

**UNIVERSITY OF COMPUTER STUDIES, MEIKTILA**

**2024-2025 ACADEMIC YEAR**

**CS-4225(Advanced Database System)**

**Library Management System**

**Group-6 September,2025**

**Member List**

|  |  |  |
| --- | --- | --- |
| No. | Name | Roll Number |
|  | Mg Thet Paing Phyo (Leader) | 4CS-1 |
|  | Ma Ain Jinn Moe | 4CS-24 |
|  | Mg Thu Htoo San | 4CS-39 |
|  | Ma Thu Shin Nway | 4CS-41 |
|  | Ma Khin Myo Thiri | 4CS-25 |
|  | Mg Kyaw Zayar Min | 4CS-50 |
|  | Mg Thiha Aung | 4CS-57 |
|  | Mg Han Sein | 4CS-52 |
|  | Mg Zin Mg Mg Phyo | 4CS-56 |
|  | Mg Yan Myo Aung | 4CS-55 |

**CONTENTS PAGES**

1. **Description 3**
2. **Objective 3**
3. **Data Dictionary 4**
4. **Entity Relationship Diagram 6**
5. **Implementation 7**
6. **Conclusion 8**
7. **Description**

This document provides a technical overview and detailed documentation for a web-based Library Management System. The system is designed to streamline library operations, including the management of books, users, borrowing transactions, and administrative tasks. Its primary goal is to provide a comprehensive, user-friendly, and efficient solution for libraries of all sizes. The development of this system utilized the following key tools: **Laravel**, **PHP**, and **phpMyAdmin**.

1. **Objective**

The primary objective of this project is to create a robust and scalable web application that efficiently manages library resources. This includes:

* To provide a single platform for librarians to add, update, and manage the library's book collection.
* To securely register and authenticate users, distinguishing between different user roles (e.g., student, faculty, librarian).
* To simplify the process of borrowing and returning books, while accurately tracking due dates and managing late fees.
* To enable users to quickly find books using various search criteria such as title, author, or ISBN.
* To provide librarians with tools to generate reports on borrowing trends, book availability, and overdue items.

### Data Dictionary

This section provides a detailed breakdown of each table and its fields.

#### 1.admin Table

* **id**: A unique, non-negative integer that serves as the primary key for each admin.
* **email**: The admin's unique email address.
* **phone\_no**: The admin's phone number.
* **address**: The physical or logical location of the administrator.
* **created\_at**: A timestamp indicating when the record was created.
* **updated\_at**: A timestamp that is automatically updated upon modification.

#### 2.books Table

#### id (Primary Key): The default primary key used by Laravel.

#### title: The title of the book.

#### author: The author of the book.

#### isbn: The book's International Standard Book Number.

#### total\_copies: The total number of copies of the book.

#### available\_copies: The number of copies currently available for borrowing.

#### shelf\_id (Foreign Key): Links to the shelves table.

#### category\_id (Foreign Key): Links to the categories table.

#### barcode: The book's barcode.

#### image: A path or URL to the book's cover image.

#### Relationships:

#### belongsTo(Category::class): Each book belongs to one category.

#### belongsTo(Shelf::class): Each book is located on one shelf.

#### hasMany(Borrowing::class): A book can have multiple borrowing records.

#### 3.book\_requests Table

* **id**: The primary key for the request records.
* **book\_id**: A foreign key linking to the id of a book in the books table.
* **requester\_name**: The full name of the requester.
* **requester\_email**: The email of the requester.
* **requester\_phone**: The phone number of the requester.
* **created\_at**: A timestamp indicating when the request was submitted.
* **updated\_at**: A timestamp that is automatically updated upon modification.

