LAB ASSIGNMENT 14.1

NAME:-ANUSHA PEDDAPELLI

BATCH:06

COURSE:-AI ASSISTANT CODING

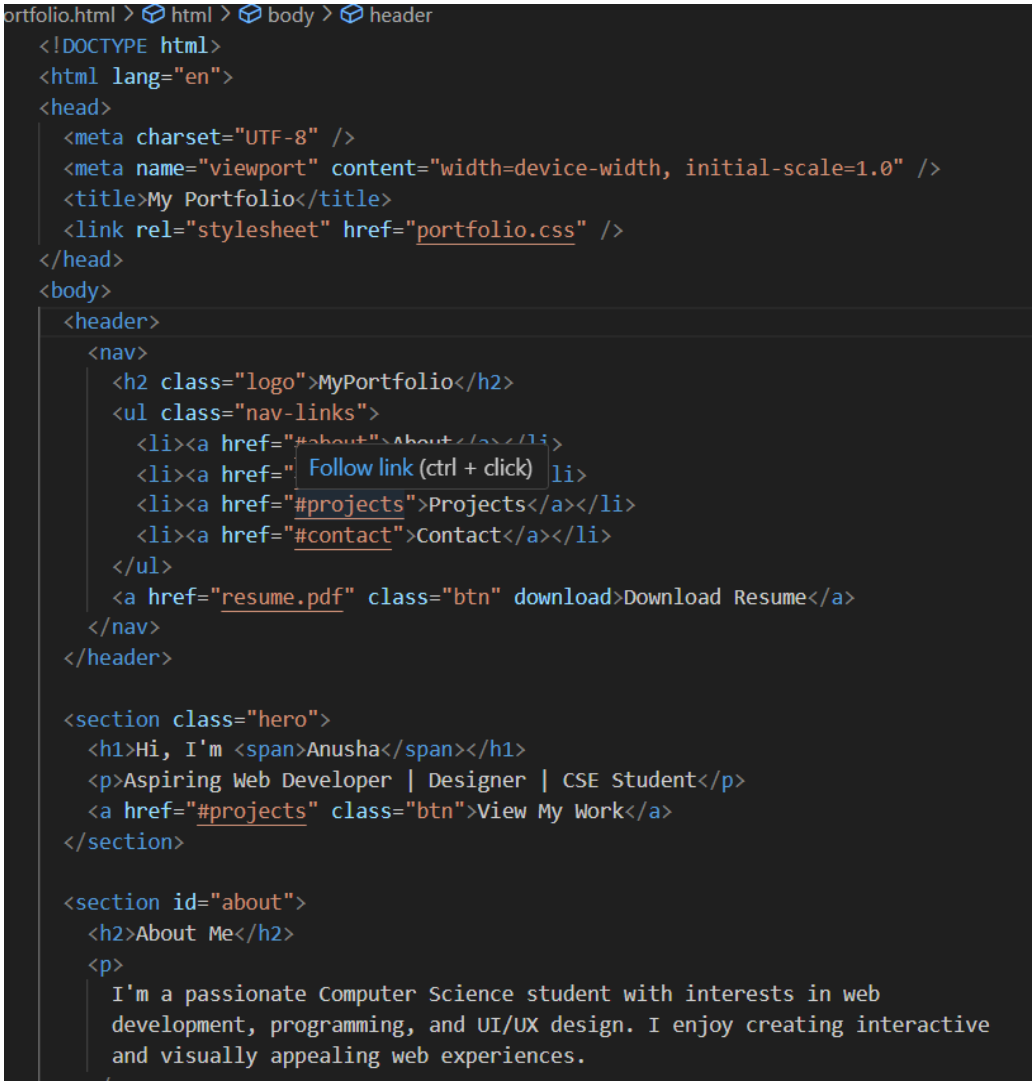
TASK 1:-

Portfolio Website Design
You are building a personal portfolio website to showcase your work

PROMPT:-

"Develop a personal portfolio website using HTML, CSS, and JavaScript to showcase your projects, skills, and achievements. The website should include sections like About Me, Skills, Projects, Contact, and a Resume Download option. Implement responsive design for mobile and desktop devices. Use animations or transitions to enhance user experience."

