ASSIGNMENT – 14

Hall ticket no : 2403a52423

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 :   
A student wants to showcase their projects, skills, and contact details in  
a portfolio website.   
• Generate an HTML structure for a personal  
portfolio page (sections: Home, About, Projects, Contact).  
• Add responsive CSS styling for the layout  
(e.g., grid/flexbox).  
• Add a hover effect on project  
cards

CODE :

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>Student Portfolio</title>

<style>

/\* Basic Reset \*/

\* {

box-sizing: border-box;

margin: 0;

padding: 0;

}

body {

font-family: Arial, sans-serif;

margin: 0;

background-color: #f4f4f4;

color: #333;

}

/\* Header Navigation \*/

header {

background-color: #333;

padding: 1rem;

}

nav ul {

display: flex;

justify-content: center;

list-style: none;

}

nav ul li {

margin: 0 1rem;

}

nav ul li a {

color: #fff;

text-decoration: none;

font-weight: bold;

}

nav ul li a:hover {

text-decoration: underline;

}

/\* Sections \*/

section {

max-width: 1000px;

margin: 2rem auto;

padding: 1rem;

background: #fff;

border-radius: 8px;

box-shadow: 0 2px 8px rgba(0,0,0,0.1);

}

/\* Home \*/

#home {

text-align: center;

}

/\* About \*/

#about {

line-height: 1.6;

}

/\* Projects \*/

#projects {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));

gap: 1.5rem;

}

.project-card {

background-color: #e0f7fa;

padding: 1rem;

border-radius: 8px;

transition: transform 0.3s, box-shadow 0.3s;

}

.project-card:hover {

transform: scale(1.05);

box-shadow: 0 4px 12px rgba(0,0,0,0.2);

cursor: pointer;

}

.project-card h3 {

margin-bottom: 0.5rem;

color: #00796b;

}

/\* Contact \*/

#contact form {

display: flex;

flex-direction: column;

}

#contact input, #contact textarea {

margin-bottom: 1rem;

padding: 0.5rem;

border: 1px solid #ccc;

border-radius: 4px;

}

#contact button {

padding: 0.75rem;

background-color: #00796b;

color: #fff;

border: none;

border-radius: 4px;

cursor: pointer;

}

#contact button:hover {

background-color: #004d40;

}

</style>

</head>

<body>

<header>

<nav>

<ul>

<li><a href="#home">Home</a></li>

<li><a href="#about">About</a></li>

<li><a href="#projects">Projects</a></li>

<li><a href="#contact">Contact</a></li>

</ul>

</nav>

</header>

<section id="home">

<h1>Hello! I am a Student</h1>

<p>Welcome to my portfolio website.</p>

</section>

<section id="about">

<h2>About Me</h2>

<p>I am passionate about web development, coding, and creating engaging user experiences. I love learning new technologies and building projects.</p>

</section>

<section id="projects">

<h2>My Projects</h2>

<div class="projects-grid">

<div class="project-card">

<h3>Project One</h3>

<p>This project is a personal blog website built with HTML, CSS, and JavaScript.</p>

</div>

<div class="project-card">

<h3>Project Two</h3>

<p>An e-commerce site with a responsive layout and real-time features.</p>

</div>

<div class="project-card">

<h3>Project Three</h3>

<p>A portfolio website to showcase various personal projects.</p>

</div>

</div>

</section>

<section id="contact">

<h2>Contact Me</h2>

<form>

<input type="text" placeholder="Your Name" required />

<input type="email" placeholder="Your Email" required />

<textarea rows="4" placeholder="Your Message" required></textarea>

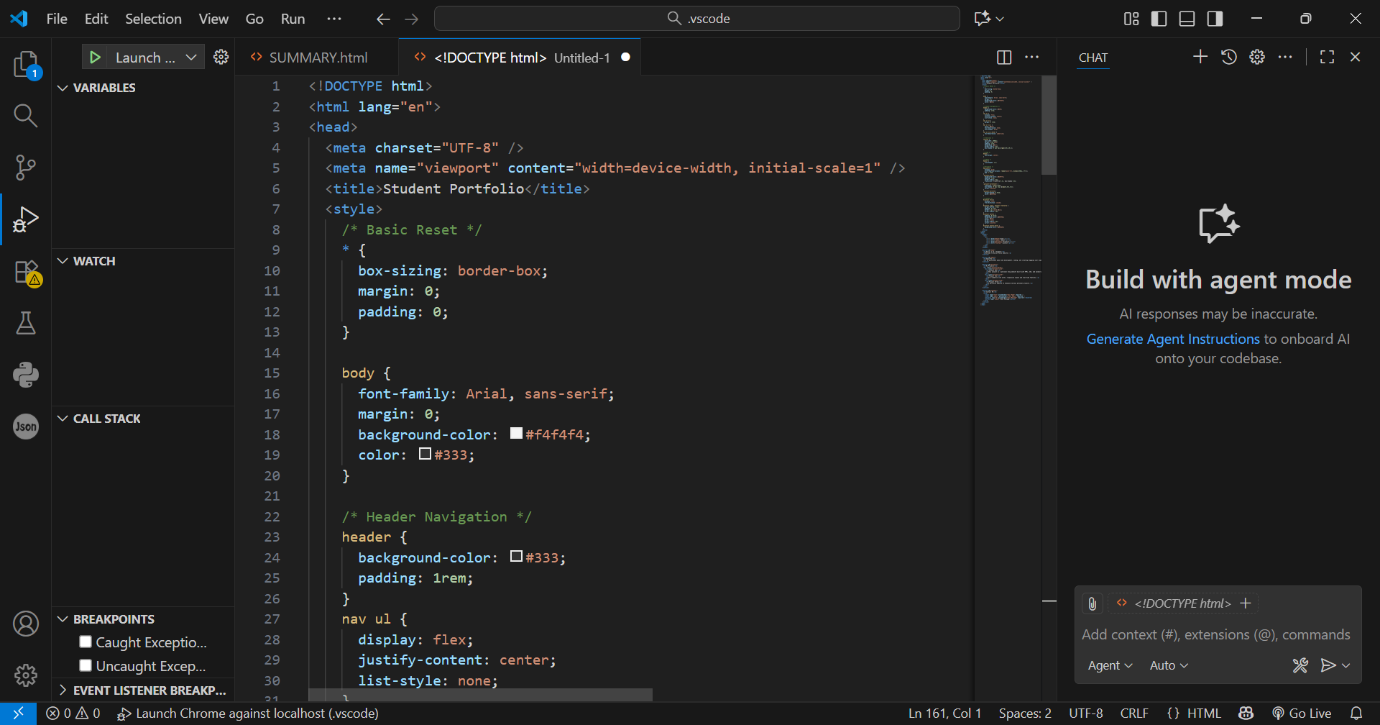
<button type="submit">Send Message</button>

