

LIBRARY MANAGEMENT SYSTEM

A Mini Project Report

Submitted by

PUVANESU.R 241901085

SANGATHAMIZHAN.S.P 241901097

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

**COMPUTER SCIENCE AND ENGINEERING(CYBER
SECURITY)**



RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project “**LIBRARY MANAGEMENT SYSTEM**” is the bonafide work of “**PUVANESU.R (241901085) , SANGATHAMIZHAN.S.P (241901097)**” who carried out the project work under my supervision.

Ms. R. Rupmala

ASSISTANT PROFESSOR

Dept. of Cyber Security,

Rajalakshmi Engineering College,

Chennai

This mini project report is submitted for the viva voce examination to be held on

_____.

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the mini project report “**LIBRARY MANAGEMENT SYSTEM**”, submitted as part of the curriculum requirements for the Bachelor of Engineering (B.E) degree affiliated to Anna University, is a bonafide work carried out by us under the supervision of Ms. R. Rupmala, Assistant Professor, Department of Computer Science Engineering and Cyber Security, Rajalakshmi Engineering College, Chennai.

This submission represents our ideas in our own words, and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources.

We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data, idea, fact, or source in our submission. We understand that any violation of the above will be grounds for disciplinary action by the institute and/or the University and may also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been obtained. This report has not previously formed the basis for the award of any degree, diploma, or similar title of any other University.

Rajalakshmi Engineering College,

PUVANESU.R

Chennai

SANGATHAMIZHAN.S.P

November 2025

ABSTRACT

The Library Management System (LMS) is a software application designed to efficiently manage the daily operations of a library. It enables librarians to handle cataloging, user registration, book issue and return, and fine calculation through a centralized digital platform. The primary objective of the system is to replace manual methods with an automated solution that improves accuracy, accessibility, and efficiency. Developed using full stack technologies such as Java, Node.js, and MySQL, this system ensures seamless data flow between the front-end and back-end. By implementing this LMS, libraries can significantly reduce administrative workload, provide faster access to resources, and enhance the overall user experience. This project presents the development of a Library Management System (LMS) using full stack technologies that integrate front-end and back-end functionalities. The system provides core modules for catalog management, user account maintenance, transaction tracking, and advanced search operations. The application leverages Java frameworks for front-end development and Node.js with MySQL for back-end services, ensuring scalability and efficiency. By automating traditional library processes, the LMS promotes data accuracy, operational transparency, and improved service delivery. Future enhancements include AI-based recommendations and mobile accessibility for on-the-go users.

KEYWORDS :

- Library Management System (LMS)
- Full Stack Development
- Java
- Node.js
- MySQL
- Catalog Management
- Book Issue and Return
- Database Management

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M.THANGAM MEGANATHAN** for their timely support and encouragement. We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance.

We express our profound gratitude to **Mr. Benedict J.N.**, Associate Professor (SG) and Head of the Department of Computer Science Engineering and Cyber Security at Rajalakshmi Engineering College, for furnishing us with essential resources, support, and an enabling environment to execute this project.

We also extend our sincere and hearty thanks to our internal guide **Ms. R. Rupmala**, for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

PUVANESU.R

SANGATHAMIZHAN.S.P

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	DECLARATION	I
	ABSTRACT	II
	ACKNOWLEDGEMENT	III
1	INTRODUCTION	1
1.1	INTRODUCTION	1
1.2	SCOPE OF THE WORK	1
1.3	PROBLEM STATEMENT	1
1.4	AIM AND OBJECTIVES OF THE PROJECT	2
2	SYSTEM SPECIFICATIONS	3
2.1	HARDWARE SPECIFICATIONS	3
2.2	SOFTWARE SPECIFICATIONS	3
3	MODULE DESCRIPTION	4
4	SAMPLE CODING	7
4.1	ARCHITECTURE	13
5	SCREENSHOTS	14
6	CONCLUSION AND FUTURE ENHANCEMENT	16
7	REFERENCES	17

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
5.1	ARCHITECTURE	13
5.2	INTRODUCTION PAGE	14
5.3	ADD BOOK	14
5.4	UPDATING BOOK	15

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Library Management System (LMS) is designed to automate and streamline the daily operations of a library. It helps in managing book records, user accounts, and transaction histories effectively. The system ensures that users can easily search for, borrow, and return books while librarians can maintain accurate data on book availability and user activity. This digital approach eliminates manual record-keeping, minimizes human error, and enhances user satisfaction by improving access to library resources.

