MAD 2 - LMS REPORT

Author

R SRIMATHI
21f3002895
21f3002895@ds.study.iitm.ac.in

I am a tech enthusiast with a passion for data science and programming. Currently, I'm in my final term of my diploma and the final year of my Bachelor's in Computer Science and Engineering. This project has been an excellent opportunity for me to apply what I've learned and further develop my skills.

Description

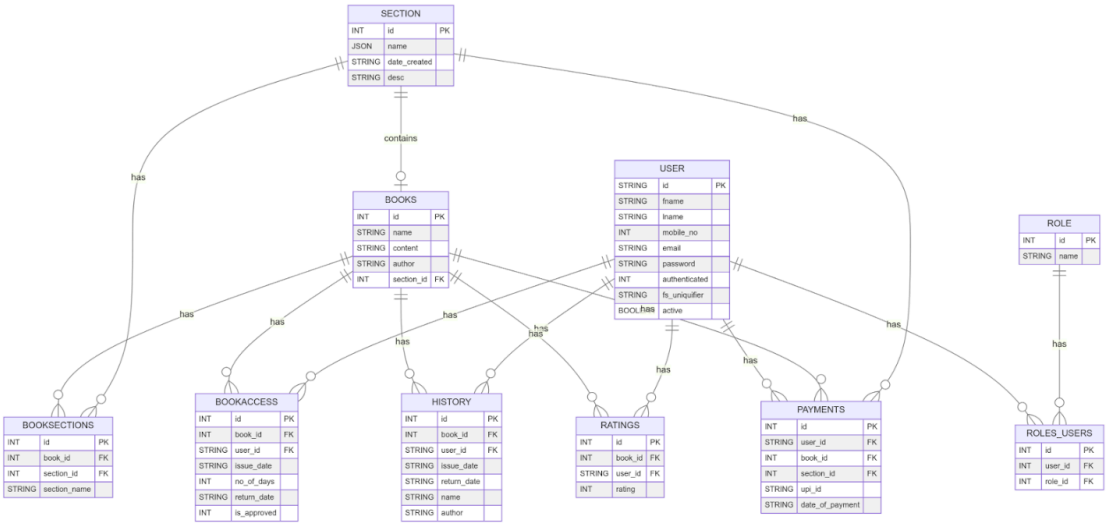
It is a Library Management System built using Flask where different users can borrow books for a specific duration and buy books from various sections or genres. The librarian has the ability to manage the books and sections, grant or reject user requests, and access detailed statistics of the application.

Technologies used

- VS Code
- Postman
- Flask
- VueJs
- Flask-Restful (For api services)
- Flask-SQLAlchemy (For databases)
- Flask-Mail (For Mailing)
- ReportLab(For PDF reports)
- Vue3-apexCharts (Chart Visualization)
- Redis (For message broker and caching)
- Flask-Caching
- Celery
- Flask-Security

DB Design Schema

I have divided the databases into several tables to simplify data access and manipulation. I meticulously designed the schema, incorporating appropriate primary and foreign keys to ensure proper linking of all elements. Below is the ER Diagram illustrating the database structure



Features implemented

Librarian Features:

- **Login:** Login for librarian.
- **CRUD Operations**: Create, Read, Update, and Delete operations for books and sections.
- **User Requests Management**: Grant or reject user requests for borrowing books.
- **Application Statistics**: View comprehensive statistics of the application.

User Features:

- **Login and Registration**: Users can create accounts and log in.
- **Borrow, Buy, and Return Books**: Users can borrow (up to 5 books at a time), buy, and return books.
- **Profile Management**: Users can view their profile, including currently reading, currently requested, and past read books.
- **Payment Portal**: Sample portal for buying books.

