LMS Application Project Report

Author
Name: Ayushi Saxena
Roll No.: 21f1006327
Email: 21f1006327@ds.study.iitm.ac.in
I am a Diploma level student of the online B.S. Data Science and Applications course of IIT Madras.

Description

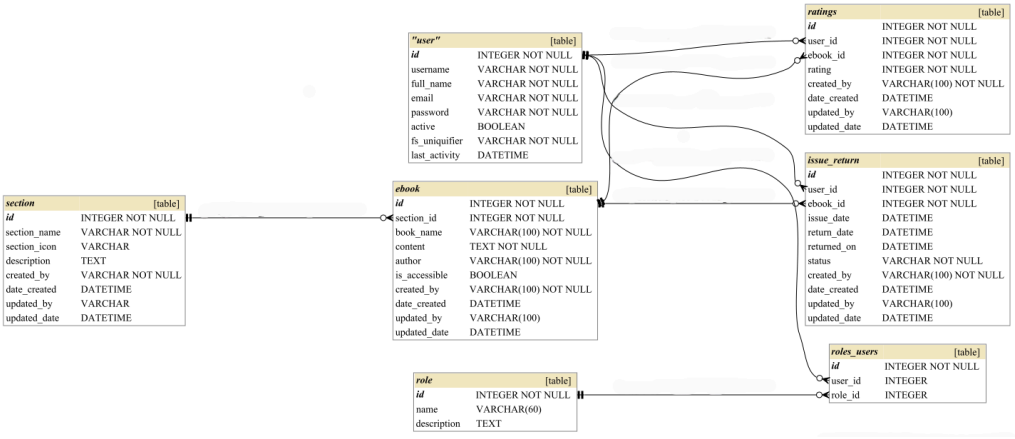
The project requires developing a digital Library Management System using Flask and Vue frameworks, along with SQLAlchemy as ORM and SQLite database. The application provides an interface for a Librarian to create and manage library sections and E-books, and track issues and return requests.It provides basic functionalities for user management such as registration, login and logout, as it is a multi-user platform. Along with that, the application can be used by users to issue, read and rate books. There are features for daily reminders and monthly reports to monitor activity.
Approach: After the necessary installations, the models were designed and database instance was created. Then the authentication system was put in place, succeeded by librarian functionalities and user functionalities. Once done, the dashboard was designed. Then the backend jobs were scheduled.

Frameworks and Libraries used

- Flask microframework for application development
- Sqlite database and Flask-SQLAlchemy as the ORM
- Flask-Security for token-based authentication
- VueJS, Bootstrap, and CSS for UI design
- ChartJS for Dashboard Statistics
- werkzeug.security to hash passwords and check hashed passwords
- smtplib, MIMEMultipart and MailHog for sending email
- Redis and Celery for scheduled jobs and caching
- Google Chat Webhook to send daily reminders

DB Schema Design

There are 7 tables in the database named library.db:-



The user table is used to store the details and login credentials of users. There are two primary roles on which access is controlled - librarian and general_user. The role table is created to define the roles and the table roles_users links the roles to the users. The section and ebook tables store the section and E-book details respectively. Each section can have many E-books, so a one-to-many relationship is established between section and ebook table, using id of section as a foreign key. E-books can be rated by users, so the ratings table has many-to-many relationships with user and ebook table. The issue_return table tracks all the requests made by users and their status. This table also has many-to-many relationships with user and ebook table to track which book request was made by which user. Audit fields have been included in all tables to track the timings of changes and who made those changes.

Architecture and Features

The app follows MVC architecture. The project folder has a file 'main.py' which has the code to run the application. The folder application contains all the python backend modules, including the models, resources, endpoints, tasks and worker. The templates folder contains an html file which is served when the application is run. The static folder contains a css file, vue components and router. The instance folder stores the database. There are configuration files for the app and celery in the root folder.

Features:-

- Responsive UI
- Common login form for librarian and users. Registration form for users.
- Token based authentication and RBAC.
- Reset password after OTP verification through email.
- Librarian features:-
    - Add, edit or delete Sections and E-Books.
    - Track user requests. Grant or reject requests.
    - Auto-revoke book access after 7 days of issuing.
    - View Issue History and download csv report for the same.
    - Monthly Activity Report on Librarian's email.
    - Dashboard with charts to simplify statistics.
- User features:-
    - Issue, read and return E-books.
    - Search and rate E-books.
    - Edit Profile.
    - Daily reminders on G-Chat to visit or return book
    - Dashboard with charts to simplify statistics.

API Resource Endpoints

Flask RESTful has been used to create a Section resource with endpoints to get and post section details.

Presentation Video Link

https://drive.google.com/file/d/1PU2XGKxrs_MWyxBE_j-3G5KshjsFLeL8/view?usp=sharing