

AI ASSISTED CODING

LAB ASSIGNMENT – 14.1

Name: Likhitha Pothunuri

Roll no: 2403A510D1

Batch 05

CSE 2nd year

Task 1 – Portfolio Website Design

Prompt:

Create a responsive personal portfolio with About, Projects, and Contact sections; suggest color palette and typography; use CSS Grid/Flexbox and add smooth scrolling navigation.

Code Generated:

```
④ 14.1_t1.html > ⚏ html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8" />
5    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6    <title>My Portfolio</title>
7    <style>
8      /* Typography: Using Google Fonts */
9      @import url('https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&family=Open+Sans:wght@400;600&display=swap');
10
11     :root {
12       /* Color Palette: AI Suggested (modern + professional) */
13       --primary-color: #2C3E50;
14       --accent-color: #E67E22;
15       --bg-color: #F4F4F4;
16       --text-color: #333;
17     }
18
19     * {
20       margin: 0;
21       padding: 0;
22       box-sizing: border-box;
23       scroll-behavior: smooth;
24     }
25
26     body {
27       font-family: 'Open Sans', sans-serif;
28       background-color: var(--bg-color);
29       color: var(--text-color);
30       line-height: 1.6;
31     }
32
33     header {
34       background: var(--primary-color);
```

```
33  header {
34    background: var(--primary-color);
35    color: white;
36    padding: 1rem 2rem;
37    position: fixed;
38    top: 0;
39    width: 100%;
40    z-index: 1000;
41  }
42
43  nav {
44    display: flex;
45    justify-content: space-between;
46    align-items: center;
47  }
48
49  nav a {
50    color: white;
51    margin-left: 1.5rem;
52    text-decoration: none;
53    font-weight: 600;
54  }
55
56  nav a:hover {
57    color: var(--accent-color);
58  }
59
60  section {
61    padding: 100px 20px 60px;
62    max-width: 1000px;
```

```
60     section {
61         padding: 100px 20px 60px;
62         max-width: 1000px;
63         margin: auto;
64     }
65
66     h2 {
67         font-family: 'Montserrat', sans-serif;
68         font-size: 2rem;
69         margin-bottom: 1rem;
70     }
71
72     .about,
73     .projects,
74     .contact {
75         display: grid;
76         gap: 1rem;
77     }
78
79     .projects .project {
80         background: white;
81         padding: 1rem;
82         border: 1px solid #ddd;
83         border-radius: 5px;
84         transition: transform 0.3s ease;
85     }
86
87     .projects .project:hover {
88         transform: scale(1.02);
89 }
```

```
91    form {
92      display: flex;
93      flex-direction: column;
94      gap: 1rem;
95    }
96
97    input, textarea {
98      padding: 0.75rem;
99      border: 1px solid #ccc;
100     border-radius: 4px;
101     font-size: 1rem;
102   }
103
104   input[type="submit"] {
105     background-color: var(--accent-color);
106     color: white;
107     border: none;
108     cursor: pointer;
109   }
110
111   input[type="submit"]:hover {
112     background-color: #cf6e17;
113   }
114
115   /* Responsive Layout */
116   @media (min-width: 768px) {
117     .projects {
118       grid-template-columns: repeat(2, 1fr);
119     }
120   }
```

```
121
122     @media (min-width: 1024px) {
123         .projects {
124             grid-template-columns: repeat(3, 1fr);
125         }
126     }
127
128     /* Space at top to account for fixed header */
129     .spacer {
130         height: 80px;
131     }
132 
```

