



UNIVERSITI MALAYSIA TERENGGANU

CSM3023 – WEB PROGRAMMING 2

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS

LAB 7 – JSP: Perform Create, Update, Retrieve and Delete (CRUD)

SEMESTER II 2023/2024

Prepared for:

DR MOHAMAD NOR BIN HASSAN

Prepared by:

MUHAMMAD ARIF HAIKAL BIN SALLEHUDDIN

(S66355)

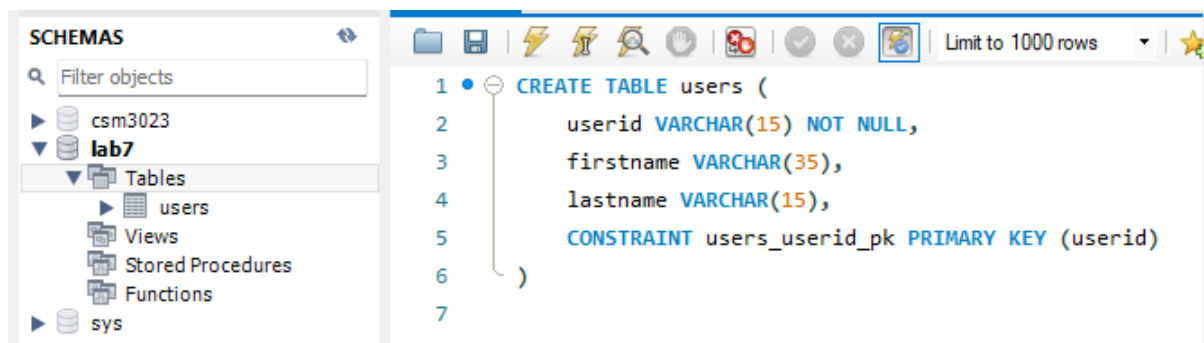
Link Github :

<https://github.com/arifhaikal2001/CSM3023-LAB-7.git>

Step 1 - Create table users in CSF3107 database schema

Create Database in Workbench :

```
CREATE TABLE users (  
    userid VARCHAR(15) NOT NULL,  
    firstname VARCHAR(35),  
    lastname VARCHAR(15),  
    CONSTRAINT users_userid_pk PRIMARY KEY (userid)  
)
```



Step 2 - Create three Java class that representing User (act as a JavaBeans to represent business object), DBConnection (to open and close database connection) and UserDao (act as a Data Access Object (DAO)) to perform CRUD process

User.java

```
package com.model;

/**
 *
 * @author ARIF HAIKAL
 */
public class User {

    private String userid;
    private String firstName;
    private String lastName;

    public String getUserid() {
        return userid;
    }

    public void setUserid(String userid) {
        this.userid = userid;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
}
```

DBConnection.java

```
package com.util;

import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

/**
 *
 * @author ARIF HAIKAL
 */
public class DBConnection {

    private static Connection myConnection = null;
    private static String myURL = "jdbc:mysql://localhost:3306/lab7";

    DBConnection() {

    }

    public static Connection getConnection() throws ClassNotFoundException
    {
        if (myConnection != null)
        {
            return myConnection;
        }
        else
            try
            {
                Class.forName("com.mysql.jdbc.Driver");
                myConnection = DriverManager.getConnection(myURL, "root", "admin@123");
            }
            catch (SQLException e)
            {
                e.printStackTrace();
            }
        return myConnection;
    }
}
```

