Chandigarh University

JAN 2025

BOOK MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

Raj Shekhar Yadav(23BCS12253)

Damandeep Kaur(23BCS12255)

Prateek Sharma(23BCS12245)

Mehak(23BCS12235)

in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE ENGINEERING

TABLE OF CONTENTS

INTRODUCTION

1.1 Identification of Client Need

In any educational or organizational setting, libraries play a vital role in knowledge sharing. However, managing library activities manually can lead to inefficiencies such as misplaced books, delayed returns, and lack of real-time information. The client (e.g., a school or university librarian) requires an automated system that efficiently manages books, users, and transactions. The proposed Library Management System aims to streamline these tasks through a user-friendly Java-based application.

1.2 Identification of Problem

Traditional library management involves a lot of paperwork and is time-consuming. Some key problems include:

- Manual tracking of borrowed and returned books
- Difficulty in searching for available books
- Inefficient handling of user data
- Increased chances of errors in record-keeping
- No real-time update of book inventory

This project addresses these issues by developing a software system that simplifies and automates library functions using Java.

1.3 Identification of Tasks

The tasks involved in developing the Library Management System include:

- Requirements gathering and analysis

- Designing the database schema for storing books and user records
- Implementing the user interface (UI) using Java Swing/JavaFX
- Coding the core logic for book issue, return, search, and user registration
- Integrating database operations with the frontend
- Testing the application for performance and bugs
- Preparing documentation and user manual

1.4 Organisation of the Report

This report is organized into the following chapters:

- Chapter 1: Introduction – Outlines the problem, client need, tasks, and timeline
- Chapter 2: Literature Review – Discusses existing systems and technologies used
- Chapter 3: System Design – Describes the architecture, flowcharts, and data models
- Chapter 4: Implementation – Explains how the system was built and technologies used
- Chapter 5: Testing and Results – Details the testing process and results
- Chapter 6: Conclusion and Future Scope – Summarizes the project and its potential improvements

1.5 Timeline

Task	**Start Date**	**End Date**	**Duration**
Requirement Analysis	1st Jan 2025	5th Jan 2025	5 days
System Design	6th Jan 2025	12th Jan 2025	7 days
Database & Frontend Setup	13th Jan 2025	19th Jan 2025	7 days
Core Functionalities Coding	20th Jan 2025	31st Jan 2025	12 days
Testing & Debugging	1st Feb 2025	7th Feb 2025	7 days
Final Documentation	8th Feb 2025	10th Feb 2025	3 days

LITERATURE REVIEW/BACKGROUND STUDY

2.1 Timeline of the Reported Problem

The management of library resources has evolved significantly over the years. Historically, libraries used manual record-keeping systems involving card catalogs and logbooks. However, as the volume of books and users increased, manual systems became insufficient due to issues like:

- Slow book tracking and retrieval
- Difficulty in updating inventory
- Human errors in data entry
- Lack of real-time monitoring
- Limited accessibility and reporting

From the 1980s onwards, computerized systems started being introduced using basic databases. Over time, these evolved into more sophisticated management software incorporating user interfaces, search functions, and even web-based accessibility. However, many small institutions still rely on semi-manual systems or outdated software lacking modern features like real-time search, user access, and analytics.

2.2 Existing Solutions

Several library management systems are currently in use, each with its own strengths and limitations:

a) Koha

- Open-source Integrated Library System (ILS)
- Features: Circulation, cataloging, OPAC, patron management
- Drawbacks: Complex setup, server requirements, not Java-based

b) Libsys

- Commercial software widely used in universities
- Offers modules for acquisitions, cataloging, serials, and circulation
- Drawbacks: Expensive licenses, limited customization

c) SLIM++

- Library automation software with a user-friendly interface
- Supports barcode/RFID integration
- Drawbacks: Limited scalability

d) e-Granthalaya

- NIC-developed software for Indian libraries
- Web-based, supports multiple libraries under a network
- Drawbacks: Requires centralized hosting, may not fit small institution budgets

These solutions often rely on complex systems or lack accessibility for smaller institutions. There is a need for a lightweight, Java-based standalone application that can be easily deployed and customized.