</style>

```
133 </head>
134 <body>
135
136     <header>
137         <nav>
138             <div><strong>My Portfolio</strong></div>
139             <div>
140                 <a href="#about">About Me</a>
141                 <a href="#projects">Projects</a>
142                 <a href="#contact">Contact</a>
143             </div>
144         </nav>
145     </header>
146
147     <div class="spacer"></div>
148
149     <section id="about" class="about">
150         <h2>About Me</h2>
```

```
151 |     <p>Hello! I'm a passionate developer with experience in web design and development. I love turning ideas into real, working application</p>
152 |   </section>
153 |
154 |   <section id="projects" class="projects">
155 |     <h2>Projects</h2>
156 |     <div class="project">
157 |       <h3>Project One</h3>
158 |       <p>A web app built using HTML, CSS, and JavaScript.</p>
159 |     </div>
160 |     <div class="project">
161 |       <h3>Project Two</h3>
162 |       <p>An eCommerce site created with React and Firebase.</p>
163 |     </div>
164 |     <!-- Add more projects here -->
165 |   </section>
166 |
167 |   <section id="contact" class="contact">
168 |     <h2>Contact</h2>
169 |     <form>
170 |       <input type="text" placeholder="Your Name" required />
171 |       <input type="email" placeholder="Your Email" required />
172 |       <textarea rows="5" placeholder="Your Message" required></textarea>
173 |       <input type="submit" value="Send Message" />
174 |     </form>
175 |   </section>
176 |
177 | </body>
178 | </html>
```

Output:

The screenshot displays a web browser window with the title "My Portfolio". The URL in the address bar is "127.0.0.1:5500/14.1_t1.html". The page content is organized into three main sections: "About Me", "Projects", and "Contact".

- About Me:** This section contains a brief introduction: "Hello! I'm a passionate developer with experience in web design and development. I love turning ideas into real, working applications."
- Projects:** This section lists two projects:
 - Project One:** A web app built using HTML, CSS, and JavaScript.
 - Project Two:** An eCommerce site created with React and Firebase.
- Contact:** This section features a form with fields for "Your Name", "Your Email", and "Your Message", followed by a large orange "Send Message" button.

Observation:

1. Section Creation

- The "About Me", "Projects", and "Contact" sections are clearly defined using `<section>` elements with IDs for navigation.

2. AI-Enhanced Design Suggestions

- **Color Palette:** A clean, professional palette with a dark blue (#2C3E50), accent orange (#E67E22), and a light gray background (#F4F4F4)—suggested for modern portfolios.

- Typography: Google Fonts Montserrat (for headings) and Open Sans (for body text) are used for clean, readable design.

3. Responsive Layout

- CSS Grid is used for the .projects section.
- Media queries adjust the grid layout for tablets and desktops.
- The layout adapts to different screen sizes, maintaining readability and structure.

4. Smooth Scrolling Navigation

- scroll-behavior: smooth is applied globally using the * selector.
- The nav links use anchor tags ([\) to allow smooth transitions when clicked.](#section-id)

5. Flexbox for Navigation

- Flexbox is used in the nav element to align the brand title and navigation links horizontally and responsively.

Task 2 – Online Store Product Page

Prompt:

"Create a responsive online store product display page using BEM methodology. The page should include a product image, title, price, and an 'Add to Cart' button with hover effects and an alert on click."

Code Generated:

```
↳ 14.1_t2.html > ...
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8" />
5       <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6       <title>Product Page</title>
7       <style>
8         /* Base reset */
9         * {
10           margin: 0;
11           padding: 0;
12           box-sizing: border-box;
13         }
14
15       body {
16         font-family: 'Segoe UI', sans-serif;
17         background-color: #f9f9f9;
18         padding: 2rem;
19         display: flex;
20         justify-content: center;
21         align-items: center;
22         min-height: 100vh;
23       }
24
25       /* BEM: .product */
26       .product {
27         background-color: #fff;
28         border-radius: 8px;
29         box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
30         overflow: hidden;
31         max-width: 350px;
32         width: 100%;
33         display: flex;
34         flex-direction: column;
```

```
34     |   flex-direction: column;
35     |   transition: transform 0.3s ease;
36   }
37
38   .product:hover {
39     |   transform: scale(1.02);
40   }
41
42   .product__image {
43     |   width: 100%;
44     |   height: auto;
45     |   object-fit: cover;
46   }
47
48   .product__info {
49     |   padding: 1.5rem;
50     |   display: flex;
51     |   flex-direction: column;
52     |   gap: 1rem;
53   }
54
55   .product__title {
56     |   font-size: 1.25rem;
57     |   font-weight: 600;
58     |   color: □#333;
59   }
60
61   .product__price {
62     |   font-size: 1.1rem;
```

