

Lesson 04 Demo 04

Generating E-Commerce application web page code

Objective: To create web pages with help using html and css with help of chat GPT tool.

Tools required: Chat GPT and Visual code.

Prerequisites: NA

Note: Generative AI tool used in this exercise can produce varied outputs even when presented with similar prompts. Thus, you may get different output for the same prompt.

Steps to be followed:

1. Create the web pages for e-commerce application as SignIn, SignUp, Dashboard etc.
2. Apply styling for web page using internal as well as external CSS.
3. Use Chat GPT prompt to generate the code.

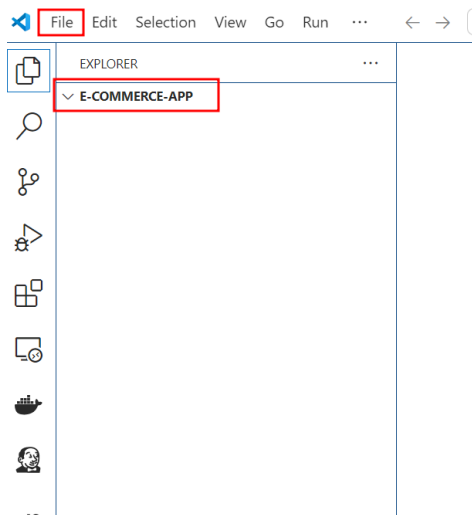
Step 1: Create the Folder as e-commerce-app and open this project in VS code

1.1 Create the folder.

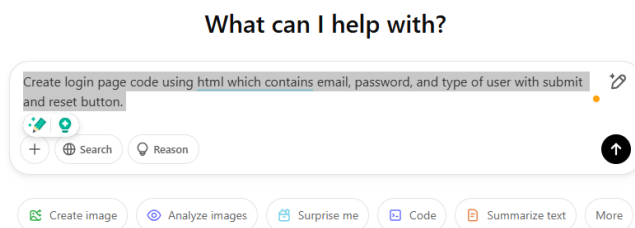


<input type="checkbox"/> Name	Date modified	Type
<input checked="" type="checkbox"/> e-commerce-app	2/17/2025 11:49 AM	File folder

1.2 Open the VS code editor and with help of File option open this Folder in VS code.



1.13. Now open the Chat GPT option and write the prompt as **Create login page code using html which contains email, password, and type of user with submit and reset button.**



1.14 you can see the login page html code will generate.

```

X Login Page
30
31
32 button {
33     background-color: #28a745;
34     color: white;
35     border: none;
36     cursor: pointer;
37 }
38 button:hover {
39     background-color: #218838;
40 }
41 .reset {
42     background-color: #dc3545;
43 }
44 .reset:hover {
45     background-color: #c82333;
46 }
47
48 </style>
49 </head>
50 <body>
51 <div class="login-container">
52 <h2>Login</h2>
53 <form action="#" method="post">
54 <input type="email" name="email" placeholder="Enter Email" required>
55 <input type="password" name="password" placeholder="Enter Password" required>
56 <select name="user_type" required>
57 <option value="">Select User Type</option>
58 <option value="admin">Admin</option>
59 <option value="user">User</option>
60 </select>
61 <button type="submit">Login</button>
62 <button type="reset" class="reset">Reset</button>
63 </form>
64 </div>
65 </body>
66 </html>
```

Here Chat GPT generate the html code for you with internal CSS styling for that web page.

