BIBILARY

PROJECT DOCUMENTATION

Author

Name : Santhosh G S	**Roll-no :** 21F1001255	**E-mail :** 21f1001255@ds.study.iitm.ac.in

I am a final-year student pursuing a Bachelor's degree in Electrical and Electronics Engineering at SSN College of Engineering in Chennai. I am passionate about creating solutions that simplify life.

Description

Bibilary is a Library Management application where an assigned librarian can add genres for books, add books in those genres, and edit them whenever needed. Librarians can also accept or reject requests from users to borrow books. From the user's point of view, they can request the required books for reading access for a specified period of time and can also purchase a book by making a payment.

Technologies used

Flask
Flask-RESTful - API for the application
Flask-SQLAlchemy - ORM for SQLite database
SQLite - Database
Jinja2 - Template
Bootstrap - Basic Styling
HTML - Markup for skeleton content of the webpage
Chart.js - Creating graphs on genre and book stats

DB Schema Design

Database schema design can be found in this document. The ER diagram of the database can be found here.

User	id(PK)	username	email	passhash	name		
Librarian	id(PK)	username	email	passhash	name		
Genre	id(PK)	name	date_created	description			
Book	id(PK)	title	authors	genre_id	price	quantity	summary
Request	id(PK)	user_id	book_id	date_requested	days_requested		
Borrow	id(PK)	user_id	book_id	date_issued	date_due		
Purchase	id(PK)	user_id	book_id	transaction_id	date_purchased	quantity	
Cart	id(PK)	user_id	book_id	quantity			
Transaction	id(PK)	user_id	date_paid				
Feedback	id(PK)	user_id	book_id	subject	ratings	review	

API Design

The application allows for CRUD operations specifically on the Genre table through the REST API created using Flask-RESTful. Additionally, read-only operations (GET) are implemented for all other tables (User, Librarian, Book, Request, Borrow, Purchase) via the API. The detailed API design along with their corresponding routes can be found in this document. The yaml file explaining the API design can be found in the root folder named api-design.yaml

Architecture

- The controller routes are implemented in routes.py, located in the root directory.
- The template HTML files are stored within the templates directory in the root directory.
- The static image files are located in the static directory, which, in turn, is present in the root directory.
- The database models are defined in models.py, situated in the root directory.
- The RESTful API, responsible for CRUD operations on the table data, can be found in api.py within the root directory.
- Environment variables are accessible in config.py in the root directory.

Features

- Every user post request is validated in the background before being stored in the database.
- The search functionality can be performed on the HOME page, MY BORROWS, and MY BOOKS pages based on the genre and title of the book.
- Users can buy a book with a certain quantity by paying the specified price.
- Both librarians and users can edit their profiles and change their passwords.
- Books that are out of stock cannot be bought or borrowed.
- Users who bought or borrowed a book can provide feedback to the book and rate it, visible to general users.
- The user registration page enforces a password policy requiring a minimum of two lowercase letters, two uppercase letters, two numbers.
- A user cannot request or borrow a book that has already been borrowed.
- A user cannot request or borrow more than 5 books at a time.

Video

Bibilary - Project Presentation.mp4