Name - Devansh Rungta
Roll Number - 23F1001518
E-Mail ID - 23f1001518@ds.study.iitm.ac.in

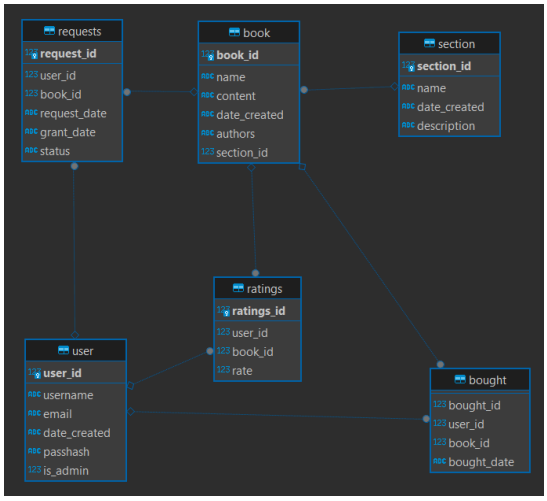
BookNest - Library Management System

The project involves developing a web-based Library Management System using Flask and SQLAlchemy. Key functionalities include user authentication, book browsing, requesting, and reading, along with admin tools for managing users, books, requests, and sections. The system aims to provide a seamless experience for users to access and interact with digital library resources while offering administrators control over system management and content distribution.

Technologies Used:

1. **Flask**: Flask is used as the web framework due to its simplicity and flexibility in building web applications and APIs in Python.
2. **Flask-RESTful**: This extension is utilized for building RESTful APIs with Flask, providing features like request parsing and serialization.
3. **Flask-SQLAlchemy**: Integration with SQLAlchemy allows seamless database management, including ORM capabilities for interacting with the database.
4. **FPDF**: FPDF is used for generating PDF files, enabling the application to create and serve PDF documents dynamically.
5. **Jinja2**: Jinja2 serves as the template engine for Flask, facilitating the rendering of HTML templates with dynamic content.
6. **SQLAlchemy**: SQLAlchemy is employed for database operations, offering a high-level ORM and SQL expression language.

DB Schema Design:



The design choices in the library management system project are driven by the need for efficient data management and user interaction. Utilizing Flask and SQLAlchemy, the system implements robust database models for users, books, sections, requests, ratings, and purchases. The system ensures smooth functionality and user access control by incorporating features

such as password hashing for security and default admin creation.

API Design:

The API design includes endpoints for managing sections and books within the library management system. Using Flask-RESTful, resources are created for sections and books, supporting GET and POST methods. Each resource retrieves data from the database and allows for the creation of new entries via JSON payloads. The API endpoints are registered with routes '/api/section' and '/api/book'.

Architecture:

1. 1. The “model.py” containing the database schema related definitions is divided into a
2. separate module from the “app.py” which contains all the views.
3. 2. Templates folder is used to serve the html files.
4. 3. Static folder contains the css.
5. 4. The “Instance” folder has the database defined.
6. 5. Venv contains the required python libraries used to build the application

Features:

1. **User Management**: Users can register, log in, and manage their accounts. Administrators have additional privileges, such as managing sections and books.
2. **Section Management**: Admins can create, update, and delete library sections, each representing a category or genre of books.
3. **Book Management**: Users can browse books by section, view book details, and search for specific titles. Admins can add, edit, and remove books from the library.
4. **Request Handling**: Users can request books for borrowing. Admins can approve or reject these requests, updating the status accordingly.
5. **Rating System**: Users can rate books, providing feedback for other users and helping to improve book recommendations.
6. **API Integration**: The project provides a RESTful API for accessing sections and books, allowing for seamless integration with other systems or applications.

Video Link:

https://drive.google.com/file/d/1Ce1GMrrUZgafYJYQ02S9v8DNuHkG6POa/view?usp=sharing

GitHub Repository Link:

https://github.com/devanshrungta/BookNest-LMS