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

Experiment – 01

Submitted By: 19ituos121
Roll No.: IT069
Name: Arjun Maniya

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.

Code:

// Source code

EX1.java :

```
package ex1;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
/**
 *
 * @author admin
 */
public class Ex1 {

    public static void main(String[] args) {
        try {
            Class.forName("org.postgresql.Driver");
            Connection conn =
DriverManager.getConnection("jdbc:postgresql://localhost:54
32/postgres","postgres","shree9592");
            new Gui("ex1",conn);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

}
```

Gui.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 ex1;

import com.sun.java.accessibility.util.AWTEventMonitor;
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.*;

/**
 *
 * @author admin
 */
public class Gui extends Frame implements ItemListener,
ActionListener {
```

```
    private Choice searchList = new Choice();
    private Choice filt = new Choice();
    private TextField tf1 = new TextField(20);
    private TextArea ta1 = new TextArea(10, 100);
    Button submit = new Button("submit");
    String query = "";
    PreparedStatement pstate = null;
    Connection conn = null;
    ResultSet rs = null;
    Label status = new Label("Result Found : 0 ");

    Gui(String name, Connection conn) {
        super(name);
        setVisible(true);
        setSize(400, 400);
        setLayout(new BorderLayout());
        this.conn = conn;
        Panel p = new Panel();
        p.setLayout(new GridLayout(4, 2));
        searchList.addItemListener(this);
        searchList.add("Telephone Number");
        searchList.add("Name");
        searchList.add("Address");
        p.add(new Label("Search By : "));
        p.add(searchList);
        p.add(new Label("Filters : "));
        p.add(filt);
        filt.setVisible(false);
        p.add(new Label("Enter detail :"));
        p.add(tf1);
        p.add(new Label(""));
        p.add(submit);
        add("North", p);
```

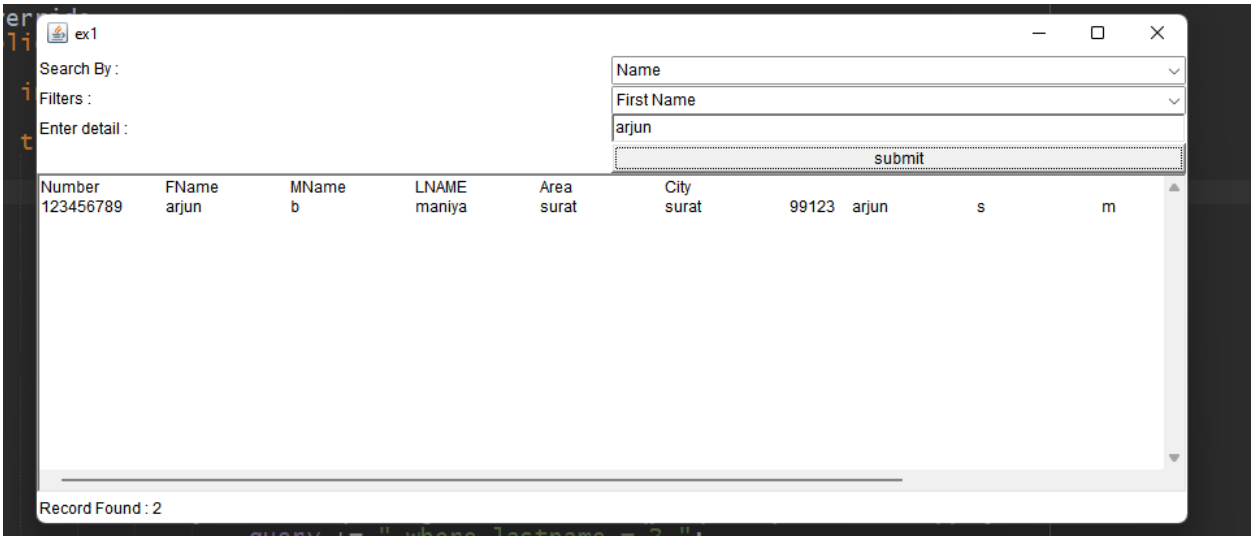
```
        add("Center", ta1);
        add("South", status);
        addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });

        submit.addActionListener(this);
        searchList.addItemListener(this);

    }

    @Override
    public void itemStateChanged(ItemEvent arg0) {
        String arg = arg0.getItem().toString();
        if (arg.equals("Telephone Number")) {
            filt.setVisible(false);
        } else if (arg.equals("Name")) {
            filt.removeAll();
            filt.setVisible(true);
            filt.add("First Name");
            filt.add("Middle Name");
            filt.add("Last Name");
        } else if (arg.equals("Address")) {
            filt.removeAll();
            filt.add("Area");
            filt.add("City");
            filt.setVisible(true);
        }
    }
```

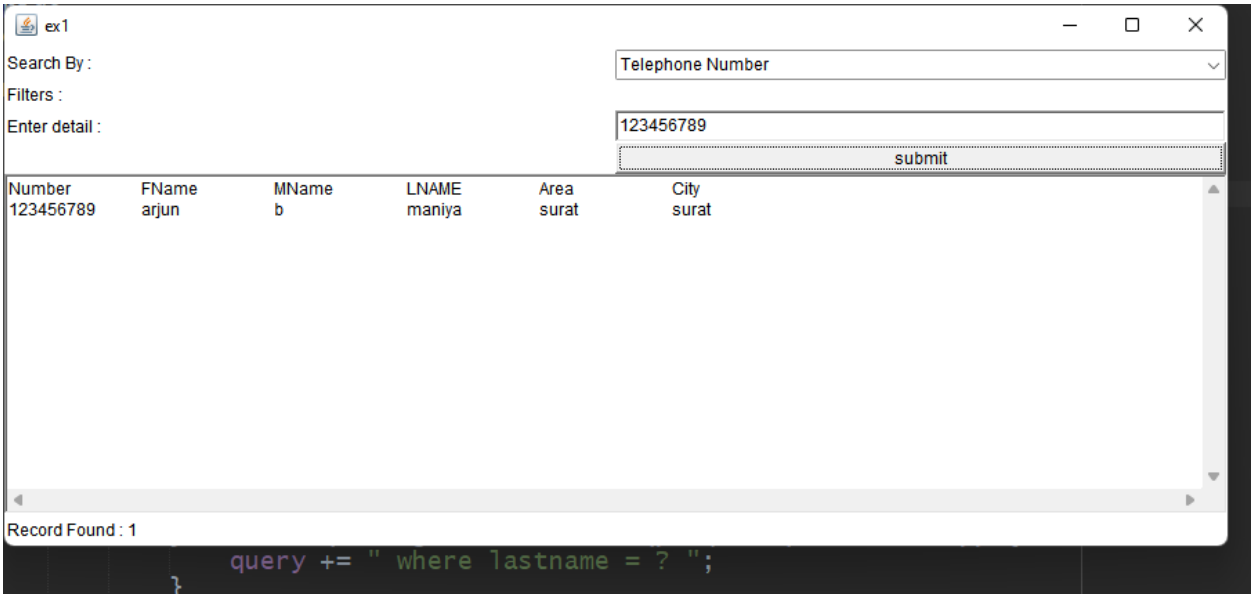
```
    @Override
    public void actionPerformed(ActionEvent arg0) {
        ta1.setText("refresh");
        query = new String("select * from ajt.telephone ");
        int len = 0;
        len = tf1.getText().toString().trim().length();

        try {
            if
(searchList.getSelectedItem().equals("Telephone Number") &&
len > 0) {
                query += " where number = ? ";
                pstate = conn.prepareStatement(query);
                pstate.setInt(1,
Integer.parseInt(tf1.getText().toString().trim()));
//
System.out.println(tf1.getText().toString().trim());
            } else if
(searchList.getSelectedItem().equals("Name") && len > 0) {
                if (filt.getSelectedItem().equals("First
Name")) {
                    query += " where firstname = ?";
                } else if
(filt.getSelectedItem().equals("Middle Name")) {
                    query += " where middlename = ? ";
                } else if
(filt.getSelectedItem().equals("Last Name")) {
                    query += " where lastname = ? ";
                }
                System.out.println(query);
                pstate = conn.prepareStatement(query);
                pstate.setString(1,
tf1.getText().toString().trim());
```

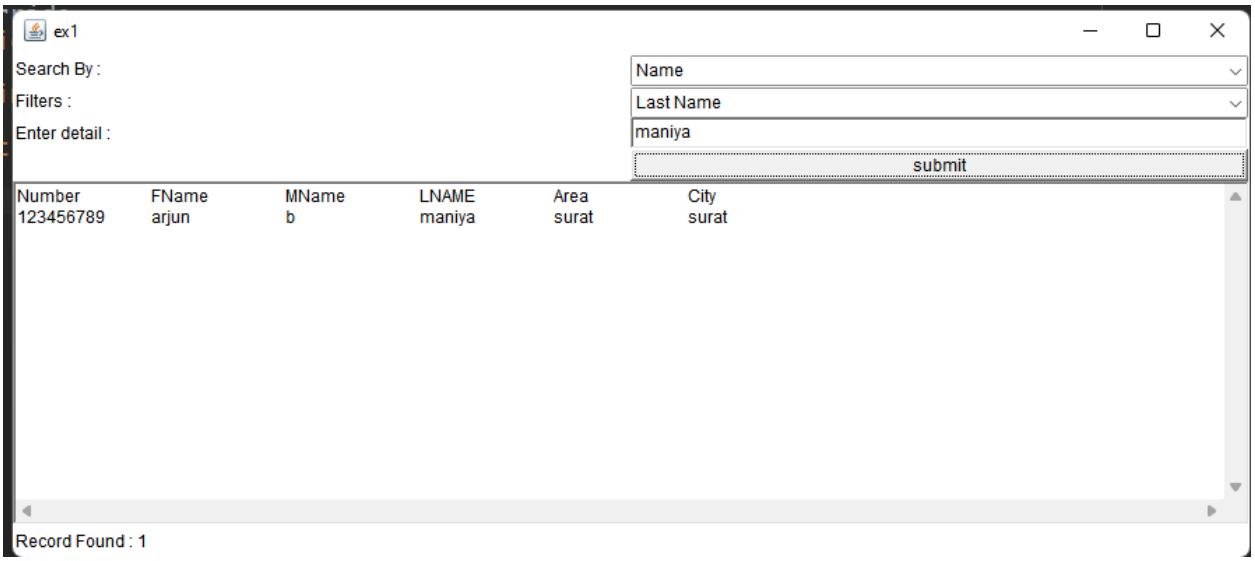
```
            } else if
(searchList.getSelectedItem().equals("Address") && len > 0)
{
                if (filt.getSelectedItem().equals("Area"));
                {
                    query += " where area = ?";
                }
                if (filt.getSelectedItem().equals("City"))
{
                    query += " where city = ?";
                }

                pstate = conn.prepareStatement(query);
                pstate.setString(1,
tf1.getText().toString().trim());

