Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT619
B.Tech. IT, Sem: VI

Experiment – 01

Submitted By
Roll No.: IT008
Name: Akshar Bhalodia

Aim: Create a GUI based application which can be used as a telephone directory application. The telephone directory is stored as a database and has one table named telephoneDir. The telephoneDir database table stores three different information: telephone no., owner name, and owner address. The owner name is made of three parts: First name, middle name, and last name. The owner address is made of five parts: house no., address 1, address 2, area name, and city name. The application allows search facility. The search is possible using three different ways:

1. Search by telephone no.
2. Search by name (one of first name, middle name, and last name) with exactly match and part of name.
3. Search by address (one of address 1, address 2, area name, and city) with exactly match and part of address.

Code:

```
package phonedir;


import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;



public class Phonedir extends Frame implements ActionListener, ItemListener{



    static Connection con = null;

    static Statement stmt= null;

    static ResultSet rs = null;

    static PreparedStatement pstat=null;

    Panel pTop=new Panel();

    TextField tf1=new TextField(20);

    TextArea ta1=new TextArea(10, 100);

    Choice c1=new Choice();

    Choice c2=new Choice();

    Button srchbtn=new Button("Search");

    Label status=new Label("Records Found = 0");

    String query=null;


    public Phonedir(){
```

```
super("My Telephone Directory");
c1.add("Telephone No");
c1.add("Name");
c1.add("Address");
c1.addItemListener(this);
srchbtn.addActionListener(this);
addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e)
{
  dispose();
}
});
setLayout(new BorderLayout());
pTop.setLayout(new GridLayout(4, 2));
pTop.add(new Label("Search Option 1 : "));
pTop.add(c1);
pTop.add(new Label("Search Option 2 : "));
pTop.add(c2);
c2.setVisible(false);
pTop.add(new Label("Enter Text : "));
pTop.add(tf1);
pTop.add(new Label(""));
pTop.add(srchbtn);
add("North",pTop);
add("Center",ta1);
```

```
        add("South",status);
        ta1.setEditable(false);

        try{
           Class.forName("org.apache.derby.jdbc.ClientDriver");
           con=DriverManager.getConnection("jdbc:derby://localhost:1527/lab1","lab1","lab1");
        }
        catch(ClassNotFoundException e)
        {
           System.out.println(""+e.toString());
        }
        catch(SQLException se)
        {
           while(se!=null)
           {
              System.out.println(""+se.toString());
              se=se.getNextException();
           }
        }
     }
  public void itemStateChanged(ItemEvent ee)
  {
     String arg=ee.getItem().toString();
     if(arg.equals("Telephone No"))
     {
        c2.setVisible(false);
    }
    else if(arg.equals("Name"))
    {
        c2.removeAll();
        c2.add("First Name");
        c2.add("Middle Name");
        c2.add("Last Name");
        c2.setVisible(true);
    }
    else if(arg.equals("Address"))
    {
        c2.removeAll();
        c2.add("Area");
        c2.add("City");
        c2.setVisible(true);
    }
}
public void actionPerformed(ActionEvent ae)
{
    ta1.setText("Refreshed");
    query=new String("select * from Phonedir");
    int len=0;
    len=tf1.getText().toString().trim().length();
    try
```

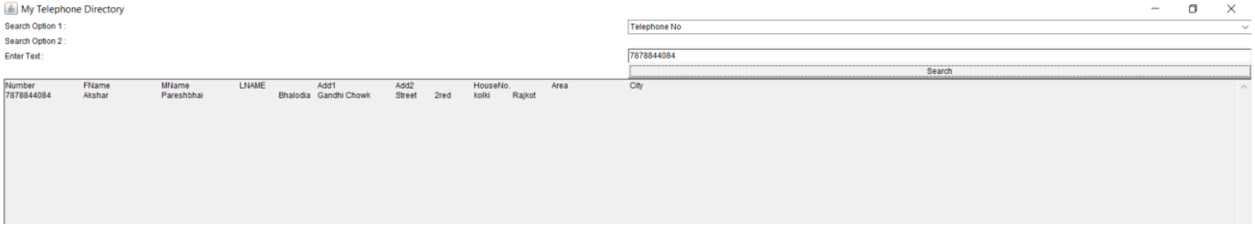
```
{
    if(c1.getSelectedItem().equals(("Telephone No")) && len>0)
    {
        query+=" where Number=?";
        pstat=con.prepareStatement(query);
        pstat.setString(1, tf1.getText().toString().trim());
    }
    else if(c1.getSelectedItem().equals(("Name")) && len>0)
    {
        if(c2.getSelectedItem().equals(("First Name")))
            query += " where fname=?";
        else if(c2.getSelectedItem().equals(("Middle Name")))
            query += " where mname=?";
        else if(c2.getSelectedItem().equals(("Last Name")))
            query += " where lname=?";
        pstat=con.prepareStatement(query);
        pstat.setString(1, tf1.getText().toString().trim());
    }
    else if(c1.getSelectedItem().equals(("Address")) && len>0)
    {
        if(c2.getSelectedItem().equals(("Area")))
            query += " where area=?";
        else if(c2.getSelectedItem().equals(("City")))
            query += " where city=?";
        pstat=con.prepareStatement(query);
        pstat.setString(1, tf1.getText().toString().trim());
    }
    else
    {
        pstat=con.prepareStatement(query);
    }
    try
    {
        System.out.println(query);
        rs=pstat.executeQuery();
    }
    catch(NullPointerException ne)
    {
        System.out.println("Text Null");
        ta1.setText("No Records Found.");
        status.setText("Records Found = 0");
    }
    if(rs!=null)

ta1.setText("Number\t\tFName\t\tMName\t\tLNAME\t\tAdd1\t\tAdd2\t\tHouseNo.\t\tArea\t\tCit
y\n");
    int count=0;
    while(rs.next())
    {
        ta1.append("" + rs.getString(1) + "\t");
        ta1.append("" + rs.getString(2) + "\t\t");
```

```
                ta1.append("" + rs.getString(3) + "\t\t");
                ta1.append("" + rs.getString(4) + "\t");
                ta1.append("" + rs.getString(5) + "\t");
                ta1.append("" + rs.getString(6) + "\t");
                ta1.append("" + rs.getString(7) + "\t");
                ta1.append("" + rs.getString(8) + "\t");
                ta1.append("" + rs.getString(9) + "\n");
                count++;
                status.setText("Records Found = " + count);
            }
        }
            catch(Exception ee){
                System.out.println("Exception " + ee);
            }


    }



        public static void main(String[] args) {
            // TODO code application logic here
            Frame dir = new Phonedir();
            dir.setVisible(true);
            dir.setSize(1300,600);
        }
```

```
}
```

Input/Output:

1.) Using Telephone Number



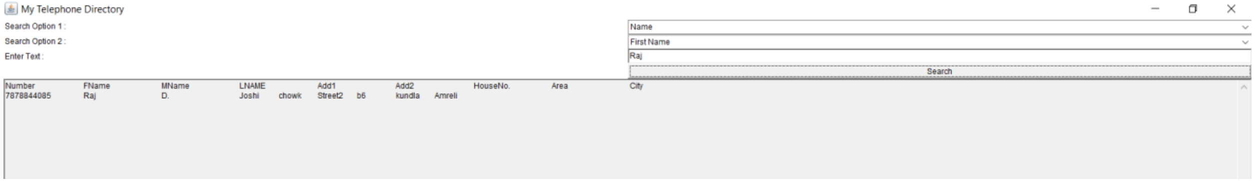
Search Option 1 : Telephone No

Search Option 2 :

Enter Text : 7878844084

```
Number      FName       MName       LNAME             Add1              Add2              HouseNo.          Area        City
7878844084  Akshar      Pareshbhai              Bhalodia  Gandhi Chowk  Street     2red   kolki      Rajkot
```

2.) Using Name



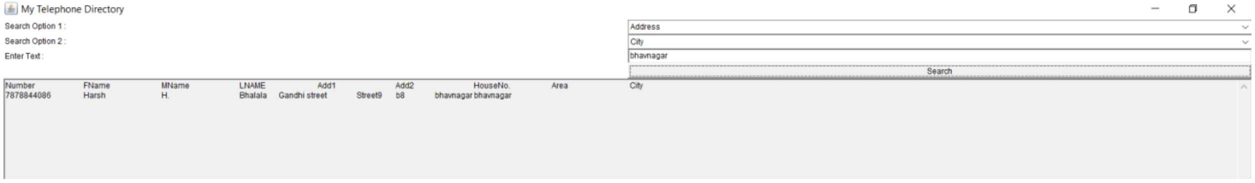
Search Option 1 : Name

Search Option 2 : First Name

Enter Text : Raj

```
Number      FName       MName       LNAME             Add1              Add2              HouseNo.          Area        City
7878844085  Raj         D.          Joshi     chowk   Street2    b6     kundla     Amreli
```

3.) Using Address



Search Option 1 : Address

Search Option 2 : City

Enter Text : bhavnagar

```
Number      FName       MName       LNAME             Add1              Add2              HouseNo.          Area        City
7878844086  Harsh       H.          Bhalala   Gandhi street     Street9   b8     bhavnagar bhavnagar
```

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT619
B.Tech. IT, Sem: VI

Experiment – 02

Submitted By
Roll No.: IT008
Name: Akshar Bhalodia

Aim: Create a GUI based application which can be used for telephone directory modification (administrator part for the above problem statement). The application allows two modification operations: create new telephone connection, and delete a telephone connection. The insert operation takes telephone no., name, and address as input parameters. The delete operation has verification step in which the user must perform the verification of the telephone connection which is about to be deleted. Once the verification is done, the application allows deleting the telephone connection. Design appropriate GUI to accommodate all stated features.

Code:

```
package phonedir;


import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

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

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;



public class Phonedirmul extends Frame implements ActionListener, ItemListener{



    static Connection con = null;

    static Statement stmt= null;

    static ResultSet rs = null;

    static PreparedStatement pstat=null;

    static PreparedStatement pstat1=null;

    static PreparedStatement pstat2=null;

    Panel pTop=new Panel();

    TextField tf0=new TextField(20);

    TextField tf1=new TextField(20);

    TextField tf2=new TextField(20);

    TextField tf3=new TextField(20);

    TextField tf4=new TextField(20);

    TextField tf5=new TextField(20);
    TextField tf6=new TextField(20);
    TextField tf7=new TextField(20);
    TextField tf8=new TextField(20);
    TextField tf9=new TextField(20);
    TextArea ta1=new TextArea(10, 100);
    Choice c0=new Choice();
    Choice c1=new Choice();
    Choice c2=new Choice();
    Button srchbtn=new Button("Execute");
    Label status=new Label("Records Found = 0");
    String query=null;
    String Insertquery=null;
    String Deletequery=null;

    public Phonedirmul(){
       super("My Telephone Directory lab2");
       c0.add("Search");
       c0.add("Insert");
       c0.add("Delete");
       c0.addItemListener(this);
       c1.add("Telephone No");
       c1.add("Name");
       c1.add("Address");
       c1.addItemListener(this);
       srchbtn.addActionListener(this);
```

```
addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e)
{
  dispose();
}
});
setLayout(new BorderLayout());
pTop.setLayout(new GridLayout(14, 2));
pTop.add(new Label("Operation : "));
pTop.add(c0);
pTop.add(new Label("Search Option 1 : "));
pTop.add(c1);
pTop.add(new Label("Search Option 2 : "));
pTop.add(c2);
c2.setVisible(false);
pTop.add(new Label("Enter Text : "));
pTop.add(tf0);


pTop.add(new Label("Enter Telephone Number : "));
pTop.add(tf1);
pTop.add(new Label("Enter First Name : "));
pTop.add(tf2);
pTop.add(new Label("Enter Middle Name : "));
pTop.add(tf3);
pTop.add(new Label("Enter Last Name : "));

pTop.add(tf4);

pTop.add(new Label("Enter Address 1 : "));

pTop.add(tf5);

pTop.add(new Label("Enter Address 2 : "));

pTop.add(tf6);

pTop.add(new Label("Enter House no. : "));

pTop.add(tf7);

pTop.add(new Label("Enter Area : "));

pTop.add(tf8);

pTop.add(new Label("Enter City : "));

pTop.add(tf9);



tf1.setVisible(false);

tf2.setVisible(false);

tf3.setVisible(false);

tf4.setVisible(false);

tf5.setVisible(false);

tf6.setVisible(false);

tf7.setVisible(false);

tf8.setVisible(false);

tf9.setVisible(false);


pTop.add(new Label(""));
```

```
        pTop.add(srchbtn);
        add("North",pTop);
        add("Center",ta1);
        add("South",status);
        ta1.setEditable(false);

        try{
          Class.forName("org.apache.derby.jdbc.ClientDriver");
          con=DriverManager.getConnection("jdbc:derby://localhost:1527/lab1","lab1","lab1");
        }
        catch(ClassNotFoundException e)
        {
          System.out.println(""+e.toString());
        }
        catch(SQLException se)
        {
          while(se!=null)
          {
            System.out.println(""+se.toString());
            se=se.getNextException();
          }
        }
      }
    public void itemStateChanged(ItemEvent ee)
    {
```

```
String arg=ee.getItem().toString();
if(arg.equals("Insert"))
{
   c1.setVisible(false);
   c2.setVisible(false);
   tf0.setVisible(false);
   tf1.setVisible(true);
   tf2.setVisible(true);
   tf3.setVisible(true);
   tf4.setVisible(true);
   tf5.setVisible(true);
   tf6.setVisible(true);
   tf7.setVisible(true);
   tf8.setVisible(true);
   tf9.setVisible(true);
}
else if(arg.equals("Search")){

   tf0.setVisible(true);
   tf1.setVisible(false);
   tf2.setVisible(false);
   tf3.setVisible(false);
   tf4.setVisible(false);
   tf5.setVisible(false);
   tf6.setVisible(false);
```

```
    tf7.setVisible(false);
    tf8.setVisible(false);
    tf9.setVisible(false);
    c1.setVisible(true);
if(arg.equals("Telephone No"))
{
    c2.setVisible(false);
}
else if(arg.equals("Name"))
{
    c2.removeAll();
    c2.add("First Name");
    c2.add("Middle Name");
    c2.add("Last Name");
    c2.setVisible(true);
}
else if(arg.equals("Address"))
{
    c2.removeAll();
    c2.add("Area");
    c2.add("City");
    c2.setVisible(true);
}
}
else{
        c1.setVisible(false);
        c2.setVisible(false);
        tf0.setVisible(false);
        tf1.setVisible(true);
        tf2.setVisible(false);
        tf3.setVisible(false);
        tf4.setVisible(false);
        tf5.setVisible(false);
        tf6.setVisible(false);
        tf7.setVisible(false);
        tf8.setVisible(false);
        tf9.setVisible(false);

    }
}

public void actionPerformed(ActionEvent ae)
{
    ta1.setText("Refreshed");
    query=new String("select * from phonedir");
    Insertquery=new String("insert into phonedir (NUMBER, FNAME, MNAME, LNAME,
ADD1, ADD2, HOUSENO, AREA, CITY)");
    Deletequery=new String("delete from phonedir where NUMBER=?");
    int len=0;
    len=tf0.getText().toString().trim().length();
```

