UNIVERSITI MALAYSIA TERENGGANU

CSM3023 – WEB PROGRAMMING 2

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS

LAB 7

SEMESTER II 2023/2024

Prepared for:

DR MOHAMAD NOR BIN HASSAN

Prepared by:

LUQMAN HAKIM BIN AZIZ

(S66292)

Week 7

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

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
			Fakhrul Adli Mohd Zaki

Table of Contents

Task 1: Perform Basic CRUD Process Using Java Servlet 5

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: Perform Basic CRUD Process Using Java Servlet

Objective:

Using Java Servlet to perform creating, retrieving, updating and deleting (CRUD) records from MySQL database.

Problem Description:

You are required to perform basic CRUD process using Java Servlet.

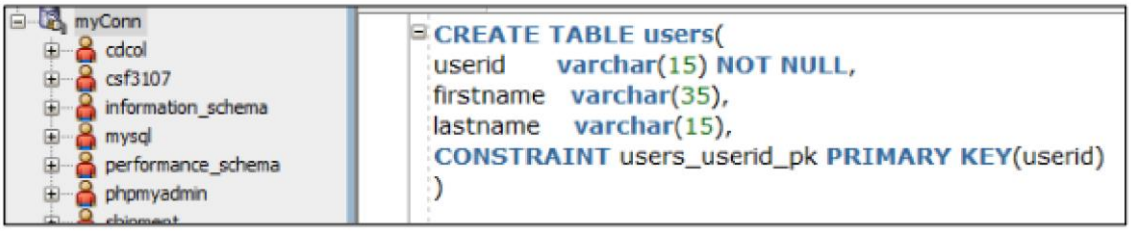
1. Create table users in CSF3203 database schema.
2. Create a three Java class that representing User (act as a JavaBeans to represent business object), DBConnection (to open and close database connection) and UserDao (act as a Data Access Object (DAO) to perform CRUD process).
3. Create UserController servlet to control the CRUD process.
4. Create index.jsp page as a main page.
5. Create listUser.jsp page to perform retrieving of a list of users.
6. Create user.jsp page to create a new record for user.
7. Create editUser.jsp page to update existing record for specific user.

Estimated time: 120 minutes

Step 1 - Create table users in CSF3107 database schema

1. Open XAMPP Control Panel.
2. Start Apache and MySQL module.

3. Open phpMyAdmin by clicking Admin button for MySQL module. 4. Select CSF3107 database schema.
5. Go to SQL tab.
6. Create table known as users.



7. Click *Go* button to execute the query.

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

1. Create new web application as CRUDServlet.
2. Create Java class User to represent the business object for user.
3. Name the package as com.model.
4. Write a coding for getter and setter for each instance variable.

```
1   package com.model;
2
3   /**
4    * @author mohamadnor
5    */
6   public class User {
7       private String userid;
8       private String firstName;
9       private String lastName;
10
11      public String getUserid() {
12          return userid;
13      }
14
15      public void setUserid(String userid) {
16          this.userid = userid;
17      }
18
19      public String getFirstName() {
20          return firstName;
21      }
```

```
22
23      public void setFirstName(String firstName) {
24          this.firstName = firstName;
25      }
26
27      public String getLastName() {
28          return lastName;
29      }
30
31      public void setLastName(String lastName) {
32          this.lastName = lastName;
33      }
34  }
```

5. Create Java class *DBConnection* to open and close the database.
6. Name the package as *com.util*
7. Write a coding for connecting and closing to database.

```
5   package com.util;
6
7   /**
8    *
9    * @author mohamadnor
10   */
11
12  import java.io.IOException;
13  import java.sql.Connection;
14  import java.sql.DriverManager;
15  import java.sql.SQLException;
16
17  public class DBConnection {
18      private static Connection myConnection=null;
19      private static String     myURL=myURL = "jdbc:mysql://localhost:3306/csf3203";
20
21      DBConnection(){
22      }
```

```
23
24      public static Connection getConnection() throws ClassNotFoundException
25      {
26          if (myConnection != null)
27          {
28              return myConnection;
29          }
30          else
31          try
32          {
33              //Establish and open MySQL database connection.....
34              Class.forName("com.mysql.jdbc.Driver");
35              myConnection = DriverManager.getConnection(myURL, "root", "admin");
36          }
37          catch (SQLException e)
38          {
39              e.printStackTrace();
40          }
41          return myConnection;
42      }
```

```
44      public void closeConnection() throws ClassNotFoundException
45      {
46          try
47          {
48              myConnection.close();
49          }
50          catch(SQLException e)
51          {
53              e.printStackTrace();
54          }
55      }
    }
```

