AI ASSISTED CODING

ASSIGNMENT-14

Name: B.Hasini

Roll no: 2503A51L13

Batch: 24BTCAICSB19

TASK-1:

Scenario:

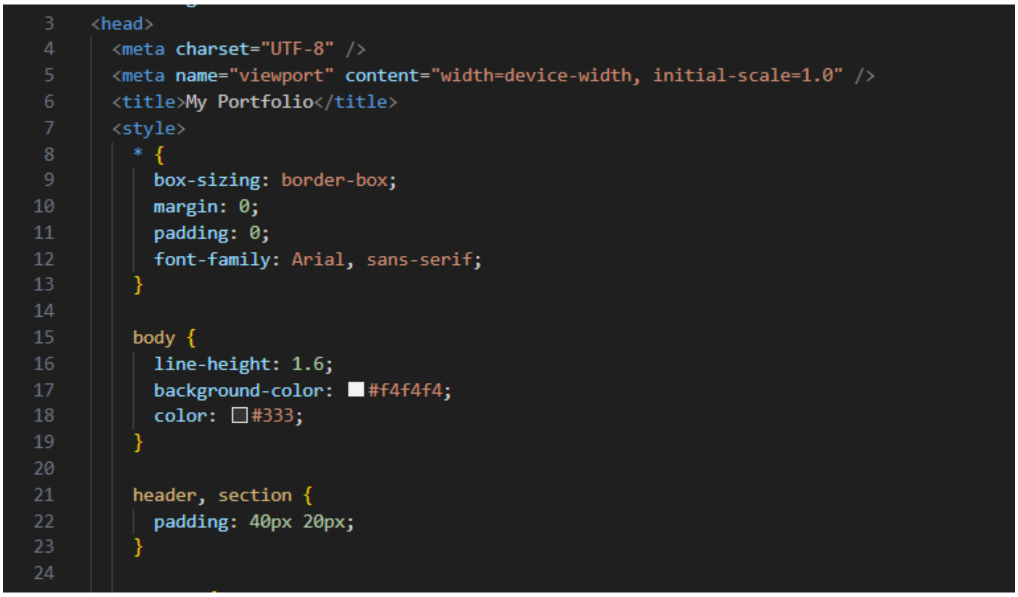
A student wants to showcase their projects, skills, and contact details in a portfolio website. Instead of writing all code manually, they want to speed up the process using GitHub Copilot.

- 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

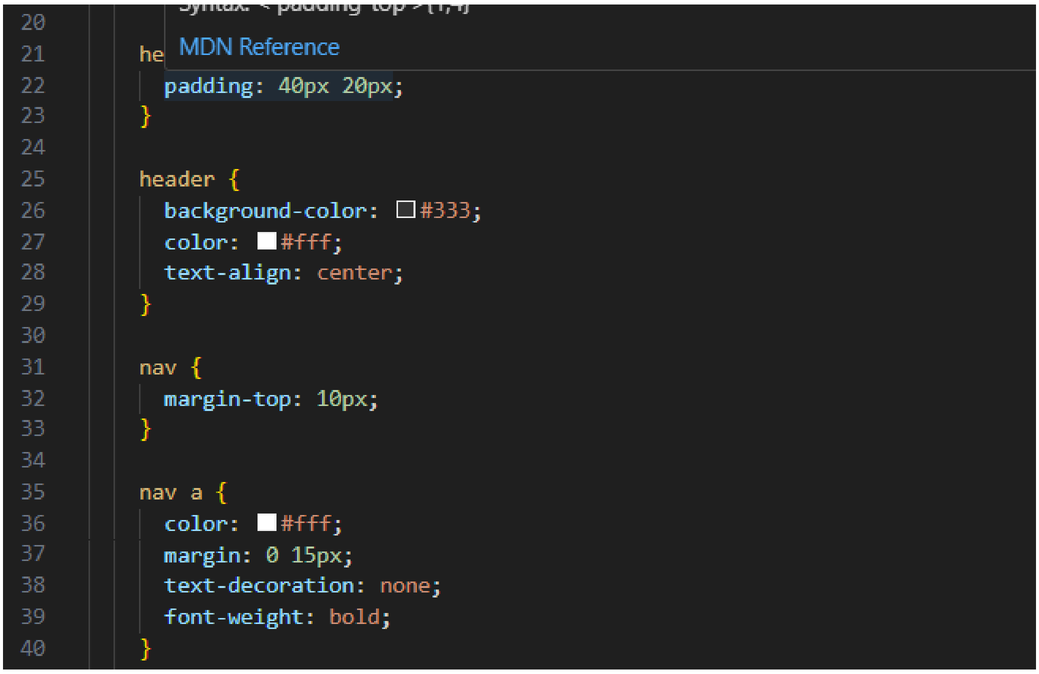
Prompt:

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.

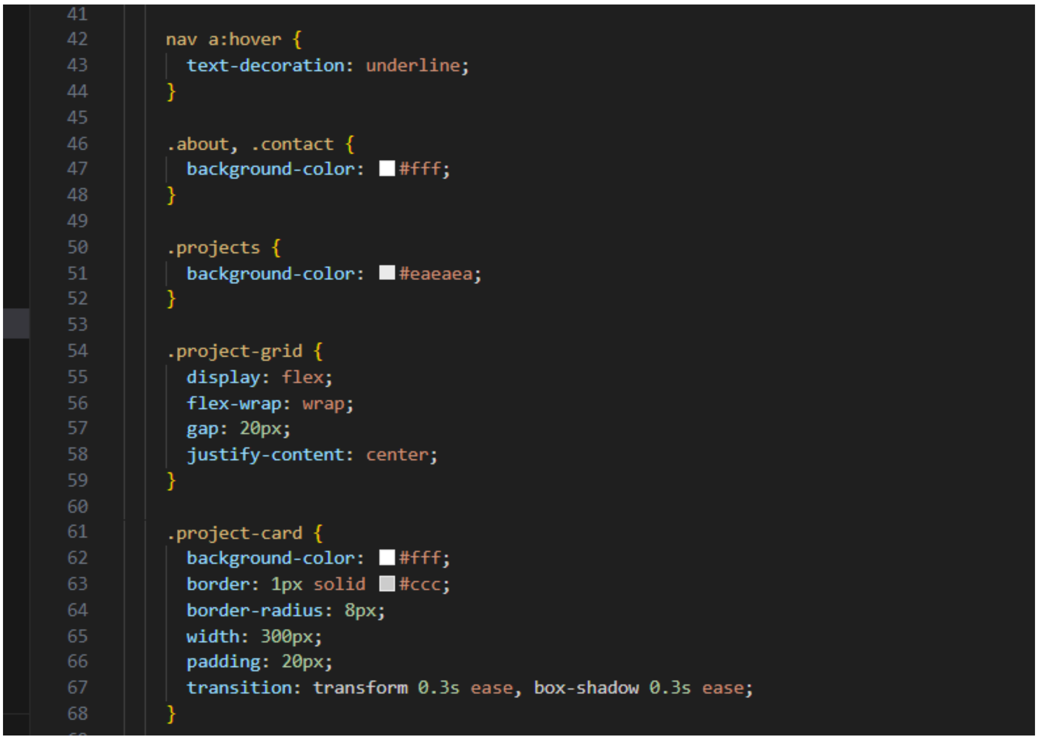
Code generated:



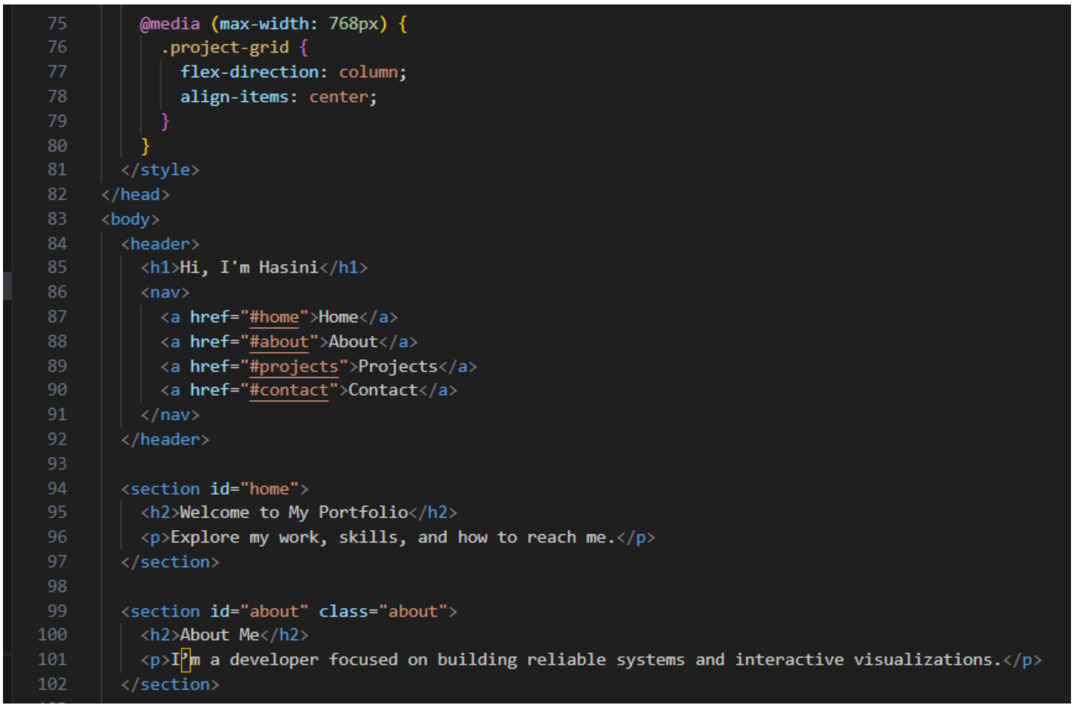
```
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>My Portfolio</title>
  <style>
    * {
      box-sizing: border-box;
      margin: 0;
      padding: 0;
      font-family: Arial, sans-serif;
    }

    body {
      line-height: 1.6;
      background-color: #f4f4f4;
      color: #333;
    }

    header, section {
      padding: 40px 20px;
    }

