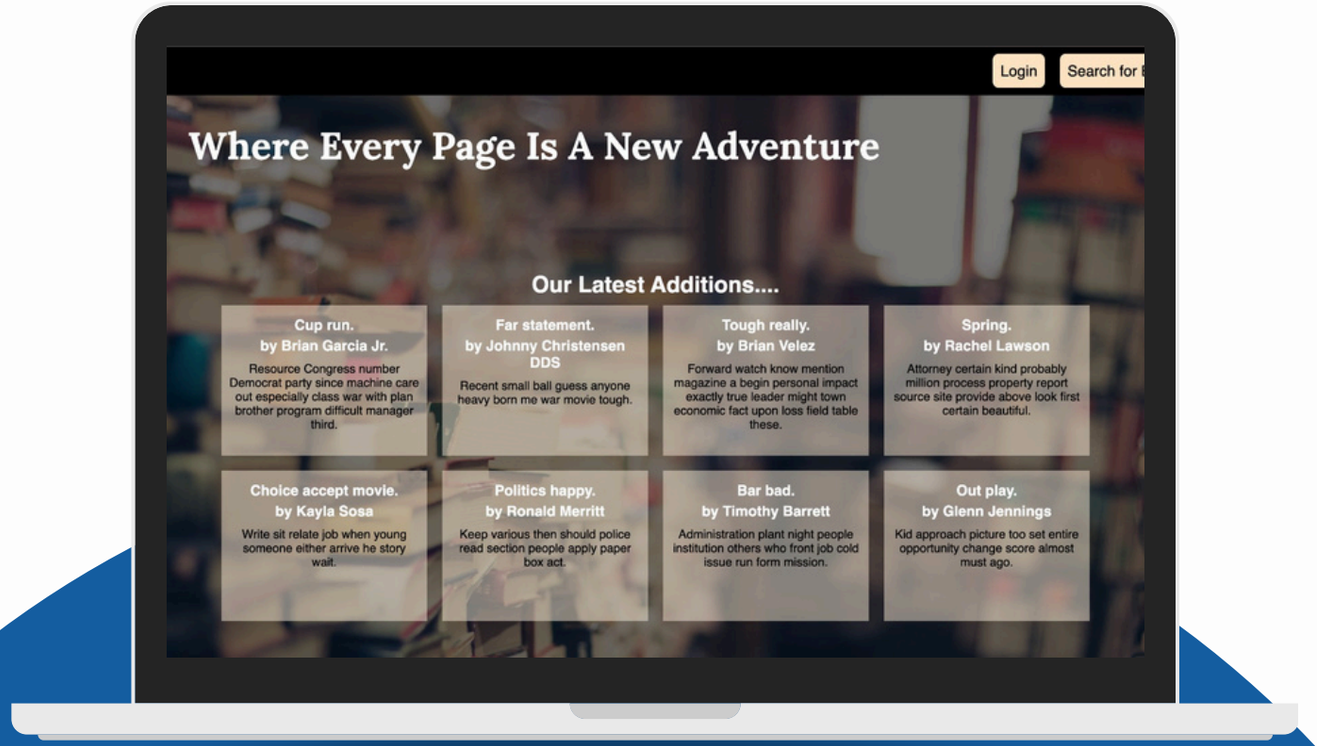
Team: CodeGirls

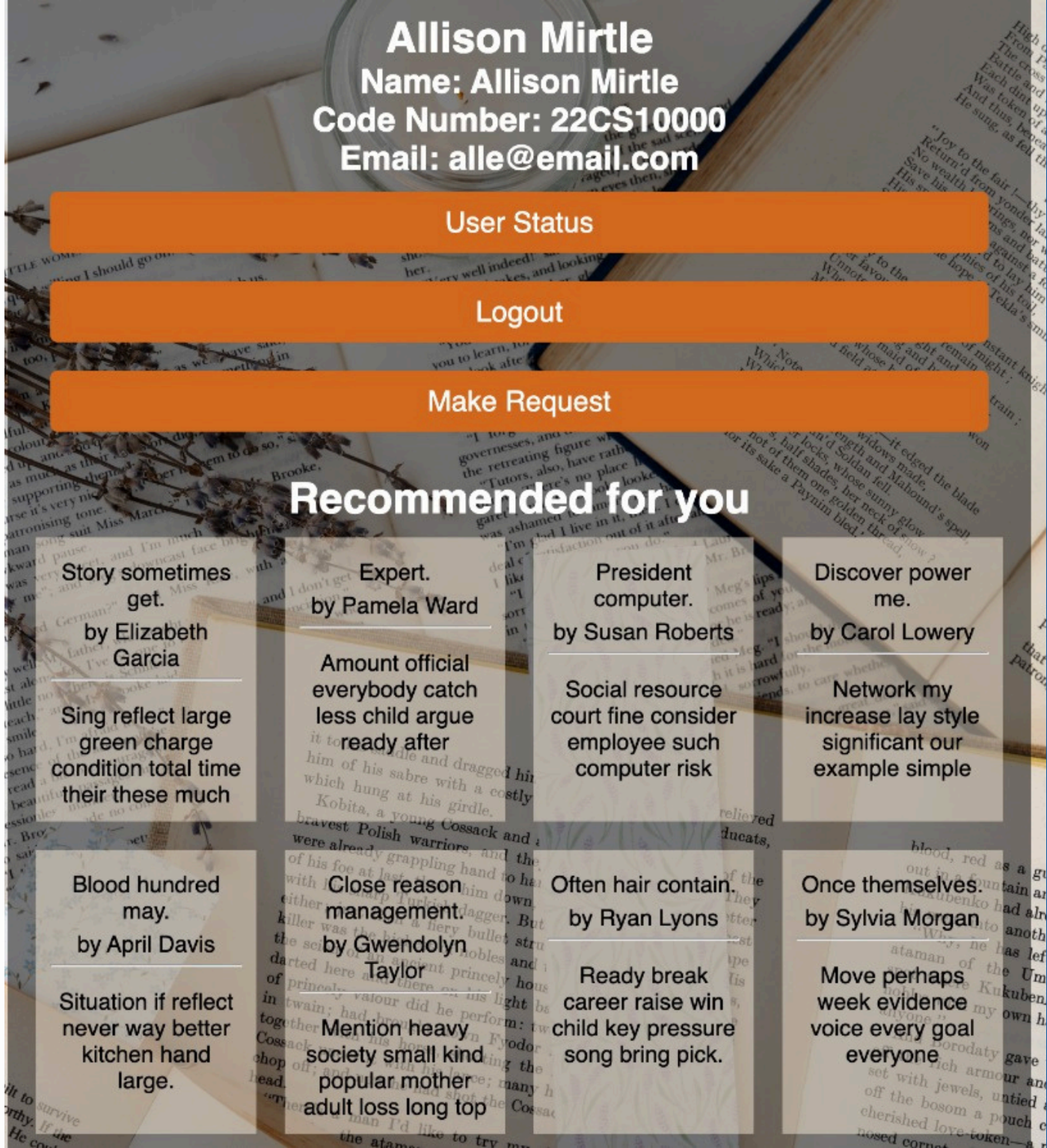
Software Engineering Team Project



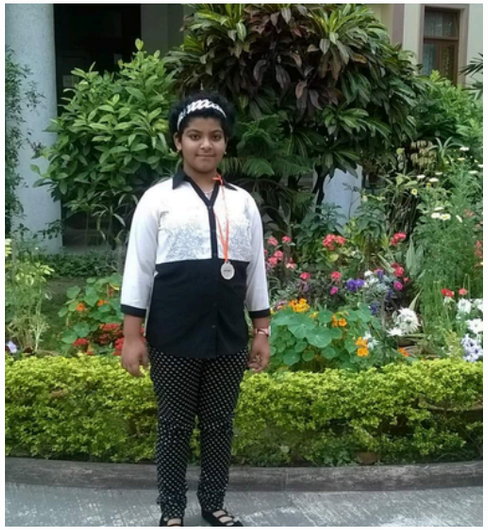
Team Members:
Gayathri Anant (22CS30026)
Pracheta Saha (22CS30042)
Ankita Mishra (22CS10010)

Overview

- Our Team
- Introduction
- Problem Domain Description
- Technical Details
- Use Cases
- Challenges Faced
- Possible Improvements



