User Story 1

Userstory1: Basic e-commerce portal (HTML/CSS) with header, nav, featured products, newsletter subscription – show message name subscribe succesfully, footer.

Codes html

```
<!-- E-Commerce Product Portal HTML Page -->
<!-- This page displays the main e-commerce portal with featured products, navigation, and newsletter subscription -->
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>E-Commerce Product Portal</title>
   <!-- Bootstrap CSS for responsive design and styling -->
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
   <!-- Custom CSS styles -->
   <link rel="stylesheet" href="style.css">
</head>
<body>
   <!-- Header section with site title -->
   <header class="bg-primary text-white py-3">
```

```
    <div class="container">
      <h1>E-Commerce Portal</h1>
    </div>
  </header>

  <!-- Navigation bar with category links -->
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container">
      <a class="navbar-brand" href="#">Home</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link" href="#">Electronics</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">Clothing</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">Books</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">Home & Garden</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>

  <!-- Main content area -->
  <main class="container my-4">
    <!-- Featured products section -->
    <section class="featured-products">
      <h2 class="mb-4">Featured Products</h2>
      <div class="row">
        <div class="col-md-4 mb-4">
          <div class="card product-card">
            <img src="headphone.jpg" class="card-img-top" alt="Product 1">
            <div class="card-body">
              <h5 class="card-title">Wireless Headphones</h5>
              <p class="card-text">$99.99</p>
            </div>
          </div>
        </div>
        <div class="col-md-4 mb-4">
          <div class="card product-card">
            <img src="watch.jpg" class="card-img-top" alt="Product 2">
            <div class="card-body">
              <h5 class="card-title">Smart Watch</h5>
              <p class="card-text">$199.99</p>
            </div>
          </div>
        </div>
        <div class="col-md-4 mb-4">
          <div class="card product-card">
            <img src="laptop.jpg" class="card-img-top" alt="Product 3">
            <div class="card-body">
              <h5 class="card-title">Laptop</h5>
              <p class="card-text">$799.99</p>
            </div>
          </div>
        </div>
      </div>
    </section>

    <!-- Newsletter subscription section -->
    <section class="newsletter mt-5">
      <h2>Subscribe to Newsletter</h2>
      <form id="newsletterForm" class="row g-3">
        <div class="col-md-6">
          <label for="email" class="form-label">Email</label>
          <input type="email" class="form-control" id="email" required>
        </div>
        <div class="col-md-6">
          <label for="name" class="form-label">Name</label>
```

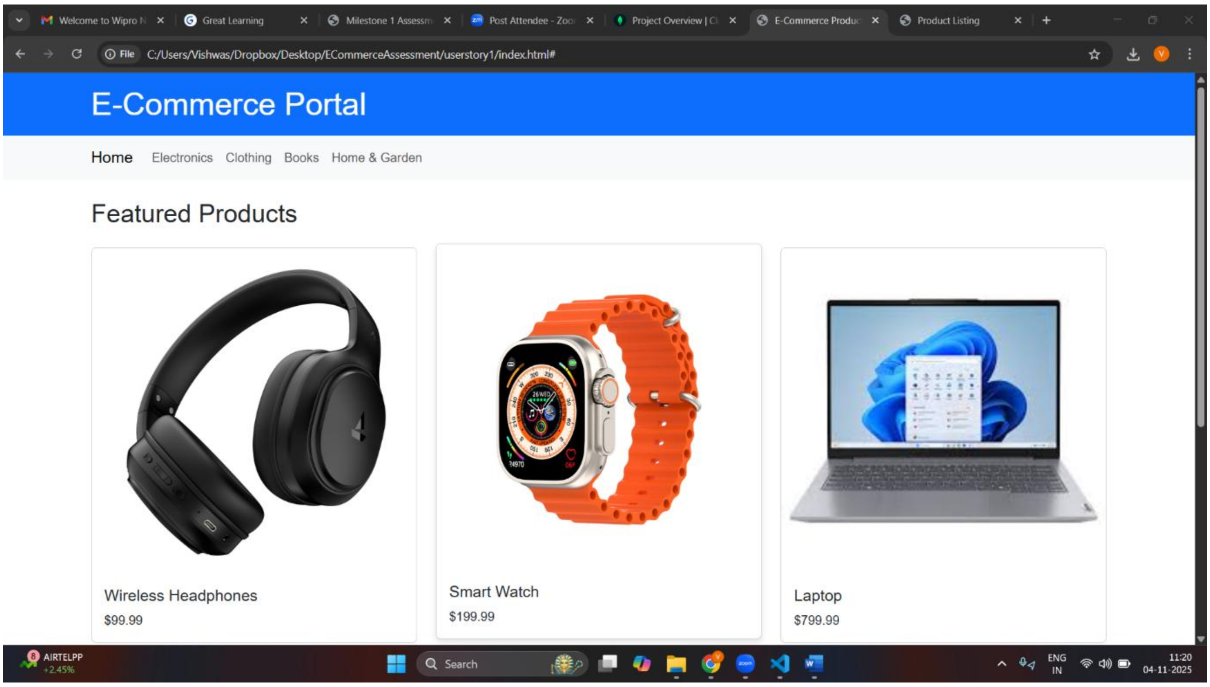
```
                    <input type="text" class="form-control" id="name" required>
                </div>
                <div class="col-12">
                    <button type="submit" class="btn btn-primary">Subscribe</button>
                </div>
            </form>
            <div id="message" class="mt-3"></div>
        </section>
    </main>

    <!-- Footer section -->
    <footer class="bg-dark text-white py-3 mt-5">
        <div class="container text-center">
            <p>&copy; 2023 E-Commerce Portal. All rights reserved.</p>
        </div>
    </footer>

    <!-- Bootstrap JavaScript for interactive components -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
    <!-- Inline JavaScript for newsletter form validation -->
    <script>
        // Event listener for newsletter form submission
        document.getElementById('newsletterForm').addEventListener('submit', function(e) {
```

```
      e.preventDefault(); // Prevent default form submission
      const email = document.getElementById('email').value;
      const name = document.getElementById('name').value;
      const messageDiv = document.getElementById('message');

      // Validate form fields and display message
      if (email && name) {
        messageDiv.innerHTML = '<div class="alert alert-success">Thank you for subscribing, ' + name + '!</div>';
        this.reset(); // Reset the form
      } else {
        messageDiv.innerHTML = '<div class="alert alert-danger">Please fill in all fields.</div>';
      }
    });
  </script>
</body>
</html>
```