```



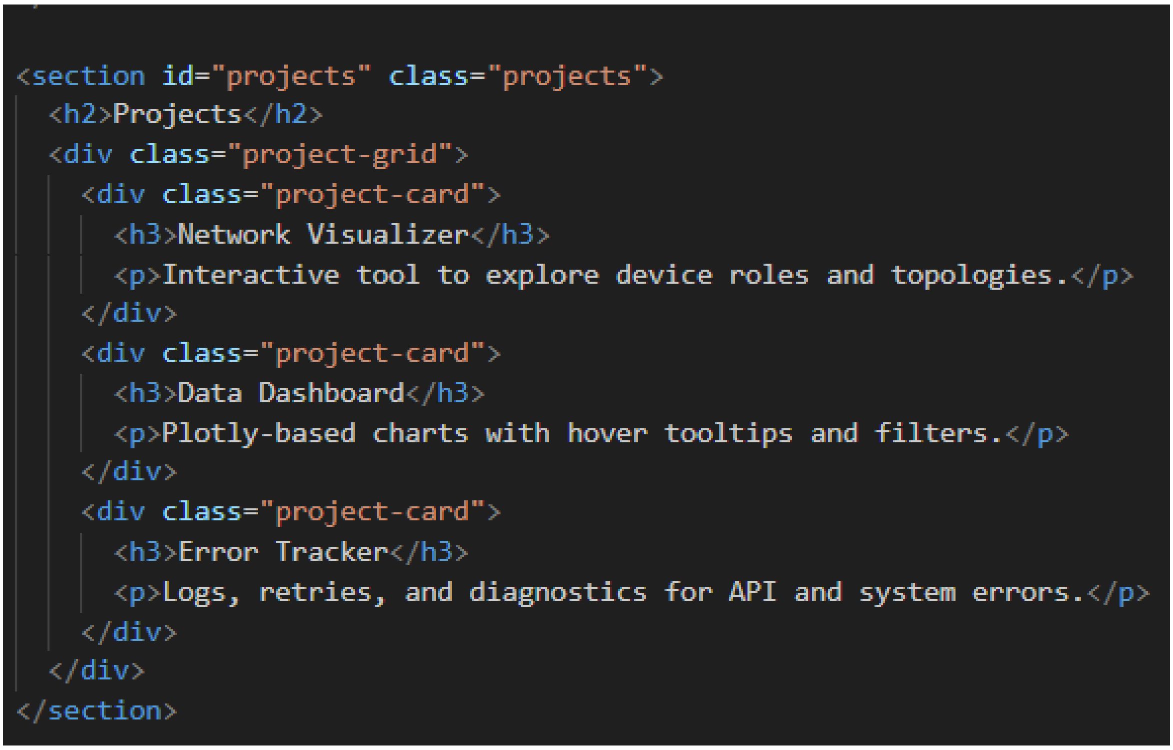
```css

he
  padding: 40px 20px;
}

header {
  background-color: #333;
  color: #fff;
  text-align: center;
}

nav {
  margin-top: 10px;
}

nav a {
  color: #fff;
  margin: 0 15px;
  text-decoration: none;
  font-weight: bold;
}
```



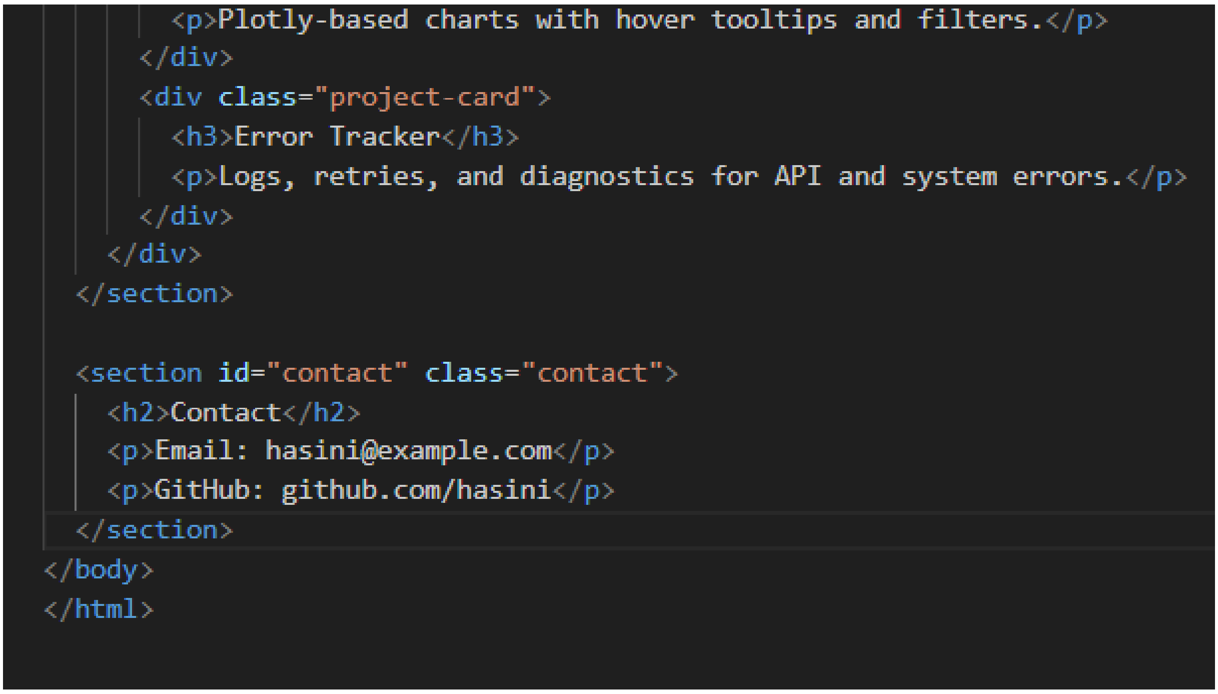
```css

nav a:hover {
  text-decoration: underline;
}

.about, .contact {
  background-color: #fff;
}

.projects {
  background-color: #eaeaea;
}

.project-grid {
  display: flex;
  flex-wrap: wrap;
  gap: 20px;
  justify-content: center;
}

.project-card {
  background-color: #fff;
  border: 1px solid #ccc;
  border-radius: 8px;
  width: 300px;
  padding: 20px;
  transition: transform 0.3s ease, box-shadow 0.3s ease;
}
```



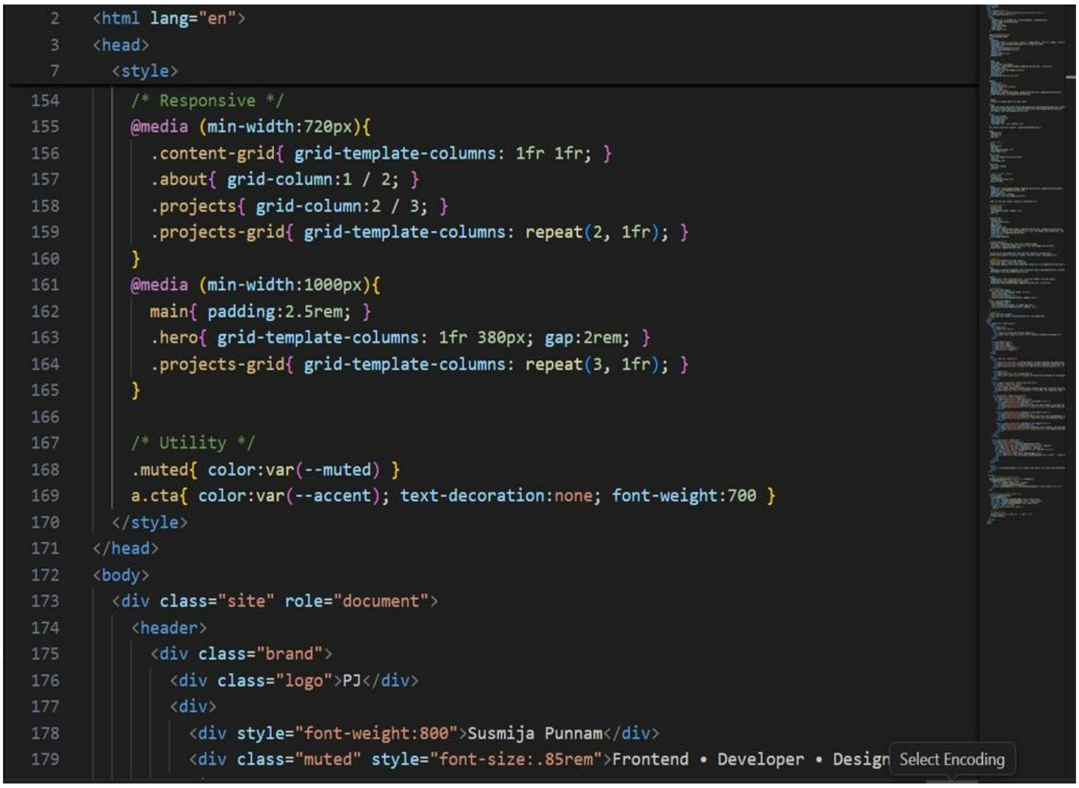
```
    @media (max-width: 768px) {
      .project-grid {
        flex-direction: column;
        align-items: center;
      }
    }
  </style>
</head>
<body>
  <header>
    <h1>Hi, I'm Hasini</h1>
    <nav>
      <a href="#home">Home</a>
      <a href="#about">About</a>
      <a href="#projects">Projects</a>
      <a href="#contact">Contact</a>
    </nav>
  </header>

  <section id="home">
    <h2>Welcome to My Portfolio</h2>
    <p>Explore my work, skills, and how to reach me.</p>
  </section>

  <section id="about" class="about">
    <h2>About Me</h2>
    <p>I'm a developer focused on building reliable systems and interactive visualizations.</p>
  </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>
```

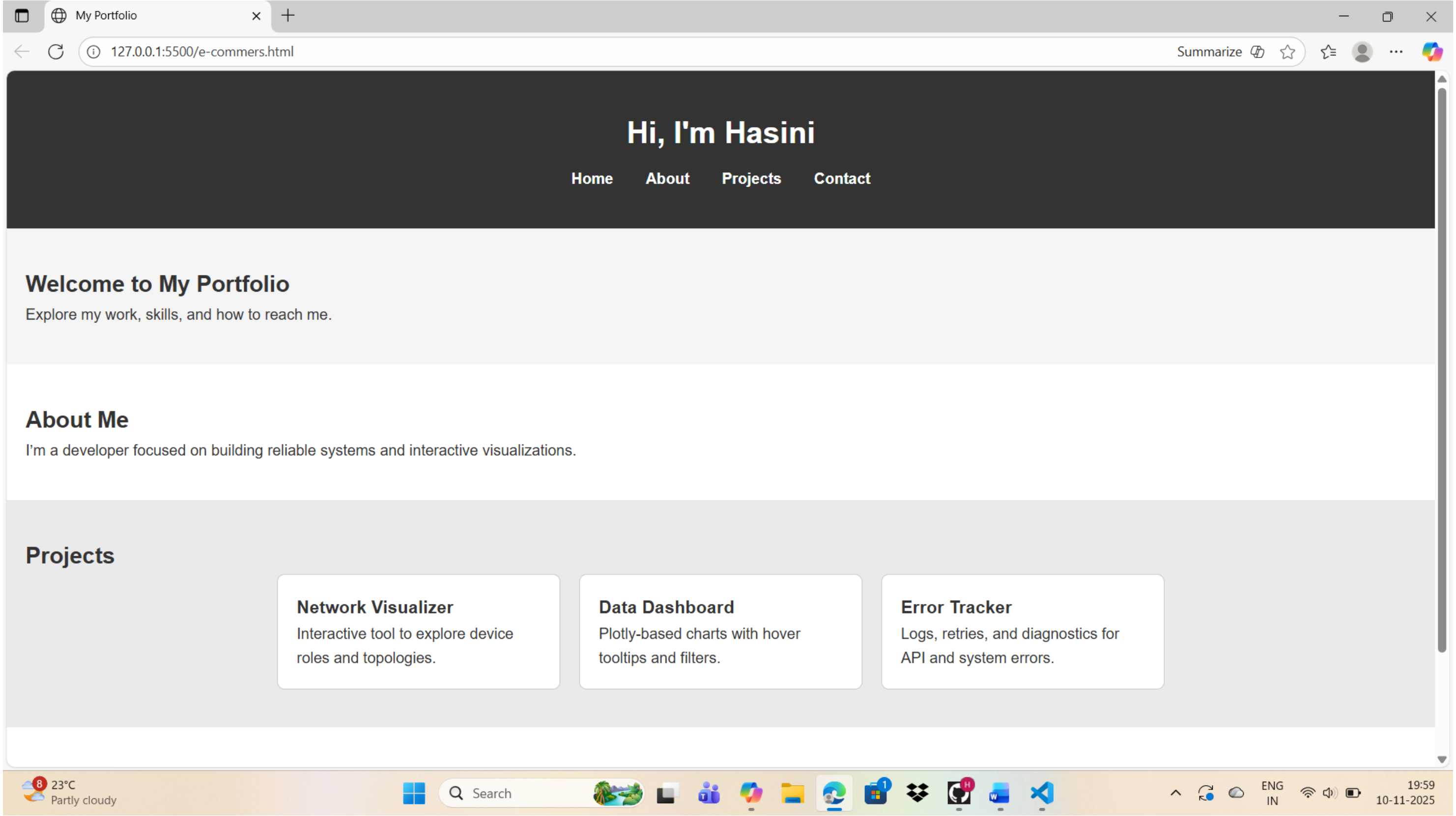


```
        <p>Plotly-based charts with hover tooltips and filters.</p>
      </div>
      <div class="project-card">
        <h3>Error Tracker</h3>
        <p>Logs, retries, and diagnostics for API and system errors.</p>
      </div>
    </div>
  </section>

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



