

Library Management System

Ishwarya D K^{*1} and Prof. Sandeepa K M^{*2}

^{*1}Student, Department of Computer Applications

^{*2}Assistant Professor, Department of Computer Applications

Maharaja Institute of Technology Mysore, Mandya, Karnataka, India.

Abstract: *A library is a place where books are stored. However, there are more activities involved in managing a library which include maintaining the records of members, issued books, returned books, due dates, fines, the reading room, and other activities. As the number of books and members increases, it becomes quite challenging to maintain the issue and return records of the books issued manually. Keeping track of registers and different spreadsheets for due books and fines might result in inconsistencies in the database of the library*

The Library Management System I intend to develop will be an application that handles different activities involved in the management of a library and provides distinct levels of access to the staff, students, faculty, and administrators.

The system will be developed in such a way that each level has different privileges depending on their role in the institution. All the activities and data including catalogues, member records, issue records, reading room records, and reports will be centralized and accessible from one platform. The system will also be secure as users will only have access to features and data specific to them thus enhancing security.

The Library Management System will be a web-based application that will be hosted on a remote web server and it will be developed using CodeIgniter 4 Framework..

Keywords: *library*

1. INTRODUCTION

A library is normally expected to perform a range of additional functions apart from buying or selling written material. Some of these functions include maintaining members' data, issuing and receiving back items, late fees, reading room maintenance, and many other roles and tasks. These duties and tasks become quite challenging when handling a large amount of material or when there are many members. This is because such volumes of data and activity cannot be handled by hand or excel sheets plus other manual methods due to the high risk of errors.

A web-based library management system is therefore proposed to handle all the activities and functions as described above and put them all together for easier daily use. The program will have different levels of access to allow members and staff perform their activities conveniently while eliminating overlaps and saving time. It will also be provided as a web application that can be accessed via browsers where only specific features will be available to specific users as opposed to using excel sheets for multiple tasks.

The above-described activities will all be stored in a single database to avoid the need of cross-referencing multiple databases which is time-consuming and insecure. The application is built using CodeIgniter 4 framework which uses Model-View-Controller (MVC) architecture to manage requests by routing them to appropriate controllers and views while processing queries to generate results. The twitter bootstrap css framework was used to develop the application's frontend and features such as responsive tables, forms, dashboards, and book catalogues among others. The database was designed using the MySQL platform and the email management system was used to send out emails and reminders to members.



II. METHODOLOGY

Step 1: Requirement Collection and System Study

The first step is to identify the information handled by the library, the users who access it, and the problems in the current process. Requirements are derived from the implemented routes and views and are organized into authentication, administration, catalogue, membership, circulation, fines, seats, reporting, and communication. Input fields, outputs, validations, and access privileges are documented for each workflow.

Step 2: Module Decomposition

The overall problem is divided into manageable modules so that each area can be designed and tested independently. Common functions such as the shared header, navigation, session values, database connection, and response handling are reused. This decomposition also supports role-specific menus without duplicating complete pages.

Step 3: Data Modelling

The database structure is designed around roles, users, librarians, students, faculty, categories, books, issued books, fines, seats, and seat bookings. Keys and relationships are selected to support lookup, reporting, and transaction updates. Fields such as status, available copies, due date, return date, fine amount, booking date, and time slots represent the changing state of the library.

Step 4: Interface and Navigation Design

Responsive forms, dashboards, cards, tables, and status labels are planned for the different user roles. The administrator receives system-wide management and reporting access. The librarian receives operational modules. Students and faculty receive simplified dashboards and self-service views. Consistent colours, buttons, form labels, and sidebar navigation reduce the learning effort required for staff and members.

Step 5: Back-End Workflow Implementation

Controller methods are implemented for login, dashboard counts, librarian and category CRUD, member search and registration, book CRUD, issue and return transactions, fines, seats, bookings, reports, profiles, and password changes. Each method obtains request input, performs checks, interacts with the database, and returns a view, redirect, or JSON response.

