Adventist University of Central Africa

P.O. Box 2461 Kigali, Rwanda | www.auca.ac.rw | info@auca.ac.rw

Names: Uwabera Ange Scovia

ID:24082

Web Technology Final Project

Topic: Library Management System

1. Purpose:

The Library Management System (LMS) is designed to streamline and automate the operations of a library. The primary purpose is to provide an efficient way for the library staff to manage student records, book details, and transactions, including borrowing and returning books. The system aims to enhance the overall management and organization of the library's resources**.**

2.Project Scope

The Library Management System encompasses the following key features:

2.1 User Roles

Admin: Responsible for managing student information, book details, borrowing and returning books, and overall system administration.

2.2 Functionality

2.2.1 CRUD Operations

Student Registration: Capture and store student information.

Book Registration: Add new books to the system with details such as title, author, and type.

Book Type Registration: Define and categorize books into types.

Borrowing and Returning Books: Keep track of books borrowed by students and their return dates.

2.2.2 Search and Filters

Search Books: Allow users to search for books based on their names.
Filter Books: Provide filters to categorize books by type.

2.2.3 Validation

Input Validation: Ensure that user inputs, such as registration details and book information, are valid and follow specified criteria.

2.2.4 Authentication

Login and Logout: Implement secure authentication mechanisms to control access to the system. Only authorized users (admins and students) should be able to log in.

3. Requirements:

3.1.1 Admin Functionality

Admin Login: Admins should be able to log in securely with a unique username and password.

Student Registration: Admins can add, update, and delete student information, including name, ID, and email.

Book Registration: Admins can add, update, and delete books with details such as title, author, and type.

Book Type Registration: Admins manage book types .

Borrowing and Returning Books: Admins can record and manage book transactions, including borrowing and returning books.

3.1.2 Student Functionality

Student Login: Students should be able to log in securely with their unique credentials.
Browse Books: Students can search available books based on title, author, or type.
Borrow and Return Books: Students can borrow books and return them within the specified due date.

3.2 Non-Functional Requirements

3.2.1 Performance

Response Time: The system should provide quick responses to user queries and actions.
Scalability: The system should be scalable to accommodate an increasing number of users and books.

3.2.2 Security

Data Encryption: User data, especially login credentials, should be encrypted to ensure secure transmission.
Access Control: The system should enforce strict access controls to prevent unauthorized access to sensitive information.

3.2.3 Usability

User Interface: The user interface should be intuitive and user-friendly for both admins and students.

4. Expected Outcomes:

The Library Management System is expected to achieve the following outcomes:

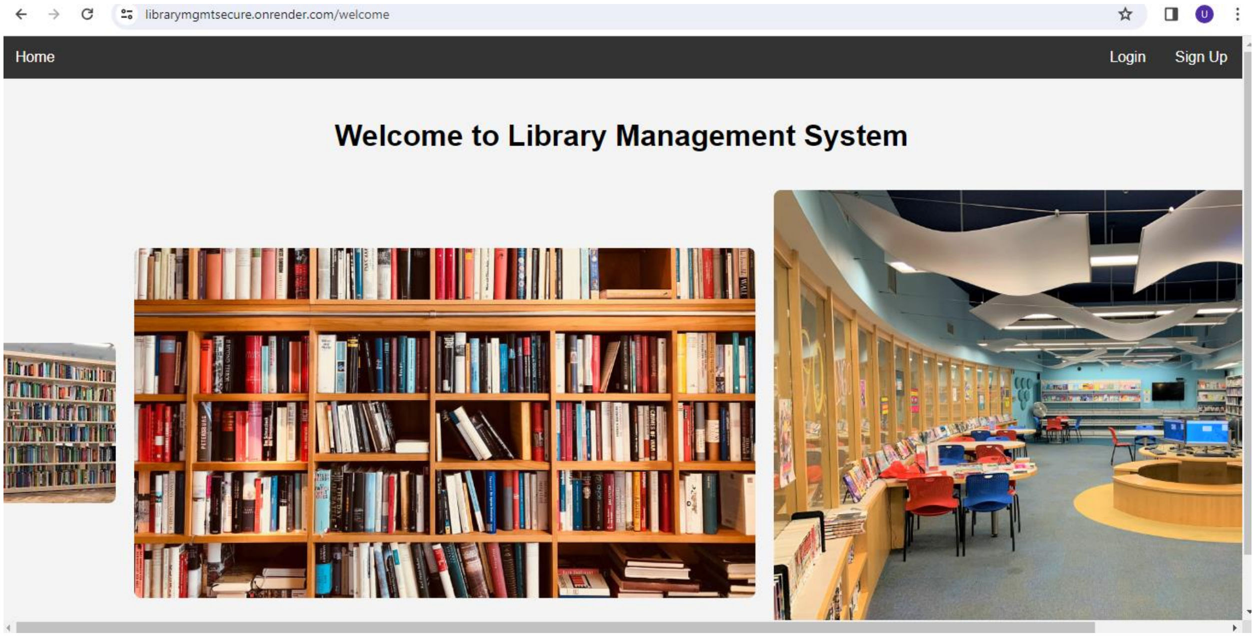
Efficient Management: Streamline the process of managing student records and book transactions.

Accuracy and Reliability: Ensure accurate and reliable data storage and retrieval.
User-Friendly Interface: Provide an intuitive and user-friendly interface for both admins and students.

