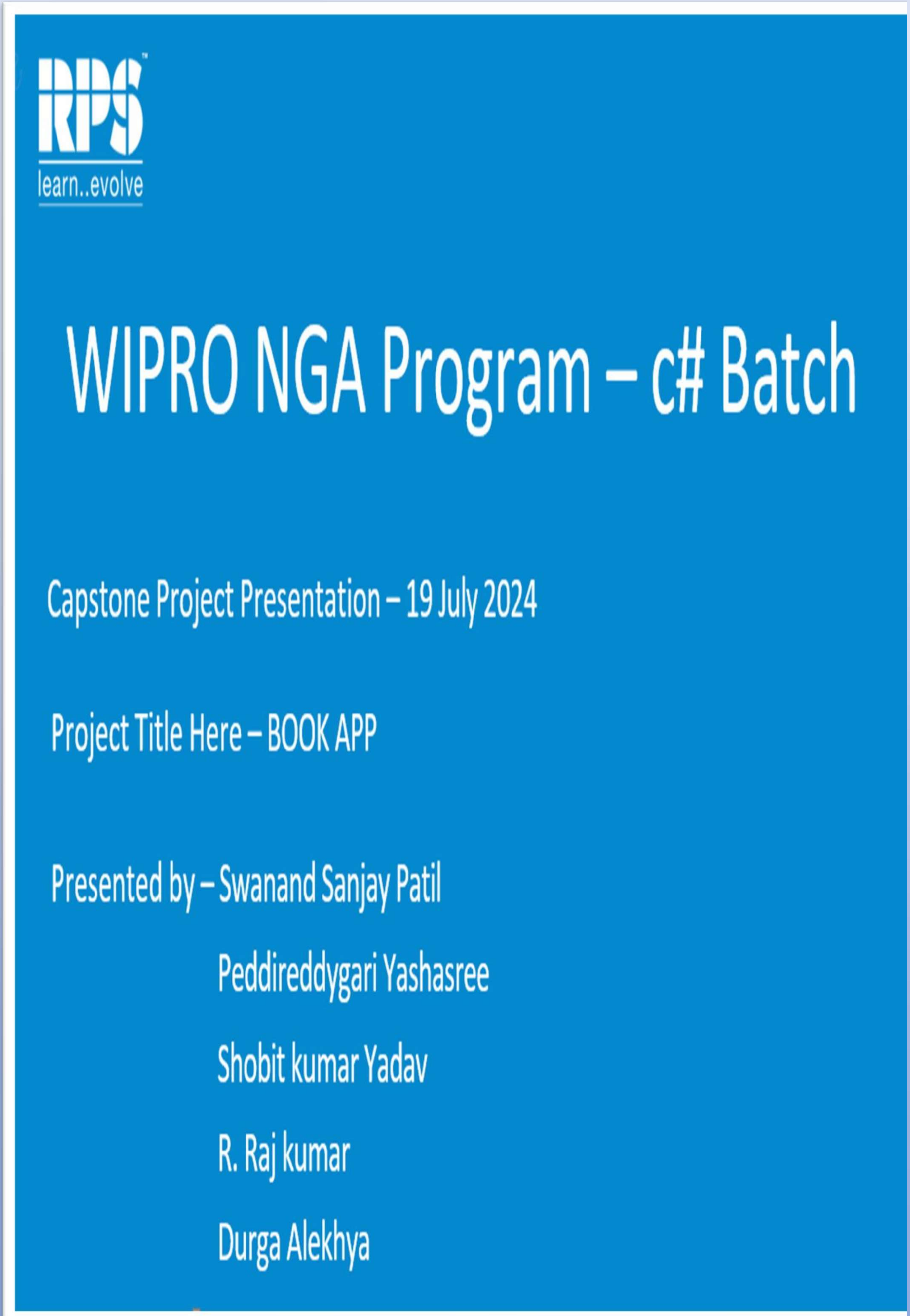


WIPRO NGA Program – c# Batch

Capstone Project Presentation – 19 July 2024

Project Title Here – BOOK APP

Presented by – Swanand Sanjay Patil

Peddireddygari Yashasree

Shobit kumar Yadav

R. Raj kumar

Durga Alekhya

BOOK APP

CONTENTS

CHAPTER-1

Introduction

1.1 Introduction

The **Book App** is designed to provide users with an interactive and personalized platform to manage and explore books. The core feature of the application is the dashboard, which acts as the central hub for users to access their favorite books, discover authors, and receive book recommendations from the external service provider, openlibrary.org.