UserDao.java

```
package com.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import java.util.ArrayList;
import java.util.List;

import com.model.User;
import com.util.DBConnection;

/**
 *
 * @author ARIF HAIKAL
 */
public class UserDao {

    private Connection connection;

    public UserDao() throws ClassNotFoundException {
        connection = DBConnection.getConnection();
    }

    public void addUser(User user) {
        try {
            PreparedStatement preparedStatement = connection
                .prepareStatement("insert into users(userid, firstname, lastname) values (?, ?, ?)");

            preparedStatement.setString(1, user.getUserid());
            preparedStatement.setString(2, user.getFirstName());
            preparedStatement.setString(3, user.getLastName());
            preparedStatement.executeUpdate();

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

    public void deleteUser(String userId) {
        try {
            PreparedStatement preparedStatement = connection
                .prepareStatement("delete from users where userid=?");

            preparedStatement.setString(1, userId);
            preparedStatement.executeUpdate();

        } catch (SQLException e) {
```

```

        e.printStackTrace();
    }
}

public void updateUser(User user) {
    try {
        PreparedStatement preparedStatement = connection
            .prepareStatement("update users set firstname=?, lastname=? " + "where
userid=?");

        preparedStatement.setString(1, user.getFirstName());
        preparedStatement.setString(2, user.getLastName());
        preparedStatement.setString(3, user.getUserid());
        preparedStatement.executeUpdate();

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

public List<User> getAllUsers() {
    List<User> users = new ArrayList<>();
    try {
        Statement statement = connection.createStatement();
        ResultSet rs = statement.executeQuery("SELECT * FROM users");
        while (rs.next()) {
            User user = new User();
            user.setUserid(rs.getString("userid"));
            user.setFirstName(rs.getString("firstname"));
            user.setLastName(rs.getString("lastname"));
            users.add(user);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return users;
}

public User getUserById(String userId) {
    User user = new User();
    try {
        PreparedStatement preparedStatement = connection
            .prepareStatement("SELECT * FROM users WHERE userid=?");
        preparedStatement.setString(1, userId);
        ResultSet rs = preparedStatement.executeQuery();
        if (rs.next()) {
            user.setUserid(rs.getString("userid"));
            user.setFirstName(rs.getString("firstname"));
            user.setLastName(rs.getString("lastname"));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

```
    return user;  
  }  
}
```

Step 3 - Create UserController servlet in order to control and redirect the request to the respective CRUD process and page

UserController.java

```
package com.controller;

/**
 *
 * @author McBois
 */
import java.io.IOException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.RequestDispatcher;
import com.dao.UserDao;
import com.model.User;

@WebServlet(name = "UserController", urlPatterns = {"/UserController"})
public class UserController extends HttpServlet {

    private static final String INSERT = "/user.jsp";
    private static final String EDIT = "/editUser.jsp";
    private static final String LIST_USER = "/listUser.jsp";
    private UserDao dao;

    public UserController() throws ClassNotFoundException {
        super();
        dao = new UserDao();
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String forward = "";
        String action = request.getParameter("action");

        if (action.equalsIgnoreCase("delete")) {
            String userId = request.getParameter("userId");
            dao.deleteUser(userId);
            forward = LIST_USER;
            request.setAttribute("users", dao.getAllUsers());
        } else if (action.equalsIgnoreCase("edit")) {
            forward = EDIT;
            String userId = request.getParameter("userId");
            User user = dao.getUserById(userId);
            request.setAttribute("user", user);
        } else if (action.equalsIgnoreCase("listUser")) {
            forward = LIST_USER;
            request.setAttribute("users", dao.getAllUsers());
        }
    }
}
```



```

    } else {
        forward = INSERT;
    }
    RequestDispatcher view = request.getRequestDispatcher(forward);
    view.forward(request, response);
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String action = request.getParameter("action");
    User user = new User();
    user.setUserid(request.getParameter("userid"));
    user.setFirstName(request.getParameter("firstName"));
    user.setLastName(request.getParameter("lastName"));

    if ("edit".equalsIgnoreCase(action)) {
        dao.updateUser(user);
    } else {
        dao.addUser(user);
    }

    RequestDispatcher view = request.getRequestDispatcher(LIST_USER);
    request.setAttribute("users", dao.getAllUsers());
    view.forward(request, response);
}

@Override
public String getServletInfo() {
    return "UserController Servlet";
}
}

```

Step 4 - Create an index.jsp page that act as a main page

index.jsp

```
<%--
    Document   : index
    Created on : 10 Jun 2024, 12:50:24 am
    Author    : McBois
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Sample Perform CRUD Using Java Servlet</title>
    </head>
    <body>
        <h1>Sample Perform CRUD Using Java Servlet!</h1>
        <jsp:forward page="/UserController?action=listUser" />
    </body>
</html>
```

Step 5 - Create listUser.jsp page to perform retrieving of a list of users.

listUser.jsp