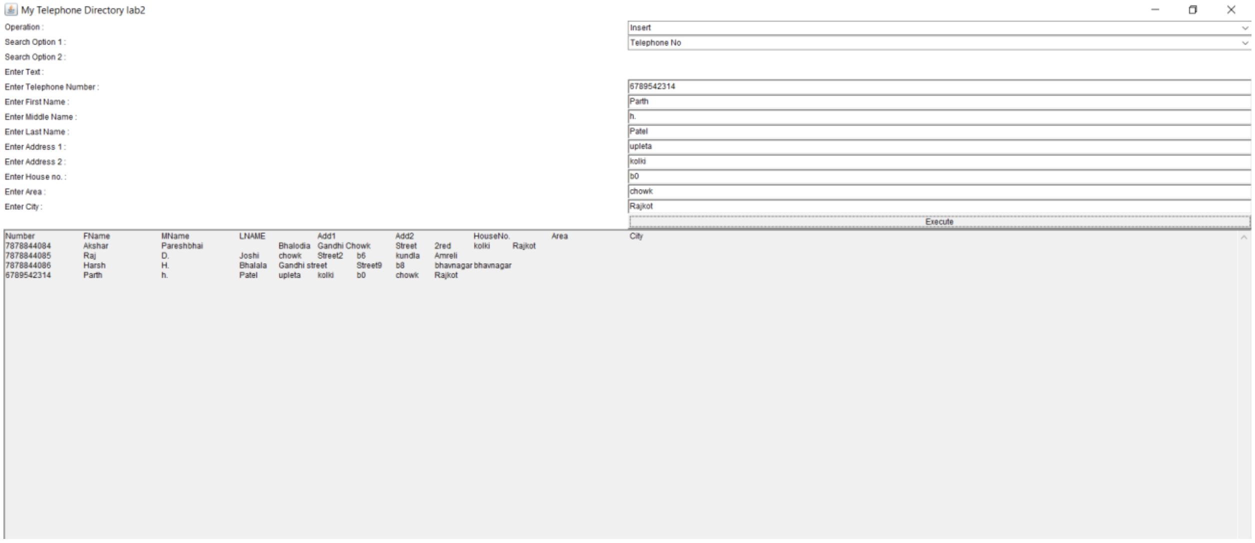
```
/*if(c0.getSelectedItem().equals(("Insert")) || c0.getSelectedItem().equals(("Delete"))){
    c1.setVisible(false);
    c2.setVisible(false);
}*/
try
{   c1.setVisible(true);
    if(c1.getSelectedItem().equals(("Telephone No")) && len>0)
    {
        query+=" where Number=?";
        pstat=con.prepareStatement(query);
        pstat.setString(1, tf0.getText().toString().trim());
    }
    else if(c1.getSelectedItem().equals(("Name")) && len>0)
    {
        if(c2.getSelectedItem().equals(("First Name")))
            query += " where fname=?";
        else if(c2.getSelectedItem().equals(("Middle Name")))
            query += " where mname=?";
        else if(c2.getSelectedItem().equals(("Last Name")))
            query += " where lname=?";
        pstat=con.prepareStatement(query);
        pstat.setString(1, tf0.getText().toString().trim());
    }
    else if(c1.getSelectedItem().equals(("Address")) && len>0)
    {
            if(c2.getSelectedItem().equals(("Area")))
                query += " where area=?";
            else if(c2.getSelectedItem().equals(("City")))
                query += " where city=?";
            pstat=con.prepareStatement(query);
            pstat.setString(1, tf1.getText().toString().trim());
        }
        else
        {
            pstat=con.prepareStatement(query);
        }
        try
        {
            System.out.println(query);
            rs=pstat.executeQuery();
        }
        catch(NullPointerException ne)
        {
            System.out.println("Text Null");
            ta1.setText("No Records Found.");
            status.setText("Records Found = 0");
        }
        if(rs!=null)
    ta1.setText("Number\t\tFName\t\tMName\t\tLNAME\t\tAdd1\t\tAdd2\t\tHouseNo.\t\tArea\t\tCity\n");
    int count=0;
    while(rs.next())
    {
        ta1.append("" + rs.getString(1) + "\t");
        ta1.append("" + rs.getString(2) + "\t\t");
        ta1.append("" + rs.getString(3) + "\t\t");
        ta1.append("" + rs.getString(4) + "\t");
        ta1.append("" + rs.getString(5) + "\t");
        ta1.append("" + rs.getString(6) + "\t");
        ta1.append("" + rs.getString(7) + "\t");
        ta1.append("" + rs.getString(8) + "\t");
        ta1.append("" + rs.getString(9) + "\n");
        count++;
        status.setText("Records Found = " + count);
    }
}
    catch(Exception ee){
        System.out.println("Exception " + ee);
    }
if(c0.getSelectedItem().equals(("Insert"))){
        Insertquery+=" values (?,?,?,?,?,?,?,?,?)";
    try{
        pstat1=con.prepareStatement(Insertquery);
        pstat1.setString(1, tf1.getText().toString().trim());
        pstat1.setString(2, tf2.getText().toString().trim());
    pstat1.setString(3, tf3.getText().toString().trim());
    pstat1.setString(4, tf4.getText().toString().trim());
    pstat1.setString(5, tf5.getText().toString().trim());
    pstat1.setString(6, tf6.getText().toString().trim());
    pstat1.setString(7, tf7.getText().toString().trim());
    pstat1.setString(8, tf8.getText().toString().trim());
    pstat1.setString(9, tf9.getText().toString().trim());

}catch(Exception ee)
{
    System.out.println("Exception ");
}
try
{
    System.out.println(Insertquery);
    pstat1.executeUpdate();
}
catch(SQLException se)
{
    System.out.println("SQL Exception");
}
catch(NullPointerException ne)
{
    System.out.println("NullPointer Exception");
}
```

```
}
        if(c0.getSelectedItem().equals(("Delete"))){
        try{
           pstat2=con.prepareStatement(Deletequery);
           pstat2.setString(1, tf1.getText().toString().trim());

        }catch(Exception ee)
        {
           System.out.println("Exception ");
        }
        try
        {
           System.out.println(Deletequery);
           pstat2.executeUpdate();
        }
        catch(SQLException se)
        {
           System.out.println("SQL Exception");
        }
        catch(NullPointerException ne)
        {
           System.out.println("NullPointer Exception");
        }
        }
}
```

```
    public static void main(String[] args) {
        // TODO code application logic here
        Frame dir = new Phonedirmul();
        dir.setVisible(true);
        dir.setSize(1300,600);
    }
}
```

Input/Output:

1.) Insert


My Telephone Directory lab2

Operation : Insert

Search Option 1 : Telephone No

Search Option 2 :

Enter Text :

Enter Telephone Number : 6789542314

Enter First Name : Parth

Enter Middle Name : h.

Enter Last Name : Patel

Enter Address 1 : upleta

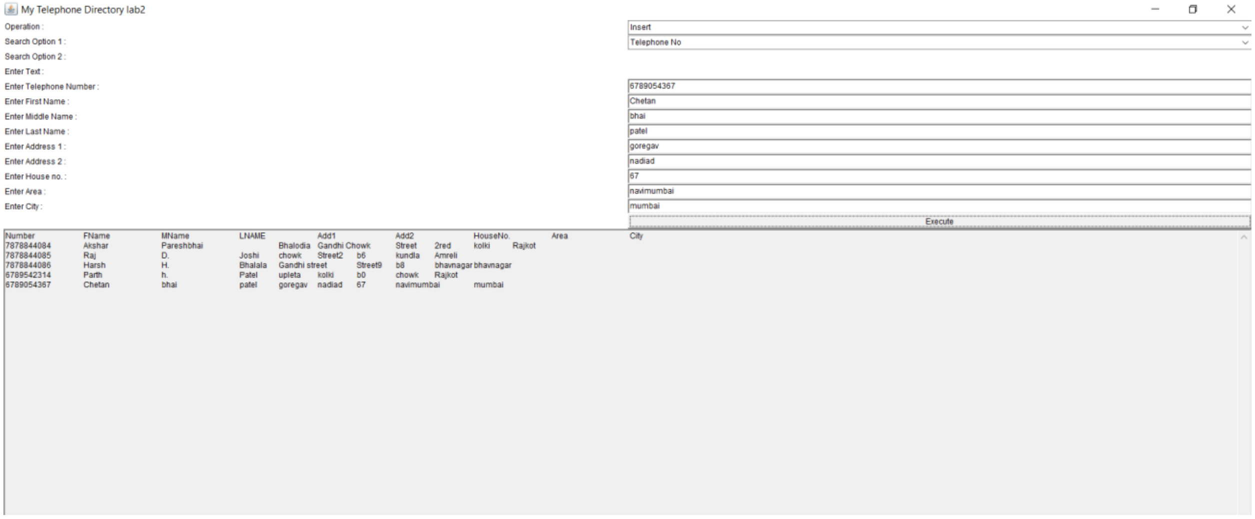
Enter Address 2 : kolki

Enter House no. : b0

Enter Area : chowk

Enter City : Rajkot

```
Number              FName               MName               LNAME               Add1                Add2                HouseNo.            Area                City
7878844084          Akshar              Pareshbhai                    Bhalodia  Gandhi Chowk        Street    2red      kolki     Rajkot
7878844085          Raj                 D.                  Joshi     chowk     Street2   b6        kundla    Amreli
7878844086          Harsh               H.                  Bhalala   Gandhi street       Street9   b8        bhavnagar bhavnagar
6789542314          Parth               h.                  Patel     upleta    kolki     b0        chowk     Rajkot
```



My Telephone Directory lab2

Operation : Insert

Search Option 1 : Telephone No

Search Option 2 :

Enter Text :

Enter Telephone Number : 6789054367

Enter First Name : Chetan

Enter Middle Name : bhai

Enter Last Name : patel

Enter Address 1 : goregav

Enter Address 2 : nadiad

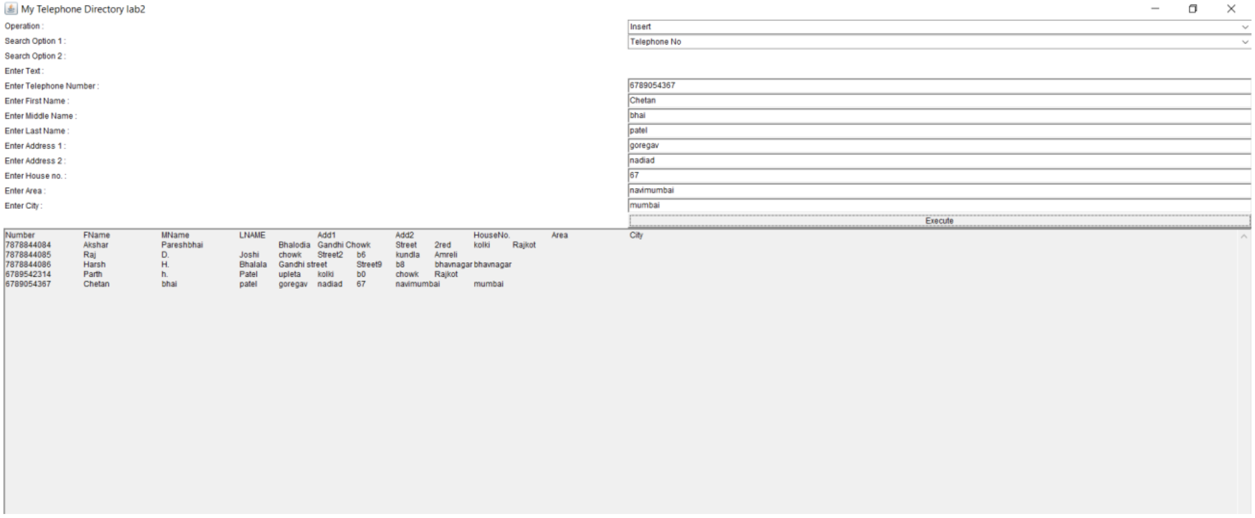
Enter House no. : 67

Enter Area : navimumbai

Enter City : mumbai

```
Number      FName       MName       LNAME       Add1        Add2        HouseNo.    Area        City
7878844084  Akshar      Pareshbhai        Bhalodia  Gandhi Chowk  Street  2red  kolki  Rajkot
7878844085  Raj         D.          Joshi  chowk  Street2  b6  kundla  Amreli
7878844086  Harsh       H.          Bhalala  Gandhi street  Street9  b8  bhavnagar bhavnagar
6789542314  Parth       h.          Patel  upleta  kolki  b0  chowk  Rajkot
6789054367  Chetan      bhai        patel  goregav  nadiad  67  navimumbai  mumbai
```

2.) Before Delete



My Telephone Directory lab2

Operation : Insert

Search Option 1 : Telephone No

Search Option 2 :

Enter Text :