Code CSS

```
/* Custom CSS styles for the E-Commerce Product Portal */

/* Styles for product cards with hover effects */
.product-card {
  transition: transform 0.3s ease, box-shadow 0.3s ease;
}

.product-card:hover {
  transform: translateY(-5px);
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

/* Styles for the newsletter section */
.newsletter {
  background-color: #f8f9fa;
  padding: 2rem;
  border-radius: 0.5rem;
}

/* Global body styles */
body {
  font-family: Arial, sans-serif;
}

/* Header title styles */
header h1 {
  margin: 0;
}
```

```
/* Footer styles */
footer {
   margin-top: auto;
}
```



E-Commerce Portal

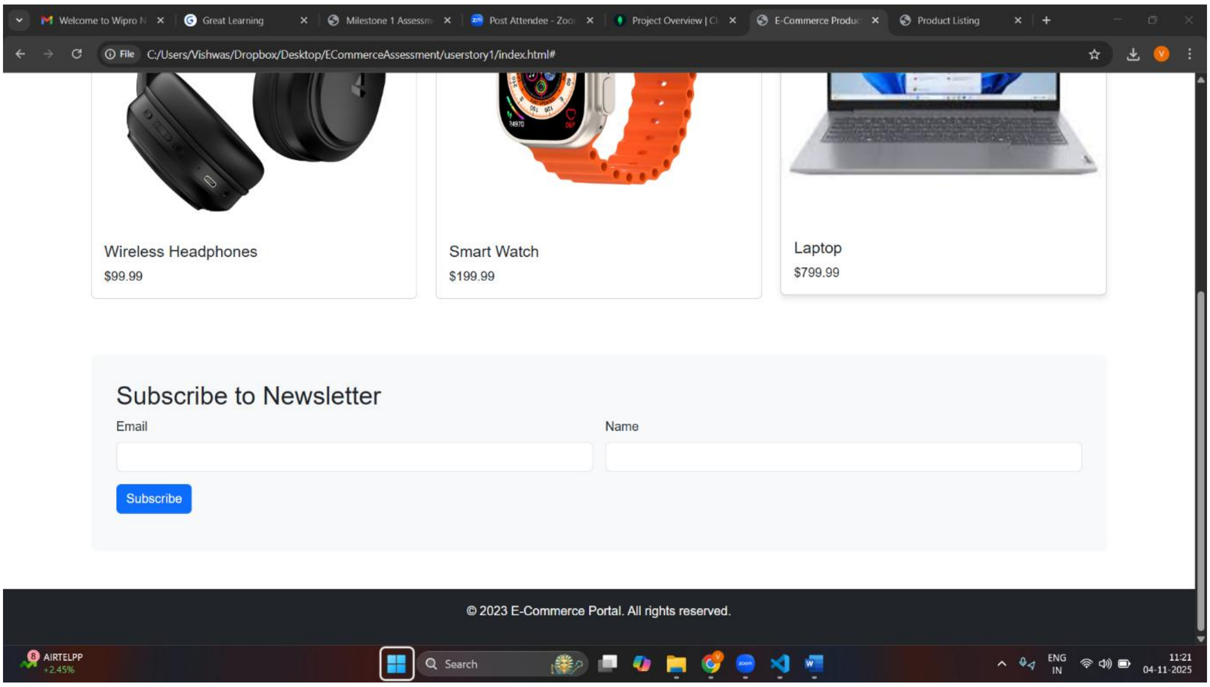
Home Electronics Clothing Books Home & Garden

Featured Products

Wireless Headphones
$99.99

Smart Watch
$199.99

Laptop
$799.99



Wireless Headphones
$99.99

Smart Watch
$199.99

Laptop
$799.99

Subscribe to Newsletter

Email

Name

Subscribe

© 2023 E-Commerce Portal. All rights reserved.

