IITM Modern Application Development - 1

Library Management System

Tharun Kumarr A - 22f3000527

Description

This project is a Library Management System through which multiple users can request and read e-books. Users and admins can register and log in to the application. The admin will have options to manage Sections and books (CRUD Operation - Create, Read, Edit and Delete), issue and revoke book access to different users, view all the users and the books borrowed by them respectively and search for books based on book title or author or section. The user will have options to view all the books present in the library database, request book access, return books anytime, have a custom profile with personal details and read any book issued right in the application. The design is minimal and straightforward so that it is easy for the user to navigate the website.

Database Design

Entity Identification

The following entities and their respective attributes were identified for the given problem statement,

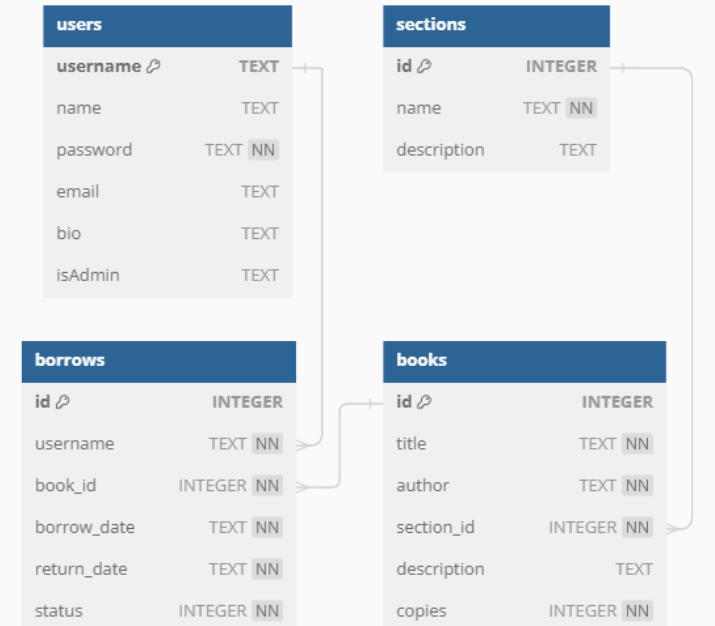
- User - username, name, password, email, bio, isAdmin
- Section - id, name, description
- Book - id, title, author, section_id, description, copies
-

The following relationships were identified,

- The book is presented in Section
- Users can borrow books
- Users/admin can return/revoke books

Schema

After identifying all the entities and their attributes, a schema diagram is constructed



Tech Stack

- **Flask** - Flask is a lightweight and versatile web framework for Python, ideal for building web applications and APIs. It provides tools, libraries, and patterns to help developers create scalable and maintainable web applications.
- **SQLite3** - SQLite3 is a lightweight, serverless relational database management system that is embedded directly into the application. It provides a simple and efficient way to manage databases.
- **Jinja 2** - Jinja2 is a powerful and widely-used templating engine for Python, commonly used in web development with frameworks like Flask. It allows for dynamic content generation by embedding Python-like expressions within HTML or other text-based templates.

API Design

- GET request - For rendering HTML pages and return data from sqlite server.
- POST request - For add and edit operations on the database. Used in adding users, books, sections, editing books, sections etc...
- DELETE request - For deleting operations performed on the database