**EXPERIMENT -01:**

**Aim: Write a JDBC application to implement DDL and DML commands**

**Execution Steps:**

**1.** Open **NetBeans** and follow these steps: **File > New Project > Java > Java Application > Project Name > Finish**.

**2.** Right-click on **Source Packages**, select **New Java File** and name it ‘JDBC’. Copy and paste the following source code into the file.

**3.** To connect your database:

* First, install Apache Derby and ensure it's in your Java folder.
* In the **Projects** section, click on **Library** and add the library named **Java DB**.
* Check the **Properties** to confirm the correct paths for both **Java DB** and the database location.

**4.** Go to **Services** and find **Databases** to locate **Java DB**. Start the server and create a database by specifying the **username**, **password**, and **database name**.

**5.** Now, click on your database and select **Connect**.

**6.** Go back to the **Projects** section and do the following:

* Select **Clean and Build**.
* Run the project and view the flow in the **Output** section.

**7.** To view the data:

* Navigate to your database, expand the tables section until you find your table name.
* Right-click on the table and select **View Data** to display your table’s data.

**Source Code:**

import java.sql.\*;

import java.util.Scanner;

public class JDBCEX {

public static void main(String[] args) {

try {

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/Y22ACM475","sharon","sharon");

Statement stmt = con.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_UPDATABLE);

Scanner sc = new Scanner(System.in);

ResultSet rs;

int ch, n, rno, s1, s2, s3, s4, s5, s6, tot;

String q, reg, sname, g;

while (true) {

System.out.println("-----MENU-----");

System.out.println("0.exit");

System.out.println("1.create table");

System.out.println("2.insert data");

System.out.println("3.adding new columns");

System.out.println("4.updating new columns");

System.out.println("5.Display data(before updating)");

System.out.println("6.Display data(after updating)");

System.out.println("7.deleting data");

System.out.println("8.dropping data");

System.out.println("Enter your choice:");

ch = sc.nextInt();

switch (ch) {

case 0:

con.close();

System.exit(0);

case 1:

try {

q = "create table student\_marks1 (rno integer, rgd varchar(10), sname varchar(20), s1 integer, s2 integer, s3 integer, s4 integer, s5 integer, s6 integer)";

stmt.executeUpdate(q);

System.out.println("Table created successfully");

} catch (SQLException e) {

System.out.println("Table already exists");

}

break;

case 2:

System.out.println("Enter number of students:");

n = sc.nextInt();

for (int i = 1; i <= n; i++) {

System.out.println("Enter student " + i + " details:");

System.out.println("Rno:");

rno = sc.nextInt();

System.out.println("Regdno:");

reg = sc.next();

System.out.println("Sname:");

sname = sc.next();

System.out.println("s1:");

s1 = sc.nextInt();

System.out.println("s2:");

s2 = sc.nextInt();

System.out.println("s3:");

s3 = sc.nextInt();

System.out.println("s4:");

s4 = sc.nextInt();

System.out.println("s5:");

s5 = sc.nextInt();

System.out.println("s6:");

s6 = sc.nextInt();

q = "insert into student\_marks1 values(" + rno + ",'" + reg + "','" + sname + "'," + s1 + "," + s2 + "," + s3 + "," + s4 + "," + s5 + "," + s6 + ")";

stmt.executeUpdate(q);

}

System.out.println(n + " records are inserted");

break;

case 3:

try {

q = "alter table student\_marks1 add column tot integer";

stmt.executeUpdate(q);

q = "alter table student\_marks1 add column grade varchar(20)";

stmt.executeUpdate(q);

System.out.println("New columns are added");

} catch (SQLException e) {

System.out.println("Already exists");

}

break;

case 4:

q = "select \* from student\_marks1";

rs = stmt.executeQuery(q);

while (rs.next()) {

tot = rs.getInt(4) + rs.getInt(5) + rs.getInt(6) + rs.getInt(7) + rs.getInt(8) + rs.getInt(9);

if (tot >= 560) g = "A";

else if (tot >= 500) g = "B";

else if (tot >= 430) g = "C";

else if (tot >= 370) g = "D";

else if (tot >= 300) g = "E";

else g = "fail";

rs.updateInt(10, tot);

rs.updateString(11, g);

rs.updateRow();

}

System.out.println("New Columns are updated");

break;

case 5:

q = "select \* from student\_marks1";

rs = stmt.executeQuery(q);

System.out.println("The student details are:");

while (rs.next()) {

System.out.println("Rno:" + rs.getString(1));

System.out.println("Regdno:" + rs.getString(2));

System.out.println("sname:" + rs.getString(3));

System.out.println("s1:" + rs.getString(4));

System.out.println("s2:" + rs.getString(5));

System.out.println("s3:" + rs.getString(6));

System.out.println("s4:" + rs.getString(7));

System.out.println("s5:" + rs.getString(8));

System.out.println("s6:" + rs.getString(9));

}

break;

case 6:

q = "select \* from student\_marks1";

rs = stmt.executeQuery(q); // Re-fetch the ResultSet

System.out.println("The student details are:");

while (rs.next()) {

System.out.println("/nRno:" + rs.getString(1));

System.out.println("Regdno:" + rs.getString(2));

System.out.println("sname:" + rs.getString(3));

System.out.println("s1:" + rs.getString(4));

System.out.println("s2:" + rs.getString(5));

System.out.println("s3:" + rs.getString(6));

System.out.println("s4:" + rs.getString(7));

System.out.println("s5:" + rs.getString(8));

System.out.println("s6:" + rs.getString(9));

System.out.println("Total:" + rs.getString(10));

System.out.println("Grade:" + rs.getString(11));

}

break;

case 7:

q = "delete from student\_marks1";

int x = stmt.executeUpdate(q);

if (x != 0) System.out.println("Table data deleted");

else System.out.println("No data to delete");

break;

case 8:

try {

q = "drop table student\_marks1";

stmt.executeUpdate(q);

System.out.println("Table deleted successfully");

} catch (SQLException e) {

System.out.println("Table does not exist");

}

break;

default:

System.out.println("Invalid choice");

break;

}

}

} catch (ClassNotFoundException | SQLException e) {

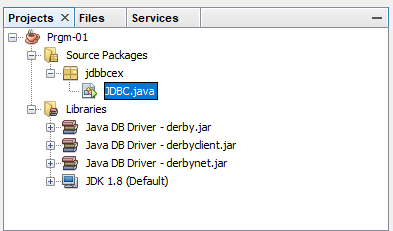
e.printStackTrace();

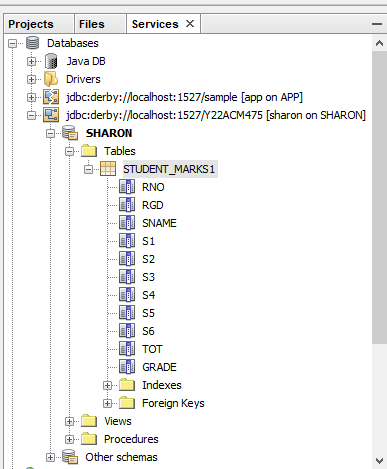
}

}

}

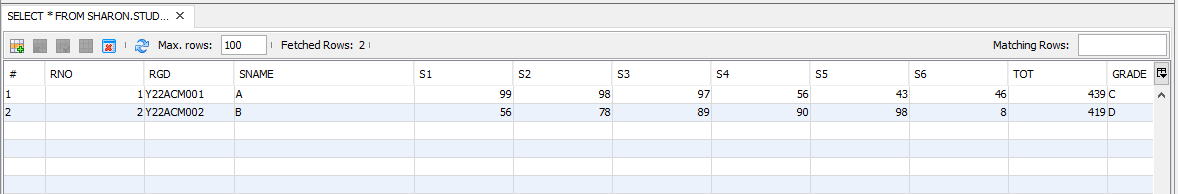
**OUTPUT:**





|  |  |
| --- | --- |
| -----MENU-----  0.exit  1.create table  2.insert data  3.adding new columns  4.updating new columns  5.Display data(before updating)  6.Display data(after updating)  7.deleting data  8.dropping data  Enter your choice:  1  Table Created Succesfully  -----MENU-----  0.exit  1.create table  2.insert data  3.adding new columns  4.updating new columns  5.Display data(before updating)  6.Display data(after updating)  7.deleting data  8.dropping data  Enter your choice:  2  Enter number of students:  2  Enter student 1 details:  Rno:  1  Regdno:  Y22ACM001  Sname:  A  s1:  99  s2:  98  s3:  97  s4:  56  s5:  43  s6:  46  Enter student 2 details:  Rno:  2  5.Display data(before updating)  6.Display data(after updating)  7.deleting data  8.dropping data  Enter your choice:  5  The student details are:  Rno:1  Regdno:Y22ACM001  sname:A  s1:90  s2:89  s3:67  s4:54  s5:53  s6:45  Rno:2  Regdno:Y22ACM002  sname:B  s1:34  s2:5  s3:67  s4:87  s5:60  s6:99  Rno:1  Regdno:Y22ACM001  sname:A  s1:99  s2:98  s3:97  s4:56  s5:43  s6:46  Rno:2  Regdno:Y22ACM002  sname:B  s1:56  s2:78  s3:89  s4:90  s5:98  s6:8  -----MENU-----  0.exit  1.create table  2.insert data  3.adding new columns  4.updating new columns  5.Display data(before updating)  -----MENU-----  0.exit  1.create table  2.insert data  3.adding new columns  4.updating new columns  5.Display data(before updating)  6.Display data(after updating)  7.deleting data  8.dropping data  Enter your choice:  7 | Regdno:  Y22ACM002  Sname:  B  s1:  56  s2:  78  s3:  89  s4:  90  s5:  98  s6:  8  2 records are inserted  -----MENU-----  0.exit  1.create table  2.insert data  3.adding new columns  4.updating new columns  5.Display data(before updating)  6.Display data(after updating)  7.deleting data  8.dropping data  Enter your choice:  3  New columns are added  -----MENU-----  0.exit  1.create table  2.insert data  3.adding new columns  4.updating new columns  5.Display data(before updating)  6.Display data(after updating)  7.deleting data  8.dropping data  Enter your choice:  4  New Columns are updated  -----MENU-----  0.exit  1.create table  2.insert data  3.adding new columns  4.updating new columns  6.Display data(after updating)  7.deleting data  8.dropping data  Enter your choice:  6  The student details are:  /nRno:1  Regdno:Y22ACM001  sname:A  s1:90  s2:89  s3:67  s4:54  s5:53  s6:45  Total:398  Grade:D  /nRno:2  Regdno:Y22ACM002  sname:B  s1:34  s2:5  s3:67  s4:87  s5:60  s6:99  Total:352  Grade:E  /nRno:1  Regdno:Y22ACM001  sname:A  s1:99  s2:98  s3:97  s4:56  s5:43  s6:46  Total:439  Grade:C  /nRno:2  Regdno:Y22ACM002  sname:B  s1:56  s2:78  s3:89  s4:90  s5:98  s6:8  Total:419  Grade:D  Table data deleted  -----MENU-----  0.exit  1.create table  2.insert data  3.adding new columns  4.updating new columns  5.Display data(before updating)  6.Display data(after updating)  7.deleting data  8.dropping data  Enter your choice: 8  Table deleted successfully |

**View Data:**



**EXPERIMENT-02:**

**Aim:**[**Write an application to demonstrate HTTP Servlets.**](#A2)

**Execution Steps:**

### **Create a New Project**

* + Open NetBeans and create a new project by navigating to:  
    **File > New Project > Java Web > Web Application.**  
    Provide the project name (e.g., Servlet Program), click **Next**, and then click **Finish.**

### **Creating and Configuring Files**

* + **index.html**:Right-click on the project, go to **New > HTML File**, and name it **index.html.**Add the required code to this file and save it.

**For GET Request:**

* + **New Servlet for GET Request**: Right-click on the project, go to **New > Servlet**, and name it **Given in Code.** Click **Next**, then **Finish.** Add the required code to this servlet and save it.
  + **Run the GET Request**: Right-click on **index.html** and select **Run.**
  + **Output for GET Request**: HTTP Error.

**For POST Request:**

* + **New Servlet for POST Request**: Follow the same process as above to create a new servlet file named **Given in Code** for the POST request. Insert the required servlet code and save it.
  + **Run the POST Request**: Right-click on **index.html** and select **Run.**
  + **Output for POST Request**: Data should be displayed.

