PROJECT REPORT

LIBRARY BOOK MANAGEMENT

ABSTRACT:

The Library Book Management System is an efficient and user-friendly application designed to streamline the process of managing library resources. By leveraging Java, JDBC, and MySQL for back-end operations and HTML/CSS for the user interface, this system provides a robust solution for librarians to add, view, update, and delete book records with ease. The program's database integration ensures data consistency and reliability, while the web-based interface enhances accessibility and usability. This paper discusses the development process, underlying technologies, and their integration, offering a practical guide for implementing a real-world book management application.

INTRODUCTION:

Libraries play a vital role in fostering education and literacy by providing access to a diverse collection of books and resources. Efficient management of these resources is essential to ensure smooth operations and better user experiences. Traditional library management methods, often reliant on manual record-keeping, are prone to errors and inefficiencies.

The Library Book Management System is designed to address these challenges by offering a digital solution for book management. Developed using Java for back-end operations and a combination of HTML and CSS for the front-end, this system provides functionalities to add, view, update, and delete book records stored in a MySQL database. The integration of Java Database Connectivity (JDBC) allows seamless communication between the application and the database, ensuring a robust and scalable solution.

This project aims to demonstrate the potential of combining programming, database management, and web technologies to create an intuitive and reliable library management system.

LITERATURE REVIEW:

Libraries play a vital role in fostering education and literacy by providing access to a diverse collection of books and resources. Efficient management of these resources is essential to ensure smooth operations and better user experiences. Traditional

library management methods, often reliant on manual record-keeping, are prone to errors and inefficiencies.

The Library Book Management System is designed to address these challenges by offering a digital solution for book management. Developed using Java for back-end operations and a combination of HTML and CSS for the front-end, this system provides functionalities to add, view, update, and delete book records stored in a MySQL database. The integration of Java Database Connectivity (JDBC) allows seamless communication between the application and the database, ensuring a robust and scalable solution.

This project aims to demonstrate the potential of combining programming, database management, and web technologies to create an intuitive and reliable library management system.

PROGRAM:

1.DATABASE SETUP

```
CREATE DATABASE LibraryDB;
USE LibraryDB;
CREATE TABLE Books (
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    author VARCHAR(255) NOT NULL,
    genre VARCHAR(100),
    published_year INT,
    available BOOLEAN DEFAULT TRUE
);
```

2.JAVA WITH JDBC

```
import java.sql.*;
import java.util.Scanner;
public class LibraryManagement {
    private static final String DB_URL = "jdbc:mysql://localhost:3306/LibraryDB";
    private static final String USER = "root"; // Replace with your MySQL username
    private static final String PASS = "password"; // Replace with your MySQL password
    public static void main(String[] args) {
```

```
        try (Connection connection = DriverManager.getConnection(DB_URL, USER,
PASS)) {
            System.out.println("Connected to the database!");
            Scanner scanner = new Scanner(System.in);
            while (true) {
                System.out.println("\nLibrary Book Management System");
                System.out.println("1. Add Book");
                System.out.println("2. View All Books");
                System.out.println("3. Update Book");
                System.out.println("4. Delete Book");
                System.out.println("5. Exit");
                System.out.print("Choose an option: ");
                int choice = scanner.nextInt();
                scanner.nextLine(); // Consume newline
                switch (choice) {
                    case 1:
                        addBook(connection, scanner);
                        break;
                    case 2:
                        viewBooks(connection);
                        break;
                    case 3:
                        updateBook(connection, scanner);
                        break;
                    case 4:
                        deleteBook(connection, scanner);
                        break;
                    case 5:
                        System.out.println("Exiting...");
```

```
                    return;
                default:
                    System.out.println("Invalid choice!");
            }
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
private static void addBook(Connection connection, Scanner scanner) throws SQLException {
    System.out.print("Enter book title: ");
    String title = scanner.nextLine();
    System.out.print("Enter book author: ");
    String author = scanner.nextLine();
    System.out.print("Enter book genre: ");
    String genre = scanner.nextLine();
    System.out.print("Enter published year: ");
    int year = scanner.nextInt();
    String sql = "INSERT INTO Books (title, author, genre, published_year) VALUES (?, ?, ?, ?)";
    try (PreparedStatement pstmt = connection.prepareStatement(sql)) {
        pstmt.setString(1, title);
        pstmt.setString(2, author);
        pstmt.setString(3, genre);
        pstmt.setInt(4, year);
        pstmt.executeUpdate();
        System.out.println("Book added successfully!");
    }
}
```

```
    private static void viewBooks(Connection connection) throws SQLException {
        String sql = "SELECT * FROM Books";
        try (Statement stmt = connection.createStatement(); ResultSet rs = stmt.executeQuery(sql)) {
            System.out.printf("%-5s %-30s %-20s %-20s %-5s %-10s%n", "ID", "Title", "Author", "Genre", "Year", "Available");
            while (rs.next()) {
                System.out.printf("%-5d %-30s %-20s %-20s %-5d %-10s%n",
                        rs.getInt("id"),
                        rs.getString("title"),
                        rs.getString("author"),
                        rs.getString("genre"),
                        rs.getInt("published_year"),
                        rs.getBoolean("available") ? "Yes" : "No");
            }
        }
    }
    private static void updateBook(Connection connection, Scanner scanner) throws SQLException {
        System.out.print("Enter book ID to update: ");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consume newline
        System.out.print("Enter new title: ");
        String title = scanner.nextLine();
        System.out.print("Enter new author: ");
        String author = scanner.nextLine();
        System.out.print("Enter new genre: ");
        String genre = scanner.nextLine();
        System.out.print("Enter new published year: ");
        int year = scanner.nextInt();
```