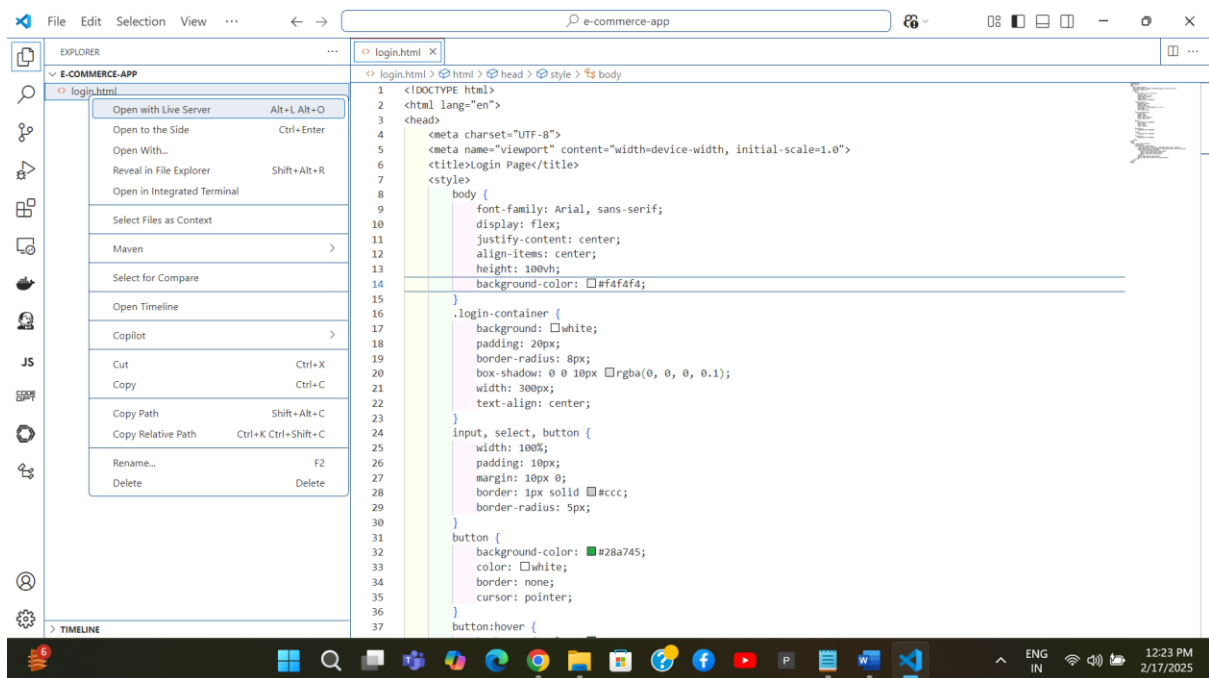
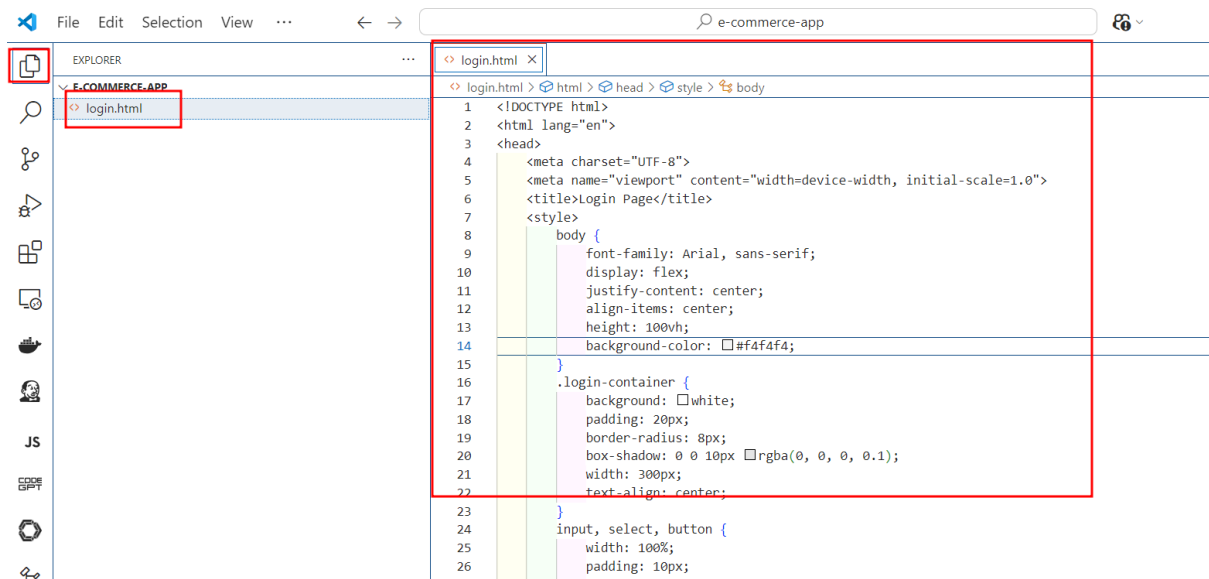
1.15: Now in VS code create the login.html page and copy and paste the code

```
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>Login Page</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      display: flex;
11      justify-content: center;
12      align-items: center;
13      height: 100vh;
14      background-color: #f4f4f4;
15    }
16    .login-container {
17      background: white;
18      padding: 20px;
19      border-radius: 8px;
20      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
21      width: 300px;
22      text-align: center;
23    }
24    input, select, button {
25      width: 100%;
26      padding: 10px;
27      margin: 10px 0;
28      border: 1px solid #ccc;
29      border-radius: 5px;
30    }
31    button {
32      background-color: #28a745;
33      color: white;
34      border: none;
35      cursor: pointer;
36    }
```

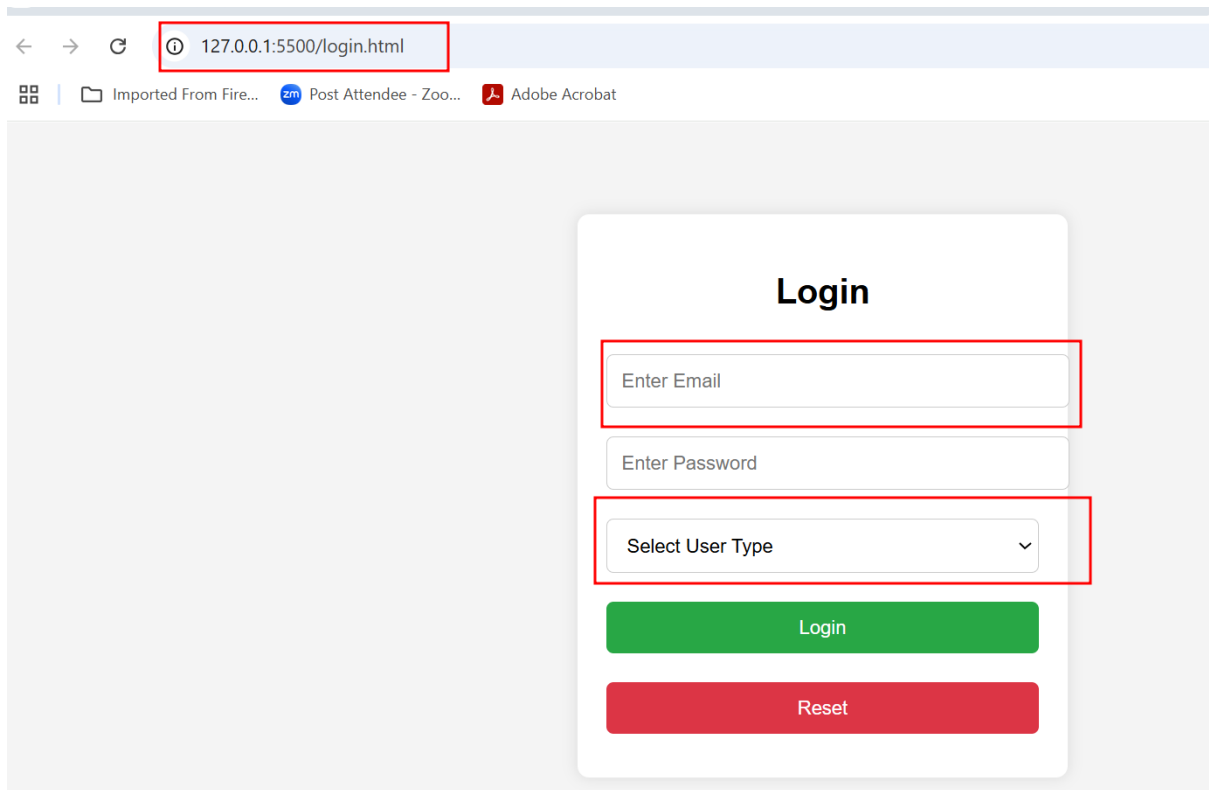
1.16 Now you need to run this page using VS code run option. There are many options present in VS code to run the html page. If you are not getting the option to run the page Please installed Liver reload features which help to run the web page on browser.

