

AI ASSISTED CODING

ASSIGNMENT-14

Name: G Hasini

Roll no.: 2503A51L41

Batch: 24BTCAICSB20

Task #1:

AI-Assisted Portfolio Website

Scenario:

A student wants to showcase their projects, skills, and contact details in a portfolio website. Instead of writing all code manually, they want to speed up the process using GitHub Copilot.

- Use Copilot to generate an HTML structure for a personal portfolio page (sections: Home, About, Projects, Contact).
- Ask Copilot to suggest responsive CSS styling for the layout (e.g., grid/flexbox).
- Customize Copilot's suggestions to add a hover effect on project cards

Prompt:

generate an HTML structure for a personal portfolio page (sections: Home, About, Projects, Contact).suggest responsive CSS styling for the layout (e.g., grid/flexbox).

add a hover effect on project cards give html css in same file.

Code generated:

```

ai_lab13 > ai_lab_14 > <> port.html > <> html > <> head > <> style > <> :root
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>Student Portfolio</title>
7   <style>
8     :root {
9       --primary: #7aa9e9;
10      --bg: #f4f4f4;
11      --text: #222;
12      --card-bg: #fff;
13      --hover: #264653;
14      --transition: 0.3s;
15    }
16    * {
17      margin: 0;
18      padding: 0;
19      box-sizing: border-box;
20      font-family: 'Segoe UI', Arial, sans-serif;
21    }
22    body {
23      background: var(--bg);
24      color: var(--text);
25      line-height: 1.6;
26    }
27    header {
28      background: var(--primary);
29      color: #fff;
30      padding: 2rem 0 1rem 0;
31      text-align: center;
32      position: sticky;
33      top: 0;
34      z-index: 100;
35    }
36    nav ul {
37      list-style: none;
38      display: flex;
39      justify-content: center;
40      gap: 2rem;
41      padding: 0;
42      margin-top: 1rem;

```

```

43    }
44    nav a {
45      color: #fff;
46      text-decoration: none;
47      font-weight: bold;
48      letter-spacing: 1px;
49      transition: color var(--transition);
50    }
51    nav a:hover {
52      color: #e9c46a;
53    }
54    main {
55      width: 90%;
56      max-width: 1200px;
57      margin: 2rem auto;
58      display: flex;
59      flex-direction: column;
60      gap: 3rem;
61    }
62    section {
63      background: var(--card-bg);
64      padding: 2rem;
65      border-radius: 12px;
66      box-shadow: 0 2px 10px rgba(0,0,0,0.05);
67    }
68    #projects .project-list {
69      display: grid;
70      grid-template-columns: repeat(auto-fit, minmax(260px, 1fr));
71      gap: 2rem;
72    }
73    .project-card {
74      background: #f9fafb;
75      border-radius: 10px;
76      box-shadow: 0 1px 4px rgba(32,40,70,0.08);
77      transition: transform var(--transition), box-shadow var(--transition), background var(--transition);
78      padding: 1.5rem;
79      cursor: pointer;
80      position: relative;