1.2 SCOPE OF THE WORK

The scope of this project encompasses the automation of major library functions, including catalog management, user registration, book issue/return tracking, and fine calculation. The system allows librarians to manage the library efficiently and provides users with an interactive interface to access library services. It can be implemented in schools, colleges, or public libraries, offering scalability and customization to suit different institutional needs.

1.3 PROBLEM STATEMENT

Traditional library systems rely heavily on manual record-keeping, which often results in errors, redundancy, and delays in book issuance and return. Managing user information and catalog details manually is time-consuming and inefficient. To overcome these issues, the Library Management System provides a centralized, computerized solution to store, organize, and retrieve data quickly, ensuring seamless communication between users and administrators.

1.4 AIM AND OBJECTIVES OF THE PROJECT

The main objective of this project is to develop an integrated system that simplifies and automates library management processes.

The specific objectives are:

- To develop a user-friendly interface for librarians and members.
- To automate cataloging, user registration, and book issue/return activities.
- To maintain an accurate and up-to-date database of library resources and users.
- To improve the efficiency, accuracy, and accessibility of library operations.
- To enable future integration with AI and mobile-based features for better scalability.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Processor	:	Intel i5
Memory Size	:	8GB (Minimum)
HDD	:	1 TB (Minimum)

2.2 SOFTWARE SPECIFICATIONS

Operating System	:	WINDOWS 10 OR ABOVE
Front - End	:	Java(java FX),HTML
Back - End	:	Node.js,MySQL
Language	:	Java,SQL,JavaScript

CHAPTER 3

MODULE DESCRIPTION

The Library Management System (LMS) is organized into two main modules—Admin Module and User Module—to ensure efficient handling of library operations. Each module performs specific functions that contribute to the smooth management of books, users, and transactions.

3.1 ADMIN MODULE

The **Admin Module** is designed for library administrators who manage the overall operations and data within the system. It provides secure access to tools required for maintaining the integrity and accuracy of library records.

Key Features:

1. Admin Login

- Provides secure login authentication using a unique username and password.
- Ensures only authorized personnel can access administrative functions.

2. Book Management

- Add new books to the library database with details such as book ID, title, author, and quantity.
- Update book information when needed.
- Delete outdated or unavailable book records.
- View the entire catalog along with availability status.

3. User Management

- Register new library members and maintain user profiles.
- View user details such as membership ID, contact information, and activity.

4. Transaction Monitoring

- Track book issue and return activities.
- Monitor overdue books and calculate fines.
- Maintain logs for auditing and future reference.

5. Data Integrity and Security

- Ensures all updates to the database are validated and securely executed.
- Prevents unauthorized modifications to book and user records.

3.2 USER MODULE

The **User Module** allows registered members to interact with the library system through a simple and user-friendly interface.

Key Features:

1. User Login

- Members log in securely using their credentials.
- Each user has a personalized dashboard displaying their activity.

2. Book Search & View

- Users can search for books based on title, author, or category.
- View availability of books in real-time.

3. Book Issue

- Submit a request to borrow a book.
- Check status of issued books, including due dates.

4. Book Return

- Return borrowed books through the system.
- View notifications for overdue books and applicable fines.

5. Transaction History

- Users can track their past book issues and returns.
- Provides transparency and easy record management.

3.3 SYSTEM FLOW OVERVIEW

Both modules work together to maintain seamless operations:

- The **Admin Module** manages the backend (books, users, transactions).
- The **User Module** provides a front-end interface for members to interact with the system.
- Data is stored in a centralized **MySQL database**, enabling efficient retrieval and updates.