            } else {
                pstate = conn.prepareStatement(query);
            }

            try {
                System.out.println(query);
                rs = pstate.executeQuery();
            } catch (Exception e) {
                e.printStackTrace();
                System.out.println("Text Null ");
                ta1.setText("No records Found");
                status.setText("Record Found : 0");
            }
            if (rs != null) {
                //TODO : modify
```

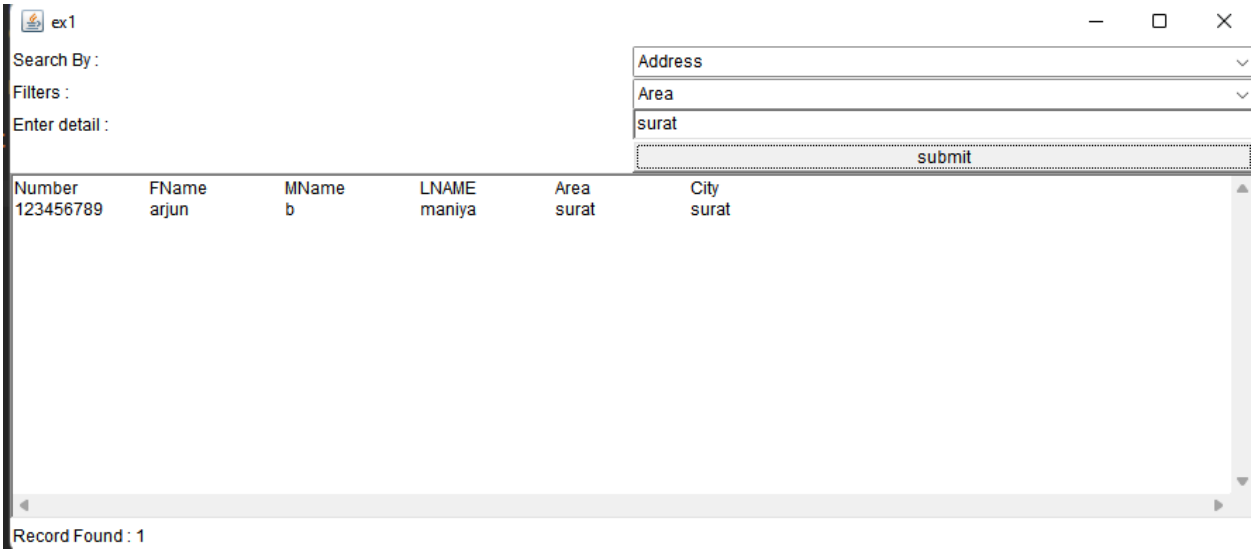
```
ta1.setText("Number\t\tFName\t\tMName\t\tLNAME\t\tArea\t\tC
ity\n");

                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\t");
                    ta1.append("" + rs.getString(5) +
"\t\t");
                    ta1.append("" + rs.getString(6) +
"\t\t");
                    count++;
                }
                status.setText("Record Found : " + count);

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

}
```

Input/Output:






ex1
Search By :
Name
Filters :
Last Name
Enter detail :
maniya
submit
Number FName MNname LNAME Area City
123456789 arjun b maniya surat surat
Record Found : 1


ex1
Search By :
Address
Filters :
Area
Enter detail :
surat
submit
Number FName MNname LNAME Area City
123456789 arjun b maniya surat surat
Record Found : 1

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

Experiment – 02

Submitted By: 19ituos121
Roll No.:IT069
Name: Arjun Maniya

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:
// Source code

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

```
package exp2;

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

/**
 *
 * @author admin
 */
public class Ex1 {

    public static void main(String[] args) {
        try {
           String dbDriver = "org.postgresql.Driver";
            String dbURL =
"jdbc:postgresql://localhost:5432/postgres";
            // Database name to access

            String dbUsername = "postgres";
            String dbPassword = "shree9592";

            Class.forName(dbDriver);
            Connection conn =
DriverManager.getConnection(dbURL, dbUsername, dbPassword);
            new Gui("ex2", conn);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

```
}
```

Gui.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 exp2;

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.*;

/**
 *
 * @author admin
 */
public class Gui extends Frame implements ItemListener,
ActionListener {

    private Choice option = new Choice();
    private Choice searchList = new Choice();
```

```
    private Choice filt = new Choice();
    private TextField tf1 = new TextField(20);
    private TextField number = new TextField(20);
    private TextField firstName = new TextField(20);
    private TextField middleName = new TextField(20);
    private TextField lastName = new TextField(20);
    private TextField area = new TextField(20);
    private TextField city = new TextField(20);
    private TextArea ta1 = new TextArea(10, 100);
    Button submit = new Button("submit");
    String query = "";
    String insertQuery = "";
    PreparedStatement pstate = null;
    Connection conn = null;
    ResultSet rs = null;
    Label status = new Label("Result Found : 0 ");
    int flag=0;

    Gui(String name, Connection conn) {
        super(name);
        setVisible(true);
        setSize(1000, 1000);
        setLayout(new BorderLayout());
        this.conn = conn;
        Panel p = new Panel();

        p.setLayout(new GridLayout(6, 2));
        option.addItemListener(this);
        option.add("select option");
        option.add("Add contact");
        option.add("search");
        p.add(new Label("Feature : "));
        p.add(option);
```

```
        searchList.addItemListener(this);
        searchList.add("Telephone Number");
        searchList.add("Name");
        searchList.add("Address");
        p.add(new Label("Search By : "));
        p.add(searchList);
        p.add(new Label("Filters : "));
        p.add(filt);
        filt.setVisible(false);
        p.add(new Label("Enter detail :"));
        p.add(tf1);

        p.add(new Label("Enter Number : "));
        p.add(number);
        p.add(new Label("Enter First name : "));
        p.add(firstName);
        p.add(new Label("Enter Middle Name : "));
        p.add(middleName);
        p.add(new Label("Enter Last name : "));
        p.add(lastName);
        p.add(new Label("Enter area : "));
        p.add(area);
        p.add(new Label("Enter city : "));
        p.add(city);
        number.setVisible(false);
        firstName.setVisible(false);
        middleName.setVisible(false);
        lastName.setVisible(false);
        city.setVisible(false);
        area.setVisible(false);

        p.add(new Label(""));
        p.add(submit);
```

```
        add("North", p);
        add("Center", ta1);
        add("South", status);
        addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });

        submit.addActionListener(this);
        searchList.addItemListener(this);

    }

    @Override
    public void itemStateChanged(ItemEvent arg0) {
        String arg = arg0.getItem().toString();

        if (arg.equals("search") || flag == 1) {
            flag = 1;
            tf1.setVisible(true);
            searchList.setVisible(true);
            number.setVisible(false);
            firstName.setVisible(false);
            middleName.setVisible(false);
            lastName.setVisible(false);
            city.setVisible(false);
            area.setVisible(false);
            if (arg0.equals("Telephone Number")) {
                filt.setVisible(false);
            } else if (arg0.equals("Name")) {
                filt.removeAll();
```

```
//                System.out.println("run name");
                filt.setVisible(true);
                filt.add("First Name");
                filt.add("Middle Name");
                filt.add("Last Name");
            } else if (arg0.equals("Address")) {
                filt.removeAll();
                filt.add("Area");
                filt.add("City");
                filt.setVisible(true);
            }
        } else if(arg.equals("Add contact") || flag == 2){
            tf1.setVisible(false);
            searchList.setVisible(false);
            number.setVisible(true);
            firstName.setVisible(true);
            middleName.setVisible(true);
            lastName.setVisible(true);
            city.setVisible(true);
            area.setVisible(true);
            status.setVisible(false);
            flag = 2;
        }
    }

    @Override
    public void actionPerformed(ActionEvent arg0) {

        int len = 0;

        try {
            ta1.setText("refresh");
```

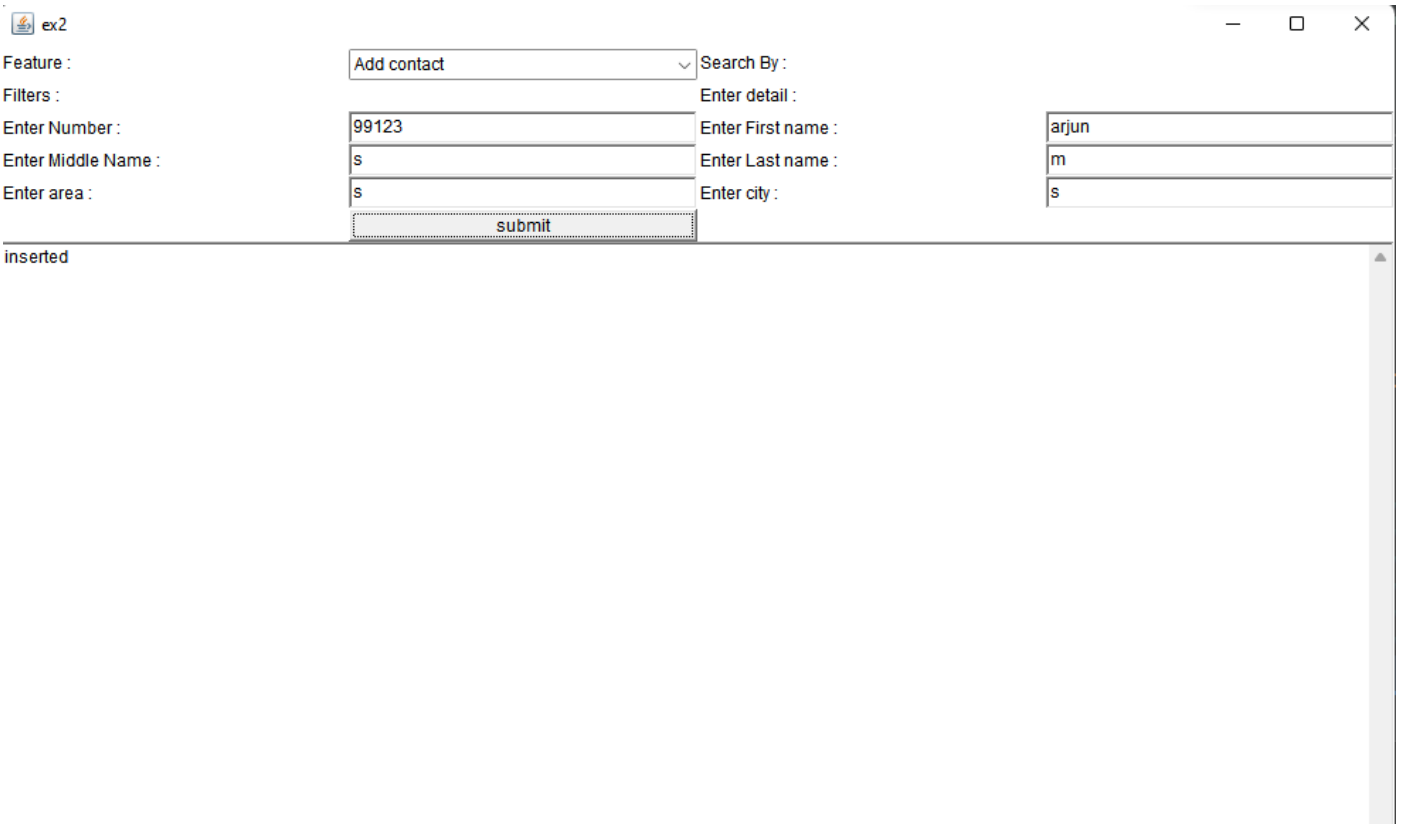
```
            query = new String("select * from ajt.telephone
");
            len = tf1.getText().toString().trim().length();
            if (option.getSelectedItem().equals("search"))
{
                if
(searchList.getSelectedItem().equals("Telephone Number") &&
len > 0) {
                    query += " where number = ? ";
                    pstate = conn.prepareStatement(query);
                    pstate.setInt(1,
Integer.parseInt(tf1.getText().toString().trim()));
//
System.out.println(tf1.getText().toString().trim());
                } else if
(searchList.getSelectedItem().equals("Name") && len > 0) {
                    if
(filt.getSelectedItem().equals("First Name")) {
                        query += " where firstname = ?";
                    } else if
(filt.getSelectedItem().equals("Middle Name")) {
                        query += " where middlename = ? ";
                    } else if
(filt.getSelectedItem().equals("Last Name")) {
                        query += " where lastname = ? ";
                    }
                    System.out.println(query);
                    pstate = conn.prepareStatement(query);
                    pstate.setString(1,
tf1.getText().toString().trim());
                } else if
(searchList.getSelectedItem().equals("Address") && len > 0)
{
```

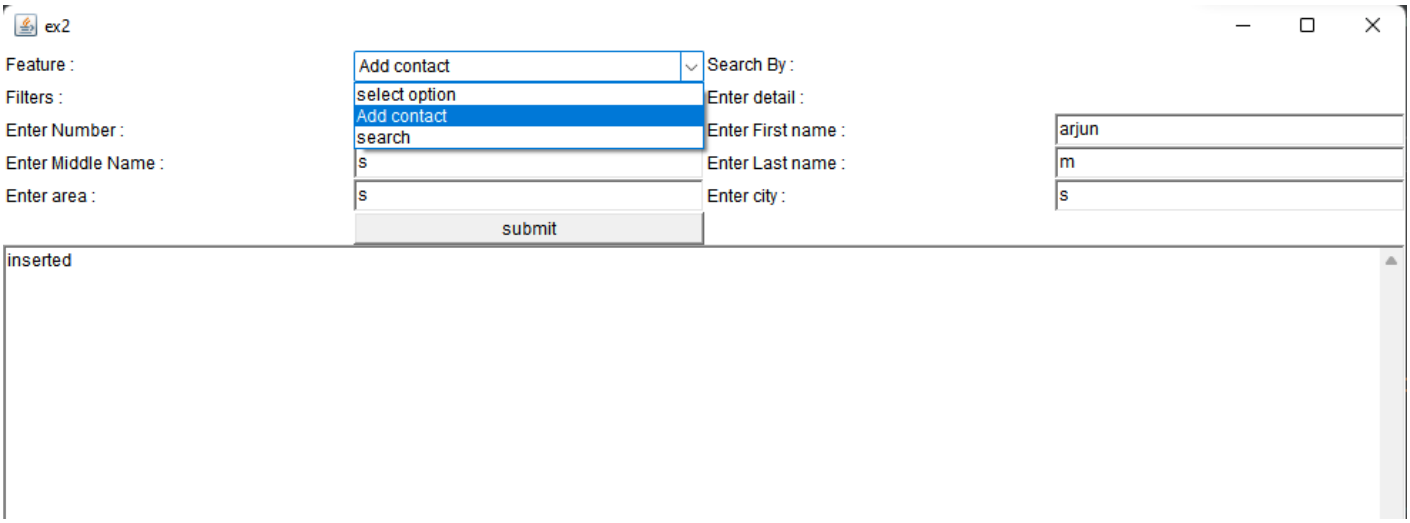
```
                    if
(filt.getSelectedItem().equals("Area"));
                    {
                        query += " where area = ?";
                    }
                    if
(filt.getSelectedItem().equals("City")) {
                        query += " where city = ?";
                    }