```
<%--
    Document : listUser
    Created on : 10 Jun 2024, 12:50:42 am
    Author : McBois
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>User List</title>
    </head>
    <body>
        <h1>List of Users</h1>
        <table border="1">
            <thead>
                <tr>
                    <th>User ID</th>
                    <th>First Name</th>
                    <th>Last Name</th>
                    <th colspan="2">Action</th>
                </tr>
            </thead>
            <tbody>
                <c:forEach items="${users}" var="user">
                    <tr>
                        <td><c:out value="${user.userid}" /></td>
                        <td><c:out value="${user.firstName}" /></td>
                        <td><c:out value="${user.lastName}" /></td>
                        <td><a
href="UserController?action=edit&userId=${user.userid}">Update</a></td>
                        <td><a
href="UserController?action=delete&userId=${user.userid}">Delete</a></td>
                    </tr>
                </c:forEach>
            </tbody>
        </table>
        <p><a href="UserController?action=insert">Add User</a></p>
    </body>
</html>
```

Step 6 - Create user.jsp page as a page for creating new record for user.

user.java

```
<%--
    Document : User
    Created on : 10 Jun 2024, 12:51:02 am
    Author : McBois
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>New Record</title>
    </head>
    <body>
        <p><b>New Record</b></p>
        <form name="frmAddUser" action="UserController" method="POST">
            <table border="0">
                <tbody>
                    <tr>
                        <td>User ID:</td>
                        <td><input type="text" name="userid" size="25" required /></td>
                    </tr>
                    <tr>
                        <td>First Name:</td>
                        <td><input type="text" name="firstName" size="40" /></td>
                    </tr>
                    <tr>
                        <td>Last Name:</td>
                        <td><input type="text" name="lastName" size="40" /></td>
                    </tr>
                    <tr>
                        <td><input type="submit" value="Submit" name="submit" /></td>
                        <td><input type="reset" value="Cancel" name="cancel" /></td>
                    </tr>
                </tbody>
            </table>
        </form>
    </body>
</html>
```

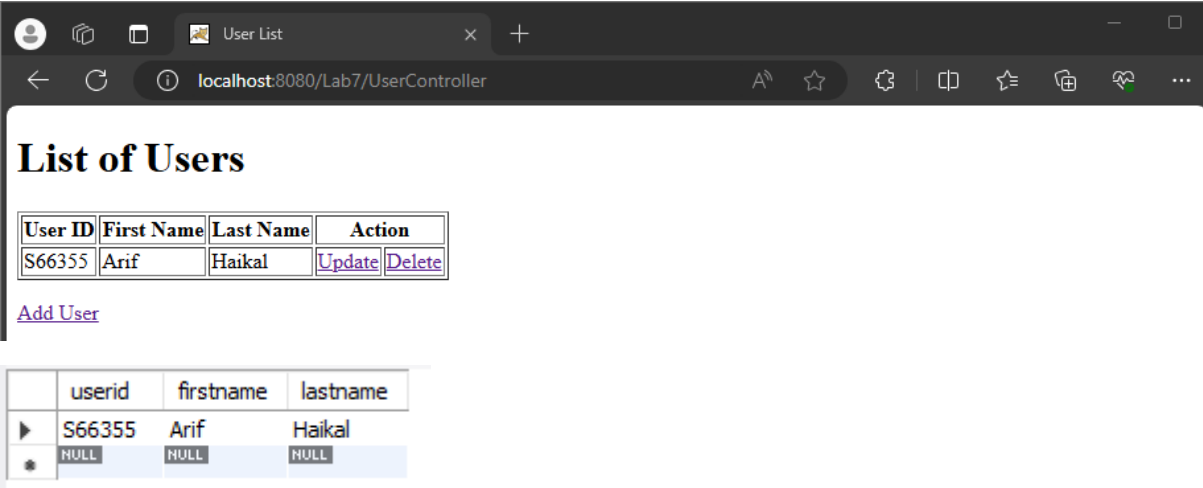
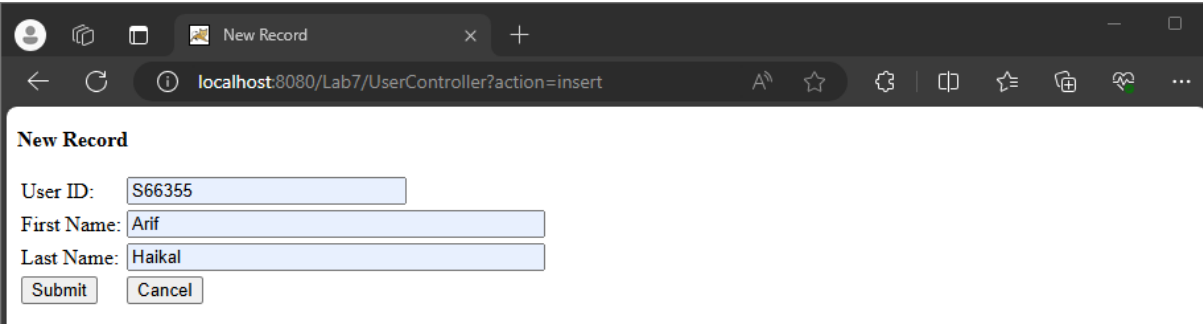
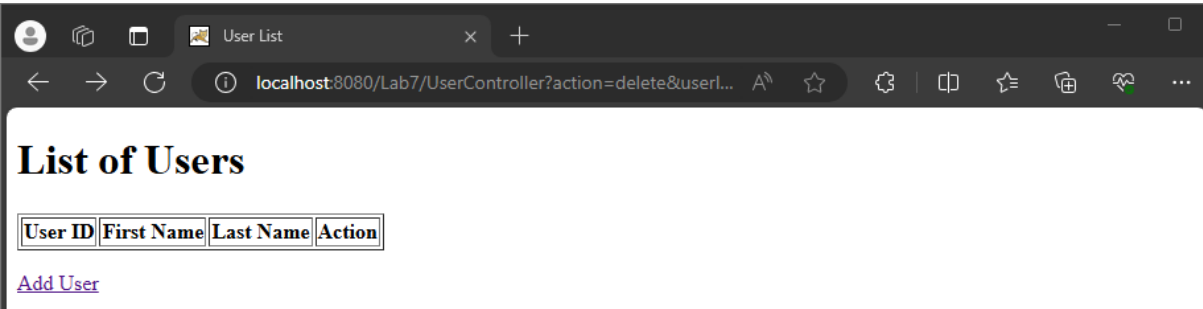
Step 7 - Create editUser.jsp page as a page for updating existing record for specific user.

editUser.java

