



UNIVERSITI MALAYSIA TERENGGANU

CSM3023 WEB BASED APPLICATION DEVELOPMENT (K1)

**BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH
HONORS**

LAB 6

SEMESTER I 2022/2023

Prepared for:

DR. MOHAMAD NOR HASSAN

Prepared by:

LUQMAN HAKIM BIN AZIZ

(S66292)

JSP: Saving and

Week 6

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

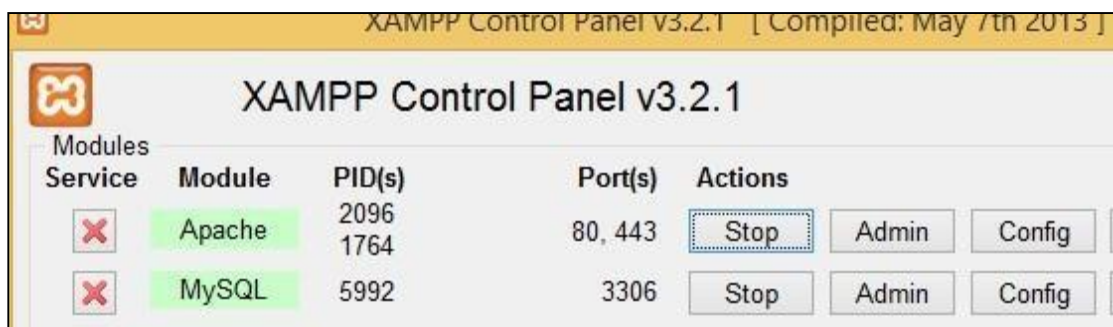
Description:

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.

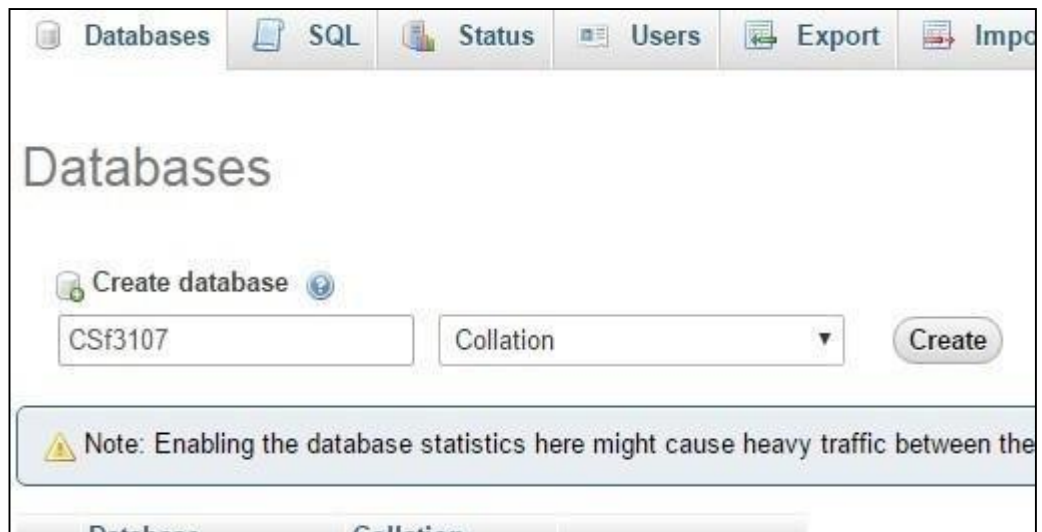
Estimated time: 20 minutes

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

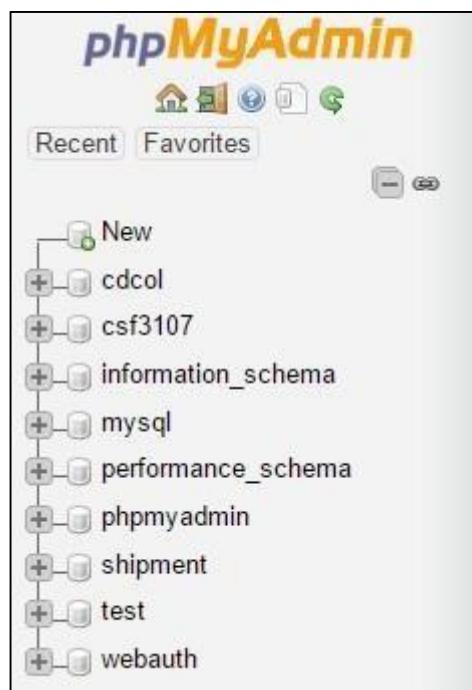
1. Start XAMPP control panel.
2. Start the Apache web server.
3. Start the MySQL database.



4. Click the *Admin* button for MySQL.
5. Go to *Database's* tab.
6. Key-in as *CSF3107* and click *Create* button.



7. Database schema successfully created.



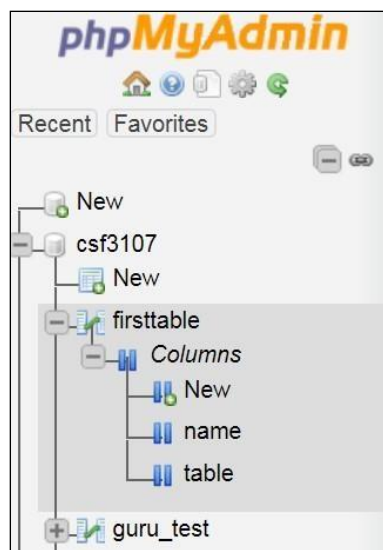
8. Use any tool to manipulate the SQL statement. Create table **FirstTable** in *csf3107* database schema.

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



10. Execute the SQL statement.

11. Table successfully created.



Step 2 - Create SampleInsertionRecord.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*.

New Web Application

Steps

1. Choose Project
2. **Name and Location**
3. Server and Settings
4. Frameworks

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

< Back **Next >** Finish Cancel Help

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

New Web Application

Steps

1. Choose Project
2. **Name and Location**
3. Server and Settings
4. Frameworks

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

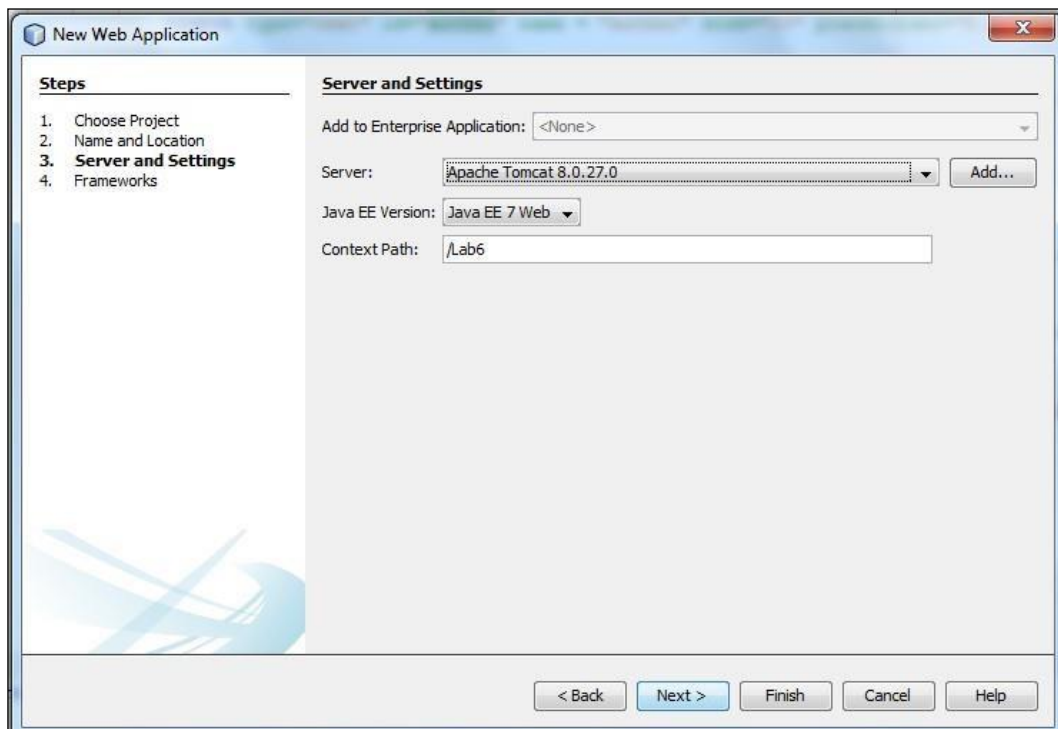
Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