Step 6: Validation and Business Rules

Validation rules protect the correctness of the data. Examples include checking mandatory login fields, verifying that a member exists, preventing duplicate library accounts, checking that a book is active and available, preventing a seat conflict for the same date/time context, and ensuring that fine and return actions reference a valid issue record. Status values are used to distinguish active, issued, returned, booked, paid, unpaid, lost, damaged, or fined states.

Step 7: Integration and Testing

The modules are integrated through shared database relationships and routes. Issue processing updates both the issue table and the available-copy count. Return processing updates the issue and inventory and may create a fine. Dashboards aggregate counts from books, members, issues, bookings, and fines. Integrated workflows are tested with normal and exceptional data to ensure that the sequence of updates is correct.

Step 8: Documentation and Review

The final step is to document the architecture, methodology, requirements, implementation, tests, and results. Diagrams are prepared to show data flow, components, use cases, sequences, and entity relationships. Interface and code snapshots are included to connect the written explanation to the implemented project. The review also identifies security and deployment improvements such as password hashing, environment-based secrets, database migrations, stronger validation, audit logs, and automated tests.

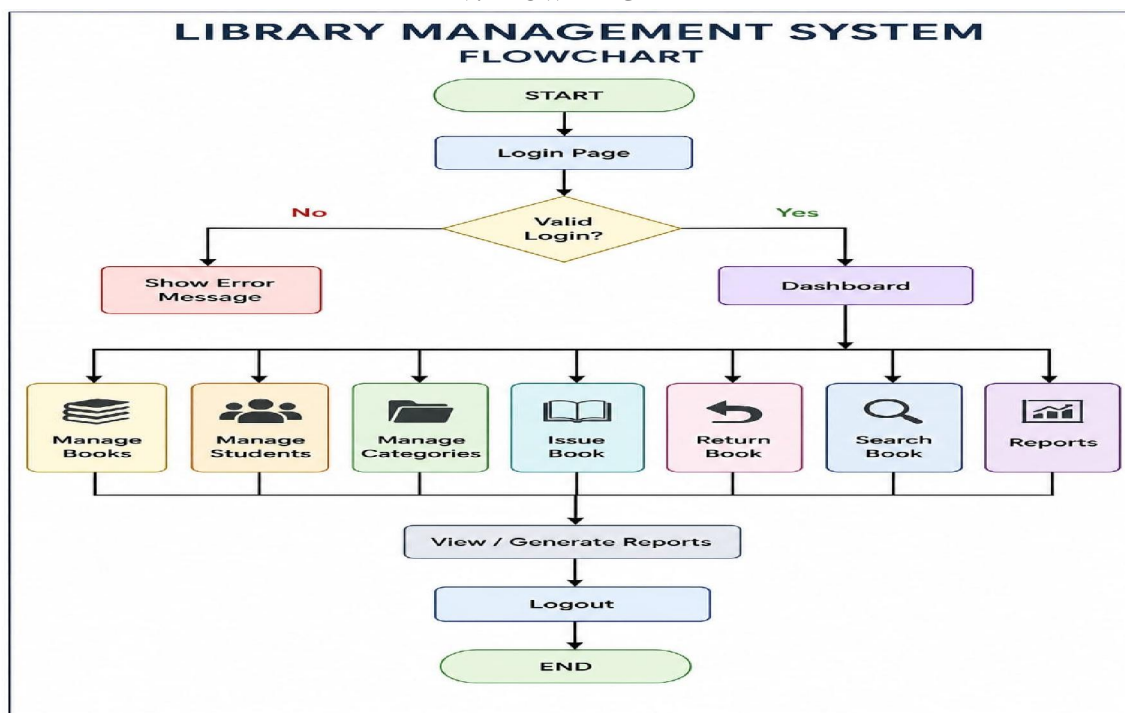


III. MODELING AND ANALYSIS

Table 1. Tools and usage

Technology	Purpose
HTML	Define page structure
CSS	Styling
Bootstrap	Responsive design
JavaScript	Client-side logic
PHP	Server-side logic
MySQL	Database
XAMPP	Local server
Visual Studio Code	Code editing and debugging
Google Chrome	Testing and running the app