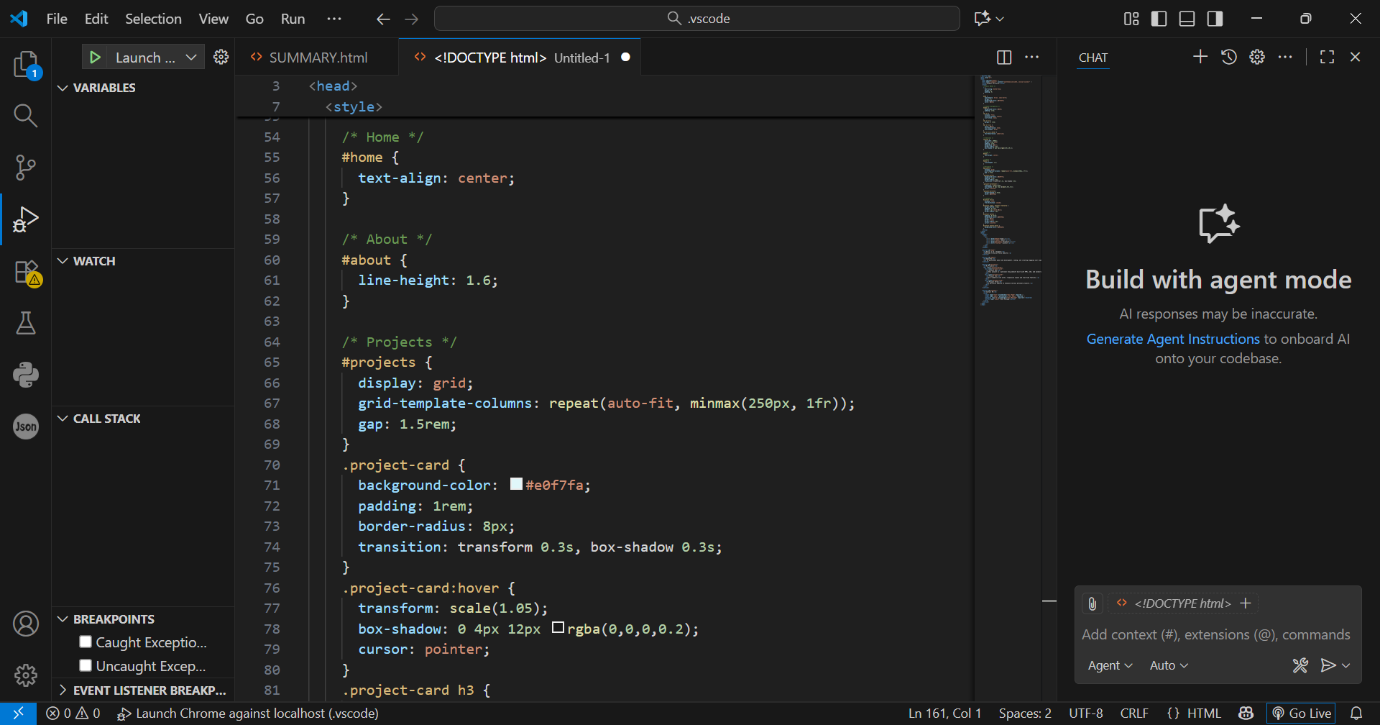
</form>

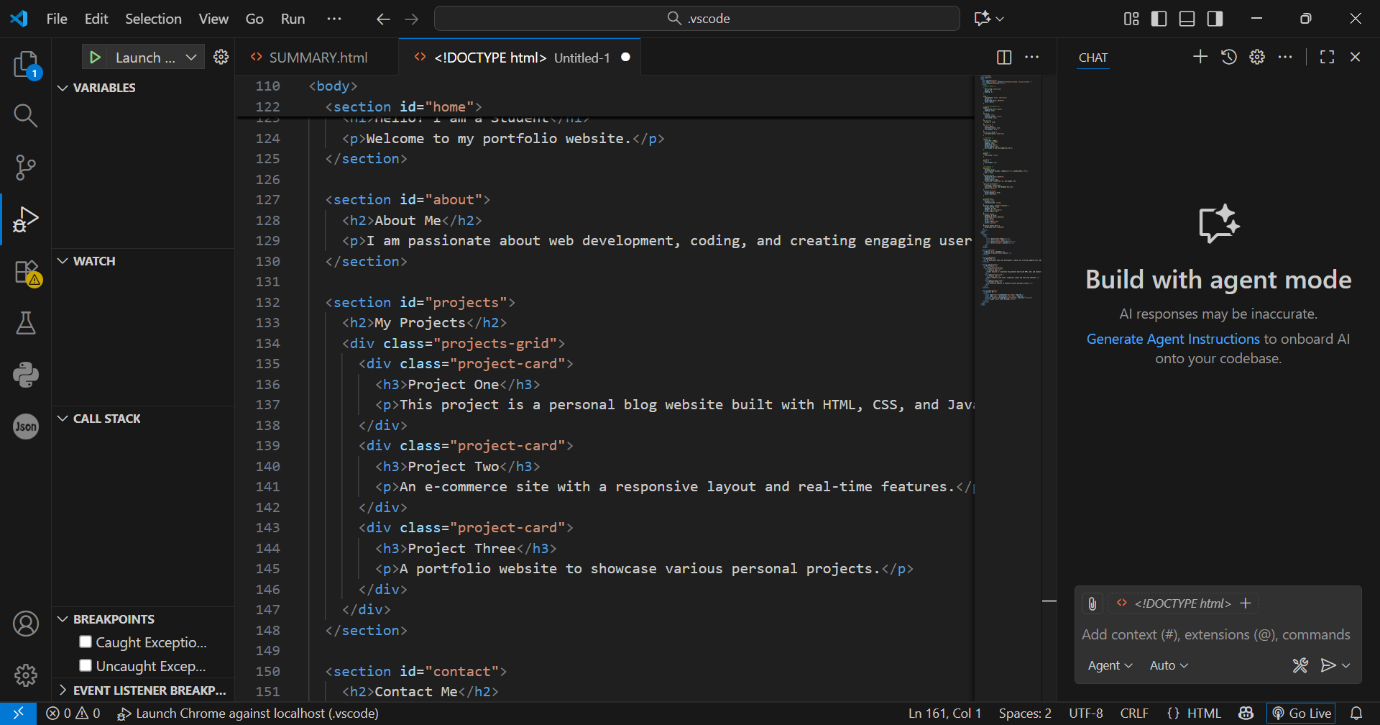
</section>

</body>

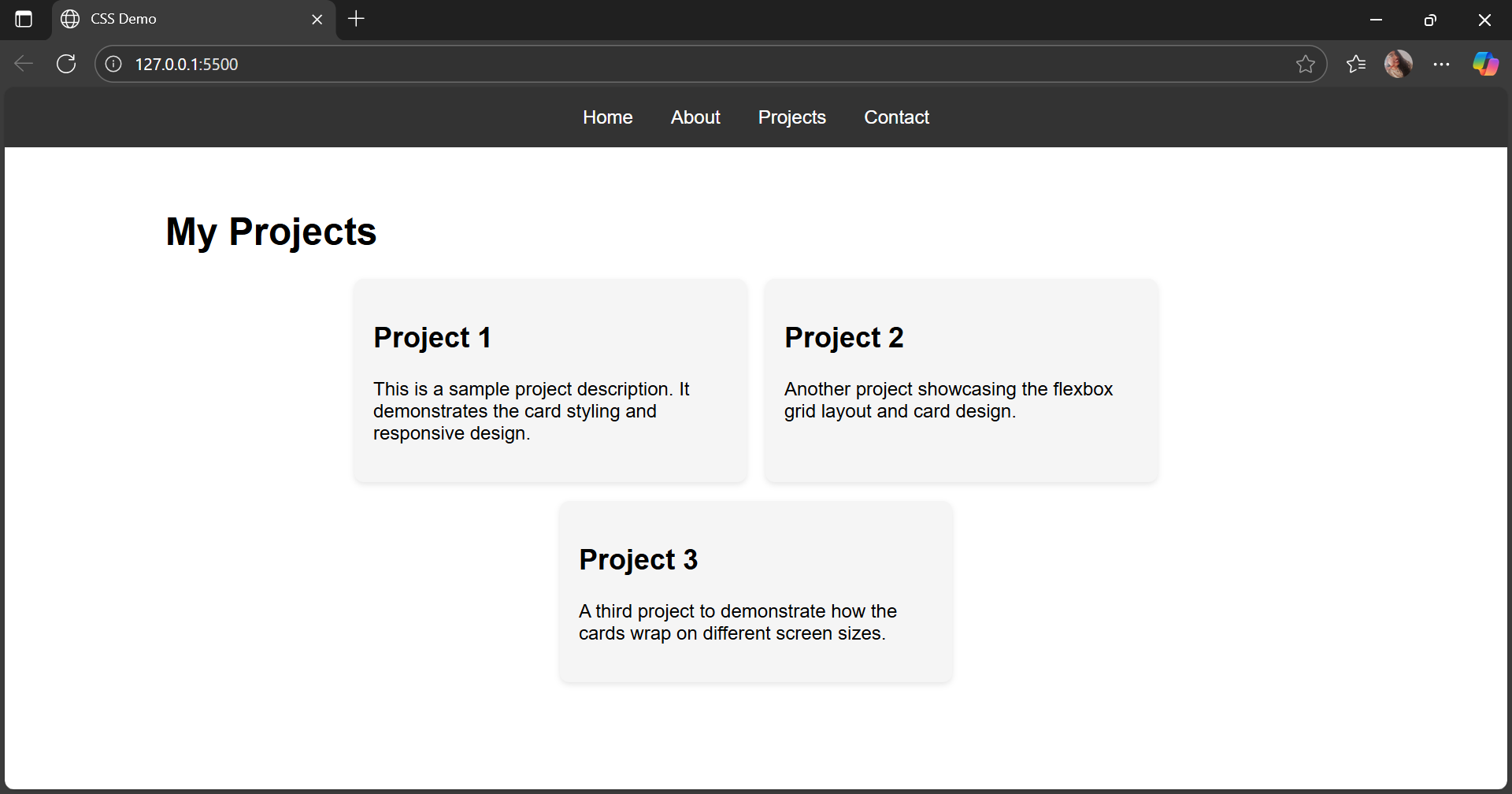
</html>







OUTPUT :



* OBSERVATION : The student needs a personal portfolio website to display projects, skills, and contact information.
* Instead of coding everything manually, the student uses GitHub Copilot to speed up development.
* Copilot is used to generate the basic HTML structure with Home, About, Projects, and Contact sections.
* Copilot suggests responsive CSS using modern layouts like grid or flexbox to ensure the site works on all devices.
* The student customizes Copilot’s code to add a hover effect on project cards, improving interactivity and design.
* This approach allows the student to focus on content and presentation while saving time on coding basics.

Task 2: AI-Generated Restaurant Landing Page  
Scenario:  
A local restaurant needs a simple landing page with a navigation bar,  
menu highlights, and an image gallery. The developer wants to quickly  
generate it using AI assistance.  
• Use Copilot to create a navigation bar with links (Home, Menu,  
Gallery, Contact).  
• Generate a menu section styled with CSS cards.  
• Add a JavaScript-based image slider for the gallery, with  
Copilot suggesting the base code

PROMPT :   
A local restaurant needs a simple landing page with a navigation bar,  
menu highlights, and an image gallery. The developer wants to quickly  
generate it using AI assistance.  
•Create a navigation bar with links (Home, Menu,  
Gallery, Contact).  
• Generate a menu section styled with CSS cards.  
• Add a JavaScript-based image slider for the gallery, with  
Copilot suggesting the base code

CODE : <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>Local Restaurant Landing Page</title>

<style>

/\* Basic Reset \*/

\* {

box-sizing: border-box;

margin: 0;

padding: 0;

}

body {

font-family: Arial, sans-serif;

background: #fff8f0;

color: #333;

}

/\* Navigation Bar \*/

header {

background-color: #d35400;

padding: 1rem 0;

box-shadow: 0 2px 5px rgba(0,0,0,0.1);

}

nav ul {

list-style: none;

display: flex;

justify-content: center;

gap: 2rem;

}

nav ul li a {

color: white;

text-decoration: none;

font-weight: bold;

font-size: 1.2rem;

transition: color 0.3s ease;

}

nav ul li a:hover {

color: #f39c12;

}

/\* Menu Section \*/

#menu {

max-width: 1000px;

margin: 2rem auto;

padding: 0 1rem;

}

#menu h2 {

text-align: center;

margin-bottom: 1.5rem;

color: #d35400;

}

.menu-cards {

display: grid;

grid-template-columns: repeat(auto-fill,minmax(250px,1fr));

gap: 1.5rem;

}

.menu-card {

background: white;

border-radius: 10px;

padding: 1rem;

box-shadow: 0 3px 10px rgba(0,0,0,0.1);

transition: transform 0.3s ease, box-shadow 0.3s ease;

}

.menu-card:hover {

transform: scale(1.05);

box-shadow: 0 6px 15px rgba(0,0,0,0.2);

cursor: pointer;

}

.menu-card h3 {

color: #e67e22;

margin-bottom: 0.5rem;

}

.menu-card p {

font-size: 0.9rem;

color: #555;

}

.menu-card .price {

margin-top: 1rem;

font-weight: bold;

color: #d35400;

}

/\* Gallery Section \*/

#gallery {

max-width: 800px;

margin: 3rem auto;

padding: 0 1rem;

text-align: center;

}

#gallery h2 {

color: #d35400;

margin-bottom: 1rem;

}

.slider {

position: relative;

overflow: hidden;

border-radius: 10px;

box-shadow: 0 2px 10px rgba(0,0,0,0.1);

}

.slides {

display: flex;

transition: transform 0.5s ease-in-out;

width: 100%;

}

.slides img {

width: 100%;

flex-shrink: 0;

