



UNIVERSITI MALAYSIA TERENGGANU

CSM3023 WEB BASED APPLICATION DEVELOPMENT (K1)

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS

LAB 6 – JSP : SAVING AND RETRIEVING FROM DATABASE

SEMESTER 4 2023/2024

Prepared for:

DR. MOHAMAD NOR HASSAN

Prepared by:

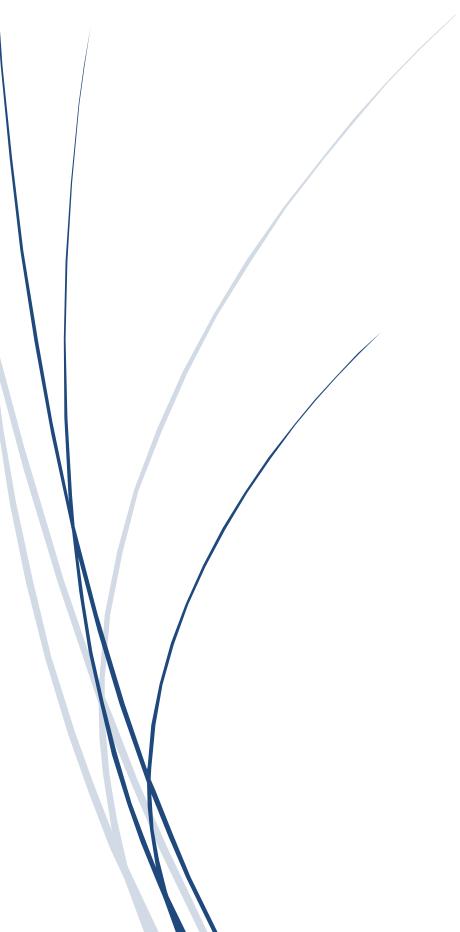
NUR ARINA BINTI ABDUL MALEK (S65361)



Week 6

JSP: Saving and Retrieving Data from Database

Web Programming 2



Lecturers

PUSAT PENGAJIAN INFORMATIK DAN MATEMATIK
GUNAAN (PPIMG), UNIVERSITI MALAYSIA TERENGGANU
(UMT)

Revision History

Revision Date	Previous Revision Date	Summary of Changes	Changes Marked
		First Issue	Mohamad Nor Hassan
		Second Issue	Dr Rabiei Mamat Dr Faizah Aplop Dr Fouad Ts Dr Rosmayati Mohemad Fakhrul Adli Mohd Zaki
13/03/2019	21/02/2019	Addition of Revision History, Table of Contents, Formatting Cover Page	Fakhrul Adli Mohd Zaki

Table of Contents

Task 1: Using JSP Page to Access a Simple MySQL Database	5
Task 2: Create Records via JSP Page	13
Task 3: Create Records Constrained by Regular Expression In JSP	20
Task 4: Perform Retrieving Records Via JSP Page.....	29
Task 5: Create A Record Using JSP Model 1	33

Arahan:

Manual makmal ini adalah untuk kegunaan pelajar-pelajar Pusat Pengajian Informatik dan Matematik Gunaan (PPIMG), Universiti Malaysia Terengganu (UMT) sahaja. Tidak dibenarkan mencetak dan mengedar manual ini tanpa kebenaran rasmi daripada penulis.

Sila ikuti langkah demi langkah sebagaimana yang dinyatakan di dalam manual. Tandakan (/) setiap langkah yang telah selesai dibuat dan tulis kesimpulan bagi setiap aktiviti yang telah selesai dijalankan.

Instruction:

This laboratory manual is for use by the students of the School of Informatics and Applied Mathematics (PPIMG), Universiti Malaysia Terengganu (UMT) only. It is not permissible to print and distribute this manual without the official authorisation of the author.

Please follow step by step as described in the manual. Tick (/) each step completed and write the conclusions for each completed activity.

Task 1: Using JSP Page to Access a Simple MySQL Database

Objective: Write a JSP that can insert data to MYSQL database as “Welcome to access MySQL database with JSP.!” and also display steps of how to connect with MYSQL database.

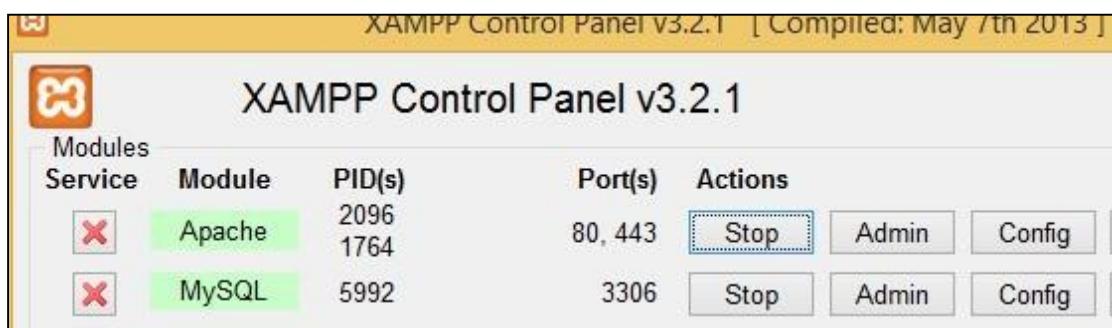
Problem Description:

1. Create a table known as *FirstTable* using database schema CF3107, create the first column as a character length 45.
2. Create *SampleInsertionRecord.jsp* page to process and acknowledge the user upon inserting record in the database.

Estimated time: 20 minutes

Step 1 - Create a table namely *FirstTable* using phpMyAdmin

1. Start XAMPP control panel.
2. Start the Apache web server.
3. Start the MySQL database.

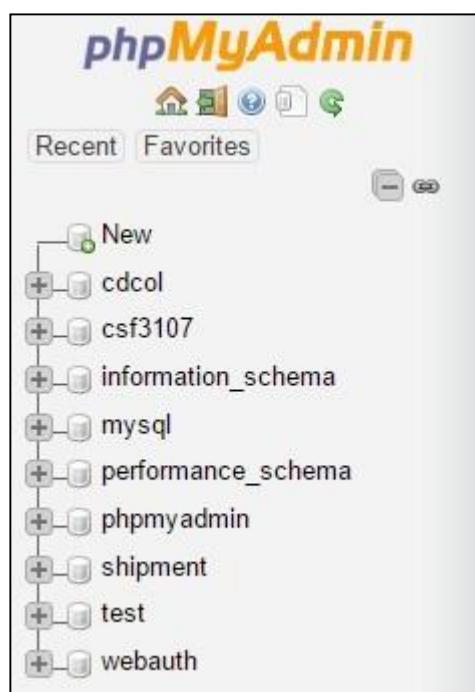


4. Click the *Admin* button for MySQL.
5. Go to *Database's* tab.

6. Key-in as *CSF3107* and click *Create* button.

The screenshot shows the MySQL Workbench interface with the 'Databases' tab selected. A 'Create database' dialog is open, containing a text input field with the value 'Csf3107'. Below the input field are dropdown menus for 'Collation' and 'Character set'. A 'Create' button is visible. A note at the bottom of the dialog box reads: '⚠ Note: Enabling the database statistics here might cause heavy traffic between the database and the server.' The 'Database' and 'Collection' tabs are also visible at the bottom of the dialog.

7. Database schema successfully created.



8. Use any tool to manipulate the SQL statement. Create table ***FirstTable*** in ***csf3107*** database schema.

9. Create ***FirstTable***'s table.

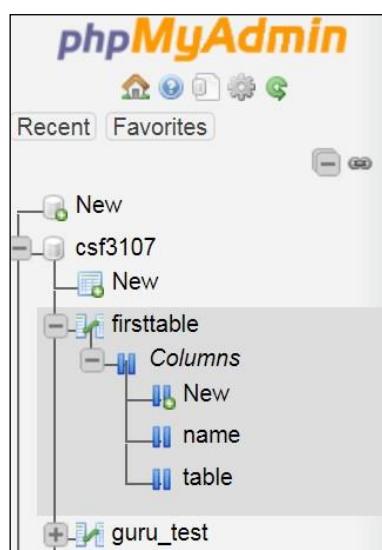


A screenshot of the MySQL Workbench interface. On the left, there's a tree view showing a 'Recent' section with 'New' and 'csf3107' entries, and a 'Favorites' section with 'New' and 'firetable'. On the right, a large window titled 'Run SQL query/queries on database csf3107:' contains a single line of SQL code:

```
1 CREATE TABLE FirstTable(NAME VARCHAR(45))
```

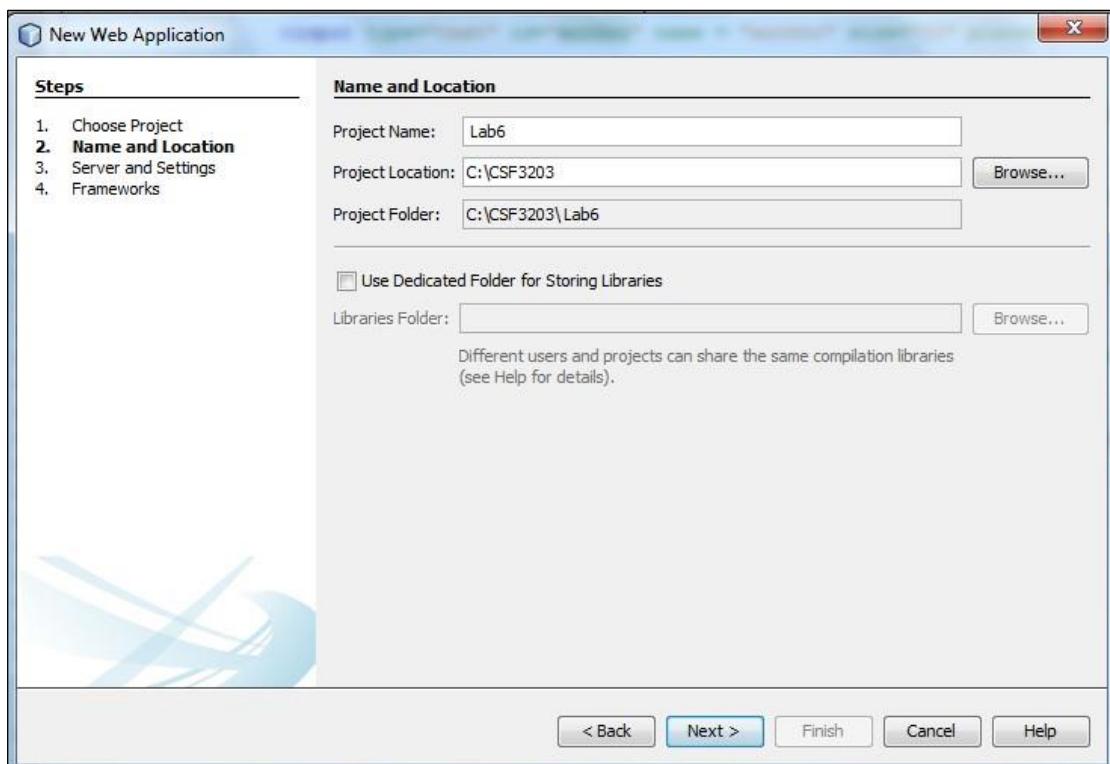
10. Execute the SQL statement.

11. Table successfully created.

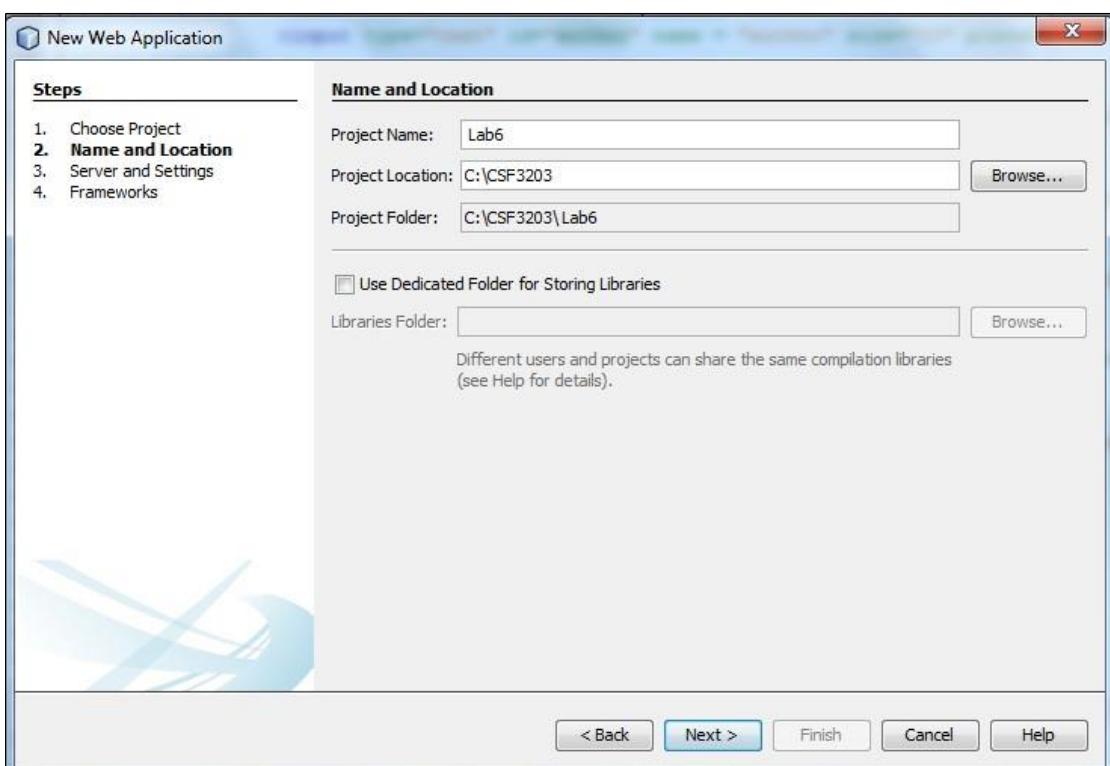


Step 2 - Create *SampleInsertRecord.jsp* to insert data in *FirstTable* table.