Time Savings: Reduce manual efforts in bookkeeping and administration.

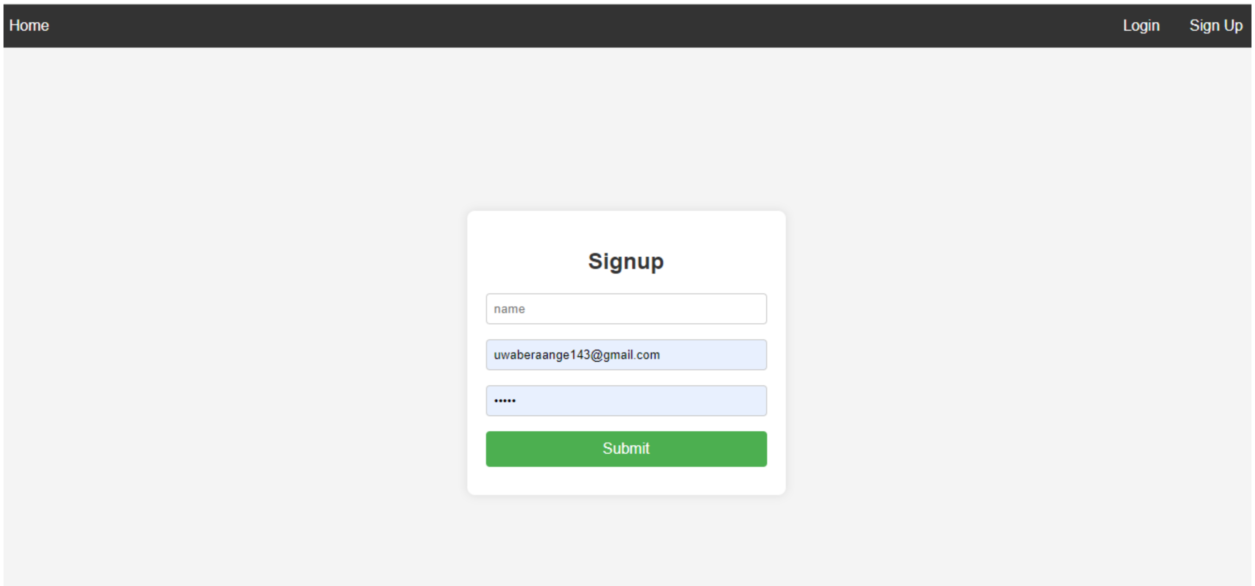
*User Documentation

This is my home page



Welcome to Library Management System

For the new user to Enter in the system need to make Registration after the Login as user



Signup

name

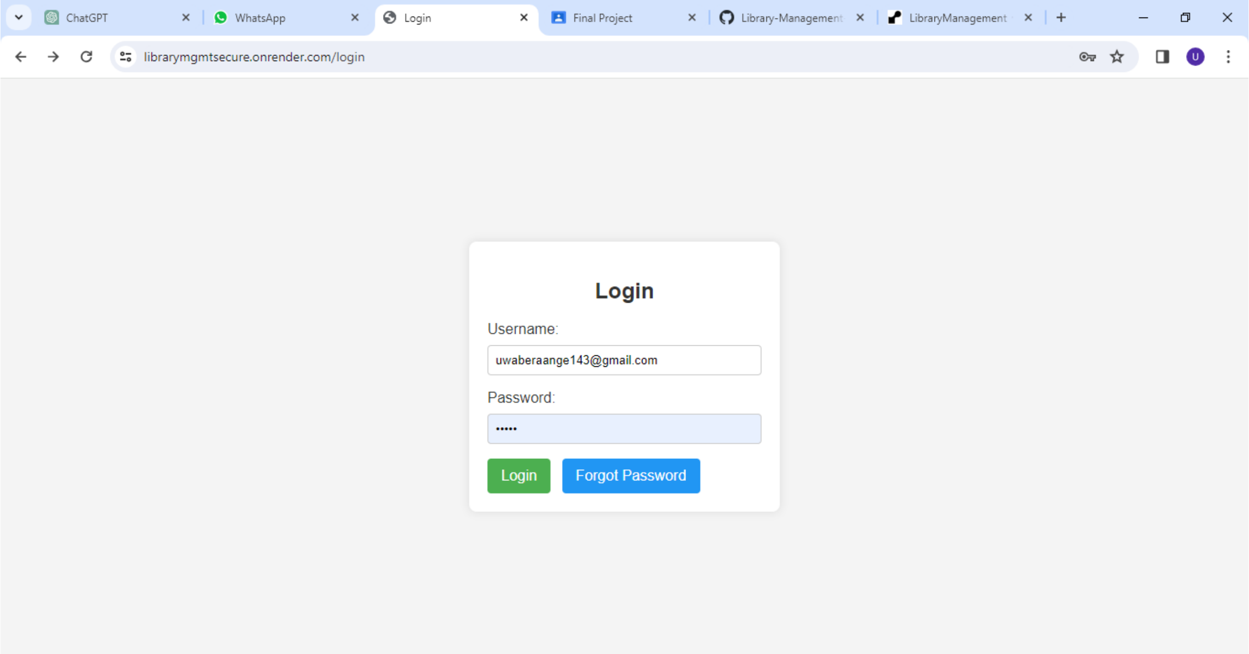
uwaberaange143@gmail.com

•••••

On Login page, user should Enter the username and password entered during registration otherwise the system will not allow him to continue. For this tutorial let use .

