Library Web Application v2

A Project for Modern Application Development 2

Author:

Chelakarri Anantha Sai Srikar

Roll number: 21f1004284

Email: 21f1004284@ds.study.iitm.ac.in

App Description:

This application has one librarian and general users who can register and login to read books. The librarian can upload books, create sections and edit/delete books and sections. General users first have to register and then after logging in to the application, all the available books and sections will be shown, from which user can request a desired book of their choice and they will be accessed with the book's content only when the librarian approves their request. The application also sends automated reminders and reports to users based on their activity.

Technologies used for the project:

The backend of the application is primarily built using the Flask framework in Python, while the frontend is developed with the Vue.js framework. Caching is done using Redis and backend tasks are done by Celery. SQLite3 serves as the database solution.

Additional technologies integrated into the project include Flask-RESTful, Flask-SQLAlchemy, Flask-JWT-Extended, Werkzeug, Flask-CORS, Flask-Mail, Pillow, and Flask-Caching, among others.

Database Schema Design:

User model: Represents users with fields for user ID, username, password, role, and login time. It includes relationships to BookRequest, Feedback, and ArchivedBookRequest models.

Book model: Represents books with details such as book ID, name, file path, author, description, image, and date. It establishes many-to-many relationships with Section through book_section_association and relates to BookRequest, Feedback, and ArchivedBookRequest.

BookRequest model: Represents user requests for books, tracking the request status, dates, and linking users and books.

Section model: Represents sections with fields for section ID, name, description, image, and date. It has a many-to-many relationship with Book through book_section_association.

Feedback model: Represents user feedback on books, including feedback text, date, and relationships to users and books.

ArchivedBookRequest model: Represents archived book requests with similar fields to BookRequest and relationships to users and books, indicating past requests.

Architecture and Features:

This project employs Vue.js for the front-end, utilizing its templates to request and retrieve data from the backend Flask application as needed. The application contains one librarian, who manages the books, sections and requests from users. The librarian can create and delete books, sections and edit them too. When a user requests a book, the request is sent to the librarian and he can grant or reject the request. On granting the request, the user will be able to see the content of the book. The general user of this application is only allowed to request a maximum of 5 books at a time. A user can access a book for 7 days. After 7 days, the book will be automatically revoked from them. There is a search functionality, where users and librarian can search the library and get their desired books and sections. Users can view their books in “My Books” section. Login reminders will be automatically sent to those users who haven’t logged in to the application in the last 24 hours and Monthly reading reports are sent to the users on 10th of every month. Caching is done to improve the performance and efficiency of the application by temporarily storing frequently accessed data. The schedule jobs, daily reminder jobs and the automatic revoke of the books is handled by Celery, which runs on a Redis server. All the frontend resides in the static folder while app.py file handles the backend.

Video Demo Link:

Link to the video