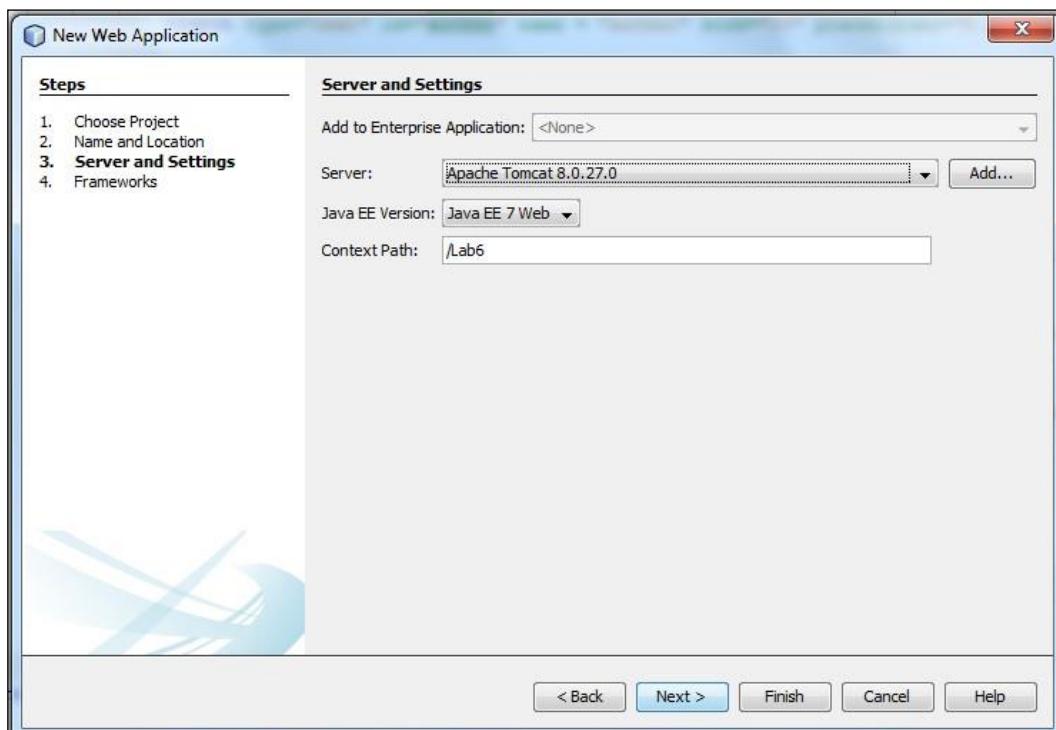
1. Go to C:\CSF3107 Lab's directory and create sub-directory as Lab 6
2. Go to NetBeans.
3. Go to File -> New Project.
4. Select Java Web -> Web Application.
5. Click the *Next* button.
6. Type Project Name: *Lab6*.



7. Choose Project Location: C:\CSF3107\Lab6

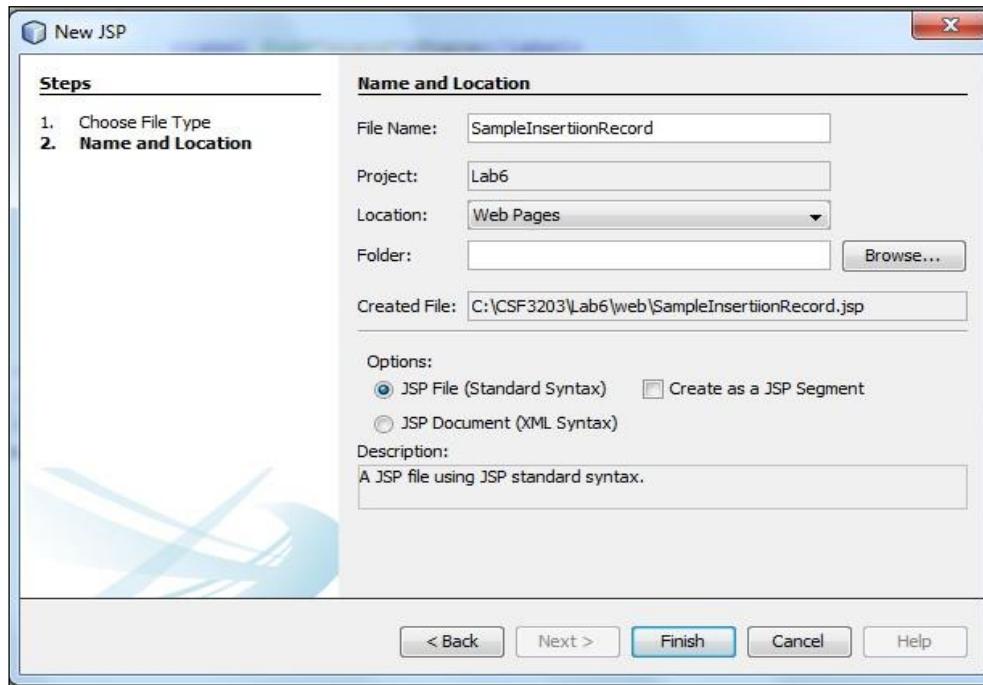


8. Click the *Next* button.
9. Select Server: *Apache Tomcat*.
10. Select Java EE Version: Java EE 7 Web.



11. Click the *Next* button.
12. Click the *Finish* button.

13. Create a new JSP's page for and rename *SampleInsertionRecord*.



14. Type header1 as *Lab 6 - Task 1 - Sample Insertion records into MySQL through JSP's page*.

```
<h1>Lab 6 - Task 1 - Sample Insertion records into MySQL through JSP's page</h1>
<%>
%>
```

15. To support the database driver, we need to use JSP Page Directive to provide directions and instructions to the container.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page language="java"%>
<%@page import="java.sql.*"%>
```

16. Write the following code:

```
<%
    int result;

    //Step 1: Load JDBC driver...
    Class.forName("com.mysql.jdbc.Driver");
    out.println("Step 1: MySQL driver loaded...!");

%>
<br>

<%
    //Step 2: Establish the connection...
    String myURL = "jdbc:mysql://localhost/csf3107";
    Connection myConnection = DriverManager.getConnection(myURL, "root", "admin");
    out.println("Step 2: Database is connected...!");

%>
<br>

<%
    //Step 3: Create a PreparedStatement object...
    out.println("Step 3: Prepared Statements created...!");

    //Prepared SQL Query as a String...
    String sInsertQry = "INSERT INTO FirstTable VALUE(?)";

    //Call method preparedStatement
    PreparedStatement myPS = myConnection.prepareStatement(sInsertQry);

%>
<br>
```

```
<%
    //Assign each value to respective columns for Book's table... (C-Create)
    out.println("Step 4: Perform insertion of record...!");
    String name = "Welcome to access MySQL database with JSP. ....!";
    myPS.setString(1, name);

    result = myPS.executeUpdate();

    if (result > 0) {
%>
<br>

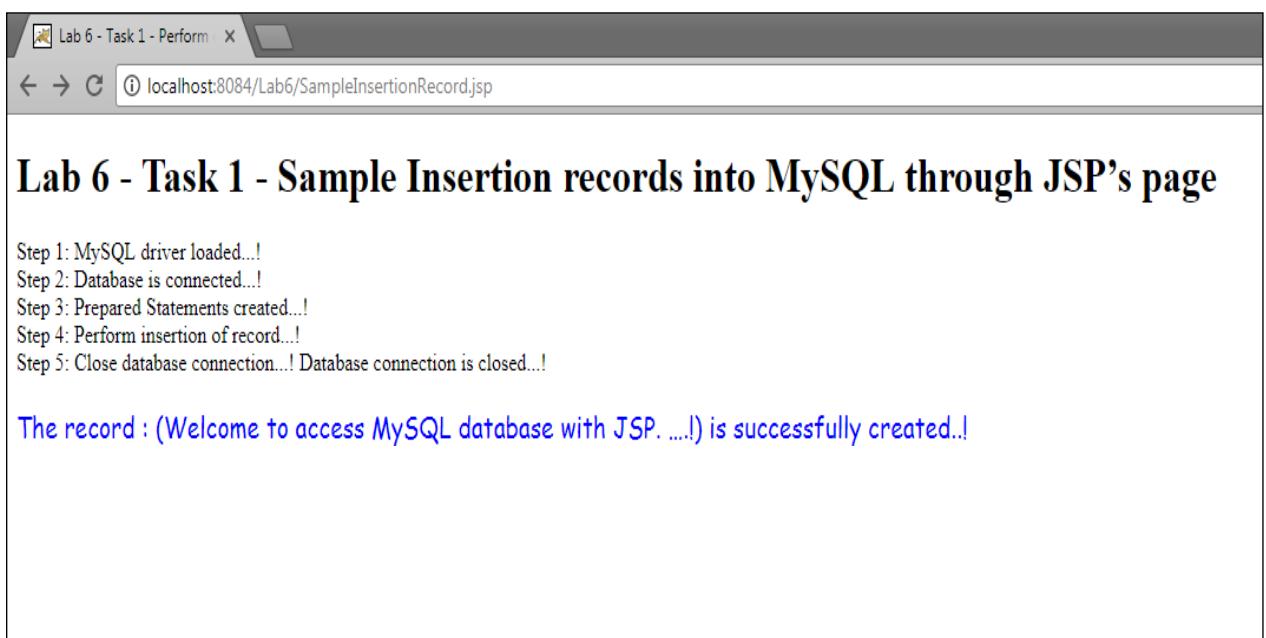
<%
        out.println("Step 5: Close database connection...!");

        out.println(" ");
        out.println("Database connection is closed...!");

        out.print("<p>" + "The record : (" + name
                + ") is successfully created..!" + "</p>");
    }
    //Step 5: Close database connection...
    myConnection.close();
%>
```

17. Save and compile **SampleInsertRecord.jsp** file.

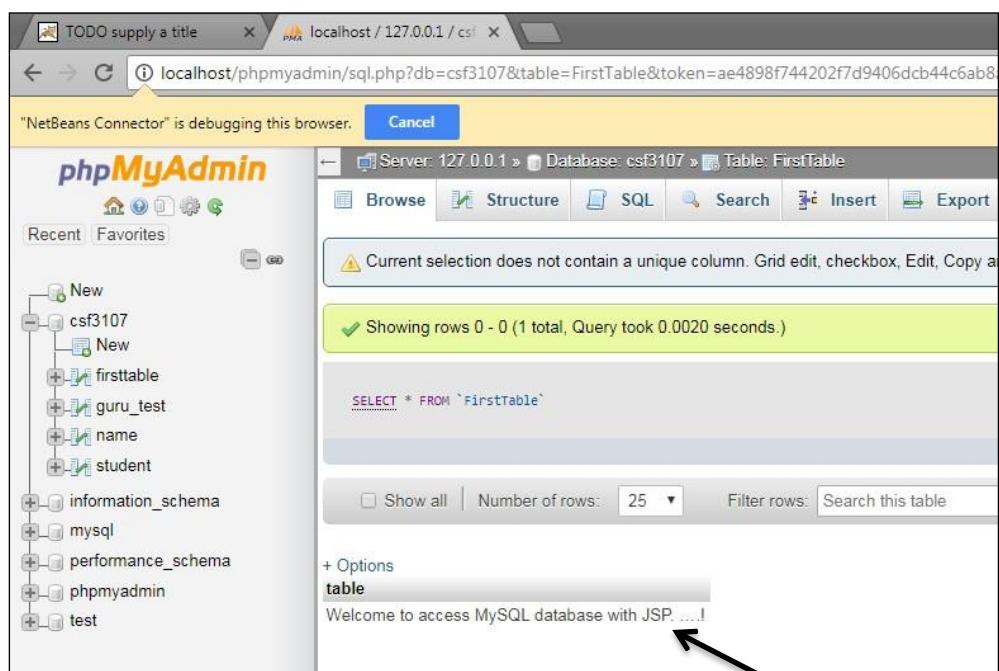
18. Run the **SampleInsertRecord.jsp** file and sample of output is shown below:



Step 3 - go to the database

1. Go to Database schema (csf3107)

2. Click on-> *csf3107* -> *FirstTable* -> then *Browser* (see the data is already there!!)



MY WORK

Database was created