```
<html lang="en">
<head>
  <style>
    /* Responsive */
    @media (min-width:720px){
      .content-grid{ grid-template-columns: 1fr 1fr; }
      .about{ grid-column:1 / 2; }
      .projects{ grid-column:2 / 3; }
      .projects-grid{ grid-template-columns: repeat(2, 1fr); }
    }
    @media (min-width:1000px){
      main{ padding:2.5rem; }
      .hero{ grid-template-columns: 1fr 380px; gap:2rem; }
      .projects-grid{ grid-template-columns: repeat(3, 1fr); }
    }

    /* Utility */
    .muted{ color:var(--muted) }
    a.cta{ color:var(--accent); text-decoration:none; font-weight:700 }
  </style>
</head>
<body>
  <div class="site" role="document">
    <header>
      <div class="brand">
        <div class="logo">PJ</div>
        <div>
          <div style="font-weight:800">Susmija Punnam</div>
          <div class="muted" style="font-size:.85rem">Frontend • Developer • Design
```

Output:



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

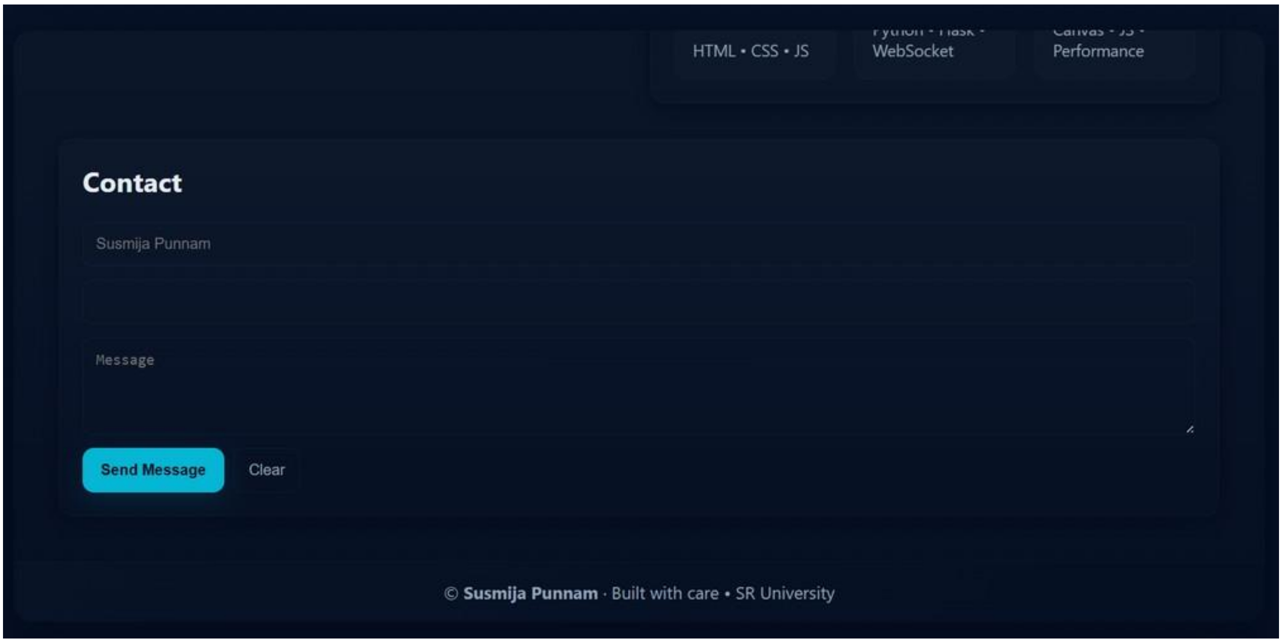
Interactive tool to explore device roles and topologies.

Data Dashboard

Plotly-based charts with hover tooltips and filters.

Error Tracker

Logs, retries, and diagnostics for API and system errors.



Susmija Punnam

© **Susmija Punnam** · Built with care • SR University

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

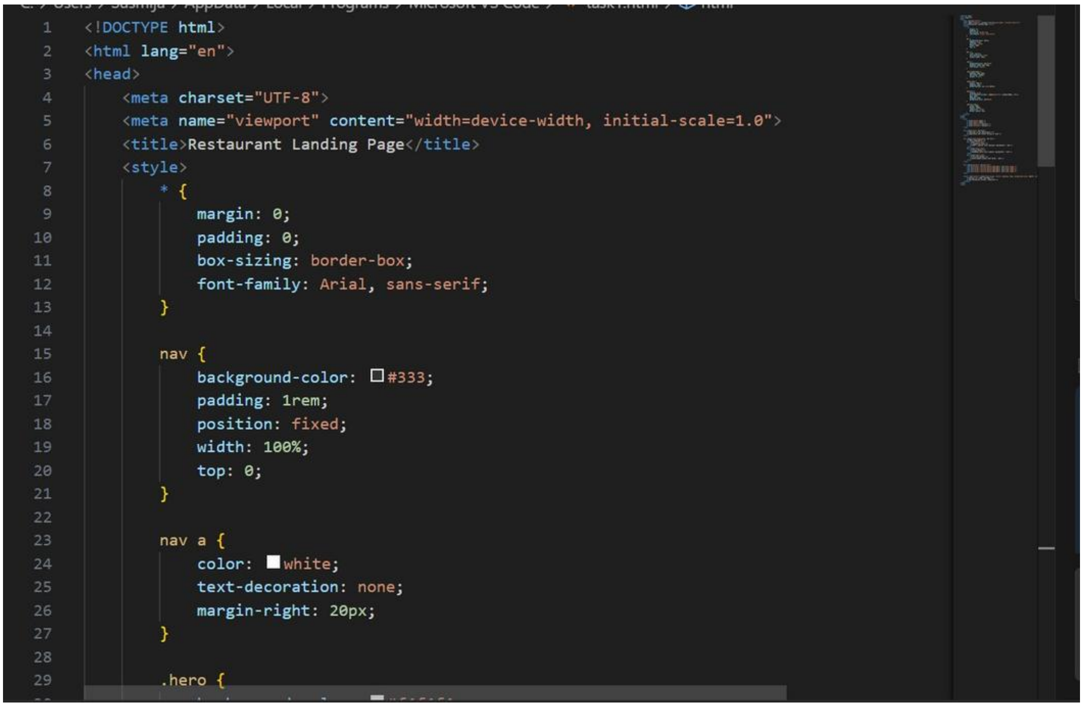
Scenario:

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.

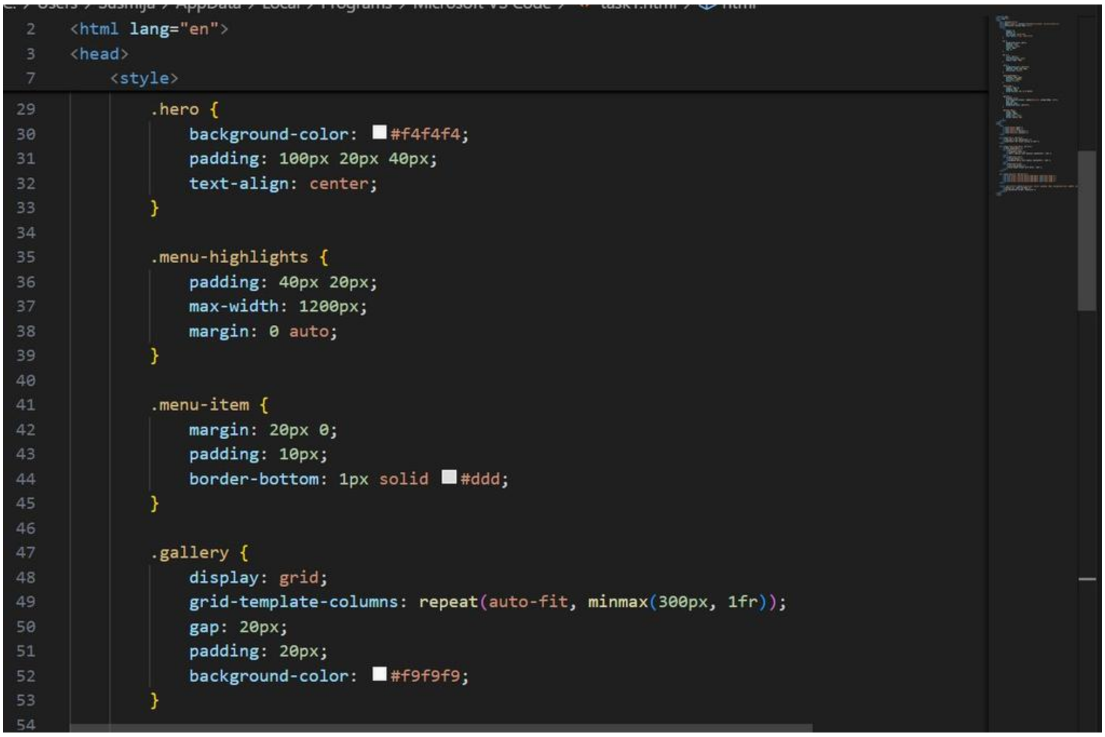
Prompt:

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

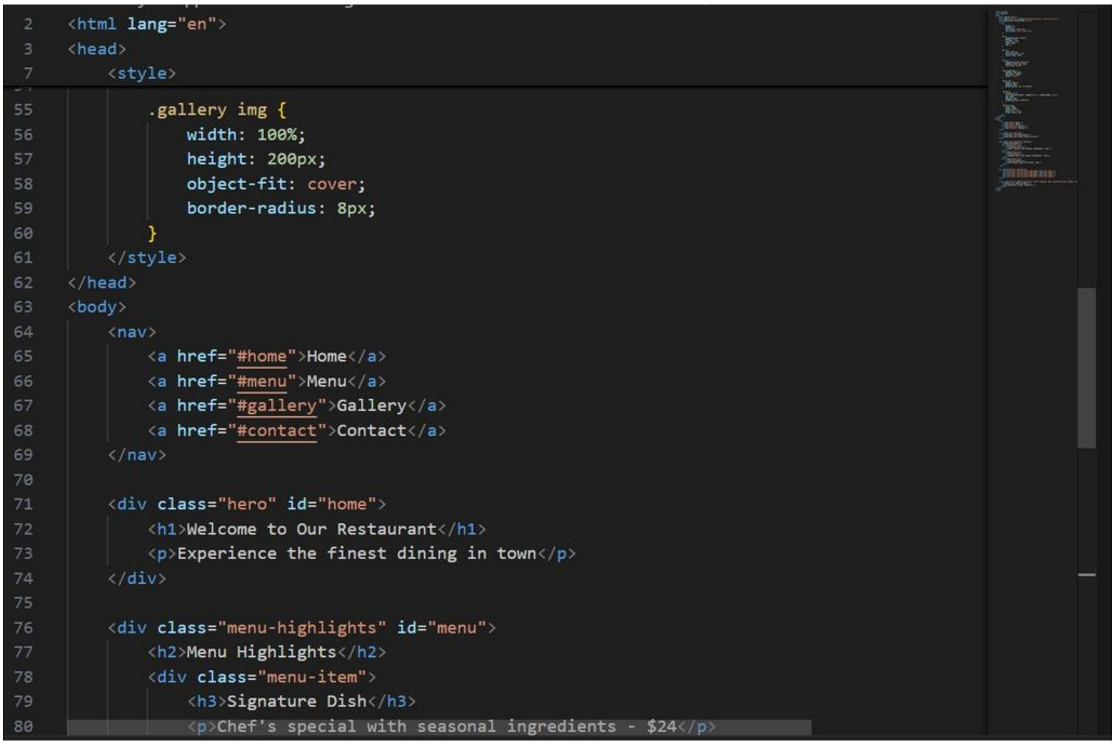
Code generated:



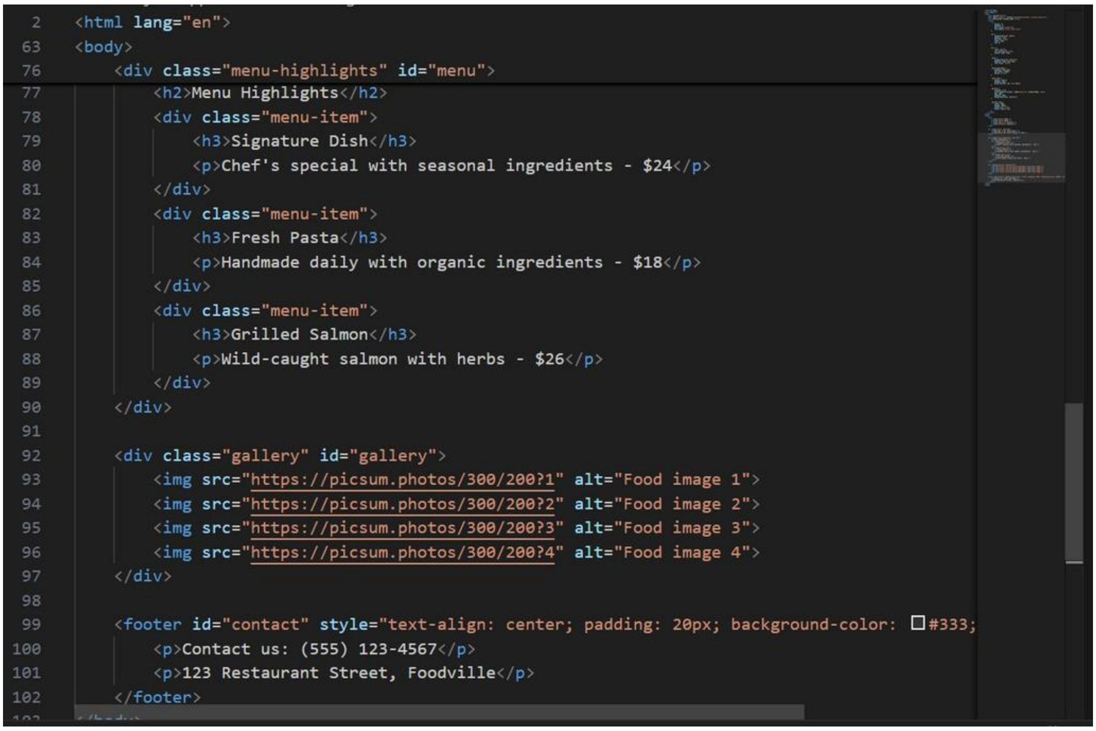
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Restaurant Landing Page</title>
    <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            font-family: Arial, sans-serif;
        }

        nav {
            background-color: #333;
            padding: 1rem;
            position: fixed;
            width: 100%;
            top: 0;
        }

        nav a {
            color: white;
            text-decoration: none;
            margin-right: 20px;
        }

        .hero {
```



