Library Management Systems
(AppDev2- Project)

Name : Manan Tilwani
Roll No. : 21f2000858
Student Email : 21f2000858@ds.study.iitm.ac.in

Description:

This app is a multi-user web app for borrowing and reading books.
To create this app, I followed the app Wireframe and the guidelines provided by the appdev2 project statement and implemented the following steps:-

- Creating the database schema of the app and tables using flask-sqlalchemy.
- Creating the flask app instance and html template pages using css and bootstrap
- Creating all the required routes to link the app to the database. Implemented the role based access control (Rbac) using flask security and provided token based authentication for the roles assigned.
- Applied CSS styling to the web pages.
- At last I implemented backend celery jobs.

Frameworks Used:

- Vue.js - The client side/ frontend part of the app is built on Vue.js.
- Flask - The server side/ backend part of the app is built on Flask.
- Redis and Celery are used for scheduled jobs/daily reminders via MailHog.
- Flask security for token based authentication.
- Smtplib and MIMEMultipart to send multipart messages using simple mail transfer protocol.
- Flask - this web application is built on flask.
- Jinja2 - for generating Monthly activity reports at backend
- Bootstrap - for templates of the web pages.
- SQLite3 - to create the database structure for the app.
- Flask-SQLAlchemy - to create and manage the relational database for the app.
- Matplotlib - to plot the book count in each section graph for the librarian dashboard.

Database:

- Database models for the app are created using flask-sqlalchemy.
- There are mainly 7 Tables used in the database: User,Book,Section,BookRequestIssueHistory,Role,Association,user_roles
- Book and Section have many to one relationship.
- Association table have many to many relationship with User and Book.
- User and Book table have one to many relationship between them.
- Users are differentiated based on their roles using the UserRoles table.

System Design:

- This web app follows MVC architecture style:-
    - Model(M) is handled by flask. Flask interacts with the database and manages the data model.
    - View(V) is handled by vue.js. Vue components are responsible for interactive user interface.
    - Controller(C) is handled by flask. Flask routes handle all the business logic at the backend.
- Instance Folder - stores the database of the app.
- Static Folder - stores all the graphs and image files and also it has subfolder pages which store all the Vuejs frontend pages.
- Main.py - the code for the flask app instance, celery app instance and initializing database for the app.
- Sample_data.py - the code of some pre saved data of the app.
- Models.py - the code for creating database tables.
- Worker.py, Celeryconfig.py, Task.py, mail_service.py - the code for celery configuration, scheduled jobs and daily reminders.
- Views.py - the code for all the backend routes and endpoints.

Features Implemented:

- Single login form for users and librarian
- Proper alert messages for the tasks performed.
- Librarian dashboard with app statistics on users,books,section,issued books,requested books and Section vs Number of books in section graph.
- Librarian can manage books,section and give access to book and revoke the access from users.
- Librarian can create,update and delete books as well as sections
- Librarian can manually revoke access of books whose due date has passed by going to particular route which tells the status of the issued books.
- User can search for books based on their name/author and search for the section name.
- Users can request for the books, read content of approved books,can rate the books.
- Users can at max at any point of time can request for 5 books.
- Monthly Activity report for admin is sent to admin's email on first day of month.
- Daily reminder mails sent to users to visit the app if inactive for 24 hours.

To run the app:

- Run the app.py file.

Presentation Video Link:

https://drive.google.com/file/d/1lj6NMobfBq1rjz0tM-xOWY5BtLdYb3cC/view?usp=sharing