Project Report

Library Management System using Core Java

Student Details

Field	Information
Name	Prateek Verma
Branch	Information Technology
Year	2nd Year
College	Babu Banarasi Das National Institute of Technology and Management (BBDNITM)
Project Title	Library Management System

1. Introduction

The Library Management System is a console-based Java application developed using core Java concepts. It aims to simplify and automate the management of books in a library, including tracking issued books, managing user records, and enabling easy book search and borrowing.

The system supports **two types of users**: **normal users** (students) and **administrators** (librarians). Each user type has access to specific features based on their roles.

2. Objectives

- To understand and implement Object-Oriented Programming (OOP) principles in a realworld application.
- To design a working prototype of a library management system using Core Java.
- To manage library operations like *viewing*, *borrowing*, *returning*, and *adding books* through a structured user interface.
- To provide basic authentication and role-based access (User/Admin).

3. Technologies Used

Technology	Purpose			
Java (JDK 17)	Main programming language			
OOP Concepts	Class, Object, Encapsulation, etc.			
Collections API	Storing and managing book/user data			
Console I/O	User input and display			
File I/O (optional)	Data persistence (if implemented)			

4. Functional Modules

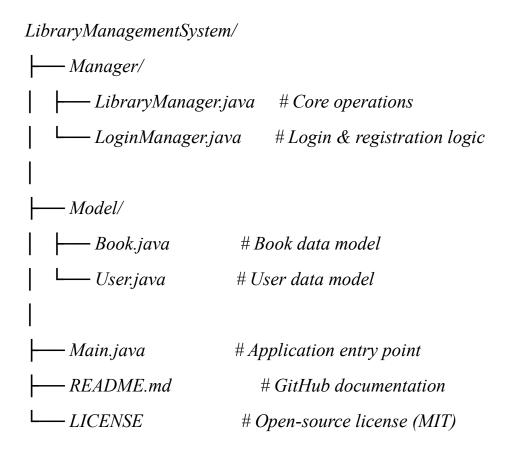
User Side:

- Register as a new user
- Login and logout
- View all books (with author, year, publisher, availability)
- Search for a book by name
- Borrow and return books using Book ID
- View list of borrowed books

Admin Side:

- Login as administrator
- Add new books to the system
- View all books
- View all registered users and their borrowed books
- Logout

5. Project Structure



6. Sample Book Display

<i>ID</i>	Book Name	Author	Publisher Ye	ar Copies/Total	
1001	Introduction to Java	John Smith	Sun Publisher	rs 2020 3/5	+
' '	Data Structures		•	,	+

7. How to Run the Project

Prerequisites:

- Java 17+
- Any Java IDE (IntelliJ, Eclipse) or terminal

Steps:

- 1. Clone or download the project
- 2. Open the project in IDE
- 3. Compile and run Main.java
- 4. Follow the console prompts to login/register and access features

8. Future Enhancements (as a 2nd-Year B. Tech Student)

- Password hashing for secure login
- Store data using file handling or a lightweight database (SQLite/MySQL)
- GUI with Swing or JavaFX for better interface
- Borrow period and overdue tracking
- Statistics (most borrowed books, active users)
- JUnit testing for key modules
- Auto-saving state on exit

9. Learning Outcomes

- Applied OOP concepts in a meaningful project
- Learned how to organize Java code into packages and modules
- Understood the basics of user authentication and role-based systems
- Improved skills in console application design and data handling
- Prepared groundwork for learning Java GUI, databases, and unit testing

10. Conclusion

The project helped in understanding how Java can be used to build structured, modular applications. It not only improves programming logic but also demonstrates how real-world systems are modeled using Object-Oriented principles. With further enhancements, this project can evolve into a complete library solution with a user-friendly interface and persistent storage.

Submitted by

Prateek Verma

B. Tech 2nd Year, Information Technology BBDNITM, Lucknow