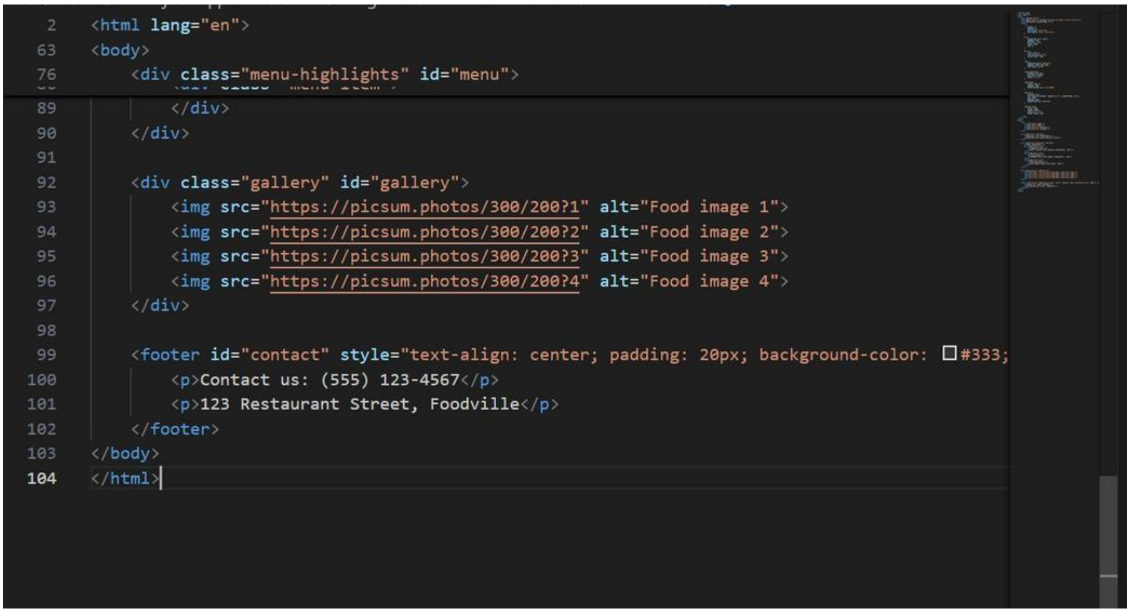
```
<html lang="en">
<head>
    <style>
        .hero {
            background-color: #f4f4f4;
            padding: 100px 20px 40px;
            text-align: center;
        }

        .menu-highlights {
            padding: 40px 20px;
            max-width: 1200px;
            margin: 0 auto;
        }

        .menu-item {
            margin: 20px 0;
            padding: 10px;
            border-bottom: 1px solid #ddd;
        }

        .gallery {
            display: grid;
            grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
            gap: 20px;
            padding: 20px;
            background-color: #f9f9f9;
        }
```



```
<html lang="en">
<head>
    <style>
        .gallery img {
            width: 100%;
            height: 200px;
            object-fit: cover;
            border-radius: 8px;
        }
    </style>
</head>
<body>
    <nav>
        <a href="#home">Home</a>
        <a href="#menu">Menu</a>
        <a href="#gallery">Gallery</a>
        <a href="#contact">Contact</a>
    </nav>

    <div class="hero" id="home">
        <h1>Welcome to Our Restaurant</h1>
        <p>Experience the finest dining in town</p>
    </div>

    <div class="menu-highlights" id="menu">
        <h2>Menu Highlights</h2>
        <div class="menu-item">
            <h3>Signature Dish</h3>
            <p>Chef's special with seasonal ingredients - $24</p>
```

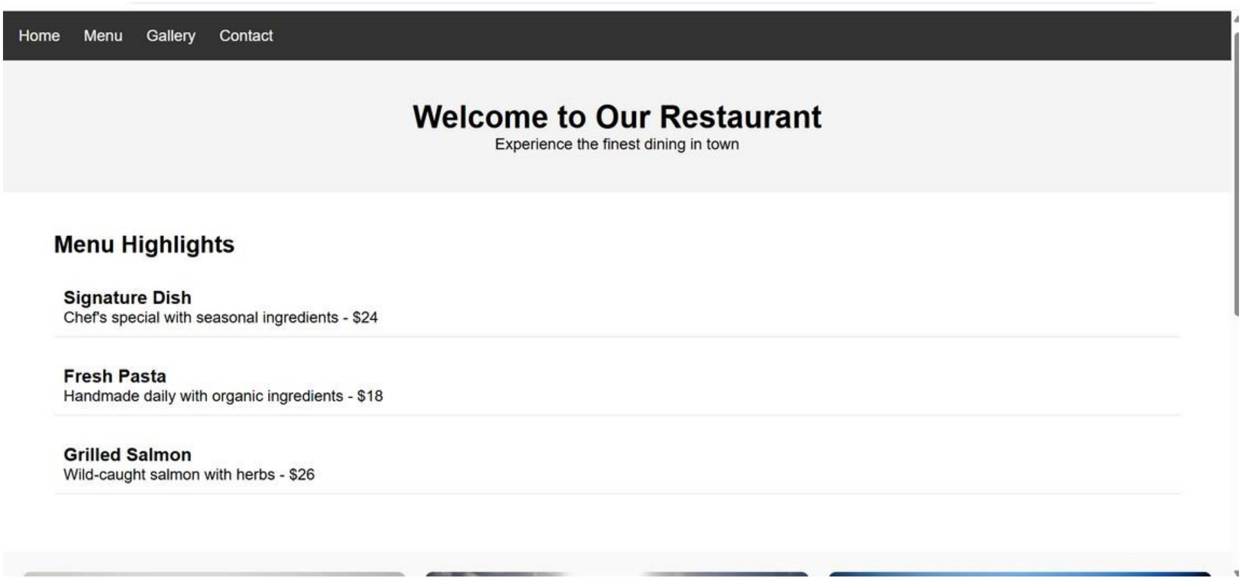


```
<html lang="en">
<body>
    <div class="menu-highlights" id="menu">
        <h2>Menu Highlights</h2>
        <div class="menu-item">
            <h3>Signature Dish</h3>
            <p>Chef's special with seasonal ingredients - $24</p>
        </div>
        <div class="menu-item">
            <h3>Fresh Pasta</h3>
            <p>Handmade daily with organic ingredients - $18</p>
        </div>
        <div class="menu-item">
            <h3>Grilled Salmon</h3>
            <p>Wild-caught salmon with herbs - $26</p>
        </div>
    </div>

    <div class="gallery" id="gallery">
        <img src="https://picsum.photos/300/200?1" alt="Food image 1">
        <img src="https://picsum.photos/300/200?2" alt="Food image 2">
        <img src="https://picsum.photos/300/200?3" alt="Food image 3">
        <img src="https://picsum.photos/300/200?4" alt="Food image 4">
    </div>

    <footer id="contact" style="text-align: center; padding: 20px; background-color: ■#333;
        <p>Contact us: (555) 123-4567</p>
        <p>123 Restaurant Street, Foodville</p>
    </footer>
```



```
<html lang="en">
<body>
    <div class="menu-highlights" id="menu">
        </div>
    </div>

    <div class="gallery" id="gallery">
        <img src="https://picsum.photos/300/200?1" alt="Food image 1">
        <img src="https://picsum.photos/300/200?2" alt="Food image 2">
        <img src="https://picsum.photos/300/200?3" alt="Food image 3">
        <img src="https://picsum.photos/300/200?4" alt="Food image 4">
    </div>

    <footer id="contact" style="text-align: center; padding: 20px; background-color: □#333;
        <p>Contact us: (555) 123-4567</p>
        <p>123 Restaurant Street, Foodville</p>
    </footer>
</body>
</html>
```

