EX.NO: 1	**Embed a map in a web page**
DATE:	

AIM:

To write a html program to embed a map with hotspots in a webpage and to show all the related information when the hotspots are clicked.

ALGORITHM:

Step 1: Start the main html program.

Step 2: Within the <body>tag, the title 'INDIA' is given using the <title> tag.

Step 3: The source image map is saved and the usemap attribute is used which creates a relationship between the <img> and the <map>.

Step 4: The <map> tag is used to define a client-side image-map.

Step 5: The <area> tag defines an area inside an image-map,the shape, coordinates and the hyperlink target for the area are mentioned within this tag.

Step 6: Using the hyperlink reference attribute the html files of different states are accessed when the defined hotspots are clicked.

Step 6: Close the <body> tag.

Step 7: Stop the program.

PROGRAM:

main.html

```html
<html>
<head>
<title>INDIA</title>
</head>
<body>
<img src="map.png" usemap="#india"/>
<map name="india">
<area shape="circle" coords="176,64,20" href="kashmir.html" alt="kashmir">
<area shape="circle" coords="187,208,20" href="delhi.html" alt="delhi">
<area shape="circle" coords="179,590,20" href="kerala.html" alt="kerala">
<area shape="circle" coords="218,575,20" href="Tamilnadu.html" alt="tamil
nadu">
</body>
</html>
```

kashmir.html

```html
<html>
<head>
<title>Kashmir</title>
</head>
<body>
```

```
<img src="C:\Users\Rose\Desktop\india\hill.jpg" width="500" height="300" >
<p style="text_align:center;">
Jammu and kashmir is a state in northern India,located mostly in the Himalayan mountains.
</p>
</body>
</html>
```

delhi.html

```
<html>
<head>
<title>Delhi</title>
</head>
<body>
<img src="C:\Users\Rose\Desktop\india\tajmahal.jpg" width="500" height="300"
>
<p style="text_align:center;">
Delhi,India's capital territory,is a massive metropolitan area in the country's north.
</p>
</body>
</html>
```

kerala.html

```
<html>
<head>
<title>Kerala</title>
</head>
<body>
<img src="C:\Users\Rose\Desktop\india\boat.jpg" width="600" height="400" >
<p style="text_align:center;">
Kerala,a state on India's tropical malabar coast ,has nearly 600km of Arabian sea
shore line.
</p>
</body>
</html>
```

tamilnadu.html

```
<html>
<head>
<title>Tamil nadu</title>
</head>
<body>
<img src="C:\Users\Rose\Desktop\india\central.jpg" width="600" height="350" >
<p style="text_align:center;">
Tamil Nadu,a southeast Indian state,is famed for its Dravidian-style hindu temples.
</p>
```

</body>

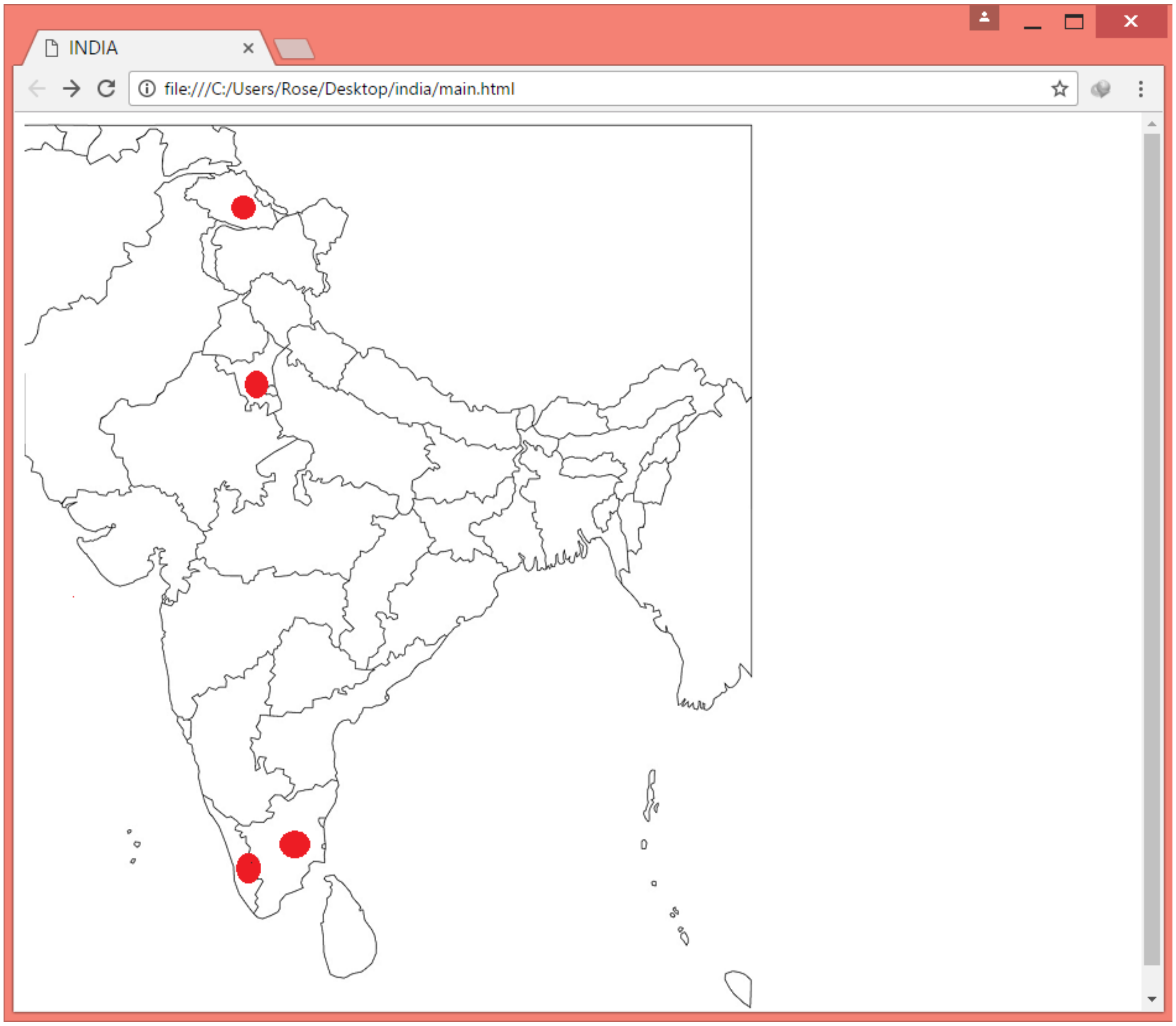
</html>

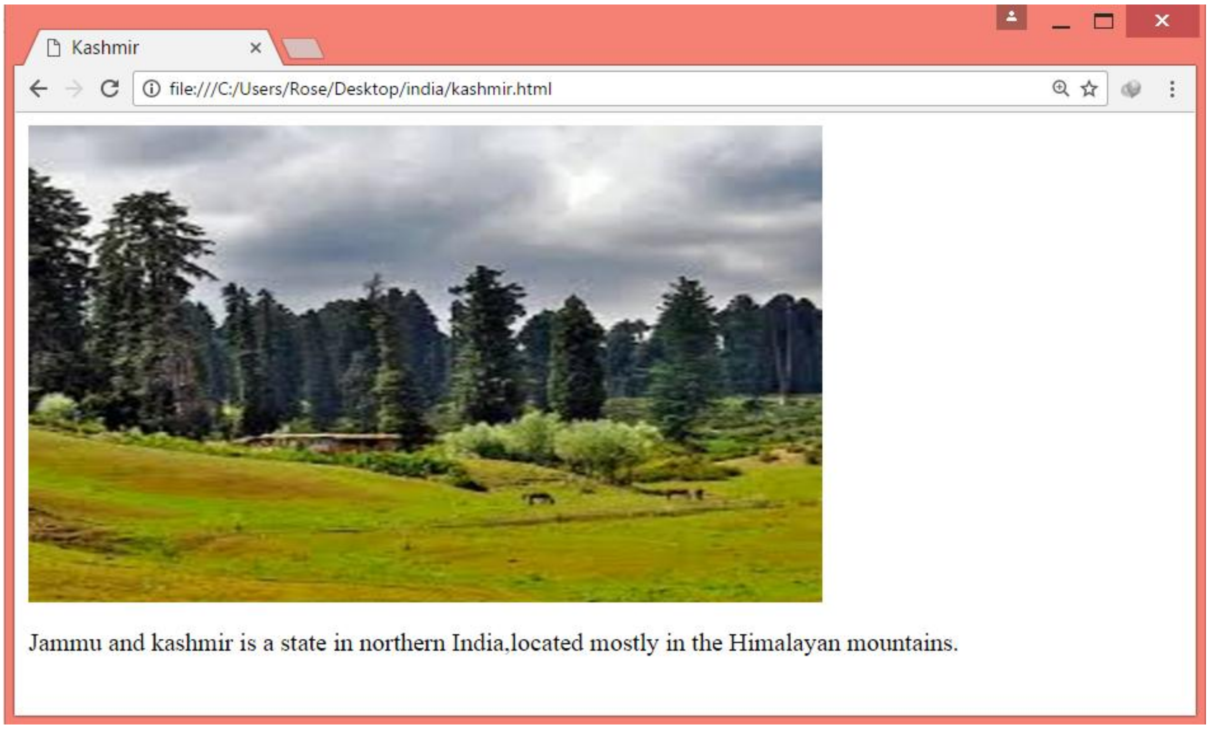
EXECUTION:

Step 1: Save all five html files with file extension as .html and image files with file extension as .jpg or .png in one directory.