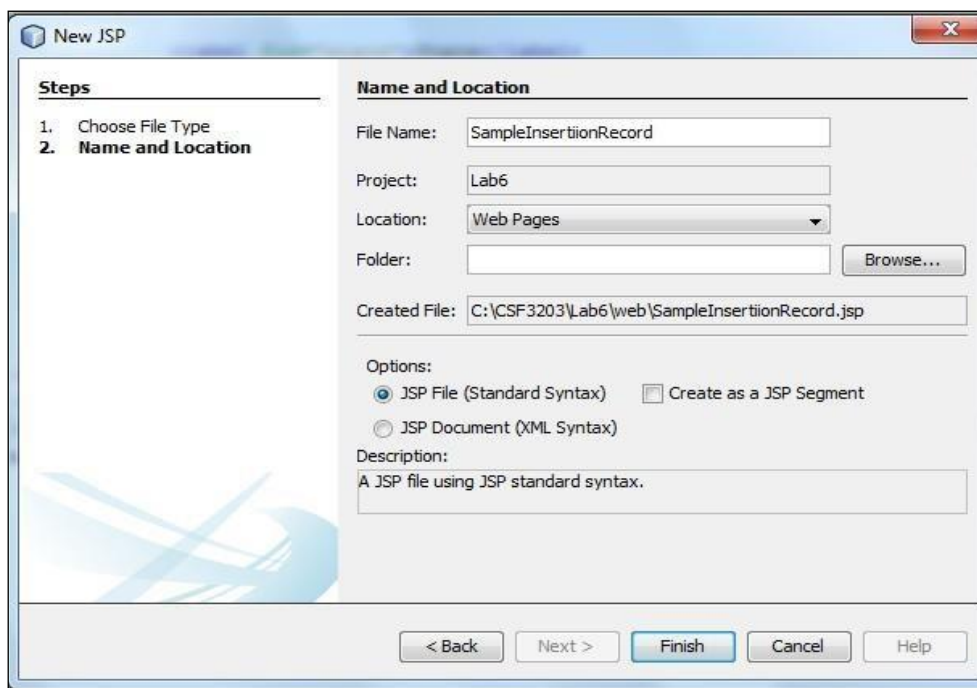
< Back **Next >** Finish Cancel Help

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/cs3107";
    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>

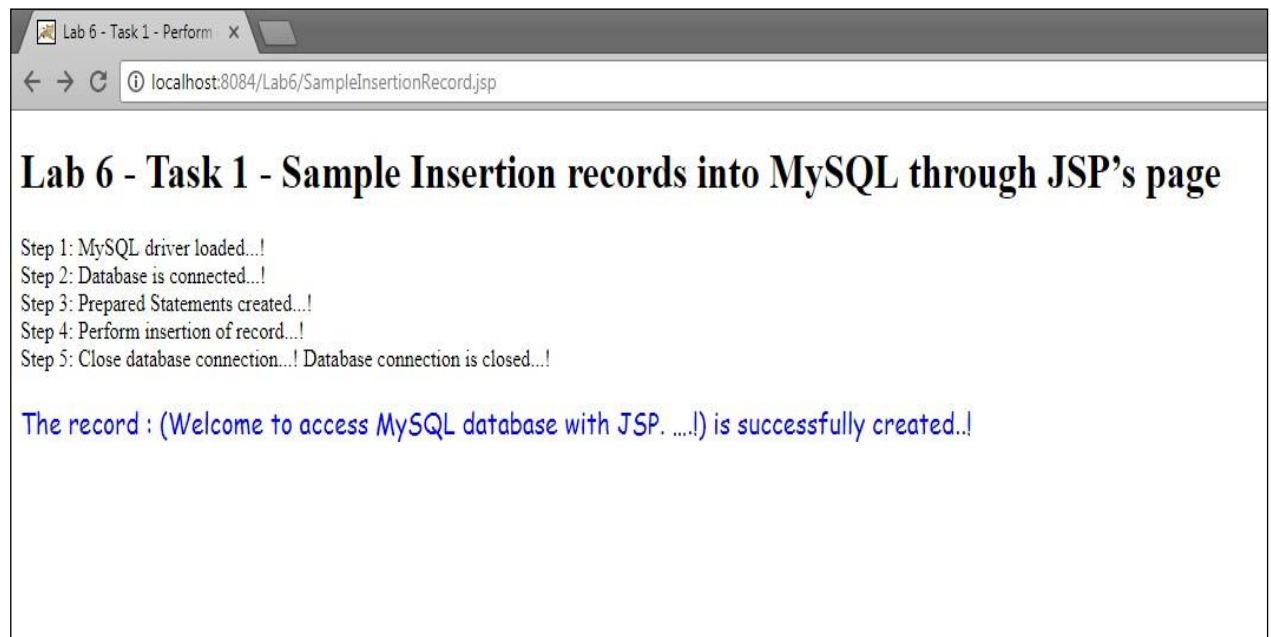
<%
    out.println("Step 5: Close database connection...!");

    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();
%>

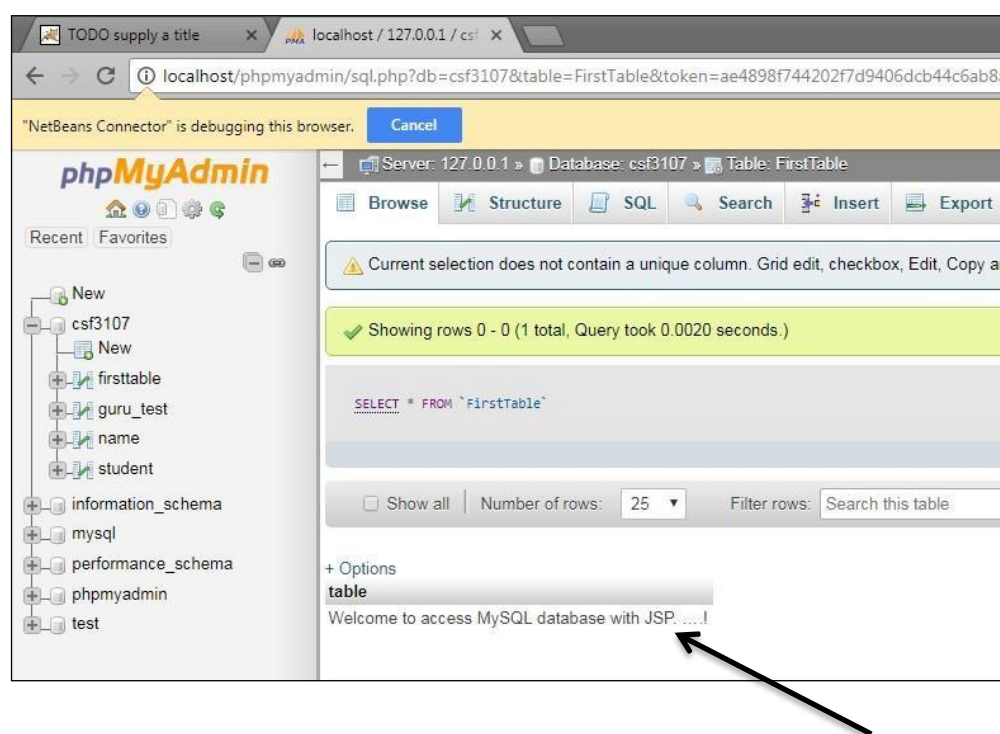
```

17. Save and compile **SampleInsertionRecord.jsp** file.
18. Run the **SampleInsertionRecord.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!!)



Coding SampleInsertionRecord.jsp

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page language="java"%>
<%@page import="java.sql.*"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Lab 6 - Task 1 - Sample Insertion records into MySQL through JSP's page</h1>

    <%
      int result;

      Class.forName("com.mysql.cj.jdbc.Driver");
      out.println("Step 1: MySQL driver loaded...!");
    %>

    <br>

    <%
      String myURL = "jdbc:mysql://localhost/cs3107";
      Connection myConnection = DriverManager.getConnection(myURL, "root", "admin");
      out.println("Step 2: Database is connected...!");
    %>

    <br>
  </body>
</html>

```



```

<%
    out.println("Step 3: Prepared Statements created...!");

    String sInsertQry = "INSERT INTO FirstTable VALUE(?)";

    PreparedStatement myPS = myConnection.prepareStatement(sInsertQry);
%>

<br>

<%
    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 succesfully created...!" + "</p>");
    }

    myConnection.close();
%>

</body>
</html>

```

Database

1 • CREATE DATABASE csf3107;

2

3 • CREATE TABLE FirstTable(NAME VARCHAR(45))

Output

localhost:8080/Lab6/SampleInsertionRecord.jsp

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...!
Step 5: Close database connection...! Database connection is closed...!

