**Front End Engineering - II**

Project Report

Semester-III (Batch-2023)

Librarium

****

**Supervised By: Submitted By:**

Dr. Sanjoy Kumar Debnath

Nishan Singh 2310990671

**Department of Computer Science and Engineering**

**Chitkara University Institute of Engineering & Technology,**

**Chitkara University, Punjab**

**Table Of Content**

|  |  |  |
| --- | --- | --- |
| Sr.no | Section | Page no |
| 1. | Abstract | 1 |
| 2. | Introduction | 2-4 |
| 3. | Problem Definition and requirement | 5 |
| 4. | Proposed Methodology | 6-8 |
| 5. | Results | 9-15 |
| 6. | References | 16 |

**Abstract:**

**LIBRARIUM**

The university library's manual system is inefficiencies, have errors, and limited access to resources. To address these challenges, we propose a Library Management System that automate and streamline library operations, improve user experience, and provide better management of library resources.

Librarium will utilize a modular approach, with separate modules for user management, book management, circulation management, and reporting.

Librarium will address real-life problems, including inefficient book searching and borrowing, inaccurate tracking of book inventory, and limited access to library resources. By automating library operations, we will reduce the workload of librarians, allowing them to focus on more critical tasks. It will also improve the user experience, providing a user-friendly interface for searching and borrowing books, and accessing library resources and services.

Innovative features include data analytics and visualization tools, enabling librarians to gain valuable insights into library usage and resource utilization. The system will also provide personalized recommendations for students and faculty, suggesting relevant books and resources based on their research interests and borrowing history.

1. **Introduction**
   1. **Background**

The Librarium project aims to address the challenges faced by traditional library systems, which often relied on manual processes that led to inefficiencies and human errors. These traditional systems made it difficult to maintain accurate records and limited access to information.

In contrast, digital library systems have become increasingly popular, providing a more efficient way to manage resources like books, journals, and digital media. By automating routine tasks and enhancing accuracy, these systems improve the overall user experience and make information more accessible.

The goal of Librarium is to create a responsive and interactive front-end for a modern library management system. This project focuses on enhancing the user experience by offering an intuitive interface for managing and accessing library resources. Users will be able to search for books, view availability, check details, and interact with library services easily, improving the overall process of managing library resources in a digital environment.

**1.2 Objectives**

* **To design an intuitive and user-friendly interface**: The primary goal is to create an easy-to-navigate interface that serves the needs of both library staff and students. By providing a streamlined user experience, we aim to minimize complexity and ensure users can quickly access the features they need, whether it's searching for books or managing accounts.
* **To maintain automatic tracking of rentals and stock in a database**: The system will automate the process of tracking book rentals, returns, and stock levels, eliminating the need for manual updates and reducing human error. A central database will store all information regarding book availability, user rentals, and inventory, ensuring that data is accurate, up-to-date, and easily accessible.
* **To ensure system responsiveness across devices**: The *Librarium* system will be designed to be fully responsive, providing a seamless experience across various devices, including desktops, tablets, and smartphones. This will allow users to access the system conveniently from anywhere, whether at home or on the go, ensuring that functionality remains intact regardless of screen size or device type.
* **To enhance the overall efficiency of library operations**: By automating routine tasks such as book checkout, return, and inventory management, the system will save time for library staff, allowing them to focus on more value-added tasks. Additionally, the system will support better decision-making by providing data-driven insights into book usage and stock levels.
* **To integrate a reliable search functionality**: The system will offer an advanced search functionality that allows users to easily find books, journals, and other resources by title, author, or category. Filters and suggestions will help users refine their searches, improving the speed and accuracy of finding relevant materials.’
* **To provide secure user authentication and account management**: The system will include secure authentication methods for both library staff and students, ensuring that only authorized individuals can access sensitive data or perform specific actions, such as adding new books or updating inventory. Personal accounts will also allow users to track their borrowed books and due dates.

**1.3 Significance**

The system is intended to serve both library staff and students, making it easy for each group to access and manage the system. For staff, *Librarium* provides simple tools to update inventory, add or remove books, and track stock levels, all from a central interface. For students, the system offers an intuitive search feature that helps users quickly locate books based on title, author, or category.

A major advantage of *Librarium* is its emphasis on responsiveness across different devices. Whether a user accesses the system via a desktop, tablet, or smartphone, the platform will remain fully functional and responsive

The integration of the Google Books API into *Librarium* adds another layer of sophistication by enabling seamless access to a vast database of books, journals, and other materials. This API allows the system to pull up-to-date information about books, such as availability, details, and even stock levels from external sources. By leveraging this API, *Librarium* not only improves the accuracy of the data but also provides users with access to a much broader selection of resources.