Username: uwaberaange143@gmail.com

Password :12345



librarymgmtsecure.onrender.com/login

Login

Username:

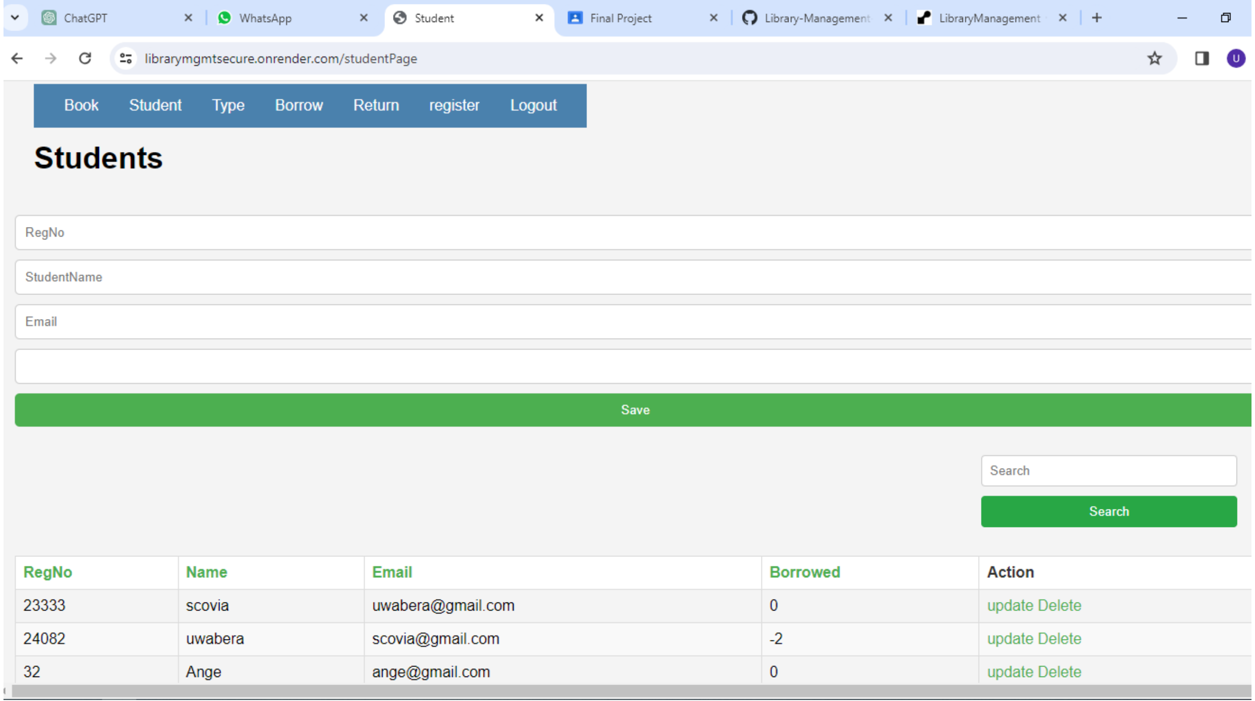
uwaberaange143@gmail.com

Password:

•••••

When login successfully user will direct to his or her account where he/she can perform many thing like Adding books, add students, add type of books, see the borrowed book, and the returned book , and also we can search all of them by their names