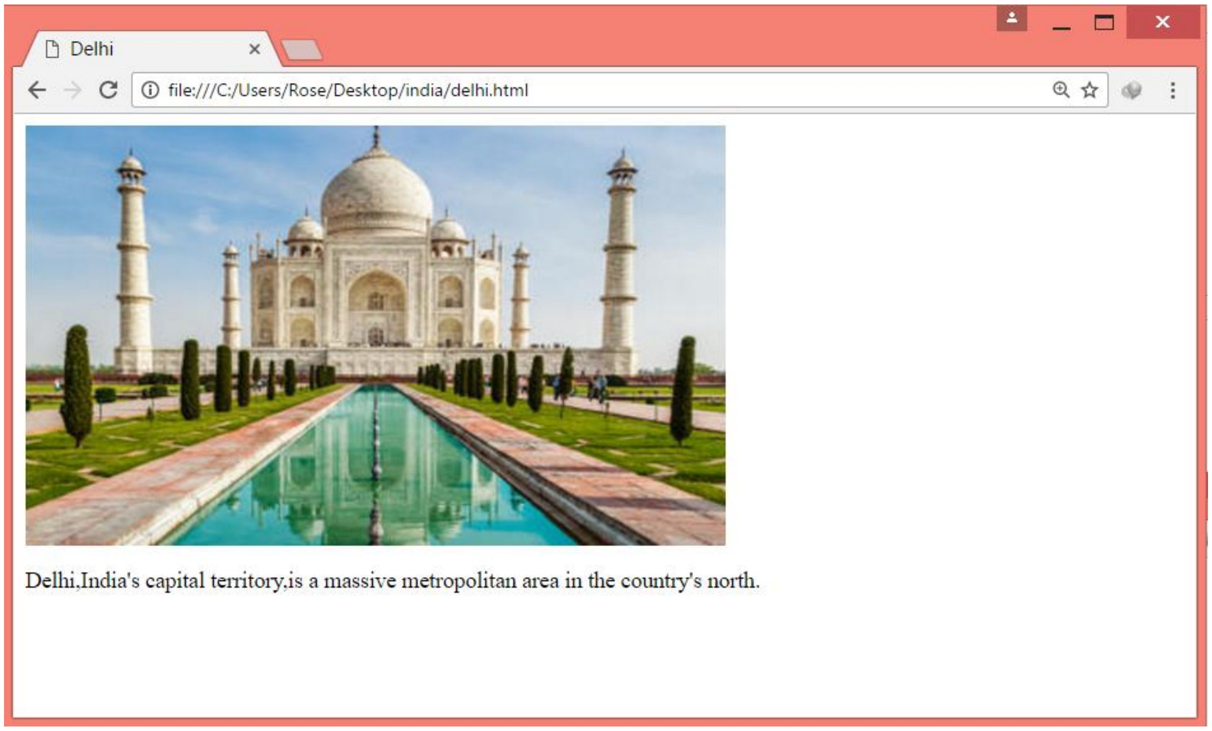
Step 2: Run the main.html in any web browser.

OUTPUT:

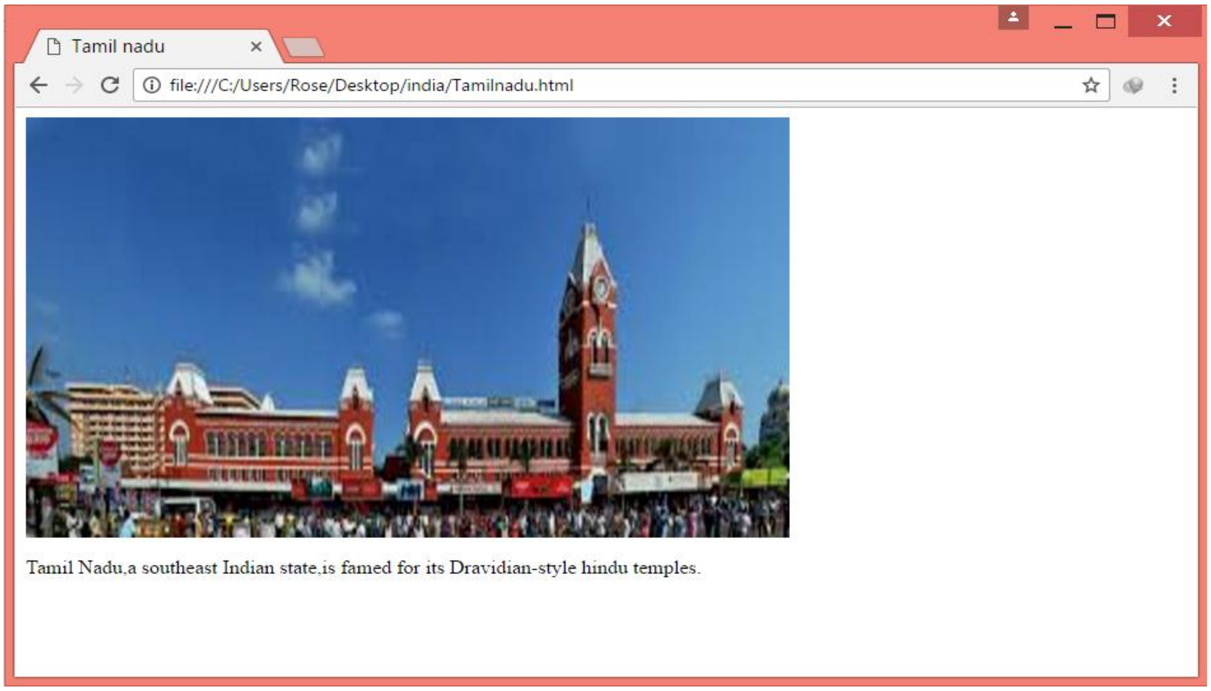




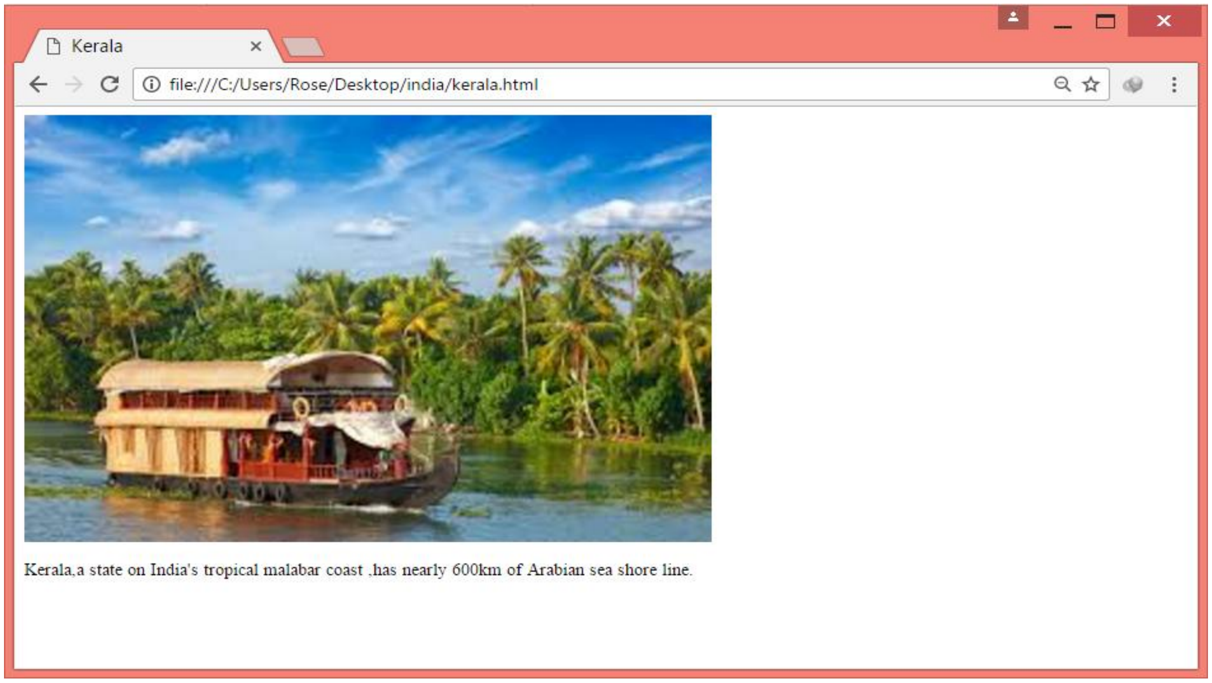
Jammu and kashmir is a state in northern India,located mostly in the Himalayan mountains.