8. Create Java class *UserDao* to perform CRUD process.
9. Name the package as *com.dao*.
10. Write codes to perform CRUD process.

```
5   package com.dao;
6
7   /**
8    *
9    * @author mohamadnor
10   */
11
13  import java.sql.Connection;
14  import java.sql.PreparedStatement;
15  import java.sql.ResultSet;
16  import java.sql.SQLException;
17  import java.sql.Statement;
18
19  import java.util.ArrayList;
20  import java.util.List;
21  import com.model.User;
22  import com.util.DBConnection;
```

```java
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 (?, ?, ? )");
            // Parameters start with 1
            preparedStatement.setString(1, user.getUserid());
            preparedStatement.setString(2, user.getFirstName());
            preparedStatement.setString(3, user.getLastName());
            preparedStatement.executeUpdate();

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

```java
    public void deleteUser(String userId) {
        try {
            PreparedStatement preparedStatement = connection
                    .prepareStatement("delete from users where userid=?");
            // Parameters start with 1
            preparedStatement.setString(1, userId);
            preparedStatement.executeUpdate();

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

```java
    public void updateUser(User user) {
        try {
            PreparedStatement preparedStatement = connection
                    .prepareStatement("update users set firstname=?, lastname=? " +
                            "where userid=?");
            // Parameters start with 1
            preparedStatement.setString(1, user.getFirstName());
            preparedStatement.setString(2, user.getLastName());
            preparedStatement.setString(3, user.getUserid());
            preparedStatement.executeUpdate();

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

```
75
76      public List<User> getAllUsers() {
77          List<User> users = new ArrayList<User>();
78          try {
79              Statement statement = connection.createStatement();
80              ResultSet rs = statement.executeQuery("select * from users");
81              while (rs.next()) {
82                  User user = new User();
83                  user.setUserid(rs.getString("userid"));
84                  user.setFirstName(rs.getString("firstname"));
85                  user.setLastName(rs.getString("lastname"));
86                  users.add(user);
87              }
88          } catch (SQLException e) {
89              e.printStackTrace();
90          }
91
92          return users;
93      }
```

```
95
96      public User getUserById(String userId) {
97          User user = new User();
98          try {
99              PreparedStatement preparedStatement = connection.prepareStatement(
100                     "select * from users where userid=?");
101             preparedStatement.setString(1,userId);
102             ResultSet rs = preparedStatement.executeQuery();
103
104             while (rs.next()) {
105                 user.setUserid(rs.getString("userid"));
106                 user.setFirstName(rs.getString("firstname"));
107                 user.setLastName(rs.getString("lastname"));
108             }
109         } catch (SQLException e) {
110             e.printStackTrace();
111         }
112
113         return user;
114     }
115 }
```

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

1. Create a Java servlet known as *UserController*.
2. Name the package as *com.controller*.
3. Import the related API and package.

```
5   package com.controller;
6
7   import java.io.IOException;
8   import java.io.PrintWriter;
9   import java.text.ParseException;
10
11  import javax.servlet.RequestDispatcher;
12  import javax.servlet.ServletException;
13  import javax.servlet.http.HttpServlet;
14  import javax.servlet.http.HttpServletRequest;
15  import javax.servlet.http.HttpServletResponse;
16
17  import com.dao.UserDao;
18  import com.model.User;
```

4. Remove *processRequest()* method.
5. Define the static instance variables and the contructor.