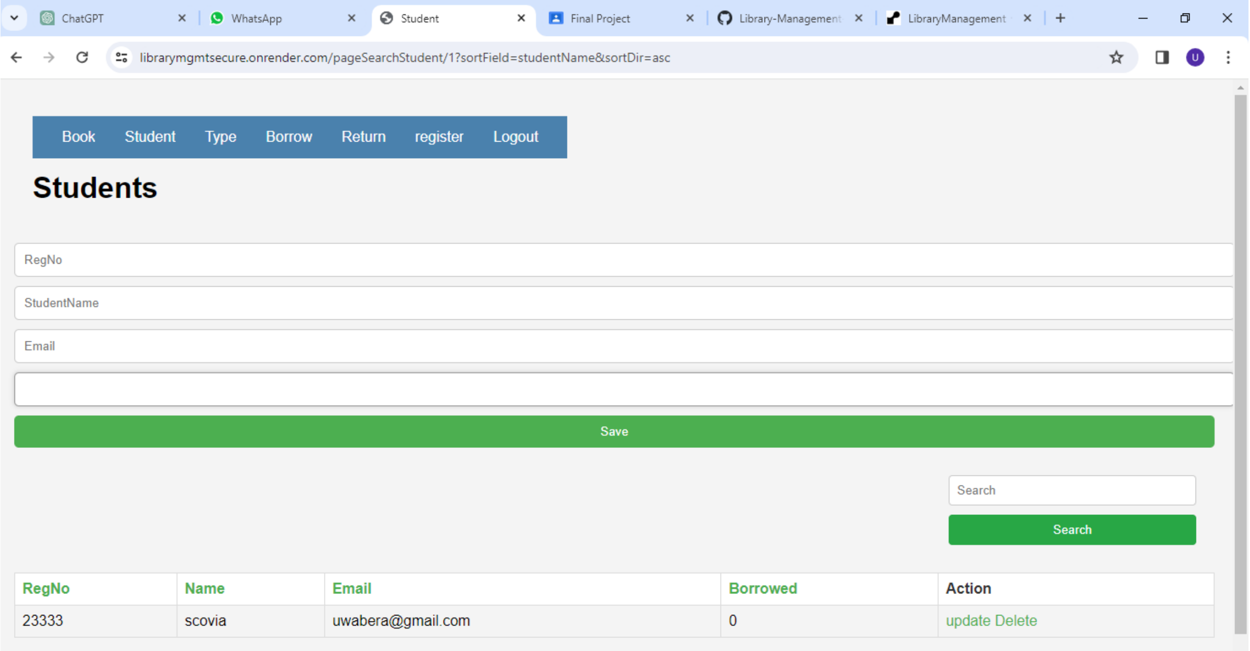
For the below picture I added student



Students

RegNo	Name	Email	Borrowed	Action
23333	scovia	uwabera@gmail.com	0	update Delete
24082	uwabera	scovia@gmail.com	-2	update Delete
32	Ange	ange@gmail.com	0	update Delete

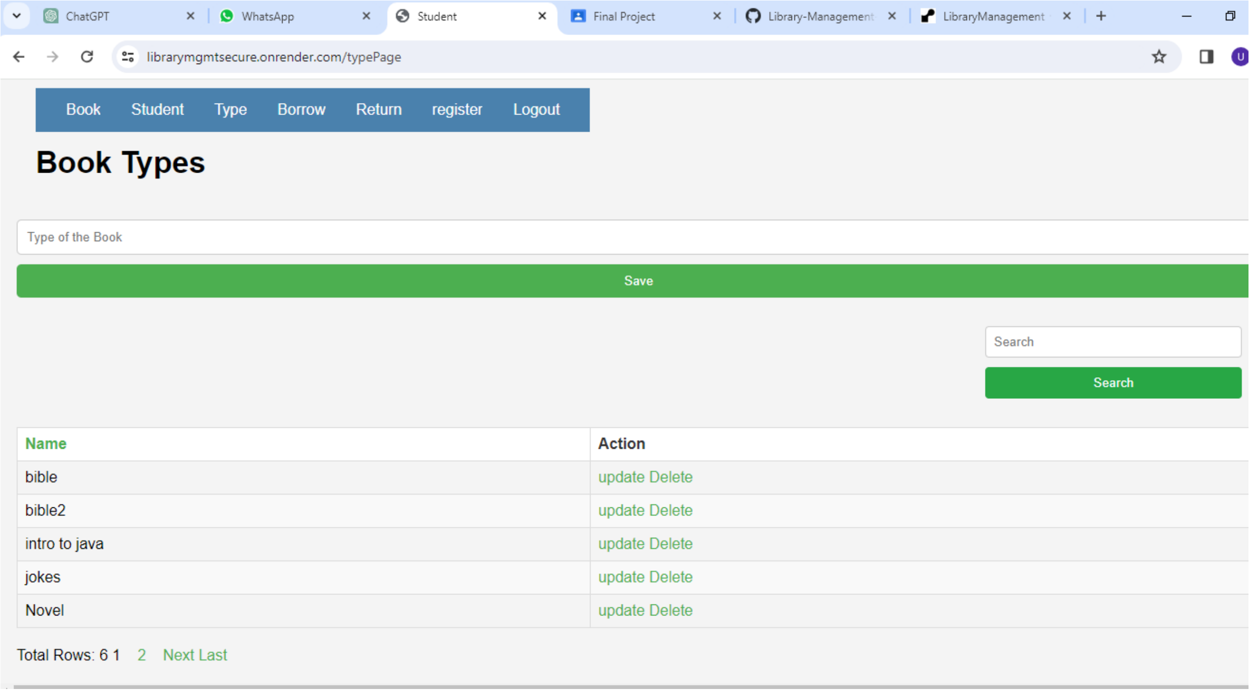
For the below picture I searched the student



