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

## Experiment – 01

**Submitted By:**
**Roll No.: IT083**
**Name: Nikhil Nasit**

**Aim:** 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: 4. Search by telephone no. 5. Search by name (one of first name, middle name, and last name) with exactly match and part of name. 6. Search by address (one of address 1, address 2, area name, and city) with exactly match and part of address.

Tools /Apparatus: JDK 1.6 or above , Netbeans IDE 6.1

**Code:**

```
package lab1;

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
```

```
public class Lab1 extends Frame implements ItemListener, ActionListener {

    Panel pTop = new Panel();
    TextField tf1 = new TextField(20);
    TextArea ta1 = new TextArea(10, 200);
    Choice c1 = new Choice();
    Choice c2 = new Choice();
    Button srchbutton = new Button("Search");
    Label status = new Label("Records Found = 0 ");

    String query = "";
    Connection con = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    public Lab1() {
        super("My Telephone Directory");
        setVisible(true);
        setSize(400, 800);

        c1.add("Telephone No");
        c1.add("Name");
        c1.add("Address");
        c1.addItemListener(this);
        srchbutton.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(srchbutton);

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

        try {
            Class.forName("org.postgresql.Driver");
            con =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/postgres",
"postgres", "@bcde");
        } 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 e) {
        String arg = e.getItem().toString();
        if (arg.equals("Telephone No")) {
            c2.setVisible(false);
```

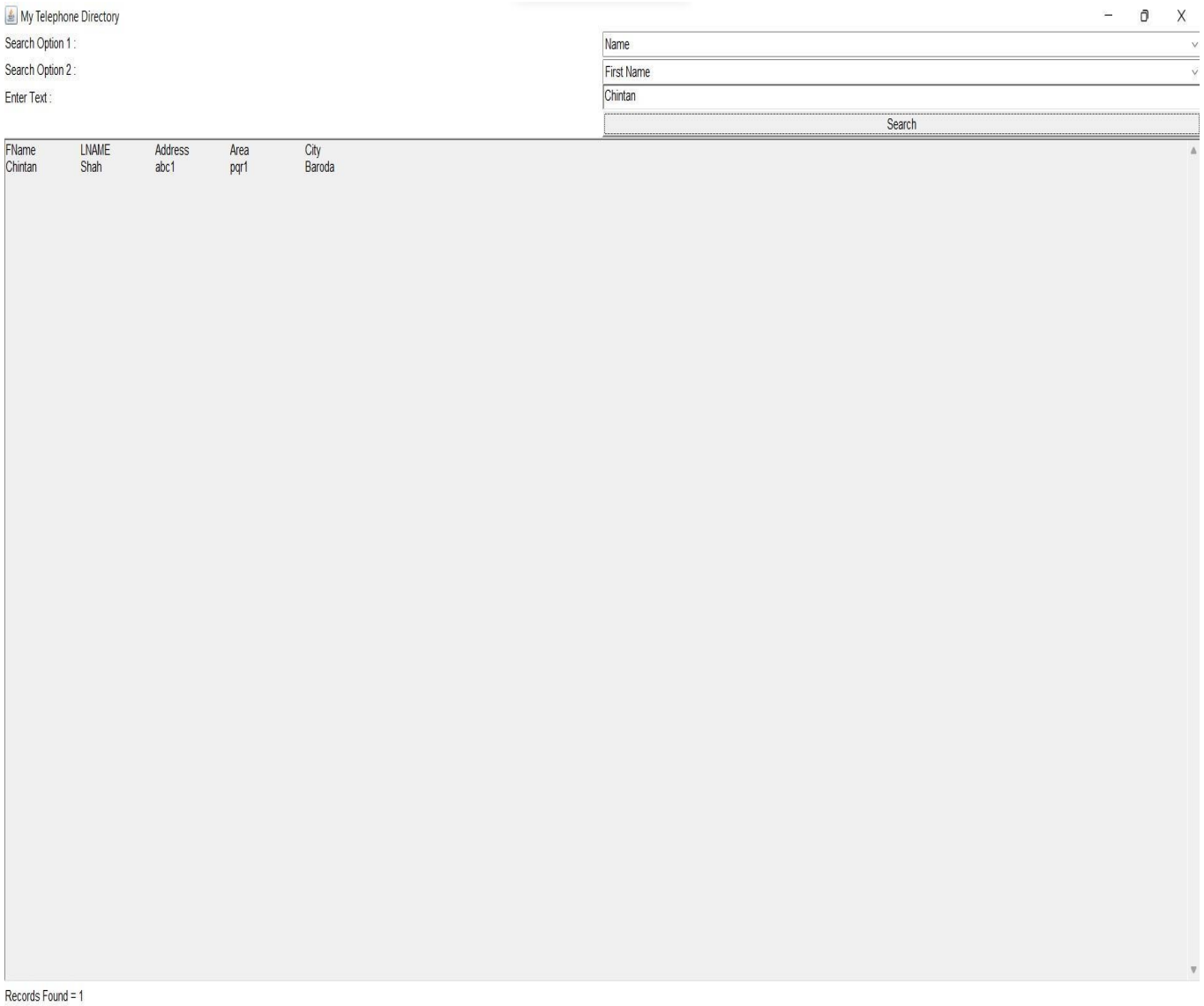
```
    } else if (arg.equals("Name")) {
        c2.removeAll();
        c2.add("First Name");
        c2.add("Last Name");
        c2.setVisible(true);
    } else if (arg.equals("Address")) {
        c2.removeAll();
        c2.add("Area");
        c2.add("City");
        c2.setVisible(true);
    }
}

public void actionPerformed(ActionEvent e) {
    ta1.setText("Refreshed");
    query = new String("select * from telephonedirectory");
    int len = 0;
    len = tf1.getText().toString().trim().length();

    try {
        if (c1.getSelectedItem().equals("Telephone No") && len > 0) {
            query += " where number=?";
            pstmt = con.prepareStatement(query);
            pstmt.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(("Last Name"))) {
                query += " where lname=?";
            }
            pstmt = con.prepareStatement(query);
            pstmt.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=?";
        }

        pstmt = con.prepareStatement(query);
        pstmt.setString(1, tf1.getText().toString().trim());
    } else {
        pstmt = con.prepareStatement(query);
    }

    try {
        System.out.println(query);
        rs = pstmt.executeQuery();
    } catch (NullPointerException ne) {
        System.out.println("Text Null3");
        ta1.setText("No Records Found");
        status.setText("Records Found = 0");
    }

    if (rs != null) {
        ta1.setText("FName\t\tLNAME\t\tAddress\t\tArea\t\tCity\n");
    }

    int count = 0;
    while (rs.next()) {
        ta1.append("" + rs.getString(1) + "\t\t");
        ta1.append("" + rs.getString(2) + "\t\t");
        ta1.append("" + rs.getString(3) + "\t\t");

        ta1.append("" + rs.getString(5) + "\t\t");
        ta1.append("" + rs.getString(6) + "\t\n");
        count++;
    }
    status.setText("Records Found = " + count);
} catch (Exception ee) {
```

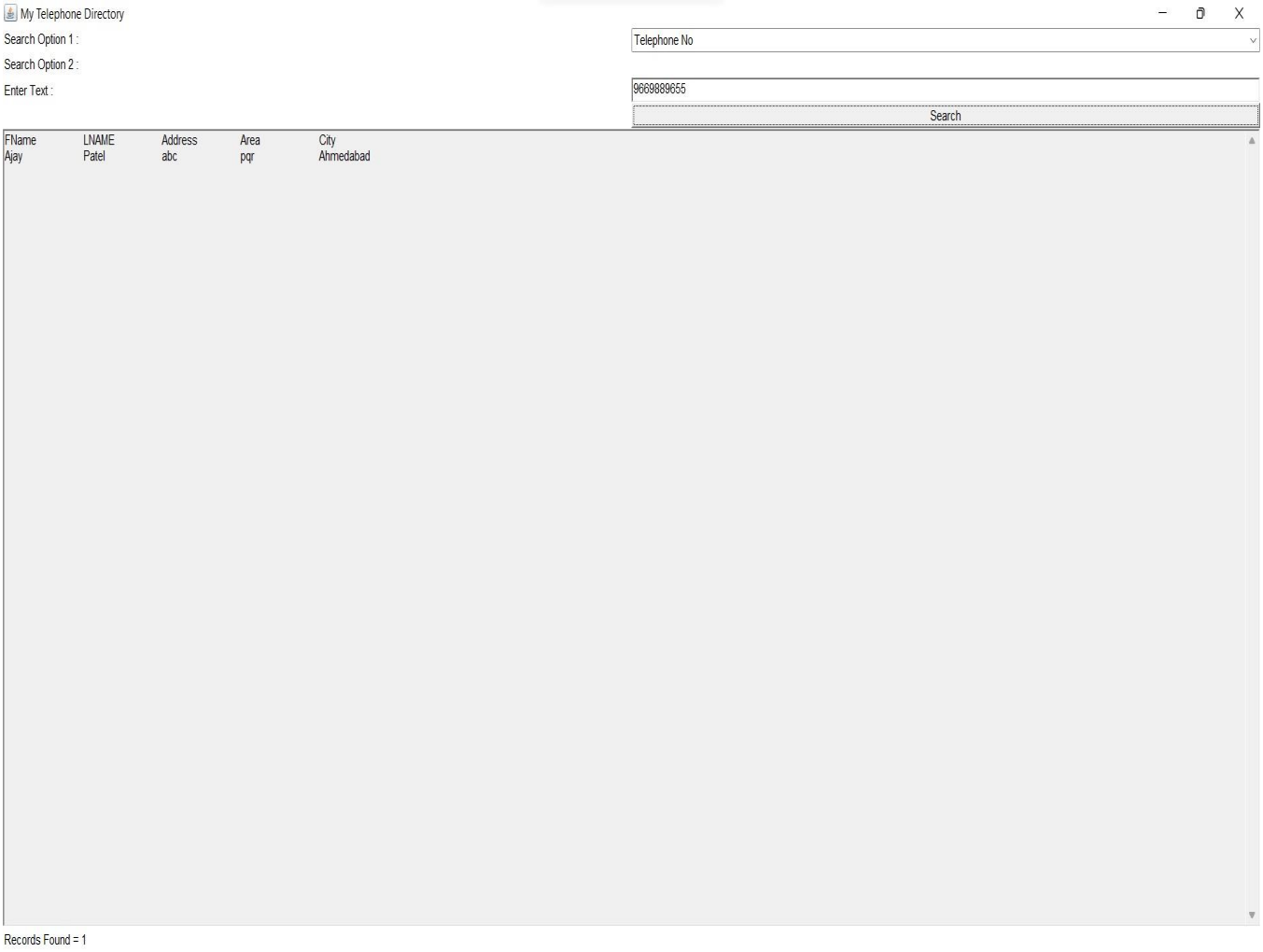
```
            System.out.println("Exception " + ee);
        }
    }

    public static void main(String[] args) {
        Frame dir = new Lab1();
    }
}
```

