

# 1. Overview

This is a console-based Library Management System written in Java. It allows users to add books to a MySQL database and view the list of books. It's a minimal version, suitable for GitHub uploads, learning JDBC, or using in academic projects.

## 2. Database Setup (MySQL)

Create a file named **library.sql** with the following content:

```
CREATE DATABASE library_db;
USE library_db;

CREATE TABLE books (
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(255),
    author VARCHAR(255),
    isbn VARCHAR(20)
);
```

## 3. Java Files

### DBConnection.java

```
import java.sql.*;

public class DBConnection {
    public static Connection getConnection() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            return DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/library_db", "root",
                "password"
            );
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

### Book.java

```
public class Book {
```

```

private String title;
private String author;
private String isbn;

public Book(String title, String author, String isbn) {
    this.title = title;
    this.author = author;
    this.isbn = isbn;
}

public String getTitle() { return title; }
public String getAuthor() { return author; }
public String getIsbn() { return isbn; }
}

```

## BookDAO.java

```

import java.sql.*;

public class BookDAO {
    public static void addBook(Book book) {
        try (Connection con = DBConnection.getConnection()) {
            String query = "INSERT INTO books (title, author, isbn)
VALUES (?, ?, ?)";
            PreparedStatement ps = con.prepareStatement(query);
            ps.setString(1, book.getTitle());
            ps.setString(2, book.getAuthor());
            ps.setString(3, book.getIsbn());
            ps.executeUpdate();
            System.out.println("Book added successfully.");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void listBooks() {
        try (Connection con = DBConnection.getConnection()) {
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM books");
            while (rs.next()) {
                System.out.println("ID: " + rs.getInt("id") +
+
+
+
                ", Title: " + rs.getString("title")
+
+
                ", Author: " +
rs.getString("author") +
                ", ISBN: " + rs.getString("isbn"));
            }
        }
    }
}

```

```

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

## Main.java

```

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        while (true) {
            System.out.println("\nLibrary Management System");
            System.out.println("1. Add Book");
            System.out.println("2. View Books");
            System.out.println("3. Exit");
            System.out.print("Enter choice: ");
            int choice = sc.nextInt();
            sc.nextLine(); // consume newline

            switch (choice) {
                case 1:
                    System.out.print("Enter Title: ");
                    String title = sc.nextLine();
                    System.out.print("Enter Author: ");
                    String author = sc.nextLine();
                    System.out.print("Enter ISBN: ");
                    String isbn = sc.nextLine();
                    Book book = new Book(title, author, isbn);
                    BookDAO.addBook(book);
                    break;
                case 2:
                    BookDAO.listBooks();
                    break;
                case 3:
                    System.out.println("Exiting...");
                    System.exit(0);
            }
        }
    }
}

```

#### 4. Technologies Used

Java (JDK 8 or later)

MySQL Database

JDBC (Java Database Connectivity)

Console-based interface

---

#### 5. Benefits of This Project

Understands Java-MySQL interaction using JDBC.

Demonstrates CRUD logic using basic SQL.

Lightweight and easy to expand to full applications.

Excellent for learning Java-based backend systems.