

LAB 7: JSP Parameter Handling & Request Forwarding

Lab Activities

You are required to extend the Music Library application by implementing parameter binding with input parameters and request forwarding between JSP pages.

1. Implement Music Search with Input Parameters

Task 1: Create a new JSP file name `search_music.jsp` to allow users to search for a song by title or artist.

- Add a form with input fields for title and artist in the search music.jsp
- The form should be submitted to result_music.jsp.

Search Music Library

Title:

Artist:

2. Process parameters in result_music.jsp

Task 1: Create a new JSP file name `result_music.jsp` to display the search result.

- Add the following code to display search results and forward to `list_music.jsp`, if no results are found.

```
<h2>Search Results</h2>
<%
    String title = request.getParameter("title");
    String artist = request.getParameter("artist");

    // Simulated search logic for demonstration purposes
    if ((title.equals("Cardigan") && artist.equals("Taylor Swift")) ||
        (title.isEmpty() && artist.equals("Taylor Swift"))) {
%>
        <p>Found: Cardigan by Taylor Swift (Indie Folk, 2020)</p>
<%
    } else {
        request.getRequestDispatcher("list_music.jsp").forward(request, response);
    }
%>
```

- Modify `list_music.jsp` to handle forwarded requests and display a message when redirected from the search page.

```
<%
    if (request.getAttribute("forwarded") != null) {
        out.println("<p>No music found for the given search.</p>");
    }
%>
```

Task 2: Implement Error handling for missing parameters.

- a. Modify `result_music.jsp` to handle missing input parameters by redirecting back to `search_music.jsp` with an error message.

```
if (title == null || title.isEmpty()) {
    response.sendRedirect("search_music.jsp?error=Title is required");
    return;
}
```

- b. Update `search_music.jsp` to display error messages using query parameters.

```
<%
    String error = request.getParameter("error");
    if (error != null) {
        out.println("<p style='color:red;'>" + error + "</p>");
    }
%>
```

- c. Deploy and Run the web application.

Task 3: Modify code in `result_music.jsp` to use JSTL

- a. Add the JSTL library by including the following tag at the top of the page.

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

- b. Replace the following code to use JSTL to display search results and forward to `list_music.jsp`, if no results are found.

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<h2>Search Results</h2>
<c:set var="title" value="${param.title}" />
<c:set var="artist" value="${param.artist}" />

<c:choose>
    <c:when test="${title eq 'Cardigan' && artist eq 'Taylor Swift'}">
        <p>Found: Cardigan by Taylor Swift (Indie Folk, 2020)</p>
    </c:when>
    <c:otherwise>
        <c:redirect url="list_music.jsp?forwarded=true" />
    </c:otherwise>
</c:choose>
```

- c. Modify `list_music.jsp` to handle forwarded requests and display a message when redirected from the search page using JSTL

```
<h1>Music Library</h1>
<c:if test="${param.forwarded eq 'true'}">
    <p>No music found for the given search.</p>
</c:if>
```

3. Implement a Logout.jsp

Task 1: Create a new JSP page named logout.jsp

- a. Add the following code to invalidate the session and redirect the user to the home page.

```
<%  
    session.invalidate();  
    response.sendRedirect("index.html");  
%>
```

- b. Add a logout link to the list_music.jsp

```
<a href="logout.jsp">Logout</a>
```

Task 2: Deploy and Test the Web Application

In this task, you build and deploy the MusicLibrary web application.

Complete the following steps:

1. Build the MusicLibrary web application. Correct any errors you encounter.
2. Deploy the MusicLibrary web application.
3. Refresh the MusicLibrary web application home page.
4. Test the Search Music page and Logout function.

Postlab Exercise

You need to extend the search function with search filtering and extract search results from database in the Music Library application. Perform the following tasks:

1. Add search filters in `search_music.jsp` that include:
 - a. Genre: a dropdown with genres.
 - b. Year range: two input fields for start year and end year.
2. Modify the `result_music.jsp` to handle the filter search results based on the selected genre and year range. Use conditional logic to filter records.
3. Modify the code in the `result_music.jsp` to display the search results from the database.