## Input/Output:

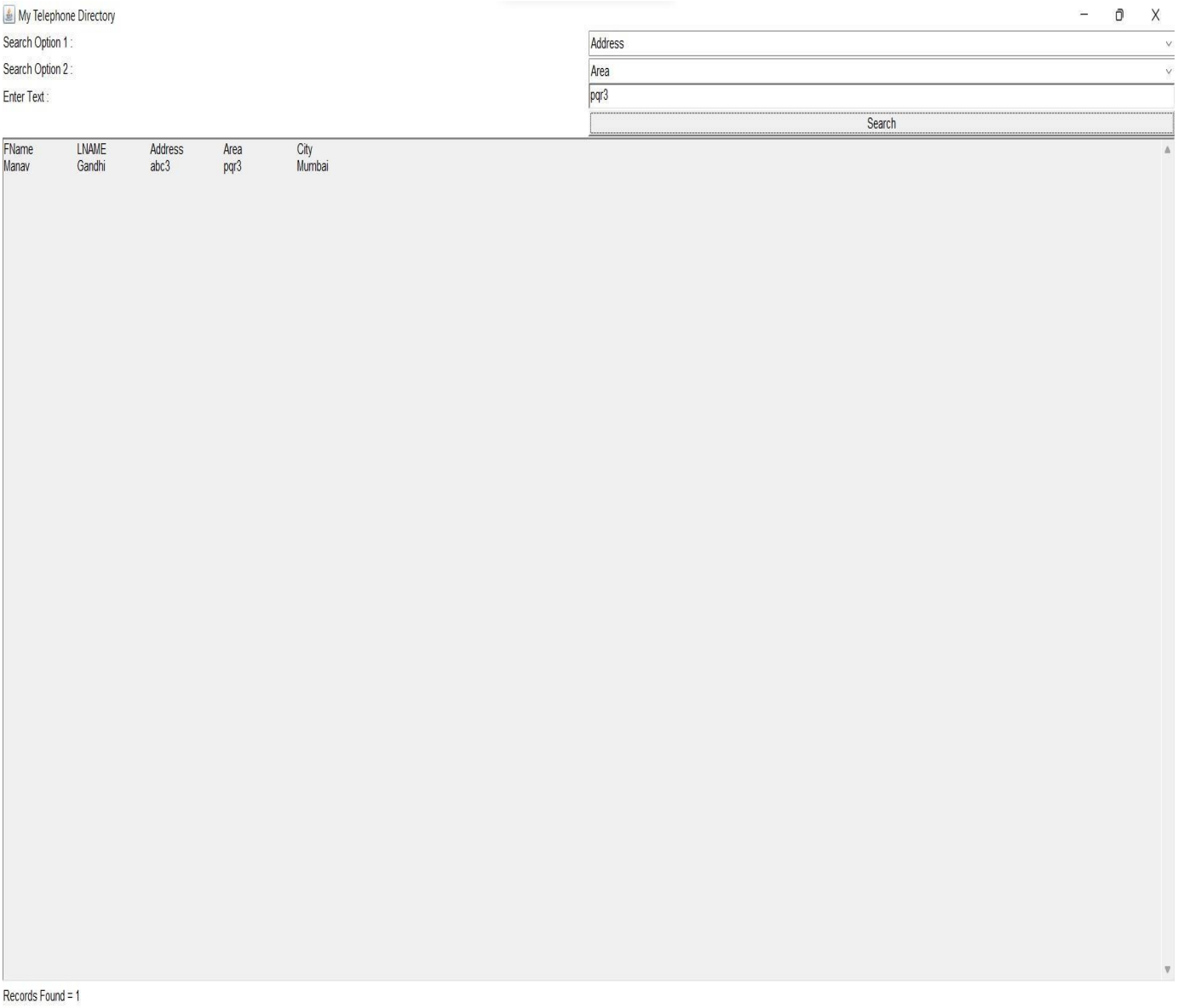
## 1.) Search by First Name



# 2.) Search by Mobile Number



# 3.) Search by Address



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

## Experiment – 02

**Submitted By:**
**Roll No.: IT083**
**Name: Nikhil Nasit**

**Aim:** 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.

Tools /Apparatus: JDK 1.6 or above , Netbeans IDE 6.1

**Code**:

```
package lab2;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class Lab2 extends Frame implements ItemListener, ActionListener {

    Panel pTop = new Panel();
    TextField tf1 = new TextField(20);
    TextField insertNumber = new TextField(20);
```

```
TextField insertFname = new TextField(20);
TextField insertLname = new TextField(20);
TextField insertAddress = new TextField(20);
TextField insertArea = new TextField(20);
TextField insertCity = new TextField(20);
TextField deleteData = new TextField(20);
TextArea ta1 = new TextArea(10, 200);
Choice c1 = new Choice();
Choice c2 = new Choice();
Button srchbutton = new Button("Search");
Button insertbutton = new Button("Insert");
Button deletebutton = new Button("Delete");
Label status = new Label("Records Found = 0 ");

String query = "";
Connection con = null;
PreparedStatement pstmt = null;
ResultSet rs = null;

public Lab2() {
   super("My Telephone Directory");
   setVisible(true);
   setSize(400, 800);

   c1.add("Telephone No");
   c1.add("Name");
   c1.add("Address");
   c1.addItemListener(this);
   srchbutton.addActionListener(this);
   insertbutton.addActionListener(this);
   deletebutton.addActionListener(this);

   addWindowListener(new WindowAdapter() {
      public void windowClosing(WindowEvent e) {
         dispose();
```

```
    }
});

setLayout(new BorderLayout());

pTop.setLayout(new GridLayout(13, 2));
pTop.add(new Label("Search Option 1 : "));
pTop.add(c1);
pTop.add(new Label("Search Option 2 : "));
pTop.add(c2);
pTop.add(new Label("Enter Text : "));
pTop.add(tf1);
pTop.add(new Label(""));
pTop.add(srchbutton);
pTop.add(new Label("Enter Number : "));
pTop.add(insertNumber);
pTop.add(new Label("Enter FirstName : "));
pTop.add(insertFname);
pTop.add(new Label("Enter LastName : "));
pTop.add(insertLname);
pTop.add(new Label("Enter Address : "));
pTop.add(insertAddress);
pTop.add(new Label("Enter Area : "));
pTop.add(insertArea);
pTop.add(new Label("Enter City : "));
pTop.add(insertCity);
pTop.add(new Label(""));
pTop.add(insertbutton);
pTop.add(new Label("Enter Number to delete data : "));
pTop.add(deleteData);
pTop.add(new Label(""));
pTop.add(deletebutton);


add("North", pTop);
```

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

        try {
            Class.forName("org.postgresql.Driver");
            con =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/postgres",
"postgres", "@bcde");
        } 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 e) {
        String arg = e.getItem().toString();
        if(arg.equals("Telephone No")){
            c2.setVisible(false);
        }
        else if(arg.equals("Name")){
            c2.removeAll();
            c2.add("First 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 e) {

    String str = e.getActionCommand();

    if(str.equals("Search")){
    ta1.setText("Refreshed");
    query= new String("select * from Directory");
    int len = 0;
    len = tf1.getText().toString().trim().length();

    try{
       if(c1.getSelectedItem().equals("Telephone No") && len>0){
          query += " where Number=?";
          pstmt=con.prepareStatement(query);
          pstmt.setString(1, tf1.getText().toString().trim());
       }
       else if(c1.getSelectedItem().equals("Name") && len>0){
         if(c2.getSelectedItem().equals(("First Name")))
         query += " where FirstName=?";
         else if(c2.getSelectedItem().equals(("Last Name")))
         query += " where LastName=?";
         pstmt = con.prepareStatement(query);
         pstmt.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=?";

          pstmt=con.prepareStatement(query);
          pstmt.setString(1,tf1.getText().toString().trim());
```

```
    }
    else{
       pstmt=con.prepareStatement(query);
    }


    try{
       System.out.println(query);
       rs=pstmt.executeQuery();
    }
    catch(NullPointerException ne){
       System.out.println("Text Null3");
       ta1.setText("No Records Found");
       status.setText("Records Found = 0");
    }

    if(rs!=null)
       ta1.setText("Number\t\tFName\t\tLNAME\t\tAddress\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) + "\n");
    count++;
    }
    status.setText("Records Found = " + count);
    }
    catch(Exception ee){
       System.out.println("Exception " + ee);
    }
}
```

