

Project Report

Library Management System using Core Java

Student Details

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

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

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

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

LibraryManagementSystem/

```

├── Manager/
│   ├── LibraryManager.java    # Core operations
│   └── LoginManager.java     # Login & registration logic
│
├── Model/
│   ├── Book.java             # Book data model
│   └── User.java              # User data model
│
├── Main.java                  # Application entry point
├── README.md                  # GitHub documentation
└── LICENSE                    # Open-source license (MIT)

```

6. Sample Book Display

| ID | Book Name | Author | Publisher | Year | Copies/Total |
|------|----------------------|---------------|----------------|------|--------------|
| 1001 | Introduction to Java | John Smith | Sun Publishers | 2020 | 3 / 5 |
| 1002 | Data Structures | Alice Johnson | TechPress | 2018 | 1 / 2 |

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