

Library Management System

Requirements and Specification Document

10/10/2023, Version - 1

Project Abstract:

This project aims to develop an advanced Library Management System (LMS) designed to streamline the efficient management of books, patrons, and loans within libraries. It is ensured that the system will remain adaptable to changing library needs by utilizing Agile methodologies. The LMS will facilitate book cataloging, patron record management, loan management with automated reminders, a user-friendly search and discovery interface, and reporting and analytics capabilities for data-driven decision-making. Ultimately, this project aims to modernize library operations by automating and enhancing various aspects of library management while ensuring flexibility and responsiveness to evolving library requirements.

Website Link:

<https://prathamgarg03.github.io/Library-Management-System-CMPT-276-/adminLogin/adminLogin.html>

GitHub Link:

<https://github.com/prathamgarg03/Library-Management-System-CMPT-276-#library-management-system-cmpt-276->

Customer:

The primary customers of this system are librarians and library staff, who rely on the system for tasks such as book cataloging and patron management. Library patrons interact with the software to search for books, manage loans, and receive reminders for overdue items. Library administrators depend on the software for data-driven decision-making through reporting and analytics. Additionally, library vendors and book suppliers may indirectly interact with the system. Understanding the diverse needs of these customer types is critical for developing a comprehensive and user-friendly library management solution.

Competitive Analysis:

The main competitors in the Library Management System market are Koha, Evergreen, and SirsiDynix, our project differentiates itself through its Agile adaptability, automatic book retrieval from online databases, a comprehensive patron experience, and advanced search capabilities. This unique blend of features empowers libraries to efficiently adapt to evolving needs, offers a more user-friendly and informative experience, enhances resource discovery, and ultimately sets our project apart as a modern solution.

User Stories:

Story 1 (Admin logging into the system)

Actors Involved: Administrator - Person operating in administrative capacity (Ex. Librarians)

Precondition: Administrator is registered in the system.

Postcondition: User jumps to Admin HomePage

Iteration 1

User gets to the website using the link, and enters their login credentials, clicks the login button and jumps to the admin homepage, which displays several options such as Books Issued, Add/Remove/Modify, Manage Students, Defaulter List. (Only the Add feature has been implemented in iteration 1)

Story 2 (Admin adding a book onto the database)

Actors Involved: Administrator - Person operating in administrative capacity (Ex. Librarians)

Precondition: User is on the Admin HomePage

Postcondition: Book is added onto the table on the Webpage

Acceptance Test: A message saying 'Book Added Successfully' is displayed on the screen

Iteration 1

User clicks on the Add/Remove/Modify button on the Admin Homepage, this navigates them to another page, where they should fill out the 'Library Form' by entering the Name, Author, Publisher and Quantity of the book, in order to add the book on the existing database. After following these steps, a message saying 'Book Added Successfully' will be displayed onto the screen and the book will be added on the table displayed on the same Webpage.

Story 3 (Student logging into the system)

Actors Involved: User - Person operating as a student.

Precondition: User is registered in the system.

Postcondition: User jumps to Student HomePage

Iteration 1

User gets to the website using the link, and clicks on the 'Not an admin' link to get to the student homepage and enters their login credentials, clicks the login button and jumps to the student homepage, which displays several options such as Search, Loans, Requests and Fines. (Only the Search feature has been implemented in iteration 1)

Story 4 (User searching a book in the database)

Actors Involved: User - Person operating as a student.

Precondition: User is registered in the system.

Postcondition: The book information pops up on the screen if it exists on the database.

