Lesson 01 Demo 06

Generating Code for the Web Page of an E-commerce Application

Objective: To create web pages using HTML and CSS code with the help of generative AI for the initial code generation of a product web page

Tools required: ChatGPT and Visual Studio Code

Prerequisites: None

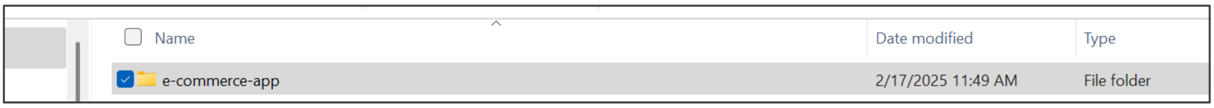
Note: The generative AI tool used in this exercise can produce varied outputs even when presented with the same prompts. Hence, you may get different outputs for the prompts provided.

Steps to be followed:

1. Create login and sign-up pages
2. Create a product page

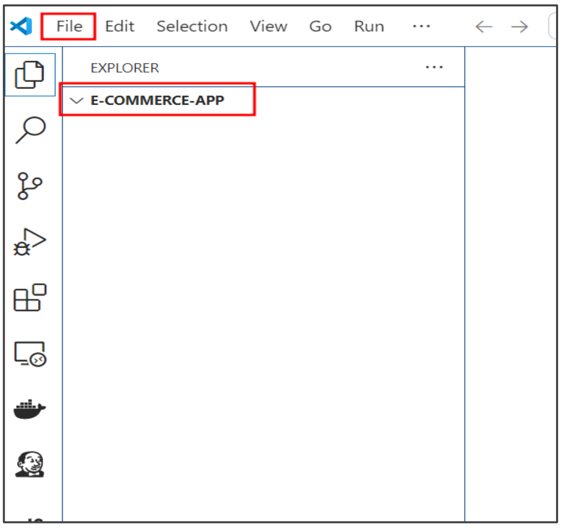
Step 1: Create login and sign-up pages

1.1 Create a folder named **e-commerce-app** on your system's local drive

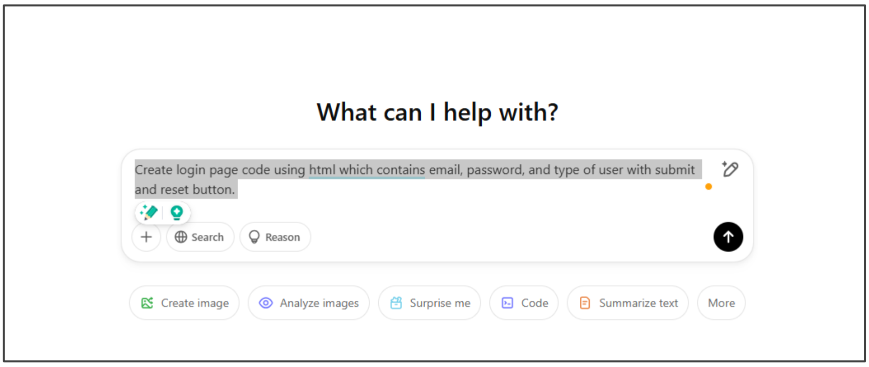


Name	Date modified	Type
e-commerce-app	2/17/2025 11:49 AM	File folder

1.2 Open VS Code, click on **File**, and open the folder created in the previous step



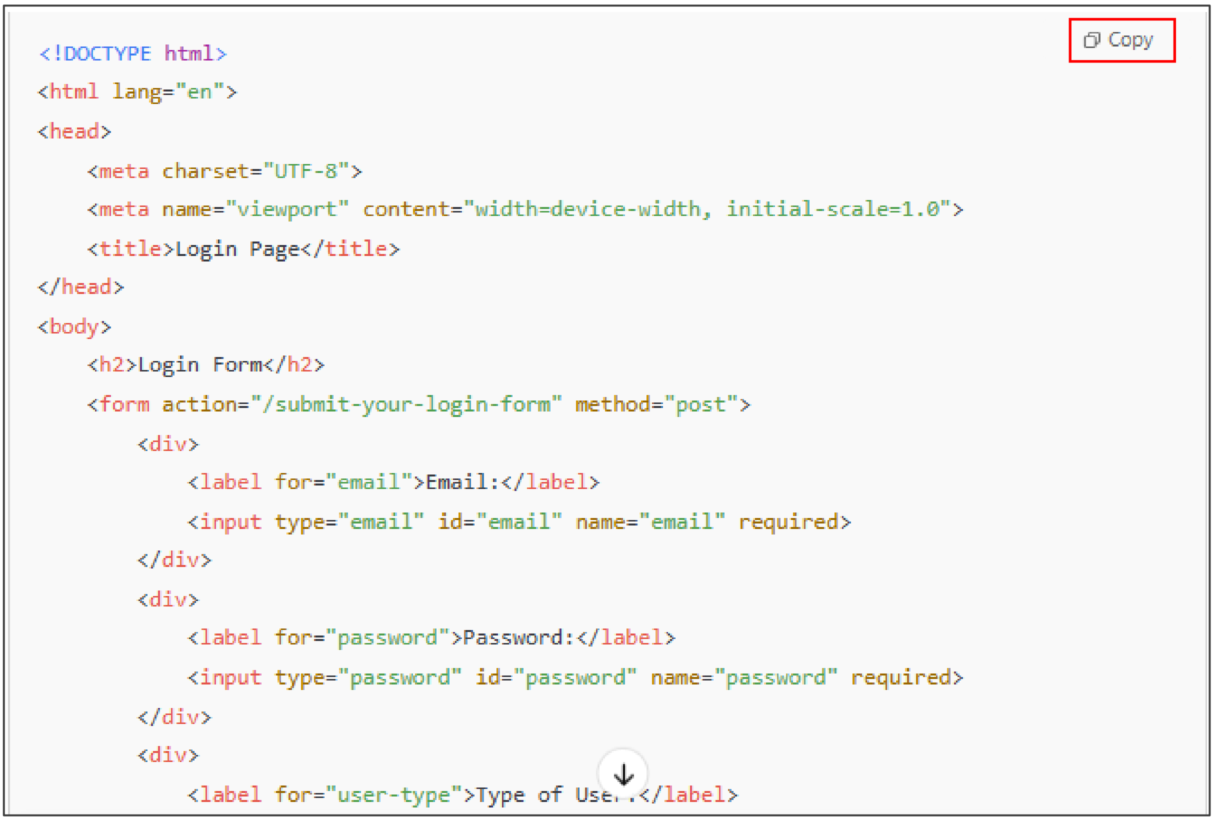
1.3 Open any web browser, navigate to ChatGPT, and enter the following prompt:
Create login page code using html which contains email, password, and type of user with submit and reset button.



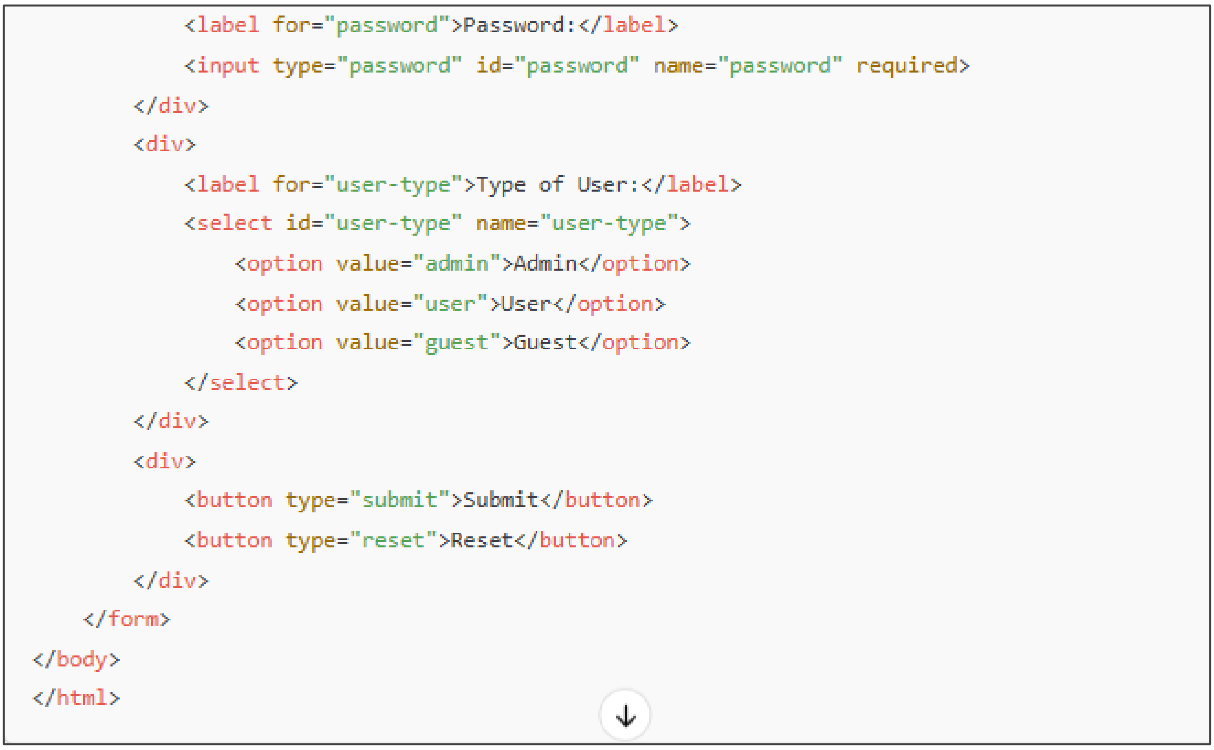
What can I help with?

Create login page code using html which contains email, password, and type of user with submit and reset button.

