**Project Overview**

**LibManage** is a command-line based Library Management System designed to streamline and digitize the core operations of a library. It allows librarians or admins to efficiently manage books, members, book issues and returns, with an integrated fine calculation system for overdue books. The project focuses on building a robust backend using Python and MySQL to provide a real-world learning experience in database handling and backend logic implementation.

The system is designed to be simple to use via the terminal, without any frontend interface, making it ideal for practicing Python, database design, CRUD operations, and real-world problem-solving without the overhead of UI development.

## Core Features

1. **Login and Authentication**
   * Secure login system for admins to access the platform.
2. **Book Management**
   * Add, update, view, search (by title, author, or genre), and delete books from the system.
3. **Member Management**
   * Register new members, view member list, avoid duplicate registrations, and delete member records.
4. **Book Issue System**
   * Issue books to registered members, assign due date (15 days from issue date), and prevent re-issuing an already issued book.
5. **Book Return System**
   * Record book return, auto-calculate fines (Rs. 5/day after due date), and update book availability.
6. **Fine Calculation**
   * Fine is automatically calculated based on delay in return and shown during return.
7. **(Planned) Reports and Analytics**
   * Track most borrowed books, overdue returns, and active members.
8. **(Planned) Logging and CSV Export**
   * Maintain transaction logs and export data for analysis/reporting.

## Technologies Used

* **Programming Language:** Python
* **Database:** MySQL
* **Database Connector:** mysql-connector-python (for database operations)
* **Development Environment:** VS Code / PyCharm / Terminal
* **Version Control:** Git (for tracking progress, optional)
* **Optional:** CSV module for export feature

## Database Schema

1. Admin

| **Field** | **Data Type** | **Description** |
| --- | --- | --- |
| username | VARCHAR (PK) | Unique admin username |
| email | VARCHAR (UNIQUE) | Admin email ID |
| password | VARCHAR | Hashed password |

1. Books

| **Field** | **Data Type** | **Description** |
| --- | --- | --- |
| book\_id | INT (PK) | Unique identifier for each book |
| title | VARCHAR | Book title |
| author | VARCHAR | Book author |
| genre | VARCHAR | Book genre/category |
| quantity | INT | Number of copies available |

1. Members

| **Field** | **Data Type** | **Description** |
| --- | --- | --- |
| member\_id | INT (PK) | Unique ID for each member |
| name | VARCHAR | Member's full name |
| email | VARCHAR (UNIQUE) | Member's email address |
| mobile\_no | VARCHAR | Member's contact number |
| date\_registered | DATE | Date of registration |

1. BookIssue

| **Field** | **Data Type** | **Description** |
| --- | --- | --- |
| book\_id | INT (FK) | Refers to Books.book\_id |
| member\_id | INT (FK) | Refers to Members.member\_id |
| issue\_date | DATE | Date when book was issued |
| due\_date | DATE | Calculated as issue date + 15 days |
| return\_date | DATE (nullable) | Actual return date (if returned) |
| fine | INT | Fine amount (if applicable) |
| status | VARCHAR | 'Issued' or 'Returned' |

### **DAY-1**

### 🔧 **Work Done:**

* Implemented secure **Admin Login** functionality.
* Integrated **MySQL authentication** using username-password matching.
* Set a **limit of 3 login attempts** to enhance security.
* Used the **colorama** Python library to add **colored terminal output** for better user experience.
* Refined user feedback messages with appropriate color codes for:
  + Input prompts
  + Successful login
  + Wrong password
  + Invalid admin
  + Login attempt warnings

### ✅ **Features Implemented Today:**

* Admin authentication with attempt limits
* Colored CLI feedback using colorama