}

/\* Slider Navigation Buttons \*/

.slider-btn {

position: absolute;

top: 50%;

transform: translateY(-50%);

background-color: rgba(211, 84, 0, 0.7);

border: none;

color: white;

font-size: 2rem;

padding: 0 0.5rem;

cursor: pointer;

border-radius: 3px;

user-select: none;

transition: background-color 0.3s ease;

}

.slider-btn:hover {

background-color: rgba(211, 84, 0, 0.9);

}

#prevBtn {

left: 10px;

}

#nextBtn {

right: 10px;

}

/\* Contact Section \*/

#contact {

max-width: 600px;

margin: 3rem auto 2rem auto;

padding: 0 1rem;

text-align: center;

}

#contact h2 {

color: #d35400;

margin-bottom: 1rem;

}

#contact p {

font-size: 1rem;

color: #555;

}

</style>

</head>

<body>

<header>

<nav>

<ul>

<li><a href="#home">Home</a></li>

<li><a href="#menu">Menu</a></li>

<li><a href="#gallery">Gallery</a></li>

<li><a href="#contact">Contact</a></li>

</ul>

</nav>

</header>

<section id="home" style="padding: 2rem 1rem; max-width: 800px; margin: auto; text-align:center;">

<h1>Welcome to Our Local Restaurant</h1>

<p>Enjoy our specially curated menu and delightful ambiance.</p>

</section>

<section id="menu">

<h2>Menu Highlights</h2>

<div class="menu-cards">

<div class="menu-card">

<h3>Grilled Chicken</h3>

<p>Succulent grilled chicken served with seasonal vegetables.</p>

<div class="price">$12.99</div>

</div>

<div class="menu-card">

<h3>Vegan Salad Bowl</h3>

<p>Fresh mixed greens, nuts, and a tangy vinaigrette dressing.</p>

<div class="price">$9.99</div>

</div>

<div class="menu-card">

<h3>Seafood Pasta</h3>

<p>Delicious pasta with shrimp, clams, and creamy sauce.</p>

<div class="price">$15.99</div>

</div>

</div>

</section>

<section id="gallery">

<h2>Gallery</h2>

<div class="slider">

<div class="slides">

<img src="https://via.placeholder.com/800x400?text=Dish+1" alt="Dish 1" />

<img src="https://via.placeholder.com/800x400?text=Dish+2" alt="Dish 2" />

<img src="https://via.placeholder.com/800x400?text=Dish+3" alt="Dish 3" />

</div>

<button class="slider-btn" id="prevBtn">&#10094;</button>

<button class="slider-btn" id="nextBtn">&#10095;</button>

</div>

</section>

<section id="contact">

<h2>Contact</h2>

<p>Phone: (123) 456-7890</p>

<p>Email: reservation@localrestaurant.com</p>

</section>

<script>

// JavaScript-based image slider

const slides = document.querySelector('.slides');

const images = document.querySelectorAll('.slides img');

const prevBtn = document.getElementById('prevBtn');

const nextBtn = document.getElementById('nextBtn');

let currentIndex = 0;

const totalImages = images.length;

function updateSlider() {

slides.style.transform = `translateX(${-currentIndex \* 100}%)`;

}

prevBtn.addEventListener('click', () => {

currentIndex = (currentIndex === 0) ? totalImages - 1 : currentIndex - 1;

updateSlider();

});

nextBtn.addEventListener('click', () => {

currentIndex = (currentIndex === totalImages - 1) ? 0 : currentIndex + 1;

updateSlider();

});

// Auto-slide every 5 seconds

setInterval(() => {

currentIndex = (currentIndex === totalImages - 1) ? 0 : currentIndex + 1;

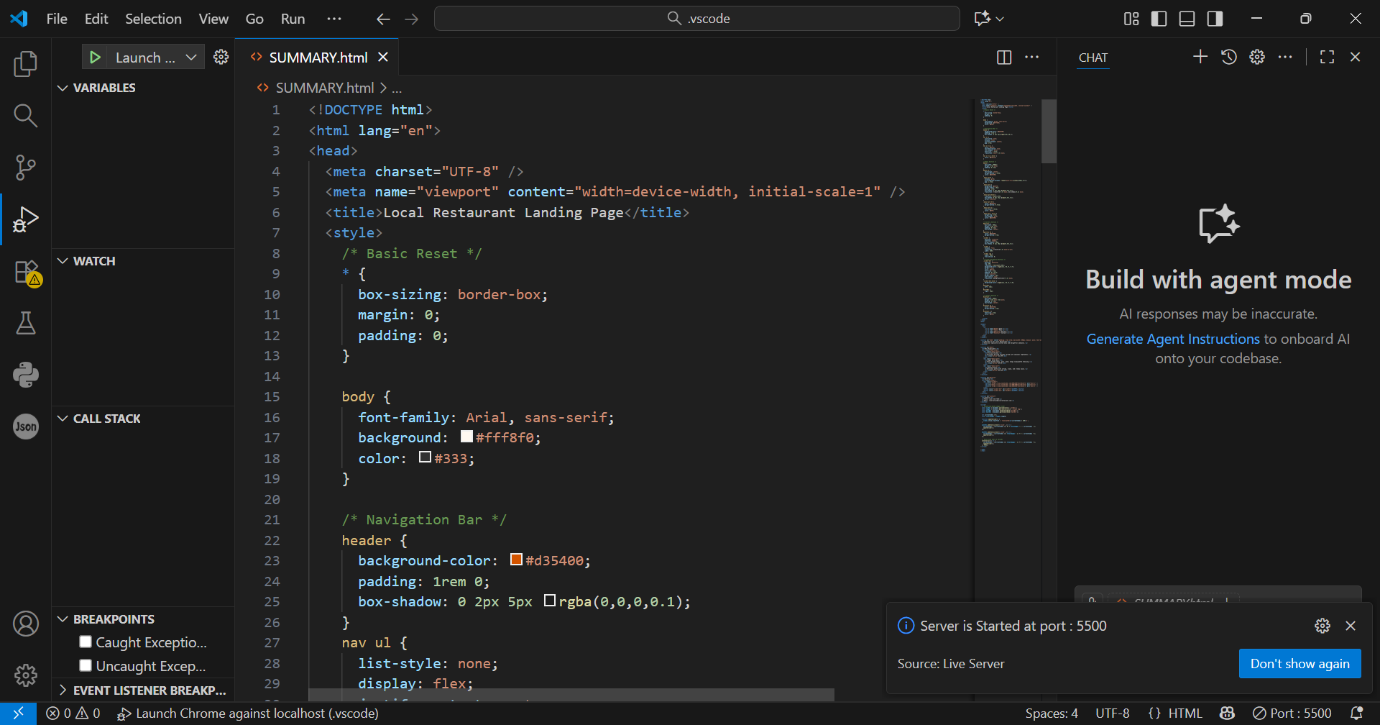
updateSlider();

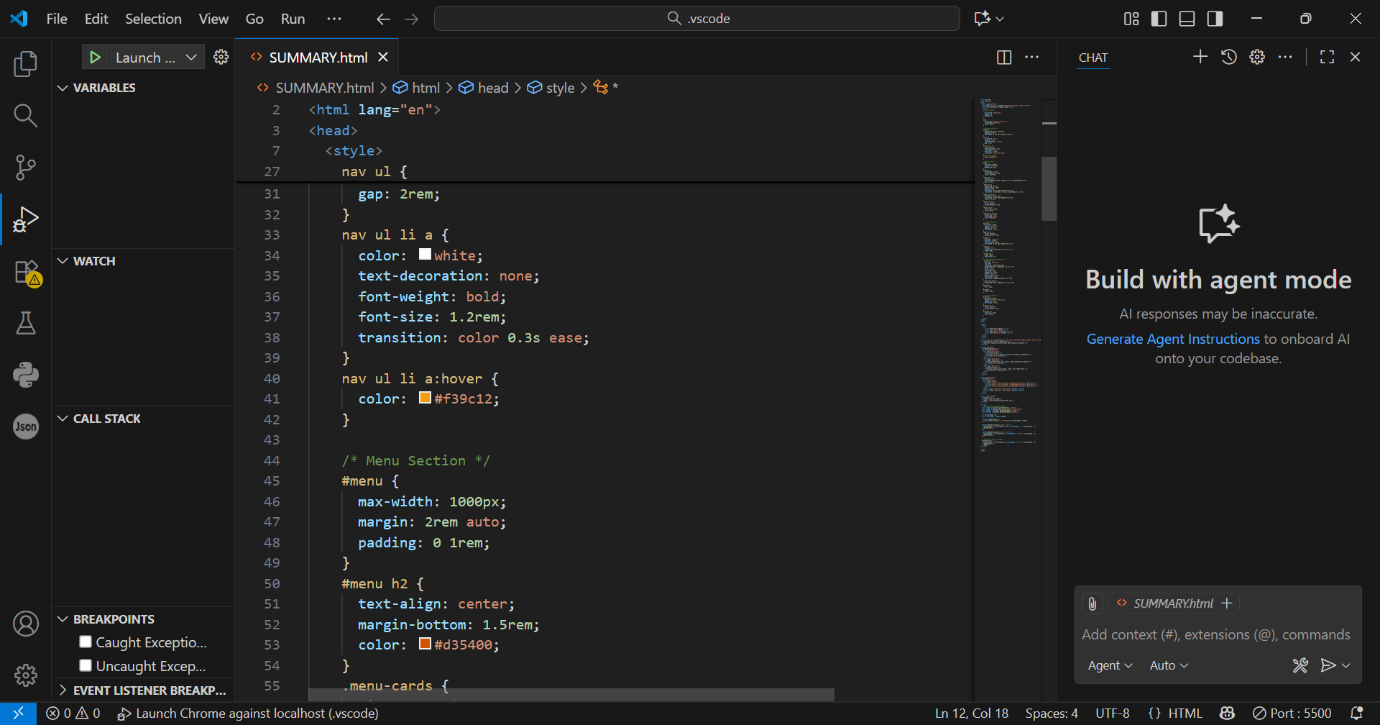
}, 5000);