1.4 Click on **Copy** to copy the HTML code generated for the login page

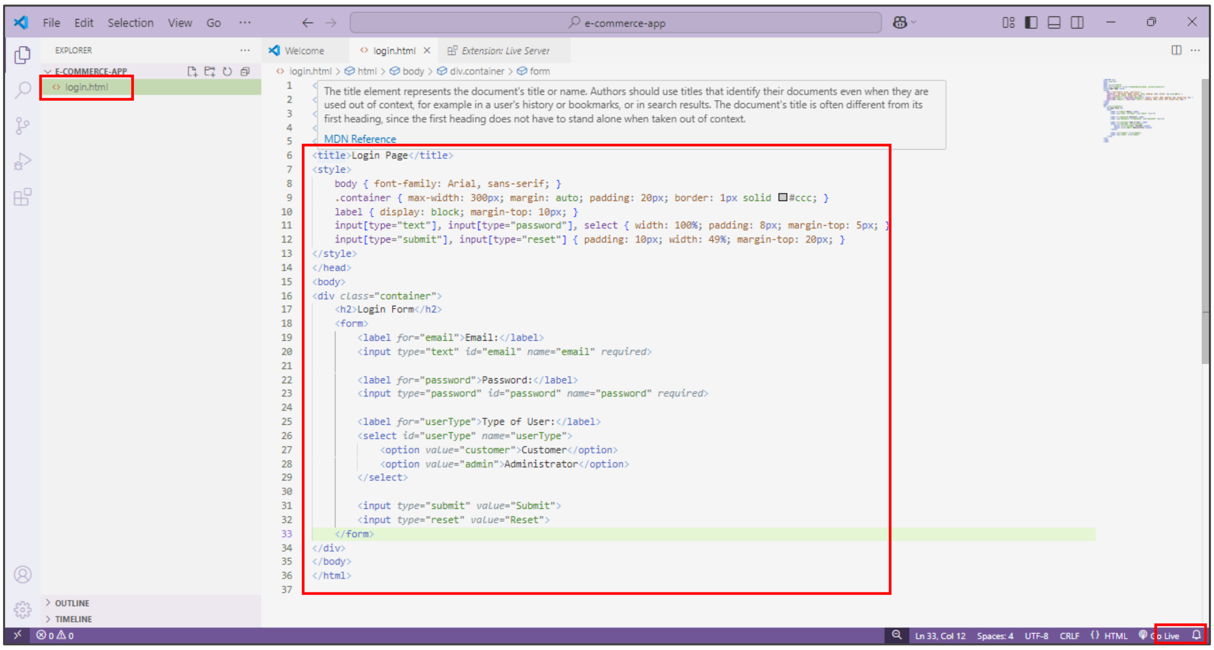


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Page</title>
</head>
<body>
    <h2>Login Form</h2>
    <form action="/submit-your-login-form" method="post">
        <div>
            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required>
        </div>
        <div>
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>
        </div>
        <div>
            <label for="user-type">Type of Use  :</label>
```



```
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>
        </div>
        <div>
            <label for="user-type">Type of User:</label>
            <select id="user-type" name="user-type">
                <option value="admin">Admin</option>
                <option value="user">User</option>
                <option value="guest">Guest</option>
            </select>
        </div>
        <div>
            <button type="submit">Submit</button>
            <button type="reset">Reset</button>
        </div>
    </form>
</body>
</html>
```

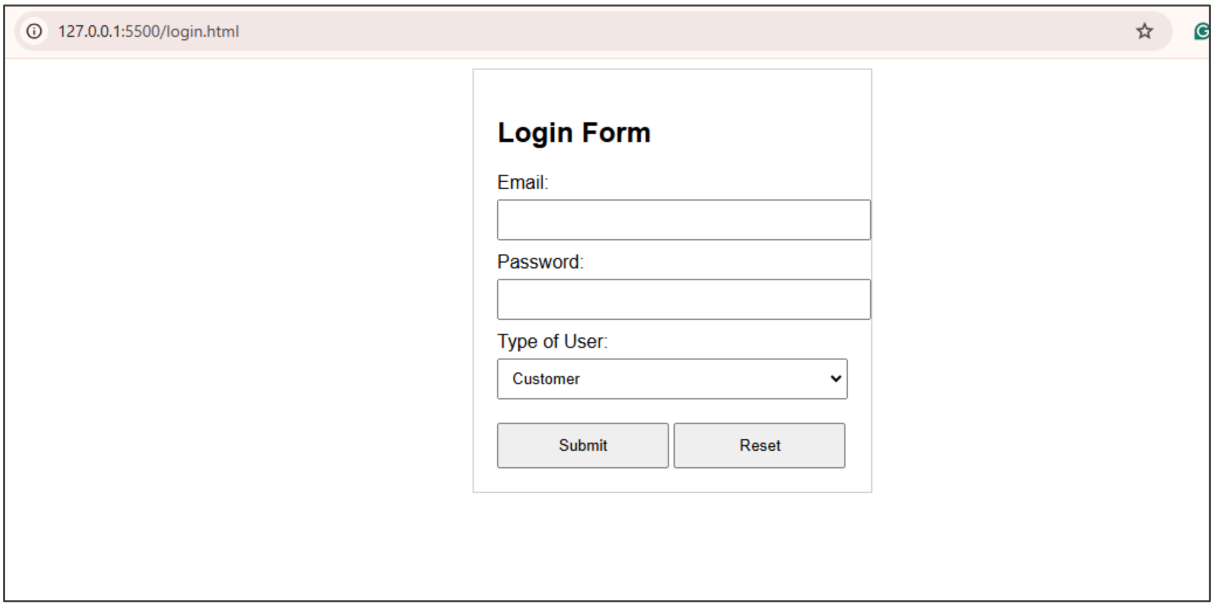
1.5 Create a file named **login.html** in the E-COMMERCE-APP folder in VS Code, paste the copied code into it, and click on **Go Live** at the bottom to view the page in a browser



The title element represents the document's title or name. Authors should use titles that identify their documents even when they are used out of context, for example in a user's history or bookmarks, or in search results. The document's title is often different from its first heading, since the first heading does not have to stand alone when taken out of context.

MDN Reference

```
<title>Login Page</title>
<style>
    body { font-family: Arial, sans-serif; }
    .container { max-width: 300px; margin: auto; padding: 20px; border: 1px solid #ccc; }
    label { display: block; margin-top: 10px; }
    input[type="text"], input[type="password"], select { width: 100%; padding: 8px; margin-top: 5px; }
    input[type="submit"], input[type="reset"] { padding: 10px; width: 49%; margin-top: 20px; }
</style>
</head>
<body>
<div class="container">
    <h2>Login Form</h2>
    <form>
        <label for="email">Email:</label>
        <input type="text" id="email" name="email" required>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>

        <label for="userType">Type of User:</label>
        <select id="userType" name="userType">
            <option value="customer">Customer</option>
            <option value="admin">Administrator</option>
        </select>