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

	name
▶	Welcome to access MySQL database with JSP...!
	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

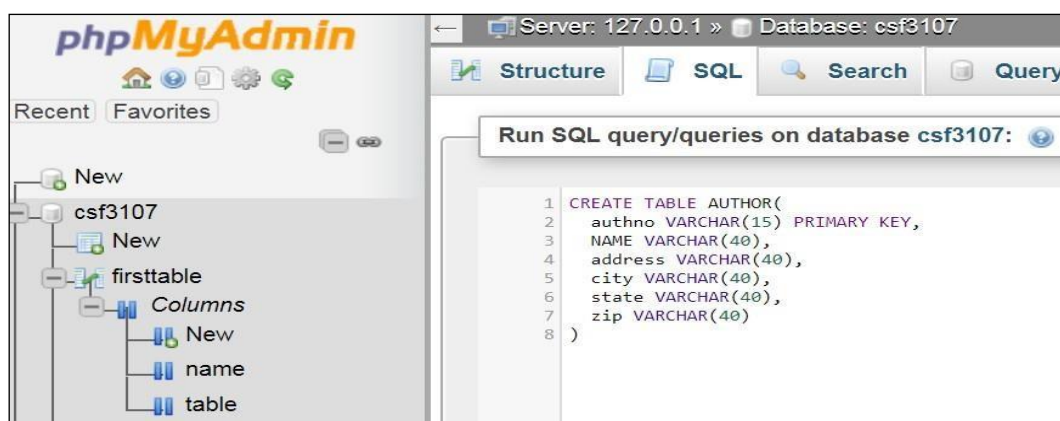
2. Create **insertAuthor.jsp** as a main interface to register a new author.

3. Create **processAuthor.jsp** page to process and acknowledge the user upon inserting record in the database.

Estimated time: 40 minutes

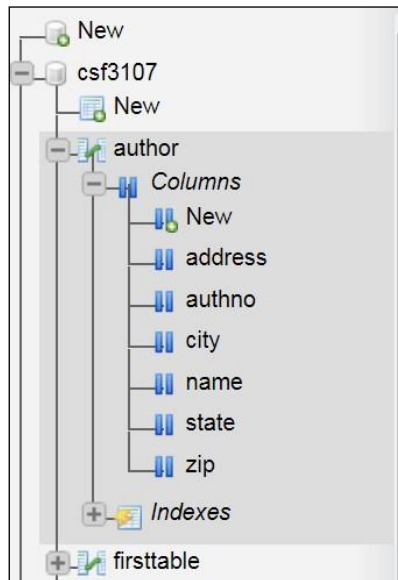
1. Use any tool to manipulate the SQL statement. Create table **author** in **csf3107** database schema.

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

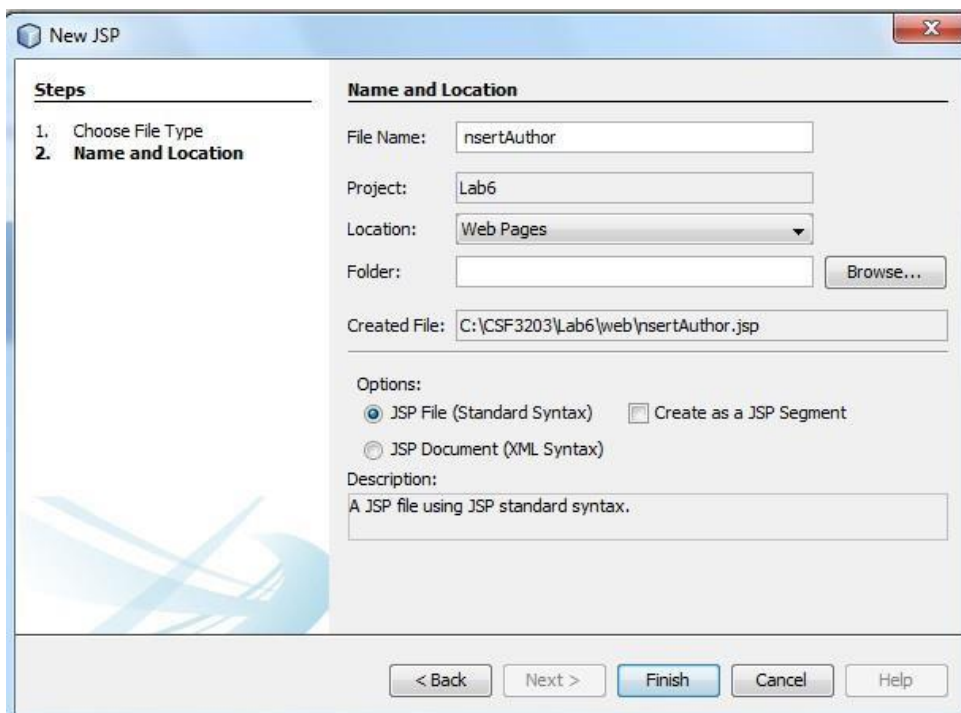


3. Execute the SQL statement.

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

Name

Address

City

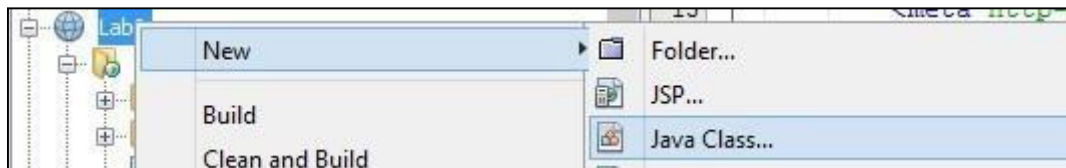
State

Zip

©2018-Dr.Faizah Binti Aplop

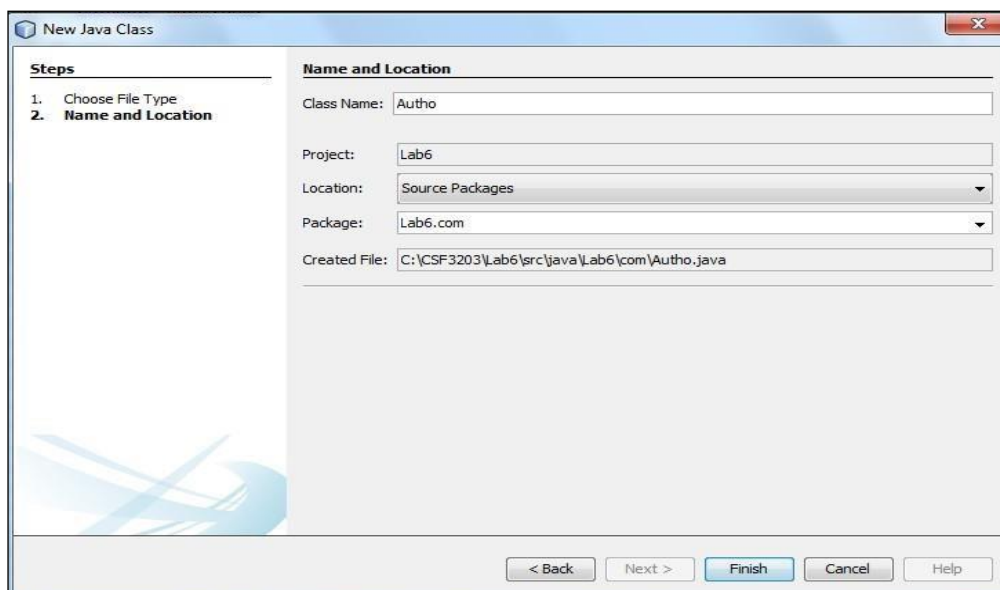
8. Go to *Lab6* project folder.