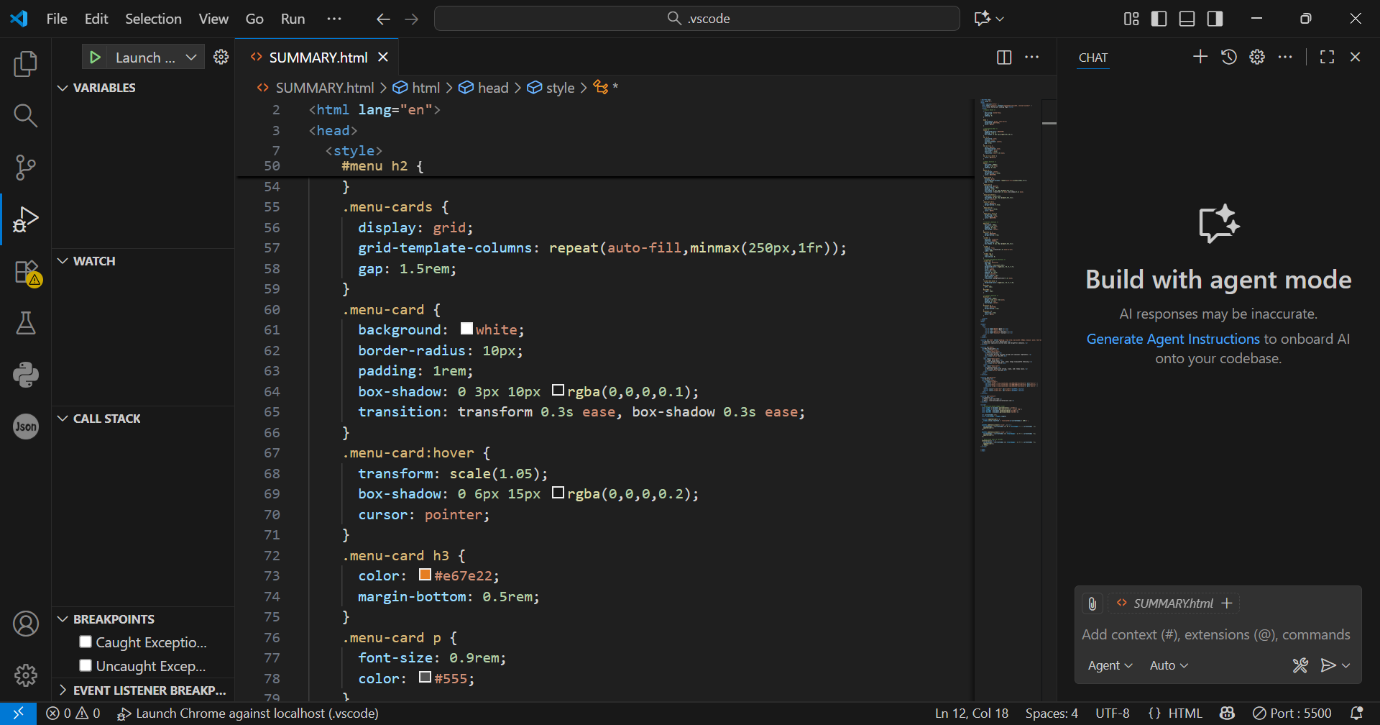
</script>

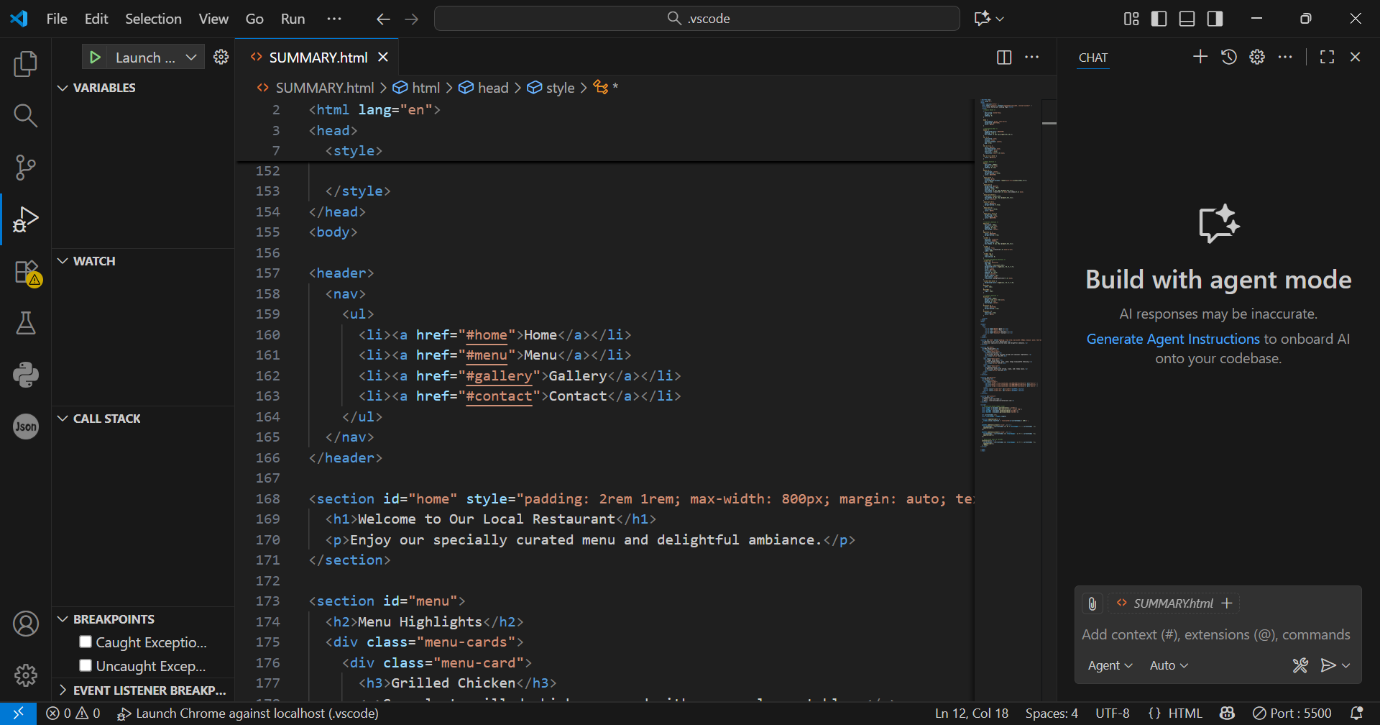
</body>

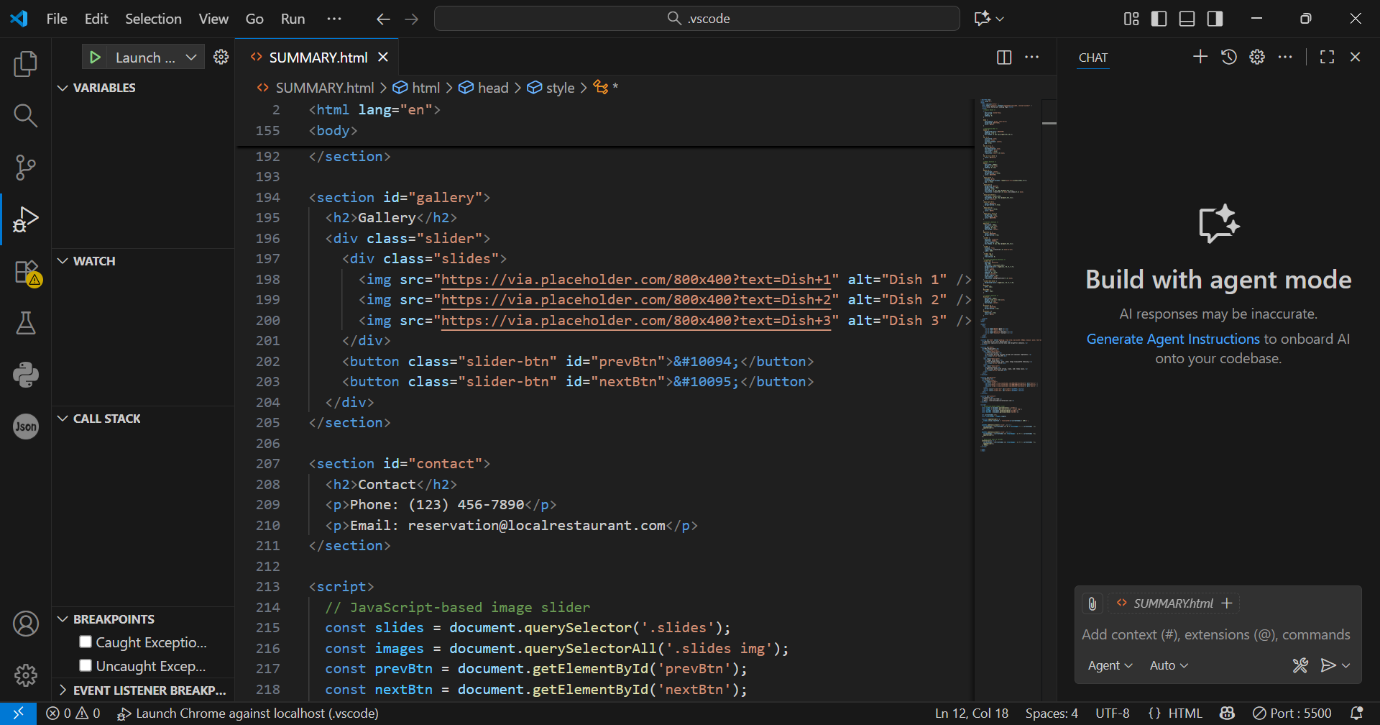
</html>



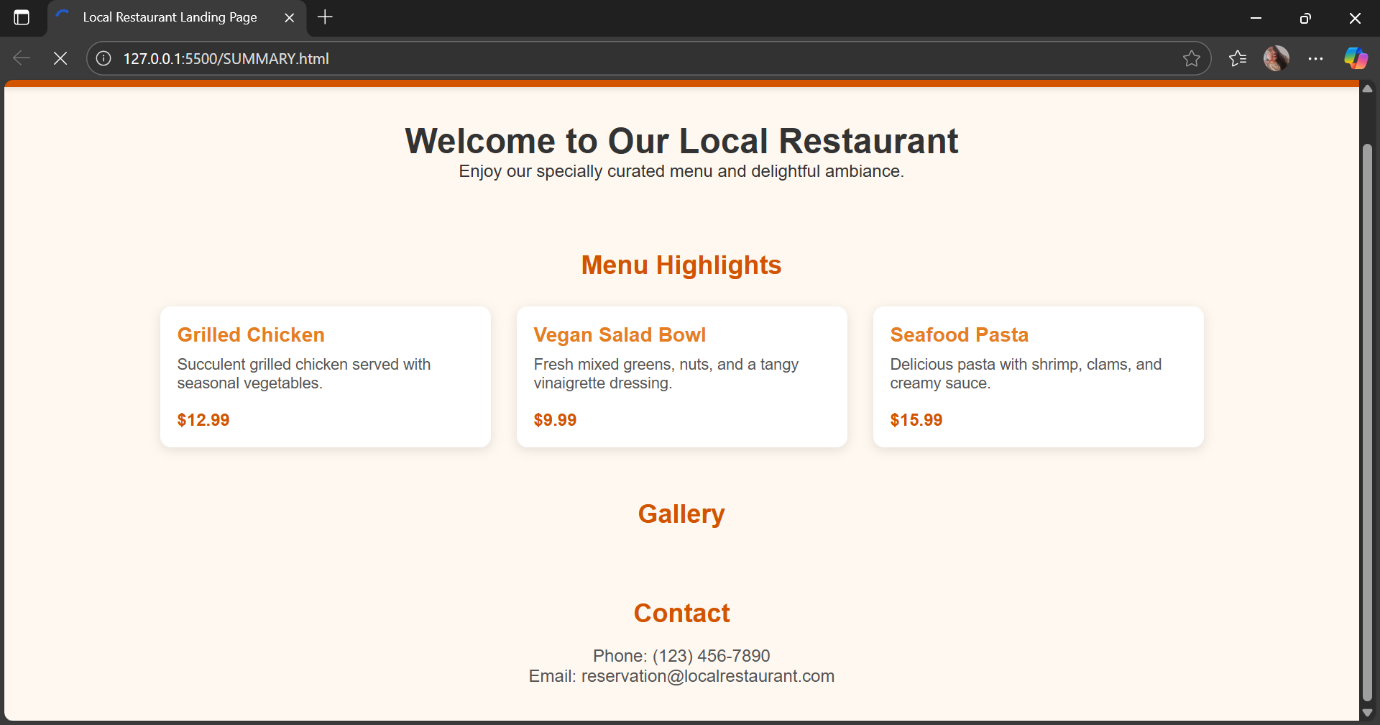








OUTPUT :



* OBSERVATION : A local restaurant requires a straightforward landing page that highlights essential features for visitors.
* The page should include a navigation bar with links to Home, Menu, Gallery, and Contact sections.
* The menu section should present the food or dish highlights using visually appealing CSS-styled cards.
* The gallery section should contain an image slider implemented with JavaScript to showcase photos dynamically.
* The developer uses GitHub Copilot AI to rapidly generate the HTML structure, responsive CSS styling, and JavaScript base code for the image slider.
* This approach accelerates development while delivering a clean, interactive, and responsive webpage tailored for the restaurant’s needs.

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.  
• Ask Copilot to generate an HTML form (fields: Name, Email,  
Phone, Department, Event Selection).  
• Use Copilot to 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.

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).  
 • Add CSS styling for an attractive form  
layout.  
• Implement JavaScript validation (e.g., email format check,  
phone number length check) using Copilot’s suggestions.

CODE : <!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>

body {

background: #f2f3f7;

font-family: 'Segoe UI', Arial, sans-serif;

margin: 0;

padding: 0;

}

.container {

max-width: 480px;

margin: 2.5rem auto;

background: #fff;

border-radius: 10px;

box-shadow: 0 4px 18px rgba(0,0,0,0.12);

padding: 2rem 2.5rem;

}

h2 {

text-align: center;

color: #3f51b5;

margin-bottom: 1.2rem;

}

form {

display: flex;

flex-direction: column;

gap: 1.1rem;

}

label {

font-weight: 500;

margin-bottom: 0.3rem;

color: #333;

}

input, select {

padding: 0.7rem;

border: 1px solid #b8b8b8;

border-radius: 6px;

font-size: 1rem;

transition: border-color 0.2s;

}

input:focus, select:focus {

border-color: #3f51b5;

outline: none;

}

.error {

color: #d32f2f;

font-size: 0.92rem;

margin-top: -0.5rem;

margin-bottom: 0.3rem;

display: none;

}

button {

background: #3f51b5;

color: #fff;

border: none;

padding: 0.9rem;

border-radius: 6px;

font-size: 1.06rem;

font-weight: bold;

cursor: pointer;

margin-top: 0.7rem;

transition: background 0.2s;

}

button:hover {

background: #1a237e;

}

</style>

</head>

<body>

<div class="container">

<h2>Tech Fest Registration</h2>

<form id="regForm" autocomplete="off" novalidate>

<div>

<label for="name">Name:</label>

<input type="text" id="name" required />

<div class="error" id="nameError">Please enter your name.</div>

</div>

<div>

<label for="email">Email:</label>

<input type="email" id="email" required />

<div class="error" id="emailError">Enter a valid email address.</div>

</div>

<div>

<label for="phone">Phone:</label>

<input type="tel" id="phone" maxlength="10" required />