```

```

81     overflow: hidden;
82   }
83   .project-card:hover {
84     transform: translateY(-8px) scale(1.025);
85     box-shadow: 0 4px 18px rgba(42,157,143,0.15);
86     background: var(--hover);
87     color: #fff;
88   }
89   .project-card h3 {
90     margin-bottom: 0.6rem;
91   }
92   .project-card:hover h3,
93   .project-card:hover p {
94     color: #e9c46a;
95   }
96   @media (max-width: 700px) {
97     nav ul {
98       flex-direction: column;
99       gap: 0.8rem;
100     }
101     main {
102       width: 98%;
103       gap: 2rem;
104     }
105     section {
106       padding: 1rem;
107     }
108     #projects .project-list {
109       grid-template-columns: 1fr;
110     }
111   }
112 </style>
113 </head>
114 <body>
115   <header>
116     <h1>Gundeti Hasini</h1>
117     <p>passionate Learner & aspiring web developer</p>
118     <nav>

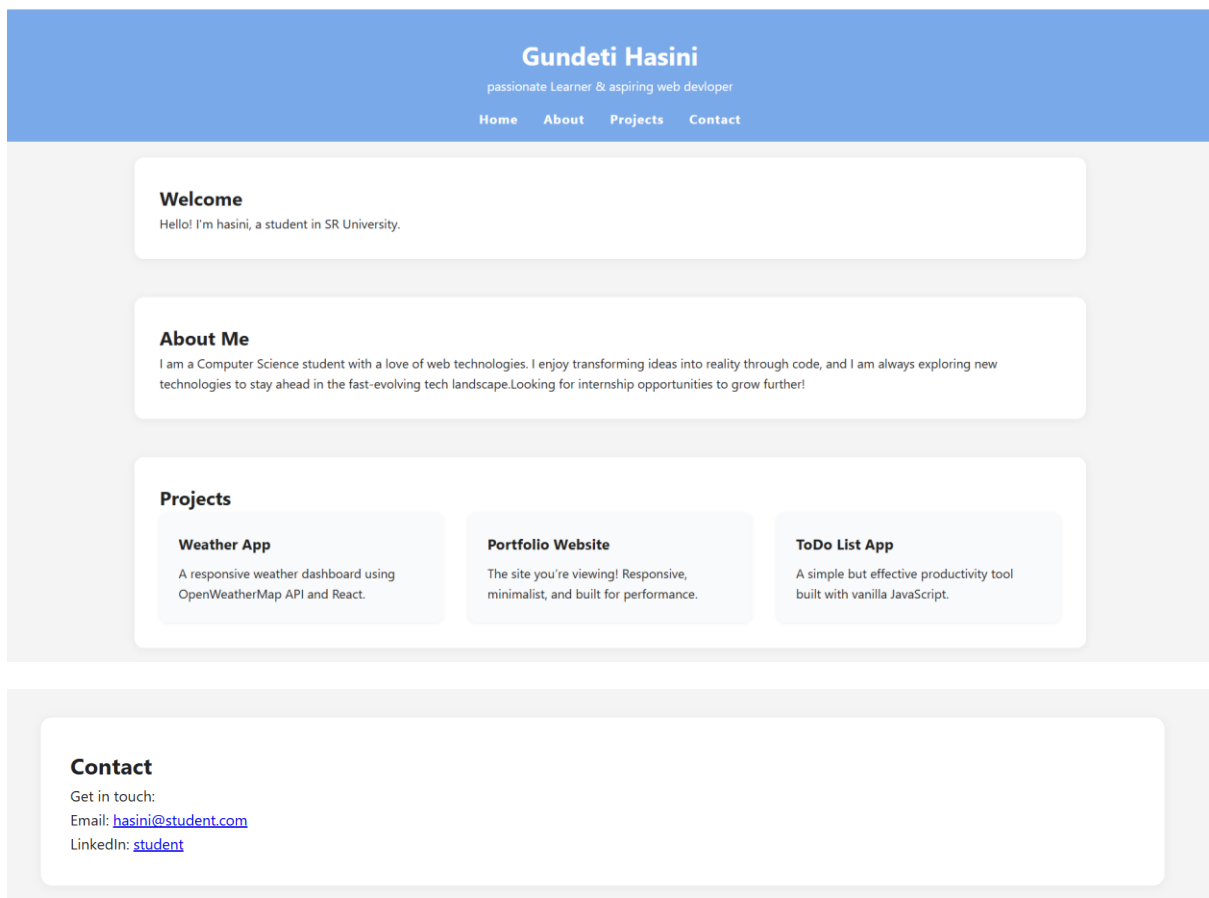
```

```

118     <nav>
119       <ul>
120         <li><a href="#home">Home</a></li>
121         <li><a href="#about">About</a></li>
122         <li><a href="#projects">Projects</a></li>
123         <li><a href="#contact">Contact</a></li>
124       </ul>
125     </nav></header>
126     <main>
127       <section id="home">
128         <h2>Welcome</h2>
129         <p>Hello! I'm hasini, a student in SR University.</p>
130       </section>
131       <section id="about">
132         <h2>About Me</h2>
133         <p>I am a Computer Science student with a love of web technologies. I enjoy transforming ideas into reality through coding.</p>
134       </section>
135       <section id="projects">
136         <h2>Projects</h2>
137         <div class="project-list">
138           <div class="project-card">
139             <h3>Weather App</h3>
140             <p>A responsive weather dashboard using OpenWeatherMap API and React.</p></div>
141           <div class="project-card">
142             <h3>Portfolio Website</h3>
143             <p>The site you're viewing! Responsive, minimalist, and built for performance.</p></div>
144           <div class="project-card">
145             <h3>ToDo List App</h3>
146             <p>A simple but effective productivity tool built with vanilla JavaScript.</p>
147           </div></div>
148       </section>
149       <section id="contact">
150         <h2>Contact</h2>
151         <p>Get in touch: <br>
152           Email: <a href="mailto:hasini@student.com">hasini@student.com</a><br>
153           LinkedIn: <a href="https://linkedin.com/in/hstudent" target="_blank">student</a></p>
154       </section></main>
155     </body></html>

```

Output:



Observation:

The code builds a responsive personal portfolio webpage with four main sections: Home, About, Projects, and Contact. It uses flexbox and CSS grid for layout, ensuring adaptability across screen sizes. The navigation is sticky, the project cards include a hover effect (lift + shadow), and a simple contact form is provided, making the design clean, modern, and user-friendly.