```
1 • CREATE DATABASE csm3023_lab6;
2
3 • CREATE TABLE firsttable(
4     NAME VARCHAR(45));
-
```

SampleInsertionRecord.jsp

The screenshot shows a code editor with Java Server Page (JSP) code. The code is used to insert a record into a MySQL database named 'csm3023_lab6' containing a table named 'firsttable'. The code is divided into several sections:

- Document header: Shows the document type, creation date (22 May 2024), and author (rynaa).
- Page declarations: Includes page content type (text/html), encoding (UTF-8), and language (java).
- HTML structure: Starts with an HTML document.
- Head section: Contains meta tags for content-type and title (JSP Page).
- Body section:
 - A heading: <h1>Lab 6 - Task 1 - Sample Insertion records into MySQL through JSP's page</h1>
 - Step 1: Prints "MySQL driver loaded..." after loading the JDBC driver.
 - Step 2: Establishes a connection to the database using the URL "jdbc:mysql://localhost:3306/csm3023_lab6".
 - Step 3: Creates a PreparedStatement object.
 - Step 4: Prepares an SQL query: "INSERT INTO first_table VALUES(?)".
 - Step 5: Calls the prepareStatement method on the connection.
 - Step 6: Assigns a value to the prepared statement placeholder.
 - Step 7: Executes the update operation.
 - Step 8: Prints a success message if the update was successful.
 - Step 9: Closes the database connection.

Output

Lab 6 - Task 1 - Sample Insertion records into MySQL through JSP's page

Step 1: MySQL driver loaded...!

Step 2: Database is connected..!

Step 3: Prepared Statements created....!

Step 4: Perform insertion of record..!

Step5: Close data connection...! Database connection is closed...!

The record : (Welcome to access MySQL database with JSP....!) is successfully created....!

Result Grid		Filter Rows:	Export:
	name		
▶	Welcome to access MySQL database with JSP....!		
	Welcome to access MySQL database with JSP....!		

Task 2: Create Records via JSP Page

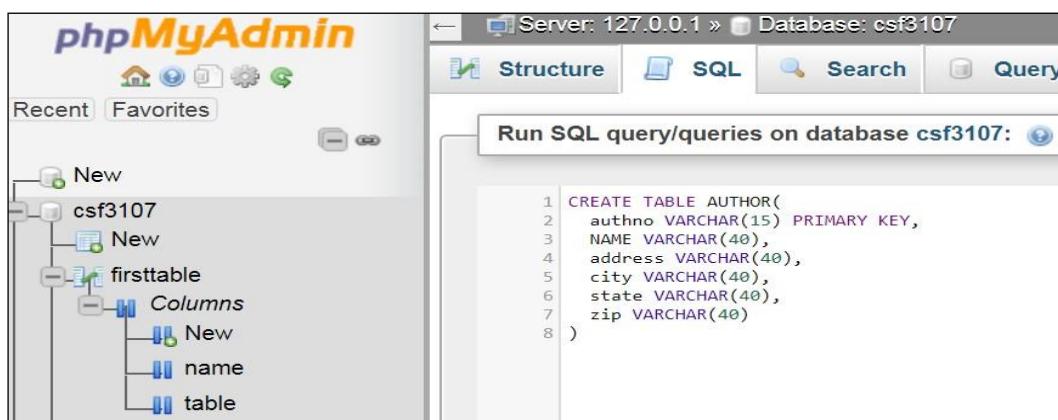
Objective: Using JSP to insert records retrieve from MySQL database.

- Problem Description:**
1. Create a table known as **Author** using database schema CF3107 using these attributes:
 - **authno** as a character length 15 and must be primary key name.
 - **address** as a character length 40
 - **city** as a character length 40
 - **state** as a character length 40
 - **zip** as a character length 40
 2. Create **insertAuthor.jsp** as a main interface to register a new author.
 3. Create **processAuthor.jsp** page to process and acknowledge the user upon inserting record in the database.

Estimated time: 40 minutes

1. Use any tool to manipulate the SQL statement. Create table **author** in **csf3107** database schema.

2. Create **author's** table.

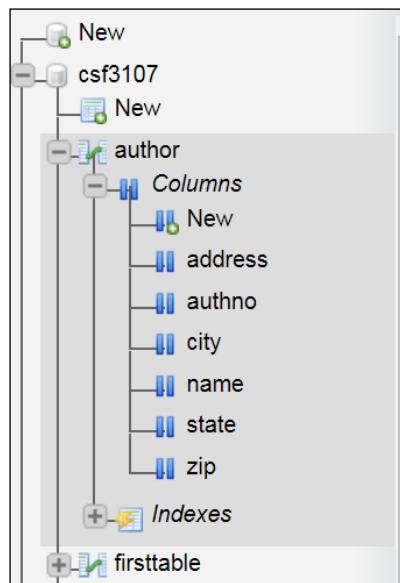


The screenshot shows the phpMyAdmin interface. On the left, the database structure is displayed for the 'csf3107' database, which contains a single table named 'firsttable'. This table has three columns: 'name', 'table', and 'Columns'. On the right, the SQL tab is active, showing the following SQL code:

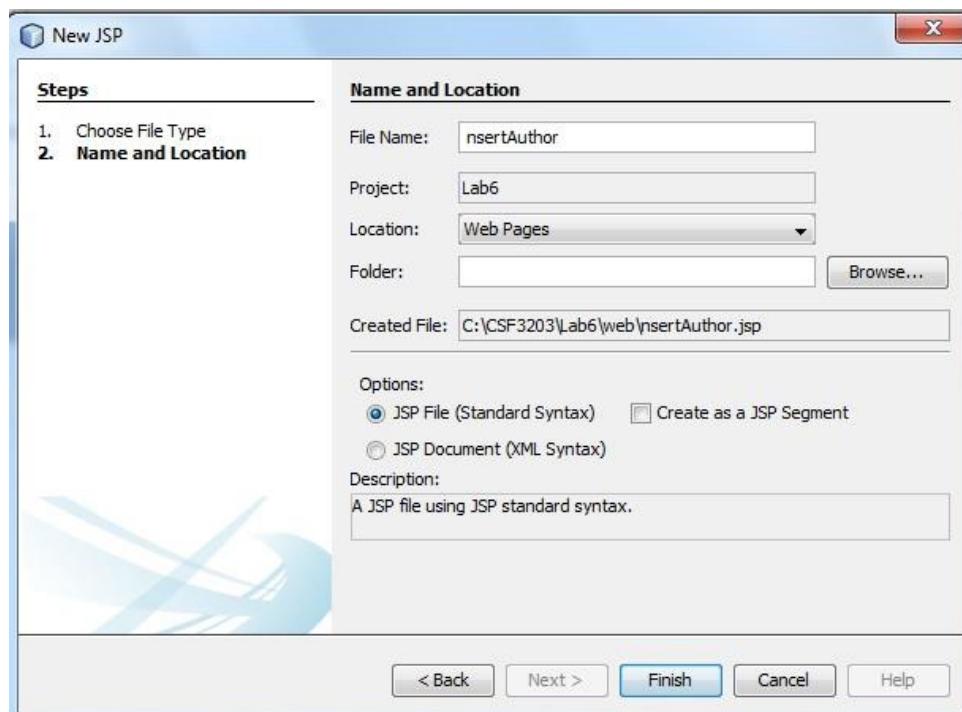
```
CREATE TABLE AUTHOR(
    authno VARCHAR(15) PRIMARY KEY,
    NAME VARCHAR(40),
    address VARCHAR(40),
    city VARCHAR(40),
    state VARCHAR(40),
    zip VARCHAR(40)
)
```

3. Execute the SQL statement.

4. Table successfully created.



5. Create a new JSP page and rename as *insertAuthor*.



6. Write an HTML code to

- a. Display six (6) labels and textfields representing *Author No*, *Name*, *Address*, *City*, *State* and *Zip* (in the combo box).
- b. Create a *Submit* button and *Cancel* button.
- c. Upon submission, redirect the page to *processAuthor.jsp* page.

7. Produce the following output;

Lab 6 - Task 2

localhost:8084/Lab6/insertAuthor.jsp

Lab 6 - Task 2 - Perform creating and retrieving records via JSP page

Author Registration

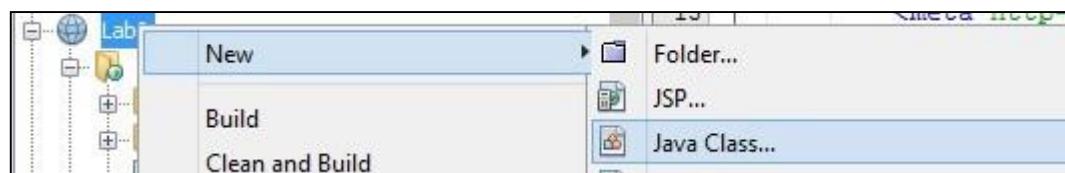
Author No	E.g.: UKXXXXX
Name	Enter your name
Address	Enter your address
City	Enter your city
State	Enter your state
Zip	Enter your zip

Submit Cancel

©2018-Dr.Faizah Binti Aplop

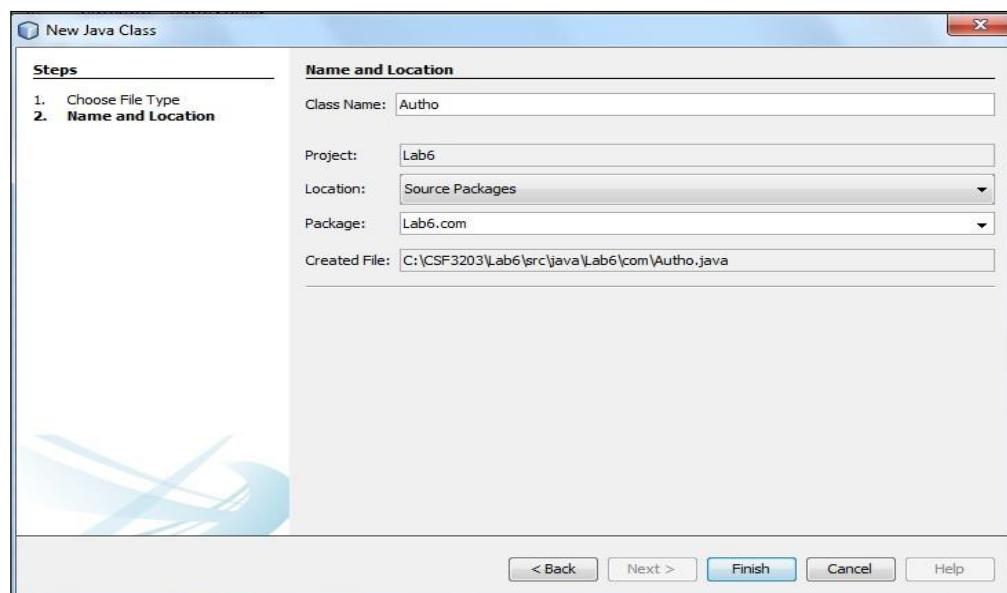
8. Go to *Lab6* project folder.

9. Right click -> New -> Java Class



10. Click Java Class

11. Rename Class Name as *Author* and package as *lab6.com*.



12. Define Six (6) instance variables for *Author* class.

```
/*
package Lab6.com;

/**
 *
 * @author fd
 */
public class author {

    private String authno;
    private String name;
    private String address;
    private String city;
    private String state;
    private String zip;
```

13. Define the *getter* and *setter* method for corresponding attributes.

```
public String getAuthno() {
    return authno;
}

public void setAuthno(String authno) {
    this.authno = authno;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}

public String getCity() {
    return city;
}

public void setCity(String city) {
    this.city = city;
}

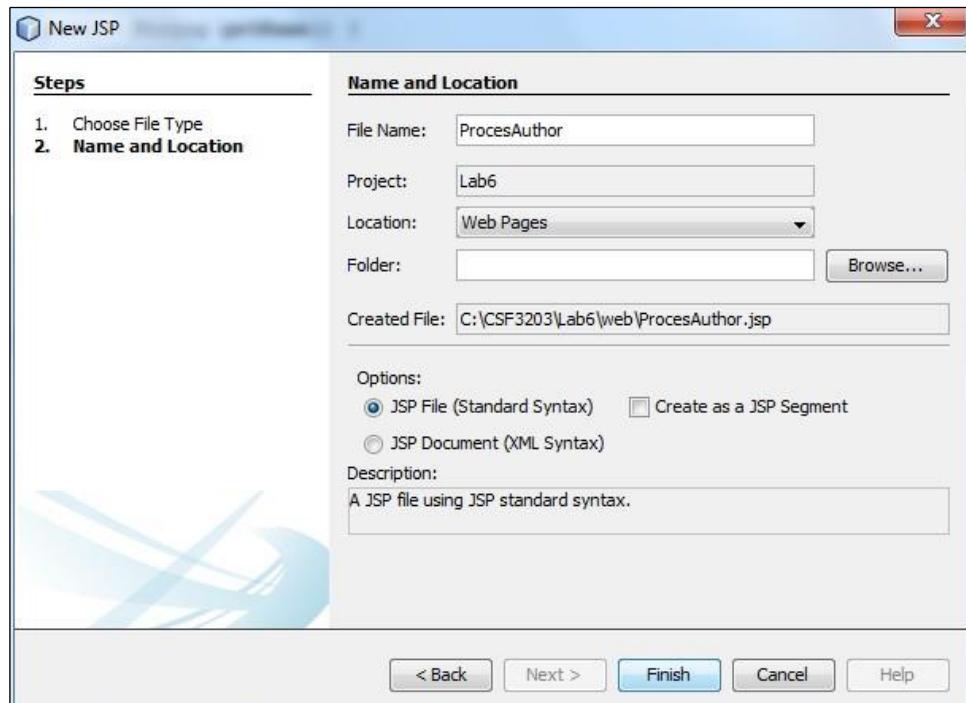
public String getState() {
    return state;
}

public void setState(String state) {
    this.state = state;
}

public String getZip() {
    return zip;
}

public void setZip(String zip) {
    this.zip = zip;
}
```

14. Create a new JSP page as *processAuthor*.



15. Add the page directive to *processAuthor.jsp* page.

```
6
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <%@page language="java"%>
9  <%@page import="java.sql.*"%>
```

16. Create an *author*'s object using JSP Standard Action tag.

```
<jsp:useBean id="myAthour" class="Lab6.com.author" scope="request"/>
```

17. Assign data entry from page *insertAuthor.jsp* page into author's bean.

18. Load the database driver and create a connection to the database.

```
<h1>Lab 6 - Task 1 - Perform creating and retrieving records via JSP page</h1>

<jsp:setProperty name="myAuthor" property="*"/>

<%
    int result;

    Class.forName("com.mysql.jdbc.Driver");

    String myURL = "jdbc:mysql://localhost/csf3107";
    Connection myConnection = DriverManager.getConnection(myURL, "root", "admin");
```

19. Create a *PreparedStatement*'s object.

```
String sInsertQry = "INSERT INTO Author(authno, name, address, city, state, zip) VALUES(?, ?, ?, ?, ?, ?)";

PreparedStatement myPS = myConnection.prepareStatement(sInsertQry);

myPS.setString(1, myAuthor.getAuthno());
myPS.setString(2, myAuthor.getName());
myPS.setString(3, myAuthor.getAddress());
myPS.setString(4, myAuthor.getCity());
myPS.setString(5, myAuthor.getState());
myPS.setString(6, myAuthor.getZip());
```

20. Execute the query and display the result.

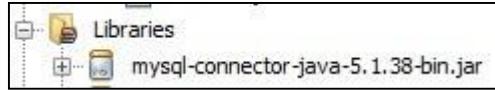
```
result = myPS.executeUpdate();
if (result > 0) {
    out.println("\tRecord successfully added into Author table...!");
    out.print("<p>" + "Record with author no " + myAuthor.getAuthno()
        + " successfully created..!" + "</p>");
    out.print("<p>" + "Details of record are; " + "</p>");
    out.print("<p>Name : " + myAuthor.getName() + "</p>");
    out.print("<p>Address : " + myAuthor.getAddress() + "</p>");
    out.print("<p>City : " + myAuthor.getCity() + "</p>");
    out.print("<p>State : " + myAuthor.getState() + "</p>");
    out.print("<p>Zip : " + myAuthor.getZip() + "</p>");
```

21. Close database connection.