CHAPTER 4

SAMPLE CODING

```
import java.util.ArrayList;
import java.util.Scanner;
// Book class
class Book {
    private int id;
    private String title;
    private String author;
    private boolean isIssued;
    public Book(int id, String title, String author) {
        this.id = id;
        this.title = title;
        this.author = author;
        this.isIssued = false;
    }
    public int getId() { return id; }
    public String getTitle() { return title; }
    public String getAuthor() { return author; }
    public boolean isIssued() { return isIssued; }
    public void issueBook() { isIssued = true; }
    public void returnBook() { isIssued = false; }
    @Override
    public String toString() {
        return id + " | " + title + " | " + author + " | " + (isIssued ? "Issued" :
"Available");
    }
}
// Library class
class Library {
```

```

private ArrayList<Book> books = new ArrayList<>();
public void addBook(Book book) {
    books.add(book);
    System.out.println("Book added successfully!");
}
public void showBooks() {
    if (books.isEmpty()) {
        System.out.println(" 📚 No books in the library yet.");
    } else {
        System.out.println("\n---- Library Books ----");
        for (Book b : books) {
            System.out.println(b);
        }
    }
}
public void issueBook(int id) {
    for (Book b : books) {
        if (b.getId() == id) {
            if (!b.isIssued()) {
                b.issueBook();
                System.out.println(" Book issued successfully!");
            } else {
                System.out.println("Book already issued!");
            }
            return;
        }
    }
    System.out.println(" Book not found!");
}
public void returnBook(int id) {
    for (Book b : books) {
        if (b.getId() == id) {

```

```

        if (b.isIssued()) {
            b.returnBook();
            System.out.println("Book returned successfully!");
        } else {
            System.out.println(" Book was not issued!");
        }
        return;
    }
}

System.out.println("Book not found!");
}

}

// Main class

public class LibraryManagementSystem {
    public static void main(String[] args) {
        Library library = new Library();
        Scanner sc = new Scanner(System.in);
        while (true) {
            System.out.println("\n===== Library Menu =====");
            System.out.println("1. Add Book");
            System.out.println("2. View Books");
            System.out.println("3. Issue Book");
            System.out.println("4. Return Book");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            int choice = sc.nextInt();
            sc.nextLine(); // consume newline
            switch (choice) {
                case 1:
                    System.out.print("Enter Book ID: ");
                    int id = sc.nextInt();
                    sc.nextLine();

```

```

        System.out.print("Enter Book Title: ");
        String title = sc.nextLine();
        System.out.print("Enter Author Name: ");
        String author = sc.nextLine();
        library.addBook(new Book(id, title, author));
        break;

    case 2:
        library.showBooks();
        break;

    case 3:
        System.out.print("Enter Book ID to issue: ");
        int issueId = sc.nextInt();
        library.issueBook(issueId);
        break;

    case 4:
        System.out.print("Enter Book ID to return: ");
        int returnId = sc.nextInt();
        library.returnBook(returnId);
        break;

    case 5:
        System.out.println("Exiting... Have a nice day!");
        sc.close();
        return;

    default:
        System.out.println(" Invalid choice! Try again.");
    }
}
}
}

```

This code establishes a connection between the Java application and the MySQL database. Ensures that the Java program can communicate with the backend database (librarydb), which stores all book and user details

```
import java.sql.*;
public class DBConnection {
    public static void main(String[] args) {
        try {
            // Load MySQL driver
            Class.forName("com.mysql.cj.jdbc.Driver");
            // Establish connection
            Connection con = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/librarydb", "root", "password");
            if (con != null) {
                System.out.println("Database Connected Successfully!");
            }
            con.close();
        } catch (Exception e) {
            System.out.println("Connection Failed: " + e);
        }
    }
}
```

This code snippet inserts a new book into the **books** table in the library database. Allows the administrator to add books to the library catalog, ensuring data is properly stored in the MySQL database

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.Scanner;
public class AddBookToLibrary {
    // Database connection details
    private static final String URL = "jdbc:mysql://localhost:3306/library"; // your DB
    name
    private static final String USER = "root"; // your MySQL username
    private static final String PASSWORD = "yourpassword"; // your MySQL
    password
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("==== Add New Book to Library ====");
        System.out.print("Enter Book ID: ");
        int id = sc.nextInt();
        sc.nextLine(); // consume newline
        System.out.print("Enter Book Title: ");
        String title = sc.nextLine();
        System.out.print("Enter Author Name: ");
        String author = sc.nextLine();
        System.out.print("Enter Quantity: ");
```