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;
}

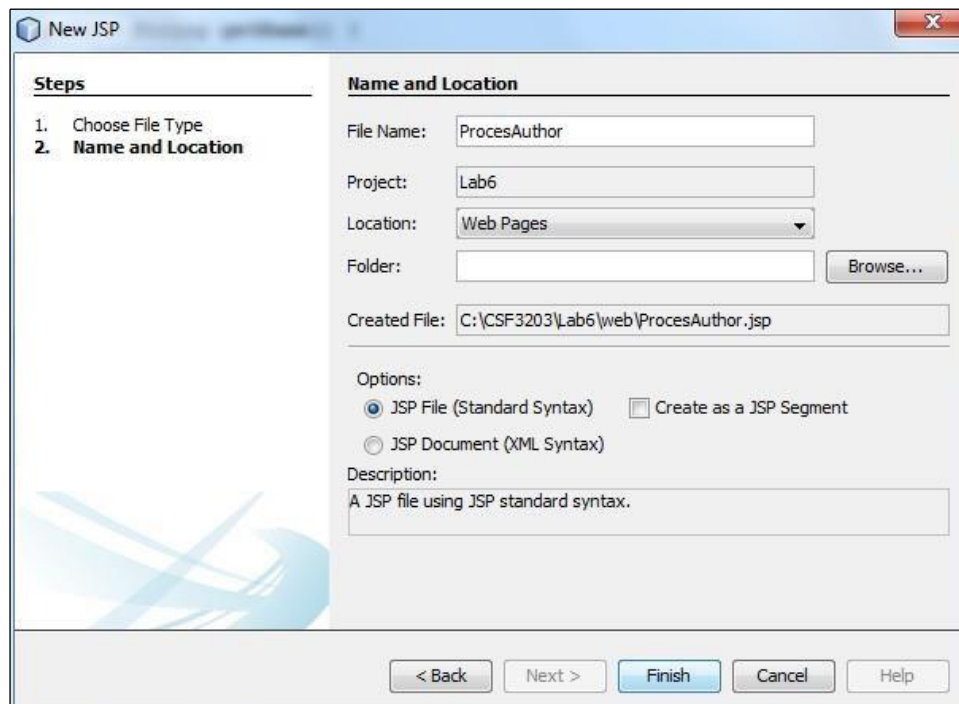
public void setState(String state) {
    this.state = state;
}

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/cs3107";
    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("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 *prosessAuthor.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

Coding insertAuthor.jsp


```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Insert Author</title>
</head>
<body>
    <h2>Lab 6 - Task 2 - Perform creating and retrieving records via JSP page</h2>
    <fieldset>
        <legend>Author Registration</legend>
        <form action="processAuthor.jsp" method="post">
            <table>
                <tr>
                    <td>Author No:</td>
                    <td><input type="text" name="authno" required></td>
                </tr>
                <tr>
                    <td>Name:</td>
                    <td><input type="text" name="name" required></td>
                </tr>
                <tr>
                    <td>Address:</td>
                    <td><input type="text" name="address" required></td>
                </tr>
                <tr>
                    <td>City:</td>
                    <td><input type="text" name="city" required></td>
                </tr>
                <tr>
                    <td>State:</td>
                    <td><input type="text" name="state" required></td>
                </tr>
                <tr>
                    <td>Zip:</td>
                    <td><input type="text" name="zip" required></td>
                </tr>
            </table>
            <input type="submit" value="Submit">
            <input type="reset" value="Cancel">
        </form>
    </fieldset>
</body>
<footer>
    &copy; Luqman Hakim
</footer>
</html>

```

Coding Author.java

```
package Lab6.com;

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;
    }
}
```

```
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;  
}  
}
```

Coding processAuthor.jsp

```
<%@ page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page language="java"%>
<%@ page import="java.sql.*" %>
<jsp:useBean id="myAuthor" class="Lab6.com.Author" scope="request"/>
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Process Author</title>
</head>
<body>
  <h1>Lab 6 - Task 2 - 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/cs3107";
    Connection myConnection = DriverManager.getConnection(myURL, "root", "admin");

    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());

    result = myPS.executeUpdate();
    if (result > 0){
      out.println("\tRecord succesfully 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>");

      //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...!");
    }
  %>
</body>
</html>
```

Database

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

Output

localhost:8080/Lab6/insertAuthor.jsp

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

Author Registration

Author No:

Name:

Address:

City:

State:

Zip:

© Luqman Hakim

localhost:8080/Lab6/processAuthor.jsp

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

Record successfully added into Author table...!

Record with author no S662922212 successfully created..!

Details of record are;

Name: Luqman Hakim Bin Aziz

Address: gong badak

City: Kuala Terengganuu

State: Terengganuu

Zip: 12weww

	authno	NAME	address	city	state	zip
▶	S66292	Luqman Hakim Bin Aziz	gong badak	Kuala Terengganu	Terengganu	12we
	S6629222	Luqman Hakim Bin Aziz	gong badak	Kuala Terengganuu	Terengganuu	12wewww
	S662922212	Luqman Hakim Bin Aziz	gong badak	Kuala Terengganuu	Terengganuu	12wewww
•	NULL	NULL	NULL	NULL	NULL	NULL

Reflection 1. What have you learnt from this exercise?

From this exercise, I have learned how to use JSP to handle form submissions, interact with a database using JavaBeans and JDBC, and display the results back to the user.

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

The steps to perform a transaction in a database are: first, create an HTML form to capture user input; second, use a JavaBean to store and transfer the data; third, establish a database connection using JDBC, prepare and execute the SQL statement to insert the data; and finally, provide feedback to the user and close the database connection properly to ensure resource management.

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.

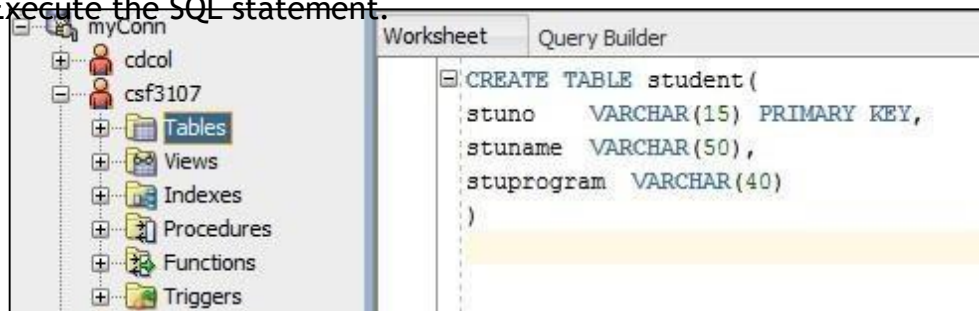
Problem Description:

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.

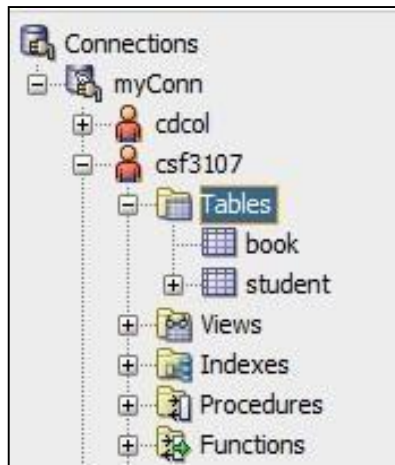
Estimated time: 50 minutes

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.