                    pstate = conn.prepareStatement(query);
                    pstate.setString(1,
tf1.getText().toString().trim());

                } else {
                    pstate = conn.prepareStatement(query);
                }

                try {
                    System.out.println(query);
                    rs = pstate.executeQuery();
                } catch (Exception e) {
                    e.printStackTrace();
                    System.out.println("Text Null ");
                    ta1.setText("No records Found");
                    status.setText("Record Found : 0");
                }
                if (rs != null) {
                    //TODO : modify

ta1.setText("Number\t\tFName\t\tMName\t\tLNAME\t\tArea\t\tC
ity\n");
```

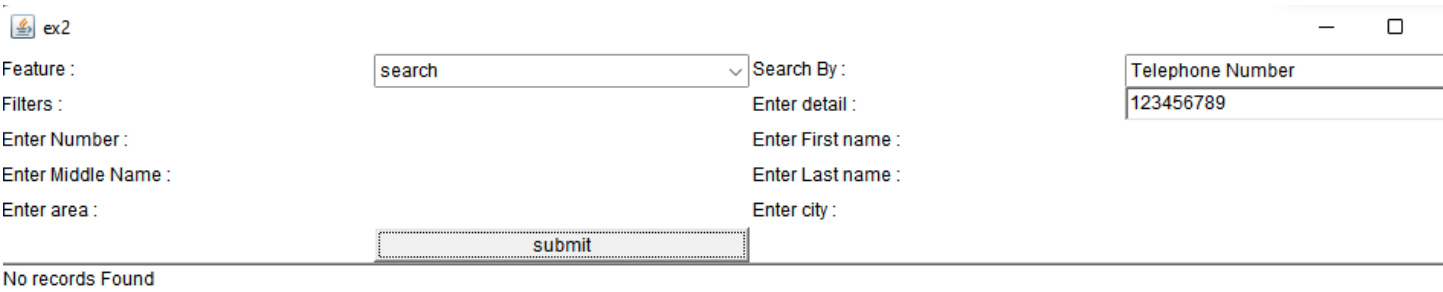
```
                    int count = 0;

                    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\t");
                        ta1.append("" + rs.getString(5) +
"\t\t");
                        ta1.append("" + rs.getString(6) +
"\t\t");
                        count++;
                    }
                    status.setText("Record Found : " +
count);

                }
            } else {
                insertQuery = new String("insert into
ajt.telephone values (?,?,?,?,?,?)");
                int number1 =
Integer.parseInt(number.getText().toString().trim());
                pstate =
conn.prepareStatement(insertQuery);
                pstate.setInt(1,number1 );
                pstate.setString(2,
firstName.getText().toString().trim());
                pstate.setString(3,
middleName.getText().toString().trim());
```

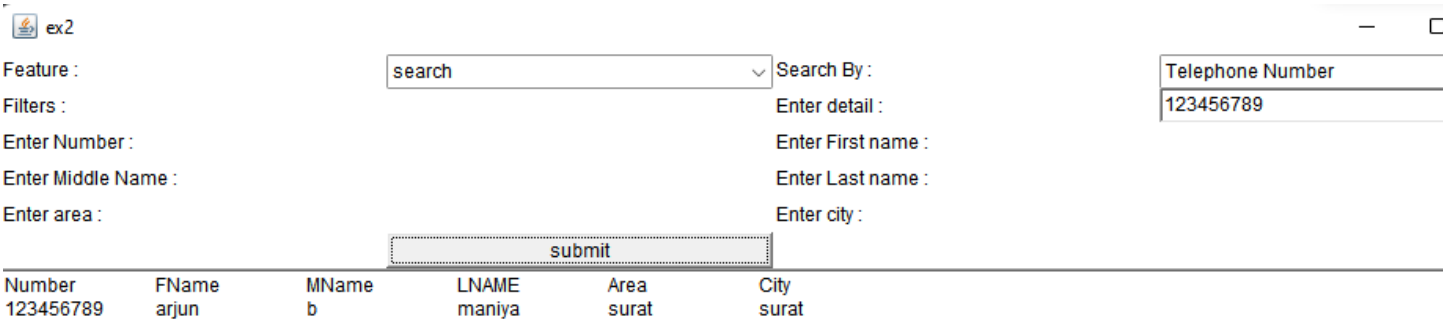
```
                pstate.setString(4,
lastName.getText().toString().trim());
                pstate.setString(5,
area.getText().toString().trim());
                pstate.setString(6,
city.getText().toString().trim());
                pstate.executeUpdate();
                ta1.setText("inserted");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

    }

}
```

Input/Output:


ex2
Feature :
Add contact
Search By :
Filters :
Enter detail :
Enter Number :
99123
Enter First name :
arjun
Enter Middle Name :
s
Enter Last name :
m
Enter area :
s
Enter city :
s
submit
inserted


ex2
Feature :
Add contact
Search By :
Filters :
select option
Add contact
search
Enter detail :
Enter Number :
Enter First name :
arjun
Enter Middle Name :
s
Enter Last name :
m
Enter area :
s
Enter city :
s
submit
inserted


ex2
Feature :
search
Search By :
Telephone Number
Filters :
Enter detail :
123456789
Enter Number :
Enter First name :
Enter Middle Name :
Enter Last name :
Enter area :
Enter city :
submit
No records Found


ex2
Feature :
search
Search By :
Telephone Number
Filters :
Enter detail :
123456789
Enter Number :
Enter First name :
Enter Middle Name :
Enter Last name :
Enter area :
Enter city :
submit
Number FName MName LNAME Area City
123456789 arjun b maniya surat surat

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

Experiment – 03

Submitted By: 19ituos121
Roll No.: IT069
Name: Arjun Maniya

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:
// Source code
INDEX>HTML :

```
<!DOCTYPE html>
<!--
Click
nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
Click
nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Html.h
tml to edit this template
-->
<html>

<head>
    <meta charset="UTF-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
<!--    <title>Welcome to Maniya</title>-->
    <link rel="stylesheet" type="text/css"
href="style.css">
</head>

<body>
    <div class="SignInContaier">
        <div class="column">
            <div class="header">
                <!--<img src="logo.png" alt="Logo"
title="Logo">-->
                <h3>Sign Up</h3>
                <!--<span>continue to maniya</span>-->
            </div>
            <form action="Register" method="post">

                <input type="text" name="userId"
id="userId" placeholder="User ID." required>
                <input type="text" name="firstName"
id="firstName" placeholder="First Name" required>

                <input type="text" name="lastName"
id="lastName" placeholder="Last Name" required>
                <input type="text" name="username"
id="username" placeholder="Username"  required>

                <input type="email" name="email" id="email"
placeholder="Email" required>
                <input type="email" name="email2"
id="email2" placeholder="Confirm Email" required>
```

```
                <input type="password" name="password"
id="password" placeholder="Password" required>
                <input type="password" name="password2"
id="password2" placeholder="Confirm Password" required>
                <input type="text" name="age" id="age"
placeholder="Age" required>
                <input type="text" name="rollNo"
id="rollNo" placeholder="Roll No." required>
                <input type="text" name="sem" id="sem"
placeholder="Semester" required>
                <input type="text" name="contact"
id="contact" placeholder="Contact Num." required>
                <input type="text" name="addr" id="addr"
placeholder="138 , Yamuna Darshan" required>


                <input type="submit" value="SUBMIT"
name="submitButton">
            </form>
        </div>
    </div>
</body>

</html>
```

Register.java :

```
/*
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
```

```
 * Click
nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servle
t.java to edit this template
 */

import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

/**
 *
 * @author arjun
 */
public class Register 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 {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use
following sample code. */
            String dbDriver = "org.postgresql.Driver";
            String dbURL =
"jdbc:postgresql://localhost:5432/postgres";
            // Database name to access

            String dbUsername = "postgres";
            String dbPassword = "shree9592";

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

            PreparedStatement ps =
con.prepareStatement("select * from ajt.students where
userid=?");
            ps.setInt(1,
Integer.parseInt(request.getParameter("userId")));
            ResultSet rs = ps.executeQuery();
            System.out.println("Result set");

            if (rs.next()) {
//
System.out.println("NewServlet.processRequest()");
                out.println("<h2>Data Found!</h2>");
            } else {
```

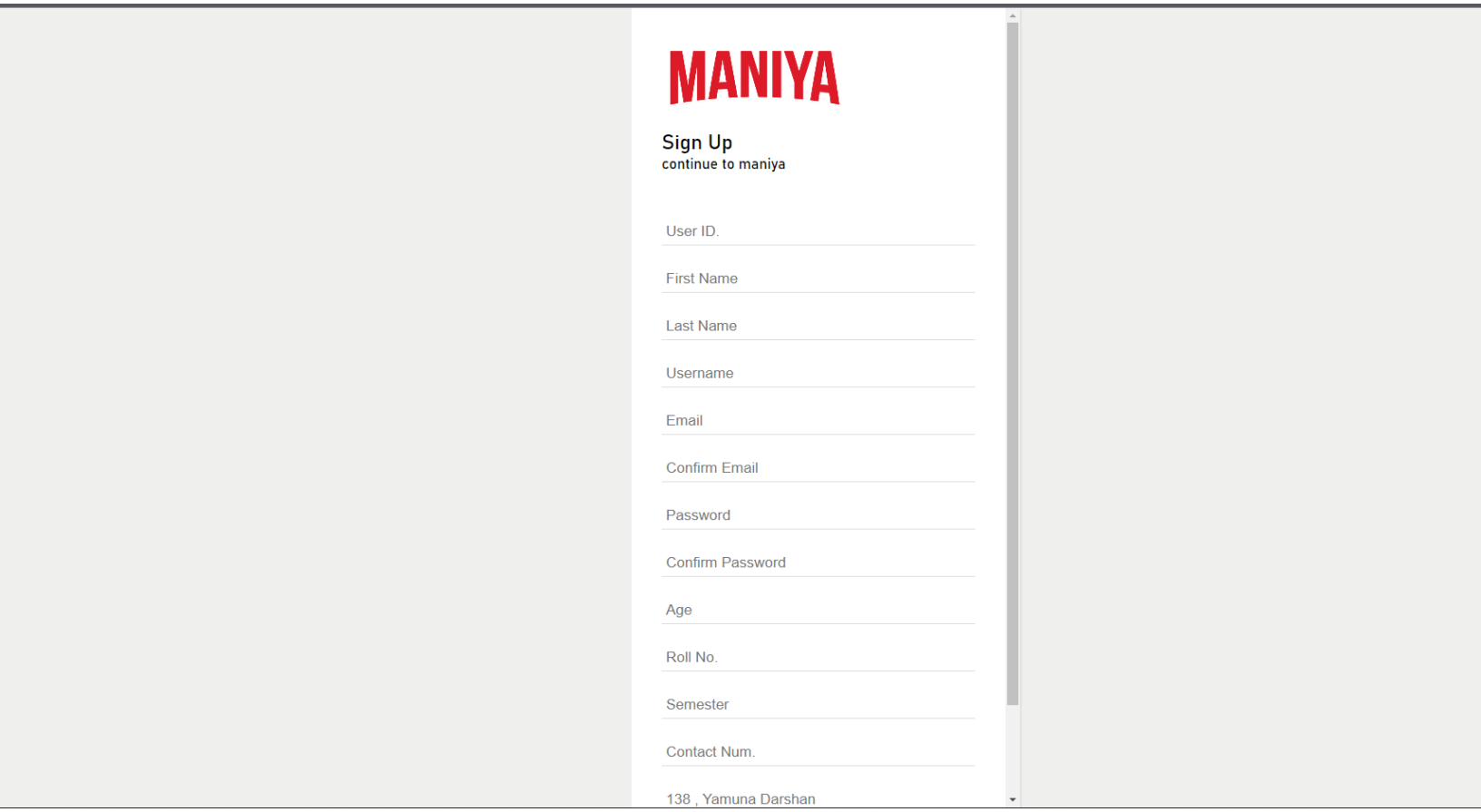
```
                String pass =
request.getParameter("password");
                String email =
request.getParameter("email");