Task #2:

AI-Generated Restaurant Landing Page

Scenario: A local restaurant needs simple landing page with navigation bar, menu highlights, and an image gallery. The developer wants to quickly generate it using AI assistance.

Prompt: a local restaurant needs simple landing page with navigation bar, menu highlights, and an image gallery.

Code generated:

```
ai_lab13 > ai_lab_14 > restr.html > html > head > style > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>Restaurant Landing Page</title>
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  <style>
8    :root {
9      --primary: #9e568a;
10     --accent: #f3f5f7;
11     --bg: #f2eef2f7;
12     --text: #222;
13   }
14   body {
15     margin: 0;
16     background: var(--bg);
17     color: var(--text);
18     font-family: 'Segoe UI', Arial, sans-serif;
19   }
20   .navbar {
21     background: var(--primary);
22     display: flex;
23     justify-content: center;
24     align-items: center;
25     padding: 1rem 0;
26     position: sticky;
27     top: 0;
28     z-index: 100;
29   }
30   .navbar ul {
31     list-style: none;
32     display: flex;
33     gap: 2rem;
34     margin: 0;
35     padding: 0;
36   }
37   .navbar a {
38     color: #fff;
39     text-decoration: none;
40     font-size: 1.1rem;
41     font-weight: bold;
42     letter-spacing: 1px;
```

```

42     letter-spacing: 1px;
43     transition: color 0.2s;
44   }
45   .navbar a:hover {
46     color: var(--accent);
47   }
48   .container {
49     max-width: 1100px;
50     margin: 0 auto;
51     padding: 2rem 1rem;
52   }
53   .title {
54     text-align: center;
55     font-size: 2.5rem;
56     color: var(--primary);
57     margin: 2rem 0 1rem 0;
58   }
59   /* Menu Section */
60   #menu {
61     margin: 3rem 0;
62   }
63   .menu-cards {
64     display: grid;
65     grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
66     gap: 2rem;
67   }
68   .menu-card {
69     background: #fff;
70     border-radius: 10px;
71     box-shadow: 0 2px 12px rgba(178,58,72,0.08);
72     padding: 1.3rem;
73     text-align: center;
74     transition: box-shadow 0.2s, transform 0.2s;
75     border: 2px solid var(--accent);
76   }
77   .menu-card:hover {
78     box-shadow: 0 6px 20px rgba(178,58,72,0.16);
79     transform: scale(1.035);
```

```

118     color: #fff;
119     border: none;
120     padding: 0.7rem 1rem;
121     border-radius: 50%;
122     cursor: pointer;
123     font-size: 1.5rem;
124     opacity: 0.7;
125     transition: opacity 0.2s;
126     z-index: 2;
127 }
128 .slider-btn:hover { opacity: 1; }
129 .slider-btn.left { left: 10px; }
130 .slider-btn.right { right: 10px; }
131 @media (max-width: 700px) {
132     .menu-cards { grid-template-columns: 1fr; }
133     .slider-container { max-width: 99vw; }
134     .slider-images img { height: 100px; }
135 }
136 </style>
137 </head>
138 <body>
139     <nav class="navbar">
140         <ul>
141             <li><a href="#home">Home</a></li>
142             <li><a href="#menu">Menu</a></li>
143             <li><a href="#gallery">Gallery</a></li>
144             <li><a href="#contact">Contact</a></li>
145         </ul>
146     </nav>
147
148     <div class="container">
149         <section id="home">
150             <h1 class="title">Welcome to Dreamy Dine</h1>
151             <p style="text-align:center; max-width:600px; margin: 0 auto;">Experience delightful cuisine at our charming local restaurant</p>
152         </section>
153         <section id="menu">
154             <h2 class="title">Menu Highlights</h2>
155             <div class="menu-cards">