User story 2

Userstory2: Product listing with filtering/sorting (HTML/JS/JSON). Product fetching , product display,

Category filtering , sort by price

Codes html

```
<!-- Product Listing HTML Page -->

<!-- This page displays a list of products with filtering and sorting options
-->

<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Product Listing</title>
    <!-- Bootstrap CSS for responsive design and styling -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
    <!-- Main container for the product listing page -->
    <div class="container my-4">
        <h1>Product Listing</h1>
        <!-- Filter and sort controls -->
        <div class="row mb-4">
            <div class="col-md-6">
                <label for="categoryFilter" class="form-label">Filter by Category:</label>
                <select id="categoryFilter" class="form-select">
                    <option value="all">All Categories</option>
                    <option value="Electronics">Electronics</option>
                    <option value="Clothing">Clothing</option>
                    <option value="Books">Books</option>
                    <option value="Home & Garden">Home & Garden</option>
                </select>
```

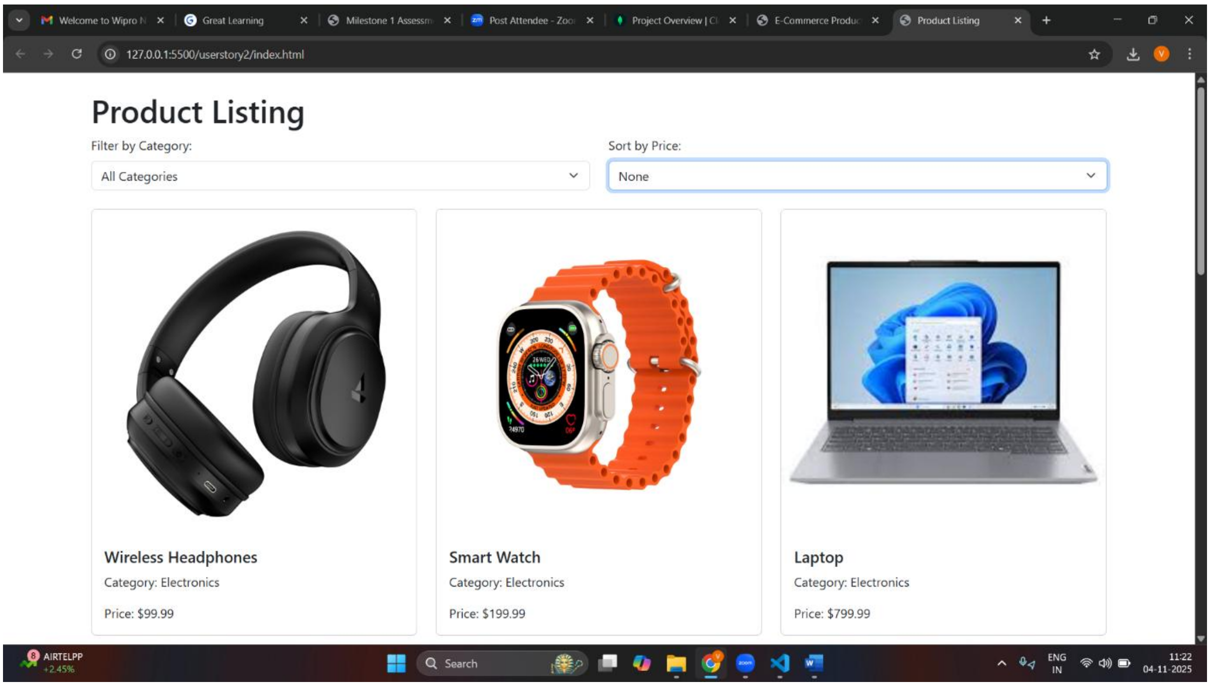
```
      </div>
      <div class="col-md-6">
        <label for="priceSort" class="form-label">Sort by Price:</label>
        <select id="priceSort" class="form-select">
          <option value="none">None</option>
          <option value="low-to-high">Low to High</option>
          <option value="high-to-low">High to Low</option>
        </select>
      </div>
    </div>
    <!-- Loading message -->
    <div id="loading" class="text-center" style="display: none;">
      <p>Loading products...</p>
    </div>
    <!-- Container for displaying products -->
    <div id="productContainer" class="row"></div>
  </div>
  <!-- JavaScript file for product functionality -->
  <script src="script.js"></script>
</body>
</html>
```