```
if(str.equals("Insert")){
   try{
      String sql = "insert into Directory values(?,?,?,?,?,?)";
      pstmt=con.prepareStatement(sql);

      String number = insertNumber.getText().trim();
      String fname = insertFname.getText().trim();
      String lname = insertLname.getText().trim();
      String address = insertAddress.getText().trim();
      String area = insertArea.getText().trim();
      String city = insertCity.getText().trim();

      pstmt.setString(1, number);
      pstmt.setString(2, fname);
      pstmt.setString(3, lname);
      pstmt.setString(4, address);
      pstmt.setString(5, area);
      pstmt.setString(6, city);

      int count=pstmt.executeUpdate();

      System.out.println(count + " record inserted");

   }
   catch(Exception exp){
      System.out.println("Exception " + exp);
   }
}

   if(str.equals("Delete")){

      try{
         String sql = "delete from Directory where number = ?";
         pstmt=con.prepareStatement(sql);
```

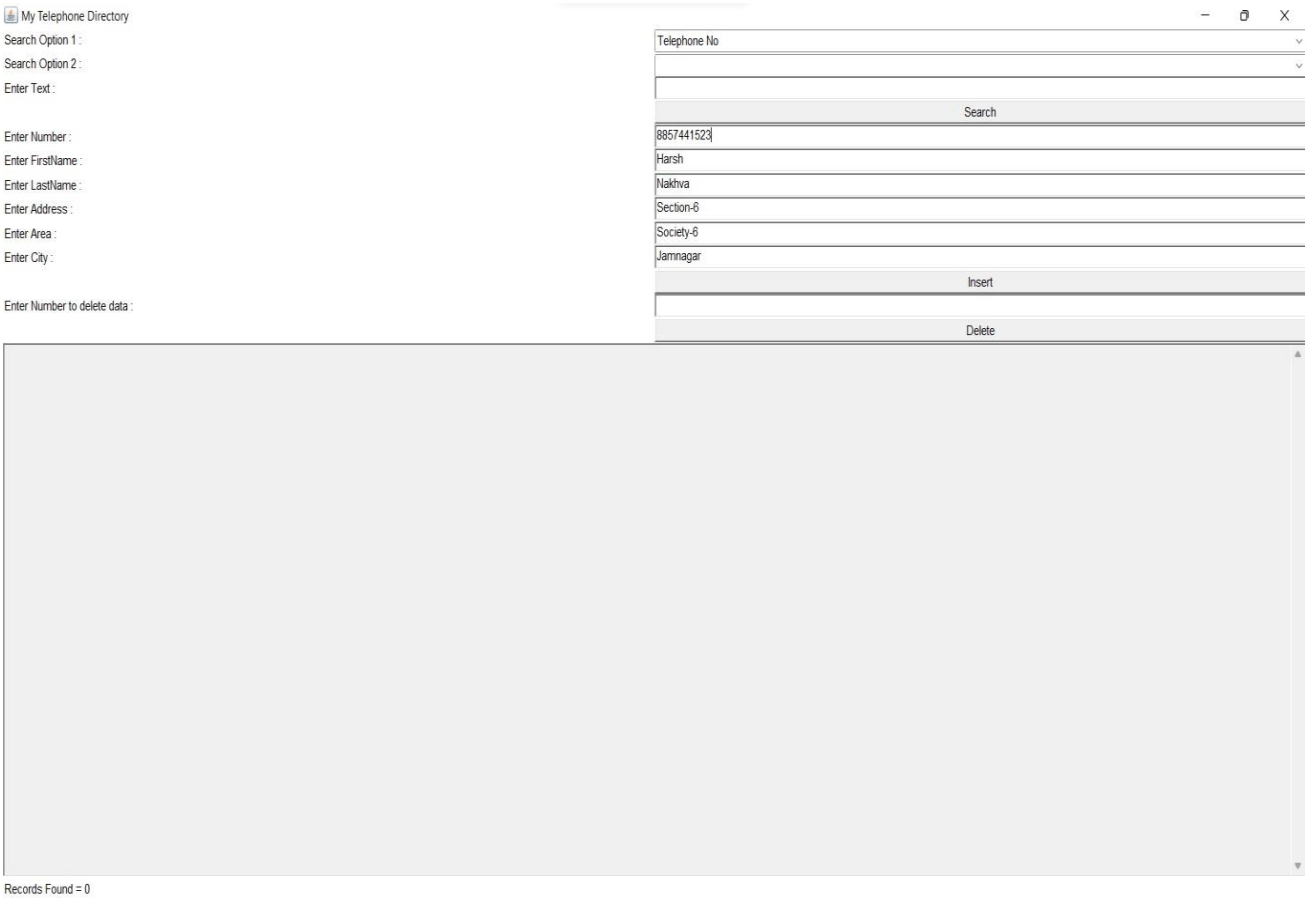
```
                String num = deleteData.getText().trim();

                pstmt.setString(1, num);

                int count = pstmt.executeUpdate();
                System.out.println(count + " record deleted");

            }
            catch(Exception exp){
                System.out.println("Exception " + exp);
            }
        }

    }

    public static void main(String[] args) {
        Frame dir = new Lab2();
    }
}
```

## Input/Output:

## 1.) Before inserting the record in the database

| # | number | firstname | lastname | address | area | city |
|---|---|---|---|---|---|---|
| 1 | 9638527411 | Deep | Changani | Section-2 | Society-2 | Jamnagar |
| 2 | 9874563211 | Utsav | Mungra | Section-3 | Society-3 | Jamnagar |
| 3 | 9876543211 | Jeet | Chovatiya | Section-4 | Society-4 | Jamnagar |
| 4 | 7412589633 | Parren | Vachhani | Section-5 | Society-5 | Navsari |
| 5 | 7412589633 | Aarya | Modi | Section-6 | Society-6 | Navsari |

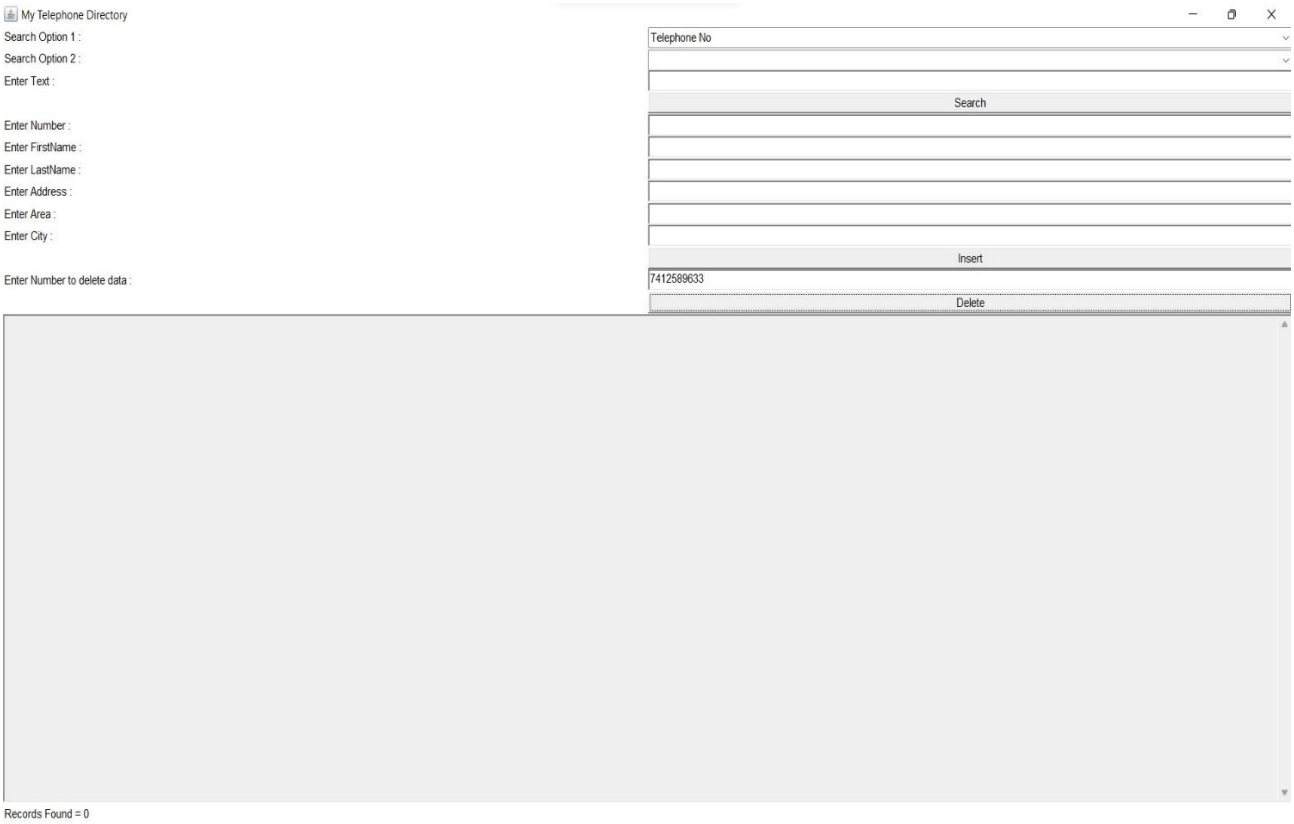
## Inserting the record



## After inserting the record in database

| # | number | firstname | lastname | address | area | city |
|---|---|---|---|---|---|---|
| 1 | 9638527411 | Deep | Changani | Section-2 | Society-2 | Jamnagar |
| 2 | 9874563211 | Utsav | Mungra | Section-3 | Society-3 | Jamnagar |
| 3 | 9876543211 | Jeet | Chovatiya | Section-4 | Society-4 | Jamnagar |
| 4 | 7412589633 | Parren | Vachhani | Section-5 | Society-5 | Navsari |
| 5 | 7412589633 | Aarya | Modi | Section-6 | Society-6 | Navsari |
| 6 | 8857441523 | Harsh | Nakhva | Section-6 | Society-6 | Jamnagar |

## 2.) Deleting the record by providing the Contact Number



## After deleting the record

| # | number | firstname | lastname | address | area | city |
|---|---|---|---|---|---|---|
| 1 | 9638527411 | Deep | Changani | Section-2 | Society-2 | Jamnagar |
| 2 | 9874563211 | Utsav | Mungra | Section-3 | Society-3 | Jamnagar |
| 3 | 9876543211 | Jeet | Chovatiya | Section-4 | Society-4 | Jamnagar |
| 4 | 8857441523 | Harsh | Nakhva | Section-6 | Society-6 | Jamnagar |

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

## Experiment – 03

**Submitted By:**
**Roll No.: IT083**
**Name: Nikhil Nasit**

**Aim:** 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.

Tools /Apparatus: JDK 1.6 or above , Netbeans IDE 6.1, Web Browser

**Code:**

**Index.html**