<div class="error" id="phoneError">Phone must be 10 digits.</div>

</div>

<div>

<label for="dept">Department:</label>

<select id="dept" required>

<option value="">--Select Department--</option>

<option value="CSE">CSE</option>

<option value="ECE">ECE</option>

<option value="EEE">EEE</option>

<option value="Mechanical">Mechanical</option>

<option value="Civil">Civil</option>

</select>

<div class="error" id="deptError">Please select your department.</div>

</div>

<div>

<label for="event">Event Selection:</label>

<select id="event" required>

<option value="">--Select Event--</option>

<option value="Coding Challenge">Coding Challenge</option>

<option value="Robotics">Robotics</option>

<option value="Hackathon">Hackathon</option>

<option value="Quiz">Quiz</option>

</select>

<div class="error" id="eventError">Please select an event.</div>

</div>

<button type="submit">Submit Registration</button>

</form>

</div>

<script>

// Real-time validation

const form = document.getElementById('regForm');

const nameEl = document.getElementById('name');

const emailEl = document.getElementById('email');

const phoneEl = document.getElementById('phone');

const deptEl = document.getElementById('dept');

const eventEl = document.getElementById('event');

const nameError = document.getElementById('nameError');

const emailError = document.getElementById('emailError');

const phoneError = document.getElementById('phoneError');

const deptError = document.getElementById('deptError');

const eventError = document.getElementById('eventError');

nameEl.addEventListener('input', () => {

if (nameEl.value.trim() === "") {

nameError.style.display = 'block';

} else {

nameError.style.display = 'none';

}

});

emailEl.addEventListener('input', () => {

const emailReg = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (!emailReg.test(emailEl.value.trim())) {

emailError.style.display = 'block';

} else {

emailError.style.display = 'none';

}

});

phoneEl.addEventListener('input', () => {

const phoneVal = phoneEl.value.trim();

if (!/^\d{10}$/.test(phoneVal)) {

phoneError.style.display = 'block';

} else {

phoneError.style.display = 'none';

}

});

deptEl.addEventListener('change', () => {

if (deptEl.value === "") {

deptError.style.display = 'block';

} else {

deptError.style.display = 'none';

}

});

eventEl.addEventListener('change', () => {

if (eventEl.value === "") {

eventError.style.display = 'block';

} else {

eventError.style.display = 'none';

}

});

form.addEventListener('submit', function (e) {

let valid = true;

if (nameEl.value.trim() === "") {

nameError.style.display = 'block';

valid = false;

}

if (!/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(emailEl.value.trim())) {

emailError.style.display = 'block';

valid = false;

}

if (!/^\d{10}$/.test(phoneEl.value.trim())) {

phoneError.style.display = 'block';

valid = false;

}

if (deptEl.value === "") {

deptError.style.display = 'block';

valid = false;

}

if (eventEl.value === "") {

eventError.style.display = 'block';

valid = false;

}

if (!valid) {

e.preventDefault();

