Modern Application Development II

Project Report
May 2024

LIBRARY MANAGEMENT SYSTEM V2

App Name: GWonder

Guddu Kumar Mishra
Roll : 22f3001315

Library Management System V2 Overview :

Description: This App is second version of my LMS project. The app is built on the Vue.js Framework for the clint side and Flask for the server side. Additional features include scheduled jobs and daily reminders using celery and token-based Authentication using flask security.

Frameworks and Technologies :

- 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 Google Chat and MailHog.
- Flask restful for managing the api calls.
- Flask security for token based authentication.
- Smtplib and MIMEMultipart to send multipart messages using simple mail transfer protocol.
- SQLite3 and Flask-SQLAlchemy - to create and manage the relational database for theapp.
- Matplotlib - to plot the app statistics graphs for the admin dashboard.
- Bootstrap - for templates of the web pages.
- Jinja2 - for generating Monthly activity reports at backend

Procedures: *First I created database schemas of the app,then a proper login system then developed the apis,logic,user-interface.At last i implemented the celery jobs.*

Future Improvements: Make it look more aesthetic , database enhancement, strengthening backend validation , implement premium features . and also there is plenty of room for backend part like handling more celery jobs etc.

GWonder Details :

" Create your own dreamy world and dive into a journey full of romance , thriller, fantasy and knowledge."

Data Model:

• The data model, implemented in SQLite using SQLAlchemy, comprises relational tables: RolesUsers , Role,User, Section , Book, BookUser, BookRequest, Feedback, and DailyVisits.

• Relationships are established through foreign keys and one-to-many, many-to-many associations, ensuring data integrity and consistency.

- Users are differentiated based on their roles using the RolesUsers table.

• This relational model facilitates efficient management of users, books, feedbacks, requests, and sections within the system.

Presentation Layer:

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.

- Requirements.txt, Imp_commands - store required dependencies.

Presentation Video Link: ***CLICK HERE***