

Author

Yatin Thakral

23f2000574

23f2000574@ds.study.iitm.ac.in

I am a second-year student pursuing a Bachelor of Technology (B.Tech) degree in Electronics and Communication Engineering (ECE) with a specialization in Artificial Intelligence and Machine Learning (AI/ML) at Netaji Subhas University of Technology.

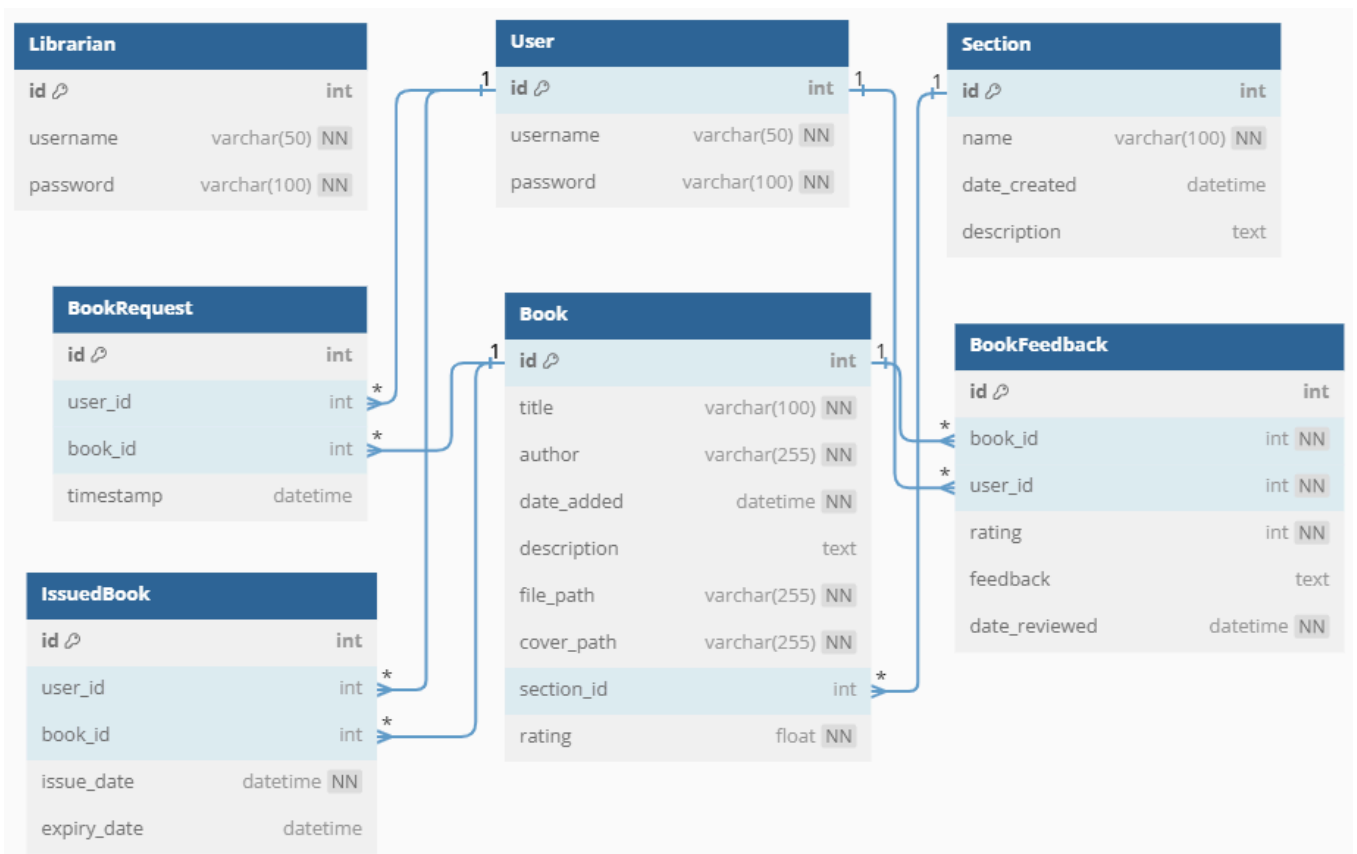
Description

The project aims to develop a Flask-based library management system where users can browse, request, borrow, and review books and librarians can efficiently organize and manage books, sections and users.

Technologies used

- Flask: Python-based web framework for developing the backend.
- Jinja2: Templating engine for generating dynamic HTML content.
- SQLAlchemy: ORM (Object-Relational Mapping) tool for database interaction.
- Bootstrap: Frontend framework for designing responsive web interfaces.
- SQLite: Database management system for storing application data.

DB Schema Design



- Normalization: The schema follows normalization principles to minimize redundancy and ensure data consistency.
- Foreign Key Relationships: Tables are linked via foreign key relationships to establish associations between entities.
- Unique Constraints: Unique constraints ensure that critical data such as usernames, book titles, and section names are not duplicated.
- Default Values: Default values are provided for certain columns like `date_added`, `timestamp`, `issue_date`, and `date_reviewed` to simplify data entry and ensure consistency.
- Data Integrity: Not null constraints and foreign key relationships enforce data integrity, preventing the insertion of incomplete or invalid data.
- Scalability: The schema is designed to scale efficiently as it separates concerns into distinct entities with clear relationships.

Architecture and Features

The project follows the MVC (Model-View-Controller) architecture, with controllers handling user interactions, models managing data, and templates rendering the views. The main controller logic, defined as routes, is present within the `app.py` file. Each route corresponds to specific functionalities, such as user authentication, book management, and dashboard rendering. Templates responsible for the frontend presentation are stored in the `templates` directory. These HTML files utilize Flask's Jinja2 templating engine for dynamic content rendering. Static files like CSS, JavaScript, and images are stored in the `static` directory.

Features implemented include user authentication, book browsing, and dashboard management. User authentication ensures secure access to the platform, while book browsing enables users to explore available titles. Dashboard management provides users with a personalized space to view borrowed books and manage account settings. Default features include basic CRUD operations for managing library resources, while additional features include responsive design for multi-device compatibility, search and filtering for efficient book discovery, and feedback and rating systems to enhance user engagement.

Video

https://drive.google.com/file/d/1vMqg9QlivqCgBU5ijQ_4Y1YSFE5QiouD/view?usp=sharing