General Features:

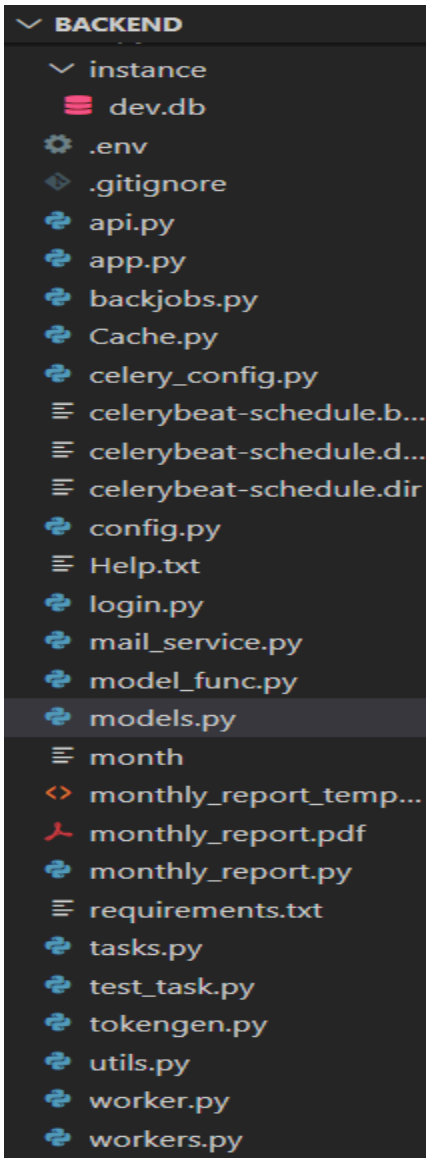
- **Report and Reminder System**:
    - Monthly report generation and daily reminders.
- **Add to Desktop**: Feature to add the application to the desktop for quick access.
- **Scheduled Tasks**:
    - Cron jobs for users and librarians and users.
- **Caching**: Improved performance using Redis for caching.
- **Search Functionality**: Search capability for both users and librarians.

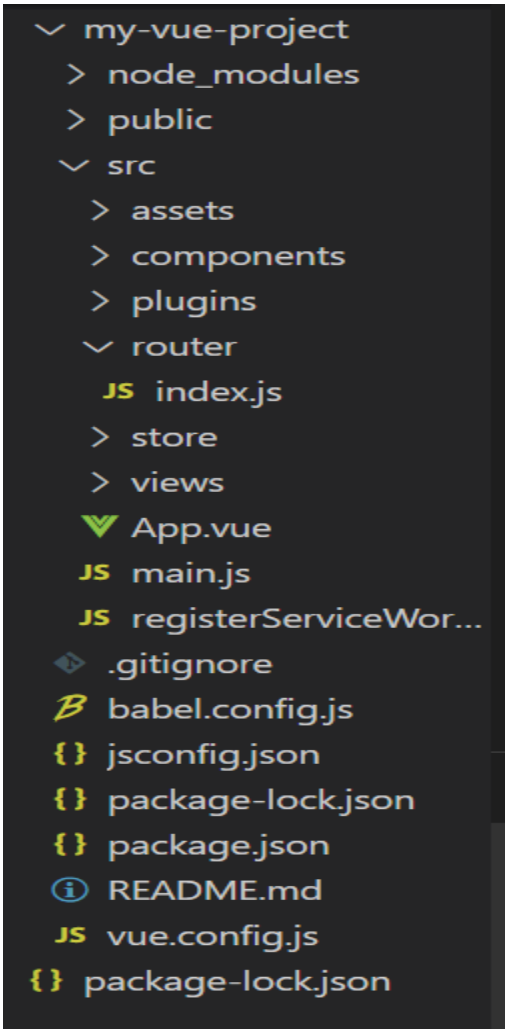
Architecture

The project is divided into two main parts: the backend and the frontend.

- **Backend**: Developed using Flask.
- **Frontend**: Developed using Vue.js (my-vue-project).

The detailed architecture of each component is shown below.





Video Link : iew-anmm-xqe (2024-08-06 17:57 GMT+5:30)