

Lesson 04 Demo 04

Generating E-Commerce product module using JavaScript

Objective: To create web pages with help using html, css and JavaScript with help of chat GPT tool which help to add the product and view the product details and display in table format.

Tools required: Chat GPT and Visual code.


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:

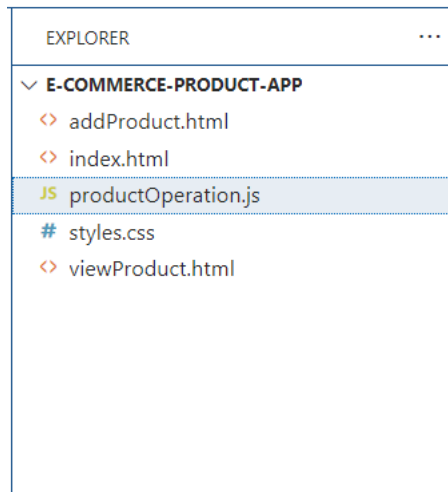
1. Create the web pages for e-commerce application as index.html, addProduct.html and viewProduct.html page
2. Apply styling for web page external CSS.
3. Use Chat GPT prompt to generate the code.

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

1.1 Create the folder.

 e-commerce-product-app

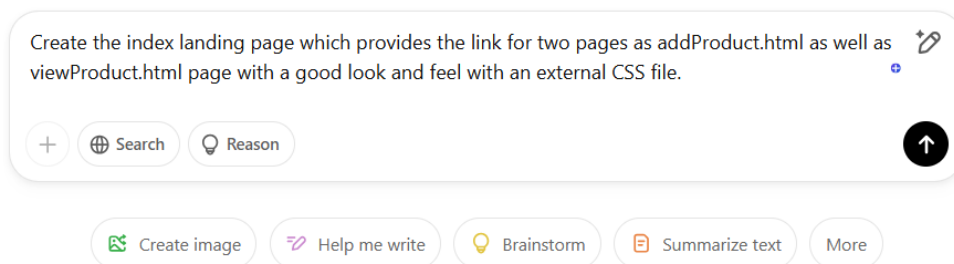
1.2 Open the VS code editor and with help of File option open this Folder in VS code and create **index.html**, **addProduct.html**, **viewProduct.html**, **productOperation.js** and **styles.css** page.



1.2 Open the chat gpt prompt and ask to generate the code for index.html page

Create the index landing page which provides the link for two pages as addProduct.html as well as viewProduct.html page with a good look and feel with an external CSS file.

What can I help with?



Here's the `index.html` file, along with an external `styles.css` file for a clean and modern look.

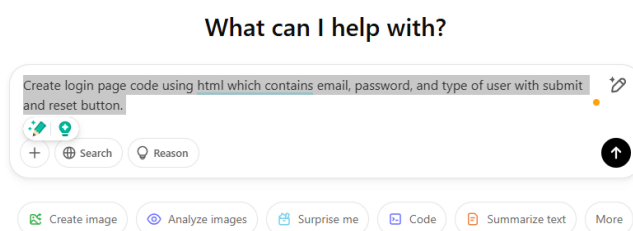
```
</> Index Landing Page

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 Management</title>
7      <link rel="stylesheet" href="styles.css">
8  </head>
9  <body>
10 <div class="container">
11     <h1>Welcome to Product Management</h1>
12     <p>Select an option below:</p>
13     <div class="links">
14         <a href="addProduct.html" class="button">Add Product</a>
15         <a href="viewProduct.html" class="button">View Products</a>
```