```
<%--
  Document   : editUser
  Created on : 10 Jun 2024, 12:51:33 am
  Author    : McBois
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Edit User</title>
  </head>
  <body>
    <p>Updating User Profile</p>
    <form name="frmEditUser" action="UserController" method="POST">
      <table border="0">
        <tbody>
          <tr>
            <td>User ID:</td>
            <td><input type="text" name="userid" readonly="readonly" value="<c:out
value='${user.userid}' />" size="25" /></td>
          </tr>
          <tr>
            <td>First Name:</td>
            <td><input type="text" name="firstName" value="<c:out
value='${user.firstName}' />" size="40" /></td>
          </tr>
          <tr>
            <td>Last Name:</td>
            <td><input type="text" name="lastName" value="<c:out
value='${user.lastName}' />" size="40" /></td>
          </tr>
          <tr>
            <td><input type="hidden" name="action" value="edit" /></td>
          </tr>
          <tr>
            <td><input type="submit" name="Submit" value="submit" /></td>
          </tr>
        </tbody>
      </table>
    </form>
  </body>
</html>
```

Output :



Update :

Edit User

localhost:8080/Lab7/UserController?action=edit&userId=S66355

Updating User Profile

User ID:

S66355

First Name:

Muhammad

Last Name:

Arif Haikal

submit

User List

localhost:8080/Lab7/UserController

List of Users

User ID	First Name	Last Name	Action
S66355	Muhammad	Arif Haikal	Update Delete

[Add User](#)

	userid	firstname	lastname
	S66355	Muhammad	Arif Haikal
*	NULL	NULL	NULL

Delete :

User List

localhost:8080/Lab7/UserController?action=delete&userl...

List of Users

User ID	First Name	Last Name	Action
---------	------------	-----------	--------

[Add User](#)

	userid	firstname	lastname
*	NULL	NULL	NULL

Reflection:**1. Why we use servlet for Java Web Application?**

Servlets allow the server to manage client requests and provide dynamic web content, we utilise them for Java web applications. Servlets are necessary for developing responsive and interactive web applications because they handle incoming requests, communicate with databases or other resources, and return results to the client.

Exercise

Q1) Implement profile registration using servlet

profileForm.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Profile Registration</title>
</head>
<body>
  <h2>Profile Registration</h2>
  <fieldset>
    <legend>Profile Registration</legend>
    <form action="profileServlet" method="post">
      Username: <br><input type="text" name="username" maxlength="15"
required><br>
      IC No: <br><input type="text" name="icno" maxlength="15" required><br>
      First Name: <br><input type="text" name="firstname" maxlength="50"
required><br>
      <br>
      <input type="submit" value="Register">
    </form>
  </fieldset>
</body>
<footer>
  &copy;2024 ARIF HAIKAL
</footer>
</html>
```