Enter Telephone Number : 6789054367

Enter First Name : Chetan

Enter Middle Name : bhai

Enter Last Name : patel

Enter Address 1 : goregav

Enter Address 2 : nadiad

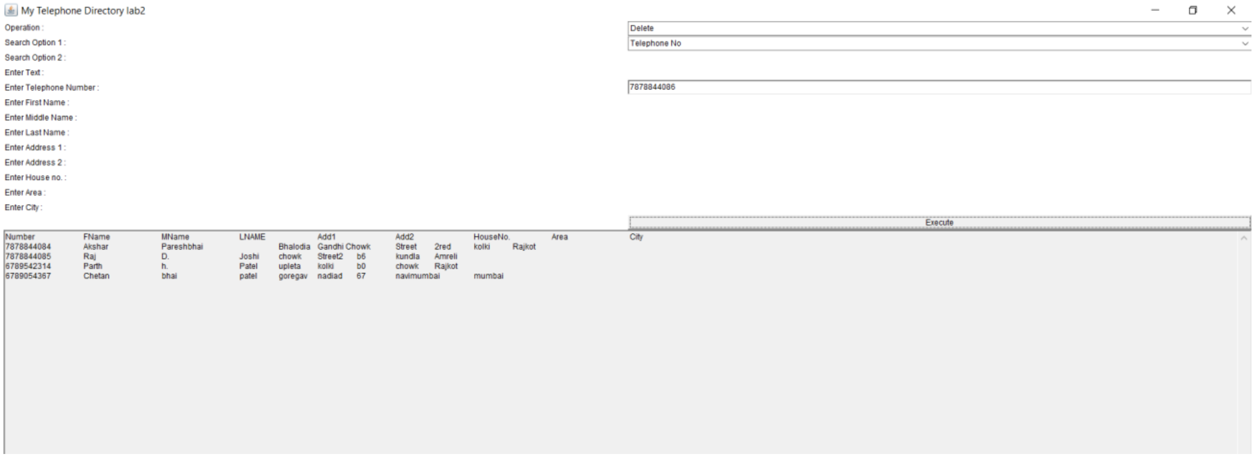
Enter House no. : 67

Enter Area : navimumbai

Enter City : mumbai

```
Number      FName       MName       LNAME       Add1        Add2        HouseNo.    Area        City
7878844084  Akshar      Pareshbhai        Bhalodia  Gandhi Chowk  Street  2red  kolki  Rajkot
7878844085  Raj         D.          Joshi  chowk  Street2  b6  kundla  Amreli
7878844086  Harsh       H.          Bhalala  Gandhi street  Street9  b8  bhavnagar bhavnagar
6789542314  Parth       h.          Patel  upleta  kolki  b0  chowk  Rajkot
6789054367  Chetan      bhai        patel  goregav  nadiad  67  navimumbai  mumbai
```

3.) After Delete



Operation : Delete

Search Option 1 : Telephone No

Search Option 2 :

Enter Text :

Enter Telephone Number : 7878844086

Enter First Name :

Enter Middle Name :

Enter Last Name :

Enter Address 1 :

Enter Address 2 :

Enter House no. :

Enter Area :

Enter City :

```
Number      FName       MName       LNAME              Add1               Add2               HouseNo.          Area        City
7878844084  Akshar      Pareshbhai         Bhalodia    Gandhi Chowk       Street      2red   kolki      Rajkot
7878844085  Raj         D.          Joshi  chowk       Street2     b6     kundla      Amreli
6789542314  Parth       h.          Patel  upleta      kolki       b0     chowk       Rajkot
6789054367  Chetan      bhai        patel  goregav     nadiad      67     navimumbai         mumbai
```

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT619
B.Tech. IT, Sem: VI

Experiment – 03

Submitted By
Roll No.: IT008
Name: Akshar Bhalodia

Aim: Create user registration functionality for student to get registered with exam- result section. The registration page takes following information from user: user ID, password, confirm password, full name, semester, roll no, email-id, and contact number. The registration servlet checks uniqueness of user ID among all users and if found unique then only stores registration information in database.

Code:

Index.html

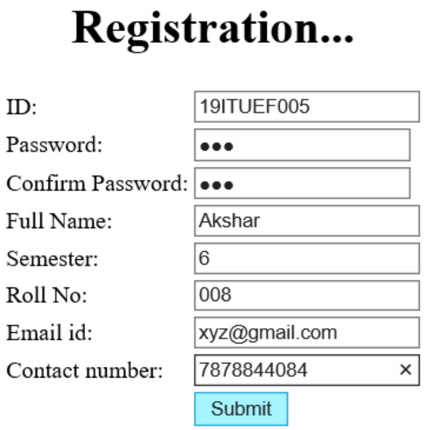
```
<!DOCTYPE html>
<html>
    <head>
        <title>Registration Form</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <h1 align="center">Registration...</h1>
```

```
<form action="Registered" method="post">
    <table align="center">
        <tr>
            <td>ID: </td>
            <td><input type="text" name="sid" id="sid"></td>
        </tr>
        <tr>
            <td>Password: </td>
            <td><input type="password" name="password" id="spswd"></td>
        </tr>
        <tr>
            <td>Confirm Password: </td>
            <td><input type="password" name="cpassword" id="cspswd"></td>
        </tr>
        <tr>
            <td>Full Name: </td>
            <td><input type="text" name="fname" id="fname"></td>
        </tr>
        <tr>
            <td>Semester: </td>
            <td><input type="text" name="sem" id="sem"></td>
        </tr>
        <tr>
            <td>Roll No: </td>
            <td><input type="text" name="rollno" id="rollno"></td>
        </tr>
        <tr>
          <td>Email id: </td>
          <td><input type="email" name="email" id="email"></td>
        </tr>
        <tr>
          <td>Contact number: </td>
          <td><input type="text" name="contact" id="contact"></td>
        </tr>
        <tr>
          <td></td>
          <td><input type="submit" value="Submit" id="submit"></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

Registered.java

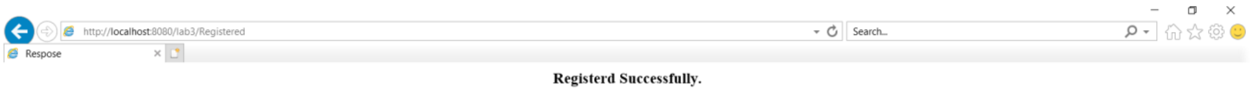
```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.*;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.*;

public class Registered extends HttpServlet
{

    Connection con;
    PreparedStatement ps;

    @Override
    public void init(ServletConfig config)
    {
        try
        {
            String driver = "org.apache.derby.jdbc.ClientDriver";
            String url = "jdbc:derby://localhost:1527/exp3";

            Class.forName(driver);
            con = DriverManager.getConnection(url,"exp3","exp3");
            ps = con.prepareStatement("insert into EXP3.DATA VALUES(?,?,?,?,?,?,?,?)");
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }


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

    {

        try (PrintWriter out = response.getWriter())

        {

            out.println("<title>Respose</title>");

            String id = request.getParameter("sid");

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

            String cpswd = request.getParameter("cpassword");

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

            int sem = Integer.parseInt(request.getParameter("sem"));

            int rollno = Integer.parseInt(request.getParameter("rollno"));

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

            Long number = Long.parseLong(request.getParameter("contact"));


            Statement stm = con.createStatement();

            ResultSet rs = stm.executeQuery("Select id from EXP3.DATA WHERE ID = '"+id+"'");


            if(rs.next())

            {

                    out.println("<h2 style='color:red;' align='center'>Sorry! User allredy availabel for
this id.</h2>");

            }

            else
        {
            ps.setString(1,id);
            ps.setString(2,pswd);
            ps.setString(3,cpswd);
            ps.setString(4,name);
            ps.setInt(5,sem);
            ps.setInt(6,rollno);
            ps.setString(7,email);
            ps.setLong(8,number);
            ps.executeUpdate();
            out.println("<h3 align='center'>Registerd Successfully.</h3>");
        }
        stm.close();
    }
    catch(SQLException s)
    {
        s.printStackTrace();
    }
}

@Override
public void destroy()
{
    try
    {
        con.close();
        ps.close();
    }
    catch(SQLException s)
    {
        s.printStackTrace();
    }

}

@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";
    }

}
```

Input/Output:



Registration...

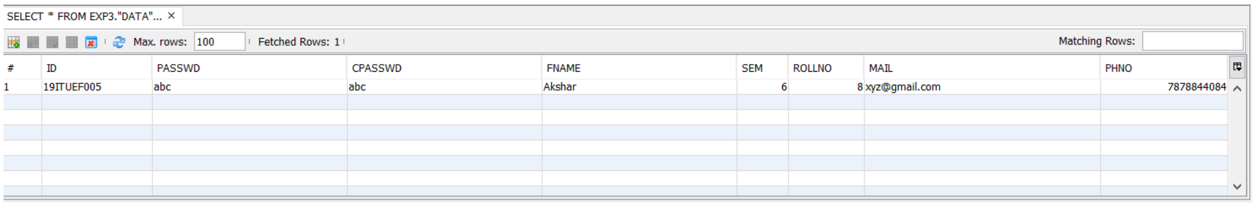
ID:	19ITUEF005
Password:	●●●
Confirm Password:	●●●
Full Name:	Akshar
Semester:	6
Roll No:	008
Email id:	xyz@gmail.com
Contact number:	7878844084



http://localhost:8080/lab3/Registered

Registerd Successfully.

Database Table After Registration:



SELECT * FROM EXP3."DATA"...

Max. rows: 100 Fetched Rows: 1 Matching Rows:

#	ID	PASSWD	CPASSWD	FNAME	SEM	ROLLNO	MAIL	PHNO
1	19ITUEF005	abc	abc	Akshar	6	8	xyz@gmail.com	7878844084

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT619
B.Tech. IT, Sem: VI

Experiment – 04

Submitted By
Roll No.: IT008
Name: Akshar Bhalodia

Aim: Create login and view result functionality with the session management. The login servlet logons the user with the exam-result section and allows access of viewing his/her exam-result

Code:

web.xml :

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

<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-
app_3_1.xsd">

    <servlet>

        <servlet-name>index</servlet-name>

        <servlet-class>com.index</servlet-class>

    </servlet>

    <servlet>

        <servlet-name>DisplayResult</servlet-name>

        <servlet-class>com.DisplayResult</servlet-class>

    </servlet>

    <servlet-mapping>

        <servlet-name>index</servlet-name>
        <url-pattern>/index</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>DisplayResult</servlet-name>
        <url-pattern>/DisplayResult</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
</web-app>
```

login.jsp :

```
<%@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="index" method="post">
            <br/>
            Username    <input type="text" name="uname"/><br/>
            Password    <input type="Password" name="pass"><br/>
            <input type="submit" name="login">
        </form>
    </body>
</html>
```

index.java :

```
package com;

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 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 index extends HttpServlet {
    Connection con = null;

    Statement st = null;

    ResultSet rs = null;

    String query = "select * from login";




    protected void processRequest(HttpServletRequest request, HttpServletResponse response)

            throws ServletException, IOException {

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

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

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


        boolean status = false;

        String userid = null;

        String name, pass, id;


        try {

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

            con= DriverManager.getConnection("jdbc:derby://localhost:1527/exp4", "exp4",
"exp4");

        } catch (NullPointerException ne) {

            System.out.println("Not Connected");

        } catch (ClassNotFoundException ex) {

            System.out.println("error class");
```

```
} catch (SQLException se) {
    System.out.println("error sql");
}



HttpSession s1=request.getSession();
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 index</title>");
    out.println("</head>");
    out.println("<body>");
    try {
        st = con.createStatement();
        rs = st.executeQuery(query);
        while (rs.next()) {
            id = rs.getString(3);
            name = rs.getString(1);
            pass = rs.getString(2);
            if (nm.equals(name) && ps.equals(pass)) {
                status = true;
                out.println("<h4>Login Successful<h4>");
                out.println("<h4>Welcome " + name + "<h4>");
```

```
                        HttpSession hs = request.getSession();

                        hs.setAttribute("ID", id);

                        break;

                    }

                }

                if (status == true) {

                    out.println("<a href=DisplayResult>Click here for Result</a >");

                } else {

                    out.println("<h4>Invalid Username or password<h4>");

                }

                st.close();

                con.close();

            } catch (SQLException se) {

                System.out.println("error sql 2");

            }

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

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

        }

    }


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

    /**

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

     *
```

```
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}
```

```
    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

DisplayResult.java :

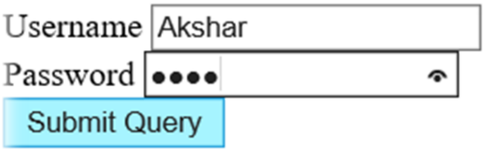
```
package com;

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.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;


public class DisplayResult extends HttpServlet {



    Connection con = null;

    PreparedStatement pst = null;

    ResultSet rs = null;

    String query = "select * from result where ID=?";

    HttpSession hs = null;

    String ID;




    protected void processRequest(HttpServletRequest request, HttpServletResponse response)

            throws ServletException, IOException {

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


        hs = request.getSession();

        try {

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

            con= DriverManager.getConnection("jdbc:derby://localhost:1527/exp4", "exp4",
"exp4");

        } catch (ClassNotFoundException ex) {

            System.out.println(ex);
```

```
} catch (SQLException se) {
    System.out.println(se);
}

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 DisplayResult</title>");
    out.println("</head>");
    out.println("<body>");
    try {
        ID = (String) hs.getAttribute("ID");
        pst = con.prepareStatement(query);
        pst.setString(1, ID);
        rs = pst.executeQuery();
        if (rs.next()) {
            int total = rs.getInt(2) + rs.getInt(3);
            double per = (double) total / 72;
            out.println("<h1>Your Result</h1>");
            out.println("<h4>Your ID : " + rs.getString(1) + "<h4>");
            out.println("<h4>AJT Marks : " + rs.getInt(2) + "<h4>");
            out.println("<h4>DPAF Marks : " + rs.getInt(3) + "<h4>");
            out.println("<h4>Total Marks : " + total + "<h4>");
```

```
                out.println("<h4>Percentage : " + String.format("%.2f",per * 100) + "%" + "<h4>");
            }
            pst.close();
            con.close();
        } catch (SQLException se) {
            System.out.println(se);
        }
        out.println("</body>");
        out.println("</html>");
        out.close();
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
```

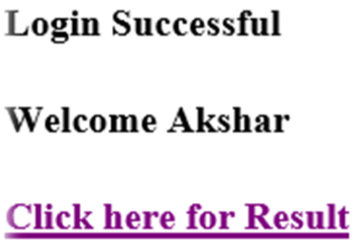
```
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

Input/Output:



Username Akshar

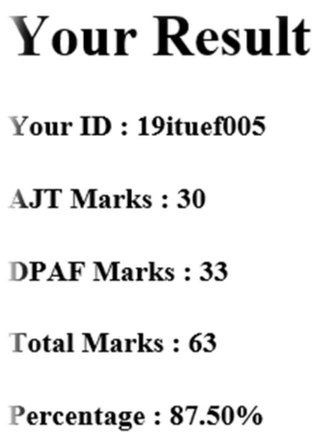
Password ●●●●



Login Successful

Welcome Akshar

Click here for Result



Your Result

Your ID : 19ituef005

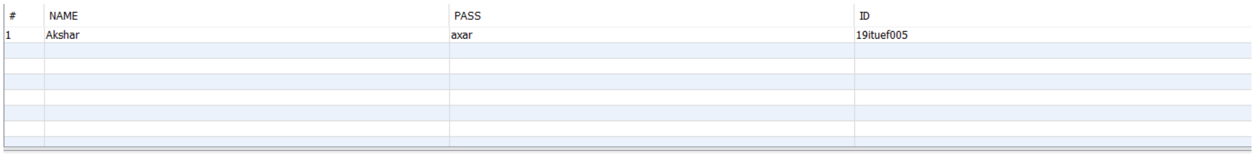
AJT Marks : 30

DPAF Marks : 33

Total Marks : 63

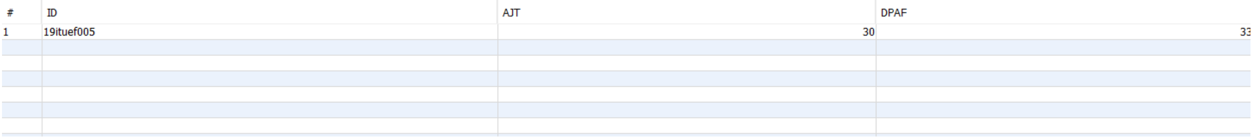
Percentage : 87.50%

Login table:



#	NAME	PASS	ID
1	Akshar	axar	19ituef005

Result table:



#	ID	AJT	DPAF
1	19ituef005	30	33

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT619
B.Tech. IT, Sem: VI

Experiment – 05

Submitted By
Roll No.: IT008
Name: Akshar Bhalodia

Aim: Write code for implementation of the two filters, Log Filter and Authentication Filter, in filter chain. Client calls the Log Filter. The Log filter logs the time of arrival of request and IP address of the client. The Log filter forwards the request to Authentication Filter. The authentication filter authenticates the client and allow to access the targeted servlet.