### **Building and Running the Project**

* + **Build the Project**: Right-click on the project and select **Clean and Build.**
  + **Run the Project**: Right-click on **index.html** or the servlet file (e.g., for GET or POST) and select **Run.**

**Source Code:**

**Index.html—Post Method**

<html>

<head>

<title>POST METHOD</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

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

Username: <input type="text" name="username"/> <br/>

Password: <input type="password" name="password"/> <br/>

<input type="submit" value="Login" />

</form>

</body>

</html>

**GetMethod.html:**

<!DOCTYPE html>

<html>

<head>

<title>GET METHOD</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<form action="LoginServlet>" method="GET">

<center>

<h1>This is servlet program</h1>

<br> Username: <input type="text" name="username"/> <br/>

<br> Password: <input type="password" name="yourPassword"/> <br/>

<input type="submit" value="NEXT" />

</center>

</form>

</body>

</html>

**LoginServlet.java**

package net.codejava.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/LoginServlet")

public class LoginServlet extends HttpServlet {

protected void doPost(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException {

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

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

System.out.println("username: " + username);

System.out.println("password: " + password);

PrintWriter writer = response.getWriter();

String htmlRespone = "<html>";

htmlRespone += "<h2>Your username is: " + username + "<br/>";

htmlRespone += "Your password is: " + password + "</h2>";

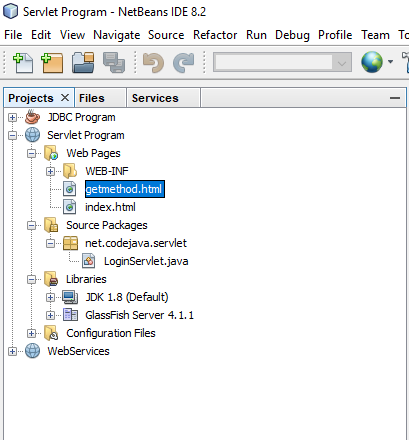
htmlRespone += "</html>";

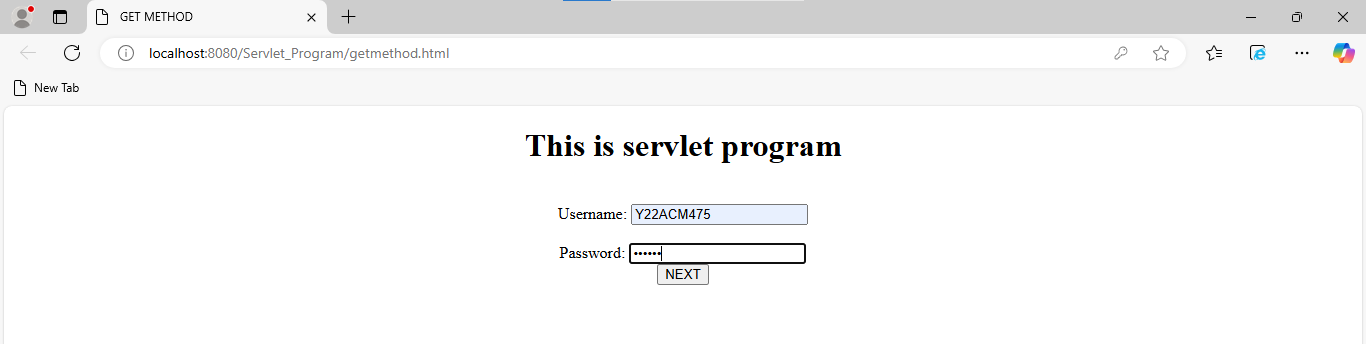
writer.println(htmlRespone);

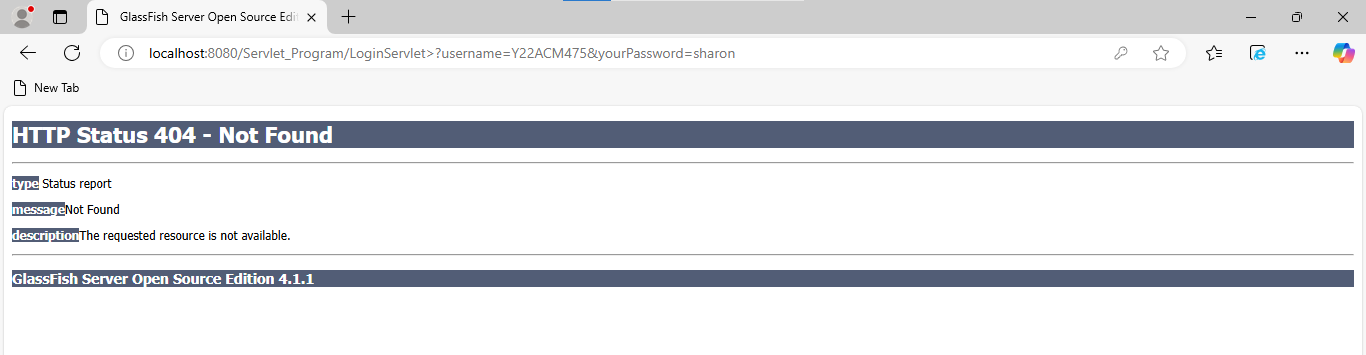
}

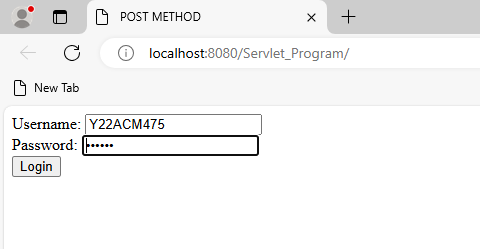
}

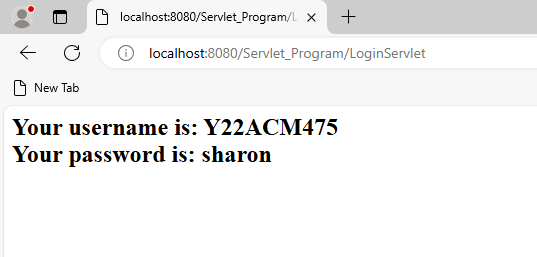
**Output:**

****

****

****

****

****

**EXPERIMENT-03:**

**Aim:**[**Write an application to demonstrate Cookie & Sessions**](#A3)**.**

**Execution Steps:**

1. Open **NetBeans** and create a new project by navigating to:**File > New Project > Java Web > Web Application.**Provide the project name (e.g., CookiesSessions), click **Next**, and then **Finish**.
2. Creating and Configuring Files
   * index.html: Right-click on the project, go to **New > HTML File**, and name it index.html Add the required code to this file and save it.
   * NewServlet: Right-click on the project, go to **New > Servlet**, and name it NewServlet. Check **Add Information to Deployment Descriptor** and click Finish.Add the necessary code to this servlet and save it.
   * Additional Servlets: Follow the same process to create the following servlet files:  
     NewServlet1.java, Brand.java, CheckedOut.java, PayMoney.java, PaymentPage.java, NewServlet2.java.
   * Project Configuration: Open the web.xml file:Navigate to **Configuration**, double-click on the file, and make any necessary changes.
3. Database Connection
   * Connect to the database: Navigate to **Services > Databases > JavaDB > Start Server.** Use the following credentials to establish a connection: **Database Name**: Y22ACM475, **Username:** SHARON, **Password**: sharon
4. Table Creation
   * UserData Table: Right-click on **Tables** and select **Create Table**. Provide the table name as UserData and add the columns: usename (VARCHAR, Not Null), password (VARCHAR, Not Null), amount (NUMERIC), ID (NUMERIC, Unique).
   * Cookie Table: Create another table named Cookie with the columns:  
     Name (VARCHAR, Not Null), Model (VARCHAR, Not Null), Version (VARCHAR), Amount (NUMERIC, Not Null), ID (NUMERIC, Unique).
   * Insert Data: Right-click on each table (e.g., UserData or Cookie) and select **View Data**. Insert the sample data directly into the table rows displayed in the table view.
5. Building and Running the Project
   * Right-click on the project and select **Clean and Build** and then **Run**

**Source Code:**

**index.html**

<!DOCTYPE html>

<html>

<head>

<title>Shopping</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<div>

<center>

<form method="get" action="NewServlet">

Username:<input type="text" name="username"/><br><br>

Password:<input type="password" name="password"/><br><br>

<input type="submit" name="submit"/>

</form>

</center>

</div>

</body>

</html>

**NewServlet.java**

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletContext;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

public class NewServlet extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException, ClassNotFoundException, SQLException {

response.setContentType("text/html;charset=UTF-8");

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/y22acm4 75","sharon","sharon");

Statement st=con.createStatement();

String a;

int r;

ResultSet res;

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

String pass=request.getParameter("password");

res=st.executeQuery("select \* from userdata where usename='"+name+"' and password='"+pass+"'");

ServletContext sc=request.getServletContext();

RequestDispatcher rd;

HttpSession ses=request.getSession(true);

Integer count=(Integer)ses.getAttribute("count");

if(count==null)

count=0;

try (PrintWriter out = response.getWriter()) {

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet NewServlet2</title>");

out.println("</head>");

out.println("<body>");

if(res.next())

{

rd=sc.getRequestDispatcher("/NewServlet1");

rd.forward(request, response);

}

else

{

count=count+1;

ses.setAttribute("count", count);

if(count<3)

{

out.println("<h1>enter correct details</h1>");

rd=sc.getRequestDispatcher("/index.html");

rd.include(request, response);

}

else

{

out.println("<h1>No of Attempts execeded </h1>");

}

}

out.println("</body>");

out.println("</html>");

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

processRequest(request, response);

} catch (ClassNotFoundException ex) {

Logger.getLogger(NewServlet.class.getName()).log(Level.SEVERE, null, ex);

} catch (SQLException ex) {

Logger.getLogger(NewServlet.class.getName()).log(Level.SEVERE, null, ex);

}

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

processRequest(request, response);

} catch (ClassNotFoundException ex) {

Logger.getLogger(NewServlet.class.getName()).log(Level.SEVERE, null, ex);

} catch (SQLException ex) {

Logger.getLogger(NewServlet.class.getName()).log(Level.SEVERE, null, ex);

}

}

@Override

public String getServletInfo() {

return "Short description";

}

}

**NewServlet1.java**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class NewServlet1 extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet NewServlet1</title>");

out.println("</head>");

out.println("<body>");

out.println("<form method='get' action='brand'>");

out.println("<h1>Choose your options:</h1>");

out.println("<h4>Select watch brand you want</h4>");

out.println("<input type='checkbox' name='n' value='noise'/>Noise<br>");

out.println("<input type='checkbox' name='n' value='fire boltt'/>Fire Boltt<br>");

out.println("<input type='checkbox' name='n' value='ptron'/>pTron<br>");

out.println("<input type='checkbox' name='n' value='boat wave'/>boAT Wave<br>");

out.println("<input type='submit' value='Next'/>");

out.println("</body>");

out.println("</html>");

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}

}

**brand.java**

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class brand extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException, ClassNotFoundException, SQLException {

response.setContentType("text/html;charset=UTF-8");

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/y22acm475","sharon","sharon");

Statement st=con.createStatement();

ResultSet res;

String[] name=request.getParameterValues("n");

String n,m,v;

int amou,id;

try (PrintWriter out = response.getWriter()) {

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet brand</title>");

out.println("</head>");

out.println("<body>");

out.println("<h1>Select the models in it ....</h1>");

out.println("<form method='get' action='checkedout'>");

for(int i=0;i<name.length;i++)

{

res=st.executeQuery("select \* from cookie where name='"+name[i]+"'");

while(res.next())

{

n=res.getString(1);

m=res.getString(2);

v=res.getString(3);

amou=res.getInt(4);

id=res.getInt(5);

out.println("<input type='checkbox' name='brand' value='"+id+"'/>");

out.println("Name: "+n);

out.println("Model: "+m);

out.println("version: "+v);

out.println("Amount: "+amou);

out.println("Id: "+id);

out.println("<br>");

}

}

out.println("<input type='submit' value='checkout'/>");

out.println("</form></body>");

out.println("</html>");

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

processRequest(request, response);

} catch (ClassNotFoundException ex) {

Logger.getLogger(brand.class.getName()).log(Level.SEVERE, null, ex);

} catch (SQLException ex) {

Logger.getLogger(brand.class.getName()).log(Level.SEVERE, null, ex);

}

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

processRequest(request, response);

} catch (ClassNotFoundException ex) {

Logger.getLogger(brand.class.getName()).log(Level.SEVERE, null, ex);

} catch (SQLException ex) {

Logger.getLogger(brand.class.getName()).log(Level.SEVERE, null, ex);

}

}

@Override

public String getServletInfo() {

return "Short description";

}

}

**checkedout.java**

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.servlet.ServletException;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class checkedout extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException, ClassNotFoundException, SQLException {

response.setContentType("text/html;charset=UTF-8");

response.setContentType("text/html;charset=UTF-8");

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/y22acm475","sharon","sharon");

Statement st=con.createStatement();

ResultSet res;

Cookie c;

String[] name=request.getParameterValues("brand");

try (PrintWriter out = response.getWriter()) {

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet brand</title>");

out.println("</head>");

out.println("<body>");

out.println("<form method='get' action='pay\_money'>");

out.println("<h1>Payment hear...</h1>");

int id,amount=0;

int total=0;

for(int i=0;i<name.length;i++)

{

res=st.executeQuery("select \* from cookie where id="+name[i]+" ");

while(res.next())

{

//total=total+res.getInt(1);

id=res.getInt(5);

amount=res.getInt(4);

total=total+amount;

out.println("product ID :"+id+" Total Amount :"+amount);

c=new Cookie(String.valueOf(id),String.valueOf(amount));

response.addCookie(c);

out.println("<br>");

}

}

out.println("<h2>Total Amount :"+total+"</h2>");

c=new Cookie("total",String.valueOf(total));

out.println("<input type='submit' value='pay'/>");

out.println("</form>");

response.addCookie(c);

out.println("</body>");

out.println("</html>");

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

processRequest(request, response);

} catch (ClassNotFoundException ex) {

Logger.getLogger(checkedout.class.getName()).log(Level.SEVERE, null, ex);

} catch (SQLException ex) {

Logger.getLogger(checkedout.class.getName()).log(Level.SEVERE, null, ex);

}

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

processRequest(request, response);

} catch (ClassNotFoundException ex) {

Logger.getLogger(checkedout.class.getName()).log(Level.SEVERE, null, ex);

} catch (SQLException ex) {

Logger.getLogger(checkedout.class.getName()).log(Level.SEVERE, null, ex);

}

}

@Override

public String getServletInfo() {

return "Short description";