```
<html>
   <head>
      <title>Students Details</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <form method="post" action="NewServlet">
      User ID : <input type="text" name="userid"/><br><br>
      Password : <input type="password" name="password"/><br><br>
      Confirm Password : <input type="password"
name="confirmpassword"/><br><br>
      Full Name : <input type="text" name="fullname"/><br><br>
      Semester : <input type="number" name="semester"/><br><br>
```

```
        Roll No. : <input type="text" name="rollno"/><br><br>
        Email ID : <input type="email" name="emailid" required/><br><br>
        Contact No. : <input type="text" name="contactnumber"/><br><br>
        <input type="submit" value="submit" name="submit"/>
    </form>
</body>
</html>
```

**NewServlet.java**

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

public class NewServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
```

```
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
//            processRequest(request, response);

    }

    // <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 {

    try {
//          System.out.println("reheryheeh");
        String dbDriver = "org.postgresql.Driver";
        String dbURL = "jdbc:postgresql://localhost:5432/postgres";
        // Database name to access

        String dbUsername = "postgres";
        String dbPassword = "@bcde";

        Class.forName(dbDriver);
        Connection con = DriverManager.getConnection(dbURL, dbUsername,
dbPassword);

        PreparedStatement ps = con.prepareStatement("select * from
student_details where id=?");
        ps.setInt(1, Integer.parseInt(request.getParameter("userid")));
        ResultSet rs = ps.executeQuery();
        System.out.println("Result set");
        PrintWriter out = response.getWriter();

        if (rs.next()) {
//            System.out.println("NewServlet.processRequest()");
            out.println("<html><body><b>ID already exists!"
                    + "</b></body></html>");
        } else {

            PreparedStatement st = con.prepareStatement("insert into student_details
values(?,?,?,?,?,?,?) ");
            st.setInt(1, Integer.parseInt(request.getParameter("userid")));
            st.setString(2, request.getParameter("password"));
            st.setString(3, request.getParameter("fullname"));
            st.setInt(4, Integer.parseInt(request.getParameter("semester")));
            st.setInt(5, Integer.parseInt(request.getParameter("rollno")));
```

```
            st.setString(6, request.getParameter("emailid"));
            st.setString(7, (request.getParameter("contactnumber")));

            st.executeUpdate();
            st.close();
            con.close();

            out.println("<html><body><b>Successfully Inserted"
                  + "</b></body></html>");
          }
        } catch (Exception e) {
          e.printStackTrace();
        }

    }

    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

## Input/Output:

## 1.) Before inserting the record

| # | id | password | fullname | semester | rollno | email | contact_number |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

## Entering the data

User ID : 1

Password : ••••

Confirm Password : ••••

Full Name : Deep

Semester : 1

Roll No. : 1

Email ID : deep@gmail.com

Contact No. : 9866554767

submit

## After inserting the record

| # | id | password | fullname | semester | rollno | email | contact_number |
|---|---|---|---|---|---|---|---|
| 1 | 1 | deeo | Deep | 1 | 1 | deep@gmail.com | 9866554767 |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

## Output

**Successfully Inserted**

## 2.) Inserting the second record

User ID : 2

Password : •••••

Confirm Password : •••••

Full Name : Utsav

Semester : 2

Roll No. : 2

Email ID : utsav@gmail.com

Contact No. : 8779888766

submit

## After inserting the second record

| # | id | password | fullname | semester | rollno | email | contact_number |
|---|---|---|---|---|---|---|---|
| 1 | 1 | deeo | Deep | 1 | 1 | deep@gmail.com | 9866554767 |
| 2 | 2 | utsav | Utsav | 2 | 2 | utsav@gmail.com | 8779888766 |

## Output

**Successfully Inserted**

## 3.) Inserting the User ID that already exist in database

User ID : 1

Password : ••••

Confirm Password : ••••

Full Name : Jeet

Semester : 1

Roll No. : 1

Email ID : jeet@gmail.com

Contact No. : 8998766754

submit

## Output

ID already exists!

## No change in database

| # | id | password | fullname | semester | rollno | email | contact_number |
|---|---|---|---|---|---|---|---|
| 1 | 1 | deeo | Deep | 1 | 1 | deep@gmail.com | 9866554767 |
| 2 | 2 | utsav | Utsav | 2 | 2 | utsav@gmail.com | 8779888766 |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

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

## Experiment – 04

**Submitted By:**
**Roll No.: IT083**
**Name: Nikhil Nasit**

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

Tools /Apparatus: JDK 1.6 or above , Netbeans IDE 6.1, Web Browser

**Code:**

**Index.html**

```
<html>
   <head>
      <title>Main</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
      <h3><a href='login.html'>Click here to Login</a></h3>
      <h3><a href='Logout'>Logout</a></h3>
      <h3><a href='InformationServlet'>Click here to get your
Information</a></h3>
   </body>
</html>
```

**Login.html**

```
<html>
   <head>
      <title>Student Information</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
      <form action="LoginServlet" method="post">
         Enter your Id : <input type="number" name="studentid" /><br><br>
         Enter password : <input type="password" name="password"><br><br>
         <input type="submit" value="submit" />
      </form>
   </body>
</html>
```

**LoginServlet.java**

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.servlet.ServletConfig;
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 LoginServlet extends HttpServlet {
```

```
    Connection con;
    @Override
    public void init(ServletConfig config) {
        try {
            Class.forName("org.postgresql.Driver");
            con = DriverManager.getConnection(config.getInitParameter("dbUrl"),
config.getInitParameter("dbName"), config.getInitParameter("dbPassword")
            );
        }
        catch (ClassNotFoundException | SQLException e) {
        }
    }

    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
        }
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        int id = Integer.parseInt(request.getParameter("studentid"));
        String password = request.getParameter("password");
        try {

            PreparedStatement ps = con.prepareStatement("select * from
student_details where id=? and password=?");

            ps.setInt(1, id);
```

```
            ps.setString(2, password);

            ResultSet rs = ps.executeQuery();

            if(rs.next()){

               out.println("<h1>Login Successfully</h1>");
               HttpSession session = request.getSession();
               session.setAttribute("id", rs.getString(1));
               session.setAttribute("password", rs.getString(2));
               session.setAttribute("fullname", rs.getString(3));
               session.setAttribute("semester", rs.getString(4));
               session.setAttribute("rollno", rs.getString(5));
               session.setAttribute("email", rs.getString(6));
               session.setAttribute("contact_number", rs.getString(7));
               out.println("<h2><a href='InformationServlet'>Click here to get your
Information</a></h2>");
            }
            else{
               out.println("<h1>Invalid Credentials</h1>");
            }

         } catch (Exception e) {
            e.printStackTrace();
         }

      }

      @Override
      public String getServletInfo() {
         return "Short description";
      }// </editor-fold>

}
```

**InformationServlet.java**

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class InformationServlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
        }
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        response.setContentType("text/html;charset=UTF-8");
        HttpSession session = request.getSession(false);

        if(session!=null){
            out.println("<table>");

            out.println("<tr>");
            out.println("<th>Id</th>");
            out.println("<th>Password</th>");
            out.println("<th>Name</th>");
```

```
            out.println("<th>Semester</th>");
            out.println("<th>RollNo</th>");
            out.println("<th>Email</th>");
            out.println("<th>ContactNumber</th>");
            out.println("</tr>");

            out.println("<tr>");
            out.println("<td>" + session.getAttribute("id") + "</td>");
            out.println("<td>" + session.getAttribute("password") + "</td>");
            out.println("<td>" + session.getAttribute("fullname") + "</td>");
            out.println("<td>" + session.getAttribute("semester") + "</td>");
            out.println("<td>" + session.getAttribute("rollno") + "</td>");
            out.println("<td>" + session.getAttribute("email") + "</td>");
            out.println("<td>" +  session.getAttribute("contact_number") + "</td>");
            out.println("</tr>");
            out.println("</table>");
            out.println("<h2><a href='Logout'>Logout</a></h4>");
        }
        else{
            out.println("<h1>Something went wrong</h1>");
        }
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>
```