#### 4.borrowings Table

#### borrow\_id (Primary Key): A unique identifier for each borrowing record.

#### member\_id (Foreign Key): Links to the members table.

#### book\_id (Foreign Key): Links to the books table.

#### borrow\_date: The date the book was borrowed.

#### due\_date: The date the book is due for return.

#### return\_date: The actual date the book was returned.

#### status: The current status of the borrowing (e.g., 'borrowed', 'returned', 'overdue').

#### renewal\_count: The number of times the borrowing has been renewed.

#### Relationships:

#### belongsTo(Member::class): Each borrowing record belongs to a member.

#### belongsTo(Book::class): Each borrowing record is for a specific book.

#### 5.categories Table

#### id (Primary Key): The default primary key.

#### name: The name of the category (e.g., 'Fiction', 'Science').

#### Relationships:

#### hasMany(Book::class): A category can contain many books.

#### 6.members Table

* **id**: The primary key for the members table.
* **name**: The member's full name.
* **email**: The member's unique email.
* **phone\_number**: The member's phone number.
* **roll\_no**: The member's unique roll number.
* **image**: A file path or URL for the member's profile picture.
* **year**: The academic year.
* **major**: The academic major.
* **gender**: The member's gender.
* **qr\_code**: A unique QR code for the member.
* **is\_admin**: A boolean flag to determine if the member has admin access.
* **registration\_date**: The date the member registered.
* **expired\_at**: The expiration date of their membership.
* **created\_at**: The date and time the record was created.
* **updated\_at**: A timestamp that is automatically updated upon modification.

**Relationships:**

* hasMany(Borrowing::class): A member can have many borrowing records.

7. reservations Table

* + id (Primary Key): The default primary key.
  + book\_id (Foreign Key): The ID of the reserved book.
  + member\_id (Foreign Key): The ID of the member who reserved the book.
  + status: The reservation status (e.g., 'pending', 'ready for pickup', 'canceled').
  + reservation\_date: The date the reservation was made.
  + expiration\_date: The date the reservation expires.

**Relationships:**

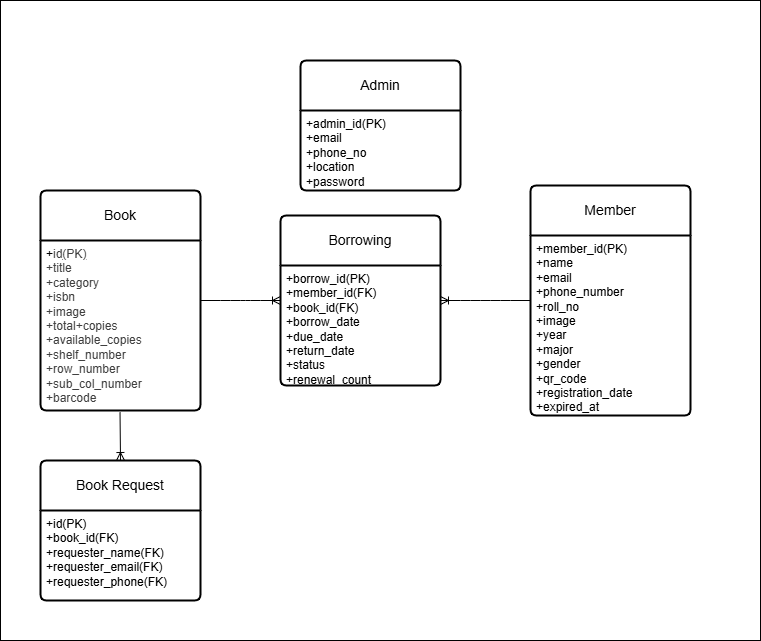
* + belongsTo(Book::class): Each reservation is for one specific book.
  + belongsTo(Member::class): Each reservation is made by one specific member.

**8.shelves Table**

* + id (Primary Key): The default primary key.
  + shelf\_number: The number of the shelf.
  + row\_number: The row where the shelf is located.
  + sub\_col\_number: The sub-column number for more precise location.

**Relationships:**

* + hasMany(Book::class): A shelf can hold many books.

