

Web based Application Development (CSM3023)

Lab Report 7  
JSP: Perform Create, Update, Retrieve and Delete (CRUD)

Programme

Computer Science (Software Engeenering)

Prepared by

NurSyaza Amira Binti Selamat

Matric No

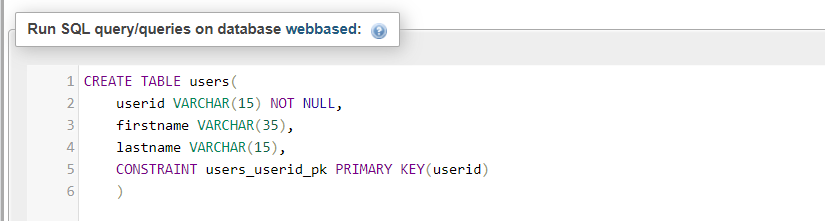
S62318

**Prepared for**

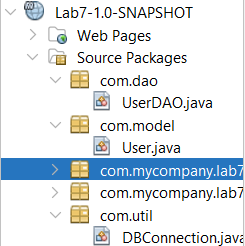
Sir Arizal

**Task 1: Perform Basic CRUIS Process Using Java Servlet**

Step 1: Create table user in webbased database schema



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 Data Access Object(DAO)) to perform CRUD process



User.java

package com.model;

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.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DBConnection implements java.io.Serializable {

private static Connection myConnection = null;

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

public DBConnection() {

}

public static Connection getConnection() throws ClassNotFoundException, SQLException {

if (myConnection != null) {

return myConnection;

} else {

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

myConnection = DriverManager.getConnection(myURL, "root", "admin");

return myConnection;

}

}

public void closeConnection() throws ClassNotFoundException, SQLException {

myConnection.close();

}

}  
  
UserDAO.java

package com.dao;

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;

public class UserDAO {

public void addUser(User user) throws ClassNotFoundException, SQLException {

PreparedStatement myPS = DBConnection.getConnection()

.prepareStatement(

"INSERT INTO users (userid, firstname, lastname)VALUES( ?, ?, ?)");

myPS.setString(1, user.getUserid());

myPS.setString(2, user.getFirstname());

myPS.setString(3, user.getLastname());

myPS.executeUpdate();

}

public void deleteUser(String userid) throws ClassNotFoundException, SQLException {

PreparedStatement myPS = DBConnection.getConnection()

.prepareStatement("DELETE FROM user WHERE userid=?");

myPS.setString(1, userid);

myPS.executeUpdate();

}

public void updateUser(User user) throws ClassNotFoundException, SQLException {

PreparedStatement myPS = DBConnection.getConnection()

.prepareStatement("UPDATE user SET firstname=?, lastname=?"

+ "WHERE userid=?");

myPS.setString(3, user.getUserid());

myPS.setString(1, user.getFirstname());

myPS.setString(2, user.getLastname());

myPS.executeUpdate();

}

public List<User> getAllUsers() throws ClassNotFoundException, SQLException {

List<User> users = new ArrayList<>();

Statement statement = DBConnection.getConnection().createStatement();

ResultSet rs = statement.executeQuery("SELECT \* FROM user");

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);

}

return users;

}

public User getUserbyId(String userid) throws ClassNotFoundException, SQLException {

User user = new User();

PreparedStatement myPS = DBConnection.getConnection()

.prepareStatement("SELECT \* FROM user WHERE userid=?");

myPS.setString(1, userid);

ResultSet rs = myPS.executeQuery();

while (rs.next()) {

user.setUserid(rs.getString("userid"));

user.setFirstname(rs.getString("firstname"));

user.setLastname(rs.getString("lastname"));