Delhi,India's capital territory,is a massive metropolitan area in the country's north.



Tamil Nadu,a southeast Indian state,is famed for its Dravidian-style hindu temples.



Kerala,a state on India's tropical malabar coast ,has nearly 600km of Arabian sea shore line.

RESULT:

Thus the html program to embed a map with hotspots in a webpage and to show all the related information when the hotspots are clicked is executed successfully.

EX.NO: 2	**Display College Information Using Various Cascading Style Sheets**
DATE:	

AIM:

To create a web page that displays college information using various style sheets.

ALGORITHM:

Step 1: Start the html program.

Step 2: The <head> element is a container for all the head elements within which a heading <h1> and a <title> tag are defined.

Step 3: The <link> tag that defines a link between a document and an external resource is used to link to external style sheets is used here to link to the external css file ‘xyz.css’.

Step 4: Several inline and internal css properties such as font-style,font-family,color,text-align are used.

Step 4: Various type of <list> tags are used to list the several instituitions of the college and the different courses offered by them. The style sheet properties are applied here like the “list-style-type:circle” which specifies the type of list-item marker in a list.

Step 5: A table with Student name, Mark and Result fields is created .

Step 6: Stop the program.

PROGRAM:

allcss.html

```
<html>
<head><h1><center>ALL STYLE SHEETS</center></h1>
<title>USE of INTERNAL and EXTERNAL STYLESHEETS</title>
<link rel="stylesheet" href="xyz.css" type="text/css">
<style type="text/css">
.vid{font-family:verdana;font-style:italic;color:red;text-align:center}
.ani{font-family:tahoma;font-style:italic;font-size:20;text-align:center;}
font{font-family:georgia;color:blue;font-size:20}
ul{list-style-type:circle}
</style>
</head>
<body>
<ol style="list-style-type:lower-alpha">
<b>R.M.K. GROUP OF INSTITUTIONS</b><br>
<li>R.M.K. ENGINEERING COLLEGE
<li>R.M.D. ENGINEERING COLLEGE
<li>R.M.K. COLLEGE OF ENGINEERING AND TECHNOLOGY
</ol>
<p class="ani"> R.M.K. COLLEGE OF ENGINEERING AND TECHNOLOGY
        <br>It is a Approved by AICTE(All India Council for Technical
        Education).It is affiliated to Anna University.<br></p>
```

```
<h2 class="vid"> R.M.K. COLLEGE OF ENGINEERING AND TECHNOLOGY
        </h2>

<br><font>It is an ISO certified Institution</font><br>

<font>

<h2>List of Courses offered</h2>

<ul>

<li>Computer Science and Engineering</li>

<li>Electronics and Communication Engineering</li>

<li>Mechanical Engineering</li>

<li>Electrical and Electronics Engineering</li>

<li>Artificial Intelligence and Data Science/li>

</ul>

</font>

<h3>Results of CSE students</h3>

<table width="100%" cellspacing="2" cellpadding="2" border="5">

<tr>

<th>S.NAME</th>

<th>MARKS</th>

<th>RESULT</th>

</tr>

<tr>

<td align="center">Ajay</td>

<td align="center">100</td>
```

```
<td align="center">pass</td>
</tr>
<tr>
<td align="center">Aditya</td>
<td align="center">99</td>
<td align="center">pass</td>
</tr>
<tr>
<td align="center">Arun</td>
<td align="center">98</td>
<td align="center">pass</td>
</tr>
</table>
</body>
</html>
```

xyz.css

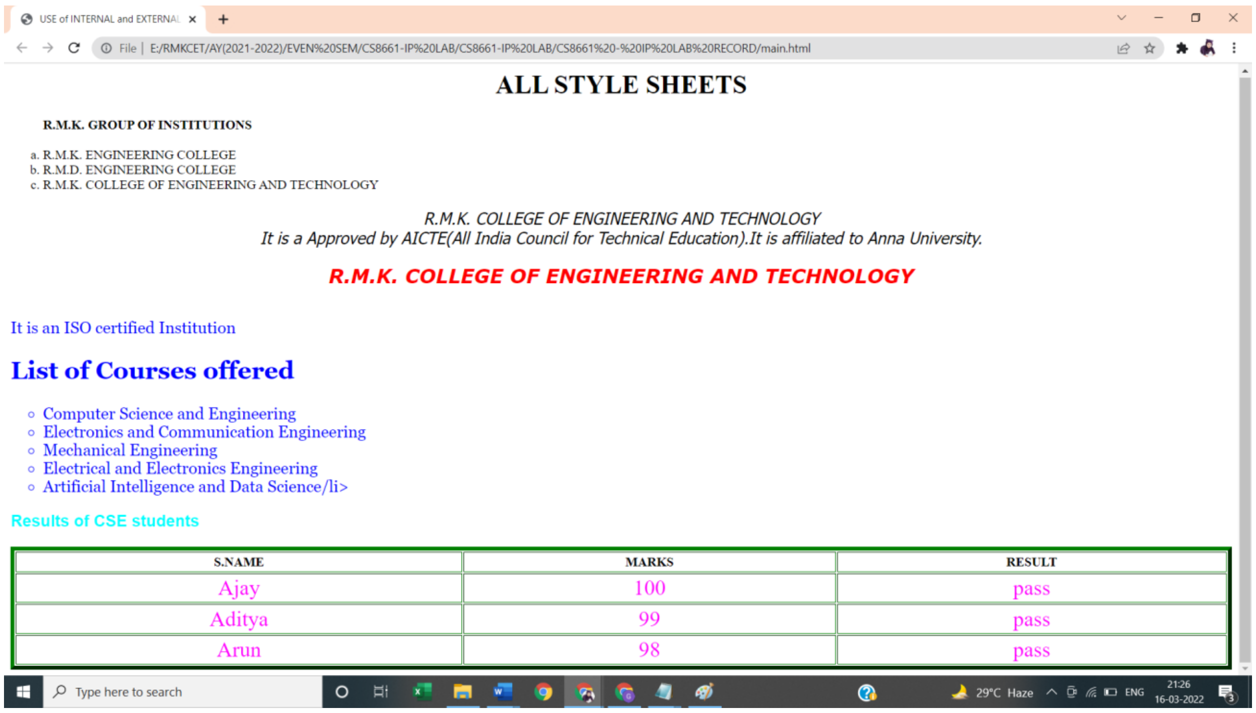
```
h3{font-family:arial;font-size:20;color:cyan}
table{border-color:green}
td{font-size:20pt;color:magenta}
```

EXECUTION:

Step 1: Save html file as ALLCSS.html and external css file as xyz.css.

Step 2: Run the AllCSS.html in any web browser.

OUTPUT:



USE of INTERNAL and EXTERNAL

E:/RMKCET/AY(2021-2022)/EVEN%20SEM/CS8661-IP%20LAB/CS8661-IP%20LAB/CS8661%20-%20IP%20LAB%20RECORD/main.html

ALL STYLE SHEETS

R.M.K. GROUP OF INSTITUTIONS

a. R.M.K. ENGINEERING COLLEGE
b. R.M.D. ENGINEERING COLLEGE
c. R.M.K. COLLEGE OF ENGINEERING AND TECHNOLOGY

R.M.K. COLLEGE OF ENGINEERING AND TECHNOLOGY
It is a Approved by AICTE(All India Council for Technical Education).It is affiliated to Anna University.

R.M.K. COLLEGE OF ENGINEERING AND TECHNOLOGY

It is an ISO certified Institution

List of Courses offered

- Computer Science and Engineering
- Electronics and Communication Engineering
- Mechanical Engineering
- Electrical and Electronics Engineering
- Artificial Intelligence and Data Science/li>

Results of CSE students

S.NAME	MARKS	RESULT
Ajay	100	pass
Aditya	99	pass
Arun	98	pass

RESULT:

Thus the creation of web page that displays college information using various style sheets is executed successfully.

EX.NO: 3	**Validate the registration, user login, user profile and payment by credit card pages using JavaScript**
DATE:	

AIM:

To validate the registration, user login, user profile and payment by credit card pages using JavaScript.

ALGORITHM:

Step 1: Start the html program.

Step 2: The <head> element is a container for all the head elements within which a heading <h1> and a <title> tag are defined.

Step 3: The <link> tag that defines a link between a document and an external resource is used to link to external style sheets is used here to link to the external css file ‘xyz.css’.

Step 4: Several inline and internal css properties such as font-style,font-family,color,text-align are used.

Step 4: Various type of <list> tags are used to list the several instituitions of the college and the different courses offered by them. The style sheet properties are applied here like the “list-style-type:circle” which specifies the type of list-item marker in a list.

Step 5: A table with Student name, Mark and Result fields is created .

Step 6: Stop the program.

