

# WEB APPLICATION DEVELOPMENT

## ASSIGNMENT – I

### QUESTION:02

#### Library Management System – Features Explanation

This project is a web-based Library Management System using Java (Servlets & JSP), MySQL, and JDBC. It allows the management of books in a library by performing basic CRUD operations (Create, Read, Update, Delete) along with a search functionality.

##### Features to Implement

1. Add Student – Allows users to add a new student (ID, Name, Email, Course, GPA).
2. View Students – Displays a list of all students.
3. Update Student – Provides an interface to modify student details.
4. Delete Student – Allows users to remove a student record.
5. Search Student (Optional) – Enables searching by name or course.

##### 2. Project Folder Structure

Folder	Contents
src/com.school.dao/	StudentDAO.java, StudentDAOImpl.java
src/com.school.model/	Student.java
src/com.school.servlet/	AddStudentServlet.java, ViewStudentServlet.java, UpdateStudentServlet.java, DeleteStudentServlet.java
src/com.school.util/	DBConnection.java
WebContent/view/	add_student.jsp, view_students.jsp, update_student.jsp, index.jsp
WebContent/css/	styles.css

##### 3. Database Schema

Column Name	Data Type	Constraints
Id	INT	PRIMARY KEY, AUTO_INCREMENT
Name	VARCHAR(100)	NOT NULL
Email	VARCHAR(100)	UNIQUE, NOT NULL
Course	VARCHAR(50)	NOT NULL
Gpa	FLOAT	NOT NULL

## **CODE:**

### **Database in MySQL**

```
CREATE DATABASE library_db; USE
library_db;

CREATE TABLE books (
    id INT PRIMARY KEY AUTO_INCREMENT,
    title VARCHAR(150) NOT NULL,
    author VARCHAR(100) NOT NULL,
    genre VARCHAR(50) NOT NULL,
    status ENUM('Available','Borrowed') NOT NULL
);
```

### **DBConnection.java**

```
package com.library.util;

import java.sql.Connection;
import java.sql.DriverManager;

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**

```
package com.library.model;

public class Book {

    private int id;

    private String title;

    private String author;

    private String genre;

    private String status;

    // Getters and Setters

}
```

### **BookDAO.java**

```
package com.library.dao;

import java.util.List;

import com.library.model.Book;

public interface BookDAO {

    void addBook(Book book);

    List<Book> getAllBooks();

    void updateBook(Book book);

    void deleteBook(int id);

}
```

### **BookDAOImpl.java**

```
public void addBook(Book book) {

    try (Connection con = DBConnection.getConnection()) {

        String sql = "INSERT INTO books(title,author,genre,status) VALUES(?,?,?,?)";

        PreparedStatement ps = con.prepareStatement(sql);

        ps.setString(1, book.getTitle());

        ps.setString(2, book.getAuthor());

        ps.setString(3, book.getGenre());

        ps.setString(4, book.getStatus());

        ps.executeUpdate();

    }

}
```

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

### **AddBookServlet.java**

```
@WebServlet("/addBook")  
  
public class AddBookServlet extends HttpServlet {  
    protected void doPost(HttpServletRequest request, HttpServletResponse response) {  
        Book book = new Book();  
        book.setTitle(request.getParameter("title"));  
        book.setAuthor(request.getParameter("author"));  
        book.setGenre(request.getParameter("genre"));  
        book.setStatus(request.getParameter("status"));  
        BookDAO dao = new BookDAOImpl();  
        dao.addBook(book);  
        response.sendRedirect("viewBooks");  
    }  
}
```

### **ViewBookServlet.java**

```
package com.library.servlet;  
  
import java.io.IOException;  
  
import java.util.List;  
  
import jakarta.servlet.RequestDispatcher;  
import jakarta.servlet.ServletException;  
import jakarta.servlet.annotation.WebServlet;  
import jakarta.servlet.http.HttpServlet;  
import jakarta.servlet.http.HttpServletRequest;  
import jakarta.servlet.http.HttpServletResponse;  
  
import com.library.dao.BookDAO;  
import com.library.dao.BookDAOImpl;
```

```

import com.library.model.Book;
@WebServlet("/viewBooks")
public class ViewBookServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        BookDAO dao = new BookDAOImpl();
        List<Book> bookList = dao.getAllBooks();

        request.setAttribute("bookList", bookList);
        RequestDispatcher rd = request.getRequestDispatcher("view/view_books.jsp");
        rd.forward(request, response);
    }
}

```

### **UpdateBookServlet.java**

```

package com.library.servlet;
import java.io.IOException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import com.library.dao.BookDAO;
import com.library.dao.BookDAOImpl;
import com.library.model.Book;
@WebServlet("/updateBook")
public class UpdateBookServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        int id = Integer.parseInt(request.getParameter("id"));
        String title = request.getParameter("title");

```