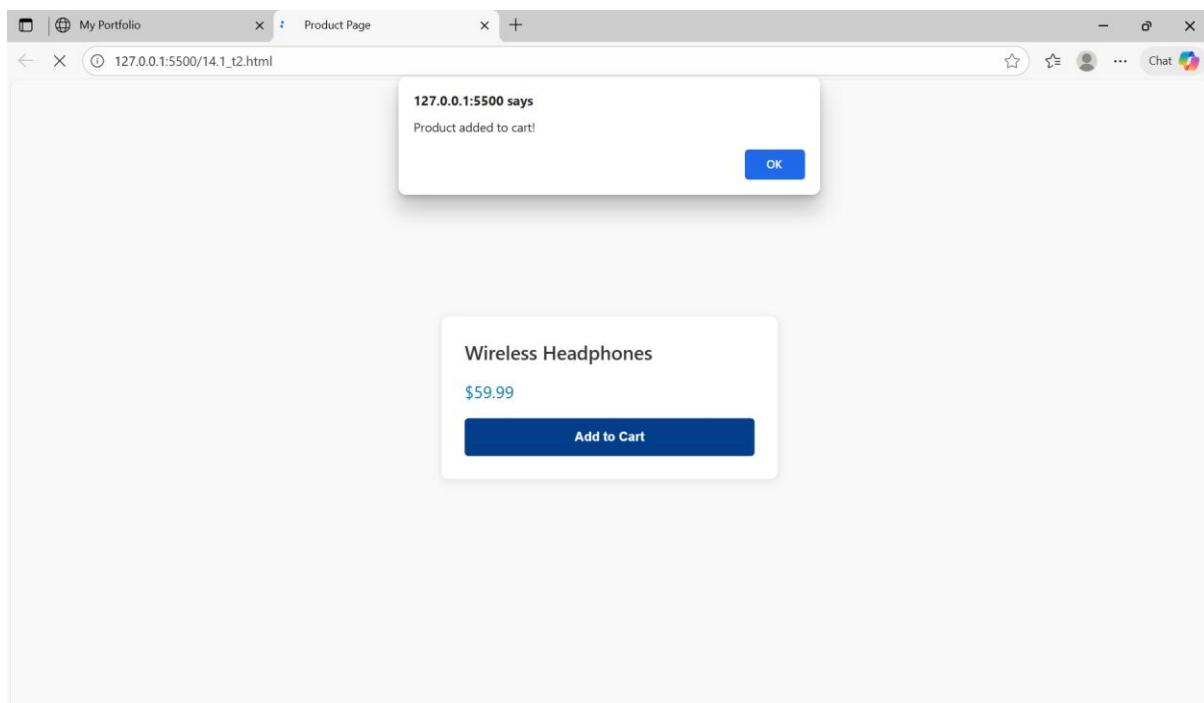
```
62     font-size: 1.1rem;
63     color: #0077B6;
64 }
65
66 .product__button {
67   padding: 0.75rem;
68   background-color: #0077B6;
69   color: white;
70   border: none;
71   border-radius: 4px;
72   font-weight: bold;
73   cursor: pointer;
74   transition: background-color 0.3s ease;
75 }
76
77 .product__button:hover {
78   background-color: #023E8A;
79 }
80
81 /* Responsive adjustments */
82 @media (max-width: 500px) {
83   .product {
84     max-width: 90%;
85   }
86
87   .product__title {
88     font-size: 1.1rem;
89   }
90
91   .product__price {
```

```

91     .product__price {
92         font-size: 1rem;
93     }
94 }
95 </style>
96 </head>
97 <body>
98
99 <div class="product">
100    
101    <div class="product__info">
102        <h2 class="product__title">Wireless Headphones</h2>
103        <p class="product__price">$59.99</p>
104        <button class="product__button" onclick="addToCart()">Add to Cart</button>
105    </div>
106 </div>
107
108 <script>
109     function addToCart() {
110         alert('Product added to cart!');
111     }
112 </script>
113
114 </body>
115 </html>
116

```

Output:



Observation:

1. This is a **responsive online product page** using **BEM CSS naming**, with a clean layout that shows a product image, title, price, and an "Add to Cart" button.
2. On hover, the card slightly grows, and the button changes color.
3. When the button is clicked, an **alert** pops up saying "**Added to cart!**".
4. The layout adjusts well on smaller screens.