Code:

Index.html

```
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="TestServlet" method="POST">
            User Name: <input type="text" name="uname"><br/>
            Password:  <input type="text" name="passwd"><br/>
            <input type="submit" name="Submit">
        </form>
    </body>
```

```
</html>
```

TestServlet.java

```
package mypack;

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 TestServlet 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 TestServlet</title>");
            out.println("</head>");
            out.println("<body>");

            String username = request.getParameter("uname");
            out.println("<b> Welcome User:"+username+"</b>");



            out.println("<h1>Servlet TestServlet at " + request.getContextPath() + "</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";
    }
}
```

AuthenticationFilter

```
package mypack;

import java.io.IOException;
import java.io.PrintStream;
import java.io.PrintWriter;
import java.io.StringWriter;
import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;



public class AuthenticationFilter implements Filter {

    private static final boolean debug = true;



    private FilterConfig filterConfig = null;

    public AuthenticationFilter() {
    }

    private void doBeforeProcessing(ServletRequest request, ServletResponse response)
            throws IOException, ServletException {
        if (debug) {
            log("AuthenticationFilter:DoBeforeProcessing");
        }



    }

    private void doAfterProcessing(ServletRequest request, ServletResponse response)
```

```
        throws IOException, ServletException {
    if (debug) {
        log("AuthenticationFilter:DoAfterProcessing");
    }



}



public void doFilter(ServletRequest request, ServletResponse response,
        FilterChain chain)
        throws IOException, ServletException {

    if (debug) {
        log("AuthenticationFilter:doFilter()");
    }

    doBeforeProcessing(request, response);

    Throwable problem = null;
    try {
        System.out.println("----AuthenticationFilter--------\n");
        HttpServletRequest req = (HttpServletRequest) request;
        HttpServletResponse res = (HttpServletResponse) response;
        String uname = req.getParameter("uname");
    String passwd = req.getParameter("passwd");
    if (uname.equals("Akshar") && passwd.equals("abc")) {
        System.out.println("Success");
        chain.doFilter(request, response);
    } else {
        System.out.println("Failure");
        res.sendRedirect("index.html");
    }

    chain.doFilter(request, response);
} catch (Throwable t) {

    problem = t;
    t.printStackTrace();
}

doAfterProcessing(request, response);

if (problem != null) {
    if (problem instanceof ServletException) {
        throw (ServletException) problem;
    }
    if (problem instanceof IOException) {
        throw (IOException) problem;
    }
```

}

```
        sendProcessingError(problem, response);
    }
}



public FilterConfig getFilterConfig() {
    return (this.filterConfig);
}

public void setFilterConfig(FilterConfig filterConfig) {
    this.filterConfig = filterConfig;
}

public void destroy() {
}



public void init(FilterConfig filterConfig) {
    this.filterConfig = filterConfig;
    if (filterConfig != null) {
        if (debug) {
            log("AuthenticationFilter:Initializing filter");
        }
    }
}
```

```
@Override
public String toString() {
    if (filterConfig == null) {
        return ("AuthenticationFilter()");
    }
    StringBuffer sb = new StringBuffer("AuthenticationFilter(");
    sb.append(filterConfig);
    sb.append(")");
    return (sb.toString());
}

private void sendProcessingError(Throwable t, ServletResponse response) {
    String stackTrace = getStackTrace(t);

    if (stackTrace != null && !stackTrace.equals("")) {
        try {
            response.setContentType("text/html");
            PrintStream ps = new PrintStream(response.getOutputStream());
            PrintWriter pw = new PrintWriter(ps);
            pw.print("<html>\n<head>\n<title>Error</title>\n</head>\n<body>\n");



            pw.print("<h1>The resource did not process correctly</h1>\n<pre>\n");
            pw.print(stackTrace);
            pw.print("</pre></body>\n</html>");
            pw.close();
            ps.close();
            response.getOutputStream().close();
        } catch (Exception ex) {
        }
    } else {
        try {
            PrintStream ps = new PrintStream(response.getOutputStream());
            t.printStackTrace(ps);
            ps.close();
            response.getOutputStream().close();
        } catch (Exception ex) {
        }
    }
}

public static String getStackTrace(Throwable t) {
    String stackTrace = null;
    try {
        StringWriter sw = new StringWriter();
        PrintWriter pw = new PrintWriter(sw);
        t.printStackTrace(pw);
        pw.close();
```

```
            sw.close();
            stackTrace = sw.getBuffer().toString();
        } catch (Exception ex) {
        }
        return stackTrace;
    }

    public void log(String msg) {
        filterConfig.getServletContext().log(msg);
    }

}
```

LogFilter

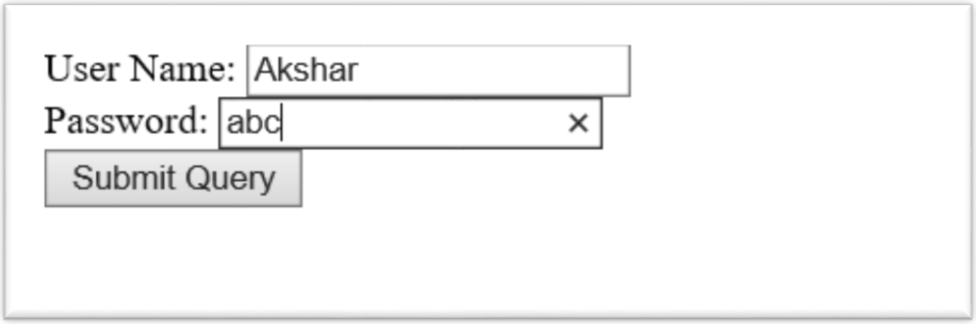
```
package mypack;

import java.io.IOException;
import java.io.PrintStream;
import java.io.PrintWriter;
import java.io.StringWriter;
import java.util.Date;
import javax.servlet.Filter;
import javax.servlet.FilterChain;
```

```
}

import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequest;


public class LogFilter implements Filter {

    private static final boolean debug = true;


    private FilterConfig filterConfig = null;

    public LogFilter() {
    }

    private void doBeforeProcessing(ServletRequest request, ServletResponse response)
            throws IOException, ServletException {
        if (debug) {
            log("LogFilter:DoBeforeProcessing");
        }


    }
```

```
private void doAfterProcessing(ServletRequest request, ServletResponse response)
        throws IOException, ServletException {
    if (debug) {
        log("LogFilter:DoAfterProcessing");
    }



}



public void doFilter(ServletRequest request, ServletResponse response,
        FilterChain chain)
        throws IOException, ServletException {

    if (debug) {
        log("LogFilter:doFilter()");
    }

    doBeforeProcessing(request, response);

    Throwable problem = null;
    try {
        System.out.println("----LogFilter--------\n");
        System.out.println("Logging Date: " + new Date());
        chain.doFilter(request, response);
    } catch (Throwable t) {

        problem = t;
        t.printStackTrace();
    }

    doAfterProcessing(request, response);


    if (problem != null) {
        if (problem instanceof ServletException) {
            throw (ServletException) problem;
        }
        if (problem instanceof IOException) {
            throw (IOException) problem;
        }
        sendProcessingError(problem, response);
    }
}


public FilterConfig getFilterConfig() {
    return (this.filterConfig);
```

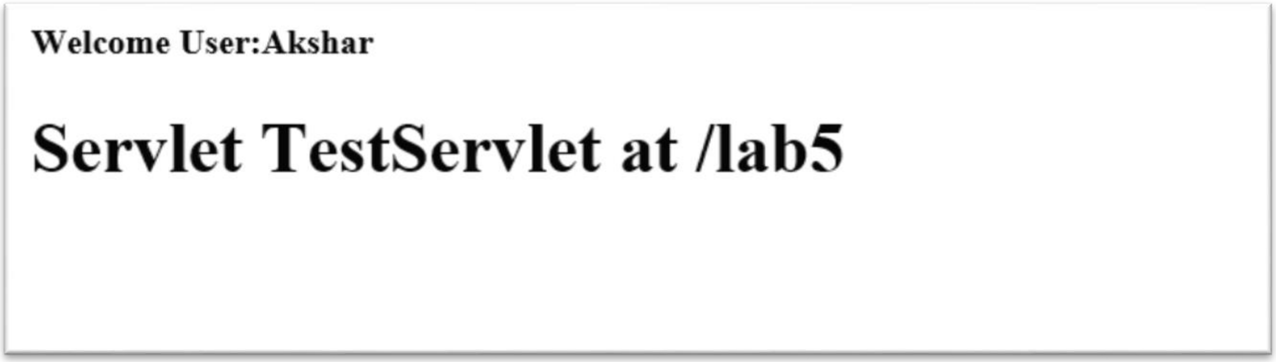
```
}



public void setFilterConfig(FilterConfig filterConfig) {
    this.filterConfig = filterConfig;
}



public void destroy() {
}



public void init(FilterConfig filterConfig) {
    this.filterConfig = filterConfig;
    if (filterConfig != null) {
        if (debug) {
            log("LogFilter:Initializing filter");
        }
    }
}



@Override
public String toString() {
    if (filterConfig == null) {
        return ("LogFilter()");
    }
    StringBuffer sb = new StringBuffer("LogFilter(");
    sb.append(filterConfig);
    sb.append(")");
    return (sb.toString());
}

private void sendProcessingError(Throwable t, ServletResponse response) {
    String stackTrace = getStackTrace(t);

    if (stackTrace != null && !stackTrace.equals("")) {
        try {
            response.setContentType("text/html");
            PrintStream ps = new PrintStream(response.getOutputStream());
            PrintWriter pw = new PrintWriter(ps);
            pw.print("<html>\n<head>\n<title>Error</title>\n</head>\n<body>\n");



            pw.print("<h1>The resource did not process correctly</h1>\n<pre>\n");
            pw.print(stackTrace);
            pw.print("</pre></body>\n</html>");
            pw.close();
            ps.close();
            response.getOutputStream().close();
        } catch (Exception ex) {
        }
    } else {
        try {
            PrintStream ps = new PrintStream(response.getOutputStream());
            t.printStackTrace(ps);
            ps.close();
            response.getOutputStream().close();
        } catch (Exception ex) {
        }
    }
}

public static String getStackTrace(Throwable t) {
    String stackTrace = null;
    try {
        StringWriter sw = new StringWriter();
        PrintWriter pw = new PrintWriter(sw);
        t.printStackTrace(pw);
        pw.close();
        sw.close();
        stackTrace = sw.getBuffer().toString();
    } catch (Exception ex) {
    }
    return stackTrace;
    }


    public void log(String msg) {
        filterConfig.getServletContext().log(msg);
    }


}
```

Input/Output:



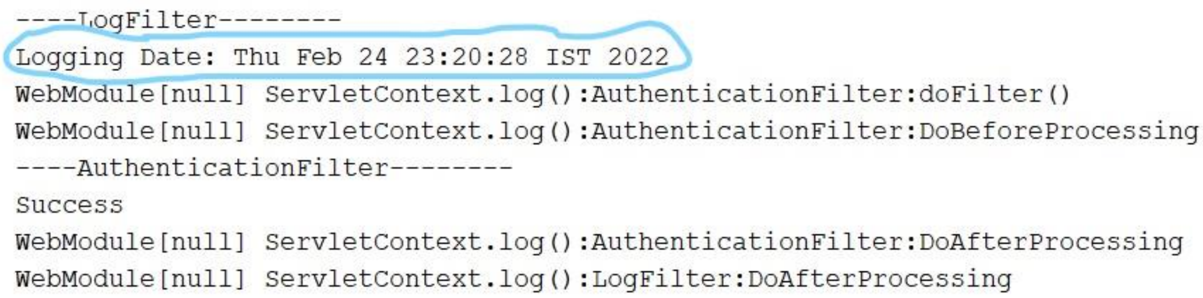
User Name: Akshar

Password: abc



Welcome User:Akshar

Servlet TestServlet at /lab5



```
----LogFilter--------
Logging Date: Thu Feb 24 23:20:28 IST 2022
WebModule[null] ServletContext.log():AuthenticationFilter:doFilter()
WebModule[null] ServletContext.log():AuthenticationFilter:DoBeforeProcessing
----AuthenticationFilter--------
Success
WebModule[null] ServletContext.log():AuthenticationFilter:DoAfterProcessing
WebModule[null] ServletContext.log():LogFilter:DoAfterProcessing
```

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT619
B.Tech. IT, Sem: VI

Experiment – 06

Submitted By
Roll No.: IT008
Name: Akshar Bhalodia

Aim: Create a JavaBean to store information about person. The details of person (person name, person age, person height, etc.) are stored in person database table. After the person is authenticated, his/her personal details are transferred from the database table (person) to JavaBean (Person) and the details are displayed in proper format using this Person JavaBean. The JavaBean is stored in session scope.