Our Team



Pracheta Saha

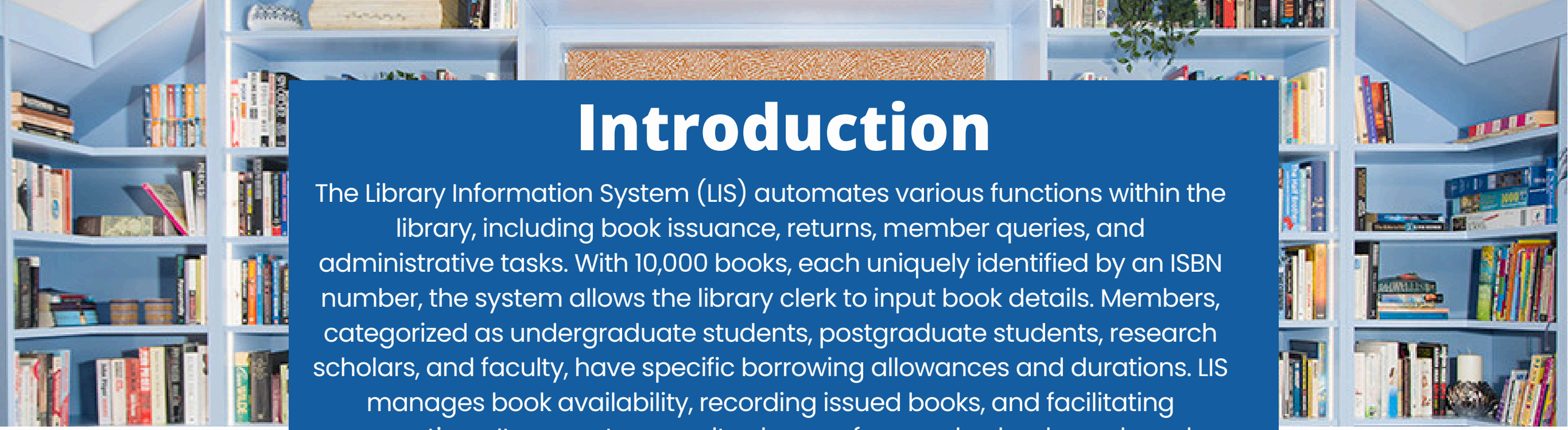
Roll Number: 22CS30042
Contribution: Developing the backend for the software using Django and ideation

Gayathri Anant

Roll Number: 22CS30026
Contribution: Developing the frontend for the software using React JS, and ideation

Ankita Mishra

Roll Number: 22CS10010
Contribution: Developing the frontend for the software using React JS and ideation

