Library Management System

**Предназначение**

* Цялостно решение за дигитализация на библиотечни процеси
* Автоматизация на ежедневните библиотечни операции
* Подобряване на ефективността на работа
* Улесняване на потребителите при заемане на книги

**Целеви Потребители**

* Библиотекари
* Администратори на библиотеки
* Читатели (крайни потребители)

**Основна Информация**

* **Име на проекта**: Система за Управление на Библиотека
* **Технологичен стек**:
  + .NET 8.0 (най-новата версия на .NET платформата)
  + Entity Framework Core (за работа с база данни)
  + SQL Server (система за управление на бази данни)
  + ASP.NET Core Identity (за автентикация и оторизация)

**Управление на Потребители**

* **Роли в системата**:
  + Администратор (пълен достъп до системата)
  + Потребител (ограничен достъп до основни функции)
* **Сигурност**:
  + Защитена автентикация
  + Оторизация базирана на роли
  + Защита на личните данни
* **Потребителски профили**:
  + Персонализирани настройки
  + История на заемания
  + Управление на лична информация

**Управление на Книги**

* **Пълен CRUD цикъл**:
  + Създаване, четене, обновяване, изтриване на записи
* **Валидация на данни**:
  + Проверка за уникалност на ISBN
  + Валидация на задължителни полета
* **Разширени функции**:
  + Поддръжка на множество автори за една книга
  + Категоризация по жанр

**Система за Заемане**

* **Проследяване на заемания**:
  + Дата на заемане, очаквана дата на връщане, реална дата на връщане
* **Статуси на заемане**:
  + Чакащо одобрение, активно заемане, върната книга, просрочено връщане
* **Мониторинг**:
  + Следене на срокове, известия за просрочване, история на заеманията

**Backend Архитектура**

* **ASP.NET Core MVC**
* **Entity Framework Core**: Code First подход, миграции
* **SQL Server интеграция**: Оптимизирани заявки, индексиране на данни

**Дизайн на Базата Данни**

* **Нормализирана структура**: Премахване на редундантност, оптимизация на заявки
* **Релации**: Един към много, много към много, каскадни изтривания
* **Ограничения**: Уникалност на данни, референциална цялост

**Модулно Тестване**

* **Покритие на кода**: ~90%
* **Използвани технологии**: NUnit, in-memory база данни, Mock обекти
* **Видове тестове**: Тестове на моделите, контролерите, бизнес логиката, интеграционни тестове

**Качество на Кода**

* **Принципи на чист код**: DRY, KISS, SOLID
* **Обработка на грешки**: Try-catch блокове, логване на грешки

**Автентикация**

* **Хеширане на пароли**: Солт за допълнителна защита
* **Управление на сесии**: Сигурни бисквитки, таймаут на сесия
* **Двуфакторна автентикация**

**Защита от Атаки**

* **XSS защита**: Валидация на входни данни
* **SQL инжекции**: Параметризирани заявки
* **CSRF защита**: Антифоргери токени

**Административен Панел**

* Управление на книги, потребители, справки и отчети

**Потребителска Част**

* Каталог на книги, заемане, профил

LibraryManagementSystem/  
├── Controllers/  
├── Models/  
├── Data/  
├── Views/  
└── Services/

**Основни таблици**: Books, Authors, Genres, Publishers, Borrowings

* Ефективно и модерно управление на библиотечни ресурси
* Подобряване на потребителското изживяване
* Оптимизиране на процесите чрез автоматизация

A computer screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a program code

AI-generated content may be incorrect.

A computer screen shot of text

AI-generated content may be incorrect.