PROJECT REPORT

MAD-2 PROJECT

IIT MADRAS BS DEGREE

Author

Name: Akshat Bhagotra

Roll Number: 21f3001511

Email: 21f3001511@ds.study.iitm.ac.in

I am a computer science student with a keen interest in machine learning. I have experience in Python, Java and SQL.

Introduction

The project aims to develop a web-based library management system using Flask, a Python web framework. The system is designed to provide users with an efficient way to manage their library activities. Users can register, log in, browse books, issue and return books, and make comments. Administrators have additional capabilities to manage sections and books. Additionally, the system sends email reminders for returning books and monthly reports for administrators. The project also incorporates caching and token-based authentication for enhanced performance and security. This report provides a comprehensive overview of the project's objectives, technologies used, architecture, database design, features implemented, and future enhancements planned.

Objectives

The primary objectives of this project are:

- To create a user-friendly interface for library users and administrators.
- To streamline the process of book issuance and returns.
- To provide features for user registration, login, and book browsing.
- To enable users to make comments on books.
- To allow administrators to manage library sections and books efficiently.
- To implement email reminders for book returns.
- To send monthly activity reports to administrators.
- To ensure the system is secure, scalable, and efficient.

Technologies Used

- **Framework**: Flask (Python web framework)

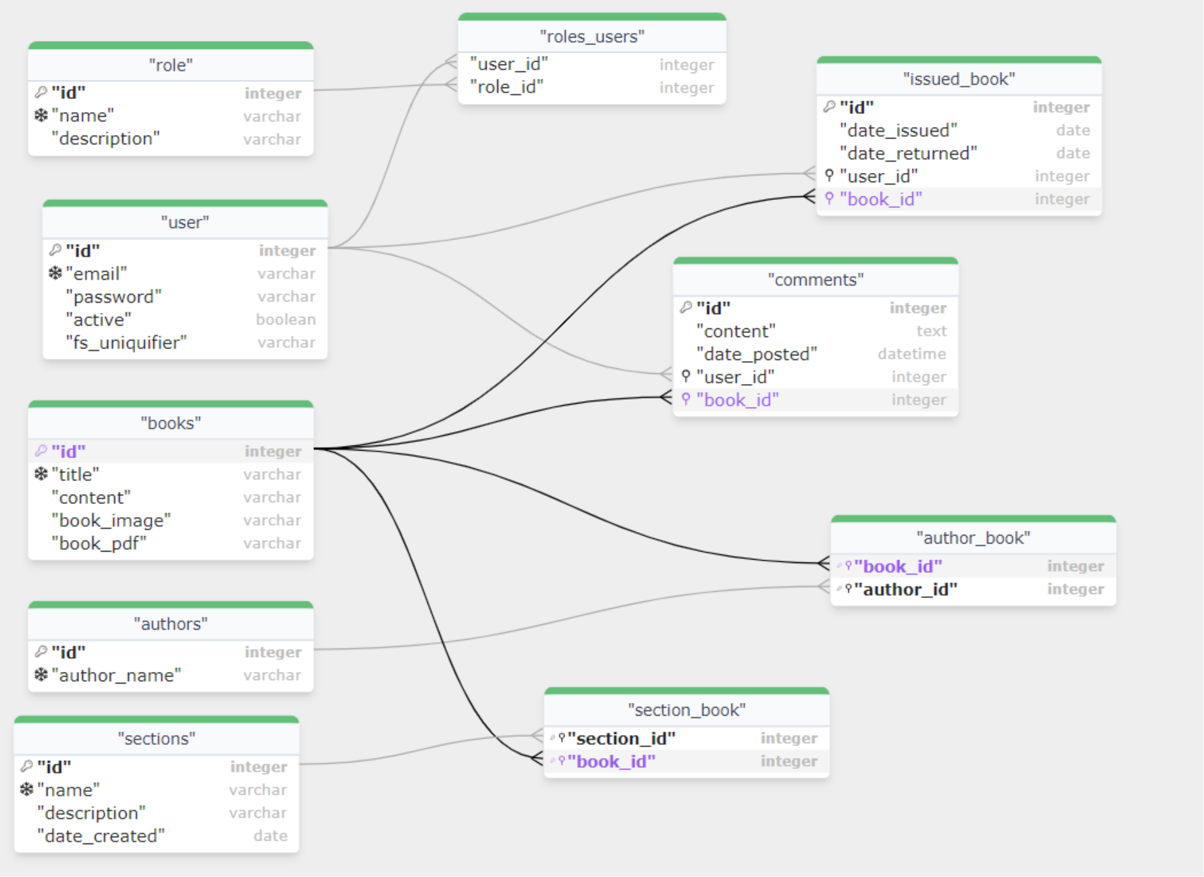
- **Frontend**: Vue.js for a dynamic user interface, Bootstrap for styling
- **Database**: SQLite for data storage
- **Authentication**: Token-based authentication for secure access
- **Caching**: Redis to enhance system performance
- **Email Service**: MailHog for email handling and testing
- **Task Management**: Celery for handling background tasks

Architecture

The system follows a modular architecture with separate components for user management, book handling, and administrative tasks. The backend is built with Flask, and the frontend is developed using Vue.js. The application uses RESTful APIs for communication between the frontend and backend.

Database Schema Design

The database is designed with a focus on efficiency and scalability, incorporating tables for users, books, sections, and issued books. Key relationships are established to ensure data integrity and support various functionalities.



Features Implemented

- **User Management**: Registration, login, and profile management
- **Book Management**: Browsing, issuing, and returning books
- **Comments**: Users can leave feedback on books
- **Administrative Controls**: Manage book sections and handle user requests
- **Background Tasks**: Scheduled jobs for sending email reminders and monthly reports
- **Caching**: Improved system performance with caching strategies
- **Token-Based Authentication**: Secure user authentication

Future Enhancements

- **Fine Management:** Implementing a fine system for late book returns and integrating payment gateways for online transactions.

Video Demonstration

https://drive.google.com/file/d/1kKLY_zSj14B9iZdO_wQPxCyHW_t-lB3u/view?usp=sharing