profileServlet.java

```
package com.exercise;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

/**
 *
 * @author ARIF HAIKAL
 */

@WebServlet("/profileServlet")
public class profileServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    // Database connection details
    private static final String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
    private static final String DB_URL = "jdbc:mysql://localhost:3306/CSM3023";
    private static final String DB_USER = "root";
    private static final String DB_PASSWORD = "admin@123";

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        String username = request.getParameter("username");
        String icno = request.getParameter("icno");
        String firstname = request.getParameter("firstname");

        Connection conn = null;
        PreparedStatement stmt = null;

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        try {
            // Load the JDBC driver
            Class.forName(JDBC_DRIVER);

            // Connect to the database
            conn = DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);

            // Prepare SQL query
            String sql = "INSERT INTO userprofile (username, icno, firstname) VALUES (?, ?, ?)";
        }
    }
}
```

```

    stmt = conn.prepareStatement(sql);
    stmt.setString(1, username);
    stmt.setString(2, icno);
    stmt.setString(3, firstname);

    // Execute SQL query
    int rows = stmt.executeUpdate();

    // Generate response
    if (rows > 0) {
        out.println("<h2>Profile registered successfully!</h2>");
    } else {
        out.println("<h2>Error registering profile!</h2>");
    }

} catch (ClassNotFoundException e) {
    out.println("<h2>JDBC Driver not found!</h2>");
    e.printStackTrace(out);
} catch (SQLException e) {
    out.println("<h2>Database error!</h2>");
    e.printStackTrace(out);
} finally {
    try {
        if (stmt != null) stmt.close();
        if (conn != null) conn.close();
    } catch (SQLException se) {
        se.printStackTrace(out);
    }
}
}
}
}
}

```

Output :