PROGRAM:

registration.html

```
<html>
<head>
<script>
function GEEKFORGEEKS()
{
        var name = document.forms["RegForm"]["Name"];
        var email = document.forms["RegForm"]["EMail"];
        var phone = document.forms["RegForm"]["Telephone"];
        var what = document.forms["RegForm"]["Subject"];
        var password = document.forms["RegForm"]["Password"];
        var address = document.forms["RegForm"]["Address"];

        if (name.value == "")
        {
                window.alert("Please enter your name.");
                name.focus();
                return false;
        }

        if (address.value == "")
        {
                window.alert("Please enter your address.");
                address.focus();
                return false;
        }

        if (email.value == "")
        {
                window.alert("Please enter a valid e-mail address.");
                email.focus();
                return false;
        }

        if (phone.value == "")
        {
                window.alert("Please enter your telephone number.");
                phone.focus();
                return false;
        }

        if (password.value == "")
        {
                window.alert("Please enter your password");
                password.focus();
                return false;
        }

        if (what.selectedIndex < 1)
        {
                alert("Please enter your course.");
                what.focus();
                return false;
        }

        return true;
}</script>

<style>
GEEKFORGEEKS {
        margin-left: 70px;
        font-weight: bold ;
        float: left;
        clear: left;
        width: 100px;
        text-align: left;
        margin-right: 10px;
        font-family:sans-serif,bold, Arial, Helvetica;
        font-size:14px;
}

div {
        box-sizing: border-box;
        width: 100%;
        border: 100px solid black;
        float: left;
        align-content: center;
        align-items: center;
}

form {
        margin: 0 auto;
        width: 600px;
}</style></head>

<body bgcolor="pink">
<h1 style="text-align: center"> REGISTRATION FORM </h1>
<form       name="RegForm"       action="/submit.php"       onsubmit="return
        GEEKFORGEEKS()" method="post">

        <p>Name: <input type="text" size=65 name="Name"> </p><br>
        <p> Address: <input type="text" size=65 name="Address"> </p><br>
        <p>E-mail Address: <input type="text" size=65 name="EMail"> </p><br>
        <p>Password: <input type="text" size=65 name="Password"> </p><br>
        <p>Telephone: <input type="text" size=65 name="Telephone"> </p><br>

        <p>SELECT YOUR COURSE
                <select type="text" value="" name="Subject">
                        <option>BE</option>
                        <option>BTECH</option>
                        <option>BBA</option>
                        <option>BCA</option>
                        <option>B.COM</option>
                        <option>GEEKFORGEEKS</option>
                </select></p><br><br>
        <p>Comments: <textarea cols="55" name="Comment"> </textarea></p>
        <p><input type="submit" value="send" name="Submit">
                <input type="reset" value="Reset" name="Reset">
        </p>
</form>
</body>
</html>
```

login.html

```
<html>
<head>
<SCRIPT LANGUAGE="JAVASCRIPT">
function essentials_of_validation(form1)
{
 var return_value=true;
 var username=form1.txtusername.value;
 var password1=form1.txtpassword1.value;
 var password2=form1.txtpassword2.value;
 if(username.length < 8)
{
 return_value=false;
 window.alert("user name less thn 8 chars");
}
if(password1.length<6)
{
 return_value=false;
 window.alert(" pwd should be > 6 char's ");
 form1.txtpassword1.value="";
 form1.txtpassword2.value="";
}
if(password1!=password2)
{
return_value=false;
window.alert("ur password mismatched ");
form1.txtpassword1.value="";
form1.txtpassword2.value="";
}
return return_value;
}
</script>
</head>
<body bgcolor="skyblue">
<marquee><b><u><font face="comic sans ms" color="light blue">login
here</font></u></b></marquee>
<form name="form1" onSubmit="essentials_of_validation(this)">
<label>Username: </label><input type="text" name="txtusername" size=20>
<br>Password: <input type="password" name="txtpassword1" size=20>
```

```
<br>Confirm
password:<input type="password" name="txtpassword2" size=20>
<br><br>
<input type="submit" value="submit">
<input type="reset" value="reset">
</form>
</body>
</html>
```

payment.html

```
<html>
<head><title>
payment</title>
<SCRIPT LANGUAGE="JAVASCRIPT">
function essentials_of_validation(form1)
{
var current=new Date();
var return_value=true;
var username=form1.txtusername.value;
var password1=form1.txtpassword1.value;
var a=form1.dd.value;
var b=form1.mm.value;
var c=form1.yyyy.value;
if(isNaN(username))
{
window.alert("Not a valid account number");
}
if(a<32 && b<13&& c>=current.getFullYear())
{
 if(c>current.getFullYear())
 {
 window.alert(" you are validated");
 }
 else if(c=current.getFullYear())
 {
```

```
 if(b>current.getMonth())
 {
 window.alert(" you are validated");
 }
 else if(b=current.getMonth())
 {
 if(a>current.getDate())
 {
 window.alert("you are validated");
 }
 }
 }
 else
 {
 window.alert("Your card has expired");
 }
}
 else
 {
 window.alert(" card has expired");
 }
}
</script>
</head>
<body bgcolor="pink">
<marquee><strong>Enjoy the Shopping with special Offers</strong></marquee>
<form name="form1" onSubmit="essentials_of_validation(this)">
<font size="+2">
Payment Through<br>
<input name="pay" type="radio" >Credit card
<input name="pay" type="radio" >Debit card<br>
Bank
<select name="bank">
<option selected>Select
<option>HSBC
```

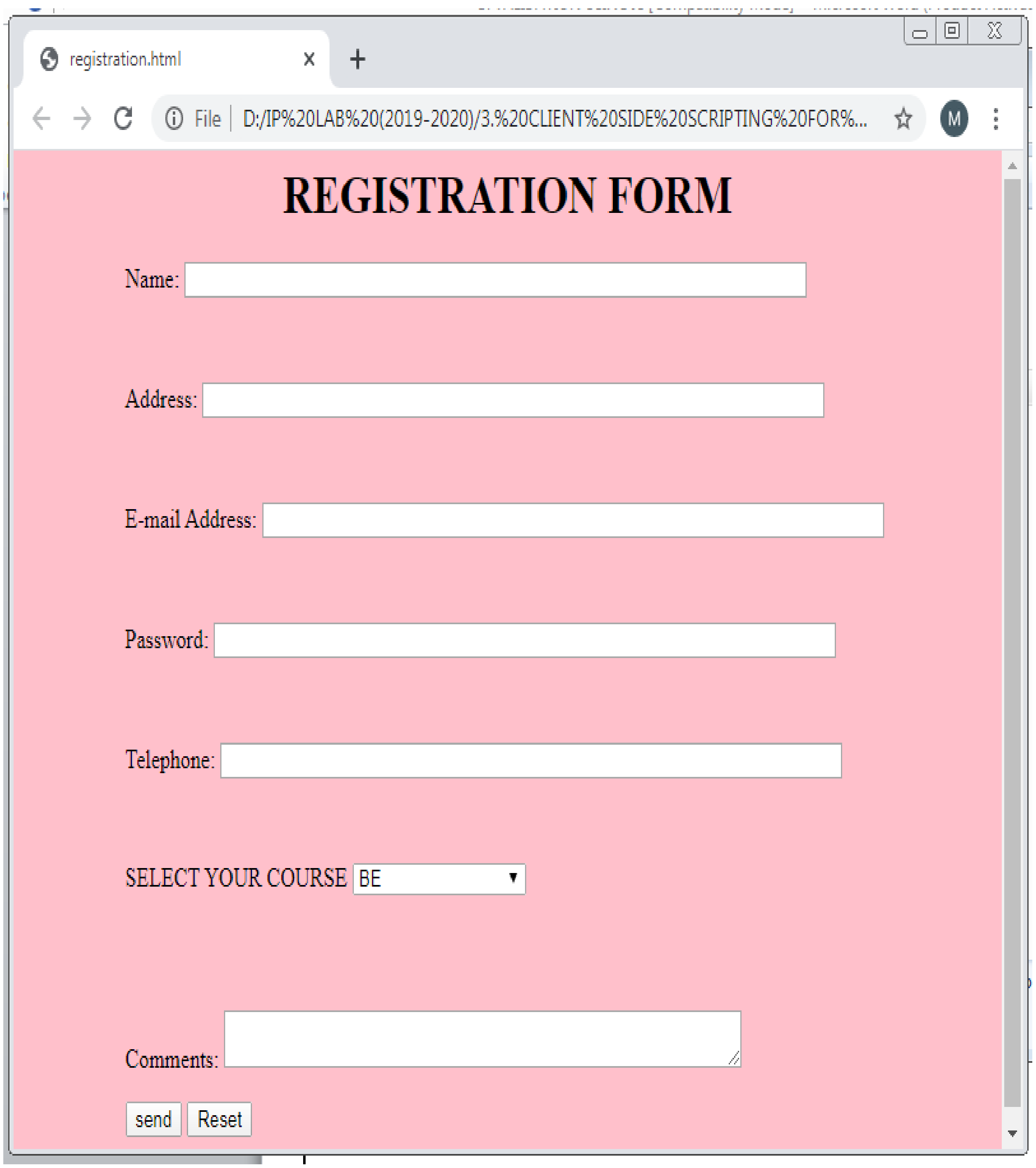
```
<option>SBI
<option>ICICI
<option>others
</select><br>
Account/Card number:<input name="txtusername" type="textbox"><br>
Net banking id/Password<input name="txtpassword1" type="password"><br>
Enter date of expiryof account/card<input name="dd" type="text" size=2>(dd)
<input name="mm" type="text" size=2>(mm)
<input name="yyyy" type="text" size=4>(yyyy)
<br>
<input type="submit" value="Accept">
<input type="reset" value="Reject">
</form>
</body>
</html>
```

EXECUTION:

Step 1: Save all three html files with file extension as .html in one directory.