}

}

**pay\_money.java**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

public class pay\_money extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

HttpSession ses=request.getSession();

ses.setMaxInactiveInterval(60);

try (PrintWriter out = response.getWriter()) {

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet pay\_money</title>");

out.println("</head>");

out.println("<body>");

out.println("<form method='get' action='payment\_page'>");

out.println("<h3>pay the amount...</h3><br>");

out.println("Username :");

out.println("<input type='text' name='username'/><br><br>");

out.println("Id :");

out.println("<input type='text'name='id'/><br><br>");

out.println("<br>");

out.println("<input type='submit' value='payment hear.....'/>");

out.println("</form>");

out.println("</body>");

out.println("</html>");

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}

}

**payment\_page.java**

import java.io.IOException;

import java.io.PrintWriter;

import static java.lang.System.out;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletContext;

import javax.servlet.ServletException;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

public class payment\_page extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException, ClassNotFoundException, SQLException {

response.setContentType("text/html;charset=UTF-8");

Class.forName("org.apache.derby.jdbc.ClientDriver");

HttpSession ses=request.getSession(false);

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/y22acm475","sharon","sharon");

Statement st=con.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,ResultSet.CONCUR\_UPDATABLE);

ResultSet res;

ServletContext sc=request.getServletContext();

RequestDispatcher rd;

Cookie[] c=request.getCookies();

int amount=0;

int total=0;

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

res=st.executeQuery("select \* from userdata where usename='"+name+"'");

if(ses==null){

System.out.println("session expired");

}

if(res.next())

{

amount=res.getInt(3);

}

for(Cookie cookie:c)

{

if(cookie.getName().equals("total"))

{

total=Integer.parseInt(cookie.getValue());

}

}

if(total<=amount)

{

amount=amount-total;

res.updateInt("amount",amount);

res.updateRow();

rd=sc.getRequestDispatcher("/NewServlet2");

rd.include(request, response);

}

else

{

rd=sc.getRequestDispatcher("/NewServlet3");

rd.include(request, response);

}

try (PrintWriter out = response.getWriter()) {

/\* TODO output your page here. You may use following sample code. \*/

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet payment\_page</title>");

out.println("</head>");

out.println("<body>");

out.println("</body>");

out.println("</html>");

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

processRequest(request, response);

} catch (ClassNotFoundException ex) {

Logger.getLogger(payment\_page.class.getName()).log(Level.SEVERE, null, ex);

} catch (SQLException ex) {

Logger.getLogger(payment\_page.class.getName()).log(Level.SEVERE, null, ex);

}

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

processRequest(request, response);

} catch (ClassNotFoundException ex) {

Logger.getLogger(payment\_page.class.getName()).log(Level.SEVERE, null, ex);

} catch (SQLException ex) {

Logger.getLogger(payment\_page.class.getName()).log(Level.SEVERE, null, ex);

}

}

@Override

public String getServletInfo() {

return "Short description";

}

}

**NewServlet2.java**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletContext;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet(urlPatterns = {"/NewServlet2"})

public class NewServlet2 extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet NewServlet2</title>");

out.println("</head>");

out.println("<body>");

out.println("<h1>payment complete</h1>");

out.println("</body>");

out.println("</html>");

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}

}

**NewServlet3.java**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class NewServlet3 extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet NewServlet3</title>");

out.println("</head>");

out.println("<body>");

out.println("<h1>payment insufficent</h1>");

out.println("</body>");

out.println("</html>");

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

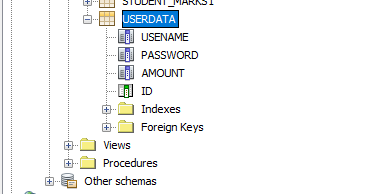
return "Short description";

}

}

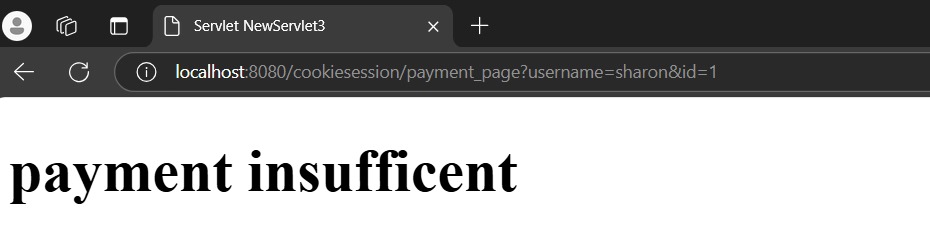
**OUTPUT:**

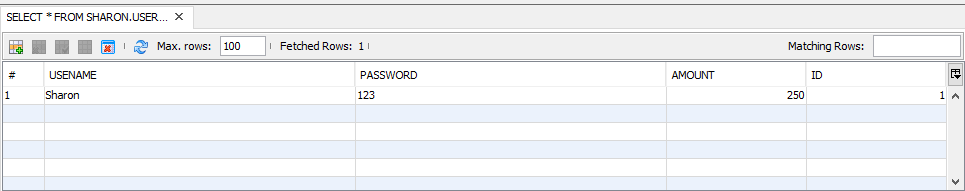
|  |  |
| --- | --- |
| C:\Users\hp\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\image (2).jpeg | C:\Users\hp\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\image (1).jpeg |

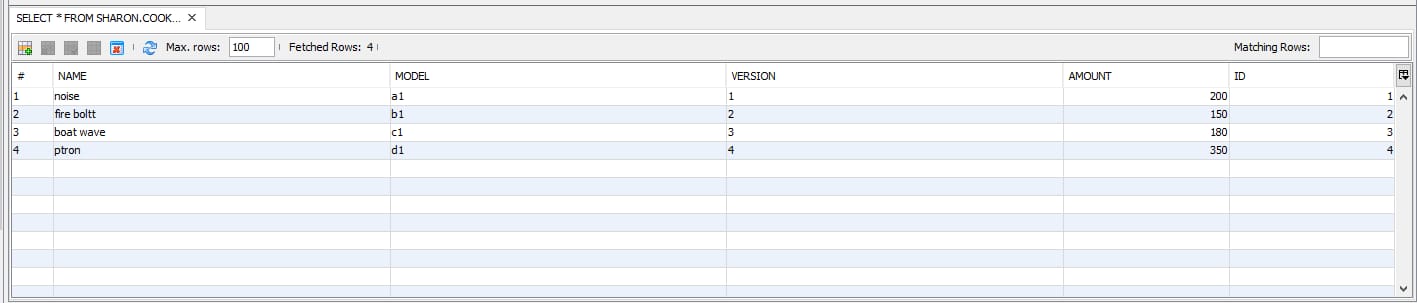


|  |  |
| --- | --- |
| C:\Users\hp\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\image (3).jpeg | C:\Users\hp\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\image (4).jpeg |
| C:\Users\hp\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\image (5).jpeg | C:\Users\hp\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\image (7).jpeg |
| C:\Users\hp\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\image (10).jpeg | C:\Users\hp\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\image (8).jpeg |
| C:\Users\hp\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\image (11).jpeg | C:\Users\hp\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\image (13).jpeg |
| C:\Users\hp\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\image (14).jpeg | C:\Users\hp\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\image (15).jpeg |

|  |  |
| --- | --- |
| C:\Users\hp\Downloads\WhatsApp Image 2024-12-08 at 19.17.03 (1).jpeg | C:\Users\hp\Downloads\WhatsApp Image 2024-12-08 at 19.17.03.jpeg |
| C:\Users\hp\Downloads\WhatsApp Image 2024-12-08 at 19.19.01.jpeg | C:\Users\hp\Downloads\WhatsApp Image 2024-12-08 at 19.19.01 (1).jpeg |
| C:\Users\hp\Downloads\WhatsApp Image 2024-12-08 at 19.20.44.jpeg | C:\Users\hp\Downloads\WhatsApp Image 2024-12-08 at 19.20.45.jpeg |
| C:\Users\hp\Downloads\WhatsApp Image 2024-12-08 at 19.22.07.jpeg | C:\Users\hp\Downloads\WhatsApp Image 2024-12-08 at 19.22.08.jpeg |







**EXPERIMENT-04:**

**Aim:** [**Write an application to integrate JSP & Servlets**](#A4)**.**

**Execution steps:**

**1.** Creating the Project

* Open NetBeans and create a new project by navigating to:**File > New Project > Java Web > Web Application**.Provide the project name (e.g., LoginRegister), click **Next**, and then **Finish**.

2. Necessary Files and Saving

* Delete the default **index.html** file that gets created automatically.
* Create index.jsp:

Right-click on the project, go to **New > JSP File**, and name it **index.jsp**.Add the required code into this file and save it.

* Create Login Servlet:

Right-click on the project, go to **New > Servlet**, and name it **Login.Servlet.Add Information to Deployment Descriptor** and click **Finish**.Insert the required code into this file and save it.

* Create Register Servlet:

Right-click on the project, go to **New > Servlet**, and name it **Register**.Add the required code and save it.

* Create Bean Class:

Right-click on the project, go to **New > Java Class**, and name it Bean.Place it in the package t and click **Finish**.Add the necessary code into this file and save it.

* Create details.jsp:

Right-click on the project, go to **New > JSP File**, and name it details.jsp.Insert the required code into the file and save it.

**3.** Database Connection:

* Connect to the database:  
  Navigate to **Services > Databases > Java DB > Start Server**.Use the credentials:**Username**: sharon,**Password**: sharon to establish the connection.

**4**. Table Creation:

* Expand the connected database in the **Services** tab to find the **Tables** section.Right-click on **Tables** and select **Create Table**.
* Enter the table name as Login and add the following columns: Name (VARCHAR, Size: 20, Not Null),Username (VARCHAR,Size: 20, Not Null),Password (VARCHAR, Size: 20, Not Null),Reg (VARCHAR, Size: 20, Not Null),Gender (VARCHAR, Size: 20, Not Null),Branch (VARCHAR, Size: 20, Not Null).Click **OK** to create the table.
* Right-click on the created table (Login) and select **View Data**.  
  Insert sample values by directly editing the rows displayed in the table view.

**6.** Clean, Build, and Run:

* **Clean and build the project**, then **run** it to display the output.

**Source Code:**

**insert.jsp**

<%--

Document : index

Created on : 12 Nov, 2024, 9:57:32 PM

Author : y22acm475

--%>

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

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

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

Username :<input type="text" name="uname"/><br><br>

Password :<input type="password" name="pass"/><br><br>

<input type="submit" value="Login"/>

</form>

<br>

<form action="register.jsp">

<input type="submit" value="Register"/>

</form>

</body>

</html>

**Login,java**

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class Login extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

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

String pa = request.getParameter("pass");

try (Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/y22acm475", "sharon", "sharon");

PreparedStatement pst = con.prepareStatement("SELECT \* FROM login WHERE username = ? AND password = ?")) {

pst.setString(1, name);

pst.setString(2, pa); // Replace with hashed password comparison in production

try (ResultSet res = pst.executeQuery()) {

try (PrintWriter out = response.getWriter()) {

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head><title>User Details</title>");

out.println("<style>");

out.println("table { width: 50%; border-collapse: collapse; }");

out.println("th, td { border: 1px solid black; padding: 10px; text-align: left; }");

out.println("th { background-color: #f2f2f2; }");

out.println("</style>");

out.println("</head>");

out.println("<body>");

out.println("<h1>User Details</h1>");

if (res.next()) {

out.println("<table>");

out.println("<tr><th>Name</th><td>" + res.getString("name") + "</td></tr>");

out.println("<tr><th>Username</th><td>" + res.getString("username") + "</td></tr>");

out.println("<tr><th>Password</th><td>Can't be displayed</td></tr>");

out.println("<tr><th>Registration Date</th><td>" + res.getString("regd") + "</td></tr>");

out.println("<tr><th>Gender</th><td>" + res.getString("gender") + "</td></tr>");

out.println("<tr><th>Branch</th><td>" + res.getString("branch") + "</td></tr>");

out.println("</table>");

} else {

out.println("<p>Invalid credentials. Please try again.</p>");

RequestDispatcher rd = request.getServletContext().getRequestDispatcher("/NewServlet");

rd.include(request, response);

}

out.println("</body>");

out.println("</html>");

}

}

} catch (SQLException ex) {

throw new ServletException("Database error", ex);

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}

}

**NewServlet.java**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class NewServlet extends HttpServlet

{

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet NewServlet</title>");

out.println("</head>");

out.println("<body>");

out.println("<form method='post' action='register.jsp'>");

out.println("<h1>You are the new user so Register Now</h1>");

out.println("<input type='submit' value='Register here'/>");

out.println("</form></body>");

out.println("</html>");

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}

}

**register.jsp**

<%--

Document : register

Created on : 12 Nov, 2024, 3:42:38 PM

Author : y22acm475

--%>

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

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

<style>

table, th, td

{

border:1px solid black;

}

</style>

</head>

<body>

<h2>Register Form</h2>

<table style="width: 30%">

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

<tr>

<th>NAME</th>

<td><input type="text" name="name" size="30%"/></td>

</tr>

<tr>

<th>USERNAME</th>

<td><input type="text" name="username" size="30%"/></td>

</tr>

<tr>

<th>PASSWORD</th>

<td><input type="password" name="password" size="30%"/></td>

</tr>

<tr>

<th>REGD</th>

<td><input type="text" name="regd" size="30%"/></td>

</tr>

<tr >

<th>GENDER</th>

<td size="30%"><input type="radio" name="gender" value="male"/>Male

<input type="radio" name="gender" value="female"/>Female

<input type="radio" name="gender" value="others"/>Others

</td>

</tr>

<tr>

<th>BRANCH</th>

<td size="30%">

<select id="branch" name="branch">

<option>Select Branch</option>

<option value="CSE">CSE</option>

<option value="AIML">ECE</option>

<option value="CBDS">EEE</option>

<option value="IT">IT</option>

</select>

</td>

</tr>

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

</form>

</table>

</body>

</html>

**bean.java**

package t;

public class bean {

private String name,username,pass,regd,gender,branch;

public String getName(){

return name;}

public void setName(String name){

this.name=name;