Iteration 1

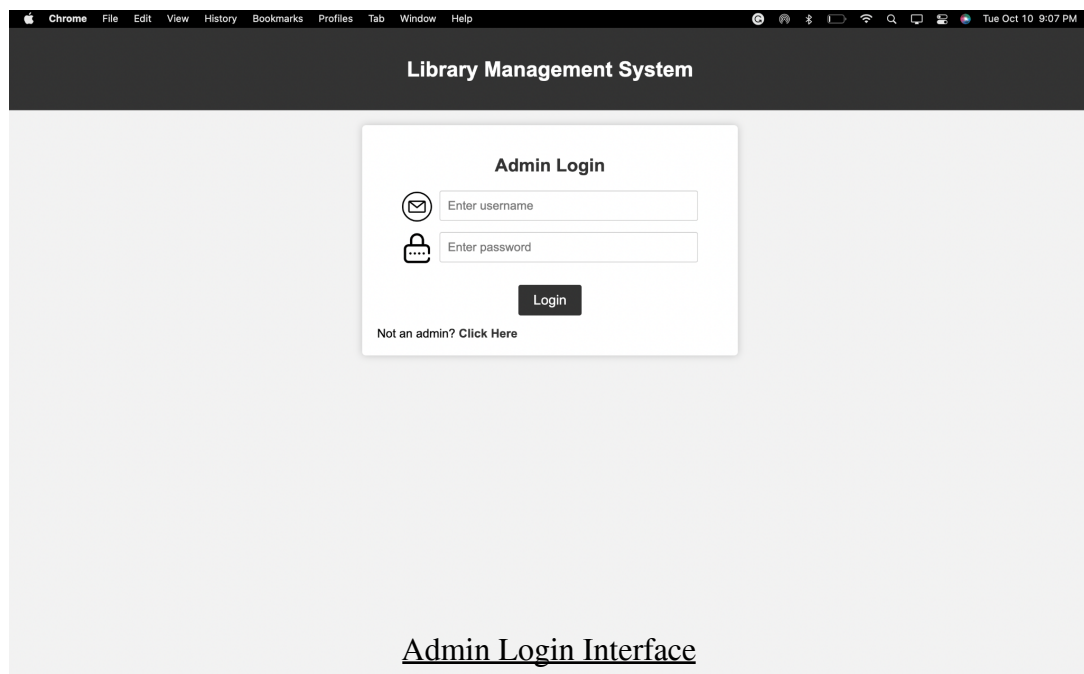
User clicks on the Search button on the Student Homepage, this navigates them to another page, where they should enter the name of the book they want to search. After following these steps, if the book exists in the database, the Name, Publisher, Author and Quantity of the book will be displayed on the screen in a table form, in case the book does not exist on the database, an error message will be displayed.

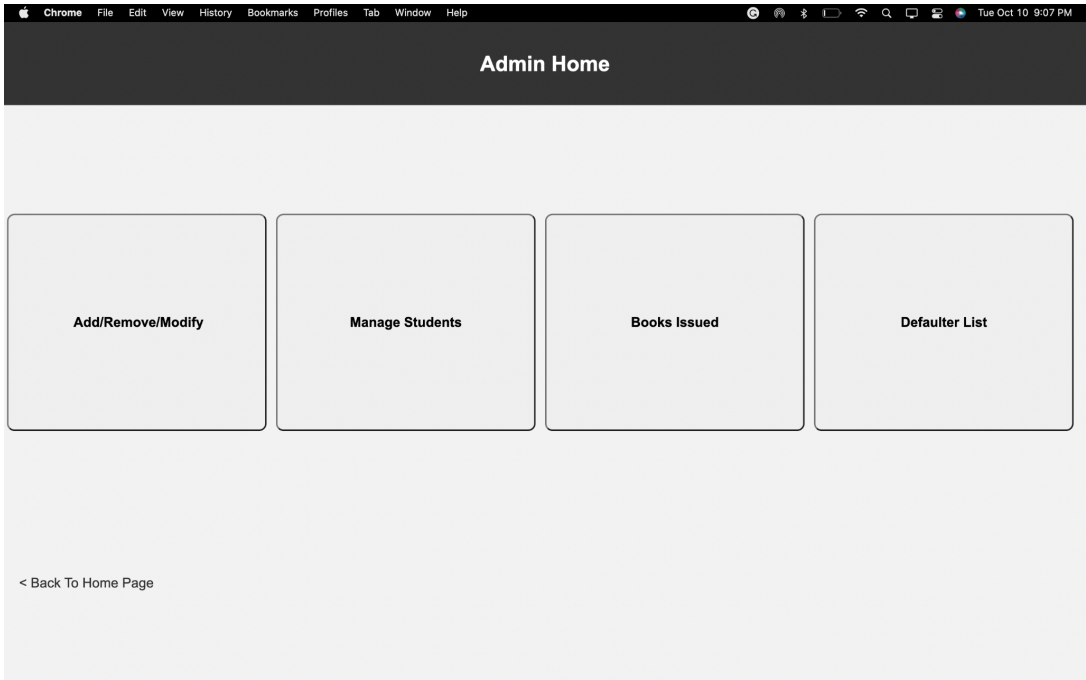
Features to be implemented in future iterations:

- Feature to remove and update the existing entries in the database
- Managing the loans and due dates of students
- Managing the list of books issued by the library.
- Maintaining a separate list of frequent offenders.

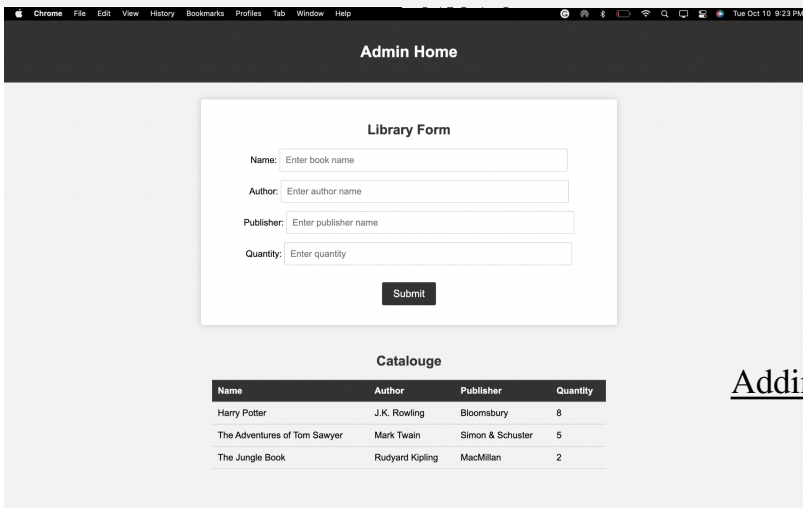
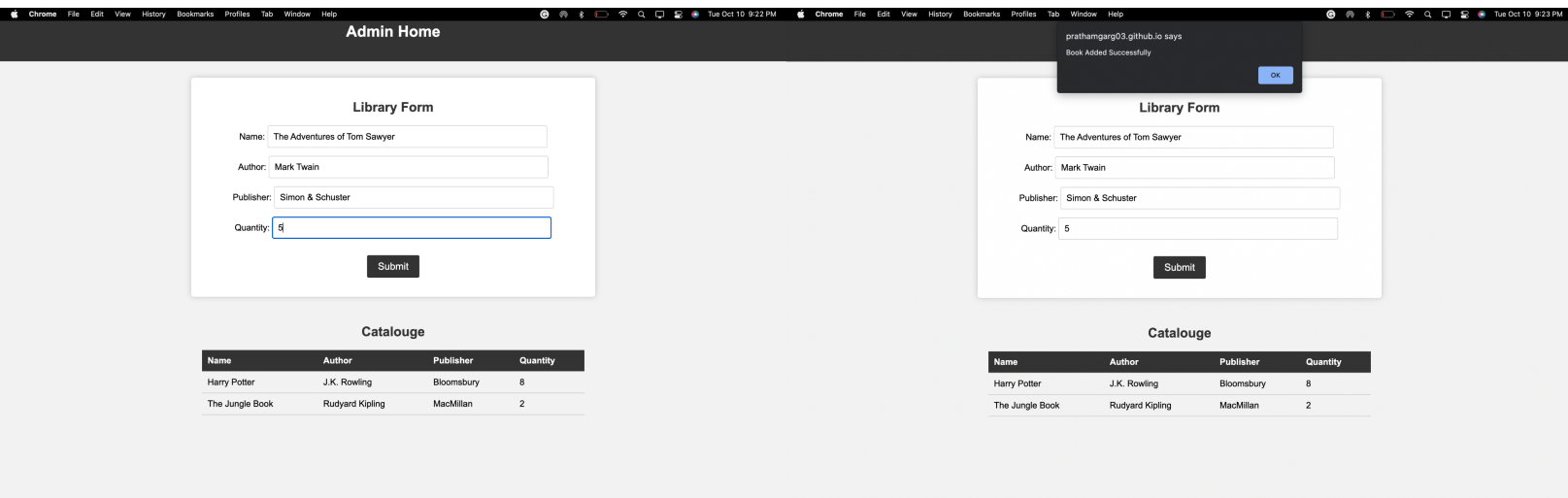
Note: Irrespective of the commits, all the group members contributed equally to the project by working on different pre-determined parts of the project.