```

int quantity = sc.nextInt();
addBook(id, title, author, quantity);
sc.close();
// Method to insert book into database
public static void addBook(int id, String title, String author, int quantity) {
    Connection conn = null;
    PreparedStatement stmt = null;
    try {
        // Step 1: Load and register JDBC driver (optional for newer Java versions)
        Class.forName("com.mysql.cj.jdbc.Driver");
        // Step 2: Establish connection
        conn = DriverManager.getConnection(URL, USER, PASSWORD);
        // Step 3: Prepare SQL query
        String sql = "INSERT INTO books (book_id, title, author, quantity) VALUES
        (?, ?, ?, ?);";
        stmt = conn.prepareStatement(sql);
        // Step 4: Set parameters
        stmt.setInt(1, id);
        stmt.setString(2, title);
        stmt.setString(3, author);
        stmt.setInt(4, quantity);
        // Step 5: Execute update
        int rows = stmt.executeUpdate();
        if (rows > 0) {
            System.out.println("☑ Book added successfully to the library database!");
        } else {
            System.out.println(" Failed to add the book.");
        }
    } catch (SQLException e) {
        System.out.println(" Database Error: " + e.getMessage());
    } catch (ClassNotFoundException e) {
        System.out.println(" JDBC Driver not found!");
    } finally {
        try {
            if (stmt != null) stmt.close();
            if (conn != null) conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