}

public String getUname(){

return username;

}

public void setUsername(String username){

this.username=username;

}

public String getPass(){

return pass;

}

public void setPass(String password){

this.pass=password;

}

public String getRegd(){

return regd;

}

public void setRegd(String regd){

this.regd=regd;

}

public String getGender(){

return gender;

}

public void setGender(String gender){

this.gender=gender;

}

public String getBranch(){

return branch;

}

public void setBranch(String branch){

this.branch=branch;

}

}

**details.jsp**

<%--

Document : details

Created on : 12 Nov, 2024, 8:02:08 AM

Author : y22acm475

--%>

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

<%@page import="t.bean"%>

<%@page import="java.sql.PreparedStatement"%>

<%@page import="java.sql.Statement"%>

<%@page import="java.sql.DriverManager"%>

<%@page import="java.sql.Connection"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<%

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/y22acm475","sharon","sharon");

Statement st=con.createStatement();

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

String uname=request.getParameter("username");

String pass=request.getParameter("password");

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

String gen=request.getParameter("gender");

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

PreparedStatement pst;

bean jb=new bean();

jb.setName(name);

jb.setUsername(uname);

jb.setPass(pass);

jb.setRegd(regd);

jb.setGender(gen);

jb.setBranch(branch);

pst=con.prepareStatement("insert into login values(?,?,?,?,?,?)");

pst.setString(1,name);

pst.setString(2,uname);

pst.setString(3,pass);

pst.setString(4,regd);

pst.setString(5,gen);

pst.setString(6,branch);

int res =pst.executeUpdate();

if(res==1){

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<h1>Registation Completed Successfully</h1>");

out.println("<style><table,th{ border:1px solid black; }>");

out.println("<td{border:1px solid black; width:30%}></style>");

out.println("</head>");

out.println("<body>");

out.println("<table>");

out.println("<tr><th>NAME</th>"+"<td>"+jb.getName()+"</td></tr>");

out.println("<tr><th>USERNAME:</th>"+"<td>"+jb.getUname()+"</td></tr>");

out.println("<tr><th>PASSWORD</th>"+"<td>can't be displayed</td></tr>");

out.println("<tr><th>REGD</th>"+"<td>"+jb.getRegd()+"</td></tr>");

out.println("<tr><th>GENDER</th>"+"<td>"+jb.getGender()+"</td></tr>");

out.println("<tr><th>BRANCH</th>"+"<td>"+jb.getBranch()+"</td></tr>");

out.println("</table>");

out.println("</body>");

out.println("</html>");

out.println("<form action='index.jsp'>");

out.println("<input type='submit' value='login'>");

}

else{

out.println("<h1>Registation fails</h1>");

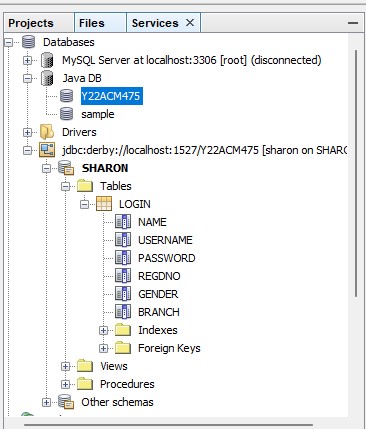
}

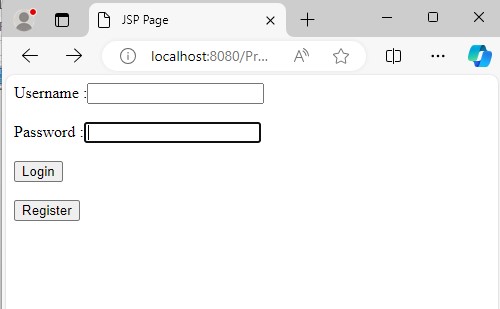
%>

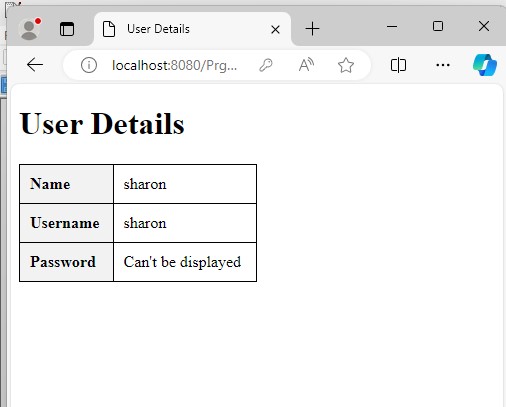
</body>

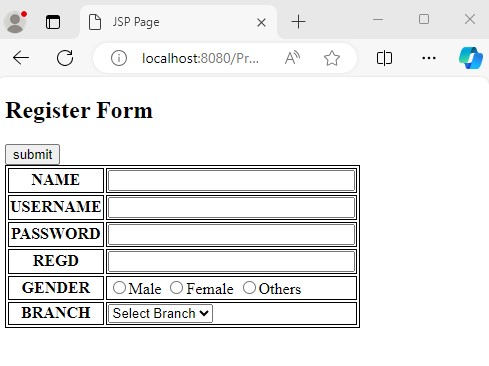
</html>

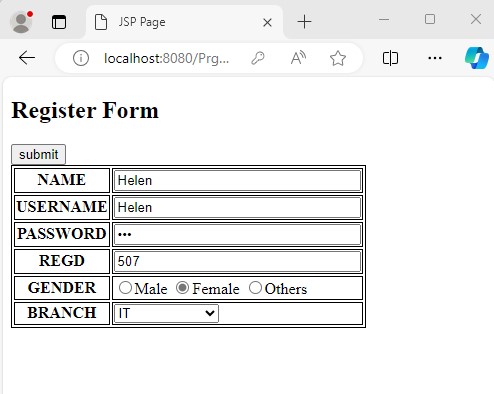
**OUTPUT:**













**EXPERIMENT-05:**

**Aim:** [**Write an application to demonstrate Standard&Custom Tags in JSP**](#A5)**.**

**Execution Steps:**

**1.** Creating the Project:

* Open NetBeans and create a new project by navigating to:  
  **File > New Project > Java Web > Web Application.**  
  Name the project, save it, and click **Finish**.

**2.** Necessary Files and Saving:

* Delete the default **index.html** file that gets created automatically.
* Create the index.jsp file:  
  Right-click on the project, select **New > JSP File**, and name it index.jsp.  
  Insert the required code into this file and save it.
* Create another JSP file:  
  Right-click on the project, select **New > JSP File**, and name it newjsp1.jsp.  
  Insert the relevant code into this file and save it.
* Add a Tag Library Descriptor (TLD) file:  
  Right-click on the project, select **New > Others > Tag Library Descriptor (TLD)**, and name the file newTaglib.  
  Add the appropriate tag definitions to the TLD file to define custom tags. Save the file.
* Create a Tag Handler:  
  Right-click on the project, select **New > Others > Tag Handler**, and provide the package name (e.g., aaa) and file name newTagHandler.  
  While creating the tag handler, click on **Browse**, navigate to WEB-INF > tlds, and select the previously created newTaglib.tld file.  
  Then click **Finish**. Insert the required logic into the tag handler and save the file.

**3.** Database Connection:

* Connect to the database:  
  Navigate to **Services > Databases > Java DB > Start Server**.Use the credentials:**Username**: sharon,**Password**: sharon to establish the connection.

**4**. Table Creation:

* Expand the connected database in the **Services** tab to find the **Tables** section.Right-click on **Tables** and select **Create Table**.
* Enter the table name as lab5 and add the following columns: regno (VARCHAR), names (VARCHAR), s1 (NUMERIC), s2 (NUMERIC), s3 (NUMERIC), s4 (NUMERIC), total (NUMERIC) with a default value of **0**, grade (VARCHAR) with a default value of **null**. Click **OK** to create the table.
* Right-click on the created table (lab5) and select **View Data**.  
  Insert sample values by directly editing the rows displayed in the table view.

**5.** Set File Paths and References:

* Set the required file paths in newjsp1.jsp and newTagHandler.java, ensuring the tag library is referenced in the JSP files using the <%@ taglib ... %> directive.

**6.** Clean, Build, and Run:

* **Clean and build the project**, then **run** it to display the output.

**Source Code:**

**index.html**

<%--

Document : index.jsp

Created on : 12 Nov, 2024, 2:24:18 PM

Author : y22acm475

--%>

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

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<form action="newjsp1.jsp">

<h1>Enter Register number :</h1>

<input type="text" name="regno" placeholder="REGNO(Y22XXXXXX)">

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

</form>

</body>

</html>

**newjsp1.jsp**

<%--

Document : newjsp1

Created on : 12 Nov, 2024, 2:27:02 PM

Author : Y22ACM475

--%>

<%@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/sql" prefix="sql" %>

<%@ taglib uri="/WEB-INF/tlds/newtag\_library.tld" prefix="ct" %>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<sql:setDataSource var='dt' driver="org.apache.derby.jdbc.ClientDriver"

url="jdbc:derby://localhost:1527/y22acm475" user="sharon" password="sharon"/>

<sql:query dataSource="${dt}" var='rs'>

select \* from lab5 where regno=?

<sql:param value="${param.regno}" />

</sql:query>

<p style="color:blue">< \*\*\*\*Welcome ${param.regno}\*\*\*\*></p>

<table border="2">

<th colspan="2">-\*-\*-Exam Results-\*-\*-</th>

<c:forEach items="${rs.rows}" var="r">

<tr><td>Regd No</td><td><c:out value="${r.regno}" /></td></tr>

<tr><td>Name</td><td><c:out value="${r.name}" /></td></tr>

<tr><td>Sub 1 :</td><td><c:out value="${r.s1}" /></td></tr>

<tr><td>Sub 2 :</td><td><c:out value="${r.s2}" /></td></tr>

<tr><td>Sub 3 :</td><td><c:out value="${r.s3}" /></td></tr>

<tr><td>Sub 4 :</td><td><c:out value="${r.s4}" /></td></tr>

<tr><td>Sub 5 :</td><td><c:out value="${r.s5}" /></td></tr>

<tr><td>Sub 6 :</td><td><c:out value="${r.s6}" /></td></tr>

<ct:NewTagHandler1 regno="${r.regno}"s1="${r.s1}" s2="${r.s2}" s3="${r.s3}"

s4="${r.s4}" s5="${r.s5}" s6="${r.s6}" />

</c:forEach>

</table><br>

<form action="index.jsp">

<input type='submit' value="Login page">

</form>

 </body>

</html>

**newtag\_lib**

<?xml version="1.0" encoding="UTF-8"?>

<taglib version="2.1" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-jsptaglibrary\_2\_1.xsd">

<tlib-version>1.0</tlib-version>

<short-name>newtag\_library</short-name>

<uri>/WEB-INF/tlds/newtag\_library</uri>

<!-- A validator verifies that the tags are used correctly at JSP

translation time. Validator entries look like this:

<validator>

<validator-class>com.mycompany.TagLibValidator</validator-class>

<init-param>

<param-name>parameter</param-name>

<param-value>value</param-value>

</init-param>

</validator>

-->

<!-- A tag library can register Servlet Context event listeners in

case it needs to react to such events. Listener entries look

like this:

<listener>

<listener-class>com.mycompany.TagLibListener</listener-class>

</listener>

-->

<tag>

<name>NewTagHandler1</name>

<tag-class>aaa.NewTagHandler1</tag-class>

<body-content>scriptless</body-content>

<attribute>

<name>regno</name>

<rtexprvalue>true</rtexprvalue>

<type>java.lang.String</type>

</attribute>

<attribute>

<name>name</name>

<rtexprvalue>true</rtexprvalue>

<type>java.lang.String</type>

</attribute>

<attribute>

<name>s1</name>

<rtexprvalue>true</rtexprvalue>

<type>int</type>

</attribute>

<attribute>

<name>s2</name>

<rtexprvalue>true</rtexprvalue>

<type>int</type>

</attribute>

<attribute>

<name>s3</name>

<rtexprvalue>true</rtexprvalue>

<type>int</type>

</attribute>

<attribute>

<name>s4</name>

<rtexprvalue>true</rtexprvalue>

<type>int</type>

</attribute>

<attribute>

<name>s5</name>

<rtexprvalue>true</rtexprvalue>

<type>int</type>

</attribute>

<attribute>

<name>s6</name>

<rtexprvalue>true</rtexprvalue>

<type>int</type>

</attribute>

</tag>

<tag>

<name>NewTagHandler1</name>

<tag-class>aaa.NewTagHandler1</tag-class>

<body-content>scriptless</body-content>

</tag>

</taglib>

**TagNewHandler1**

package aaa;

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.servlet.jsp.JspWriter;

import javax.servlet.jsp.JspException;

import javax.servlet.jsp.tagext.JspFragment;

import javax.servlet.jsp.tagext.SimpleTagSupport;

public class NewTagHandler1 extends SimpleTagSupport {

private String regno;

private int s1;

private int s2;

private int s3;

private int s4;

private int s5;

private int s6;

/\*\*

\* Called by the container to invoke this tag. The implementation of this

\* method is provided by the tag library developer, and handles all tag

\* processing, body iteration, etc.