}

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;

import java.io.IOException;

import jakarta.servlet.RequestDispatcher;

import jakarta.servlet.ServletException;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import com.dao.UserDAO;

import com.model.User;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

public class UserController extends HttpServlet {

private static String INSERT = "/User.jsp";

private static String EDIT = "/EditUser.jsp";

private static String LIST\_USER = "/ListUser.jsp";

private UserDAO dao;

public UserController() throws ClassNotFoundException {

super();

dao = new UserDAO();

}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

/\*\*

\* Handles the HTTP <code>GET</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@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");

try {

dao.deleteUser(userid);

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(UserController.class.getName()).log(Level.SEVERE, null,

ex);

}

forward = LIST\_USER;

try {

request.setAttribute("users", dao.getAllUsers());

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(UserController.class.getName()).log(Level.SEVERE, null,

ex);

}

} else if (action.equalsIgnoreCase("edit")) {

try {

forward = EDIT;

String userid = request.getParameter("userid");

User user = dao.getUserbyId(userid);

request.setAttribute("user", user);

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(UserController.class.getName()).log(Level.SEVERE, null,

ex);

}

} else if (action.equalsIgnoreCase("listUser")) {

try {

forward = LIST\_USER;

request.setAttribute("users", dao.getAllUsers());

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(UserController.class.getName()).log(Level.SEVERE, null,

ex);

}

} else if (action.equalsIgnoreCase("insert")) {

forward = INSERT;

}

RequestDispatcher view = request.getRequestDispatcher(forward);

view.forward(request, response);

}

/\*\*

\* Handles the HTTP <code>POST</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@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 (action.equalsIgnoreCase("edit")) {

try {

dao.updateUser(user);

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(UserController.class.getName()).log(Level.SEVERE, null,

ex);

}

} else {

try {

dao.addUser(user);

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(UserController.class.getName()).log(Level.SEVERE, null,

ex);

}

}

RequestDispatcher view = request.getRequestDispatcher(LIST\_USER);

try {

request.setAttribute("users", dao.getAllUsers());

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(UserController.class.getName()).log(Level.SEVERE, null,

ex);

}

view.forward(request, response);

}

/\*\*

\* Returns a short description of the servlet.

\*

\* @return a String containing servlet description

\*/

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

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

Index.jsp

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

<%@page import="com.controller.UserController"%>

<!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>

<footer>NurSyaza Amira @2024</footer>

</html>  
  
Step 5: Create ListUser.jsp page to perform retrieving of a list of users

ListUser.jsp

<%@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>

<p>List of Users..!</p>

<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=<c:out

value="${user.userid}" />">Update</a></td>

<td><a href="UserController?action=delete&userid=<c:out

value="${user.userid}" />">Delete</a></td>

</tr>

</c:forEach>

</tbody>

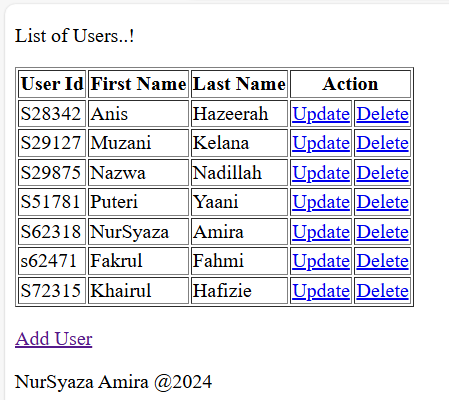
</table>

<p><a href="UserController?action=insert">Add User</a></p>

</body>

<footer>NurSyaza Amira @2024</footer>

</html>



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

User.jsp

<%@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="<%=request.getContextPath()%>/UserController"

method="POST">

<table border="0">

<tr>

<td>User ID: </td>

<td><input type="text" name="userid" value="" size="40"

required></td>

</tr>

<tr>

<td>First Name:</td>

<td><input type="text" name="firstname" value="" size="40"

required></td>

</tr>

<tr>

<td>Last Name:</td>

<td><input type="text" name="lastname" value="" size="40"

