

AI ASSISTED CODING

ASSIGNMENT-14

Name: B.Hasini

Roll no: 2503A51L13

Batch: 24BTCAICSB19

TASK-1:

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:

```
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       * {
9         box-sizing: border-box;
10        margin: 0;
11        padding: 0;
12        font-family: Arial, sans-serif;
13      }
14
15     body {
16       line-height: 1.6;
17       background-color: #f4f4f4;
18       color: #333;
19     }
20
21     header, section {
22       padding: 40px 20px;
23     }
24
```

```
20      Syntax: <padding-top> (l, t, r, b)
21      See MDN Reference
22      padding: 40px 20px;
23      }
24
25      header {
26          background-color: #333;
27          color: #fff;
28          text-align: center;
29      }
30
31      nav {
32          margin-top: 10px;
33      }
34
35      nav a {
36          color: #fff;
37          margin: 0 15px;
38          text-decoration: none;
39          font-weight: bold;
40      }
41
42      nav a:hover {
43          text-decoration: underline;
44      }
45
46      .about, .contact {
47          background-color: #fff;
48      }
49
50      .projects {
51          background-color: #eaeaea;
52      }
53
54      .project-grid {
55          display: flex;
56          flex-wrap: wrap;
57          gap: 20px;
58          justify-content: center;
59      }
60
61      .project-card {
62          background-color: #fff;
63          border: 1px solid #ccc;
64          border-radius: 8px;
65          width: 300px;
66          padding: 20px;
67          transition: transform 0.3s ease, box-shadow 0.3s ease;
68      }
```

```
75      @media (max-width: 768px) {
76        .project-grid {
77          flex-direction: column;
78          align-items: center;
79        }
80      }
81    </style>
82  </head>
83  <body>
84    <header>
85      <h1>Hi, I'm Hasini</h1>
86      <nav>
87        <a href="#home">Home</a>
88        <a href="#about">About</a>
89        <a href="#projects">Projects</a>
90        <a href="#contact">Contact</a>
91      </nav>
92    </header>
93
94    <section id="home">
95      <h2>Welcome to My Portfolio</h2>
96      <p>Explore my work, skills, and how to reach me.</p>
97    </section>
98
99    <section id="about" class="about">
100      <h2>About Me</h2>
101      <p>I'm a developer focused on building reliable systems and interactive visualizations.</p>
102    </section>
```

```
<section id="projects" class="projects">
  <h2>Projects</h2>
  <div class="project-grid">
    <div class="project-card">
      <h3>Network Visualizer</h3>
      <p>Interactive tool to explore device roles and topologies.</p>
    </div>
    <div class="project-card">
      <h3>Data Dashboard</h3>
      <p>Plotly-based charts with hover tooltips and filters.</p>
    </div>
    <div class="project-card">
      <h3>Error Tracker</h3>
      <p>Logs, retries, and diagnostics for API and system errors.</p>
    </div>
  </div>
</section>
```

```

    <p>Plotly-based charts with hover tooltips and filters.</p>
  </div>
  <div class="project-card">
    <h3>Error Tracker</h3>
    <p>Logs, retries, and diagnostics for API and system errors.</p>
  </div>
</div>
</section>

<section id="contact" class="contact">
  <h2>Contact</h2>
  <p>Email: hasini@example.com</p>
  <p>GitHub: github.com/hasini</p>
</section>
</body>
</html>

```

```

2   <html lang="en">
3     <head>
7       <style>
154      /* Responsive */
155      @media (min-width:720px){
156        .content-grid{ grid-template-columns: 1fr 1fr; }
157        .about{ grid-column:1 / 2; }
158        .projects{ grid-column:2 / 3; }
159        .projects-grid{ grid-template-columns: repeat(2, 1fr); }
160      }
161      @media (min-width:1000px){
162        main{ padding:2.5rem; }
163        .hero{ grid-template-columns: 1fr 380px; gap:2rem; }
164        .projects-grid{ grid-template-columns: repeat(3, 1fr); }
165      }
166
167      /* Utility */
168      .muted{ color:var(--muted) }
169      a.cta{ color:var(--accent); text-decoration:none; font-weight:700 }
170    </style>
171  </head>
172  <body>
173    <div class="site" role="document">
174      <header>
175        <div class="brand">
176          <div class="logo">PJ</div>
177          <div>
178            <div style="font-weight:800">Susmija Punnam</div>
179            <div class="muted" style="font-size:.85rem">Frontend • Developer • Design Select Encoding

```

Output:

The screenshot shows a web browser window with the title "My Portfolio". The URL in the address bar is "127.0.0.1:5500/e-commers.html". The page content is as follows:

Hi, I'm Hasini

Home About Projects Contact

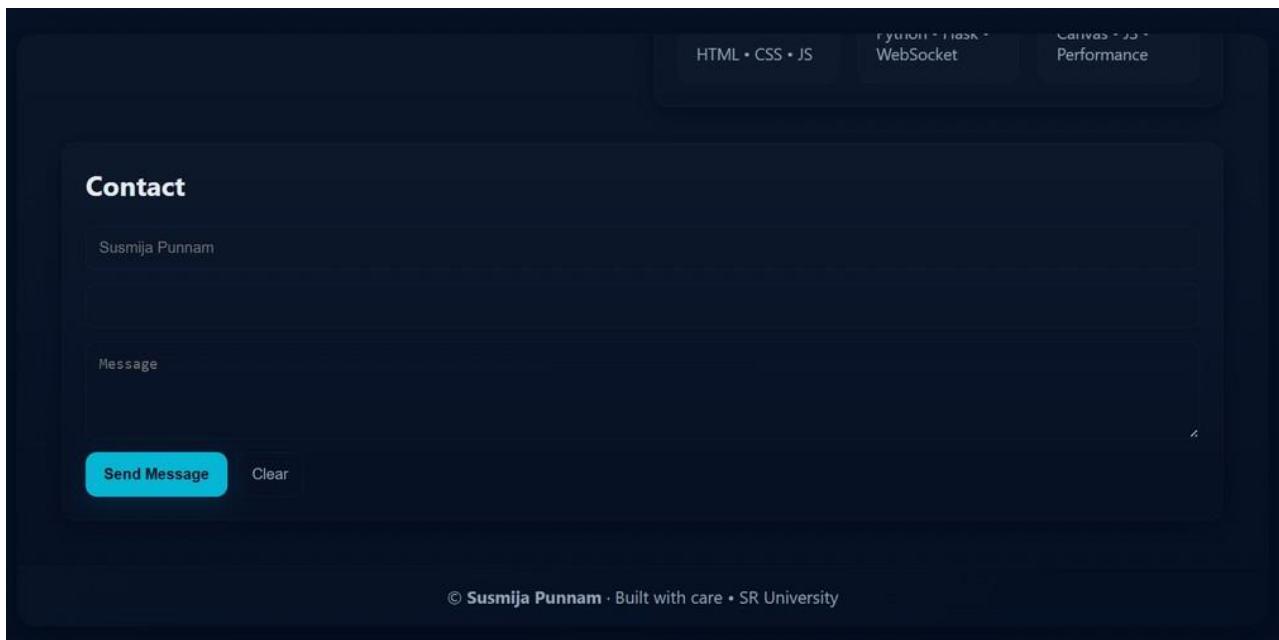
Welcome to My Portfolio
Explore my work, skills, and how to reach me.

About Me
I'm a developer focused on building reliable systems and interactive visualizations.

Projects

Network Visualizer Interactive tool to explore device roles and topologies.	Data Dashboard Plotly-based charts with hover tooltips and filters.	Error Tracker Logs, retries, and diagnostics for API and system errors.
---	---	---

At the bottom of the screen, the Windows taskbar is visible, showing the Start button, a search bar, pinned icons for File Explorer, Microsoft Edge, and other applications, and system status indicators like battery level, signal strength, and the date and time (10-11-2025).



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

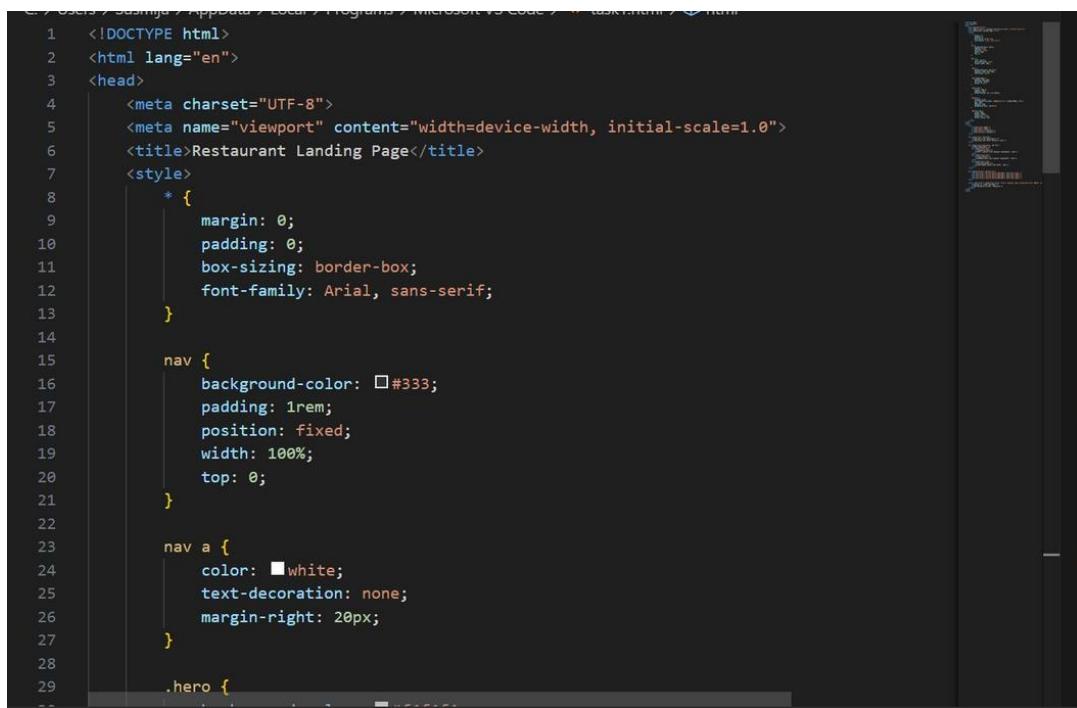
Scenario:

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.