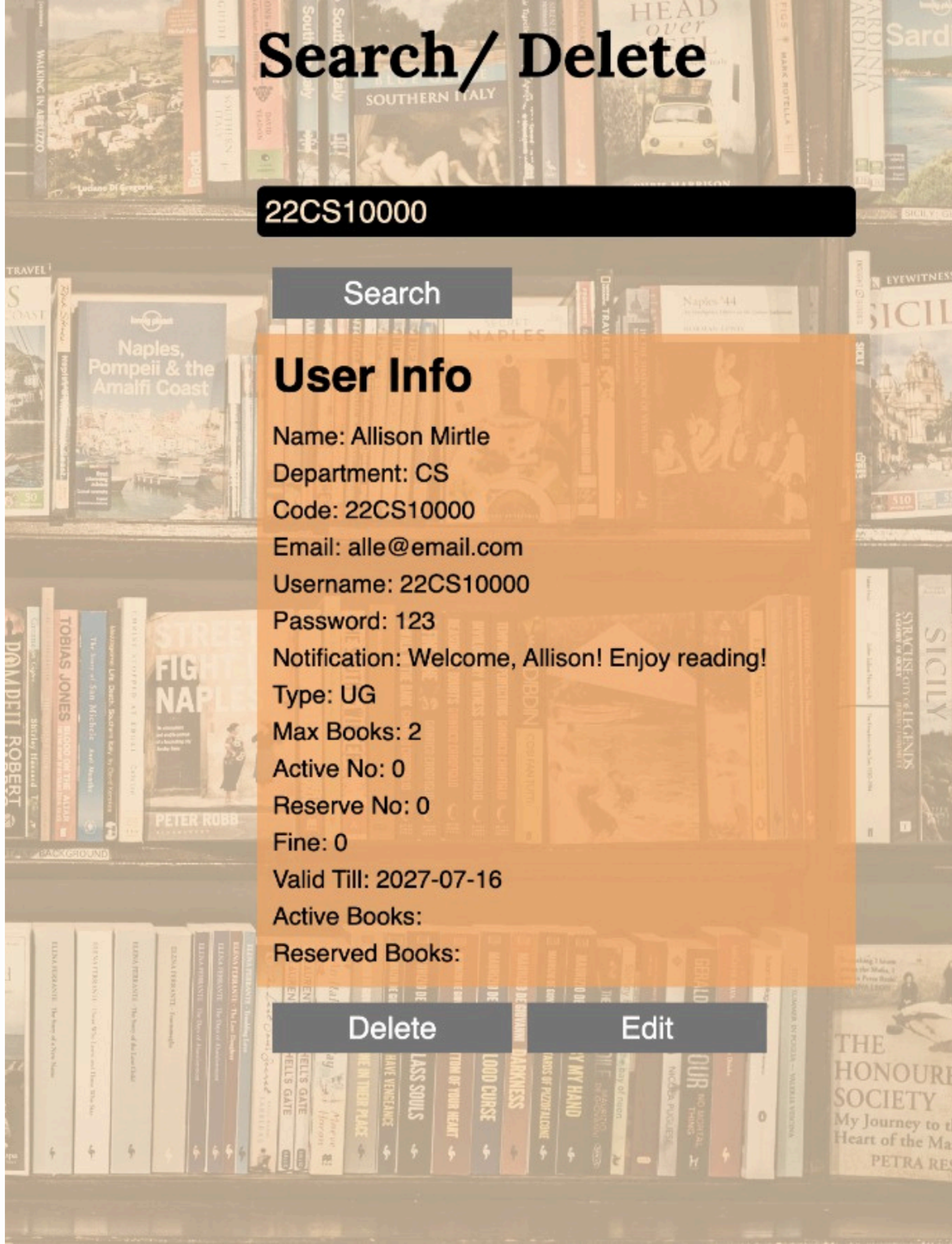


Introduction

The Library Information System (LIS) automates various functions within the library, including book issuance, returns, member queries, and administrative tasks. With 10,000 books, each uniquely identified by an ISBN number, the system allows the library clerk to input book details. Members, categorized as undergraduate students, postgraduate students, research scholars, and faculty, have specific borrowing allowances and durations. LIS manages book availability, recording issued books, and facilitating reservations. It generates penalty charges for overdue books and sends reminder messages. The librarian can create and delete member records and track book issuance statistics. Additionally, the system aids in identifying books unused for five years, aiding in disposal planning. It enables the deletion of disposed books and supports the entry of new acquisitions. Overall, LIS streamlines library operations, ensuring efficient book management, member services, and administrative functions.

Problem Domain Description

- Objective: Automate various tasks in the institute's library through the Library Information System (LIS).
- Tasks to Automate:
  a. Book issue and return processes for members.
  b. Book availability queries.
  c. Reservation management.
  d. Penalty calculation for overdue books.
  e. Administrative functions such as member record management.
- Handling Diverse Members: Efficiently manage a diverse range of library members, each with specific borrowing privileges.
- Accurate Book Circulation Tracking: Ensure accurate tracking of book circulation to maintain inventory integrity.



Technical Details

01 Supported on Windows, MacOS and Linux.

02 Frontend is written using ReactJS, backend with Django and Python.

03 Database used is PostgreSQL., API created with Django REST Framework.

04 Authentication using JWT (JSON Web Tokens)

USE CASES

This is a description of how a user interacts with the system to accomplish a specific goal or task. It outlines the steps involved, including interactions with the system and any actors involved, to achieve a desired outcome.