Code:

index.jsp

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

<!DOCTYPE html>

<jsp:useBean id="personBean" class="com.PersonBean" scope="request" />

<html>

    <head>

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

        <title>User Registration</title>

    </head>

    <body >

        <form action="process.jsp" method="post" name="formregistration">

            <table align="center" name="tblouter" width="50%">

                <tr height="300" valign="top">

                    <td align="center" border="1" colspan="10">
```

```
<br>
<table align="center" border="1" cellpadding="2" name="tblfirstchild"
    width="100%">
  <tr>
    <td align="center" colspan="2">
      <font size="2" ><b>User Registration</b></font>
    </td>

  </tr>
  <tr>
    <td align="right" width="25%">User Name:</td>
    <td align="left">
      <input maxlength="35" name="userName" type="text" size="15"
          value="<%=personBean.getUserName()%>">
      <br>
      <font size="2" color="red">
      <%=personBean.getErrorMsg("userName")%></font>
    </td>
  </tr>
  <tr>
    <td align="right" width="25%">Password:</td>
    <td align="left">
      <input maxlength="35" name="password1" type="password" size="15"
          value="<%=personBean.getPassword1()%>">
      <br>
```

```
                  <font size="2" color="red">
                  <%=personBean.getErrorMsg("password1")%></font>
                </td>
              </tr>
              <tr>
                <td colspan="2" align="center">
                  <input name="cmdSubmit" type="submit" value="Submit">
                  <input namd="cmdReset" type="reset" value="Reset">
                </td>
              </tr>
            </table>
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

process.jsp

```
<%@page language="java" import="java.util.*" %>
<jsp:useBean id="personBean" class="com.PersonBean" scope="request">
  <jsp:setProperty name="personBean" property="*"/>
</jsp:useBean>
<%
  if (personBean.validate(request.getParameter("userName"),
      request.getParameter("password1"))) {
```

```
%>
<jsp:forward page="success.jsp"/>
<%} else {
%>
<jsp:forward page="index.jsp"/>
<%
    }
%>
```

success.jsp

```
<jsp:useBean id="personBean" class="com.PersonBean" scope="request" />
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <form name="FrmUserRegistration" >
            <table align="center" name="tblOuter" width="50%">
                <tr height="300" valign="top" >
                    <td align="center" border="1" colspan="10">
                        <br>
                        <table align="center" border="1" cellpadding="2" name="tblFirstChild"
                               width="100%">
                            <tr >
```

```
          <td align="center" colspan="2">
            <font size="2" ><b>Welcome <jsp:getProperty name="personBean"
property="userName"/></b>
            </font>
          </td>
        </tr>
        <tr>
          <td align="right" width="25%">
            First Name:
          </td>
          <td align="left">
            <jsp:getProperty name="personBean" property="firstName"/>
          </td>
        </tr>
        <tr>
          <td align="right" width="25%">
            Last Name:
          </td>

          <td align="left">
            <jsp:getProperty name="personBean" property="lastName"/>
          </td>
        </tr>
        <tr>
          <td align="right" width="25%">
```

```
                    Email Address:
                </td>
                <td align="left">
                    <jsp:getProperty name="personBean" property="email"/>
                </td>
            </tr>
            <tr>
                <td align="right" width="25%">
                    Age:
                </td>
                <td align="left">
                    <jsp:getProperty name="personBean" property="age"/>
                </td>
            </tr>
            <tr>
                <td align="right" width="25%">
                    Height:
                </td>
                <td align="left">
                    <jsp:getProperty name="personBean" property="height"/> Feet
                </td>
            </tr>
        </table>
    </td>
</tr>
```

```
            </table>
        </form>
    </body>
</html>
```

PersonBean.java

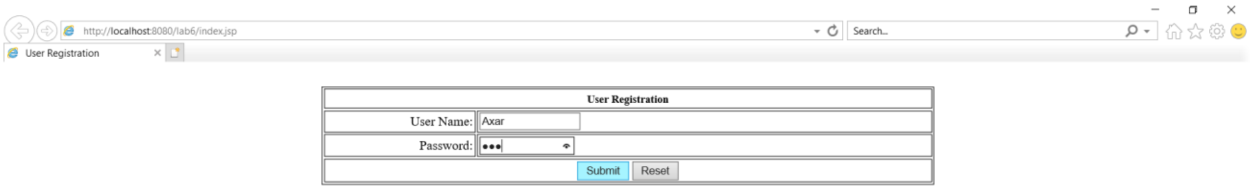
```
package com;



import java.util.*;
import java.beans.*;
import com.DatabaseConnectivity;
import java.util.*;

public class PersonBean {
    private String firstName;
    private String lastName;
    private String email;
    private String userName;
    private String password1;
    private String password2;
    private String age;
    private String height;
    private Hashtable error;
    DatabaseConnectivity user;
```

```
public PersonBean() {

    firstName = "";

    lastName = "";

    email = "";

    userName = "";

    password1 = "";

    age = "";

    height = "";

    error = new Hashtable();

}


public boolean validate(String u, String p) {

    boolean allOk;

    if (userName.equals("")) {

        error.put("userName", "Enter the User Name");

        userName = "";

        allOk = false;

    }


    if (password1.equals("")) {

        error.put("password1", "Enter a valid password");

        password1 = "";

        allOk = false;

    }
    user = new DatabaseConnectivity(u, p);
    allOk = user.checkUser();
    if (allOk) {
        user.getUserData();
        System.out.println("========================================");
        firstName = user.firstName;
        lastName = user.lastName;
        userName = user.userName;
        age = user.age;
        email = user.email;
        height = user.height;

    }
    System.out.println("$$$$$$$$$$$$$$$$$$$$$$$$$$" + allOk);
    return allOk;
}

public String getErrorMsg(String s) {
    String errorMsg = (String) error.get(s.trim());
    return (errorMsg == null) ? "" : errorMsg;
}
public void setErrors(String key, String msg) {
    error.put(key, msg);
}
public String getFirstName() {
    return firstName;
}
public void setFirstName(String fname) {
    firstName = fname;
}
public String getLastName() {
    return lastName;
}
public void setlastName(String lname) {
    lastName = lname;
}
public String getEmail() {
    return email;
}
public void setEmail(String eml) {
    email = eml;
}
public String getUserName() {
    return userName;
}
public void setUserName(String uname) {
    userName = uname;
}
public String getPassword1() {
    return password1;
    }
    public void setPassword1(String p1) {
        password1 = p1;
    }
    public String getAge() {
        return age;
    }
    public void setAge(String a) {
        age = a;
    }
    public String getHeight() {
        return height;
    }

    public void setHeight(String hg) {
        height = hg;
    }
}
```

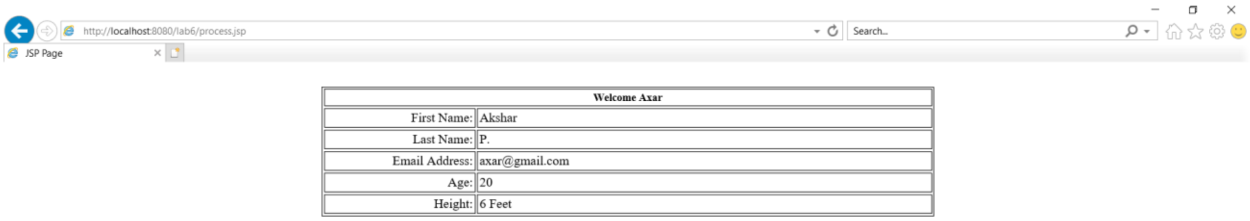
DatabaseConnectivity.java

```
package com;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;



public class DatabaseConnectivity {
    Connection db;
    String SelectQuery;
    Statement ps;
    public String userName, password1, firstName, lastName, email, age, height;

    public DatabaseConnectivity(String userName, String password1) {
        try {
            Class.forName("org.apache.derby.jdbc.ClientDriver");

            this.userName = userName;
            this.password1 = password1;
        } catch (Exception e) {
            System.out.println("Phase 1: " + e);
        }
    }

    public boolean checkUser() {
        boolean check = false;
        System.out.println("-----------------" + userName + "-------" + password1);
        SelectQuery="SELECT * from person where uname='"+userName+"' and
pass1='"+password1+"'";
        try {
            db = DriverManager.getConnection("jdbc:derby://localhost:1527/lab6", "lab6", "lab6");
            ps = db.createStatement();
            ResultSet Res = ps.executeQuery(SelectQuery);
            check = Res.next();
            ps.close();
            db.close();
        } catch (Exception e) {
            System.out.println("Phase 2: " + e);
        }
        return check;
    }

    public void getUserData() {
        SelectQuery="SELECT * from person where uname='"+userName+"' and
pass1='"+password1+"'";
        try {
            db = DriverManager.getConnection("jdbc:derby://localhost:1527/lab6", "lab6", "lab6");
            ps = db.createStatement();
            ResultSet Res = ps.executeQuery(SelectQuery);
            if (Res.next()) {
                firstName = Res.getString("fname");

                lastName = Res.getString("lname");
                email = Res.getString("email");
                age = Res.getString("age");
                height = Res.getString("HEIGHT");
                System.out.println("===="+firstName+"-"+lastName+"-"+email+"-"+age+"-"+height);
            }
            ps.close();
            db.close();
        }
        catch (Exception e)
        {
            System.out.println("Phase 3: " + e);
        }
    }
}
```

Input/Output:



User Registration	
User Name:	Axar
Password:	●●●
Submit Reset	



Welcome Axar	
First Name:	Akshar
Last Name:	P.
Email Address:	axar@gmail.com
Age:	20
Height:	6 Feet

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT619
B.Tech. IT, Sem: VI

Experiment – 07

Submitted By
Roll No.: IT008
Name: Akshar Bhalodia

Aim: Create a JSP based Web application which allows the user to edit his registration information (Refer EXPERIMENT-4). If login is successful, the user authentication servlet creates the welcome message for the user in session scope and then forwards the request to JSP page which handles the edit operation. Use the JSTL core library for variable creations, use and iterations, and JSTL SQL library for interaction with the database.

Code:

Index.jsp

```
<!DOCTYPE html>

<html>

    <head>

        <title>Exam Registration</title>

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

    </head>

    <body style="background-color:burlywood">

        <center><h2>Registration</h2></center>

        <form action="Registration">

        <table>

            <tr>

                <td>USERID:</td>
                <td><input type="text" name="uid"></td>
            </tr>
            <tr>
                <td>PASSWORD:</td>
                <td><input type="password" name="pass"></td>
            </tr>
            <tr>
                <td>CONFORM PASSWORD:</td>
                <td><input type="password" name="cpass"></td>
            </tr>
            <tr>
                <td>FullName:</td>
                <td><input type="text" name="fname"></td>
            </tr>
            <tr>
                <td>SEMESTER:</td>
                <td>
                    <input type="radio" name="sem" value="1"> 1
                    <input type="radio" name="sem" value="2"> 2
                    <input type="radio" name="sem" value="3"> 3
                    <input type="radio" name="sem" value="4"> 4
                    <input type="radio" name="sem" value="5"> 5
                    <input type="radio" name="sem" value="6"> 6
                    <input type="radio" name="sem" value="7"> 7
                    <input type="radio" name="sem" value="8"> 8<br>
```

```
                </td>
            </tr>
            <tr>
                <td>ROLLNO:</td>
                <td><input type="text" name="rno"></td>
            </tr>
            <tr>
                <td>EMAIL:</td>
                <td><input type="text" name="email"></td>
            </tr>
            <tr>
                <td>CONTACT NO:</td>
                <td><input type="text" name="contact"></td>
            </tr>
            <tr>
                <td><input type="submit" value = "REGISTER"></td>
            </tr>
        </table>
        </form>
    </body>
</html>
```

Page.jsp

```
<%@ page import="java.io.*,java.util.*,java.sql.*"%>

<%@ page import="javax.servlet.http.*,javax.servlet.*" %>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>


<html>

   <head>

      <title>sql:update Tag</title>

   </head>

   <body>

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

                    url="jdbc:derby://localhost:1527/exp7"

                    user="exp7"  password="exp7"/>

      <c:set var="Userid" value= "${sessionScope.akshar}" />

      <%

         if (request.getMethod().equalsIgnoreCase("POST")) {

      %>

      <c:set var="pass" value="${param.pass}"/>

      <c:set var="cpass" value="${param.cpass}"/>

      <c:set var="fname" value="${param.fname}"/>

      <c:set var="rno" value="${param.rno}"/>

      <c:set var="sem" value="${param.sem}"/>

      <c:set var="email" value="${param.email}"/>

      <c:set var="cnumber" value="${param.cnumber}"/>
```

```
    <sql:update dataSource="${db}" var="rs1">
        UPDATE Registration SET password = ? ,cpassword = ? ,fname = ?,semester = ?,rollno
= ?,email = ?,cnumber = ? WHERE USERID=?
        <sql:param value="${pass}" />
        <sql:param value="${cpass}" />
        <sql:param value="${fname}" />
        <sql:param value="${sem}" />
        <sql:param value="${rno}" />
        <sql:param value="${email}" />
        <sql:param value="${cnumber}" />
        <sql:param value="${Userid}" />
    </sql:update>
    <c:if test="${rs1!=0}">
        <c:out value="Your Profile Data Updated"/>
    </c:if>
    <c:if test="${rs1==0}">
        <c:out value="Your Profile Data Not Updated"/>
    </c:if>
    <% } else { %>
    <sql:query dataSource="${db}" var="rs">
        SELECT * from Registration WHERE USERID=?
        <sql:param value="${Userid}" />
    </sql:query>
    <form action="page.jsp" method="Post">
```

```
<table>
    <c:forEach var="table" items="${rs.rows}">
        <tr>
            <td>USERID:</td>
            <td><input type="text" name="uid" value="${table.userid}" readonly ></td>
        </tr>
        <tr>
            <td>PASSWORD:</td>
            <td><input type="password" name="pass" value="${table.password}"></td>
        </tr>
        <tr>
            <td>CONFORM PASSWORD:</td>
            <td><input type="password" name="cpass" value="${table.cpassword}"></td>
        </tr>
        <tr>
            <td>FullName:</td>
            <td><input type="text" name="fname" value="${table.FName}"></td>
        </tr>
        <tr>
            <td>SEMESTER:</td>
            <td>
                <input type="text" name="sem" value="${table.semester}">
            </td>
        </tr>
        <tr>
```

```
                    <td>ROLLNO:</td>
                    <td><input type="text" name="rno" value="${table.rollno}"></td>
                </tr>
                <tr>
                    <td>EMAIL:</td>
                    <td><input type="text" name="email" value="${table.email}"></td>
                </tr>
                <tr>
                    <td>CONTACT NO:</td>
                    <td><input type="text" name="cnumber" value="${table.cnumber}"></td>
                </tr>
                <tr>
                    <td><input type="submit" value = "Save Changes"></td>
                </tr>
            </c:forEach>
        </table>
    </form>
    <%}%>
  </body>
</html>
```

