Project Report: Flask + Vue.js Library Management System

Maanav A – 21F1004840

1. Overview

This project is a full-stack Library Management System using Flask for the backend and Vue.js for the frontend. It includes user authentication, book borrowing/returning, and overdue book handling. Celery with Redis is used for background tasks like automatically returning overdue books.

2. Project Structure

Backend (Flask):

- API for user management, book borrowing/return, and overdue book handling.

- SQLAlchemy for database management (MySQL/PostgreSQL).

- Celery for background tasks, Redis as the message broker.

Frontend (Vue.js):

- Vue.js for the user interface, including book lists, borrowing features, and account management.

- Axios for API interaction with the Flask backend.

3. Core Features

- User login and authentication (admin and regular users).

- Book borrowing (users can borrow up to 5 books at a time).

- Celery handles overdue book returns in the background.

- Admins can manage the library (add/remove books).

- Responsive frontend with real-time updates.

4. Celery Setup

1. Install Redis

Ensure Redis is installed and running locally or via a cloud provider. Update the Redis URL in `config.py`

2. Start Celery

Run Celery workers from the backend directory:

```
celery -A app.celery worker --loglevel=info
```

3. Run Celery Beat

Celery Beat handles periodic tasks:

```
celery -A app.celery beat --loglevel=info
```

6. Dependencies

Backend (Python):

- Flask, Flask-SQLAlchemy, Flask-Migrate, Celery, Redis, SQLAlchemy

Frontend (Node.js):

- Vue.js, Axios, Vue Router

7. Future Improvements

- Email reminders for overdue books.

- Advanced book search and filter.

- Admin dashboard with borrowing statistics and analytics.

Demo Video Link:
https://drive.google.com/file/d/1BbC3HM5b2bQhzFS49ruuRbrHsCXTnUF5/view?usp=sharing