ASP.NET MVC Product Management Assessment

Instructions:

1. Create a new ASP.NET MVC Core project with Entity Framework code first approach and complete the following tasks.

2. Use any library like Bootstrap or jQuery to style your views.

3. Use identity management

4. Create login and logout functionality

5. Create roles like Admin, Users

Note: Admin can add, edit, delete, products, users, categories, suppliers. User registration should be available for normal users, globally. Once the user registers, they can add product to cart.

Note: Please use non-generic repository pattern.

Task 1: Model Creation

Create the following models with the following properties:

Product Model

- ProductId (int, primary key)

- Name (string)

- Description (string)

- Price (decimal)

- CategoryId (int)

- SupplierId (int)

Category Model

- CategoryId (int, primary key)

- Name (string)

Supplier Model

- SupplierId (int, primary key)

- Name (string)

- ContactName (string)

- ContactEmail (string)

- ContactPhone (string)

Shopping Cart

-CartId (int, primary key)

-UserId (int)

-ProductId(int)

User

Table will be created based on identity

Role

Table will be created based on identity

Task 2: Database Context

Dataseed for admin user and roles (“Admin and “User”)

Task 3: Controller Creation

Create Auth Controller for Register and Login functionality.

Create a controller called " AdminContext " with the following actions:

-List users

-List Categories

-List Products

-Suppliers

-Create menu items for Products, Users, Categories, Suppliers and Cart

Create a controller called " ProductContext " with the following actions:

Create a database context called "ProductContext" that inherits from "DbContext" and add a DbSet property for each of the models you created in Task 1.

- Index: display a list of all products in the database with their corresponding category and supplier information.

- Create: display a form to create a new product with a dropdown list of categories and suppliers.

- Store: store the new product in the database.

- Edit: display a form to edit an existing product with a dropdown list of categories and suppliers.

- Update: update an existing product in the database.

- Delete: delete an existing product from the database.

- List all the users along with admin in the user list page.

- Cart list page will display which user have added item in their cart

- User can delete and add product items in Cart.

Note: Use jQuery Ajax

Task 4: Views

Create views for each action using Razor syntax. The views should be styled using a front-end framework or library like Bootstrap or jQuery.

Bonus Task

Add search functionality to the Index action that allows the user to search for products by name, category, or supplier.

- Use ASP.NET MVC Core and Entity Framework code first approach.

- You will be evaluated based on the accuracy and completeness of your code as well as the organization and efficiency of your solution.