```

```

156         <div class="menu-card">
157             <h3>Grilled Italian Panini</h3>
158             <p>Fresh mozzarella, pesto, tomato on toasted bread</p>
159             <p><strong>$25</strong></p>
160         </div>
161         <div class="menu-card">
162             <h3>Sizzling Veg Sizzler</h3>
163             <p>Seasonal veggies served hot on a sizzler plate</p>
164             <p><strong>$32</strong></p>
165         </div>
166         <div class="menu-card">
167             <h3>Creamy Mushroom Soup</h3>
168             <p>Smooth, rich, and perfect for all seasons</p>
169             <p><strong>$18</strong></p>
170         </div>
171         <div class="menu-card">
172             <h3>Classic Chocolate Brownie</h3>
173             <p>Warm, gooey brownie topped with vanilla ice cream</p>
174             <p><strong>$15</strong></p>
175         </div>
176     </div>
177 </section>
178 <section id="gallery">
179     <h2 class="title">Gallery</h2>
180     <div class="slider-container">
181         <button class="slider-btn left">&#8249;</button>
182         <div class="slider-images">
183             
184             
185             
186             
187         </div>
188         <button class="slider-btn right">&#8250;</button>
189     </div>
190 </section>
191 <section id="contact">

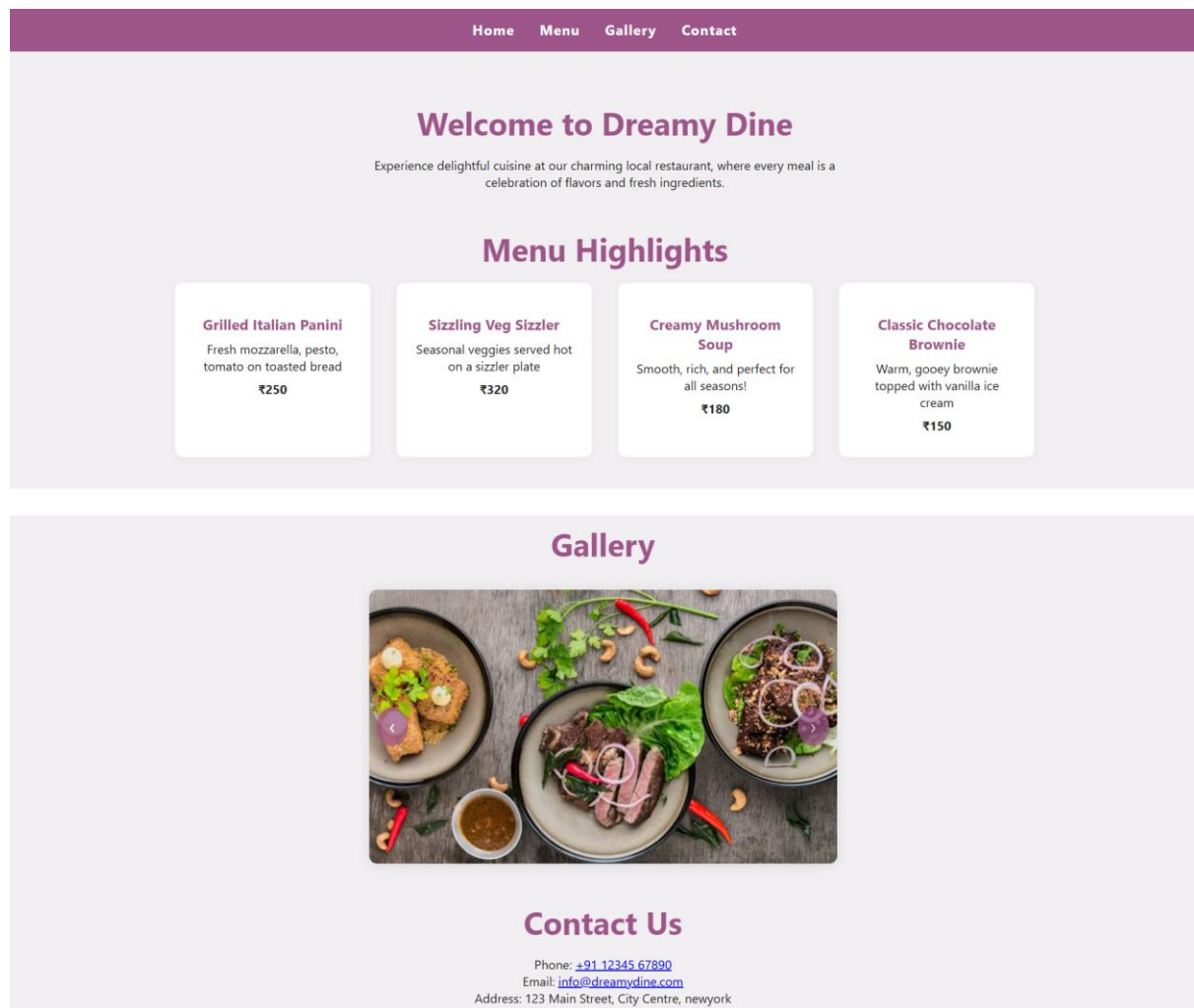
```

```

191     <section id="contact">
192         <h2 class="title">Contact Us</h2>
193         <p style="text-align:center;">
194             Phone: <a href="tel:+911234567890">+91 12345 67890</a><br>
195             Email: <a href="mailto:info@dreamydine.com">info@dreamydine.com</a><br>
196             Address: 123 Main Street, City Centre, New York
197         </p></section></div>
198 <script>
199     const images = document.querySelectorAll('.slider-images img');
200     const leftBtn = document.querySelector('.slider-btn.left');
201     const rightBtn = document.querySelector('.slider-btn.right');
202     let currentIndex = 0;
203     function showImage(idx) {
204         images.forEach((img, i) => {
205             img.classList.toggle('active', i === idx);
206         });
207     }
208     leftBtn.addEventListener('click', () => {
209         currentIndex = (currentIndex - 1 + images.length) % images.length;
210         showImage(currentIndex);
211     });
212     rightBtn.addEventListener('click', () => {
213         currentIndex = (currentIndex + 1) % images.length;
214         showImage(currentIndex);
215     });
216     let startX = null;
217     document.querySelector('.slider-container').addEventListener('touchstart', e => {
218         startX = e.touches[0].clientX;
219     });
220     document.querySelector('.slider-container').addEventListener('touchend', e => {
221         if (startX === null) return;
222         let endX = e.changedTouches[0].clientX;
223         if (endX - startX > 40) leftBtn.click();
224         else if (startX - endX > 40) rightBtn.click();
225         startX = null;
226     });
227 </script></body></html>