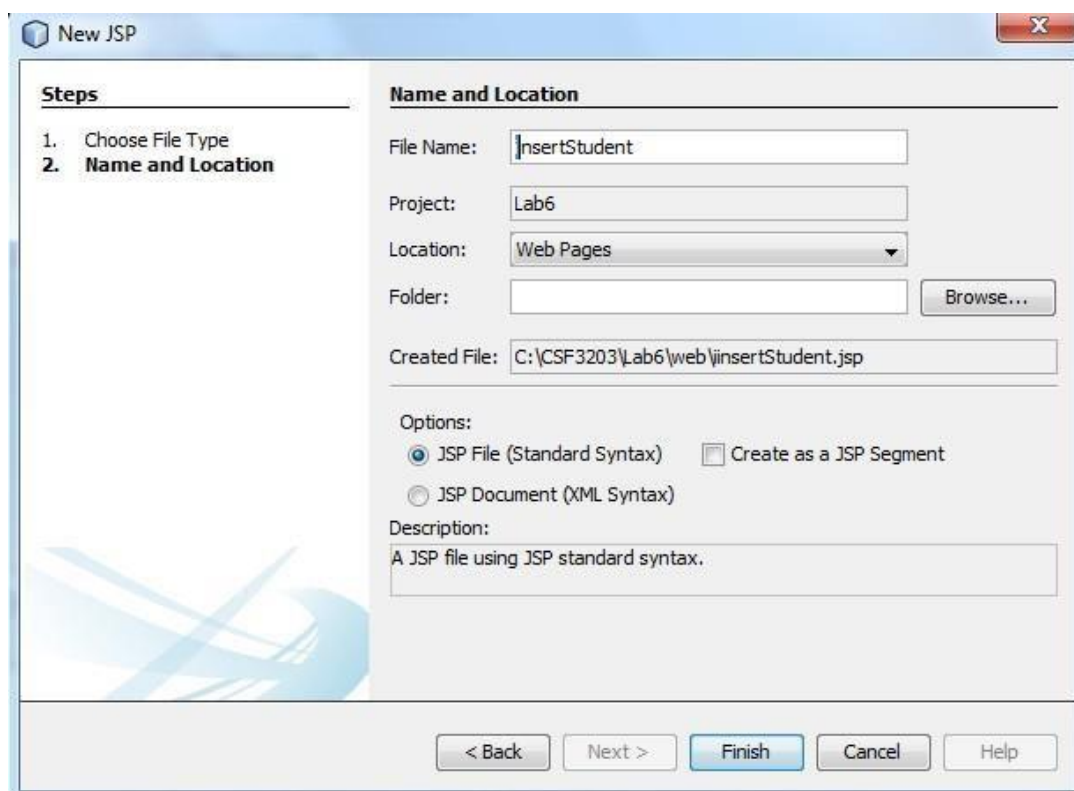


3. Table successfully created.



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

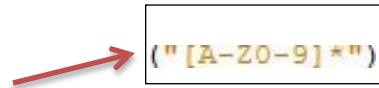
1. Create a new JSP's page and rename as *insertStudent*.



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.

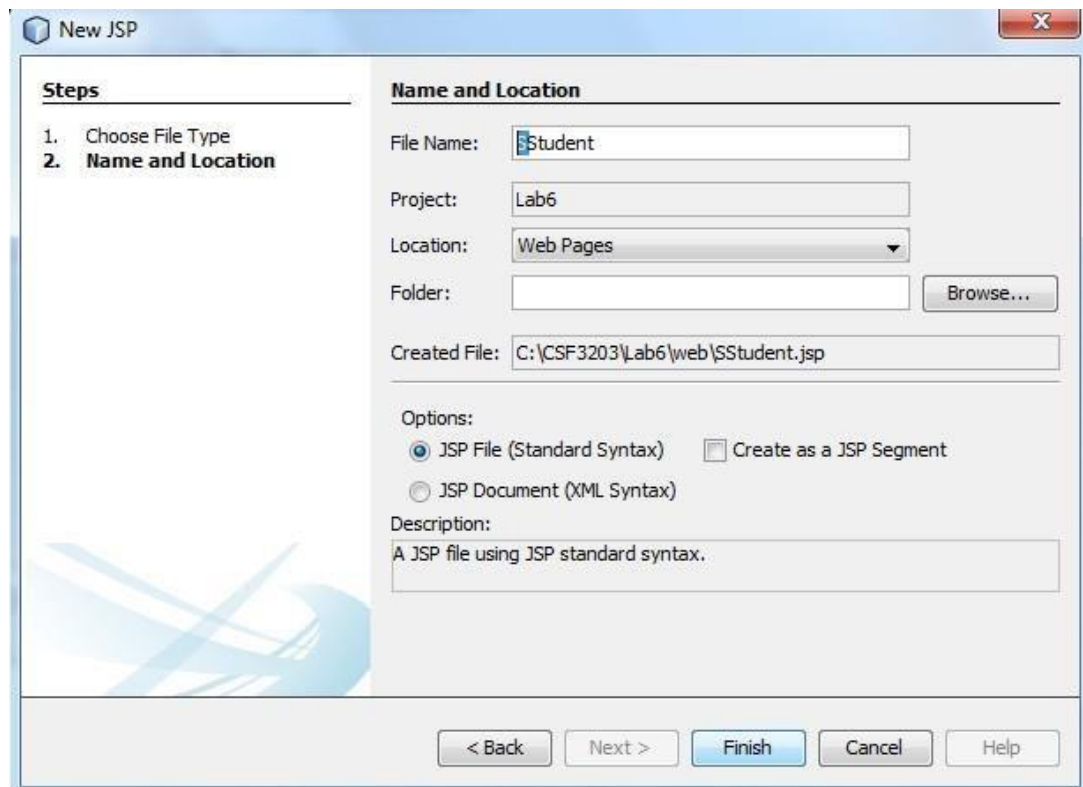
3. Produce the following output;

