style="color: var(--error-color);">*</span></label>

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

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

    <label for="phone">Phone Number <span aria-hidden="true" style="color: var(--error-color);">*</span></label>

    <input type="tel" id="phone" name="phone" aria-required="true" aria-describedby="phoneError"

      placeholder="e.g. +1 123 456 7890" required pattern="^\+?\d{1,4}[\s.-]?\(?\d+\)?([\s.-]?\d+)*$" />

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

    <label for="session">Select Session <span aria-hidden="true" style="color: var(--error-color);">*</span></label>

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

      <option value="">-- Choose a session --</option>
```

```
    <option value="morning">Morning: Future of Technology</option>
    <option value="afternoon">Afternoon: AI and Data Science</option>
    <option value="evening">Evening: Networking & Career Growth</option>
  </select>
  <div class="error-message" id="sessionError">Please select a session.</div>
  <button type="submit" aria-label="Submit registration form">Register</button>
 </form>
 <script>
document.getElementById('regForm').addEventListener('submit', function(event) {
    event.preventDefault();
    const errors = this.querySelectorAll('.error-message');
    errors.forEach(error => error.style.display = 'none');
    let valid = true;
    const nameInput = this.name;
    if (!nameInput.value || nameInput.value.trim().length < 3) {
      document.getElementById('nameError').style.display = 'block';
      valid = false;
```

```
    }
    const emailInput = this.email;
    const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!emailInput.value || !emailPattern.test(emailInput.value)) {
      document.getElementById('emailError').style.display = 'block';
      valid = false;
    }
    const phoneInput = this.phone;
    const phonePattern = /^\+?\d{1,4}[\s.-]?\(?\d+\)?([\s.-]?\d+)*$/;
    if (!phoneInput.value || !phonePattern.test(phoneInput.value))
{
      document.getElementById('phoneError').style.display =
'block';
      valid = false;
    }
    const sessionInput = this.session;
    if (!sessionInput.value) {
      document.getElementById('sessionError').style.display =
'block';
      valid = false;
    }
    if (valid) {
```