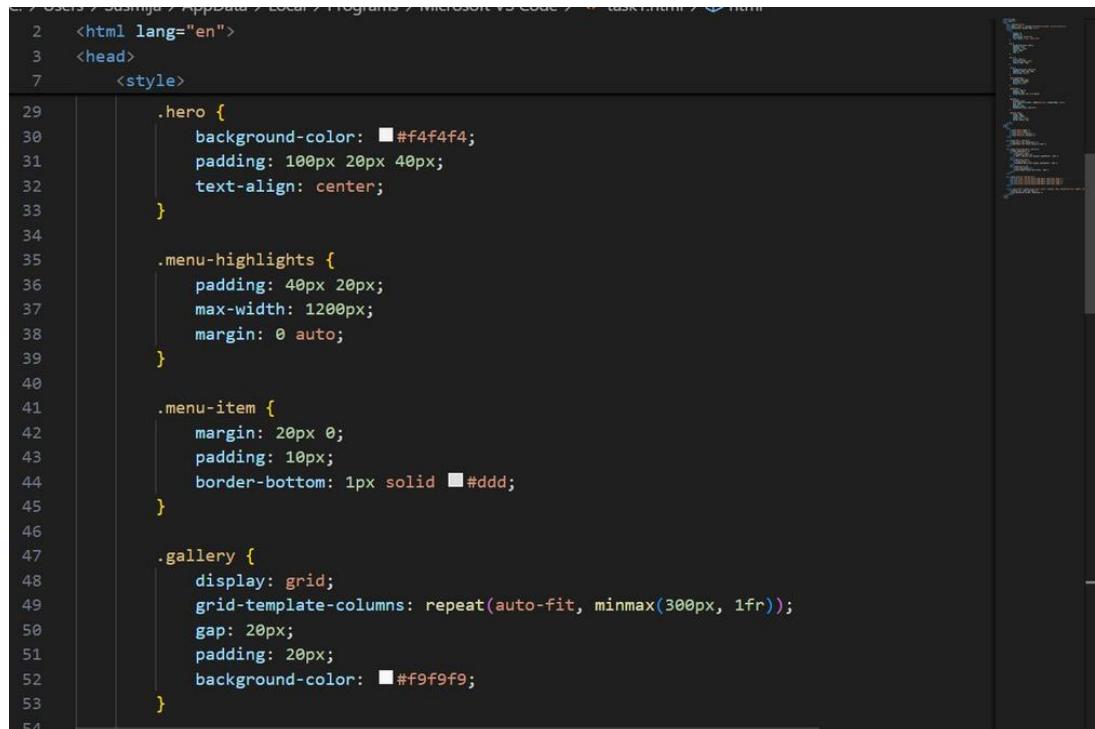
Prompt:

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

Code generated:



```
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>Restaurant Landing Page</title>
7      <style>
8          * {
9              margin: 0;
10             padding: 0;
11             box-sizing: border-box;
12             font-family: Arial, sans-serif;
13         }
14
15         nav {
16             background-color: #333;
17             padding: 1rem;
18             position: fixed;
19             width: 100%;
20             top: 0;
21         }
22
23         nav a {
24             color: white;
25             text-decoration: none;
26             margin-right: 20px;
27         }
28
29         .hero {
```



```
2  <html lang="en">
3  <head>
7  <style>
29         .hero {
30             background-color: #f4f4f4;
31             padding: 100px 20px 40px;
32             text-align: center;
33         }
34
35         .menu-highlights {
36             padding: 40px 20px;
37             max-width: 1200px;
38             margin: 0 auto;
39         }
40
41         .menu-item {
42             margin: 20px 0;
43             padding: 10px;
44             border-bottom: 1px solid #ddd;
45         }
46
47         .gallery {
48             display: grid;
49             grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
50             gap: 20px;
51             padding: 20px;
52             background-color: #f9f9f9;
53         }
```

```
2  <html lang="en">
3  <head>
7  	<style>
55  .gallery img {
56  	width: 100%;
57  	height: 200px;
58  	object-fit: cover;
59  	border-radius: 8px;
60  }
61  </style>
62 </head>
63 <body>
64 	<nav>
65  	<a href="#home">Home</a>
66  	<a href="#menu">Menu</a>
67  	<a href="#gallery">Gallery</a>
68  	<a href="#contact">Contact</a>
69  </nav>
70
71 	<div class="hero" id="home">
72  	<h1>Welcome to Our Restaurant</h1>
73  	<p>Experience the finest dining in town</p>
74  </div>
75
76 	<div class="menu-highlights" id="menu">
77  	<h2>Menu Highlights</h2>
78  	<div class="menu-item">
79  	<h3>Signature Dish</h3>
80  	<p>Chef's special with seasonal ingredients - $24</p>
81  </div>
82  	<div class="menu-item">
83  	<h3>Fresh Pasta</h3>
84  	<p>Handmade daily with organic ingredients - $18</p>
85  </div>
86  	<div class="menu-item">
87  	<h3>Grilled Salmon</h3>
88  	<p>Wild-caught salmon with herbs - $26</p>
89  </div>
90  </div>
91
92 	<div class="gallery" id="gallery">
93  	
94  	
95  	
96  	
97  </div>
98
99 	<footer id="contact" style="text-align: center; padding: 20px; background-color: #333;">
100  	<p>Contact us: (555) 123-4567</p>
101  	<p>123 Restaurant Street, Foodville</p>
102  </footer>
```

```
2  <html lang="en">
63 <body>
76 	<div class="menu-highlights" id="menu">
77  	<h2>Menu Highlights</h2>
78  	<div class="menu-item">
79  	<h3>Signature Dish</h3>
80  	<p>Chef's special with seasonal ingredients - $24</p>
81  </div>
82  	<div class="menu-item">
83  	<h3>Fresh Pasta</h3>
84  	<p>Handmade daily with organic ingredients - $18</p>
85  </div>
86  	<div class="menu-item">
87  	<h3>Grilled Salmon</h3>
88  	<p>Wild-caught salmon with herbs - $26</p>
89  </div>
90  </div>
91
92 	<div class="gallery" id="gallery">
93  	
94  	
95  	
96  	
97  </div>
98
99 	<footer id="contact" style="text-align: center; padding: 20px; background-color: #333;">
100  	<p>Contact us: (555) 123-4567</p>
101  	<p>123 Restaurant Street, Foodville</p>
102  </footer>
```

```

2   <html lang="en">
3     <body>
4       <div class="menu-highlights" id="menu">
5         ...
6       </div>
7     </div>
8
9     <div class="gallery" id="gallery">
10      
11      
12      
13      
14    </div>
15
16    <footer id="contact" style="text-align: center; padding: 20px; background-color: #333;">
17      <p>Contact us: (555) 123-4567</p>
18      <p>123 Restaurant Street, Foodville</p>
19    </footer>
20  </body>
21 </html>

