CSE 3023

WEB-BASED APPLICATION DEVELOPMENT

LAB WEEK 7

SERVLET: PERFORM CREATE, RETRIEVE, UPDATE AND DELETE (CRUD)

LAB DEMONSTRATOR

SIR ARIZAL SHAMSIL

PREPARED BY

NAME	MATRIC NO.
NUR HIDAYATUL AIN BINTI HALUWI	S62367

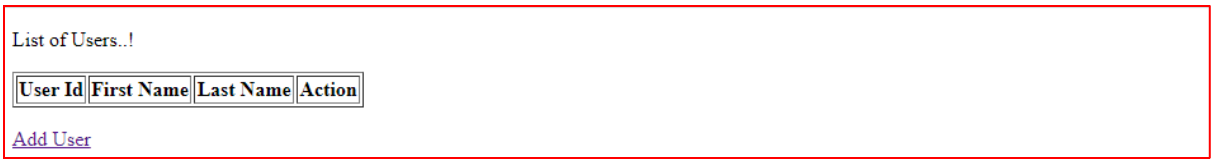
FACULTY OF OCEAN ENGINEERING TECHNOLOGY AND INFORMATICS

UNIVERSITI MALAYSIA TERENGGANU

Task 1: Perform Basic CRUD Process Using Java Servlet

Running the program and perform CRUD process.

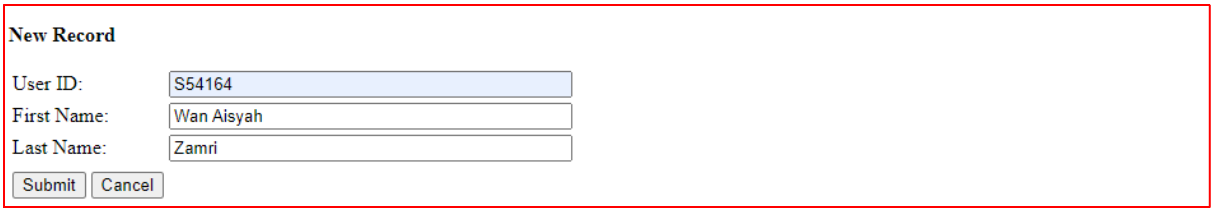
1.1 Run index.jsp



List of Users..!

User Id	First Name	Last Name	Action

1.2 Click Add User button to create new record.

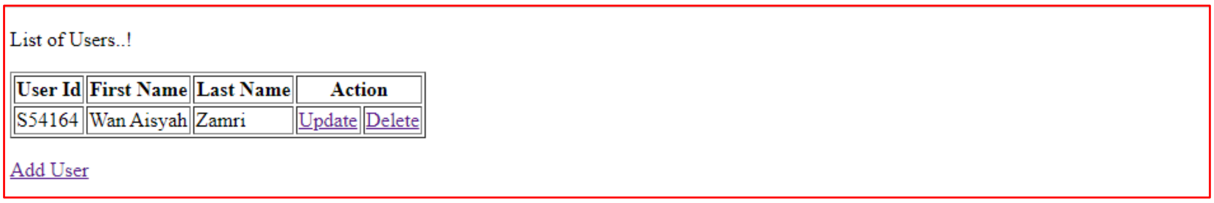


New Record

User ID: S54164

First Name: Wan Aisyah

Last Name: Zamri

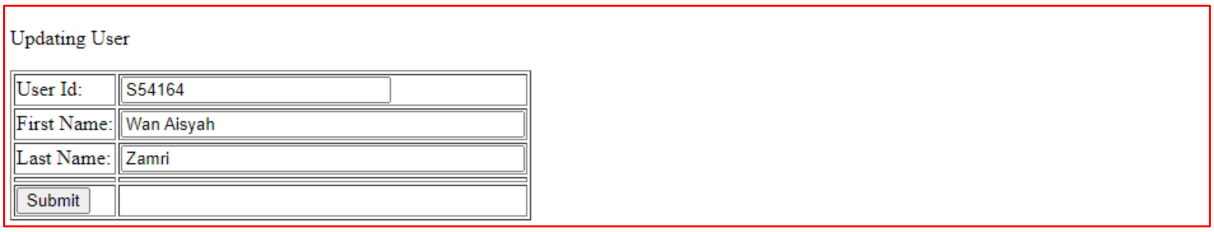


List of Users..!