Profile Registration

Profile Registration

Username:
muhdarifhaikal

IC No:
010328140795

First Name:
ARIF

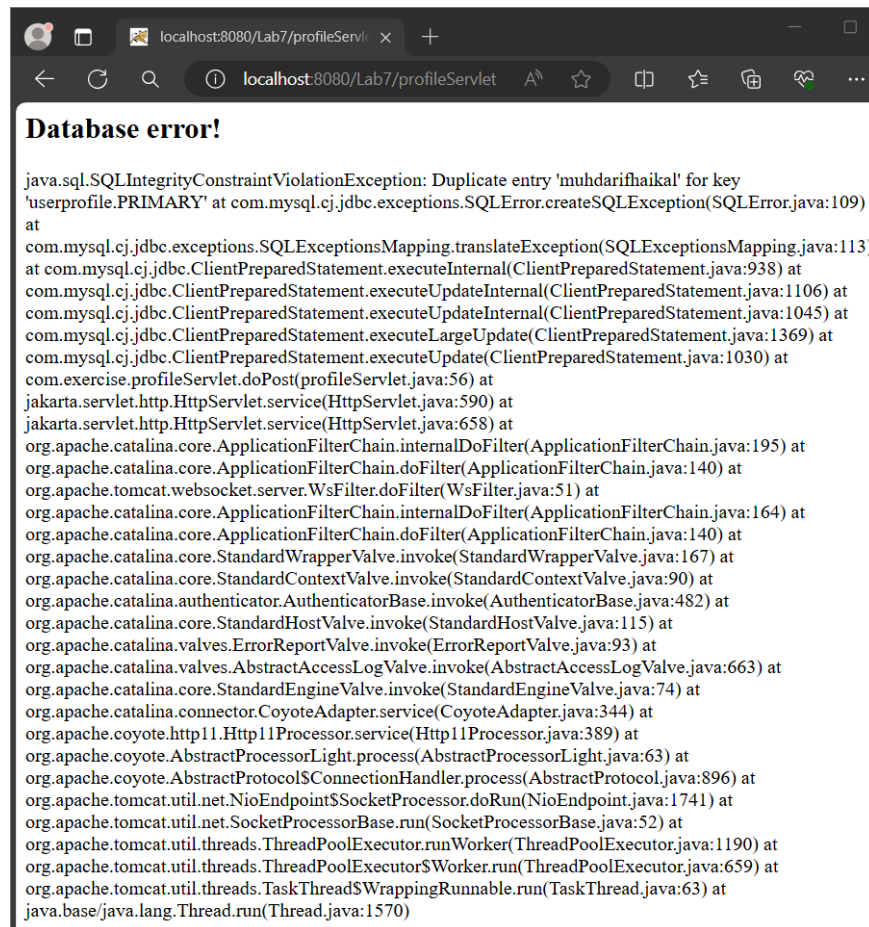
Register

©2024 ARIF HAIKAL

Profile registered successfully!

	username	icno	firstname
▶	muhdarifhaikal	010328140795	ARIF
*	NULL	NULL	NULL

When you duplicate :



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/Lab7/profileServlet'. The page content is titled 'Database error!' and displays a detailed Java stack trace. The error is a 'java.sql.SQLIntegrityConstraintViolationException: Duplicate entry 'muhdarifhaikal' for key 'userprofile.PRIMARY'' occurring at 'com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:109)'. The stack trace continues through various MySQL and Tomcat classes, including 'com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException', 'com.mysql.cj.jdbc.ClientPreparedStatement.executeInternal', 'com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdateInternal', 'com.mysql.cj.jdbc.ClientPreparedStatement.executeLargeUpdate', 'com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdate', 'com.exercise.profileServlet.doPost', 'jakarta.servlet.http.HttpServlet.service', and 'org.apache.catalina.core.ApplicationFilterChain.internalDoFilter'.

```
java.sql.SQLIntegrityConstraintViolationException: Duplicate entry 'muhdarifhaikal' for key
'userprofile.PRIMARY' at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:109)
at
com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:113)
at com.mysql.cj.jdbc.ClientPreparedStatement.executeInternal(ClientPreparedStatement.java:938) at
com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdateInternal(ClientPreparedStatement.java:1106) at
com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdateInternal(ClientPreparedStatement.java:1045) at
com.mysql.cj.jdbc.ClientPreparedStatement.executeLargeUpdate(ClientPreparedStatement.java:1369) at
com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdate(ClientPreparedStatement.java:1030) at
com.exercise.profileServlet.doPost(profileServlet.java:56) at
jakarta.servlet.http.HttpServlet.service(HttpServlet.java:590) at
jakarta.servlet.http.HttpServlet.service(HttpServlet.java:658) at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:195) at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:140) at
org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:51) at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:164) at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:140) at
org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:167) at
org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:90) at
org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:482) at
org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:115) at
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:93) at
org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.java:663) at
org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:74) at
org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:344) at
org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:389) at
org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:63) at
org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:896) at
org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1741) at
org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:52) at
org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1190) at
org.apache.tomcat.util.threads.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:659) at
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:63) at
java.base/java.lang.Thread.run(Thread.java:1570)
```

Q2) Applying session in student registration.

