

A Web-Based Library Management System Using MySQL

1st Rohit Sudhir Sukale

*Department of Computer Engineering
Vishwakarma Institute Of Technology*

Pune, India

rohit.sukale25@vit.edu

2nd Samruddhi Sunil Bate

*Department of Computer Engineering
Vishwakarma Institute Of Technology*

Pune, India

samruddhi.bate25@vit.edu

3rd Tejas Eknath Naik

*Department of Computer Engineering
Vishwakarma Institute Of Technology*

Pune, India

tejas.naik25@vit.edu

4th Aditya Santosh Galande

*Department of Computer Engineering
Vishwakarma Institute Of Technology*

Pune, India

aditya.galande25@vit.edu

5th Prof. Snehal Khajurji

*Department of Computer Engineering
Vishwakarma Institute Of Technology*

Pune, India

snehal.khajurji@vit.edu

Abstract— This project presents a comprehensive web-based Library Management System (LMS) developed to optimize and automate the core functions of academic libraries. Essential tasks like book cataloging, book issuance, return processing, student record handling, due-date tracking, and fine calculation are all streamlined and digitized by the system. The LMS lowers the risk of human error, speeds up daily operations, and guarantees accurate, current records by substituting traditional manual procedures with an integrated software platform.

The architecture is divided into three layers: a secure data layer, a robust application layer, and a responsive presentation layer. The frontend interface is implemented using HTML5, Tailwind CSS, and JavaScript, ensuring a highly interactive and mobile-friendly user experience. The backend is powered by Node.js and Express.js, which provide scalable API endpoints for processing user requests, validating inputs, performing fine calculations, and managing system logic. The data layer utilizes MySQL, a relational database that ensures data consistency through normalization, foreign key constraints, and ACID-compliant transactions.

When combined, these elements create a system that improves operational effectiveness, facilitates real-time book availability monitoring, and offers a dependable digital substitute for manual record-keeping. In addition to lessening the workload for librarians, the LMS enhances transparency, preserves data integrity, and speeds up information retrieval. As a result, the system significantly improves the overall management, accessibility, and productivity of library services within academic institutions.

I. INTRODUCTION

Traditional library management practices rely heavily on manual entries, physical registers, and paper-based tracking of books. These traditional approaches are frequently slow, ineffective, and prone to human error, which can result in lost documents, erroneous availability status, and delays in producing crucial reports. As the volume of books and student transactions increases, managing large collections becomes

more challenging, resulting in operational irregularities and lower service quality

To address these challenges, this project introduces a fully digital and automated web-based Library Management System that centralizes core operations—such as managing books and student records, issuing and returning books with automatic fine calculation, generating detailed reports, and tracking real-time availability—into a modular, user-friendly platform suitable for schools, colleges, and educational institutions seeking efficient and transparent library automation

II. RESEARCH GAP IN EXISTING SYSTEMS

Despite the fact that library software is used by many institutions, the majority of current systems still have a number of drawbacks. Many older systems rely on outdated interfaces, offer limited automation, or lack proper integration between book management, issuing, returning, and reporting. The lack of real-time book availability tracking is a major gap that makes it more difficult for students and librarians to know the precise status of books at any given time.

Another major issue is that many existing solutions do not include an automatic fine calculation mechanism, forcing librarians to manually check due dates and compute penalties—an error-prone and time-consuming process. Several systems also fail to provide complete and meaningful reports, such as overdue lists, return history, and student-wise issue summaries, which are essential for decision-making.

Furthermore, many older systems are challenging to use on various devices, such as laptops, tablets, or mobile phones, because they lack a contemporary, responsive, and user-friendly interface. Security is another concern, as some traditional systems do not offer proper authentication, role management, or data validation, which can lead to unauthorized access or inaccurate entries.

These gaps highlight the need for a modern, fully automated, web-based Library Management System that ensures better usability, real-time monitoring, strong data accuracy, and smooth management of all library-related operations. This project aims to deliver that.

III. LITERATURE REVIEW

Library Management Systems have changed over time, shifting from traditional paper registers to digital tools that aim to improve efficiency and accuracy. Several studies and implementations show a clear need for automation, particularly in managing book circulation, cataloging, and student records.

Early systems mainly focused on basic catalog storage. They provided simple features for adding books and keeping member lists. However, these systems missed key automation functions like calculating fines, generating detailed reports, and checking real-time availability. Research into web-based systems brought some improvements by allowing online access, but many platforms still struggled to create smooth workflows for issuing and returning books.