        <input type="submit" value="Submit">
        <input type="reset" value="Reset">
    </form>
</div>
</body>
</html>
```

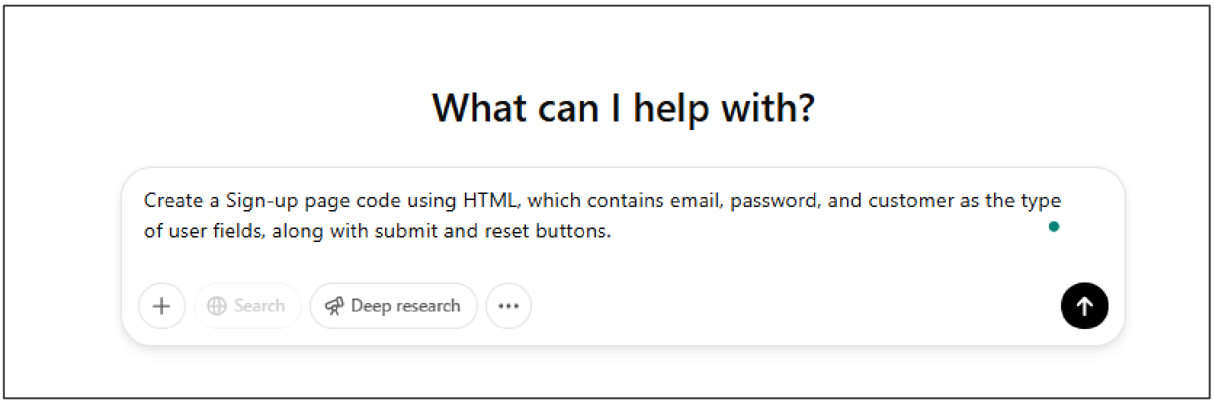


Type of User: Customer

Note: By default, the live server runs on port number 5500.

Note: VS Code offers several options for running an HTML page. If you do not see the **Go live** option, install the Live Server extension which facilitates running the web page in a browser.

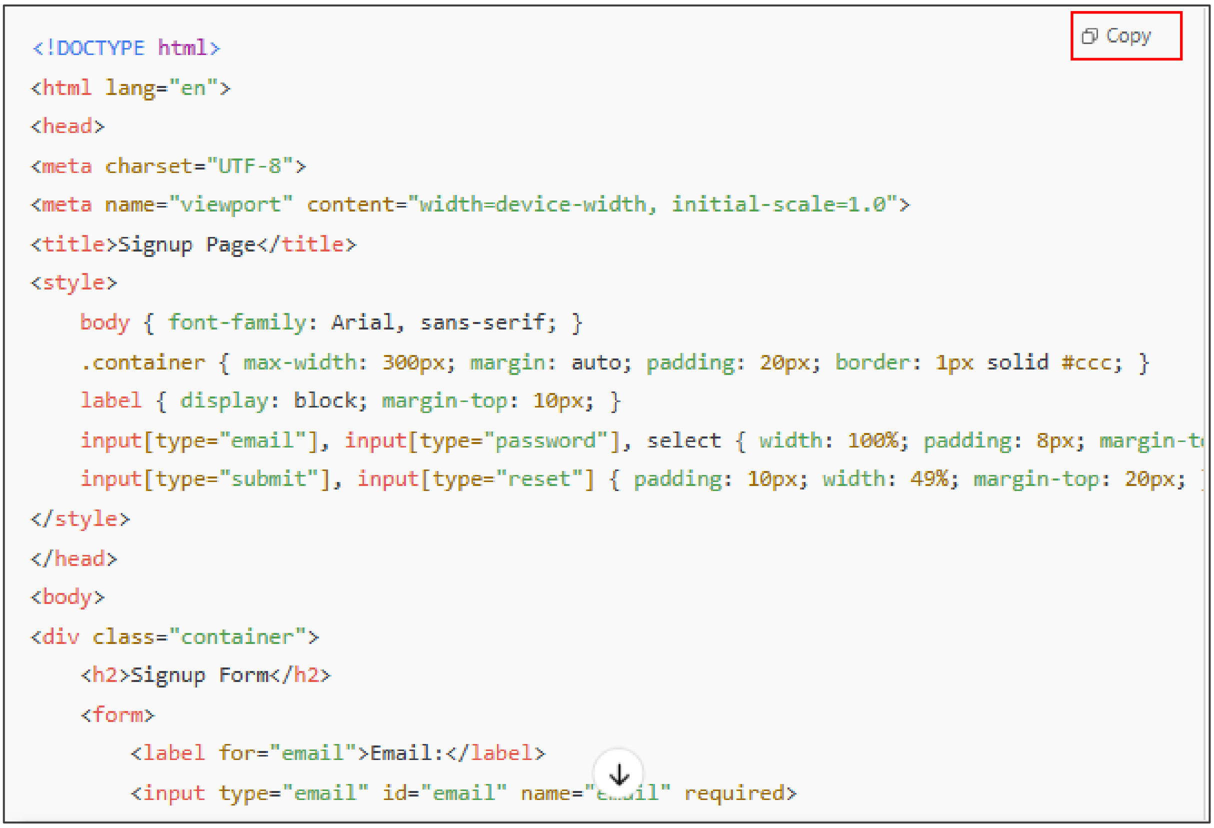
1.6 Open ChatGPT and enter the following prompt to generate code for a sign-up page:
Create a Sign-up page code using HTML, which contains email, password, and customer as the type of user fields, along with submit and reset buttons.



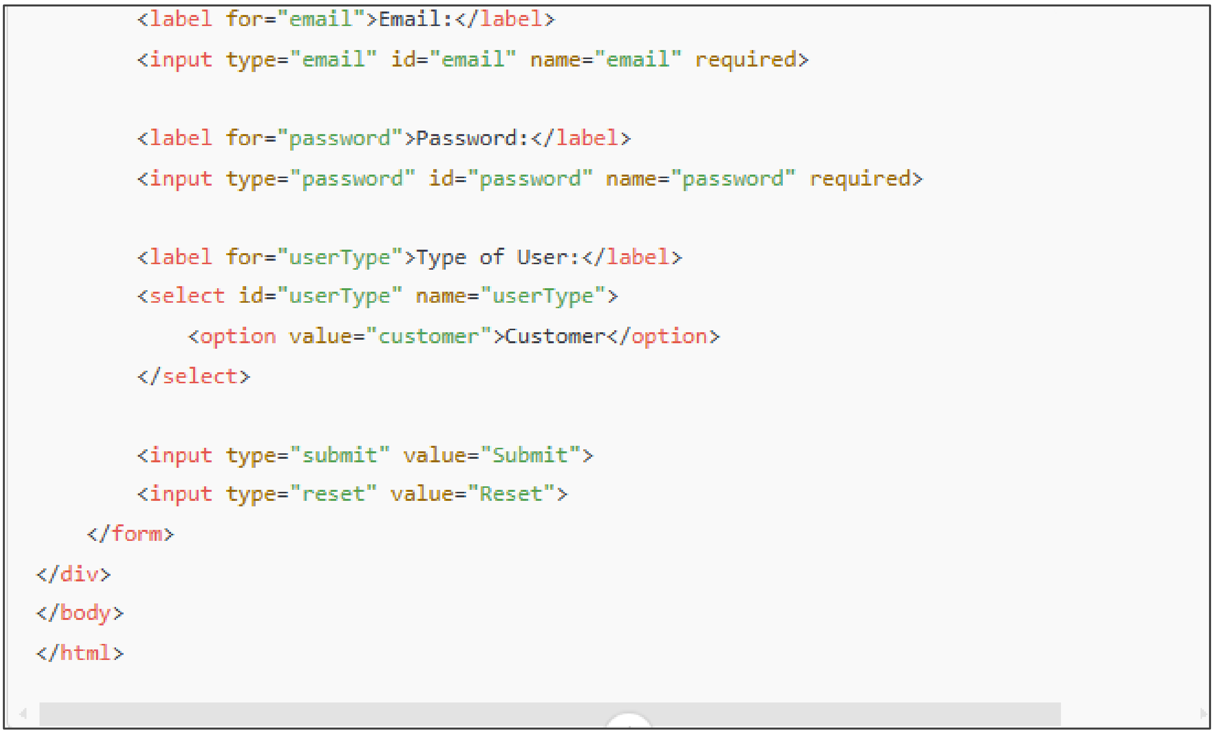
What can I help with?

Create a Sign-up page code using HTML, which contains email, password, and customer as the type of user fields, along with submit and reset buttons.

1.7 Click on **Copy**

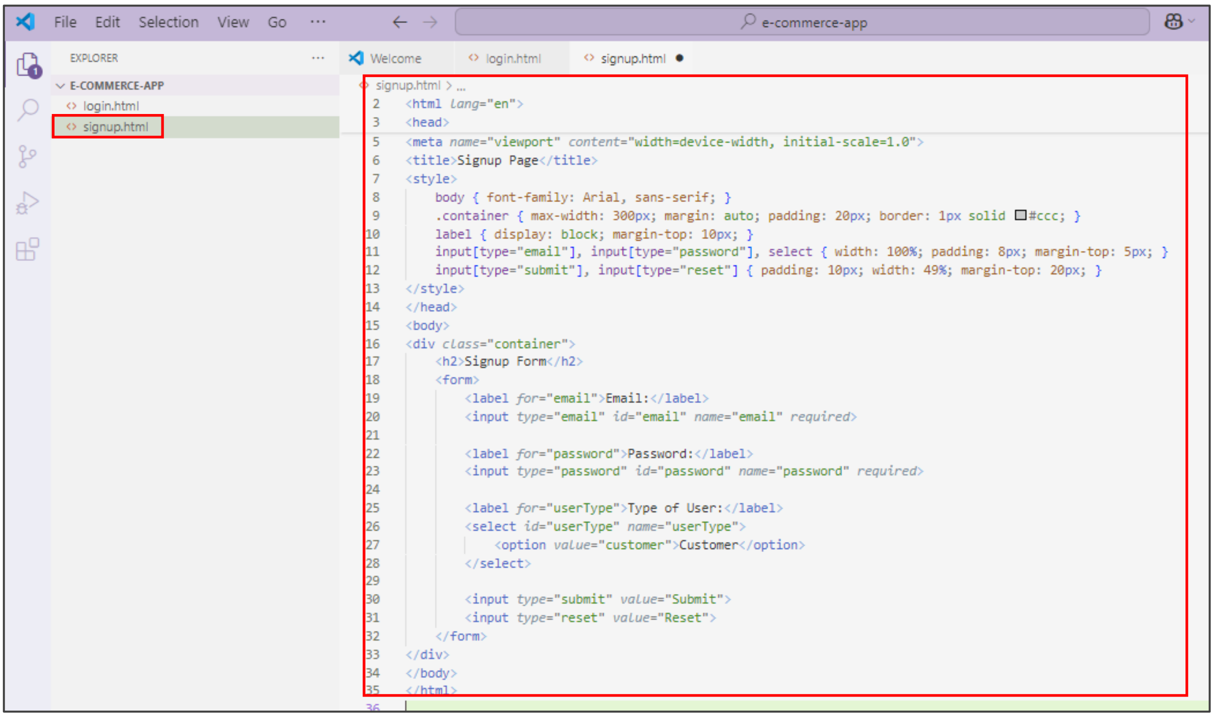


```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Signup Page</title>
<style>
    body { font-family: Arial, sans-serif; }
    .container { max-width: 300px; margin: auto; padding: 20px; border: 1px solid #ccc; }
    label { display: block; margin-top: 10px; }
    input[type="email"], input[type="password"], select { width: 100%; padding: 8px; margin-t
    input[type="submit"], input[type="reset"] { padding: 10px; width: 49%; margin-top: 20px;
</style>
</head>
<body>
<div class="container">
    <h2>Signup Form</h2>
    <form>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
```



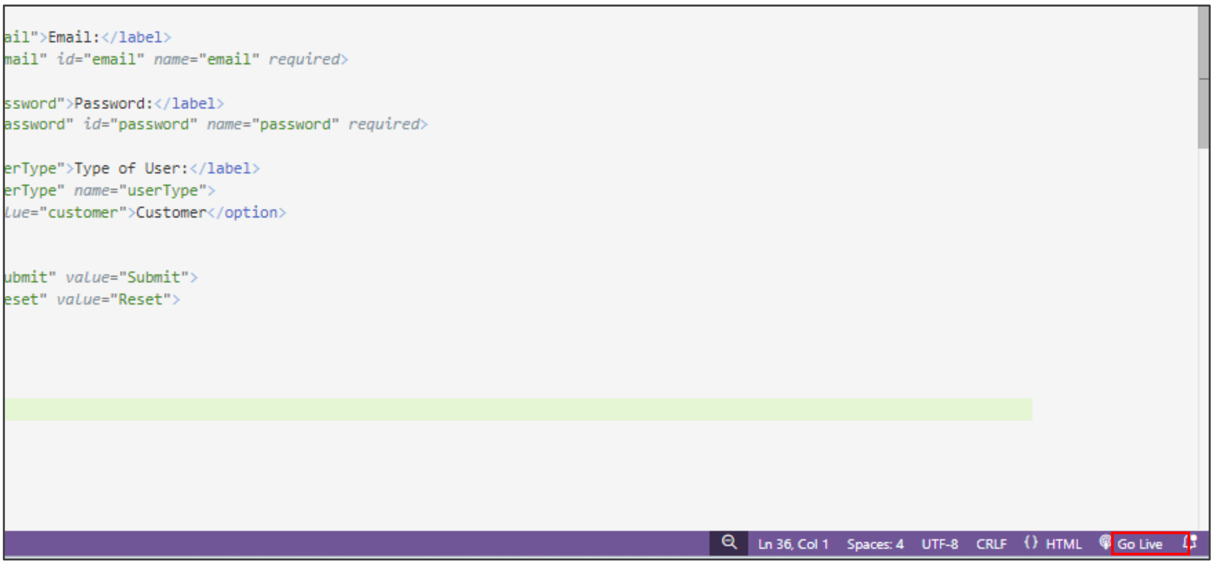
```
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>

        <label for="userType">Type of User:</label>
        <select id="userType" name="userType">
            <option value="customer">Customer</option>
        </select>