studentRegister.jsp

```
<%--
    Document : studentRegister
    Created on : 7 Jun 2024, 5:21:13 pm
    Author : ARIF HAIKAL
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <title>Student Registration</title>
</head>
<body>
    <h2>Student Registration</h2>
    <fieldset>
        <legend>Student Registration</legend>
        <form action="confirmRegister.jsp" method="post">
            Student ID: <br><input type="text" name="studentid" required><br>
            Name: <br><input type="text" name="name" required><br>
            <br>
            <input type="submit" value="Submit">
        </form>
    </fieldset>
    <br>
</body>
<footer>
    &copy;2024 ARIF HAIKAL
</footer>
</html>
```

confirmRegister.jsp

```
<%--
  Document   : confirmRegister
  Created on : 7 Jun 2024, 5:20:04 pm
  Author    : ARIF HAIKAL
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="jakarta.servlet.http.HttpSession" %>
<%
  String studentid = request.getParameter("studentid");
  String name = request.getParameter("name");

  // Use the implicit session object directly without declaring it
  session.setAttribute("studentid", studentid);
  session.setAttribute("name", name);
%>
<!DOCTYPE html>
<html>
<head>
  <title>Confirm Registration</title>
</head>
<body>
  <h2>Confirm Registration</h2>
  <fieldset>
    <p>Student ID: <%= studentid %></p>
    <p>Name: <%= name %></p>
  </fieldset>
  <br>
  <form action="notificationRegister.jsp" method="post">
    <input type="submit" value="Proceed">
  </form>

</body>
</html>
```

notificationRegister.jsp

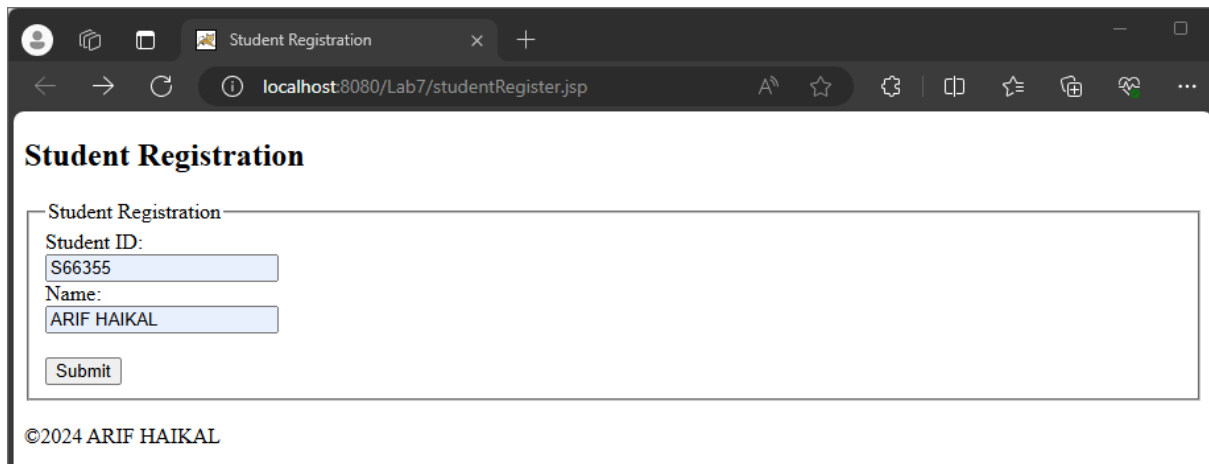
```
<%--
  Document   : notificationRegister
  Created on : 5 Jun 2024, 3:43:09 pm
  Author    : ARIF HAIKAL
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="jakarta.servlet.http.HttpSession" %>
<%
  // Use the implicit session object directly without declaring it
  HttpSession currentSession = request.getSession(false);
  if (currentSession == null || currentSession.getAttribute("studentid") == null ||
currentSession.getAttribute("name") == null) {
    response.sendRedirect("studentRegister.jsp");
    return;
  }

  String studentid = (String) currentSession.getAttribute("studentid");
  String name = (String) currentSession.getAttribute("name");

  currentSession.invalidate();
%>
<!DOCTYPE html>
<html>
<head>
  <title>Registration Notification</title>
</head>
<body>
  <h2>Registration Notification</h2>
  <fieldset>
    <p>Student ID: <%= studentid %></p>
    <p>Name: <%= name %></p>
    <p>Your registration has been successfully completed.</p>
  </fieldset>
</body>
<br>
<footer>
  &copy;2024 ARIF HAIKAL
</footer>
</html>
```


