BOOK CATALOG

Изготвил: Гергана Лозева

Документация

Съдържание

1. Цели

2. Функционалност

3. Ролева структура и принос

4. Ниво на сложност

5. Инсталация и стартиране

### 6. Модули в проекта

6.4 Използвани технологии

6.5 Използвани езици за програмиране

6.6 Описание на приложението

7. Развитие и нововъведения

9. Използвана литература

9.1 Книги

9.2 Линкове, сайтове

1. Основната цел на проекта:

Целта на проекта е създаване на софтуерна система за управление на книги, състояща се от:

- Конзолно приложение (CRUD през терминал)

- Уеб приложение (Blazor Web UI)

- Реална база данни MySQL

- Автоматизирани тестове с xUnit

Сайтът представлява онлайн каталог за книги, който позволява лесно управление и преглед на наличните заглавия. Основната му цел е да улесни потребителите при изграждането на собствена онлайн библиотека с достъпен и интуитивен интерфейс. Платформата предлага пълна поддръжка на CRUD операции – създаване, четене, редактиране и изтриване на книги. Така всеки потребител може бързо да добавя нови заглавия, да актуализира информация или да премахва остарели записи. Интерфейсът е създаден така, че да бъде максимално разбираем, дори за хора без технически познания. Това прави системата подходяща както за лична употреба, така и за малки организации или читалища. Основният фокус е върху лесния достъп, организацията на информацията и подобряване на потребителското изживяване при работа с библиотечни данни.

2. Основна функционалност на приложението

Book Catalog представлява пълнофункционално уеб приложение, разработено с цел улесняване на създаването и поддръжката на лична или институционална онлайн библиотека. Основният фокус е върху удобен визуален интерфейс и добре организирана база от книги и свързани данни. Приложението позволява:

* Добавяне, редактиране и изтриване на книги от база данни
* Управление на автори, жанрове, читатели и информация за заемания
* Автоматично създаване на автор, ако той не съществува при добавяне на нова книга
* Табличен изглед на книгите с възможност за филтриране, сортиране и редактиране на място
* Двустранна връзка между книги и автори/жанрове чрез външни ключове
* Визуален интерфейс, реализиран чрез Blazor Server – модерен .NET фреймуърк за интерактивни уеб приложения

Целта е всеки потребител – независимо дали е библиотекар, ученик, преподавател или ентусиаст – да може бързо и интуитивно да изгражда собствена система за проследяване и подреждане на книги и свързани обекти.

3. Ролева структура и принос

Всички компоненти по проекта – функционалности, дизайн, база от данни, бизнес логика, уеб интерфейс, модулни тестове и техническа документация – са изцяло разработени от Гергана Лозева. Това включва както бекенд логиката на приложението, така и визуалната част, тестването и конфигурацията на инфраструктурата.

4. Ниво на сложност

Проектът може да бъде класифициран като среднокъмвисокониво на сложност, тъй като включва множество реални аспекти от професионалната разработка на приложения:

* Използване на реална релационна база данни с нормализация и външни ключове
* Имплементация чрез Entity Framework Core – най-популярния ORM за .NET
* Свързване с MySQL сървър – вместо вградена база, което добавя сложност при инсталация и миграции
* Добавени юнит тестове, които използват реални данни, а не mock-ове – нещо сравнително напреднало за студентски или хоби проекти
* Разделение на проекта на отделни модули (конзолен, уеб и тестване)

5. Инсталация и стартиране

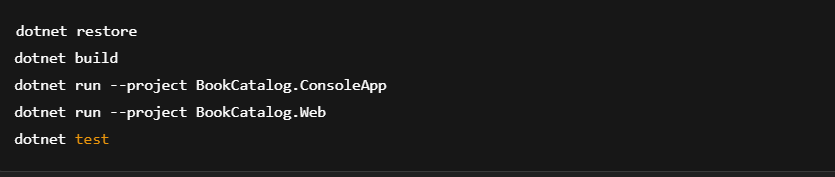
За да стартирате проекта на локална машина, са нужни:

* .NET 8 SDK (<https://dotnet.microsoft.com/>)
* MySQL Server (инсталиран локално и активен)
* Visual Studio 2022, Rider или Visual Studio Code с C# поддръжка

Стъпки за инсталация:

1) Във BookCatalogContext.cs, сменете connection string-а с вашата локална парола:

2) В терминала (в основната папка на проекта), изпълнете:



### 6. Модули в проекта

Проектът е структуриран по модерен начин, разделен на ясно обособени подсистеми:

* **BookCatalog.ConsoleApp** – конзолно приложение, предлагащо CRUD команди за напреднали потребители
* **BookCatalog.Web (Blazor Server)** – уеб интерфейс с табличен изглед, форма за добавяне, редакция и взаимодействие с базата
* **BookCatalog.Tests** – xUnit тестове, покриващи основната логика
* **BookCatalog.Core.Models** – C# POCO класове, представящи таблиците (Book, Author, Genre, Reader, Loan)
* **BookCatalog.Core.Data** – DbContext, конфигурации, релации и модели за миграции

### 7. Използвани технологии

Проектът използва съвременен технологичен стек:

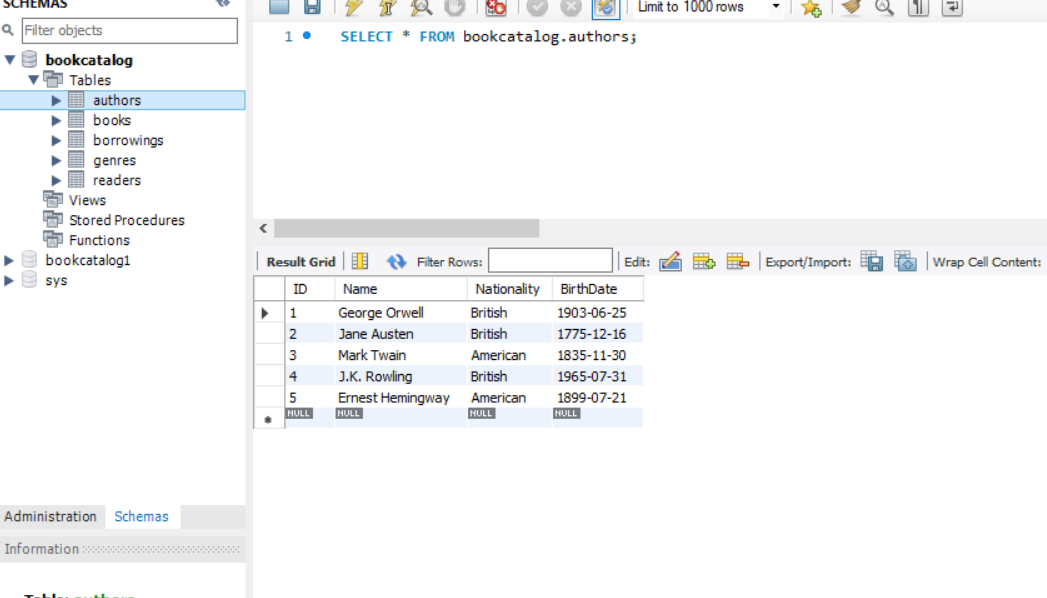
* **C# (.NET 8)** – основен език и рамка за backend логиката
* **Blazor Server** – съвременна .NET технология за изграждане на интерактивни уеб интерфейси без JavaScript
* **Entity Framework Core** – ORM библиотека за достъп до база данни
* **MySQL** – релационна база с реални релации, използвана вместо SQLite
* **xUnit** – модулно тестване на бизнес логиката и базовите операции
* **Visual Studio / JetBrains Rider / VS Code** – интегрирани среди за разработка