                if
(pass.equals(request.getParameter("password2"))) {

                    if
(email.equals(request.getParameter("email2"))) {

                        PreparedStatement st =
con.prepareStatement("insert into ajt.students
values(?,?,?,?,?,?,?,?,?,?) ");
                        st.setInt(1,
Integer.parseInt(request.getParameter("userId")));
                        st.setString(2,
request.getParameter("username"));
                        st.setString(3, pass);
                        st.setString(4,
request.getParameter("firstName") + " " +
request.getParameter("lastName"));
                        st.setString(5,
request.getParameter("addr"));
                        st.setInt(6,
Integer.parseInt(request.getParameter("rollNo")));
                        st.setInt(7,
Integer.parseInt(request.getParameter("sem")));
                        st.setString(8, email);
                        st.setLong(9,
Long.parseLong(request.getParameter("contact")));
                        st.setInt(10,
Integer.parseInt(request.getParameter("age")));
```

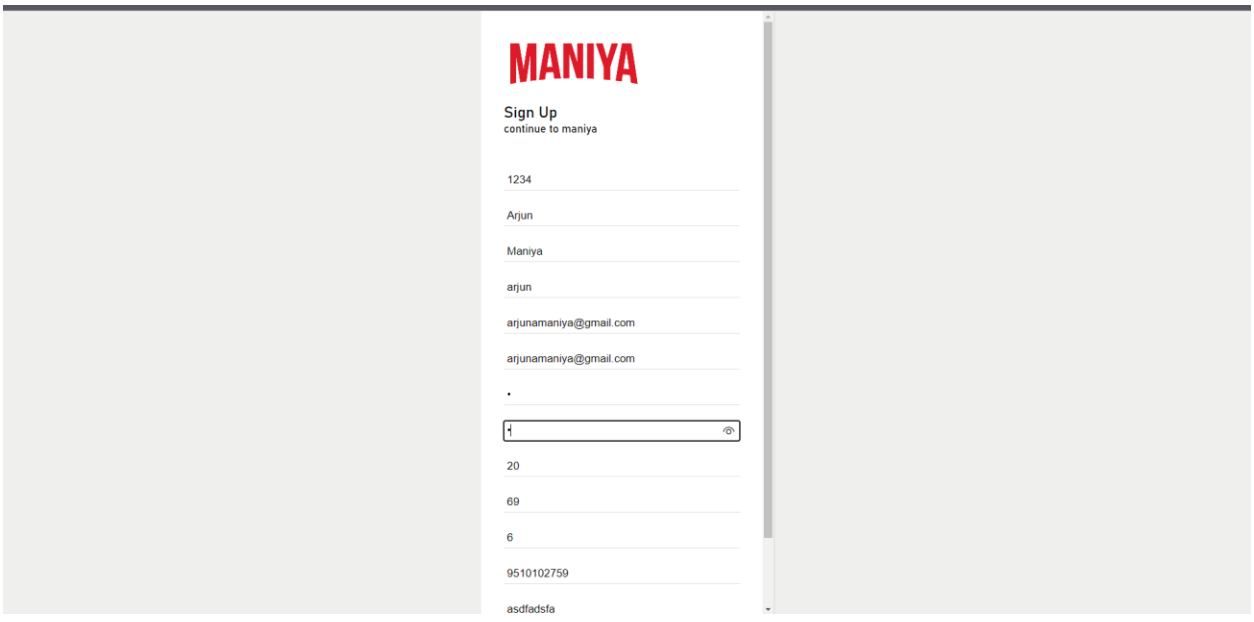
```
                        st.executeUpdate();
                        st.close();
                        con.close();

                        out.println("Successfully
Inserted");
                        System.out.println("inserted");
                    } else {
                        out.print("Confirm email not
match");
                    }

                } else {
                    out.print("Confirm Password Does not
Match");
                }
            }
        } catch (Exception e) {
            e.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:

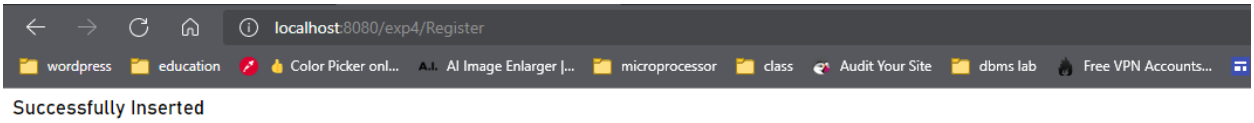
Input:


MANIYA
Sign Up
continue to maniya
User ID.
First Name
Last Name
Username
Email
Confirm Email
Password
Confirm Password
Age
Roll No.
Semester
Contact Num.
138 , Yamuna Darshan

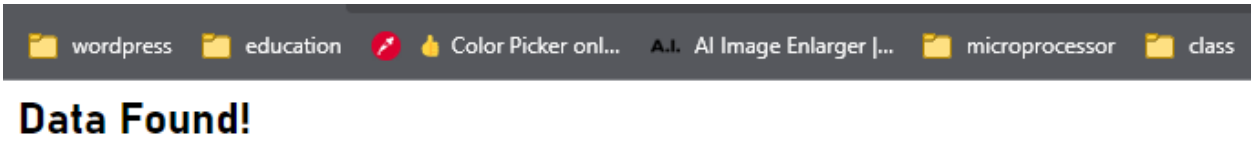

MANIYA
Sign Up
continue to maniya
1234
Arjun
Maniya
arjun
arjunamaniya@gmail.com
arjunamaniya@gmail.com
20
69
6
9510102759
asdfadsfa

Output:

If user is not registered in our database then successfully register :



If user already in data base then :



Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT69
B.Tech. IT, Sem: VI

Experiment – 04

Submitted By: 19ituos121
Roll No.: IT069
Name: Arjun Maniya

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:
// Source code

Index.html :

```
<!DOCTYPE html>
<!--
Click
nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
Click
nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Html.h
tml to edit this template
-->
<html>
    <head>
        <title>login form</title>
        <link rel="stylesheet" type="text/css"
href="style.css">
    </head>
```

```
    <body>
        <div class="SignInContaier">
            <div class="column">
                <div class="header">
                    <h3>Sign In</h3>
                    <span>Continue to Result Portal </span>
                </div>
                <form action="Login" method="POST">
                    <input type="text" name="userid"
id="userid" placeholder="user id" >
                    <input type="password" name="password"
id="password" placeholder="Password" >
                    <input type="submit" value="SUBMIT"
name="submitButton">
                    <!--<script>alert("entered");</script>-
->
                </form>
                <!--<span> have account ? Make one <a
href="" class="signInMessage">here</a></span>-->
            </div>
        </div>
    </body>
</html>
```

Login.java :

```
/*
```

```
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click
nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servle
t.java to edit this template
 */


import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.http.HttpSession;

/**
 *
 * @author admin
 */
public class Login 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 {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use
following sample code. */
           int userid =
Integer.parseInt(request.getParameter("userid"));
            System.out.println("user id " + userid );
           String password =
request.getParameter("password");
            System.out.println("servlet called");
           if(LoginCheck.validate(userid, password))
           {
               HttpSession session = request.getSession();
               session.setAttribute("userid",userid);
               RequestDispatcher rd =
request.getRequestDispatcher("welcome");
               rd.forward(request, response);
           }
           else{
               out.print("username or password are wrong
check again");
               RequestDispatcher rd =
request.getRequestDispatcher("index.html");
               rd.forward(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 {
        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>

}
```

LoginCheck.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
 */

import java.sql.*;
/**
 *
 * @author admin
```

```
 */
public class LoginCheck {
    public static boolean validate(int userid ,String
password)
    {
        System.out.println("user id = " + userid);
        try {
            Class.forName("org.postgresql.Driver");
            java.sql.Connection conn =
DriverManager.getConnection("jdbc:postgresql://localhost:54
32/postgres", "postgres", "shree9592");
            System.out.println("connection done");

            PreparedStatement pstmt =
conn.prepareStatement("select * from ajt.students where
userid = ? and password = ?");
            pstmt.setInt(1,(userid));
            pstmt.setString(2,password);

            ResultSet rs = pstmt.executeQuery();
            if(rs.next())
            {
                return true;
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
        return false;
    }
}
```

Welcome.java

```
/*
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click
nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servle
t.java to edit this template
 */


import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

/**
 *
 * @author admin
 */
public class Welcome 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 {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use
following sample code. */
            int username =
Integer.parseInt(request.getParameter("userid"));
            out.print("welcome "+username + "<br>");

            out.print("<br> Check your marks <a
href='Marks'>here!</a>");