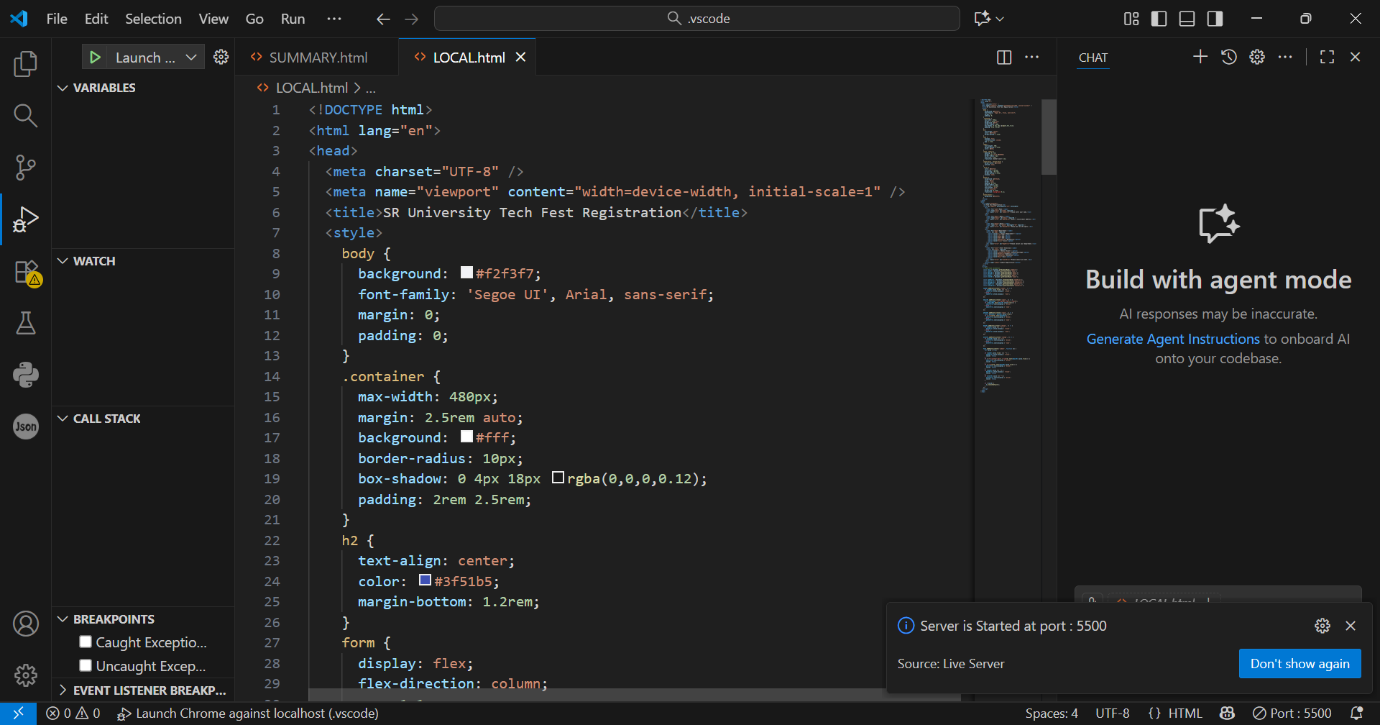
}

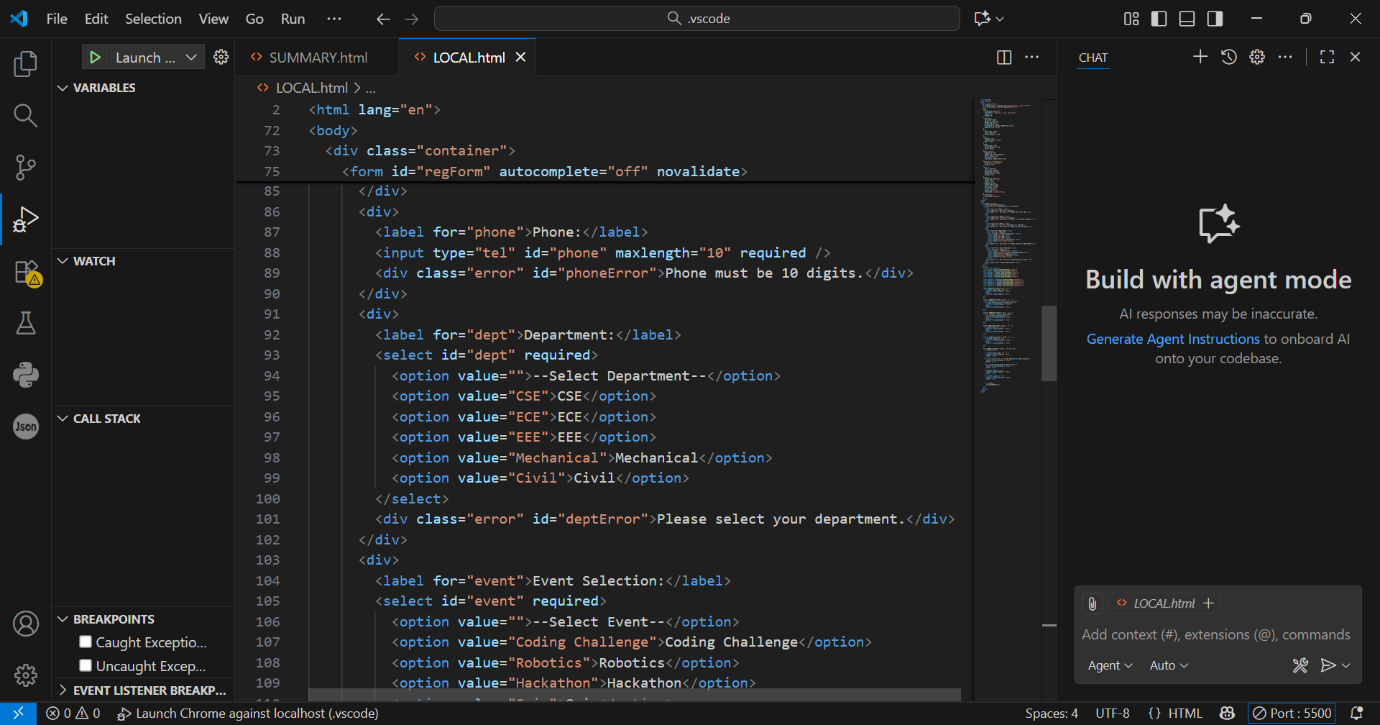
});

