PROJECT REPORT

MAD 1(JAN 2024)

by: Bhavesh Srihan
Roll no: 21f1004112

1. Introduction

This report outlines the development and implementation of a Library Management System, utilizing Flask for the backend and Vue for the frontend. Key features include JWT token-based authentication, search functionality, book rating, one-time purchase and download and pdf viewing capabilities. The system allows users to access and manage library resources.

2. Project Structure

Below is the representation of the project file structure for better understanding:-

```
LMS/
├── venv/
├── lms/
│   ├── __init__.py
│   ├── models.py
│   ├── api/
│   ├── templates/
│   └── …
├─- lms-ui/
│   ├── public/
│   ├── src/
│   └── ...
├── README.md
├── requirements.txt
├── app.py
└── ...
```

3. Database Structure

The application comprises separate database tables for users, authors, and sections, along with dedicated tables for books. Additionally, there are tables for transactions, where users pay for access to books for a specified duration, and for ratings, where users can rate books on a scale of 1 to 5. These tables are linked through association tables, facilitating many-to-many relationships between books and users or authors, and one-to-many relationships between books and sections.

Furthermore, a role table is implemented to enable role-based authentication. Users can assume one of three roles:

The USER role grants access to payment and book access functionalities.
The MANAGER role allows CRUD (Create, Read, Update, Delete) operations on sections, authors, and books.
At the highest hierarchy sits the ADMIN role, which provides access to metadata about the application's functioning. Administrators also possess the authority to grant or revoke managerial access and can block users from accessing the application.

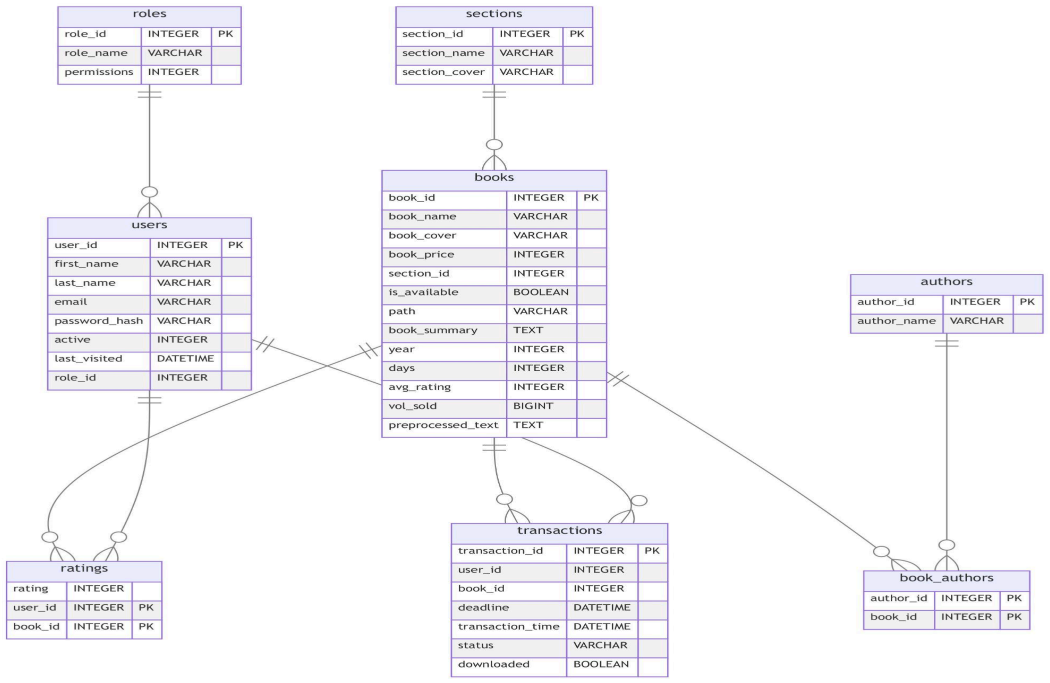

Fig. ER model for database.

4. Main Features

a. **RBAS**: The application incorporates a role-based authentication system utilizing JWT for user verification and registration. Additionally, it features automatic token refreshing using refresh tokens, with the implementation of the double-submit method for CSRF protection. Users are allowed to update their profiles seamlessly within the system.

b. **Search functionality**:A search feature has been integrated into the application, enabling users to find their desired books by searching for book names, summaries, authors, or the sections to which they belong. On backend, the search functionality utilizes fuzzy search techniques in conjunction with text lemmatization to enhance the quality of search results.

c. **Caching**:Redis has been integrated into the application to implement caching for various data sets, including book details, user details, and section-wise search details. The emphasis has been placed on maintaining data freshness in most scenarios, ensuring that users consistently access up-to-date information.

d. **Backend tasks** have been organized using Celery, with three periodic tasks designed:
    i. **Daily Reminders**: Users who haven't visited the application within the last 24 hours receive a daily reminder email. The email includes information about the number of days since their last visit, encouraging them to engage with the platform again.
    ii. **Monthly Report**: At the beginning of each month, an email is sent to every user, summarizing their activity on the site for the previous month. This report details the books purchased and the total amount spent during that period, providing users with insights into their monthly engagement.
    iii. **Removing Expired Transactions**: Every half-hour, the system automatically queries the database to identify and remove transactions with expired deadlines. Changes are also propagated to the caching mechanism to ensure that users are promptly informed about expired transactions, maintaining transparency and system accuracy.

e. **A PDF viewer** has been integrated into the application using pdf.js, enabling users to read PDF documents page by page. The viewer includes zoom-in functionality, allowing users to adjust the magnification level for better readability and navigation within the PDF files.

f. **Feedback System**: The application includes a rating functionality, allowing users to provide feedback on books they have accessed by assigning a rating out of 5. Users can share their opinions and experiences with the book, helping others make informed decisions. Additionally, the application displays both the average rating and the number of reads for each book, empowering users to make better choices based on community feedback and engagement metrics.

Video link: LMS Demo