Name: Priyanka Narula

Roll number: 21f3002839

Email id: 21f3002839@ds.study.iitm.ac.in

Description

This project primarily focuses on establishing a robust Online Library Management System. This system is designed to cater multiple types of users, with access granted on the basis of user type (general or admin). There are multiple genre's and each genre has a set of books available for the user to read. The admin can perform CRUD Operations on genre and books .

Technologies Used

- **Flask** as it helps me build web applications in Python, making it easier to handle user requests, manage routing, and create web pages.
- **Flask-SQLAlchemy** to work with databases in my Flask application.
- **Jinja2** as the templating engine to generate dynamic HTML content. It allows me to combine Python code with HTML templates, making it easier to display data from my application.
- **Bootstrap** for styling tables , forms and buttons .

DB Schema Design

- **User Table:**

  Contains user information such as *username, password, first name, last name and type*. Each user has a *unique ID* and can be associated with book requests.

- **Genre Table:**

  Represents different genre of books available in the library. Each genre has a *unique ID and name.*

- **Books Table**:

  Stores information about books available in the library. Includes details like *book name*, *description, authors, date added,* and the genre it belongs to. Each book is associated with a specific genre.

- **Issue Table:**

  Tracks requests made by users to borrow books. Includes *user name*, *book name*, *request date*, *return date*, and *request status*. Users can request books, and their requests are stored here.

- **Request Table:**

  Records book issued to users, indicating which books are approved by admin.Links *user names, book names, authors, issue dates*, and related book requests.

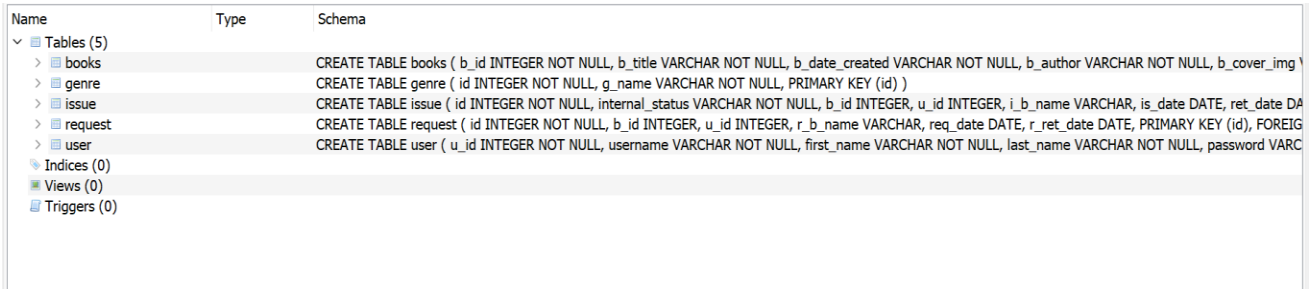
Architecture and Features

app.py file contains the setup for my Flask application, including creating the Flask app object, setting up the database connection, and importing necessary files like routes.py

routes.py file is responsible for defining the routes (URL endpoints) of my application that users can access. It contains the logic for handling HTTP requests and returning appropriate responses, such as rendering HTML templates, processing form data, and interacting with the database through models.

models.py file defines the database models for the application using Flask-SQLAlchemy. It contains Python classes that represent tables in your database, including fields and relationships between tables.

ER Diagram



Basic Routes for Users and Admin:

- Register, login, logout and profile routes to manage their accounts and authentication.

CRUD Operations for Sections and Books:

- Routes for admins to create, read, update, and delete genre and books in the library.

Admin-Specific Routes:

- Routes for admins to view all books, sections, book requests, issue history, and user feedback..
- Routes to accept/deny book requests and revoke book access for users.

User-Specific Routes:

- User actions include requesting books, viewing book issue history, and managing currently issued books.
- Users can also read a book once approved and return books **.**

Search Functionality:

- Users can search based on:
  - Section name
  - Book name
  - Book authors

Video

https://drive.google.com/file/d/1zUFWcv6Ns5uSYVjxKnUy8Pjs1yRRr6fy/view?usp=drive_link