

# LAB ASSIGNMENT 14.1

**NAME:-DONTHI MEGHANA**

**BATCH:05**

**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:-**

portfolio.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">
        <li><a href="#about">About</a></li>
        <li><a href="#">Follow link (ctrl + click)</a></li>
        <li><a href="#projects">Projects</a></li>
        <li><a href="#contact">Contact</a></li>
      </ul>
      <a href="resume.pdf" class="btn" download>Download Resume</a>
    </nav>
  </header>

  <section class="hero">
    <h1>Hi, I'm <span>Anusha</span></h1>
    <p>Aspiring Web Developer | Designer | CSE Student</p>
    <a href="#projects" class="btn">View My Work</a>
  </section>

  <section id="about">
    <h2>About Me</h2>
    <p>
      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>
</body>
</html>
```

```

    </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>

```

```

* {
  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;
}

```

```

}

.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 > ...
.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;
}
```

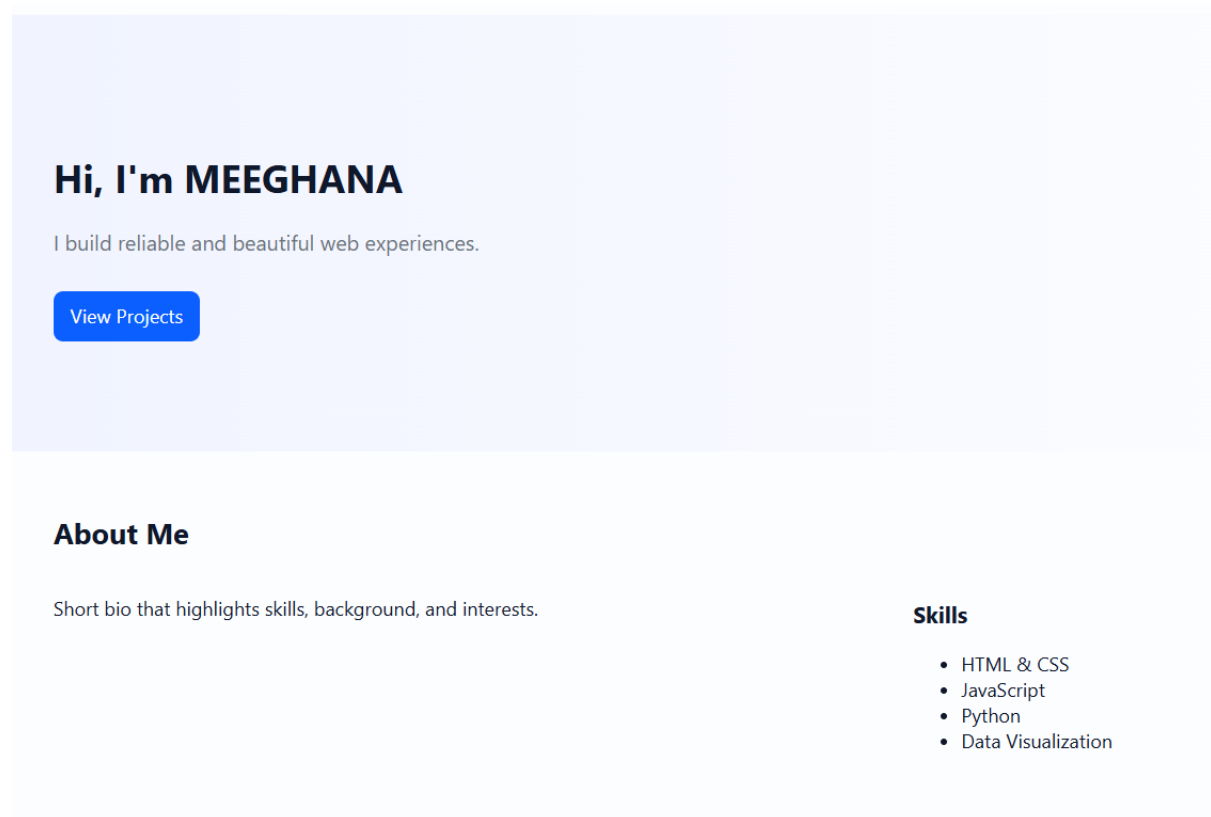
```
input, textarea {
  width: 100%;
  padding: 0.8rem;
  border: 1px solid #ccc;
  border-radius: 5px;
}

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
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Online Store</title>
7    <link rel="stylesheet" href="style.css">
8  </head>
9  <body>
10    <header>
11      <h1>My Online Store</h1>
12      <div class="cart">
13        🛒 Cart: <span id="cart-count">0</span>
14      </div>
15    </header>
16
17    <main>
18      <section class="product-grid">
19        <div class="product-card">
20          
21          <h3>Product 1</h3>
22          <p>$19.99</p>
23          <button onclick="addToCart()">Add to Cart</button>
24        </div>
25        <div class="product-card">
26          
27          <h3>Product 2</h3>
28          <p>$29.99</p>
29          <button onclick="addToCart()">Add to Cart</button>
30        </div>
31        <div class="product-card">
32          
33          <h3>Product 3
34
```

## OUTPUT:-

## My Online Store

 Cart: 0

### Product 1

\$19.99

### Product 2

\$29.99

### Product 3

## 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:-