\* @throws javax.servlet.jsp.JspException

\* @throws java.io.IOException

\*/

@Override

public void doTag() throws JspException, IOException {

JspWriter out = getJspContext().getOut();

int total = s1 + s2 + s3 + s4 + s5 + s6;

String grade;

String color;

if (total >= 550) {

grade = "A";

color = "green";

}

else if (total >= 450) {

grade = "B";

color = "blue";

}

else if (total >= 350) {

grade = "C";

color = "orange";

}

else {

grade = "F";

color = "red";

}

out.write("<tr><td>Total</td><td>" + total + "</td></tr>");

out.write("<tr><td>Grade</td><td style='color:" + color + "'>" + grade + "</td></tr>");

try {

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/y22acm475","sharon","sharon");

Statement st=con.createStatement();

ResultSet rs;

rs=st.executeQuery("select \* from lab5 where regno='"+regno+"'");

if(rs.next()){

st.execute("update lab5 set total="+total);

st.execute("update lab5 set grade="+grade);

}

JspFragment f = getJspBody();

if (f!= null) {

f.invoke(out);

}}

catch (java.io.IOException ex){

throw new JspException("Error in NewTagHandler tag", ex);

}

catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(NewTagHandler1.class.getName()).log(Level.SEVERE, null, ex);

}

}

public void setRegno(String regno){

this.regno = regno;

}

public void setS1(int s1) {

this.s1 = s1;

}

public void setS2(int s2) {

this.s2 = s2;

}

public void setS3(int s3) {

this.s3 = s3;

}

public void setS4(int s4) {

this.s4 = s4;

}

public void setS5(int s5) {

this.s5 = s5;

}

public void setS6(int s6) {

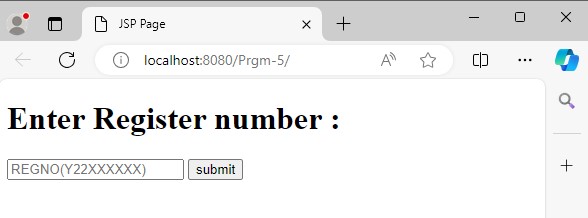
this.s6=s6;

}

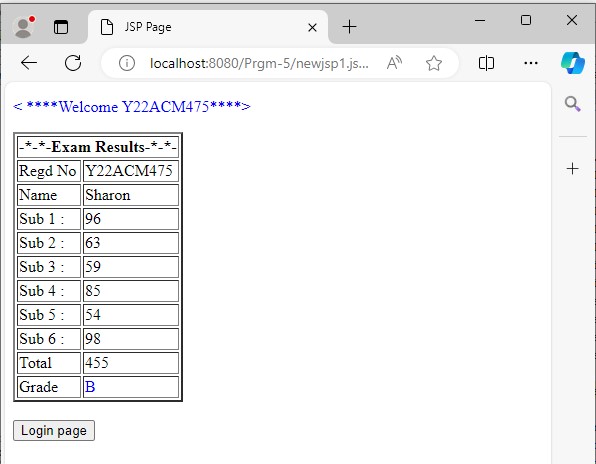
}

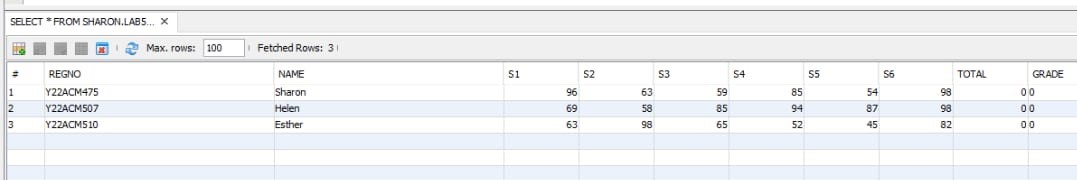
**OUTPUT:**

|  |  |
| --- | --- |
| C:\Users\hp\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\image (16).jpeg | C:\Users\hp\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\image (17).jpeg |









**EXPERIMENT-06**

**Aim:** **Write an application to demonstrate Java Server Faces (JSF) Validators, Event handlers and converters.**

**Execution Steps:**

1. **Open NetBeans** and follow:  
   **File** > **New Project** > **Java Web** > **Web Application**.  
   **Name** your project and **save** it.
2. While creating the **Web Application**, make sure you select **JavaServer Faces** under **Frameworks** and click **Finish**.
3. Check whether you select **GlassFish Server** in the dropdown list and the **JavaDB driver** in the **Library** section.  
   Check the **properties** and move on.
4. Creating and saving necessary files:
   * You will have an **index.xhtml** file. Insert the **code** and **save** the file.For the file **preview.xhtml**, **right-click** on the project and select **New**.Search for **XHTML**, insert the **code**, and **save**.
   * For **Java classes**, **right-click** on the project and select **New**.  
     Search for **Servlet** and add **3 servlets**:**EmailCheck, PhnConverter,RegNo.**Insert the **code** and **save**.
   * For a **Java class**, **right-click** on the project and select **New**.  
     Search for **Java Class**, name it as **ManagedBean**, insert the **code**, and **save**.
   * Make sure you **check the box** for adding the **descriptors** to **web.xml** each time you create a **Servlet** or **Java class**.
   * Ensure that **all class names match** as expected.
5. **Clean and build** the project, then click **Run**.

**Source Code:**

**index.xhtml**

<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://xmlns.jcp.org/jsf/html"

xmlns:f="http://xmlns.jcp.org/jsf/core">

<h:head>

<title>Registration Page</title>

<style>

.cen {

text-align: center;

}

</style>

</h:head>

<h:body>

<center>

<h:form>

<h1>Registration Page</h1>

<h:panelGrid columns="2">

<h:outputLabel value="First Name:" for="fname" />

<h:inputText id="fname" value="#{mb.fname}" required="true">

<f:validateRequired />

</h:inputText>

<h:outputLabel value="Last Name:" for="lname" />

<h:inputText id="lname" value="#{mb.lname}" required="true">

<f:validateRequired />

</h:inputText>

<h:outputLabel value="Father Name:" for="father" />

<h:inputText id="father" value="#{mb.father}" required="true">

<f:validateRequired />

</h:inputText>

<h:outputLabel value="Email Address:" for="email" />

<h:inputText id="email" value="#{mb.email}" required="true">

<f:validator validatorId="newval" />

</h:inputText>

<h:outputLabel value="Register Number:" for="regdno" />

<h:inputText id="regdno" value="#{mb.regdno}" required="true">

<f:validator validatorId="regd" />

</h:inputText>

<h:outputLabel value="Password:" for="password" />

<h:inputSecret id="password" value="#{mb.password}" required="true" />

<h:outputLabel value="Confirm Password:" for="confirmpass" />

<h:inputSecret id="confirmpass" value="#{mb.confirmpass}" required="true" />

<h:outputLabel value="Gender:" for="gender" />

<h:selectOneRadio id="gender" value="#{mb.gender}">

<f:selectItem itemLabel="Male" itemValue="Male" />

<f:selectItem itemLabel="Female" itemValue="Female" />

</h:selectOneRadio>

<h:outputLabel value="Language:" for="lang" />

<h:selectManyCheckbox id="lang" value="#{mb.lang}">

<f:selectItem itemLabel="English" itemValue="English" />

<f:selectItem itemLabel="Hindi" itemValue="Hindi" />

<f:selectItem itemLabel="Telugu" itemValue="Telugu" />

</h:selectManyCheckbox>

<h:outputLabel value="Phone Number:" for="phone" />

<h:inputText id="phone" value="#{mb.phone}">

<f:converter converterId="phn" />

</h:inputText>

</h:panelGrid>

<h:commandButton action="preview.xhtml" value="Submit" />

</h:form>

</center>

</h:body>

</html>

**preview.html**

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://xmlns.jcp.org/jsf/html"

xmlns:f="http://xmlns.jcp.org/jsf/core">

<head>

<title>User Information</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<style>

body {

font-family: Arial, sans-serif;

margin: 20px;

}

h2 {

text-align: center;

}

.info-table {

margin: 0 auto;

border-collapse: collapse;

width: 50%;

}

.info-table th, .info-table td {

border: 1px solid #ddd;

padding: 8px;

text-align: left;

}

.info-table th {

background-color: #f2f2f2;

}

</style>

</head>

<body>

<h2>User Information</h2>

<div>

<h:panelGrid columns="2" class="info-table">

<h:outputLabel value="First Name:" />

<h:outputText value="#{mb.fname}" />

<h:outputLabel value="Last Name:" />

<h:outputText value="#{mb.lname}" />

<h:outputLabel value="Father Name:" />

<h:outputText value="#{mb.father}" />

<h:outputLabel value="Email:" />

<h:outputText value="#{mb.email}" />

<h:outputLabel value="Register Number:" />

<h:outputText value="#{mb.regdno}" />

<h:outputLabel value="Gender:" />

<h:outputText value="#{mb.gender}" />

<h:outputLabel value="Phone Number:" />

<h:outputText value="#{phn.phone}" />

</h:panelGrid>

</div>

</body>

</html>

**managedbean.java**

import java.util.Date;

import javax.faces.bean.ManagedBean;

import javax.faces.bean.RequestScoped;

@ManagedBean(name = "mb")

@RequestScoped

public class managedbean {

String regdno,fname,lname,father,email,password,confirmpass,gender,lang[],phone,state,dis,man;

Date dob;

public Date getDob() {

return dob;

}

public void setDob(Date dob) {

this.dob = dob;

}

public String getRegdno() {

return regdno;

}

public void setRegdno(String regdno) {

this.regdno = regdno;

}

public String getFname() {

return fname;

}

public void setFname(String fname) {

this.fname = fname;

}

public String getLname() {

return lname;

}

public void setLname(String lname) {

this.lname = lname;

}

public String getFather() {

return father;

}

public void setFather(String father) {

this.father = father;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getConfirmpass() {

return confirmpass;

}

public void setConfirmpass(String confirmpass) {

this.confirmpass = confirmpass;

}

public String getGender() {

return gender;

}

public void setGender(String gender) {

this.gender = gender;

}

public String[] getLang() {

return lang;

}

public void setLang(String[] lang) {

this.lang = lang;

}

public String getPhone() {

return phone;

}

public void setPhone(String phone) {

this.phone = phone;

}

public String getMan() {

return man;

}

public void setMan(String man) {

this.man = man;

}

}

**emailcheck.java**

import javax.faces.application.FacesMessage;

import javax.faces.component.UIComponent;

import javax.faces.context.FacesContext;

import javax.faces.validator.FacesValidator;

import javax.faces.validator.Validator;

import javax.faces.validator.ValidatorException;

@FacesValidator("newval")

public class EmailCheck implements Validator {

@Override

public void validate(FacesContext fc, UIComponent uic, Object value) throws ValidatorException {

if (value == null || value.toString().trim().isEmpty()) {

// You might want to handle required fields separately.

FacesMessage msg = new FacesMessage("Email is required");

throw new ValidatorException(msg);

}

String email = value.toString();

// Regex for validating email addresses

String emailRegex = "^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,}$";

if (!email.matches(emailRegex)) {

FacesMessage msg = new FacesMessage("Please enter a valid email address (e.g., user@example.com)");

throw new ValidatorException(msg);

}

}

}

**phnConverter.java**

import javax.faces.component.UIComponent;

import javax.faces.context.FacesContext;

import javax.faces.convert.Converter;

import javax.faces.convert.FacesConverter;

import java.util.logging.Level;

import java.util.logging.Logger;

@FacesConverter("phn")

public class PhnConverter implements Converter {

private static final String COUNTRY\_CODE = "+91";

private static final Logger LOGGER = Logger.getLogger(PhnConverter.class.getName());

@Override

public Object getAsObject(FacesContext fc, UIComponent uic, String value) {

if (value == null || value.isEmpty()) {

return null;

}

if (!isValidPhoneNumber(value)) {

LOGGER.log(Level.WARNING, "Invalid phone number: {0}", value);

return null;

}

return COUNTRY\_CODE + value.trim();

}

@Override

public String getAsString(FacesContext fc, UIComponent uic, Object object) {

if (object == null) {

return "";

}

String phoneNumber = object.toString();

return phoneNumber.replace(COUNTRY\_CODE, "").trim();

}

private boolean isValidPhoneNumber(String phoneNumber) {

return phoneNumber.matches("\\d{10}");

}

}

**regdno.java**

import javax.faces.application.FacesMessage;

import javax.faces.component.UIComponent;

import javax.faces.context.FacesContext;

import javax.faces.validator.FacesValidator;

import javax.faces.validator.Validator;

import javax.faces.validator.ValidatorException;

@FacesValidator("regd")

public class Regdno implements Validator {

@Override

public void validate(FacesContext context, UIComponent component, Object value) throws ValidatorException {

String regdno = (String) value;

if (regdno == null || !regdno.matches("\\d{9}")) {

FacesMessage msg = new FacesMessage("Invalid Register Number");

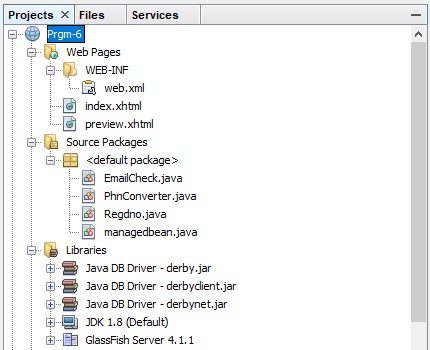
msg.setSeverity(FacesMessage.SEVERITY\_ERROR);