The screenshot shows the VS Code interface with the 'Extensions: Marketplace' view on the left. The 'Live Server' extension by Ritwick Dey is highlighted. The main editor shows the 'Live Server' extension details page, which includes a description, a 'Launch a development local Server with live reload feature for static & dynamic pages' button, and a 'Details' tab. The 'Details' tab shows the extension's version (v5.7.9), download count (60M), and rating (4.5 stars). It also includes a link to the GitHub repository and a 'Launch a local development server with live reload feature for static & dynamic pages' button.

1.17 After installed you can run this page on server.

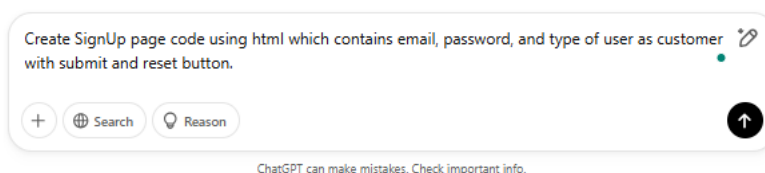


By default live server run on default port number 5500



1.18. Now we will create the Signup.html page

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



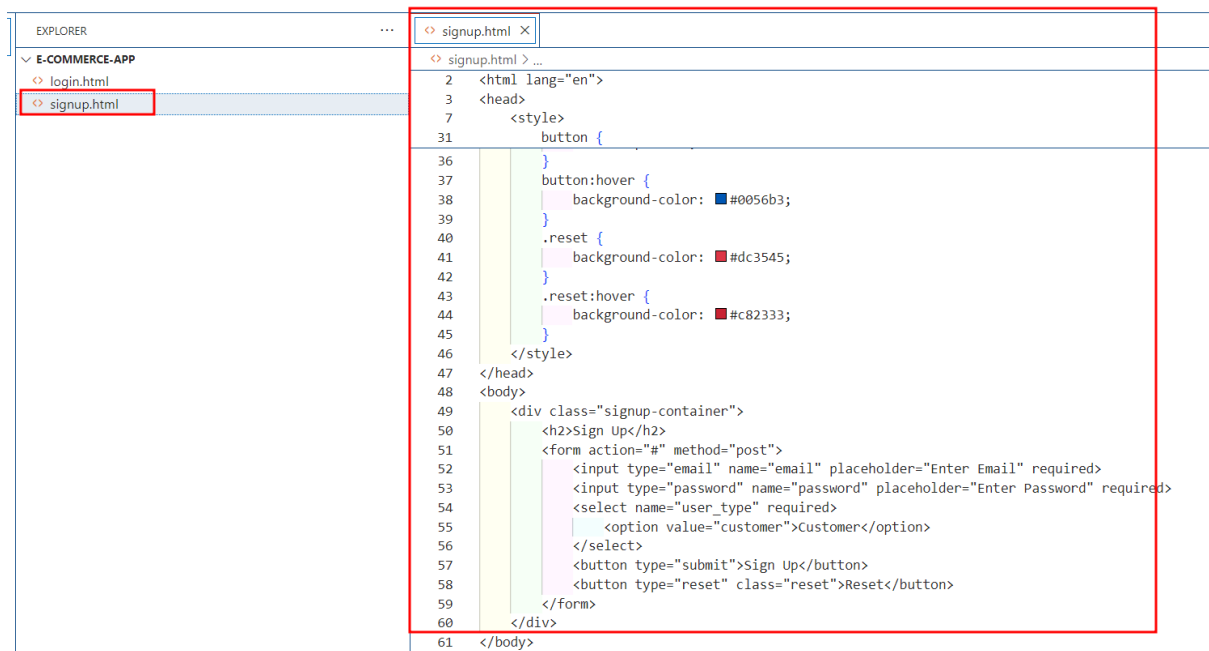
1.19 You can see the output generate as

```

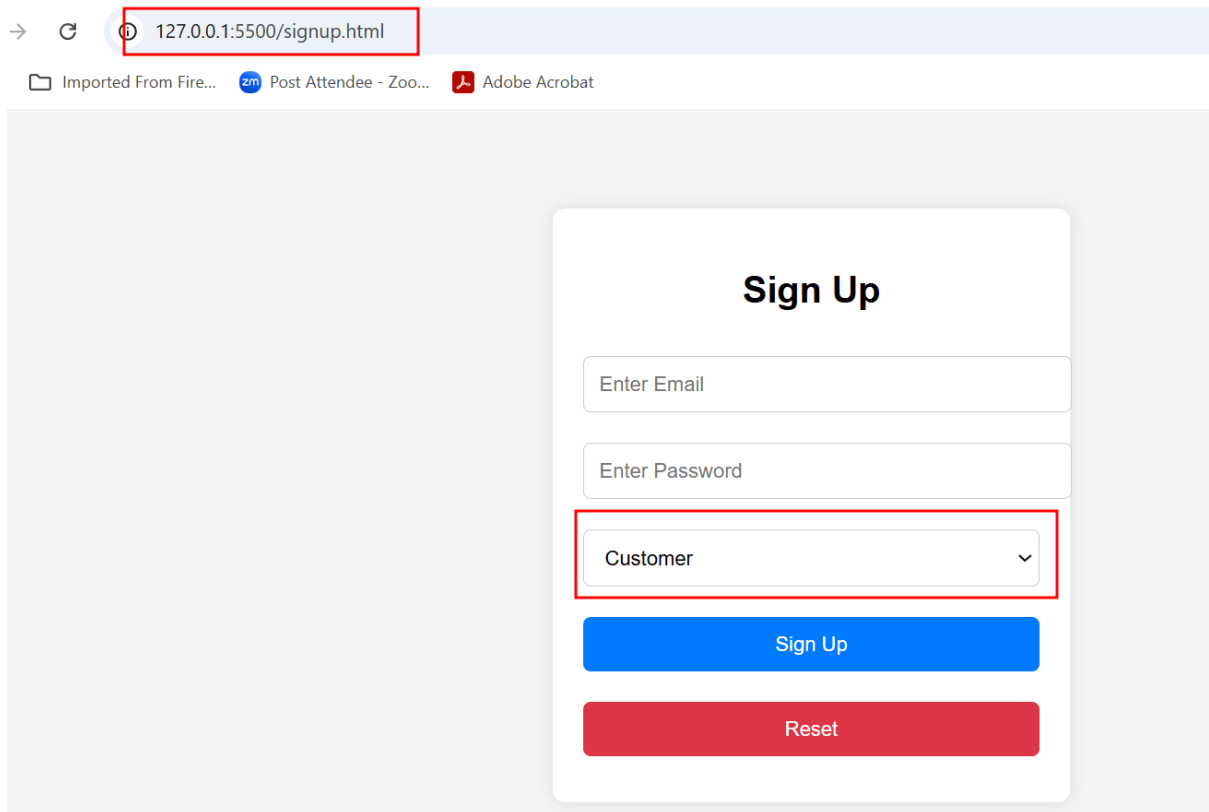
1         background-color: #dc3545;
2     }
3     .reset:hover {
4         background-color: #c82333;
5     }
6 }
7 </style>
8 </head>
9 <body>
10    <div class="signup-container">
11        <h2>Sign Up</h2>
12        <form action="#" method="post">
13            <input type="email" name="email" placeholder="Enter Email" required>
14            <input type="password" name="password" placeholder="Enter Password" required>
15            <select name="user_type" required>
16                <option value="customer">Customer</option>
17            </select>
18            <button type="submit">Sign Up</button>
19            <button type="reset" class="reset">Reset</button>
20        </form>
21    </div>
22 </body>
23 </html>

