
ABOUT ME:

NAME: SOHAN SRI SHAMANTH V R

ROLL NO:22F3001144

EMAIL: 22f3001144@ds.study.iitm.ac.in

PROJECT NAME : LIBRARY MANAGEMENT SYSTEM

OVERVIEW

This is a multi-user app built by using the where a student/user by registering oneself, can request books available in the library, download it after paying the required amount for each book. Here the other role is of Admin's where he can add books to a particular section, edit content of the book, grant/cancel requests requested by the student/user. Both Admin and user can view the stats of requests and books in each section, etc..

TECHNOLOGIES USED

1. Flask libraries for the backend jobs
2. VueJs (Vue-2) for the frontend user and admin Interaction with the app
3. SQLAlchemy for the database handling and user datastore
4. 3.io module to store the pdf files of books as binary data
5. Caching the routes which are frequently to improve performance
6. Celery for the Job Scheduling
7. Redis which helps in Message Queuing
8. Matplotlib for showing analysis chart

DATABASE SCHEMA DESIGN

User Table → id(PK), email, password, wallet_amount, active, fs_uniquifier, no_of_books_owned

Section Table → id(PK), section-id, section-name, time of creation, description

Book Table → id(PK), book-id, book-name, author, section-id, content, amount, filename

Requests Table → id(PK), book-id, book-name, requested-date, granted-date, granted till which-date, status, user_id (who made the request), section-id

UserRoles Table → id(PK), user_id, role_id

Role Table → id(PK), name, description

Feedback Table → id(PK),book_id,user-id,rating,description

Some relationships between tables

1. section_id column in Book table is a foreign key for section_id in Section table
2. user_id in UserRoles Table is a foreign key for id attribute in User table and role_id in Role Table is a foreign key for id attribute in Role Table,
3. user_id in Request table is a foreign key for id column in User and section_id column is a foreign key for section_id column in Section table
4. book is a relationship for the table book with backref named “section” in the Section table with Book table, which also has cascade delete option which deletes all the books present in the respective section if the section is deleted i.e., Section Table would be a parent and Book would be a child.

Features and Workflow:

Vue js has been used as a fronted and Python Flask for backend jobs, now that when we turn on the localhost server and use the router link to navigate through the links which are different and classified for both admin and user, fetch the information and render the components. For the task scheduling we use the Celery-Redis servers for exporting csv and use the analysis for monthly analysis as it is for August 2024 we provide only one file for that month.

Then we use the caching for the frequently used routes for better performance.

API Design:

/api/resource/admin to fetch the books available in the library to display it in both admin and User Dashboard and can be even used to upload the books using POST to the library

/user-login/ for the login and use the token to validate the in the frontend

/profile to view the profile for the current user

Video link : Folder link for videos there are two videos one for LMS application and one for the celery beat scheduling and daily remainders

Folder - link - [Click here](#)