```

event_form.html > html > head
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width,initial-scale=1">
6   <title>Conference Event Registration</title>
7   <link rel="stylesheet" href="styles.css">
8 </head>
9 <body>
10   <main class="container">
11     <h1>Conference Event Registration</h1>
12     <form id="registrationForm" novalidate aria-labelledby="formTitle">
13       <p id="formTitle" class="visually-hidden">Conference registration form collects attendee information and sess
14     </p>
15     <div class="form-group">
16       <label for="name">Full name</label>
17       <input type="text" id="name" name="name" required aria-required="true" aria-describedby="nameHelp">
18       <div id="nameHelp" class="help">Enter your full name.</div>
19       <div class="error" id="nameError" aria-live="polite"></div>
20     </div>
21
22     <div class="form-group">
23       <label for="email">Email address</label>
24       <input type="email" id="email" name="email" required aria-required="true" aria-describedby="emailHelp">
25       <div id="emailHelp" class="help">We'll send confirmation to this email.</div>
26       <div class="error" id="emailError" aria-live="polite"></div>
27     </div>
28
29     <div class="form-group">
30       <label for="phone">Phone number</label>
31       <input type="tel" id="phone" name="phone" inputmode="tel" aria-describedby="phoneHelp">
32       <div id="phoneHelp" class="help">Optional, but helpful for urgent updates.</div>
33       <div class="error" id="phoneError" aria-live="polite"></div>

```

```

34     </div>
35     <div class="form-group">
36       <div>
37         <input type="radio" id="session1" name="session" value="ai" required aria-required="true">
38         <label for="session1">AI in Practice</label>
39       </div>
40       <div>
41         <input type="radio" id="session2" name="session" value="ml">
42         <label for="session2">Machine Learning Foundations</label>
43       </div>
44       <div>
45         <input type="radio" id="session3" name="session" value="ds">
46         <label for="session3">Data Strategies</label>
47       </div>
48       <div class="error" id="sessionError" aria-live="polite"></div>
49     </div>
50   </form>
51   <div class="form-actions">
52     <button type="submit" id="submitBtn">Register</button>
53     <button type="reset">Reset</button>
54   </div>
55   <div id="result" role="status" aria-live="polite"></div>
56 </main>
57 </body>
58 </html>
59
60 <script src="script.js"></script>
61
62
63
64
65

```

```

: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:-**

## Conference Event Registration

Full name

Enter your full name.

Email address

We'll send confirmation to this email.

Phone number

Optional, but helpful for urgent updates.

Session selection

☐

AI in Practice

☐

Machine Learning Foundations

☐

Data Strategies

Register

Reset

### 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
1  <!doctype html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <meta name="viewport" content="width=device-width,initial-scale=1">
6    <title>Fetch JSON Demo</title>
7    <link rel="stylesheet" href="fetch_styles.css">
8  </head>
9  <body>
10   <main class="container">
11     <h1>Posts from JSONPlaceholder</h1>
12     <div id="controls">
13       <button id="loadBtn">Load posts</button>
14       <button id="errorBtn">Simulate error</button>
15     </div>
16     <div id="status" role="status" aria-live="polite"></div>
17     <ul id="list" class="list" aria-live="polite"></ul>
18   </main>
19
20   <script src="fetch_script.js"></script>
21 </body>
22 </html>

```

```

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

```



```

5 fetch_script.js > ...
1 const API_URL = 'https://jsonplaceholder.typicode.com/posts?_limit=8';
2 const loadBtn = document.getElementById('loadBtn');
3 const errorBtn = document.getElementById('errorBtn');
4 const statusEl = document.getElementById('status');
5 const listEl = document.getElementById('list');
6

```

Zencoder

```

7 function setLoading() {
8   statusEl.textContent = 'Loading posts...';
9   listEl.innerHTML = '';
10  for (let i = 0; i < 4; i++) {
11    const li = document.createElement('li');
12    li.className = 'item';
13    const skeletonTitle = document.createElement('div');
14    skeletonTitle.className = 'skeleton';
15    const skeletonBody = document.createElement('div');
16    skeletonBody.className = 'skeleton';
17    li.appendChild(skeletonTitle);
18    li.appendChild(skeletonBody);
19    listEl.appendChild(li);
20  }
21 }
22

```

Zencoder

```

23 function renderPosts(posts) {
24   statusEl.textContent = '';
25   listEl.innerHTML = '';
26   posts.forEach(post => {
27     // Escape text content to avoid XSS
28     const li = document.createElement('li');
29     li.className = 'item';
30     const title = document.createElement('h3');
31     title.textContent = post.title;
32

```

```

33     posts.forEach(post => {
34       title.textContent = post.title;
35       const body = document.createElement('p');
36       body.textContent = post.body;
37       li.appendChild(title);
38       li.appendChild(body);
39       listEl.appendChild(li);
40     });
41   });
42 }
43

```

Zencoder

```

40 function showError(err) {
41   statusEl.innerHTML = `<span class="error">Error: ${String(err)}</span>`;
42   listEl.innerHTML = '';
43 }
44

```

Zencoder

```

45 async function fetchPosts(simulateError = false) {
46   try {
47     setLoading();
48     const url = simulateError ? API_URL + 'INVALID' : API_URL;
49     const res = await fetch(url, {cache: 'no-store'});
50     if (!res.ok) throw new Error('Network response not OK: ' + res.status);
51     const data = await res.json();
52     renderPosts(data);
53   } catch (err) {
54     showError(err.message || err);
55   }
56 }
57
58 loadBtn.addEventListener('click', () => fetchPosts(false));
59 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.