```

1.20 Now create the signup.html page in Vs code and copy and paste generate html code from chat gpt prompt.

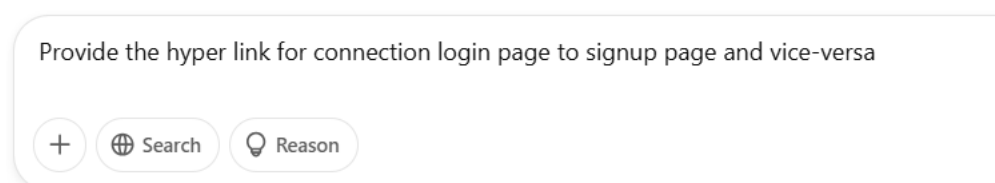


1.21 To test the code run this page.



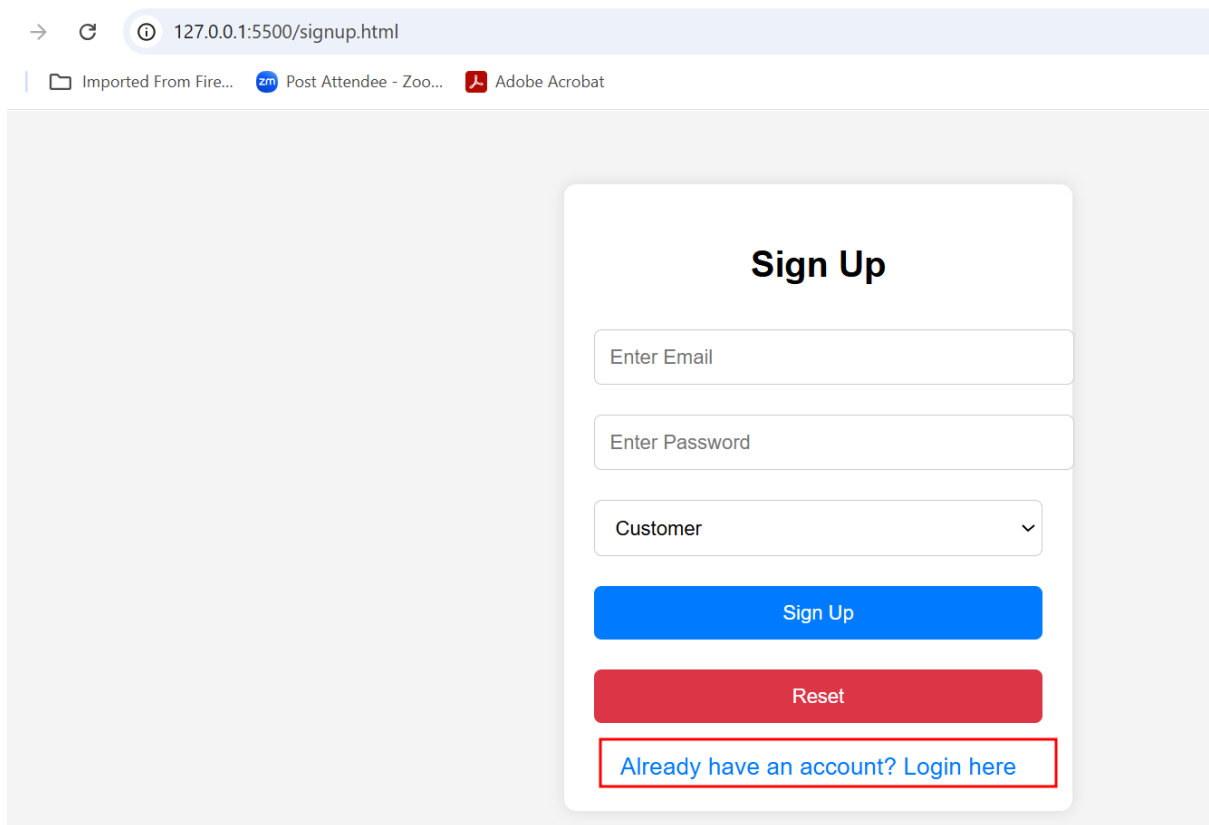
1.22 : Now ask for chat gpt to connect both page.