```

Output:



Observation:

The code designs a simple and responsive restaurant landing page with a sticky navigation bar, a menu section styled using CSS cards, and an interactive JavaScript-based image slider for the gallery. The layout is clean, user-friendly, and effectively highlights the restaurant's offerings while ensuring easy navigation.

Task #3:

AI-Powered Event Registration Form

Scenario: SR University is hosting a tech fest. They need a web-based registration

form for students. The form must validate user input in real-time.

Prompt: SR University is hosting a tech fest. They need a web-based

registration form for students. The form must validate user input in real-time. generate an HTML form (fields: Name, Email, Phone, Department, Event Selection). Assist in adding CSS styling for an attractive form layout.

- Implement JavaScript validation (e.g., email format check, phone number length check) using Copilot's suggestions. Give js html css in single file.

Code generated:

```
ai_lab13 > ai_lab14 > tech.html > html > head > style > root
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>SR University Tech Fest Registration</title>
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <style>
8 :root {
9   --primary: #441bb3;
10  --accent: #34be82;
11  --error: #d7263d;
12  --bg: #f7fafc;
13  --card-bg: #fff;
14  --text: #222;
15 }
16 body {
17   background: var(--bg);
18   font-family: 'Segoe UI', Arial, sans-serif;
19   min-height: 100vh;
20   display: flex;
21   justify-content: center;
22   align-items: center;
23 }
24 .form-container {
25   background: var(--card-bg);
26   padding: 2rem 2.5rem;
27   border-radius: 16px;
28   box-shadow: 0 4px 18px 0 rgba(41,55,240,0.13);
29   width: 100%;
30   max-width: 410px;
31 }
32 .form-container h2 {
33   text-align: center;
34   color: var(--primary);
35   margin-bottom: 1rem;
36 }
37 .event-form label {
38   display: block;
39   margin-top: 1.1rem;
40   font-weight: 500;
41 }
42 .event-form input,
```

```
42 .event-form input,
43 .event-form select {
44   width: 100%;
45   margin-top: 0.4rem;
46   padding: 0.6rem 0.9rem;
47   border: 2px solid #e7eaf3;
48   border-radius: 8px;
49   font-size: 1rem;
50   transition: border-color 0.25s;
51   outline: none;
52 }
53 .event-form input:focus,
54 .event-form select:focus {
55   border-color: var(--primary);
56 }
57 .error-msg {
58   color: var(--error);
59   font-size: 0.95em;
60   margin-top: 0.25rem;
61   display: none;
62 }
63 .event-form button {
64   width: 100%;
65   padding: 0.9rem;
66   background: var(--primary);
67   color: #fff;
68   border: none;
69   border-radius: 8px;
70   font-size: 1.07rem;
71   margin-top: 1.4rem;
72   cursor: pointer;
73   font-weight: 600;
74   transition: background 0.18s;
75 }
76 .event-form button:disabled {
77   background: #abb4ef;
78   cursor: default;
79 }
80 .success-msg {
```

```

81     color: var(--accent);
82     font-weight: bold;
83     text-align: center;
84     margin-top: 1rem;
85     display: none;
86 }
87 </style></head>
88 <body>
89   <div class="form-container">
90     <h2>SRU Tech Fest Registration</h2>
91     <form class="event-form" id="regForm" autocomplete="off">
92       <label for="name">Name:</label>
93       <input type="text" id="name" required>
94       <div class="error-msg" id="nameError">Name is required.</div>
95       <label for="email">Email:</label>
96       <input type="email" id="email" required>
97       <div class="error-msg" id="emailError">Please enter a valid email.</div>
98       <label for="phone">Phone:</label>
99       <input type="tel" id="phone" maxlength="10" required>
100      <div class="error-msg" id="phoneError">Enter a valid 10-digit phone number.</div>
101      <label for="department">Department:</label>
102      <select id="department" required>
103        <option value="">Select</option>
104        <option>Computer Science</option>
105        <option>Electronics</option>
106        <option>Mechanical</option>