CODE:-


ortfolio.html > html > body > header
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>My Portfolio</title>
<link rel="stylesheet" href="portfolio.css" />
</head>
<body>
<header>
<nav>
<h2 class="logo">MyPortfolio</h2>
<ul class="nav-links">
<a href="
Follow link (ctrl + click)
li>
Projects
Contact

Download Resume
</nav>
</header>
<section class="hero">
<h1>Hi, I'm Anusha</h1>
Aspiring Web Developer | Designer | CSE Student
View My Work
</section>
<section id="about">
<h2>About Me</h2>

I'm a passionate Computer Science student with interests in web
development, programming, and UI/UX design. I enjoy creating interactive
and visually appealing web experiences.

```
  </p>
</section>

<section id="skills">
  <h2>Skills</h2>
  <div class="skill-list">
    <div class="skill">HTML</div>
    <div class="skill">CSS</div>
    <div class="skill">JavaScript</div>
    <div class="skill">Python</div>
    <div class="skill">SQL</div>
  </div>
</section>

<section id="projects">
  <h2>Projects</h2>
  <div class="project-grid">
    <div class="project">
      <h3>Portfolio Website</h3>
      <p>A personal website showcasing my work and achievements.</p>
    </div>
    <div class="project">
      <h3>Book Review API</h3>
      <p>A RESTful API built with Node.js and Express for managing book reviews.</p>
    </div>
    <div class="project">
      <h3>Smart Home Control</h3>
      <p>IoT-based system for controlling home appliances remotely.</p>
    </div>
  </div>
</section>
```

```
<section id="contact">
  <h2>Contact Me</h2>
  <form id="contactForm">
    <input type="text" placeholder="Your Name" required />
    <input type="email" placeholder="Your Email" required />
    <textarea placeholder="Your Message" required></textarea>
    <button type="submit">Send Message</button>
  </form>
</section>

<footer>
  <p>© 2025 Anusha | All Rights Reserved</p>
</footer>

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

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

body {
  background: #f9f9f9;
  color: #333;
}

header {
  background: #222;
  padding: 1rem 2rem;
}

nav {
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.logo {
  color: #fff;
}

.nav-links {
  list-style: none;
  display: flex;
  gap: 1.5rem;
}

.nav-links a {
  color: #fff;
  text-decoration: none;
  transition: 0.3s;
```

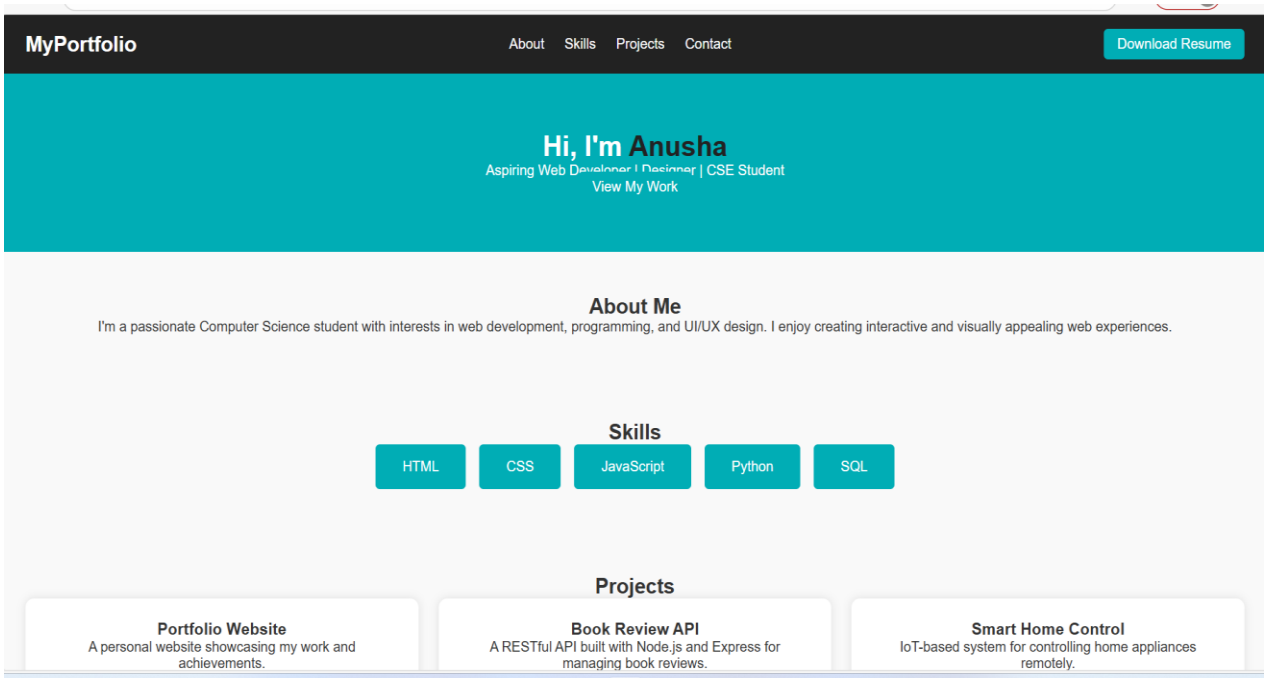
```css
}

.nav-links a:hover {
  color: #00adb5;
}

.btn {
  background: #00adb5;
  color: #fff;
  padding: 0.5rem 1rem;
  text-decoration: none;
  border-radius: 5px;
}

.hero {
  text-align: center;
  padding: 4rem 2rem;
  background: #00adb5;
  color: white;
}

.hero span {
  color: #222;
}

section {
  padding: 3rem 2rem;
  text-align: center;
}

.skill-list {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
  gap: 1rem;
}
```