Ultimately, the *Librarium* project represents a significant advancement in how libraries can manage their resources and engage with users. By reducing manual tasks, improving data accuracy, and enhancing the user experience, *Librarium* helps libraries run more efficiently while offering students and staff a modern, accessible platform for managing and accessing library materials. The project stands as a prime example of how technology can be leveraged to improve traditional systems and provide a more efficient and enjoyable experience for all users involved.

1. **Problem Definition and Requirements**

Libraries often face challenges related to resource management, including tracking the status of books, handling book returns, and maintaining records accurately. The project addresses these challenges by creating a front end that allows for easy data entry, searching, and real-time updates.

**Problem Definition**:

* The current manual system for managing library operations is plagued by issues such as not knowing whether the desired book is in stock or not.
* Inefficient book searching and borrowing processes
* Inaccurate tracking of book inventory and availability
* No alert system for users when due date for the books approaches near;

**Software Requirements**:

* Google Books API: Used for retrieving book information, enabling search functionality, and managing book data in the system.
* MongoDB: To manage and store data easily we have implemented MongoDB.
* **Express.js**: Used to create the backend API enabling efficient communication with MongoDB to fetch, update, and manage book data.

1. **Proposed Methodology**

* **System Design**:

Librarium will be designed using a modular approach, with separate modules for user management, book management, circulation management, and reporting.

It will provide user to check whether the book desired is in stock or not and will also provide an alert system.

* **Development**:

It was developed using a combination of open-source and proprietary technologies, including HTML, CSS, JavaScript, and a database management system . Later on converted to React using

Tailwind.

* **Testing and Quality Assurance**:

Librarium will undergo thorough testing and quality assurance to ensure that it meets the requirements and is free of bugs and errors.

* **Implementation and Deployment**:

Librarium will be implemented and deployed in phases, with training and support provided to library staff and users.

**Features and Functionality:**

* **Login Page:** A responsive login page has been developed to provide secure and easy access for both library staff and students. The page includes input fields for email or phone number and password, allowing users to log in to the system based on their registered credentials. The login system ensures secure authentication by validating the input data before granting access to the library resources. Additionally, the page is designed to be fully responsive, meaning it adapts seamlessly to different devices, ensuring that users can easily log in regardless of whether they are using a desktop, tablet, or smartphone.**Main Interface:** Created the main interface with a navigation bar, background image, and content area. Structured layout to provide a clear and user-friendly interface.
* **MongoDB:**  *Librarium* uses **MongoDB** as the database solution to handle all data storage needs. MongoDB is a NoSQL database, which allows for the flexible and scalable storage of library data, such as book details, user information, and transaction logs. The system fetches data from MongoDB in real-time, ensuring that the information displayed to users is always current and accurate. Whether a user is searching for a book or managing rental transactions, MongoDB ensures that data is quickly retrieved and presented efficiently. The use of MongoDB also enables the system to scale easily as the library's database grows, ensuring the system remains performant even as more books and users are added.
* **NodeJS with ExpressJS:** To handle backend tasks and provide smooth communication between the front-end interface and the database, *Librarium* utilizes **NodeJS** in conjunction with the **ExpressJS** framework. NodeJS provides a powerful, non-blocking I/O environment, which allows the system to handle multiple requests simultaneously without slowing down, ensuring a fast and responsive user experience. Express.js is used to structure the server and manage routing efficiently, enabling the system to handle requests such as adding or removing books, updating inventory, and fetching book details. Express.js simplifies the creation of API endpoints for CRUD (Create, Read, Update, Delete) operations, ensuring seamless interactions with the MongoDB database. The combination of NodeJS and Express ensures that the backend is scalable, fast, and able to handle the demands of a growing user base.