Recent studies have examined RFID-based and IoT-enhanced library automation, which offer potential for better tracking. Yet, these technologies often come with high costs and complicated setups, making them less viable for small or medium educational institutions. Research on digital library platforms indicates that user experience is vital. Outdated interfaces and complicated navigation often discourage librarians from fully adopting available systems.

Several academic papers stress the need for centralized database management to ensure accuracy, avoid data duplication, and maintain consistent transaction records. MySQL-based systems have shown to be effective because of their reliability, structured data management, and support for multiple users.

Overall, the literature reveals a clear gap. Most existing solutions either lack sufficient automation, have poor user interfaces, or fail to integrate all essential library functions into a single, accessible platform. This project aims to tackle these issues by creating a modern, responsive, and fully automated system using Node.js, MySQL, and a clean frontend interface to enhance usability, efficiency, and transparency in library operations.

IV. RESEARCH METHODOLOGY

The research method used in this project is a structured and repetitive process. It ensures that we create a reliable, efficient, and user-friendly Library Management System. This method includes requirement analysis, system design, implementation, and testing. It allows for constant improvement at every stage.

1. Problem Identification and Requirement Analysis

This stage involves examining the limitations of manual library processes and gathering detailed requirements from librarians and users. This helps clarify what the digital system needs to accomplish.

2. System Design and Architecture Planning

After finalizing the requirements, a structured three-tier architecture is planned. This outlines how the user interface, backend services, and MySQL database will work together to ensure smooth and reliable operations.

3. Database Modeling

A properly organized MySQL database is designed with logical relationships between tables. This ensures that all book, student, and transaction data stays accurate, consistent, and easy to retrieve.

4. Development and Module Implementation

The system is built by adding each feature, such as managing books and students, issuing and returning books, calculating fines, and generating reports, according to the planned architecture and workflow.

5. Testing and Validation

Once development is complete, the system undergoes multiple rounds of testing. This verifies that each module works correctly, data flows smoothly across components, and users can perform all library tasks without errors.

6. Deployment and Evaluation

The final system is deployed in a real or simulated environment and assessed based on performance, usability, accuracy, and its overall effectiveness in addressing the problems identified during the initial research.

We deployed the final system on a local server and evaluated it based on accuracy, speed, ease of use, and reliability. The results showed significant improvements over manual processes.

7. Documentation & Final Reporting

The final stage is preparing clear and organized documentation. This documentation explains the system's design, workflow, implementation details, and results.

SYSTEM DESIGN

The design of the Library Management System aims to create a smooth, efficient, and user-friendly platform that digitizes all main library operations. It begins by identifying how librarians interact with books, students, and daily transactions. Then, it translates these needs into a clear flow of processes. The goal is to make the digital system easy to navigate and reduce the complexity of traditional manual tasks.

The overall architecture follows a three-tier structure. This separates the system into the presentation layer, business logic layer, and data layer. This separation helps maintain clarity, improves modularity, and allows each layer to operate independently while interacting smoothly with the others. This setup not only makes the system easier to maintain but also improves scalability for future upgrades and new features.

The frontend design is crucial for user experience. It is built with HTML, Tailwind CSS, and JavaScript to provide a clean, responsive interface that functions well on different devices. Librarians can easily access modules like Books, Students, Issue Book, Return Book, Reports, and Dashboard. The design also ensures that all pages load quickly and display information in a clear and organized way.

The backend design uses Node.js and Express.js to manage data flow, handle user actions, and perform all essential operations. This includes validating inputs, issuing and returning books, calculating fines, updating availability, and generating real-time reports. The backend structure relies on modular routing, keeping each operation separate and preventing the system from becoming complex or hard to maintain.

The database design focuses on MySQL, where all data is stored in structured tables like Books, Students, IssuedBooks, and ReturnRecords. The tables are linked through primary and foreign keys to keep records accurate and consistent. This structure ensures that data remains organized, easy to retrieve, and protected from duplication or corruption. Proper indexing makes searches fast and efficient.

Finally, the system design includes key supporting elements such as secure authentication, error handling, rules for calculating fines, and constraints to ensure smooth operations. Each module is designed to work independently but still connect with other modules. This results in a reliable, flexible, and easy-to-manage Library Management System suitable for real-world use in educational institutions.