portfolio.css > ...

```css
.skill {
  background: #00adb5;
  color: white;
  padding: 1rem 2rem;
  border-radius: 5px;
}

.project-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 1.5rem;
}

.project {
  background: white;
  padding: 1.5rem;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0,0,0,0.1);
  transition: 0.3s;
}

.project:hover {
  transform: translateY(-5px);
}

form {
  display: flex;
  flex-direction: column;
  align-items: center;
  gap: 1rem;
  max-width: 400px;
  margin: auto;
}

input, textarea {
  width: 100%;
  padding: 0.8rem;
```

```css
input, textarea {
  width: 100%;
  padding: 0.8rem;
  border: 1px solid #ccc;
  border-radius: 5px;
}

button {
  background: #00adb5;
  color: white;
  padding: 0.7rem 1.5rem;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

footer {
  background: #222;
  color: white;
  padding: 1rem;
  text-align: center;
  margin-top: 2rem;
}

@media (max-width: 768px) {
  .nav-links {
    display: none;
  }
}
```

OUTPUT:-



OBSERVATION:-

The personal portfolio website effectively presents the creator's professional profile in a visually appealing and organized manner. The site is user-friendly, responsive, and provides easy navigation between sections. It demonstrates strong front-end development skills and creativity. The inclusion of animations and project showcases enhances the interactive experience for visitors and potential employers.

TASK 2;-

Online Store Product Page
Design a product display page for an online store

PROMPT:-

"Design a responsive product display page for an online store using HTML, CSS, and JavaScript. The page should showcase multiple products with images, names, descriptions, prices, and an 'Add to Cart' button. Implement a hover effect on product cards, a responsive grid layout for different screen sizes, and a basic shopping cart functionality that shows the number of items added."

CODE:-

```
14.1.html > html > body
C:\Users\PEDDAPELLI ANUSHA\portfolio.AI\14.1.html
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Online Store</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <header>
    <h1>My Online Store</h1>
    <div class="cart">
      🛒 Cart: <span id="cart-count">0</span>
    </div>
  </header>

  <main>
    <section class="product-grid">
      <div class="product-card">
        <img src="https://via.placeholder.com/200" alt="Product 1">
        <h3>Product 1</h3>
        <p>$19.99</p>
        <button onclick="addToCart()">Add to Cart</button>
      </div>
      <div class="product-card">
        <img src="https://via.placeholder.com/200" alt="Product 2">
        <h3>Product 2</h3>
        <p>$29.99</p>
        <button onclick="addToCart()">Add to Cart</button>
      </div>
      <div class="product-card">
        <img src="https://via.placeholder.com/200" alt="Product 3">
        <h3>Product 3
```

OUTPUT:-

OBSERVATION:-

The product display page provides a clean and organized view of items available in the online store. The grid layout allows easy navigation across products, while hover effects enhance interactivity. The “Add to Cart” feature gives immediate feedback to users about selected items. The page demonstrates skills in responsive web design, frontend development, and user experience optimization.