**JavaScript**

**Objective:** Add interactivity and dynamic behaviour to the website. **Features and Functionality:**

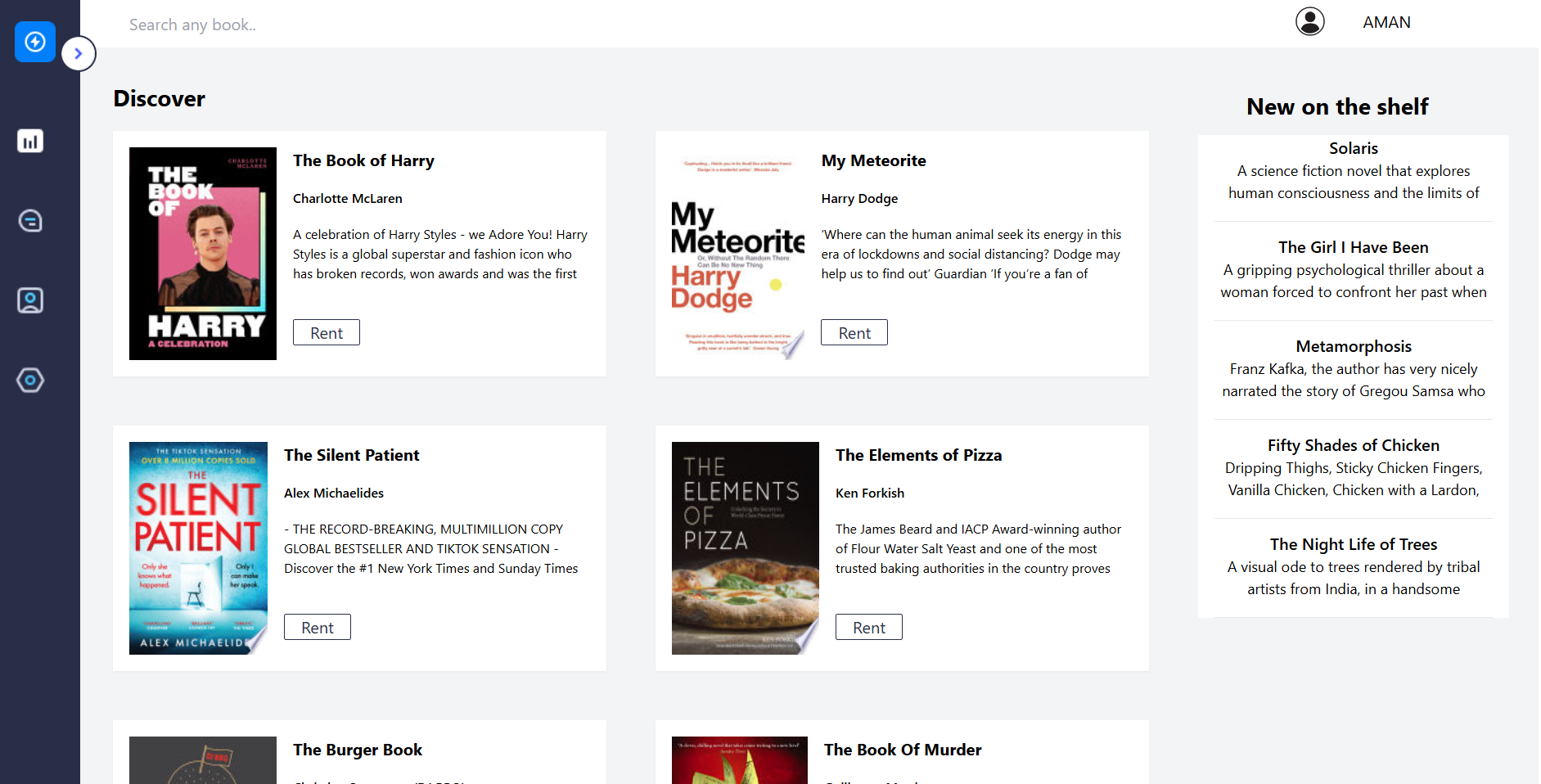
1. **Interactive Elements:**
   * Implement dynamic form validation for login and registration.
   * Enhance navigation bar with dropdown menus and mobile-friendly behaviour.
2. **Search Functionality:**
   * Enable search suggestions and filtering in the knowledge base.
3. **Admin Page**:
   * The Admin Page allows admin to add , remove and delete books from database.
   * It allows to increase stock of new book
   * Used Google Book API to fetch books images and other data;