```

Output:

Home Menu Gallery Contact

Welcome to Our Restaurant

Experience the finest dining in town

Menu Highlights

Signature Dish

Chef's special with seasonal ingredients - \$24

Fresh Pasta

Handmade daily with organic ingredients - \$18

Grilled Salmon

Wild-caught salmon with herbs - \$26

Home Menu Gallery Contact







Contact us: (555) 123-4567
123 Restaurant Street, Foodville

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:

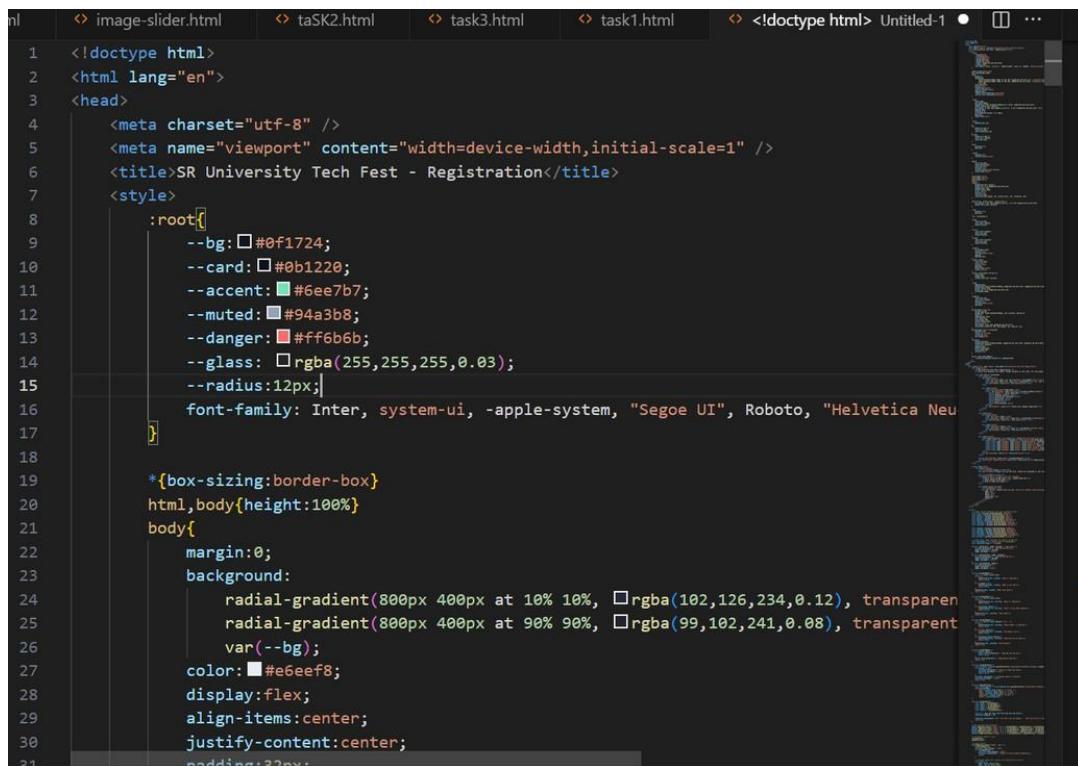
SR University is hosting a tech fest. They need a web-based registration form for students. The form must validate user input in real-time.