```
24  public class UserController extends HttpServlet {
25
26      private static String INSERT = "/user.jsp";
27      private static String EDIT = "/editUser.jsp";
28      private static String LIST_USER = "/listUser.jsp";
29      private UserDao dao;
30
31      public UserController() throws ClassNotFoundException {
32          super();
33          dao = new UserDao();
34      }
```

6. Write a code for *doGet()* method in order to determine the respective CRUD process and redirect to related page request.

```
@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 if (action.equalsIgnoreCase("insert")) {
       forward = INSERT;
    }

    RequestDispatcher view = request.getRequestDispatcher(forward);
    view.forward(request, response);
}
```

7. Write a code for doPost() method in order to perform creating or updating the record and finally, redirect to related page request.

```
67      @Override
68      protected void doPost(HttpServletRequest request, HttpServletResponse response)
69              throws ServletException, IOException {
70
71          String action = request.getParameter("action");
72
73          User user = new User();
74          user.setUserid(request.getParameter("userid"));
75          user.setFirstName(request.getParameter("firstName"));
76          user.setLastName(request.getParameter("lastName"));
77
78         if( action.equalsIgnoreCase("edit") )
79         {
80             dao.updateUser(user);
81         }
82         else
83         {
84            dao.addUser(user);
85         }
86
87         RequestDispatcher view = request.getRequestDispatcher(LIST_USER);
88         request.setAttribute("users", dao.getAllUsers());
89         view.forward(request, response);
90     }
   }
```

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

1. Create jsp page and key-in filename as *index.jsp*.
2. Write and HTML markup and JSP action tag to forward the page to UserController servlet with URL parameter as *action=listUser*.

```
7   <%@page contentType="text/html" pageEncoding="UTF-8"%>
8   <!DOCTYPE html>
9   <html>
10      <head>
11          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12          <title>Sample Perform CRUD Using Java Servlet</title>
13      </head>
14      <body>
15              <h1>Sample Perform CRUD Using Java Servlet!</h1>
16              <jsp:forward page="/UserController?action=listUser" />
17      </body>
18  </html>
19
```

3. Compile the file

- Create

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

1. Create jsp page and key-in filename as listUser.jsp.
2. Add standard.jar and jstl.jar in library project folder.
3. Add the taglib directive to listUser.jsp.

```
4       Author     : mohamadnor
5   --%>
6
7   <%@page contentType="text/html" pageEncoding="UTF-8"%>
8   <%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
9   <%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
```

4. Write and HTML markup and JSTL syntax to display the records.

```
11  <!DOCTYPE html>
12  <html>
13      <head>
14          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
15          <title>User Lists</title>
16      </head>
17      <body>
            <p>List of Users..!</p>
            <table border=1>
20          <thead>
21              <tr>
22                  <th>User Id</th>
23                  <th>First Name</th>
24                  <th>Last Name</th>
25                  <th colspan=2>Action</th>
26              </tr>
27          </thead>
28          <tbody>
```

```
30              <c:forEach items="${users}" var="user">
31                  <tr>
32                      <td><c:out value="${user.userid}" /></td>
33                      <td><c:out value="${user.firstName}" /></td>
34                      <td><c:out value="${user.lastName}" /></td>
35                      <td><a href="UserController?action=edit&userId=<c:out value="${user.userid}"/>">Update</a></td>
36                      <td><a href="UserController?action=delete&userId=<c:out value="${user.userid}"/>">Delete</a></td>
37                  </tr>
38              </c:forEach>
39          </tbody>
40      </table>
41      <p><a href="UserController?action=insert">Add User</a></p>
42      </body>
43  </html>
```

5. Compile the page

- Create

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

1. Create jsp page and key-in file name as user.jsp.
2. Create HTML markup and call UserController servlet from this page.