107        <option>Civil</option>
108        <option>Management</option></select>
109      <div class="error-msg" id="deptError">Please select your department.</div>
110      <label for="event">Event Selection:</label>
111      <select id="event" required>
112        <option value="">Select</option>
113        <option>Hackathon</option>
114        <option>Robotics</option>
115        <option>Web Design</option>
116        <option>Quiz</option>
117        <option>AI Workshop</option></select>
118      <div class="error-msg" id="eventError">Please select an event.</div>

```

```

119     <button type="submit" id="submitBtn" disabled>Register</button>
120     <div class="success-msg" id="successMsg">Registration successful! See you at the fest!</div>
121   </form> </div>
122 <script>
123   const name = document.getElementById('name');
124   const email = document.getElementById('email');
125   const phone = document.getElementById('phone');
126   const department = document.getElementById('department');
127   const eventField = document.getElementById('event');
128   const submitBtn = document.getElementById('submitBtn');
129   const nameError = document.getElementById('nameError');
130   const emailError = document.getElementById('emailError');
131   const phoneError = document.getElementById('phoneError');
132   const deptError = document.getElementById('deptError');
133   const eventError = document.getElementById('eventError');
134   const successMsg = document.getElementById('successMsg');
135   const regForm = document.getElementById('regForm');
136   function validateName() {
137     if (!name.value.trim()) {
138       nameError.style.display = 'block';
139       return false;
140     }
141     nameError.style.display = 'none';
142     return true;
143   }
144   function validateEmail() {
145     const regex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
146     if (!regex.test(email.value.trim())) {
147       emailError.style.display = 'block';
148       return false;
149     }
150     emailError.style.display = 'none';
151     return true;
152   }
153   function validatePhone() {
154     const val = phone.value.trim();
155     if (!/^d{10}$/.test(val)) {
156       phoneError.stvle.displav = 'block';

```

```

157     return true;
158     function validateDept() {
159         if (!department.value) {
160             deptError.style.display = 'block';
161             return false;
162             deptError.style.display = 'none';
163             return true;
164         }
165         function validateEvent() {
166             if (!eventField.value) {
167                 eventError.style.display = 'block';
168                 return false;
169             }
170             eventError.style.display = 'none';
171             return true;
172         }
173         function validateForm() {
174             return validateName() & validateEmail() & validatePhone() & validateDept() & validateEvent();
175         }
176         name.addEventListener('input', () => { validateName(); enableSubmit(); });
177         email.addEventListener('input', () => { validateEmail(); enableSubmit(); });
178         phone.addEventListener('input', () => { validatePhone(); enableSubmit(); });
179         department.addEventListener('change', () => { validateDept(); enableSubmit(); });
180         eventField.addEventListener('change', () => { validateEvent(); enableSubmit(); });
181         function enableSubmit() {
182             submitBtn.disabled = !validateForm();
183         }
184         regForm.addEventListener('submit', function(e) {
185             e.preventDefault();
186             if (!validateForm()) {
187                 enableSubmit();
188                 return;
189             }
190             successMsg.style.display = 'block';
191             regForm.reset();
192             submitBtn.disabled = true;
193             setTimeout(() => { successMsg.style.display = 'none'; }, 4000);
194         });
195     }
196     </script></body></html>

