INTERNSHIP PROJECT-1 DOCUMENT

ON “LIBRARY MANAGEMENT SYSTEM (JDBC & MYSQL)”

Submitted by:	Submitted To:-
Navpreet Singh (INTERN)	Kanduri Abhinay (FOUNDER)
	RITHIN VERMA (CTO)

INDEX

INTRODUCTION:-

The Library Management System is a console-based Java application that enables users to manage library records, including retrieving, issuing, updating, and deleting book information. By leveraging Java JDBC with a MySQL database, this system provides persistent storage, allowing efficient and scalable handling of library data. This project aims to provide a simple yet practical tool for library operations using SQL and Java, serving as a foundation for more advanced systems.

SOFTWARE REQUIREMENTS:-

Programming Language: Java (JDK 8 or higher)

IDE: VS Code, IntelliJ IDEA, or any Java-compatible editor

Libraries: Uses only Java Standard Library APIs (I/O, Serialization)

Operating System: Platform-independent (Windows, Linux, Mac)

Database: MySQL (tested with port 3308)

Other Requirements: MySQL server running with user access and the specified password and database.

DESIGN:-

The application is structured as follows:

Database Table: Stores book details, borrower information, issue dates, and book type.

Books Main Class: Handles all user interaction, menu, and database CRUD operations.

Persistence: Data is stored and managed in a MySQL table using JDBC for secure and reliable access and transactions.

User Interaction: A console menu system for retrieving, issuing, updating, or deleting records.

INPUT:-

```
+----------------------+

|  Library Managment  |
+-----------------------+
| 1. To data retrieval  |
| 2. To issue book      |
| 3. Book returned      |
|    (delete data)      |
| 4. Update Values      |
+-----------------------+
```

Users select actions via terminal by entering command numbers..

IMPLEMENTATION:-

The main Books class communicates with MySQL using JDBC, performing all CRUD operations through SQL queries and prepared statements.

User input is collected via Scanner and mapped to the corresponding database fields.

Each command (retrieve, issue, delete, update) is handled by a dedicated case in a continuous loop, with input validation and messaging.

All changes are immediately reflected in the database for data durability.

TESTING:-

The application was tested as follows:

Manual Input Testing: Entered multiple book records (with a variety of details, dates, and types) and verified retrieval, editing, and deletion.

Edge Cases: Tested with non-existent IDs; attempted to add, update, or delete using invalid data.

Database Validation: Confirmed by checking MySQL directly after each operation.

Menu Flow: Ensured appropriate handling for invalid menu selections or incorrect input formats (e.g., wrong date).

Exception Handling: Verified that loss of database connection and SQL errors are caught and do not crash the application.

ADVANTAGES:-

Simplicity: Straightforward interface and code structure, ideal for learning purposes.

Real Persistence: Uses a real relational database for permanent, scalable storage.

Standard Tools: Requires only JDBC and MySQL—no external libraries needed.

Extensibility: The structure allows adding new features (e.g., user membership, fine calculation).

Cross-Platform: Runs on any OS with Java and MySQL installed.

CONCLUSION:-

This Library Management System demonstrates essential enterprise Java programming concepts—JDBC, SQL operations, exception handling, and menu-driven console applications. The project reinforced knowledge of backend programming, SQL, and practical software design. It serves as a foundation for future development of full-featured library systems using modern frameworks.

REFERENCES:-

GeeksForGeeks: Java & JDBC Tutorials

JavaTPoint: Java Database Connectivity

Oracle Documentation: Official Java/JDBC Reference

MySQL Documentation