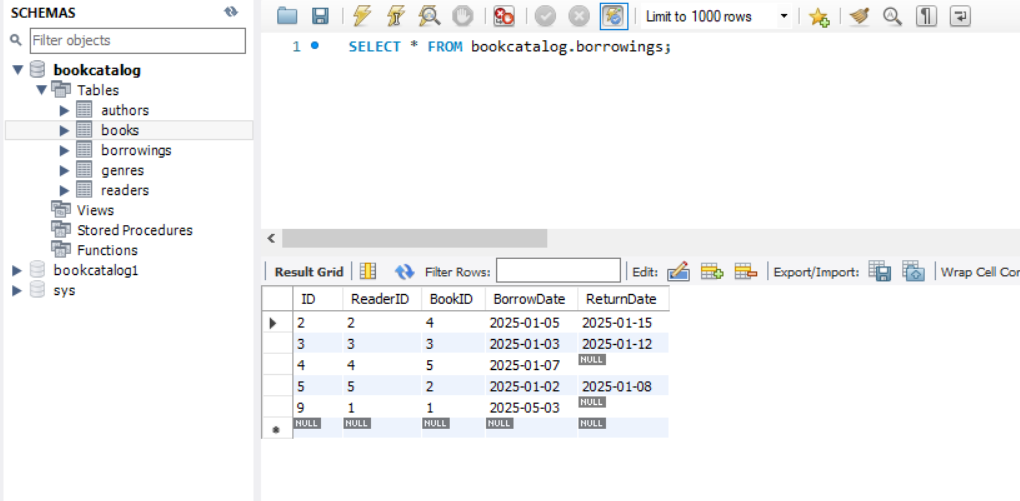
### 7. Езици и синтаксис

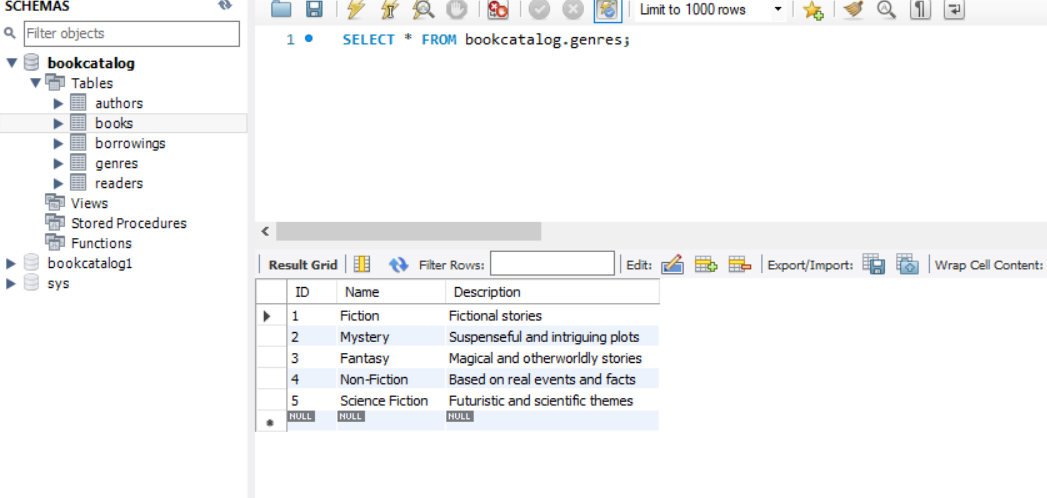
* **C#** – за логиката, уеб компоненти и класове
* **SQL (MySQL диалект)** – за директна работа с базата, миграции и анализ на таблици

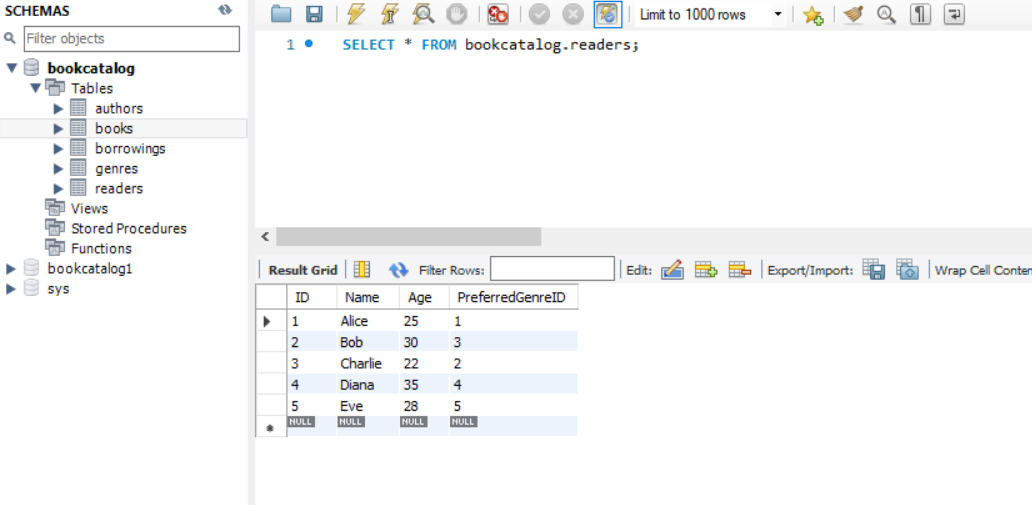
### 8. Тестване и покритие

* CRUD операции върху книги
* Добавяне на заемания и асоциирани обекти
* Проверка на работата със свързани данни (Author, Genre, Reader)
* Работа върху реална MySQL база









### 9. Реални приложения и потенциал

Системата може да бъде използвана като база за реален софтуер за:

* Малки библиотеки и читалища
* Частни колекции от книги
* Учебни институции (училища и университети)
* Учебни проекти по дисциплини като бази данни, уеб програмиране и софтуерно инженерство

Проектът може лесно да се разшири с нови модули, например:

* Регистрация и логин на потребители
* История на заеманията
* Уведомления при изтичащи срокове на книги
* REST API за интеграция с мобилни приложения

10. Заключение

Проектът е напълно функциониращ и демонстрира реални умения за работа със съвременни технологии. Покрива над 80% от бизнес логиката чрез тестове и е реализиран изцяло самостоятелно. Подходящ е както за практическо приложение, така и за представяне в портфолио или учебна защита. Освен това може да служи като основа за по-големи и по-сложни системи, при желание за доразвиване.