        }
    }

    // <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>
```

```
}
```

Marks.java

```
/*
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click
nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servle
t.java to edit this template
 */

import jakarta.servlet.http.HttpSession;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author arjun
 */
public class Marks 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
     * @throws java.sql.SQLException
     */
    protected void processRequest(HttpServletRequest
request, HttpServletResponse response)
            throws ServletException, IOException,
SQLException, ClassNotFoundException {
        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);
            int username =
(int)session.getAttribute("userid");
            Class.forName("org.postgresql.Driver");
            java.sql.Connection conn =
DriverManager.getConnection("jdbc:postgresql://localhost:54
32/postgres", "postgres", "shree9592");
            System.out.println("connection done");
            PreparedStatement pstmt =
conn.prepareStatement("select * from ajt.marks where
userid=? ");
```

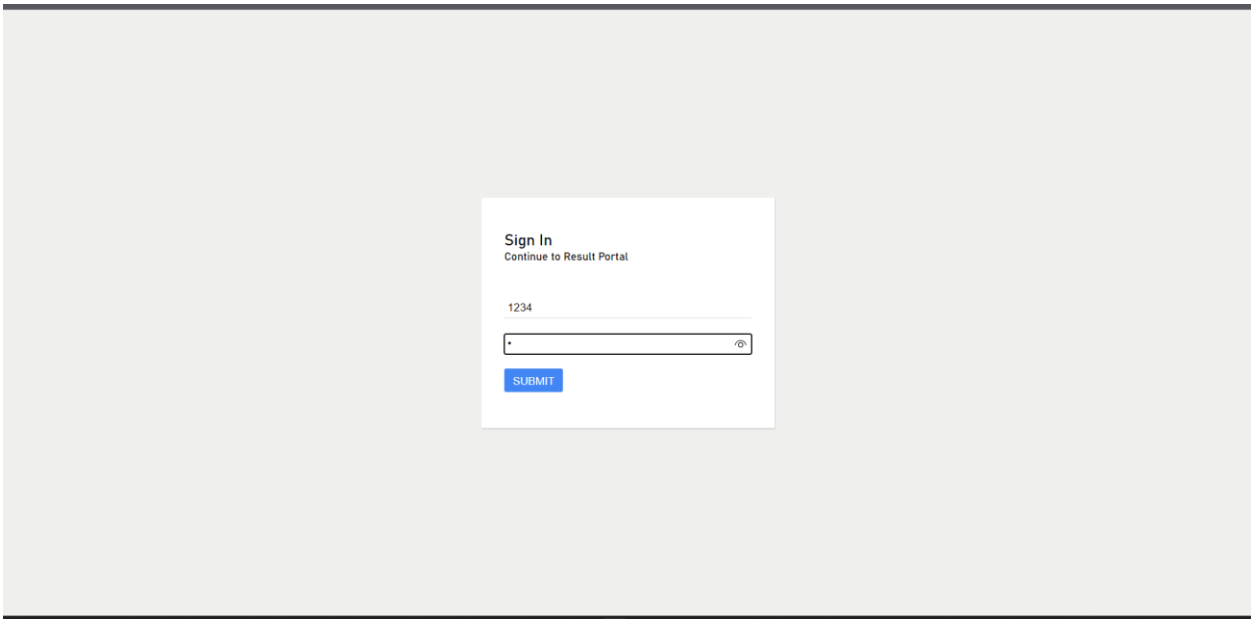
```
            pstmt.setInt(1,username);
            out.print("hi ");
            ResultSet rs = pstmt.executeQuery();
            String start = "<br><h2>";
            String end = "</h2>";
            while(rs.next())
            {
                start += "<br> paper 1  marks " +
rs.getInt("paper1");
                start += "<br> paper 2  marks " +
rs.getInt("paper2");
                start += "<br> paper 3  marks " +
rs.getInt("paper3");
            }
            out.print(start+end);

        }
    }

    // <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 {
        try {
            processRequest(request, response);
        } catch (SQLException ex) {

Logger.getLogger(Marks.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (ClassNotFoundException ex) {

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

    /**
     * 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 {
            processRequest(request, response);
        } catch (SQLException ex) {
```

```
Logger.getLogger(Marks.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (ClassNotFoundException ex) {

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

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

}
```

Input/Output:

Input:



Output:

Welcome page here user can enter after successful login :

Showing student Marks :

---

hi

paper 1 marks 45
paper 2 marks 54
paper 3 marks 65

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT69
B.Tech. IT, Sem: VI

Experiment – 05

Submitted By: 19ituos121
Roll No.: IT069
Name: Arjun Maniya

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:
// Source code
Auth.java :

```
package filters;

import java.io.IOException;
import java.io.PrintStream;
import java.io.PrintWriter;
import java.io.StringWriter;
import jakarta.servlet.Filter;
import jakarta.servlet.FilterChain;
import jakarta.servlet.FilterConfig;
import jakarta.servlet.ServletException;
import jakarta.servlet.ServletRequest;
import jakarta.servlet.ServletResponse;
import jakarta.servlet.annotation.WebFilter;
```

```
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

/**
 *
 * @author Zainab
 */
@WebFilter(filterName = "Authenticate", urlPatterns =
{"/authenticate"})
public class Auth implements Filter {

    private static final boolean debug = true;
    // The filter configuration object we are associated
with. If
    // this value is null, this filter instance is not
currently
    // configured.
    private FilterConfig filterConfig = null;

    public Auth() {
    }

    private void doBeforeProcessing(ServletRequest request,
ServletResponse response)
            throws IOException, ServletException {
        if (debug) {
            log("Authenticate:DoBeforeProcessing");
        }
        // Write code here to process the request and/or
response before
        // the rest of the filter chain is invoked.
        // For example, a logging filter might log items on
the request object,
```

```
        // such as the parameters.
        /*
for (Enumeration en = request.getParameterNames();
en.hasMoreElements(); ) {
 String name = (String)en.nextElement();
 String values[] = request.getParameterValues(name);
 int n = values.length;
 StringBuffer buf = new StringBuffer();
 buf.append(name);
 buf.append("=");
 for(int i=0; i < n; i++) {
 buf.append(values[i]);
 if (i < n-1)
 buf.append(",");
 }
BTech-IT, Sem-VI, Term Work, Advanced Java Technology,
IT619
 log(buf.toString());
}
         */
    }

    private void doAfterProcessing(ServletRequest request,
ServletResponse response)
            throws IOException, ServletException {
        if (debug) {
            log("Authenticate:DoAfterProcessing");
        }
        // Write code here to process the request and/or
response after
        // the rest of the filter chain is invoked.
        // For example, a logging filter might log the
attributes on the
```

```
        // request object after the request has been
processed.
        /*
for (Enumeration en = request.getAttributeNames();
en.hasMoreElements(); ) {
 String name = (String)en.nextElement();
 Object value = request.getAttribute(name);
 log("attribute: " + name + "=" + value.toString());
}
         */
        // For example, a filter might append something to
the response.
        /*
PrintWriter respOut = new
PrintWriter(response.getWriter());
respOut.println("<P><B>This has been appended by an
intrusive filter.</B>");
         */
    }

    /**
     *
     * @param request The servlet request we are processing
     * @param response The servlet response we are creating
     * @param chain The filter chain we are processing
     *
     * @exception IOException if an input/output error
occurs
     * @exception ServletException if a servlet error
occurs
     */
```

```
    public void doFilter(ServletRequest request,
ServletResponse response, FilterChain chain) throws
            IOException, ServletException {
        PrintWriter out = response.getWriter();
        if (debug) {
            log("TestFilter:doFilter()");
        }
        doBeforeProcessing(request, response);
        HttpServletRequest req = (HttpServletRequest)
request;
        HttpServletResponse res = (HttpServletResponse)
response;
        String uname = req.getParameter("uname");
        String passwd = req.getParameter("passwd");
        if (uname.equals("arjun") &&
passwd.equals("arjun")) {
            out.println("\n Success..... Servlet Filter is
Forwarding Request");
            //chain.doFilter(request, response);
        } else {
            out.println("\n Failure..... Servlet Filter has
not Forwarded Request");
            res.sendRedirect("index.jsp");
        }
        Throwable problem = null;
        try {
            chain.doFilter(request, response);
        } catch (Throwable t) {
            out.print("Error");
        }
    }

    /**
```

```
     * Return the filter configuration object for this
filter.
     */
    public FilterConfig getFilterConfig() {
        return (this.filterConfig);
    }

    /**
     * Set the filter configuration object for this filter.
     *
     * @param filterConfig The filter configuration object
     */
    public void setFilterConfig(FilterConfig filterConfig)
{
        this.filterConfig = filterConfig;
    }

    /**
     * Destroy method for this filter
     */
    public void destroy() {
    }

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

    /**
```

```
     * Return a String representation of this object.
     */
    @Override
    public String toString() {
        if (filterConfig == null) {
            return ("Authenticate()");
        }
        StringBuffer sb = new
StringBuffer("Authenticate(");
        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<b
ody>\n"); //NOI18N
                // PENDING! Localize this for next official
release
                pw.print("<h1>The resource did not process
correctly</h1>\n<pre>\n");
                pw.print(stackTrace);
                pw.print("</pre></body>\n</html>");
//NOI18N
```

```
                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);
```

```
    }
}
```

log.java :

```
package filters;

import java.io.IOException;
import java.io.PrintStream;
import java.io.PrintWriter;
import java.io.StringWriter;
import java.util.Date;
import jakarta.servlet.Filter;
import jakarta.servlet.FilterChain;
import jakarta.servlet.FilterConfig;
import jakarta.servlet.ServletException;
import jakarta.servlet.ServletRequest;
import jakarta.servlet.ServletResponse;
import jakarta.servlet.http.HttpServletRequest;

public class Log implements Filter {

    private static final boolean debug = true;
    private FilterConfig filterConfig = null;

    public Log() {
    }

    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");
        }
    }

    /**
     *
     * @param request The servlet request we are processing
     * @param response The servlet response we are creating
     * @param chain The filter chain we are processing
     *
     * @exception IOException if an input/output error
occurs
     * @exception ServletException if a servlet error
occurs
     */
    public void doFilter(ServletRequest request,
ServletResponse response, FilterChain chain)
            throws IOException, ServletException {
        PrintWriter out = response.getWriter();
        if (debug) {
            log("TestFilter:doFilter()");
        }
        doBeforeProcessing(request, response);
        System.out.println("doFilter() method is called in
" + this.getClass().getName() + "\n");
```

```
        HttpServletRequest req = (HttpServletRequest)
request;
        String ipAddress = req.getRemoteAddr();
        out.println("IP Address is " + ipAddress + " Time
is " + new Date().toString() + "\n");
        Throwable problem = null;
        try {
            chain.doFilter(request, response);
        } catch (Throwable t) {
            System.out.println("ERORR in loginfilter");
        }
    }

    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<b
ody>\n"); //NOI18N
                // PENDING! Localize this for next official
release
                pw.print("<h1>The resource did not process
correctly</h1>\n<pre>\n");
                pw.print(stackTrace);
                pw.print("</pre></body>\n</html>");
//NOI18N
                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);
    }
}
```

Main.java :

```
package filters;

import java.io.IOException;
import java.io.PrintStream;
import java.io.PrintWriter;
import java.io.StringWriter;
import java.util.Date;
import jakarta.servlet.Filter;
import jakarta.servlet.FilterChain;
import jakarta.servlet.FilterConfig;
import jakarta.servlet.ServletException;
import jakarta.servlet.ServletRequest;
import jakarta.servlet.ServletResponse;
import jakarta.servlet.http.HttpServletRequest;

public class Log implements Filter {

    private static final boolean debug = true;
    private FilterConfig filterConfig = null;

    public Log() {
    }

    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");
        }
    }

    /**
     *
     * @param request The servlet request we are processing
     * @param response The servlet response we are creating
     * @param chain The filter chain we are processing
     *
     * @exception IOException if an input/output error
occurs
     * @exception ServletException if a servlet error
occurs
     */
    public void doFilter(ServletRequest request,
ServletResponse response, FilterChain chain)
            throws IOException, ServletException {
        PrintWriter out = response.getWriter();
        if (debug) {
            log("TestFilter:doFilter()");
        }
        doBeforeProcessing(request, response);
        System.out.println("doFilter() method is called in
" + this.getClass().getName() + "\n");
        HttpServletRequest req = (HttpServletRequest)
request;
```

```
        String ipAddress = req.getRemoteAddr();
        out.println("IP Address is " + ipAddress + " Time
is " + new Date().toString() + "\n");
        Throwable problem = null;
        try {
            chain.doFilter(request, response);
        } catch (Throwable t) {
            System.out.println("ERORR in loginfilter");
        }
    }

    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<b
ody>\n"); //NOI18N
                // PENDING! Localize this for next official
release
                pw.print("<h1>The resource did not process
correctly</h1>\n<pre>\n");
                pw.print(stackTrace);
                pw.print("</pre></body>\n</html>");
//NOI18N
                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);
    }
}
```

index.html :

```
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>login Page</title>
    </head>
    <body>
        <form action="Main" method="POST">
            User Name: <input type="text"
name="uname"/><br/><br/>
            Password: <input type="text"
name="passwd"/><br/><br/>
            <input type="submit" name="Submit"/>
        </form>
    </body>
</html>
```

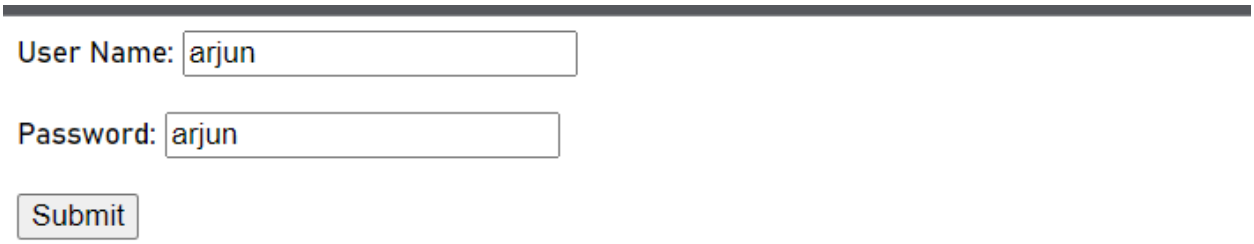
web.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="4.0"
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_4_0.xsd">
    <filter>
        <filter-name>Log</filter-name>
        <filter-class>filters.Log</filter-class>
    </filter>
    <filter>
```