Step 2: Run the html files in any web browser.

OUTPUT:



registration.html

File | D:/IP%20LAB%20(2019-2020)/3.%20CLIENT%20SIDE%20SCRIPTING%20FOR%...

REGISTRATION FORM

Name:

Address:

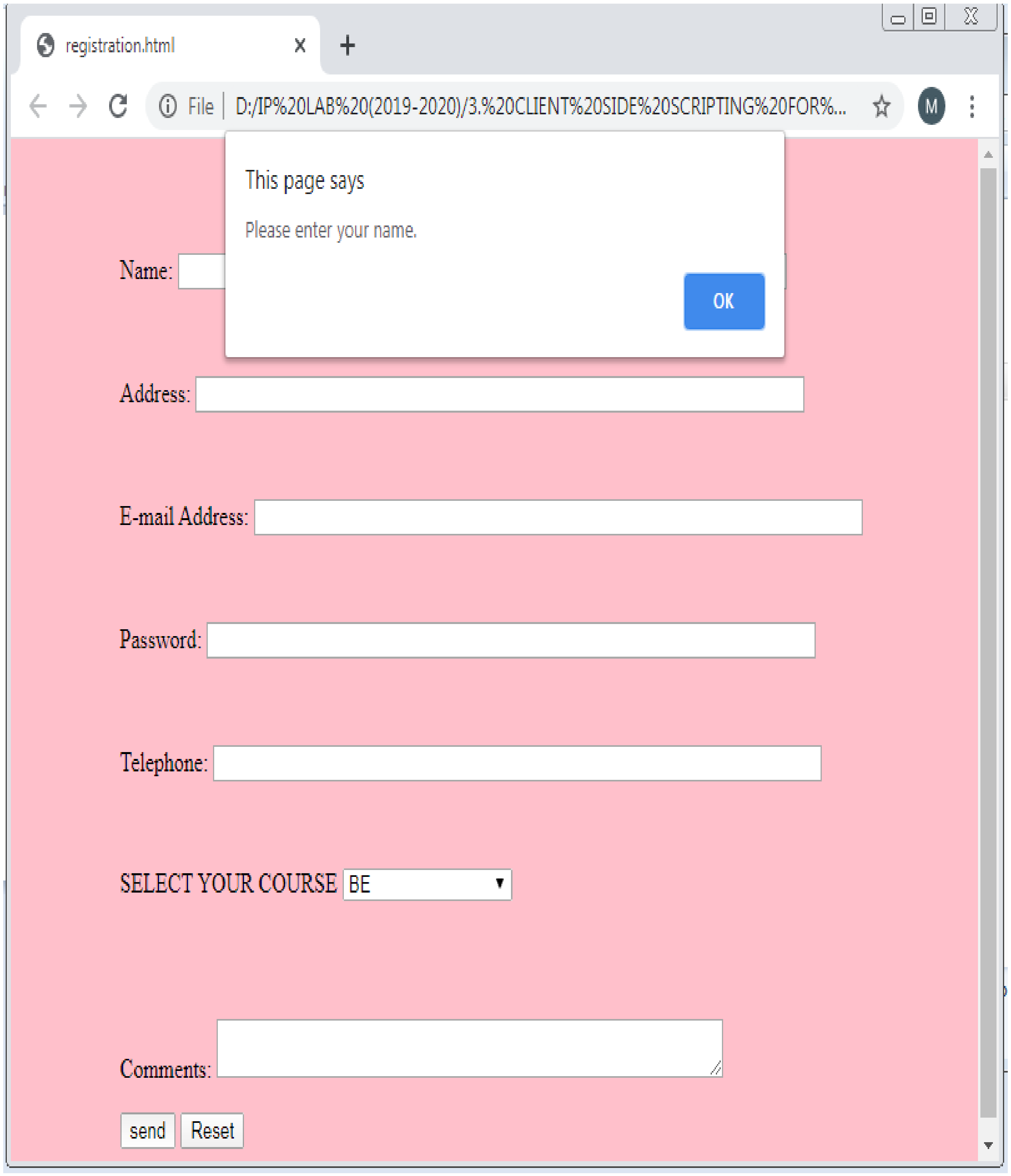
E-mail Address:

Password:

Telephone:

SELECT YOUR COURSE BE

Comments:


registration.html
File | D:/IP%20LAB%20(2019-2020)/3.%20CLIENT%20SIDE%20SCRIPTING%20FOR%...
OK
send
Reset

This page says

Please enter your name.

Name:

Address:

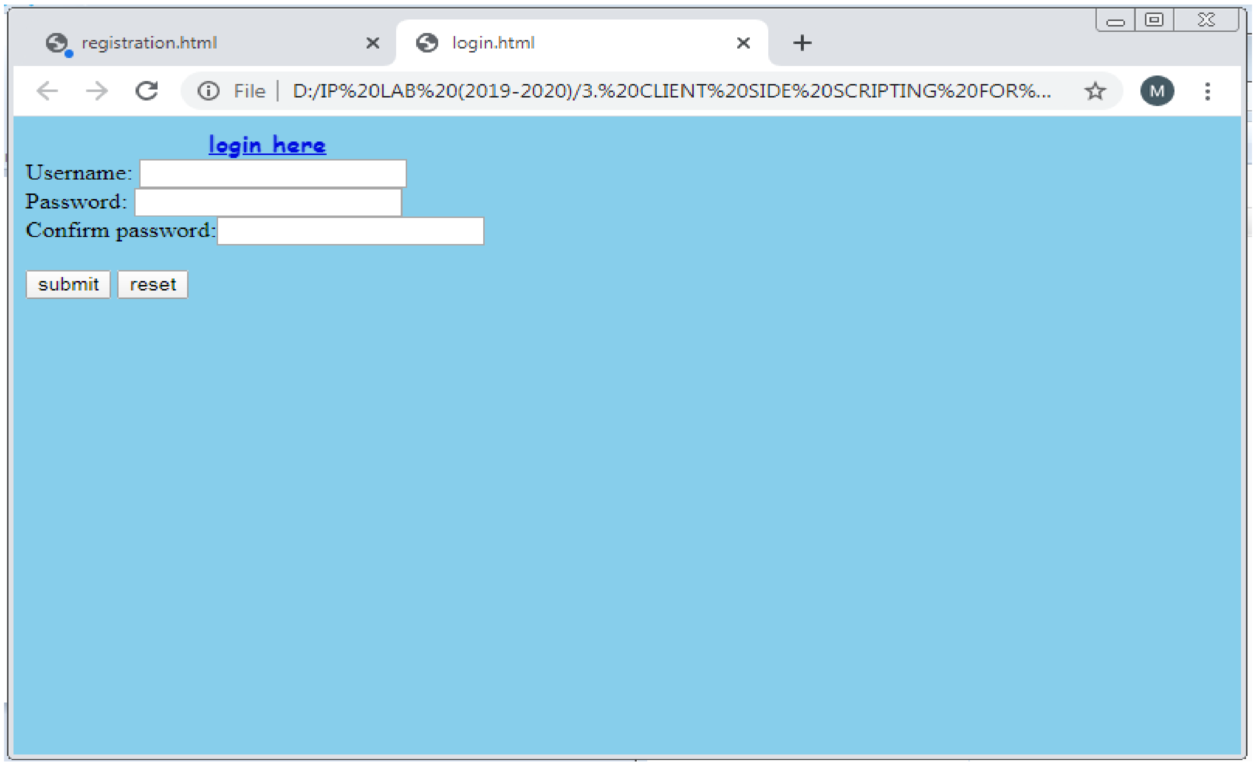
E-mail Address:

Password:

Telephone:

SELECT YOUR COURSE BE

Comments:


registration.html
login.html
D:/IP%20LAB%20(2019-2020)/3.%20CLIENT%20SIDE%20SCRIPTING%20FOR%...

