IIT Madras BS Degree in Data Science & Programing

REPORT SUBMISSION

Morden Application Development (MAD 1)

Application on **: *LIBRARY MANAGEMENT SYSTEM***

~ Submitted by

Name - Bristi Biswas

Roll No. – 22f3001833

Submitted on – 30.03.2024

- **Presentation Video Link - View Video**

Models Used & it's Attributes

USER	ADMIN	BOOKS	CATEGORY	BOOK REQUESTS	ORDERS
id	id	book_id	category_id	request_id	order_id
username	adminname	book_name	category_name	book_id	book_id
password	password	category_id	description	book_name	book_name
email id	email id	author_name	date_created	author_name	category_id
		price		category_id	author_name
				user_id	price
				username	user_id
				date_of_issue	username
				date_of_return	payment(status)
				status	date_of_issue
				user_request_date	
				feedback	

Database used – Sqlite 3

Requirements – Bootstrap, Picsum, Font-awesome, Chart.js

Description of Entities

- **User :** It stores details of the registered users of the application which includes their username, password & email ids.

- **Librarian :** The one who manages the entire system & stores information such as the name of the admin, password & email id.

- **Category :** It stores details of the various sections/categories including category name, description and the dates on which each category is created.

- **Books :** Stores information about the books available on the system which includes their titles, name of the authors, prices & along with the category id to which it belongs to.

- **BookRequest :** Stores information of the books that are requested by the users including the details of the requested book, user details & the status.

- **Orders :** Stores details of the books of which the user placed an order including the user details, payment information & status.

APPROACH & WORKING FUNCTIONALITIES

1. There is an index page which is basically the first appearance of the app to the users which has two separate logins one for Users & other for the Librarian.

2. If the user is new then he/she can simply sign up to the application with a username & password after submitting which, it will take them back to the login page & logging into which it will take it to their own dashboard.

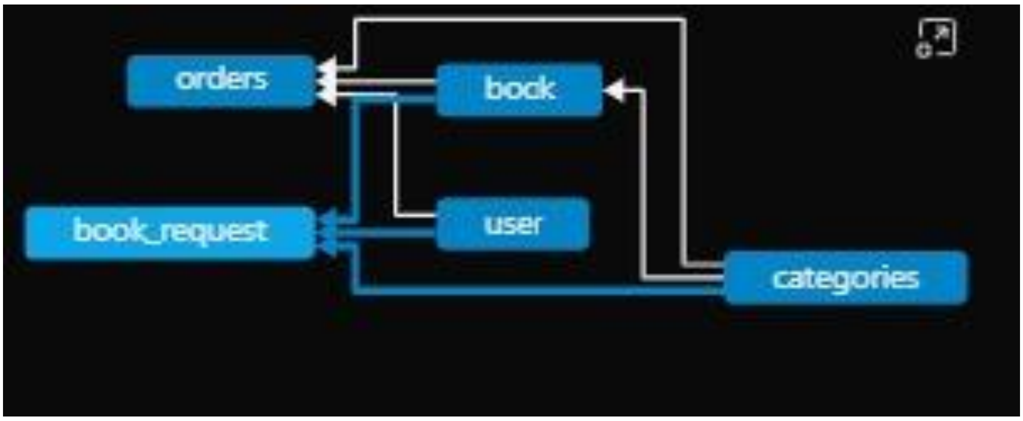
3. The user dashboard contains different sections in the navigation bar:

    - **View Books –** contains all the books under it's respective category with a basic search functionality for user easy accessibility on the basis of categories, books, author name & maximum price limit. The images of the books are auto generated using picsum. Each book can be requested for 7 days or can be ordered by paying the price for permanent accessibility (as pdfs).
    - **User Profile** – where the user can update his/her username, new password & email id only if he/she confirms the old password properly.
    - **My Books –** which navigates to a page divided into 3 tables one for the recent requests (but not granted yet), next is the current books which is granted by the admin for 7 days including the date of issue as well as return& the user can return before it as well, the last table is the rejected ones where the book requests are rejected by the admin.
    - **My Orders** – which navigates to a page divided into 2 tables one is the cart where user adds the books which he/she wants to buy & then after reviewing by the admin they get to pay the price & download the book using pdf. For simplicity of pdfs, I have used windows.print() method.
    - **Logout –** to logout from the application.

4. The application has only one Librarian who can login with his/her credentials which takes it to the Librarian dashboard which contains add category where the Librarian can add new category to the application by filling up certain details of the new category on the database, it also has different sections in the navigation bar.

- **Sections-** opens up to a table of all the categories including the no. of books it currently has, the date of it's creation & certain CRUDE operations to edit, update or delete the category. It has two buttons for each category one is to add new books under that category & other is show category which shows all the books under that category along with actions like to edit & delete the books.
- **Admin Profile-** where the admin can update his/her username, new password & email id only if he/she confirms the old password properly.
- **Request-Management-** which navigates to a page which is divided into 3 tables. One is the Current requests where user requests for a book & the admin can either grant the book by filling up a form for date of issue & date of return or reject any request. Next table is the Request Granted Table which shows all the user & book details of which the admin has granted the requests but it also has an action for each rows of requests to revoke it anytime whenever the admin wants to. And the last table is Returned Books which shows the books which users returns within the alloted time frame. The auto-revoke function invokes when the user fails to return the books before or on the date of return.
- **Order-Management-** which navigates to a page which is divided into 3 tables. One is Requested Transactions where the admin can view the user & books details & notify user to pay. Other table is the Pending Transactions where the user is asked to pay the price of the book for permanent accessibility. The last table is Paid Transactions which contains the details of the users who has paid the necessary payment.
- **Statistics-** Distribution of number of books associated to a particular category
- **Logout-** to logout from the application

ER Diagram



This ER Diagram is taken from the in-built technical functionality of the database SQLite 3

The Entity-Relationship Diagram visualizes the representations of relationships between various entities in the database

- **User** has **Many-to-Many** relationship with **BookRequest** and **Orders** which means one user can request many books but maximum limit is 5 as per the constraints in the controllers and one book can be requested by multiple users and also order as many books he/she wants.

- **Category** has **One-to-Many** relationship with **Book** which means that one category can have multiple books whereas each book belongs to only one category.

- **BookRequest** has **One-to-Many** relationship with **Book** which means one book can be multiple requests. But each request is for only one book.

- **Orders** has **One-to-Many** relationship with **Book** which means one book can have multiple orders but each order is only for one book.

Overview & Challenges Faced

The project has helped me to not only implement the logics but also understand the technical functionalities from the core. Though it was quite challenging but at the same time learnt a lot. Some of the core functionalities like the maximum book limit & the auto revoke made me introduce to some methods & working of Python in the controllers. I have also used flash message technique where everytime there is a change in the system there is a message kind of an alert both for the user & admin. CRUDE operations from the admin side like adding new books & sections , deleting & updating them as well. Managing the Requested & Ordered Books based on the basis of the current status & creating the statistics was full of learning & understanding core concepts. Bootstrap on the other hand, made the navbars, buttons & cards aesthetically relevant to the user & admin. I have also used Search functionality for the user where they can search by books, category, author name & maximum price. Also used a bunch of html templates & applied Jinja. All over, tried to make it as presentable as possible with the knowledge from the theory of Mordern Application Development 1.

Thank You