

LIBRARY MANAGEMENT SYSTEM

Final Project Report

BY RISHU RAJ GAUTAM

22F2000034 | Modern Application Development - I | Jan, 2024 Term

AUTHOR

Name - Rishu Raj Gautam

Roll No – 22f2000034

Email – 22f2000034@ds.study.iitm.ac.in

About me - I leverage data to drive impactful applications and strategic decisions. Combining a background in Mathematics & Computer Applications with a passion for coding, I'm adept at Python, web development frameworks, and machine learning. I'm also enthusiastic about education, creating content to empower others.

DESCRIPTION OF PROJECT

A library may operate more efficiently by managing books, users, transactions, and comments with the use of a Library Management System, a comprehensive software solution. With tools for managing user accounts, cataloguing books, processing loan and return procedures, gathering feedback, and more, this system meets the demands of users, librarians, and administrators.

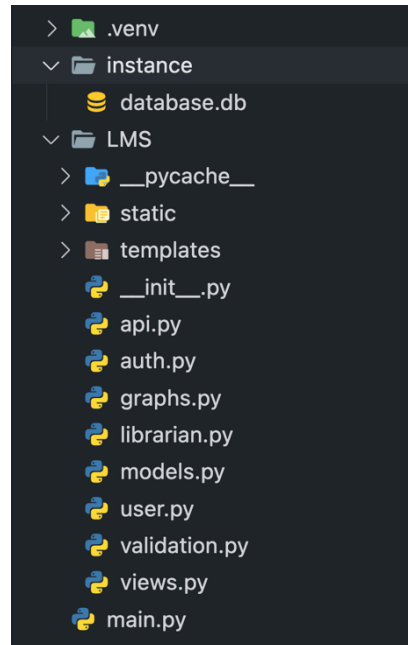
TECHNOLOGIES USED

- **Flask:** Backend framework for building the web application.
- **SQLAlchemy:** ORM (Object-Relational Mapping) tool for database interactions.
- **SQLite :** Database management system for storing application data.
- **HTML/CSS/JavaScript:** Frontend technologies for user interface design and interactivity.
- **Flask-Login:** Extension for managing user sessions and authentication.
- **Datetime:** Python library for handling date and time operations.

ARCHITECTURE

Here, The Main.py file contains the main code to run the webapp.LMS(Library Management System) is a module which contains all the files for app. The description of files inside LMS are as follows

- **Api.py** – This file contains UserAPI,SectionAPI,BookAPI
- **Auth.py** – This file contains Authorization logic like login, signup
- **Graphs.py** – This file contains graphs logic which I have generated for dashboard
- **Librarian.py** – This contain all functionalities related to Librarian
- **Models.py** – This contains DB Classes
- **User.py** – This contains all functionalities related to User
- **Validation.py** – This contains error handling logic for API
- **Views.py** – This contain Normal view for guest user
- **Templates** – This contains all HTML templates for User,Librarian,authorization
- **Static** – This contains CSS and image files



FEATURES

User Management:

- Registration and authentication for users (librarians and regular users).
- User roles (librarian or regular user) to control access and permissions.
- User profile management including personal details like name, email, contact number, etc.

Book Management:

- Cataloging and indexing of books with details such as title, author, ISBN, content, availability status, etc.
- Classification of books into sections for easy navigation and organization.
- Ability to add, edit, and delete books from the system.

Transaction Management:

- Tracking of book transactions including borrowing and returning activities.
- Automatic updates to book availability status upon transactions.
- Due date management and reminders for users to return borrowed books on time.

Borrow Request Management:

- Facility for users to request book borrowings.
- Approval workflow for librarian review and processing of borrow requests.
- Monitoring and updating the status of borrow requests (e.g., pending, approved, rejected).

