Name: Subhrajit Samanta
Roll: 22f1001198
Email: 22f1001198@ds.study.iitm.ac.in

Project Report

Library Management System V2

1. Introduction

The Library Management System V2 is a multi-user web application designed to manage and maintain e-books across various sections within an online library. The system enables users to browse, request, and access e-books, while librarians have the authority to manage sections, e-books, and user access. The system is built using the Flask framework for the backend, VueJS for the frontend, and integrates SQLite for data storage, Redis for caching, and Celery for batch processing.

2. Technology Stack

2.1. Backend

- **Flask:** Used for building the API to handle all server-side logic and operations.
- **SQLite:** Lightweight database for storing all application data.
- **Redis:** Used for caching and handling real-time notifications and batch jobs.
- **Celery:** Task queue to manage asynchronous tasks such as sending reminders and generating reports.

2.2. Frontend

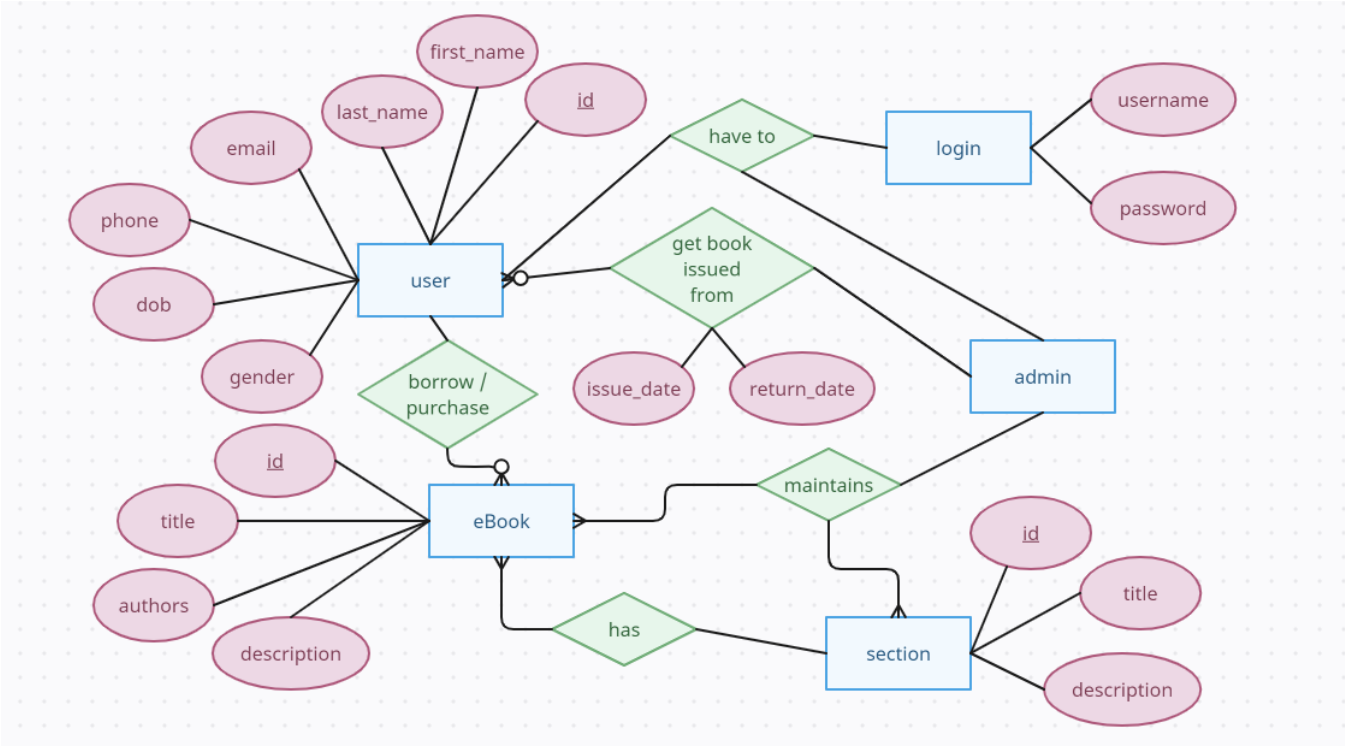
- **VueJS:** Modern JavaScript framework for a responsive and dynamic user interface.
- **Bootstrap:** CSS framework used for styling HTML elements and creating unified, responsive UI.
- **Jinja2:** Template engine integrated with Flask, used where dynamic content generation is required, but not used for the UI.

3. Database Schema

1. **roles_users:** id (PK), user_id (FK user.id), role_id (FK role.id).
2. **user:** id (PK), first_name, last_name, username, email, password, phone, dob, gender, active, fs_uniquifier
3. **role:** id (PK), name, description
4. **section:** id (PK), title, date_created, description, active
5. **ebook:** id (PK), title, section_id (FK section.id), authors_name, description, date_created, active, rating
6. **book_issued:** id (PK), ebook_id (FK ebook.id), user_id (FK user.id), issue_date, return_date

7. **book_request:** id (PK), ebook_id (FK ebook.id), user_id (FK user.id), request_date, required_day
8. **return_list:** id (PK), ebook_id (FK ebook.id), user_id (FK user.id), issue_date, return_date
9. **revoke_list:** id (PK), ebook_id (FK ebook.id), user_id (FK user.id), revoke_date
10. **delete_ebook:** id (PK), ebook_id (FK ebook.id), delete_date
11. **delete_section:** id (PK), section_id (FK section.id), delete_date
12. **rating:** id (PK), ebook_id (FK ebook.id), user_id (FK user.id), rating, feedback, date
13. **purchase:** id (PK), ebook_id (FK ebook.id), user_id (FK user.id), date.

4. ER Diagram



4. Roles and Responsibilities

4.1. Librarian: Manage Sections & eBooks, Grant/Revoke Access on eBook, Monitoring user & eBook.

4.2. General User: Browse Sections & e-Books, Request e-Books, Return e-Books, Provide Rate & Feedback, Profile Management.

5. Features Implemented

1. Implemented login and registration forms with role-based access control using Flask security.
2. Added functionality to create, delete sections in the library.
3. Enabled creating and deleting e-books within specific sections.
4. Librarians can view requests and grant/revoke e-book access.

5. Implemented search for e-books and sections based on various criteria.
6. Users can request up to 5 e-books and must return them within a specified period. Otherwise, the access for that will be automatically revoked the time period.
7. Users can provide ratings and feedback for e-books.
8. Users can download e-books as PDFs after purchase.
9. Created a mock payment portal for handling e-book purchases.
10. Scheduled daily reminders to users via email.
11. Created a monthly report for librarians, sent via email on the first day of each month.
12. Enabled librarians to trigger the export of e-book data as a CSV file.
13. Generated monthly activity reports for user, sent via email on the first day of each month.
14. Implemented caching to improve API performance.
15. Used external libraries like ChartJS for data visualization.

6. Video link:

https://drive.google.com/file/d/1VfPmutzXKGmhjPMhN6BPYa9v1HYKsMKm/view?usp=sharing