User Interface Requirements:

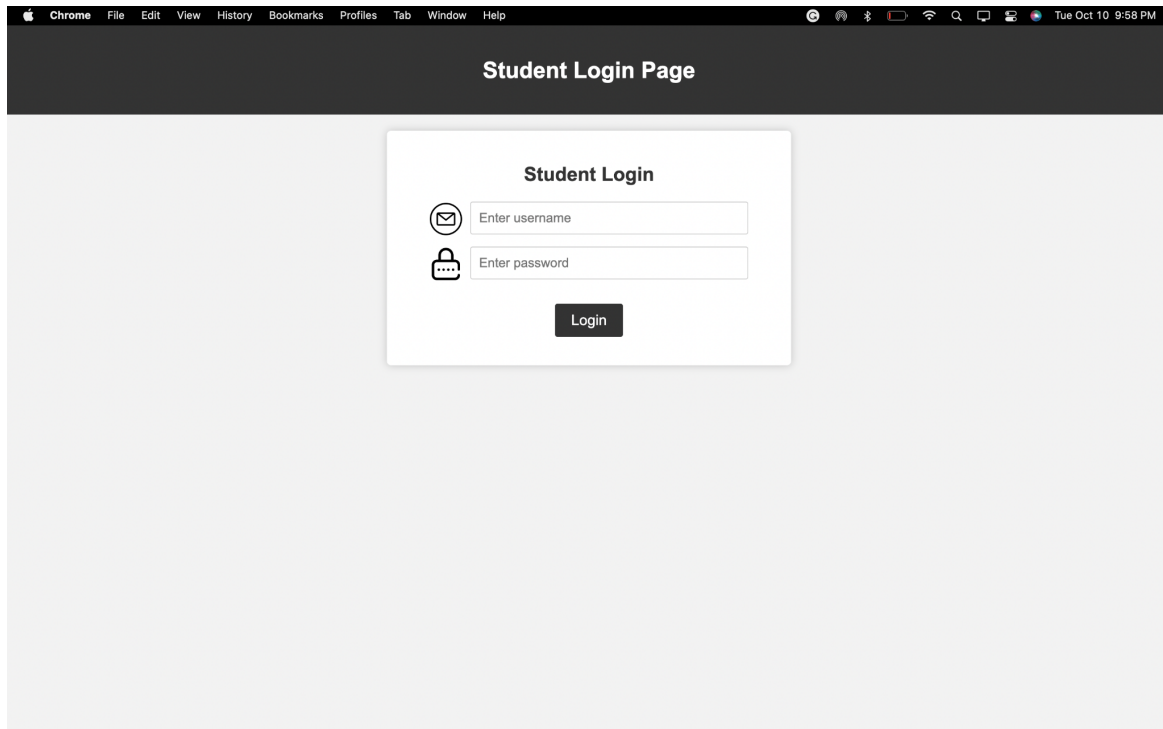




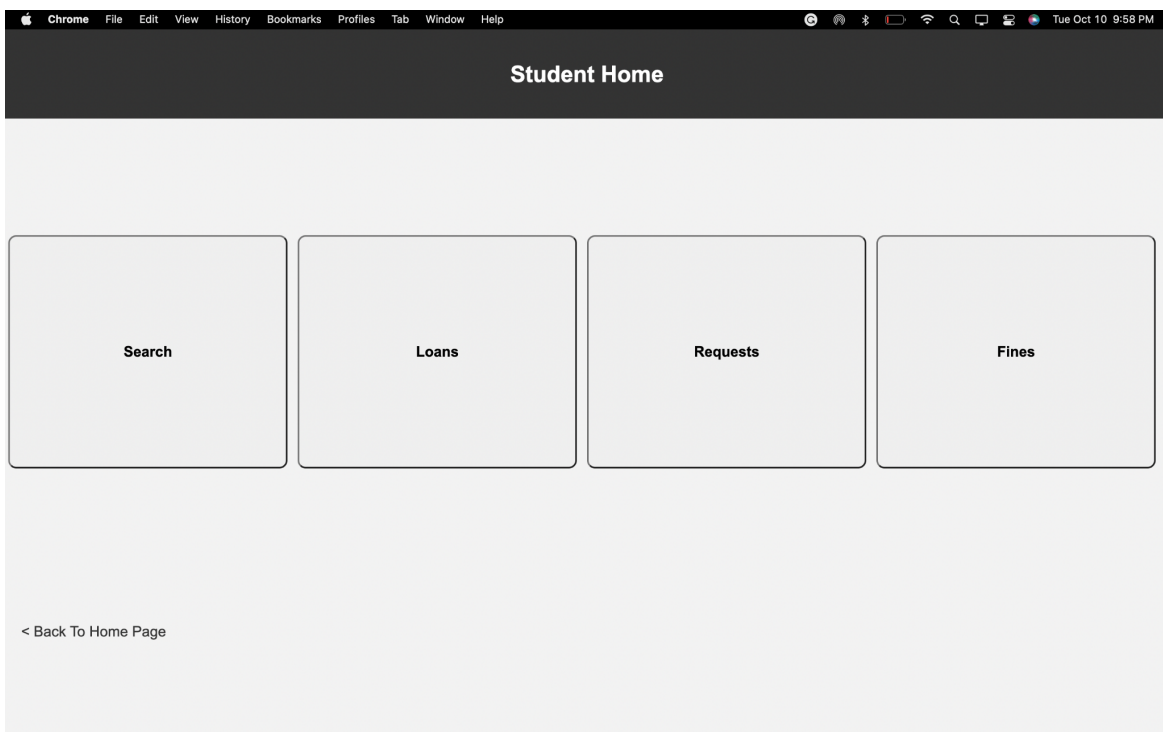