**React**

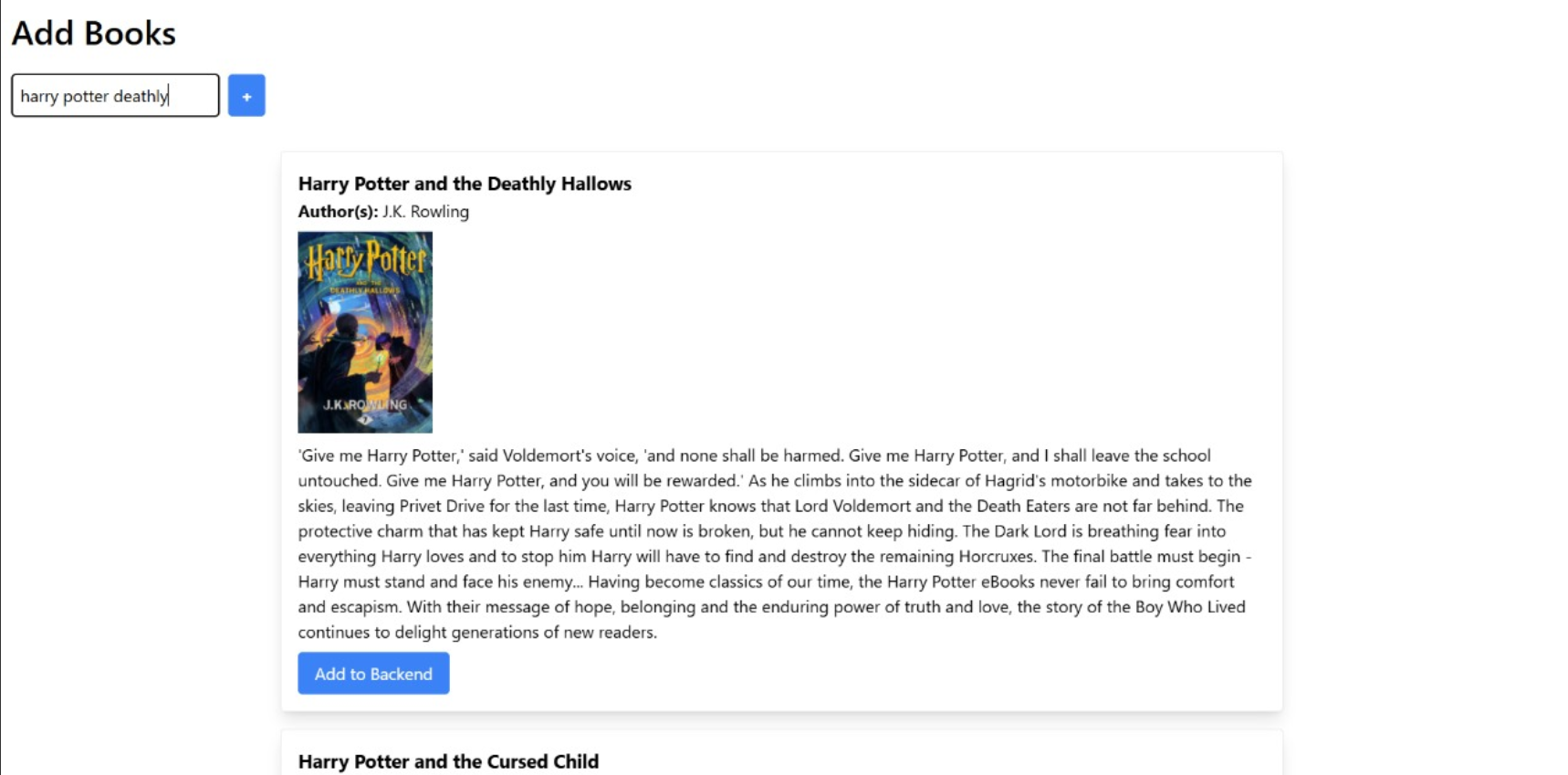
**Objective:** Develop a dynamic and scalable web application with a modern tech stack. **Features and Functionality:**

1. **Component Architecture:**
   * Convert the main interface into reusable React components (e.g., Navbar, AboutPage, Userinfo, Register etc ).
2. **Routing:**
   * Set up React Router for seamless navigation between different sections (e.g., login, knowledge base).
   * Used useNavigation method for routing from one page to another seamlessly.
3. **Results:**