```
        alert(`Thank you, ${nameInput.value}, for registering for the ${sessionInput.options[sessionInput.selectedIndex].text} session!`);
        this.reset();
      }
    });
  </script>
</body>
</html>
```

Output :



Observation :

The form is visually appealing and uses a modern color palette and the Roboto font for readability.All fields (name, email, phone, session) are required, with clear labels and error indicators for accessibility.Responsive design centers the form and adapts well to different screen sizes.JavaScript provides client-side validation for minimum name length, email format, phone pattern, and session selection.

Task Description #4 (Data – Fetch API & Render List with Loading/Error States)

• Task: 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 JS using fetch() to load API data, show items, and handle loading/error states.

Code :

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />
```

```
  <title>API Data Fetch with Pagination</title>
  <style>
    @import
url('https://fonts.googleapis.com/css2?family=Inter:wght@400;600
&display=swap');
    :root {
      --primary-color: #4a90e2;
      --error-color: #e74c3c;
      --bg-color: #f5f7fa;
      --text-color: #333;
      --font-family: 'Inter', sans-serif;
      --border-radius: 8px;
      --shadow: 0 4px 12px rgba(74, 144, 226, 0.2);
    }
    body {
      margin: 0;
      font-family: var(--font-family);
      background-color: var(--bg-color);
      color: var(--text-color);
      display: flex;
      justify-content: center;
```

```
  padding: 2rem;
  min-height: 100vh;
  box-sizing: border-box;
  flex-direction: column;
  align-items: center;
}
.container {
  max-width: 600px;
  width: 100%;
  background: white;
  border-radius: var(--border-radius);
  padding: 1.5rem 2rem;
  box-shadow: var(--shadow);
  box-sizing: border-box;
  margin-bottom: 1.5rem;
}
h1 {
  margin-top: 0;
  color: var(--primary-color);
  font-weight: 600;
  text-align: center;
```

```
  margin-bottom: 1rem;
}
#statusMessage {
  text-align: center;
  font-size: 1rem;
  padding: 0.8rem 0;
  color: var(--primary-color);
  min-height: 1.5rem;
}

#statusMessage.error {
  color: var(--error-color);
}
ul.data-list {
  list-style: none;
  padding: 0;
  margin: 0;
}
ul.data-list li {
  background-color: #f0f4ff;
  margin-bottom: 1rem;
```

```
    border-radius: var(--border-radius);
    padding: 1rem 1.2rem;
    box-shadow: inset 0 0 6px rgba(74, 144, 226, 0.15);
    font-weight: 500;
    font-size: 1rem;
  }

  .skeleton {
    background: linear-gradient(90deg, #e0e6ff 25%, #c0ccff 37%, #e0e6ff 63%);
    animation: loading 1.4s infinite ease-in-out;
    border-radius: var(--border-radius);
    height: 1.2rem;
    margin-bottom: 1rem;
    width: 100%;
    max-width: 450px;
  }
  @keyframes loading {
    0% { background-position: 200% 0; }
    100% { background-position: -200% 0; }
  }
```

```
.pagination {
  display: flex;
  gap: 1rem;
  justify-content: center;
}
.pagination button {
  background-color: var(--primary-color);
  border: none;
  color: white;
  border-radius: var(--border-radius);
  padding: 0.6rem 1.2rem;
  font-size: 1rem;
  font-weight: 600;
  cursor: pointer;
  transition: background-color 0.3s ease;
  min-width: 80px;
}
.pagination button:disabled {
  background-color: #a0c3ff;
  cursor: not-allowed;
}
```

```
    .pagination button:hover:not(:disabled) {
      background-color: #3a6bcc;
    }
  </style>
</head>
<body>
  <div class="container" role="region" aria-live="polite" aria-busy="false">
    <h1>API Data Fetch</h1>
    <div id="statusMessage" aria-live="assertive" aria-atomic="true"></div>
    <ul class="data-list" id="dataList" aria-label="List of items"></ul>
  </div>
  <div class="pagination" role="navigation" aria-label="Pagination navigation">
    <button id="prevBtn" aria-label="Previous page" disabled>Previous</button>
    <button id="nextBtn" aria-label="Next page">Next</button>
  </div>
  <script>
    const API_URL = 'https://jsonplaceholder.typicode.com/posts';
    const dataList = document.getElementById('dataList');
```