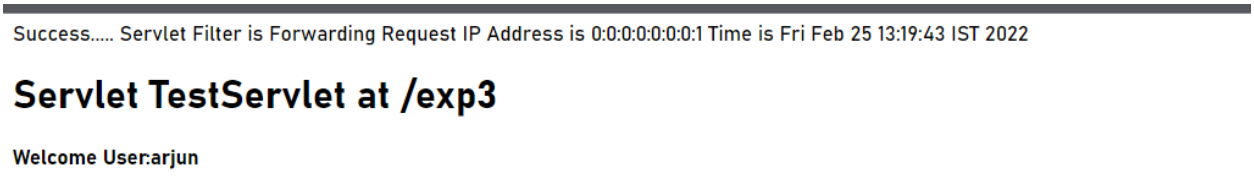
```
        <filter-name>Auth</filter-name>
        <filter-class>filters.Auth</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>Auth</filter-name>
        <servlet-name>Main</servlet-name>
    </filter-mapping>
    <filter-mapping>
        <filter-name>Log</filter-name>
        <servlet-name>Main</servlet-name>
    </filter-mapping>
    <servlet>
        <servlet-name>Main</servlet-name>
        <servlet-class>filters.Main</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>Main</servlet-name>
        <url-pattern>/Main</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
</web-app>
```

Input/Output:

Input:


User Name: arjun
Password: arjun
Submit

Output:


Success..... Servlet Filter is Forwarding Request IP Address is 0:0:0:0:0:0:0:1 Time is Fri Feb 25 13:19:43 IST 2022
Servlet TestServlet at /exp3
Welcome User:arjun

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT69
B.Tech. IT, Sem: VI

Experiment – 06

Submitted By:
Roll No.:
Name:

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:
// Source code
index.jsp :

```
<%--
    Document   : main
    Created on : Feb 2, 2022, 1:30:32 PM
    Author     : admin
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <link rel="stylesheet" href="style.css"/>
```

```
        <title>User Register</title>
    </head>
    <body>
       <div class="SignInContaier">
            <div class="column">
                <div class="header">
                    <h3>Sign In</h3>
                    <span>Continue to Login</span>
                </div>
                <form action="Login" method="POST" >
                    <input type="text" name="username"
id="username" placeholder="Username" >
                    <input type="password" name="password"
id="password" placeholder="Password" >
                    <input type="submit" value="SUBMIT"
name="submitButton">
                    <!--<script>alert("entered");</script>-
->
                </form>
                <!--<span> have account ? Make one <a
href="" class="signInMessage">here</a></span>-->
            </div>
        </div>
    </body>
</html>
```

PersonDetails.jsp :

```
<%--
    Document   : second
    Created on : Feb 2, 2022, 4:56:26 PM
    Author     : admin
--%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="ex6.PersonBean" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Hello
            <jsp:useBean id="st" class="ex6.PersonBean"
scope="session"></jsp:useBean>

        Name    : <jsp:getProperty property="name"
name="st"/><br>
        Age     : <jsp:getProperty property="age"
name="st"/><br>
        height     : <jsp:getProperty property="hight"
name="st"/><br>

        </h1>

    </body>
</html>
```

DBconn.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 ex6;

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

/**
 *
 * @author admin
 */
public class DBConn {

    static Connection conn = null;

    public static Connection getConnection() {
        try {
            if(conn == null)
            {
                Class.forName("org.postgresql.Driver");
                conn =
DriverManager.getConnection("jdbc:postgresql://localhost:54
32/postgres", "postgres", "shree9592");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return conn;
    }
}
```

Login.java :

```
/*
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click
nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servle
t.java to edit this template
 */
package ex6;

import jakarta.servlet.RequestDispatcher;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

/**
 *
 * @author admin
 */
public class Login 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 {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String username =
request.getParameter("username");
            String pswd = request.getParameter("password");
            String password = "";

            try {
                Connection conn = DBConn.getConnection();
                PreparedStatement pstmt =
conn.prepareStatement("select * from ajt.person where
name=?");
                pstmt.setString(1, username);
                ResultSet rs = pstmt.executeQuery();

                if (rs.next()) {
                    password = rs.getString("password");
                    if (pswd.equals(password)) {
                        PersonBean st = new PersonBean();
                        st.setName(rs.getString("name"));
```

```
                        st.setAge(rs.getInt("age"));
                        st.setHight(rs.getInt("height"));

                        HttpSession hs =
request.getSession();
                        hs.setAttribute("st", st);

                        RequestDispatcher rd =
request.getRequestDispatcher("PersonDetail.jsp");
                        rd.forward(request, response);
                    } else {
                        out.println("<h1 align='center'
style='color:red;'>Incorrect Password!</h1>");
                    }
                } else {
                    out.println("<h1 align='center'
style='color:red;'>User not Found! Try again.</h1>");
                }
            } catch (SQLException e) {
                e.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";
    }
}
```

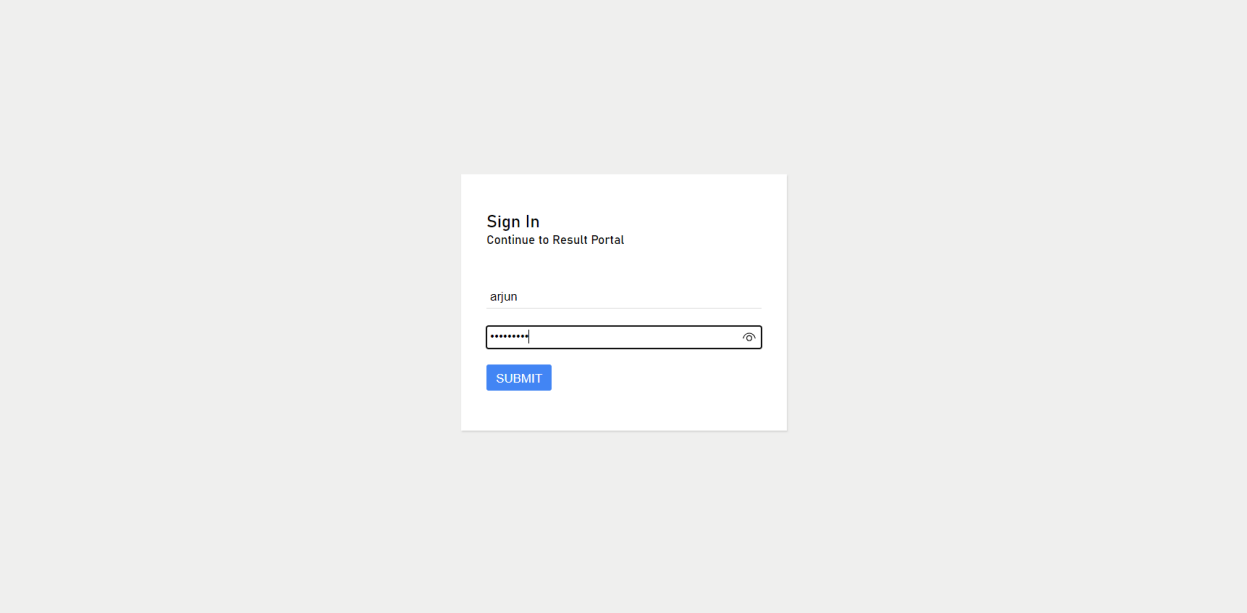
PersonBean.java

```
/*
 * 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 ex6;

/**
 *
 * @author admin
 */
public class PersonBean implements java.io.Serializable{
```

```
    private String name;
    private int hight,age;

    public PersonBean() {
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getHight() {
        return hight;
    }

    public void setHight(int hight) {
        this.hight = hight;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }


}
```

Input/Output:

Input:



Output:

```
Hello Name : arjun
Age : 19
height : 6
```

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT69
B.Tech. IT, Sem: VI

Experiment – 07

Submitted By: 19ituos121
Roll No.: IT069
Name: Arjun Maniya

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:
// Source code
index.jsp :

```
<!DOCTYPE html>
<!--
Click
nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
Click
nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Html.h
tml to edit this template
-->
<html>
    <head>
        <title>login form</title>
```

```
        <link rel="stylesheet" type="text/css"
href="style.css">
    </head>
    <body>
        <div class="SignInContaier">
            <div class="column">
                <div class="header">
                    <h3>Sign In</h3>
                    <span>Continue to Result Portal </span>
                </div>
                <form action="valid.jsp" method="POST">
                    <input type="text" name="uname"
id="uname" placeholder="uname" >
                    <input type="password" name="password"
id="password" placeholder="Password" >
                    <input type="submit" value="SUBMIT"
name="submitButton">
                    <!--<script>alert("entered");</script>-
->
                </form>
                <!--<span> have account ? Make one <a
href="" class="signInMessage">here</a></span>-->
            </div>
        </div>
    </body>
</html>
```

validate.jsp :

```
<%--
    Document   : valid
    Created on : 23-Feb-2022, 8:04:08 AM
```

```
    Author     : arjun
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <%@ taglib uri="http://java.sun.com/jsp/jstl/core"
prefix="c"%>
        <%@ taglib uri="http://java.sun.com/jsp/jstl/sql"
prefix="sql"%>
        <link rel="stylesheet" type="tsxt/css"
href="style.css"/>
        <title>validation Page</title>
    </head>
    <body>
        <sql:setDataSource var="ds"
driver="org.postgresql.Driver"

url="jdbc:postgresql://localhost:5432/postgres"
                           user="postgres"
password="shree9592"/>
        <c:set var="username" value="${param.uname}"
scope="session"/>
        <c:set var="password" value="${param.password}"
scope="session"/>
        <sql:query dataSource="${ds}" var="result">
            SELECT * from ajt.students where username=? and
password=?
            <sql:param value="${username}"/>
            <sql:param value="${password}"/>
```

```
        </sql:query>
        <c:if test="${result.rowCount > 0}">
            <h1>Hello <c:out value="${username}"/>!</h1>
            <div>edit Detail</div>
            <c:forEach var="st" items="${result.rows}">
            <form action="edit.jsp" method="POST">
                <input type="text" name="uname" id="uname"
value="${username}" readonly>
                <input type="text" name="email"
value="${st.email}" placeholder="email">
                <input type="text" name="address"
value="${st.address}" placeholder="address" >
                <input type="text" name="contact"
value="${st.contact}" placeholder="contact number">
                <input type="password" name="password"
value="${st.password}" placeholder="password" >
                <input type="submit" value="SUBMIT"
name="submitButton">
                <!--<script>alert("entered");</script>-->
            </form>
            </c:forEach>