Output:



Welcome to Our Restaurant

Experience the finest dining in town

Menu Highlights

Signature Dish

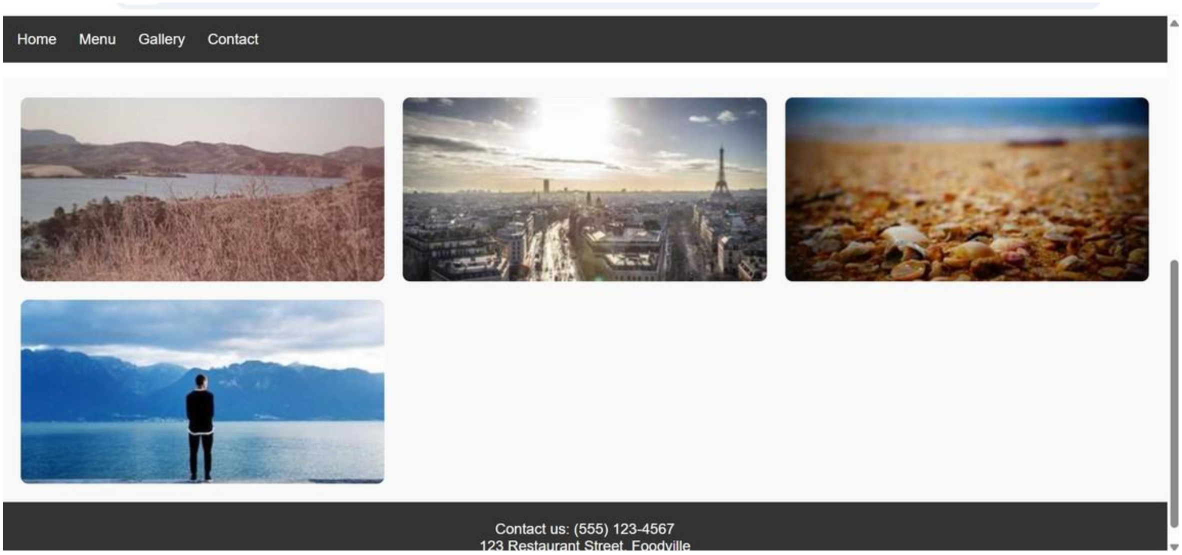
Chef's special with seasonal ingredients - $24

Fresh Pasta

Handmade daily with organic ingredients - $18

Grilled Salmon

Wild-caught salmon with herbs - $26



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:

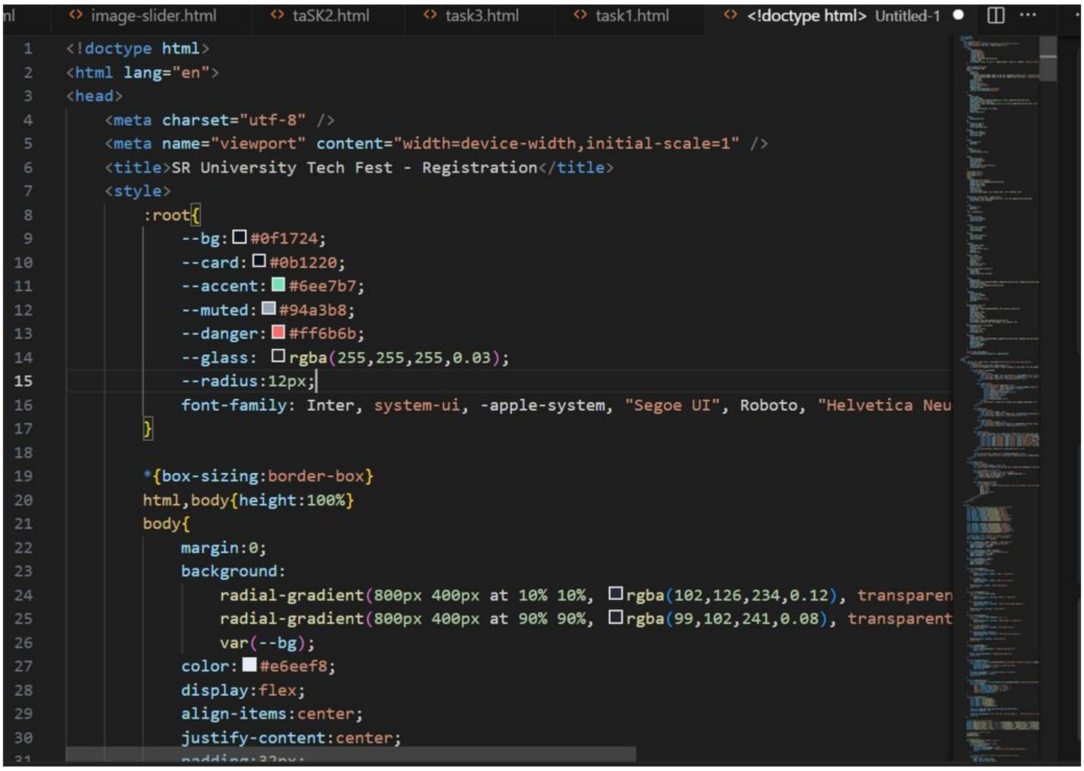
SR University is hosting a tech fest. They need a web-based registration form for students. The form must validate user input in real-time.

Prompt:

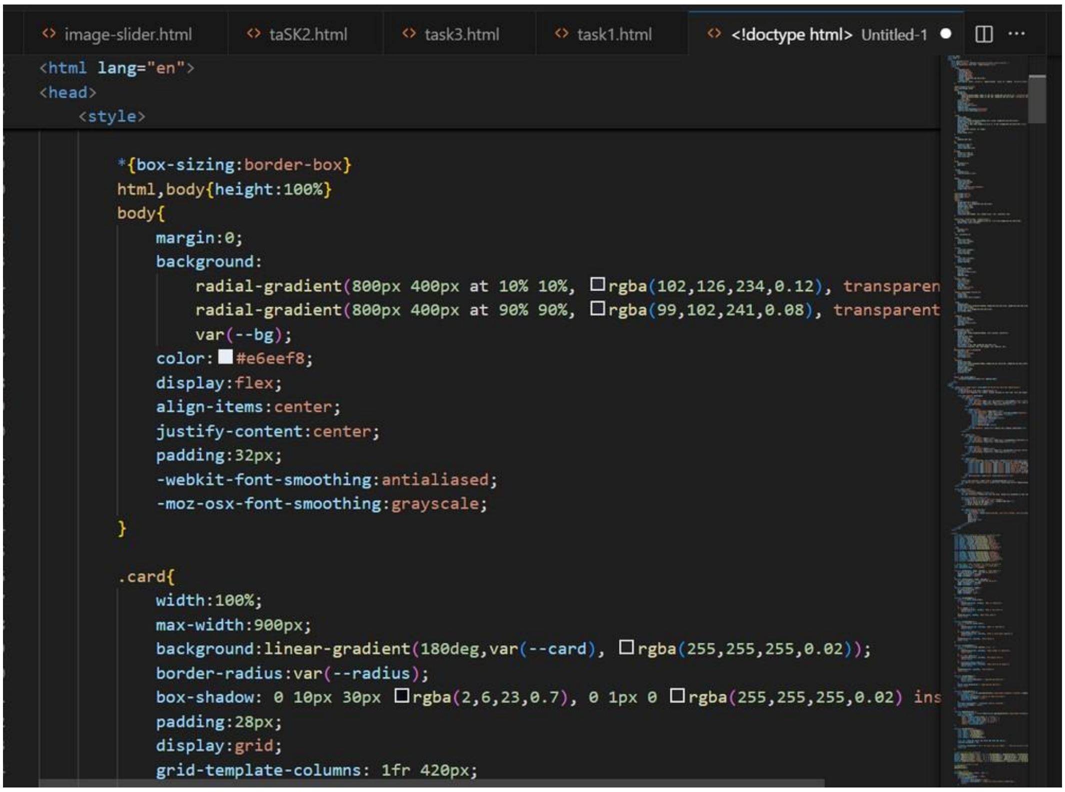
SR University is hosting a tech fest. They need a web-basedregistration form for students. The form must validate user input in real- time.generate an HTML form (fields: Name, Email, Phone, Department, Event Selection).Assist in adding CSS styling for an attractive form layout.

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

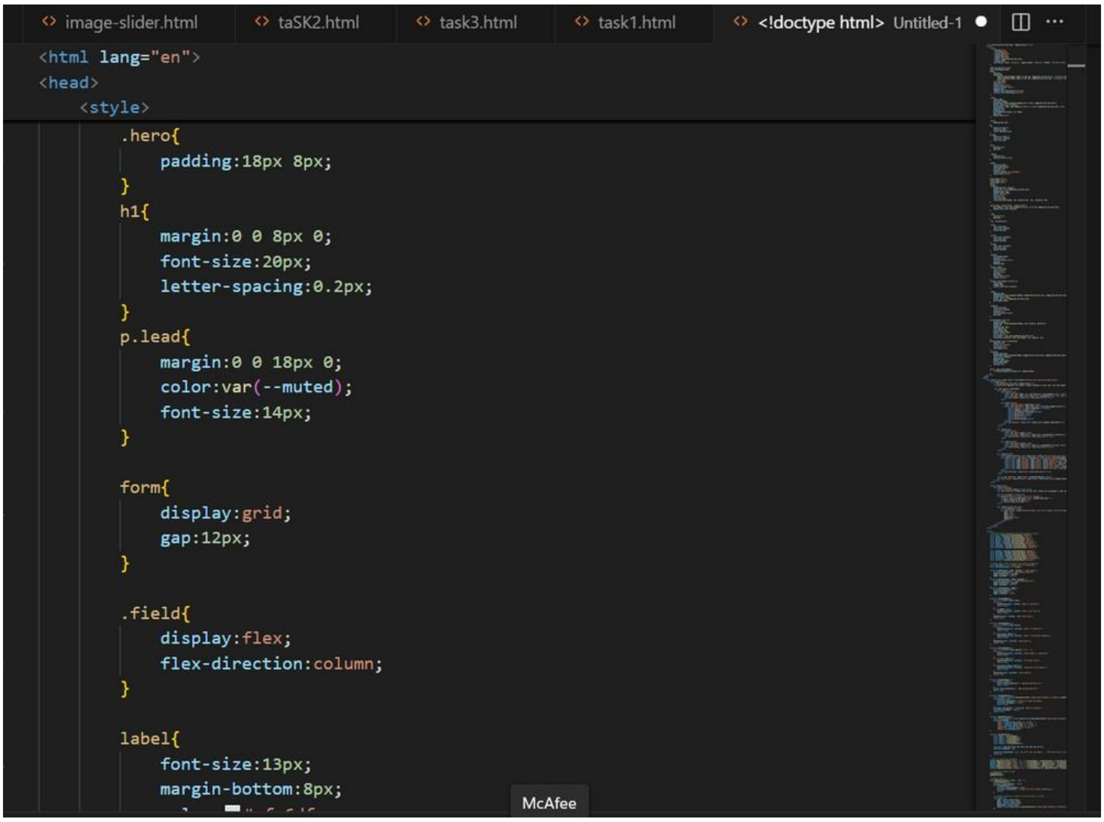
Code generated:



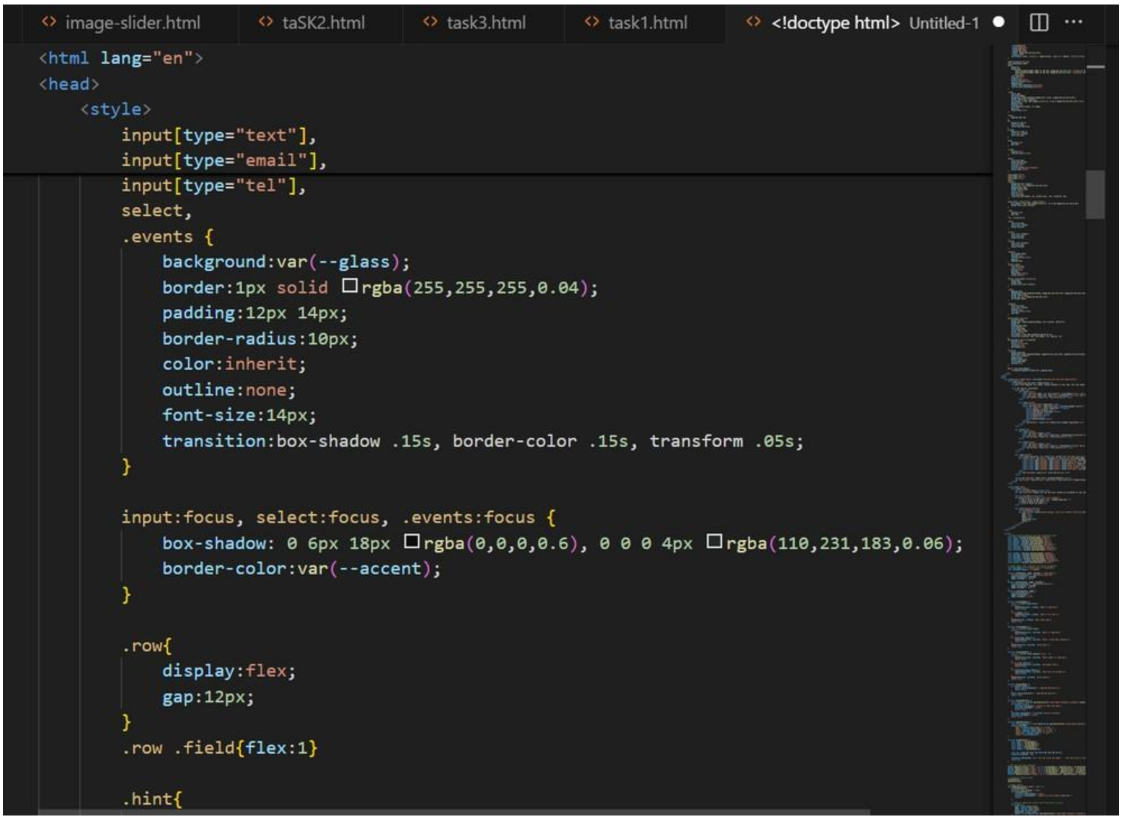
```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width,initial-scale=1" />
    <title>SR University Tech Fest - Registration</title>
    <style>
        :root{
            --bg:#0f1724;
            --card:#0b1220;
            --accent:#6ee7b7;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

        *{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), transparen
                radial-gradient(800px 400px at 90% 90%, rgba(99,102,241,0.08), transparent
                var(--bg);
            color:#e6eef8;
            display:flex;
            align-items:center;
            justify-content:center;
```



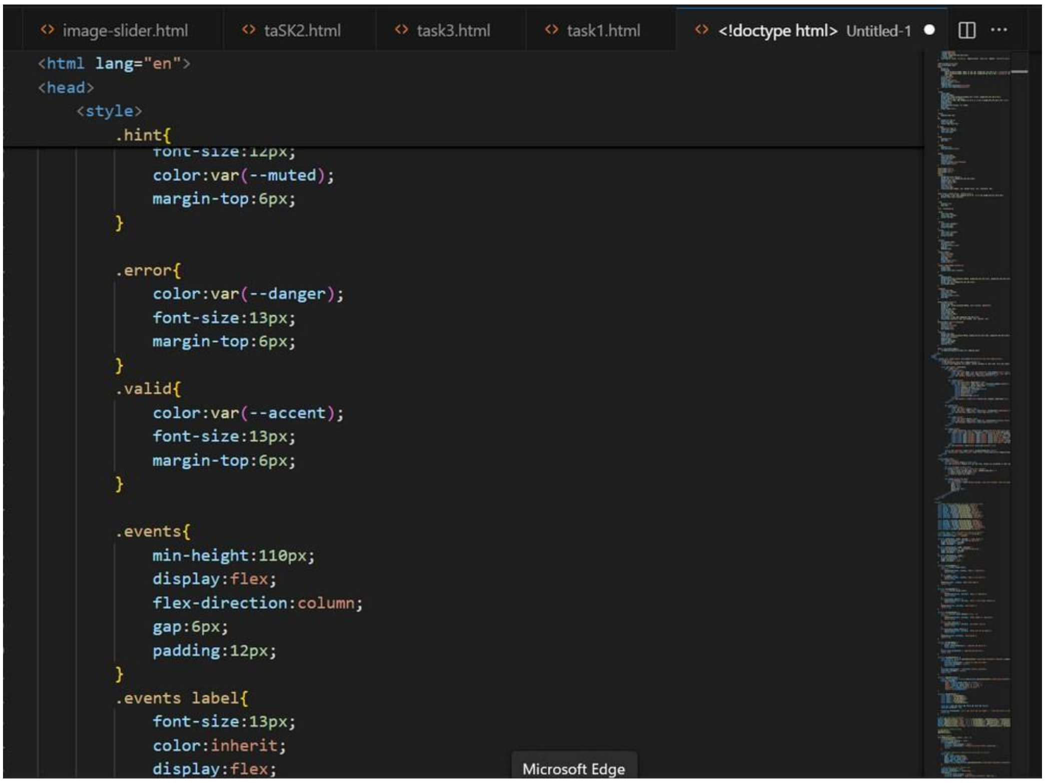
```
<html lang="en">
<head>
    <style>

        *{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), transparen
                radial-gradient(800px 400px at 90% 90%, rgba(99,102,241,0.08), transparent
                var(--bg);
            color:#e6eef8;
            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), rgba(255,255,255,0.02));
            border-radius:var(--radius);
            box-shadow: 0 10px 30px rgba(2,6,23,0.7), 0 1px 0 rgba(255,255,255,0.02) ins
            padding:28px;
            display:grid;
            grid-template-columns: 1fr 420px;
```



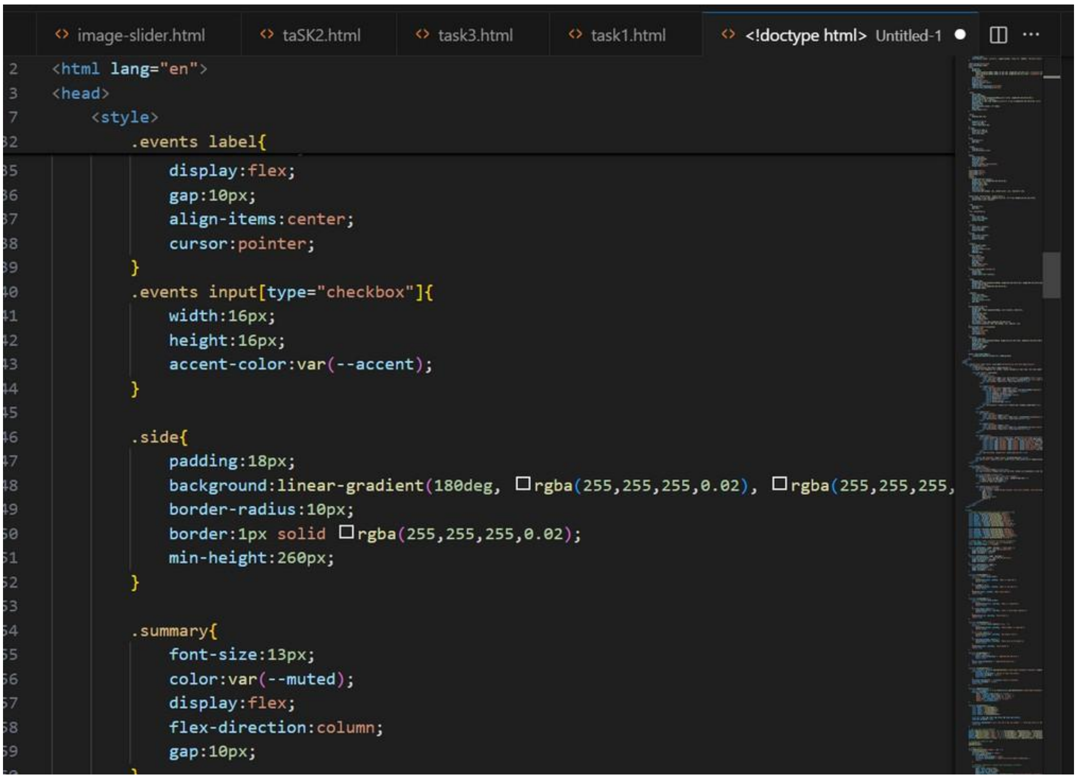
```
<html lang="en">
<head>
    <style>
        .hero{
            padding:18px 8px;
        }
        h1{
            margin:0 0 8px 0;
            font-size:20px;
            letter-spacing:0.2px;
        }
        p.lead{
            margin:0 0 18px 0;
            color:var(--muted);
            font-size:14px;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

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

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

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



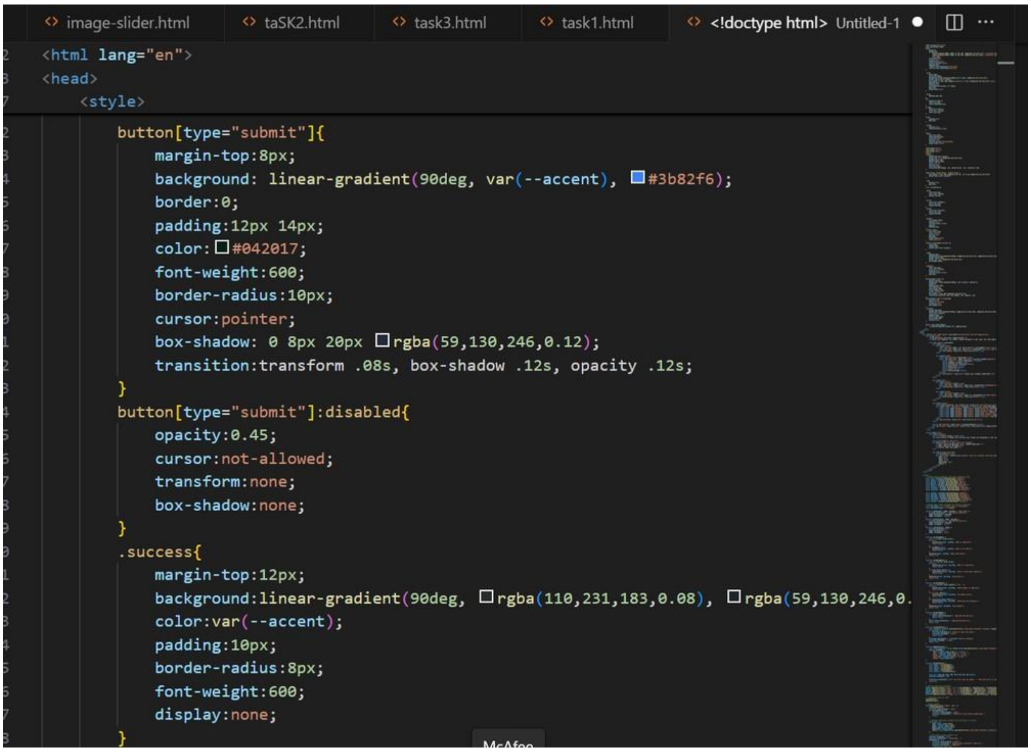
```
<html lang="en">
<head>
    <style>
        input[type="text"],
        input[type="email"],
        input[type="tel"],
        select,
        .events {
            background:var(--glass);
            border:1px solid □rgba(255,255,255,0.04);
            padding:12px 14px;
            border-radius:10px;
            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 0 4px □rgba(110,231,183,0.06);
            border-color:var(--accent);
        }