```
<%@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>
        <br/>
        <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" value="" size="25" required /></td>
                    </tr>
                    <tr>
                        <td>First Name :</td>
                        <td><input type="text" name="firstName" value="" size="40" /></td>
                    </tr>
```

```
                    <tr>
                        <td>Last Name :</td>
                        <td><input type="text" name="lastName" value="" size="40" /></td>
                    </tr>
                    <tr>
                        <td><input type="hidden" name="action" value="insert" /></td>
                    </tr>
                    <tr>
                        <td>
                            <input type="submit" value="Submit" name="submit" />
                            <input type="reset" value="Cancel" name="cancel" />
                        </td>
                    </tr>
                </tbody>
            </table>
        </form>
    </body>
</html>
```

3. Compile the file.

- Create

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

1. Create jsp page and key-in file name as editUser.jsp.
2. Add the taglib directive to editUser.jsp.

```
7   <%@page contentType="text/html" pageEncoding="UTF-8"%>
8   <%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
9   <%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
10
```

3. Write and HTML markup and JSTL syntax to display the records.

- Create

```
<!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" value="Submit" name="submit" />
                </td>
            </tr>
           </tbody>
           </table>
        </form>
    </body>
</html>
```

4. Compile the file.

Running the program and perform CRUD process

1. Run index.jsp page.
2. Click Add User button to create new record.
3. Click hyperlink Update in order to update an existing record.

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

Coding 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;
   }
}
```

Coding 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;

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 (?,
?, ?)");

         //parameters start with 1
         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<User>();
    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();

            while (rs.next()){
                user.setUserid(rs.getString("userid"));
                user.setFirstName(rs.getString("firstname"));
                user.setLastName(rs.getString("lastname"));
            }
        }
        catch (SQLException e){
            e.printStackTrace();
        }
        return user;
    }
}
```

Coding DBConnection.java

```
package com.util;

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

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

    DBConnection(){
    }

    public static Connection getConnection() throws ClassNotFoundException{
        if (myConnection != null){
            return myConnection;
        }
        else
            try{
                //establish and open mysql database connection
```

```
            Class.forName("com.mysql.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 UserController.java

```
package com.controller;

import java.io.IOException;
import java.io.PrintWriter;
import java.text.ParsePosition;

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;
/**
 *
 * @author User
 */
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");
            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 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")){
        dao.updateUser(user);
    }
    else{
        dao.addUser(user);
    }

    RequestDispatcher view = request.getRequestDispatcher(LIST_USER);
    request.setAttribute("users", dao.getAllUsers());
    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>
```

```
}
```

Coding index.jsp

```
<%--
   Document   : index
   Created on : 15 Jun 2024, 3:16:40 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>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>
```

Coding listUser.jsp

```
<%--
   Document   : listUser
   Created on : 15 Jun 2024, 3:17:18 am
   Author     : Luqman Hakim
--%>

<%@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>
```

Coding user.jsp

```
<%--
    Document   : user
    Created on : 15 Jun 2024, 3:17:52 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>New Record</title>
    </head>
    <body>
```