Provide the hyper link for connection login page to signup page and vice-versa



1.23 replaced new generate code with existing signup page code which provide link to connect login page.

Signup.html page



1.24 : Please take the help of this code with hyperlink tag and few styling effect and which help to connect login page to signup page and test it.

Signup.htm page style code copy and paste in login.html page

```
<> signup.html <> login.html X
<> login.html > html > head > style > .link
2  <html lang="en">
3  <head>
7    <style>
31      button {
35        cursor: pointer;
36      }
37      button:hover {
38        background-color: #218838;
39      }
40      .reset {
41        background-color: #dc3545;
42      }
43      .reset:hover {
44        background-color: #c82333;
45      }
46      .link {
47        display: block;
48        margin-top: 10px;
49        color: #007bff;
50        text-decoration: none;
51      }
52      .link:hover {
53        text-decoration: underline;
54      }
55    </style>
56  </head>
57  <body>
58    <div class="login-container">
59      <h2>Login</h2>
60      <form action="#" method="post">
```

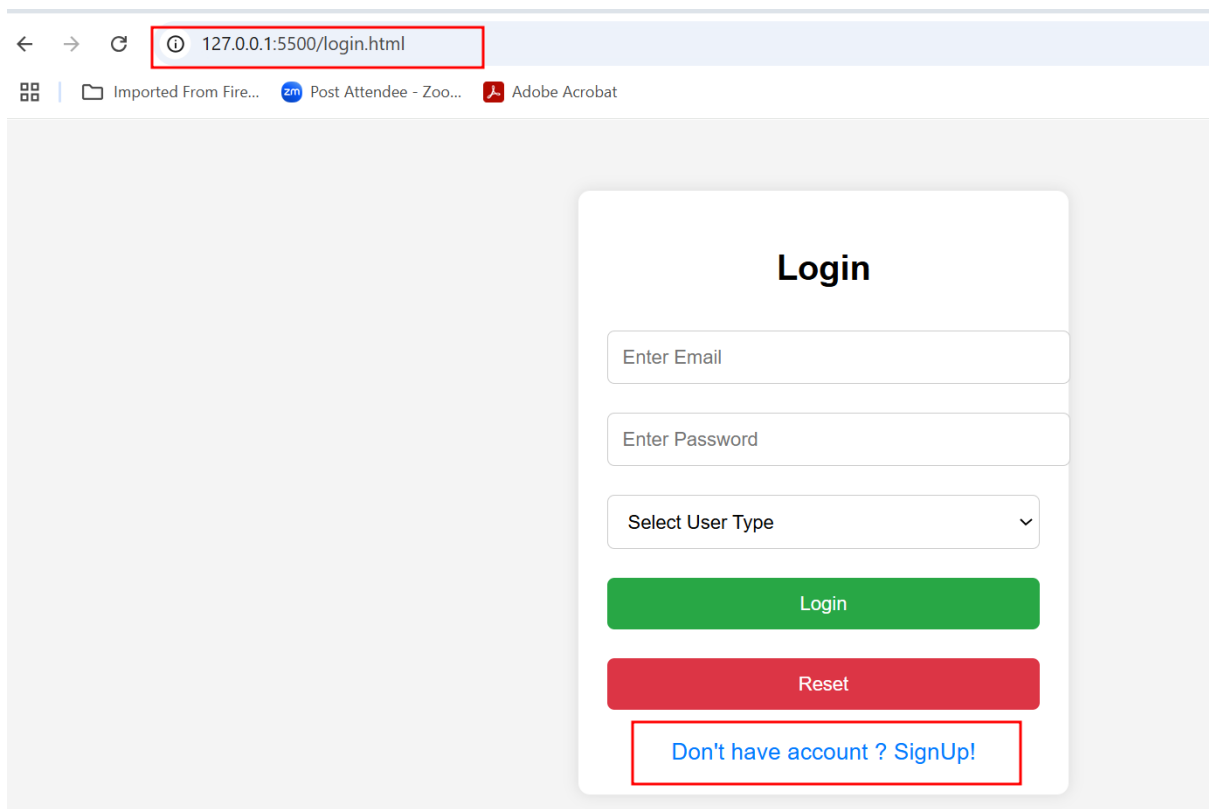
Signup.html page hyperlink code copy and paste in login.html page and change the href path and message.