Code js

```
// JavaScript for Product Listing Page
// Handles fetching, displaying, filtering, and sorting of products
// DOM elements
const productContainer =
document.getElementById('productContainer');
const categoryFilter = document.getElementById('categoryFilter');
const priceSort = document.getElementById('priceSort');
const loadingMessage = document.getElementById('loading');

// Array to store products data
let products = [];

// Function to fetch products from JSON file
async function fetchProducts() {
    loadingMessage.style.display = 'block';
    try {
        const response = await fetch('products.json');
        if (!response.ok) {
            throw new Error('Network response was not ok');
        }
        products = await response.json();
        displayProducts(products);
    } catch (error) {
        console.error('Error fetching products:', error);
        productContainer.innerHTML = '<p class="text-danger">Failed to
load products. Please try again later.</p>';
    } finally {
        loadingMessage.style.display = 'none';
    }
}

// Function to display products in the container
function displayProducts(productsToDisplay) {
    productContainer.innerHTML = '';
    productsToDisplay.forEach(product => {
        const productCard = `
            <div class="col-md-4 mb-4">
                <div class="card">
                    <img src="${product.image}" class="card-img-top"
alt="${product.name}">
                    <div class="card-body">
                        <h5 class="card-title">${product.name}</h5>
                        <p class="card-text">Category: ${product.category}</p>
                        <p class="card-text">Price: $${product.price}</p>
                    </div>
                </div>
            </div>
        `;
        productContainer.innerHTML += productCard;
    });
}

// Function to filter and sort products based on user selections
function filterAndSortProducts() {
    let filteredProducts = products;
    const selectedCategory = categoryFilter.value;
    if (selectedCategory !== 'all') {
        filteredProducts = filteredProducts.filter(product => product.category === selectedCategory);
    }

    const sortOrder = priceSort.value;
    if (sortOrder === 'low-to-high') {
        filteredProducts.sort((a, b) => a.price - b.price);
    } else if (sortOrder === 'high-to-low') {
        filteredProducts.sort((a, b) => b.price - a.price);
    }

    displayProducts(filteredProducts);
}

// Event listeners for filter and sort controls
categoryFilter.addEventListener('change', filterAndSortProducts);
priceSort.addEventListener('change', filterAndSortProducts);

// Initial fetch of products on page load
fetchProducts();
```

code json data:

```
[
{
    "id": 1,
    "name": "Wireless Headphones",
    "category": "Electronics",
    "price": 99.99,
    "image": "headphone.jpg"
},
{
    "id": 2,
    "name": "Smart Watch",
    "category": "Electronics",
    "price": 199.99,
    "image": "watch.jpg"
},
{
    "id": 3,
    "name": "Laptop",
    "category": "Electronics",
    "price": 799.99,
    "image": "laptop.jpg"
},
{
    "id": 4,
    "name": "T-Shirt",
    "category": "Clothing",
    "price": 19.99,
    "image": "tshirt.jpg"
  },
  {
    "id": 5,
    "name": "Jeans",
    "category": "Clothing",
    "price": 49.99,
    "image": "jeans.jpg"
  },
  {
    "id": 6,
    "name": "Book: JavaScript Guide",
    "category": "Books",
    "price": 29.99,
    "image": "javascript book.jpg"
  },
  {
    "id": 7,
    "name": "Garden Hose",
    "category": "Home & Garden",
    "price": 24.99,
    "image": "garden hose.jpg"
  },
  {
    "id": 8,
    "name": "Coffee Maker",
        "category": "Home & Garden",
        "price": 79.99,
        "image": "coffee maker.jpg"
    }
]
```



Product Listing

Filter by Category:

All Categories

Sort by Price:

None

Wireless Headphones

Category: Electronics

Price: $99.99

Smart Watch

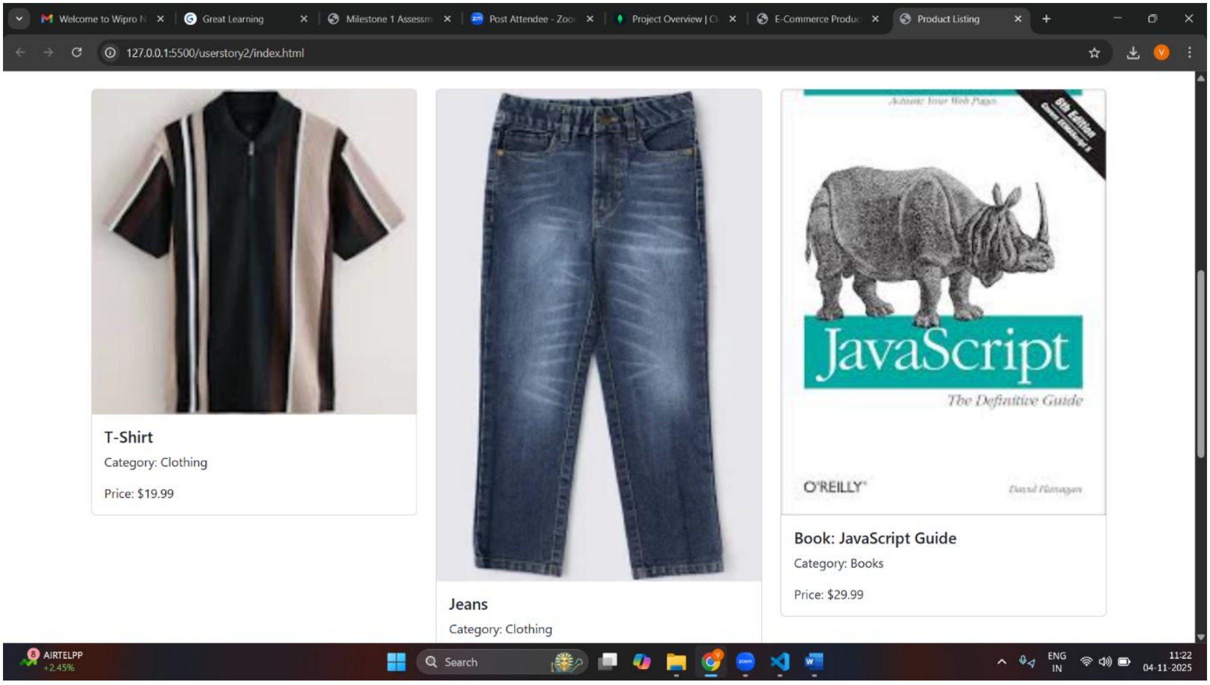
Category: Electronics

Price: $199.99

Laptop

Category: Electronics

Price: $799.99



T-Shirt

Category: Clothing

Price: $19.99

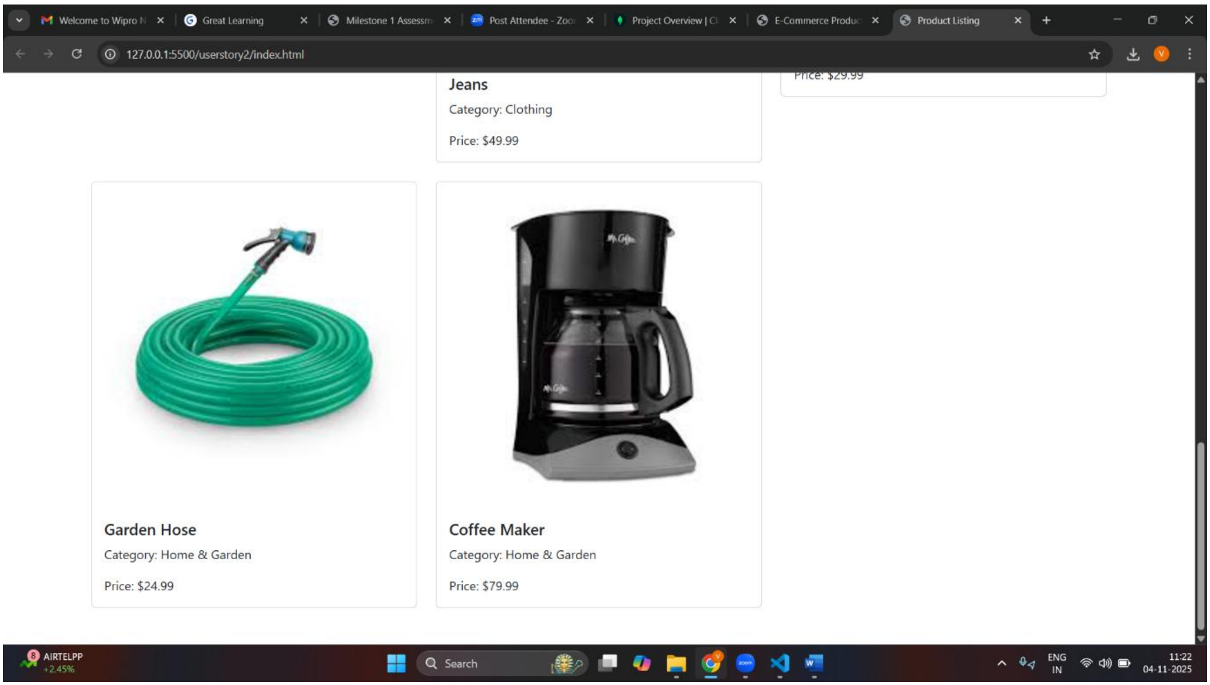
Jeans

Category: Clothing

Book: JavaScript Guide

Category: Books

Price: $29.99