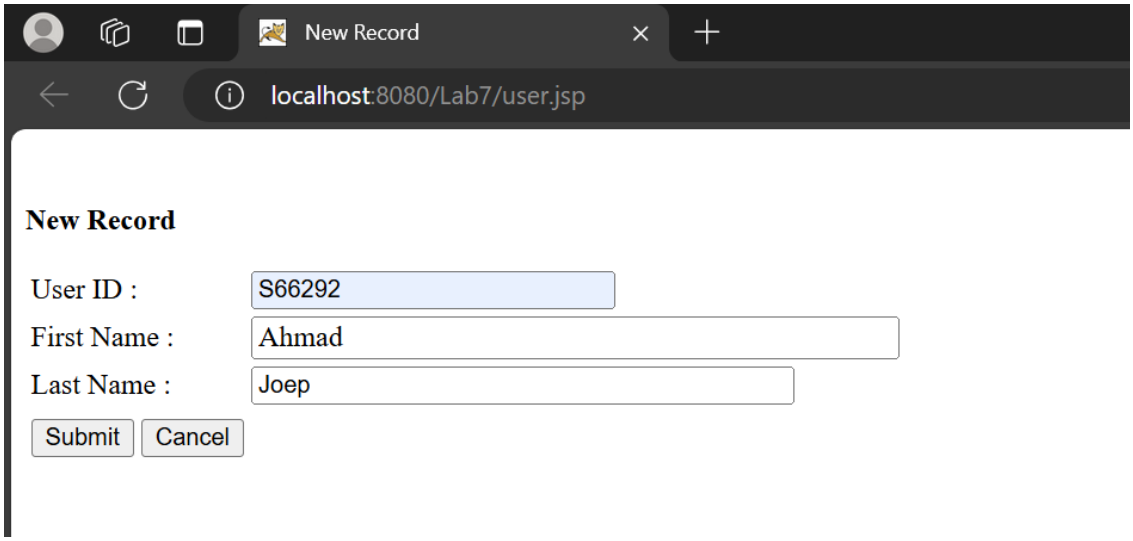
```
      <br/>
      <p>
        <b>New Record</b>
      </p>
      <form name="frmAddUser" action="UserController" method="POST">
        <table border="0">
          <tbody>
            <tr>
              <td>User ID :</td>
              <td><input type="text" name="userid" value="" size="25"
required=""></td>
            </tr>
            <tr>
              <td>First Name :</td>
              <td><input type="text" name="firstName" value="" size="40" ></td>
            </tr>
            <tr>
              <td>Last Name :</td>
              <td><input type="text" name="lastName" value="" size="40" ></td>
            </tr>
            <tr>
              <td><input type="hidden" name="action" value="insert" ></td>
            </tr>
            <tr>
              <td>
                <input type="submit" value="Submit" name="submit">
                <input type="reset" value="Cancel" name="cancel" >
              </td>
            </tr>
          </tbody>
        </table>
      </form>
    </body>
</html>
```

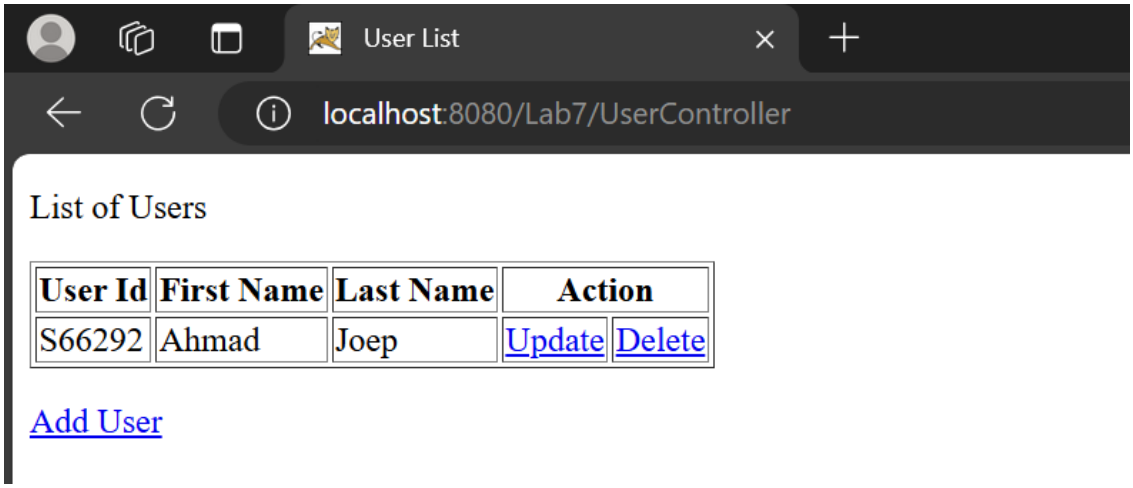
Coding editUser.jsp

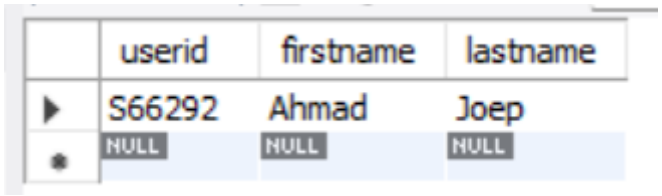
```
<%--
    Document   : editUser
    Created on : 15 Jun 2024, 3:15:55 am
    Author     : Luqman Hakim
--%>

<%@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" value="Submit" name="submit" >
            </td>
          </tr>
        </tbody>
      </table>
    </form>
  </body>
</html>
```