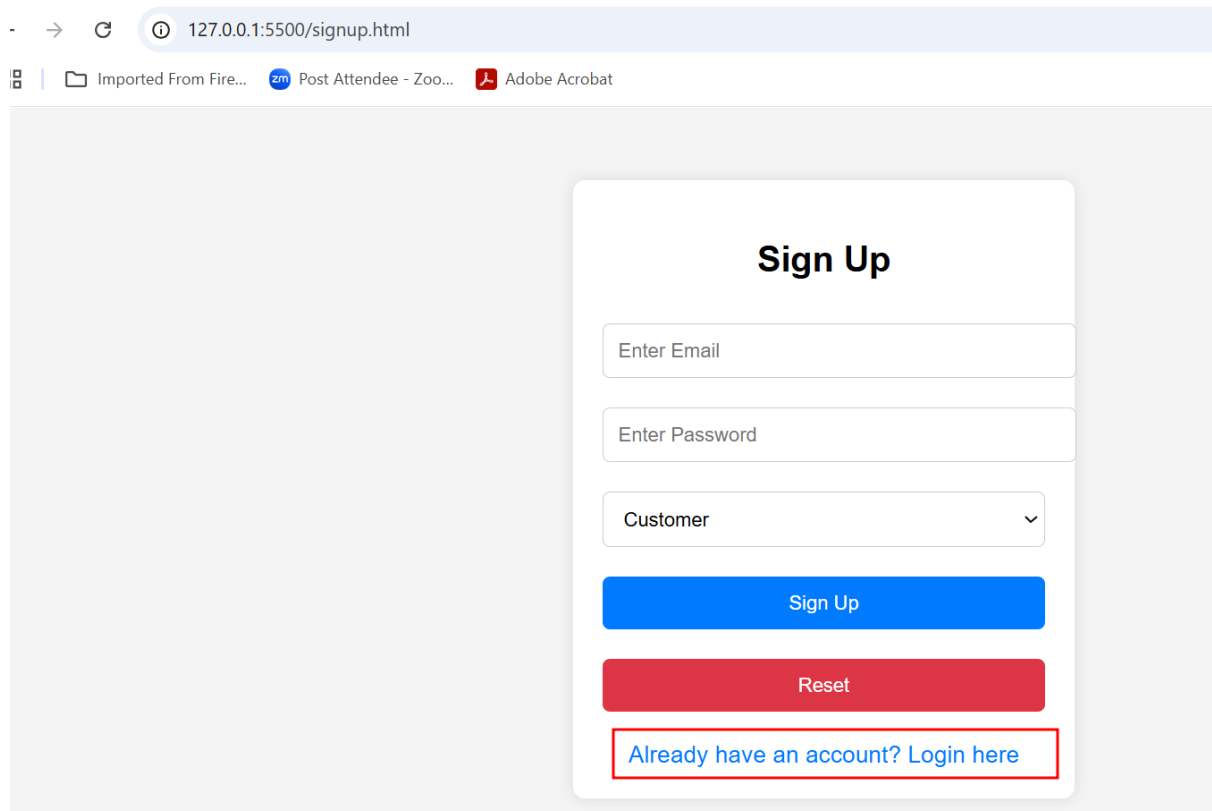
```

56 </style>
57 </head>
58 <body>
59 <div class="login-container">
60 <h2>Login</h2>
61 <form action="#" method="post">
62 <input type="email" name="email" placeholder="Enter Email" required>
63 <input type="password" name="password" placeholder="Enter Password"
64 <select name="user_type" required>
65 <option value="">Select User Type</option>
66 <option value="admin">Admin</option>
67 <option value="user">User</option>
68 </select>
69 <button type="submit">Login</button>
70 <button type="reset" class="reset">Reset</button>
71 </form>
72 <a href="signup.html" class="link">Don't have account ? SignUp!</a>
73 </div>
74 </body>
75 </html>
76

```

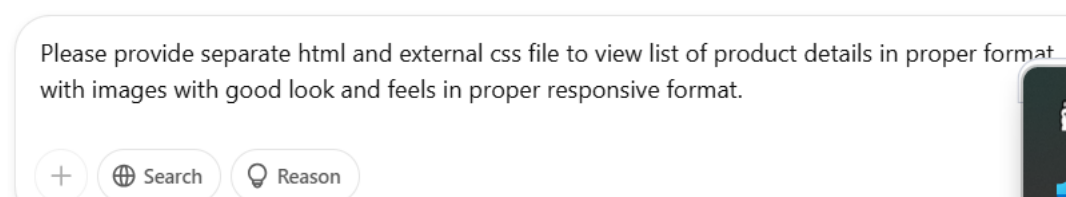
1.25 Now run both the page and test it is working or not.