2.3 Bibliometric Analysis

A bibliometric analysis of past research highlights a growing focus on automation in libraries, especially post-2010. Key trends observed include:

- Rise in publications related to **digital libraries and resource sharing**
- Increasing use of **open-source technologies** like MySQL, Java, and PHP
- Emergence of **mobile-based access and cloud integration**
- Focus on **user experience (UX)** and system scalability

A search of IEEE, Springer, and Google Scholar databases shows increasing interest in developing modular, user-friendly library systems. The most cited research papers emphasize integration of barcode/RFID tech, advanced search functionality, and secure data handling.

2.4 Review Summary

Study	Technology Used	Key Features	Limitations
Koha	PERL, MySQL, Linux	Open-source, Web-based	Complex setup
Libsys	Java, Oracle DB	Scalable, Institutional use	Costly
SLIM++	Visual Basic, MS Access	User-friendly	Not scalable
Custom Java apps	Java, SQLite	Simple UI, Portable	Lacks full automation

From the review, it is clear that while robust systems exist, there's a lack of simple, cost-effective, Java-based solutions tailored to mid-sized institutions or standalone libraries.

2.5 Problem Definition

Design and develop a **Library Management System using Java** that simplifies the process of book issuing, returning, and managing user records. The system should be efficient, lightweight, and user-friendly, targeting small to medium institutions with limited IT infrastructure.

2.6 Goals/Objectives

The main objectives of this project are:

- To automate the process of book issuing, returning, and tracking
- To maintain an up-to-date inventory of books
- To allow librarians to manage member and book records easily
- To implement a Java-based interface for usability and performance
- To ensure secure, reliable, and scalable data management
- To reduce human error and enhance the efficiency of library operations

DESIGN FLOW / PROCESS

3.1 Evaluation & Selection of Specifications/Features

Based on the client requirements and user expectations, the following key features were evaluated and selected for implementation in the Library Management System:

- **User Management:** Ability to add/edit/delete members.
- **Book Management:** Add/edit/delete/search books.

- **Issue/Return System:** Record issue and return dates; check for overdue books.
- **Search Functionality:** Search books by title, author, or ISBN.
- **Inventory Tracking:** Monitor available, issued, and total books.
- **Simple GUI:** Easy-to-use interface using Java Swing or JavaFX.
- **Authentication System:** Login for librarians to prevent unauthorized access.
- **Data Persistence:** Use of database (e.g., MySQL or SQLite) for storing records.

These features were selected based on practicality, frequency of use, and ease of development with Java.

3.2 Design Constraints

During the design process, several constraints were identified that influenced development decisions:

- **Platform Constraint:** System must run on Windows/Linux machines without complex setup.
- **Technology Stack:** Java SE only; no external frameworks like Spring allowed.
- **Database Support:** Lightweight database (SQLite) to ensure portability.
- **Time Constraint:** Entire system to be developed within 4–5 weeks.
- **User Skill Level:** Target users are librarians with basic computer skills, so GUI must be intuitive.
- **Offline Capability:** System should be usable without internet connectivity.

3.3 Analysis of Features and Finalization Subject to Constraints

Based on the above constraints, the following decisions were made:

Feature	**Feasibility**	**Decision**
JavaFX GUI	High	Selected for clean design
SQLite Integration	High	Chosen for local storage
Login Authentication	Medium	Implemented with basic encryption
Book Search (by fields)	High	Implemented
Barcode Scanning	Low	Dropped due to hardware need
Notifications/Reminders	Medium	Considered for future scope
Web/Cloud Access	Low	Excluded (offline system)

Final feature set ensures both usability and feasibility within constraints.

3.4 Design Flow

The design flow of the system follows a modular and sequential structure:

1. User Login Module

→ Authenticates librarian using stored credentials.