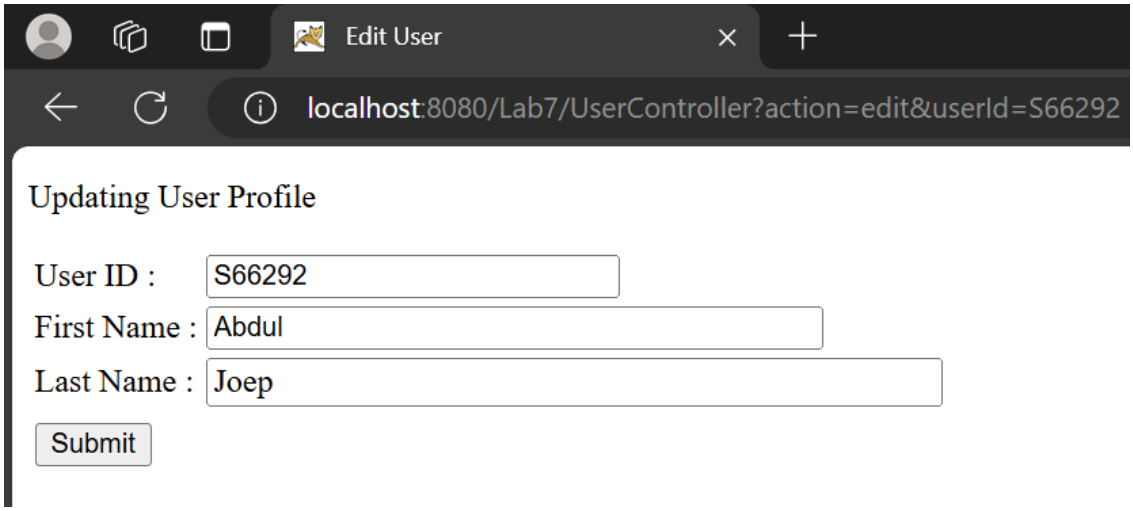
Output Add User

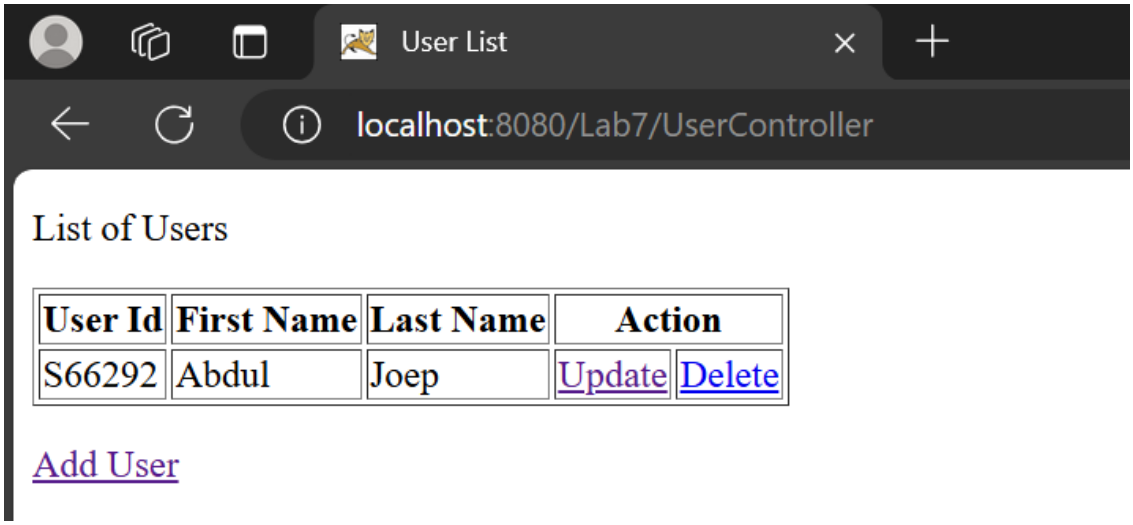


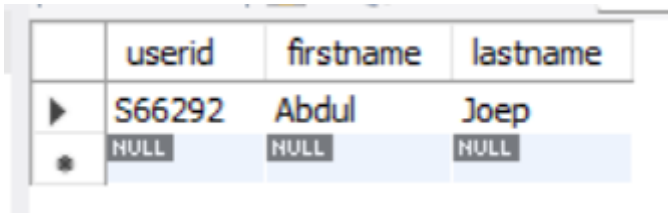




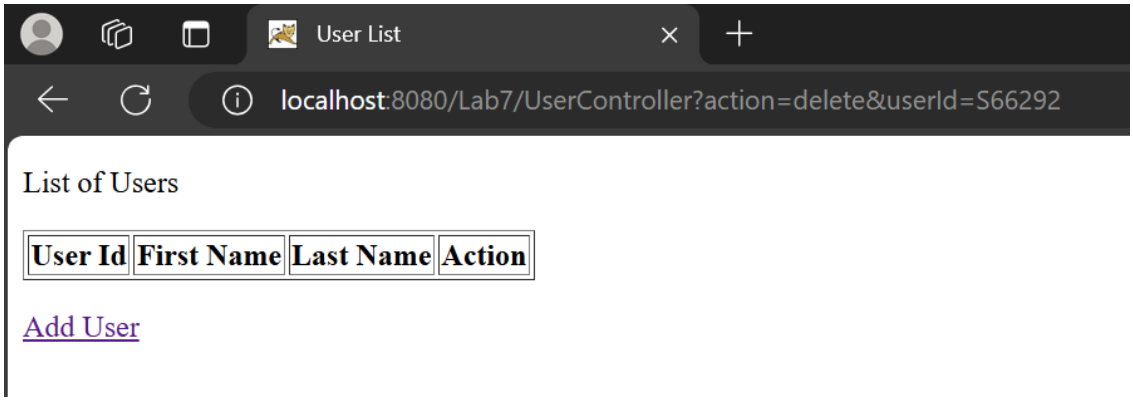
Output update

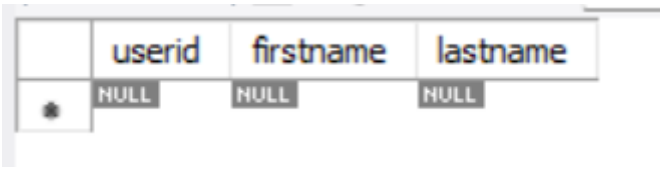






Output delete





Reflection:

1. Why we use servlet for Java Web Application?

Servlets are used in Java web applications to handle requests and responses between the web server and client, enabling dynamic content generation and interaction. They provide a robust, platform-independent method for building scalable and efficient web applications by leveraging Java's capabilities. Servlets also integrate seamlessly with other Java technologies, facilitating secure and maintainable web development.

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.

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*)