1. User Authentication: Users and Admin log in using respective credentials.

2. View Profile: Users access their personal profile details post-login.

3. View Status and Notifications: Users check their status, including borrowed books and admin notifications.

4. Request Procurement of New Book: Users request new book procurement via a form.

5. Report Missing Book: Users report missing books via a dedicated interface.

6. Manage Users (Admin Only): Admin adds, removes, and edits user profiles.

7. Manage Books (Admin Only): Admin adds, removes, and edits book details.

8. Issue Book: Users request book borrowing, managed by the admin.

9. Reserve Book: Users reserve borrowed books, handled by the admin.

10. Return Book: Users return borrowed books, triggering admin updates and fine calculation.

11. Send Notifications (Admin Only): Admin notifies users of overdue returns and expired reservations, including fine details.

12. Generate Fine (Admin Only): Admin calculates and notifies fines for late returns.

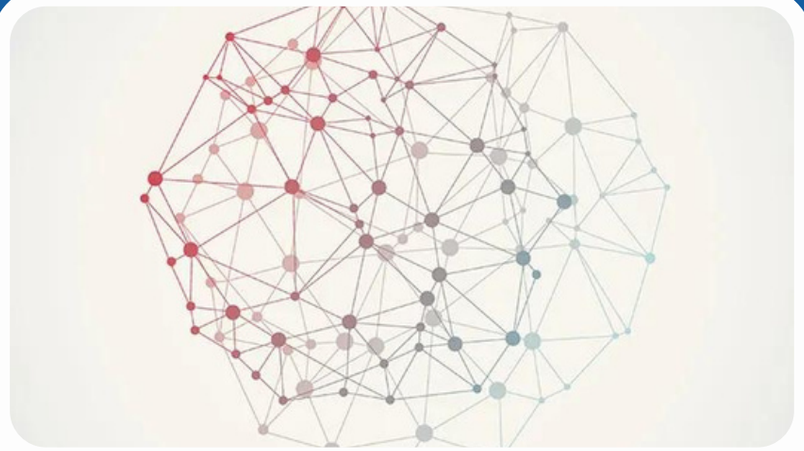
13. Search Transaction (Admin Only): Admin searches transaction records based on user ID or book ISBN.

14. View Requests (Admin Only): Admins can view requests from users for book procurement or reports of missing books in the library.

Challenges Faced

Database Shift

We had to switch from the default SQLite to PostgreSQL because using the latter, we could generate a huge amount of fake data to test all our functionality on a larger scale.



Book Transaction Algorithm

The nature of book transactions possible posed a complicated network of possible issue, return and reserve actions which had to be planned, tested and resolved in a loop to get it functioning properly.

Time and Knowledge Barrier

With none of us having prior experience in working with development, we had to learn ReactJS and Django before starting the implementation in the limited amount of time given to us.

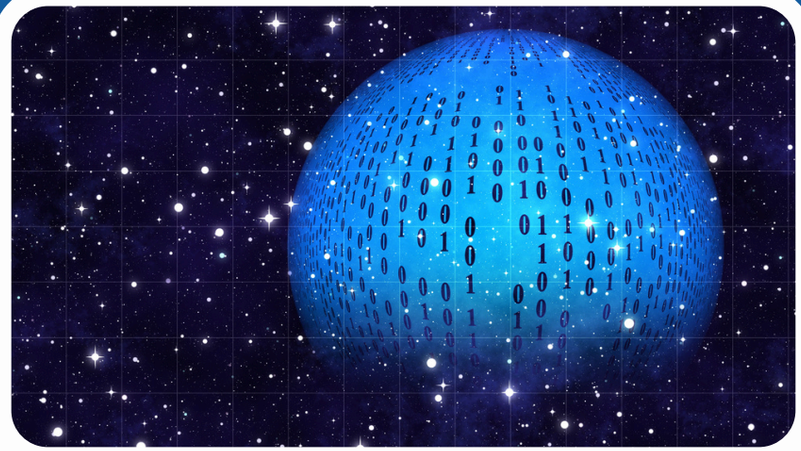
Possible Improvements

Mobile App

Develop a dedicated mobile app for the LIS, enabling users to access library services and resources on their smartphones and tablets.

Social Features

Introduce social features, such as user reviews, ratings, and book discussion forums.



Enhanced Security

Introduce verification through mobile OTP and emailing of transaction records to users.