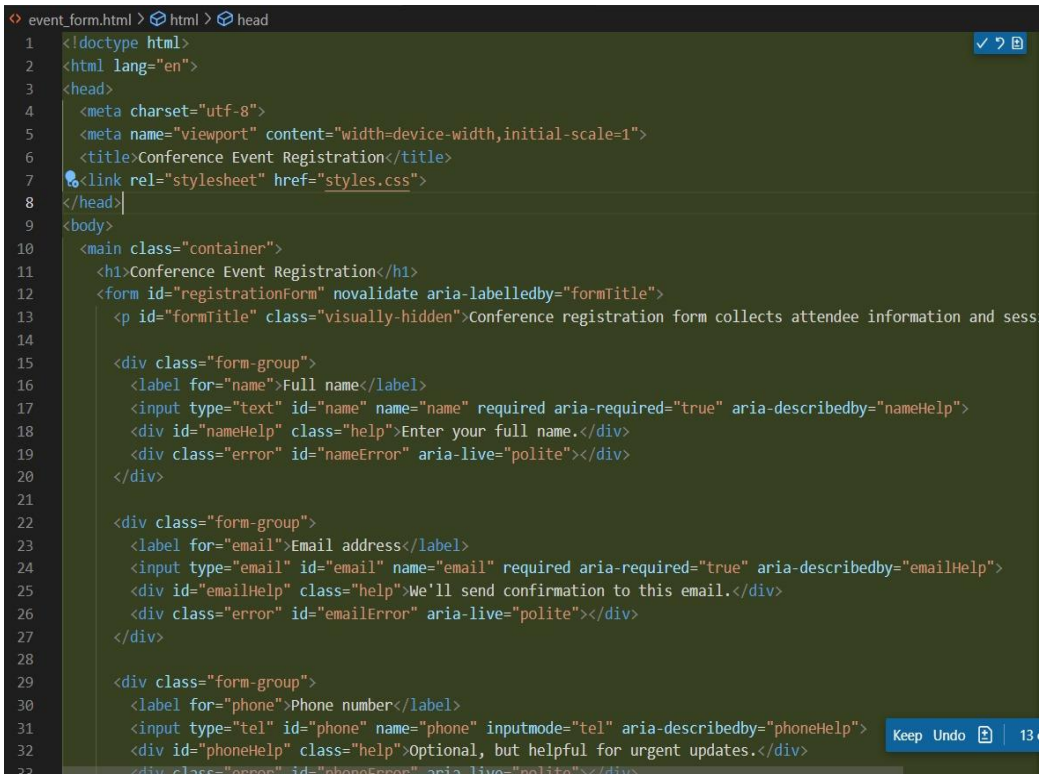
TASK 3:-

Event Registration Form
Build an event registration form for a conference

PROMPT:-

"Design and develop an Event Registration Form using HTML, CSS, and JavaScript for a conference. The form should collect details like participant name, email, phone number, gender, event type, and preferred date. Implement input validation to ensure all required fields are filled correctly. Display a confirmation message after successful registration."