1.26 Now we will create Produce view page which display list of product.

Please provide separate html and external css file to view list of product details in proper format with images with good look and feels in proper responsive format.



```

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 Details</title>
7   <link rel="stylesheet" href="styles.css">
8 </head>
9 <body>
10 <div class="product-container">
11 <div class="product">
12   
13   <h2>Product Name</h2>
14   <p class="description">This is a great product that provides excellent fe
15   <p class="price">$99.99</p>

```

Answer in chat instead >

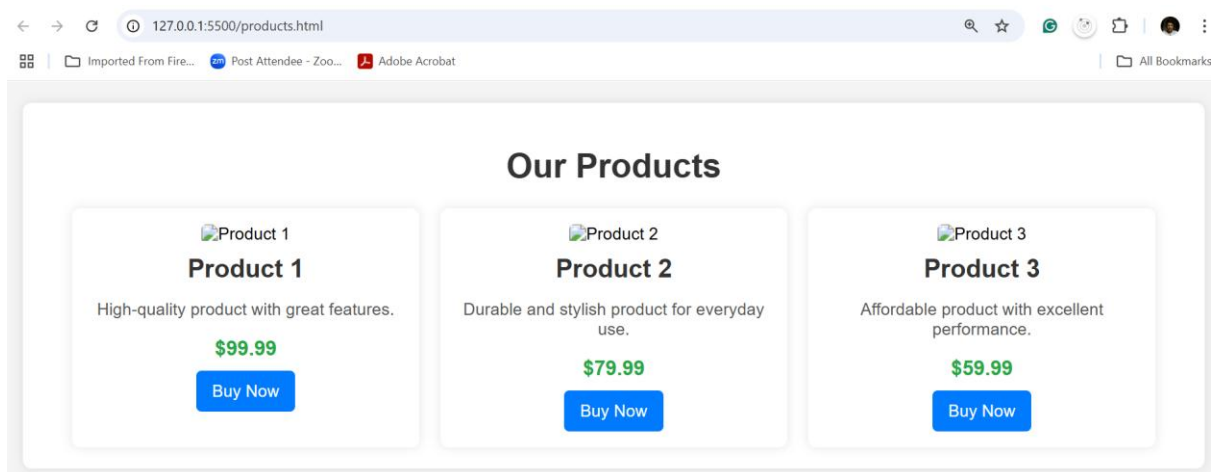
Here's the HTML file for viewing product details. Now, I'll provide the external CSS file to style it properly.

```

</> Styles
1 body {
2   font-family: Arial, sans-serif;

```

1.27: Now create **products.html** page and **styles.css** file and copy and paste the code. Then run the products.html page to view the output.



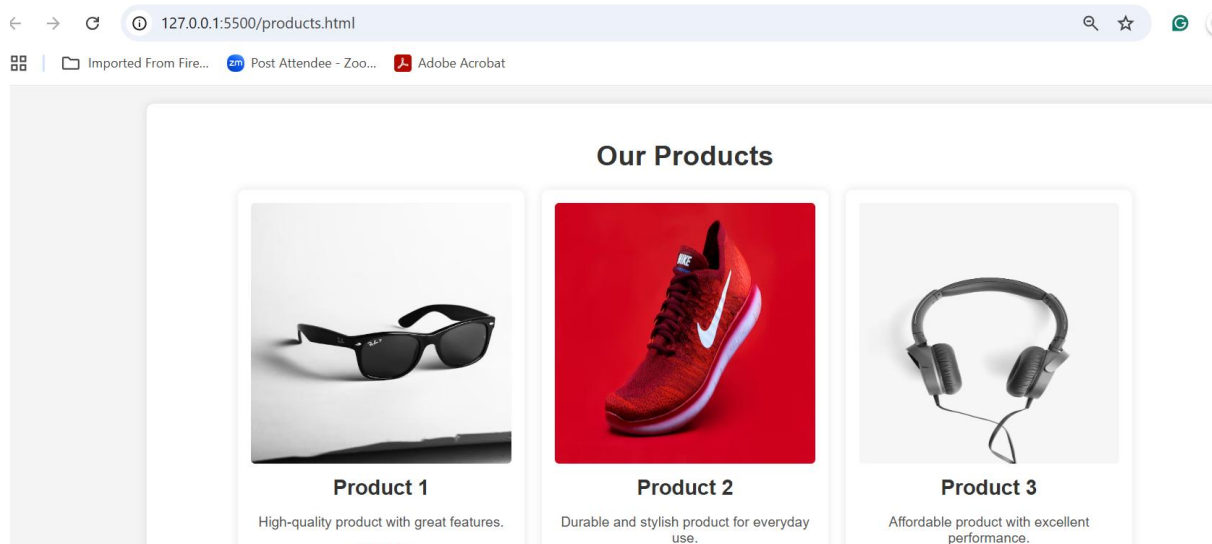
1.28 Refer any online website which provide you image url or else download images in local machine and place in image tag with URL or local image path.

Few reference url as

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

```
products.html > html > body > div.product-list-container > div.product-list > div.product > img
2  en">
3
5  me="viewport" content="width=device-width, initial-scale=1.0">
6  roduct List</title>
7  l="stylesheet" href="styles.css">
8
9
10 ss="product-list-container">
11 Our Products</h1>
12 class="product-list">
13 <div class="product">
14   
17   <h2>Product 1</h2>
18   <p class="description">High-quality product with great features.</p>
19   <p class="price">$99.99</p>
20   <button class="buy-button">Buy Now</button>
21 </div>
22 <div class="product">
23   
26   <h2>Product 2</h2>
27   <p class="description">Durable and stylish product for everyday use.</p>
28   <p class="price">$79.99</p>
29   <button class="buy-button">Buy Now</button>
30 </div>
31 <div class="product">
32   
```