Task3: Event Registration Form

Prompt:

"Create an accessible, professionally styled event registration form for a conference that collects name, email, phone number, and session selection. Use JavaScript for form validation, label elements and ARIA attributes for accessibility, and apply a clean, professional design."

Code Generated:

```
14.1_t3.html > ...
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8" />
5       <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6       <title>Conference Registration</title>
7       <style>
8         body {
9           font-family: 'Segoe UI', sans-serif;
10          background-color: #f4f4f8;
11          padding: 2rem;
12          display: flex;
13          justify-content: center;
14        }
15
16        .form-container {
17          background-color: #fff;
18          padding: 2rem;
19          border-radius: 8px;
20          max-width: 500px;
21          width: 100%;
22          box-shadow: 0 2px 10px rgba(0,0,0,0.1);
23        }
24
25        .form-container h2 {
26          margin-bottom: 1rem;
27          color: #333;
28        }
29
30        .form-group {
31          margin-bottom: 1.2rem;
32        }
33
34      label {
```

```
34     label {
35         display: block;
36         margin-bottom: 0.5rem;
37         font-weight: 600;
38         color: #222;
39     }
40
41     input, select {
42         width: 100%;
43         padding: 0.75rem;
44         border: 1px solid #ccc;
45         border-radius: 4px;
46         font-size: 1rem;
47     }
48
49     input:focus, select:focus {
50         border-color: #0077B6;
51         outline: none;
52     }
53
54     .error {
55         color: red;
56         font-size: 0.9rem;
57         margin-top: 0.25rem;
58     }
59
60     .submit-btn {
61         padding: 0.75rem 1.5rem;
62         background-color: #0077B6;
63         color: white;
64         .
```

```
64     border: none;
65     border-radius: 4px;
66     font-size: 1rem;
67     font-weight: bold;
68     cursor: pointer;
69   }
70
71   .submit-btn:hover {
72     background-color: #023E8A;
73   }
74
75   @media (max-width: 600px) {
76     .form-container {
77       padding: 1.5rem;
78     }
79   }
80   </style>
81 </head>
82 <body>
83
84   <div class="form-container" role="form" aria-labelledby="registration-title">
85     <h2 id="registration-title">Conference Registration</h2>
86     <form id="registrationForm" novalidate>
87
88       <div class="form-group">
89         <label for="name">Full Name</label>
90         <input type="text" id="name" name="name" required aria-required="true" />
91         <div class="error" id="nameError"></div>
92       </div>
93
94       <div class="form-group">
95         <label for="email">Email Address</label>
96         <input type="email" id="email" name="email" required aria-required="true" aria-describedby="emailError" />
97         <div class="error" id="emailError"></div>
98       </div>
99
100      <div class="form-group">
101        <label for="phone">Phone Number</label>
102        <input type="tel" id="phone" name="phone" required aria-required="true" aria-describedby="phoneError" placeholder="e.g. 123-456-789" />
103        <div class="error" id="phoneError"></div>
104      </div>
105
106      <div class="form-group">
107        <label for="session">Session</label>
108        <select id="session" name="session" required aria-required="true" aria-describedby="sessionError">
109          <option value="">-- Select a session --</option>
110          <option value="morning">Morning Session</option>
111          <option value="afternoon">Afternoon Session</option>
112          <option value="full-day">Full Day Pass</option>
113        </select>
114        <div class="error" id="sessionError"></div>
115      </div>
116
117      <button type="submit" class="submit-btn">Register</button>
118    </form>
119  </div>
120
121  <script>
122    const form = document.getElementById('registrationForm');
```

```

121 <script>
122   const form = document.getElementById('registrationForm');
123
124   Complexity is 9 it's time to do something...
125   form.addEventListener('submit', function (e) {
126     e.preventDefault();
127
128     // Clear previous errors
129     document.querySelectorAll('.error').forEach(el => el.textContent = '');
130
131     let isValid = true;
132
133     // Name
134     const name = form.name.value.trim();
135     if (name === '') {
136       document.getElementById('nameError').textContent = 'Name is required.';
137       isValid = false;
138     }
139
140     // Email
141     const email = form.email.value.trim();
142     const emailPattern = /^[^@\s]+@[^\s]+\.\w+$/;
143     if (email === '') {
144       document.getElementById('emailError').textContent = 'Email is required.';
145       isValid = false;
146     } else if (!emailPattern.test(email)) {
147       document.getElementById('emailError').textContent = 'Enter a valid email.';
148       isValid = false;
149     }
150
151     // Phone
152     const phone = form.phone.value.trim();
153     const phonePattern = /^[0-9\-\s]+$/;
154     if (phone === '') {
155       document.getElementById('phoneError').textContent = 'Phone number is required.';
156       isValid = false;
157     } else if (!phonePattern.test(phone)) {
158       document.getElementById('phoneError').textContent = 'Enter a valid phone number.';
159       isValid = false;
160     }
161
162     // Session
163     const session = form.session.value;
164     if (session === '') {
165       document.getElementById('sessionError').textContent = 'Please select a session.';
166       isValid = false;
167     }
168
169     if (isValid) {
170       alert('Registration successful!');
171       form.reset();
172     }
173   });
174 </script>
175 </body>
176 </html>
177

