Capstone Project 1: Library Management

Title: Design and develop a Library Management System

Problem Statement:

A good Samaritan and well-wishers in your neighbourhood have a situation on their hands with lots of used books, of all categories being handed over to them. They would like your team to help them with a Single Page web Application (SPA) that can be used to organize the Books, into Libraries based on different Categories, making them searchable.

Technologies Involved:

- Windows OS
- .Net 8.0 or 9.0 SDK
- Visual Studio 2020 or 2022
- Microsoft SQL Server 2019 or 2022
- SQL Server Management Studio 2019
- Angular 19
- Visual Studio Code

Agile User Stories:

Create a LibMgmt database using MS SQL Server
Develop an ASP.Net Web API to communicate with the database and provide data in JSON format.
Develop an angular SPA to talk to the ASP.Net web API and provide users with a fantastic web based UI.

Sprint Plan:

Sprint 1 – Design of the Database LibMgmtDb

1. Create Libraries table with the following fields for capturing Library information:

- LibraryId: Integer
- Name: String
- Address: String
- MaximumCapacity: Integer
- Books: ICollection<Book>
  Description: Represents a library entity with attributes including name, address, maximum capacity, and a list of books belonging to the library.

2. Create Books table with the following fields for capturing Book information:

- BookId: Integer
- Title: String
- Author: String
- Category: String
- Price: Decimal
- LibraryId: int
- Library : Library
  Description: Represents a book entity with attributes such as title, author, category, price, and library Id.

Sprint 2 – Design and develop an ASP.Net Web API for accessing the LibMgmtDb database

1. Create an ASP.Net Web API Project called LibMgmtAPI, which will be implemented using the MVC architecture.
2. Add the packages required for implementing EF Db-First approach, since the database has already been created.
3. Run Scaffolding using Nuget Package Manager window, to automatically generate the model classes and DbContext derived class for the database.

Sprint 3 – Implement Services and Controller classes

1. Create an IBookManager interface with the following methods:

AddBook(Book book): int
UpdateBook(int id, Book book): int
DeleteBook(int id): int
GetBooks(): List<Book>
FindBookById(int id): Book
FindBookByName(string name) : Book

Description: Declares the methods for managing books, including adding, editing, deleting, and listing books, as well as database operations related to books.

2. Create an ILibraryManager interface with the following methods:

AddLibrary(): int
GetLibraries(): List<Library>
FindLibraryById(int id): Library
FindLibraryByName(string name): Library
UpdateLibrary(int id, Library library) : int
DeleteLibrary(int id): int

Description: Defines methods for managing libraries, including adding and listing libraries, and database operations related to libraries.

3. Create classes implementing the above Interfaces:

a. BookManager : IBookManager

Description: Implements the methods defined in the IBookManager interface for managing books. It includes operations such as adding, editing, deleting, and listing books and database operations.

b. LibraryManager : ILibraryManager

Description: Implements the methods defined in the ILibraryManager interface for managing libraries. It includes operations such as adding and listing libraries and database operations.

4. Implement end point action methods in the LibraryController, to allow an Admin user to add, modify, delete and list libraries – all, by Id and by name.

Each method in the controller class calls the corresponding Library Manager method which interacts with the database.

5. Implement the same methods in the **BookController class.**

Sprint 4: Apply JWT authentication using ASP.Net Identity, in your project

1. Implement authentication using ASP.Net Identity, generating JWT, to ensure your resource end-points are secure.

Create an IAuthService interface with the following methods:

a. Register(RegisterModel model, string role) : (int, string)
b. Login(LoginModel model) : (int, string)
Description: Defines methods for managing Users, including adding and listing users, and database operations related to users.

Create the implementation class for the AuthService interface.

2. Implement AccountController with end point action methods for authentication and managing the Users

Sprint 5 – Develop an SPA using angular to communicate with the ASP.Net Web API

1. Create the models folder to accommodate all the model classes

2. Create the following components RegistrationComponent, LoginComponent, NavbarComponent, AdminComponent, BookComponent, LibraryComponent, HomeComponent and ErrorComponent.

These tasks collectively lay the groundwork for the application's frontend and backend functionalities, enabling user authentication, user registration, library management, and book management features. Additionally, the defined interfaces ensure consistency and structure within the application's data models.

3. Implementing the registration and login forms using **template-driven forms.**
4. Implement routing in the application.
5. Create an AuthGuard, to protect the routes that need to be accessible only to logged in users.
6. Apply Bootstrap for all pages developed above. Use proper styling

Deliverables:

A SQL Server based database that can store and manage the books information, into libraries

An ASP.Net Web API that keeps t he data safe, providing data only to authenticated users.

An Angular web app that can communicate with the ASP.Net Web API and provide users a intuitive user experience.