Registration.java

```
import java.io.IOException;

import java.io.PrintWriter;

import java.sql.*;

import javax.servlet.*;

import javax.servlet.http.*;


public class Registration extends HttpServlet {


    Connection cn = null;

    ResultSet rs = null;

    String driver = "org.apache.derby.jdbc.ClientDriver";

    String url = "jdbc:derby://localhost:1527/exp7";

    String query = "insert into registration values (?,?,?,?,?,?,?,?)";

    PreparedStatement pst = null;


    protected void processRequest(HttpServletRequest request, HttpServletResponse response)

            throws ServletException, IOException {

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

        PrintWriter out = response.getWriter();

        HttpSession session = request.getSession();

        String userid;

        try {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Registration</title>");
            out.println("</head>");
            out.println("<body style=\"background-color:powderblue\">");
            try {
                Class.forName(driver);
                cn = DriverManager.getConnection(url, "exp7", "exp7");
            } catch (Exception e) {
            }
            int flag = 0;
            try {
                pst = cn.prepareStatement(query);
                PreparedStatement pst1 = cn.prepareStatement("select * from registration where userid
= ?");
                pst1.setString(1, request.getParameter("uid"));
                ResultSet rs1 = pst1.executeQuery();
                if ((request.getParameter("uid") == "") || (request.getParameter("pass") == "") ||
(request.getParameter("cpass") == "") || (request.getParameter("fname") == "") ||
request.getParameter("sem") == "" || request.getParameter("rno") == "" ||
request.getParameter("email") == "" || request.getParameter("contact") == "") {
                    out.println("Enter whole data!!!!!");
                } else if (rs1.next() == true) {
                    out.println("<p>UserID Is Already Exist</p>");
                    out.println("<p>Please, Insert Correct Data: </p>");
                } else {
```

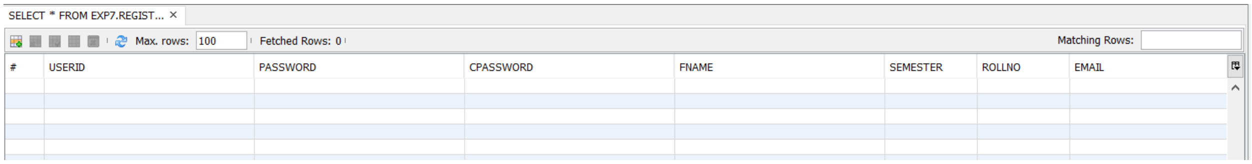
```
        if (request.getParameter("pass").equals(request.getParameter("cpass"))) {
            pst.setString(1, request.getParameter("uid"));
            pst.setString(2, request.getParameter("pass"));
            pst.setString(3, request.getParameter("cpass"));
            pst.setString(4, request.getParameter("fname"));
            pst.setInt(5, Integer.parseInt(request.getParameter("sem")));
            pst.setInt(6, Integer.parseInt(request.getParameter("rno")));
            pst.setString(7, request.getParameter("email"));
            pst.setString(8, request.getParameter("contact"));
            int ix = pst.executeUpdate();
            out.println(ix + " records inserted");
            userid = request.getParameter("uid");

            session.setAttribute("akshar", userid);

            cn.close();

            flag = 1;
        } else {
            out.println("<p>Conform Password dosen't Matched</p>");
            out.println("<p>Please, Insert Correct Data: </p>");
        }
    }
} catch (Exception e) {
}
```

```
        if (flag == 1) {
            out.println("<p>our Data Is Inserted Successfully</p>");


            out.println("<a href=\"page.jsp\">Edit Details</a>");


        } else {
            out.println("<a href=\"index.jsp\">Back TO Registration</a>");
            out.println("<a href=\"page.jsp\">Edit Details</a>");
        }
    } catch (Exception e) {
    } finally {
        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";
    }
}
```

Input/Output:

Database Table before registration:

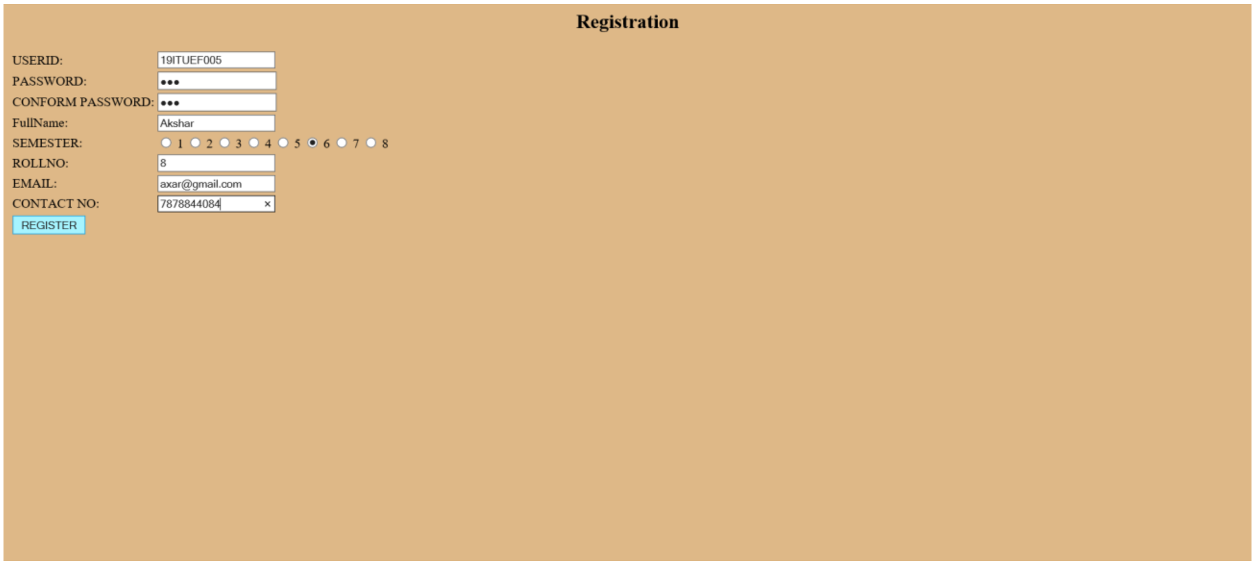


SELECT * FROM EXP7.REGIST... ×

Max. rows: 100   Fetched Rows: 0   Matching Rows:

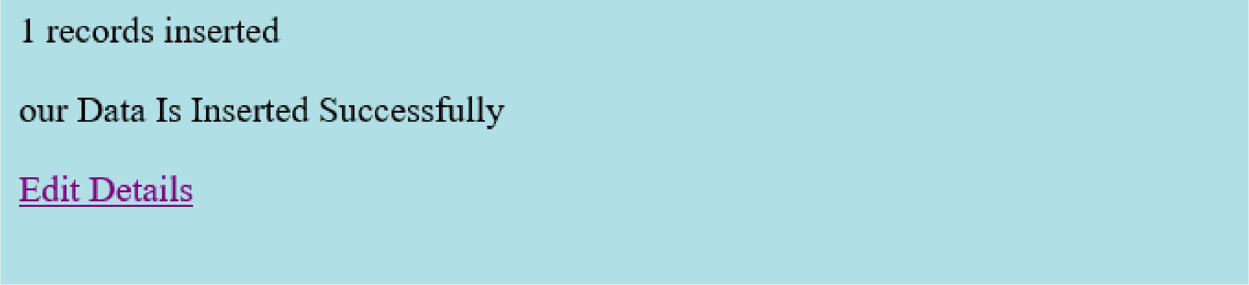
#	USERID	PASSWORD	CPASSWORD	FNAME	SEMESTER	ROLLNO	EMAIL

Registration:



Registration

USERID:	19ITUEF005
PASSWORD:	•••
CONFORM PASSWORD:	•••
FullName:	Akshar
SEMESTER:	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ● 6 ○ 7 ○ 8
ROLLNO:	8
EMAIL:	axar@gmail.com
CONTACT NO:	7878844084

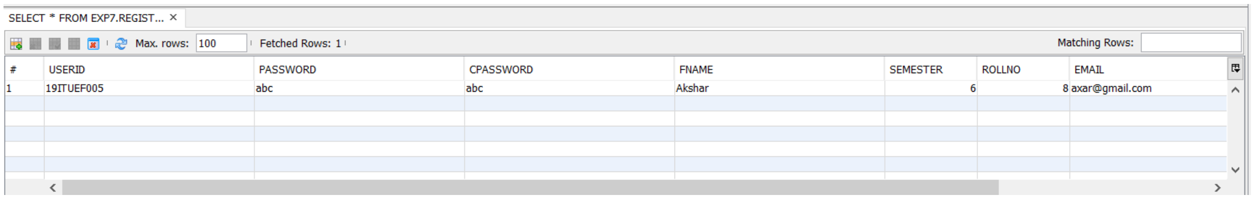


1 records inserted

our Data Is Inserted Successfully

Edit Details

After Registration Database table:

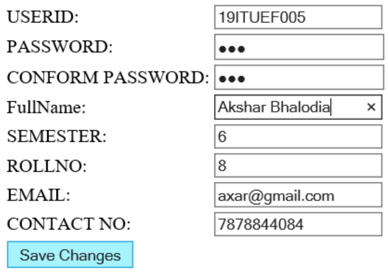


SELECT * FROM EXP7.REGIST...

Max. rows: 100 Fetched Rows: 1 Matching Rows:

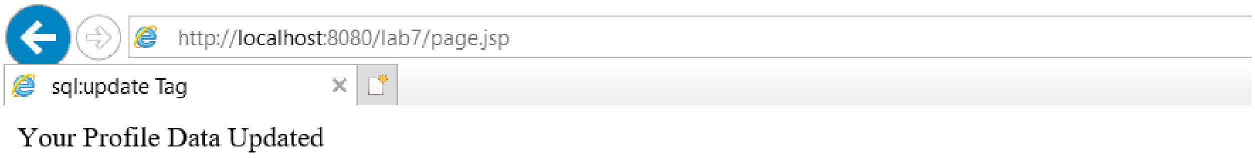
#	USERID	PASSWORD	CPASSWORD	FNAME	SEMESTER	ROLLNO	EMAIL
1	19ITUEF005	abc	abc	Akshar	6	8	axar@gmail.com

Edit Details:



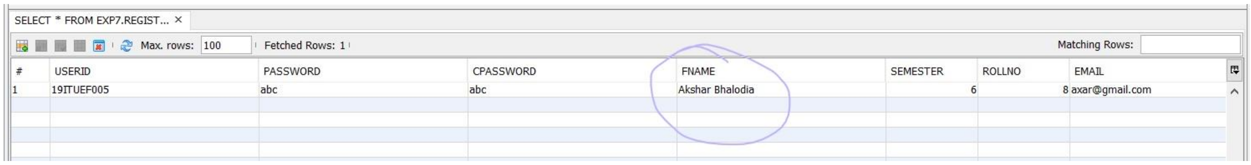
USERID:	19ITUEF005
PASSWORD:	●●●
CONFORM PASSWORD:	●●●
FullName:	Akshar Bhalodia
SEMESTER:	6
ROLLNO:	8
EMAIL:	axar@gmail.com
CONTACT NO:	7878844084

Save Changes:



Your Profile Data Updated

Database Table After Updation:



#	USERID	PASSWORD	CPASSWORD	FNAME	SEMESTER	ROLLNO	EMAIL
1	19ITUEF005	abc	abc	Akshar Bhalodia	6	8	axar@gmail.com

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT619
B.Tech. IT, Sem: VI

Experiment – 08

Submitted By
Roll No.: IT008
Name: Akshar Bhalodia

Aim: Write steps to provide Basic Authentication to a Web Application. The application has two secure directories (secureAdmin and secureUser) corresponding to two users – Admin and User. Theapplication has two html files: (i) pageA.html under SecureAdmin directory and (ii) pageU.html under secureUser directory.

Code:

web.xml

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

<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">

    <session-config>

        <session-timeout>

            30

        </session-timeout>

    </session-config>

    <security-constraint>

        <display-name>adminConstraint</display-name>

        <web-resource-collection>
            <web-resource-name>adminResource</web-resource-name>
            <description/>
            <url-pattern>/secure_admin/*</url-pattern>
        </web-resource-collection>
        <auth-constraint>
            <description/>
            <role-name>Admin</role-name>
        </auth-constraint>
    </security-constraint>
    <security-constraint>
        <display-name>userConstraint</display-name>
        <web-resource-collection>
            <web-resource-name>userResource</web-resource-name>
            <description/>
            <url-pattern>/secure_user/*</url-pattern>
        </web-resource-collection>
        <auth-constraint>
            <description/>
            <role-name>User</role-name>
        </auth-constraint>
    </security-constraint>
    <login-config>
        <auth-method>BASIC</auth-method>
```