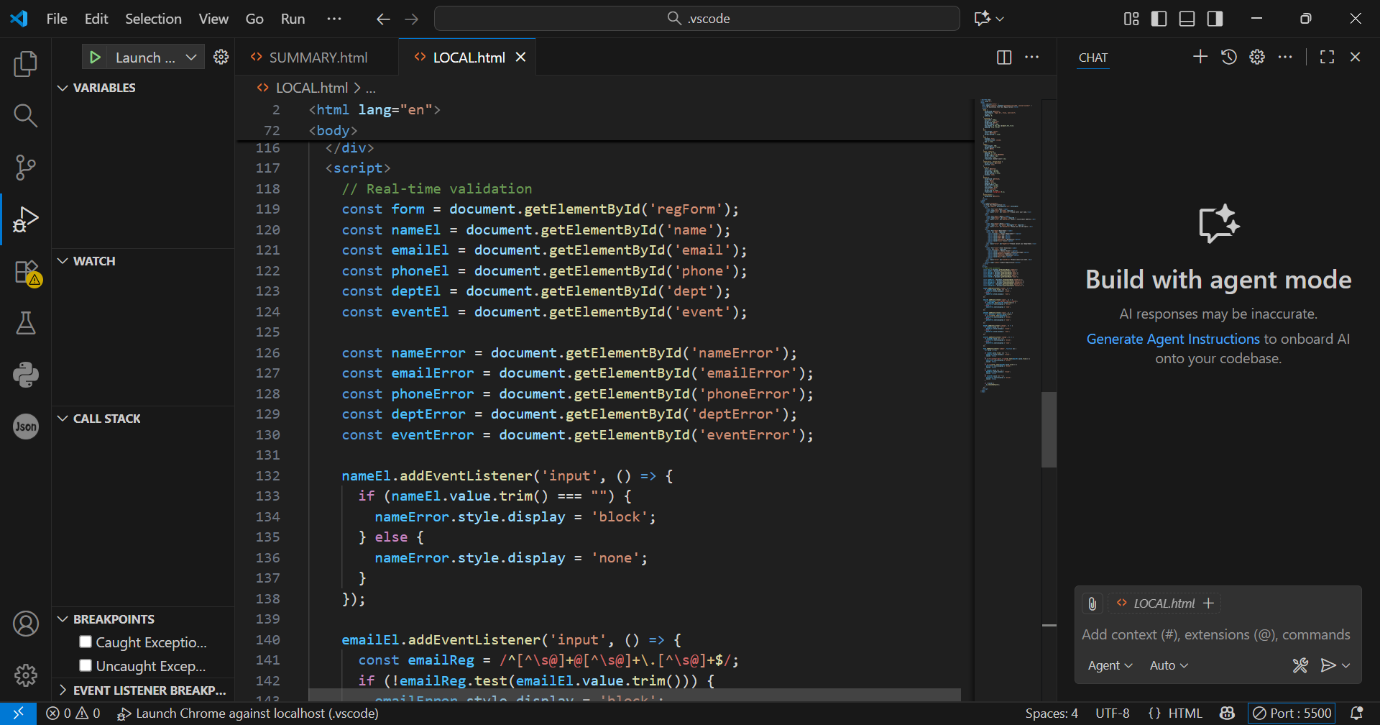
</script>

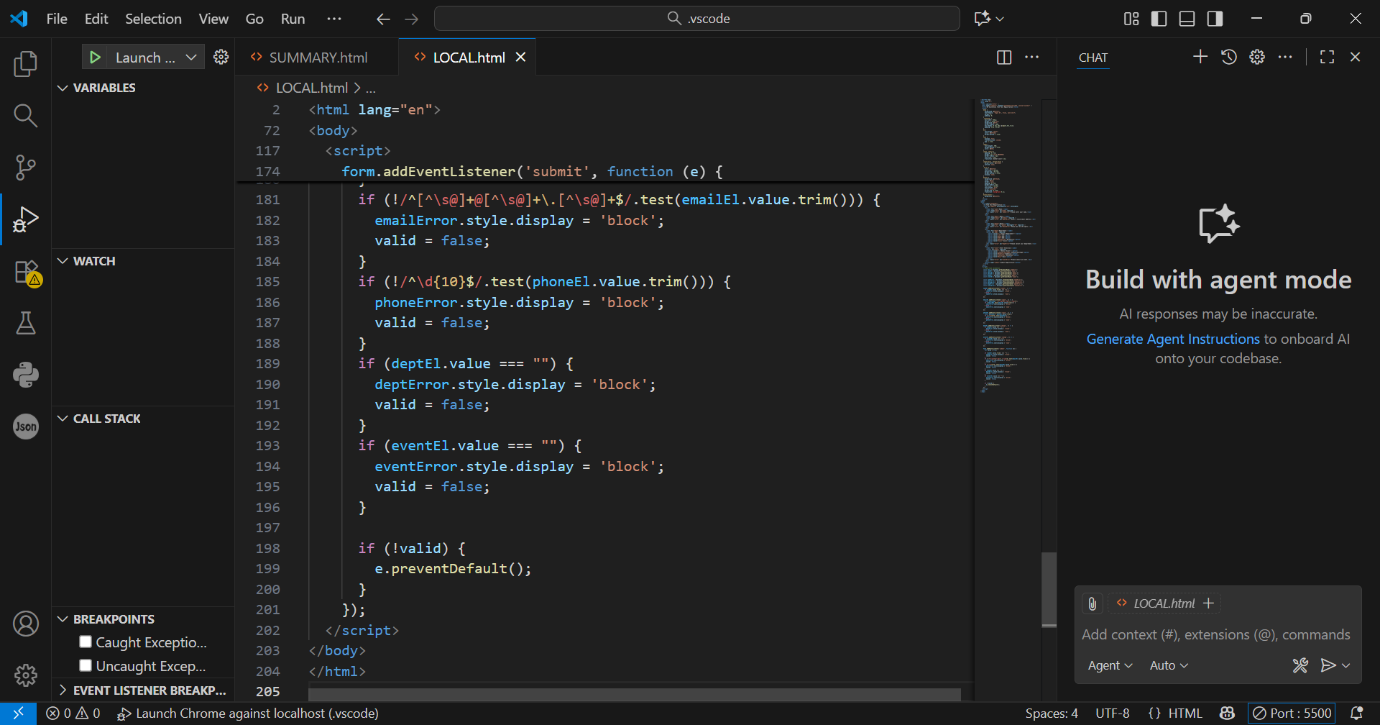
</body>

</html>

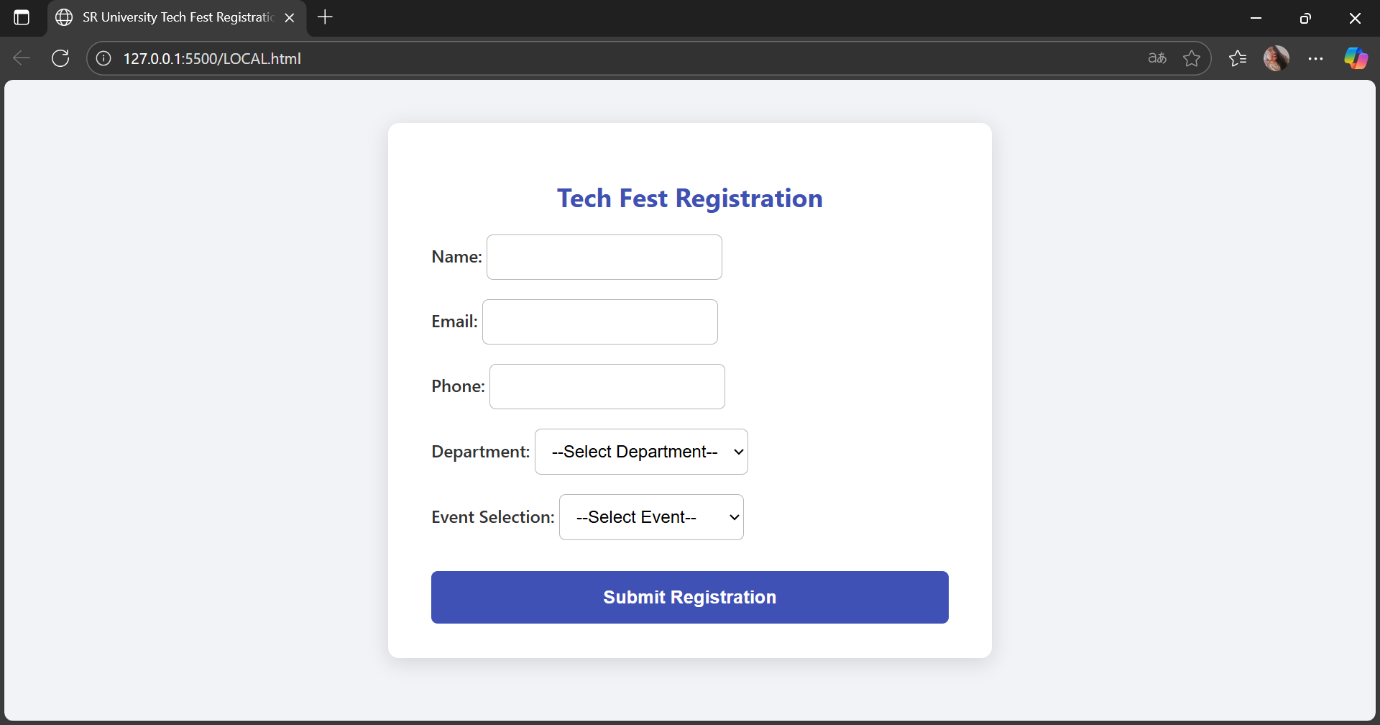








OUTPUT :



* OBSERVATION : Develop a web-based event registration form for SR University's tech fest.
* The form includes fields: Name, Email, Phone, Department, and Event Selection.
* Use GitHub Copilot to generate the HTML form markup.
* Style the form attractively with CSS, with Copilot assistance.
* Implement real-time JavaScript validation for input fields:
  + Validate name is entered.
  + Check email format correctness.
  + Ensure phone number is exactly 10 digits.
  + Require selection of Department and Event.
* Provide immediate feedback on invalid inputs before form submission.
* Enhance user experience and data accuracy while speeding up development using AI assistance.

Task 4: AI-Assisted E-Commerce Product Page  
Scenario:  
A startup wants a basic e-commerce product page to display products  
with prices and an “Add to Cart” button.  
• Use Copilot to generate a grid-based product catalog in  
HTML/CSS.  
• Implement a JavaScript “Add to Cart” functionality with  
Copilot’s guidance.  
• Modify Copilot’s suggestions to include a cart counter at the  
top-right corner of the 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.  
• Implement a JavaScript “Add to Cart” functionality with  
Copilot’s guidance.  
• Include a cart counter at the top-right corner of the page.

CODE : <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>Startup E-Commerce Page</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

background-color: #f4f4f4;

}

header {

display: flex;

justify-content: flex-end;

padding: 1rem;

background-color: #333;

position: relative;

}

#cart {

position: absolute;

top: 10px;

right: 20px;

background: #ff5722;

color: white;

padding: 0.3rem 0.7rem;

border-radius: 20px;

font-weight: bold;

font-size: 1rem;

}

.product-grid {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));

gap: 2rem;

padding: 2rem;

margin: 0 auto;

max-width: 1000px;

}

.product-card {

background-color: white;

padding: 1rem;

border-radius: 8px;

box-shadow: 0 2px 8px rgba(0,0,0,0.1);

display: flex;

flex-direction: column;

align-items: center;

}

.product-card h3 {

margin-bottom: 0.5rem;

color: #333;

}

.price {

font-size: 1.2rem;

margin-bottom: 1rem;

color: #e91e63;

}

button {

padding: 0.5rem 1rem;

background-color: #2196f3;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

font-weight: bold;

transition: background-color 0.2s;

}

button:hover {

background-color: #1976d2;

}

</style>

</head>

<body>

<header>

<div id="cart">Cart: 0</div>

</header>

<div class="product-grid">

<div class="product-card">

<h3>Product 1</h3>

<p class="price">$19.99</p>

<button onclick="addToCart()">Add to Cart</button>

</div>

<div class="product-card">

<h3>Product 2</h3>

<p class="price">$29.99</p>

<button onclick="addToCart()">Add to Cart</button>

</div>

<div class="product-card">

<h3>Product 3</h3>

<p class="price">$39.99</p>

<button onclick="addToCart()">Add to Cart</button>

</div>

<div class="product-card">

<h3>Product 4</h3>

<p class="price">$49.99</p>

<button onclick="addToCart()">Add to Cart</button>

</div>

</div>

<script>

let cartCount = 0;

function addToCart() {

cartCount++;

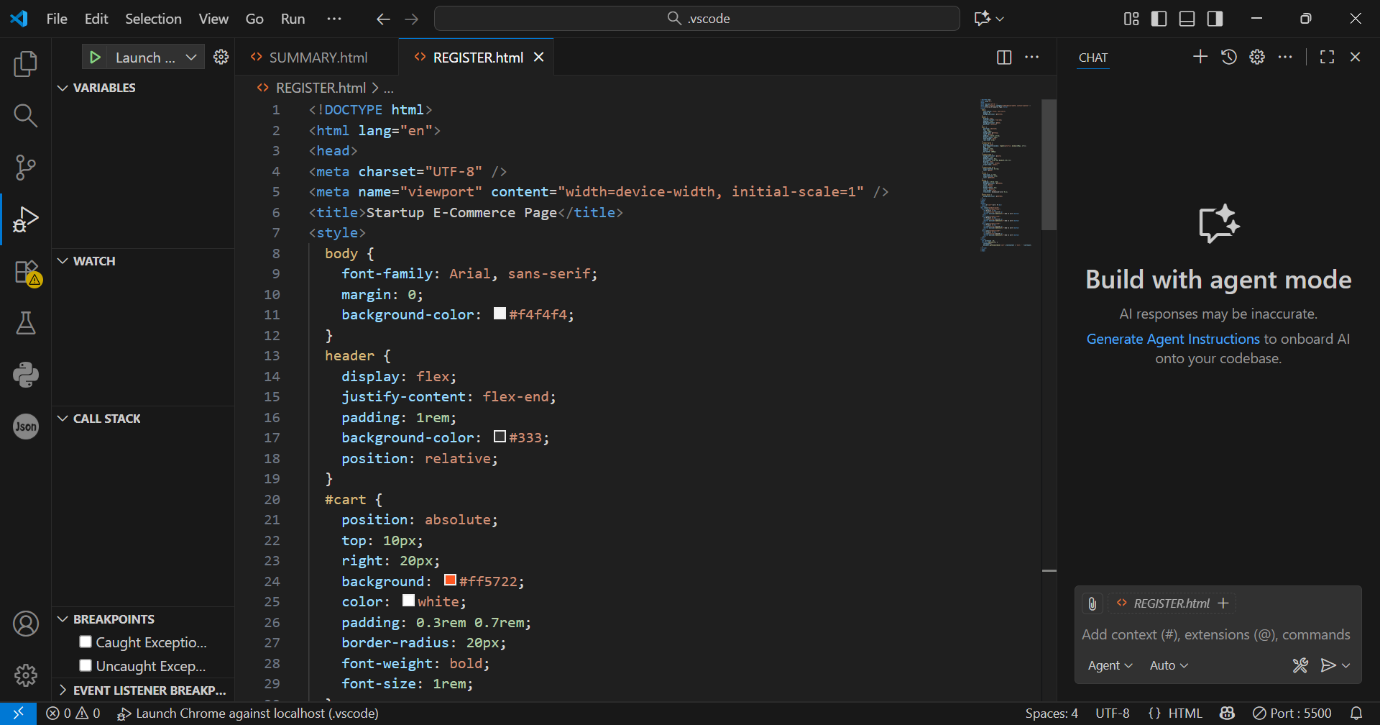
document.getElementById('cart').textContent = 'Cart: ' + cartCount;

