Author : Achin Aggarwal
Roll number : 22f1001390

Student email : 22f1001390@ds.study.iitm.ac.in

About Me :
I am a passionate student pursuing computer science with a keen interest in software development.

Description :

The project entails developing a Library Management System (LMS) catering to two user roles: librarian and student. Librarians can manage sections and books, including creation, editing, and deletion. Students can request to issue books, which librarians can approve or revoke. Additionally, users can return books.

Technologies Used :

- Flask: A lightweight web framework for Python, chosen for its simplicity and flexibility in building web applications.
- Flask-RESTful: Used to create RESTful APIs for smooth communication between the frontend and backend.
- Flask-SQLAlchemy: Integrated for database management, providing ORM support and simplifying database operations.
- Aniso8601, Blinker, Click, Werkzeug, and others: Necessary dependencies for various functionalities within the Flask framework.

Database Schema Design :

- User : Stores user information including roll number, username, password hash, and role (librarian or not).
- Section : Represents sections in the library with attributes such as name, description, and creation date.
- Book : Contains details about books including name, section ID, content, author, issue date, and return date.
- Request : Records requests made by users to issue books.
- Confirmation : Stores confirmation details for issued books.
- Issue : Tracks issued books along with issue and return dates.
- Feedback : Captures user feedback on books, including comments and ratings.

API Design :

The API is designed to provide information about library sections.
- GetSection : Retrieves details of all library sections including ID, name, description, and creation date.

Architecture and Features :

The project follows a structured architecture with separate files for different functionalities.
- Files : Main components include app.py, routes.py, models.py, config.py, api.py, requirements.txt, and README.md.
- Templates : HTML templates are stored in the templates folder for rendering dynamic content.
- Instance Folder : Contains the SQLite database file.

Features Implemented :

- Robust login system for librarians and users.
- CRUD operations for librarians to manage sections and books.
- User dashboard displaying available sections and books after login.
- Book issuance and revocation functionalities for librarians.
- API implementation for displaying section details.
- Visualization using Chart.js depicting section vs book graphs.
- Limitation on issuing 5 books per user and automatic revocation after return dates.

Video :

Link :- https://drive.google.com/file/d/1ykfw2Kf6R8ieWSHo84pJsDqne_DUzUvd/view?usp=sharing