IV. FLOW DIAGRAM



V. SNAP SHOTS

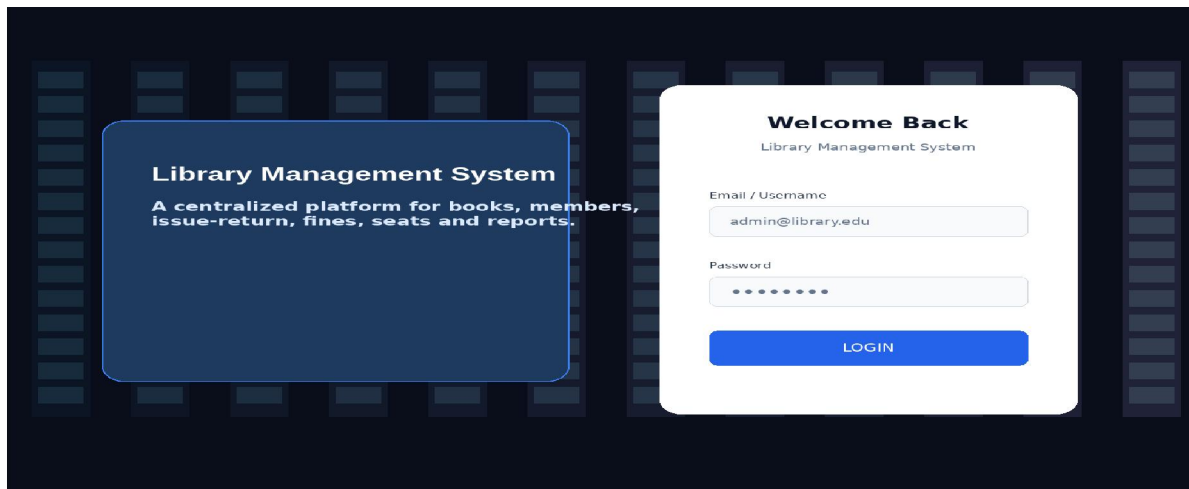


Figure 1: library login Page

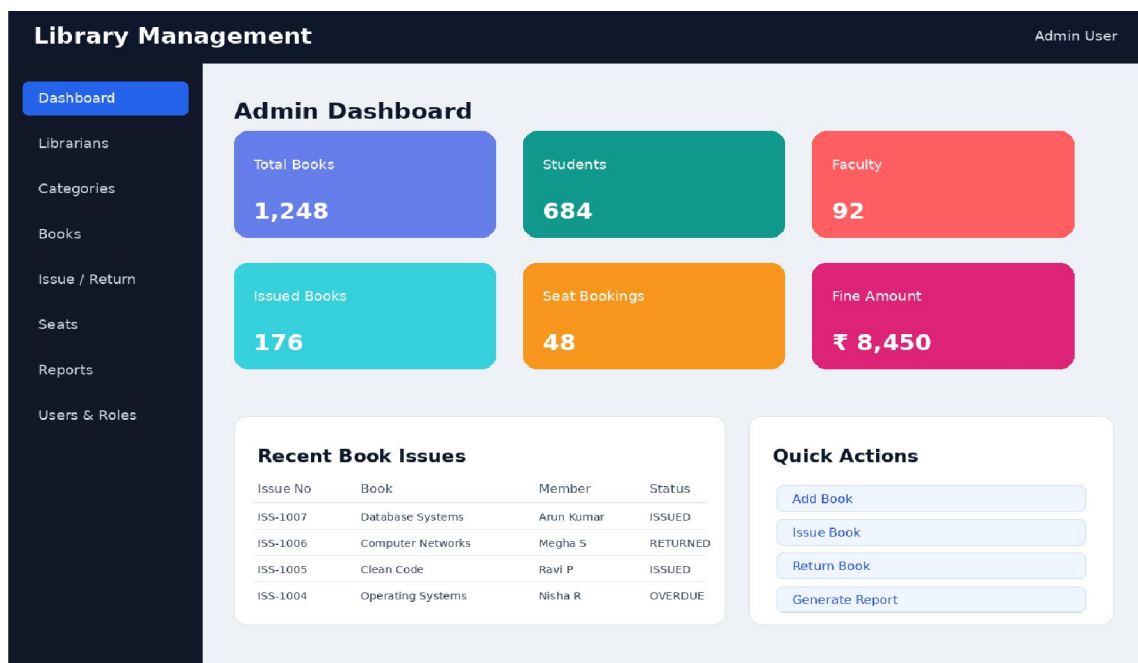


Figure 2: Admin Page



Library Management

Admin User

- Dashboard
- Librarians
- Categories
- Books**
- Issue / Return
- Seats
- Reports
- Users & Roles

Book Catalogue and Inventory

Book List

+ Add Book

Search by title, author or ISBN...

Code	ISBN	Title	Author	Category	Total	Available	Status
BK1001	9780132350884	Clean Code	R. C. Martin	Programming	12	8	ACT Edit
BK1002	9780262033848	Introduction to Algorithms	Cormen et al.	Algorithms	8	5	ACT Edit
BK1003	9780133594140	Database System Concepts	Silberschatz	Database	10	4	ACT Edit
BK1004	9781119456339	Computer Networks	Tanenbaum	Networks	7	2	ACT Edit
BK1005	9781491950357	Designing Data-Intensive Apps	Martin Kleppmann	Architecture	6	6	ACT Edit

Figure 3: Book Catalogue and Inventory

Library Management

Admin User

- Dashboard
- Librarians
- Categories
- Books
- Issue / Return**
- Seats
- Reports
- Users & Roles

Issue Book

Issue Book to Student or Faculty

Member Type

Student

Member ID / Register No.