```

        String author = request.getParameter("author");
        String genre = request.getParameter("genre");
        String status = request.getParameter("status");
        Book book = new Book();
        book.setId(id);
        book.setTitle(title);
        book.setAuthor(author);
        book.setGenre(genre);
        book.setStatus(status);
        BookDAO dao = new BookDAOImpl();
        dao.updateBook(book);
        response.sendRedirect("viewBooks");
    }
}

```

### **DeleteBookServlet.java**

```

package com.library.servlet;

import java.io.IOException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import com.library.dao.BookDAO;
import com.library.dao.BookDAOImpl;
@WebServlet("/deleteBook")
public class DeleteBookServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        int id = Integer.parseInt(request.getParameter("id"));
        BookDAO dao = new BookDAOImpl();
    }
}

```

```
        dao.deleteBook(id);

        response.sendRedirect("viewBooks");
    }
}
```

### **index.jsp**

```
<h2>Library Management System</h2>

<a href="view/add_book.jsp">Add Book</a><br>

<a href="viewBooks">View Books</a>
```

### **add\_book.jsp**

```
<form action="../addBook" method="post">

Title: <input type="text" name="title"><br>

Author: <input type="text" name="author"><br>

Genre: <input type="text" name="genre"><br>

Status:

<select name="status">

<option>Available</option>

<option>Borrowed</option>

</select><br>

<input type="submit" value="Add Book">

</form>
```

### **view\_books.jsp**

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

<% @ page import="java.util.List" %>

<% @ page import="com.library.model.Book" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">
```

```

<title>View Books</title>

</head>

<body>

<h2>Library Book List</h2>

<a href="add_book.jsp">Add New Book</a>

<br><br>

<table border="1" cellpadding="10">

  <tr>

    <th>ID</th>

    <th>Title</th>

    <th>Author</th>

    <th>Genre</th>

    <th>Status</th>

    <th>Actions</th>

  </tr>

  <%

    List<Book> bookList = (List<Book>) request.getAttribute("bookList");

    if (bookList != null) {
      for (Book book : bookList) {
%>

    <tr>

      <td><%= book.getId() %></td>

      <td><%= book.getTitle() %></td>

      <td><%= book.getAuthor() %></td>

      <td><%= book.getGenre() %></td>

      <td><%= book.getStatus() %></td>

      <td>

        <a href="update_book.jsp?id=<%= book.getId() %>

```

```

        &title=<%= book.getTitle() %>
        &author=<%= book.getAuthor() %>
        &genre=<%= book.getGenre() %>
        &status=<%= book.getStatus() %>">
        Edit</a>
        |
        <a href=" ../deleteBook?id=<%= book.getId() %>">
        Delete</a>
    </td>
</tr>
</tr>

<%
    }
}
%>
</table>
<br><a href=" ../index.jsp">Back to Home</a>
</body>
</html>
update_book.jsp
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Update Book</title>
</head>
<body>
<h2>Update Book</h2>

```

```

<%
    String id = request.getParameter("id");
    String title = request.getParameter("title");
    String author = request.getParameter("author");
    String genre = request.getParameter("genre");
    String status = request.getParameter("status");
%>
<form action="../updateBook" method="post">
    <input type="hidden" name="id" value="<%= id %>">
    Title:
    <input type="text" name="title" value="<%= title %>"><br><br>
    Author:
    <input type="text" name="author" value="<%= author %>"><br><br>
    Genre:
    <input type="text" name="genre" value="<%= genre %>"><br><br>
    Status:
    <select name="status">
        <option value="Available" <%= "Available".equals(status) ? "selected" : "" %>>
            Available
        </option>
        <option value="Borrowed" <%= "Borrowed".equals(status) ? "selected" : "" %>>
            Borrowed
        </option>
    </select>
    <br><br>
    <input type="submit" value="Update Book">
</form>
<br>
<a href="../viewBooks">Back to Book List</a>
</body>
</html>

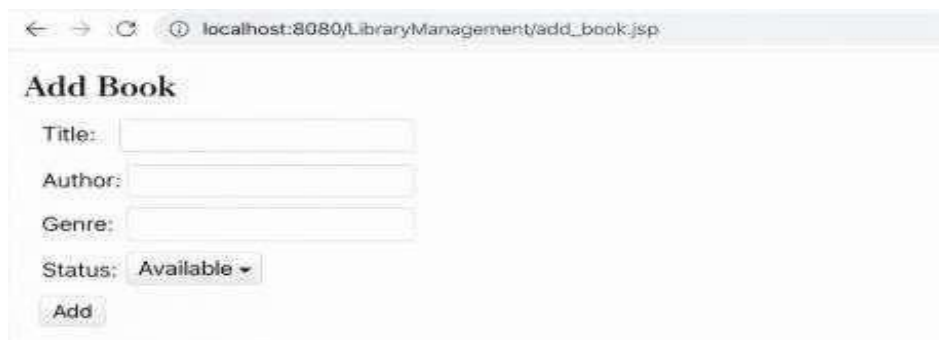
```