```
//Step 5: Close database connection...
System.out.println("Step 5: Close database connection...!");
myConnection.close();
System.out.println(" ");
System.out.println("Database connection is closed...!");
```

22. Save and compile *processAuthor.jsp* file.

IMPORTANT: Please add **MySQL Java connector** to your project before running the program.



23. Run *insertAuthor.jsp* page.

24. Key-in the record.

25. Click *Submit* button.

4. The record will save in the database, and user get a notification.

Lab 6 - Task 1 - Perform creating and retrieving records via JSP page

Record successfully added into Author table...!

Record with author no gsk23322 successfully created..!

Details of record are;

Name : Fouad

Address : Malaysia

City : KT

State : UMT

Zip : 23

Reflection

1. What have you learnt from this exercise?

2. Define step by step before you successfully perform the transaction in a database.

MY WORK

Author table was created

```
6 • CREATE TABLE AUTHOR(
7     authno VARCHAR(15) PRIMARY KEY,
8     NAME VARCHAR(40),
9     address VARCHAR(40),
10    city VARCHAR(40),
11    state VARCHAR(40),
12    zip VARCHAR(40)
13 );
14
```

insertAuthor.jsp

```
<%-->
1 Document      : insertAuthor
2 Created on   : 12 May 2024, 1:09:28 am
3 Author        : rynaa
4 --%>
5
6 <%@page contentType="text/html" pageEncoding="UTF-8"%>
7 <!DOCTYPE html>
8 <html>
9   <head>
10    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
11    <title>JSP Page</title>
12  </head>
13  <body>
14    <h1>Lab 6 - Task 2 - Perform creating and retrieving records via JSP page</h1>
15    <fieldset>
16      <legend>Author Registration</legend>
17      <form action="ProcessAuthor.jsp" method="post">
18        <table>
19          <tr>
20            <td>
21              <label for="authno">Author No</label>
22            </td>
23            <td>
24              <input type="text" id="authno" name="authno" placeholder="E.g. URXXXXX">
25            </td>
26          </tr>
27          <tr>
28            <td>
29              <label for="name">Name</label>
30            </td>
31            <td>
32              <input type="text" id="NAME" name="NAME" placeholder="Enter your name">
33            </td>
```

```

5      </tr>
6      <tr>
7          <td>
8              <label for="address">Address</label>
9          </td>
10         <td>
11             <input type="text" id="address" name="address" placeholder="Enter your address">
12         </td>
13     </tr>
14     <tr>
15         <td>
16             <label for="city">City</label>
17         </td>
18         <td>
19             <input type="text" id="city" name="city" placeholder="Enter your city">
20         </td>
21     </tr>
22     <tr>
23         <td>
24             <label for="state">State</label>
25         </td>
26         <td>
27             <input type="text" id="state" name="state" placeholder="Enter your state">
28         </td>
29     </tr>
30     <tr>
31         <td>
32             <label for="zip">Zip</label>
33         </td>
34         <td>
35             <input type="text" id="zip" name="zip" placeholder="Enter your zip">
36         </td>
37     </tr>
38     <tr>
39         <td></td>

```

```

68             <tr>
69                 <td>
70                     <button type="submit" value="Submit">Submit</button>
71                     <button type="reset" value="Reset">Cancel</button>
72                 </td>
73             </tr>
74         </table>
75     </form>
76 </fieldset>
77 <footer><p>©2024-Nur Arina</p></footer>
78 </body>
79 </html>

```

output

Author Registration

Author No	<input type="text" value="E.g. UKXXXXX"/>
Name	<input type="text" value="Enter your name"/>
Address	<input type="text" value="Enter your address"/>
City	<input type="text" value="Enter your city"/>
State	<input type="text" value="Enter your state"/>
Zip	<input type="text" value="Enter your zip"/>

©2024-Nur Arina

Author.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Template
 * Click nbfs://nbhost/SystemFileSystem/Template
 */
package com.mycompany.lab6;

/**
 *
 * @author rynaa
 */
public class Author {
    private String authno;
    private String NAME;
    private String address;
    private String city;
    private String state;
    private String zip;

    public String getAuthno() {
        return authno;
    }

    public void setAuthno(String authno) {
        this.authno = authno;
    }

    public String getNAME() {
        return NAME;
    }

    public void setNAME(String NAME) {
        this.NAME = NAME;
    }
}
```

```
34
35     public String getAddress() {
36         return address;
37     }
38
39     public void setAddress(String address) {
40         this.address = address;
41     }
42
43     public String getCity() {
44         return city;
45     }
46
47     public void setCity(String city) {
48         this.city = city;
49     }
50
51     public String getState() {
52         return state;
53     }
54
55     public void setState(String state) {
56         this.state = state;
57     }
58
59     public String getZip() {
60         return zip;
61     }
62
63     public void setZip(String zip) {
64         this.zip = zip;
65     }
66
67 }
```

com.mycompany.lab6.Author >

processAuthor.jsp

```
Source History |           
```

```
<%--  
1 Document : ProcessAuthor  
2 Created on : 12 May 2024, 1:20:17 am  
3 Author : rynaa  
4-->  
5  
6 <%@page contentType="text/html" pageEncoding="UTF-8"%>  
7 <%@page language="java"%>  
8 <%@page import="java.sql.*"%>  
9 <!DOCTYPE html>  
10 <html>  
11 <head>  
12 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
13 <title>JSP Page</title>  
14 </head>  
15 <body>  
16 <h1>Lab 6 - Task 1 - Perform and retrieving records via JSP page</h1>  
17 <jsp:useBean id="myAuthor" class="com.mycompany.lab6.Author" type="request"/>  
18  
19 <jsp:setProperty name="myAuthor" property="*"/>  
20  
21 <%  
22 int result;  
23  
24 Class.forName("com.mysql.cj.jdbc.Driver");  
25  
26 String myURL = "jdbc:mysql://localhost:3306/csm3023_lab6";  
27 Connection myConnection = DriverManager.getConnection(myURL, "root", "admin");  
28  
29 String sInsertQry = "INSERT INTO author(authno, NAME, address, city, state, zip) VALUES(?, ?, ?, ?, ?, ?);  
30  
31 PreparedStatement myPS = myConnection.prepareStatement(sInsertQry);  
32  
33 myPS.setString(1, myAuthor.getAuthno());  
34  
35  
36  
37  
38  
39  
40  
41  
42 if (result > 0){  
43     out.println("\tRecord successfully added into Author table...!");  
44     out.print("<p>" + "Record with author no " +myAuthor.getAuthno()  
45             + " successfully created..!" + "</p>");  
46     out.print("<p>" + "Details of record are; " + "</p>");  
47     out.print("<p>Name : " + myAuthor.getNAME() + "</p>");  
48     out.print("<p>Address : " + myAuthor.getAddress() + "</p>");  
49     out.print("<p>City : " + myAuthor.getCity() + "</p>");  
50     out.print("<p>State : " + myAuthor.getState() + "</p>");  
51     out.print("<p>Zip : " + myAuthor.getZip() + "</p>");  
52 }  
53 //Step5: close database connection..  
54 System.out.println("Step 5: Close database connection..!");  
55 myConnection.close();  
56 System.out.println(" ");  
57 System.out.println("Database connection is closed..!");  
58  
59 %>  
60 </body>  
61 </html>
```

output

localhost:8080/Lab6/ProcessAutho...

Lab 6 - Task 1 - Perform and retrieving records via JSP page

Record successfully added into Author table...!

Record with author no gsk23322 successfully created..!

Details of record are;

Name : Fouad

Address : Malaysia

City : KT

State : UMT

Zip : 23

Task 3: Create Records Constrained by Regular Expression In JSP

Objective: Using JSP Standard Action, scriptlets and regular expression to insert records retrieve from MySQL database.

- Problem Description:**
1. Create a table known as **student** using database schema CF3107 using these attributes:
 - **stuid** as a character length 15 and must be the primary key
 - **stuname** as a character length 50
 - **stuprogram** as a character length 40
 - **address** as a character length 40
 2. Create **insertStudent.jsp** as a main interface to register new book.
 3. Create **processStudent.jsp** page to process and acknowledge the user upon inserting record in the database.
 4. Create **displayStudent.jsp** page to populate records.
 5. Create **errorStudent.jsp** to handle an error.

Estimated time: 50 minutes

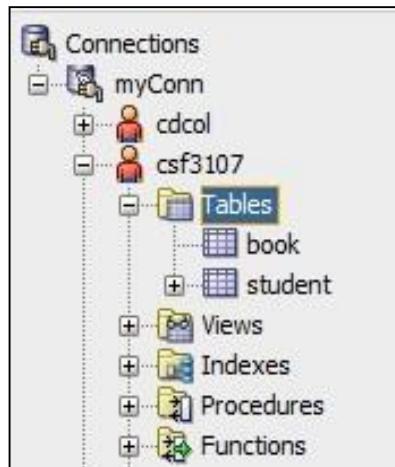
Step 1 - Create a table book using phpMyAdmin

1. Create a table as a student in the *csf3107* database schema.
2. Execute the SQL statement.

The screenshot shows the phpMyAdmin interface. On the left, there's a tree view of the database structure under 'myConn'. The 'Tables' node under 'csf3107' is selected. On the right, the 'Worksheet' tab is active, displaying the following SQL code:

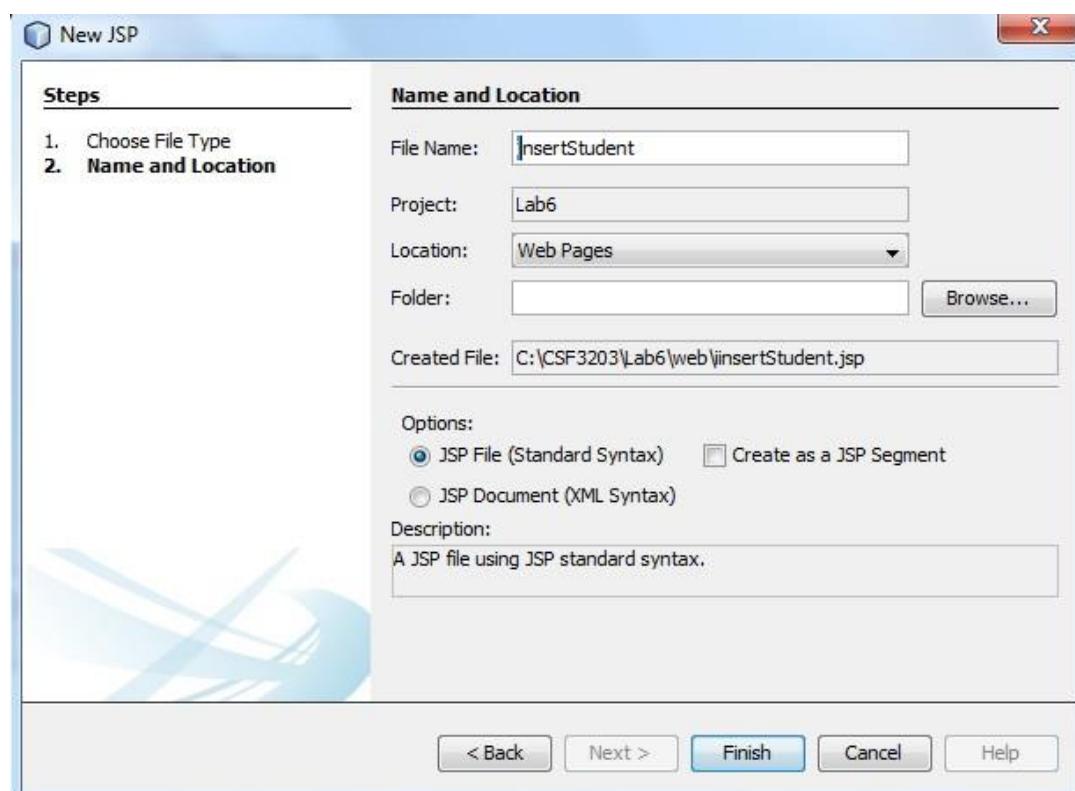
```
CREATE TABLE student(
    stuno    VARCHAR(15) PRIMARY KEY,
    stuname  VARCHAR(50),
    stuprogram  VARCHAR(40)
)
```

3. Table successfully created.



Step 2 - Create *insertStudent.jsp* as a main interface to register a new student

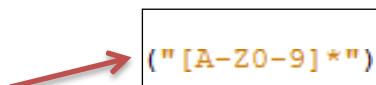
1. Create a new JSP's page and rename as *insertStudent*.



2. Write an HTML code to

- a. Display three (3) labels and textfields representing *Student ID*, *Name* and *Program (in the combo box)*.
- b. The first field must be started with captain letters then numbers input.

(Use the following regular expression in *Book JavaBeans* file)



- c. Create a *Submit* button and *Cancel* button.

3. Produce the following output;

The screenshot shows a "Student Registration" form. It includes fields for "Student No" (with placeholder "E.g.: UIXXXXX"), "Name" (with placeholder "Enter your name"), and "Program" (a dropdown menu currently showing "BSc. Soft. Eng." which is selected, and other options like "BSc. with IM", "BSc. in Networking", and "BSc. in Robotics"). Below the form are two buttons: "Submit" and "Cancel". At the bottom left of the form area, it says "©2016-Mohamad Nor".

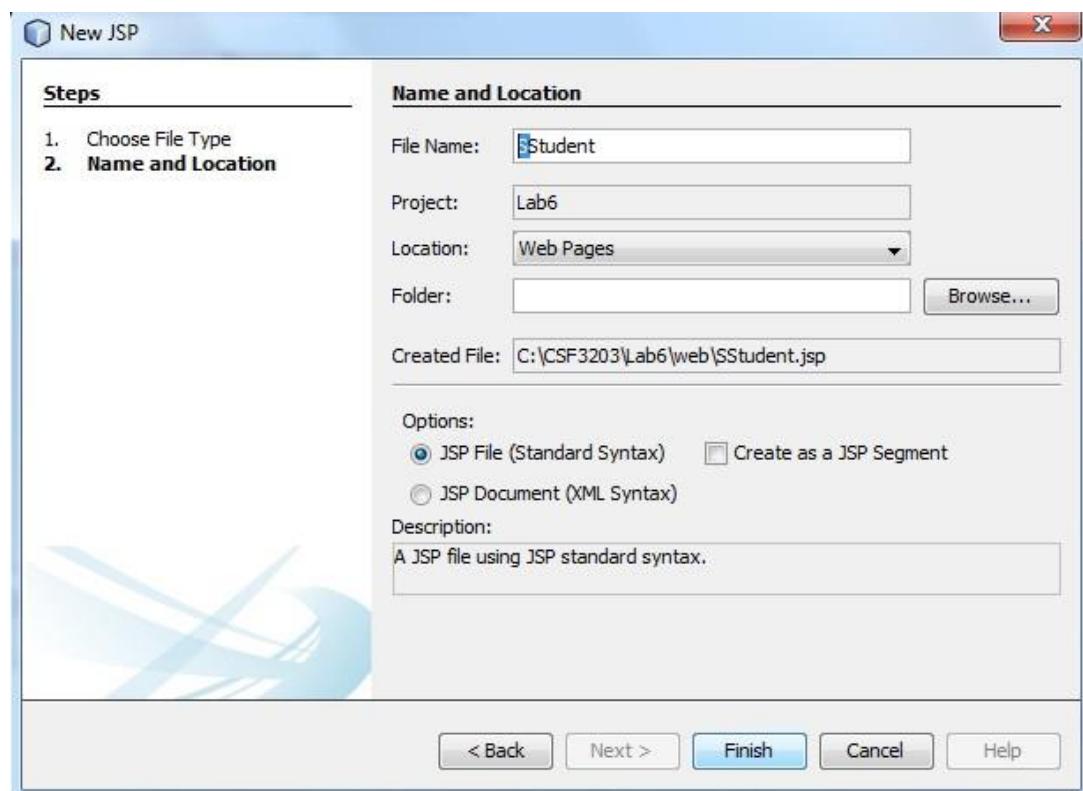
Step 3 - Create Book JavaBeans

1. Go to *Lab6* project folder.
2. Right click -> New -> Java Class



3. Click Java Class

4. Rename Class Name as *Book* and package as *lab6.com*.



5. Define **THREE (3)** instance variables for *Book* class.

```
6  package lab9.com;
7
8  /**
9  *
10 * @author mnor
11 */
12 public class Student
13 {
14     //Create attributes...
15     private String stuno;
16     private String name;
17     private String program;
18 }
```