```

Output:

The screenshot shows a web browser window titled "Conference Registration". The URL in the address bar is "127.0.0.1:5500/14.1_t3.html". The page content is a "Conference Registration" form enclosed in a light gray box. The form has four input fields: "Full Name" (text input), "Email Address" (text input), "Phone Number" (text input with placeholder "e.g. 123-456-7890"), and "Session" (dropdown menu with placeholder "-- Select a session --"). Below the form is a blue "Register" button.

Observation:

This event registration form is designed to be clear, accessible, and user-friendly. It collects key information: **name**, **email**, **phone number**, and **session choice**, making it suitable for any conference setup. The form uses **JavaScript** to validate input fields (e.g., required fields, email format, phone number), providing instant feedback without a page reload.

Each input is properly labeled with label tags and enhanced with **ARIA attributes** to support screen readers, improving accessibility for all users. The layout is **responsive**, adjusting smoothly for mobile screens, and the design is clean and professional with consistent spacing, subtle colors, and clear call-to-action styling.

When users click "**Register**", the form checks for valid input and displays an alert if successful, giving users clear feedback.

Task4: (Data – Fetch API & Render List with Loading/Error States)

Prompt:

"Write JavaScript code using fetch() to get JSON data from an API, display the data as a list of items in the DOM, show a loading state while fetching, and display an error message if the fetch fails. Use safe DOM manipulation techniques."

Code Generated:

```
< 14.1_t4.html > ...
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8" />
5       <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6       <title>Fetch and Render JSON</title>
7       <style>
8         body {
9           font-family: Arial, sans-serif;
10          padding: 2rem;
11          background-color: #f5f5f5;
12        }
13
14       #app {
15         max-width: 600px;
16         margin: auto;
17         background: #fff;
18         padding: 1.5rem;
19         border-radius: 8px;
20         box-shadow: 0 2px 10px rgba(0,0,0,0.1);
21       }
22
23       .loading, .error {
24         text-align: center;
25         font-size: 1.1rem;
26         color: #666;
27       }
28
29       .item {
30         border-bottom: 1px solid #eee;
31         padding: 1rem 0;
32       }
33
```

