Software Engineering Internship Assignment

Prepared By: G.K.M Kaveesh Yoshitha

2024.11.01

Contents

01. Introduction

The objective of this project was to design and implement a comprehensive Library Management System that facilitates efficient management of library resources. This application serves as a robust solution for users to effectively handle book records, encompassing essential functionalities such as the creation, retrieval, updating, and deletion of book entries.

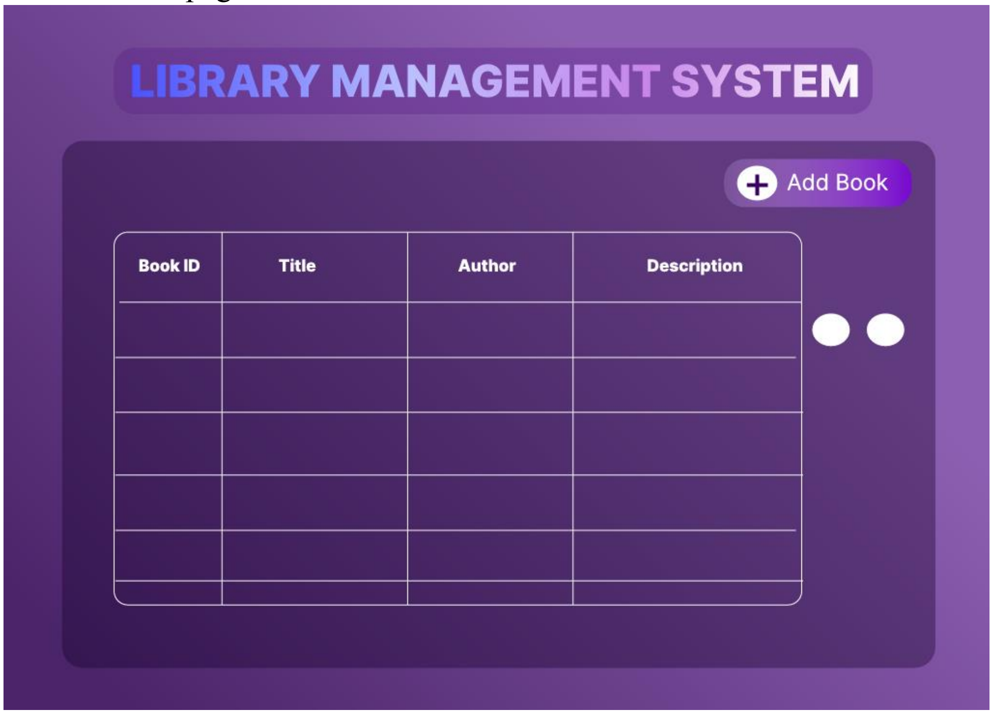
The system architecture comprises a backend API developed using C# .NET, integrated with an SQLite database for data storage, while the frontend interface is crafted with React and TypeScript. This combination of technologies ensures a responsive user experience and a reliable backend framework, allowing for seamless interaction between the user and the library's data.

02. Development Process

Planning

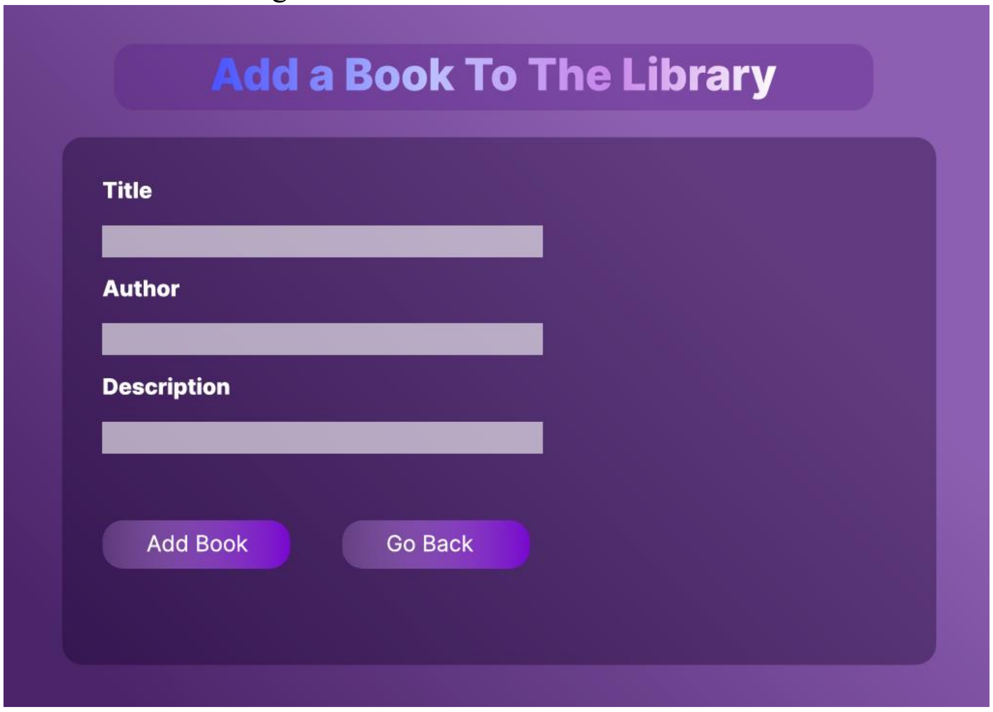
The initial phase of the project involved creating a detailed design of the frontend using Figma. I chose a purple-themed color palette for the entire application due to its modern aesthetic appeal. Attached below are screenshots of all the pages designed in Figma:

- Home page

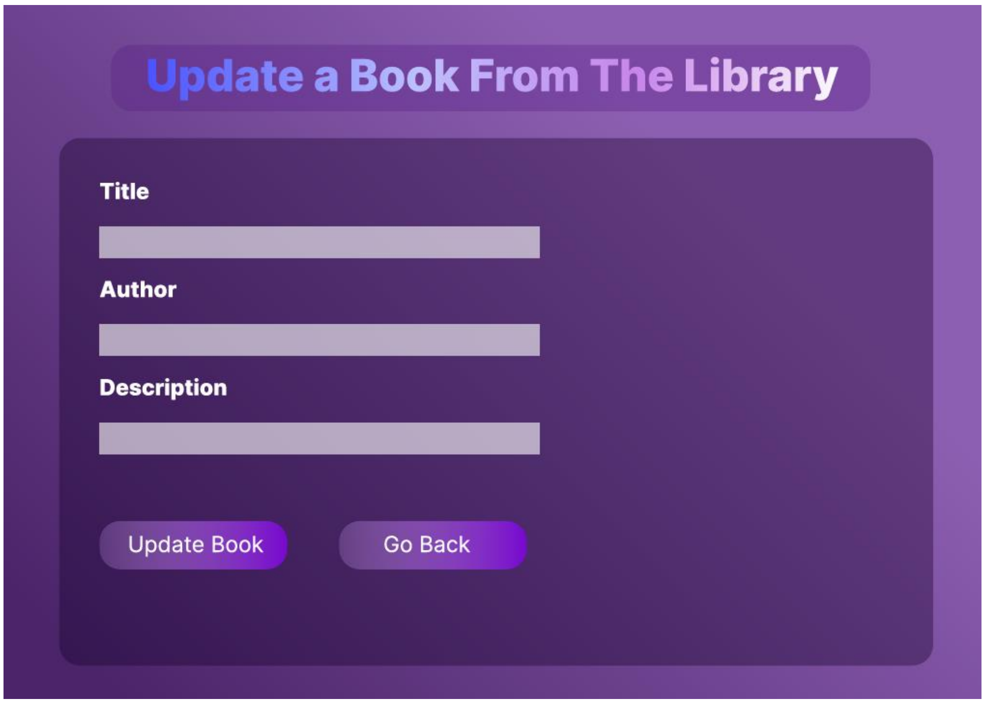


Book ID	Title	Author	Description

- Add Book Page



- Update Book Page



Frontend Implementation

For the frontend, I utilized React.js with TypeScript, leveraging libraries such as Axios for HTTP requests, React Router for navigation, and hooks like useState and useEffect for state management. I also incorporated Font Awesome for icons and React Toastify for notifications. Tailwind CSS was employed for styling the frontend components.

The application consists of three primary pages: the Home Page, Add Book Page, and Update Book Page. The Home Page features a grid displaying all book data, accompanied by buttons for editing, deleting, and adding books. The edit functionality navigates to the Update Book Page, while the add button directs users to the Add Book Page.

Backend Implementation

The backend was developed using ASP.NET Web API, with SQLite serving as the database management system. I created RESTful API endpoints for the following operations: adding a book, viewing all books, updating a book, deleting a book, and retrieving a book by its ID. To integrate with SQLite, I installed the following packages:

- Microsoft.EntityFrameworkCore.Tools

- Microsoft.EntityFrameworkCore.SQLite

I utilized Entity Framework to automate the creation and management of database tables and data.

01. View Books
Endpoint: /api/Library
Method: GET
Description: Retrieves all book details from the database.

02. View Book by ID
Endpoint: /api/Library/${id}
Method: GET
Description: Retrieves details of a specific book based on its ID.

03. Create a Book
Endpoint: /api/Library
Method: POST
Description: Adds a new book to the database.
Request Body Example:

```
{
        "id": 0,
        "title": "Name of the Book",
        "author": "Name of the Author",
        "description": "Description about the
        book"
}
```

04. Update a Book
Endpoint: /api/Library/${id}
Method: PUT
Description: Updates a specific book based on its ID.
Request Body Example:

```
{
  "id": 0,
  "title": "Updated Name of the Book",
  "author": "Updated Name of the Author",
  "description": "Updated Description about the book"
}
```

05. Delete a Book
Endpoint: /api/Library/${id}
Method: DELETE
Description: Deletes a specific book based on its ID

03. Challenges Faced

I. **Time Constraints**
I received the project on October 28, 2024, with a deadline of November 3, 2024. Due to a scheduled surgery on November 2, 2024, I initiated the project on October 29, 2024, and aimed to complete it by November 1, 2024. Despite the time limitations, I successfully implemented all mandatory requirements, including CRUD operations. However, I was unable to develop the optional feature of user authentication.

II. **Low Knowledge about SQLite and typescript**
Having not previously worked with a React and TypeScript stack or utilized SQLite for database management, I faced a steep learning curve within a constrained timeframe. Nevertheless, I was able to acquire the necessary knowledge and successfully apply it throughout the project.

04. Key Insights and Lessons Learned

I. **SQLite Proficiency**
This project provided valuable experience in working with SQLite, enhancing my ability to manage data effectively.

II. **TypeScript Familiarity**
I gained practical experience in developing applications using React and TypeScript, which has strengthened my skill set.

III. **Understanding of RESTful APIs**
The project allowed me to revisit and solidify my understanding of RESTful APIs, including handling user requests and integrating the frontend with the backend. Additionally, I refined my skills in using Axios to facilitate communication between the React frontend and the backend.

05. Conclusion

The requirements for this project involved creating a Library Management System that enables users to add, view, update, and delete books while providing an engaging user interface. I am pleased to report that these requirements were successfully met, resulting in a functional and aesthetically appealing application.