MAD-2 PROJECT REPORT ON LIBRARY MANAGEMENT SYSTEM

NAME - **YASHWANT RAJ**
Roll No. - **21f1001547**
Email - 21f1001547@ds.study.iitm.ac.in
I am a diploma level student in IITM BS online degree program.

Title

Library Management System

Introduction

The Library Management System (LMS) is a comprehensive web application designed to streamline library operations and enhance user experience. It provides functionalities for both librarians and users, including secure login, book management, borrowing requests, and daily reminders for users. The system allows librarians to efficiently manage books, track user activities, and generate monthly reports. Users can easily browse available books, make borrowing requests, and receive notifications about their access. With a user-friendly interface and robust backend, the LMS aims to improve the overall efficiency of library management while providing a seamless experience for all users apart from allowing users to read books, provide feedback (through ratings or reviews) and check their statistics(eq : Books Read till now etc).

System Overview

The application features a backend built with Flask, handling API requests, user session management, and interactions with a SQLite database. The frontend, developed with Vue.js, provides an interactive interface where users can explore music, manage playlists, and access recommended tracks based on their preferences.

Data Models

User: Represents a user of the library, storing details such as username, email, password, role (e.g., user, librarian, admin), and the last visit timestamp. It provides methods for creating and retrieving user data.
Librarian: Inherits from the User model, specifically representing librarians in the system. It links the user account to the librarian profile and allows librarians to manage library operations.
Admin: Similar to the Librarian model, this represents admin users who have elevated privileges. It links to the User model, allowing admins to perform administrative tasks within the system.
Section: Represents different sections of the library, storing details like section name, creation date, and description. It provides a method to create new sections and convert section data to a dictionary format.
Book: Represents books in the library, including attributes like title, author, content, section affiliation, uploader, and rating. It offers methods for creating books and converting book data to a dictionary format.

BookRequest: Manages requests made by users to borrow books. It tracks user and book associations, request duration, status (pending, granted, revoked), and timestamps for requests. It includes methods for creating requests and managing access to books.
Review: Represents user reviews for books, containing information about the user, book, rating, and optional comments. This model provides a method to create reviews and convert review data to a dictionary format.

System Architecture

The system follows a client-server architecture. The client side is built using Vue.js providing an interactive user interface for the admin and users. On the server side, Flask is used to handle API requests and interactions with the database.

Functionality

Librarian : The librarian is the administrator of the website. The librarian can grant or reject a book requested by the user. Also the librarian can upload books of different sections on the website, monitor the statistics such as the no. of books uploaded in each section and the no. of books reviewed.

User : The user can check out the books uploaded on the website, search for a book by it's title, author, section name and rating. The user can also request to read a book to the librarian and also provide feedback on the book after reading it. The user can also check out the stats such as the no. of books completed by the user and section wise books read.

User Interfaces

User Dashboard : It provides the users with options to search for the books they have access to or the books they have already completed as well as the option to navigate to the books they don't have access to.

Librarian Dashboard : It provides the librarian with the option to navigate to sections where it can check the different sections of books uploaded, the requests raised by the users and the stats of the platform.

Other Core Functionalities

User Authentication : Secure login and registration processes implemented using jwt authentication.
Sending scheduled emails and a Monthly Report to each user at a specific time of the day has also been implemented using Redis and Celery.

Project Presentation Video Link

MAD2_LMS_Demo.mp4