The Shopping list app
project name

Document history

Rev	Group Members	Description
1	Ingabire Princesse Darlene	System Documentation for the app
2	Rango Kami Miguel Samuel	

1. Project Subject

This project is a web-based Shopping List application built using ASP.NET Core. It allows users to manage a list of shopping items, offering functionalities such as adding, viewing, editing, and deleting items. The application supports user authentication, enabling personalized lists and enhancing security by restricting access to certain features based on user roles.

2. Specification of the technologies used

Framework: ASP.NET Core 6.0

Database: SQL Server

ORM: Entity Framework Core

Authentication: Cookie-based Authentication

IDE: Visual Studio 2022

Languages: C#

3. Instructions for Initial Project Setup

Clone the Repository: Clone the project repository from your source control system.

Open in Visual Studio: Open the project in Visual Studio 2022 or later.

Configure Database Connection: Update the appsettings.json file with your SQL Server connection string.

Run Migrations: Execute the following commands in the Package Manager Console to apply the database migrations.

Run the Application: Press F5 or use the dotnet run command to start the application.

4. Description of Project Structure

Controllers: Handles HTTP requests and returns responses.

Data: Contains database context classes for Entity Framework Core.

Models: Contains classes representing the data structure.

Views: Contains Razor views for the UI.

wwwroot: Contains static files like CSS, JavaScript, and images.

Program.cs: Entry point of the application.

appsettings.json: Configuration settings.

Startup.cs: Configures services and the app's request pipeline

Models

1. Item

Represents an item in the shopping list.

Fields:
Id (int): Primary key.
Name (string): Name of the item. Required.
Description (string): Description of the item. Optional.
Quantity (int): Number of items. Required.
IsPurchased (bool): Indicates if the item is purchased. Defaults to false.
UserId (string): Foreign key referencing the user who created the item.

2. User

Represents a user in the system.

Fields:
Id (string): Primary key.
UserName (string): Username. Required.
Email (string): Email address. Required and must be unique.
PasswordHash (string): Password hash.
Role (string): User role, e.g., Admin, User.

Controllers

1. HomeController

Handles requests related to the home page.

Methods:
Index
HTTP Method: GET
Parameters: None
Description: Displays the home page.
Returns: View for the home page.

2. ItemController

Handles CRUD operations for items.

Methods:

Index

HTTP Method: GET
Parameters: None
Description: Displays a list of all items.
Returns: View with a list of items.

Create

HTTP Method: GET, POST
Parameters: Item object (POST)
Description: Displays the item creation form on GET; processes form submission on POST.
Returns: Redirects to Index after creation.

Edit

HTTP Method: GET, POST
Parameters: id (int) for GET, Item object for POST
Description: Displays the item edit form on GET; processes updates on POST.
Returns: Redirects to Index after editing.

Delete

HTTP Method: POST
Parameters: id (int)
Description: Deletes an item.
Returns: Redirects to Index.

3. AccountController

Handles user authentication and account management.

Login

HTTP Method: GET, POST
Parameters: Login model (POST)
Description: Displays the login form on GET; processes login on POST.
Returns: Redirects to the home page after login.

Register

HTTP Method: GET, POST
Parameters: Registration model (POST)
Description: Displays the registration form on GET; processes registration on POST.
Returns: Redirects to the login page after registration.

Logout

HTTP Method: POST
Parameters: None
Description: Logs out the user.
Returns: Redirects to the home page.

Description of the User System

Roles:

Admin: Can manage all users and items.
User: Can manage their own items.

Role Assignment: Roles are assigned during user registration or manually by an admin.

Capabilities:
Logged-in Users: Can create, view, edit, and delete their own items.
Guests: Can only view the home page and login/register.

User Information:

User-Specific: Items created by the user.
Global: General application settings and shared resources.

Interesting Features

Item Filtering: Users can filter items by their purchased status.
User-specific Data: Each user has a personalized list of items.
Session Management: Keeps user preferences and session data consistent across requests.