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

        .hint{
```



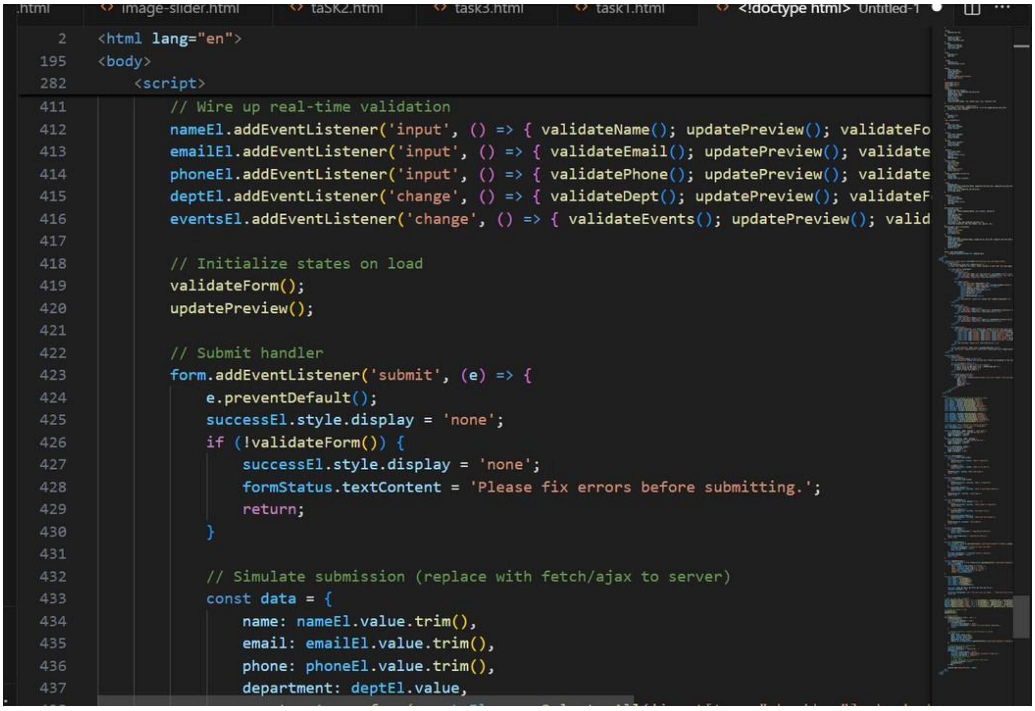
```
<html lang="en">
<head>
    <style>
        .hint{
            font-size:12px;
            color:var(--muted);
            margin-top:6px;
        }

        .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;
```



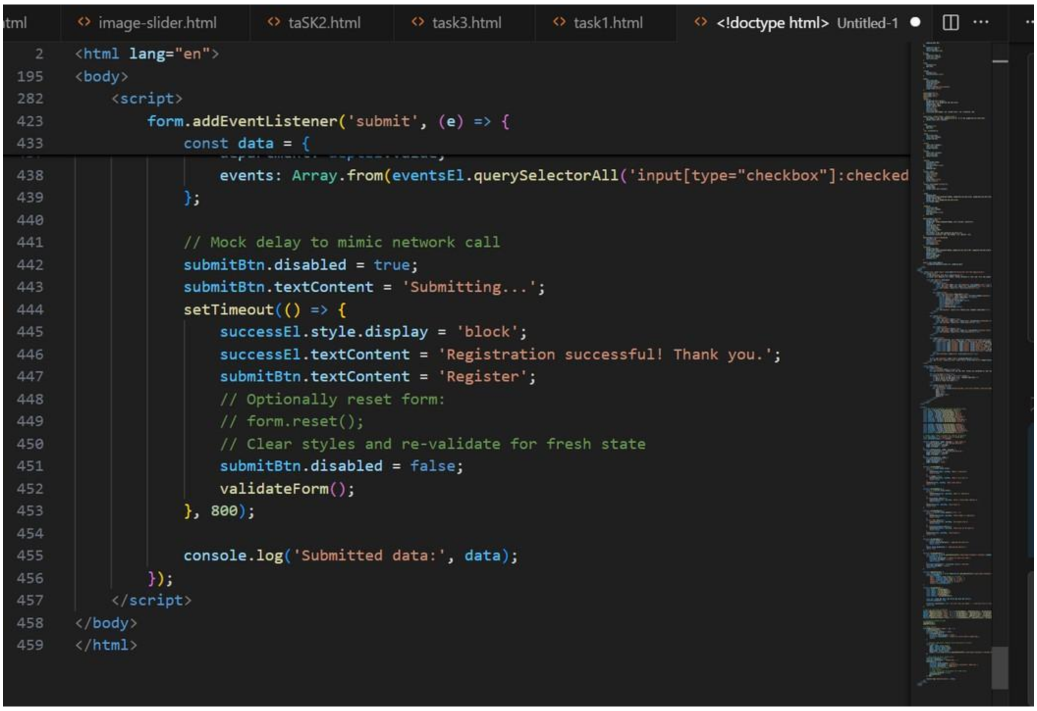
```
<html lang="en">
<head>
    <style>
        .events label{
            display:flex;
            gap:10px;
            align-items:center;
            cursor:pointer;
        }
        .events input[type="checkbox"]{
            width:16px;
            height:16px;
            accent-color:var(--accent);
        }

        .side{
            padding:18px;
            background:linear-gradient(180deg, rgba(255,255,255,0.02), rgba(255,255,255,
            border-radius:10px;
            border:1px solid rgba(255,255,255,0.02);
            min-height:260px;
        }

        .summary{
            font-size:13px;
            color:var(--muted);
            display:flex;
            flex-direction:column;
            gap:10px;
        }
```



```
<html lang="en">
<head>
    <style>
        button[type="submit"]{
            margin-top:8px;
            background: linear-gradient(90deg, var(--accent), #3b82f6);
            border:0;
            padding:12px 14px;
            color:#042017;
            font-weight:600;
            border-radius:10px;
            cursor:pointer;
            box-shadow: 0 8px 20px rgba(59,130,246,0.12);
            transition:transform .08s, box-shadow .12s, opacity .12s;
        }
        button[type="submit"]:disabled{
            opacity:0.45;
            cursor:not-allowed;
            transform:none;
            box-shadow:none;
        }
        .success{
            margin-top:12px;
            background:linear-gradient(90deg, rgba(110,231,183,0.08), rgba(59,130,246,0.
            color:var(--accent);
            padding:10px;
            border-radius:8px;
            font-weight:600;
            display:none;
        }
```

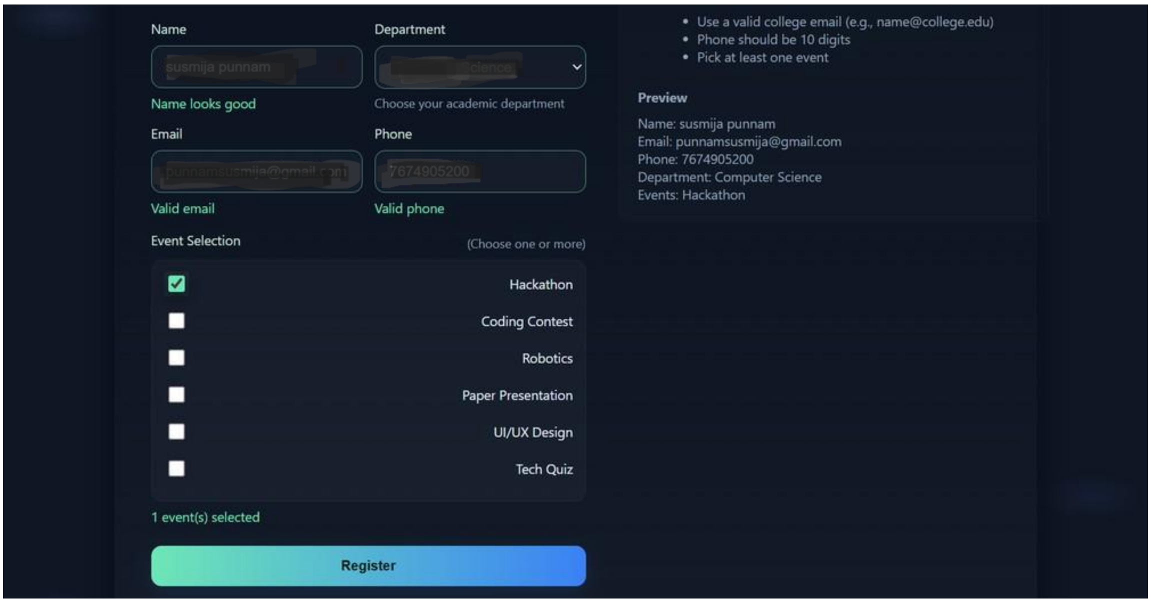


```
<html lang="en">
<body>
    <script>
        // Wire up real-time validation
        nameEl.addEventListener('input', () => { validateName(); updatePreview(); validateFo
        emailEl.addEventListener('input', () => { validateEmail(); updatePreview(); validate
        phoneEl.addEventListener('input', () => { validatePhone(); updatePreview(); validate
        deptEl.addEventListener('change', () => { validateDept(); updatePreview(); validateF
        eventsEl.addEventListener('change', () => { validateEvents(); updatePreview(); valid

        // Initialize states on load
        validateForm();
        updatePreview();

        // Submit handler
        form.addEventListener('submit', (e) => {
            e.preventDefault();
            successEl.style.display = 'none';
            if (!validateForm()) {
                successEl.style.display = 'none';
                formStatus.textContent = 'Please fix errors before submitting.';
                return;
            }

            // Simulate submission (replace with fetch/ajax to server)
            const data = {
                name: nameEl.value.trim(),
                email: emailEl.value.trim(),
                phone: phoneEl.value.trim(),
                department: deptEl.value,
```



```
<html lang="en">
<body>
    <script>
        form.addEventListener('submit', (e) => {
            const data = {
                events: Array.from(eventsEl.querySelectorAll('input[type="checkbox"]:checked
            };

