Author

Samadhan Kashinath Patil
21f1006852
21f1006852@ds.study.iitm.ac.in

Description:

A Library Management System Web Application where Sections and Books can be created. And Books can be issued by the user.

Technologies Used:

- Flask: Provides the core web framework for building and managing the application.
- Flask-CORS: Allows to control and handle cross-origin requests from different domains.
- Flask-JWT-Extended: Ensures secure authentication and authorization for users.
- Flask-Caching: Improves performance by caching frequently used data.
- Flask-RESTful: Simplifies the creation of RESTful APIs and resource management.
- Flask-SQLAlchemy: Integrates SQLAlchemy for working with databases and data models.
- Celery: Manages asynchronous tasks to improve user experience and handle background operations.
- Redis: Provides caching and task management capabilities.

API Design:

User Management API: Login, Logout, register feature

Section Management API: CRUD implementation of sections.

Book Management API: CRUD implementation of books.

Book issuing API: Book issue, Book approve

Reporting API: Generate report in CSV File for Admin

Cache Management API

Celery and Asynchronous Task API

DB Schema Design

There are 5 tables, User, Sections, Books, BookCart and CartItems. The whole schema can be seen here https://drive.google.com/file/d/1VMntwsAeT9wqyx1HgJIR2RqAYcskKMst/view?usp=sharing (Click on the Link)

Architecture and Features

\backend: models.py app.py, `requirements.txt`, and configuration files
\backend\tools: task.py, mailer.py, worker.py

\backend\templates: HTML Templates for Mailing

\backend\instances: Contains the database.

\frontend: Contains Vue templates and JavaScript files.

\frontend\public: index.html, logo.png

\frontend\src: App.vue, main.js, routers.js

\frontend\src\components: Vue Components

\frontend\src\mixins: UserMixin.js

\frontend\src\styles: global.css

Features Implemented:

Default features include user registration, login/logout. Seamless section and book management are facilitated through APIs for creating, updating, and retrieving books belongs to particular section. The book issuing system sends a requests to the admin and once the admin approve it and sends issuing confirmations via email. Automated tasks, such as daily reminders and monthly reports, are efficiently handled using Celery for asynchronous processing. Additional feature such as monthly report for admin in the form of csv is available on one click. The project's caching mechanism optimises performance by storing frequently accessed data in memory, contributing to an overall smooth and responsive user interface

Video

https://drive.google.com/file/d/1JzAUWcz8jLodINRS_DC2NyvyY4-0HqAo/view?usp=sharing