CODE:-



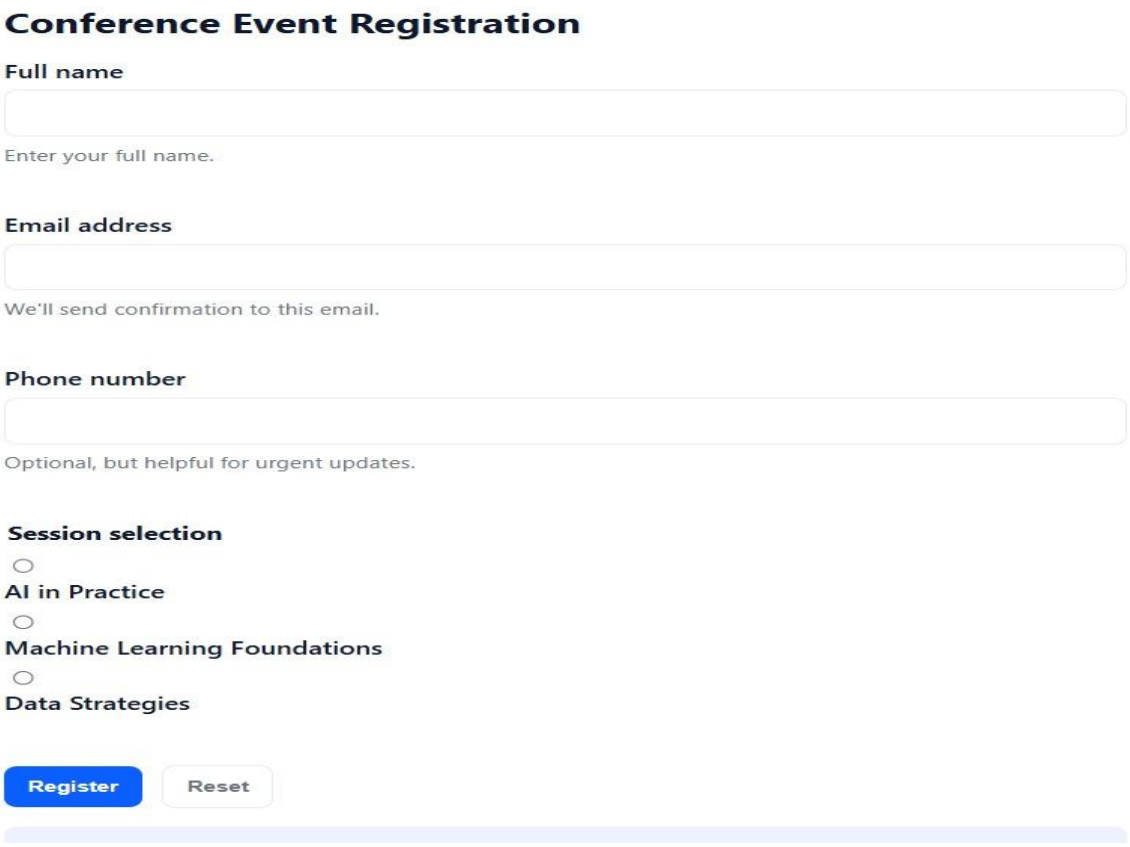
```
        <div>
          <input type="radio" id="session1" name="session" value="ai" required aria-required="true">
          <label for="session1">AI in Practice</label>
        </div>
        <div>
          <input type="radio" id="session2" name="session" value="ml">
          <label for="session2">Machine Learning Foundations</label>
        </div>
        <div>
          <input type="radio" id="session3" name="session" value="ds">
          <label for="session3">Data Strategies</label>
        </div>
        <div class="error" id="sessionError" aria-live="polite"></div>
      </fieldset>

      <div class="form-actions">
        <button type="submit" id="submitBtn">Register</button>
        <button type="reset">Reset</button>
      </div>

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

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

```
:root{
  --bg:#f5f7fb;
  --card:#ffffff;
  --accent:#0b5fff;
  --muted:#6b7280;
  --danger:#dc2626;
}