```

Output:

The screenshot displays a registration form titled "SRU Tech Fest Registration". The form contains the following fields and values:

- Name:** Hasini
- Email:** hasini@gmail.com
- Phone:** 1234567890
- Department:** Computer Science (selected from a dropdown menu)
- Event Selection:** AI Workshop (selected from a dropdown menu)
- Register Button:** A blue button labeled "Register" is visible at the bottom of the form.

Below the form, a green message states: "Registration successful! See you at the fest!"

Observation:

The code provides a modern, responsive registration form with fields for student details and event selection. It uses CSS for an attractive layout and JavaScript for real-time validation (checking email format, phone length, and required fields). Error messages are displayed instantly, and a success message confirms registration, making the form both user-friendly and reliable.

Task#4:

AI-Assisted E-Commerce Product Page

Prompt: A startup wants a basic e-commerce product page to display products with prices and an “Add to Cart” button.

- generate a grid-based product catalog in HTML/CSS.
- include a cart counter at the top-right corner of the page. give html,css and js in the same file.

Code generated:

```
lab13 > ai_lab_14 > cart.html > html > head > style > h1
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Startup E-Commerce Product Page</title>
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <style>
8     :root {
9       --primary: #d18072;
10      --accent: #74726c;
11      --bg: #f7f9fc;
12      --card-bg: #fff;
13      --text: #222;
14      --shadow: 0 2px 10px rgba(0,0,0,0.08);
15    }
16    body {
17      margin: 0;
18      background: var(--bg);
19      font-family: 'Segoe UI', Arial, sans-serif;
20      color: var(--text);
21    }
22    .cart-counter {
23      position: fixed;
24      top: 25px;
25      right: 40px;
26      background: var(--primary);
27      color: #fff;
28      font-size: 1.06rem;
29      font-weight: bold;
30      border-radius: 50px;
31      padding: 0.55em 1em;
32      box-shadow: var(--shadow);
33      z-index: 50;
34      display: flex;
35      align-items: center;
36      gap: 0.5em;
37    }
38    .cart-icon {
39      font-size: 1.5em;
40      vertical-align: middle;
41    }
42    .container {
```

```

42 .container {
43   max-width: 1200px;
44   margin: 2.5rem auto;
45   padding: 0 1rem;
46 }
47 h1 {
48   text-align: center;
49   color: var(--primary);
50   margin-bottom: 2.5rem;
51 }
52 .product-grid {
53   display: grid;
54   grid-template-columns: repeat(auto-fit, minmax(240px, 1fr));
55   gap: 2rem;
56 }
57 .product-card {
58   background: var(--card-bg);
59   border-radius: 14px;
60   box-shadow: var(--shadow);
61   padding: 1.4rem 1rem 1rem 1rem;
62   text-align: center;
63   transition: box-shadow 0.23s, transform 0.23s;
64   position: relative;
65   display: flex;
66   flex-direction: column;
67   align-items: center;
68 }
69 .product-card: hover {
70   box-shadow: 0 4px 22px rgba(0,119,237,0.17);
71   transform: scale(1.028);
72 }
73 .product-card img {
74   width: 100%;
75   max-width: 118px;
76   height: 110px;
77   object-fit: contain;
78   margin-bottom: 1rem;
79   border-radius: 4px;
80   background: #f1f3f7;
81 }
82 .product-card h3 {
83   margin: 0.5rem 0 0.2rem 0;
84   color: var(--primary);
85 }
86 .product-card p {