```
}
```

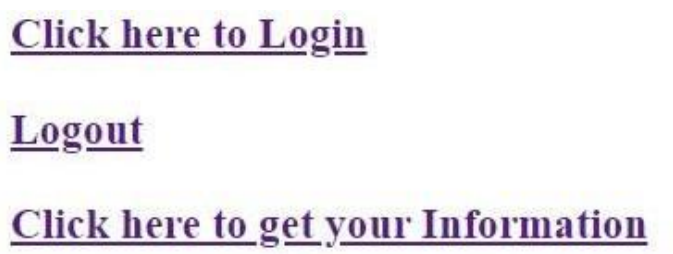
**Logout.java**

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class Logout extends HttpServlet {

   protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
         throws ServletException, IOException {
      response.setContentType("text/html;charset=UTF-8");
      try ( PrintWriter out = response.getWriter()) {
         /* TODO output your page here. You may use following sample code. */
         HttpSession session = request.getSession(false);
         if(session==null){
            out.println("<h1>Please Login</h1>");
         }
         else{
         session.invalidate();
         out.println("<h1>Logout successfully</h1>");
         }
         out.println("<h2><a href='index.html'>Index</a><h2>");
      }
   }

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

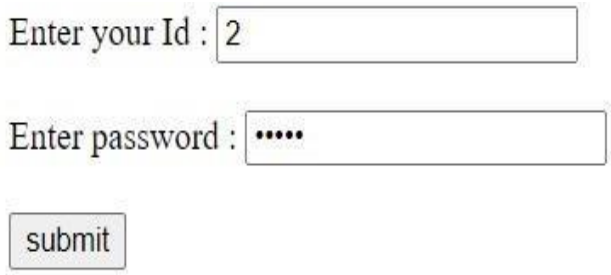
```
        processRequest(request, response);
    }
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

## Input/Output:

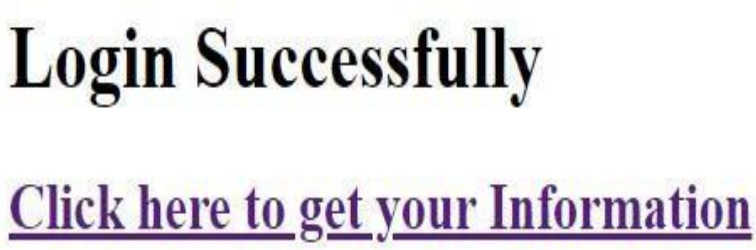
### 1.) Index Page




### 2.) Enter credentials




### 3.) Login Successfully



## 4.) Displaying student information

| Id | Password | Name | Semester | RollNo | Email | ContactNumber |
|---|---|---|---|---|---|---|
| 2 | utsav | Utsav | 2 | 2 | utsav@gmail.com | 8779888766 |

Logout

## 5.) Logout

# Logout successfully

Index

## 6.) Entering invalid Credentials

Enter your Id : 2

Enter password : ••••

submit

## 7.) Output

# Invalid Credentials

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

## Experiment – 05

**Submitted By:**
**Roll No.: IT083**
**Name: Nikhil Nasit**

**Aim:** 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.

Tools /Apparatus: JDK 1.6 or above , Netbeans IDE 6.1, Web Browser

**Code:**

**Index.html**

```
<html>
   <head>
      <title>Filter</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
      <form action="Welcome" method="post">
         ID:
         <input type="text" name="studentid" id="studentid"><br><br>
         Password: <input type="password" name="password"
id="password"><br><br>
         <input type="submit" value="Login" id="submit">
```

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

**LogFilter.java**

```
package com;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.Date;
import javax.servlet.*;

public class LogFilter implements Filter {

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {

    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response,
FilterChain chain) throws IOException, ServletException {
        try {
            System.out.println("LogFilter");
            PrintWriter out = response.getWriter();
            out.println(new Date());
            chain.doFilter(request, response);

        } catch (Throwable t) {
            t.printStackTrace();
        }
    }
```

```
    @Override
    public void destroy() {

    }

}
```

**AuthFilter.java**

```
package com;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.*;
import javax.servlet.http.HttpServletRequest;
public class AuthFilter implements Filter {

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {

    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response,
FilterChain chain) throws IOException, ServletException {
        try {
            HttpServletRequest req = (HttpServletRequest) request;
            int id = Integer.parseInt(req.getParameter("studentid"));
            String password = req.getParameter("password");

            if (id == 1 && password.equals("deeo")) {
                chain.doFilter(request, response);
            } else {
                PrintWriter out = response.getWriter();
                out.println();
```

```
            out.println("Sorry Invalid Credentials! Entry Restricted");
          }

      } catch (Throwable t) {
          t.printStackTrace();
      }
   }

   @Override
   public void destroy() {

   }

}
```

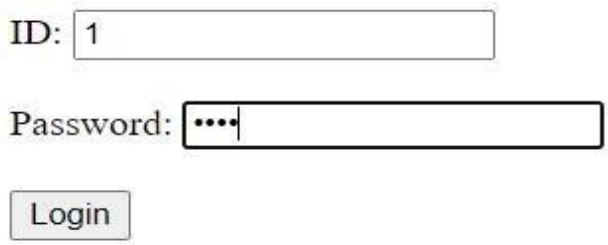
**WelcomeServlet.java**

```
package com;

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 Welcome extends HttpServlet {

   protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
          throws ServletException, IOException {
      response.setContentType("text/html;charset=UTF-8");
      try ( PrintWriter out = response.getWriter()) {
          /* TODO output your page here. You may use following sample code. */
            out.println();
```

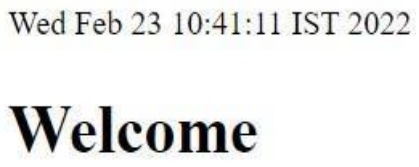
```
            out.println("<h1>Welcome</h1>");
        }
    }

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

}
```

## Input/Output:

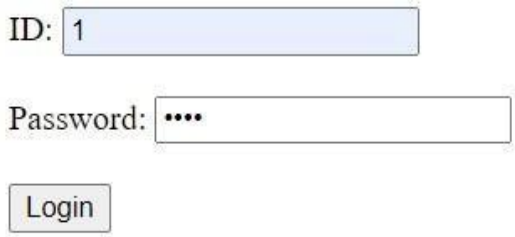
## 1.) Entering the valid ID and password




## Displaying the output via Filters




## 2.) Entering the wrong credentials



## Displaying the error message i.e. Invalid Credentials

```
Wed Feb 23 10:39:52 IST 2022

Sorry Invalid Credentials! Entry Restricted
```

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

## Experiment – 06

**Submitted By:**
**Roll No.: IT083**
**Name: Nikhil Nasit**

**Aim:** 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. Tools /Apparatus: JDK 1.6 or above , Netbeans IDE 6.1, Web Browser

**Code:**

**Index.jsp**

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>Student Information</title>
   </head>
   <body>
      <form action="LoginServlet" method="post">
         Enter your Id : <input type="number" name="studentid" /><br><br>
         Enter password : <input type="password" name="password"><br><br>
         <input type="submit" value="submit" />
      </form>
   </body>
```

```
</html>
```

**Display.jsp**

```
<%@page import="com.Student" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>Display Details</title>
   </head>
   <body>
      <jsp:useBean id="student" class="com.Student"
scope="session"></jsp:useBean>

      Id      :<jsp:getProperty property="id" name="student"/><br>
      Name    :<jsp:getProperty property="fullname" name="student"/><br>
      Roll No :<jsp:getProperty property="rollno" name="student"/><br>
      Semester:<jsp:getProperty property="semester" name="student"/><br>
      Email ID:<jsp:getProperty property="email" name="student"/><br>
      Mobile No.:<jsp:getProperty property="contact_number"
name="student"/><br>
   </body>
</html>
```

**DataBaseConnection.java**

```
package com;


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

public class DataBaseConnection {

   static Connection con = null;

   public static Connection getConnection() {
      try {
         if (con != null) {
            return con;
         } else {
            Class.forName("org.postgresql.Driver");
            con =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/postgres",
"postgres", "@bcde");
            return con;
         }
      } catch (Exception e) {
         e.printStackTrace();
      }
      return con;
   }
}
```

**Student.java**

```
package com;

public class Student {

    private int id;
    private String fullname;
    private String semester;
    private int rollno;
    private long contact_number;
    private String email;

    public Student() {
    }


    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getFullname() {
        return fullname;
    }

    public void setFullname(String fullname) {
        this.fullname = fullname;
    }

    public String getSemester() {
        return semester;
    }
```

```
    public void setSemester(String semester) {
        this.semester = semester;
    }

    public int getRollno() {
        return rollno;
    }

    public void setRollno(int rollno) {
        this.rollno = rollno;
    }

    public long getContact_number() {
        return contact_number;
    }

    public void setContact_number(long contact_number) {
        this.contact_number = contact_number;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

}
```

**LoginServlet.java**

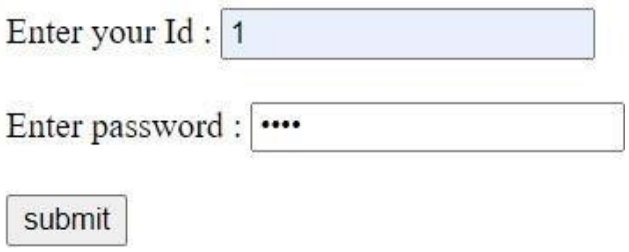
```
package com;

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.RequestDispatcher;
import javax.servlet.ServletConfig;
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 LoginServlet extends HttpServlet {

   protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
           throws ServletException, IOException {
       response.setContentType("text/html;charset=UTF-8");
       try ( PrintWriter out = response.getWriter()) {
           /* TODO output your page here. You may use following sample code. */

       }
   }

   @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 {
        PrintWriter out = response.getWriter();

        int id = Integer.parseInt(request.getParameter("studentid"));
        String password = request.getParameter("password");
        try {

            Connection con = DataBaseConnection.getConnection();
            PreparedStatement ps = con.prepareStatement("select * from
student_details where id=? and password=?");

            ps.setInt(1, id);
            ps.setString(2, password);