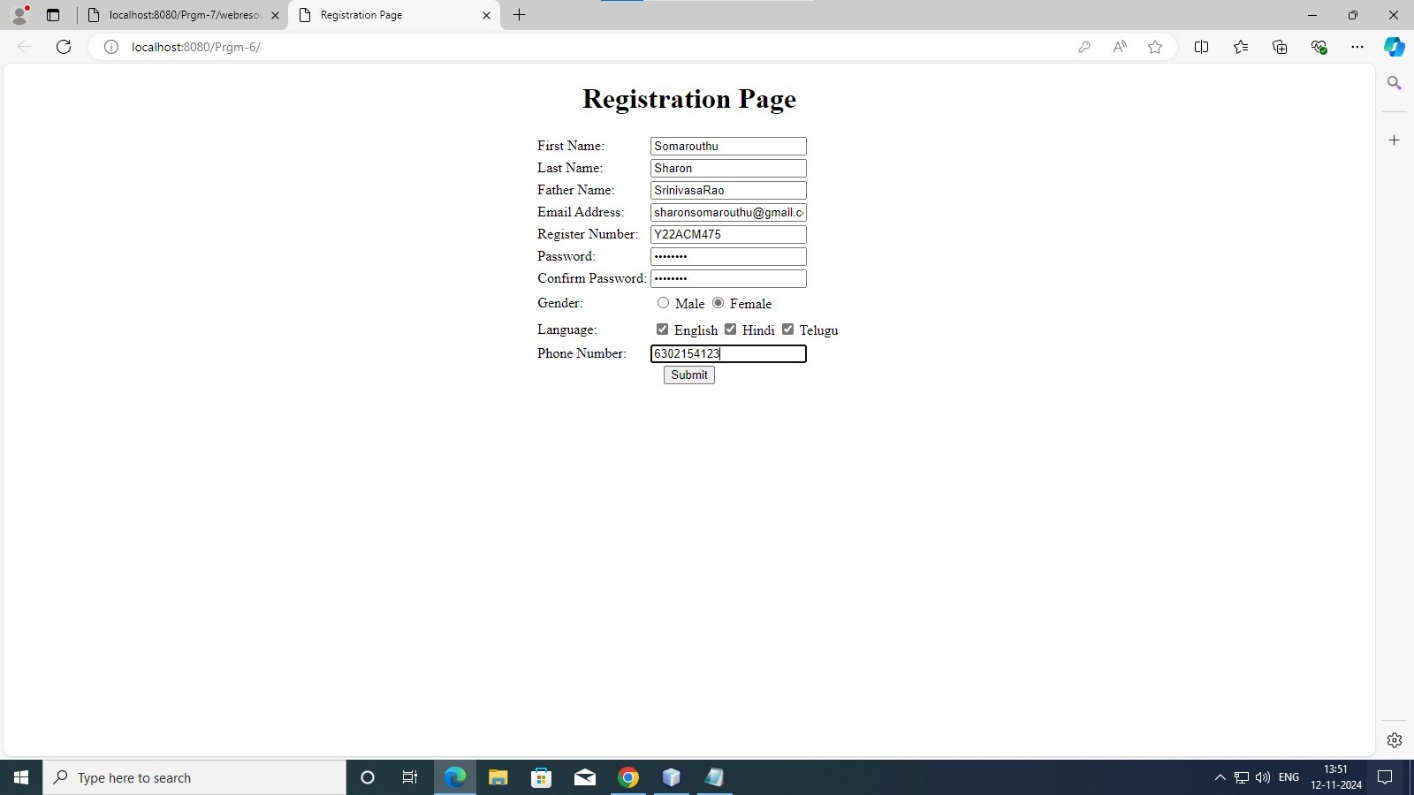
throw new ValidatorException(msg);

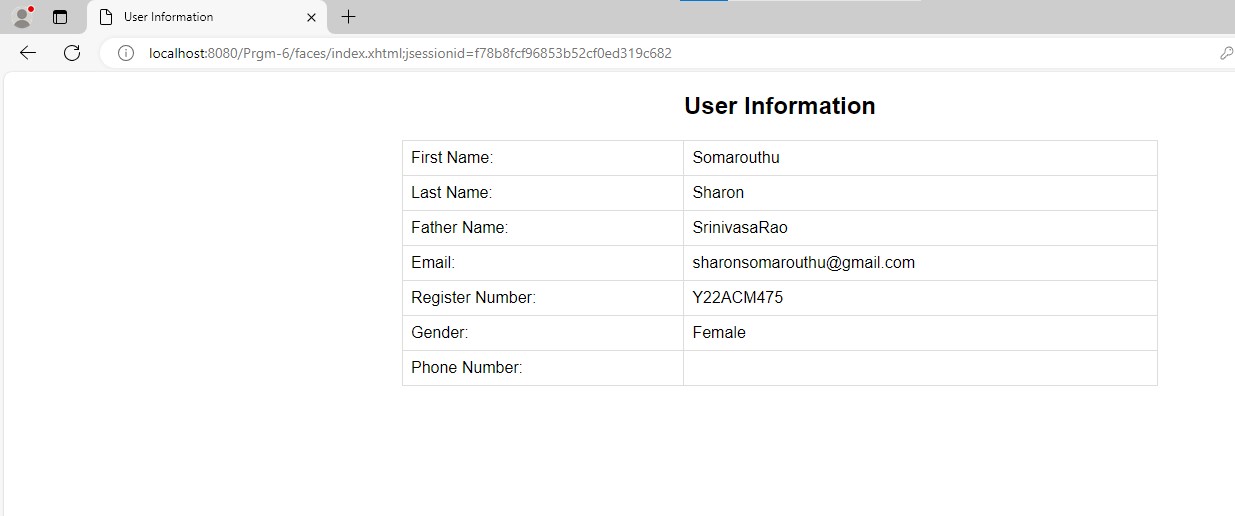
}

}}

**OUTPUT:**







**EXPERIMENT-07:**

**Aim:** [**Write an application to demonstrate web service**](#A7).

**Execution Steps:**

1. **Open NetBeans** and follow:  
   **File** > **New Project** > **JavaWeb** > **Web Application**.  
   **Name** your project and **save** it.
2. Make sure you select **GlassFish Server** in the dropdown list and added **JavaDB driver** in the **Library** section. Check for the **properties** and move on.
3. Creating necessary files and saving
   * You will have an **index**. Insert the **code** and **save** the file.
   * For **Java classes**, **right-click** on the project and select **New**.  
     Search for **Servlet** and add **3 servlets** named **GenericResources**,**Application**, and **MyClasses**.  
     Insert the **code** and **save**.
   * Make sure you **check the box** for adding the **descriptors** to **web.xml** each time you create a **Servlet** or **Java class**.
   * Make sure all **class names** match, and also check for the **correct project name** in the **URL** provided.
4. **Clean and build** the project, then click **Run**.

**Source Code:**

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Arithmetic Operations Application</title>

</head>

<body>

<form action="http://localhost:8080/Prgm-7/webresources/generic" method="get">

<div align="center">

<h1>

Enter a Number: <input type="text" name="a"/><br/><br/>

Enter b Number:<input type="text" name="b"/><br/><br/><br/>

<div style="color: red;" align="center">

<input type="submit" value="Calculate"/>

</div>

</h1>

</div>

</form>

</body>

</html>

**GenericResource.java**

package pkg;

import javax.ws.rs.\*;

import javax.ws.rs.core.\*;

@Path("generic")

public class GenericResource {

@Context

private UriInfo context;

public GenericResource() {

}

@GET

@Produces(MediaType.TEXT\_HTML)

public String getHtml() {

try {

// Parse input numbers

int a = Integer.parseInt(context.getQueryParameters().getFirst("a"));

int b = Integer.parseInt(context.getQueryParameters().getFirst("b"));

int addition = a + b;

int subtraction = a - b;

int multiplication = a \* b;

double division = b != 0 ? (double) a / b : Double.NaN;

int modulo = b != 0 ? a % b : Integer.MIN\_VALUE;

long factorialA = factorial(a);

long factorialB = factorial(b);

boolean isPrimeA = isPrime(a);

boolean isPrimeB = isPrime(b);

boolean isPerfectA = isPerfect(a);

boolean isPerfectB = isPerfect(b);

String multiplicationTableA = multiplicationTable(a);

String multiplicationTableB = multiplicationTable(b);

int sumEven = sumOfEvenNumbers(a, b);

int sumOdd = sumOfOddNumbers(a, b);

StringBuilder result = new StringBuilder("<div align='center'><h1>Results:</h1>");

result.append("<h2>Basic Operations</h2>");

result.append("<table border='1' style='border-collapse: collapse;'><tr><th>Operation</th><th>Result</th></tr>");

result.append("<tr><td>Addition</td><td>").append(addition).append("</td></tr>");

result.append("<tr><td>Subtraction</td><td>").append(subtraction).append("</td></tr>");

result.append("<tr><td>Multiplication</td><td>").append(multiplication).append("</td></tr>");

result.append("<tr><td>Division</td><td>").append(Double.isNaN(division) ? "undefined" : division).append("</td></tr>");

result.append("<tr><td>Modulo</td><td>").append(modulo == Integer.MIN\_VALUE ? "undefined" : modulo).append("</td></tr>");

result.append("<tr><td>Sum of Even Numbers</td><td>").append(sumEven).append("</td></tr>");

result.append("<tr><td>Sum of Odd Numbers</td><td>").append(sumOdd).append("</td></tr>");

result.append("</table>");

result.append("<h2>Factorial</h2>");

result.append("<table border='1' style='border-collapse: collapse;'><tr><th>Number</th><th>Factorial</th></tr>");

result.append("<tr><td>").append(a).append("</td><td>").append(factorialA).append("</td></tr>"); result.append("<tr><td>").append(b).append("</td><td>").append(factorialB).append("</td></tr>");

result.append("</table>");

result.append("<h2>Prime Check</h2>");

result.append("<table border='1' style='border-collapse: collapse;'><tr><th>Number</th><th>Is Prime?</th></tr>");

result.append("<tr><td>").append(a).append("</td><td>").append(isPrimeA ? "Yes" : "No").append("</td></tr>");

result.append("<tr><td>").append(b).append("</td><td>").append(isPrimeB ? "Yes" : "No").append("</td></tr>");

result.append("</table>");

result.append("<h2>Perfect Number Check</h2>");

result.append("<table border='1' style='border-collapse: collapse;'><tr><th>Number</th><th>Is Perfect?</th></tr>");

result.append("<tr><td>").append(a).append("</td><td>").append(isPerfectA ? "Yes" : "No").append("</td></tr>");

result.append("<tr><td>").append(b).append("</td><td>").append(isPerfectB ? "Yes" : "No").append("</td></tr>");

result.append("</table>");

result.append("<h2>Multiplication Tables</h2>");

result.append("<h3>Table for ").append(a).append(":</h3><p>").append(multiplicationTableA).append("</p>");

result.append("<h3>Table for ").append(b).append(":</h3><p>").append(multiplicationTableB).append("</p>");

result.append("</div>");

return result.toString();

} catch (NumberFormatException e) {

return "<div align='center'><h1>Error: Please enter valid integers.</h1></div>";

}

}

private long factorial(int number) {

if (number < 0) return -1; // Error for negative numbers

long result = 1;

for (int i = 1; i <= number; i++) {

result \*= i;

}

return result;

}

private boolean isPrime(int number) {

if (number <= 1) return false;

for (int i = 2; i <= Math.sqrt(number); i++) {

if (number % i == 0) return false;

}

return true;

}

private int sumOfEvenNumbers(int a, int b) {

int sum = 0;

for (int i = Math.min(a, b); i <= Math.max(a, b); i++) {

if (i % 2 == 0) {

sum += i;

}

}

return sum;

}

private int sumOfOddNumbers(int a, int b) {

int sum = 0;

for (int i = Math.min(a, b); i <= Math.max(a, b); i++) {

if (i % 2 != 0) {

sum += i;

}

}

return sum;

}

private String multiplicationTable(int number) {

StringBuilder table = new StringBuilder();

for (int i = 1; i <= 10; i++) {

table.append(number).append(" x ").append(i).append(" = ").append(number \* i).append("<br/>");

}

return table.toString();

}

private boolean isPerfect(int number) {

if (number < 1) return false;

int sum = 0;

for (int i = 1; i < number; i++) {

if (number % i == 0) {

sum += i;

}

}

return sum == number;

}

@PUT

@Consumes(MediaType.TEXT\_HTML)

public void putHtml(String content) {

// Not implemented

}

}

**ApplicationConfig.java**

package pkg;

import java.util.Set;

import javax.ws.rs.core.Application;

@javax.ws.rs.ApplicationPath("webresources")

public class ApplicationConfig extends Application {

@Override

public Set<Class<?>> getClasses() {

Set<Class<?>> resources = new java.util.HashSet<>();

addRestResourceClasses(resources);

return resources;

}

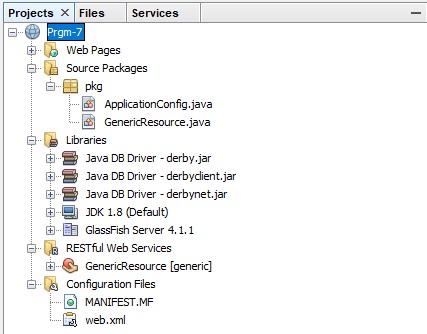
private void addRestResourceClasses(Set<Class<?>> resources) {

resources.add(pkg.GenericResource.class);