Students

RegNo	Name	Email	Borrowed	Action
23333	scovia	uwabera@gmail.com	0	update Delete

For the below picture I added the book type

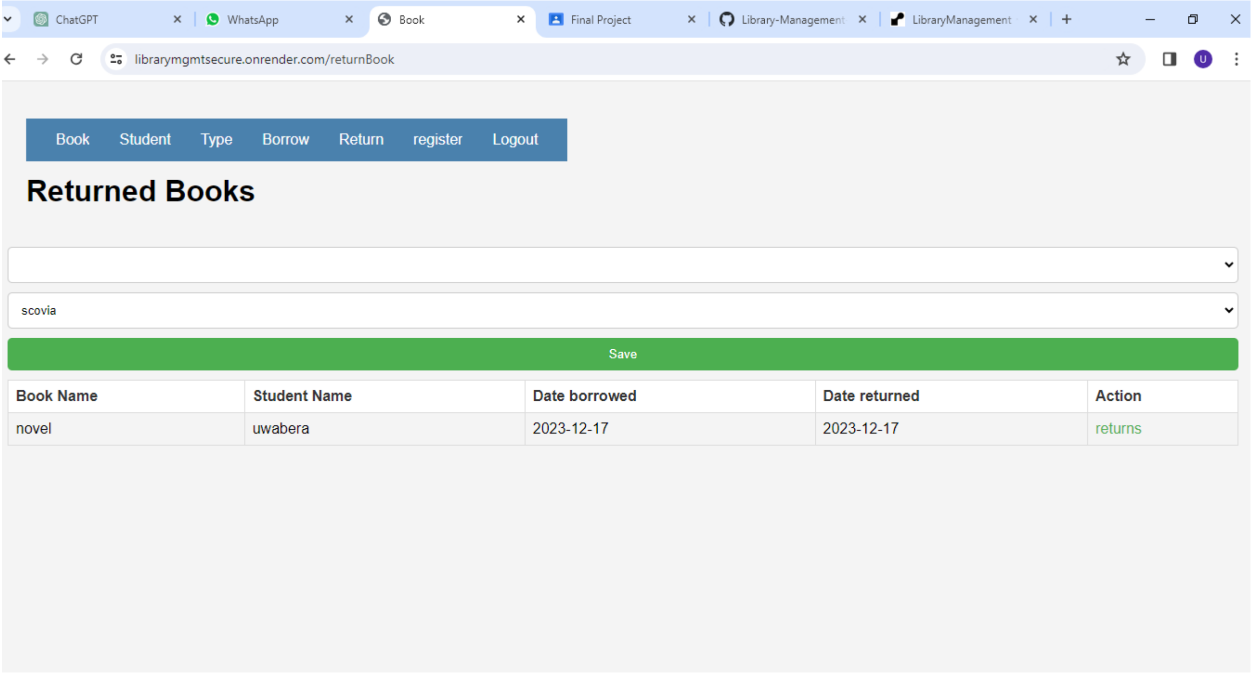


Book Types

Name	Action
bible	update Delete
bible2	update Delete
intro to java	update Delete
jokes	update Delete
Novel	update Delete

Total Rows: 6 1 2 Next Last

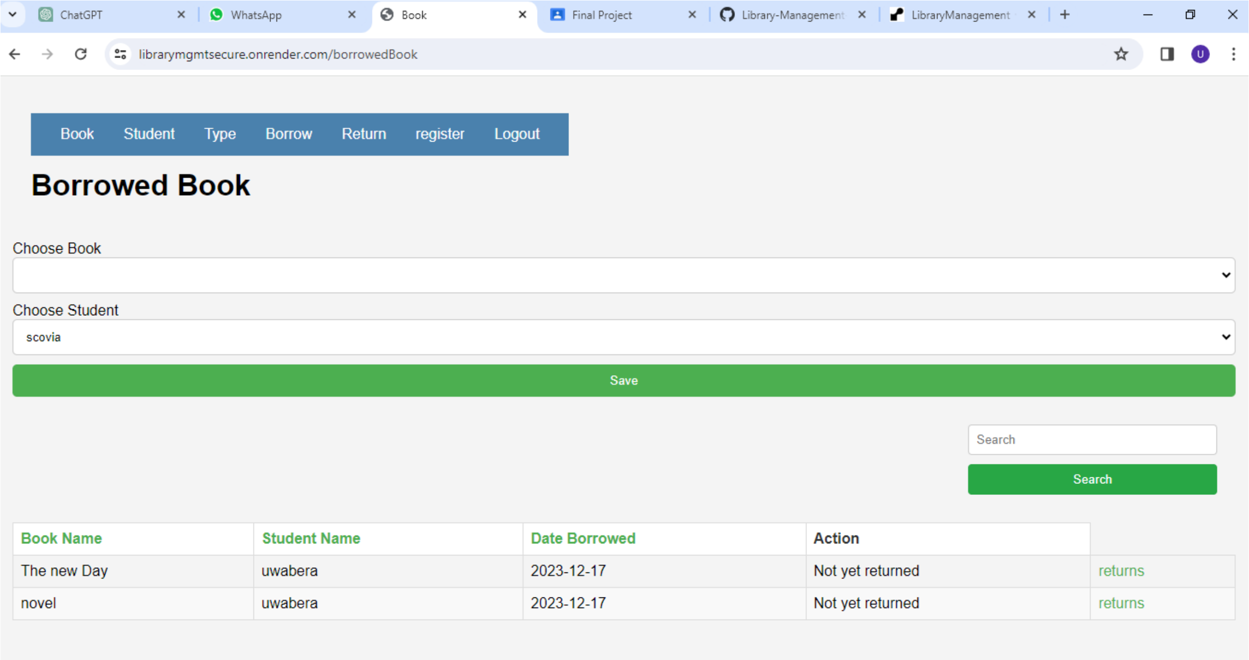
For the below picture I can view the returned books



Returned Books

Book Name	Student Name	Date borrowed	Date returned	Action
novel	uwabera	2023-12-17	2023-12-17	returns