User Id	First Name	Last Name	Action	
S54164	Wan Aisyah	Zamri	Update	Delete

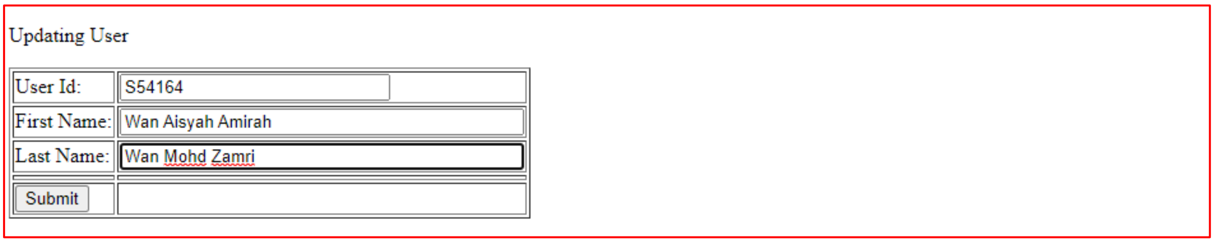
The record is successfully added into the list!

1.3 Click hyperlink Update in order to update an existing record.



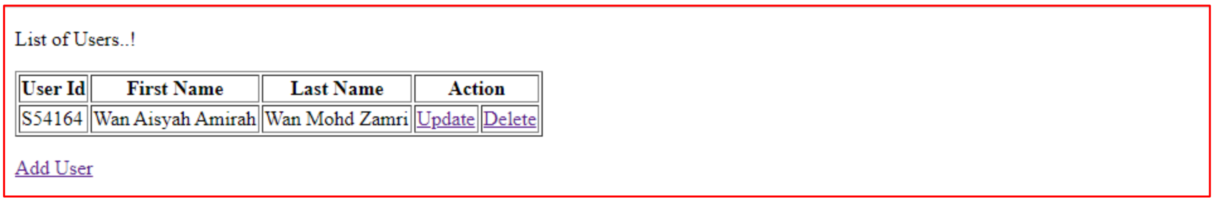
Updating User

User Id:	S54164
First Name:	Wan Aisyah
Last Name:	Zamri
Submit	



Updating User

User Id:	S54164
First Name:	Wan Aisyah Amirah
Last Name:	Wan Mohd Zamri
Submit	



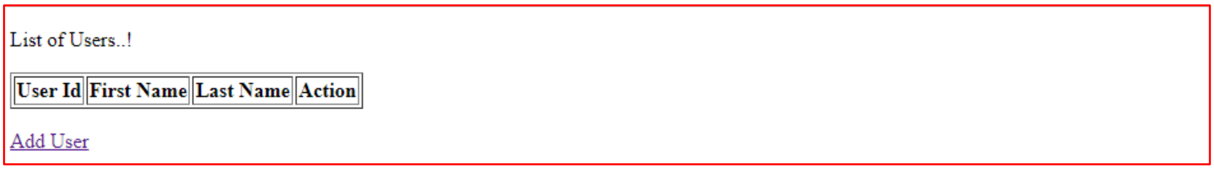
List of Users..!

User Id	First Name	Last Name	Action	
S54164	Wan Aisyah Amirah	Wan Mohd Zamri	Update	Delete

Add User

The changes made for User ID: S54164 is updated in the list.

1.4 Click hyperlink Delete in order to delete an existing record.



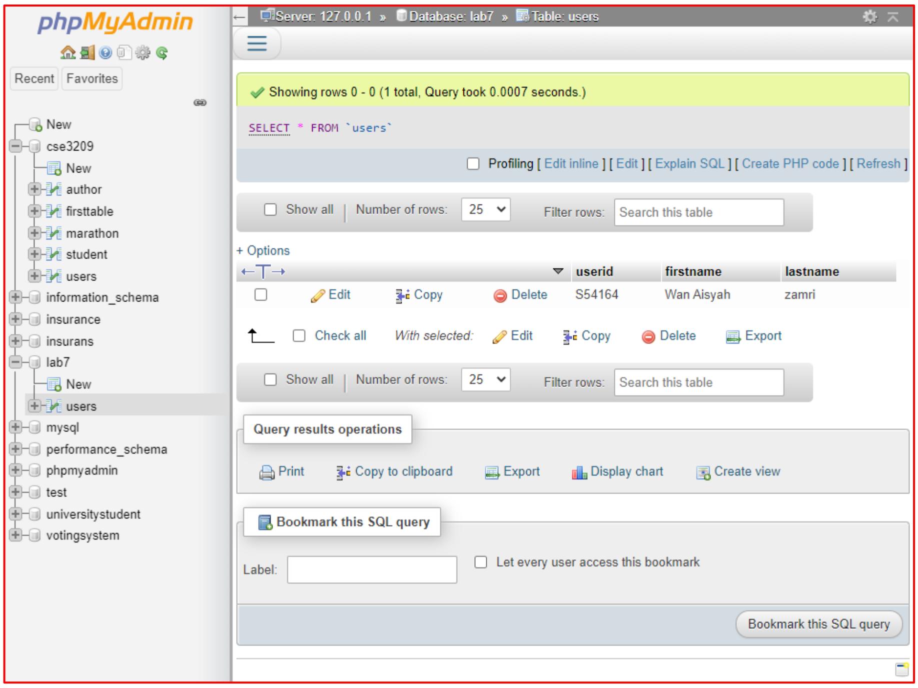
List of Users..!