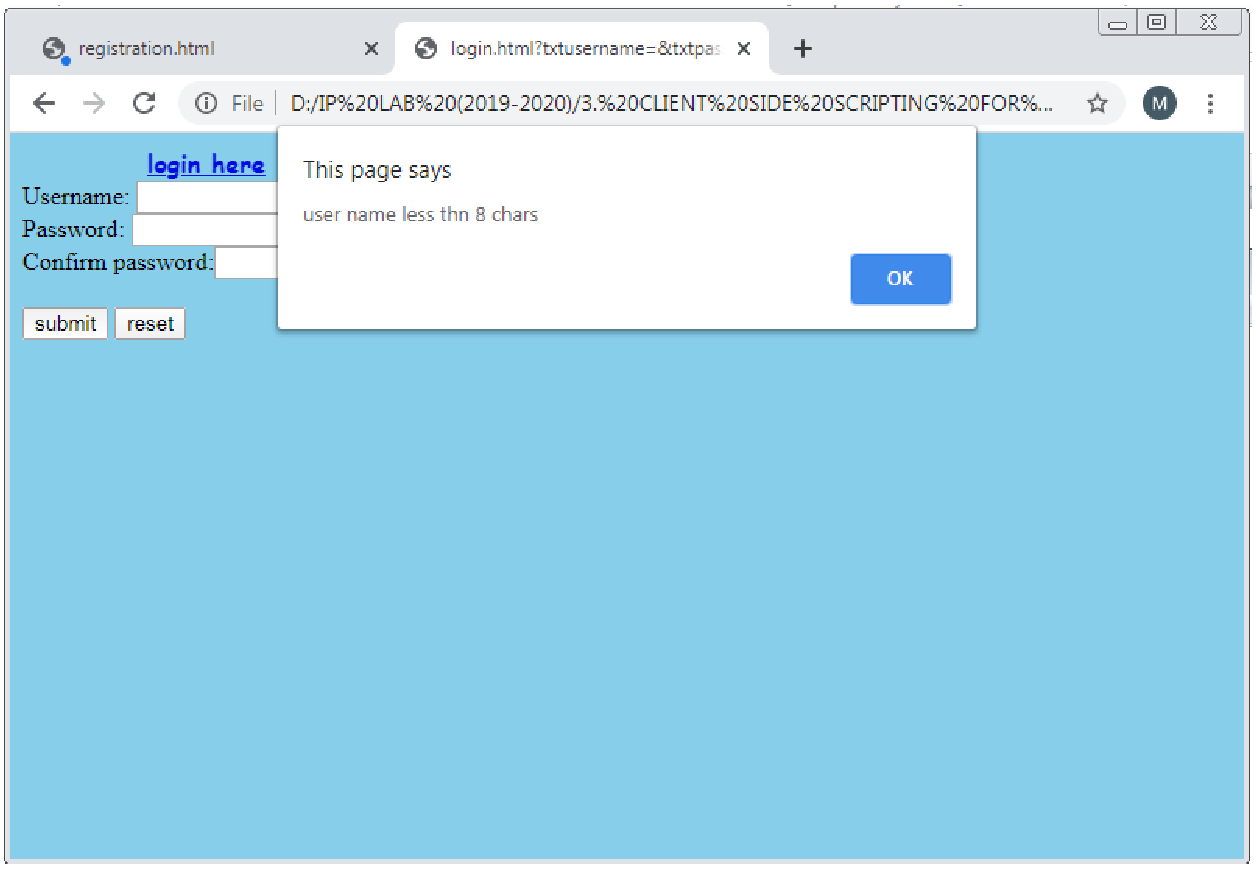
login here

Username:

Password:

Confirm password:

submit reset


registration.html
login.html?txtusername=&txtpas
D:/IP%20LAB%20(2019-2020)/3.%20CLIENT%20SIDE%20SCRIPTING%20FOR%...

login here

Username:

Password:

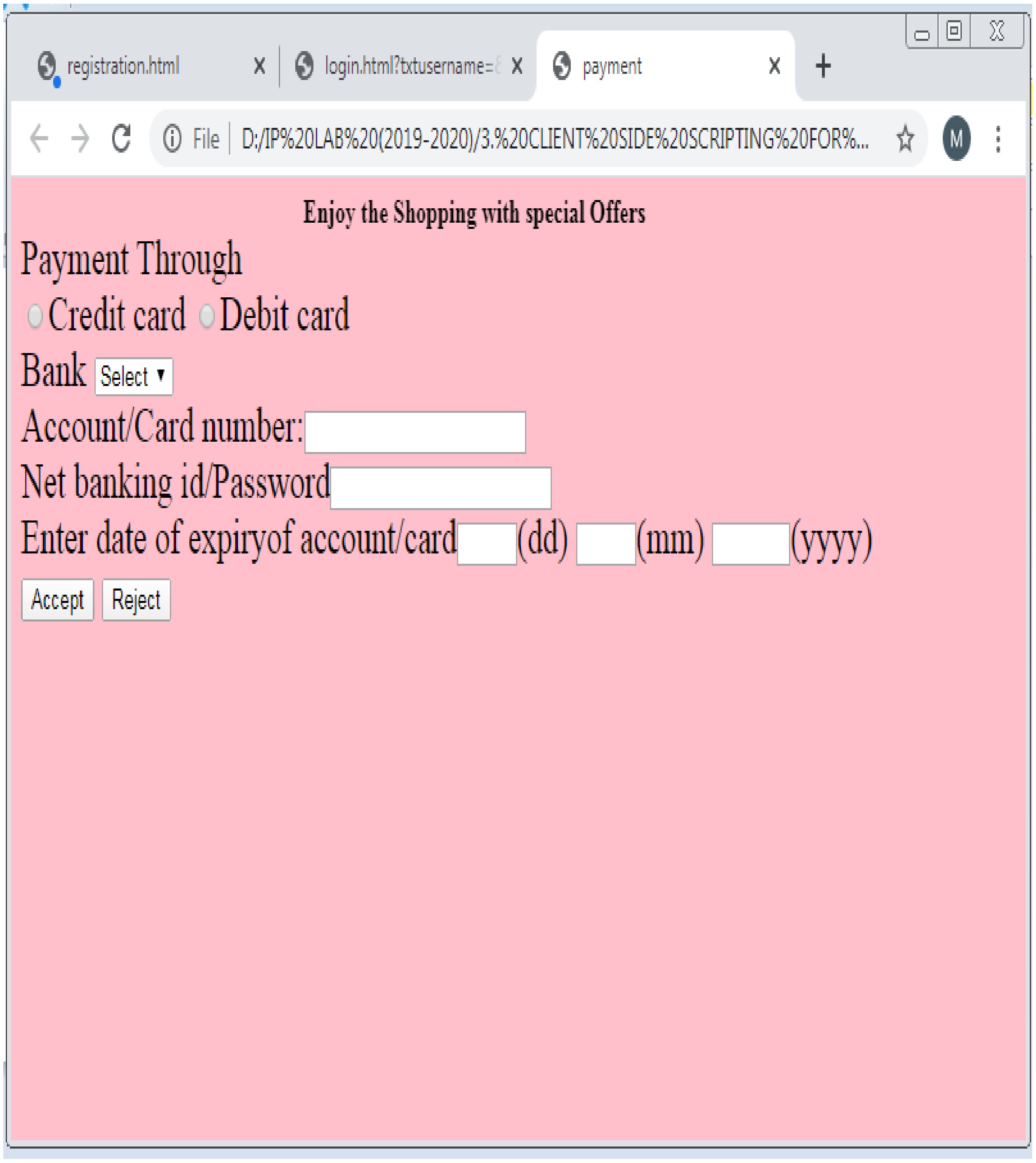
Confirm password:

submit reset

This page says

user name less thn 8 chars

OK


registration.html
login.html?txtusername=
payment
File
D:/IP%20LAB%20(2019-2020)/3.%20CLIENT%20SIDE%20SCRIPTING%20FOR%...

