Library Management

Author

Jayanthi Nikhil Anand
21f1003041
21f1003041@ds.study.iitm.ac.in

I am currently pursuing BTech in Computer Science and Engineering from Anurag University, Hyderabad. I am also a data science and machine learning enthusiast.

Description

Library Management system is an e-book reading platform with user and admin login functionalities. Admin manages books, granting or rejecting user requests, revoking granted books, while the users request books, read reviews of the books and even return the books. The frontend employs Vue components and routing, while the backend utilizes Flask, SQLAlchemy, and Celery for asynchronous tasks. This includes monthly reports, daily reminders, and caching for optimization.

Technologies used

- Flask 2.2.2 - For app routing, template rendering, and redirecting.
- Flask-SQL Alchemy - For connecting the SQLite database as a server with the flask application.
- Celery [Redis] – For caching and scheduling asynchronous tasks.
- Flask_JWT_Extended – For Tokenization
- VueJS – For frontend tasks
- Bootstrap – For Styling

DataBase Schema

User

- id(Primary key)
- username(Unique)
- email(Unique)
- password
- lastAct

UserBook

- id(Primary key)
- user_id(FK-User)
- book_id(FK-Book)
- issue-date
- due_date
- return-date
- status

Section

- id(Primary key)
- name

Book

- id(Primary key)
- section_id(FK-Section)
- name
- content
- authors

GrantedBook

- id(Primary key)
- user_id(FK-User)
- book_id(FK-Book)
- granted_date

Feedback

- id(Primary key)
- user_id(FK-User)
- book_id(FK-Book)
- rating
- comment

API Design

API's are responsible for handling requests from admin for performing CRUD on sections and books, performing all the activities in the admin dashboard, handling users requests, etc. Here are the main elements implemented:

Authentication and Authorization

- The API uses JWT(JSON Web Tokens) for user authentication. It provides endpointsfor both user and admin login.
- The jwt_required() decorator is used to secure certain routes, ensuring that only authenticated users can access them.

User Functionality

The user related routes are listed below:

- "/signup"
- "/userlogin"
- "/user/home"
- "/user/books"
- "/user/request-book/<int:book_id>"

- /”feedback”
- “/book/<int:book_id>/reviews”
- “/user/search-books”
- “/user/mybooks/<int:book_id>”

Admin Fucntionaloity

The admin related routes are listed below:

- “/adminlogin”
- “/admin/home”
- “/admin/home/<int:section_id>”
- “/sections/<int:section_id>”
- “/sections/<int:section_id>/add_book”
- “/books/<int:book_id>/update”
- “/books/<int:book_id>/delete”
  Etc

Architecture and Features

The project folder named Library Management consists of all the required files. This folder has two subfolders, one for backend and the other for frontend. Backend has the models, controllers, config, mail_config, celery_config, app files, etc. While the frontend folder consists of all the components, App and main javascript files, etc.

Video

https://drive.google.com/file/d/1z7CIbkRcx3rG0IpnZYYb_Q5HNK3ONX-T/view?usp=sharing