            ResultSet rs = ps.executeQuery();

            if (rs.next()) {

                out.println("<h1>Login Successfully</h1>");
                Student st = new Student();
                st.setId(rs.getInt(1));
                st.setFullname(rs.getString(3));
                st.setSemester(rs.getString(4));
                st.setRollno(rs.getInt(5));
                st.setEmail(rs.getString(6));
                st.setContact_number(rs.getLong(7));

                HttpSession session = request.getSession();
                session.setAttribute("student", st);
```

```
            RequestDispatcher rd = request.getRequestDispatcher("display.jsp");
            rd.forward(request, response);
        } else {
            out.println("<h1>Invalid Credentials</h1>");
        }

    } catch (Exception e) {
        e.printStackTrace();
    }
}

@Override
public String getServletInfo() {
    return "Short description";
}// </editor-fold>

}
```

## Input/Output:

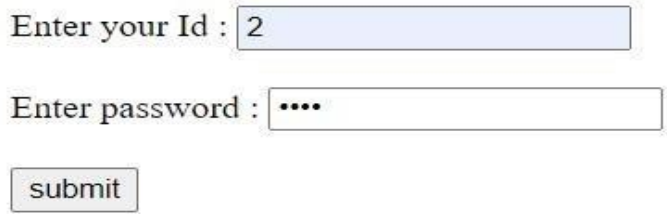
## 1.) Entering the details




## Output:

Id :1
Name :Deep
Roll No :1
Semester:1
Email ID:deep@gmail.com
Mobile No.:9866554767

## 2.) Entering invalid credentials




## Output:

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

## Experiment – 07

**Submitted By:**
**Roll No.: IT083**
**Name: Nikhil Nasit**

**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.

Tools /Apparatus: JDK 1.6 or above , Netbeans IDE 6.1, Web Browser

**Code:**

**index.html**

```
<!DOCTYPE html>
<html>
   <head>
      <title>Login</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
      <h1>Enter Your Details For Login</h1>
      <form action="Authentication" method="post">
         ID: <input type="text" name="studentid" id="studentid"><br><br>
```

```
        Password: <input type="password" name="password"
id="password"><br><br>
        <input type="submit" value="Login" id="submit">

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

**Authentication.java**

```
package com;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.servlet.RequestDispatcher;
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 Authentication extends HttpServlet {
   protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
      response.setContentType("text/html;charset=UTF-8");
      try ( PrintWriter out = response.getWriter()) {
         /* TODO output your page here. You may use following sample code. */
         int id = Integer.parseInt(request.getParameter("studentid"));
         String password = request.getParameter("password");
         String pswd = "";
```

```
            try {
                Connection con = DataBaseConnection.getConnection();
                Statement stm = con.createStatement();
                ResultSet rs = stm.executeQuery("select * from student_details where id
='" + id + "'");

                if (rs.next()) {
                    pswd = rs.getString(2);
                    if (password.equals(pswd))  {
                        HttpSession hs = request.getSession();
                        hs.setAttribute("id", id);
                        hs.setAttribute("password", password);

                        stm.executeUpdate("delete from student_details where id='" + id +
"'");

                        RequestDispatcher rd =
request.getRequestDispatcher("display.jsp");
                        rd.forward(request, response);
                    } else {
                        out.println("<h1>Incorrect Password!</h1>");
                    }
                } else {
                    out.println("<h1>User not Found! Try again.</h1>");
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }

        }
    }

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

```
    @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";
    }// </editor-fold>

}
```

**display.jsp**

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

<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Display Page</title>
    </head>
    <body>
        <h1>Enter details to be updated.</h1>
        <h3>Enter details in all fields.</h3>
        <form action="update.jsp" method="post">
```

```
        Full Name: <input type="text" name="fname" id="fname"><br><br>
        Semester: <input type="text" name="sem" id="sem"><br><br>
        Roll No: <input type="text" name="rollno" id="rollno"><br><br>
        Email id: <input type="email" name="email" id="email"><br><br>
        Contact number: <input type="text" name="contact"
id="contact"><br><br>
        <input type="submit" value="Update" id="submit">
      </form>
   </body>
</html>
```

**update.jsp**

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>Update Details</title>
      <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
      <%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
   </head>

   <body>
      <sql:setDataSource var="ds" driver="org.postgresql.Driver"
                  url="jdbc:postgresql://localhost:5432/postgres"
                  user="postgres"  password="@bcde"/>

      <c:set var="id" value="${sessionScope.id}"></c:set>
      <c:set var="password" value="${sessionScope.password}"></c:set>
      <c:set var="name" value="${param.fname}"></c:set>
      <c:set var="sem" value="${param.sem}"></c:set>
      <c:set var="rollno" value="${param.rollno}"></c:set>
      <c:set var="email" value="${param.email}"></c:set>
      <c:set var="mno" value="${param.contact}"></c:set>
```

```
        <sql:update dataSource="${ds}" var="updateCount">
            insert into student_details values
('${id}','${password}','${name}',${sem},${rollno},'${email}',${mno})
        </sql:update>
        <h1>Details Updated successfully!</h1>
    </body>
</html>
```

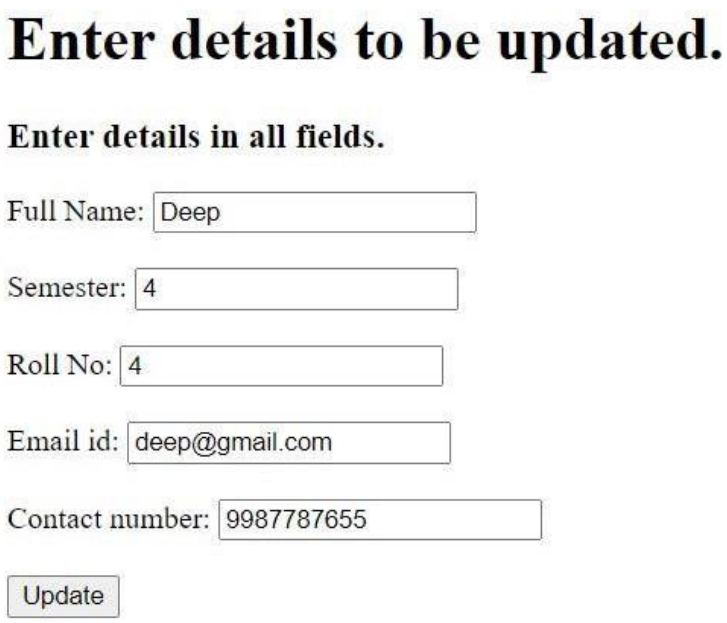
**Input/Output:**

**1.) Before updating the record in Database**

| # | id | password | fullname | semester | rollno | email | contact_number |
|---|---|---|---|---|---|---|---|
| 1 | 2 | utsav | Utsav | 2 | 2 | utsav@gmail.com | 8779888766 |
| 2 | 1 | deeo | Deep | 1 | 1 | deep11@gmail.com | 9987665576 |

**2.) Entering credentials**

## 3.) Entering details to be updated




## 4.) Details updated successfully


## 5.) After updating the record in Database

| # | id | password | fullname | semester | rollno | email | contact_number |
|---|---|---|---|---|---|---|---|
| 1 | 2 | utsav | Utsav | 2 | 2 | utsav@gmail.com | 8779888766 |
| 2 | 1 | deeo | Deep | 4 | 4 | deep@gmail.com | 9987787655 |

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

## Experiment – 08

**Submitted By:**
**Roll No.: IT083**
**Name: Nikhil Nasit**

**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.

Tools /Apparatus: JDK 1.6 or above , Netbeans IDE 6.1, Web Browser

**Code:**
**Index.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>
      <p>Request a secure Admin page <a
            href="secureAdmin/pageA.html">here!</a></p>

      <p>Request a secure User page <a
            href="secureUser/pageU.html" >here!</a></p>
   </body>
</html>
```

**PageA.html**

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>Admin secure area</title>
    </head>
    <body>
        <h1>Admin secure area</h1>
    </body>
</html>
```

**PageU.html**

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>User secure area</title>
    </head>
    <body>
        <h1>User Secure Area</h1>
    </body>
</html>
```

**web.xml**

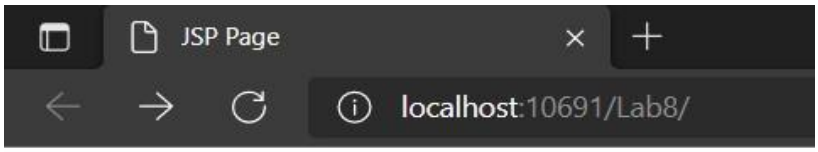
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
   <session-config>
      <session-timeout>
         30
      </session-timeout>
   </session-config>
   <security-constraint>
      <display-name>AdminConstraint</display-name>
      <web-resource-collection>
         <web-resource-name>Admin</web-resource-name>
         <description/>
         <url-pattern>/secureAdmin/*</url-pattern>
      </web-resource-collection>
      <auth-constraint>
         <description/>
         <role-name>AdminRole</role-name>
      </auth-constraint>
   </security-constraint>
   <security-constraint>
      <display-name>UserConstraint</display-name>
      <web-resource-collection>
         <web-resource-name>User</web-resource-name>
         <description/>
         <url-pattern>/secureUser/*</url-pattern>
      </web-resource-collection>
      <auth-constraint>
         <description/>
         <role-name>AdminRole</role-name>
         <role-name>UserRole</role-name>
      </auth-constraint>
   </security-constraint>
   <login-config>
      <auth-method>BASIC</auth-method>
      <realm-name>file</realm-name>
   </login-config>
   <security-role>
      <description>Admin can access
</description>
```