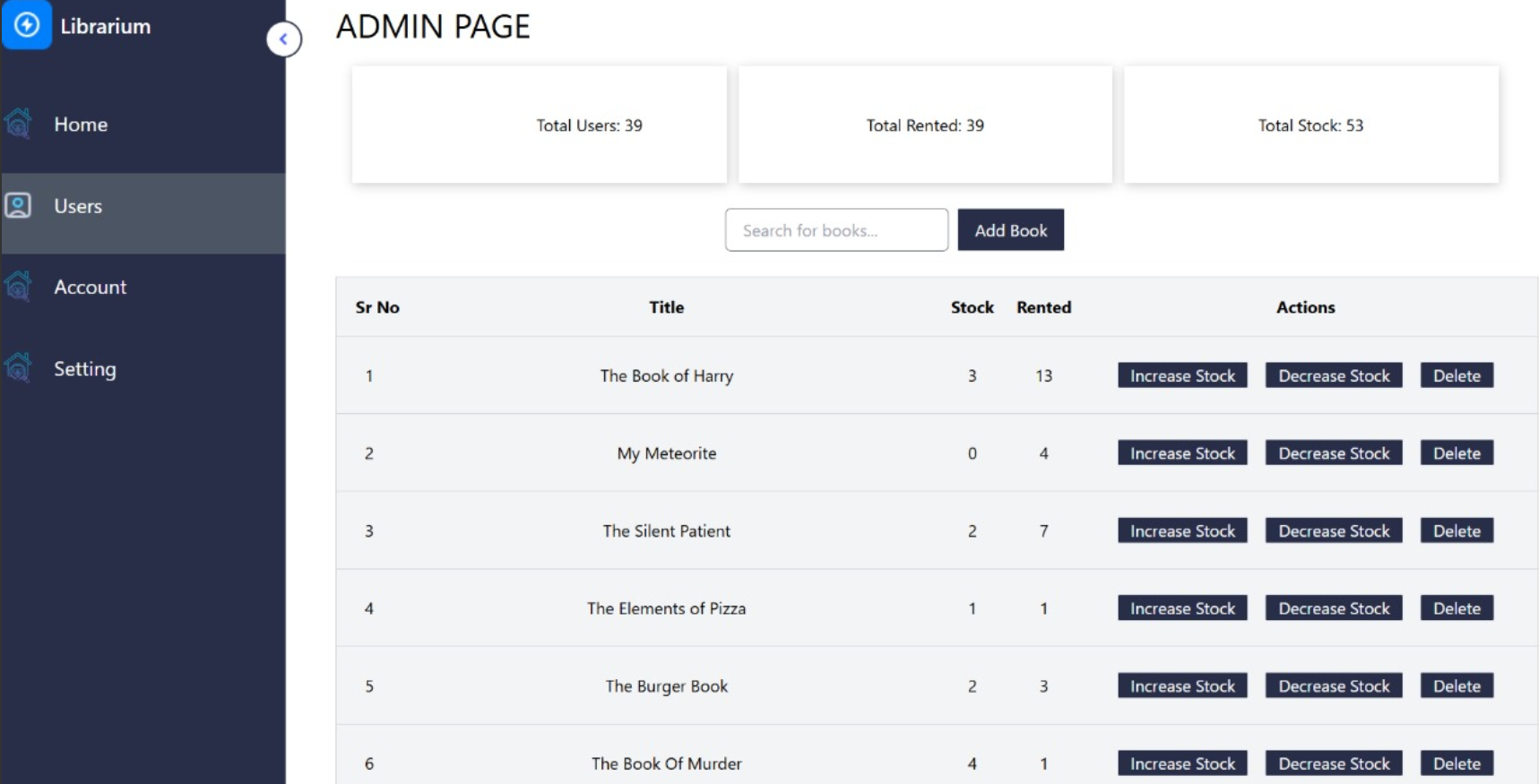
**Main Page :**

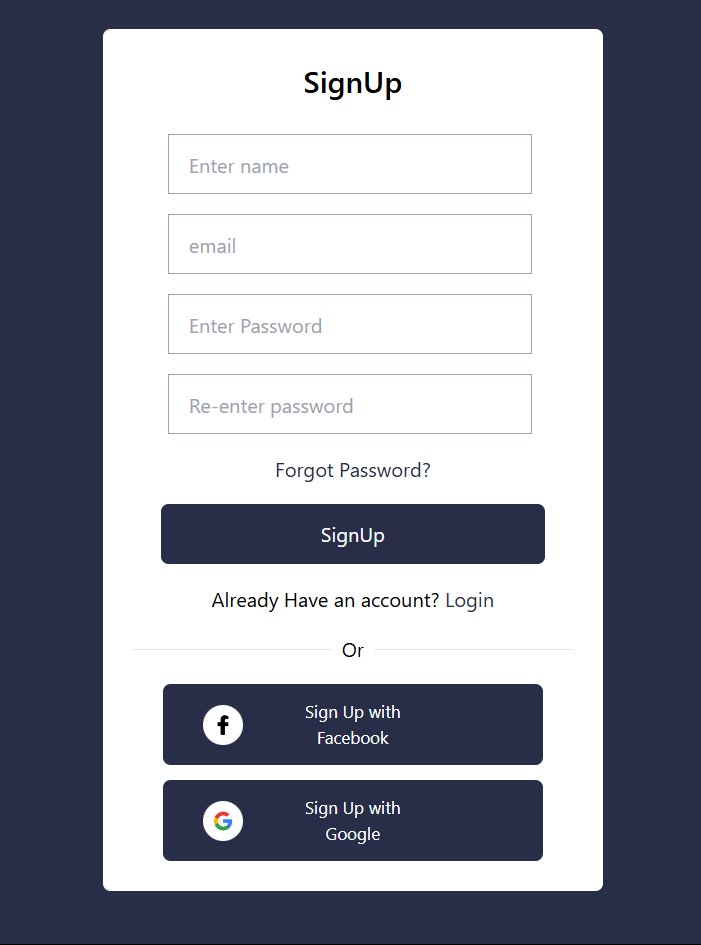