Jeans

Category: Clothing

Price: $49.99

Price: $29.99

Garden Hose

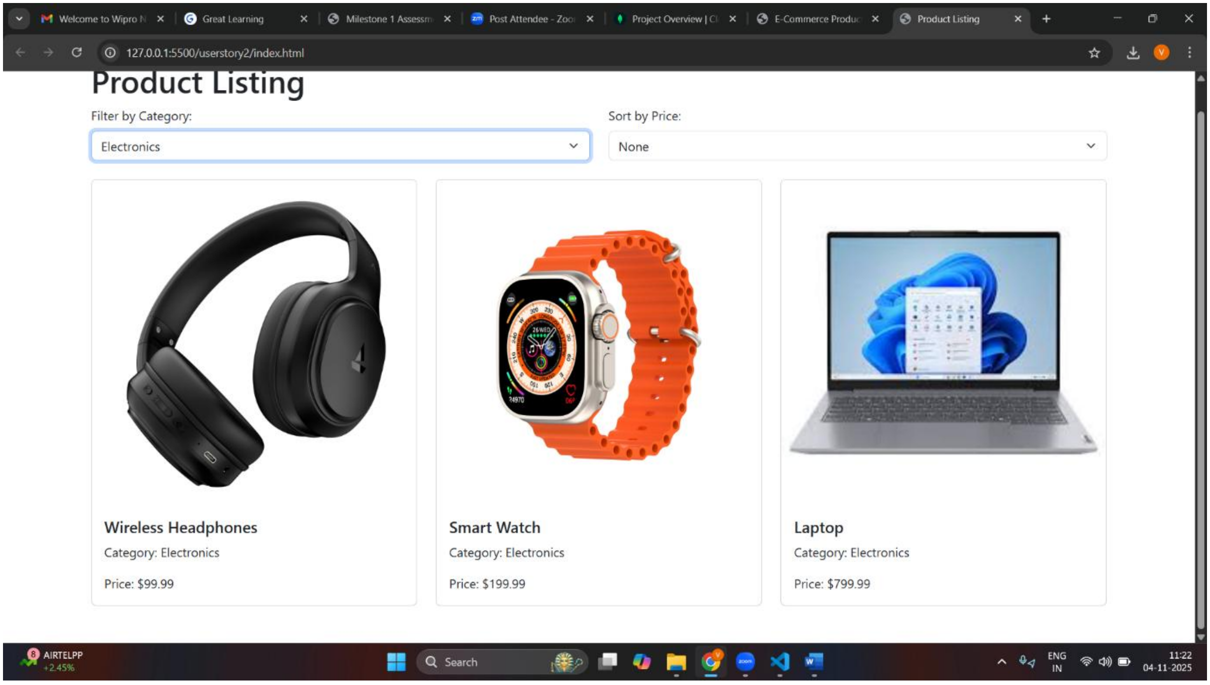
Category: Home & Garden

Price: $24.99

Coffee Maker

Category: Home & Garden

Price: $79.99



Product Listing

Filter by Category: Electronics

Sort by Price: None

Wireless Headphones

Category: Electronics

Price: $99.99

Smart Watch

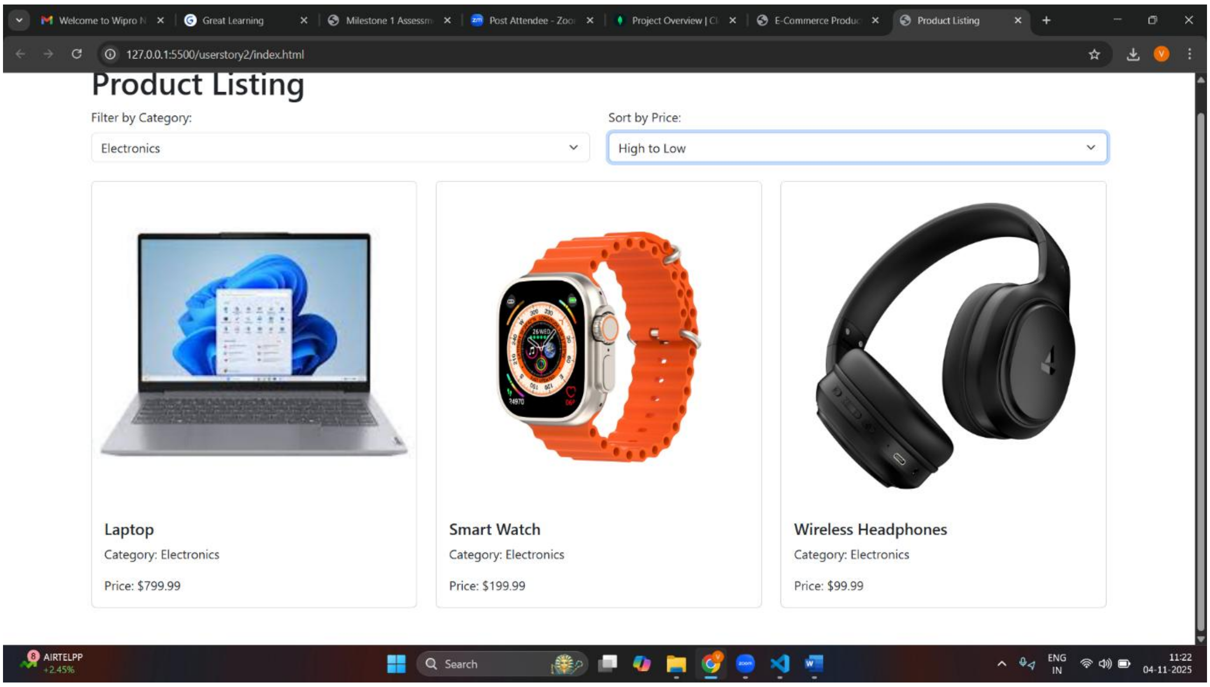