1.29 View once again with updated product details



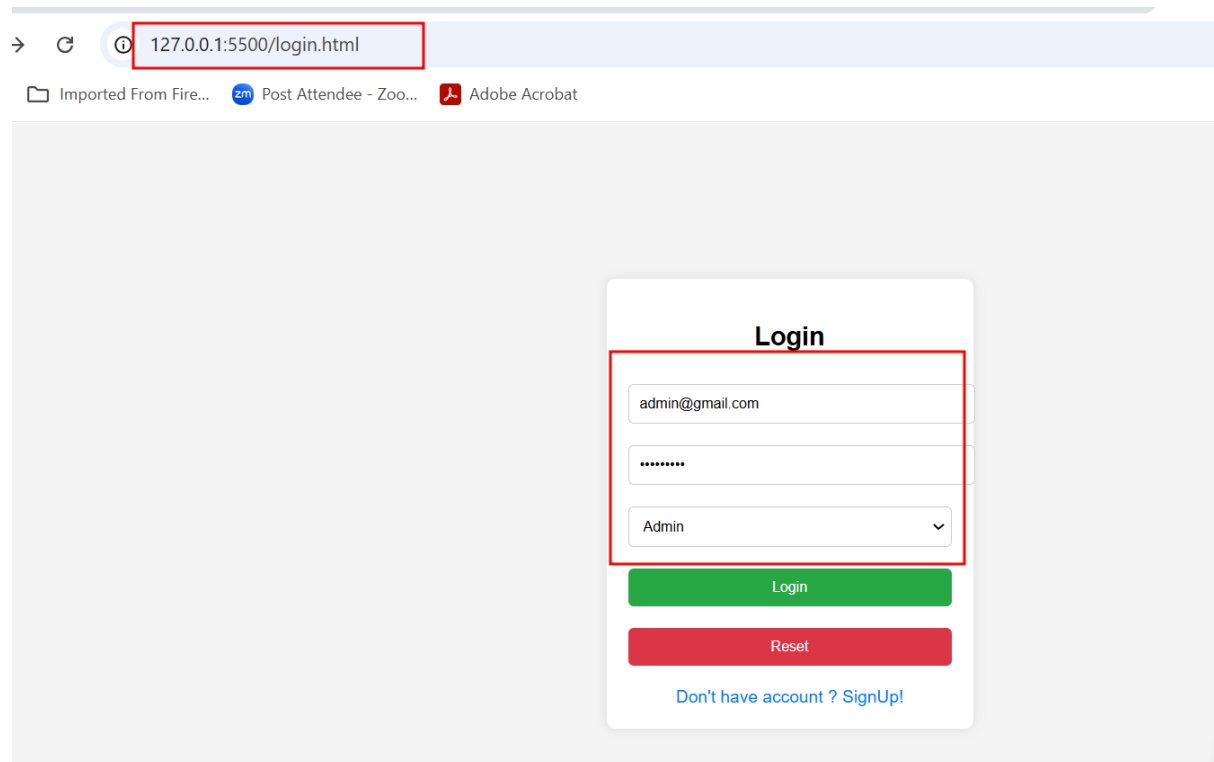
1.30 Now we will connect this page with login.html page. So in action attribute provide the path of this page. Which open after dummy login. Please check the method as get here. Because here we are not using any server side technology yet.

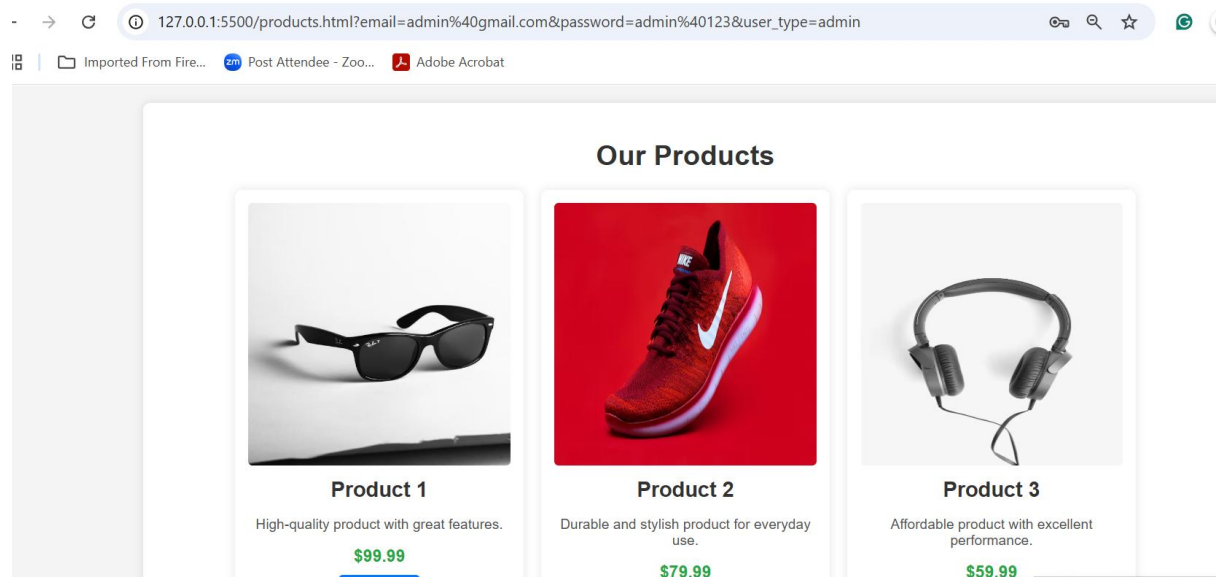
```

</head>
<body>
  <div class="login-container">
    <h2>Login</h2>
    <form action="products.html" method="get">
      <input type="email" name="email" placeholder="Enter Email" required>
      <input type="password" name="password" placeholder="Enter Password" required>
      <select name="user_type" required>
        <option value="">Select User Type</option>
        <option value="admin">Admin</option>
        <option value="user">User</option>
      </select>
      <button type="submit">Login</button>
      <button type="reset" class="reset">Reset</button>
    </form>
    <a href="signup.html" class="link">Don't have account ? SignUp!</a>
  </div>
</body>

```

1.30 Run the login.html page





Here we are not checking email, password and type of user as of now.