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.

```

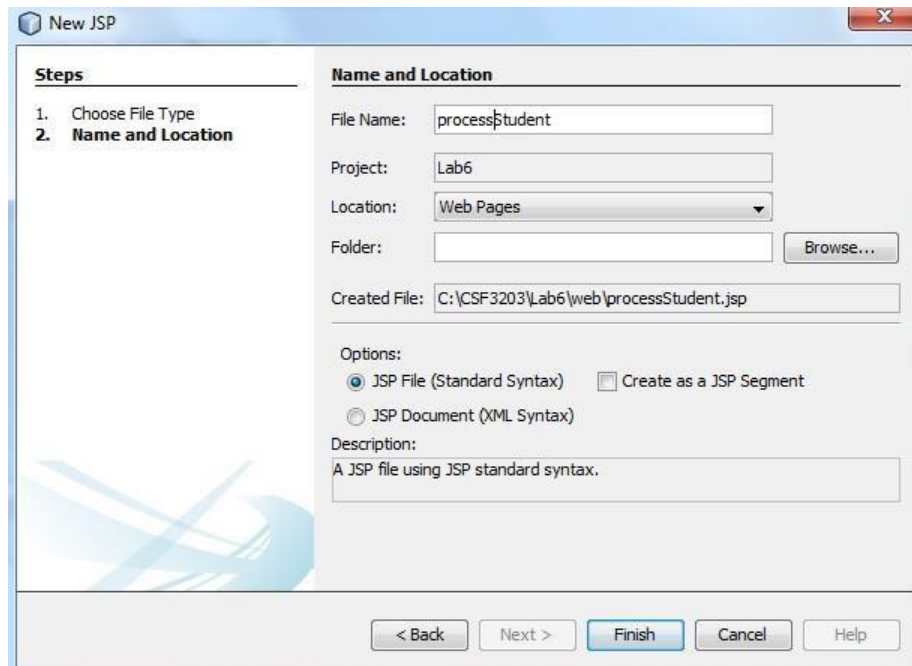
public String getStuno() {
    Pattern pt = Pattern.compile("[A-Z0-9]*");
    Matcher mt = pt.matcher(stuno);
    boolean bl = mt.matches();
    if (bl == true) {
        valid = stuno;
    } else {
        valid = invalid;
    }
    return valid;
}

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/cs3107";
    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.

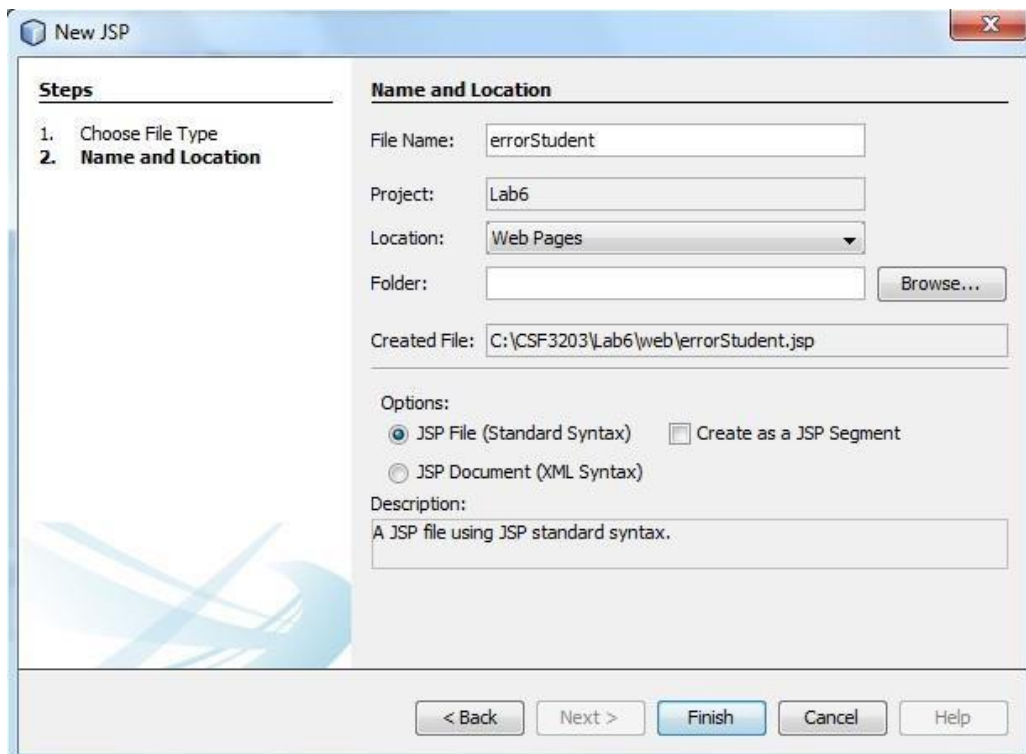
```

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

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

Coding insertStudent.jsp


```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Hero Academy</title>
  </head>
  <body>
    <fieldset>
      <legend>Student Registration</legend>
      <form action="processStudent.jsp" method="post">
        <table>
          <tr>
            <td>
              <label for="stuno">Student No</label>
            </td>
            <td>
              <input type="text" id="stuno" name="stuno" placeholder="E.g.: UKXXXXXX">
            </td>
          </tr>
          <tr>
            <td>
              <label for="name">Name</label>
            </td>
            <td>
              <input type="text" id="name" name="name" placeholder="Enter your name">
            </td>
          </tr>
          <tr>
            <td><label for="program">Program</label></td>
            <td>
              <select id="program" name="program">
                <option value="BSc. Soft. Eng.">BSc. Soft. Eng.</option>
                <option value="BSc. with IM">BSc. with IM</option>
                <option value="BSc. in Networking">BSc. in Networking</option>
                <option value="BSc. in Robotics">BSc. in Robotics</option>
              </select>
            </td>
          </tr>
          <tr>
            <td>
              <button type="submit" value="Submit">Submit</button>
              <button type="reset" value="Reset">Cancel</button>
            </td>
          </tr>
        </table>
      </form>
    </fieldset>
    <footer>
      <p>&copy; Lucman Hakim</p>
    </footer>
  </body>
</html>

```

Coding Student.java


```

package Lab6.com;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Student {
    //Create attributes...
    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;
    }
}

```

```

    public String getName() {
        return name;
    }

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

    public String getProgram() {
        return program;
    }

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

```

Coding processStudent.jsp

```

<%@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="Lab6.com.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/csf3107";
            Connection myConnection = DriverManager.getConnection(myURL, "root", "admin");
            System.out.println("Step 2: Database is connected...!");
        %>
    </body>
</html>

```

```

//Step 3: Create a PreparedStatement object...
System.out.println("Step3: Prepared Statements created...!");
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 Student's table.. (C-Create)
myPS.setString(1, myStudent.getStuno());
myPS.setString(2, myStudent.getName());
myPS.setString(3, myStudent.getProgram());

result = myPS.executeUpdate();
if (result > 0){
    out.println("\tRecord successfully added into Student 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>");
}

//Step5: close database connection..!
System.out.println("Step 5: Close database connection..!");
myConnection.close();
System.out.println(" ");
System.out.println("Database connection is closed..!");
%>
</body>
</html>

```

Coding errorStudent.jsp

```

<%@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>
        <form id="errorFrm" action="insertStudent.jsp" method="post">
            <h1>Lab 6 - Task 1 - Perform creating and retrieving records via JSP page</h1>
            <p>Error occur when inserting record...!</p>
            <p>Error Message: ${exception.getMessage()}</p>
            <br>
        </form>
    </body>
</html>

```

Database

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

Output

Student Registration

Student No: S6629222

Name: Khalid

Program: BSc. in Networking

Submit Cancel

©Luqman Hakim

Record successfully added into Student table...!

Record with student no S6629222 successfully created..!

Details of record are;

Student ID : S6629222

Name : Khalid

Program : BSc. in Networking

	stuno	stuname	stuprogram
▶	S66292	Luqman	BSc. with IM
	S6629222	Khalid	BSc. in Networking
✱	NULL	NULL	NULL

Reflection

1. What have you learnt from this exercise?

From this exercise, I learned how to use JSP and JDBC to interact with a database, including handling form data and executing SQL statements.

2. Define step by step before you successfully perform the transaction in a database.
 - First, load the JDBC driver.
 - Second, establish a connection to the database.
 - Third, create a PreparedStatement to execute the SQL query with the provided parameters.
 - Finally, execute the query and handle the results, then close the database connection.

Task 4: Perform Retrieving Records Via JSP Page

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

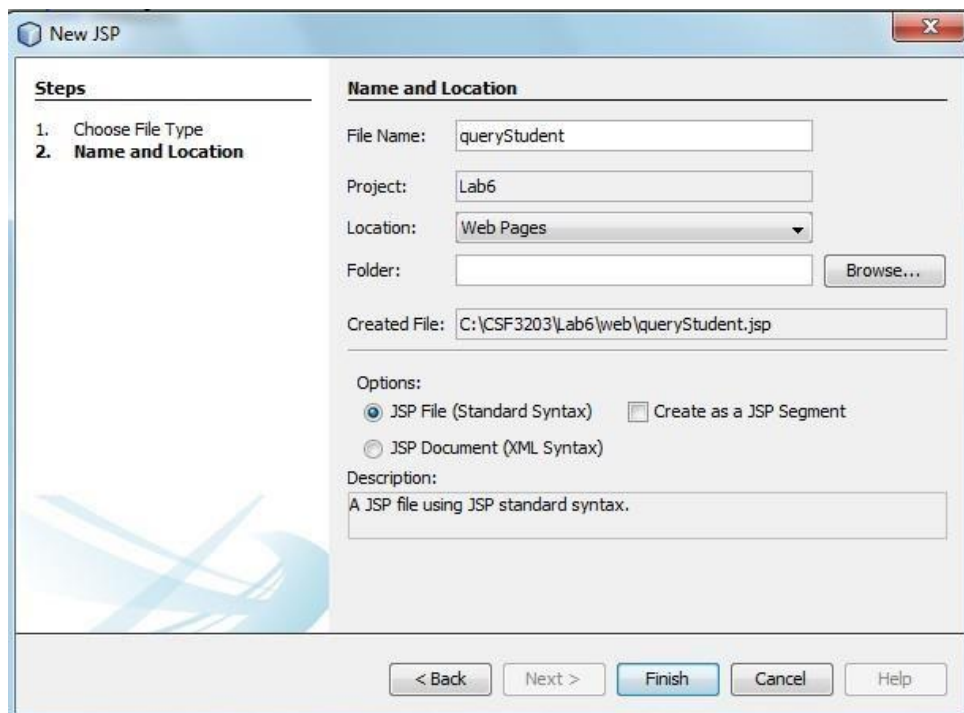
Problem Description: Retrieve student records and populate in the table.