Category: Electronics

Price: $199.99

Laptop

Category: Electronics

Price: $799.99



Product Listing

Filter by Category: Electronics

Sort by Price: High to Low

Laptop
Category: Electronics
Price: $799.99

Smart Watch
Category: Electronics
Price: $199.99

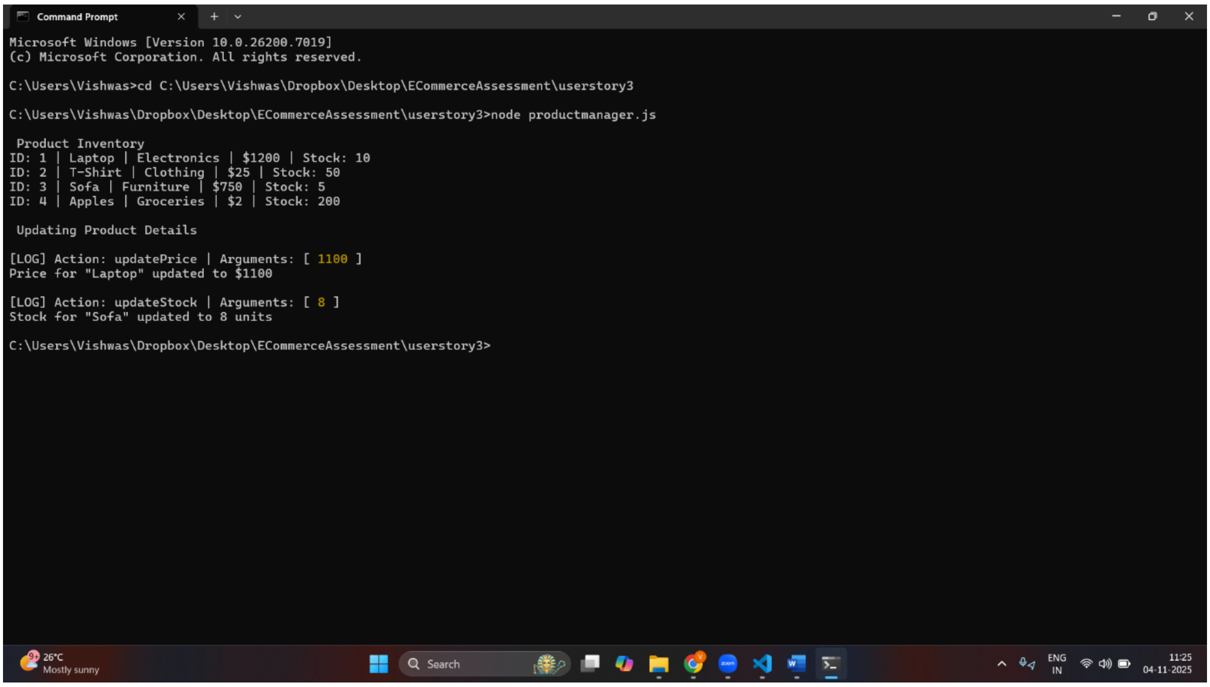
Wireless Headphones
Category: Electronics
Price: $99.99

