**Project Title: Online Book Management System**

**Introduction:**

The Online Book Management System is a web-based application designed to provide users with a platform for managing their personal book collections, connecting with friends to share books, and engaging in book-related activities. This system leverages modern technologies such as .NET Core for web and API development and MSSQL for database management.

**Project Scope:**

The primary goal of this project is to create an intuitive and feature-rich web application that allows users to perform essential book-related tasks, such as adding, deleting, updating, and listing books. The scope of the project extends beyond basic book management to include the following key features:

**Extended Book Information**: Users can input additional details about each book, such as the number of pages, price, and the language in which the book is published.

**User Reviews:** The system enables users to share their thoughts and opinions on the books they've read, facilitating discussions and helping others make informed reading choices.

**Book Sharing:** Users can share their books with friends and fellow readers, expanding the reach of each book within the community.

**Search Functionality:** An efficient search feature allows users to quickly locate specific books within their collection, making it easy to find and manage their reading materials.

**Target Audience:**

This application is designed for book enthusiasts, avid readers, and anyone looking for a convenient way to manage their book collections, connect with others who share their passion for reading, and discover new books of interest.

**Technical Details:**

The technical foundation of this project includes .NET Core for building robust web and API applications and MSSQL for managing the database. These technologies have been chosen for their reliability, scalability, and the extensive developer ecosystem they offer.

In the following sections of this documentation, we will delve into the detailed specifications, architecture, and implementation details of the Online Book Management System, providing a comprehensive guide for project stakeholders, developers, and users.

**User Roles and Descriptions:**

**Registered User:**

**Description:** A registered user is an individual who has created an account within the Online Book Management System. They can access the full range of features and functionalities provided by the application.

**Capabilities:**

**Book Management:** Registered users can add new books to their collection, edit existing book details, and remove books from their library.

**Book Reviews:** Users can write and post reviews for books they have read, allowing them to share their opinions and insights with the community.

**Book Sharing:** Registered users can share their books with friends or other users, expanding the reach of their collection and fostering a sense of community.

**Friendship Management:** Users can connect with other users on the platform, sending and accepting friend requests.

**Search and Discovery:** They have access to the search functionality to find specific books, friends, or book-related content within the platform.

**Guest User:**

**Description:** A guest user is an individual who accesses the application without creating an account or logging in. Guest users have limited access to the platform's features.

**Capabilities:**

**Browsing**: Guest users can browse and search for books and book-related content within the platform.

**Book Details:** They can view detailed information about books, including reviews, ratings, and descriptions.

**No Data Modification:** Guest users cannot add, edit, or delete books, write reviews, or engage in book sharing activities.

**Limited Interactions:** Guest users cannot send or receive friend requests or interact with the social features of the platform.

**Anonymous:** Guest users can use the platform anonymously without the need for registration.

**Admin User:**

**Description:** An admin user is a privileged user role with elevated access rights. Admins are responsible for managing and moderating the platform's content and user activities.

**Capabilities:**

**User Management:** Admin users can manage registered users, review reported content, and enforce community guidelines.

**Content Moderation:** They have the authority to review and remove inappropriate or violating content, including reviews, book entries, and user accounts.

**System Configuration:** Admins can configure system settings, manage user roles, and oversee the overall functionality of the platform.

**Reports and Analytics:** They have access to data analytics tools to gain insights into user behavior and platform usage.

**Notification Broadcast:** Admins can send important announcements or notifications to all users.

**Requirements:**

**Backend**

**- ASP.NET Core WEB API (programming language):** is a framework for building web APIs using the ASP.NET Core platform, allowing developersto create RESTful services that can be consumed by various clients overHTTP.

**- Visual Studio 2022 (IDE):** is the latest version of Microsoft's integrated development environment (IDE) for software development, providing a comprehensive set of tools and features to streamline the process of building applications for various platforms, including desktop, web, mobile, and cloud.

**- Microsoft SQL Server (database):** is a relational database managementsystem (RDBMS) developed by Microsoft, commonly used for storing and managing large amounts of structured data. It offers robust features for data management, querying, and scalability, making it a popular choice for enterprise-level applications.