Users can register and create a profile, complete with a profile image. Once registered, users can log in and use the app's search functionality to find specific books or authors. The app categorizes content into three main sections on the dashboard:

1. **Favorite Books**: This section displays all the favorite books that the user has added.
2. **Authors**: This section showcases various authors.
3. **Recommendations**: This section provides book recommendations sourced from openlibrary.org.

Objectives:

The **Book App** aims to provide a comprehensive and user-friendly platform for managing and exploring books. The primary objectives are:

1. **Create a Dashboard View**:
   - Develop a dashboard accessible via the Angular route /dashboard with three distinct sections:
     - **Favorite Books**: Display favorite books with ID #favorite.
     - **Authors**: Display authors with ID #author.
     - **Recommendations**: Display book recommendations from a third-party service (openlibrary.org) with ID #recommended.

2. **Enhance User Experience**:
   - Enable users to register and create a profile, including uploading a profile image.
   - Provide a seamless login experience for registered users.
3. **Implement Search Functionality**:
   - Allow users to search for specific books or authors efficiently.
4. **Manage Favorites**:
   - Enable users to view all favorite book cards under the Favorite section.
5. **Integrate External Book Recommendations**:
   - Display book recommendations from the third-party service provider.

CHAPTER-2
SOFTWARE AND HARDWARE SPECIFICATION

Software Specifications

1. **Frontend**
   - **Framework**: Angular
   - **Languages**: TypeScript, HTML, CSS
   - **Libraries**: Bootstrap for styling, RxJS for reactive programming
2. **Backend**
   - **Framework**: ASP.NET Core
   - **Language**: C#
   - **Database**: SQL Server
   - **Authentication**: JWT (JSON Web Token)
   - **API Documentation**: Swagger
   - **Libraries**: Entity Framework Core for ORM, AutoMapper for object-object mapping
3. **Containerization**
   - **Tool**: Docker
   - **Configuration Files**: Dockerfile, docker-compose.yml
4. **Version Control**
   - **Platform**: GitLab
5. **Testing**
   - **Frontend**: Karma and Jasmine for unit testing
   - **Backend**: NUnit for unit testing
   - **End-to-End Testing**: Protractor

Hardware *Specifications*

1. **Development Environment**

   - **Processor**: Quad-core CPU (Intel i5)
   - **RAM**: Minimum 8 GB
   - **Storage**: Minimum 250 GB SSD
   - **Operating System**: Windows 10.

2. **Production Environment**
   - **Processor**: Quad-core CPU (Intel i5)
   - **RAM**: Minimum 16 GB
   - **Storage**: Minimum 500 GB SSD
   - **Operating System**: Windows Server.
   - **Docker Support**: Ensure the operating system supports Docker Engine and Docker Compose

3. **Database Server**
   - **Processor**: Quad-core CPU (Intel i5)
   - **RAM**: Minimum 16 GB
   - **Storage**: Minimum 500 GB SSD with RAID configuration for redundancy
   - **Operating System**: Windows Server.

CHAPTER-3

REVIEW OF LITERATURE

Literacy survey is the most important step in the software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next steps are to be determined which operating system and language can be used for developing the tool. Once the programmers need lot of external support. This support can be obtained senior programmers, from book or from websites. Before the building the system the above considerations are taken for developing the personal system.

Technologies :

1. Angular

Overview: Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Developed by Google, it provides a robust framework for building dynamic and responsive web applications.

Significance:

- **Component-Based Architecture**: Angular's modular structure allows for reusable components, enhancing code maintainability and scalability.
- **Two-Way Data Binding**: Simplifies the process of displaying and updating data in the UI, making the development process more efficient.
- **Dependency Injection**: Facilitates better organization and testing of the code by providing a way to inject services and other dependencies into components.
- **Extensive Tooling and Libraries**: Angular comes with a rich set of tools, including Angular CLI, which aids in the development, testing, and deployment of applications.

2. ASP.NET Core

Overview: ASP.NET Core is a cross-platform, high-performance framework for building modern, cloud-based, and internet-connected applications. Developed by Microsoft, it is known for its flexibility, performance, and productivity.

Significance:

- **Cross-Platform**: Supports development on Windows, macOS, and Linux, making it versatile for different development environments.
- **Performance**: Optimized for high performance, making it suitable for scalable and robust web applications.

- **Modular Architecture**: Allows developers to include only the necessary packages, reducing the application's footprint.
- **Built-in Dependency Injection**: Simplifies the management of service dependencies and enhances testability.

3. SQL Server

Overview: SQL Server is a relational database management system (RDBMS) developed by Microsoft. It is used to store and manage data in a structured way.

Significance:

- **Scalability and Performance**: Handles large volumes of data and complex queries efficiently.
- **Security Features**: Provides advanced security features like encryption, access controls, and auditing.
- **Integration with .NET**: Seamlessly integrates with ASP.NET Core, facilitating smooth data operations and management.
- **Tools and Services**: Comes with a suite of tools for data management, reporting, and analytics.