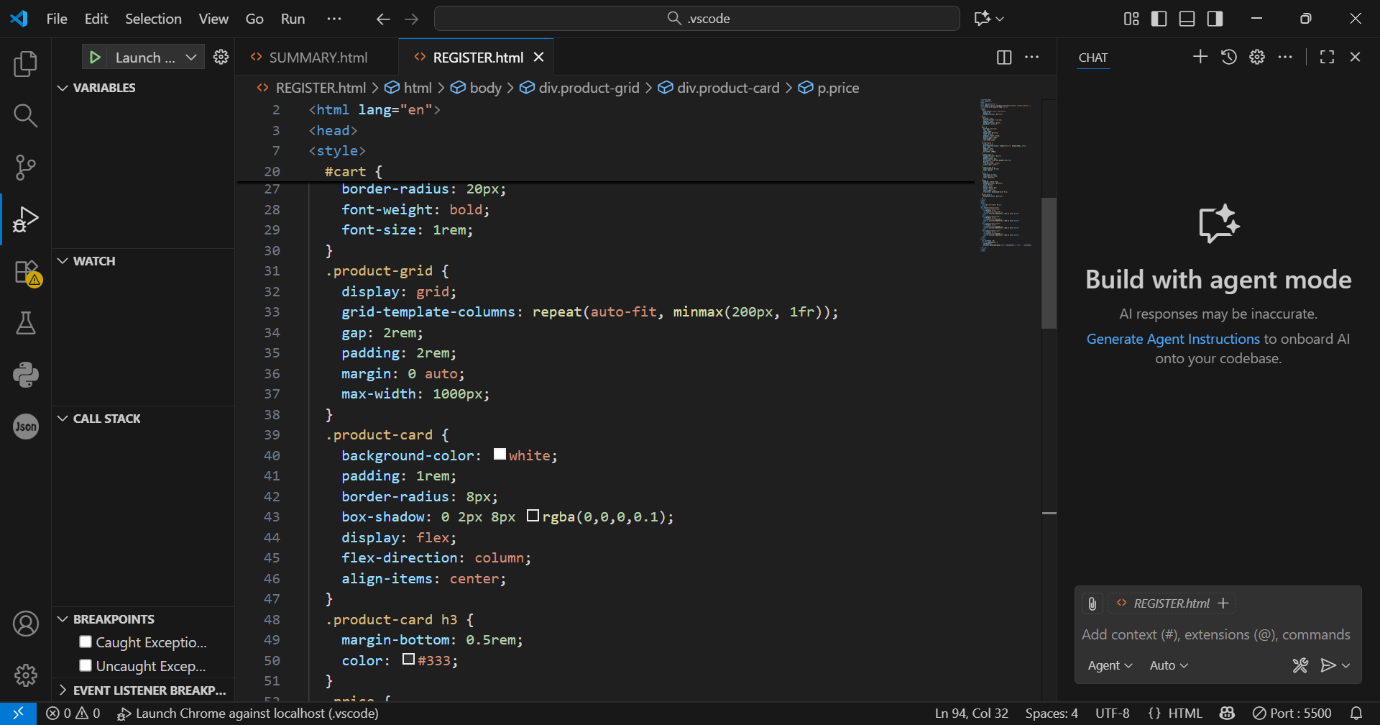
}

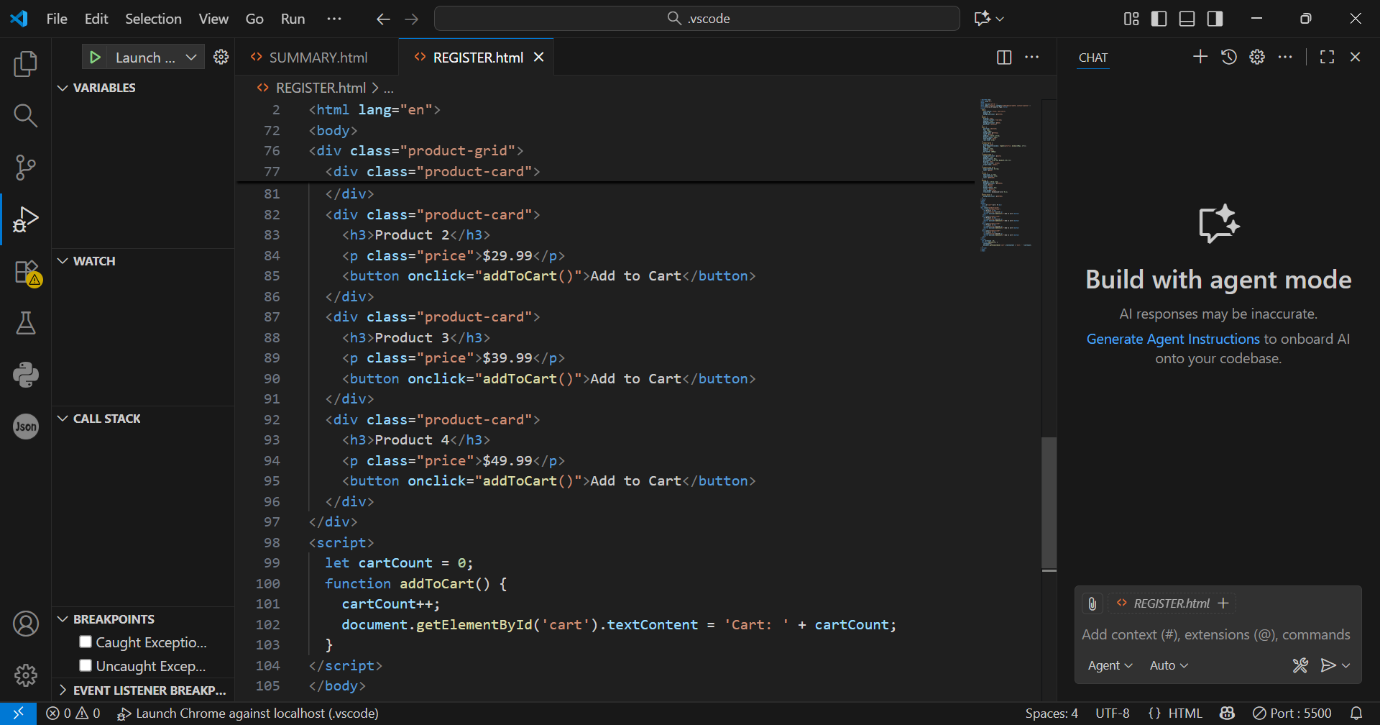
</script>

</body>

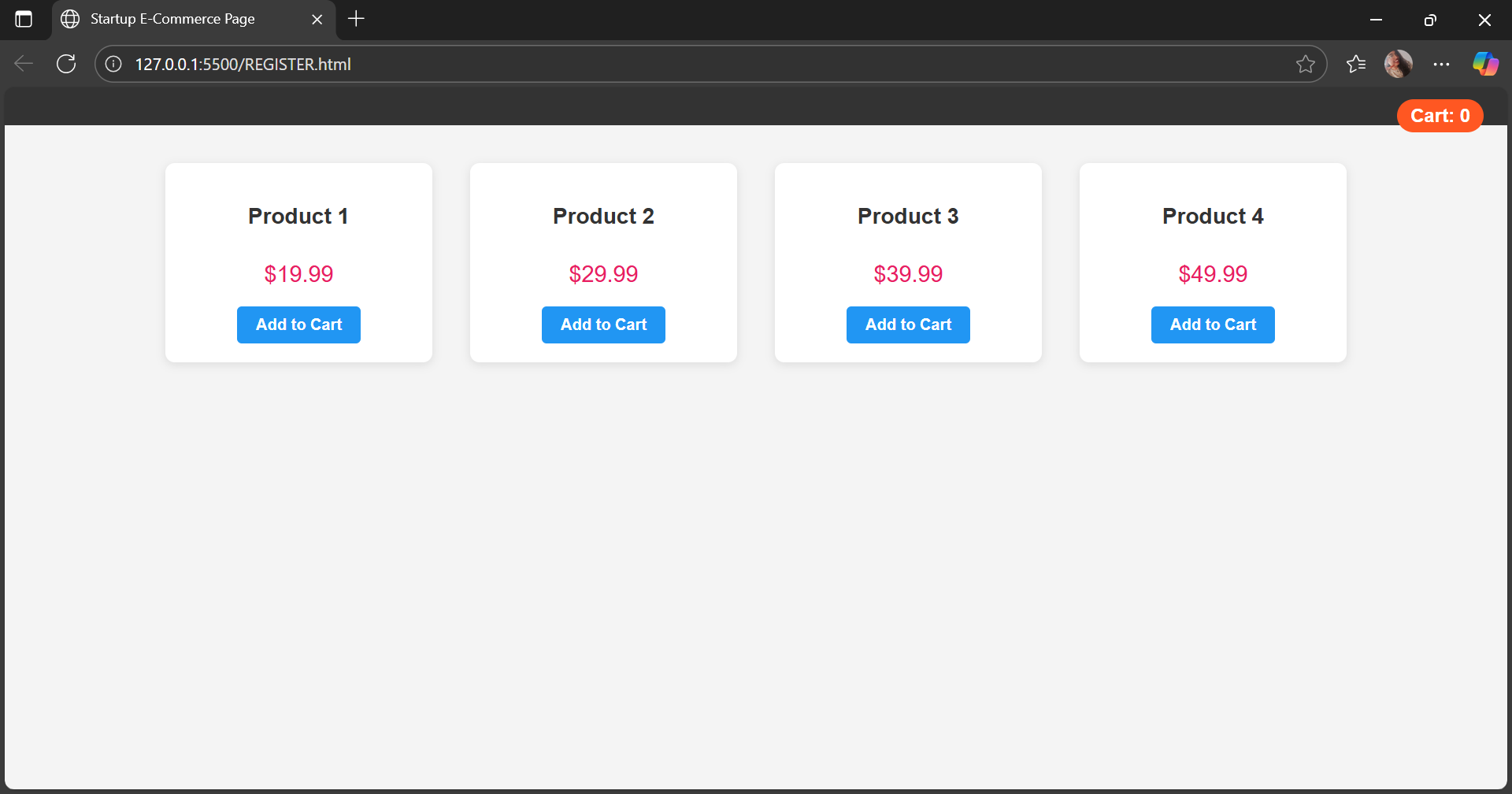
</html>







OUTPUT :



* OBSERVATION : Develop a basic e-commerce product page for a startup to showcase products with prices and an “Add to Cart” button.
* Use GitHub Copilot to generate an HTML grid layout for displaying the product catalog.
* Style the product page with CSS to create visually appealing product cards arranged in a responsive grid.
* Implement JavaScript functionality to add products to the cart.
* Customize the JavaScript code to include a live cart counter displayed at the top-right corner of the page.
* The cart counter updates dynamically as users add items, enhancing user experience and interaction.
* Utilize AI assistance to speed up development and ensure clean, modern code.