}

}

**OUTPUT:**



|  |  |
| --- | --- |
|  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Results:Basic Operations  |  |  | | --- | --- | | **Operation** | **Result** | | Addition | 8 | | Subtraction | -2 | | Multiplication | 15 | | Division | 0.6 | | Modulo | 3 | | Sum of Even Numbers | 4 | | Sum of Odd Numbers | 8 |  Factorial  |  |  | | --- | --- | | **Number** | **Factorial** | | 3 | 6 | | 5 | 120 |  Prime Check  |  |  | | --- | --- | | **Number** | **Is Prime?** | | 3 | Yes | | 5 | Yes |  Perfect Number Check  |  |  | | --- | --- | | **Number** | **Is Perfect?** | | 3 | No | | 5 | No | | Multiplication TablesTable for 3: 3 x 1 = 3 3 x 2 = 6 3 x 3 = 9 3 x 4 = 12 3 x 5 = 15  3 x 6 = 18 3 x 7 = 21 3 x 8 = 24 3 x 9 = 27 3 x 10 = 30 Table for 5: 5 x 1 = 5 5 x 2 = 10 5 x 3 = 15 5 x 4 = 20 5 x 5 = 25 5 x 6 = 30 5 x 7 = 35 5 x 8 = 40 5 x 9 = 45 5 x 10 = 50 |

**EXPERIMENT-08:**

**Aim:**[**Write an application using Web sockets**](#A8).

**Execution Steps**:

1. **Open NetBeans and follow:File** > **New Project** > **JavaWeb** > **Web Application.**Provide the project name (e.g., Program-8), click **Next**, and then **Finish**.
2. Creating and Configuring Files
   * index.html: Right-click on the project, go to **New > HTML File**, and name it index.html Add the required code to this file and save it.
   * Device: Right-click on the project, go to **New > Servlet**, and name it Employee. Check **Add Information to Deployment Descriptor** and click Finish.Add the necessary code to this servlet and save it.
   * Additional Servlets: Follow the same process to create the following servlet files:  
     DeviceSessionHandler.java,DeviceWebSocketServer.java
   * style.css: Right-click on the project, go to **New > Other >Cascading style sheets**, and name it style.css Add the required code to this file and save it.
   * websocket.js:Right-click on the project, go to **New > Other>Java Script File**, and name it websocket.js Add the required code to this file and save it.
3. Building and Running the Project
   * Right-click on the project and select **Clean and Build** and then **Run**

**Source Code:**

**Index.html**

<!DOCTYPE html>

<html>

<head>

<title></title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<script src="websocket.js"></script>

<link rel="stylesheet" type="text/css" href="style.css">

<link rel="stylesheet" href="style.css" type="text/css"/>

<link rel="stylesheet" href="style.css" type="text/css">

<link rel="stylesheet" href="style.css" type="text/css"/>

</head>

<body>

<div id="wrapper">

<h1>Java Websocket Home</h1>

<p>Welcome to the Java WebSocket Home. Click the Add a device button to start adding devices.</p>

<br />

<div id="addDevice">

<div class="button"> <a href="#" OnClick="showForm()">Add a device</a> </div>

<form id="addDeviceForm">

<h3>Add a new device</h3>

<span>Name: <input type="text" name="device\_name" id="device\_name"></span>

<span>Type:

<select id="device\_type">

<option name="type" value="Appliance">Appliance</option>

<option name="type" value="Electronics">Electronics</option>

<option name="type" value="Lights">Lights</option>

<option name="type" value="Other">Other</option>

</select></span>

<span>Description:<br />

<textarea name="description" id="device\_description" rows="2" cols="50"></textarea>

</span>

<input type="button" class="button" value="Add" onclick=formSubmit()>

<input type="reset" class="button" value="Cancel" onclick=hideForm()>

</form>

</div>

<br />

<h3>Currently connected devices:</h3>

<div id="content">

</div>

</div>

</body>

</html>

**Style.css**

body {

font-family: Arial, Helvetica, sans-serif;

font-size: 80%;

background-color: #1f1f1f;

}

#wrapper {

width: 960px;

margin: auto;

text-align: left;

color: #d9d9d9;

}

p {

text-align: left;

}

.button {

display: inline;

color: #fff;

background-color: #f2791d;

padding: 8px;

margin: auto;

border-radius: 8px;

-moz-border-radius: 8px;

-webkit-border-radius: 8px;

box-shadow: none;

border: none;

}

.button:hover {

background-color: #ffb15e;

}

.button a, a:visited, a:hover, a:active {

color: #fff;

text-decoration: none;

}

#addDevice {

text-align: center;

width: 960px;

margin: auto;

margin-bottom: 10px;

}

#addDeviceForm {

text-align: left;

width: 400px;

margin: auto;

padding: 10px;

}

#addDeviceForm span {

display: block;

}

#content {

margin: auto;

width: 960px;

}

.device {

width: 180px;

height: 110px;

margin: 10px;

padding: 16px;

color: #fff;

vertical-align: top;

border-radius: 8px;

-moz-border-radius: 8px;

-webkit-border-radius: 8px;

display: inline-block;

}

.device.off {

background-color: #c8cccf;

}

.device span {

display: block;

}

.deviceName {

text-align: center;

font-weight: bold;

margin-bottom: 12px;

}

.removeDevice {

margin-top: 12px;

text-align: center;

}

.device.Appliance {

background-color: #5eb85e;

}

.device.Appliance a:hover {

color: #a1ed82;

}

.device.Electronics {

background-color: #0f90d1;

}

.device.Electronics a:hover {

color: #4badd1;

}

.device.Lights {

background-color: #c2a00c;

}

.device.Lights a:hover {

color: #fad232;

}

.device.Other {

background-color: #db524d;

}

.device.Other a:hover {

color: #ff907d;

}

.device a {

text-decoration: none;

}

.device a:visited, a:active, a:hover {

color: #fff;

}

.device a:hover {

text-decoration: underline;

}

**websocket.js**

window.onload = init;

var socket = new WebSocket("ws://localhost:8080/WebsocketHome/actions");

socket.onmessage = onMessage;

function onMessage(event) {

var device = JSON.parse(event.data);

if (device.action === "add") {

printDeviceElement(device);

}

if (device.action === "remove") {

document.getElementById(device.id).remove();

//device.parentNode.removeChild(device);

}

if (device.action === "toggle") {

var node = document.getElementById(device.id);

var statusText = node.children[2];

if (device.status === "On") {

statusText.innerHTML = "Status: " + device.status + " (<a href=\"#\" OnClick=toggleDevice(" + device.id + ")>Turn off</a>)";

} else if (device.status === "Off") {

statusText.innerHTML = "Status: " + device.status + " (<a href=\"#\" OnClick=toggleDevice(" + device.id + ")>Turn on</a>)";

}

}

}

function addDevice(name, type, description) {

var DeviceAction = {

action: "add",

name: name,

type: type,

description: description

};

socket.send(JSON.stringify(DeviceAction));

}

function removeDevice(element) {

var id = element;

var DeviceAction = {

action: "remove",

id: id

};

socket.send(JSON.stringify(DeviceAction));

}

function toggleDevice(element) {

var id = element;

var DeviceAction = {

action: "toggle",

id: id

};

socket.send(JSON.stringify(DeviceAction));

}

function printDeviceElement(device) {

var content = document.getElementById("content");

var deviceDiv = document.createElement("div");

deviceDiv.setAttribute("id", device.id);

deviceDiv.setAttribute("class", "device " + device.type);

content.appendChild(deviceDiv);

var deviceName = document.createElement("span");

deviceName.setAttribute("class", "deviceName");

deviceName.innerHTML = device.name;

deviceDiv.appendChild(deviceName);

var deviceType = document.createElement("span");

deviceType.innerHTML = "<b>Type:</b> " + device.type;

deviceDiv.appendChild(deviceType);

var deviceStatus = document.createElement("span");

if (device.status === "On") {

deviceStatus.innerHTML = "<b>Status:</b> " + device.status + " (<a href=\"#\" OnClick=toggleDevice(" + device.id + ")>Turn off</a>)";

} else if (device.status === "Off") {

deviceStatus.innerHTML = "<b>Status:</b> " + device.status + " (<a href=\"#\" OnClick=toggleDevice(" + device.id + ")>Turn on</a>)";

//deviceDiv.setAttribute("class", "device off");

}

deviceDiv.appendChild(deviceStatus);

var deviceDescription = document.createElement("span");

deviceDescription.innerHTML = "<b>Comments:</b> " + device.description;

deviceDiv.appendChild(deviceDescription);

var removeDevice = document.createElement("span");

removeDevice.setAttribute("class", "removeDevice");

removeDevice.innerHTML = "<a href=\"#\" OnClick=removeDevice(" + device.id + ")>Remove device</a>";

deviceDiv.appendChild(removeDevice);

}

function showForm() {

document.getElementById("addDeviceForm").style.display = '';

}

function hideForm() {

document.getElementById("addDeviceForm").style.display = "none";

}

function formSubmit() {

var form = document.getElementById("addDeviceForm");

var name = form.elements["device\_name"].value;

var type = form.elements["device\_type"].value;

var description = form.elements["device\_description"].value;

hideForm();

document.getElementById("addDeviceForm").reset();

addDevice(name, type, description);

}

function init() {

hideForm();

}

**Device.java**

package org.example.model;

public class Device {

private int id;

private String name;

private String status;

private String type;

private String description;

public Device() {

}

public int getId() {

return id;

}

public String getName() {

return name;

}

public String getStatus() {

return status;

}

public String getType() {

return type;

}

public String getDescription() {

return description;

}

public void setId(int id) {

this.id = id;

}

public void setName(String name) {

this.name = name;

}

public void setStatus(String status) {

this.status = status;

}

public void setType(String type) {

this.type = type;

}

public void setDescription(String description) {

this.description = description;

}

}

**DeviceSessionHandler.java**

package org.example.websocket;

import java.io.IOException;

import java.util.ArrayList;

import javax.enterprise.context.ApplicationScoped;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.json.JsonObject;

import javax.json.spi.JsonProvider;

import javax.websocket.Session;

import org.example.model.Device;

@ApplicationScoped

public class DeviceSessionHandler {

private int deviceId = 0;

private final Set<Session> sessions = new HashSet<>();

private final Set<Device> devices = new HashSet<>();

public void addSession(Session session) {

sessions.add(session);

for (Device device : devices) {

JsonObject addMessage = createAddMessage(device);

sendToSession(session, addMessage);

}

}

public void removeSession(Session session) {

sessions.remove(session);

}

public List<Device> getDevices() {

return new ArrayList<>(devices);

}

public void addDevice(Device device) {

device.setId(deviceId);

devices.add(device);

deviceId++;

JsonObject addMessage = createAddMessage(device);

sendToAllConnectedSessions(addMessage);

}

public void removeDevice(int id) {

Device device = getDeviceById(id);

if (device != null) {

devices.remove(device);

JsonProvider provider = JsonProvider.provider();

JsonObject removeMessage = provider.createObjectBuilder()

.add("action", "remove")

.add("id", id)

.build();

sendToAllConnectedSessions(removeMessage);

}

}

public void toggleDevice(int id) {

JsonProvider provider = JsonProvider.provider();

Device device = getDeviceById(id);

if (device != null) {

if ("On".equals(device.getStatus())) {

device.setStatus("Off");

} else {

device.setStatus("On");

}

JsonObject updateDevMessage = provider.createObjectBuilder()

.add("action", "toggle")

.add("id", device.getId())

.add("status", device.getStatus())

.build();

sendToAllConnectedSessions(updateDevMessage);

}

}

private Device getDeviceById(int id) {

for (Device device : devices) {

if (device.getId() == id) {

return device;

}

}

return null;

}

private JsonObject createAddMessage(Device device) {

JsonProvider provider = JsonProvider.provider();

JsonObject addMessage = provider.createObjectBuilder()

.add("action", "add")

.add("id", device.getId())

.add("name", device.getName())

.add("type", device.getType())

.add("status", device.getStatus())

.add("description", device.getDescription())

.build();

return addMessage;

}

private void sendToAllConnectedSessions(JsonObject message) {

for (Session session : sessions) {

sendToSession(session, message);

}

}

private void sendToSession(Session session, JsonObject message) {

try {

session.getBasicRemote().sendText(message.toString());

} catch (IOException ex) {

sessions.remove(session);

Logger.getLogger(DeviceSessionHandler.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

**DeviceWebSocketServer.java**

package org.example.websocket;

import javax.websocket.OnClose;

import javax.websocket.OnError;

import javax.websocket.OnMessage;

import javax.websocket.OnOpen;

import javax.websocket.Session;

import javax.websocket.server.ServerEndpoint;

import javax.enterprise.context.ApplicationScoped;

import javax.inject.Inject;

import java.io.StringReader;

import javax.json.Json;

import javax.json.JsonObject;

import javax.json.JsonReader;

import org.example.model.Device;

import java.util.logging.Level;

import java.util.logging.Logger;

@ApplicationScoped

@ServerEndpoint("/actions")

public class DeviceWebSocketServer {

@Inject

private DeviceSessionHandler sessionHandler;

@OnOpen

public void open(Session session) {

sessionHandler.addSession(session);

}

@OnClose

public void close(Session session) {

sessionHandler.removeSession(session);

}

@OnError

public void onError(Throwable error) { Logger.getLogger(DeviceWebSocketServer.class.getName()).log(Level.SEVERE, null, error);