        <input type="submit" value="Submit">
        <input type="reset" value="Reset">
    </form>
</div>
</body>
</html>
```

1.8 Create a **signup.html** page in the same folder in VS Code and paste the generated HTML code from ChatGPT into it

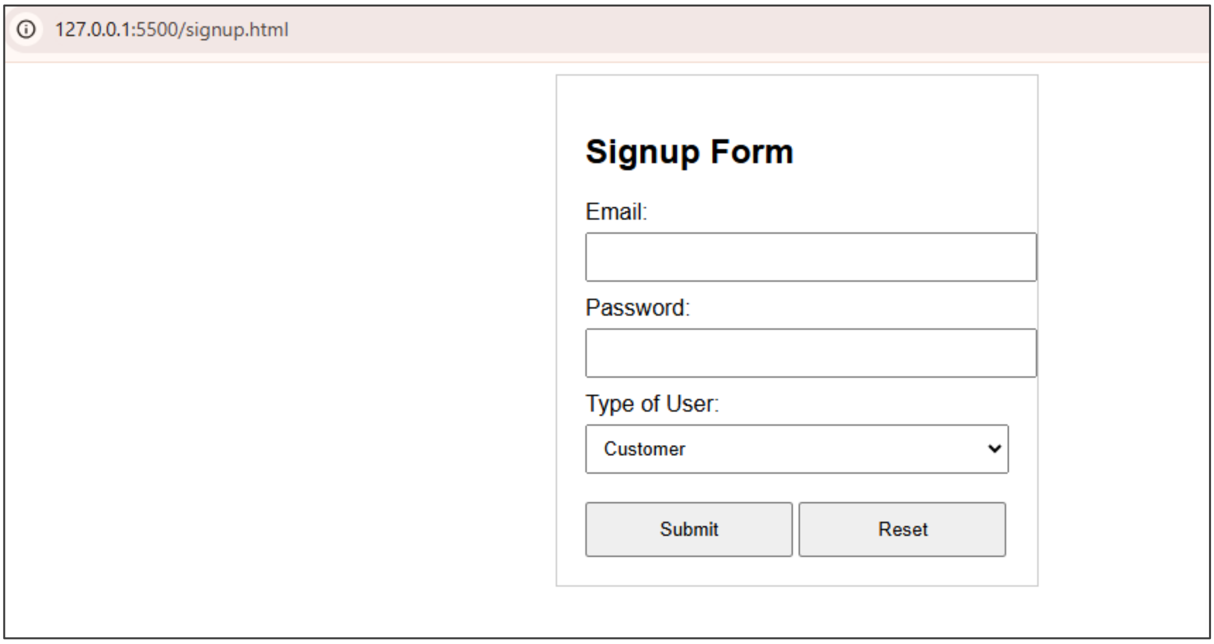


```
<html lang="en">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Signup Page</title>
<style>
    body { font-family: Arial, sans-serif; }
    .container { max-width: 300px; margin: auto; padding: 20px; border: 1px solid #ccc; }
    label { display: block; margin-top: 10px; }
    input[type="email"], input[type="password"], select { width: 100%; padding: 8px; margin-top: 5px; }
    input[type="submit"], input[type="reset"] { padding: 10px; width: 49%; margin-top: 20px; }
</style>
</head>
<body>
<div class="container">
    <h2>Signup Form</h2>
    <form>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>

        <label for="userType">Type of User:</label>
        <select id="userType" name="userType">
            <option value="customer">Customer</option>
        </select>

        <input type="submit" value="Submit">
        <input type="reset" value="Reset">
    </form>
</div>
</body>
</html>
```

1.9 Click on **Go Live** to execute the code



```
ail">Email:</label>
mail" id="email" name="email" required>

ssword">Password:</label>
assword" id="password" name="password" required>

erType">Type of User:</label>
erType" name="userType">
lue="customer">Customer</option>


ubmit" value="Submit">
eset" value="Reset">
```



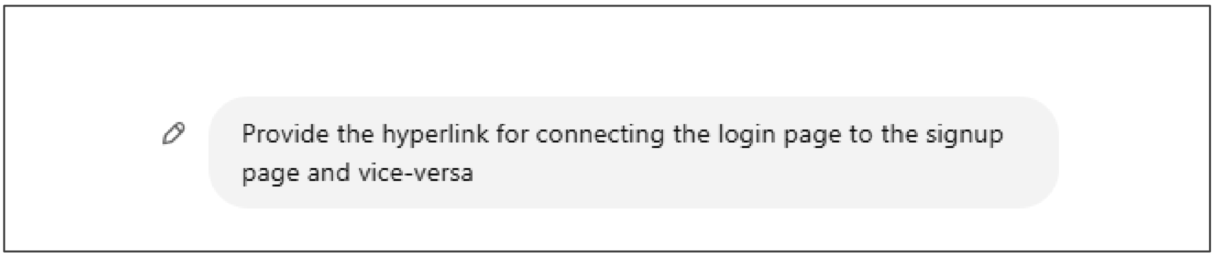
Signup Form

Email:

Password:

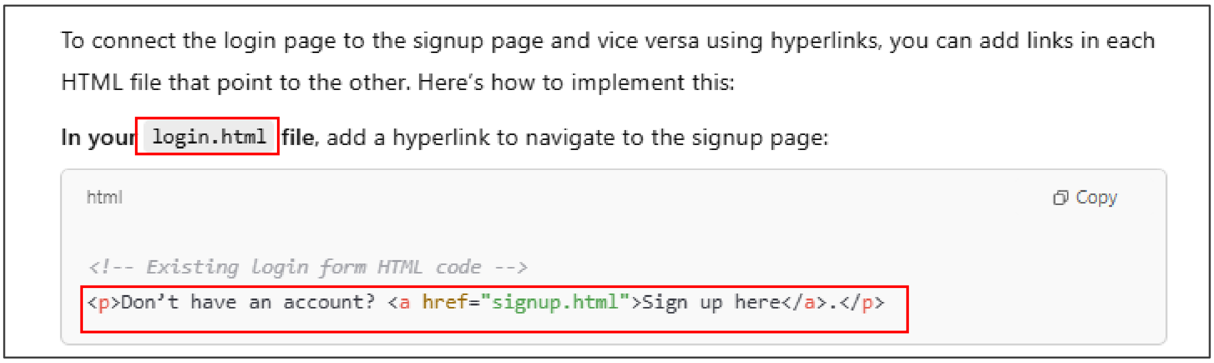
Type of User:

1.10 Enter the following prompt to generate code for connecting the created pages: **Provide the hyperlink for connecting the login page to the signup page and vice-versa**



Provide the hyperlink for connecting the login page to the signup page and vice-versa

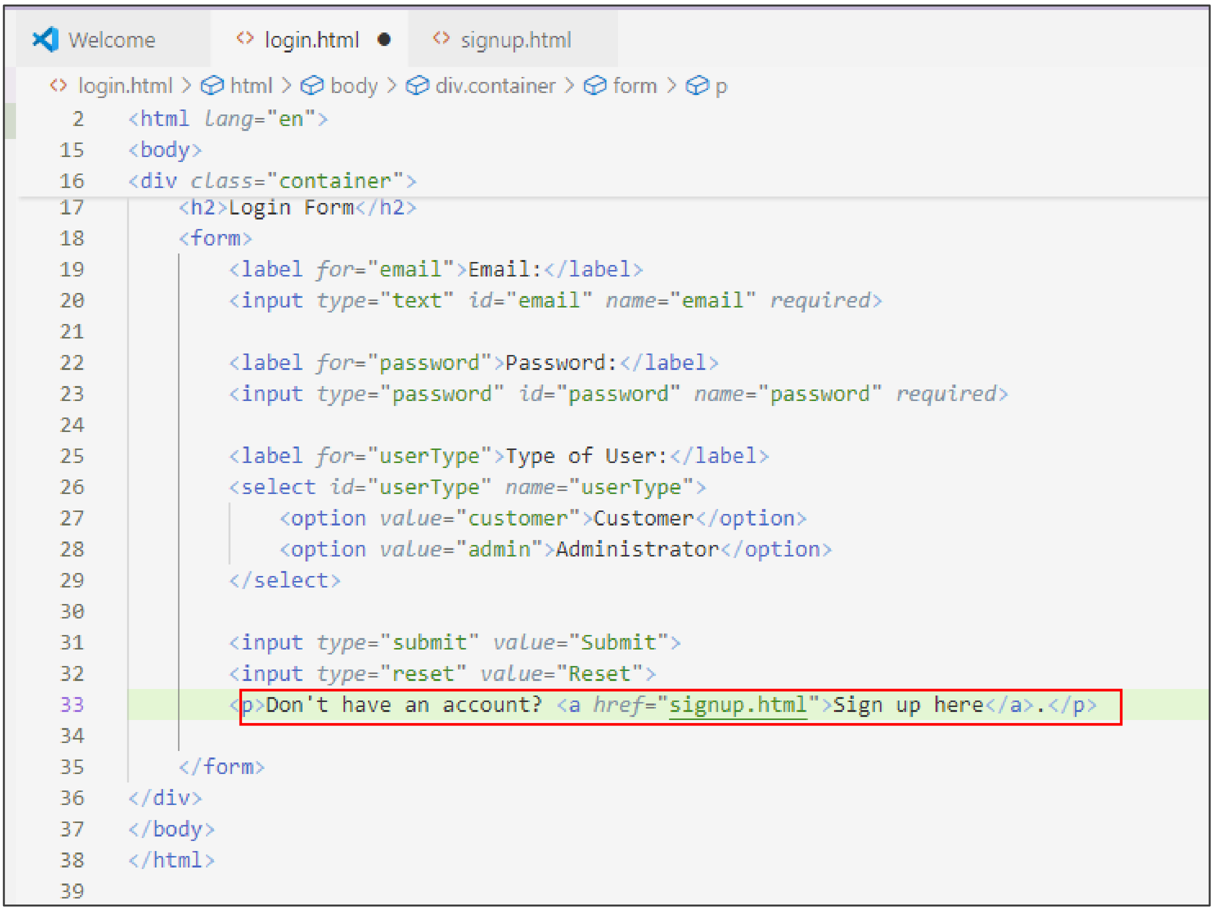
1.11 Copy the code for login.html and paste it into the file just above the <form> tag



To connect the login page to the signup page and vice versa using hyperlinks, you can add links in each HTML file that point to the other. Here's how to implement this:

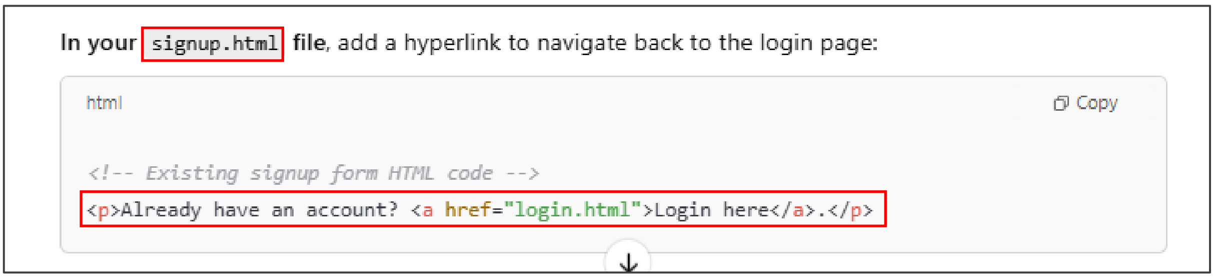
In your `login.html` **file**, add a hyperlink to navigate to the signup page:

```html
<!-- Existing login form HTML code -->
<p>Don't have an account? <a href="signup.html">Sign up here</a>.</p>
```



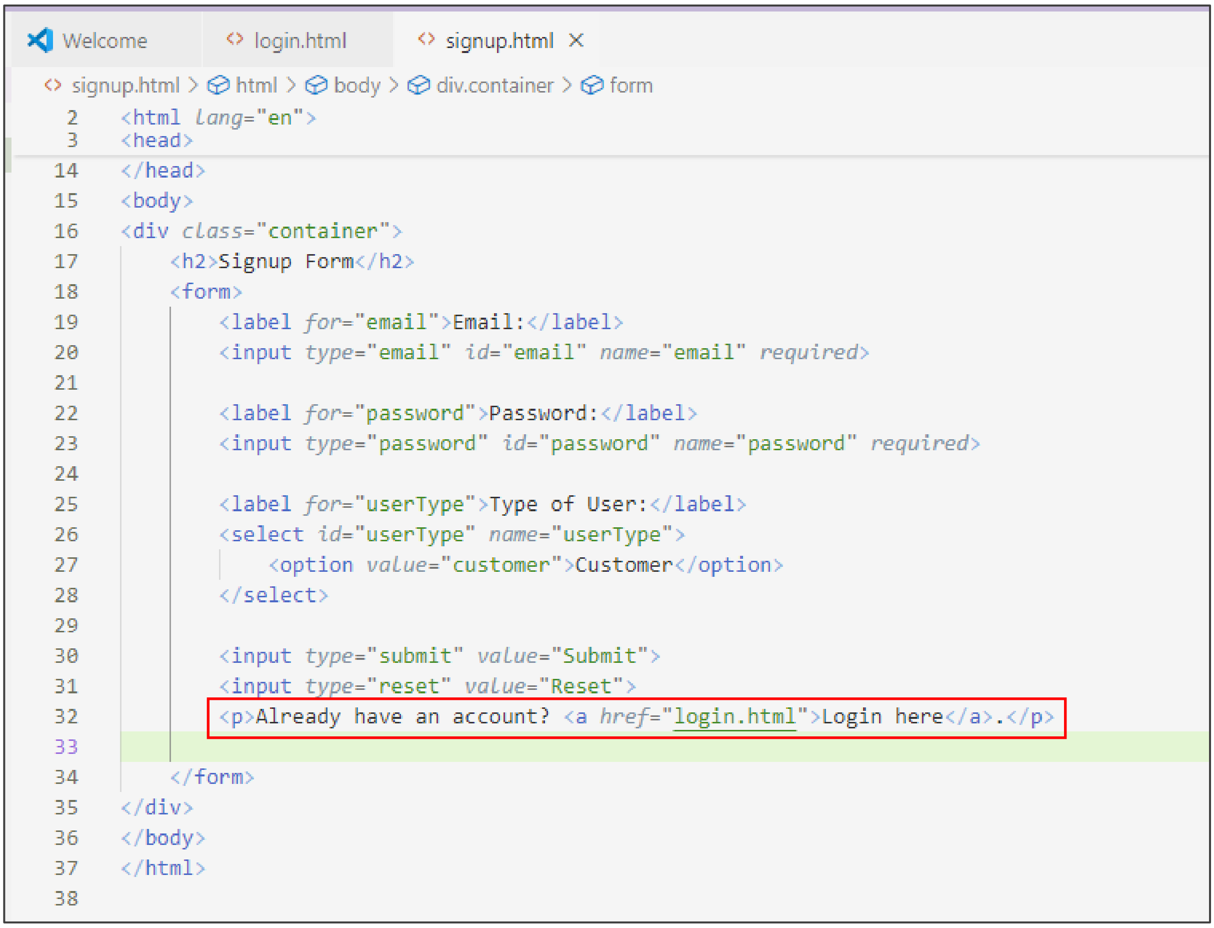
```html
<html lang="en">
<body>
<div class="container">
    <h2>Login Form</h2>
    <form>
        <label for="email">Email:</label>
        <input type="text" id="email" name="email" required>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>

        <label for="userType">Type of User:</label>
        <select id="userType" name="userType">
            <option value="customer">Customer</option>
            <option value="admin">Administrator</option>
        </select>

        <input type="submit" value="Submit">
        <input type="reset" value="Reset">
        <p>Don't have an account? <a href="signup.html">Sign up here</a>.</p>

    </form>
</div>
</body>
</html>
```

1.12 Copy the code for signup.html and paste it into the file just above the <form> tag



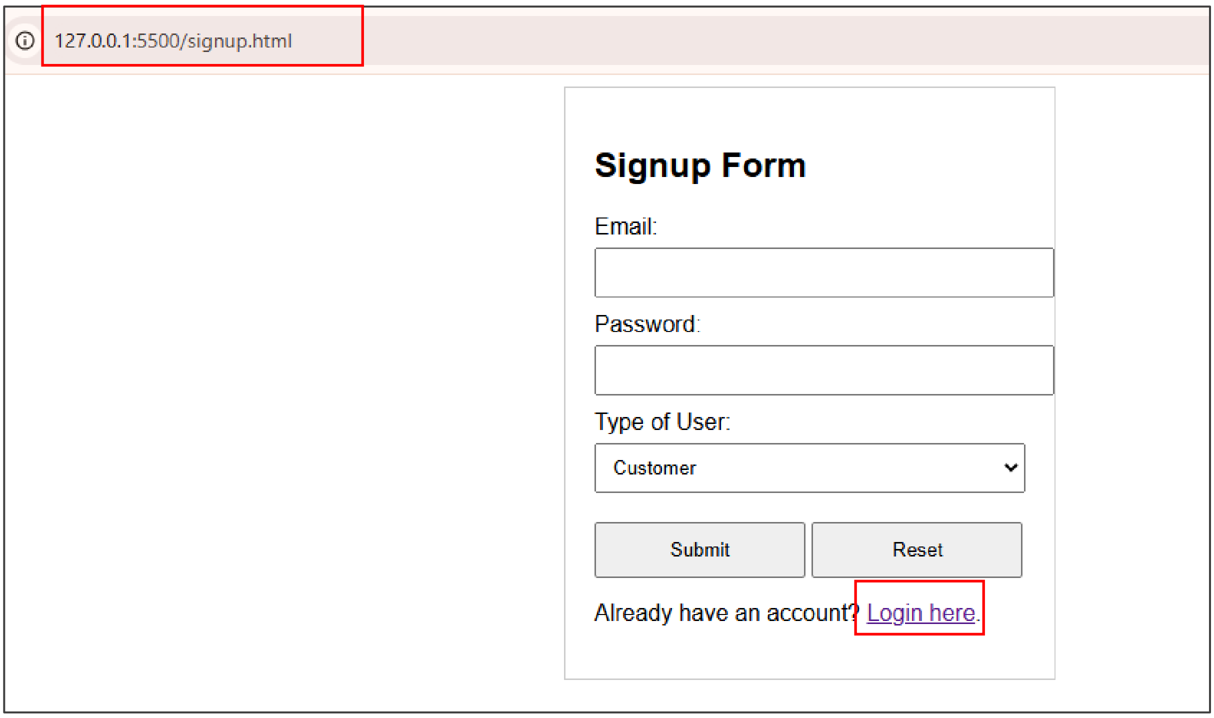
In your `signup.html` **file**, add a hyperlink to navigate back to the login page:

```html
<!-- Existing signup form HTML code -->
<p>Already have an account? <a href="login.html">Login here</a>.</p>
```



```
<html lang="en">
<head>
</head>
<body>
<div class="container">
    <h2>Signup Form</h2>
    <form>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>

        <label for="userType">Type of User:</label>
        <select id="userType" name="userType">
            <option value="customer">Customer</option>
        </select>

        <input type="submit" value="Submit">
        <input type="reset" value="Reset">
        <p>Already have an account? <a href="login.html">Login here</a>.</p>