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:



A screenshot of a code editor showing a file named 'task3.html'. The code is a combination of CSS and HTML. It includes a head section with meta tags for charset and viewport, and a title 'SR University Tech Fest - Registration'. The body section contains a style block with various CSS variables and rules. The style block includes a ':root' selector with properties like --bg, --card, --accent, --muted, --danger, --glass, and --radius. It also includes a '*' selector with 'box-sizing: border-box' and a 'body' selector with 'height: 100%', 'margin: 0', and a background gradient. The code is written in a dark-themed code editor with syntax highlighting.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width,initial-scale=1" />
    <title>SR University Tech Fest - Registration</title>
    <style>
        :root{
            --bg:#0f1724;
            --card:#0b1220;
            --accent:#6ee7b7;
            --muted:#94a3b8;
            --danger:#ff6b6b;
            --glass:rgba(255,255,255,0.03);
            --radius:12px;
            font-family: Inter, system-ui, -apple-system, "Segoe UI", Roboto, "Helvetica Neue", sans-serif;
        }

        *{box-sizing:border-box}
        html,body{height:100%}
        body{
            margin:0;
            background:
                radial-gradient(800px 400px at 10% 10%, rgba(102,126,234,0.12), transparent),
                radial-gradient(800px 400px at 90% 90%, rgba(99,102,241,0.08), transparent)
                var(--bg);
            color:#e6eff8;
            display:flex;
            align-items:center;
            justify-content:center;
            padding:0 20px;
        }
    </style>
</head>
<body>
</body>
</html>
```

The screenshot shows a code editor with several tabs at the top: image-slider.html, taSK2.html, task3.html, task1.html, and Untitled-1. Untitled-1 is the active tab, displaying the following CSS code:

```
*{box-sizing:border-box}
html,body{height:100%}
body{
  margin:0;
  background:
    radial-gradient(800px 400px at 10% 10%, transparent, var(--bg));
  color:#e6eff8;
  display:flex;
  align-items:center;
  justify-content:center;
  padding:32px;
  -webkit-font-smoothing:antialiased;
  -moz-osx-font-smoothing:grayscale;
}

.card{
  width:100%;
  max-width:900px;
  background:linear-gradient(180deg,var(--card), transparent);
  border-radius:var(--radius);
  box-shadow: 0 10px 30px rgba(2,6,23,0.7), 0 1px 0 transparent;
  padding:28px;
  display:grid;
  grid-template-columns: 1fr 420px;
```

The screenshot shows a code editor with several tabs at the top: image-slider.html, taSK2.html, task3.html, task1.html, and Untitled-1. Untitled-1 is the active tab, displaying the following CSS code:

```
<html lang="en">
<head>
  <style>
    .hero{
      padding:18px 8px;
    }
    h1{
      margin:0 0 8px 0;
      font-size:20px;
      letter-spacing:0.2px;
    }
    p.lead{
      margin:0 0 18px 0;
      color:var(--muted);
      font-size:14px;
    }

    form{
      display:grid;
      gap:12px;
    }

    .field{
      display:flex;
      flex-direction:column;
    }

    label{
      font-size:13px;
      margin-bottom:8px;
    }
```

```
<html lang="en">
<head>
    <style>
        input[type="text"],
        input[type="email"],
        input[type="tel"],
        select,
        .events {
            background:var(--glass);
            border:1px solid #rgba(255,255,255,0.04);
            padding:12px 14px;
            border-radius:10px;
            color:inherit;
            outline:none;
            font-size:14px;
            transition:box-shadow .15s, border-color .15s, transform .05s;
        }

        input:focus, select:focus, .events:focus {
            box-shadow: 0 6px 18px #rgba(0,0,0,0.6), 0 0 4px #rgba(110,231,183,0.06);
            border-color:var(--accent);
        }

        .row{
            display:flex;
            gap:12px;
        }
        .row .field{flex:1}

        .hint{
```

```
<html lang="en">
<head>
    <style>
        .hint{
            font-size:12px;
            color:var(--muted);
            margin-top:6px;
        }

        .error{
            color:var(--danger);
            font-size:13px;
            margin-top:6px;
        }

        .valid{
            color:var(--accent);
            font-size:13px;
            margin-top:6px;
        }

        .events{
            min-height:110px;
            display:flex;
            flex-direction:column;
            gap:6px;
            padding:12px;
        }
        .events label{
            font-size:13px;
            color:inherit;
            display:flex;
```

Microsoft Edge

```
2 <html lang="en">
3 <head>
4   <style>
5     .events label{
6       display:flex;
7       gap:10px;
8       align-items:center;
9       cursor:pointer;
10    }
11   .events input[type="checkbox"]{
12     width:16px;
13     height:16px;
14     accent-color:var(--accent);
15   }
16
17   .side{
18     padding:18px;
19     background:linear-gradient(180deg, □rgba(255,255,255,0.02), □rgba(255,255,255,
20     border-radius:10px;
21     border:1px solid □rgba(255,255,255,0.02);
22     min-height:260px;
23   }
24
25   .summary{
26     font-size:13px;
27     color:var(--muted);
28     display:flex;
29     flex-direction:column;
30     gap:10px;
31   }
32 }
```