*{box-sizing:border-box}
body{font-family:Inter,Segoe UI,Roboto,Arial,sans-serif;background:var(--bg);color:#0f172a;margin:0;padding:40px}
.container{max-width:720px;margin:0 auto;background:var(--card);padding:28px;border-radius:12px;box-shadow:0 6px 18px rgba(12, 20, 40, .08)}
h1{margin-top:0;font-size:1.5rem}
.form-group{margin-bottom:16px}
label{display:block;font-weight:600;margin-bottom:6px}
input[type="text"],input[type="email"],input[type="tel"]{width:100%;padding:10px 12px;border:1px solid #e6edf3;border-radius:8px;font-size:1rem}
.help{font-size:.85rem;color:var(--muted);margin-top:6px}
.error{color:var(--danger);font-size:.9rem;margin-top:8px;min-height:1.2em}
.form-actions{display:flex;gap:12px;align-items:center;margin-top:18px}
button{background:var(--accent);color:white;border:none;padding:10px 14px;border-radius:8px;cursor:pointer;font-weight:600}
button[type="reset"]{background:transparent;color:var(--muted);border:1px solid #e6edf3}
fieldset{border:0;padding:0;margin:0}
legend{font-weight:700;margin-bottom:8px}
.visually-hidden{position:absolute!important;height:1px;width:1px;overflow:hidden;clip:rect(1px,1px,1px,1px);white-space:nowrap}
#result{margin-top:16px;padding:12px;border-radius:8px;background:#eef2ff;color:#0b3a96;font-weight:600}
@media (max-width:520px){body{padding:20px}.container{padding:20px}}
```

```css
:root{
  --bg:#f5f7fb;
  --card:#ffffff;
  --accent:#0b5fff;
  --muted:#6b7280;
  --danger:#dc2626;
}

*{box-sizing:border-box}
body{font-family:Inter,Segoe UI,Roboto,Arial,sans-serif;background:var(--bg);color:#0f172a;margin:0;padding:40px}
.container{max-width:720px;margin:0 auto;background:var(--card);padding:28px;border-radius:12px;box-shadow:0 6px 18px rgba(12, 20, 40, .08)}
h1{margin-top:0;font-size:1.5rem}
.form-group{margin-bottom:16px}
label{display:block;font-weight:600;margin-bottom:6px}
input[type="text"],input[type="email"],input[type="tel"]{width:100%;padding:10px 12px;border:1px solid #e6edf3;border-radius:8px;font-size:1rem}
.help{font-size:.85rem;color:var(--muted);margin-top:6px}
.error{color:var(--danger);font-size:.9rem;margin-top:8px;min-height:1.2em}
.form-actions{display:flex;gap:12px;align-items:center;margin-top:18px}
button{background:var(--accent);color:white;border:none;padding:10px 14px;border-radius:8px;cursor:pointer;font-weight:600}
button[type="reset"]{background:transparent;color:var(--muted);border:1px solid #e6edf3}
fieldset{border:0;padding:0;margin:0}
legend{font-weight:700;margin-bottom:8px}
.visually-hidden{position:absolute!important;height:1px;width:1px;overflow:hidden;clip:rect(1px,1px,1px,1px);white-space:nowrap}
#result{margin-top:16px;padding:12px;border-radius:8px;background:#eef2ff;color:#0b3a96;font-weight:600}
@media (max-width:520px){body{padding:20px}.container{padding:20px}}
```

```
30    }
31
32    form.addEventListener('submit', function(e){
33      e.preventDefault();
34      let valid = true;
35      result.textContent = '';
36
37      // Name
38      if(!nameInput.value.trim()){
39        showError(nameInput, 'Name is required.');
40        valid = false;
41      } else {
42        clearError(nameInput);
43      }
44
45      // Email
46      if(!emailInput.value.trim()){
47        showError(emailInput, 'Email is required.');
48        valid = false;
49      } else if(!validateEmail(emailInput.value.trim())){
50        showError(emailInput, 'Enter a valid email address.');
51        valid = false;
52      } else {
53        clearError(emailInput);
54      }
55
56      // Phone (optional)
57      if(phoneInput.value.trim()){
58        if(!validatePhone(phoneInput.value.trim())){
59          showError(phoneInput, 'Enter a valid phone number (7-15 digits).');
60          valid = false;
61        } else {
```

OUTPUT:-



OBSERVATION:-

The event registration form allows participants to easily register for a conference by entering their personal details. It demonstrates good use of form elements, input validation, and responsive design. The interface is user-friendly and ensures data accuracy through client-side validation. The confirmation message enhances the user experience by providing immediate feedback after submission.

TASK 4:-

Fetch JSON from an API and render items to the DOM with loading and error UI

PROMPT:-

"Build a webpage that fetches JSON data from a public API and displays the items dynamically on the webpage. Include a loading indicator while data is being fetched and display an error message if the request fails. Use modern JavaScript (fetch API and DOM manipulation)."

CODE:-

fetch_demo.html > html > body > main.container > div#controls