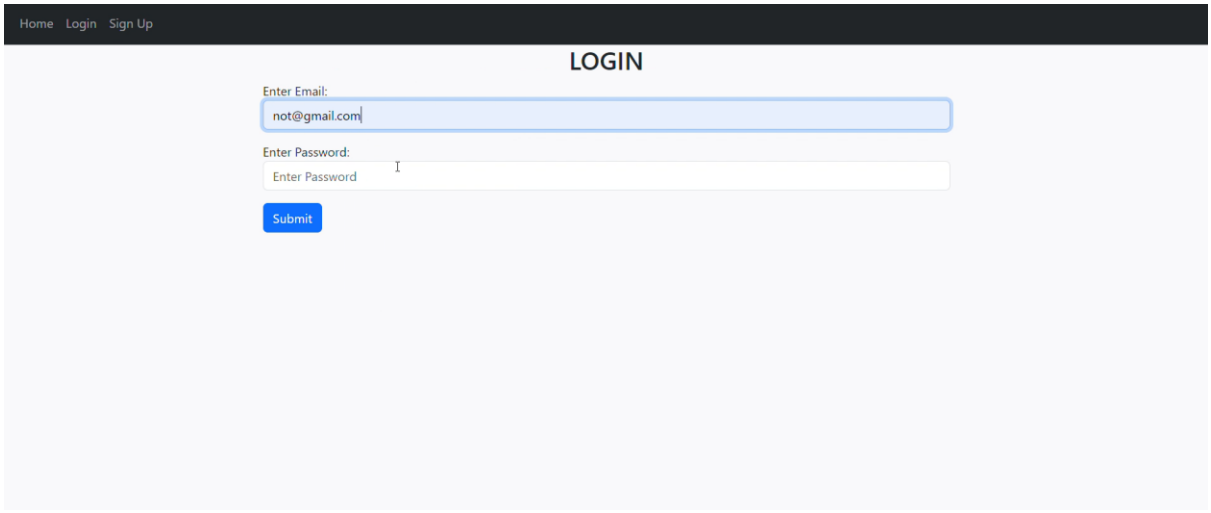
```
        String sql = "UPDATE Books SET title = ?, author = ?, genre = ?, published_year
= ? WHERE id = ?";
        try (PreparedStatement pstmt = connection.prepareStatement(sql)) {
            pstmt.setString(1, title);
            pstmt.setString(2, author);
            pstmt.setString(3, genre);
            pstmt.setInt(4, year);
            pstmt.setInt(5, id);
            pstmt.executeUpdate();
            System.out.println("Book updated successfully!");
        }
    }
    private static void deleteBook(Connection connection, Scanner scanner) throws
SQLException {
        System.out.print("Enter book ID to delete: ");
        int id = scanner.nextInt();
        String sql = "DELETE FROM Books WHERE id = ?";
        try (PreparedStatement pstmt = connection.prepareStatement(sql)) {
            pstmt.setInt(1, id);
            pstmt.executeUpdate();
            System.out.println("Book deleted successfully!");
        }
    }
}
```

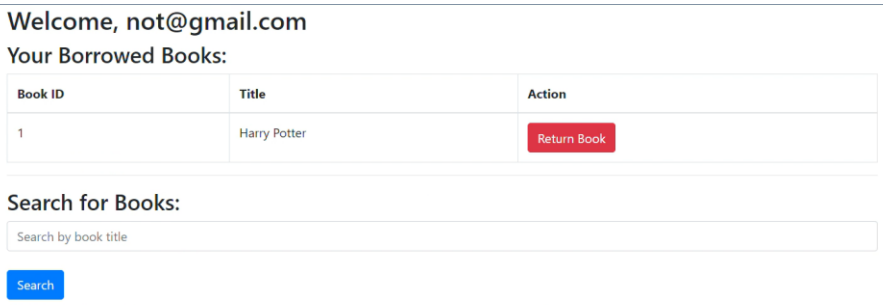
3.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>Library Management</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <h1>Library Book Management</h1>
        <form action="LibraryServlet" method="post">
            <label for="action">Choose Action:</label>
            <select name="action" id="action">
                <option value="view">View Books</option>
                <option value="add">Add Book</option>
                <option value="update">Update Book</option>
                <option value="delete">Delete Book</option>
            </select>
            <button type="submit">Submit</button>
        </form>
    </div>
</body>
</html>
```

OUTPUT:





CONCLUSION:

The Library Book Management System successfully addresses the challenges of traditional library resource management by offering a digital, user-friendly, and reliable solution. The use of Java and MySQL ensures robust back-end operations, while the integration of HTML and CSS provides an intuitive front-end experience. By

automating key processes such as adding, updating, viewing, and deleting book records, the system reduces human error, improves efficiency, and enhances the overall user experience.

Future enhancements could include additional features such as user authentication, book borrowing and return tracking, and advanced search capabilities. This project demonstrates the effectiveness of integrating programming, database management, and web technologies to create scalable and impactful applications, making it a valuable resource for libraries aiming to modernize their operations.