6. Define the *getter* and *setter* method for corresponding attributes.

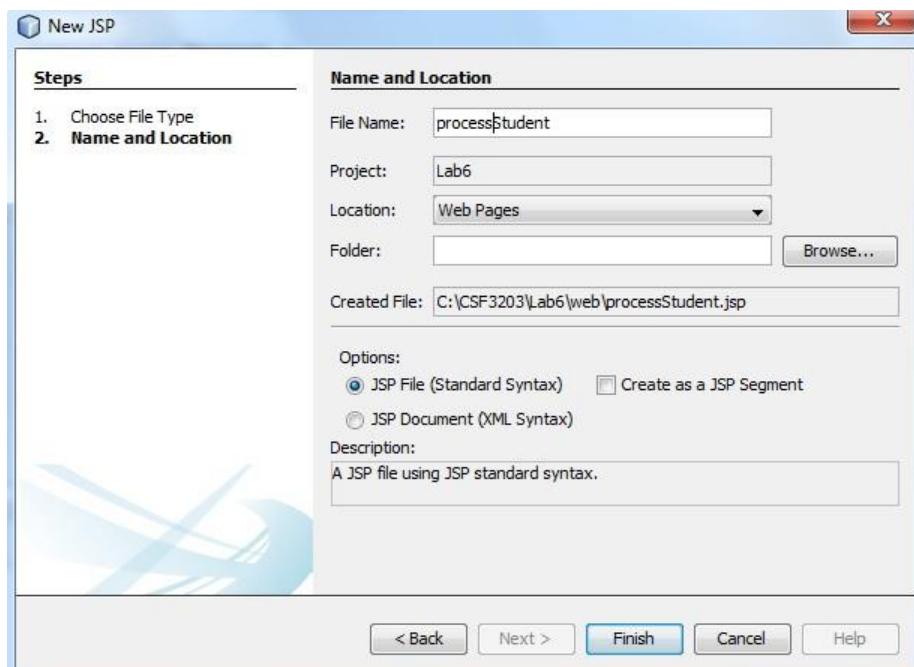
```
28     public String getName() {
29         return name;
30     }
31
32     public void setName(String name) {
33         this.name = name;
34     }
35
36     public String getProgram() {
37         return program;
38     }
39
40     public void setProgram(String program) {
41         this.program = program;
42     }
43 }
```

7. Define *getter* and *setter* method plus regular expression for *stuno* attribute.

```
public String getStuno() {
    Pattern pt = Pattern.compile("[A-Z0-9]*");
    Matcher mt = pt.matcher(stuno);
    boolean bl = mt.matches();
    if (bl == true) {
        valid = stuno;
    } else {
        valid = invalid;
    }
    return valid;
}
public void setStuno(String stuno) {
    this.stuno = stuno;
}
```

Step 4 - Create *processBook.jsp* to insert a record into the database

1. Create a new JSP's page for and rename as *processStudent*.



2. Add the page directive to *processStudent.jsp* page.

```
1  <%--  
2   Document  : processStudent  
3   Created on : 27-Apr-2016, 15:38:30  
4   Author    : Mohamad Nor Hassan  
5  --%>  
6  
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>  
8  <%@page language="java"%>  
9  <%@page import="java.sql.*"%>  
10 <%@page errorPage="errorStudent.jsp" %>  
11
```

3. Create a *Student*'s object using JSP Standard Action tag.

```
17 <%-- Create an object for Student-->  
18 <jsp:useBean id="myStudent" class="Lab6.com.Student" scope="request"/>  
19
```

4. Assign data entry from page *insertStudent.jsp* page into Student's bean.

```
-->  
23 <%-- Assign data entry from page insertStudent.jsp page into Student's bean-->  
24 <jsp:setProperty name="myStudent" property="*"/>
```

5. Load the database driver and create a connection to the database.

```
<%
    int result;

    //Step 1: Load JDBC driver...
    Class.forName("com.mysql.jdbc.Driver");
    System.out.println("Step 1: MySQL driver loaded...!");

    //Step 2: Establish the connection...
    String myURL = "jdbc:mysql://localhost/csf3107";
    Connection myConnection = DriverManager.getConnection(myURL, "root", "admin");
    System.out.println("Step 2: Database is connected...!");
```

6. Create a *PreparedStatement*'s object.

```
//Step 3: Create a PreparedStatement object...
System.out.println("Step 3: Prepared Statements created...!");

//Prepared SQL Query as a String...
String sInsertQry = "INSERT INTO Student(stuno, stuname, stuprogram) VALUES(?, ?, ?)" ;
System.out.println("\tSQL Query: " + sInsertQry);

//Call method preparedStatement
PreparedStatement myPS = myConnection.prepareStatement(sInsertQry);

//Assign each value to respective columns for Book's table... (C-Create)
System.out.println("Step 4: Perform insertion of record...!");
myPS.setString(1, myStudent.getStuno());
myPS.setString(2, myStudent.getName());
myPS.setString(3, myStudent.getProgram());
```

7. Execute the query and display the result.

```
//Step 4: Execute the query...
result = myPS.executeUpdate();
if ( result > 0 )
{
    System.out.println("\tRecord successfully added into Book's table...!");
    out.print("<p>" + "Record with student no " + myStudent.getStuno() +
              " successfully created..!" + "</p>");
    out.print("<p>" + "Details of record are; " + "</p>");
    out.print("<p>Student ID : " + myStudent.getStuno() + "</p>");
    out.print("<p>Name : " + myStudent.getName() + "</p>");
    out.print("<p>Program : " + myStudent.getProgram() + "</p>");
}
```

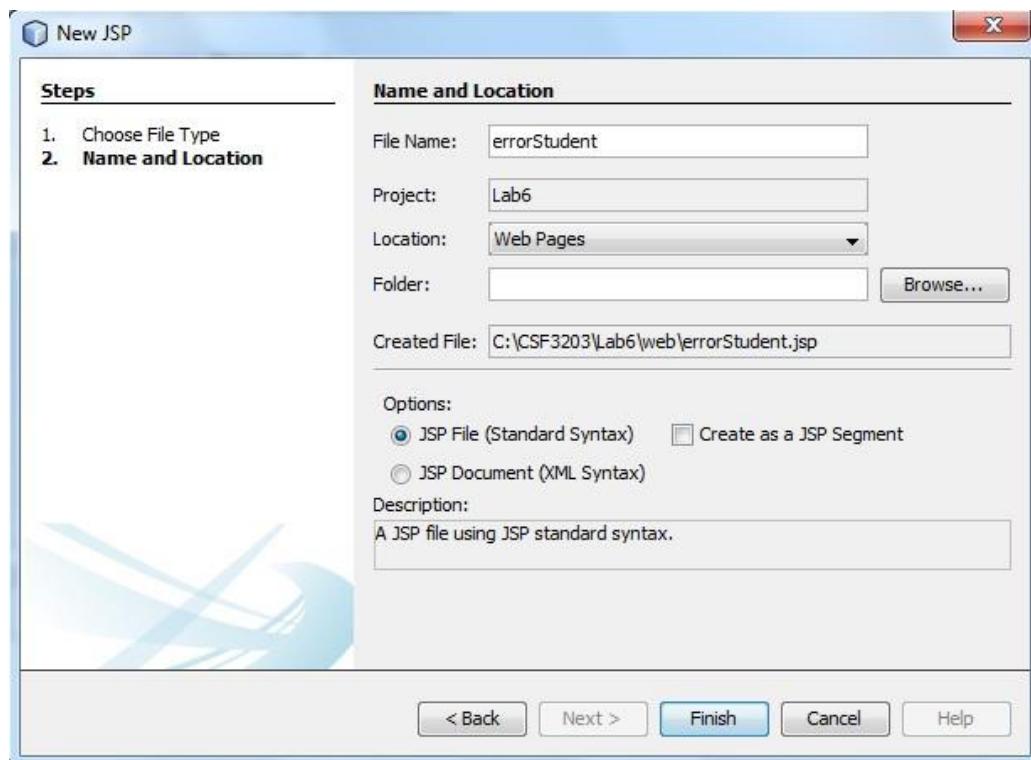
8. Close database connection.

```
//Step 5: Close database connection...
System.out.println("Step 5: Close database connection...!");
myConnection.close();
System.out.println(" ");
System.out.println("Database connection is closed...!");

%>
```

Step 5 - Create *errorBook.jsp* to display any error message

1. Create new JSP's file and rename as *errorStudent.jsp*.



2. Define the page directive to declare that this is an error page.
3. Complete remaining of code.

```
15      </head>
16  
```

```
17  
```

```
18      <form id="errorFrm" action="insertStudent.jsp" method="post">
19          <h1>Lab 9 - Task 1 - Perform creating and retrieving records via JSP page</h1>
20          <_ : when inserting record...!</p>
21          <p><jsp:expression> exception.getMessage() </jsp:expression></p>
22          <br>
23      </form>
24  
```

```
</body>
</html>
```

The code editor displays the JSP file 'errorStudent.jsp'. The code includes a head section, a body section starting with a form tag. Inside the form, there is an h1 header with the text 'Lab 9 - Task 1 - Perform creating and retrieving records via JSP page'. Below the h1 is a p tag containing a jsp:expression that outputs the exception message. The code is numbered from 15 to 24 on the left.

4. Save and compile *errorStudent.jsp*'s file

Step 6 - Running the program and create a new database

1. Run *insertStudent.jsp* page.

2. Key-in the record.

Student Registration

Student No: UK88888

Name: Mohamad Nor Hassan

Program: BSc. in Networking

Submit Cancel

©2016-Mohamad Nor

3. Click *Submit* button.

4. The record will save in the database, and user get a notification.

Record with student no UK88888 successfully created..!

Details of record are;

Student ID : UK88888

Name : Mohamad Nor

Program : BSc. in Networking

Reflection

1. What have you learnt from this exercise?

2. Define step by step before you successfully perform the transaction in a database.

MY WORK

Student table was created

```
15 • CREATE TABLE student(
16     stuno VARCHAR(15) PRIMARY KEY,
17     stuname VARCHAR(50),
18     stuprogram VARCHAR(40)
19 );
20
```

insertStudent.jsp

```
1  <%--%
2   Document : insertStudent
3   Created on : 12 May 2024, 1:58:03 am
4   Author    : rynaa
5   --%>
6
7   <%@page contentType="text/html" pageEncoding="UTF-8"%>
8   <!DOCTYPE html>
9   <html>
10  <head>
11      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12      <title>JSP Page</title>
13  </head>
14  <body>
15      <fieldset>
16          <legend>Student Registration</legend>
17          <form action="processStudent.jsp" method="post">
18              <table>
19                  <tr>
20                      <td>
21                          <label for="stuno">Student No</label>
22                      </td>
23                      <td>
24                          <input type="text" id="stuno" name="stuno" placeholder="E.g.: UKXXXXXX">
25                      </td>
26                  </tr>
27                  <tr>
28                      <td>
29                          <label for="name">Name</label>
30                      </td>
31                      <td>
32                          <input type="text" id="name" name="name" placeholder="Enter your name">
33                      </td>
34                  </tr>
```

```

34          </tr>
35          <tr>
36              <td><label for="program">Program</label></td>
37              <td>
38                  <select id="program" name="program">
39                      <option value="BSc. Soft. Eng.">BSc. Soft. Eng.</option>
40                      <option value="BSc. with IM">BSc. with IM</option>
41                      <option value="BSc. in Networking">BSc. in Networking</option>
42                      <option value="BSc. in Robotics">BSc. in Robotics</option>
43                  </select>
44              </td>
45          </tr>
46          <tr>
47              <td>
48                  <button type="submit" value="Submit">Submit</button>
49                  <button type="reset" value="Reset">Cancel</button>
50              </td>
51          </tr>
52      </table>
53  </form>
54 </fieldset>
55  <footer>
56      <p>© 2024-Nur Arina</p>
57  </footer>
58 </body>
59 </html>

```

errorStudent.jsp

```

1 <%-->
2     Document : errorStudent
3     Created on : 12 May 2024, 2:33:28 am
4     Author   : rynaa
5 --%>
6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE html>
9 <html>
10    <head>
11        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12        <title>JSP Page</title>
13    </head>
14    <body>
15        <form id="errorFrm" action="insertStudent.jsp" method="post">
16            <h1>Lab 6 - Task 1 - Perform creating and retrieving records via JSP page</h1>
17            <p>Error occur when inserting record...!</p>
18            <p>Error Message: ${exception.getMessage()}</p>
19            <br>
20        </form>
21    </body>
22 </html>