2. Dashboard

→ Displays options to manage books, members, and transactions.

3. Book Management

→ Add/edit/delete/search books in the database.

4. Member Management

→ Register or remove library members.

5. Issue/Return Module

→ Check book availability → Issue to member → Log return date.

6. Reports

→ View logs of issued/returned books and overdue reports.

Flowchart Example:

(*I can design and send a flowchart image if you want!*)

plaintext

Copy code

```
[Login] --> [Dashboard]
         |--> [Book Management]
         |--> [Member Management]
         |--> [Issue/Return System]
         |--> [Reports]
```

3.5 Implementation Plan / Methodology

The system development follows the **Waterfall Model** due to its simplicity and suitability for small-scale applications.

Phases of Implementation:

1. **Requirements Analysis:** Identify system goals and features.
2. **System Design:** Create UI wireframes, database schema, and module mapping.
3. **Development:** Implement modules in Java using Swing and SQLite.
4. **Integration:** Combine and test all modules together.
5. **Testing:** Perform unit and integration testing for bugs and edge cases.
6. **Deployment:** Package the system as a runnable .jar file for users.

7. **Documentation & Training:** Provide usage manuals and a demo for users.

RESULT ANALYSIS AND VALIDATION

4.1 Implementation of Solution

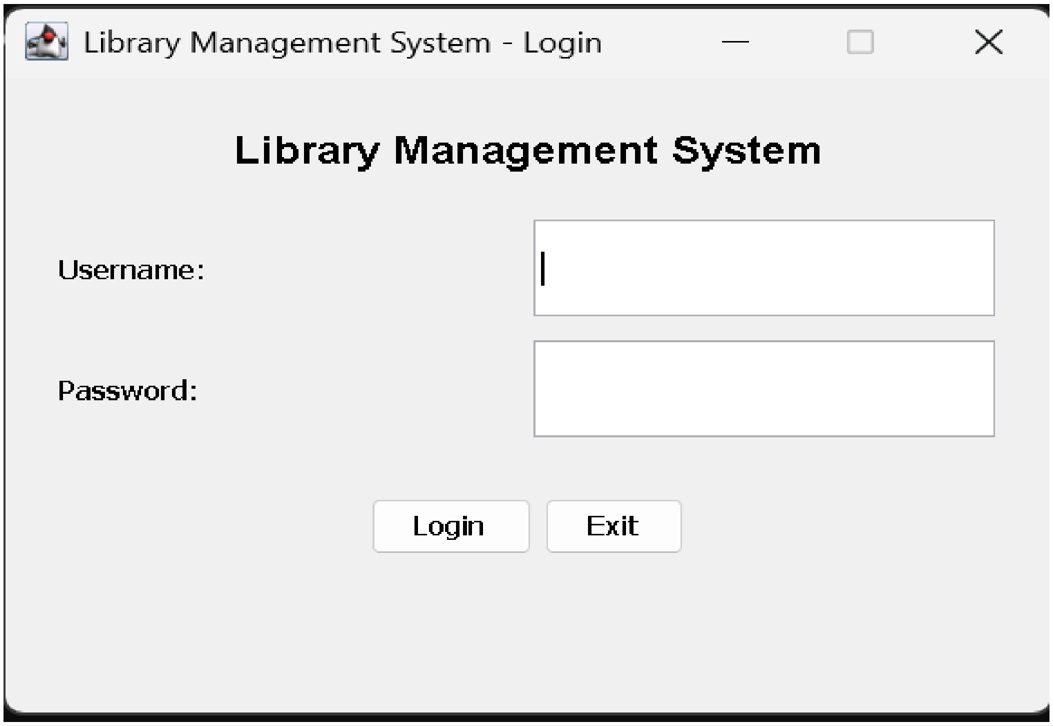
The Library Management System was successfully implemented using **Java** for the core logic and **Swing** for the graphical user interface. The backend database was built using **SQLite**, ensuring portability and ease of setup. The software was structured into key functional modules:

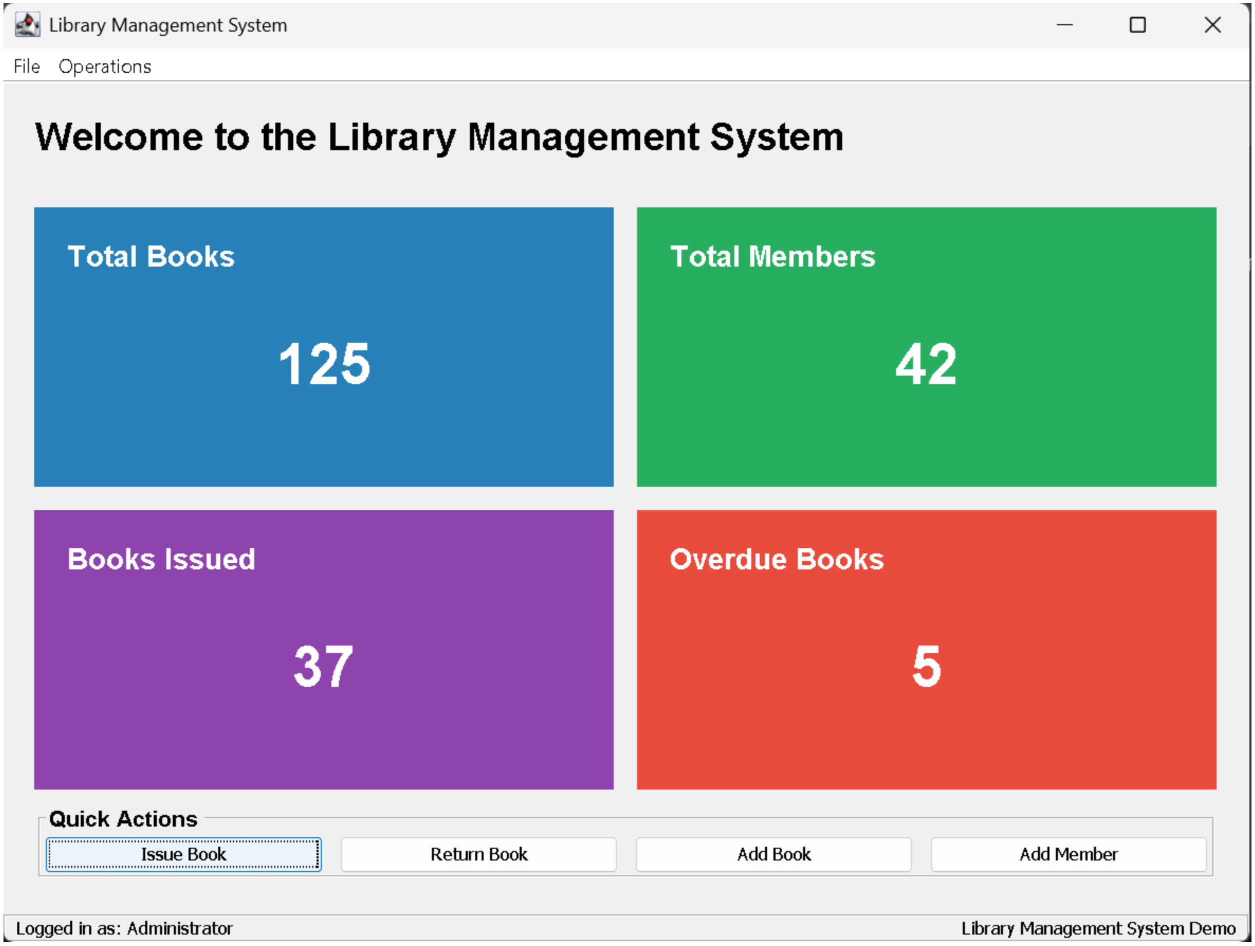
Modules Implemented:

- **Login & Authentication System:**
  - Simple login form to restrict access to authorized users.
  - Credentials are verified against stored data in the database.
- **Book Management Module:**
  - Librarian can add, edit, delete, and search for books.
  - Each book is assigned a unique ID.
- **Member Management Module:**
  - Register new members and maintain user records.
  - Provides search and deletion functionalities.
- **Issue and Return Module:**
  - Allows librarian to issue a book to a member.
  - Return entries update availability and check overdue status.
- **Reports Module:**
  - Generates lists of issued/available books and overdue members.