Output :



Student Registration

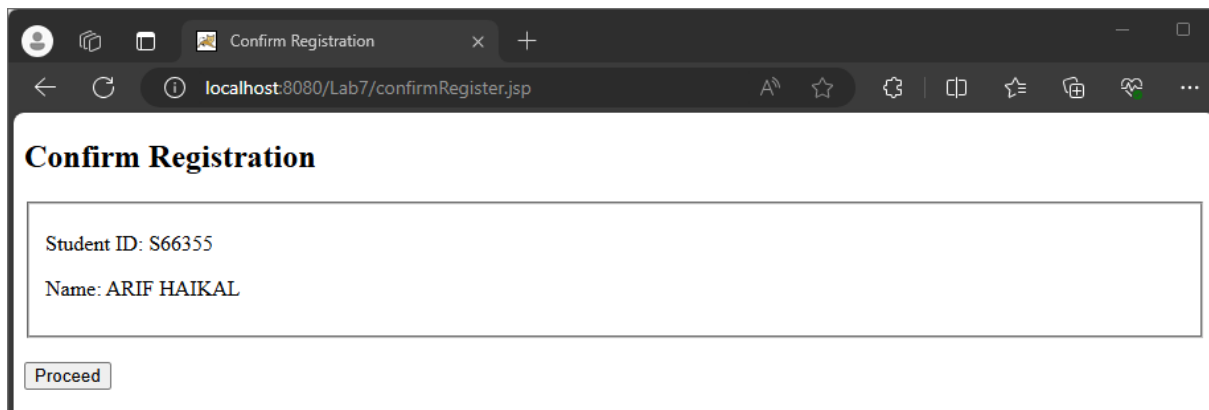
Student Registration

Student ID:
S66355

Name:
ARIF HAIKAL

Submit

©2024 ARIF HAIKAL

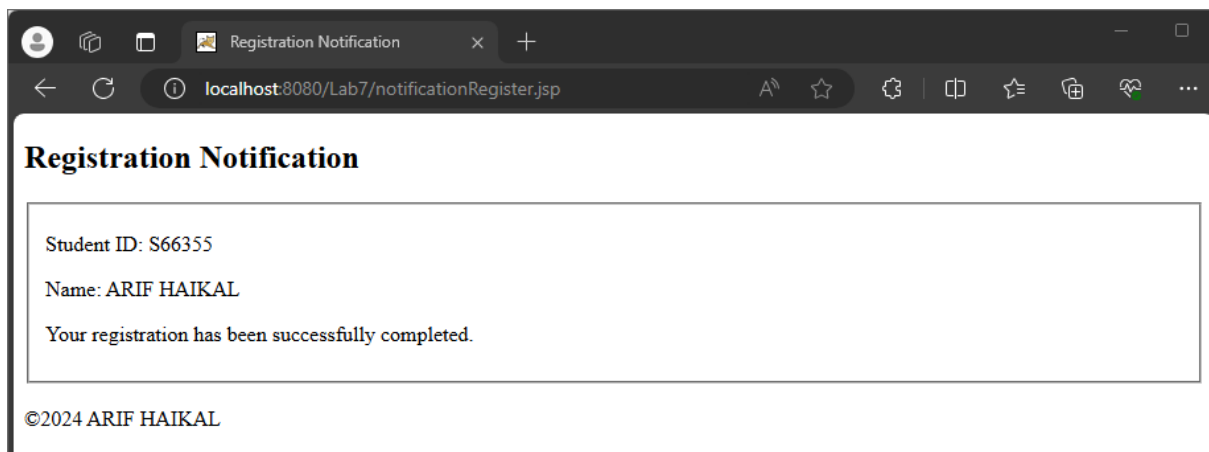


Confirm Registration

Student ID: S66355

Name: ARIF HAIKAL

Proceed



Registration Notification

Student ID: S66355

Name: ARIF HAIKAL

Your registration has been successfully completed.

©2024 ARIF HAIKAL