Project Name: ShopForHome

ShopForHome is a popular store offering home décor products. Due to COVID-19, all offline shopping has stopped, and the store wants to move its operations to an online platform by creating a web application.

There are two types of users in this application:

1. **User**
2. **Admin**

User Stories:

1. **Login, Logout, and Registration:**
   - As a User, I should be able to log in, log out, and register on the application.
2. **View Products by Category:**
   - As a User, I should be able to browse products based on categories (e.g., Furniture, Home Décor, Lighting).
3. **Product Filtering:**
   - As a User, I should be able to filter products by price, rating, and category.
4. **Add to Cart:**
   - As a User, I should be able to add products to the shopping cart.
5. **Manage Cart Quantity:**
   - As a User, I should be able to increase or decrease the quantity of items in my shopping cart.
6. **Unlimited Product Addition:**
   - As a User, I should be able to add an unlimited number of products to the shopping cart.
7. **Wishlist Feature:**
   - As a User, I should have a wishlist option where I can save products that I want to purchase later.
8. **Discount Coupons:**
   - As a User, I should receive different discount coupons for promotions or sales events.

Admin Stories:

1. **Login, Logout, and Registration:**
   - As an Admin, I should be able to log in, log out, and register on the application.

2. **User Management (CRUD):**
   - As an Admin, I should be able to create, read, update, and delete user accounts.
3. **Product Management (CRUD):**
   - As an Admin, I should be able to create, read, update, and delete products in the catalog.
4. **Bulk Product Upload via CSV:**
   - As an Admin, I should be able to upload product details in bulk using a CSV file.
5. **Stock Management:**
   - As an Admin, I should be able to view the stock of products and receive notifications when the stock level drops below 10.
6. **Sales Reports:**
   - As an Admin, I should be able to generate sales reports for a specified duration.
7. **Discount Coupons for Specific Users:**
   - As an Admin, I should be able to create and assign discount coupons to specific sets of users.

Instructions:

1. The **front end** of the application should be implemented using **React** to handle the user interface, components, and interactions.
2. The **back end** of the application should be implemented using **ASP.NET Core MVC**, handling business logic, database operations, and API development.
3. Use the `assets/images` folder to store product images.
4. Provide the database schema in a `.sql` file.
5. Create separate **Web API controllers** for:
   - Search functionality
   - Sales reports
   - Discount coupons
6. Use **Entity Framework (EF)** or any other ORM tool to communicate with the database.
7. Use **Swagger** for testing the Web APIs and providing API documentation.

Sprint Plan:

Sprint I Objectives:

1. Create the Use Case document.
2. Design the database schema with all necessary tables and their relationships.
3. Select **Entity Framework** (or any other ORM tool) to communicate with the database.
4. Identify the required controllers for the back end (ASP.NET Core MVC).

5. Create the initial **React components** and set up the static e-commerce views for product listings and images.

Sprint II Objectives:

1. Implement the **DBContext** object using the selected ORM tool (e.g., Entity Framework).
2. Develop **CRUD operations** for Users and Products in the back end (ASP.NET Core MVC).
3. Implement the **Bulk product upload** feature using CSV files in the back end.
4. Build the **Cart functionality** (add, update, delete) using React for the front end.

Sprint III Objectives:

1. Develop the **Web API** for search functionality, reports, and discount coupons in ASP.NET Core MVC.
2. Integrate **Swagger UI** and test the Web API.
3. Implement **search and filtering** features in the React front end that consume the Web API.
4. Develop **stock management** and **sales report** generation features for Admin users in ASP.NET Core MVC.

Reference:

For further reference on the e-commerce store design and functionality, please visit:

- https://www.pepperfry.com/