```
2 <html lang="en">
3 <head>
4   <style>
5     button[type="submit"]){
6       margin-top:8px;
7       background: linear-gradient(90deg, var(--accent), □#3b82f6);
8       border:0;
9       padding:12px 14px;
10      color:□#042017;
11      font-weight:600;
12      border-radius:10px;
13      cursor:pointer;
14      box-shadow: 0 8px 20px □rgba(59,130,246,0.12);
15      transition:transform .08s, box-shadow .12s, opacity .12s;
16    }
17    button[type="submit"]:disabled{
18      opacity:0.45;
19      cursor:not-allowed;
20      transform:none;
21      box-shadow:none;
22    }
23    .success{
24      margin-top:12px;
25      background:linear-gradient(90deg, □rgba(110,231,183,0.08), □rgba(59,130,246,0.
26      color:var(--accent));
27      padding:10px;
28      border-radius:8px;
29      font-weight:600;
30      display:none;
31    }
32 }
```

```
2 <html lang="en">
195 <body>
282   <script>
411     // Wire up real-time validation
412     nameEl.addEventListener('input', () => { validateName(); updatePreview(); validateFo
413     emailEl.addEventListener('input', () => { validateEmail(); updatePreview(); validate
414     phoneEl.addEventListener('input', () => { validatePhone(); updatePreview(); validate
415     deptEl.addEventListener('change', () => { validateDept(); updatePreview(); validateF
416     eventsEl.addEventListener('change', () => { validateEvents(); updatePreview(); valid
417
418     // Initialize states on load
419     validateForm();
420     updatePreview();
421
422     // Submit handler
423     form.addEventListener('submit', (e) => {
424       e.preventDefault();
425       successEl.style.display = 'none';
426       if (!validateForm()) {
427         successEl.style.display = 'none';
428         formStatus.textContent = 'Please fix errors before submitting.';
429         return;
430       }
431
432       // Simulate submission (replace with fetch/ajax to server)
433       const data = {
434         name: nameEl.value.trim(),
435         email: emailEl.value.trim(),
436         phone: phoneEl.value.trim(),
437         department: deptEl.value,
438       };
439     });
440
441     // Mock delay to mimic network call
442     submitBtn.disabled = true;
443     submitBtn.textContent = 'Submitting...';
444     setTimeout(() => {
445       successEl.style.display = 'block';
446       successEl.textContent = 'Registration successful! Thank you.';
447       submitBtn.textContent = 'Register';
448       // Optionally reset form:
449       // form.reset();
450       // Clear styles and re-validate for fresh state
451       submitBtn.disabled = false;
452       validateForm();
453     }, 800);
454
455     console.log('Submitted data:', data);
456   });
457 </script>
458 </body>
459 </html>
```