```
      <role-name>AdminRole</role-name>
    </security-role>
    <security-role>
      <description>User can access
</description>
      <role-name>UserRole</role-name>
    </security-role>
</web-app>
```

**glassfish-web.xml**

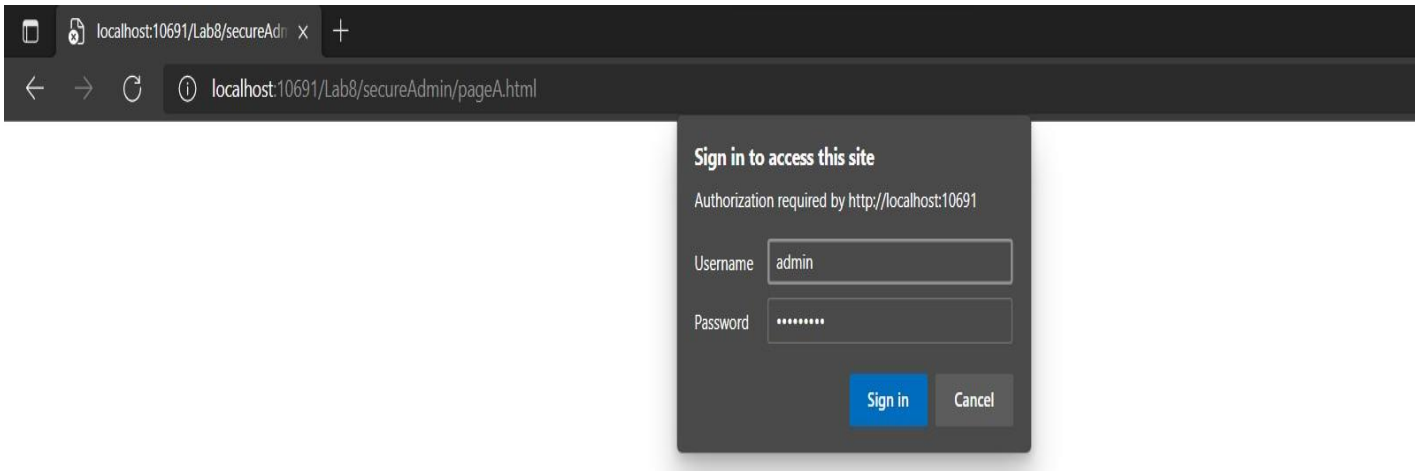
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE glassfish-web-app PUBLIC "-//GlassFish.org//DTD GlassFish Application
Server 3.1 Servlet 3.0//EN" "http://glassfish.org/dtds/glassfish-web-app_3_0-1.dtd">
<glassfish-web-app error-url="">
  <security-role-mapping>
    <role-name>AdminRole</role-name>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

## Output:




Request a secure Admin page here!

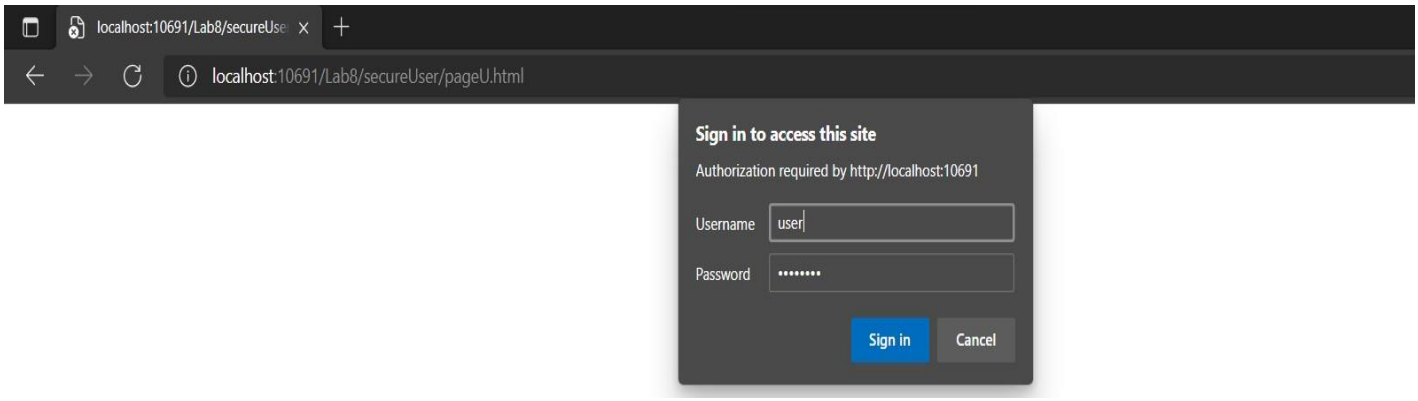
Request a secure User page here!





# Admin secure area

\





# User Secure Area

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

## Experiment – 09

**Submitted By:**
**Roll No.: IT083**
**Name: Nikhil Nasit**

**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.

Tools /Apparatus: JDK 1.6 or above , Netbeans IDE 6.1, Web Browser

**Code:**
**index.jsp**

```
<%@taglib uri="WEB-INF/mytag.tld" prefix="m"%>
<%@taglib uri="WEB-INF/mytag.tld" prefix="my"%>

<%@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>
   Current Date and Time is: <m:today/>
   <my:customtag align="left"/>

</html>
```

**context.html**

```
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/Lab9"/>
```

**mytag.tlg**

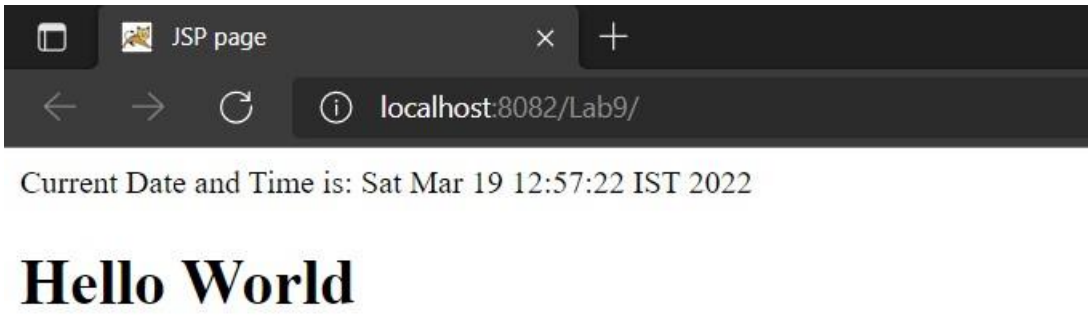
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE taglib PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.2//EN"
"http://java.sun.com/dtd/web-jsptaglibrary_1_2.dtd">

<taglib>
  <tlib-version>1.0</tlib-version>
  <jsp-version>1.2</jsp-version>
  <tag>
    <name>today</name>
    <tag-class>com.MyTagHandler</tag-class>
  </tag>
  <tag>
    <name>customtag</name>
    <tag-class>com.MyTagHandler</tag-class>
    <attribute>
      <name>align</name>
      <required>true</required>
    </attribute>
  </tag>
</taglib>
```

**MyTagHandler.java**

```
package com;

import java.util.Calendar;
import javax.lang.model.SourceVersion;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.tagext.SimpleTagSupport;
import javax.servlet.jsp.tagext.TagSupport;
public class MyTagHandler extends TagSupport{

  private String align;
```

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


public int doStartTag() throws JspException {
    JspWriter out=pageContext.getOut();
    try{
     out.print(Calendar.getInstance().getTime());
     out.println("<h1 align="+this.align+">Hello World</h1>");
    }catch(Exception e)
    {System.out.println(e);}
    return SKIP_BODY;
}
}
```

**Output:**



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

## Experiment – 10

**Submitted By:**
**Roll No.: IT083**
**Name: Nikhil Nasit**

**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. The server stores account data in database. Design appropriate interface and test implementation on network.
Tools /Apparatus: Editor (Notepad), command prompt.

**Code:**
**BankInterface.java**

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Interface.java to edit this template
 */
package lab10;
import java.rmi.*;
/**
 *
 * @author Dell
 */
public interface BankInterface extends Remote{
   public String deposit(int accno,double amt) throws RemoteException;
   public String withdraw(int accno,double amt) throws RemoteException;
   public String checkBalance(int accno) throws RemoteException;
}
```

**BankProxy.java**

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package lab10;

import java.rmi.*;
import java.sql.*;
import java.rmi.server.*;

/**
 *
 * @author DELL
 */

public class BankProxy extends UnicastRemoteObject implements BankInterface {

   String url = "jdbc:postgresql://localhost:5432/postgres";
   String driver = "org.postgresql.Driver";
   Connection con = null;
   PreparedStatement ps = null;
   ResultSet rs = null;

   BankProxy() throws RemoteException {
      try {
         Class.forName(driver);
         con = DriverManager.getConnection(url, "postgres", "@bcde");
      } catch (ClassNotFoundException | SQLException se) {
         se.printStackTrace();
      }
   }

   @Override
   public String deposit(int accno, double amt) throws RemoteException {
      String viewBalance = new String("Select amt from Bank where accno=?");
      double balance = 0.0;
      String accHolder = "";

      try {
         ps = con.prepareStatement(viewBalance);
```

```
            ps.setInt(1, accno);

            rs = ps.executeQuery();

            while (rs.next()) {
                accHolder = rs.getString(2);
                balance = rs.getDouble(3);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        if (amt < 0.0) {
            return new String("Invalid Deposit Amount:\nAmount must be greater than 0.0");
        } else {
            String q = "Update Bank set amt=? where accno=?";

            try {
                ps = con.prepareStatement(q);
                ps.setDouble(1, balance + amt);
                ps.setInt(2, accno);
                ps.executeUpdate();

                System.out.println("Account number : " + accno);
                System.out.println("Account Holder : " + accHolder);
                System.out.println("Account Balance : " + balance);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return "Deposited Successfully\n";
    }

