Vivek Kumar Kamal
21f3002412

Library Management System Report

https://drive.google.com/file/d/1EaPHBbvfuzkJCV7m1PjFdRdHqUSGJlpd/view?usp=drive_link

Project Overview:
This Library Management System is a comprehensive digital solution designed to streamline the operations of a modern library. It facilitates the management of e-books, user accounts, book requests, and provides insightful statistics for librarians. The system incorporates features such as user authentication, e-book browsing, borrowing and returning processes, and feedback mechanisms, enhancing the overall library experience for both users and administrators.

Technologies Used:
1. Backend:
   - Flask (Python web framework)
   - SQLAlchemy (ORM for database operations)
   - Flask-JWT-Extended (for JWT-based authentication)
   - Celery (for task scheduling)
   - Redis (as message broker for Celery)

2. Frontend:
   - Vue.js (JavaScript framework for building user interfaces)
   - Axios (for making HTTP requests to the backend)

3. Database:
   - SQLite (lightweight relational database)

4. Additional Libraries:
   - Flask-CORS (for handling Cross-Origin Resource Sharing)
   - Passlib (for password hashing)
   - Python-dateutil (for date and time utilities)

Database Schema:
The database consists of the following main tables:

1. users:
   - id (Primary Key)
   - email (unique)
   - password_hash
   - name
   - is_librarian (Boolean)
   - created_at
   - is_active (Boolean)

2. ebooks:
   - id (Primary Key)
   - title (unique)
   - authors
   - content
   - created_at
   - rating (average rating)
   - total_ratings
   - total_borrows

3. requests:
   - id (Primary Key)
   - status (pending, granted, revoked)
   - request_date
   - issue_date
   - return_date
   - due_date
   - user_id (Foreign Key to users)
   - ebook_id (Foreign Key to ebooks)

4. read_books:
   - id (Primary Key)
   - read_date
   - user_id (Foreign Key to users)
   - ebook_id (Foreign Key to ebooks)
   - request_id (Foreign Key to requests)

5. sections:
   - id (Primary Key)
   - name (unique)
   - description
   - created_at

6. book_section (Association table):
   - book_id (Foreign Key to ebooks)
   - section_id (Foreign Key to sections)

API Endpoints:

1. Authentication:
   - POST /auth/register: Register a new user
   - POST /auth/login: User login
   - POST /auth/logout: User logout
   - POST /auth/refresh: Refresh JWT token

2. E-books:
   - GET /api/ebooks: List all e-books

- GET /api/ebooks/<ebook_id>: Get details of a specific e-book
- POST /api/ebooks: Add a new e-book (librarian only)
- PUT /api/ebooks/<ebook_id>: Update an e-book (librarian only)
- DELETE /api/ebooks/<ebook_id>: Delete an e-book (librarian only)

3. Sections:
   - GET /api/sections: List all sections
   - GET /api/sections/<section_id>: Get details of a specific section
   - POST /api/sections: Add a new section (librarian only)
   - PUT /api/sections/<section_id>: Update a section (librarian only)
   - DELETE /api/sections/<section_id>: Delete a section (librarian only)

4. Requests:
   - GET /api/requests: List all requests (librarian only)
   - GET /api/requests/<request_id>: Get details of a specific request
   - POST /api/requests: Create a new book request
   - PUT /api/requests/<request_id>: Update request status (librarian only)

5. User-specific:
   - GET /api/user/requests: Get user's book requests
   - GET /api/returned-books: Get user's returned books
   - POST /api/return-book/<request_id>: Return a book

6. Feedback:
   - POST /api/feedback: Submit feedback for a book
   - GET /api/user/feedbacks: Get user's feedbacks

7. Statistics:
   - GET /api/statistics: Get library statistics (librarian only)

8. Search:
   - GET /api/search?q=<query>: Search for e-books

Key Features:

1. User Authentication: Secure JWT-based authentication system.
2. Role-based Access Control: Separate interfaces and permissions for regular users and librarians.
3. E-book Management: Add, update, and delete e-books (librarian only).
4. Book Borrowing System: Users can request, borrow, and return e-books.
5. Feedback System: Users can rate and provide feedback on books they've read.
6. Search Functionality: Users can search for books by title, author, or section.
7. Statistics Dashboard: Librarians can view important statistics about library usage.
8. Automated Reminders: Celery tasks send daily reminders for due books and monthly activity reports.