required></td>

</tr>

<tr>

<td><input type="hidden" name="action" value="insert"></td>

<td></td>

</tr>

<tr>

<td>

<input type="submit" name="submit" value="Submit">

<input type="reset" name="cancel" value="Cancel">

</td>

<td></td>

</tr>

</table>

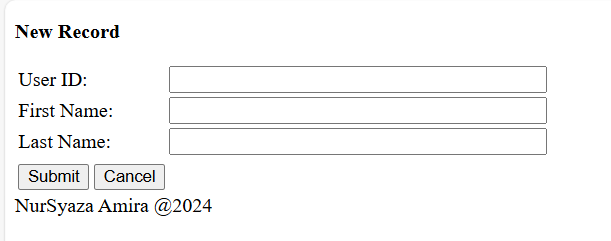
</form>

</body>

<footer>NurSyaza Amira @2024</footer>

</html>

Step 7: Create EditUser.jsp page as a page for updating existing record for specific user.



EditUser.jsp

Bottom of Form

<%@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</p>

<form name="frmEditUser" action="UserController" method="POST">

<table border="1">

<tbody>

<tr>

<td>User Id:</td>

<td><input type="text" name="userid" value="<c:out

value="${user.userid}" />" size="25" readonly="readonly"></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>

<td></td>

</tr>

<tr>

<td>

<input type="submit" name="submit" value="Submit">

</td>

<td></td>

</tr>

</tbody>

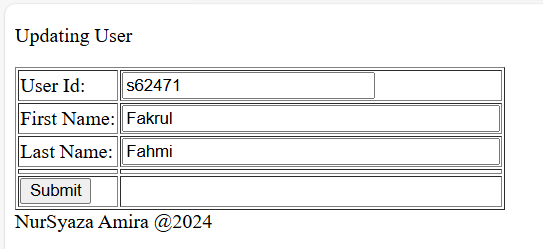
</table>

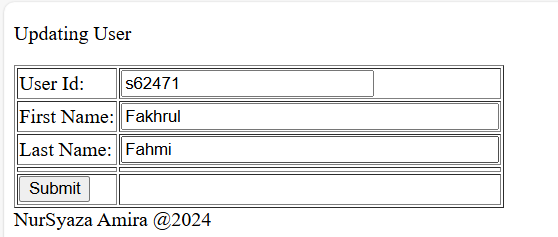
</form>

</body>

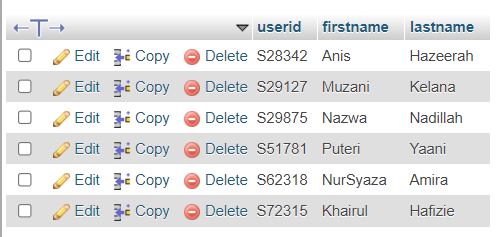
<footer>NurSyaza Amira @2024</footer>

</html>









**Exercise**

Q1) Implement profile registration using servlet

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

• icno as a character length 15

• firstname as varchar(50)

2. Create an entry form.

3. Create a servlet known as profileServlet.

4. Use profileServlet to ackhowledge user about the profile registration.  
  
userprofile table

CREATE TABLE userprofile (

username CHAR(15) PRIMARY KEY,

icno CHAR(15),

firstname VARCHAR(50)

);  
  
profileRegister.jsp

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

<!DOCTYPE html>

<html>

<head>

<title>Profile Registration</title>

</head>

<body>

<h2>Profile Registration</h2>

<form action="profileServlet" method="post">

<table>

<tr>

<td>Username:</td>

<td><input type="text" name="username" required></td>

</tr>

<tr>

<td>IC Number:</td>

<td><input type="text" name="icno" required></td>

</tr>

<tr>

<td>First Name:</td>

<td><input type="text" name="firstname" required></td>

</tr>

<tr>

<td colspan="2"><input type="submit" value="Register"></td>

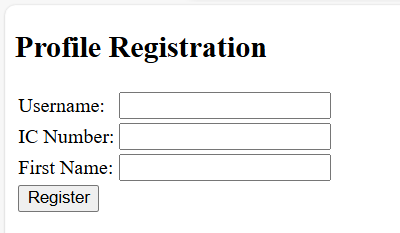
</tr>

</table>

</form>

</body>

</html>





profileServlet.java  
  
package com.controller;

import java.io.IOException;

import java.sql.Connection;