```

```

87   margin: 0.3rem 0 0.5rem 0;
88   font-size: 1.02rem;
89   color: #333;
90 }
91 .product-card .price {
92   font-size: 1.05rem;
93   font-weight: bold;
94   color: var(--accent);
95   margin-bottom: 1rem;
96 }
97 .add-cart-btn {
98   background: var(--primary);
99   color: #fff;
100   border: none;
101   border-radius: 9px;
102   padding: 0.62rem 1.3rem;
103   font-size: 1.04rem;
104   font-weight: bold;
105   box-shadow: 0 1px 7px rgba(0,119,237,0.09);
106   transition: background 0.16s;
107   cursor: pointer;
108 }
109 .add-cart-btn:active,
110 .add-cart-btn: hover {
111   background: #005db0;
112 }
113 @media (max-width: 800px) {
114   .container { padding: 0 0.5rem; }
115   .product-grid { gap: 1rem; }
116 }
117 @media (max-width: 500px) {
118   h1 { font-size: 1.6rem; }
119   .cart-counter { padding: 0.4em 0.7em; font-size: 1rem; top: 15px; right: 12px; }
120   .product-card img { height: 75px; max-width: 92px; }
121 }
122 
```

```

123 </style></head>
124 <body>
125   <div class="cart-counter">
126     <span class="cart-icon">&#128722;</span>
127     <span id="cartCount">0</span></div>
128   <div class="container">
129     <h1>Shop Our Products</h1>
130     <div class="product-grid">

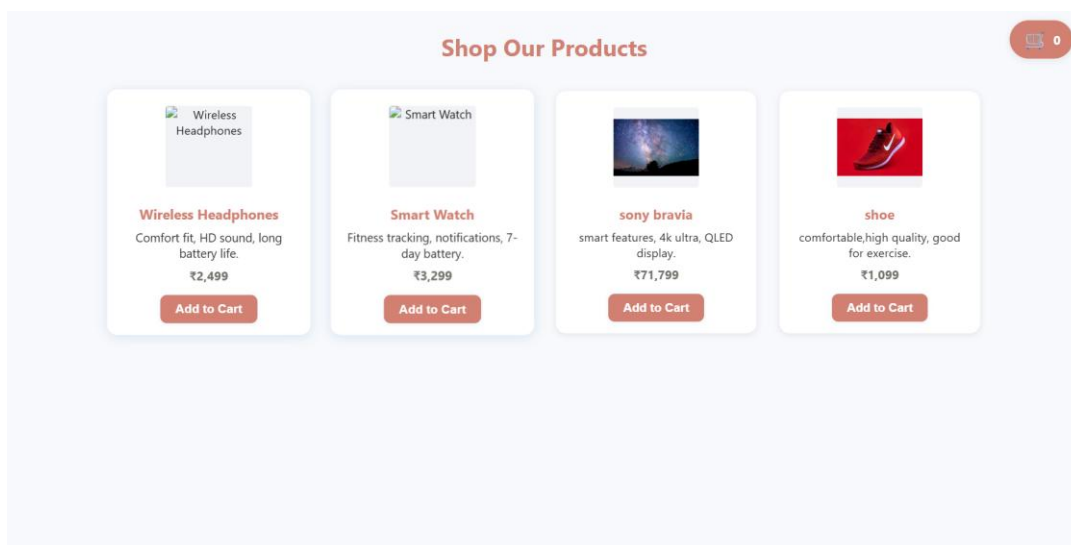
```

```

118 <button class="add-cart-btn">Add to Cart</button></div>
119 <div class="product-card">
120 
122 <h3>Smart Watch</h3>
123 <p>Fitness tracking, notifications, 7-day battery.</p>
124 <div class="price">₹3,299</div>
125 <button class="add-cart-btn">Add to Cart</button></div>
126 <div class="product-card">
127 
129 <h3>sony bravia</h3>
130 <p>smart features, 4k ultra, QLED display.</p>
131 <div class="price">₹71,799</div>
132 <button class="add-cart-btn">Add to Cart</button></div>
133 <div class="product-card">
134 
136 <h3>shoe </h3>
137 <p>comfortable,high quality, good for exercise.</p>
138 <div class="price">₹1,099</div>
139 <button class="add-cart-btn">Add to Cart</button>
140 </div></div></div>
141 <script>
142 let cartCount = 0;
143 const cartCountSpan = document.getElementById('cartCount');
144 const addCartBtns = document.querySelectorAll('.add-cart-btn');
145 addCartBtns.forEach(btn => {
146   btn.addEventListener('click', () => {
147     cartCount++;
148     cartCountSpan.textContent = cartCount;
149     btn.textContent = 'Added!';
150     btn.disabled = true;
151     setTimeout(() => {
152       btn.textContent = 'Add to Cart';
153       btn.disabled = false;
154     }, 1200);});});
155 </script></body></html>

```

Output:



Observation:

The code creates a basic e-commerce product page using HTML, CSS, and JavaScript in a single file. It displays products in a responsive grid layout with images, names, prices, and an “Add to Cart” button. A cart counter is placed at the top-right corner inside the header and dynamically updates whenever the user clicks “Add to Cart.” The implementation is simple, lightweight, and serves as a starting point for building a full shopping cart system.