Feedback System:

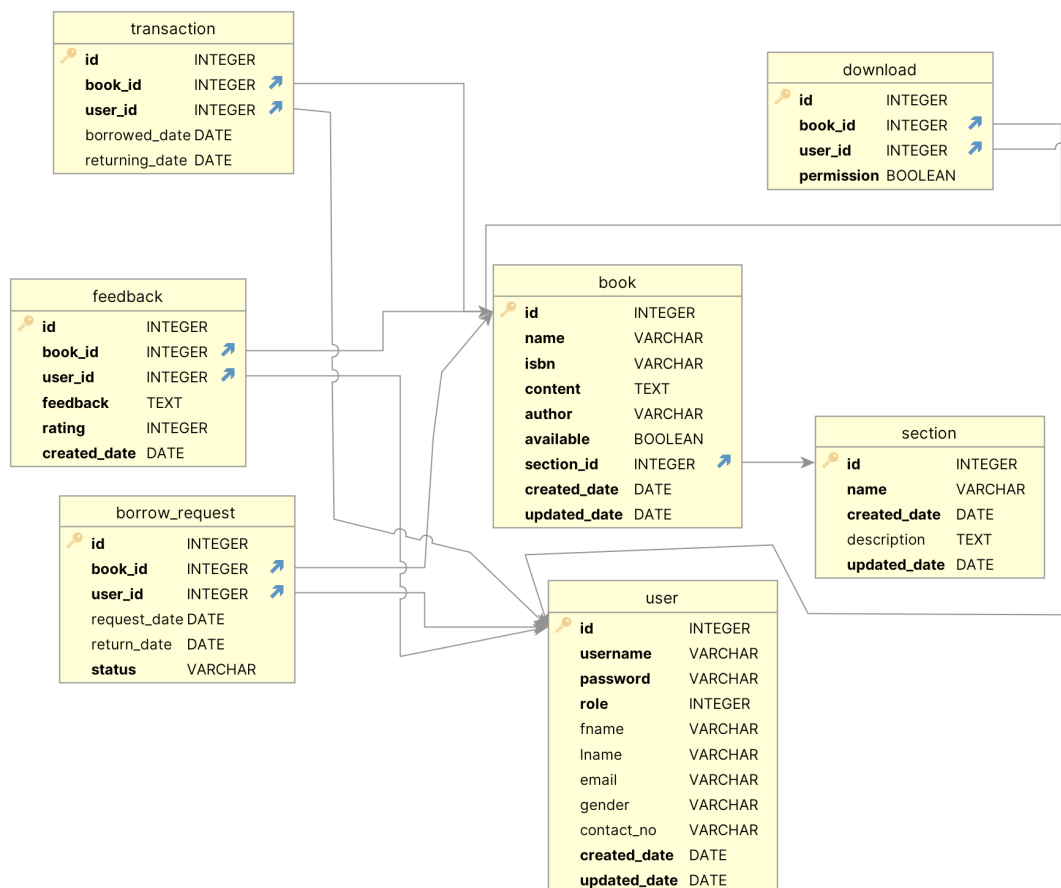
- Collection of user feedback on borrowed books.

Download Access Control:

- Provision for users to download digital copies of books (if available).
- Permission-based access control to regulate downloading privileges.

DB SCHEMA DESIGN

The database schema consists of tables for users, sections, books, transactions, borrow requests, downloads, and feedback. Relationships are established between these tables to track user activities such as borrowing, requesting, downloading, and providing feedback on books. Each table contains relevant fields such as user details, book information, transaction dates, request statuses, and feedback text, ensuring comprehensive data management within the library system.



FUTURE ENHANCEMENTS

- Implementing a reservation system for requested books.
- Adding advanced search and filtering options for books.
- Enhancing the user interface for better user experience.
- Integrating email notifications for overdue books, approved requests, etc.

VIDEO LINK

Video demonstration of my project is available here.

<https://drive.google.com/file/d/1vh2i2XRaovL6sj4rwyCP6NqnG1F5SDYK/view?usp=sharing>

THANK YOU