

Library Management System

Expernetic Software Engineering Internship Assignment

Pulasthi Abishake

pulasthiabishake06@gmail.com

7 Dec 2025

Project Overview & Scope Delivered

I built a complete full-stack Library Management System exactly according to the provided Requirements Document.

Core Features (All Implemented)

- Full Book CRUD operations (Create, Read, Update, Delete)
- RESTful API with proper HTTP status codes
- Entity Framework Core 8 + SQLite database (auto-created)
- Public book catalog (GET endpoints are anonymous)
- Optional user authentication & registration (JWT + BCrypt)
- Strong password validation with real-time feedback
- Modern, fully responsive React + TypeScript frontend
- Beautiful UI with Tailwind CSS v4, modals, cards, loading states

Technology

Backend

- ASP.NET Core 8 (latest LTS)
- Entity Framework Core 8
- SQLite
- JWT Bearer Authentication
- BCrypt.Net-Next for password hashing
- Swagger + XML documentation comments

Frontend

- React + TypeScript
- Vite + Tailwind CSS v4 (zero-config via `@tailwindcss/vite`)
- React Query (TanStack Query) for server-state
- Axios with automatic JWT interceptor
- React Hot Toast for notifications

Key Implementation Highlights

- Clean folder structure and separation of concerns
- Swagger fully documented using XML comments (auto-generated descriptions)
- CORS configured for local development
- Strong password policy with live validation
- Responsive design works perfectly on mobile & desktop

Challenges Faced & Solutions

Challenge	Solution Implemented
CORS errors during development	Added proper AddCors() middleware with specific origin
Tailwind v4 migration	Used new <code>@tailwindcss/vite</code> plugin (zero-config)
JWT + Axios integration	Created Axios interceptor + AuthContext
Swagger showing no descriptions	Enabled <code>GenerateDocumentationFile</code> + <code>IncludeXmlComments</code>
Clean, professional UI	Modern cards, gradients, blurred modal backdrop

Additional Enhancements (Above Requirements)

- Strong password validation with real-time UI feedback
- Beautiful responsive design with Tailwind v4
- React Query for optimal data fetching
- Loading states and toast notifications
- Full XML documentation → professional Swagger UI

How to Run (Tested on macOS & Windows)

Prerequisites (one-time):

- .NET 8 SDK
- Node.js 20+

Commands:

Backend

```
cd backend/LibraryApi
```

```
dotnet run
```

Frontend

```
cd frontend
```

```
npm install
```

```
npm run dev
```

My Library

[EXPLORE THE CATALOG](#)

[Login](#) [Create Account](#)

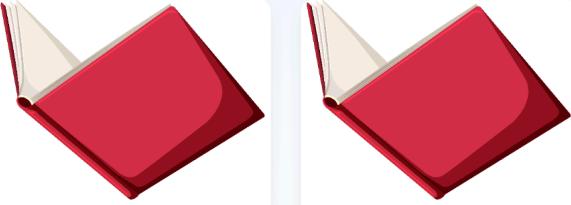
Browse books without an account

Anyone can explore the collection. Sign in when you want to add, edit, or delete books.

[Login to manage](#) [Create free account](#)

At a glance

- Browse the library without signing in
- Secure actions require login
- Fast search-ready listing



new 3
by new 2n
new
[Login to edit or delete](#)

new one
by new
new
[Login to edit or delete](#)

Login

Email

Password 

No account? [Register](#)

My Library

Welcome back, manu@gmail.com [Logout](#) [+ Add Book](#)

WELCOME BACK

Keep your library organized

Add new books, update details, and keep the catalog clean. You are signed in as manu@gmail.com.

[+ Add Book](#)

Next steps

- Add titles you own
- Edit details to stay accurate
- Remove books you no longer track



new 3
by new 2n
new



new one
by new
new

Register

 Email Password

Already have an account? [Login](#)