}

@OnMessage

public void handleMessage(String message, Session session) {

try (JsonReader reader = Json.createReader(new StringReader(message))) {

JsonObject jsonMessage = reader.readObject();

if ("add".equals(jsonMessage.getString("action"))) {

Device device = new Device();

device.setName(jsonMessage.getString("name"));

device.setDescription(jsonMessage.getString("description"));

device.setType(jsonMessage.getString("type"));

device.setStatus("Off");

sessionHandler.addDevice(device);

}

if ("remove".equals(jsonMessage.getString("action"))) {

int id = (int) jsonMessage.getInt("id");

sessionHandler.removeDevice(id);

}

if ("toggle".equals(jsonMessage.getString("action"))) {

int id = (int) jsonMessage.getInt("id");

sessionHandler.toggleDevice(id);

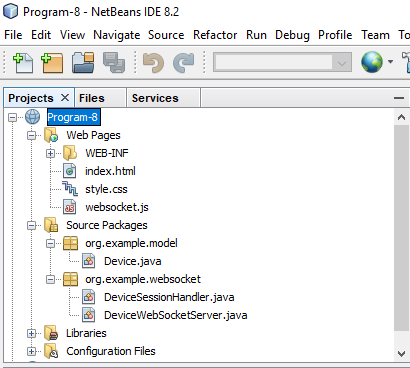
}

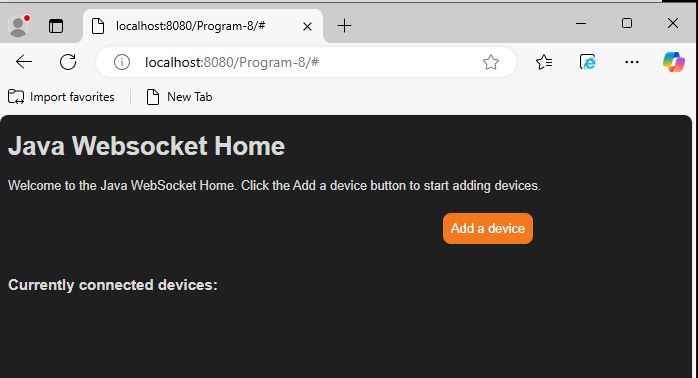
}

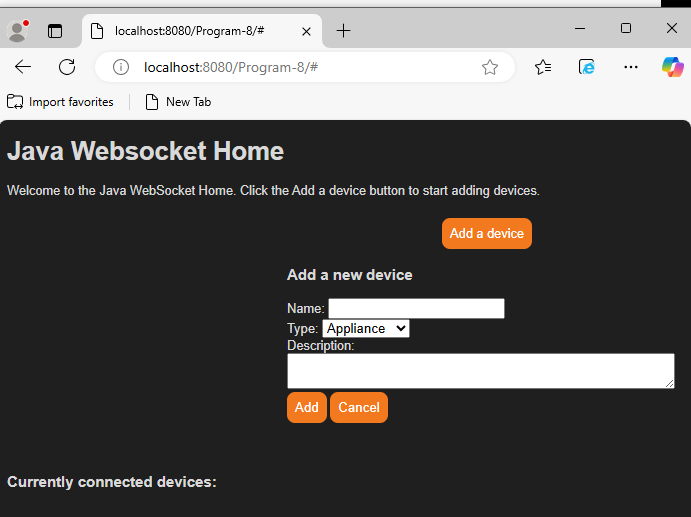
}

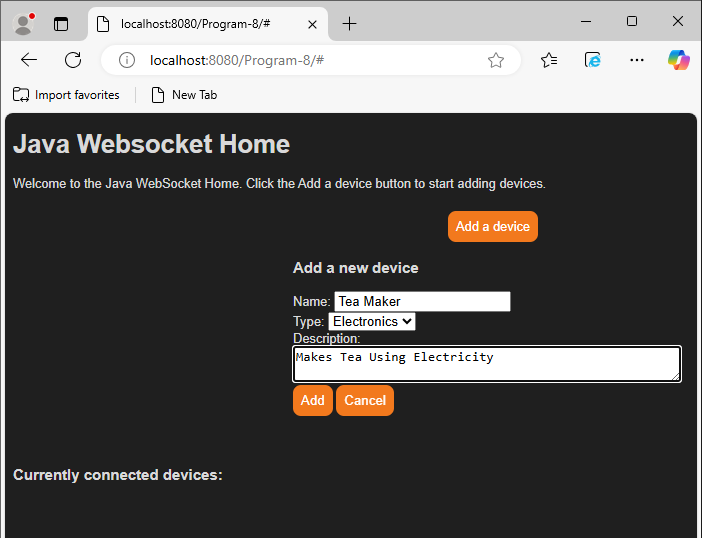
}

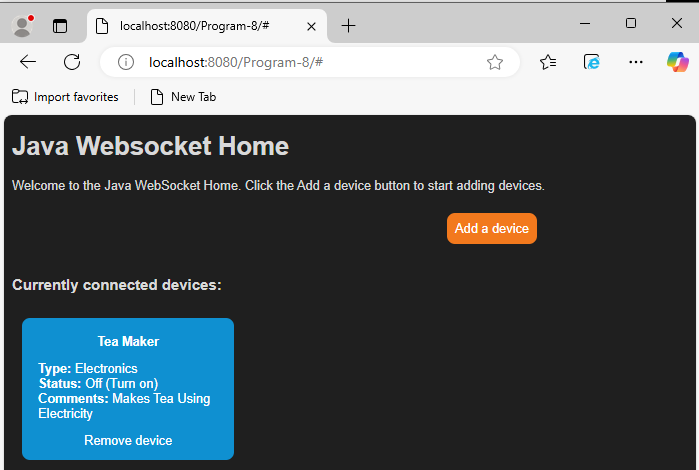
**OUTPUT:**

****

****







**EXPERIMENT-09:**

**Aim:**[**Write an application to demonstrate Session Bean & Entity Bean (persistence).**](#A9)

**Execution Steps:**

1. Open **NetBeans** and Follow**: JavaWeb > Web Application.**Provide the project name (e.g., Program-9), click **Next**, and then **Finish**.
2. Creating and Configuring Files
   * index.html: Right-click on the project, go to **New > HTML File**, and name it index.html Add the required code to this file and save it.
   * Employee: Right-click on the project, go to **New > Servlet**, and name it Employee. Check **Add Information to Deployment Descriptor** and click **Finish**.Add the necessary code to this servlet and save it.
   * Additional Servlets: Follow the same process to create the following servlet files:EmployeeServlet.java,EmployeeSessionBean.java
3. Building and Running the Project
   * Right-click on the project and select **Clean and Build** and then **Run**

**Source Code:**

**Employee.java:**

import java.io.Serializable;

public class Employee implements Serializable {

private Long id;

private String name;

private String position;

private double salary;

public Employee(Long id, String name, String position, double salary) {

this.id = id;

this.name = name;

this.position = position;

this.salary = salary;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getPosition() {

return position;

}

public void setPosition(String position) {

this.position = position;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}}

**EmployeeServlet.java:**

import javax.ejb.EJB;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.List;

@WebServlet("/EmployeeServlet")

public class EmployeeServlet extends HttpServlet {

@EJB

private EmployeeSessionBean employeeSessionBean;

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

List<Employee> employees = employeeSessionBean.getAllEmployees();

response.setContentType("text/html");

PrintWriter out = response.getWriter();

out.println("<html><body>");

out.println("<h2>Employee List</h2>");

out.println("<table border='1'>");

out.println("<tr><th>Name</th><th>Position</th><th>Salary</th></tr>");

for (Employee employee : employees) {

out.println("<tr>");

out.println("<td>" + employee.getName() + "</td>");

out.println("<td>" + employee.getPosition() + "</td>");

out.println("<td>" + employee.getSalary() + "</td>");

out.println("</tr>");

}

out.println("</table>");

out.println("</body></html>");

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

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

double salary = Double.parseDouble(request.getParameter("salary"));

employeeSessionBean.addEmployee(name, position, salary);

response.sendRedirect("EmployeeServlet");

}

}

**EmployeeSessionBean.java:**

import javax.ejb.Stateless;

import java.util.ArrayList;

import java.util.List;

@Stateless

public class EmployeeSessionBean {

private List<Employee> employeeList = new ArrayList<>();

private static Long idCounter = 1L;

public void addEmployee(String name, String position, double salary) {

Employee employee = new Employee(idCounter++, name, position, salary);

employeeList.add(employee);

}

public List<Employee> getAllEmployees() {

return employeeList;

}}

**index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Add Employee</title>

</head>

<body>

<h2>Add Employee</h2>

<form action="EmployeeServlet" method="POST">

NAME: <input type="text" name="name" required /><br><br/>

JOB ROLE: <input type="text" name="position" required /><br><br/>

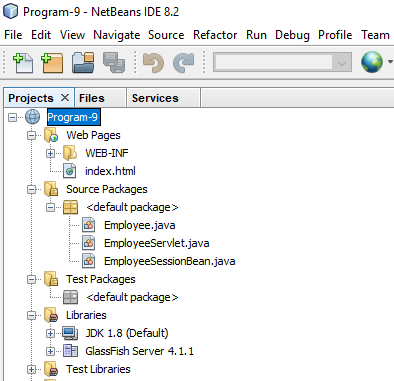
SALARY: <input type="text" name="salary" required /><br><br/>

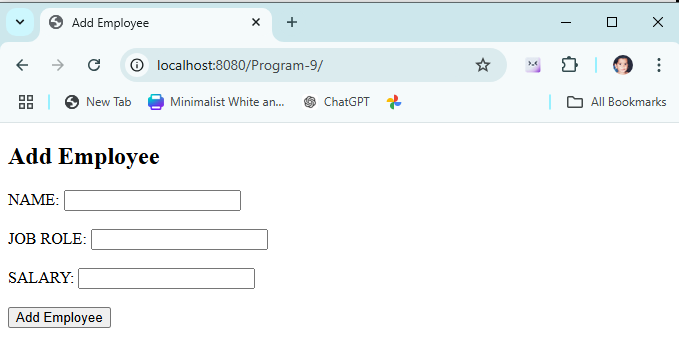
<input type="submit" value="Add Employee" />

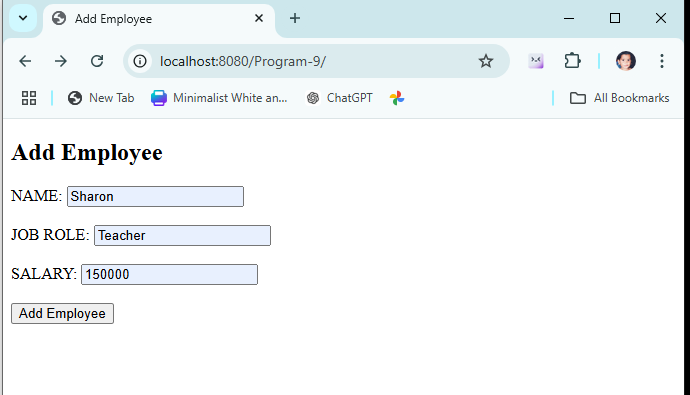
</form>

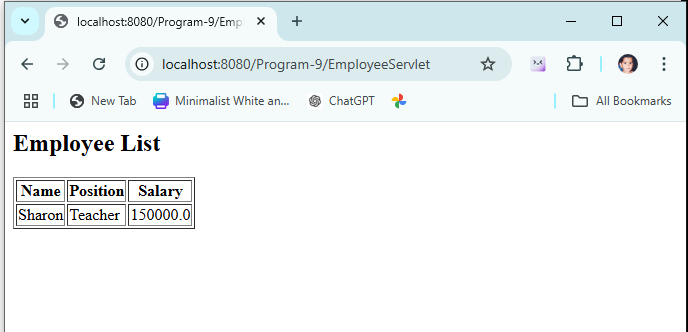
</body></html>

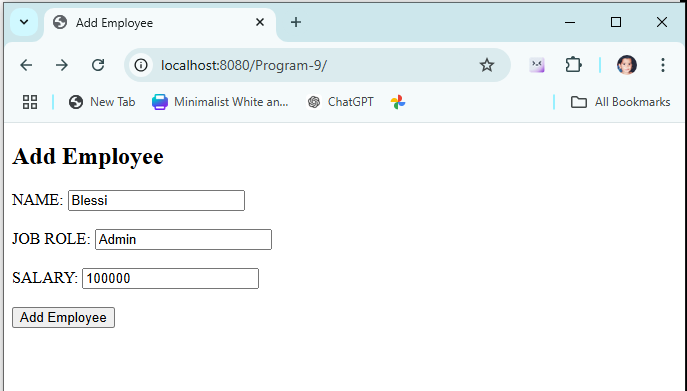
**OUTPUT:**

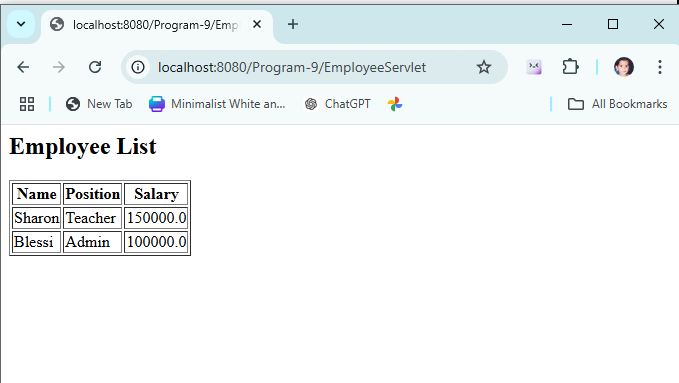
****

****



****

****

****