    </form>
</div>
</body>
</html>

```

1.13 Now, run both pages. Click **Login here.** on the sign-up page and **Sign up here.** on the login page to check if the hyperlinks work.



127.0.0.1:5500/signup.html

Signup Form

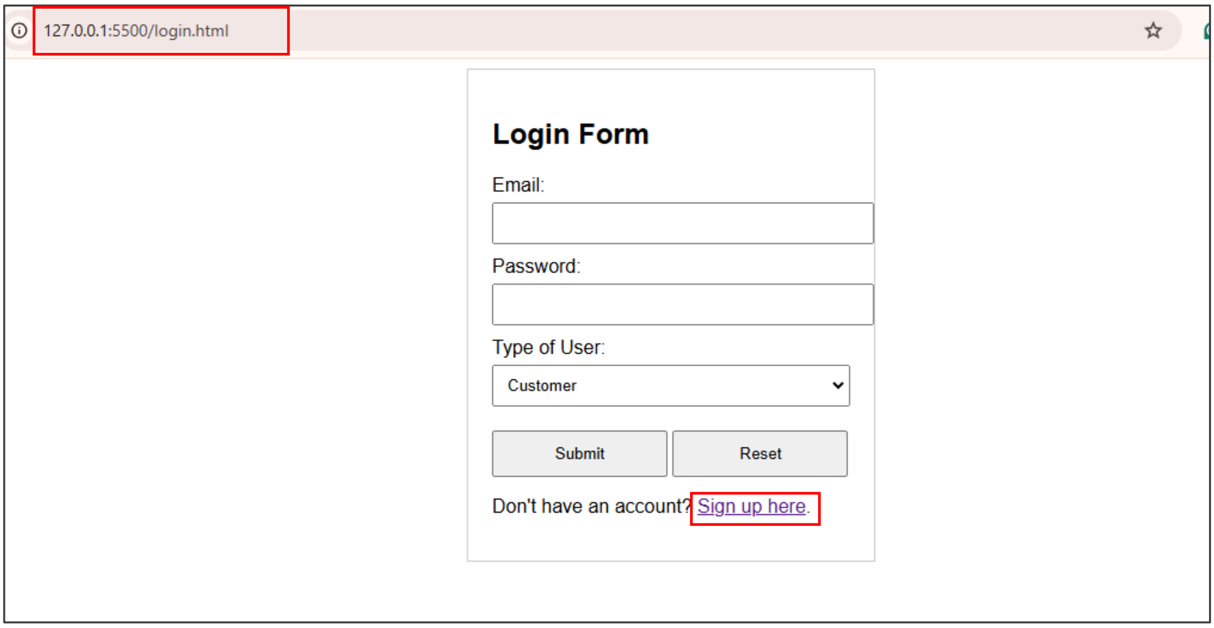
Email:

Password:

Type of User:

Customer

Already have an account? Login here.



Login Form

Email:

Password:

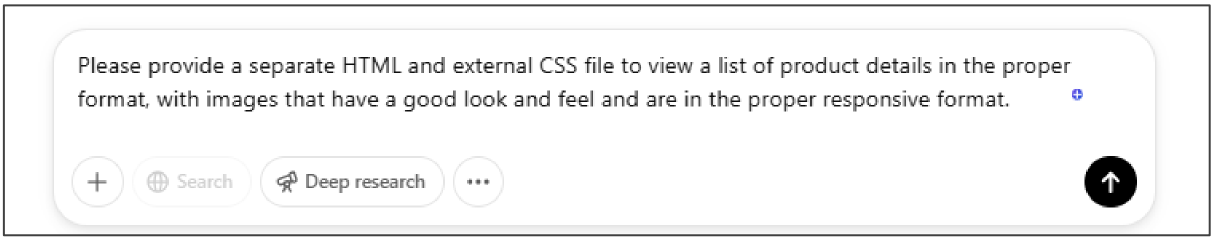
Type of User:

Customer

Don't have an account? Sign up here.

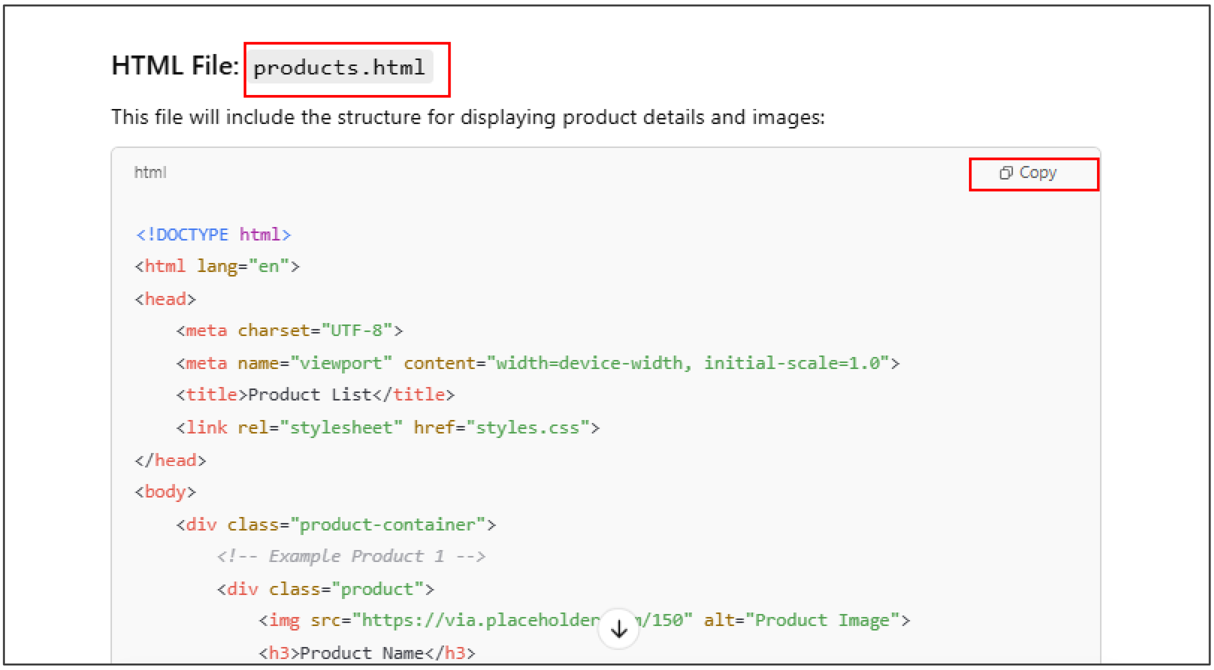
Step 2: Create a product page

2.1 Enter the following prompt in ChatGPT to generate the code for the product page:
Please provide a separate HTML and external CSS file to view a list of product details in the proper format, with images that have a good look and feel and are in the proper responsive format.



Please provide a separate HTML and external CSS file to view a list of product details in the proper format, with images that have a good look and feel and are in the proper responsive format.

2.2 Copy the code for products.html

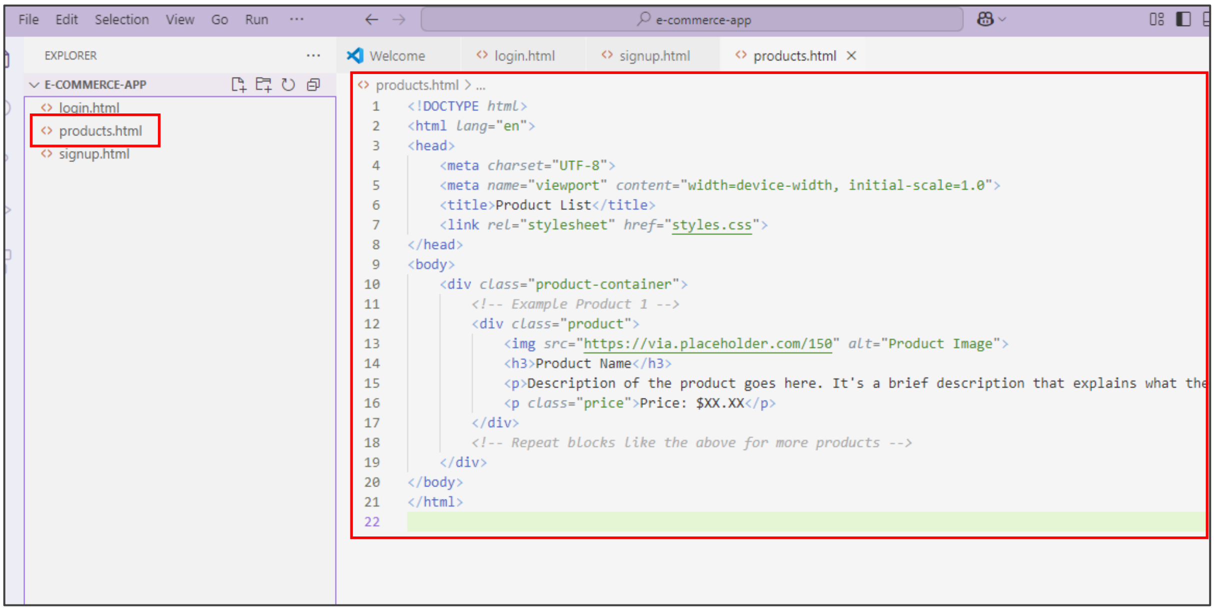


HTML File: `products.html`

This file will include the structure for displaying product details and images:

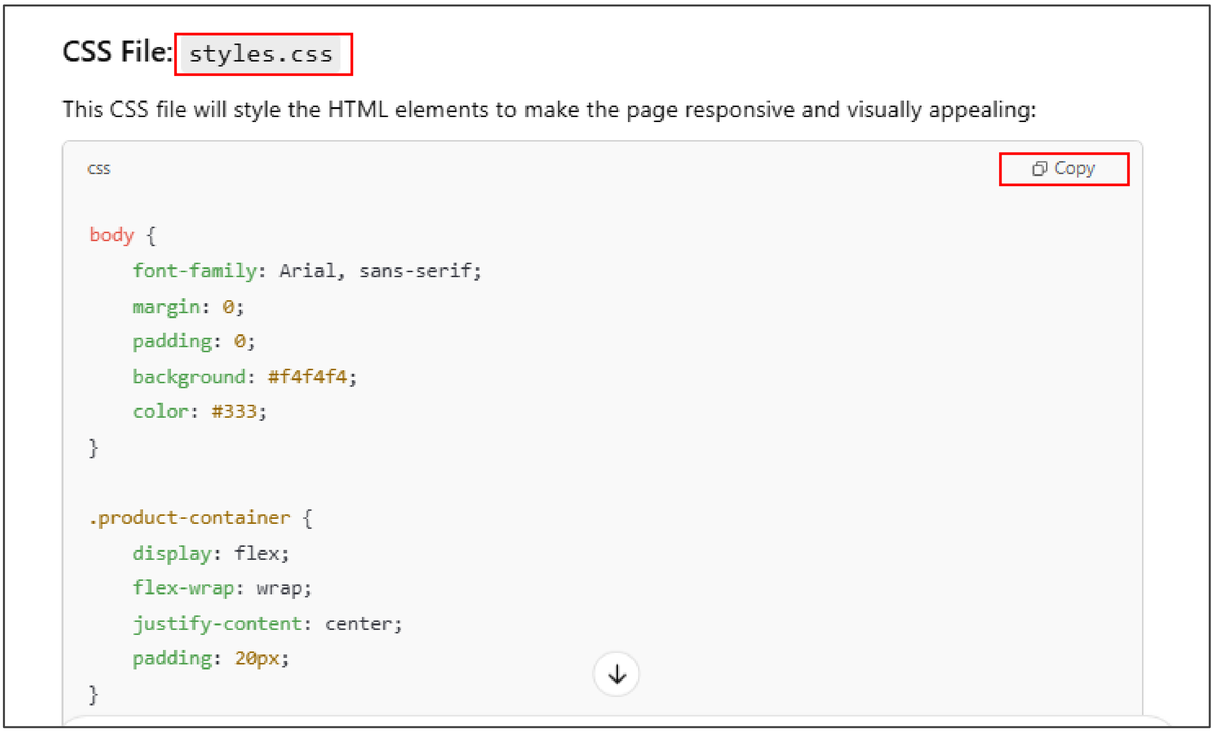
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Product List</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="product-container">
        <!-- Example Product 1 -->
        <div class="product">
            <img src="https://via.placeholder /150" alt="Product Image">
            <h3>Product Name</h3>
```