```
2 <html lang="en">
195 <body>
282   <script>
423     form.addEventListener('submit', (e) => {
433       const data = {
434         events: Array.from(eventsEl.querySelectorAll('input[type="checkbox"]':checked
435       );
436
437       // Mock delay to mimic network call
441       submitBtn.disabled = true;
442       submitBtn.textContent = 'Submitting...';
443       setTimeout(() => {
444         successEl.style.display = 'block';
445         successEl.textContent = 'Registration successful! Thank you.';
446         submitBtn.textContent = 'Register';
447         // Optionally reset form:
448         // form.reset();
449         // Clear styles and re-validate for fresh state
450         submitBtn.disabled = false;
451         validateForm();
452       }, 800);
453
454       console.log('Submitted data:', data);
455     });
456   </script>
457 </body>
458 </html>
```

Output:

The screenshot shows a registration form with the following fields and validation messages:

- Name:** susmija punnam (Valid email)
- Department:** Computer Science (Choose your academic department)
- Email:** punnamsusmija@gmail.com (Valid email)
- Phone:** 7674905200 (Valid phone)

Event Selection: (Choose one or more)

- Hackathon
- Coding Contest
- Robotics
- Paper Presentation
- UI/UX Design
- Tech Quiz

1 event(s) selected

Preview:

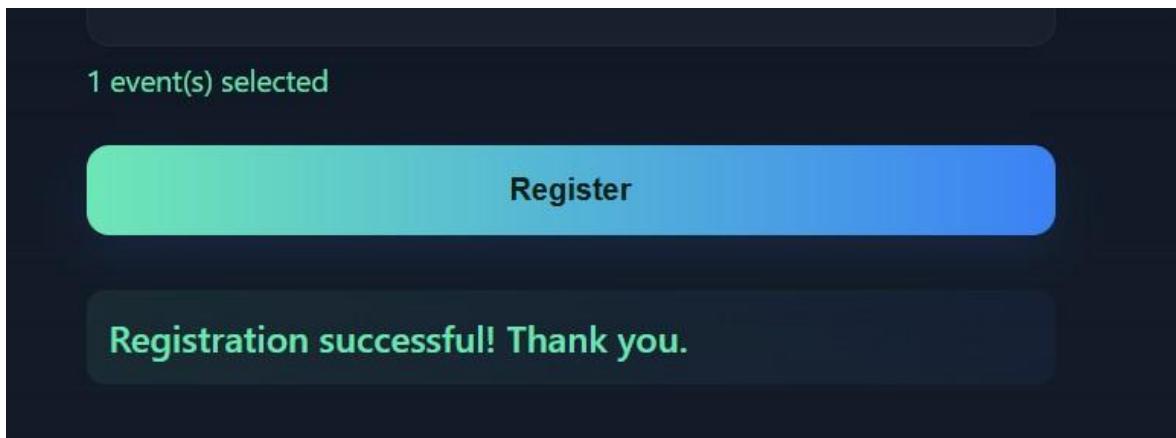
- Use a valid college email (e.g., name@college.edu)
- Phone should be 10 digits
- Pick at least one event

Registration Preview:

- Name: susmija punnam
- Email: punnamsusmija@gmail.com
- Phone: 7674905200
- Department: Computer Science
- Events: Hackathon

Buttons:

- Register (Large blue button)



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:

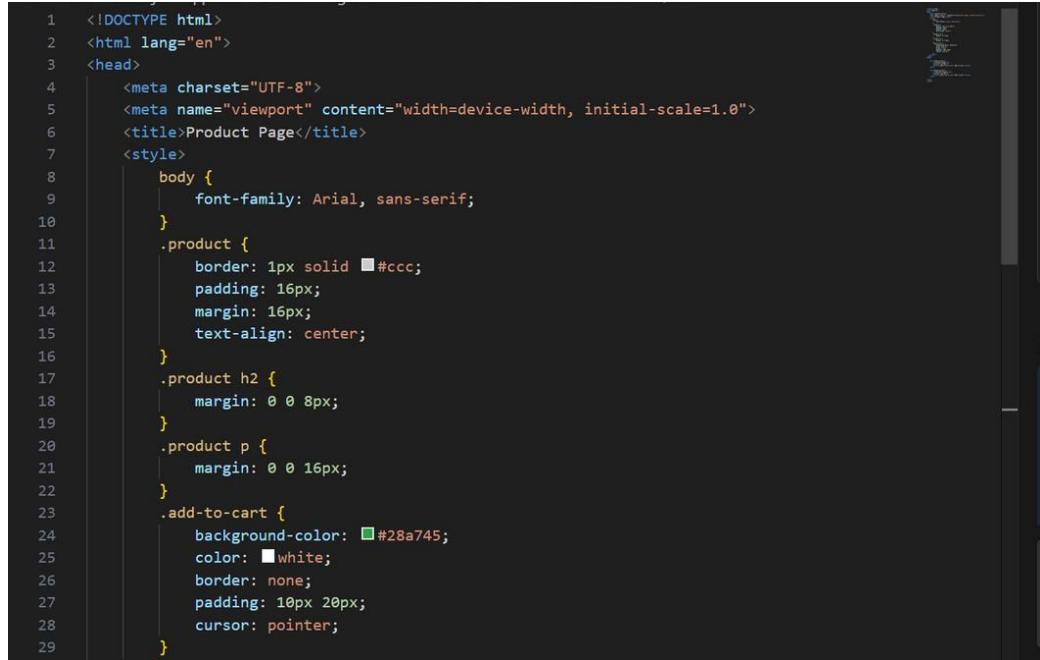
AI-Assisted E-Commerce Product Page

Prompt:

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

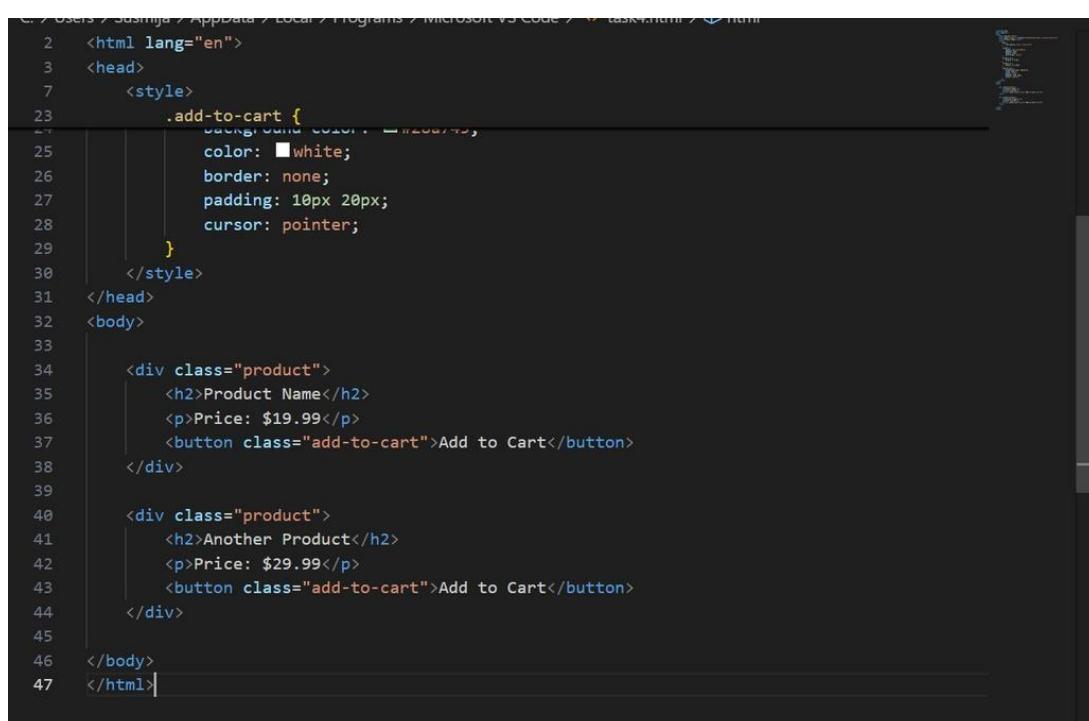
- generate a grid-based product catalog in HTML/CSS.
- include a cart counter at the top-right corner of the page. give html,css and js in the same file.

Code generated:

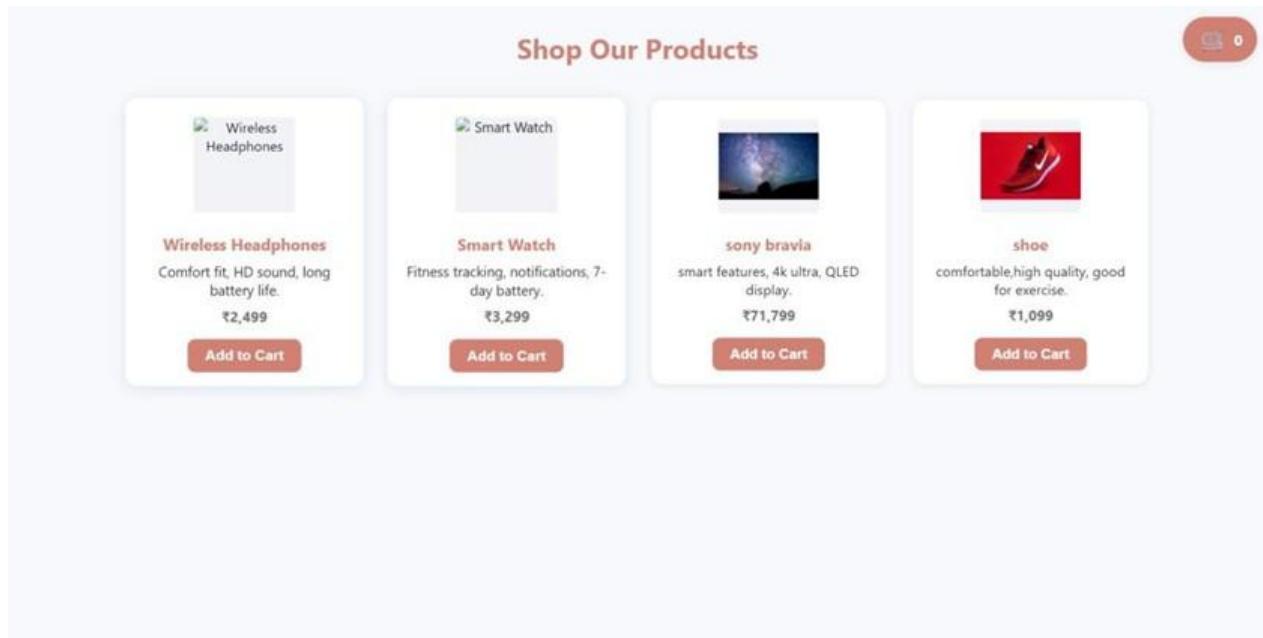