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;

import com.util.DBConnection;

@WebServlet("/profileServlet")

public class profileServlet extends HttpServlet {

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;

try {

conn = DBConnection.getConnection();

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);

stmt.executeUpdate();

request.setAttribute("message", "Profile registered successfully!");

} catch (Exception e) {

e.printStackTrace();

request.setAttribute("message", "Error registering profile.");

} finally {

if (stmt != null) {

try {

stmt.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if (conn != null) {

try {

conn.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

request.getRequestDispatcher("profileConfirmation.jsp").forward(request, response);

}

}

profileConfirmation.jsp

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

<!DOCTYPE html>

<html>

<head>

<title>Profile Registration Confirmation</title>

</head>

<body>

<h2>Profile Registration Confirmation</h2>

<p>${message}</p>

<a href="profileRegister.jsp">Register Another Profile</a>

</body>

</html>



Q2) Applying session in student registration.

1. Create main interface for student registration; studentid, name.

(studentRegister.jsp)

2. When student click Submit button, it will redirect to confirmation page

(confirmRegister.jsp)

3. When user click Proceed button, current page will forward notification

to end user via Notification page (notificationRegister.jsp)

studentRegister.jsp

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

<!DOCTYPE html>

<html>

<head>

<title>Student Registration</title>

</head>

<body>

<h2>Student Registration</h2>

<form action="confirmRegister.jsp" method="post">

<table>

<tr>

<td>Student ID:</td>

<td><input type="text" name="studentid" required></td>

</tr>

<tr>

<td>Name:</td>

<td><input type="text" name="name" required></td>

</tr>

<tr>

<td colspan="2"><input type="submit" value="Submit"></td>

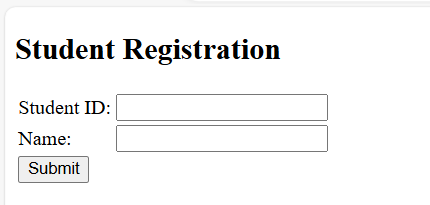
</tr>

</table>

</form>

</body>

</html>



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

<%@page import="jakarta.servlet.http.HttpSession"%>

<!DOCTYPE html>

<html>

<head>

<title>Confirm Registration</title>

</head>

<body>

<h2>Confirm Registration</h2>

<%

String studentid = request.getParameter("studentid");

String name = request.getParameter("name");

HttpSession session = request.getSession();

session.setAttribute("studentid", studentid);

session.setAttribute("name", name);

%>

<p>Student ID: <%= studentid %></p>

<p>Name: <%= name %></p>

<form action="notificationRegister.jsp" method="post">

<input type="submit" value="Proceed">

</form>

</body>

</html>

notifcationRegister.jsp

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

<%@page import="jakarta.servlet.http.HttpSession"%>

<!DOCTYPE html>

<html>

<head>

<title>Registration Notification</title>

</head>

<body>

<h2>Registration Notification</h2>

<%

HttpSession session = request.getSession(false);

if (session != null) {

String studentid = (String) session.getAttribute("studentid");

String name = (String) session.getAttribute("name");

if (studentid != null && name != null) {

out.println("<p>Student ID: " + studentid + "</p>");

out.println("<p>Name: " + name + "</p>");

out.println("<p>Registration successful!</p>");

session.invalidate(); // Invalidate the session after use

} else {

out.println("<p>No student information found. Please register again.</p>");

}

} else {

out.println("<p>No session found. Please register again.</p>");

}

%>

<a href="studentRegister.jsp">Register Another Student</a>

</body>

</html>