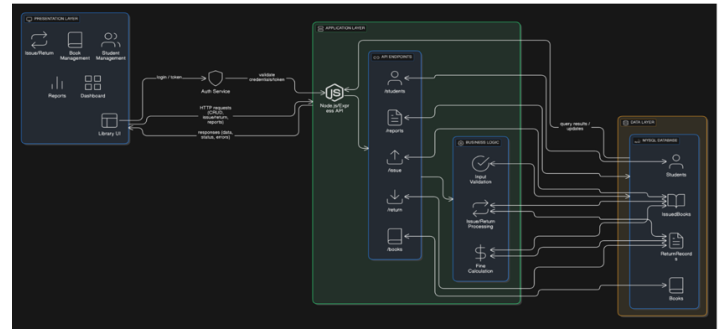


Fig. no 1. System Design

IMPLEMENTATION

The Library Management System was implemented using a modular and systematic approach. This ensured that each part worked efficiently and integrated well with the others. The development process started with setting up the project environment. Node.js and Express.js were configured for backend operations, and the MySQL database was prepared for structured data storage. This initial setup laid the groundwork for building all the features that followed and ensured that the system could handle real-time data processing.

The frontend was created with HTML, Tailwind CSS, and JavaScript, resulting in a clean, modern, and responsive user interface. Each page, including the Dashboard, Books Management, Students Management, Issue Book, Return Book, and Reports, was designed to help librarians navigate with ease. Tailwind CSS helped maintain a consistent look, while JavaScript allowed for smooth interactions like loading dynamic content, validating forms, and updating tables.

For the backend, Node.js and Express.js were used. Each feature was developed as a separate API route, which improved organization and maintainability. These routes handled essential tasks, such as adding and updating book records, storing student information, issuing and returning books, calculating fines, and generating various reports. The backend also included error handling and validation to avoid incorrect inputs and maintain data reliability.

The MySQL database consisted of four main tables: Books, Students, IssuedBooks, and ReturnRecords. Each table had appropriate primary keys, foreign keys, and constraints to keep accurate relationships between the data. SQL queries supported tasks like searching, issuing, returning, and updating availability. The database was optimized with indexing to ensure faster access during operations like searching for books or tracking student records.

Integration among the frontend, backend, and database was achieved using asynchronous API calls. This enabled real-time updates across all modules.

To ensure system correctness, multiple testing phases were included. Each module was tested independently before being tested again as part of the complete system. The final implementation successfully delivered a functional, automated, and user-friendly Library Management System that supports daily operations in schools and colleges with improved speed, accuracy, and convenience.

DATABASE DESIGN: TABLES OVERVIEW

Table 1: Books

Field	Type	Description
book_id (PK)	int	Unique ID for each book, auto-generated.
title	varchar(150)	Name or title of the book.
author_name	varchar(100)	Author of the book.
genre	varchar(50)	Category or genre of the book.
total_copies	int	Total number of copies the library owns.
available_copies	int	Copies currently available for issuing.

Explanation:

This table stores complete details about every book in the library. It helps track how many copies exist and how many are available for issuing in real time.

Table 2: Students

Field	Type	Description
student_id (PK)	int	Unique ID assigned to each student.
student_name	varchar(100)	Full name of the student.
department	varchar(50)	Department or branch of the student.
year	int	Year of study (e.g., 1st, 2nd, 3rd).
contact_no	varchar(15)	Student’s mobile number.
email	varchar(100)	Unique email ID for communication.

Explanation:

This table stores information about students who borrow books. It helps the system identify which student issued or returned a book.

Table 3: IssuedBooks

Field	Type	Description
issue_id (PK)	int	Unique ID for each issue transaction.

Field	Type	Description
student_id (FK)	int	Links to the Students table.
book_id (FK)	int	Links to the Books table.
issue_date	date	Date on which the book was issued.
due_date	date	Date by which the book must be returned.
return_date	date	Date on which the book was returned (if any).
status	varchar(20)	Current status: ‘Issued’ or ‘Returned’.

Explanation:

This table records every book issue transaction. It helps track overdue books, due dates, and whether a book is returned or still issued.

Table 4: ReturnRecords

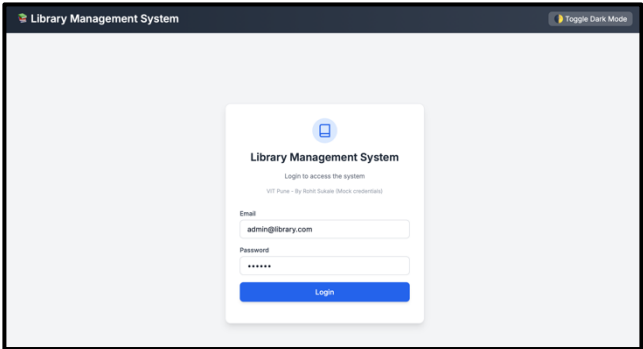
Field	Type	Description
return_id (PK)	int	Unique ID for each return entry.
issue_id	int	References the issue transaction.
return_date	date	The date the book was actually returned.
fine_amount	decimal(10,2)	Fine calculated based on delay (if applicable).

