Author
Rajat Srivastava
22f3003195
22f3003195@ds.study.iitm.ac.in
I enrolled in this course in September 2022 and I am currently at diploma level. I want to present you this project report represents the culmination of my efforts and learning experiences throughout the Mad-2 course.

Description

I had to develop a Library Management web application by using HTML,CSS and VueJs for frontend and Flask for backend.Here I created a separate login for Librarian and user and new user can also register.Librarian can create,update and delete Sections as well as Ebooks.User can request access for an ebook and can see contents of ebook once access granted by librarian.Users would receive daily reminders to visit app as well as a monthly report through mail which I implemented by using Celery.

Technologies Used

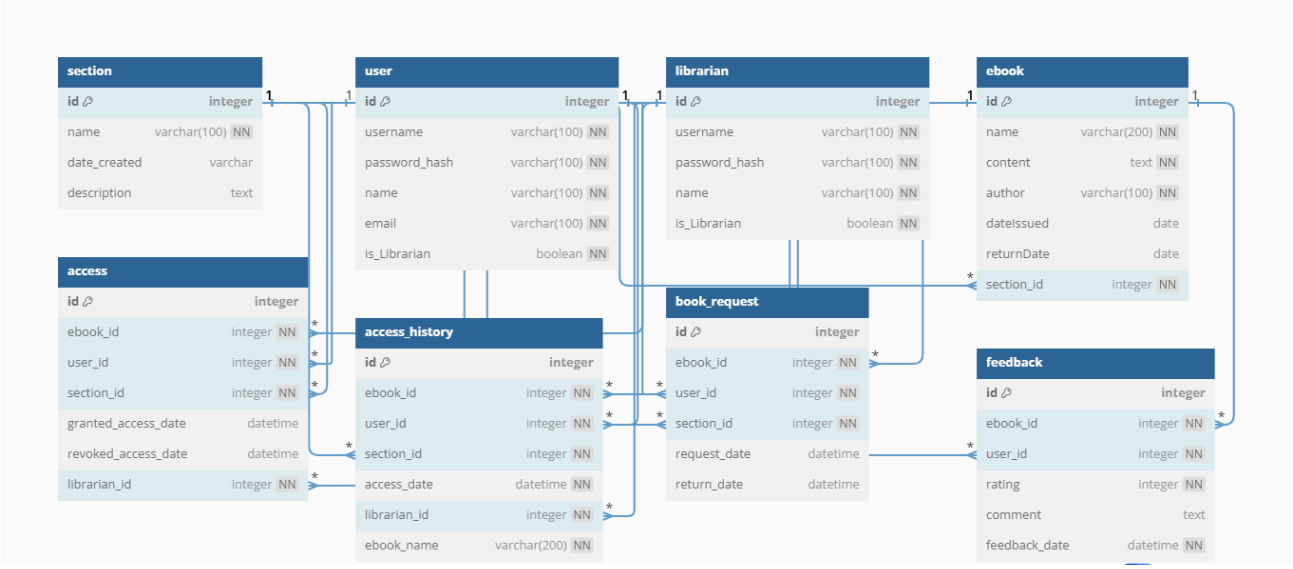
Flask: A Python framework for building web applications.
SQLAlchemy: An Object-Relational Mapping (ORM) library for Python, used it for interacting with database.
SQLite: A relational database management system.I used it as a database because it is a very powerful tool for small and medium scale applications.
 Flask-CORS: A Flask extension for handling Cross-Origin Resource Sharing (CORS) headers.I used it for securely handling cross origin requests.
Flask-RESTful: An extension for creating RESTful APIs in Flask.Created APIs using this.
Werkzeug: A utility library for handling authentication and password hashing.
Python: The programming language used for building the application.
HTML/CSS: For front-end to make web pages.
Vue.js: Create dynamic and interactive web applications with its intuitive and versatile framework.
Celery: It streamlines asynchronous task execution in Python, making distributed computing and task scheduling a breeze for developers.

API Design

The Flask API provides endpoints for user authentication, section and ebook management, access requests, feedback submission, and more, with token-based authentication and JSON data serialization.

Video Link
https://www.youtube.com/watch?v=5JD8b6JCnF4

DB Schema Design



In the database schema, several one-to-many relationships exist. Users can have multiple access instances, each tied to a specific user. Librarians can grant access, forming a one-to-many relationship with users. Ebooks are accessible by multiple users, with each access instance linked to a specific ebook. Ebooks can also receive multiple book requests, each associated with a specific ebook. Users can make several book requests, forming a one-to-many relationship. Sections contain multiple ebooks, with each ebook linked to a specific section. Users provide feedback on multiple ebooks, and each feedback instance relates to a specific user and ebook.

Architecture and Features

In the project folder there is a static folder and a template folder, inside static folder there are all js files for frontend and and in template folder there is index.html. Then there i app.py which has all the api endpoints as well as app configurations. In “models.py” file all the database models are described using Flask-SQLAlchemy.Celeryconfig.py contains celery configurations and worker.py contains worker code of celery.In mail_services.py there are configurations of SMTP and tasks.py contain tasks that are scheduled using celery.

Features include that an existing user can login and a new user can register.User can request access for a book.Librarian can grant or deny access and also can revoke access after granting.Users can search for a particular book from it’s name or author’s name also for all books of the section. Users would receive daily reminder through email for visiting app as well a monthly activity report.