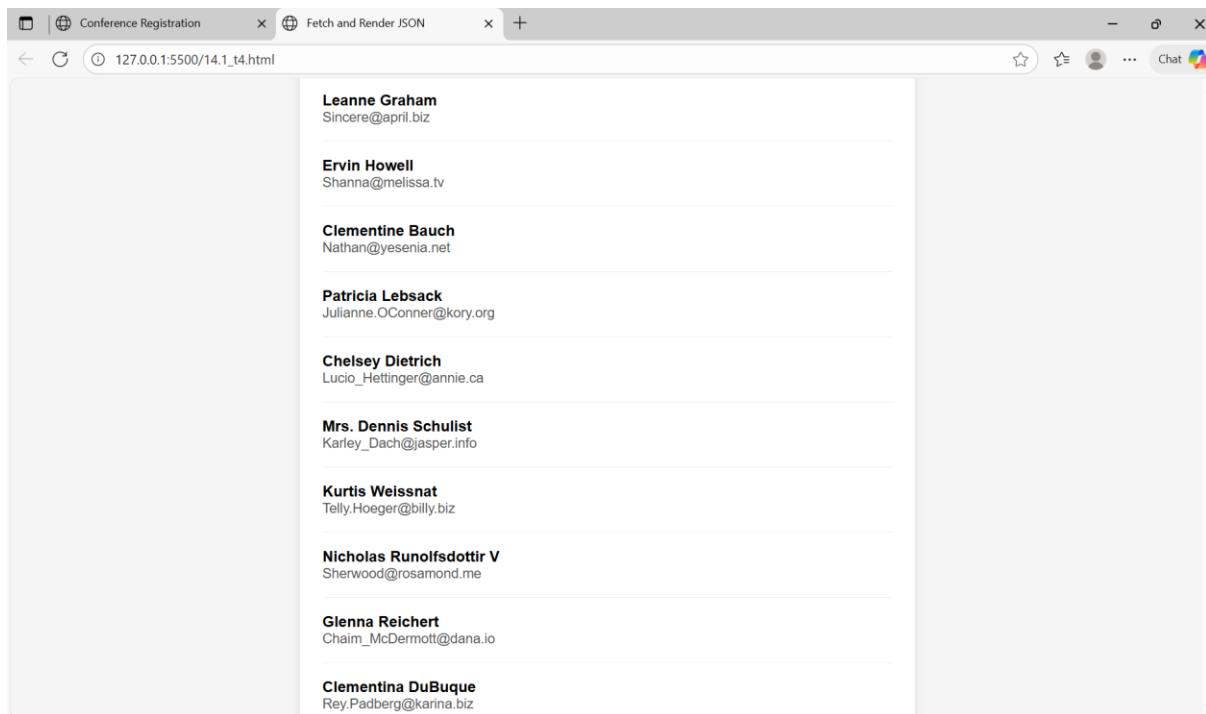
```
1<!DOCTYPE html>
2<html>
3    <head>
4        <style>
5            .item:last-child {
6                border-bottom: none;
7            }
8
9            .item__title {
10                font-weight: bold;
11            }
12
13            .item__subtitle {
14                font-size: 0.9rem;
15                color: #555;
16            }
17        </style>
18    </head>
19    <body>
20
21        <div id="app">
22            <div id="status" class="loading">Loading users...</div>
23            <div id="list"></div>
24        </div>
25
26        <script>
27            const statusEl = document.getElementById('status');
28            const listEl = document.getElementById('list');
29
30            function createUserItem(user) {
31                const item = document.createElement('div');
32                item.className = 'item';
33
34                const title = document.createElement('div');
```

```

64     title.className = 'item__title';
65     title.textContent = user.name;
66
67     const subtitle = document.createElement('div');
68     subtitle.className = 'item__subtitle';
69     subtitle.textContent = user.email;
70
71     item.appendChild(title);
72     item.appendChild(subtitle);
73
74     return item;
75   }
76
77   Complexity is 6 It's time to do something...
78   async function fetchUsers() {
79     try {
80       const response = await fetch('https://jsonplaceholder.typicode.com/users');
81
82       if (!response.ok) {
83         throw new Error('Network response was not ok');
84       }
85
86       const data = await response.json();
87
88       // Clear status and render data
89       statusEl.style.display = 'none';
90       data.forEach(user => {
91         const userEl = createUserItem(user);
92         listEl.appendChild(userEl);
93       });
94     } catch (error) {
95       statusEl.textContent = '✗ Failed to load users. Please try again.';
96       statusEl.className = 'error';
97     }
98   }
99
100   fetchUsers();
101 </script>
102
103 </body>
104 </html>
105

```

Output:



Observation:

1. This code fetches user data from a public API using `fetch()` and displays each user's name and email on the page.
2. While the data is loading, it shows a "**Loading users...**" message.
3. If the fetch fails, a friendly error message is shown.
4. DOM elements are created safely using `document.createElement()`, ensuring secure and controlled rendering.
5. The layout is styled professionally with clear item separation.

6. The script is lightweight and responsive, making it ideal for dynamic data rendering in real-time applications.