```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Product Page</title>
7      <style>
8          body {
9              font-family: Arial, sans-serif;
10         }
11         .product {
12             border: 1px solid #ccc;
13             padding: 16px;
14             margin: 16px;
15             text-align: center;
16         }
17         .product h2 {
18             margin: 0 0 8px;
19         }
20         .product p {
21             margin: 0 0 16px;
22         }
23         .add-to-cart {
24             background-color: #28a745;
25             color: white;
26             border: none;
27             padding: 10px 20px;
28             cursor: pointer;
29         }

```



```
C:\>Users\>Sushma\>AppData\>Local\>Programs\>Microsoft VS Code\>task.html>task.html
1  <html lang="en">
2  <head>
3      <style>
4          .add-to-cart {
5              background-color: #28a745;
6              color: white;
7              border: none;
8              padding: 10px 20px;
9              cursor: pointer;
10         }
11     </style>
12 </head>
13 <body>
14
15     <div class="product">
16         <h2>Product Name</h2>
17         <p>Price: $19.99</p>
18         <button class="add-to-cart">Add to Cart</button>
19     </div>
20
21     <div class="product">
22         <h2>Another Product</h2>
23         <p>Price: $29.99</p>
24         <button class="add-to-cart">Add to Cart</button>
25     </div>
26
27 </body>
28 </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.