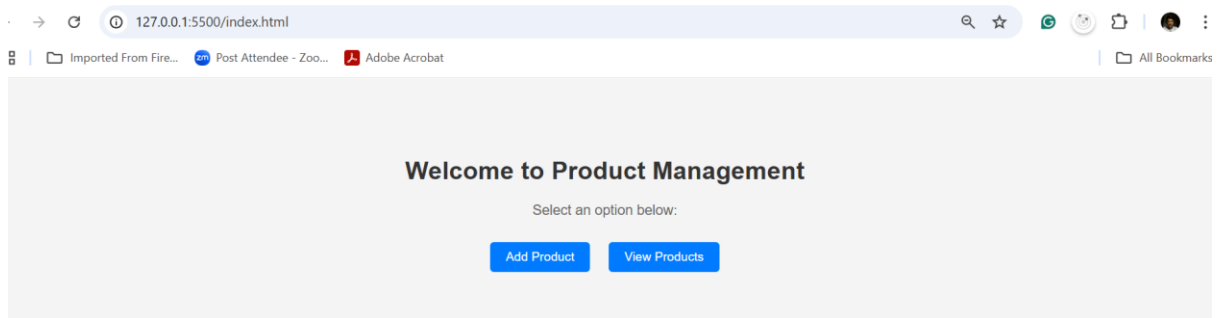
Answer in chat instead >

Now, you just need an external `styles.css` file for a modern and visually appealing design. Would you like me to create that as well?

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.**

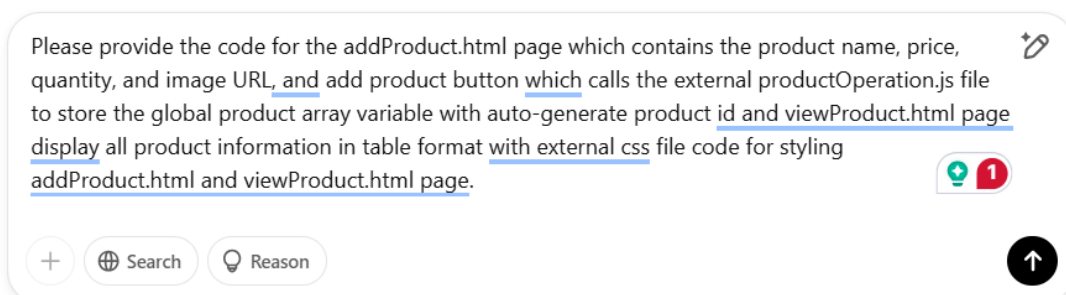


1.14 Copy and paste code from chat gpt to index.html and styles.css file code in vs code project.



1.15 : Now we need to code for addProduct page which contains product name, price, quantity and image url which call external css file and store the product in array variables.

Please provide the code for the addProduct.html page which contains the product name, price, quantity, and image URL, and add product button which calls the external productOperation.js file to store the global product array variable with auto-generate product id and viewProduct.html page display all product information in table format with external css file code for styling addProduct.html and viewProduct.html page. Use sessionStorage to share the data between one page to another pages.



Here is the code for the `addProduct.html` and `viewProduct.html` pages, along with the necessary `productOperation.js` and CSS styling files:

1. `addProduct.html`

```
html
Copy Edit

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Add Product</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Add Product</h1>
    <form id="productForm">
      <label for="productName">Product Name:</label>
      <input type="text" id="productName" required>
      <label for="productPrice">Price:</label>
      <input type="number" id="productPrice" required>
    </form>
  </div>
</body>
</html>
```

1.16 copy and paste generated code in respective file.

`addProduct.html`

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Add Product</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <div class="links">
      <a href="viewProduct.html" class="button">View Products</a>
    </div>
    <h1>Add Product</h1>
    <form id="productForm">
      <label for="productName">Product Name:</label>
      <input type="text" id="productName" required>
      <br><br>
      <label for="productPrice">Price:</label>
      <input type="number" id="productPrice" required>
    </form>
  </div>
</body>
</html>
```

```

        <br><br>

        <label for="productQuantity">Quantity:</label>
        <input type="number" id="productQuantity" required>
        <br><br>

        <label for="productImage">Image URL:</label>
        <input type="text" id="productImage" required>
        <br><br>

        <button type="button" onclick="addProduct()">Add Product</button>
    </form>
</div>