Enjoy the Shopping with special Offers

Payment Through

Credit card Debit card

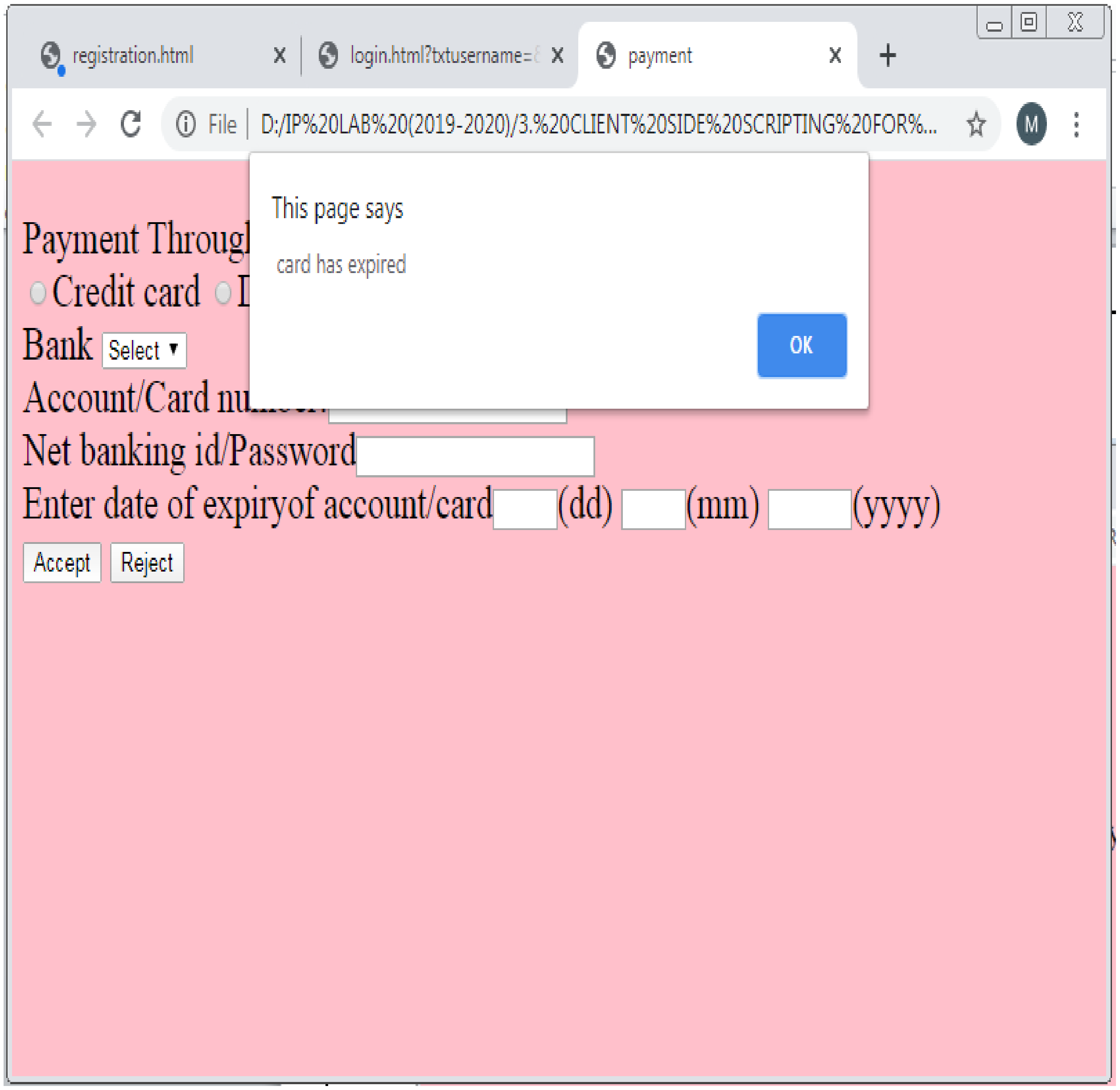
Bank Select

Account/Card number:

Net banking id/Password

Enter date of expiryof account/card (dd) (mm) (yyyy)

Accept Reject



registration.html | login.html?txtusername= | payment

D:/IP%20LAB%20(2019-2020)/3.%20CLIENT%20SIDE%20SCRIPTING%20FOR%...

This page says

card has expired

OK

Payment Throug

Credit card

Bank Select

Account/Card nu

Net banking id/Password

Enter date of expiryof account/card (dd) (mm) (yyyy)

Accept Reject

RESULT:

Thus the validation for the registration, user login, user profile and payment by credit card pages using JavaScript is executed successfully.

EX.NO: 4 (i)	**Invoke Servlets From HTML Forms**
DATE:	

AIM:

To write a servlet program to invoke servlets from HTML forms.

ALGORITHM:

HTML:

Step 1: Start the program.

Step 2: Define form element with attribute name, method and action.

Step 3: call servlet program in form action attribute.

Step 4: Get username and password from user by using input type as text and password.

Step 5: submit username and password using input type as submit.

Step 6: Stop the program.

SERVLET:

Step 1: Start the program.

Step 2: Import necessary packages to access the predefined classes and its methods.

Step 3: Define doPost() method by passing request and response object.

Step 4: Create an object for PrintWriter class.

Step 5: Read user details from the form using getParameter() method.

Step 6: Display the user details by using object of PrintWriter class.

Step 7: Stop the program.

PROGRAM:

param.html

```
<html>

<body>

<h1>Form Details</h1>

<form name="myform" method="post" action="paramname"><br>

<h3>Enter Your Name:<input type="text" name="username" value=""></h3><br>

<h3>Enter Your Password:<input type="password" name="pass" value=""></h3>

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

</form>

</body>

</html>
```

paramname.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class paramname extends HttpServlet
{
	public void doPost(HttpServletRequest req, HttpServletResponse res) throws
ServletException,IOException
	{
		res.setContentType("text/html");
		PrintWriter pw=res.getWriter();
		String name=req.getParameter("username");
		String pass=req.getParameter("pass");
		pw.println("<b>Your Data is Here:</b>");
		pw.println("USERNAME="+name);
		pw.println("PASSWORD="+pass);
		pw.close();
	}
}
```

web.xml:

```
<web-app>
 <servlet>
<servlet-name>paramname</servlet-name>
<servlet-class>paramname</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>paramname</servlet-name>
<url-pattern>/paramname</url-pattern>
</servlet-mapping>
<welcome-file-list>
<welcome-file>param.html</welcome-file>
</welcome-file-list>
</web-app>
```

EXECUTION:

Step 1: Save html file as param.html, servlet file as paramname.java and xml file as web.xml

Step 2: Set class path for servlet-api.jar files as follows

i. My Computer→Properties→Advanced System Settings→Environment Variables→User variables→New
ii. Set Variable name as classpath and Variable value as your jar file location.
iii. Example C:\Program Files\Apache Software Foundation\Tomcat 7.0\lib\servlet-api.jar;
iv. Click ok→ok→ok→close.

Step 3: Open cmd prompt and goto your servlet program directory.

Step 4: Set path for javac compiler in cmd prompt as follows

```
set path=" C:\Program Files\Java\jdk1.7.0_79\bin";
```

Step 5: Compile your servlet program as follows