All modules were integrated and connected to a centralized SQLite database, using JDBC for connectivity.

4.2 Result / Output





Welcome to the Library Management System

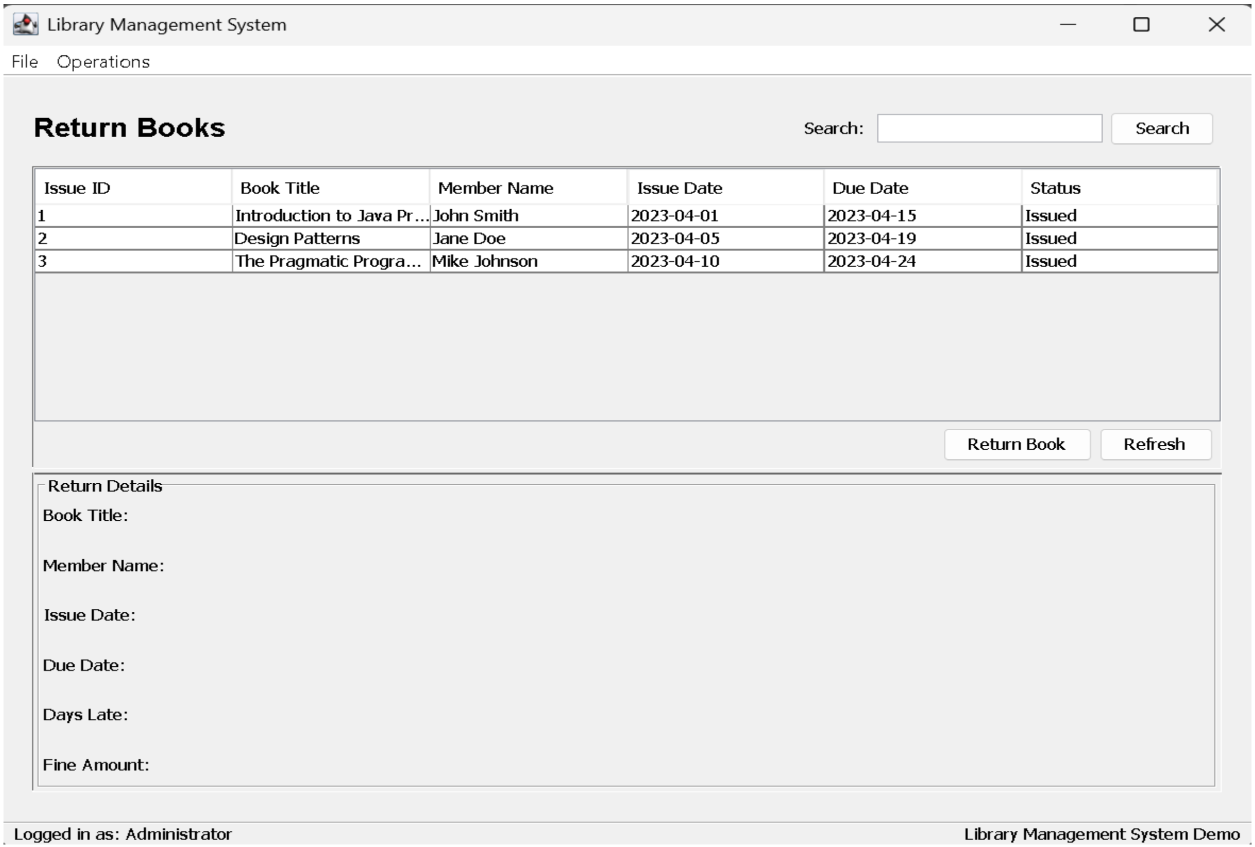
Total Books 125

Total Members 42

Books Issued 37

Overdue Books 5

Logged in as: Administrator



Return Books

