Library Management system Report

Author:

Name- Abhinav Tiwari

Roll No.-22f1001616

Mail id -22f1001616@ds.study.iitm.ac.in

About Me: My self Abhinav Tiwari , i am from Sultanput, Utter Pradesh(U.P.).I am perusing BS Degree as a full time student.

Description:

"Explore our Library Management System (LMS) for a great reading experience! You can find all kinds of books, from old favorites to new releases, on our easy-to-use platform. Admins can easily create events and customize the reading journey for everyone. Welcome to BookScape LMS, where organized book management meets the fun of discovering new stories!"

Technologies used:

- Flask for application code
- Jinja2 templates + Bootstrap for HTML generation and css styling
- SQLite for data storage
- Flask_login is used for secure user login
- Flask_restful and Flask_cors for building RESTful APIs

DataBase Schema Design:

In User schema the **User** class includes columns for **id**, **username**, **name**, and **password**. The **id** column serves as the primary key, and it auto-increments for each new user. The **username** column is marked as unique and not null to ensure that each user has a distinct username. The **name** column is not null, capturing the user's name. Lastly, the **password** column is marked as not null for ensuring the security of user accounts. The class also includes relationships for reviews and book requests, similar to the provided **Admin** schema.

In Librarian schema, the Librarian class has columns for l_id, l_username, l_name, and l_password. The l_id column serves as the primary key, and it auto-increments for each new librarian. The l_username column is marked as unique and not null to ensure that each librarian has a distinct username. The l_name column is not null, capturing the librarian's name. Lastly, the l_password column is marked as not null for ensuring the security of librarian accounts. The get_id method is included as required by the UserMixin class, returning the string representation of the librarian's ID.

In E-Book schema, the EBook class has columns for id, name, image, pdf, release_date, author_name, total_pages, return_date, return_date_expected, content, section_id, section_name, and price. The id column serves as the primary key, and it auto-increments for each new eBook. The section_id column is a foreign key referencing the id column of the Section table, establishing a relationship with the section where the eBook belongs.

Additionally, relationships are defined for reviews (reviews_relation), book requests (book_requests), and the section to which the eBook belongs (section). The average_rating column is included to store the average rating of the eBook, initialized with a default value of 0.

In Review schema, the Review class has columns for id, user_id, rating, user_review, and ebook_id. The id column serves as the primary key, and it auto-increments for each new review. The user_id and ebook_id columns are foreign keys referencing the id columns of the User and EBook tables, respectively, establishing relationships with the user who wrote the review and the eBook being reviewed.The reviewer and book_relation relationships are defined to connect the Review class with the User and EBook classes, respectively.

In Section schema, the Section class has columns for id, name, description, and date_created. The id column serves as the primary key, and it auto-increments for each new section. The book_requests relationship connects the Section class to the BookRequest class using the foreign key relationship, and the books relationship connects it to the EBook class using the back_populates attribute, establishing a bidirectional relationship between Section and EBook

In Book-Rrquest schema, the BookRequest class has columns for id, request_id, request_time, return_time, status, user_id, ebook_id, section_id, is_requested, is_returned, is_revoked, and is_issued. The id column serves as the primary key, and it auto-increments for each new book request. The request_id column is marked as unique to ensure each request has a unique identifier. ForeignKey relationships are established for user_id, ebook_id, and section_id columns. The is_requested, is_returned, is_revoked, and is_issued columns are Boolean flags indicating the request status. Relationships are defined with unique backref names to link the BookRequest class to the User, EBook, and Section classes.

In Downlode schema, the Downlode class has columns for id, user_id, ebook_id, section_id, status, is_payed, and is_downloded. The id column serves as the primary key, and it auto-increments for each new download. ForeignKey relationships are established for user_id, ebook_id, and section_id columns. The is_payed and is_downloded columns are Boolean flags indicating the payment and download status, respectively. Relationships are defined with unique backref names to link the Downlode class to the User, EBook, and Section classes.

API Design:

Our system employs a consistent table schema to facilitate seamless CRUD operations through dedicated API endpoints. This design enables effortless data manipulation using standard HTTP requests, including GET, POST, PUT, and DELETE. By offering a standardized approach to interacting with the data, the API ensures the synchronization of the database with the application's requirements. This not only streamlines the process of maintaining and updating the application but also enhances its scalability and adaptability over time.

Architecture and Features:

Our project adopts the MVC architecture, with controllers managing user requests in "app.py" and templates providing a user-friendly display from the "templates" folder. Models handle data and database interactions.Default features include secure login, advanced filtering, and booking management. Additional enhancements, such as user ratings, show reviews, and revenue reports, amplify user experience and empower administrators with valuable insights. Efficiency and security drive our implementation, ensuring a seamless and secure platform for users to explore, review, and book shows.

Click here for LMS project Video:

https://drive.google.com/file/d/1iqbylMxN0zia2CadKuBSY3s9pRw6LC_6/view?usp=sharing