```

Student.java

```
/*
 * package com.mycompany.lab6;
 *
 * import java.util.regex.Matcher;
 * import java.util.regex.Pattern;
 */
/**
 * @author rynaa
 */
public class Student {
    private String stuno;
    private String name;
    private String program;

    public String getStuno() {
        Pattern pt = Pattern.compile("[A-Z0-9]*");
        Matcher mt = pt.matcher(stuno);
        boolean bl = mt.matches();
        String valid = "";
        String invalid = "Invalid input please reenter!";
        if(bl == true){
            valid = stuno;
        }else{
            valid = invalid;
        }
        return valid;
    }

    public void setStuno(String stuno) {
        this.stuno = stuno;
    }
}
```

```
33         this.stuno = stuno;
34     }

35     public String getName() {
36         return name;
37     }

38     public void setName(String name) {
39         this.name = name;
40     }

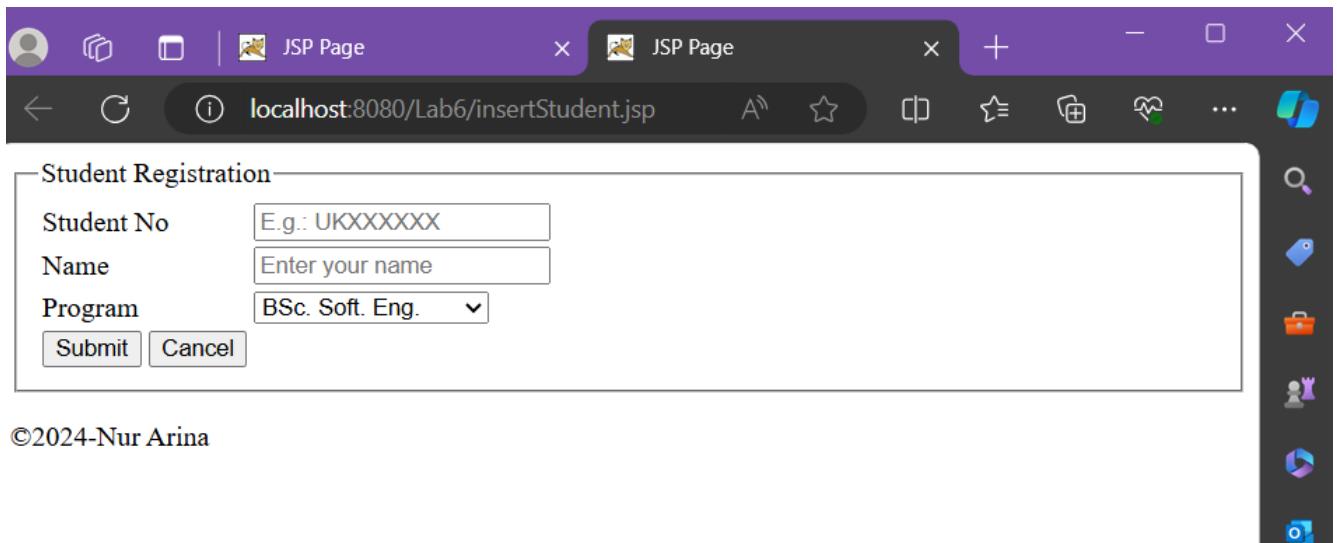
41     public String getProgram() {
42         return program;
43     }

44     public void setProgram(String program) {
45         this.program = program;
46     }

47     public void setProgram(String program) {
48         this.program = program;
49     }

50 }
```

output



©2024-Nur Arina

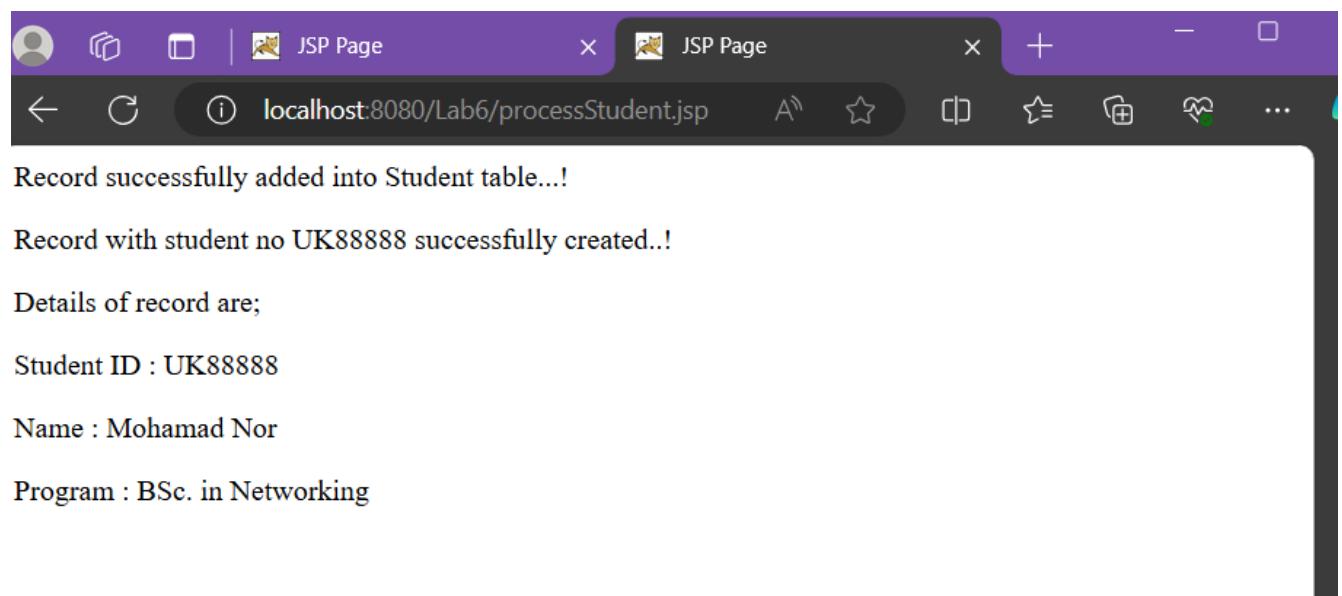
processStudent.jsp

```
<%--  
    Document : processStudent  
    Created on : 12 May 2024, 2:30:34 am  
    Author   : rynaa  
--%>  
  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<%@page language="java"%>  
<%@page import="java.sql.*"%>  
<%@page errorPage="errorStudent.jsp"%>  
<!DOCTYPE html>  
<html>  
    <head>  
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
        <title>JSP Page</title>  
    </head>  
    <body>  
        <jsp:useBean id="myStudent" class="com.mycompany.lab6.Student" scope="request"/>  
        <jsp:setProperty name="myStudent" property="*"/>  
  
        <%  
            int result;  
  
            //Step 1: Load JDBC driver..  
            Class.forName("com.mysql.cj.jdbc.Driver");  
            System.out.println("Step 1: MySQL driver loaded...!");  
  
            //Step 2: Establish the connection  
            String myURL = "jdbc:mysql://localhost:3306/csm3023_lab6";  
            Connection myConnection = DriverManager.getConnection(myURL, "root", "admin");  
            System.out.println("Step 2: Database is connected...!");  
  
            //Step 3: Create a PreparedStatement object...  
            System.out.println("Step3: Prepared Statements created...!");  
        %>
```

```

32         //Step 3: Create a PreparedStatement object...
33         System.out.println("Step3: Prepared Statements created...!");
34         String sInsertQry = "INSERT INTO student(stuno, stuname, stuprogram) VALUES(?, ?, ?)";
35         System.out.println("\tSQL Query: " + sInsertQry);
36
37         //Call method preparedStatement
38         PreparedStatement myPS = myConnection.prepareStatement(sInsertQry);
39
40         //Assign each value to respective columns for Student's table.. (C-Create)
41         System.out.println("Step 4 : Perform insertion of record...!");
42         myPS.setString(1, myStudent.getStuno());
43         myPS.setString(2, myStudent.getName());
44         myPS.setString(3, myStudent.getProgram());
45
46         result = myPS.executeUpdate();
47         if (result > 0){
48             out.println("\tRecord successfully added into Student table...!");
49             out.print("<p>" + "Record with student no " +myStudent.getStuno()
50                     + " successfully created..!" + "</p>");
51             out.print("<p>" + "Details of record are; " + "</p>");
52             out.print("<p>Student ID : " + myStudent.getStuno() + "</p>");
53             out.print("<p>Name : " + myStudent.getName() + "</p>");
54             out.print("<p>Program : " + myStudent.getProgram() + "</p>");
55         }
56     }
57     //Step5: close database connection..
58     System.out.println("Step 5: Close database connection..!");
59     myConnection.close();
60     System.out.println(" ");
61     System.out.println("Database connection is closed..!");
62
63     %>
64 
```

Output



Task 4: Perform Retrieving Records Via JSP Page

Objective: Use Java Scriptlet to query a list of records.

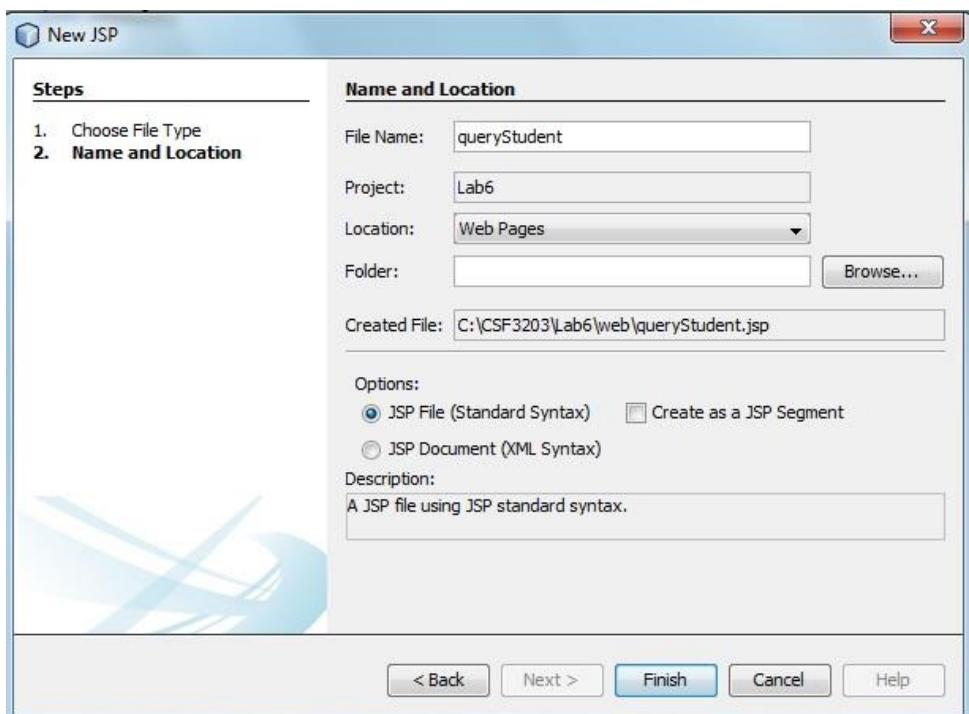
Problem Description: Retrieve student records and populate in the table.

Estimated time: 30 minutes

1. Run program *insertStudent.jsp* from Task 3.
2. Insert the following records;

UK12489	Ahmad Salam	BSc with IM
UK56789	Rosnah Azman	BSc Soft. Eng.
UK67342	Liew Cheng Huat	Bsc in Robotics

3. Go to *Lab6*'s project.
4. Create a new JSP's file.
5. Ke-in file name as *queryStudent*.



6. Rename title as Lab 6 - *Task 3*.

7. Rename `<h1>` as Lab 6 - Task 4 : Retrieving record vis JSP page.
8. Use JSP page directive to include the information such as content type, and use Java SQL API.

```

1  [ ] <%-- 
2      Document    : queryStudent
3      Created on : 27-Apr-2016, 18:01:17
4      Author      : Mohamad Nor Hassan
5  --%>
6  <%@page contentType="text/html" pageEncoding="UTF-8"%>
7  <%@page import="java.sql.*"%>

```

9. Use a Java scriptlet to create a simple structure of HTML table.

```

31  [ ] <%-- 
32          out.print("<table>");
33          out.print("<thead>");
34          out.print("<tr>");
35          out.print("<th>" + "ISBNNo" + "</th>");
36          out.print("<th>" + "Author" + "</th>");
37          out.print("<th>" + "Title" + "</th>");
38          out.print("</tr>");
39          out.print("</thead>");
40          out.print("<tbody>");
41  --%>

```

10. Then, load the database driver and connect into the database.

```

<%
//Step 1: Load JDBC driver...
Class.forName("com.mysql.jdbc.Driver");
System.out.println("Step 1: MySQL driver loaded...!");

//Step 2: Establish the connection...
String myURL = "jdbc:mysql://localhost/csf3107";
Connection myConnection = DriverManager.getConnection(myURL, "root", "admin");
System.out.println("Step 2: Database is connected...!");

```

11. Create Statement for the query.

```
//Step 3: Create a statement object...
Statement myStatement = myConnection.createStatement();
```

12. Perform query to retrieve records from the Student's table.

```
//Step 4: Perform retrieve record from Student's table... (R-Retrieve)
String myQuery = "SELECT * FROM student";
ResultSet myResultSet = myStatement.executeQuery(myQuery);
```

13. Fetch the record into HTML's table.

```
while ( myResultSet.next() )
{
    out.print("<tr>");
    out.print("<td width=\"20%\">" + myResultSet.getString(1) + "</td>");
    out.print("<td width=\"40%\">" + myResultSet.getString(2) + "</td>");
    out.print("<td width=\"40%\">" + myResultSet.getString(3) + "</td>");
    out.print("</tr>");
}
```

14. Close the database connection.

```
//Step 5: Close database connection...
System.out.println("Step 5: Close database connection...!");
myConnection.close();
System.out.println(" ");
System.out.println("Database connection is closed...!");

        out.print("</tbody>");
        out.print("</table>");
%>
```

15. Enhance the CSS for the table.

```
<style>
    table {
        border-collapse: collapse;
    }

    td, th {
        border: 1px solid #999;
        padding: 0.5rem;
        text-align: left;
    }

    th {
        background: gold;
    }
</style>
```

16. Save *queryStudent.jsp*

17. Compile and run *queryStudent.jsp*.

18. You should get the following output.

ISBNNo	Author	Title
UK12489	Ahmad Salam	BSc. with IM
UK56789	Rosnah Azman	BSc. Soft. Eng.
UK67342	Liew Cheng Huat	BSc. in Robotics
UK88888	Mohamad Nor	BSc. in Networking

Reflection

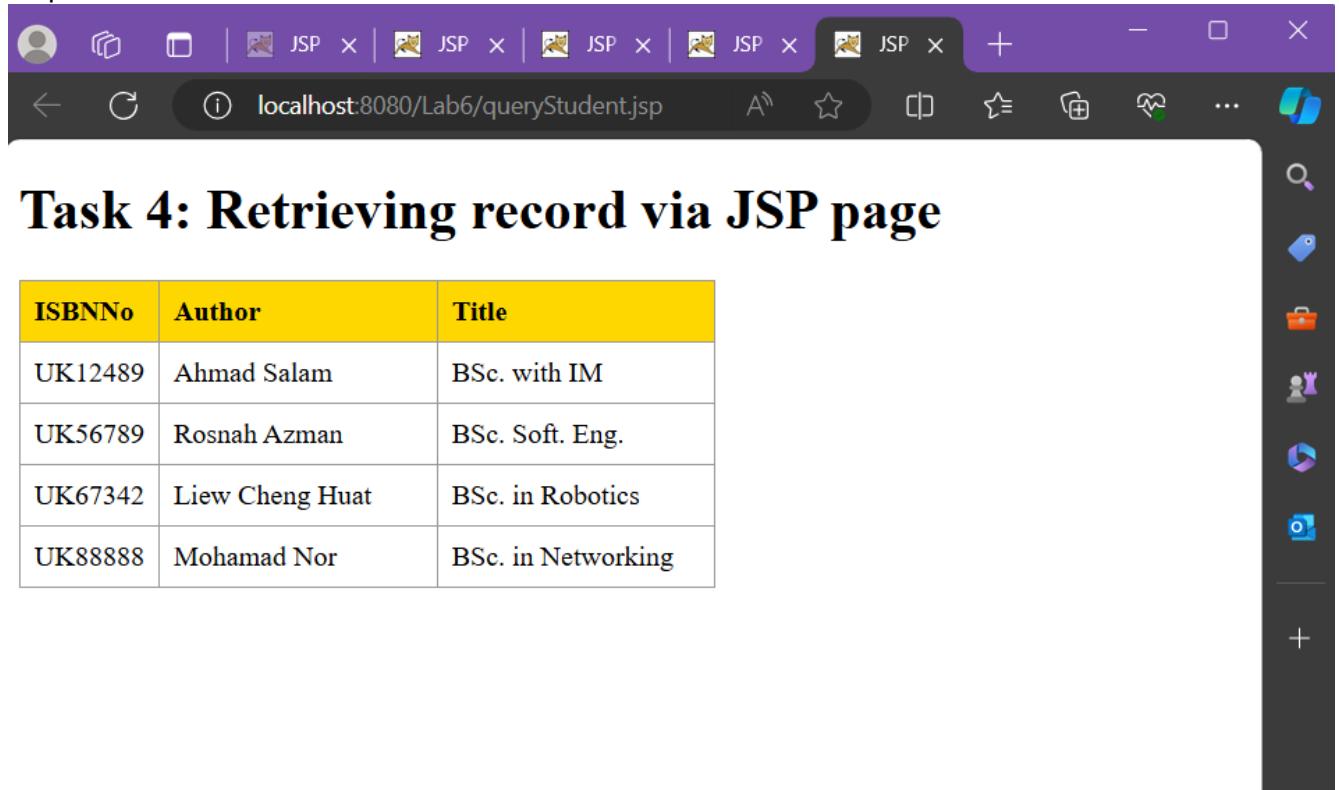
1. What have you learnt from this exercise?
2. Explain the differences when using *Statement()* and *PreparedStatement()*.