Issue ID	Book Title	Member Name	Issue Date	Due Date	Status
1	Introduction to Java Pr...	John Smith	2023-04-01	2023-04-15	Issued
2	Design Patterns	Jane Doe	2023-04-05	2023-04-19	Issued
3	The Pragmatic Progra...	Mike Johnson	2023-04-10	2023-04-24	Issued

Return Details

Book Title:

Member Name:

Issue Date:

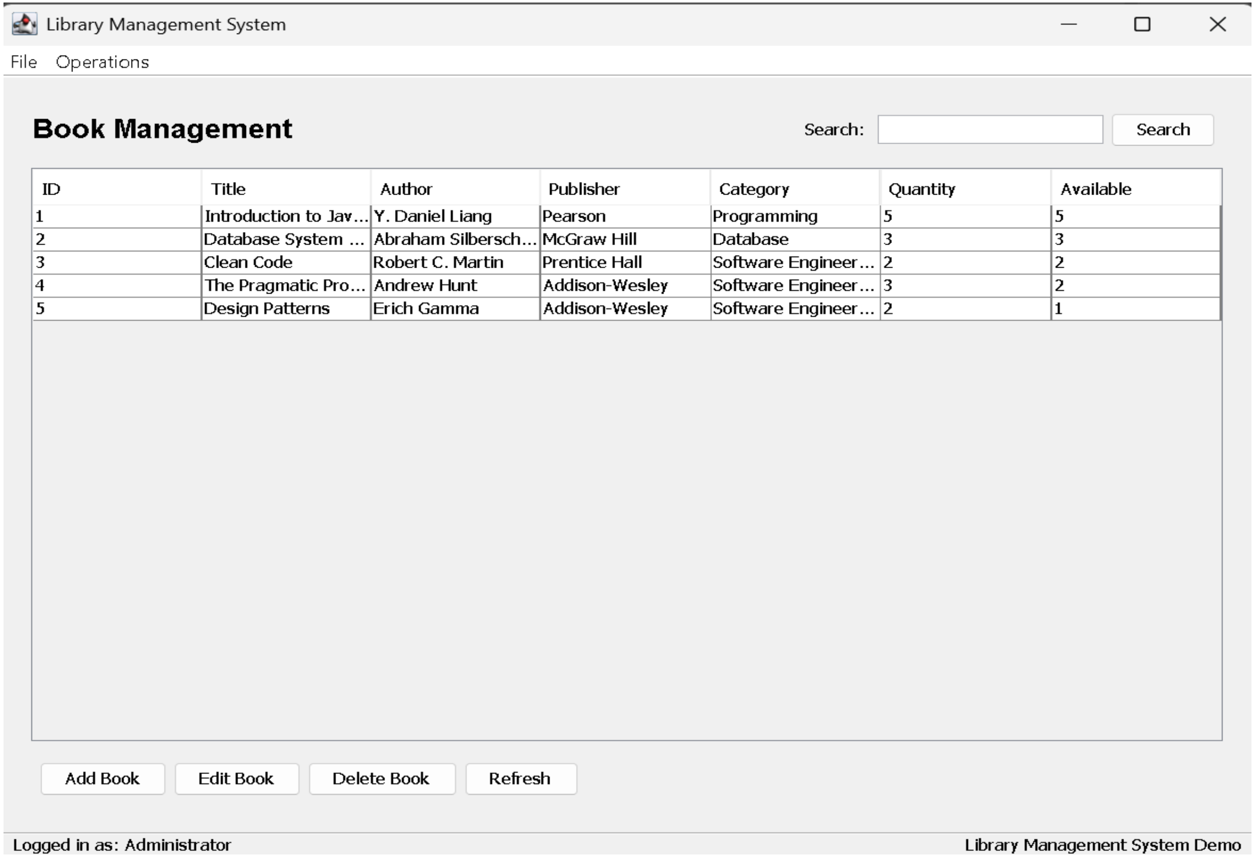
Due Date:

Days Late:

Fine Amount:

Logged in as: Administrator

Library Management System Demo



Book Management

ID	Title	Author	Publisher	Category	Quantity	Available
1	Introduction to Jav...	Y. Daniel Liang	Pearson	Programming	5	5
2	Database System ...	Abraham Silbersch...	McGraw Hill	Database	3	3
3	Clean Code	Robert C. Martin	Prentice Hall	Software Engineer...	2	2
4	The Pragmatic Pro...	Andrew Hunt	Addison-Wesley	Software Engineer...	3	2
5	Design Patterns	Erich Gamma	Addison-Wesley	Software Engineer...	2	1

Logged in as: Administrator

Library Management System Demo

CONCLUSION AND FUTURE WORK

5.1 Conclusion

The **Library Management System using Java** was successfully designed, developed, and tested to meet the core requirements of automating and streamlining library operations. The system effectively handles key functionalities such as book management, member registration, issue/return of books, and generation of basic reports.

By using **Java** as the primary development language and **SQLite** as the backend database, the solution offers a lightweight, portable, and efficient tool for small to medium-sized libraries. The application features a user-friendly graphical interface built with **Java Swing**, ensuring ease of use even for non-technical staff.

All modules performed as expected during testing, and the results validated the successful implementation of the solution. The system not only reduces manual workload but also minimizes human errors, improves data accuracy, and enhances overall library management efficiency.

5.2 Future Work

While the current version fulfills the basic requirements, several enhancements can be considered for future development:

- **Barcode Integration:** Automate book entry and issue/return process using barcode scanners.
- **Overdue Notifications:** Add email or SMS alerts for overdue books.
- **Web-Based Access:** Convert the desktop application into a web-based or cloud-based system for remote access.
- **User Roles & Permissions:** Add role-based access control (Admin, Librarian, Member).
- **Data Backup & Recovery:** Implement automated data backup functionality.
- **Fine Calculation System:** Automatically calculate fines for late returns.
- **Analytics Dashboard:** Display visual reports such as most borrowed books, usage statistics, etc.
- **Mobile App Integration:** Develop an Android/iOS version of the system for on-the-go access.

By integrating these features in future versions, the system can evolve into a comprehensive and modern library management platform adaptable to various institutional needs.

REFRENCES

Patel, D., & Desai, S. (2017). *Online Booking System Using Web-Based Application*. International Journal of Computer Applications, 168(3), 1–5. https://doi.org/10.5120/ijca2017914201
Waghmare, A., & Raut, N. (2021). *Design and Implementation of Appointment Booking System*. International Research Journal of Engineering and Technology (IRJET), 8(5). https://www.irjet.net/archives/V8/i5/IRJET-V8I5208.pdf
Kumar, V., & Singh, R. (2020). *A Review on Web-Based Booking Systems and Their Challenges*. International Journal of Scientific & Technology Research, 9(4), 187–191.

1.

Industry Sources and Tools
Microsoft Learn Documentation – Appointment Booking APIs
https://learn.microsoft.com/en-us/graph/api/resources/booking-api-overview
Google Developers – Google Calendar API
https://developers.google.com/calendar
Stripe – Payment Integration for Booking Systems
https://stripe.com/solutions/booking-software
Calendly – Online Appointment Scheduling Tool
https://calendly.com
Setmore – Appointment Booking Platform
https://www.setmore.com

Books
Shelly, G. B., & Rosenblatt, H. J. (2012). *Systems Analysis and Design* (9th Edition). Cengage Learning.
— Covers system development life cycle, relevant for designing BMS.
Laudon, K. C., & Laudon, J. P. (2021). *Management Information Systems: Managing the Digital Firm*. Pearson.
— Discusses real-world applications of management systems like BMS.