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/cs3107";
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

queryStudent.jsp


```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.sql.*"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Hero Academy Student 2024/2025</title>
        <style>
            table{
                border-collapse: collapse;
            }
            td, th{
                border: 1px solid #999;
                padding: 0.5rem;
                text-align: left;
            }
            th{
                background: gold;
            }
        </style>
    </head>
    <body>
        <h1>Task 4: Retrieving record via JSP page</h1>

```

```

<%
    out.print("<table>");
    out.print("<thead>");
    out.print("<tr>");
        out.print("<th>" + "ISBNNo" + "</th>");
        out.print("<th>" + "Author" + "</th>");
        out.print("<th>" + "Title" + "</th>");
    out.print("</tr>");
    out.print("</thead>");
    out.print("<tbody>");

%>
<%
    //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/csf3107";
    Connection myConnection = DriverManager.getConnection(myURL, "root", "admin");
    System.out.println("Step 2: Database is connected...!");

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

    String myQuery = "SELECT * FROM student";
    ResultSet myResultSet = myStatement.executeQuery(myQuery);

    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>");
    }

```

```

    }
    //Step5: 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>");

    %>
</body>
</html>

```

Output

← ↻ ⓘ localhost:8080/Lab6/queryStudent.jsp

Task 4: Retrieving record via JSP page

ISBNNo	Author	Title
S66292	Luqman	BSc. with IM
S6629222	Khalid	BSc. in Networking
UK12489	Ahmad Salam	BSc. with IM
UK56789	Rosnah Azman	BSc. Soft. Eng.
UK67342	Liew Cheng Huat	BSc. in Robotics

	stuno	stuname	stuprogram
▶	S66292	Luqman	BSc. with IM
	S6629222	Khalid	BSc. in Networking
	UK12489	Ahmad Salam	BSc. with IM
	UK56789	Rosnah Azman	BSc. Soft. Eng.
	UK67342	Liew Cheng Huat	BSc. in Robotics
●	NULL	NULL	NULL

Reflection 1. What have you learnt from this exercise?

Through this exercise, I've learned how to integrate Java Servlets and JSP to interact with a MySQL database using JDBC. This process involved setting up database connections, executing queries, handling exceptions, and presenting retrieved data dynamically in HTML tables.

2. Explain the differences when using *Statement()* and *PreparedStatement()*.

- **Statement:** Used for executing static SQL queries at runtime. It can be less secure as it directly concatenates input with SQL statements, making it vulnerable to SQL injection attacks.
- **PreparedStatement:** Preferred for executing dynamic SQL queries with parameters. It provides better performance and security by precompiling SQL queries, reducing database load, and protecting against SQL injection through parameterized queries. It also improves code readability and maintainability.

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).

Marathon Registration

IC No	<input type="text" value="E.g.: 921110-10-2514"/>
Name	<input type="text" value="Enter your name"/>
Category	<div>5 KM ▼<div>5 KM7 KM10 KM</div></div>
<input type="button" value="Submit"/> <input type="button" value="Cancel"/>	

©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*.

```

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">

```



```

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*.
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();

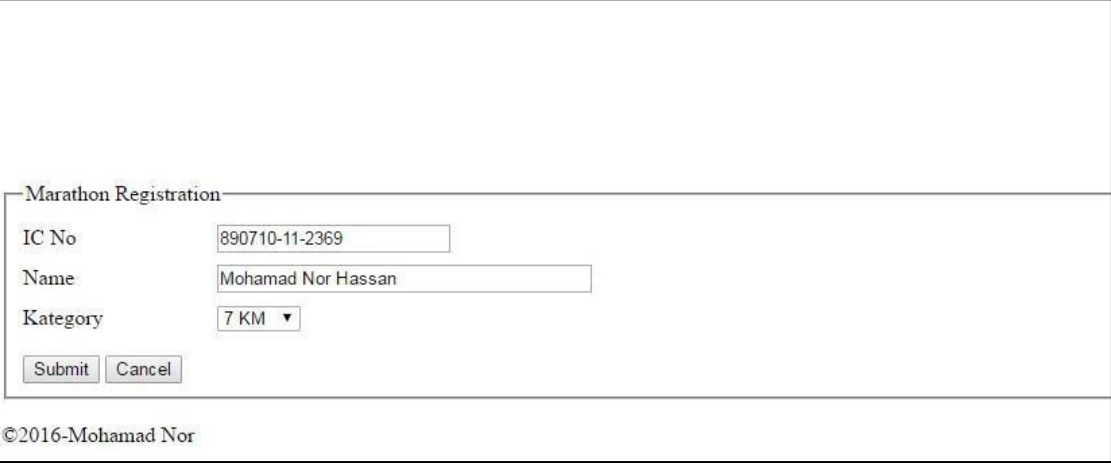
    //Step 2: Add the records...
    result = object1.addDetails(myMarathon);

    //Step 3: Determine whether the transactino 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 connnection...!
    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.



The screenshot shows a web form titled "Marathon Registration". It contains three input fields: "IC No" with the value "890710-11-2369", "Name" with the value "Mohamad Nor Hassan", and "Category" with a dropdown menu showing "7 KM". Below the fields are two buttons: "Submit" and "Cancel". At the bottom left of the form, there is a copyright notice: "©2016-Mohamad Nor".

14. The output will appear in a web browser.



The screenshot shows the output of the registration process in a web browser. It displays a red message: "Record with IC No 890710-11-2369 successfully created..!". Below this, it says "Details of record are;" followed by the record details: "Ic No : 890710-11-2369", "Name : Mohamad Nor Hassan", and "Category : 7 KM". At the bottom left, there is a copyright notice: "©2016-Mohamad Nor".

Reflection

1. What have you learnt from this exercise?

This exercise taught me how to integrate JavaServer Pages (JSP) with Java classes to handle database operations effectively, using JDBC for database connectivity and prepared statements to execute SQL queries securely.

2. Describe the benefits of using JavaBeans.

JavaBeans provide several benefits, such as encapsulation of data fields and business logic into reusable components, promoting code reusability and maintainability. They enable easy integration with JSPs through the `<jsp:useBean>` and `<jsp:setProperty>` tags, facilitating clean separation of presentation logic from data handling in web applications. Additionally,

JavaBeans support features like serialization, making them suitable for distributed computing and storage in various environments.

Coding registerMarathon.jsp

```
<%--
    Document   : registerMarathon
    Created on : 15 Jun 2024, 2:10:49 am
    Author    : Luqman Hakim
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Superhero Marathon 2025</title>
    </head>
    <body>
        <h1>Superhero Marathon 2025</h1>
        <fieldset>
            <legend>Marathon Registration</legend>
            <form action="processMarathon.jsp" method="post">
                <table>
                    <tr>
                        <td>
                            <label for="icno">IC No</label>
                        </td>
                        <td>
                            <input type="text" id="icno" name="icno" placeholder="E.g.: 921110-10-2514">
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <label for="name">Name</label>
                        </td>
                        <td>
                            <input type="text" id="name" name="name" placeholder="Enter your
name">
                        </td>
                    </tr>
                </table>
            </form>
        </fieldset>
    </body>
</html>
```

```

<tr>
  <td>
    <label for="category">Category</label>
  </td>
  <td>
    <select id=category name="category">
      <option value="5 KM">5 KM</option>
      <option value="7 KM">7 KM</option>
      <option value="10 KM">10 KM</option>
    </select>
  </td>
</tr>
<tr>
  <td>
    <button type="submit" value="Submit">Submit</button>
    <button type="reset" value="Reset">Cancel</button>
  </td>
</tr>
</table>
</form>
</fieldset>
<footer>
  <p>&copy;Luqman Hakim</p>
</footer>
</body>
</html>

```

Coding Marathon.java

