Modern Application Development 1

Project Report

January 2024

Library Management System

Code Name : LibManage

Andrew David Ratnam Sathiadass

DS23f1002382

Library Management System Overview:

Description: This system, using Python, Flask, and Bootstrap with SQLite, enables a Librarian to manage books, genres, and users, while Readers can register, login, request books, and provide feedback.

Project Goals: Demonstrate Database Systems and Model-View-Controller Architecture understanding.

Frameworks: Flask, WTForms, SQLAlchemy, Matplotlib, Flask-RESTful.

Resources: GitHub Repo 1, GitHub Repo 2, GitHub Repo 3.

Procedures: Started with Database (Model) development, expedited development post model refinement.

Schedule: See Contributions.

Presentation Video Link: View Presentation

Future Improvements: Enhance database design, implement CRUD API with JWT, strengthen SQL validation, connect feedback to user profiles, integrate JavaScript for client-side efficiency, automate book cover image generation.

Delving Deeper: Unraveling the Layers of LibManage

Ahoy there, wanderer of the digital realm! Prepare to embark on a whimsical journey through the intricacies of our system. As we casually saunter through the layers of our creation, let us uncover the mystique behind our database design. From the charming allure of the presentation layer to the logic that dances through our veins, and the models that whimsically shape our reality—we invite you to join us on this carefree exploration of data brilliance. So, sit back, relax, and let us meander through the serendipity of our digital escapade.

1.Data Base Design

- The database is in the Third Normal Form (3NF). Each table is free of repeating groups and exhibits dependencies solely on primary keys, ensuring data integrity and minimizing redundancy.

2. Presentation Layer

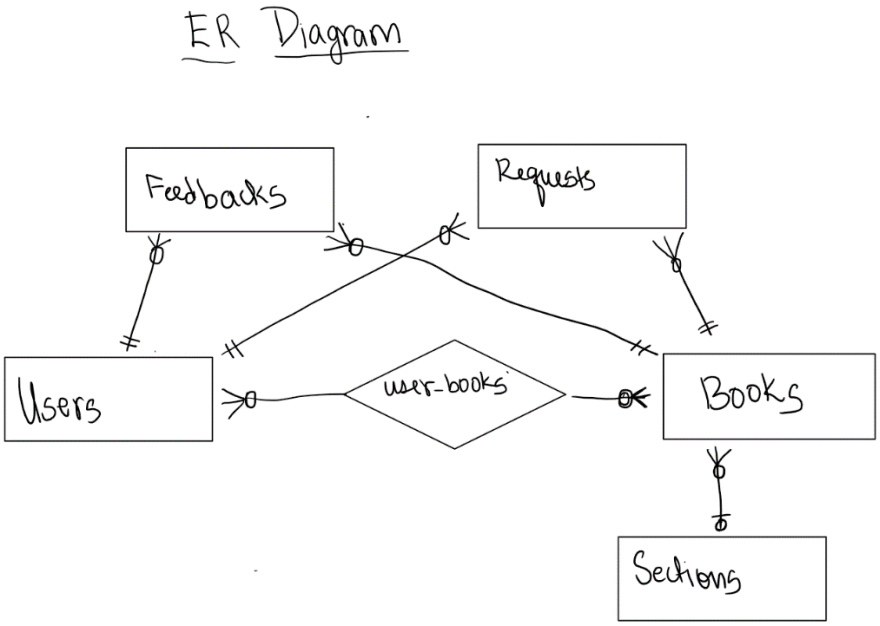
- The application leverages Flask routes and static HTML templates enhanced by Bootstrap, seamlessly blending dynamic functionality and aesthetic appeal for an immersive user experience

3. Business Logic

- The application incorporates WTForms and Flask Forms for robust input validation, bolstered by SQL validations pre-data retrieval to ensure basic security measures.
- Furthermore, user authentication safeguards critical actions such as user or book deletion and request approval.

4. Data Model

- The data model, implemented in SQLite using SQLAlchemy, comprises relational tables: Users, Books, Sections, Feedbacks, and Requests.
- Relationships are established through foreign keys and many-to-many associations, ensuring data integrity and consistency.
- This relational model facilitates efficient management of users, books, feedback, requests, and sections within the system.



Notation	Meaning
	Relationship
	One
	Many
	One and ONLY One
	Zero or One
	One or Many
	Zero or Many

➢ Conclusion

- Demonstrates rudimentary database and MVC understanding.
- Future enhancements aim for functionality, security, and user experience.
- Fork the repository for additional features addition. [GitHub Repository Link]