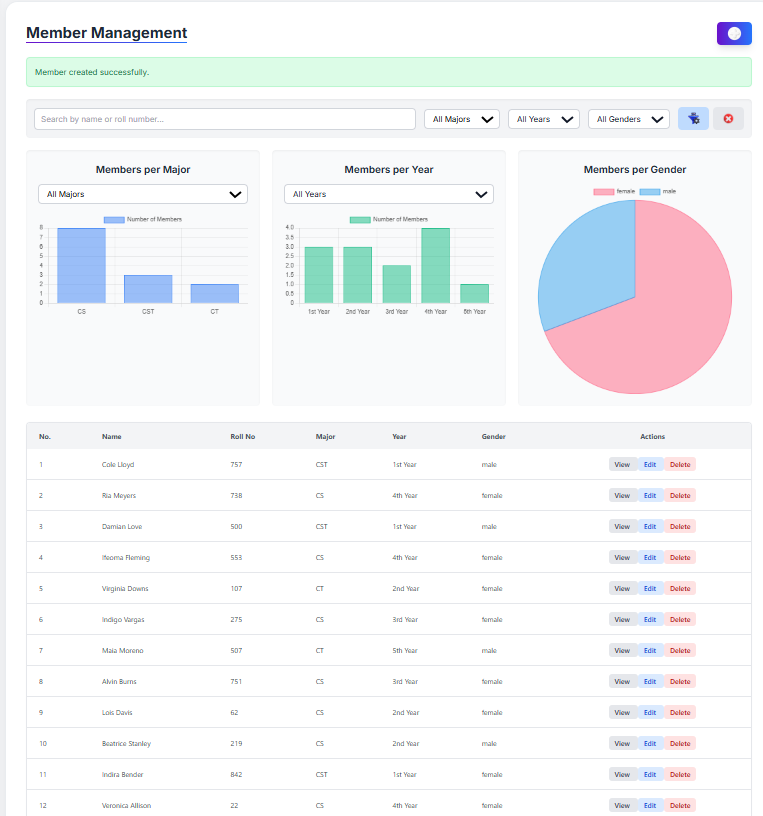
1. **Entity Relationship Diagram**

The system's database schema is based on five main entities: **Admin**, **Books**, **Book Requests**, **Borrowings**, and **Members**. The relationships between these entities are as follows:

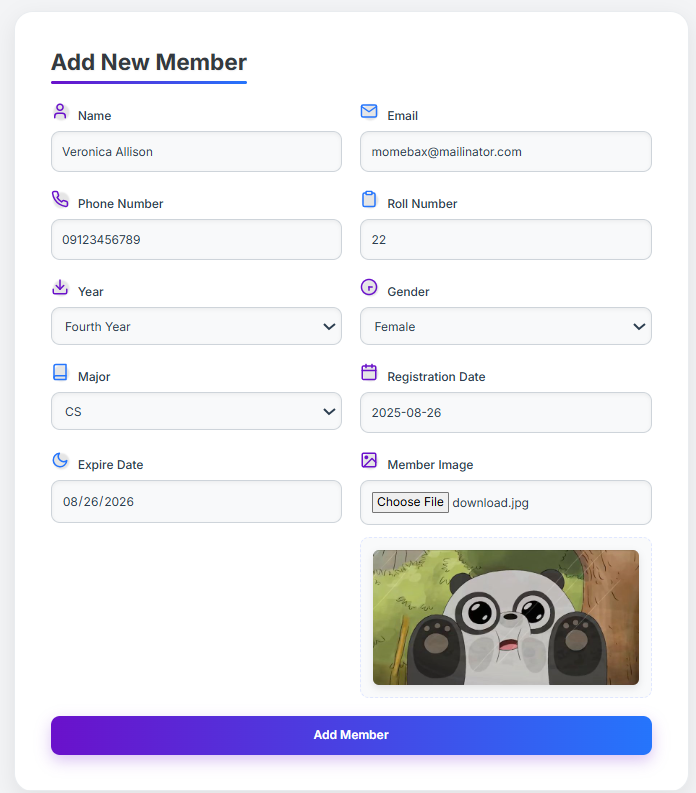
* **Members and Borrowings:** There is a **one-to-many** relationship between members and borrowings. A single member can have multiple borrowing records, but each borrowing record is associated with only one member.
* **Books and Borrowings:** There is a **one-to-many** relationship between books and borrowings. A single book can be borrowed many times, but each borrowing record is for one specific book.
* **Books and Book Requests:** There is a **one-to-many** relationship between books and book\_requests. A single book can have multiple requests, but each request is for one specific book.
* **Members and Book Requests:** The book\_requests table can be conceptually linked to the members table by matching the requester\_email with the email in the members table.
* **Admin and Members:** The admin table and the is\_admin flag in the members table suggest that administrative roles are handled at different levels. The admin table might be for super-admins, while the is\_admin flag is for general members who have been granted admin privileges.