Admin Homepage



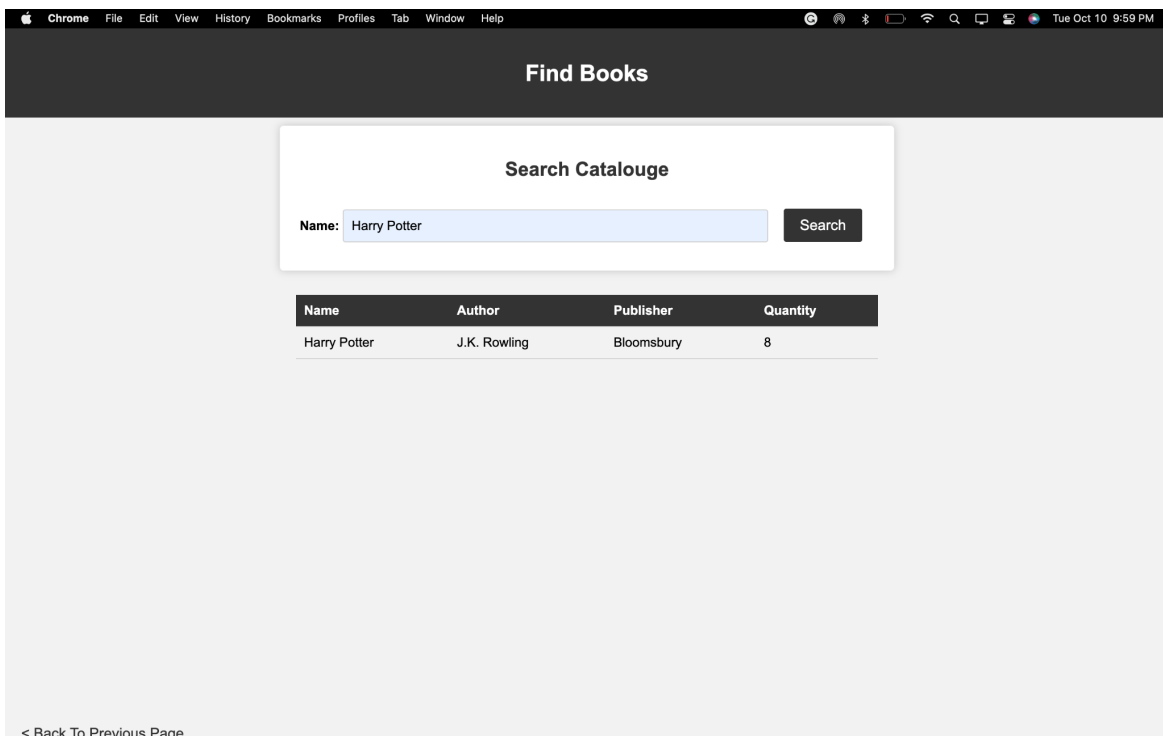
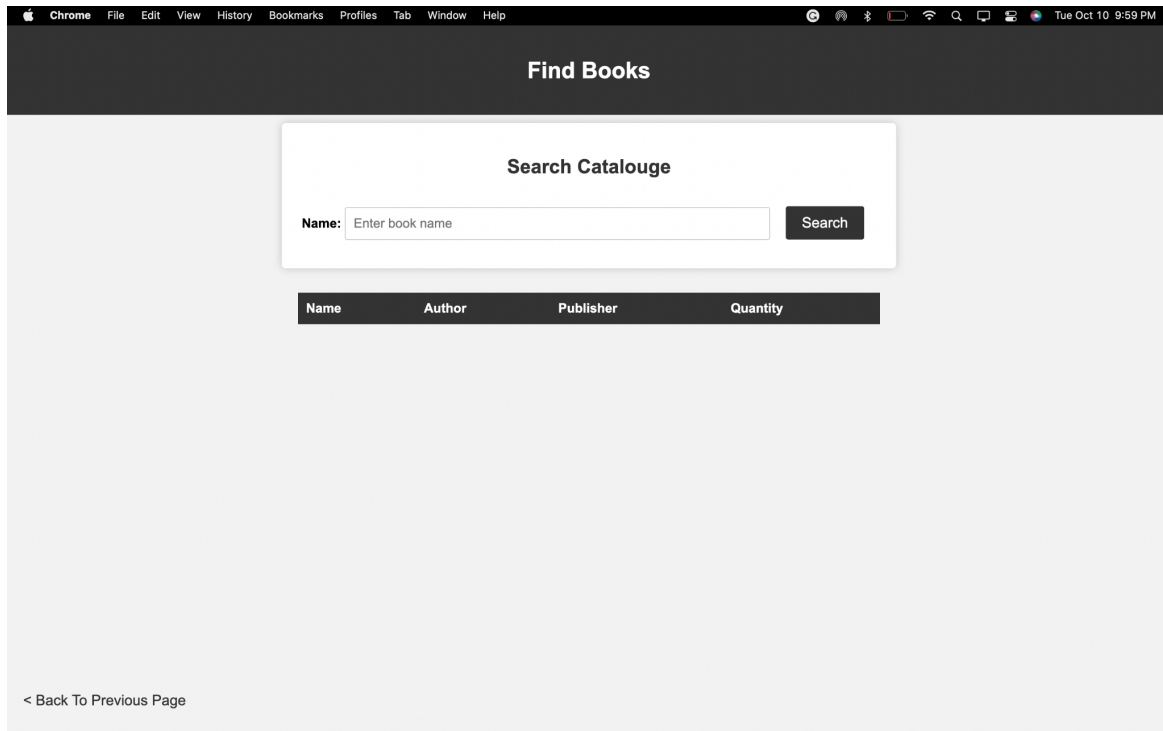
Adding a book onto the database



Student Login Page



Student Home Page



Searching a book on the Database