MY WORK

queryStudent.jsp

```
1  <%--  
2   Document : queryStudent  
3   Created on : 12 May 2024, 2:58:59 am  
4   Author    : rynaa  
5   --%>  
6  
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>  
8  <%@page import="java.sql.*"%>  
9  <!DOCTYPE html>  
10 <html>  
11  <head>  
12    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
13    <title>JSP Page</title>  
14  </head>  
15  <style>  
16    table {  
17      border-collapse: collapse;  
18    }  
19    td, th {  
20      border: 1px solid #999;  
21      padding: 0.5rem;  
22      text-align: left;  
23    }  
24    th{  
25      background: gold;  
26    }  
27  </style>  
28  <body>  
29    <h1>Task 4: Retrieving record via JSP page</h1>  
30  
31    <%  
32      out.print("<table>");  
33      out.print("<thead>");  
34  
35          out.print("<thead>");  
36          out.print("<tr>");  
37              out.print("<th>" + "ISBNNo" + "</th>");  
38              out.print("<th>" + "Author" + "</th>");  
39              out.print("<th>" + "Title" + "</th>");  
40              out.print("</tr>");  
41          out.print("</thead>");  
42          out.print("<tbody>");  
43      %>  
44      //Step 1: Load JDBC driver..  
45      Class.forName("com.mysql.cj.jdbc.Driver");  
46      System.out.println("Step 1: MySQL driver loaded...!");  
47  
48      //Step 2: Establish the connection  
49      String myURL = "jdbc:mysql://localhost:3306/csm3023_lab6";  
50      Connection myConnection = DriverManager.getConnection(myURL, "root", "admin");  
51      System.out.println("Step 2: Database is connected...!");  
52  
53      //Step 3: Create a PreparedStatement object...  
54      Statement myStatement = myConnection.createStatement();  
55  
56      String myQuery = "SELECT * FROM student";  
57      ResultSet myResultSet = myStatement.executeQuery(myQuery);  
58  
59      while(myResultSet.next()){  
60          out.print("<tr>");  
61              out.print("<td width=\"20%\">" + myResultSet.getString(1) + "</td>");  
62              out.print("<td width=\"40%\">" + myResultSet.getString(2) + "</td>");  
63              out.print("<td width=\"40%\">" + myResultSet.getString(3) + "</td>");  
64          out.print("</tr>");  
65      }  
66      //Step5: close database connection...!  
67      System.out.println("Step 5: Close database connection..!");  
68  
69      //Step5: close database connection...!  
70      System.out.println("Step 5: Close database connection..!");  
71      myConnection.close();  
72      System.out.println(" ");  
73      System.out.println("Database connection is closed..!");  
74  
75      out.print("</tbody>");  
76      out.print("</table>");  
77  
78  %>  
79  
80  </body>  
81  
82 </html>
```

output



The screenshot shows a Microsoft Edge browser window with a purple title bar. The title bar has several tabs, all labeled "JSP" with an "X" icon. The active tab is "localhost:8080/Lab6/queryStudent.jsp". The main content area of the browser displays a table with four rows. The table has three columns: "ISBNNo", "Author", and "Title". The first row is highlighted with a yellow background. The data in the table is as follows:

ISBNNo	Author	Title
UK12489	Ahmad Salam	BSc. with IM
UK56789	Rosnah Azman	BSc. Soft. Eng.
UK67342	Liew Cheng Huat	BSc. in Robotics
UK88888	Mohamad Nor	BSc. in Networking

Task 4: Retrieving record via JSP page

ISBNNo	Author	Title
UK12489	Ahmad Salam	BSc. with IM
UK56789	Rosnah Azman	BSc. Soft. Eng.
UK67342	Liew Cheng Huat	BSc. in Robotics
UK88888	Mohamad Nor	BSc. in Networking

Task 5: Create A Record Using JSP Model 1

Objective: Use JavaBeans to perform SQL transaction.

Problem Description: Create a sample web form to register the Marathon event.

Estimated time: 40 minutes

1. Choose Project *Lab6*.

2. Create a new JSP's file.



3. Type file name as *registerMarathon*.

4. Prepare the following Graphical User Interface (GUI).

The screenshot shows a web page titled "Marathon Registration". The form contains the following fields:

- IC No: A text input field with placeholder text "E.g.: 921110-10-2514".
- Name: A text input field with placeholder text "Enter your name".
- Category: A dropdown menu currently showing "5 KM" as the selected option. Other options in the menu are "5 KM", "7 KM", and "10 KM".
- Buttons: Two buttons at the bottom left labeled "Submit" and "Cancel".

At the bottom of the page, there is a copyright notice: "©2016-Mohamad Nor".

5. Create a JavaBeans *Marathon*.

```
2  */
3   * Bean    : Marathon.java
4   * Author  : Mohamad Nor Hassan
5   * Date    : 27 April 2016
6   */
7   public class Marathon {
8       private String icno;
9       private String name;
10      private String category;
11
12      public String getIcno() {
13          return icno;
14      }
15
16      public void setIcno(String icno) {
17          this.icno = icno;
18      }
19
20      public String getName() {
21          return name;
22      }
23
24      public void setName(String name) {
25          this.name = name;
26      }
27
28      public String getCategory() {
29          return category;
30      }
31
32      public void setCategory(String category) {
33          this.category = category;
34      }
35  }
```

6. Create a *Database* class that has two methods; *getConnection()*, and *closeConnection()*

```
1 package lab9.com;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.SQLException;
7 import java.util.logging.Level;
8 import java.util.logging.Logger;
9 import lab9.com.Marathon;
10
11 /**
12 * Bean      : Database.java
13 * Author   : Mohamad Nor Hassan
14 * Date     : 27 April 2016
15 */
16 public class Database {
17     private static Connection myConnection = null;
18     private static String myURL = "jdbc:mysql://localhost:3306/csf3107";
19     private int result = 0;
20
21     public static Connection getConnection() throws ClassNotFoundException {
22
23         if (myConnection != null) {
24             return myConnection;
25         }
26         else try {
27
28             Class.forName("com.mysql.jdbc.Driver");
29             myConnection = DriverManager.getConnection(myURL, "root", "admin");
30         }
31         catch (SQLException e) {
32             e.printStackTrace();
33         }
34         return myConnection;
35     }
36
37     public void closeConnection() throws ClassNotFoundException
38     {
39         try {
40             myConnection.close();
41         }
42         catch(SQLException e){
43             e.printStackTrace();
44         }
45     }
46 }
```

7. Create a *MarathonDAO* class to perform SQL transaction for business object *Marathon* and store it into package *lab6.com*.

```

1  /*
2  */
3  import java.sql.Connection;
4  import java.sql.PreparedStatement;
5  import java.sql.SQLException;
6  import lab9.com.Database;
7
8  /**
9   * Bean      : MarathonDAO.java
10  * Author    : Mohamad Nor Hassan
11  * Date     : 27 April 2016
12  */
13 public class MarathonDAO
14 {
15     private Connection connection;
16     private int result = 0;
17     public MarathonDAO() throws ClassNotFoundException
18     {
19         connection = Database.getConnection();
20     }
21
22     public int addDetails (Marathon marathon)
23     {
24         try {
25             String mySQL = "INSERT INTO marathon(icno, name, category) values (?, ?, ?)";
26             PreparedStatement preparedStatement = connection.prepareStatement(mySQL);
27
28             System.out.println("IC No      = " + marathon.getIcno());
29             System.out.println("Name       = " + marathon.getName());
30             System.out.println("Category  = " + marathon.getCategory());
31
32             //Parameters
33             preparedStatement.setString(1, marathon.getIcno());
34             preparedStatement.setString(2, marathon.getName());
35             preparedStatement.setString(3, marathon.getCategory());
36             result = preparedStatement.executeUpdate();
37
38         } catch (SQLException e) {
39             e.printStackTrace();
40         }
41         return result;
42     }
43 }

```

8. Create a new file name known as *processMarathon.jsp*.

9. Import related classes in package *lab6.com*.

```

Document      : processMarathon
Created on   : 27-Apr-2016, 19:15:15
Author        : Mohamad Nor Hassan
--%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.sql.*"%>
<%@page import="lab9.com.Database"%>
<%@page import="lab9.com.Marathon"%>
<%@page import="lab9.com.MarathonDAO"%>

```

10. Instantiate an object *Marathon*.

```
<!-- Create an object for Marathon-->
<jsp:useBean id="myMarathon" class="lab6.Marathon" scope="request"/>
```

11. Create a Java Scriptlet to invoke respective object for inserting record in *marathon*'s table.

```
<%
    int result;

    //Step 1: Create Database object...
    Database myDB = new Database();

    MarathonDAO object1 = new MarathonDAO();

    //Step 2: Add the records...
    result = object1.addDetails(myMarathon);

    //Step 3: Determine whether the transaction is sucess...
    if ( result > 0 )
    {
        System.out.println("\tRecord successfully added into Book's table...!");
        out.print("<p>" + "Record with IC No " + myMarathon.getIcno() +
                 " successfully created..!" + "</p>");
        out.print("<p>" + "Details of record are: " + "</p>");
        out.print("<p>Ic No : " + myMarathon.getIcno() + "</p>");
        out.print("<p>Name : " + myMarathon.getName() + "</p>");
        out.print("<p>Category : " + myMarathon.getCategory() + "</p>");
    }

    //Step 4: Close database connection...
    System.out.println("Step 5: Close database connection...!");
    myDB.closeConnection();
    System.out.println(" ");
    System.out.println("Database connection is closed...!");
%>
```

12. Compile and save *processMarathon.jsp*.

13. Run *registerMarathon.jsp* and key-in related record.

Marathon Registration

IC No	890710-11-2369
Name	Mohamad Nor Hassan
Category	7 KM ▾

©2016-Mohamad Nor

14. The output will appear in a web browser.

Record with IC No 890710-11-2369 successfully created..!

Details of record are;

Ic No : 890710-11-2369

Name : Mohamad Nor Hassan

Category : 7 KM

©2016-Mohamad Nor

Reflection

1. What have you learnt from this exercise?

2. Describe the benefits of using JavaBeans.

MY WORK

RegisterMarathon.jsp

```
1  <%--  
2   Document : registerMarathon  
3   Created on : 13 May 2024, 1:40:04 am  
4   Author : rynaa  
5 --%>  
6  
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>  
8  <!DOCTYPE html>  
9  <html>  
10 <head>  
11    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
12    <title>JSP Page</title>  
13  </head>  
14  <body>  
15    <h1>Superhero Marathon 2025</h1>  
16    <fieldset>  
17      <legend>Marathon Registration</legend>  
18      <form action="processMarathon.jsp" method="post">  
19        <table>  
20          <tr>  
21            <td>  
22              <label for="icno">IC No</label>  
23            </td>  
24            <td>  
25              <input type="text" id="icno" name="icno" placeholder="E.g.: 921110-10-2514">  
26            </td>  
27          </tr>  
28          <tr>  
29            <td>  
30              <label for="name">Name</label>  
31            </td>  
32            <td>  
33              <input type="text" id="name" name="name" placeholder="Enter your name">  
34            </td>
```

```
5  </tr>  
6  <tr>  
7    <td>  
8      <label for="category">Category</label>  
9    </td>  
0    <td>  
1      <select id="category" name="category">  
2        <option value="5 KM">5 KM</option>  
3        <option value="7 KM">7 KM</option>  
4        <option value="10 KM">10 KM</option>  
5      </select>  
6    </td>  
7  </tr>  
8  <tr>  
9    <td>  
0      <button type="submit" value="Submit">Submit</button>  
1      <button type="reset" value="Reset">Cancel</button>  
2    </td>  
3  </tr>  
4  </table>  
5  </form>  
6  </fieldset>  
7  <footer>  
8    <p>&copy; Superhero Marathon Asia 2025</p>  
9  </footer>  
0  </body>  
1 </html>
```

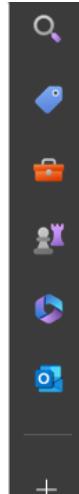
Output

Superhero Marathon 2025

Marathon Registration

IC No	E.g.: 921110-10-2514
Name	Enter your name
Category	5 KM ▾
<input type="button" value="Submit"/>	<input type="button" value="Cancel"/>

©Superhero Marathon Asia 2025



Marathon.java

```
5 package com.mycompany.lab6;
6
7 /**
8 * 
9 * @author rynaa
10 */
11 public class Marathon {
12     private String icno;
13     private String name;
14     private String category;
15
16     public String getIcno() {
17         return icno;
18     }
19
20     public void setIcno(String icno) {
21         this.icno = icno;
22     }
23
24     public String getName() {
25         return name;
26     }
27
28     public void setName(String name) {
29         this.name = name;
30     }
31
32     public String getCategory() {
33         return category;
34     }
35
36     public void setCategory(String category) {
37         this.category = category;
38     }
39
40 }
```

```
3     public void setCategory(String category) {
4         this.category = category;
5     }
6
7 }
```