4JK21CS042

Book

Clean Code — Available: 8

Issue Date

14-07-2026

Due Date

28-07-2026

ISSUE BOOK

RESET

Figure 4: Issue Book Interface



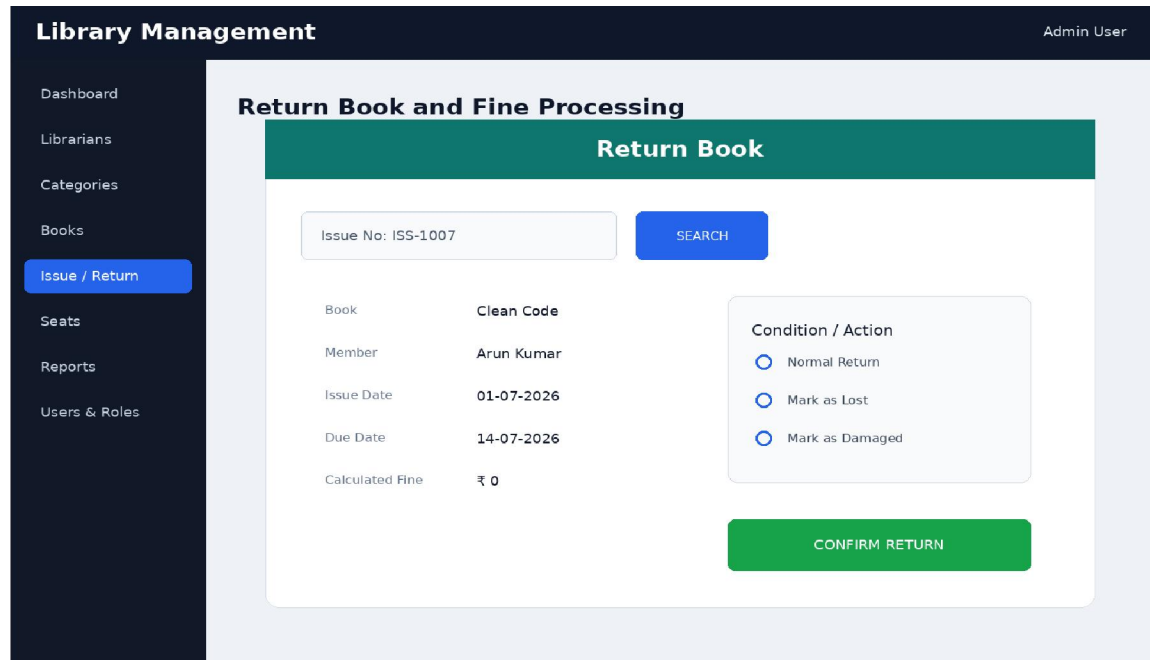


Figure 5: Return and Fine Processing

VI. RESULTS AND DISCUSSION

The Library Management System was designed and tested. The system was designed to perform all the required operations in managing a library automated system. This application was developed to enable the librarian manage books, category, authors, publishers, students records, book issue, book return and fine management in a web based system. The modules were tested and they performed their functions efficiently and the information was stored in the MySQL database effectively.

The login module provides the application with a secured access control system. The application requires a password and a user name to log in to the system. The application allows the librarian to add, update, search and delete any book details from the system. It was also found that the system kept track of the student's records when issuing and returning of the books and it was automated hence saving time and human efforts.

VII. CONCLUSION

The Library Management System demonstrates how the major activities of an academic library can be integrated into a centralized web application. The project covers role-based access, librarian and user administration, book categories and inventory, student and faculty registration, issue and return processing, fines, seats, bookings, profiles, dashboards, reports, email reminders, and an optional assistant. These functions address the primary limitations of manual registers, spreadsheets, and disconnected tools.

The use of PHP, CodeIgniter 4, MySQL, Bootstrap, JavaScript, jQuery, AJAX, and DataTables provides a practical open-source technology base. The MVC-oriented project organization, relational data model, role-specific views, and modular controllers make the application understandable and extendable. The design diagrams and requirements show how actors, components, data stores, and transactions are connected.



