

Name: Harsh Kumar Upadhyay

Roll Number: 21f1002332

e-Mail ID: Harsh

Library Management System - V2

Multi-user app to issue, grant/revoke and maintain e-books across sections like a library.

Frontend: <http://localhost:8080/>

Mails: <http://localhost:8025/>

Frameworks used

SQLite for data storage

VueJS for UI

Bootstrap

Redis for caching

Flask for API

Jinja2 templates

SQLite for database

Redis and Celery for batch jobs

Roles

The platform will have **two** roles;

1. librarian - the maintainer of sections and e-books in the library

An librarian can monitor all the users, see all the statistics

A librarian will create sections, e-books in the library.

A librarian can grant/revoke access for an e-book.

2. General user - an individual who wants to access an e-book from the library

A user can go through all the available sections and e-books in the library and can request for an e-book.

A user can request for maximum of 5 e-books at a time

A user can update their profile page, rate an e-book.

Terminologies

Section : All the e-books in the library can be classified in a section. For e.g. Philosophy, science, arts etc.

E-book : An e-book in a digital format of the content that can be viewed and rated and is accessible on any device that runs the web.

Database Model:

1. Table 'roles_users'

id
user_id
Role_id

2. Table 'role'

'Id'
'name'
'description'
'Permissions'

3. Table 'user'

'id'
'email'
'username'
'password'
'last_login_at'
'current_login_at'
'last_login_ip'
'current_login_ip'
'login_count'
'active'
'fs_uniquifier'
'granted'
'revoked'

4. Table 'section'

'id'
'name'
'description'
'created_at'
'updated_at'
'Books'

5. Table 'book'

'id'
'name'
'authors'
'description'
'content'
'created_at'
'updated_at'
'section_id'
'section_name'
'user_id'
'status'
'username'
'user_email'
'issued_at'
'like'
'return_at'
'requested'
'returned'

Core Functionalities

1. Admin login and user login (RBAC)

Login/Register form

Only one librarian identified by its role.

Flask security based authentication for role based access control

Suitable model to store and differentiate all the types of user of the app.

2. Librarian Dashboard - for the librarian

The librarian added automatically whenever a new database is created

Librarian dashboard which displays all the relevant statistics of the application, e.g. active users, grant requests, e-books issued, revoked etc.

3. Section Management - for the librarian

Create a new section to add ebooks from a genre.

Update an existing section

Delete an existing section

4. e-book Management - for the librarian

Create e-books in a particular section

Edit an existing e-book

Delete an existing e-book.

5. Search for sections, e-books

Search for relevant e-books based on their section, authors, title etc.

6. General user functionalities

Ability to request for/return a particular e-book

Request for a maximum of 5 e-books at a time

A user can access a book for a specific period of time

Users can give feedback for an ebook.

Note: Can decide to implement any **ONE** of either automatic revoking of an e-book after a specific period of time or a **manual revoking of the e-book by the librarian.**

7. Librarian functionalities

Ability to view all the issue/return requests from all the users

Ability to grant/revoke a particular e-book to a user

A librarian should be able to monitor active users, grant requests, e-books issued, revoked etc.

8. Backend Jobs

Scheduled Job - Daily reminders - The application should send daily reminders to users on g-chat using Google Chat Webhooks or SMS or mail

Check if a user has not visited/has a return date approaching
If yes, then send the alert asking them to visit/return the book
The reminder can be sent in the evening, every day (students can choose the time)

Scheduled Job - Monthly Activity Report - Monthly report for the librarian created using HTML and sent via mail.

The activity report consists of sections & e-books issued, return dates, ratings received on sections/ebooks etc.

For monthly report to be sent, start a job on first day of every month
→ create a report using above parameters → send it as email

User Triggered Async Job - Export as CSV - CSV format details for the issued/returned/granted e-books done by the librarian.

This export is meant to download the Name, Content, Author(s), Date issued, Return date etc.

Have a dashboard from where the librarian can trigger the export
This should trigger a batch job, send an alert once done

9. Performance and Caching

Add caching where required to increase the performance

Add cache expiry

API Performance

Video Link:  **App Dev2 Project** (<https://youtu.be/p0H1SWhKpXg?si=SWRFTEqTBTH51T0F>)