**5.Implementation**

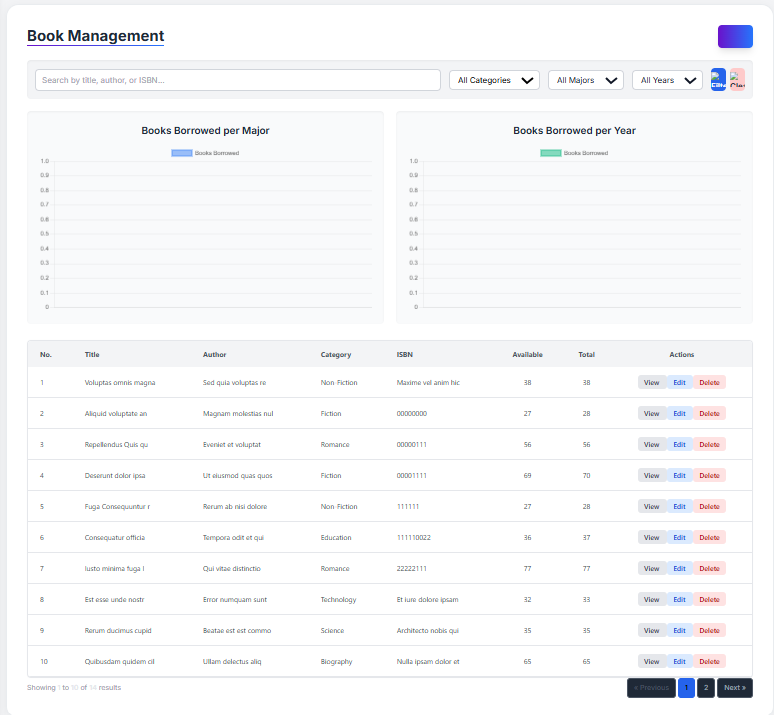
1.MemberList

****

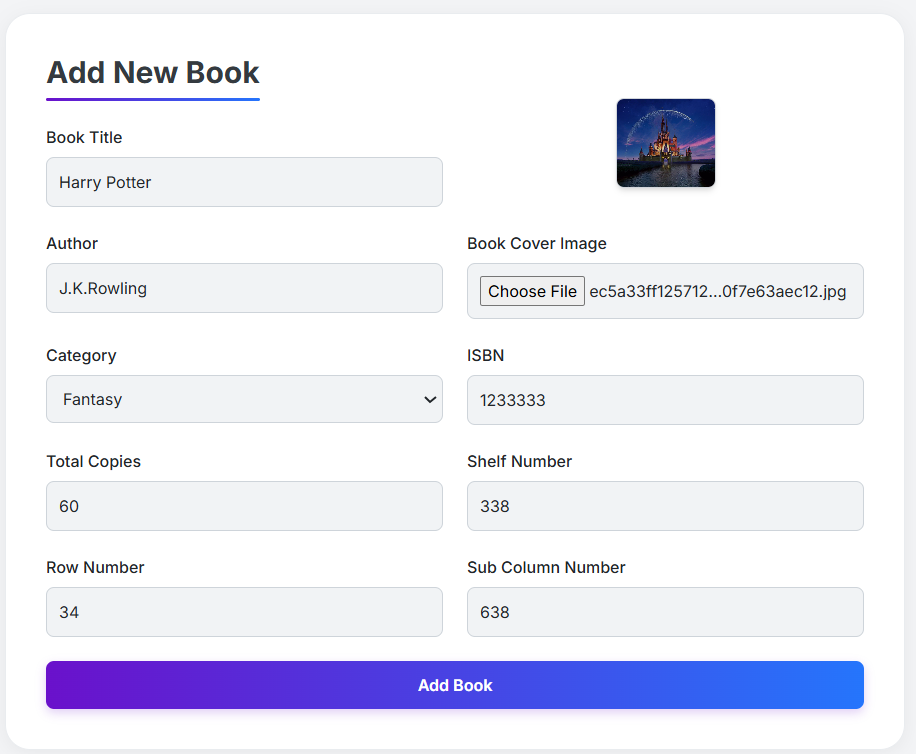
2.Register Library Member



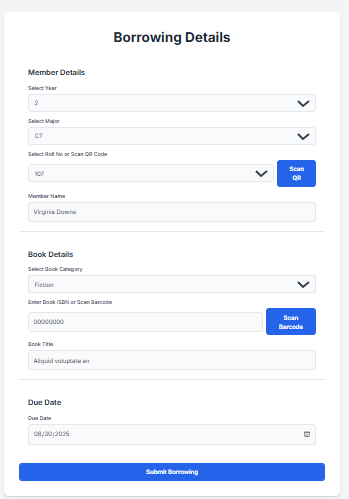
3. Book List



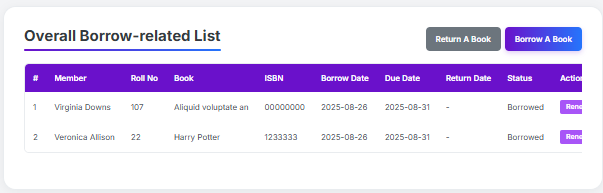
4.Add a book



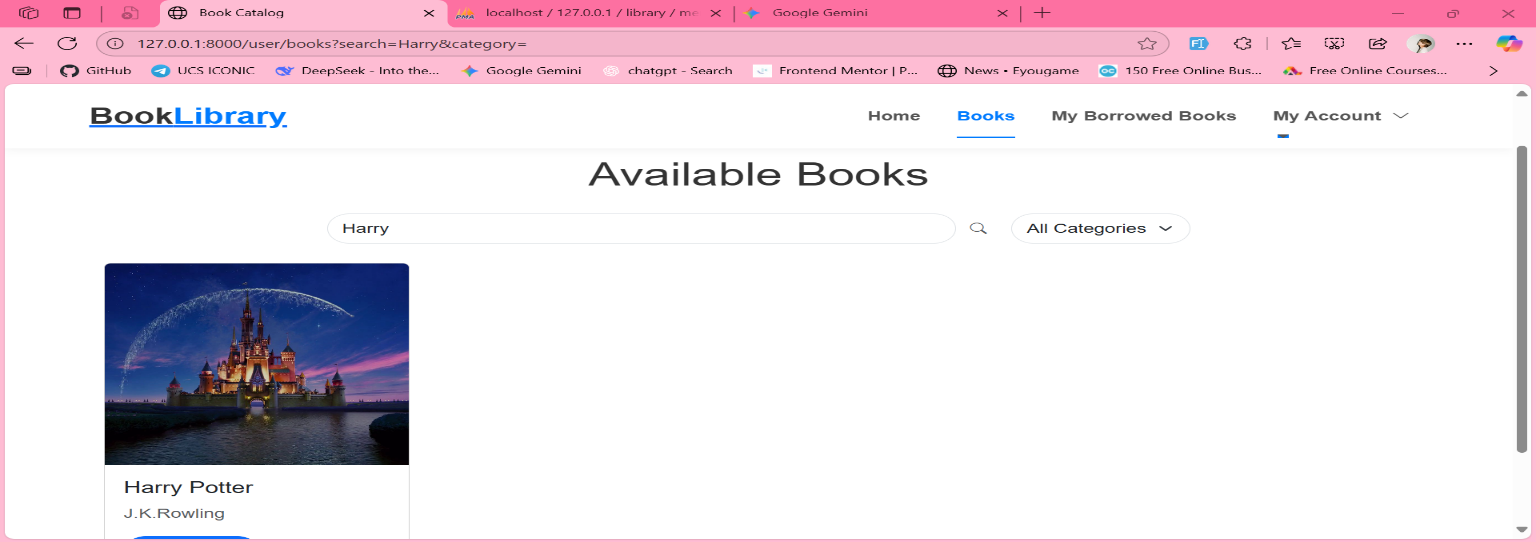