User story 3

Userstory3: Product management in TypeScript with classes/interfaces enum

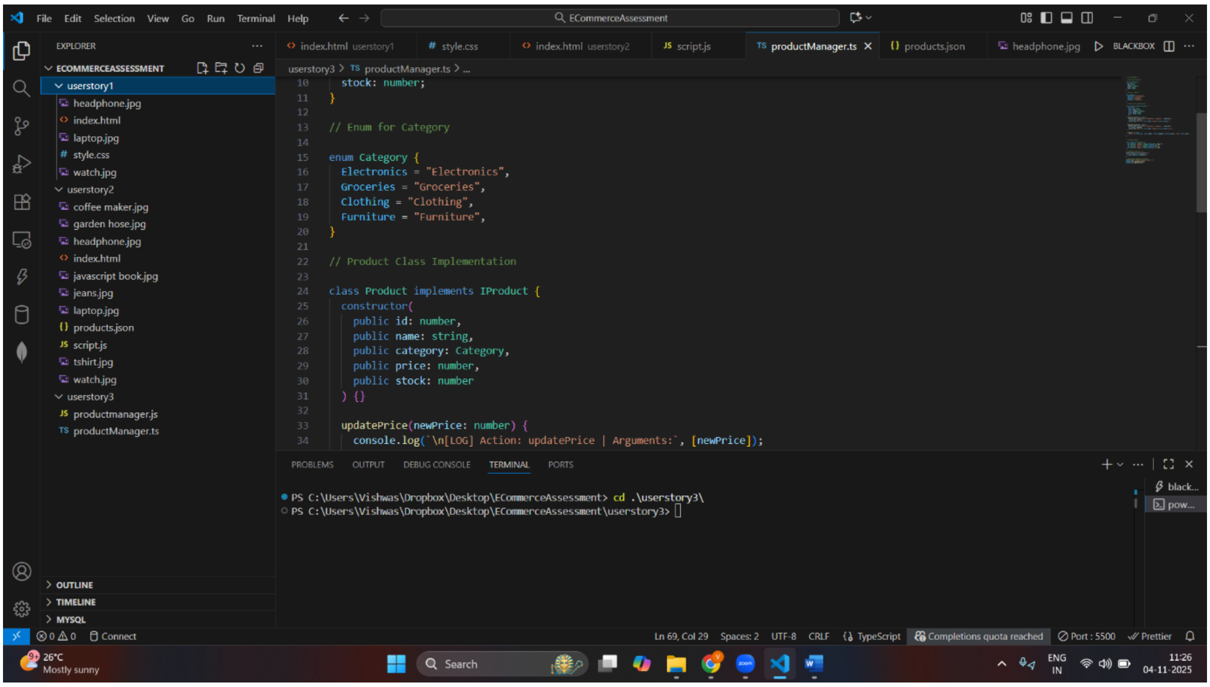
```
// productManager.ts

// Interface Definition

interface IProduct {
  id: number;
  name: string;
  category: Category;
  price: number;
  stock: number;
}

// Enum for Category

enum Category {
  Electronics = "Electronics",
  Groceries = "Groceries",
  Clothing = "Clothing",
  Furniture = "Furniture",
}

// Product Class Implementation

class Product implements IProduct {
  constructor(
    public id: number,
    public name: string,
    public category: Category,
    public price: number,
    public stock: number
  ) {}
  updatePrice(newPrice: number) {
    console.log(`\n[LOG] Action: updatePrice | Arguments:`, [newPrice]);
    this.price = newPrice;
    console.log(`Price for "${this.name}" updated to $${this.price}`);
  }

  updateStock(newStock: number) {
    console.log(`\n[LOG] Action: updateStock | Arguments:`, [newStock]);
    this.stock = newStock;
    console.log(`Stock for "${this.name}" updated to ${this.stock} units`);
  }

  display(): string {
    return `ID: ${this.id} | ${this.name} | ${this.category} | $${this.price} | Stock: ${this.stock}`;
  }
}

// Storing Products
```

```
const productss: Product[] = [
  new Product(1, "Laptop", Category.Electronics, 1200, 10),
  new Product(2, "T-Shirt", Category.Clothing, 25, 50),
  new Product(3, "Sofa", Category.Furniture, 750, 5),
  new Product(4, "Apples", Category.Groceries, 2, 200),
];

// Iteration using for...of
console.log("\n Product Inventory ");
for (const product of productss) {
  console.log(product.display());
}

// Demonstrate Decorator in Action
console.log("\n Updating Product Details ");
productss[0].updatePrice(1100);
productss[2].updateStock(8);
```



```
Microsoft Windows [Version 10.0.26200.7019]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vishwas>cd C:\Users\Vishwas\Dropbox\Desktop\ECommerceAssessment\userstory3

C:\Users\Vishwas\Dropbox\Desktop\ECommerceAssessment\userstory3>node productmanager.js

 Product Inventory
ID: 1 | Laptop | Electronics | $1200 | Stock: 10
ID: 2 | T-Shirt | Clothing | $25 | Stock: 50
ID: 3 | Sofa | Furniture | $750 | Stock: 5
ID: 4 | Apples | Groceries | $2 | Stock: 200

 Updating Product Details

[LOG] Action: updatePrice | Arguments: [ 1100 ]
Price for "Laptop" updated to $1100

[LOG] Action: updateStock | Arguments: [ 8 ]
Stock for "Sofa" updated to 8 units

C:\Users\Vishwas\Dropbox\Desktop\ECommerceAssessment\userstory3>
```

structure



```
ECOMMERCEASSESSMENT
  userstory1
    headphone.jpg
    index.html
    laptop.jpg
    style.css
    watch.jpg
  userstory2
    coffee maker.jpg
    garden hose.jpg
    headphone.jpg
    index.html
    javascript book.jpg
    jeans.jpg
    laptop.jpg
    products.json
    script.js
    tshirt.jpg
    watch.jpg
  userstory3
    productmanager.js
    productManager.ts
```

```
  stock: number;
}

// Enum for Category

enum Category {
  Electronics = "Electronics",
  Groceries = "Groceries",
  Clothing = "Clothing",
  Furniture = "Furniture",
}

// Product Class Implementation

class Product implements IProduct {
  constructor(
    public id: number,
    public name: string,
    public category: Category,
    public price: number,
    public stock: number
  ) {}

  updatePrice(newPrice: number) {
    console.log(`\n[LOG] Action: updatePrice | Arguments:`, [newPrice]);
```

```
PS C:\Users\Vishwas\Dropbox\Desktop\ECommerceAssessment> cd .\userstory3\
PS C:\Users\Vishwas\Dropbox\Desktop\ECommerceAssessment\userstory3>
```