For the below picture I can view the borrowed books



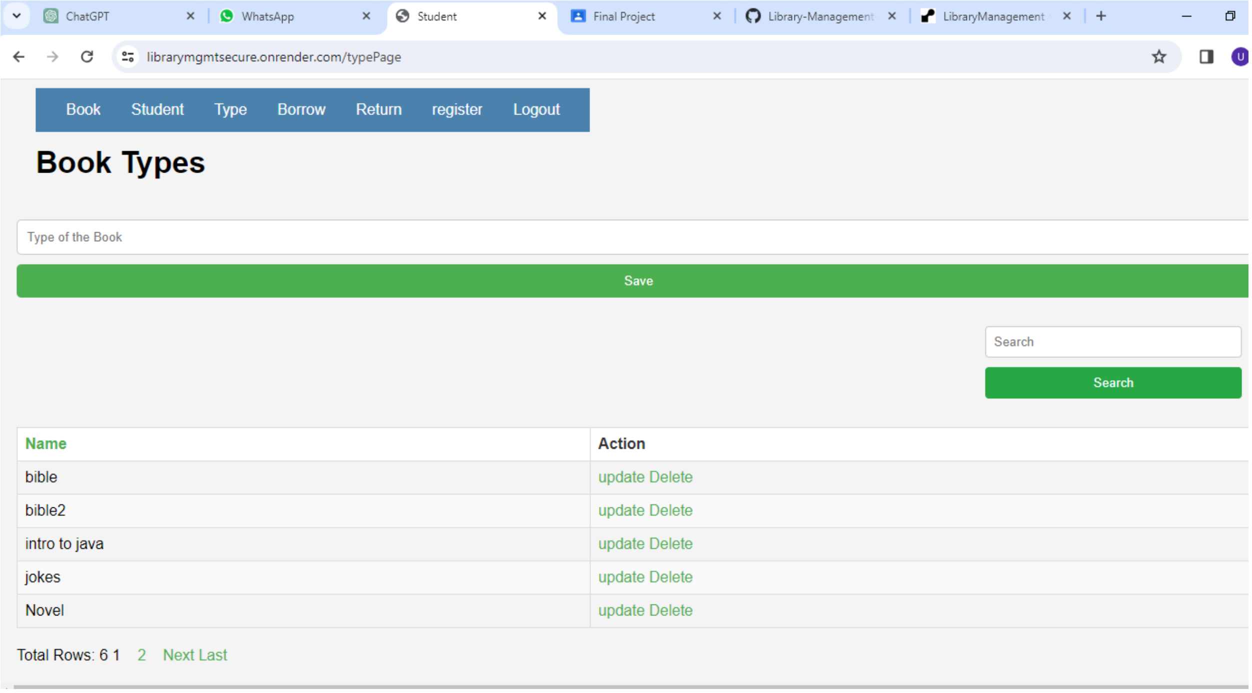
Borrowed Book

Choose Book

Choose Student: scovia

Book Name	Student Name	Date Borrowed	Action	
The new Day	uwabera	2023-12-17	Not yet returned	returns
novel	uwabera	2023-12-17	Not yet returned	returns

For the below picture I can add and view the book types and also there is pagination

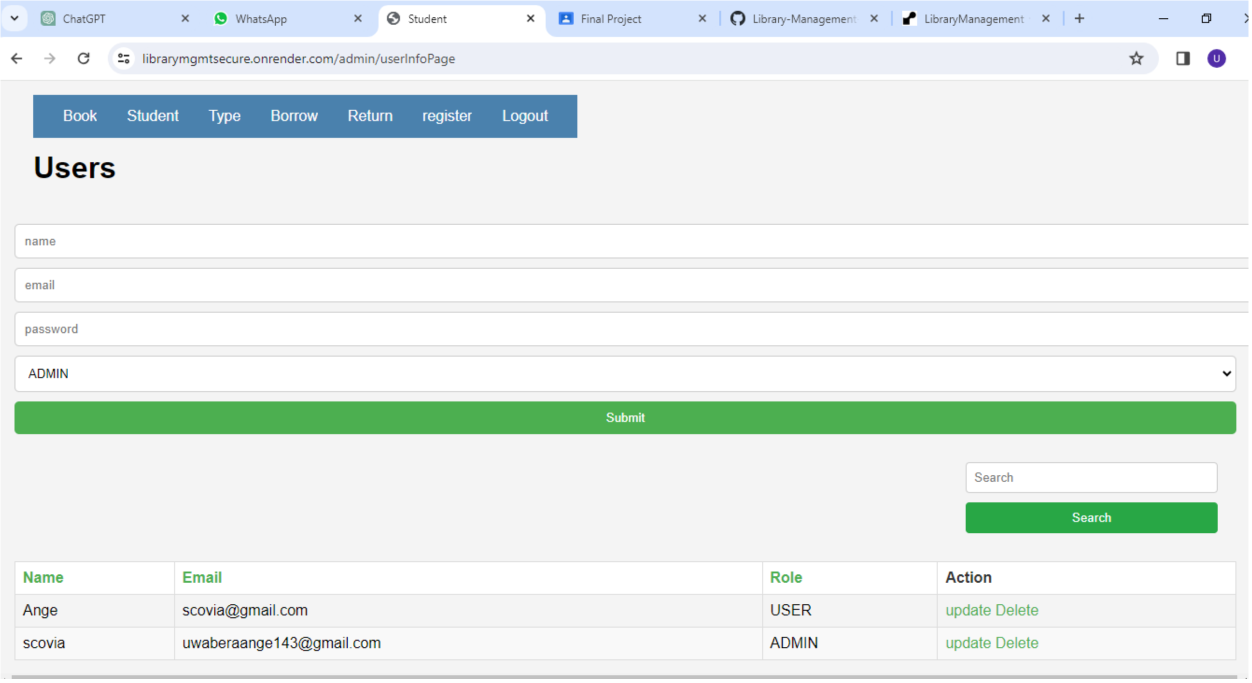


Book Types

Name	Action
bible	update Delete
bible2	update Delete
intro to java	update Delete
jokes	update Delete
Novel	update Delete