    @Override
    public String withdraw(int accno, double amt) throws RemoteException {
        String viewBalance = new String("Select amt from Bank where accno=?");
        double balance = 0.0;
        String accHolder = "";

        try {
            ps = con.prepareStatement(viewBalance);
            ps.setInt(1, accno);

            rs = ps.executeQuery();
```

```
            while (rs.next()) {
                accHolder = rs.getString(2);
                balance = rs.getDouble(3);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        if (amt < 0.0) {
            return new String("Invalid Withdrawal Amount:\nAmount must be greater than 0.0");
        } else if (amt > balance) {
            return new String("Insufficient Balance\n");
        } else {
            String q = "Update Bank set amt=? where accno=?";

            try {
                ps = con.prepareStatement(q);
                ps.setDouble(1, balance - amt);
                ps.setInt(2, accno);
                ps.executeUpdate();

                System.out.println("Account number : " + accno);
                System.out.println("Account Holder : " + accHolder);
                System.out.println("Account Balance : " + balance);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return "Witdrawal Successful\n";
    }

    @Override
    public String checkBalance(int accno) throws RemoteException {
        String viewBalance = new String("Select amt from Bank where accno=?");
        double balance = 0.0;
        String accHolder = "";

        try {
            ps = con.prepareStatement(viewBalance);
            ps.setInt(1, accno);

            rs = ps.executeQuery();

            while (rs.next()) {
```

```
            accHolder = rs.getString(2);
            balance = rs.getDouble(3);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }

    System.out.println("Account number : " + accno);
    System.out.println("Account Holder : " + accHolder);
    System.out.println("Account Balance : " + balance);

    return "Your Balanace is : " + balance + "\n";
  }
}
```

**BankClient.java**

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package lab10;

import java.rmi.*;
import java.rmi.registry.LocateRegistry;

/**
 *
 * @author DELL
 */
public class BankClient {

  public static void main(String[] args) {
    try {
//            System.setSecurityManager(new java.rmi.RMISecurityManager());
//          registry = LocateRegistry.createRegistry( 1099 );
        String bankServerURL = "rmi://localhost:1099/BankServer";
        BankInterface bank = (BankInterface) Naming.lookup(bankServerURL);

        System.out.println("Account : " + bank.checkBalance(10001));
        System.out.println("Account : " + bank.deposit(10001, 3000.0));
        System.out.println("Account : " + bank.withdraw(10002, 5000.0));
        System.out.println("Account : " + bank.checkBalance(10004));
```

```
            System.out.println("Account : " + bank.deposit(10004, 40000.0));
        } catch (Exception e) {
            System.out.println("Exception " + e);
        }
    }
}
```

**BankServer.java**

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package lab10;

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

/**
 *
 * @author DELL
 */
public class BankServer {


    /**
     * @param args the command line arguments
     */
//    BankServer()
//    {
//
//    }
    public static void main(String[] args) {
        // TODO code application logic here
        try {
            BankInterface bank = new BankProxy();
            Naming.rebind("rmi://localhost:1099/BankServer", bank);
        } catch (Exception e) {
            System.out.println("Exception " + e);
        }
    }

}
```

## Output:

| # | accno | accholder | amt |
|---|---|---|---|
| 1 | 10001 | Deep Changani | 10207.00 |
| 2 | 10002 | Utsav Mungra | 35900.50 |
| 3 | 10003 | Jeet Covatiya | 59799.00 |
| 4 | 10004 | Harsh Nakhva | 60090.00 |
| 5 | 10005 | Hem Solanki | 69462.00 |

```
D:\netbeans\AJT\Labs\Lab10\src\lab10>rmic lab10.BankProxy
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:\netbeans\AJT\Labs\Lab10\src\lab10>start rmiregistry

D:\netbeans\AJT\Labs\Lab10\src\lab10>java lab10.BankServer
java.lang.ClassNotFoundException: org.postgresql.Driver
        at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:418)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:355)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:351)
        at java.lang.Class.forName0(Native Method)
        at java.lang.Class.forName(Class.java:264)
        at lab10.BankProxy.<init>(BankProxy.java:27)
        at lab10.BankServer.main(BankServer.java:27)

D:\netbeans\AJT\Labs\Lab10\src\lab10>set classpath=%classpath%;.; D:\netbeans\AJT\Labs\Lab10\src\lab10;D:\netbeans\postgresql-42.3.1.jar

D:\netbeans\AJT\Labs\Lab10\src\lab10>java lab10.BankServer
```

```
D:\netbeans\AJT\Labs\Lab10\src\lab10>set classpath=%classpath%;.; D:\netbeans\AJT\Labs\Lab10\src\lab10;D:\netbeans\postgresql-42.3.1.jar

D:\netbeans\AJT\Labs\Lab10\src\lab10>javac -d . BankClient.java

D:\netbeans\AJT\Labs\Lab10\src\lab10>java lab10.BankClient
Account : Your Balanace is : 0.0

Account : Deposited Successfully

Account : Insufficient Balance

Account : Your Balanace is : 0.0

Account : Deposited Successfully


D:\netbeans\AJT\Labs\Lab10\src\lab10>
```

| # | accno | accholder | amt |
|---|---|---|---|
| 1 | 10002 | Utsav Mungra | 35900.50 |
| 2 | 10003 | Jeet Covatiya | 59799.00 |
| 3 | 10005 | Hem Solanki | 69462.00 |
| 4 | 10001 | Deep Changani | 3000.00 |
| 5 | 10004 | Harsh Nakhva | 40000.00 |

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

## Experiment – 11

**Submitted By:**
**Roll No.: IT083**
**Name: Nikhil Nasit**

**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.

Tools /Apparatus: JDK 1.6 or above , Netbeans IDE 6.1, Web Browser

**Code:**
**index.html**

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>JSP Page</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
```

```
        <h1>Income Tax Calculation</h1>
        <!--<a href="myservlet">Click here to visit stateless session bean</a>-->

        <form method="post" action="IncomeServlet">
            Enter Annual Salary : <input type="text" name="annualSalary"/>
            <br><br>
            <input type="submit" name="submit"/>
        </form>
    </body>
</html>
```

**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>IncomeServlet</servlet-name>
        <servlet-class>test.IncomeServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>IncomeServlet</servlet-name>
        <url-pattern>/IncomeServlet</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
</web-app>
```

**IncomeServlet.java**

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package test;

import java.io.IOException;
import java.io.PrintWriter;
import javax.ejb.EJB;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author DELL
 */
public class IncomeServlet extends HttpServlet {

    @EJB
    private IncomeTaxBeanLocal incomeTaxBean;

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        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>Income Servlet</title>");
            out.println("</head>");
```

```
            out.println("<body>");
            String str1 = request.getParameter("annualSalary");

            double n1 = Double.parseDouble(str1);
            out.println("<h1>Servlet Income Servlet at " + request.getContextPath() + "</h1>");

            out.println("Tax on AnnualSalary of " + str1 + " is : " +
incomeTaxBean.calculateTax(n1));

            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>

}
```

**IncomeTaxBean.java**

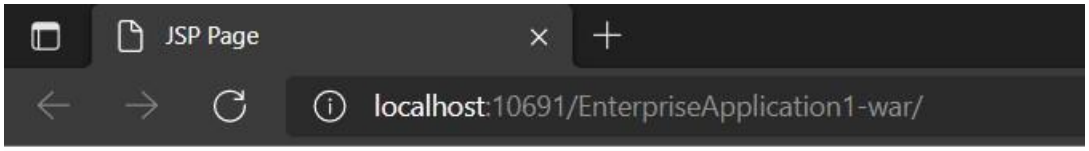
```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package test;

import javax.ejb.Stateless;

/**
 *
 * @author DELL
 */
@Stateless
public class IncomeTaxBean implements IncomeTaxBeanLocal {

    @Override
    public double calculateTax(double annualSalary) {
        double tax = 0.0;
        double sal= annualSalary;
//        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.
        if(sal>=100000.0)
        {
            sal-=100000.0;
        }
        if(sal>=100000.0)
        {
            tax += 0.1*(100000.0);
```

```
            sal-=100000.0;
        }
        if(sal>=100000.0)
        {
            tax += 0.2*(100000.0);
            sal-=100000.0;
        }
        if(sal>=100000.0)
        {
            tax += 0.3*(100000.0);
            sal-=100000.0;
        }
        tax+=sal;
        return tax;
    }

    // Add business logic below. (Right-click in editor and choose
    // "Insert Code > Add Business Method")

}
```

**IncomeTaxBeanLocal.java**

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package test;

/**
 *
 * @author Dell
 */
public interface IncomeTaxBeanLocal {
    public double calculateTax(double annualSalary);
}
```

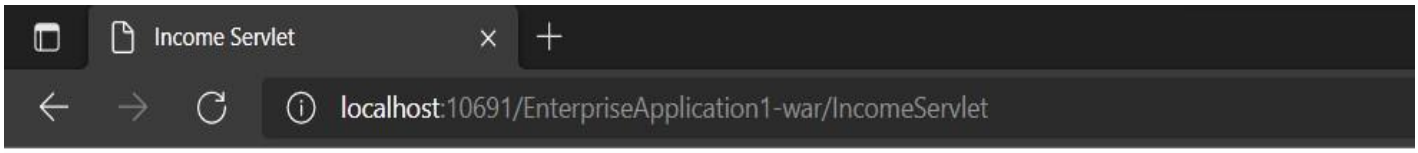
**Output:**




# Income Tax Calculation

Enter Annual Salary : 455443

Submit




# Servlet Income Servlet at /EnterpriseApplication1-war

Tax on AnnualSalary of 455443 is : 115443.0