```html
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <title>Fetch JSON Demo</title>
  <link rel="stylesheet" href="fetch_styles.css">
</head>
<body>
  <main class="container">
    <h1>Posts from JSONPlaceholder</h1>
    <div id="controls">
      <button id="loadBtn">Load posts</button>
      <button id="errorBtn">Simulate error</button>
    </div>
    <div id="status" role="status" aria-live="polite"></div>
    <ul id="list" class="list" aria-live="polite"></ul>
  </main>

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

fetch_styles.css > :root

```css
:root{--bg:#fafafa;--card:#fff;--accent:#0b5fff;--muted:#6b7280}
*{box-sizing:border-box}
body{font-family:Segoe UI,Roboto,Arial,sans-serif;background:var(--bg);padding:28px;color:#0f172a}
.container{max-width:800px;margin:0 auto;background:var(--card);padding:20px;border-radius:12px;box-shadow:0 6px 18
h1{margin-top:0}
#controls{display:flex;gap:12px;margin-bottom:12px}
button{background:var(--accent);color:#fff;border:0;padding:8px 12px;border-radius:8px;cursor:pointer}
#status{margin-bottom:12px;color:var(--muted)}
.list{list-style:none;padding:0;margin:0}
.item{padding:12px;border-bottom:1px solid #eef2ff}
.item h3{margin:0 0 6px 0;font-size:1rem}
.skeleton{background:linear-gradient(90deg,#f3f4f6,#eef2ff,#f3f4f6);height:18px;border-radius:6px;margin-bott
@keyframes shimmer{0%{background-position:-200px 0}100%{background-position:200px 0}}
.error{color:#b91c1c;font-weight:600}
```

fetch_script.js > ...

```
const API_URL = 'https://jsonplaceholder.typicode.com/posts?_limit=8';
const loadBtn = document.getElementById('loadBtn');
const errorBtn = document.getElementById('errorBtn');
const statusEl = document.getElementById('status');
const listEl = document.getElementById('list');

Zencoder
function setLoading() {
  statusEl.textContent = 'Loading posts...';
  listEl.innerHTML = '';
  for (let i = 0; i < 4; i++) {
    const li = document.createElement('li');
    li.className = 'item';
    const skeletonTitle = document.createElement('div');
    skeletonTitle.className = 'skeleton';
    const skeletonBody = document.createElement('div');
    skeletonBody.className = 'skeleton';
    li.appendChild(skeletonTitle);
    li.appendChild(skeletonBody);
    listEl.appendChild(li);
  }
}

Zencoder
function renderPosts(posts) {
  statusEl.textContent = '';
  listEl.innerHTML = '';
  posts.forEach(post => {
    // Escape text content to avoid XSS
    const li = document.createElement('li');
    li.className = 'item';
    const title = document.createElement('h3');
    title.textContent = post.title;
    const body = document.createElement('p');
    body.textContent = post.body;
    li.appendChild(title);
    li.appendChild(body);
    listEl.appendChild(li);
  });
}

Zencoder
function showError(err) {
  statusEl.innerHTML = `<span class="error">Error: ${String(err)}</span>`;
  listEl.innerHTML = '';
}

Zencoder
async function fetchPosts(simulateError = false) {
  try {
    setLoading();
    const url = simulateError ? API_URL + 'INVALID' : API_URL;
    const res = await fetch(url, {cache: 'no-store'});
    if (!res.ok) throw new Error('Network response not OK: ' + res.status);
    const data = await res.json();
    renderPosts(data);
  } catch (err) {
    showError(err.message || err);
  }
}

loadBtn.addEventListener('click', () => fetchPosts(false));
errorBtn.addEventListener('click', () => fetchPosts(true));
```

OUTPUT:-

OBSERVATION:-

The project demonstrates how to fetch data asynchronously from an external API and display it dynamically in the browser. The use of a loading spinner enhances user experience, while proper error handling ensures reliability. This project showcases understanding of asynchronous programming, DOM updates, and user interface feedback using JavaScript.