****

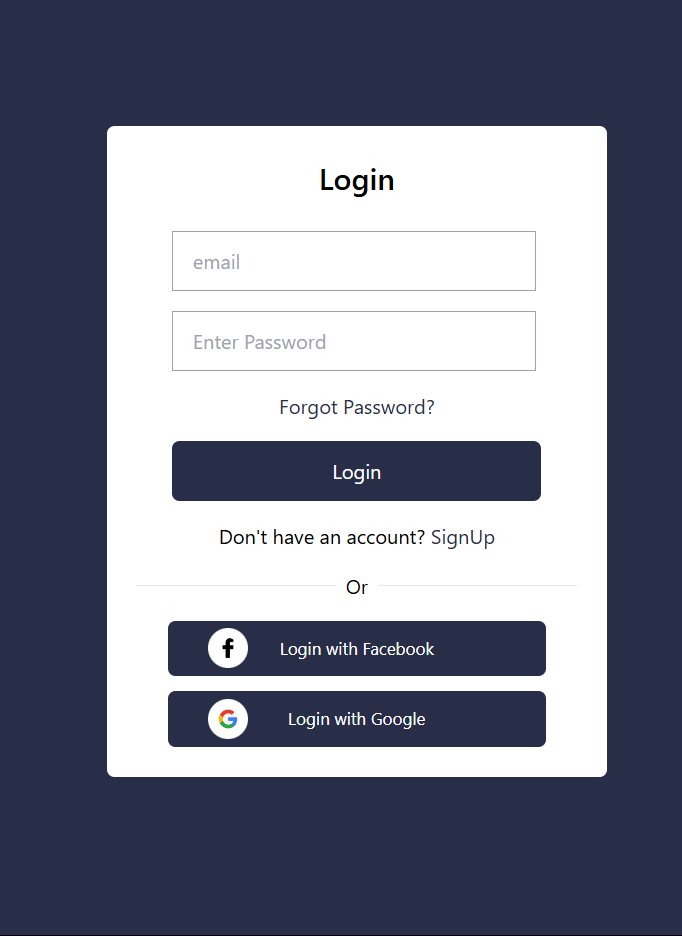
**Admin Page for Adding Books in Database**