        </c:if>
    </body>
</html>
```

edit.jsp :

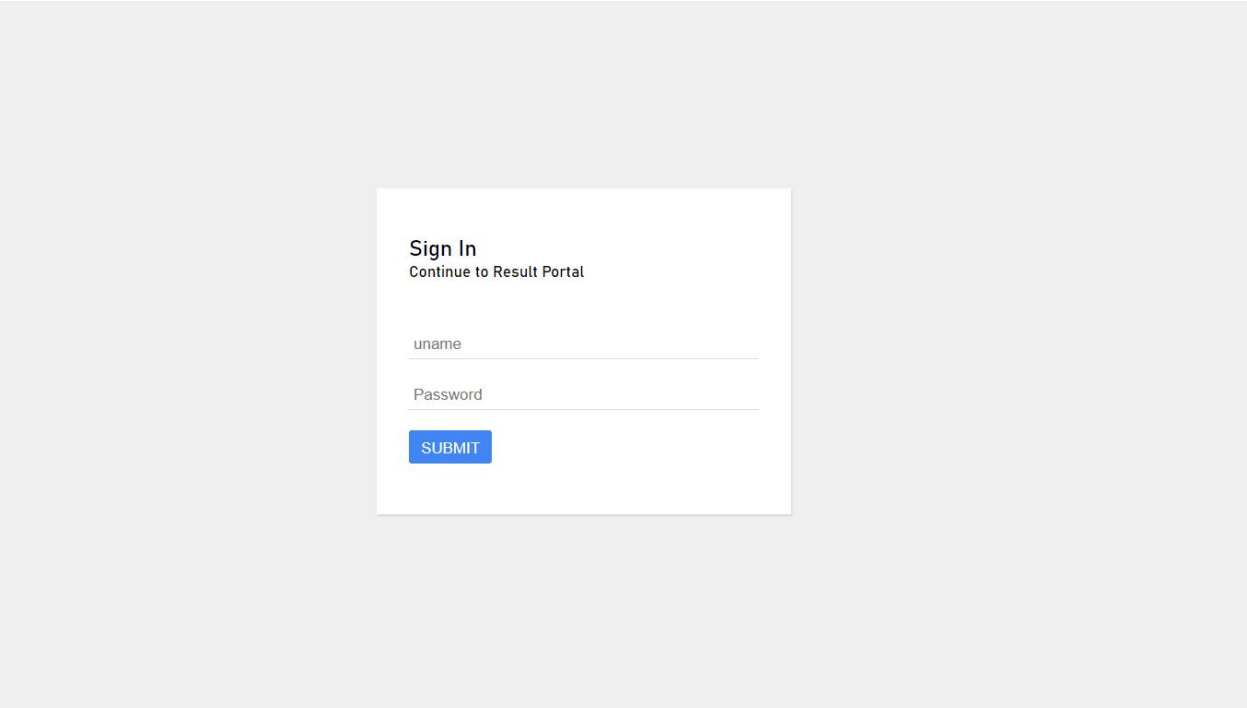
```
<%--
    Document   : edit
    Created on : 23-Feb-2022, 12:42:51 PM
```

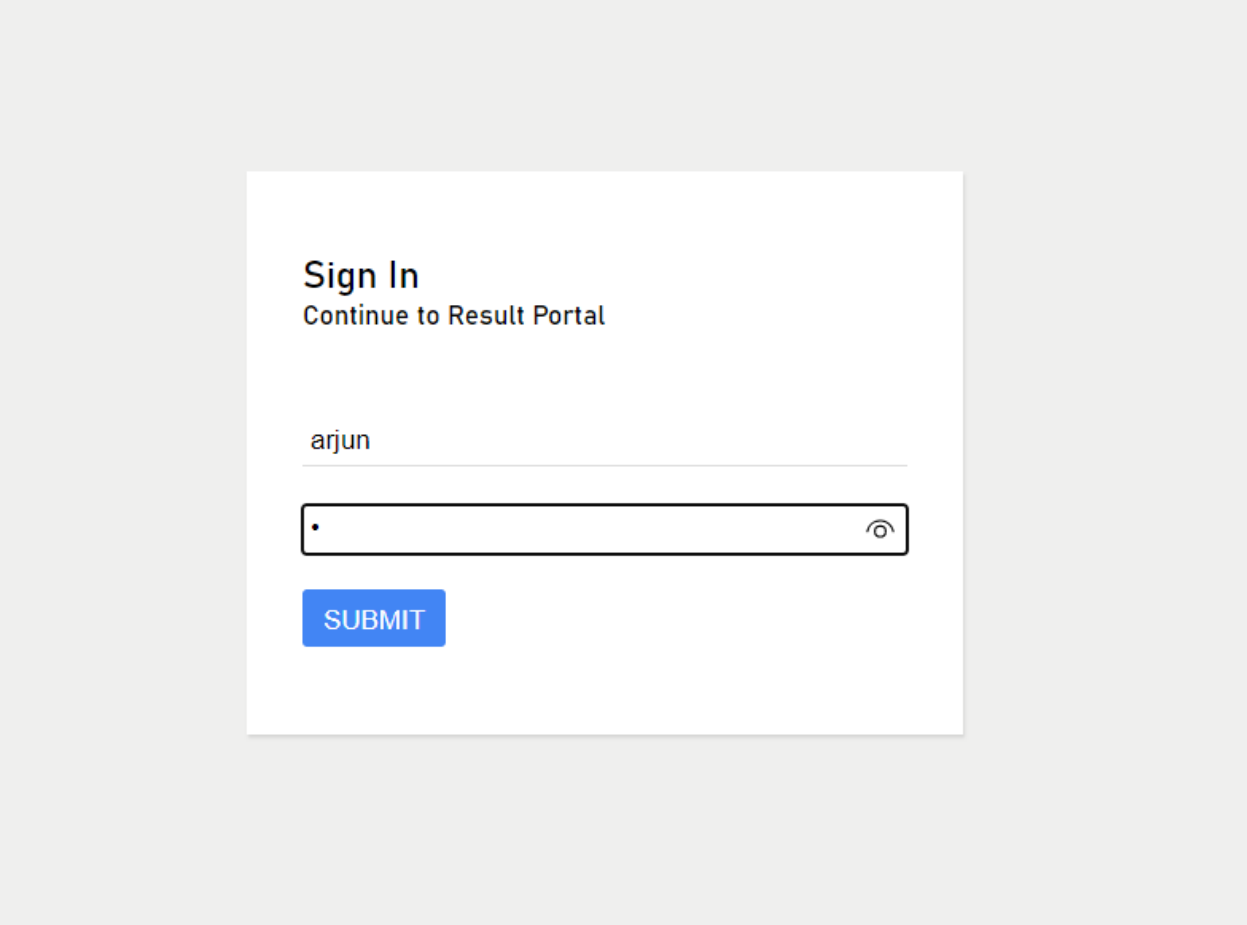
```
    Author     : arjun
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <%@ taglib uri="http://java.sun.com/jsp/jstl/core"
prefix="c"%>
        <%@ taglib uri="http://java.sun.com/jsp/jstl/sql"
prefix="sql"%>
        <link rel="stylesheet" type="text/css"
href="style.css">
        <title>Edit details</title>

    </head>
    <body>
        <sql:setDataSource var="ds"
driver="org.postgresql.Driver"

url="jdbc:postgresql://localhost:5432/postgres"
                            user="postgres"
password="shree9592"/>

        <c:set var="password" value="${param.password}"
scope="session"/>

        password ${password}
        <hr>contact Number  : ${param.contact}
        <hr>Email : ${param.email}
        <hr>address : ${param.address}
```

```
        <hr>user name : ${param.uname}



                <sql:update dataSource="${ds}"
var="result">
                    UPDATE ajt.students SET password =
?,address= ?,contact=?,email=? WHERE username = ?
                    <sql:param value="${password}"/>
                    <sql:param value="${param.address}"/>
                    <sql:param
value="${Long.parseLong(param.contact)}"/>
                    <sql:param value="${param.email}"/>
                    <sql:param value="${param.uname}"/>
                </sql:update>
                <hr>
                <div><h2> your details change succesfully
</h2></div>
                <hr>




    </body>
</html>
```

Input/Output:

Input:





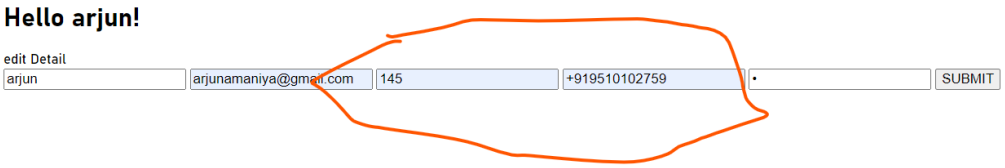
Before changing user details:

#	userid	username	password	name	address	rollno	sem	
5	1234	arjun	s	Arjun Maniya	asdfadsfa	69	6	arjunar

Output:

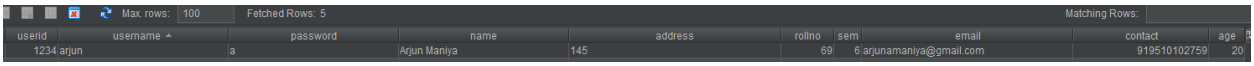
- set details by default as it was in database it remain same if user do not want to change it . (so this is details before change in database)

- detail that we have change :



- **after detail change :**

- **database :**



Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT69
B.Tech. IT, Sem: VI

Experiment – 08

Submitted By: 19ituos121
Roll No.: IT069
Name: Arjun Maniya

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:
// Source code
Index.html :

```
<%@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>
```

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>
```

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>
```

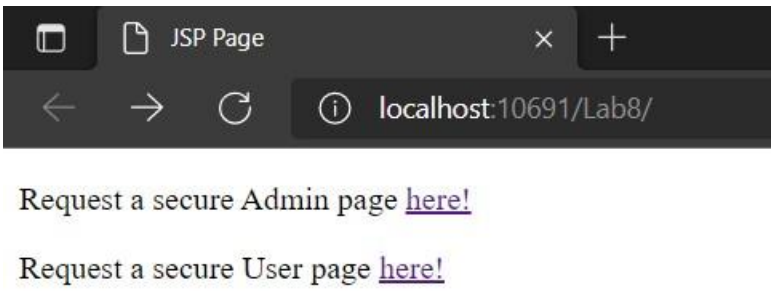
pageA.html :

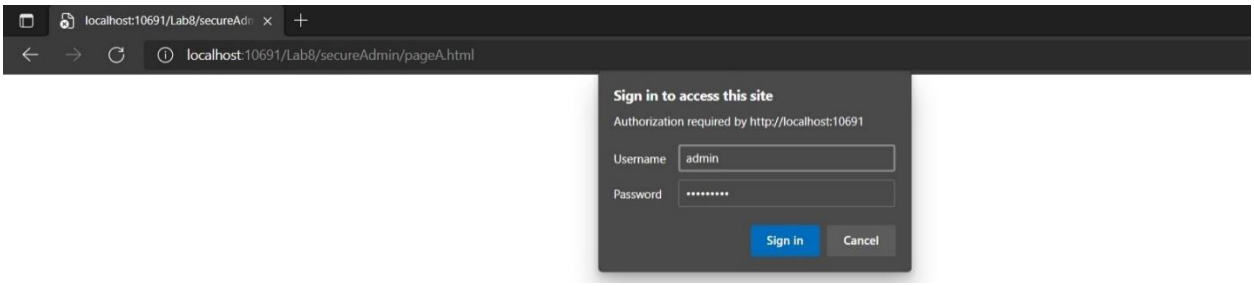
```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in
Project Properties.
```