4. Docker

Overview: Docker is an open-source platform that automates the deployment, scaling, and management of applications using containerization.

Significance:

- **Consistency Across Environments**: Ensures that applications run the same way regardless of where they are deployed (development, testing, production).
- **Isolation**: Each container runs in its own isolated environment, which helps in avoiding conflicts and improving security.
- **Scalability**: Simplifies the process of scaling applications horizontally by adding more container instances.
- **Resource Efficiency**: Containers share the host OS kernel, making them lightweight compared to traditional virtual machines.

5. JWT Authentication

Overview: JSON Web Tokens (JWT) are a compact, URL-safe means of representing claims to be transferred between two parties. Used for authentication and information exchange.

Significance:

- **Security**: Provides a secure way of transmitting information between parties as a JSON object.

- **Stateless Authentication**: Eliminates the need for server-side session storage, making the system more scalable.
- **Cross-Domain Support**: Can be used across different domains and is widely supported by various technologies.

6. Swagger

Overview: Swagger is a set of tools for designing, building, documenting, and consuming RESTful web services.

Significance:

- **API Documentation**: Automatically generates interactive API documentation, making it easier for developers to understand and use the API.
- **Client Code Generation**: Can generate client libraries in various programming languages, speeding up the integration process.
- **Standardization**: Ensures that APIs adhere to a consistent and well-documented standard, enhancing maintainability and usability.

7. Testing Frameworks

Overview: The testing frameworks used include Karma and Jasmine for front-end testing, and NUnit for back-end testing.

Significance:

- **Karma**: A test runner that allows executing JavaScript code in multiple real browsers, ensuring compatibility and correctness.
- **Jasmine**: A behavior-driven development framework for testing JavaScript code, offering a clean and straightforward syntax.
- **NUnit**: A unit-testing framework for .NET applications, supporting test-driven development and ensuring code reliability.

8.Web API's

Overview:

Web APIs (Application Programming Interfaces) enable communication between different software applications over the web. They allow applications to access functionalities or data from other applications or services, typically through HTTP protocols. Web APIs are integral to modern software architecture, enabling interoperability, scalability, and modular development.

CHAPTER-4

SOFTWARE REQUIREMENT ANALYSIS

4.1 Existing System

The existing system, if any, may have limitations or gaps that the new **Book App** aims to address. Typically, an existing book management system might include:

- **Basic User Management**: Allowing users to register and log in.
- **Static Book Lists**: Displaying a static list of books without personalized recommendations.
- **Limited Search Functionality**: Basic search capabilities that might not be efficient or user-friendly.

4.2 Proposed System

The proposed **Book App** system aims to overcome the limitations of the existing system and introduce new features to enhance user experience, security, and scalability.

Key Features:

- **Comprehensive User Management**: Users can register with profile images, log in securely using JWT authentication, and manage their profiles.
- **Dynamic Dashboard**: A responsive dashboard view with sections for favorite books, authors, and recommendations.
- **Favorites Management**: Ability to add and view favorite books.
- **External Recommendations**: Integration with openlibrary.org to provide personalized book recommendations.

4.3 Feasibility Study

A feasibility study is essential to ensure that the proposed system is viable and practical. This involves analyzing the technical, economic, and operational aspects of the project.

Technical Feasibility:

- **Technology Stack**: The proposed technology stack (Angular, ASP.NET Core, SQL Server, Docker, JWT, Swagger) is modern, widely adopted, and well-supported.
- **Integration**: Integration with openlibrary.org and use of Docker for containerization are technically feasible and align with current best practices.
- **Scalability**: The proposed system is designed to be scalable, using containerization to handle increased loads.

CHAPTER-5

MODULES

5.1 User Management Module

Description: Handles all user-related functionalities, including registration, authentication, and profile management.

Key Features:

- **Registration**: Allows new users to sign up by providing personal details and uploading a profile image.
- **Login**: Enables registered users to log in securely using their credentials.
- **JWT Authentication**: Implements JSON Web Token (JWT) for secure authentication and authorization.
- **Profile Management**: Allows users to view and update their profile information.

5.2 Dashboard Module

Description: The central module of the application that displays the main dashboard view, including favorite books, authors, and book recommendations.

Key Features:

- **Default View**: The dashboard is the default view upon accessing the application.
- **Favorite Books**: Displays a list of favorite books added by the user.
- **Authors**: Shows a list of authors.
- **Recommendations**: Integrates with the openlibrary.org API to fetch and display book recommendations.

5.3 Favorite Management Module

Description: Manages the functionality related to adding and viewing favorite books.