****

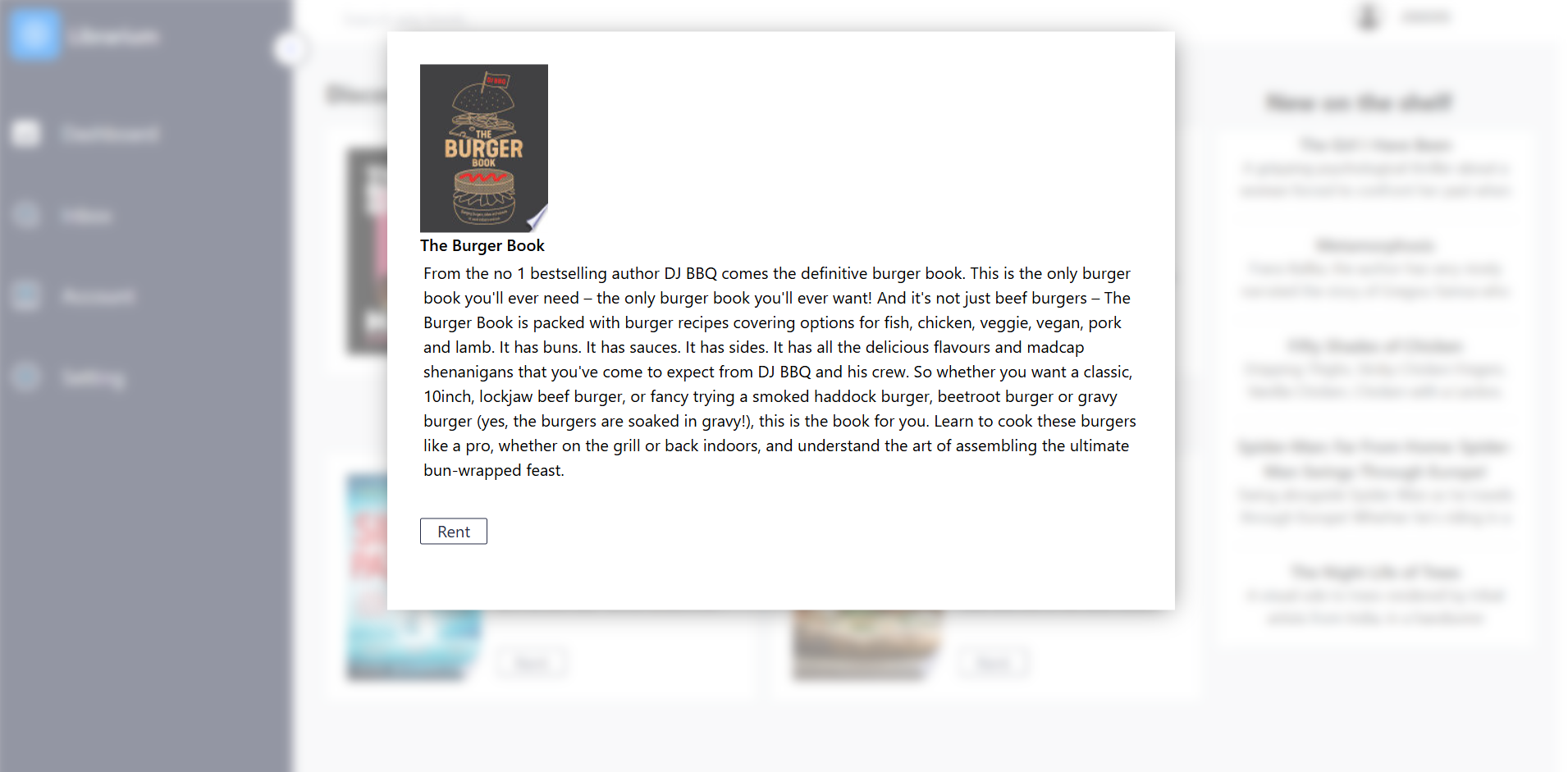
**Overview of Admin Page**

****

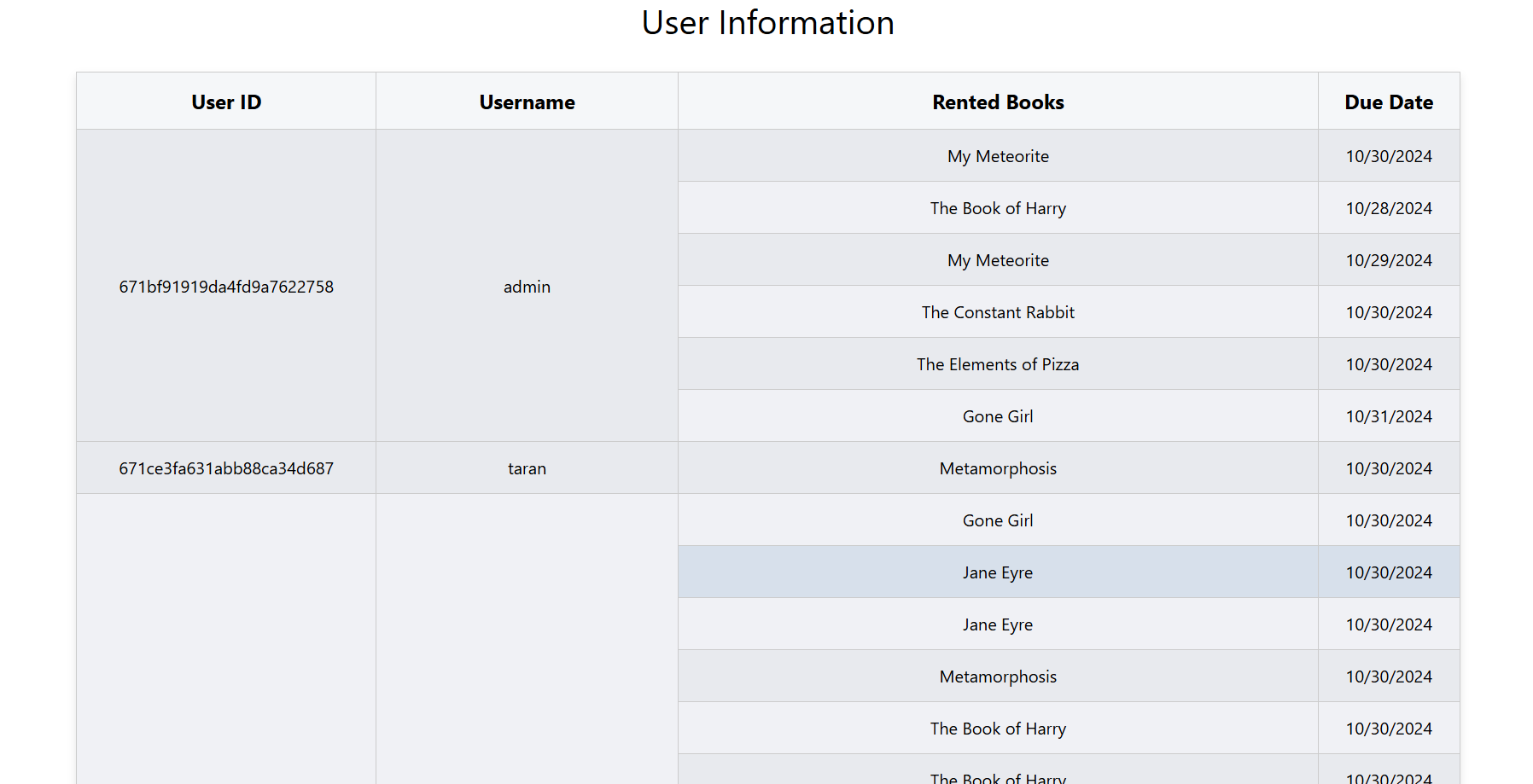
****

****

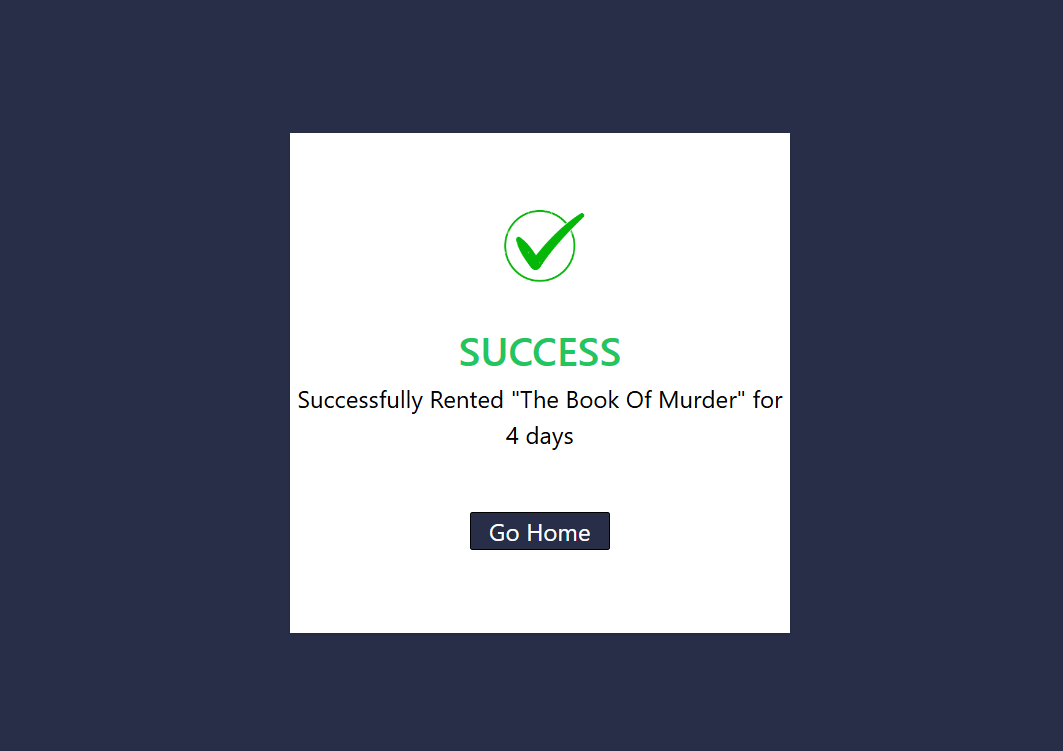
**About Book Page:**

****

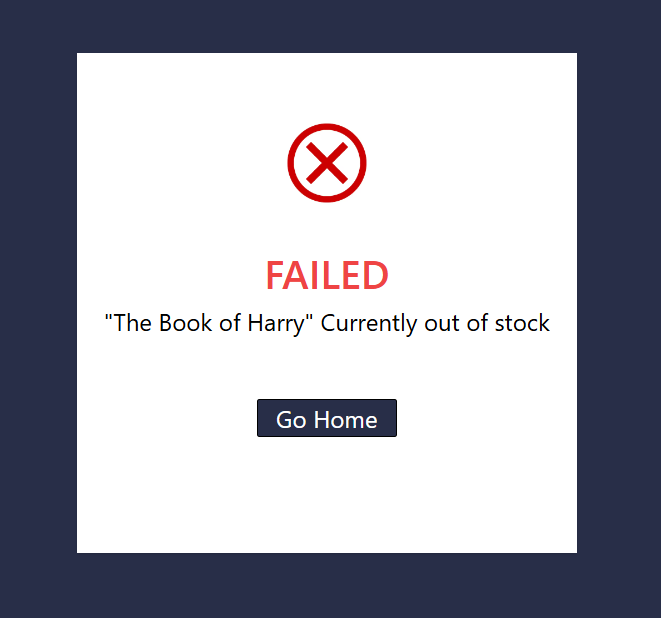
**Users info Page:**

****

**Successful rental page:**

****

**Out of Stock Page:**

****

**References:**

**Google Books API:**

Google. (n.d.). Google Books API. Retrieved from https://developers.google.com/books