ACKNOWLEDGEMENTS

I express my gratitude to the project guide for their invaluable guidance, and unwavering support throughout the project duration. I'm equally thankful to the faculties of MIT Mysore further insightful suggestion and motivation which makes well experience in my work. I extend my sincere appreciation to the management of the institute for providing the necessary resources, that helps in achieving successful project. I'm deeply grateful to my classmates and friends for their coordination, assistance and encouragement during the process of development and I finally, acknowledged with gratitude the contribution of all those directly or indirectly supported in upbringing the project.

REFERENCES

- [1] T. W. Araya and A. Mengsteab, "Designing web-based library management system," *International Journal of Engineering Research & Technology*, vol. 9, no. 10, pp. 272–277, 2020.
- [2] R. Achary, "Performance analysis of RFID and IoT-based smart library management system," in *Proceedings of the 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, pp. 68–72, 2021, doi: 10.1109/ICOSEC51865.2021.9591890.
- [3] M. R. Maepa and M. N. Moeti, "IoT-based smart library seat occupancy and reservation system using RFID and FSR technologies for South African universities of technology," in *Proceedings of the International Conference on Artificial Intelligence and Its Applications (icARTi '21)*, Art. no. 10, pp. 1–8, 2021, doi: 10.1145/3487923.3487933.
- [4] M. Asim, M. Arif, and M. Rafiq, "Applications of Internet of Things in university libraries of Pakistan: An empirical investigation," *The Journal of Academic Librarianship*, vol. 48, no. 6, Art. no. 102613, 2022, doi: 10.1016/j.acalib.2022.102613.
- [5] T. Gopalakrishnan, "Cloud-based library management and book tracking through the Internet of Things," *Journal of IoT in Social, Mobile, Analytics, and Cloud*, vol. 4, no. 4, pp. 234–243, 2022, doi: 10.36548/jismac.2022.4.001.