Key Features:

- **Add to Favorites**: Allows users to add books to their favorites list.
- **View Favorites**: Displays all favorite book cards under the Favorite section on the dashboard.

5.4 Recommendation Module

Description: Integrates with a third-party book service provider (openlibrary.org) to fetch and display book recommendations.

Key Features:

- **Fetch Recommendations**: Connects to the openlibrary.org API to retrieve recommended books.
- **Display Recommendations**: Shows the recommended books under the Recommendations section on the dashboard.

5.5 **Author Module**

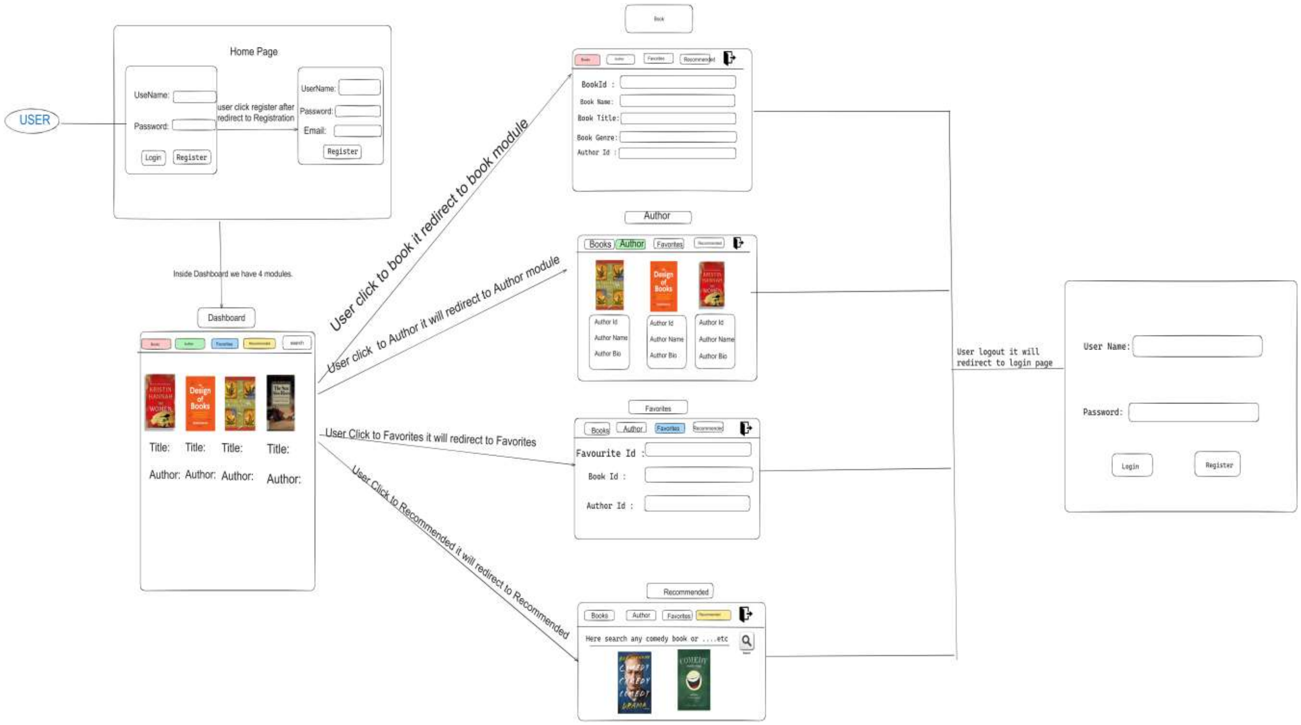
Description: The Author Module handles the retrieval, display, and management of author information within the application.

Key Features:

- **View Authors**: Displays a list of authors on the dashboard.
- **Search Authors**: Enables users to search for authors by name.
- **Author Details**: Provides detailed information about individual authors