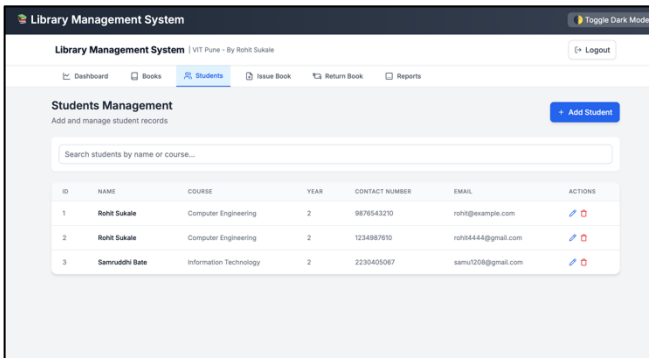
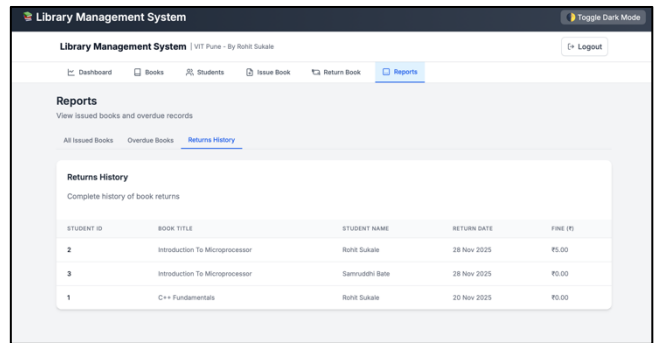
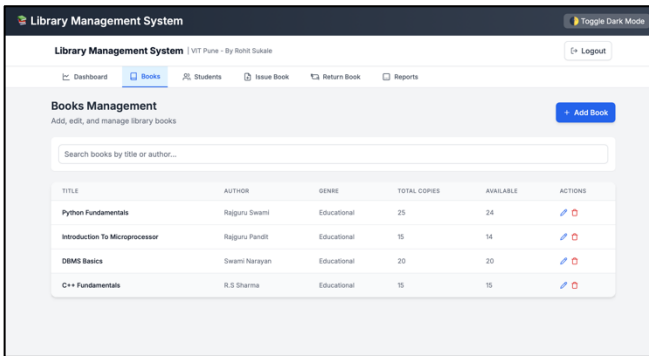
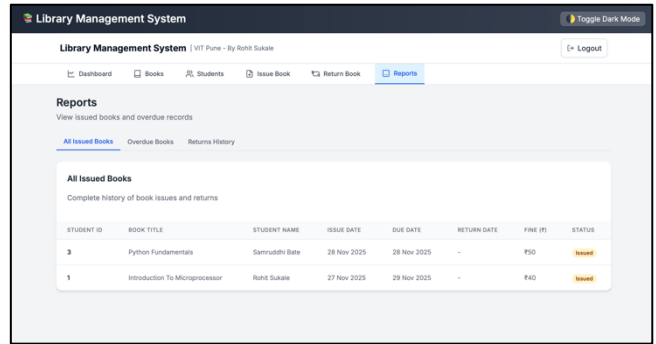
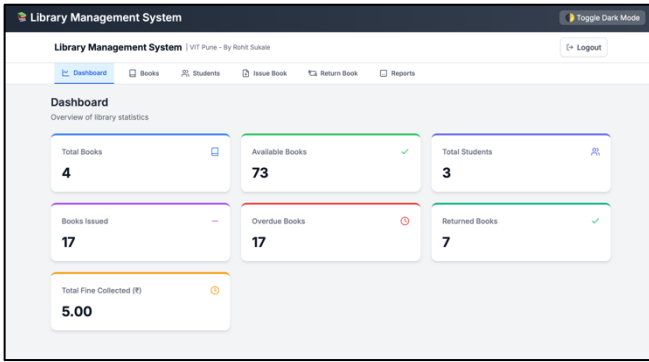
Explanation:

This table maintains the return history and fine details. It helps generate reports on fines, returned books, and overall return activity.

D. EVALUATION AND VALIDATION

In comparison to manual allocation, the accuracy, fairness, efficiency, and user satisfaction of the suggested AI-Based Smart Allocation Engine were assessed.

