

# Online BookStore

## Description

Designed and developed a dynamic e-commerce platform integrating leading-edge technologies to deliver a seamless user experience and efficient system management. Leveraged a powerful technology stack including Asp.net Core MVC, SQL Server, Entity Framework, Bootstrap v5, HTML5, and JavaScript.

## Project Goal

The goal of this project is to create a seamless, scalable, and efficient online book store that offers users an intuitive browsing and shopping experience. The application should enable customers to browse and purchase books with ease, while providing administrators with efficient tools for managing the store's content, users, and orders. By using modern web technologies, the platform will offer high performance, security, and flexibility, ensuring a sustainable and robust solution for future scalability. Key goals include:

**User Experience:** Provide an intuitive, responsive, and secure shopping experience for users on all devices.

**Scalability:** Build the application with scalability in mind, ensuring it can handle increasing product and user volumes.

**Efficient Management:** Provide administrators with an easy-to-use back-end to manage products, orders, and users.

**Secure Transactions:** Integrate a secure payment gateway and implement robust authentication and authorization mechanisms to protect user data.

**High Performance:** Optimize the application for performance, including efficient product search and display features (via DataTables API).

# Requirements

## Functional Requirements

### 1. Functional Requirements

#### 1.1 Admin Requirements

- **Admin Login:** The admin must be able to log in to the website.
- **Register New Admin:** The admin must be able to register another admin user.
- **Add Users:** The admin must have the capability to add new users to the system.
- **Determine User Authority:** The admin must be able to assign roles and permissions to users.
- **Add Categories:** The admin must be able to create new categories for books.
- **Add Products:** The admin must be able to add new books to the store.
- **Control Sales and Prices:** The admin must be able to manage book prices and sales.
- **Manage User Details:** The admin must be able to view and edit user information.
- **Manage Orders:** The admin must be able to manage orders, including canceling or expediting them.
- **Manage Admin Account:** The admin must be able to manage their own account settings.
- **Process Orders and Payments:** The admin must have the ability to process orders and handle payments.

#### 1.2 Client Requirements

- **Client Registration:** Clients must be able to register for a new account.
- **Client Login:** Clients must be able to log in to their accounts.
- **View Product Details:** Clients must be able to view detailed information about products.
- **Place Orders and Process Payments:** Clients must be able to place orders and complete payment transactions.
- **Check Order Status:** Clients must be able to check the status of their orders. ●
- **Search for Products:** Clients must be able to search for products easily. ●

# Non-Functional Requirements

## 2. Non-Functional Requirements

### *2.1 Security*

- **Data Encryption:** User data must be encrypted both in transit and at rest to ensure privacy and security.

### *2.2 Performance*

- **Loading Speed:** The website should load within a few seconds for users to enhance user experience.
- **Concurrent Users:** The system should support up to 2000 concurrent users without performance degradation.

### *2.3 Backup and Recovery*

- **Daily Backups:** The website should perform daily backups of the database.
- **Recovery Plan:** A plan should be in place for restoring services in the event of data loss or corruption.

### *2.4 Integration*

- **Payment Processing:** The website should integrate with banking systems to facilitate payment processing.
- **Social Media Integration:** The website should allow users to log in and register using their social media accounts.

### *2.5 Usability*

- **User Interface:** The website should have a simple and intuitive interface.
- **Homepage Display:** The website should prominently display featured books on the homepage.

## 2.6 Maintainability

- **Code Documentation:** The codebase should be well-documented to facilitate easy updates and maintenance.

## 2.7 Scalability

- **Scalable Architecture:** The architecture should support scaling to accommodate a 50% increase in traffic during peak seasons, such as holidays.

# Technologies and Patterns Utilized

## 1. Architectural and Design Patterns

- **ASP.NET Core Identity:** Provides robust authentication and authorization, ensuring secure user management.
- **Automated Database Migrations:** Facilitates easy setup and updates with integrated seed database migrations.
- **Dependency Injection:** Enhances code organization and maintainability by managing dependencies effectively.
- **Unit of Work Pattern:** Manages database transactions effectively to ensure consistency.
- **N-Tier Architecture:** Ensures scalability and flexibility of the application by separating concerns.
- **Repository Pattern:** Utilized for efficient database management and separation of concerns.

## 2. Frontend Technologies

- **HTML:** Structures the web pages for content presentation.
- **CSS:** Styles and designs the user interface.
- **Bootstrap:** A CSS framework that helps build responsive and fast-loading interfaces.
- **JavaScript:** Adds interactivity to the website.

### 3. Backend Technologies

- **ASP.NET MVC:** The framework used for building the web application.
- **C#:** The programming language used for developing the application logic.
- **Entity Framework:** Facilitates object-relational mapping (ORM) for interaction with the database.
- **LINQ :** Used for querying data from collections and databases in a concise manner.

### 3. Database

- **SQL Server:** The database used to store application data.

### 4. Additional Technologies

- **IIS:** Using as a virtual server during development time o

### Features

- **Product Management:** Streamlined functionality for adding, editing, and deleting products.
- **Shopping Cart:** Intuitive shopping cart functionality for a better user experience.
- **Order Management System:** Comprehensive system for tracking orders, shipping, and cancellation options.
- **User and Role Management:** Flexible management of users and roles with secure registration and password recovery mechanisms.
- **Stripe Payment Integration:** Seamless transactions through the integration of the Stripe payment gateway.
- **Multi-Image Upload:** Functionality for enhancing product presentation with multiple images.
- **View Components:** Utilized for building reusable UI components.
- **External Login:** Integration of social login options for easier user access.
- **Email Confirmation:** Mechanism for confirming user registrations to enhance security.
- **DataTables API:** Integrated for efficient and dynamic product display.