CHAPTER-6

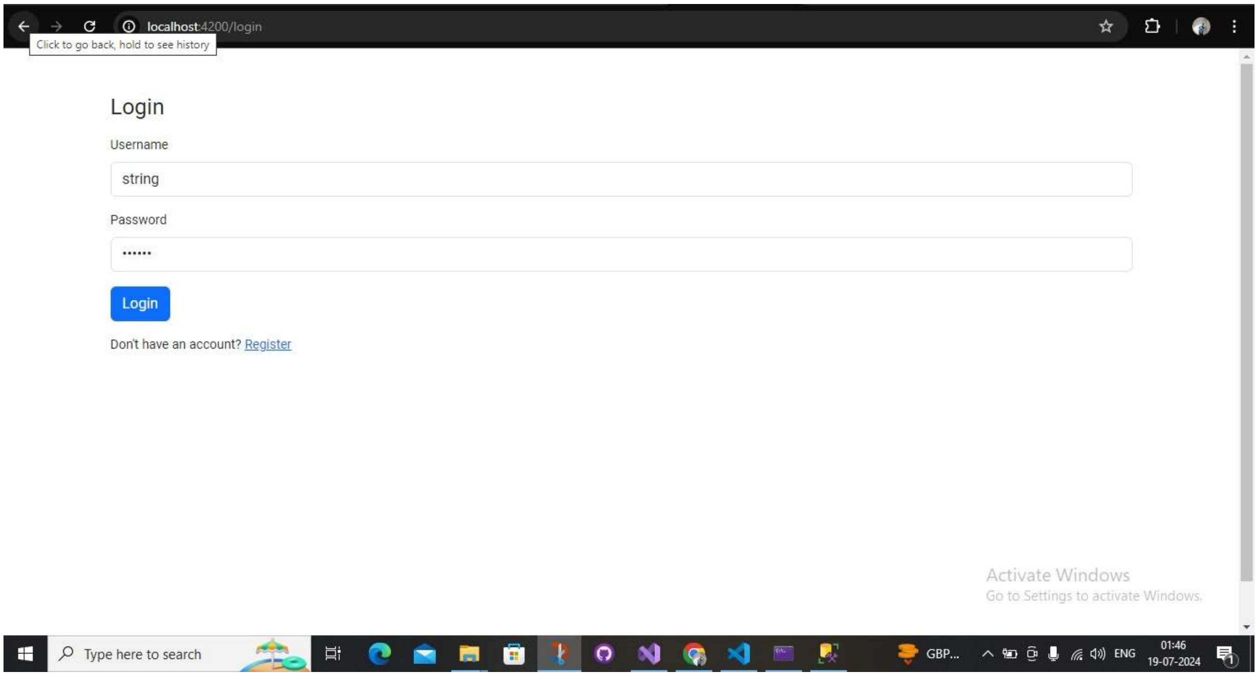
WIREFRAME & NAVIGATIONAL FLOW



CHAPTER-7

SCREENSHORTS

HOME SCREEN:



Login

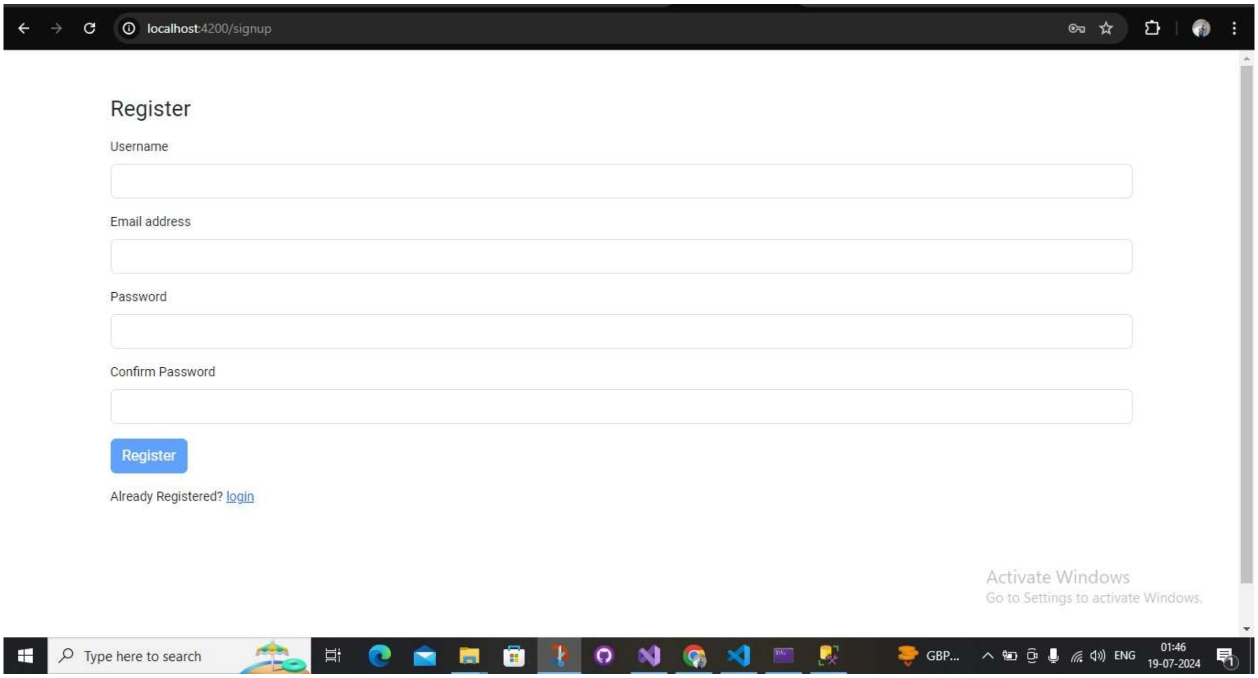
Username

string

Password

••••••

Don't have an account? Register



Register

Username

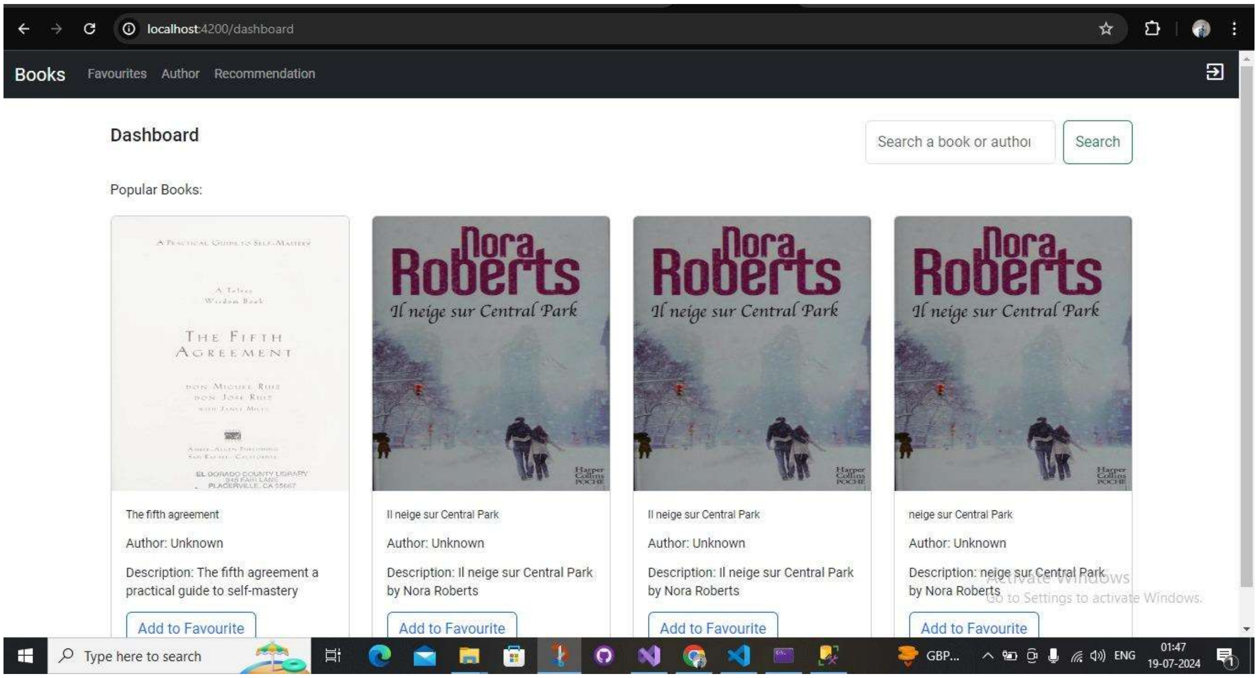
Email address

Password

Confirm Password

Already Registered? login

DASHBOARD:



Dashboard

Popular Books:

The fifth agreement

Author: Unknown

Description: The fifth agreement a practical guide to self-mastery

Il neige sur Central Park

Author: Unknown

Description: Il neige sur Central Park by Nora Roberts

Il neige sur Central Park

Author: Unknown

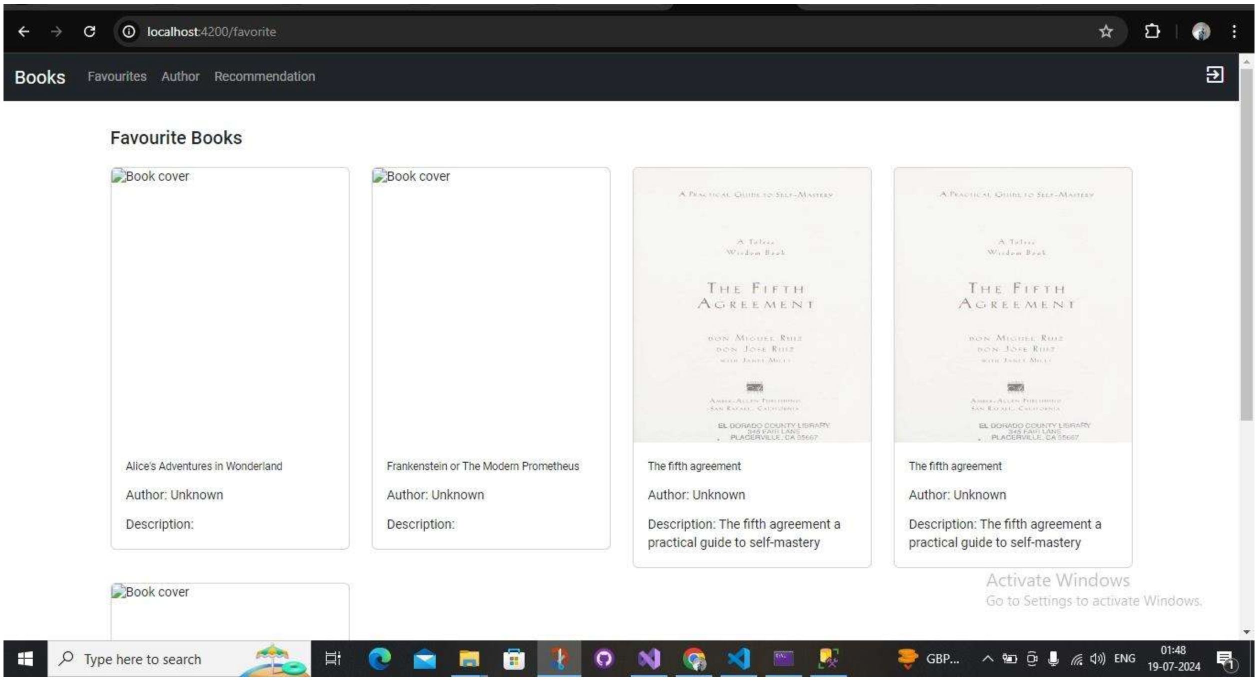
Description: Il neige sur Central Park by Nora Roberts

neige sur Central Park

Author: Unknown

Description: neige sur Central Park by Nora Roberts

FAVOURITES:



Favourite Books

Alice's Adventures in Wonderland

Author: Unknown

Description:

Frankenstein or The Modern Prometheus

Author: Unknown

Description:

The fifth agreement

Author: Unknown

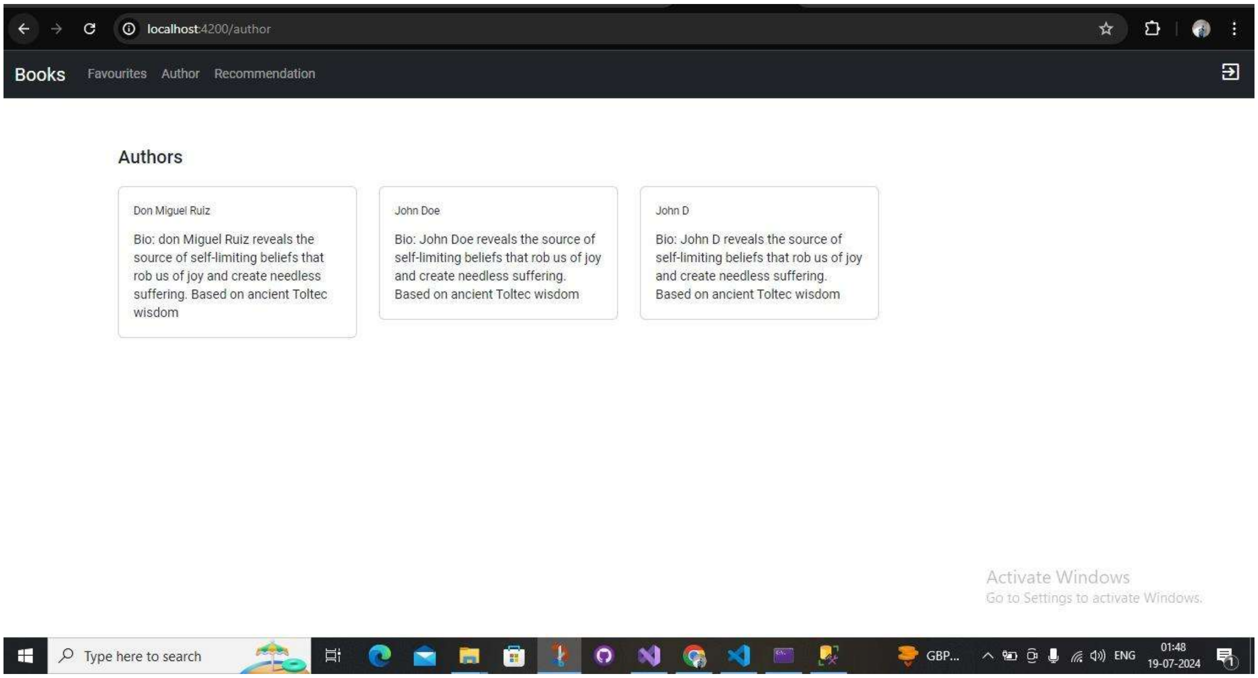
Description: The fifth agreement a practical guide to self-mastery

