Expernetic - Software Engineer Intern – Assignment

Library Management System

Sathira Wijeratne

August 2025

Project Overview

This report outlines the development process, implementation details, challenges faced, and key insights gained from the Software Engineering Internship assignment. The goal was to build a full-stack Library Management System with a C#.NET backend and a React/TypeScript frontend within one calendar week.

Development Process

I started the project by initializing the backend and frontend using Visual Studio, but since I was not that familiar with Visual Studio I transition to Visual Studio Code for development as I am more comfortable with that IDE. There were issues at first, while the project ran through Visual Studio, it did not run through Visual Studio Code due to port different ports being used depending on which IDE the project was run.

Once that was fixed, I adopted a backend-first approach, which allowed me to establish the core data model and API endpoints before building the user interface.

Throughout the process, I found myself constantly learning and adapting. I relied on online resources, including tutorials, guides, documentation, and videos, to understand new concepts and figure out solutions. The project was a full-stack effort, with ASP.NET Core handling the backend and React with TypeScript powering the frontend.

Backend Implementation

The backend was developed using ASP.NET Core 8.0, focusing on creating a robust and secure RESTful API for managing books and users.

- **Database:** Entity Framework Core was used with SQLite database. This was a new experience for me, and I had to learn about things like migrations to manage the database schema changes. The database schema included a Book entity for storing book titles, authors, and descriptions, as well as a User entity to handle authentication.
- **API:** I designed RESTful API endpoints for all the required CRUD operations on the Book entity. To handle user management and secure certain endpoints, I also implemented JWT-based authentication as well as endpoints to login and register.
- **Security:** For user authentication, I configured JWT Bearer authentication to handle token validation. A critical part of this was implementing password security using BCrypt for hashing to ensure user passwords were not stored in plain text.
- **Architecture:** The API was designed with a clear RESTful structure. I also implemented global exception handling to centralize error management and used data annotations for input validation, which was a new concept for me. Since the API documentation was generated using Swagger/OpenAPI, it made it easier to test the endpoints.

Frontend Implementation

The frontend was built with React 18 and TypeScript, using Vite as the build tool.

- **Tech Stack:** React and TypeScript was a new combination for me. I also integrated Material-UI (MUI) for the visual components. I picked it because I had used it previously for a different project, althrough I was still new to it. Therefore I had to learn on the fly to style the application. React Router DOM was used to handle navigation and protected routes.
- **UI Components:** I developed different components for key functionalities, including forms for user registration and login, and a main UI for viewing, adding, editing, and deleting book records. The design was made to be responsive, ensuring it works well on both mobile and desktop devices.
- **State Management:** I used the React Context API to manage the application's state for responsiveness, specifically to identify whether the current display was a desktop or mobile device. This was my first time using the Context API, and it was a valuable learning experience.

Challenges Faced

This assignment presented several challenges, primarily because I had to learn many of the core technologies from scratch within a short timeframe.

- **Lack of Familiarity:** A major hurdle was my lack of experience with C# and .NET. I also had no prior exposure to SQLite when it came to web applications, Entity Framework, or database migrations, all of which were essential for the backend.
- **New Technologies:** On the frontend, I was unfamiliar with TypeScript with React and Vite. I also chose to use Material-UI for styling, which required me to learn a new component library.
- **Specific Implementation Issues:** I faced specific challenges like implementing JWT authentication, and global exception handling. I also had port issues when running the project in Visual Studio Code, which eventually resolved themselves but caused initial delays. The Context API in React was another new concept I had to grasp for state management.

Additional Features

Beyond the core requirements, I added a few features to enhance the application's functionality and robustness.

- **Global Exception Handling:** I implemented a centralized middleware for global exception handling on the backend. This was a crucial addition for better error management and a more robust application.
- **Authentication Security:** I focused on a secure authentication flow by incorporating BCrypt password hashing and JWT tokens, which was a significant learning opportunity.

Key Insights Gained

Completing this project provided me with several valuable insights and hands-on experience that I can apply to future projects.

- **Full-Stack Architecture:** I gained a better understanding of how a backend API, built with a robust framework like ASP.NET Core, integrates with a frontend framework like React.
- **Security Best Practices:** I learned the importance of security in web development, specifically the practical application of JWT for authentication.
- **Database Management:** The experience with Entity Framework taught me about Object-Relational Mapping (ORM) and the importance of database migrations for managing schema changes.
- **Modern Development:** Working with React and TypeScript demonstrated the benefits of type safety and a modern build tool like Vite.
- **API Design:** The process of creating the API reinforced my understanding of RESTful design principles and the need for consistent error handling. Overall, this assignment was an intensive but rewarding experience that taught me the value of a structured development process and the ability to learn and apply new technologies.