2.3 Create a file named **products.html** in VS code in the E-COMMERCE-APP folder and paste the code into it



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Product List</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="product-container">
        <!-- Example Product 1 -->
        <div class="product">
            <img src="https://via.placeholder.com/150" alt="Product Image">
            <h3>Product Name</h3>
            <p>Description of the product goes here. It's a brief description that explains what the
            <p class="price">Price: $XX.XX</p>
        </div>
        <!-- Repeat blocks like the above for more products -->
    </div>
</body>
</html>

```

2.4 Copy the code for styles.css

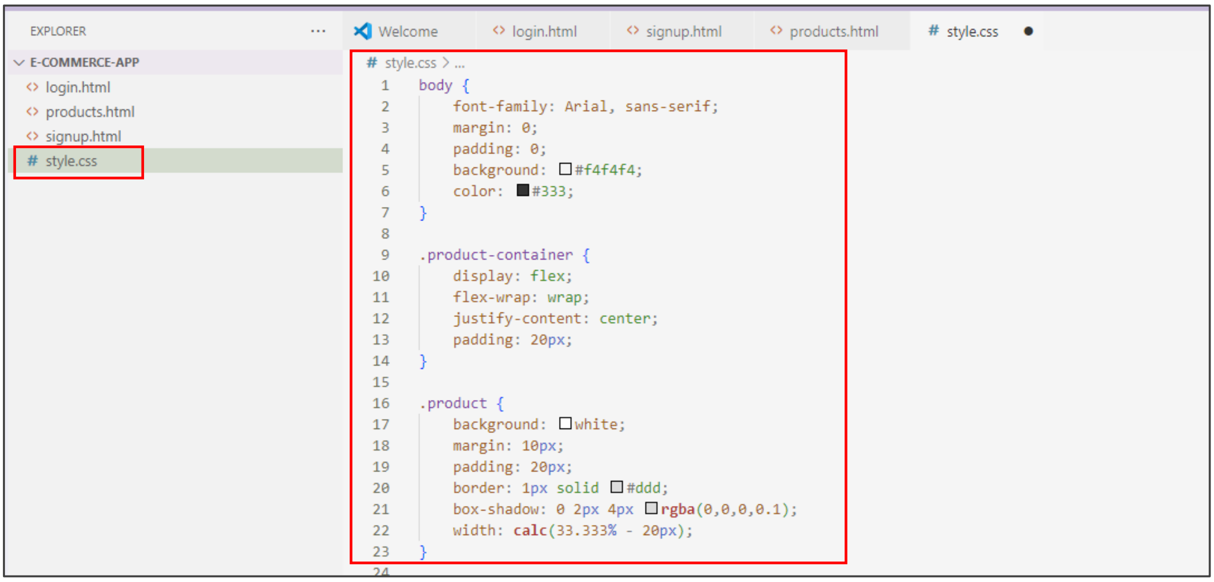


CSS File: `styles.css`

This CSS file will style the HTML elements to make the page responsive and visually appealing:

```css
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background: #f4f4f4;
    color: #333;
}

.product-container {
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
    padding: 20px;
}
```

2.5 Create a new file named **style.css** in the E-COMMERCE-APP folder and paste the code into it

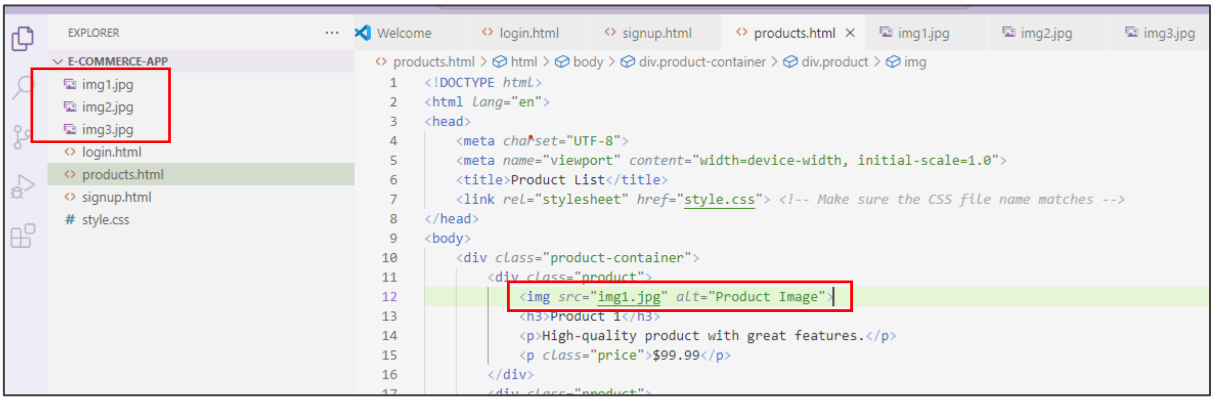


```css
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background: #f4f4f4;
    color: #333;
}

.product-container {
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
    padding: 20px;
}

.product {
    background: white;
    margin: 10px;
    padding: 20px;
    border: 1px solid #ddd;
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
    width: calc(33.333% - 20px);
}
```

2.6 Download images from any of the URLs provided below to the E-COMMERCE-APP folder and insert them into the products.html page in an image tag using the command:
<img src=" <image name> " alt="Product Image">

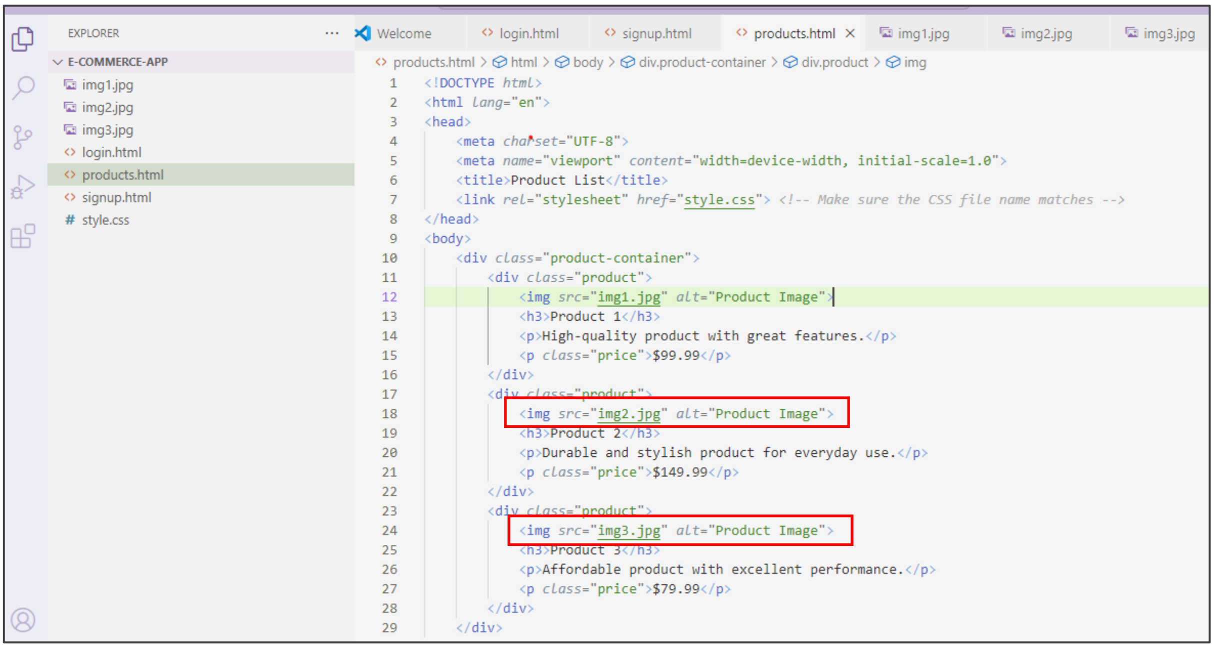
Reference URLs:

a. https://unsplash.com/s/photos/product?utm_source=chatgpt.com
b. https://www.pexels.com/search/product/?utm_source=chatgpt.com
c. https://www.shopify.com/stock-photos?utm_source=chatgpt.com



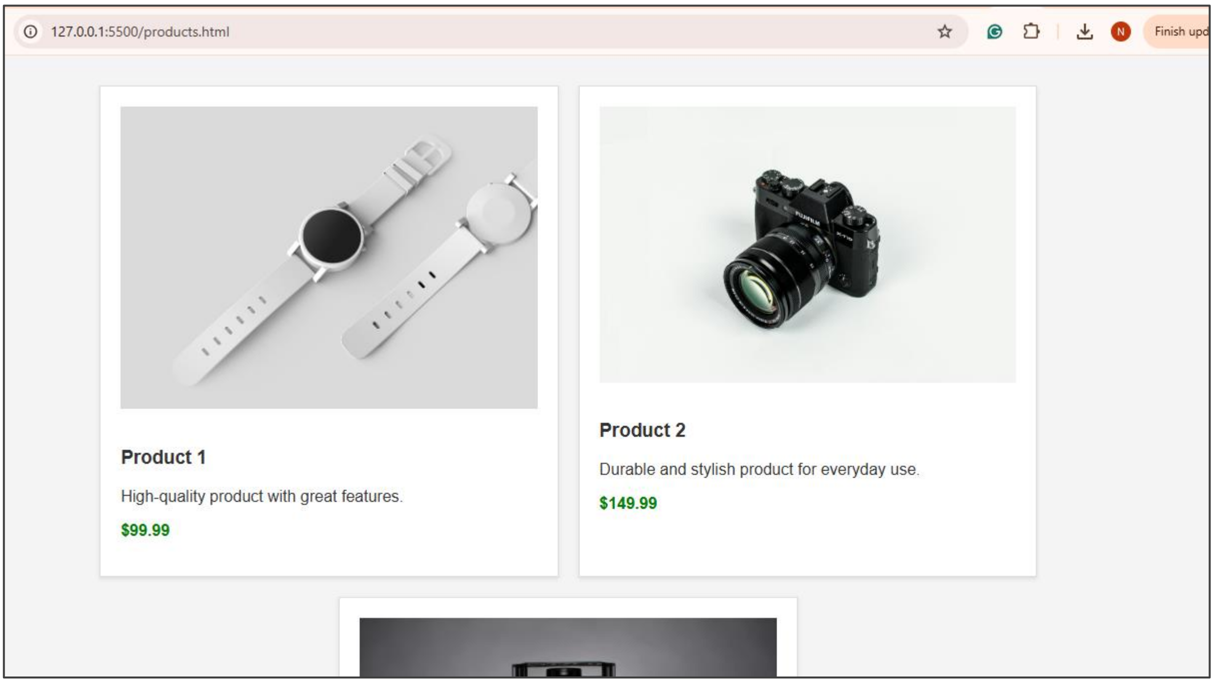
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Product List</title>
    <link rel="stylesheet" href="style.css"> <!-- Make sure the CSS file name matches -->
</head>
<body>
    <div class="product-container">
        <div class="product">
            <img src="img1.jpg" alt="Product Image">
            <h3>Product 1</h3>
            <p>High-quality product with great features.</p>
            <p class="price">$99.99</p>
        </div>
```