```
        <realm-name>file</realm-name>
    </login-config>
    <security-role>
        <description/>
        <role-name>Admin</role-name>
    </security-role>
    <security-role>
        <description/>
        <role-name>User</role-name>
    </security-role>
</web-app>
```

glassfish-web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE glassfish-web-app PUBLIC "-//GlassFish.org//DTD GlassFish Application Server 3.1 Servlet 3.0//EN" "http://glassfish.org/dtds/glassfish-web-app_3_0-1.dtd">
<glassfish-web-app error-url="">
  <security-role-mapping>
    <role-name>Admin</role-name>
    <principal-name>testadmin</principal-name>
  </security-role-mapping>
  <security-role-mapping>
    <role-name>User</role-name>
    <principal-name>testuser</principal-name>
  </security-role-mapping>

  <class-loader delegate="true"/>

  <jsp-config>

    <property name="keepgenerated" value="true">

      <description>Keep a copy of the generated servlet class' java code.</description>

    </property>

  </jsp-config>

</glassfish-web-app>
```

pageA.html

```
<!DOCTYPE html>

<html>

    <head>

        <title>secure admin page</title>

        <meta charset="UTF-8">

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

    </head>

    <body>

        <div>secure  admin area</div>

    </body>

</html>
```

pageU.html

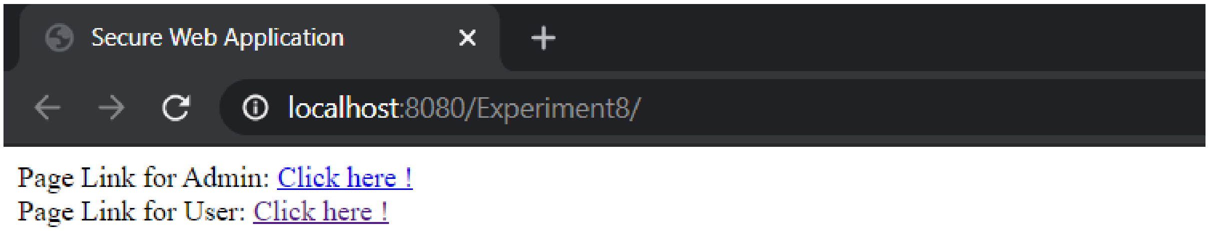
```
<!DOCTYPE html>
<html>
    <head>
        <title>secure user area</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <div>secure  user area</div>
    </body>
</html>
```

index.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Secure  Web Application</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <div>Page Link for Admin:  <a href="secure_admin/pageA.html">Click here !</a></div>
        <div>Page Link for User:  <a href="secure_user/pageU.html">Click here !</a></div>
    </body>
</html>
```

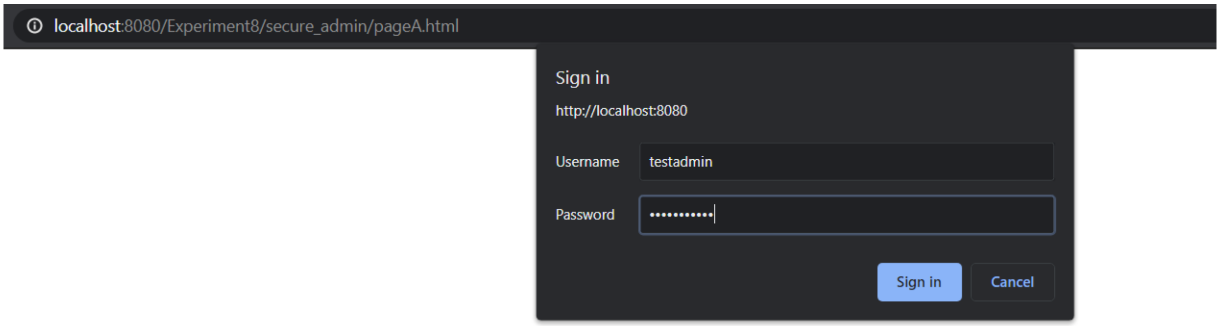
Input/Output:

Main



Page Link for Admin: Click here !
Page Link for User: Click here !

Click on admin



localhost:8080/Experiment8/secure_admin/pageA.html

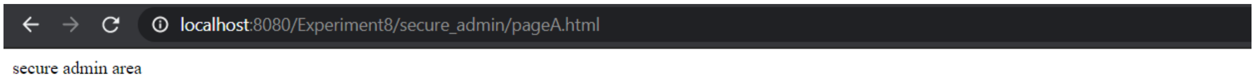
Sign in

http://localhost:8080

Username: testadmin

Password: •••••••••••

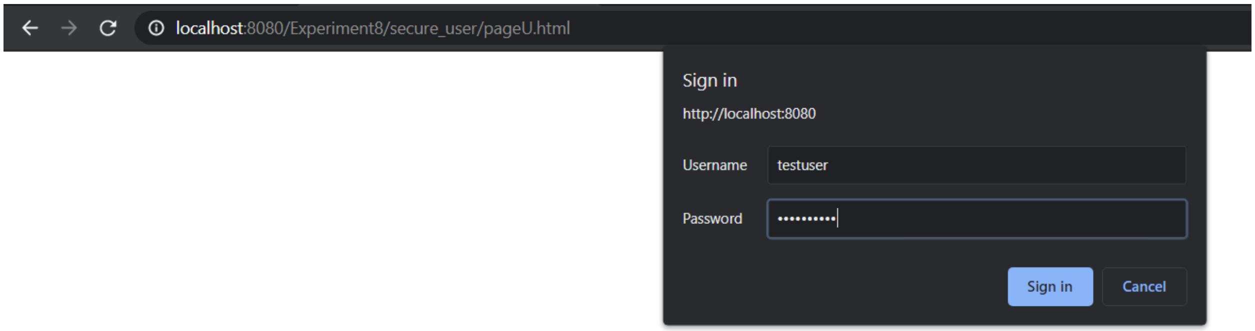
After click on OK



localhost:8080/Experiment8/secure_admin/pageA.html

secure admin area

Click on user



localhost:8080/Experiment8/secure_user/pageU.html

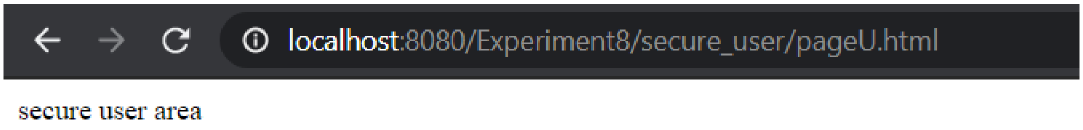
Sign in

http://localhost:8080

Username: testuser

Password: ••••••••••

After click on OK



localhost:8080/Experiment8/secure_user/pageU.html

secure user area

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT619
B.Tech. IT, Sem: VI

Experiment – 09

Submitted By
Roll No.: IT008
Name: Akshar Bhalodia

Aim: Create custom tags: date and header. The date tag is used to display current date and header tag is used to print the header in proper format. The header tag has following attributes: align, border, bgcolor, color, font, and size. Show the usage of these two tags in your JSP page. The align, color, font, and size are for alignment of text, color of text, font-family for text, and size of text respectively. The border, and bgcolor are for border size of box containing text, and background color of box respectively.

Code:

Date.tld

```
<?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>date</short-name>

  <uri>/WEB-INF/tlds/Date</uri>

  <tag>

    <name>DateHandler</name>

    <tag-class>EXP9.DateHandler</tag-class>

    <body-content>empty</body-content>
  </tag>
</taglib>
```

Header.tld

```
<?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>header</short-name>
    <uri>/WEB-INF/tlds/header</uri>
    <tag>
        <name>HeaderHandler</name>
        <tag-class>EXP9.HeaderHandler</tag-class>
        <body-content>JSP</body-content>
        <attribute>
            <name>align</name>
            <rtexprvalue>true</rtexprvalue>
            <type>java.lang.String</type>
        </attribute>
        <attribute>
            <name>color</name>
            <rtexprvalue>true</rtexprvalue>
            <type>java.lang.String</type>
        </attribute>
        <attribute>
```

```
            <name>font</name>
            <rtexprvalue>true</rtexprvalue>
            <type>java.lang.String</type>
        </attribute>
        <attribute>
            <name>size</name>
            <rtexprvalue>true</rtexprvalue>
            <type>java.lang.String</type>
        </attribute>
        <attribute>
            <name>border</name>
            <rtexprvalue>true</rtexprvalue>
            <type>java.lang.String</type>
        </attribute>
        <attribute>
            <name>bgcolor</name>
            <rtexprvalue>true</rtexprvalue>
            <type>java.lang.String</type>
        </attribute>
    </tag>
</taglib>
```

DateHandler.java

```
package EXP9;


import java.util.Calendar;
import javax.servlet.jsp.JspWriter;

import javax.servlet.jsp.JspException;

import javax.servlet.jsp.tagext.JspFragment;

import javax.servlet.jsp.tagext.SimpleTagSupport;


public class DateHandler extends SimpleTagSupport {


    @Override

    public void doTag() throws JspException {

        JspWriter out = getJspContext().getOut();


        try {

            out.print(Calendar.getInstance().getTime());

        }


        catch (java.io.IOException ex) {

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

        }

    }


}
```

HeaderHandler.java

```
package EXP9;


import java.io.IOException;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.tagext.BodyContent;
import javax.servlet.jsp.tagext.BodyTagSupport;

public class HeaderHandler extends BodyTagSupport {

    private String align;
    private String color;
    private String font;
    private String size;
    private String border;
    private String bgcolor;

    public HeaderHandler() {
        super();
    }

    @Override
    public int doStartTag() throws JspException {
        if(bgcolor==null)
            setBgcolor("white");
        if(align==null)
            setAlign("left");
        if(font==null)
    }
        setFont("Arial");
    if(color==null)
        setColor("Black");
    if(size==null)
        setSize("5");
    if(border==null)
        setBorder("0");

    String header="<table bgcolor=\'"+bgcolor+"\' border=\'"+border+"\' align=\'"+align+"\'>";
    header += "<tr><td><font face=\'"+font+"\' color=\'"+color+"\' size=\'"+size+"\'>";
    try{
        pageContext.getOut().write(header);
    }catch(IOException ioe){
        ioe.printStackTrace();
    }
    return EVAL_BODY_INCLUDE;
}

@Override
public int doEndTag() throws JspException {
    try{
        pageContext.getOut().write("</font></td></tr></table>");
    }catch(IOException ioe){
        ioe.printStackTrace();
    }
    return EVAL_PAGE;
}

public void setAlign(String align) {
    this.align = align;
}

public void setColor(String color) {
    this.color = color;
}

public void setFont(String font) {
    this.font = font;
}

public void setSize(String size) {
    this.size = size;
}

public void setBorder(String border) {
    this.border = border;
}

public void setBgcolor(String bgcolor) {
    this.bgcolor = bgcolor;
    }
}
```

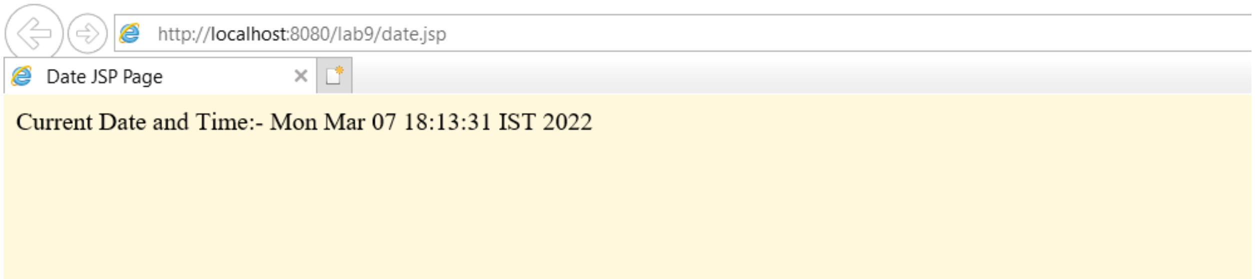
date.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib uri="/WEB-INF/tlds/Date.tld" prefix="m" %>


<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Date JSP Page</title>
    </head>
    <body style="background-color: cornsilk">
        Current Date and Time:- <m:DateHandler/>
    </body>
</html>
```

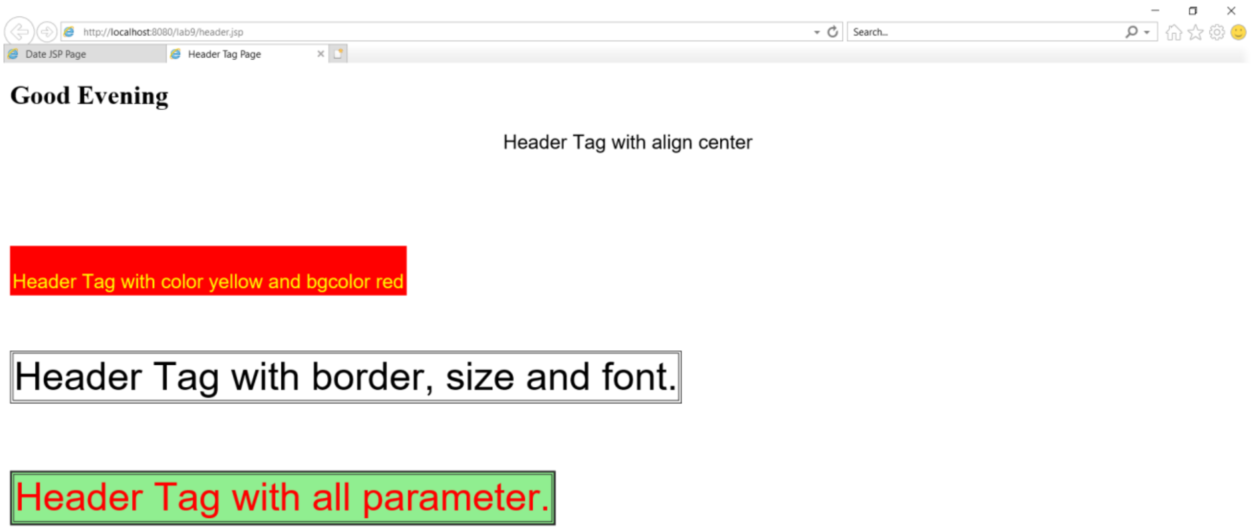
header.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="header" uri="/WEB-INF/tlds/Header.tld" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Header Tag Page</title>
    </head>
    <body>

        <h1>Good Evening</h1>

        <header:HeaderHandler align="center">

            Header Tag with align center

        </header:HeaderHandler><br><br><br><br><br><br>

        <header:HeaderHandler color="yellow" bgcolor="red"><br>

            Header Tag with color yellow and bgcolor red<br>

        </header:HeaderHandler><br><br><br><br><br><br><br>

        <header:HeaderHandler size="15" border="1" font="Arial">

            Header Tag with border, size and font.

        </header:HeaderHandler><br><br><br><br><br><br><br><br>

        <header:HeaderHandler size="20" border="2" font="Arial" align="Left"
bgcolor="lightgreen" color="red">

            Header Tag with all parameter.