<script src="productOperation.js"></script>
</body>
</html>

```

viewProduct.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>View Products</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <div class="links">
            <a href="addProduct.html" class="button">Add Product</a>
        </div>
        <h1>Product List</h1>
        <table id="productTable">
            <thead>
                <tr>
                    <th>Product ID</th>
                    <th>Product Name</th>
                    <th>Price</th>
                    <th>Quantity</th>
                    <th>Image</th>
                </tr>
            </thead>

```

```

        <tbody>
            <!-- Product rows will be populated by JavaScript -->
        </tbody>
    </table>
</div>

<script src="productOperation.js"></script>
</body>
</html>

```

productOperation.js

```

// Load products from sessionStorage or initialize an empty array
let products = JSON.parse(sessionStorage.getItem('products')) || [];

// Function to add product
function addProduct() {
    const productName = document.getElementById('productName').value;
    const productPrice = parseFloat(document.getElementById('productPrice').value);
    const productQuantity = parseInt(document.getElementById('productQuantity').value);
    const productImage = document.getElementById('productImage').value;

    // Generate product ID automatically (simple increment)
    const productId = products.length + 1;

    // Create new product object
    const newProduct = {
        id: productId,
        name: productName,
        price: productPrice,
        quantity: productQuantity,
        image: productImage
    };

    // Add product to global array
    products.push(newProduct);

    // Save products to localStorage
    sessionStorage.setItem('products', JSON.stringify(products));

    // Clear form
    document.getElementById('productForm').reset();

    // Optionally alert user that the product has been added

```

```

    alert('Product added successfully!');
}

// Function to display products
function displayProducts() {
    const tableBody = document.querySelector('#productTable tbody');
    tableBody.innerHTML = ""; // Clear previous rows

    // Loop through products and add them to the table
    if (products.length === 0) {
        const row = document.createElement('tr');
        row.innerHTML = `<td colspan="5">No products available</td>`;
        tableBody.appendChild(row);
    } else {
        products.forEach(product => {
            const row = document.createElement('tr');

            row.innerHTML = `
                <td>${product.id}</td>
                <td>${product.name}</td>
                <td>${product.price.toFixed(2)}</td>
                <td>${product.quantity}</td>
                <td></td>
            `;

            tableBody.appendChild(row);
        });
    }
}

// Display products when page loads for viewProduct.html
if (window.location.href.includes('viewProduct.html')) {
    displayProducts();
}

```

styles.css

```

body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    text-align: center;
    margin: 0;
    padding: 0;
}

```



```
}

.container {
  margin-top: 100px;
}

h1 {
  color: #333;
}

p {
  font-size: 18px;
  color: #666;
}

.links {
  margin-top: 20px;
}

.button {
  display: inline-block;
  padding: 10px 20px;
  margin: 10px;
  font-size: 16px;
  color: #fff;
  background-color: #007BFF;
  text-decoration: none;
  border-radius: 5px;
  transition: background 0.3s;
}

.button:hover {
  background-color: #0056b3;
}

/* Basic styling for both addProduct.html and viewProduct.html */
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f4f4f4;
}

.container {
  width: 80%;
  margin: auto;
}
```

```
padding: 20px;
background-color: white;
box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
}
```

```
h1 {
  text-align: center;
  margin-bottom: 20px;
}
```

```
label {
  display: block;
  margin: 5px 0;
}
```

```
input[type="text"], input[type="number"] {
  width: 100%;
  padding: 8px;
  margin: 5px 0;
  border: 1px solid #ccc;
  border-radius: 5px;
}
```

```
button {
  width: 100%;
  padding: 10px;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
```

```
button:hover {
  background-color: #45a049;
}
```

```
table {
  width: 100%;
  border-collapse: collapse;
  margin-top: 20px;
}
```

```
table, th, td {
  border: 1px solid #ddd;
}
```

```

th, td {
    padding: 10px;
    text-align: left;
}

th {
    background-color: #f4f4f4;
}

.product-image {
    width: 50px;
    height: 50px;
    object-fit: cover;
}

```

1.16 : in addProduct.html page provide link for viewProduct.html page and vice-versa.