The fifth agreement

Author: Unknown

Description: The fifth agreement a practical guide to self-mastery

Authors:



Authors

Don Miguel Ruiz

Bio: don Miguel Ruiz reveals the source of self-limiting beliefs that rob us of joy and create needless suffering. Based on ancient Toltec wisdom

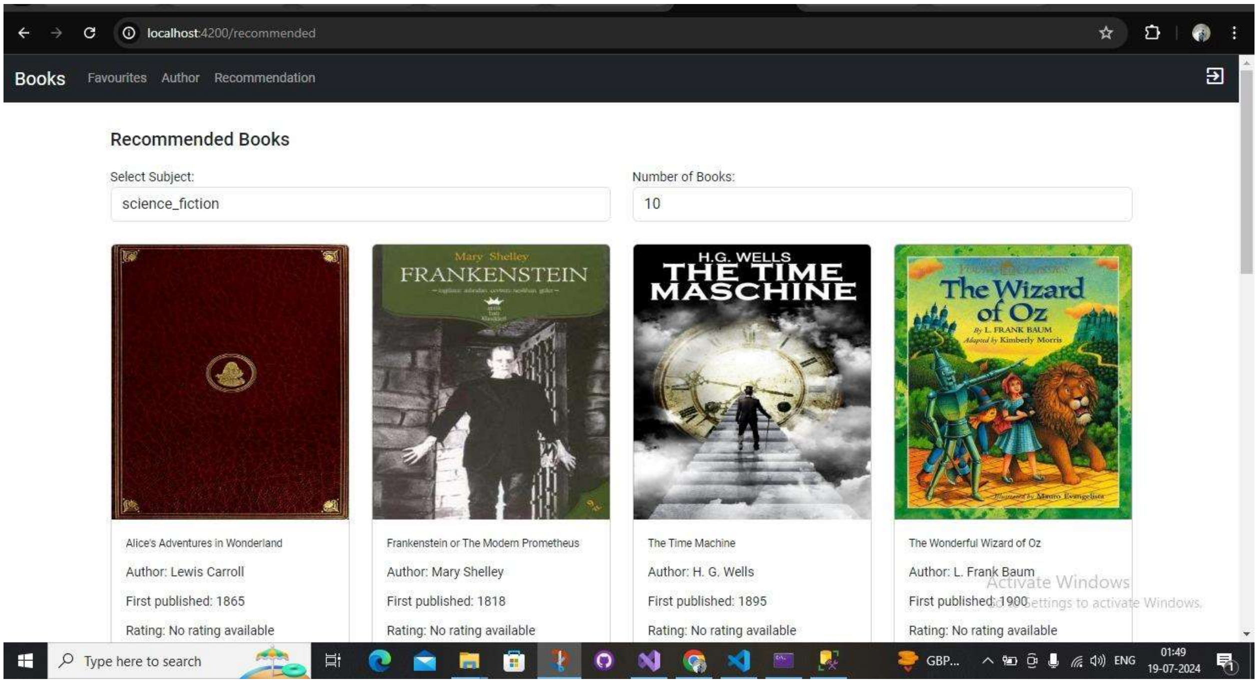
John Doe

Bio: John Doe reveals the source of self-limiting beliefs that rob us of joy and create needless suffering. Based on ancient Toltec wisdom

John D

Bio: John D reveals the source of self-limiting beliefs that rob us of joy and create needless suffering. Based on ancient Toltec wisdom

RECOMMENDED BOOK:



Recommended Books

Select Subject: science_fiction

Number of Books: 10

Alice's Adventures in Wonderland
Author: Lewis Carroll
First published: 1865
Rating: No rating available

Frankenstein or The Modern Prometheus
Author: Mary Shelley
First published: 1818
Rating: No rating available

The Time Machine
Author: H. G. Wells
First published: 1895
Rating: No rating available

The Wonderful Wizard of Oz
Author: L. Frank Baum
First published: 1900
Rating: No rating available

CHAPTER-8

CONCLUSION

The **Book App** project has successfully met its objectives, providing a powerful, user-friendly platform for managing book-related information. By leveraging modern technologies, best practices, and a modular design, the application ensures a high level of functionality, security, and scalability. The project not only addresses the needs of current users but also provides a solid foundation for future growth and enhancements. Overall, the **Book App** is a significant achievement, offering a comprehensive, scalable, and engaging solution for book enthusiasts.