        </header:HeaderHandler>

    </body>

</html>
```

Input/Output:



```
Current Date and Time:- Mon Mar 07 18:13:31 IST 2022
```



Good Evening

Header Tag with align center

Header Tag with color yellow and bgcolor red

Header Tag with border, size and font.

Header Tag with all parameter.

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT619
B.Tech. IT, Sem: VI

Experiment – 10

Submitted By
Roll No.: IT008
Name: Akshar Bhalodia

Aim: create a RMI based client-server application. The server allows access of bank account object to client through RMI mechanism. The account object allows following operations: deposit, withdraw and balance. Design appropriate interface and test implementation on network.

Code:

BankIntf.java

```
import java.rmi.*;


public interface BankIntf extends Remote {

  public double Deposit(double a) throws RemoteException;

  public double Withdraw(double a) throws RemoteException;

  public double showBalance() throws RemoteException;

}
```

BankImplement.java

```
import java.rmi.*;

import java.rmi.server.*;

public class BankImplement extends UnicastRemoteObject

  implements BankIntf {

        double bal = 500000;
  public BankImplement() throws RemoteException {
  }
  public double Deposit(double a) throws RemoteException {
            bal += a;
    return bal;
  }
  public double Withdraw(double a) throws RemoteException {
            bal -= a;
    return bal;
  }
  public double showBalance() throws RemoteException {
    return bal;
  }
}
```

BankServer.java

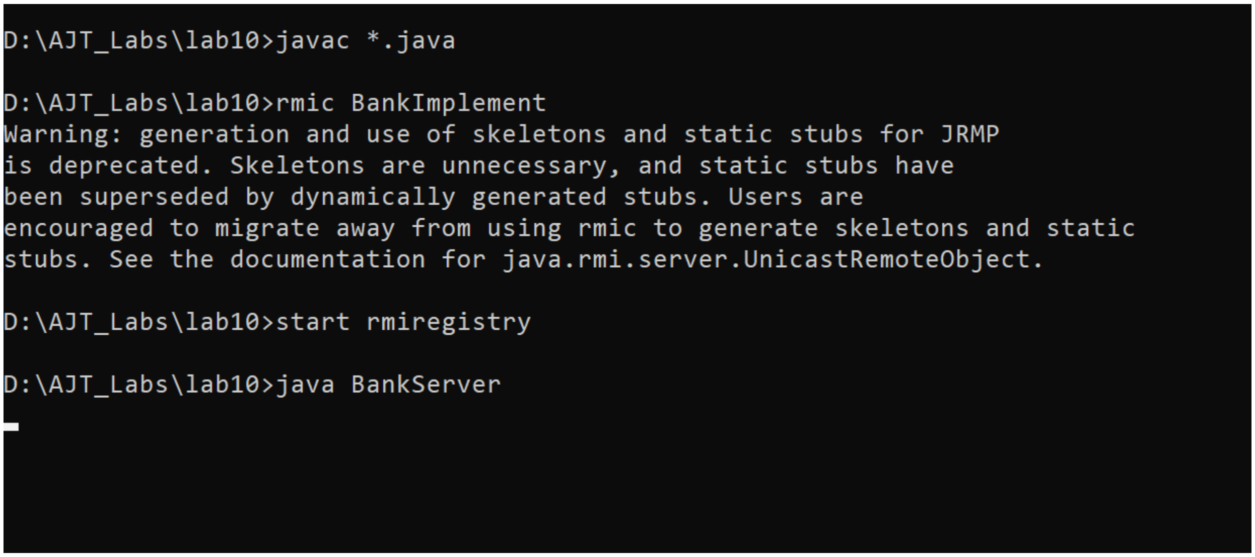
```
import java.rmi.*;
import java.net.*;

public class BankServer {
  public static void main(String args[]) {
    try {
      BankImplement bankImpl = new BankImplement();
      Naming.rebind("BankServer", bankImpl);
    }
    catch(Exception e) {
      System.out.println("Exception: " + e);
    }
  }
}
```

BankClient.java

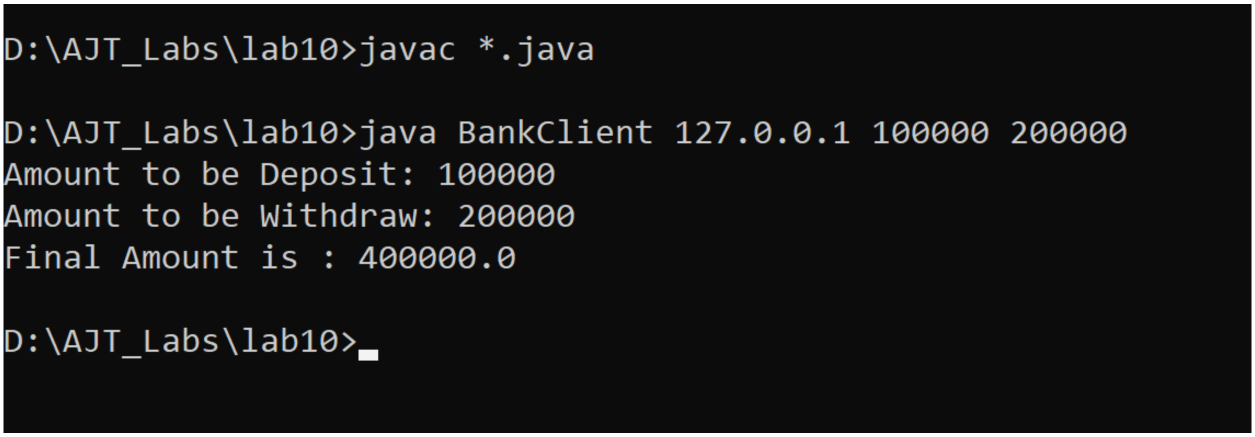
```
import java.rmi.*;

public class BankClient{
        public static void main(String args[]) {
                try {
                  String bankURL = "rmi://" + args[0] + "/BankServer";
                  BankIntf bankIntf =
                                              (BankIntf)Naming.lookup(bankURL);
                  System.out.println("Amount to be Deposit: " + args[1]);
                  double d1 = Double.valueOf(args[1]).doubleValue();
                  System.out.println("Amount to be Withdraw: " + args[2]);

                  double d2 = Double.valueOf(args[2]).doubleValue();
                  bankIntf.Deposit(d1);
                  bankIntf.Withdraw(d2);
                  System.out.println("Final Amount is : " + bankIntf.showBalance());
                  }
                  catch(Exception e){
```

```
                System.out.println(e);
            }
        }
}
```

Output:



```
D:\AJT_Labs\lab10>javac *.java

D:\AJT_Labs\lab10>rmic BankImplement
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.

D:\AJT_Labs\lab10>start rmiregistry

D:\AJT_Labs\lab10>java BankServer
```



```
D:\AJT_Labs\lab10>javac *.java

D:\AJT_Labs\lab10>java BankClient 127.0.0.1 100000 200000
Amount to be Deposit: 100000
Amount to be Withdraw: 200000
Final Amount is : 400000.0

D:\AJT_Labs\lab10>
```

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT619
B.Tech. IT, Sem: VI

Experiment – 11

Submitted By
Roll No.: IT008
Name: Akshar Bhalodia

Aim: Create and use a session bean to calculate the income-tax on annual income. The bean takes salary (annual income), and total investment as arguments to business method and returns calculated income-tax as result. The business rules for calculating income-tax are as follows. No income-tax on first 100,000 Rs. of salary. 10% tax on next 100,000 Rs. of remaining salary, 20% on next 100,000 Rs. of remaining salary, 30% on next 100,000 Rs. of remaining salary, and 100% on remaining salary. The investment of maximum Rs.100,000 is considered as non-chargeable income.

Code:

- **TaxCalculatorRemote.java**

```
package ejbs;


import java.rmi.RemoteException;
import javax.ejb.EJBObject;
public interface TaxCalculatorRemote extends EJBObject{
   double calculateTaxAmount(double income) throws RemoteException;
}
```

- **TaxCalculatorRemoteHome.java**

```
package ejbs;


import java.rmi.RemoteException;
import javax.ejb.CreateException;
import javax.ejb.EJBHome;

public interface TaxCalculatorRemoteHome extends EJBHome {
    ejbs.TaxCalculatorRemote create() throws CreateException, RemoteException;
}
```

- **TaxCalculatorBean.java**

```
package ejbs;

import javax.ejb.SessionBean;
import javax.ejb.SessionContext;

public class TaxCalculatorBean implements SessionBean {
    private SessionContext context;

    @Override
    public void setSessionContext(SessionContext aContext) {
        context = aContext;
    }

    @Override
    public void ejbActivate() {
    }

    @Override
    public void ejbPassivate() {
    }
```

```
@Override
public void ejbRemove() {
}

public void ejbCreate() {
}

public double calculateTaxAmount(double income) {
    double TaxAmount;
    if(income<=100000)
    {
        TaxAmount=0;
        return TaxAmount;
    }
    else if(income<=200000)
    {
        TaxAmount=(income-100000)*0.1;
        return TaxAmount;
    }
    else if(income<=300000)
    {
        TaxAmount=(income-200000)*0.20+10000;
        return TaxAmount;
    }
    else if(income<=400000)
    {
            TaxAmount=(income-300000)*0.3+30000;
            return TaxAmount;
        }
        else
        {
            TaxAmount=(income-400000)+60000;
            return TaxAmount;
        }
    }
}
```

- **ejb-jar.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<ejb-jar xmlns="http://xmlns.jcp.org/xml/ns/javaee"
         version="3.2"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/ejb-jar_3_2.xsd">
    <enterprise-beans>
        <session>
            <display-name>TaxCalculatorSB</display-name>
            <ejb-name>TaxCalculatorBean</ejb-name>
            <home>ejbs.TaxCalculatorRemoteHome</home>
            <remote>ejbs.TaxCalculatorRemote</remote>
            <ejb-class>ejbs.TaxCalculatorBean</ejb-class>
            <session-type>Stateless</session-type>
            <transaction-type>Container</transaction-type>
    </session>
  </enterprise-beans>
  <assembly-descriptor>
    <container-transaction>
      <method>
        <ejb-name>TaxCalculatorBean</ejb-name>
        <method-name>*</method-name>
      </method>
      <trans-attribute>Required</trans-attribute>
    </container-transaction>
  </assembly-descriptor>
</ejb-jar>
```

- **glassfish-ejb-jar.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE glassfish-ejb-jar PUBLIC "-//GlassFish.org//DTD GlassFish Application Server 3.1 EJB 3.1//EN" "http://glassfish.org/dtds/glassfish-ejb-jar_3_1-1.dtd">
<glassfish-ejb-jar>
  <enterprise-beans>
    <ejb>
      <ejb-name>TaxCalculatorBean</ejb-name>
      <jndi-name>ejb/TaxCalculatorBean</jndi-name>
    </ejb>
  </enterprise-beans>
</glassfish-ejb-jar>
```

- **Test.java**

```
package com;

import ejbs.TaxCalculatorRemote;

import ejbs.TaxCalculatorRemoteHome;

import java.io.IOException;

import java.io.PrintWriter;

import java.rmi.RemoteException;

import javax.ejb.CreateException;

import javax.naming.Context;

import javax.naming.InitialContext;

import javax.naming.NamingException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class Test extends HttpServlet {


    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)

            throws ServletException, IOException {

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

        PrintWriter out = response.getWriter();

        try {

            out.println("<html>");

            out.println("<head>");

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

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

            out.println("<body>");

            TaxCalculatorRemote robj = lookupTaxCalculatorBean();
            double income1,income2,income3,income4,income5;
            income1 = 120000;
            income2 = 226000;
            income3 = 340000;
            income4 = 455500;
            income5 = 1000000;
            out.println("Tax amount for person 1 is " + robj.calculateTaxAmount(income1));
            out.println("Tax amount for person 2 is " + robj.calculateTaxAmount(income2));
            out.println("Tax amount for person 3 is " + robj.calculateTaxAmount(income3));
            out.println("Tax amount for person 4 is " + robj.calculateTaxAmount(income4));
            out.println("Tax amount for person 5 is " + robj.calculateTaxAmount(income5));
            out.println("</body>");
            out.println("</html>");
        } finally {
            out.close();
        }
    }

    @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);
    }


    TaxCalculatorRemote lookupTaxCalculatorBean() {
        try {
            Context c = new InitialContext();
            Object remote = c.lookup("java:comp/env/TaxCalculatorBean");
            TaxCalculatorRemoteHome rv = (TaxCalculatorRemoteHome)
javax.rmi.PortableRemoteObject.narrow(remote, ejbs.TaxCalculatorRemoteHome.class);
            return rv.create();
        } catch (NamingException ne) {

java.util.logging.Logger.getLogger(getClass().getName()).log(java.util.logging.Level.SE
VERE, "exception caught", ne);
            throw new RuntimeException(ne);
        } catch (CreateException ce) {

java.util.logging.Logger.getLogger(getClass().getName()).log(java.util.logging.Level.SE
VERE, "exception caught", ce);
            throw new RuntimeException(ce);
        } catch (RemoteException re) {

java.util.logging.Logger.getLogger(getClass().getName()).log(java.util.logging.Level.SE
VERE, "exception caught", re);
            throw new RuntimeException(re);
        }
    }
}
```

- **Web.xml**

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

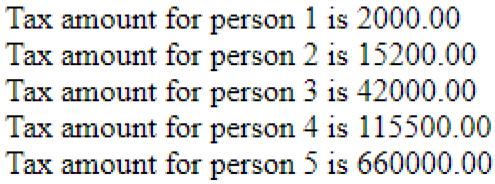
```
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">

  <servlet>

    <servlet-name>Test</servlet-name>

    <servlet-class>com.Test</servlet-class>

  </servlet>

  <servlet-mapping>

    <servlet-name>Test</servlet-name>

    <url-pattern>/Test</url-pattern>

  </servlet-mapping>

  <session-config>

    <session-timeout>

      30

    </session-timeout>

  </session-config>

  <ejb-ref>

    <ejb-ref-name>TaxCalculatorBean</ejb-ref-name>

    <ejb-ref-type>Session</ejb-ref-type>

    <home>ejbs.TaxCalculatorRemoteHome</home>

    <remote>ejbs.TaxCalculatorRemote</remote>

  </ejb-ref>

</web-app>
```

- **glassfish-web.xml**

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

```
<!DOCTYPE glassfish-web-app PUBLIC "-//GlassFish.org//DTD GlassFish Application
Server 3.1 Servlet 3.0//EN" "http://glassfish.org/dtds/glassfish-web-app_3_0-1.dtd">

<glassfish-web-app error-url="">

  <class-loader delegate="true"/>

  <jsp-config>

    <property name="keepgenerated" value="true">

      <description>Keep a copy of the generated servlet class' java code.</description>

    </property>

  </jsp-config>

   <ejb-ref>

        <ejb-ref-name>TaxCalculatorBean</ejb-ref-name>

        <jndi-name>ejb/TaxCalculatorBean</jndi-name>

    </ejb-ref>

</glassfish-web-app>
```

Input/Output:



```
Tax amount for person 1 is 2000.00
Tax amount for person 2 is 15200.00
Tax amount for person 3 is 42000.00
Tax amount for person 4 is 115500.00
Tax amount for person 5 is 660000.00
```