PROJECT DOCUMENTATION

Author

Name: Prem Kumar
Roll No.: 21f1000531
Email: 21f1000531@ds.study.iitm.ac.in

Description

Introducing E-Library, a dynamic digital platform designed for book enthusiasts and knowledge seekers. With E-Library, you can explore a vast collection of books, delve into various genres, and immerse yourself in captivating stories and informative content. The app offers easy access to e-books, audiobooks, and other digital resources, ensuring a convenient and enjoyable reading experience.

E-Library simplifies the discovery of new books and authors, enabling you to create personalised reading lists based on your interests and preferences. Whether you prefer fiction, non-fiction, academic texts, or self-help books, E-Library caters to diverse reading tastes.

Technologies Used

The E-Library app is built on Flask, a strong web framework for backend development. It uses Flask-Security for user authentication and Flask-SQLAlchemy for managing the database efficiently.

On the frontend, Vue.js is used along with Bootstrap for a visually appealing design, Vue-router ensures smooth navigation, while Pinia handles the app's state effectively.

For better performance, the app uses Redis for caching, supported by Flask-Caching. Email sending is managed by flask_mail, and flask_cors handles cross-origin resource sharing. HTTP requests are managed with the 'requests' library.

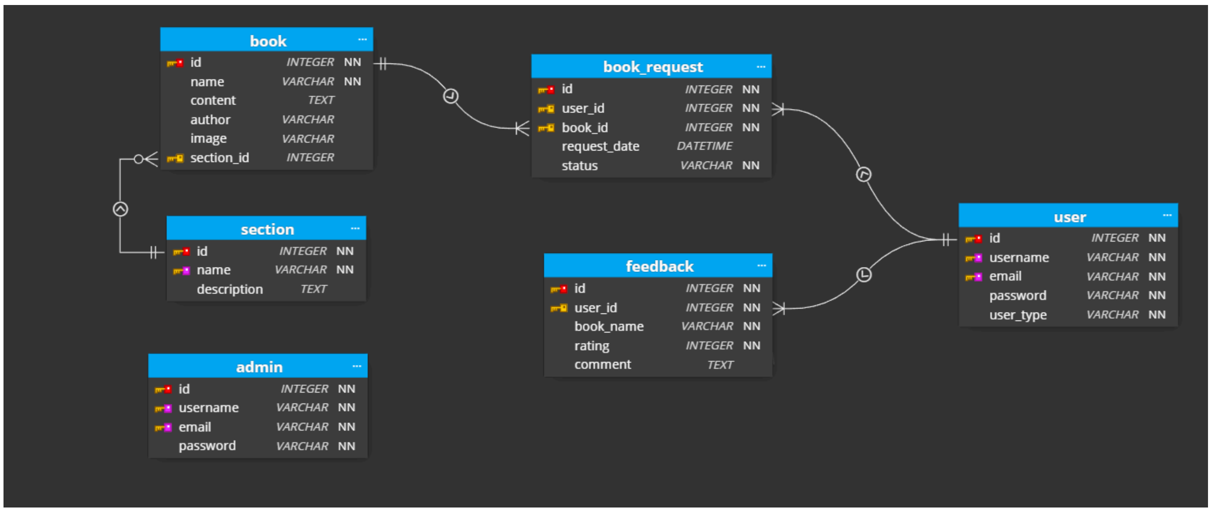
Additionally, Redis and Celery work together for batch job handling, making task management efficient within the app.

API Design

Our E-Library app's API architecture, built on Flask with blueprints for streamlined functionality, comprises multiple endpoints catering to various aspects. The Book Data API facilitates book operations like creation, editing, and searching, ensuring a comprehensive book management system. The User Data API handles user tasks such as authentication, registration, and role modification, ensuring smooth user interactions. Additionally, the Section API manages book sections for organised navigation, and the Section API allows users to create and manage personalised sections. Leveraging Flask-RESTful and

Flask-SQLAlchemy, our API guarantees efficient data handling and a user-friendly experience within the E-Library app..

db Schema Design
The database schema for the E-Library app includes tables for User, Book, Section, BookRequest, and Feedback. Users have essential details like username, email, and role. Books contain information such as name, content, author, and image. Sections categorise books for easier navigation. BookRequest tracks user requests for books. Feedback stores user ratings and comments about books. This schema, powered by Flask-SQLAlchemy, efficiently manages user interactions and book data in E-Library applications.



Architecture and Features

In the root folder we have the app.py file, models.py, resources.py file,a readme.md file, a requirements.txt file, the instance folder with the SQLite database file, the component folder which contains the vue templates and the static folder with images file etc.
1. app.py **:** This file acts as a controller for the app and has all the routes to different pages and contains a Python code for defining database models using SQLAlchemy and Flask-Login for the E-Library web application. All the api endpoints have been created and managed here. For Sending a Monthly Report using Celery.
2. workers.py : For managing Cache using redis.
There's a registration form for users with proper front-end and back-end validation. There's a login form for users and Admin. The Search bar on top helps the user and admin to search for books and sections. The app has CRUD features for Books & Sections Admin can edit/delete their Books/Sections. Books & Sections can be deleted by the admin.

Presentation Video : [App Presentation Video]