From **index.html** page code

```

8  </head>
9  <body>
0      <div class="container">
1          <h1>Welcome to Product Management</h1>
2          <p>Select an option below:</p>
3          <div class="links">
4              <a href="addProduct.html" class="button">Add Product</a>
5              <a href="viewProduct.html" class="button">View Products</a>
6          </div>
7      </div>
8  </body>
9  </html>
0

```

addProduct.html

```

</head>
<body>
  <div class="container">
    <div class="links">
      <a href="viewProduct.html" class="button">View Products</a>
    </div>
    <h1>Add Product</h1>
    <form id="productForm">
      <label for="productName">Product Name:</label>
      <input type="text" id="productName" required>
      <br><br>

      <label for="productPrice">Price:</label>
      <input type="number" id="productPrice" required>
      <br><br>

      <label for="productQuantity">Quantity:</label>
      <input type="number" id="productQuantity" required>
      <br><br>

      <label for="productImage">Image URL:</label>
      <input type="text" id="productImage" required>
      <br><br>

      <button type="button" onclick="addProduct()">Add Product</button>
    </form>
  </div>

```

viewProduct.html

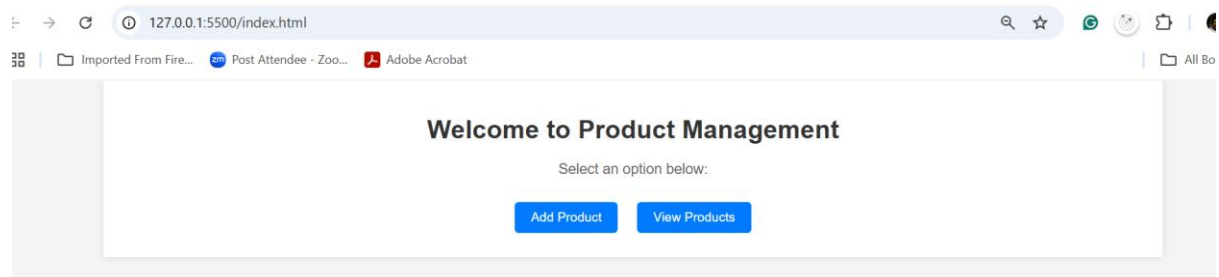
```

<body>
  <div class="container">
    <div class="links">
      <a href="addProduct.html" class="button">Add Product</a>
    </div>
    <h1>Product List</h1>
    <table id="productTable">
      <thead>
        <tr>

```

1.17 Now run the application and test it add operation and view operation working or not.

Run the index.html page first



View Products

Add Product

Product Name:

Laptop

Price:

87000

Quantity:

10

Image URL:

data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAQABAAID/2wCEAAkGBxMTEhUTEhUUVGBUYFxcXGBCYGBcXGBYXFcYHSggGBomHRCVlJElhJSkrLi4uFx8zODMlNygtLisBCgoKDg0OGhAQGI0HSUt

Add Product

127.0.0.1:5500/addProduct.html

Imported From Fire... Post Attendee - Zoo... Ado

127.0.0.1:5500 says
Product added successfully!

OK

Add Product

Product Name:

Laptop

Price:

87000

Quantity:

10

Image URL:

data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAQABAAID/2wCEAAkGBxMTEhUTEhUUVGBUYFxcXGBCYGBcXGBYXFcYHSggGBomHRCVlJElhJSkrLi4uFx8zODMlNygtLisBCgoKDg0OGhAQGI0HSUt

Add Product

View Products

Add Product

Product Name:

pen drive

Price:

2600

Quantity:

50

Image URL:


data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAQABAAID/2wCEAAkGBxITEhUQEhIVFhUXGBcXFRcWFhUXFRgYFRcWFxgXGhcYHSggGBslHRUVITEhJSkrLi4uFx8zODMlNygtLysBCgoKDg0OFxQFy0mlCUtL

Add Product

Now you can view the product details.

Add Product

Product List

Product ID	Product Name	Price	Quantity	Image
1	Laptop	\$87000.00	10	
2	pen drive	\$2600.00	50	