Author

Anushka Yadav
21f1004806@ds.study.iitm.ac.in
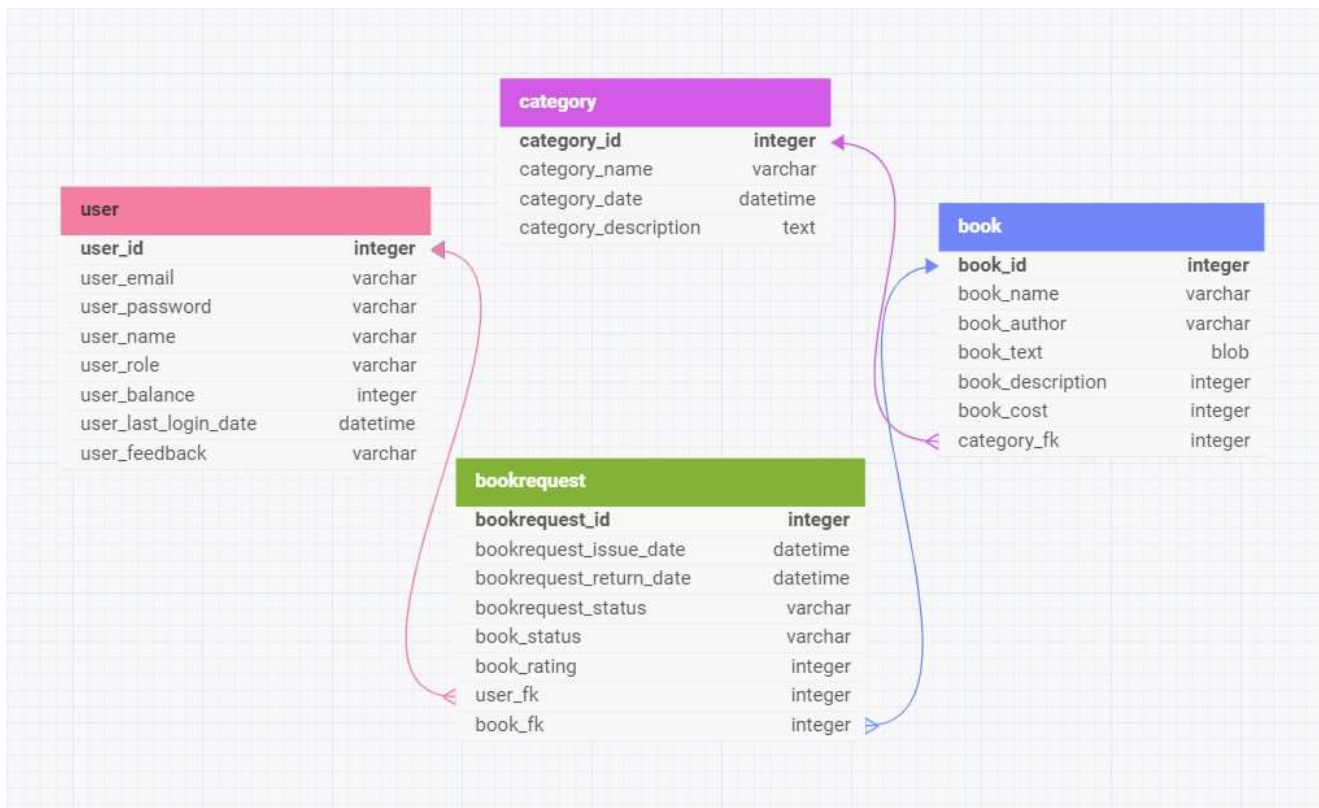This is Project Report on MAD2 Project Course.

Description

The Library Management System app enables users to browse and request e-books, while admin can manage section, books and, user access. It features user authentication, book requests, automated access control, feedback submission, and scheduled jobs. The system offers APIs for CRUD operation and dashboard creation.

Technologies used

1. Flask for API: Flask is used to create the backend API endpoints that handle requests and

2. Bootstrap: Used for styling.

3. SQLite for database SQLite manages data storage for Categories, Books, and Users and Book requests.

4. Redis for caching: Redis is used for caching frequently accessed data.

5. Redis and Celery for batch jobs: Redis and Celery handle background tasks and batch jobs, ensuring efficient processing.

DB Schema Design

Architecture and Features

1. CRUD operations are centralized in `api.py`, while REST controllers reside in `views.py`.

2. `mail_service.py`, `worker.py`, `tasks.py`, and `instances.py` handle caching and background tasks using Celery.

3. JWT authorization is managed in `auth.py`.

Features:

1. Admins can CRUD categories and books, search for books, authors, and categories, and monitor overall book status.

2. Admins can accept or reject book requests, and a statistics page displays request trends.

3. Users are limited to 5 pending book requests, and can view and return issued books

Scheduled Jobs

1. Monthly Activity Report: Automatically generated and sent on the first day of each month.

2. Daily Reminders: Sent to users inactive for over 1 day.

3. Access Revocation: Overdue books have access revoked automatically.

Additional Features

1. Content Purchase: Users can download books by paying the price

2. Dynamic Search: Search functionality for books, authors, and categories.

3. Styling and Aesthetic: UI designed for user experience and aesthetics.

4. Feedback System: Users can provide feedback on their experience.