User Id	First Name	Last Name	Action

Add User

Upon clicking the delete button, no record exists in the list anymore.

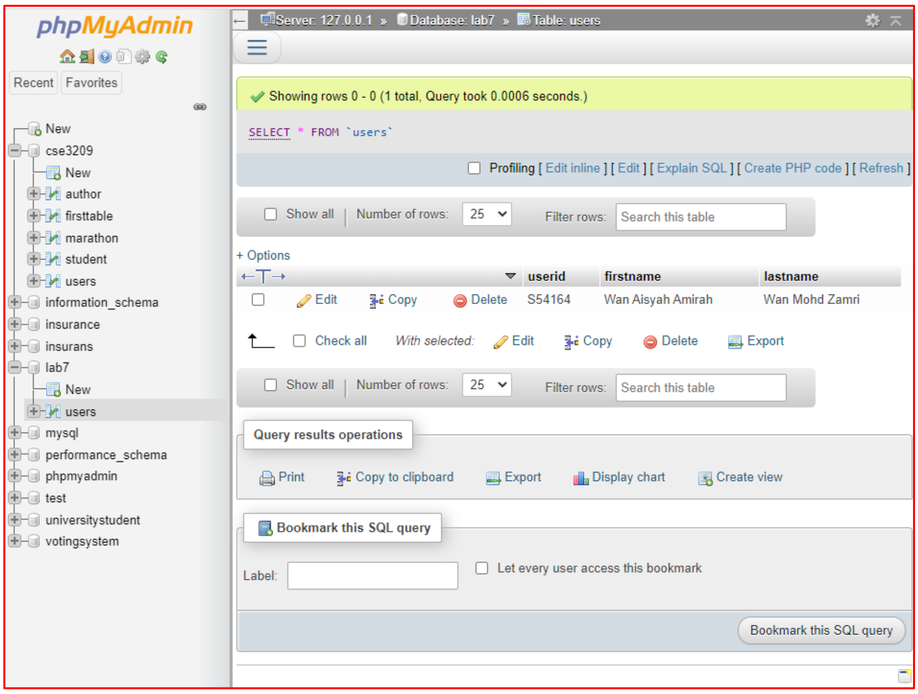
1.5 phpMyAdmin View


phpMyAdmin
Recent
Favorites
New
cse3209
New
author
firsttable
marathon
student
users
information_schema
insurance
insurans
lab7
New
users
mysql
performance_schema
phpmyadmin
test
universitystudent
votingsystem
Server: 127.0.0.1 » Database: lab7 » Table: users
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
+ Options
□ Edit Copy Delete
□ Check all With selected: Edit Copy Delete Export
□ Show all | Number of rows: 25 Filter rows: Search this table
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Label:
□ Let every user access this bookmark
Bookmark this SQL query

✓ Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)

```
SELECT * FROM `users`
```

userid	firstname	lastname
S54164	Wan Aisyah	zamri


phpMyAdmin
Recent
Favorites
New
cse3209
New
author
firsttable
marathon
student
users
information_schema
insurance
insurans
lab7
New
users
mysql
performance_schema
phpmyadmin
test
universitystudent
votingsystem
Server: 127.0.0.1 » Database: lab7 » Table: users
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
+ Options
□ Edit Copy Delete
□ Check all With selected: Edit Copy Delete Export
□ Show all | Number of rows: 25 Filter rows: Search this table
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Label:
□ Let every user access this bookmark
Bookmark this SQL query

✓ Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)

```
SELECT * FROM `users`
```

userid	firstname	lastname
S54164	Wan Aisyah Amirah	Wan Mohd Zamri

1.6 Source Codes

A) User.java

```
package com.model;

/**
 *
 * @author
 */
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;
    }
}
```

B) DBConnection.java

```
package com.util;

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

/**
 *
 * @author
 */
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();
    }

}
```

C) UserController.java

```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;
/**
 *
 * @author
 */
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>

}
```

D) UserDao.java

```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;

/**
 *
 * @author
 */
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;
    }

}
```

E) user.jsp

```
<%--
    Document   : user

--%>

<%@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>
</html>
```

F) index.jsp

```
<%--
    Document   : index

--%>

<%@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>
</html>
```

G) listUser.jsp

```
<%--
    Document   : listUser

--%>

<%@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>
</html>
```

H) editUser.jsp

```
<%--
    Document   : editUser

--%>

<%@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>
</html>
```

1.7 Refelction

1. What have you learnt from this exercise?

   Based on this exercise, I have learnt on how to perform creating, retrieving, updating, and deleting CRUD records from MySQL database.