ARCHITECTURE

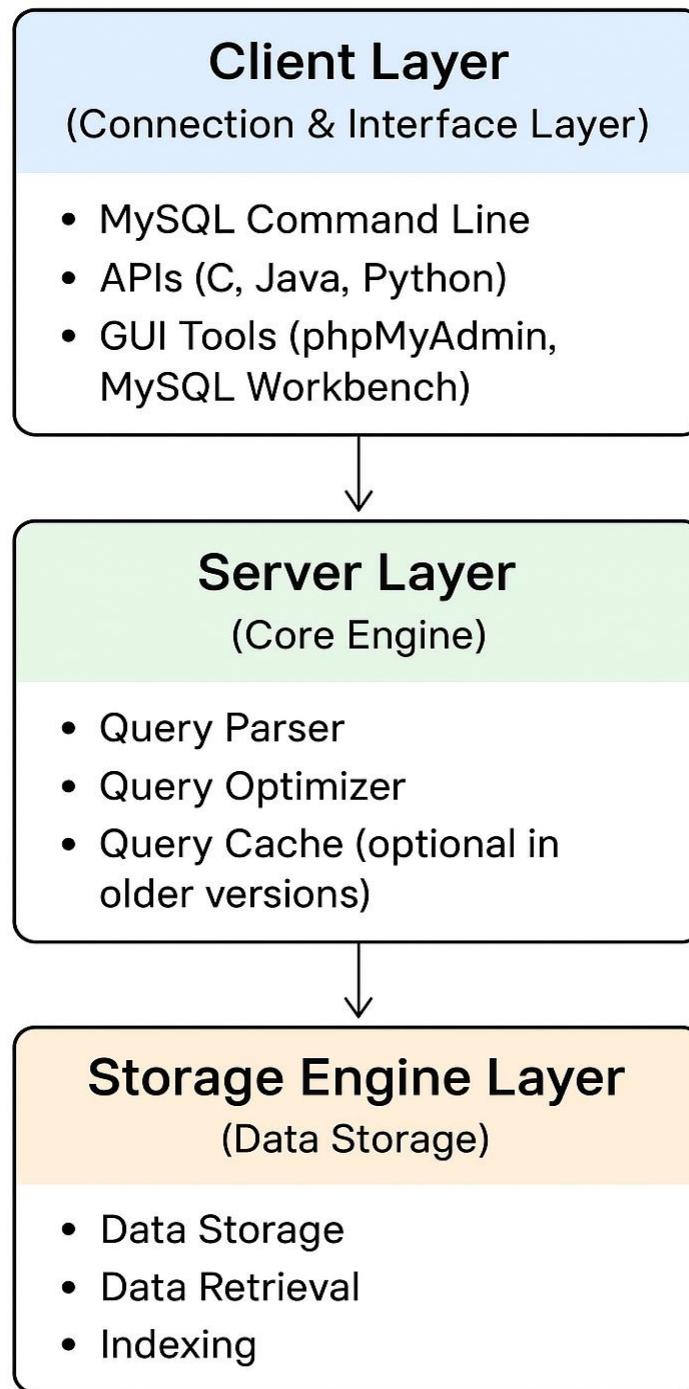


Fig 4.1 Architecture

CHAPTER 5

SCREEN SHOTS

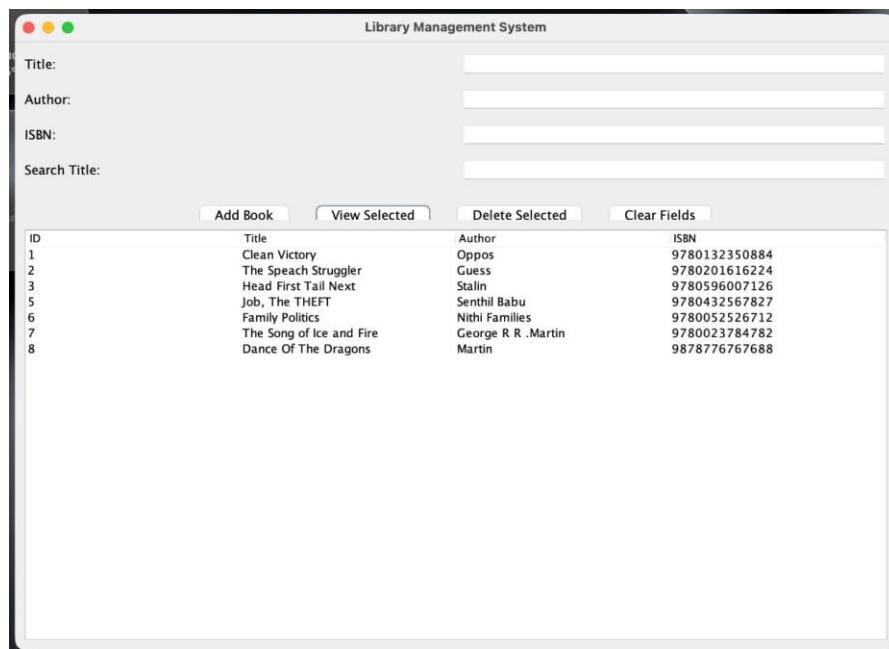


Fig 5.1 Introduction page

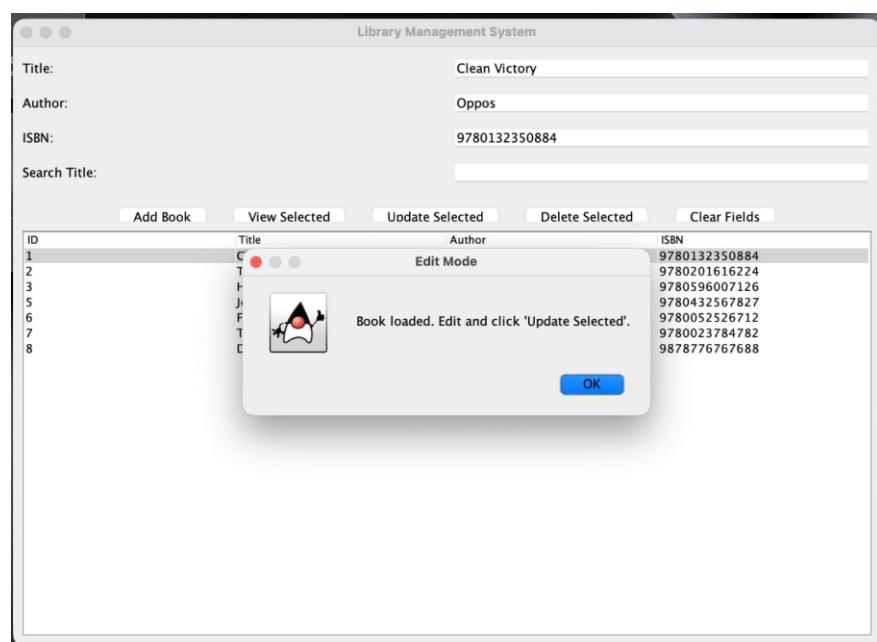


Fig 5.2 ADDING BOOK

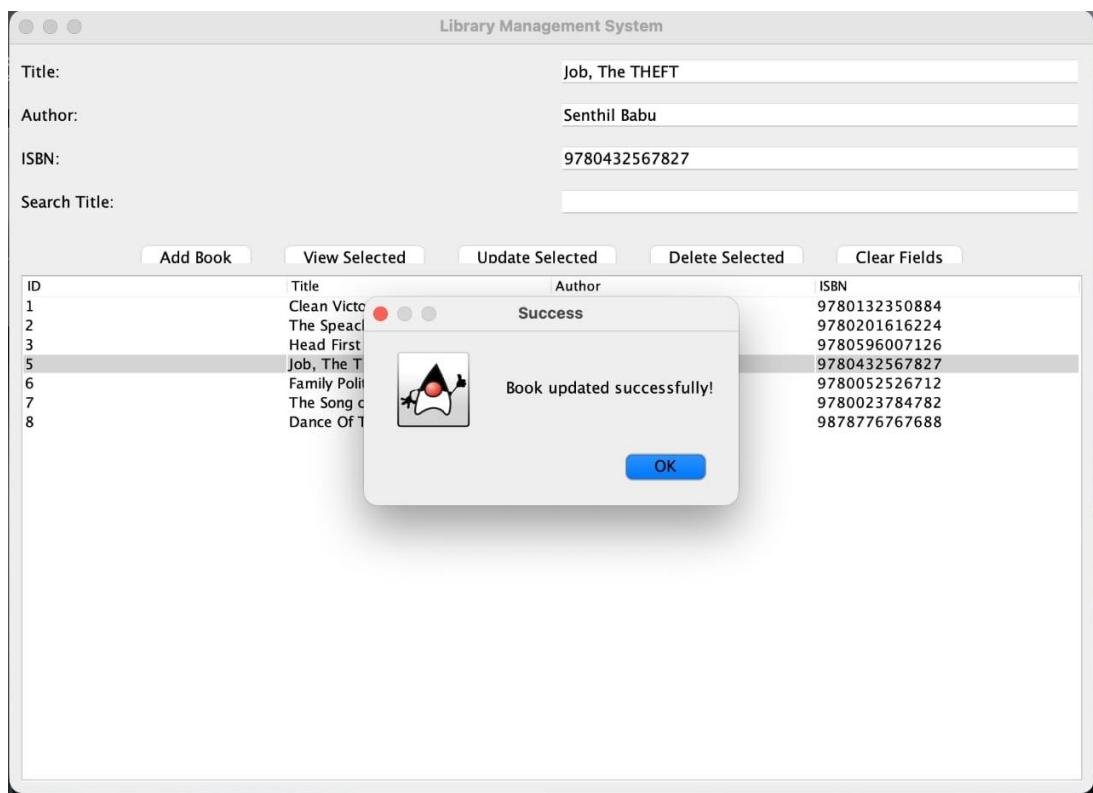


Fig 5.3 UPDATING BOOK

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

The Library Management System effectively digitalizes the traditional library setup, making it easier to manage books and users. It enhances efficiency by automating repetitive tasks such as catalog maintenance, issue/return operations, and fine calculation. The system ensures data consistency and improves user satisfaction through its interactive interface.

In the future, the system can be enhanced by integrating features like AI-based book recommendations, barcode/RFID scanning, and mobile accessibility, enabling a fully automated and modern library experience.

REFERENCES

1. <https://www.w3schools.com/java/>
2. <https://www.geeksforgeeks.org/javafx-tutorial/>
3. <https://www.tutorialspoint.com/mysql/>
4. <https://www.javatpoint.com/nodejs-tutorial>
5. <https://www.oracle.com/java/technologies/javase/>