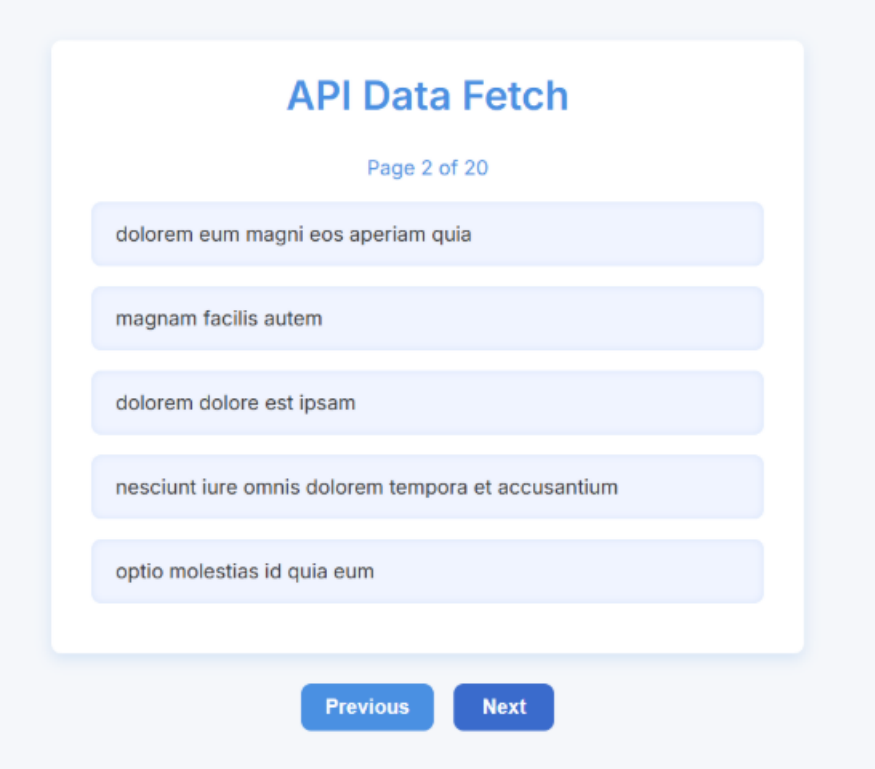
```
    const statusMessage =
document.getElementById('statusMessage');
    const container = document.querySelector('.container');
    const prevBtn = document.getElementById('prevBtn');
    const nextBtn = document.getElementById('nextBtn');
    const ITEMS_PER_PAGE = 5;
    let currentPage = 1;
    let totalItems = 0;
    function showSkeletons(count = ITEMS_PER_PAGE) {
      dataList.innerHTML = '';
      for (let i = 0; i < count; i++) {
        const skeleton = document.createElement('li');
        skeleton.className = 'skeleton';
        dataList.appendChild(skeleton);
      }
    }
    function clearUI() {
      dataList.innerHTML = '';
      statusMessage.textContent = '';
      statusMessage.classList.remove('error');
    }
```

```
function renderItems(items) {
  clearUI();
  items.forEach(item => {
    const li = document.createElement('li');
    li.textContent = item.title;
    dataList.appendChild(li);
  });
}
function showError(message) {
  clearUI();
  statusMessage.textContent = message;
  statusMessage.classList.add('error');
}

async function fetchData(page) {
  container.setAttribute('aria-busy', 'true');
  statusMessage.classList.remove('error');
  statusMessage.textContent = 'Loading data...';
  showSkeletons();
  try {
```

```
      const response = await
fetch(`${API_URL}?_limit=${ITEMS_PER_PAGE}&_page=${page}`);
      if (!response.ok) {
        throw new Error(`Error loading data: ${response.status}
${response.statusText}`);
      }
      const data = await response.json();
      totalItems = parseInt(response.headers.get('X-Total-Count'))
|| 0;
      renderItems(data);
      statusMessage.textContent = `Page ${page} of
${Math.ceil(totalItems / ITEMS_PER_PAGE)}`;
      prevBtn.disabled = page === 1;
      nextBtn.disabled = page >= Math.ceil(totalItems /
ITEMS_PER_PAGE);
    } catch (error) {
      showError(error.message);
    } finally {
      container.setAttribute('aria-busy', 'false');
    }
  }
  prevBtn.addEventListener('click', () => {
```

```
    if (currentPage > 1) {
      currentPage--;
      fetchData(currentPage);
    }
  });
  nextBtn.addEventListener('click', () => {
    if (currentPage < Math.ceil(totalItems / ITEMS_PER_PAGE)) {
      currentPage++;
      fetchData(currentPage);
    }
  });
  fetchData(currentPage);
 </script>
</body>
</html>
```

Output :



Observation :

This code is a well-designed web application that fetches data from an external API and displays it on the page with pagination, loading effects, and error handling. It uses the

fetch() API to retrieve JSON data from https://jsonplaceholder.typicode.com/posts, showing five items per page with “Next” and “Previous” navigation buttons. While the data is being fetched, skeleton loaders and a “Loading data...” message appear to give users visual feedback. The interface is styled using modern CSS features like variables, box shadows, and Google Fonts, resulting in a clean and professional design. Accessibility is also considered with ARIA roles, labels, and live regions for screen readers. Error handling is implemented effectively, displaying clear messages when issues occur. Overall, the code is well-structured, readable, and demonstrates good practices in asynchronous programming, DOM manipulation, and UI design.