Database.java

```
5 package com.mycompany.lab6;
6
7 import java.sql.Connection;
8 import java.sql.DriverManager;
9 import java.sql.SQLException;
10 import java.util.logging.Level;
11 import java.util.logging.Logger;
12
13 /**
14 * @author rynaa
15 */
16 public class Database {
17     private static Connection myConnection = null;
18     private static String myURL = myURL = "jdbc:mysql://localhost:3306/csm3023_lab6";
19     private int result = 0;
20
21     public static Connection getConnection() throws ClassNotFoundException{
22         if(myConnection != null){
23             return myConnection;
24         }
25         else try{
26             Class.forName("com.mysql.cj.jdbc.Driver");
27             myConnection = DriverManager.getConnection(myURL, "root", "admin");
28         }
29         catch(SQLException e){
30             e.printStackTrace();
31         }
32         return myConnection;
33     }
34
35     public void closeConnection() throws ClassNotFoundException
36     {
37         try{
38             myConnection.close();
39         }
40         catch(SQLException e){
41             e.printStackTrace();
42         }
43     }
44
45 }
```

```
32         return myConnection;
33     }
34
35     public void closeConnection() throws ClassNotFoundException
36     {
37         try{
38             myConnection.close();
39         }
40         catch(SQLException e){
41             e.printStackTrace();
42         }
43     }
44
45 }
```

MarathonDAO.java

```
5 package com.mycompany.lab6;
6
7 import java.sql.Connection;
8 import java.sql.PreparedStatement;
9 import java.sql.SQLException;
10 import com.mycompany.lab6.Database;
11 /**
12 * @author rynaa
13 */
14 public class MarathonDAO {
15     private Connection connection;
16     private int result = 0;
17     public MarathonDAO() throws ClassNotFoundException{
18         connection = Database.getConnection();
19     }
20     public int addDetails(Marathon marathon){
21         try{
22             String mySQL = "INSERT INTO marathon(icno, name, category) VALUES (?, ?, ?)";
23             PreparedStatement preparedStatement = connection.prepareStatement(mySQL);
24
25             System.out.println("IC no = " + marathon.getIcno());
26             System.out.println("Name = " + marathon.getName());
27             System.out.println("Category= " + marathon.getCategory());
28
29             //Parameters
30             preparedStatement.setString(1, marathon.getIcno());
31             preparedStatement.setString(2, marathon.getName());
32             preparedStatement.setString(3, marathon.getCategory());
33             result = preparedStatement.executeUpdate();
34         }
```

```
34             result = preparedStatement.executeUpdate();
35         }catch(SQLException e){
36             e.printStackTrace();
37         }
38         return result;
39     }
40 }
```

processAuthor.jsp

```
1 <%-- 
2     Document : ProcessAuthor
3     Created on : 16 May 2024, 1:20:17 am
4     Author    : rynaa
5 --%>
6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <%@page language="java"%>
9 <%@page import="java.sql.*"%>
10 <!DOCTYPE html>
11 <html>
12     <head>
13         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
14         <title>JSP Page</title>
15     </head>
16     <body>
17         <h1>Lab 6 - Task 1 - Perform and retrieving records via JSP page</h1>
18         <jsp:useBean id="myAuthor" class="com.mycompany.lab6.Author" scope="request"/>
19
20         <jsp:setProperty name="myAuthor" property="" />
21
22         <%
23             int result;
24
25             Class.forName("com.mysql.cj.jdbc.Driver");
26
27             String myURL = "jdbc:mysql://localhost:3306/csm3023_lab6";
28             Connection myConnection = DriverManager.getConnection(myURL, "root", "admin");
29
30             String sInsertQry = "INSERT INTO author(authno, NAME, address, city, state, zip) VALUES(?, ?, ?, ?, ?, ?)";
31
32             PreparedStatement myPS = myConnection.prepareStatement(sInsertQry);
33
34             myPS.setString(1, myAuthor.getAuthno());
```

```
34         myPS.setString(1, myAuthor.getAuthno());
35         myPS.setString(2, myAuthor.getNAME());
36         myPS.setString(3, myAuthor.getAddress());
37         myPS.setString(4, myAuthor.getCity());
38         myPS.setString(5, myAuthor.getState());
39         myPS.setString(6, myAuthor.getZip());
40
41         result = myPS.executeUpdate();
42         if (result > 0){
43             out.println("\tRecord successfully added into Author table...!");
44             out.print("<p>" + "Record with author no " +myAuthor.getAuthno()
45                     + " successfully created..!" + "</p>");
46             out.print("<p>" + "Details of record are; " + "</p>");
47             out.print("<p>Name : " + myAuthor.getNAME() + "</p>");
48             out.print("<p>Address : " + myAuthor.getAddress() + "</p>");
49             out.print("<p>City : " + myAuthor.getCity() + "</p>");
50             out.print("<p>State : " + myAuthor.getState() + "</p>");
51             out.print("<p>Zip : " + myAuthor.getZip() + "</p>");
52         }
53         //Step5: close database connection..
54         System.out.println("Step 5: Close database connection..!");
55         myConnection.close();
56         System.out.println(" ");
57         System.out.println("Database connection is closed..!");
58     %>
59     </body>
60 </html>
```

Exercise

Implement user login

1. Create a table known as **userprofile** using database schema CF3107 using these attributes.
 - **username** as a character length 15 and must be primary key
 - **password** as a character length 10
 - **firstname** as varchar(50)
 - **lastname** as varchar(50)
2. Create **insertUser.html** as the main interface to register a new user.
3. Create **processUser.jsp** page to process the record.
4. Create **login.jsp** page to login to the system.
5. Create **doLogin.jsp** to validate username and password. If validation is successful, redirect the page to **main.jsp** page that displays the username, firstname and lastname.
6. If validation is unsuccessful, redirect the page to **login.jsp** with message ‘Invalid username or password..!’

MY WORK

Userprofile table was created

```
21 • CREATE TABLE userprofile(
22     username VARCHAR(15) PRIMARY KEY,
23     password VARCHAR(10),
24     firstname VARCHAR(50),
25     lastname VARCHAR(50)
26 );
27
28
29
~~
```

insertUser.html

```
<!DOCTYPE html>
<!--
Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Html.html to edit this template
-->
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Department of Quality UMT</h1>
        <fieldset>
            <legend>User Registration</legend>
            <form action="processUser.jsp">
                <table>
                    <tr>
                        <td>
                            <label for="username">Username</label>
                        </td>
                        <td>
                            <input type="text" id="username" name="username" placeholder="Enter a username">
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <label for="password">Password</label>
                        </td>
                        <td>
                            <input type="password" id="password" name="password" placeholder="Enter a password">
                        </td>
                    </tr>
                    <tr>
                        <td>
```

```

35         <td>
36             <label for="firstname">Firstname</label>
37         </td>
38         <td>
39             <input type="text" id="firstname" name="firstname" placeholder="E.g.:Ecah">
40         </td>
41     </tr>
42     <tr>
43         <td>
44             <label for="lastname">Lastname</label>
45         </td>
46         <td>
47             <input type="text" id="lastname" name="lastname" placeholder="E.g.:bin Daud">
48         </td>
49     </tr>
50     <tr>
51         <td>
52             <button type="submit" value="Submit">Submit</button>
53             <button type="reset" value="Reset">Cancel</button>
54         </td>
55     </tr>
56 </table>
57 </form>
58 <footer>
59     <p><img alt="copy icon" style="vertical-align: middle;"/> Quality UMT - 2024</p>
60 </body>
61 </html>

```

processUser.jsp

```

1 <%--%
2     Document : processUser
3     Created on : 17 May 2024, 2:18:24 pm
4     Author    : rynaa
5 --%>
6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <%@page language="java"%>
9 <%@page import="java.sql.*"%>
10 <%@page errorPage="login.jsp"%>
11 <!DOCTYPE html>
12 <html>
13     <head>
14         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
15         <title>Department of Quality UMT</title>
16     </head>
17     <body>
18         <jsp:useBean id="myUser" class="com.mycompany.lab6.User" scope="request"/>
19         <jsp:setProperty name="myUser" property="*"/>
20
21         <%
22             int result;
23
24             //Step 1: Load JDBC driver..
25             Class.forName("com.mysql.cj.jdbc.Driver");
26             System.out.println("Step 1: MySQL driver loaded...!");
27
28             //Step 2: Establish the connection
29             String myURL = "jdbc:mysql://localhost:3306/csm3023_lab6";
30             Connection myConnection = DriverManager.getConnection(myURL, "root", "admin");
31             System.out.println("Step 2: Database is connected...!");
32
33             //Step 3: Create a PreparedStatement object...
34             System.out.println("Step3: Prepared Statements created...!");

```

```

34     System.out.println("Step3: Prepared Statements created...!");
35     String sInsertQry = "INSERT INTO userprofile(username, password, firstname, lastname) VALUES(?, ?, ?, ?)";
36     System.out.println("\tSQL Query: " + sInsertQry);
37
38     //Call method preparedStatement
39     PreparedStatement myPS = myConnection.prepareStatement(sInsertQry);
40
41     //Assign each value to respective columns for Student's table.. (C-Create)
42     myPS.setString(1, myUser.getUsername());
43     myPS.setString(2, myUser.getPassword());
44     myPS.setString(3, myUser.getFirstname());
45     myPS.setString(4, myUser.getLastname());
46
47     result = myPS.executeUpdate();
48     if (result > 0){
49         out.print("\tRecord successfully added into User table...!");
50         out.print("<p>" + "Record with Username " +myUser.getUsername()
51             + " successfully created.!"+ "</p>");
52         out.print("<p>" + "Details of record are: " + "</p>");
53         out.print("<p>Username : " + myUser.getUsername() + "</p>");
54         out.print("<p>Firstname : " + myUser.getFirstname() + "</p>");
55         out.print("<p>Lastname : " + myUser.getLastname() + "</p>");
56     }
57     //Step5: close database connection..
58     System.out.println("Step 5: Close database connection..!");
59     myConnection.close();
60     System.out.println(" ");
61     System.out.println("Database connection is closed..!");
62
63     %>
64   </body>
65 </html>

```

User.java

```

/*
package com.mycompany.lab6;
/**
 *
 * @author User
 */
public class User {
    private String username;
    private String password;
    private String firstname;
    private String lastname;

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getFirstname() {
        return firstname;
    }
}

```

```

34     }
35
36     public void setFirstname(String firstname) {
37         this.firstname = firstname;
38     }
39
40     public String getLastname() {
41         return lastname;
42     }
43
44     public void setLastname(String lastname) {
45         this.lastname = lastname;
46     }
47 }

```

Login.jsp

```

<%-->
Document : login
Created on : 17 May 2024, 3:28:54 pm
Author    : rynaa
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Department of Quality UMT</h1>
        <fieldset>
            <legend>User Login</legend>
            <form action="doLogin.jsp">
                <table>
                    <tr>
                        <td>
                            <label for="username">Username</label>
                        </td>
                        <td>
                            <input type="text" id="username" name="username" placeholder="Enter a username">
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <label for="password">Password</label>
                        </td>
                        <td>
                            <input type="password" id="password" name="password" placeholder="Enter a password">
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <button type="submit" value="Submit">Submit</button>
                            <button type="reset" value="Reset">Cancel</button>
                        </td>
                    </tr>
                </table>
            </form>
        </fieldset>
        <%
            String errorMessage = request.getParameter("error");
            if (errorMessage != null && !errorMessage.isEmpty()) {
                out.println("<p style='color: red;'>" + errorMessage + "</p>");
            }
        %>
        <footer>
            <p>&copy;Quality UMT - 2024</p>
        </footer>
    </body>
</html>

```

doLogin.jsp

```
<%--  
Document : doLogin  
Created on : 17 May 2024, 4:19:30 pm  
Author : rynaa  
--%>  
  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<%@ page import="java.sql.* , lab6.com.User" %>  
<%@ page import="java.io.* , java.util.*" %>  
<!DOCTYPE html>  
<html>  
    <head>  
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
        <title>Department of Quality UMT</title>  
    </head>  
    <body>  
        <h1>Department of Quality UMT</h1>  
        <%  
            // Retrieve username and password from the request  
            String username = request.getParameter("username");  
            String password = request.getParameter("password");  
  
            // Check if username and password are not null and not empty  
            if (username != null && !username.isEmpty() && password != null && !password.isEmpty()) {  
                // Establish database connection  
                Connection conn = null;  
                PreparedStatement pstmt = null;  
                ResultSet rs = null;  
                try {  
                    Class.forName("com.mysql.cj.jdbc.Driver");  
                    String myURL = "jdbc:mysql://localhost:3306/csm3023_lab6";  
                    conn = DriverManager.getConnection(myURL, "root", "admin");  
  
                    // Query to check if the username and password are valid  
                    String query = "SELECT * FROM userprofile WHERE username = ? AND password = ?";  
                    pstmt = conn.prepareStatement(query);  
                    pstmt.setString(1, username);  
                    pstmt.setString(2, password);  
                    rs = pstmt.executeQuery();  
  
                    if (rs.next()) {  
                        // If user exists and credentials are valid, redirect to main.jsp  
                        User user = new User();  
                        user.setUsername(rs.getString("username"));  
                        user.setFirstname(rs.getString("firstname"));  
                        user.setLastname(rs.getString("lastname"));  
  
                        // Set user object in session for later use  
                        session.setAttribute("user", user);  
                        response.sendRedirect("main.jsp");  
                    } else {  
                        // If invalid username or password, redirect back to login.jsp with error message  
                        response.sendRedirect("login.jsp?error=Invalid+username+or+password");  
                    }  
                } catch (Exception e) {  
                    e.printStackTrace();  
                } finally {  
                    // Close resources  
                    if (rs != null) rs.close();  
                    if (pstmt != null) pstmt.close();  
                    if (conn != null) conn.close();  
                }  
            } else {  
                // If username or password is empty, redirect back to login.jsp with error message  
                response.sendRedirect("login.jsp?error=Username+and+password+are+required");  
            }  
        %>
```

Main.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page language="java"%>
<%@ page import="lab6.com.User" %>
<%@ page import="java.io.*, java.util.*" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <%
            // Retrieve user object from session
            lab6.com.User user = (lab6.com.User) session.getAttribute("user");

            // Check if user object exists in session
            if (user != null) {
        %>
        <h1>Welcome, <%= user.getFirstname() %> <%= user.getLastname() %></h1>
        <p>Your username is: <%= user.getUsername() %></p>

        <!-- Add other content here -->

        <p><a href="logout.jsp">Logout</a></p>
        <%
            } else {
                // If user object does not exist in session, redirect to login page
                response.sendRedirect("login.jsp");
            }
        %>
    </body>
```

Output

Department of Quality UMT

User Registration

Username	<input type="text" value="Enter a username"/>
Password	<input type="password" value="Enter a password"/>
Firstname	<input type="text" value="E.g.:Ecah"/>
Lastname	<input type="text" value="E.g.:bin Daud"/>
<input type="button" value="Submit"/> <input type="button" value="Cancel"/>	

©Quality UMT - 2024