Coding registerProfile.html

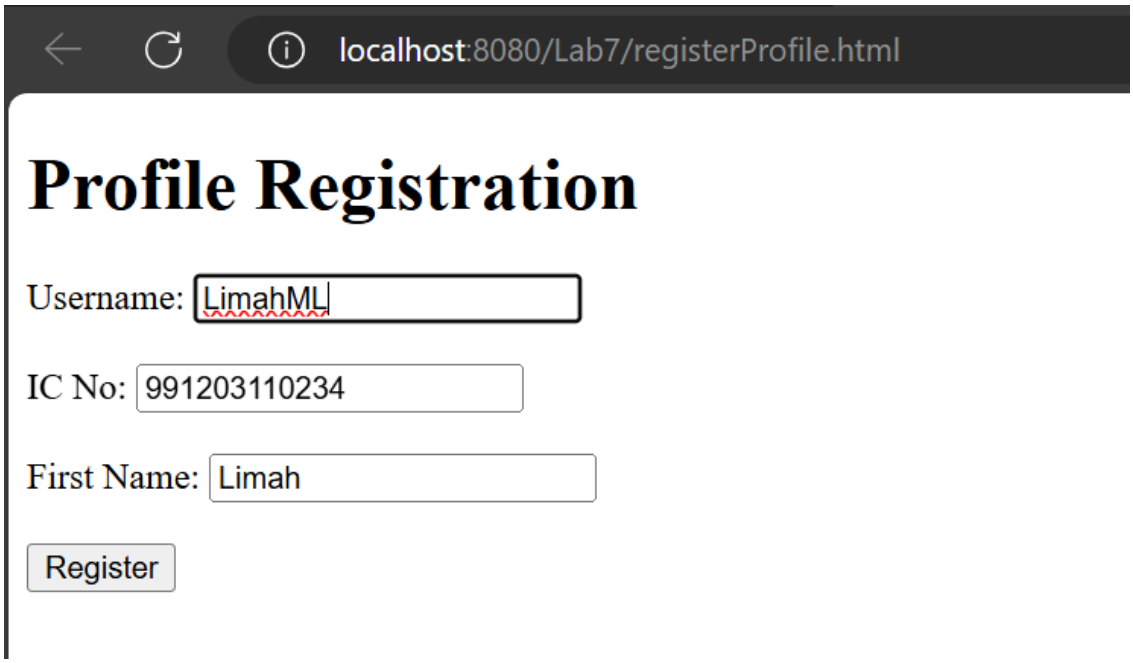
```
<!DOCTYPE html>
<html>
<head>
  <title>Profile Registration</title>
</head>
<body>
  <h1>Profile Registration</h1>
  <form action="profileServlet" method="post">
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" required>
    <br><br>
    <label for="icno">IC No:</label>
    <input type="text" id="icno" name="icno" required>
    <br><br>
    <label for="firstname">First Name:</label>
    <input type="text" id="firstname" name="firstname" required>
    <br><br>
    <button type="submit">Register</button>
  </form>
</body>
</html>
```

Coding profileServlet.java

```
package com.controller;

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;

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

   @Override
   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 pstmt = null;

      try {
         Class.forName("com.mysql.cj.jdbc.Driver");
         conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/CF3107",
"root", "password");
         String sql = "INSERT INTO userprofile (username, icno, firstname) VALUES (?, ?,
?)";
         pstmt = conn.prepareStatement(sql);
         pstmt.setString(1, username);
         pstmt.setString(2, icno);
         pstmt.setString(3, firstname);

         int result = pstmt.executeUpdate();

         response.setContentType("text/html");
         PrintWriter out = response.getWriter();
         if (result > 0) {
            out.println("<h1>Profile registered successfully!</h1>");
            out.println("<p>Username: " + username + "</p>");
```

```
            out.println("<p>IC No: " + icno + "</p>");
            out.println("<p>First Name: " + firstname + "</p>");
        } else {
            out.println("<h1>Profile registration failed!</h1>");
        }

    } catch (ClassNotFoundException | SQLException e) {
        e.printStackTrace();
    } finally {
        try {
            if (pstmt != null) pstmt.close();
            if (conn != null) conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
  }
}
```

Output Q1



Coding studentRegister.jsp

```
<%--
    Document   : studentRegister
    Created on : 15 Jun 2024, 1:38:49 pm
```

```
   Author     : Luqman Hakim
--%>

<%@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 Luqman Hakim
</footer>
</html>
```

Coding confirmRegister.jsp

```
<%--
   Document   : confirmRegister
   Created on : 15 Jun 2024, 1:39:39 pm
   Author     : Luqman Hakim
--%>

<%@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>
```

Coding notificationRegister.jsp

```
<%--
    Document   : notificationRegister
    Created on : 15 Jun 2024, 1:40:19 pm
    Author     : Luqman Hakim
--%>

<%@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 Luqman Hakim
</footer>
</html>
```

Output Q2