```
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>
```

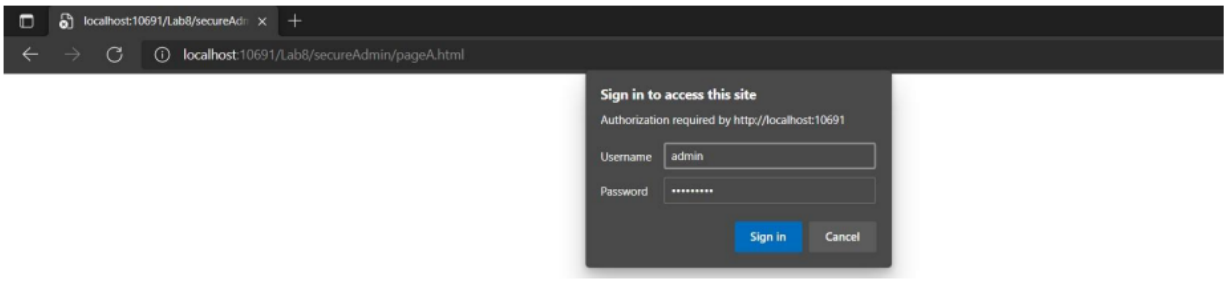
Input/Output:

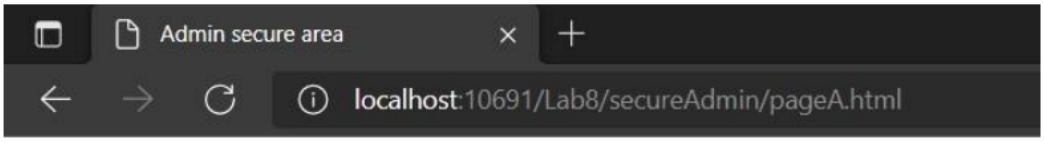
Input:

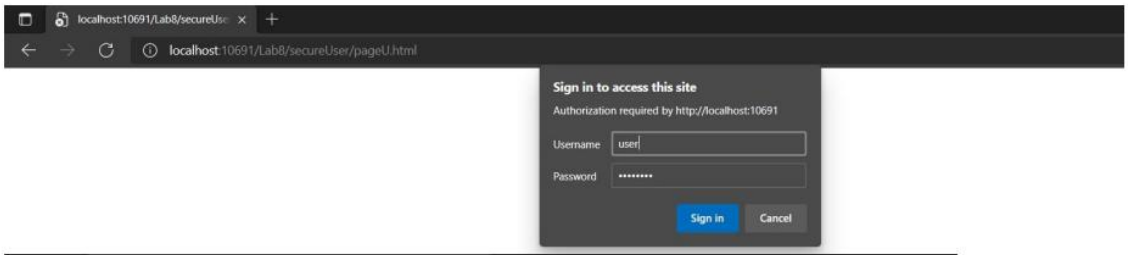


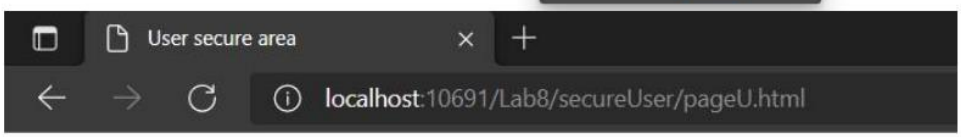


Output:









User Secure Area

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT69
B.Tech. IT, Sem: VI

Experiment – 09

Submitted By: 19ituos121
Roll No.: IT069
Name: Arjun Maniya

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:
// Source code

index.jsp :

```
<%--
    Document   : index
    Created on : 23-Feb-2022, 4:34:19 PM
    Author     : arjun
--%>

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

```
<%@ taglib uri="WEB-INF/tlds/myTag.tld" prefix="m" %>
Current Date and Time is: <m:Date/>

<m:Header align="center" color="Black" bgcolor="yellow"
border="solid" font="monospace" size="20"></m:Header>
```

HeaderHandler.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 myTagHandler;

import javax.servlet.jsp.JspException;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.tagext.TagSupport;
/**
 *
 * @author arjun
 */
public class HeaderHandler extends TagSupport{

    String align;
    String border;
    String bgcolor, color, font;
    int size;

    public void setAlign(String align) {
```

```
        this.align = align;
    }

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

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

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

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

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



    @Override
    public int doStartTag() throws JspException {
        JspWriter out = pageContext.getOut();

        try {
            out.print("<div  style = 'color:"+this.color+";
font-family:"+this.font+"; border-style:"+this.border+";
background-color : "+this.bgcolor + ";font-
```

```
size:"+this.size+"; text-align : "+this.align+"' > IT069
Arjun Maniya </div>");
        } catch (Exception e) {
            e.printStackTrace();
        }

        return SKIP_BODY;
    }



}
```

DateHandler :

```
/*
 * 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 myTagHandler;

import java.io.IOException;
import java.util.Calendar;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.tagext.TagSupport;

/**
 *
 * @author admin
```

```
 */
public class DateHandler extends TagSupport {

    /**
     * Called by the container to invoke this tag. The
implementation of this
     * method is provided by the tag library developer, and
handles all tag
     * processing, body iteration, etc.
     * @throws javax.servlet.jsp.JspException
     */

    @Override
     public int doStartTag() throws JspException {
         JspWriter out = pageContext.getOut();

        try{
            out.print(Calendar.getInstance().getTime());
        }
        catch(IOException e)
        {
        }
        return SKIP_BODY;
    }


}
```

myTag.tdl :

```
<?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>mytag</short-name>
  <uri>/WEB-INF/tlds/myTag</uri>

  <tag>
    <name>Date</name>
    <tag-class>myTagHandler.DateHandler</tag-class>
  </tag>
  <tag>
    <name>Header</name>
    <tag-class>myTagHandler.HeaderHandler</tag-class>
    <attribute>
      <name>align</name>
      <required>true</required>
      <rtexprvalue>true</rtexprvalue>
      <type>java.lang.String</type>
    </attribute>
    <attribute>
      <name>border</name>
      <required>true</required>
      <rtexprvalue>true</rtexprvalue>
      <type>java.lang.int</type>
    </attribute>
    <attribute>
      <name>bgcolor</name>
      <required>true</required>
      <rtexprvalue>true</rtexprvalue>
      <type>java.lang.int</type>
    </attribute>
```

```
    <attribute>
      <name>font</name>
      <required>true</required>
      <rtexprvalue>true</rtexprvalue>
      <type>java.lang.int</type>
    </attribute>
    <attribute>
      <name>color</name>
      <required>true</required>
      <rtexprvalue>true</rtexprvalue>
      <type>java.lang.String</type>
    </attribute>
    <attribute>
      <name>size</name>
      <required>true</required>
      <rtexprvalue>true</rtexprvalue>
      <type>java.lang.int</type>
    </attribute>
  </tag>
</taglib>
```

Input/Output:

Input:

```
<%@ taglib uri="WEB-INF/tlds/myTag.tld" prefix="m" %>
Current Date and Time is: <m:Date/>

<m:Header align="center" color="black" bgcolor="yellow" border="solid" font="monospace" size="20"></m:Header>
```

Output:

Current Date and Time is: Wed Mar 02 23:55:29 IST 2022

IT069 Arjun Maniya

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT69
B.Tech. IT, Sem: VI

Experiment – 10

Submitted By: 19ituos121
Roll No.: IT069
Name: Arjun Maniya

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**.**

Code:
// Source code

Bank.java :

```
package mp;


import java.rmi.*;




/**
 *
 * @author admin
 */
public interface Bank extends Remote{
    public void deposite(int acc_no,int money) throws
RemoteException;
```

```
    public void withdraw(int acc_no,int money) throws
RemoteException;
    public void showBalance(int acc_no) throws
RemoteException;

}
```

BankImpl.java :

```
package mp;

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

/**
 *
 * @author admin
 */
public class BankImpl extends UnicastRemoteObject implements
Bank {

    Connection conn = DBConn.getConnection();
    PreparedStatement pstmt = null;

    BankImpl() throws RemoteException{
         super();
    }

    public void deposite(int acc_no, int money) {
        try {
            pstmt = conn.prepareStatement("select * from bank
where acc_no = ?");
            pstmt.setInt(1, acc_no);
            ResultSet rs = pstmt.executeQuery();
            while (rs.next()) {
```

```
                pstmt = conn.prepareStatement("update bank
set balance = ? where acc_no = ?");
                pstmt.setInt(1, rs.getInt("balance") +
money);
                pstmt.executeUpdate();
            }
            printData(acc_no);
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }


    public void withdraw(int acc_no, int money) {
        try {
            pstmt = conn.prepareStatement("select * from bank
where acc_no = ?");
            pstmt.setInt(1, acc_no);
            ResultSet rs = pstmt.executeQuery();
            while (rs.next()) {
                pstmt = conn.prepareStatement("update bank
set balance = ? where acc_no = ?");
                pstmt.setInt(1, rs.getInt("balance") -
money);
                pstmt.executeUpdate();
            }
            printData(acc_no);
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }


    public void showBalance(int acc_no)  {
        try {
            pstmt = conn.prepareStatement("select * from bank
where acc_no = ?");
```

```
            pstmt.setInt(1, acc_no);
            ResultSet rs = pstmt.executeQuery();
            while (rs.next()) {
                System.out.println( rs.getInt("balance") +
"");
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }

    }

    private void printData(int acc_no) {
        try {
            pstmt = conn.prepareStatement("select * from bank
where acc_no = ?");
            pstmt.setInt(1, acc_no);
            ResultSet rs = pstmt.executeQuery();
            while (rs.next()) {
                System.out.println("\n Name : " +
rs.getString("name") + "\n Account Number : " +
rs.getInt("acc_no") + "\n Balance : " +
rs.getInt("balance"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

}
```

DBConn.java :

```
package mp;
```

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


/**
 *
 * @author admin
 */
public class DBConn {

    public static Connection conn = null;

    public static Connection getConnection() {
        if (conn == null) {
            try {
                Class.forName("org.postgresql.Driver");
                conn =
DriverManager.getConnection("jdbc:postgresql://localhost:5432
/postgres", "postgres", "shree9592");
                System.out.println("connection done");
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return conn;
    }
}
```

MyClient.java :

```
package mp;


import java.rmi.*;

public class MyClient {
```

```
    public static void main(String args[]) throws Exception {
        Bank b = (Bank)
Naming.lookup("rmi://localhost:6999/maniya");


       b.deposite(12345, 1000);
       b.withdraw(12345, 500);
       b.showBalance(12345);

    }
}
```

MyServer.java :

```
package mp;



/**
 *
 * @author admin
 */
import java.rmi.*;

public class MyServer {

    public static void main(String args[]) throws Exception {
        Remote r = new BankImpl();
        Naming.rebind("rmi://localhost:6999/maniya", r);
    }
}
```

Input/Output:

Input:

Max. rows: 100 | Fetched Rows: 5 | Matching Rows:

#	ACCNO	ACCHOLDER	AMT
1	1	abc	50000
2	2	def	20000.50
3	3	ghi	6000.00
4	4	jkl	10000.00
5	5	mno	50000.00

```
C:\Users\Windows\Onedrive\Documents\NetBEansProjects\ajtLab10>java.ajtlab10.BankClient
Account : Your Balance is : 0.0
Account : Deposited Successfully
Account : Insufficent Balance
Account : Your Balance is : 0.0
Account : Deposited Successfully
```

Max. rows: 100 | Fetched Rows: 5 | Matching Rows:

#	ACCNO	ACCHOLDER	AMT
1	1	abc	3000
2	2	def	20000.50
3	3	ghi	6000.00
4	4	jkl	10000.00
5	5	mno	50000.00

Dharmsinh Desai University, Nadiad
Department of Information Technology
Advanced Java Technology, IT69
B.Tech. IT, Sem: VI

Experiment – 11

Submitted By: 19ITOUS121
Roll No.: IT069
Name: Arjun Maniya

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:
// Source 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>
```

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);
}
```

IncomeServlet :

```
/*
 * 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>

}
```

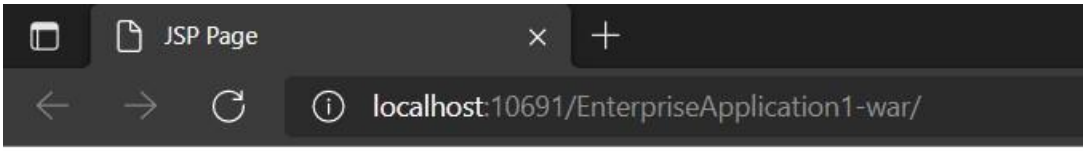
IncomeTaxBean.java :

```
/*
 * 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")

}
```

Input/Output:

Input:

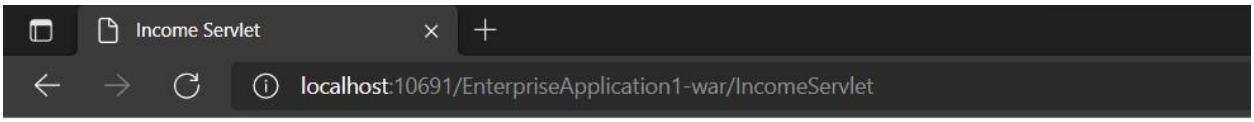


Income Tax Calculation

Enter Annual Salary : 455443

Submit

Output:



Servlet Income Servlet at /EnterpriseApplication1-war

Tax on AnnualSalary of 455443 is : 115443.0