Total Rows: 6 1 2 Next Last

For the below picture I can add and view the Users



Users

name

email

password

ADMIN

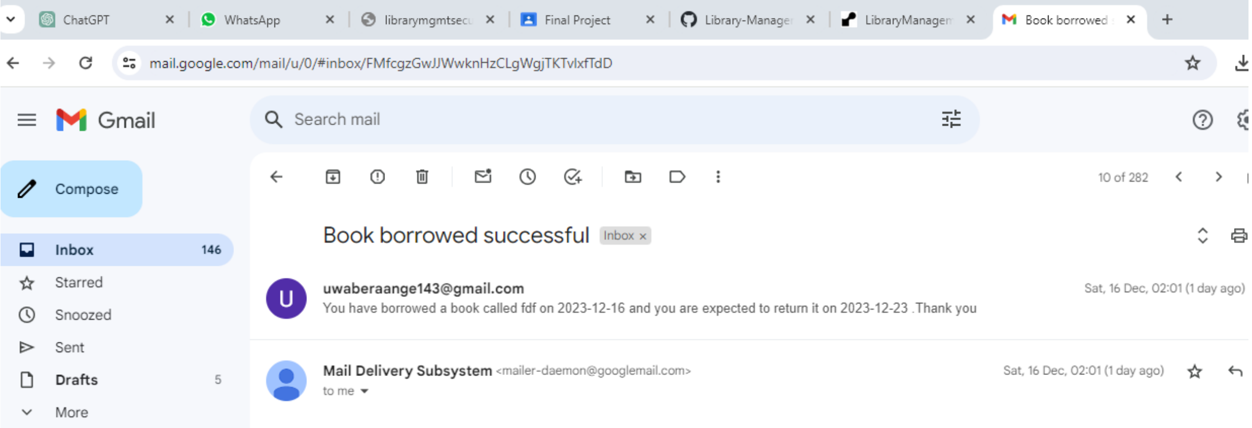
Submit

Search

Search

Name	Email	Role	Action
Ange	scovia@gmail.com	USER	update Delete
scovia	uwaberaange143@gmail.com	ADMIN	update Delete

When you borrow the book we send you the email and before two days of returning the book we sends another email to remind you

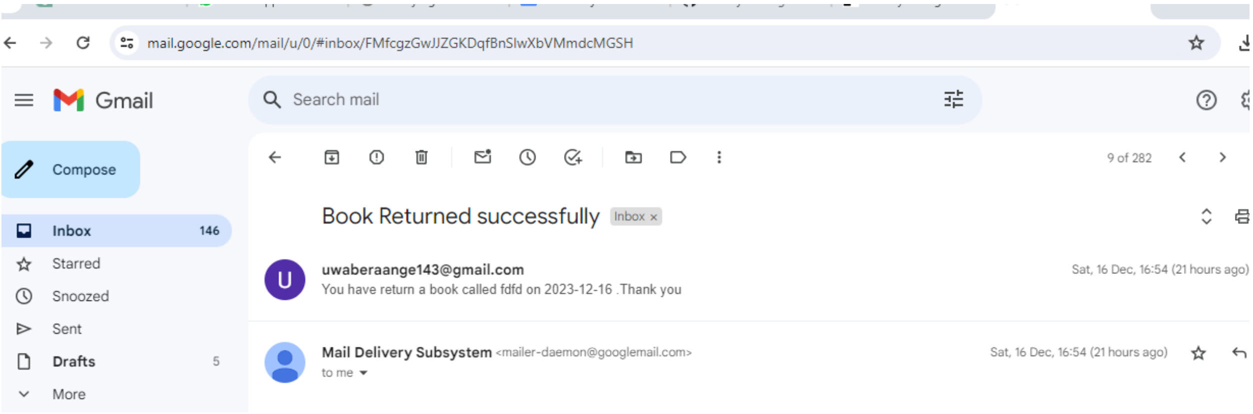


Book borrowed successful Inbox ×

uwaberaange143@gmail.com — Sat, 16 Dec, 02:01 (1 day ago)
You have borrowed a book called fdf on 2023-12-16 and you are expected to return it on 2023-12-23 .Thank you

Mail Delivery Subsystem <mailer-daemon@googlemail.com> — Sat, 16 Dec, 02:01 (1 day ago)
to me

And after returning the book we sends another email saying that we have received it