**- SmarterASP.net (backend + database deployment):** is an online shopping comparison website that helps users find the best deals on various products by aggregating information from different retailers and presenting it in a user-friendly interface. ASP.NET could potentially be used as the underlying technology for building and powering the website's backend infrastructure.

**Frontend**

**- React (programming language):** is an open-source JavaScript library forbuilding user interfaces. It allows developers to create reusable UI components and efficiently update the user interface in response to changes in data, providing a flexible and efficient approach to building interactive web applications.

**- Visual Studio Code (IDE):** is a free and open-source source code editor developed by Microsoft. It provides a lightweight and highly customizable environment for editing code, with support for a wide range of programming languages, extensions, and integrated tools for debugging and version control.

**- Vercel (frontend deployment):** is a cloud platform for deploying and hosting websites and web applications. It specializes in providing fast and scalable static site hosting, serverless functions, and continuous deployment, enabling developers to easily deploy and manage their projects with minimal configuration and maximum performance.

**Repository**

A computer screen shot of a computer

Description automatically generated**- GitHub (project repositories):** is a web-based platform that provides hosting for version control repositories. It offers collaboration features, issue tracking, and pull request functionality, making it widely used for software development and open-source projects.  
  
  
 **Database Design:**

**Project Architecture:**

**MVC (Model-View-Controller):** The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application.The library application will consist of 2 parts. Instead of consisting of a single project, we will develop the application in different projects for back-end and front-end. We will use React in the front-end project and provide data from the relevant endpoints here. model and controller components are included in the back-end project, and the view component will be in the front-end project.  
  
  
  
**Project Development Stages:**The project will primarily consist of 3 parts: database, backend and frontend projects. We will successfully complete these 3 parts in a 4-week period until the final presentation of the project. In the first week, a database model will be created, backend integration will be made, and backend and frontend projects will be established. In the 2nd week, some endpoints related to users and books will be completed and navbars will be created in the frontend project. In the 3rd week, with the completion of endpoints such as authors and book sharing, favorites, likes, all work will be completed in the backend project and we will concentrate on the frontend area in the remaining period. In the process until the final presentation, that is, in the 4th week, project documentation will be completed, demo presentations will be worked on in the projects and if there are any errors or problems, we will focus on them. After the user and admin panels are completed, the projects and database will be uploaded to free servers and will be ready for the final presentation.  
 **Project Management:**  
  
  
  
  
A screenshot of a calendar

Description automatically generated  
A screenshot of a computer

Description automatically generated

**Security:**

Authentication is the process of verifying the identity of a user or system, typically through the use of credentials like passwords, tokens, or biometric data. It ensures that the entity accessing a resource is who they claim to be.

Authorization, on the other hand, is the process of defining and enforcing access control rules and permissions that determine what actions an authenticated user or system can perform within a system or on a resource. It ensures that authenticated entities are granted appropriate levels of access and privileges based on their roles or attributes.  
  
High-level security measures have been taken by using tokens and refresh tokens in our project. On the database side, we prevent access to data because authentication with username and password is required. Additionally, Since we keep the github folder where our project is located private, we have prevented external attacks on the project settings. Finally, since we have configured the cors settings for the backend project, we only respond to requests from the frontent project. In this way, we take great precautions against foreign requests that may come from outside.

**Deployment and Publishment:**

Our project consists of 3 parts: backend, frontend and database. When we deploy the database to the server, we will be able to use our data easily. When we deploy the backend project, we will be able to easily test our endpoints, and finally, when we deploy the frontend project, users will be able to use the interface of the final library application. That's why we first deployed the database, then the backend project, and lastly the front project. We uploaded the database and backend project to smarterasp.net, which has a 60-day free version, and the frontend project with Vercelle, one of the most used deploy applications. Since we handle the deployment operations free of charge, we have limited use of the database and backend project, and we cannot use the domain name we want in the frontend project. In the future, after the project is implemented in real life, we plan to purchase the domain name, use better deployment applications and upload our projects there.