```

package Lab6.com;

public class Marathon {
  private String icno;  private
  String name;
  private String category;

  public String getIcno() {
return icno;
  }

  public void setIcno(String icno) {
    this.icno = icno;
  }
}

```

```

    }

    public String getName() {
return name;
    }

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

    public String getCategory() {
return category;
    }

    public void setCategory(String category) {
        this.category = category;
    }
}

```

Coding Database.java

```

package
Lab6.com;

import java.sql.Connection; import
java.sql.DriverManager; import
java.sql.SQLException; import
java.util.logging.Level; import
java.util.logging.Logger;
import Lab6.com.Marathon;

public class Database {
    private static Connection myConnection = null;
    private static String myURL = myURL = "jdbc:mysql://localhost:3306/csf3107";
private int result = 0;

    public static Connection getConnection() throws ClassNotFoundException{
if(myConnection != null){
        return myConnection;
    }
else try{
        Class.forName("com.mysql.cj.jdbc.Driver");
        myConnection = DriverManager.getConnection(myURL, "root", "admin");

```

```

    }
    catch(SQLException e){
e.printStackTrace();
    }
    return myConnection;
}

public void closeConnection() throws ClassNotFoundException
{
    try{
        myConnection.close();
    }
    catch(SQLException e){
e.printStackTrace();
    }
}
}

```

Coding MarathonDAO.java

```

package
Lab6.com;

import java.sql.Connection; import
java.sql.PreparedStatement; import
java.sql.SQLException;
import Lab6.com.Database;

public class MarathonDAO {
private Connection connection;
private int result = 0;
public MarathonDAO() throws ClassNotFoundException{
    connection = Database.getConnection();
}
public int addDetails(Marathon marathon){
try{
    String mySQL = "INSERT INTO marathon(icno, name, category) VALUES (?, ?, ?)";
    PreparedStatement preparedStatement = connection.prepareStatement(mySQL);

    System.out.println("IC no  = " + marathon.getIcno());
    System.out.println("Name  = " + marathon.getName());
    System.out.println("Category= " + marathon.getCategory());

```

```

        //Parameters
        preparedStatement.setString(1, marathon.getIdno());
preparedStatement.setString(2, marathon.getName());
preparedStatement.setString(3, marathon.getCategory());
result = preparedStatement.executeUpdate();
}catch(SQLException e){        e.printStackTrace();
    }
    return result;
}
}

```

Coding processMarathon.jsp

```

<%--
    Document   : processMarathon
    Created on : 15 Jun 2024, 2:13:27 am
    Author    : Luqman Hakim
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page language="java"%>
<%@page import="java.sql.*"%>
<%@page import="Lab6.com.Database"%>
<%@page import="Lab6.com.Marathon"%>
<%@page import="Lab6.com.MarathonDAO"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Superhero Marathon 2025</title>
    </head>
    <body>
        <jsp:useBean id="myMarathon" class="Lab6.com.Marathon" scope="request"/>
        <jsp:setProperty name="myMarathon" property="*" />

        <%
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 transactions is success..
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>");          }else{
    out.print("<p>Crazyy this is an error! maybe you already registerd before?</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...!");
%>
</body>
</html>

```

Database

```

20 • CREATE TABLE Marathon (
21     icno VARCHAR(15) PRIMARY KEY,
22     name VARCHAR(50),
23     category VARCHAR(50)
24 );

```

Output

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/Lab5/registerMarathon.jsp'. The page title is 'Superhero Marathon 2025'. Below the title is a 'Marathon Registration' form. The form contains three input fields: 'IC No' with the value '010207030629', 'Name' with the value 'Ahmad Mohsin', and 'Category' with a dropdown menu showing '10 KM'. Below these fields are two buttons: 'Submit' and 'Cancel'. At the bottom left of the page, there is a copyright notice: '©Luqman Hakim'.



icno	name	category
010207030629	Ahmad Mohsin	10 KM
NULL	NULL	NULL

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..!'

Coding insertUser.html

```
<!DOCTYPE html>  
<html>
```

```

<head>
  <title>Department of Quality UMT</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</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>
            <label for="firstname">Firstname</label>
          </td>
          <td>
            <input type="text" id="firstname" name="firstname"
placeholder="E.g.:Jamal">
          </td>
        </tr>
        <tr>
          <td>
            <label for="lastname">Lastname</label>
          </td>
          <td>
            <input type="text" id="lastname" name="lastname" placeholder="Enter a
lastname">
          </td>
        </tr>
      </table>
    </form>
  </fieldset>

```

```

        <td>
            <input type="text" id="lastname" name="lastname" placeholder="E.g.:bin
Abdullah">
        </td>
    </tr>
    <tr>
        <td>
            <button type="submit" value="Submit">Submit</button>
            <button type="reset" value="Reset">Cancel</button>
        </td>
    </tr>
</table>
</form>
</fieldset>
<footer>
    <p>&copy;Luqman Hakim</p>
</footer>
</body>
</html>

```

Coding processUser.jsp

```

<%--
    Document   : processUser
    Created on : 15 Jun 2024, 2:36:32 am
    Author    : Luqman Hakim
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page language="java"%>
<%@page import="java.sql.*"%>
<%@page errorPage="login.jsp"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Department of Quality UMT</title>
    </head>
    <body>
        <jsp:useBean id="myUser" class="Lab6.com.User" scope="request"/>
        <jsp:setProperty name="myUser" 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/csf3017";
    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...!");
    String sInsertQry = "INSERT INTO userprofile(username, password, firstname,
lastname) VALUES(?, ?, ?, ?)";
    System.out.println("\tSQL Query: " + sInsertQry);

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

    //Assign each value to respective columns for Student's table.. (C-Create)
    myPS.setString(1, myUser.getUsername());      myPS.setString(2,
myUser.getPassword());      myPS.setString(3, myUser.getFirstname());
    myPS.setString(4, myUser.getLastname());

    result = myPS.executeUpdate();
    if (result > 0){
        out.println("\tRecord successfully added into User table...!");
        out.print("<p>" + "Record with Username " +myUser.getUsername()
            + " successfully created..!" + "</p>");
    out.print("<p>" + "Details of record are; " + "</p>");
    out.print("<p>Username : " + myUser.getUsername() + "</p>");
    out.print("<p>Firstname : " + myUser.getFirstname() + "</p>");
    out.print("<p>Lastname : " + myUser.getLastname() + "</p>");
    }
    //Step5: close database connection..!
    System.out.println("Step 5: Close database connection...!");
    myConnection.close();      System.out.println(" ");
    System.out.println("Database connection is closed..!");
%>
</body>

```

</html>

Coding login.jsp

<%--

Document : login

Created on : 15 Jun 2024, 2:37:00 am

Author : Luqman Hakim

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!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>

<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;Luqman Hakim</p>
</footer>
</body>
</html>

```

Coding doLogin.jsp

```

<%--
    Document   : doLogin
    Created on : 15 Jun 2024, 2:37:52 am
    Author    : Luqman Hakim
--%>

<%@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/csf3017";
    conn = DriverManager.getConnection(myURL, "root", "admin");

    // Query to check if the username and password are valid
    String query = "SELECT * FROM user 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");
}
%>
</body>
</html>

```

Database

```

27 • CREATE TABLE user(
28     username VARCHAR(15) PRIMARY KEY,
29     password VARCHAR(10),
30     firstname VARCHAR(50),
31     lastname VARCHAR(50)
32 );

```

Output

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/Lab6/insertUser.html'. The page title is 'Department of Quality UMT'. Below the title is a 'User Registration' form. The form contains four input fields: 'Username' with the value 'Aliff', 'Password' with masked characters '....', 'Firstname' with the value 'Aliff', and 'Lastname' with the value 'Huat'. At the bottom of the form are two buttons: 'Submit' and 'Cancel'. The footer of the page reads '©Luqman Hakim'.