2.7 Repeat step 2.6 for the other two products



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Product List</title>
    <link rel="stylesheet" href="style.css"> <!-- Make sure the CSS file name matches -->
</head>
<body>
    <div class="product-container">
        <div class="product">
            <img src="img1.jpg" alt="Product Image">
            <h3>Product 1</h3>
            <p>High-quality product with great features.</p>
            <p class="price">$99.99</p>
        </div>
        <div class="product">
            <img src="img2.jpg" alt="Product Image">
            <h3>Product 2</h3>
            <p>Durable and stylish product for everyday use.</p>
            <p class="price">$149.99</p>
        </div>
        <div class="product">
            <img src="img3.jpg" alt="Product Image">
            <h3>Product 3</h3>
            <p>Affordable product with excellent performance.</p>
            <p class="price">$79.99</p>
        </div>
    </div>
```

2.8 Now, run the products.html page to view the output



Product 1

High-quality product with great features.

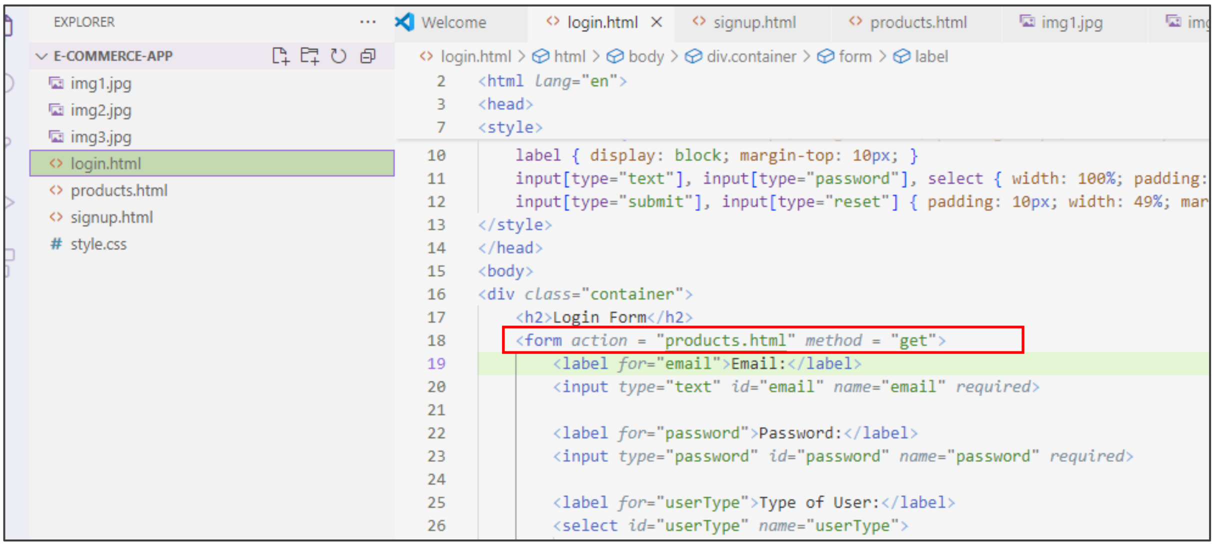
$99.99

Product 2

Durable and stylish product for everyday use.

$149.99

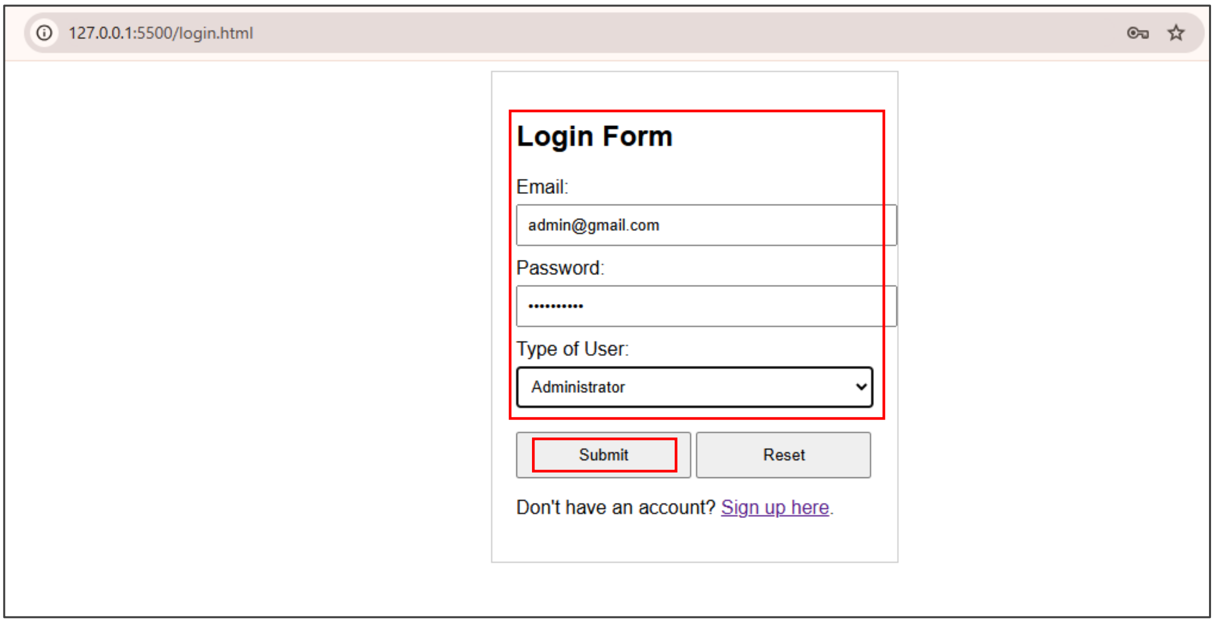
2.9 Link the products.html page to the login.html page using the action attribute in login.html with the command:
<form *action* = "products.html" *method* = "get">



```
<html lang="en">
<head>
<style>
    label { display: block; margin-top: 10px; }
    input[type="text"], input[type="password"], select { width: 100%; padding:
    input[type="submit"], input[type="reset"] { padding: 10px; width: 49%; mar
</style>
</head>
<body>
<div class="container">
    <h2>Login Form</h2>
    <form action = "products.html" method = "get">
        <label for="email">Email:</label>
        <input type="text" id="email" name="email" required>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>

        <label for="userType">Type of User:</label>
        <select id="userType" name="userType">
```

Note: The products.html page will open the products page in the browser after a dummy login. The form uses the GET method, as no server-side technology is being used yet.

2.10 Run the products.html page, fill in the credentials, and click on **Submit**



127.0.0.1:5500/login.html

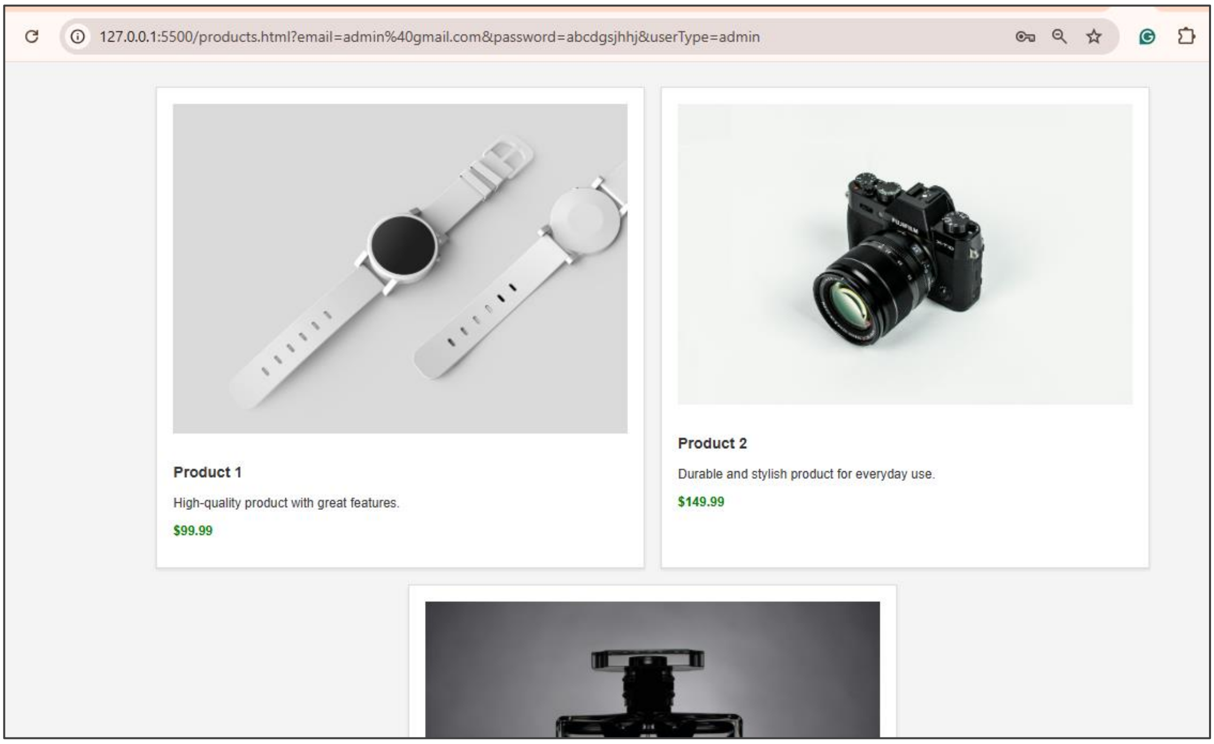
Login Form

Email: admin@gmail.com

Password: ••••••••••

Type of User: Administrator

Don't have an account? Sign up here.



127.0.0.1:5500/products.html?email=admin%40gmail.com&password=abcdgsjhhj&userType=admin

Product 1

High-quality product with great features.

$99.99

Product 2

Durable and stylish product for everyday use.

$149.99

By following these steps, you have successfully created an e-commerce application web page with a login page, sign-up page, and product page, utilizing HTML and CSS to style these pages and incorporate links between them for easy navigation. This setup simulates a basic e-commerce platform, preparing it for further development and integration of server-side technologies.