            // Mock delay to mimic network call
            submitBtn.disabled = true;
            submitBtn.textContent = 'Submitting...';
            setTimeout(() => {
                successEl.style.display = 'block';
                successEl.textContent = 'Registration successful! Thank you.';
                submitBtn.textContent = 'Register';
                // Optionally reset form:
                // form.reset();
                // Clear styles and re-validate for fresh state
                submitBtn.disabled = false;
                validateForm();
            }, 800);

            console.log('Submitted data:', data);
        });
    </script>
</body>
</html>
```

Output:



Name: [redacted]

Name looks good

Department: [redacted]

Choose your academic department

Email: [redacted]

Valid email

Phone: [redacted]

Valid phone

Event Selection (Choose one or more)

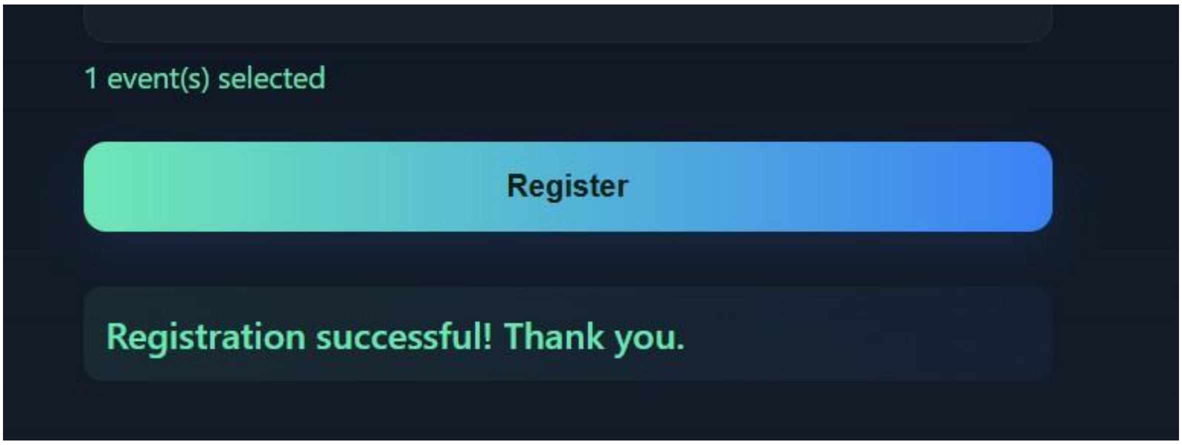
☑ Hackathon
☐ Coding Contest
☐ Robotics
☐ Paper Presentation
☐ UI/UX Design
☐ Tech Quiz

1 event(s) selected

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

Preview

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



1 event(s) selected

Registration successful! Thank you.

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.

<u>Task#4</u>:

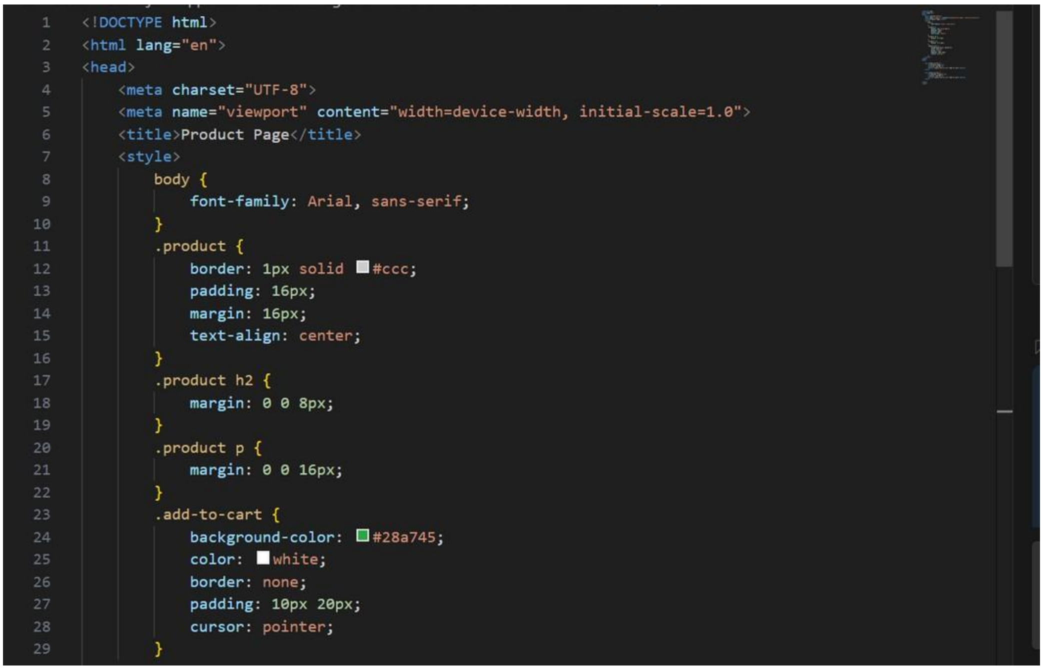
AI-Assisted E-Commerce Product Page

Prompt:

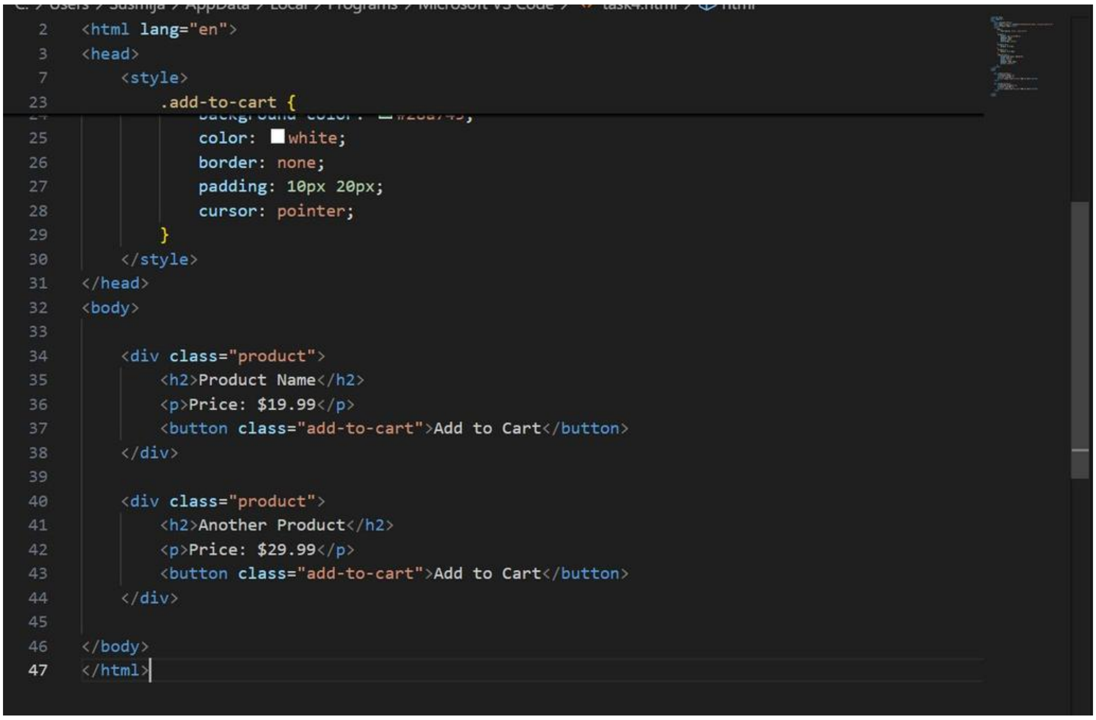
A startup wants a basic e-commerce product page to display products with prices and an “Add to Cart” button.

- 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:

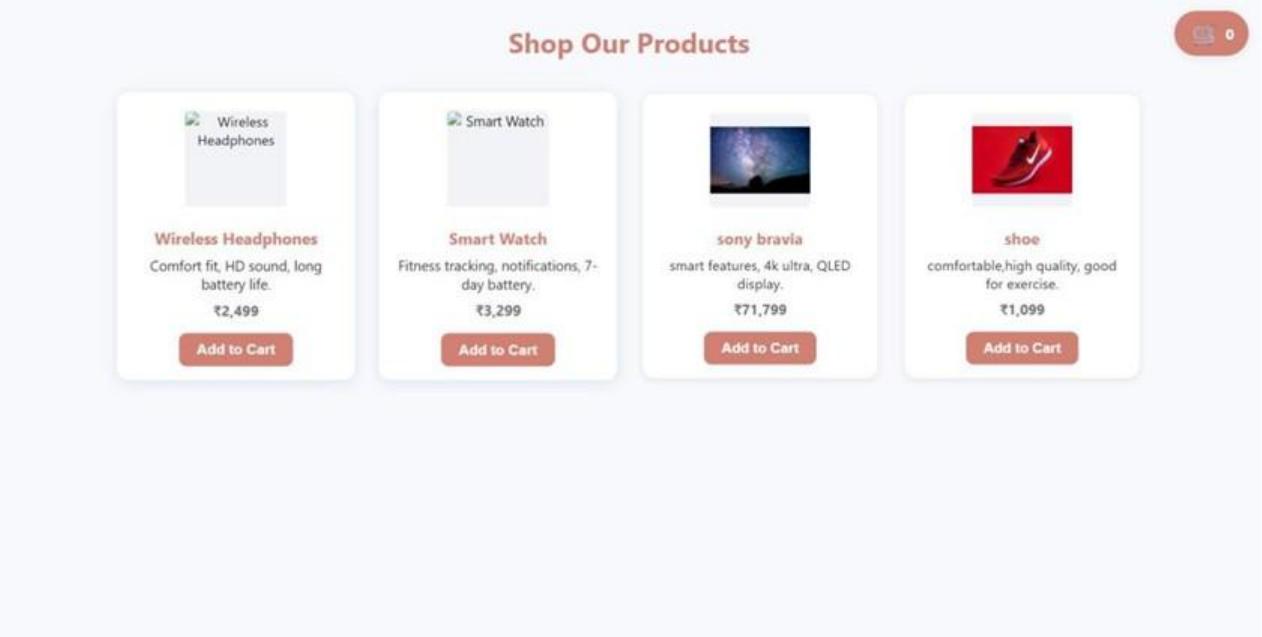


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Product Page</title>
    <style>
        body {
            font-family: Arial, sans-serif;
        }
        .product {
            border: 1px solid #ccc;
            padding: 16px;
            margin: 16px;
            text-align: center;
        }
        .product h2 {
            margin: 0 0 8px;
        }
        .product p {
            margin: 0 0 16px;
        }
        .add-to-cart {
            background-color: #28a745;
            color: white;
            border: none;
            padding: 10px 20px;
            cursor: pointer;
        }
```



```html
<html lang="en">
<head>
    <style>
        .add-to-cart {
            color: white;
            border: none;
            padding: 10px 20px;
            cursor: pointer;
        }
    </style>
</head>
<body>

    <div class="product">
        <h2>Product Name</h2>
        <p>Price: $19.99</p>
        <button class="add-to-cart">Add to Cart</button>
    </div>

    <div class="product">
        <h2>Another Product</h2>
        <p>Price: $29.99</p>
        <button class="add-to-cart">Add to Cart</button>
    </div>

</body>
</html>
```

OUTPUT:



Shop Our Products

Wireless Headphones
Comfort fit, HD sound, long battery life.
₹2,499

Smart Watch
Fitness tracking, notifications, 7-day battery.
₹3,299

sony bravia
smart features, 4k ultra, QLED display.
₹71,799

shoe
comfortable,high quality, good for exercise.
₹1,099

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.