Bookly Library App

Modern application development-1

Submitted by

Saksham Sirohi

Roll number: **22f1001635**

IITM Online BS Degree Program,

Indian Institute of Technology, Madras, Chennai

Tamil Nadu, India, 600036

Project Report

Made By:

Saksham Sirohi

22f1001635

22f1001635@ds.study.iitm.ac.in

Introduction

Hi! I'm Saksham, a versatile developer proficient in Python and Java, passionate about leveraging modern technologies. Eager to combine technical skills with growing data science interests to extract valuable insights. Collaborative team player & quick learner, committed to delivering innovative, high-quality solutions.

Project Description

The library management web application built with Flask provides a comprehensive platform for managing a library's collection and serving its patrons. Key features include user management (librarians and regular users), book and section management, book requests and approvals, book purchase and feedback, and reporting/analytics. Librarians can add, edit, and remove books and sections, while users can browse the library, request and purchase books, and provide feedback. The application follows a modular architecture, leveraging SQLAlchemy for database operations and Matplotlib for data visualization, ensuring a robust and feature-rich library management solution.

Technologies Used

Flask: The Python web framework used to build the application, enabling efficient routing and view management.

SQLAlchemy: The ORM tool used for interacting with the database, providing a Pythonic abstraction layer for database operations.

Flask-Login: Utilized for user authentication and session management, ensuring secure user interactions.

Flask-WTF: Integrated for handling forms and data validation, enhancing user input and submission.

Bootstrap: The front-end framework used to create a responsive and visually appealing user interface.

Matplotlib: Employed for generating dynamic statistical charts and visualizations, providing valuable insights into library usage and trends.

SQLite: The relational database management system used to efficiently store and manage user, book, section, request, and feedback data.

DB Schema Design

The User model has fields for user identification, authentication, and membership tiers. The Section model categorizes books, while the Book model stores book details, content, and thumbnails. The Request model tracks book requests, status, and return dates. The Feedback model records user feedback for books.

This schema establishes one-to-many and many-to-many relationships between the entities, ensuring data integrity and enabling features like book management, user authentication, request approvals, and feedback collection. The schema follows best practices for database design, providing a robust foundation for the library management application.

API Design

The application's API design includes CRUD endpoints for managing books and sections, with input validation and role-based access control. The API is integrated into the existing Flask application, enabling a smooth transition to a more comprehensive API-based architecture if needed.

Architecture and Features

The library management application follows a modular architecture, with Flask routes handling user interactions, Jinja2 templates rendering the views, and WTForms managing user input. Librarians have access to features like adding/editing books and sections, approving book requests, and viewing usage statistics. Users can browse the library, request and purchase books, and provide feedback.

Video

https://drive.google.com/drive/folders/1pyACsShajxgK8UZtb7k-svBDzprXrfR-?usp=sharing