5.Borrow a book



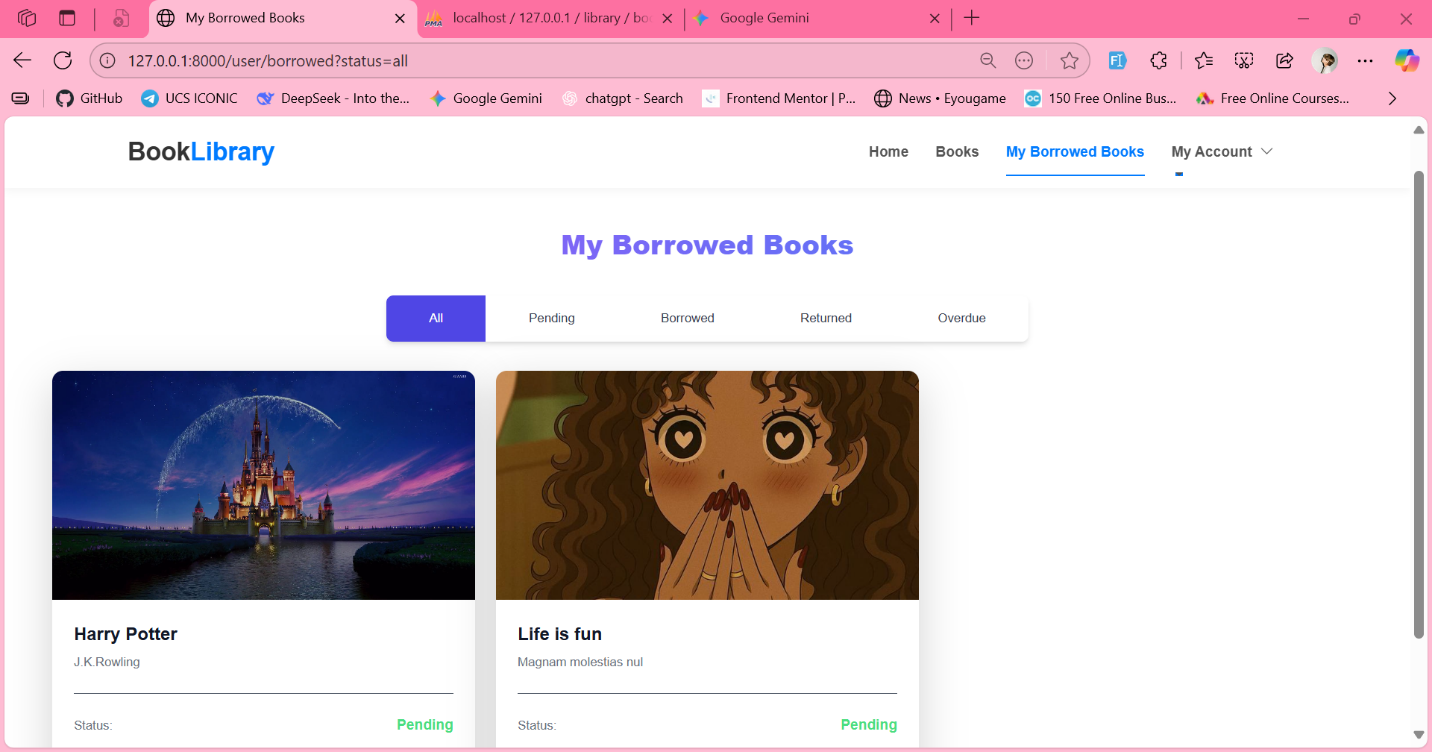
6.Borrowing book list



7.User-side



8.User borrowed book list



### 6. Conclusion

In conclusion, this Library Management System project provides a more comprehensive solution for modernizing library operations. The expanded database design, with dedicated tables for administrators, members, and book requests, ensures a scalable, secure, and user-friendly application. By implementing the described features, the system effectively addresses the core needs of managing books, users, and borrowing transactions, paving the way for a more efficient and accessible library.