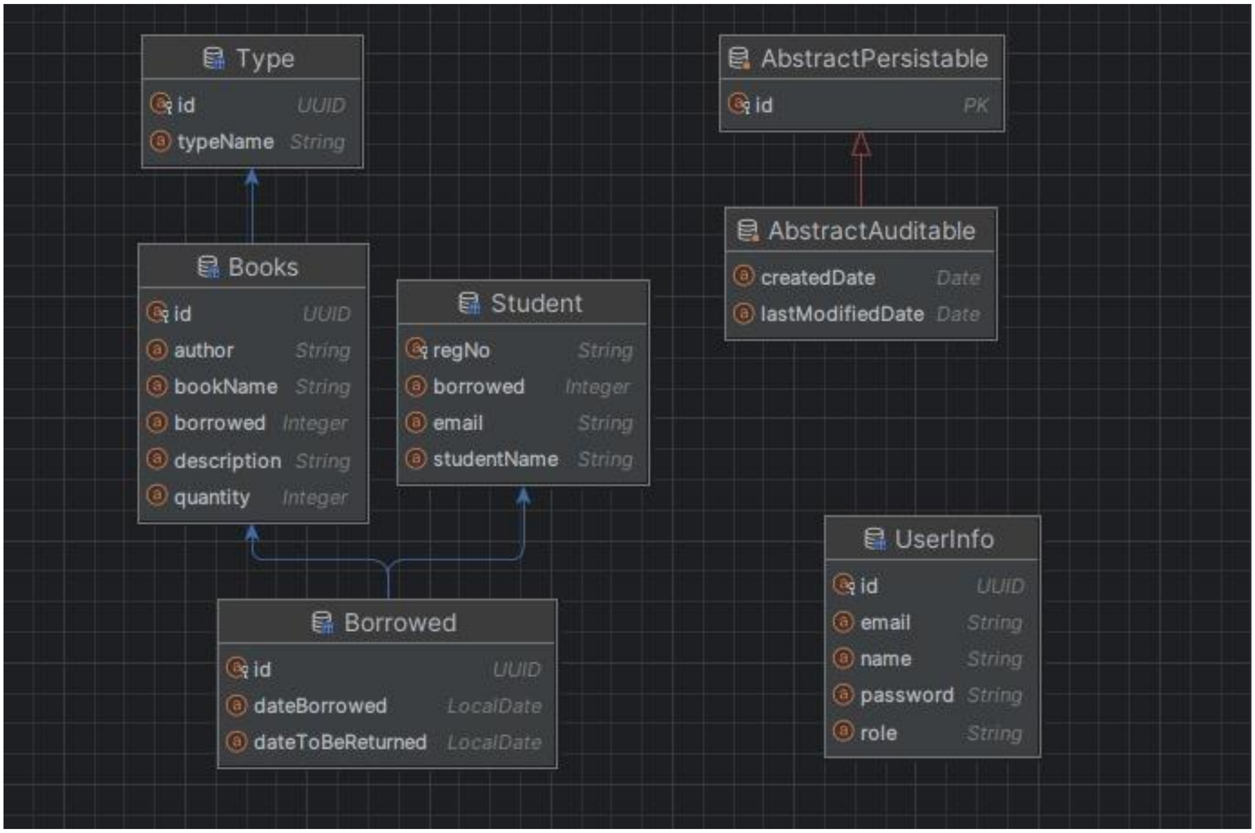
mail.google.com/mail/u/0/#inbox/FMfcgzGwJJZGKDqfBnSlwXbVMmdcMGSH

Book Returned successfully

uwaberaange143@gmail.com
You have return a book called fdfd on 2023-12-16 .Thank you
Sat, 16 Dec, 16:54 (21 hours ago)

Mail Delivery Subsystem <mailer-daemon@googlemail.com>
to me
Sat, 16 Dec, 16:54 (21 hours ago)

Below is the database schema of my project



Type	
id	UUID
typeName	String

AbstractPersistable	
id	PK

AbstractAuditable	
createdDate	Date
lastModifiedDate	Date

Books	
id	UUID
author	String
bookName	String
borrowed	Integer
description	String
quantity	Integer

Student	
regNo	String
borrowed	Integer
email	String
studentName	String

Borrowed	
id	UUID
dateBorrowed	LocalDate
dateToBeReturned	LocalDate

UserInfo	
id	UUID
email	String
name	String
password	String
role	String

*Technical Documentation:

Architecture Overview:

The Library Management System is developed using the Spring Boot framework, which follows a layered architecture pattern. The system consists of the following layers:

a**. Presentation Layer:** This layer handles user interaction and displays the user interfaces. It includes controllers responsible for handling HTTP requests and rendering appropriate responses.

b. **Service Layer:** The service layer contains the business logic of the application. It encapsulates the core functionalities and acts as an intermediary between the presentation and data access layers.

c. **Data Access Layer:** This layer interacts with the database and handles data persistence. It includes repositories or DAOs (Data Access Objects) responsible for querying and manipulating data.

d. **Database**: The system uses a relational database management system (RDBMS) such as MySQL for data storage.

Implementation Details:
The Library Management System is implemented using Java programming language and the Spring Boot framework. Some key implementation details include:

a. **Spring Boot:** The system leverages the Spring Boot framework for rapid application development and to simplify configuration and deployment.

b. **Spring MVC:** Spring MVC (Model-View-Controller) is used to handle the presentation layer and manage the flow of requests and responses.

c. **Spring Data JPA:** Spring Data JPA is used to simplify the implementation of the data access layer. It provides abstractions and automatic CRUD (Create, Read, Update, Delete) operations for interacting with the database.

d. **Hibernate:** Hibernate is used as an Object-Relational Mapping (ORM) tool to facilitate the mapping between Java objects and the underlying database tables.

e. **Security**: Spring Security is utilized for implementing authentication and authorization mechanisms, ensuring secure access to the system's resources.

f. **Dependency Management**: Maven is used for managing project dependencies and building the application.

Additional Technical Details:
Here are some additional technical details relevant to the Library Management System:

a. Server-Side Technologies: The system runs on a Java Servlet container such as Apache Tomcat.

b. Front-End Technologies: The user interfaces are built using HTML, CSS, thymeleaf and JavaScript frameworks.

e. Deployment: The system can be deployed on various platforms for source code is on github and hosted on Heroku.

5. Conclusion

The Library Management System is designed to meet the needs of modern libraries by providing an automated and user-friendly solution for managing student information, book details, and transactions. The implementation of CRUD operations, search, filters, validations, and secure authentication ensures a robust and effective system.

I deployed my software using Render.
I choose Render because , it is a cloud platform that provides a straightforward and scalable way to deploy web applications and services. It supports various programming languages, frameworks, and databases

Thanks!!