## OUTPUT:

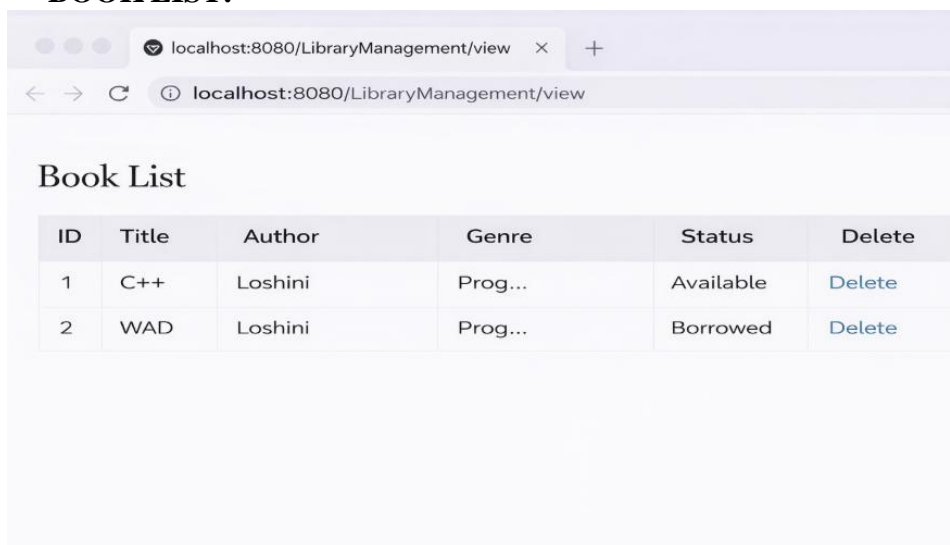
### WELCOME PAGE:



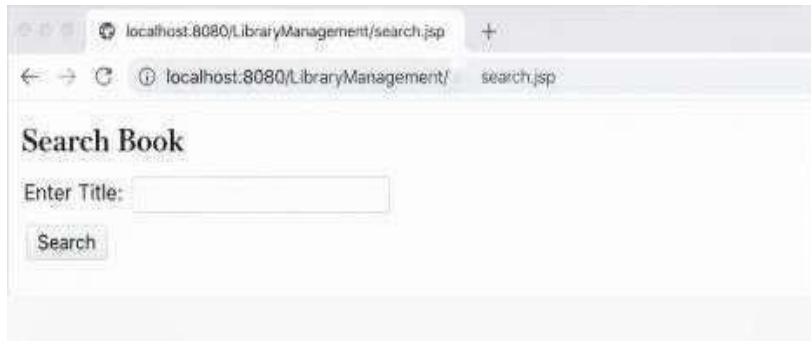
### ADD BOOK:



### BOOK LIST:



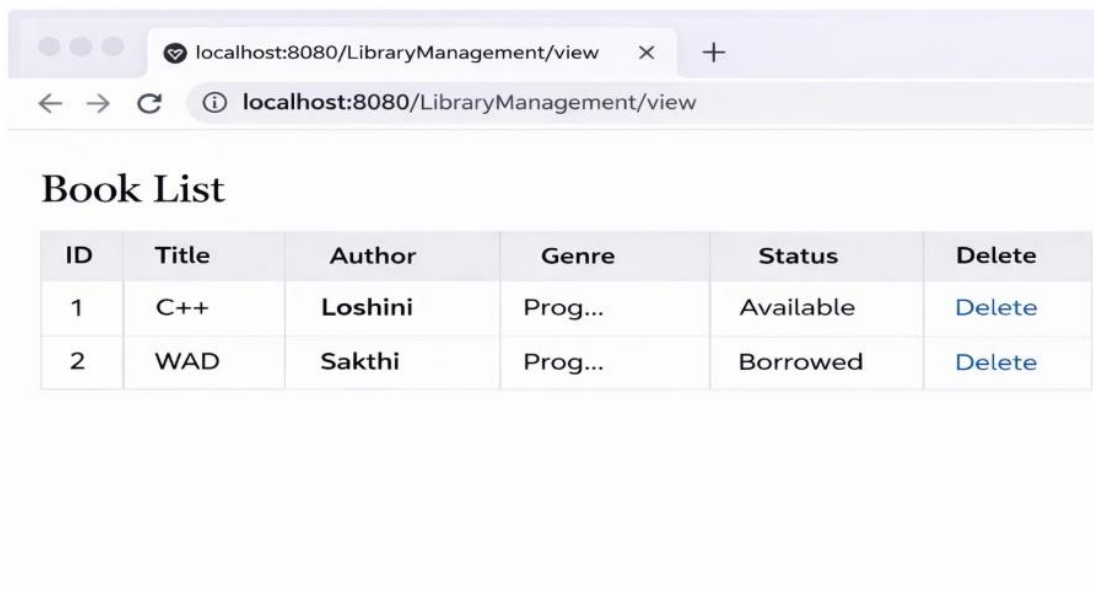
## SEARCH BOOK:



Search Book

Enter Title:

## BOOK LIST VIEW:



ID	Title	Author	Genre	Status	Delete
1	C++	Loshini	Prog...	Available	<a href="#">Delete</a>
2	WAD	Sakthi	Prog...	Borrowed	<a href="#">Delete</a>