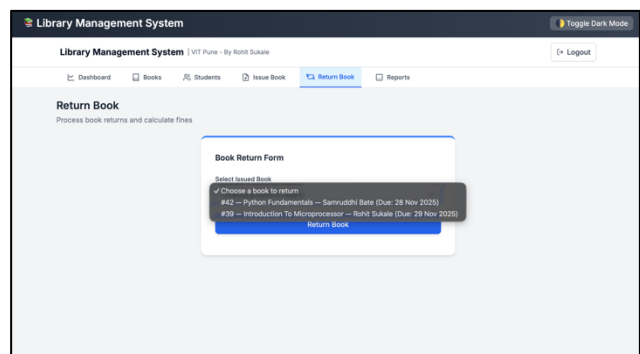
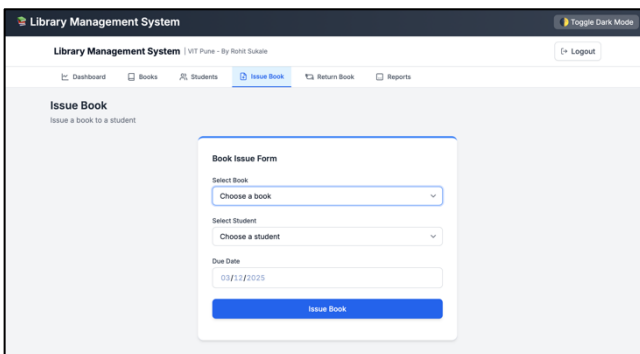




E. CONCLUSION AND FUTURE WORK

The development of this web-based Library Management System effectively tackles the main challenges faced by traditional library operations. By combining a modern frontend, a strong Node.js backend, and a well-organized MySQL database, the system offers a smooth and reliable way to manage books, students, issuing, returning, and calculating fines. The automated workflows greatly reduce human error, improve data accuracy, and save valuable time for librarians. Features like real-time book availability, detailed reporting, and an easy-to-use interface make the system more efficient and user-friendly, ultimately boosting productivity in library management at educational institutions.

While the current system meets the basic needs of daily library activities, there are several areas to improve it further. Future work may include adding QR code or barcode scanning for quicker book issuance and returns, creating a student login portal so users can track their borrowed books and due dates, and enabling online book reservations. Additional improvements, such as showing book cover images, generating downloadable PDF reports, sending automated email or SMS reminders, and adding analytics dashboards for insights into library usage, can further enhance the system. These changes would help make the system a more intelligent and flexible platform suitable for larger institutions.



ACKNOWLEDGMENT

I want to thank everyone who supported and guided me during this project. I am especially grateful to my project mentor and faculty members for their helpful feedback, encouragement, and technical advice. Their input helped me refine the system and improve its overall quality. I also appreciate my classmates and friends for their useful suggestions and for helping me test the application at different stages of development. Lastly, I am thankful to my institution for providing the resources and environment that made this project possible.

REFERENCES

- [1] Web-Based Library Management System – IJERT
<https://www.ijert.org/designing-web-based-library-management-system>
- [2] Modern Library Management System – ResearchGate
https://www.researchgate.net/publication/391232217_MODERN_LIBRARY_MANAGEMENT_SYSTEM
- [3] Library Management System Using Web Technologies – IJRPR
<https://ijrpr.com/uploads/V6ISSUE8/IJRPR52231.pdf>
- [4] Library Automation & Management – IJARCS
<https://ijarcs.info/index.php/Ijarcs/article/view/4478>
- [5] Library Management System – IJCSMC
<https://www.ijcsmc.com/docs/papers/March2017/V6I3201712.pdf>
- [6] Design & Implementation of Library Management System
https://www.researchgate.net/publication/321746127_Design_and_Implementation_of_Library_Management_System
- [7] A Review on Library Management Systems
<https://www.ijert.org/research/a-review-on-library-management-system-IJERTV9IS040592.pdf>
- [8] Web-Based Library Automation System – Academia
https://www.academia.edu/83773348/Web_Based_Library_Automation_System
- [9] Node.js Official Documentation
<https://nodejs.org/en/docs>
- [10] Express.js REST API Guide
<https://expressjs.com/en/guide/routing.html>
- [11] MySQL Documentation (Database Engine, Constraints, Schema Design)
<https://dev.mysql.com/doc/>
- [12] REST API Best Practices – Microsoft
<https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design>
- [13] Tailwind CSS Official Docs
<https://tailwindcss.com/docs>
- [14] MDN Web Docs – JavaScript
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [15] HTML5 Standard Documentation
<https://html.spec.whatwg.org/>