```
javac  paramname.java
```

It generates Bytecode file (or Object file) as paramname.class

Step 6: Goto C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps

Step 7: In the above webapps folder create your own new project folder. For example I created and named it as "india".

Step 8: Goto C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\india

Step 9: In the above india folder create one folder which is named as WEB-INF(should be capital letters) .

Step 10: Goto C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\india\WEB-INF

Step 11: In the above WEB-INF folder create one folder which is named as classes (should be small letters).

Step 12: Now place your html, xml and servlet class file in the following directories.

a) Place param.html in C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\india

b) Place web.xml in C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\india\WEB-INF

c) Place paramname.class in C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\india\WEB-INF\classes

Step 13: Start the Apache Tomcat server as follows:

a) Goto C:\Program Files\Apache Software Foundation\Tomcat 7.0\bin.

b) Open Tomcat7w→Click start button.

Step 14: Open your browser and type in URL location as follows:

localhost:8089

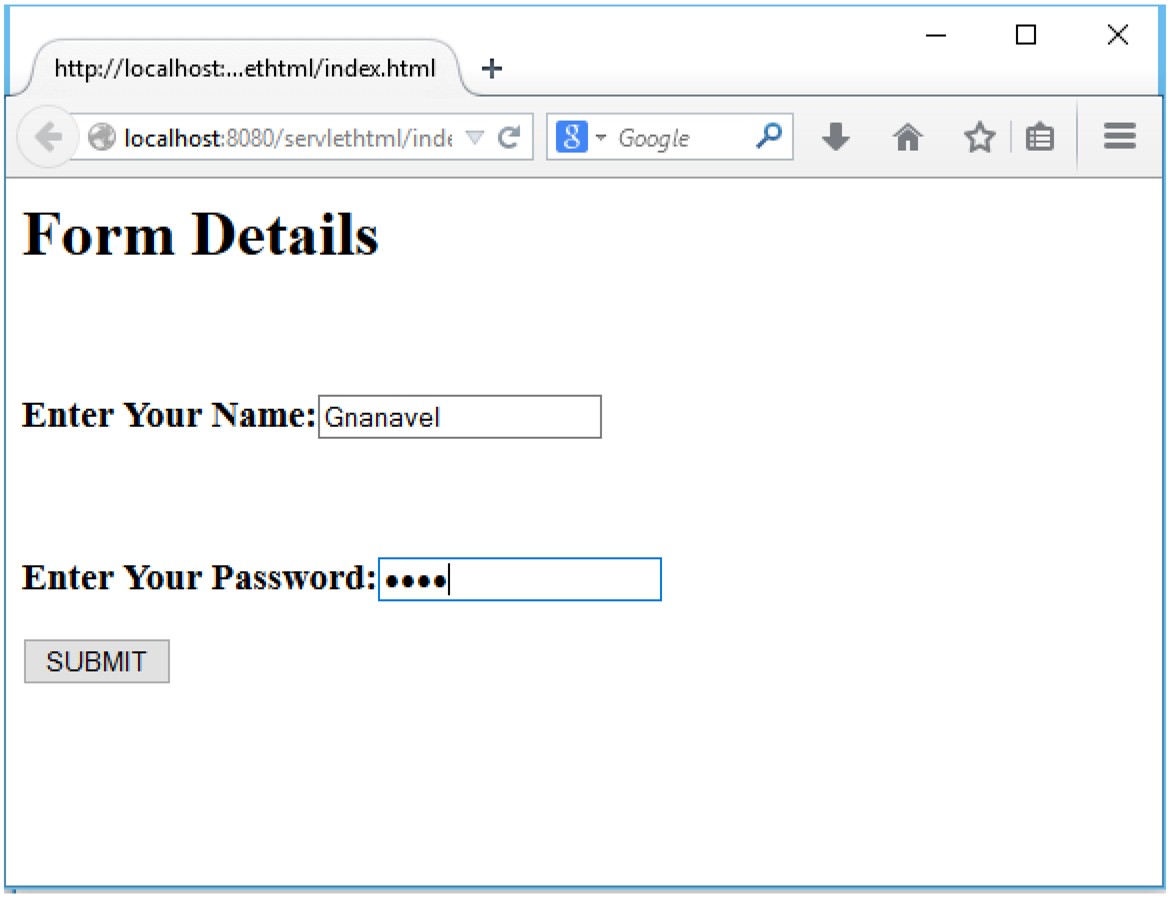
It will show the home page of Apache Tomcat Server.

Step 15: Execute your program as follows:

localhost:8089/project_folder_name/html_file_name.

Example: localhost:8089/india/param.html

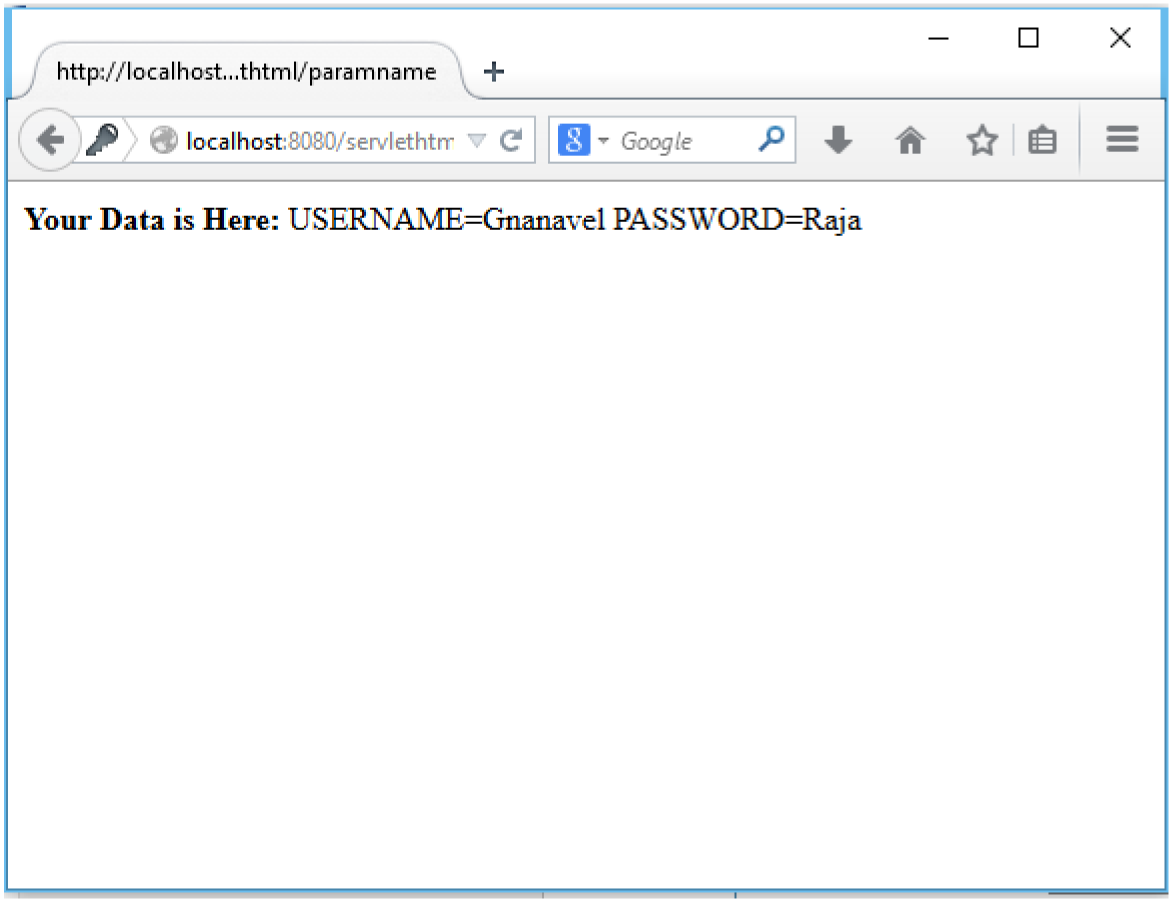
OUTPUT:


http://localhost:...ethtml/index.html
localhost:8080/servlethtml/inde
Google
SUBMIT

Form Details

Enter Your Name: Gnanavel

Enter Your Password: ••••



Your Data is Here: USERNAME=Gnanavel PASSWORD=Raja

RESULT:

Thus the servlet program to invoke servlets from HTML forms is executed successfully.

EX.NO: 4 (ii)	Session Tracking Using Hidden Form Fields
DATE:	

AIM:

To write a java program to implement session tracking in servlets using hidden form fields.

ALGORITHM:

HTML:

Step 1: Start the program.

Step 2: Define form element with attribute name, method and action.

Step 3: call first servlet program in form action attribute.

Step 4: Get username and password from user by using input type as text and password.

Step 5: submit username and password using input type as submit.

Step 6: Stop the program.

SERVLET1:

Step 1: Start the program.

Step 2: Import necessary packages to access the predefined classes and its methods.

Step 3: Define doPost() method by passing request and response object.

Step 4: Create an object for PrintWriter class.

Step 5: Read user name from the form using getParameter() method.

Step 6: Call second servlet program by using hidden as a input type in the form.

Step 7: Stop the program.

SERVLET2:

Step 1: Start the program.

Step 2: Import necessary packages to access the predefined classes and its methods.

Step 3: Define doGet() method by passing request and response object.

Step 4: Create an object for PrintWriter class.

Step 5: Read user name from the hidden form field using getParameter() method.

Step 6: Display the user name using object of PrintWriter class.

Step 7: Stop the program.

PROGRAM:

index.html

```
<html>
<body>
<form method="post" action="First">
Name:<input type="text" name="user" /><br/><br/>
Password:<input type="password" name="pass" ><br/>
<input type="submit" value="submit">
</form></body>
</html>
```

First.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class First extends HttpServlet
{
        protected void doPost(HttpServletRequest request, HttpServletResponse
        response) throws ServletException, IOException
        {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        //getting value submitted in form from HTML file
        String user = request.getParameter("user");
        out.print("Welcome "+user);

        //creating a new hidden form field
        out.print("<form action='Second'>");
        out.print("<input type='hidden' name='uname' value='"+user+"'>");
        out.print("<input type='submit' value='go'>");
        out.print("</form>");
        }
}
```

Second.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class Second extends HttpServlet
{
        protected void doGet(HttpServletRequest request, HttpServletResponse
        response) throws ServletException, IOException
        {
                response.setContentType("text/html");
                PrintWriter out = response.getWriter();

                //getting parameter from the hidden field
                String user = request.getParameter("uname");
                out.println("Welcome "+user);
        }
}
```

web.xml:

```
<web-app>

      <servlet>

      <servlet-name>First</servlet-name>
```

```
    <servlet-class>First</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>First</servlet-name>
    <url-pattern>/First</url-pattern>
  </servlet-mapping>
    <servlet>
    <servlet-name>Second</servlet-name>
    <servlet-class>Second</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Second</servlet-name>
    <url-pattern>/Second</url-pattern>
  </servlet-mapping>
    <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>
  </web-app>
```

EXECUTION:

Step 1: Save html file as index.html, two servlet file as First.java and Second.java and xml file as web.xml

Step 2: Set class path for servlet-api.jar files as follows

i. My Computer→Properties→Advanced System Settings→Environment Variables→User variables→New
ii. Set Variable name as classpath and Variable value as your jar file location.
iii. Example C:\Program Files\Apache Software Foundation\Tomcat 7.0\lib\servlet-api.jar;

iv. Click ok→ok→ok→close.

Step 3: Open cmd prompt and goto your servlet program directory.

Step 4: Set path for javac compiler in cmd prompt as follows

```
set path=" C:\Program Files\Java\jdk1.7.0_79\bin";
```

Step 5: Compile your first servlet program as follows

```
javac  First.java
```

It generates Bytecode file (or Object file) as First.class

Step 6: Compile your first servlet program as follows

```
javac  Second.java
```

It generates Bytecode file (or Object file) as Second.class

Step 6: Goto C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps

Step 7: In the above webapps folder create your own new project folder. For example I created and named it as "hiddenform ".

Step 8: Goto C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\hiddenform.

Step 9: In the above india folder create one folder which is named as WEB-INF(should be capital letters) .

Step 10: Goto C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\hiddenform\WEB-INF

Step 11: In the above WEB-INF folder create one folder which is named as classes (should be small letters).

Step 12: Now place your html, xml and servlet class file in the following directories.

a) Place index.html in C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\hiddenform.

b) Place web.xml in C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\ hiddenform\WEB-INF

c) Place First.class and Second.class in C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\ hiddenform\WEB-INF\classes

Step 13: Start the Apache Tomcat server as follows:

a) Goto C:\Program Files\Apache Software Foundation\Tomcat 7.0\bin.

b) Open Tomcat7w→Click start button.

Step 14: Open your browser and type in URL location as follows:

localhost:8089

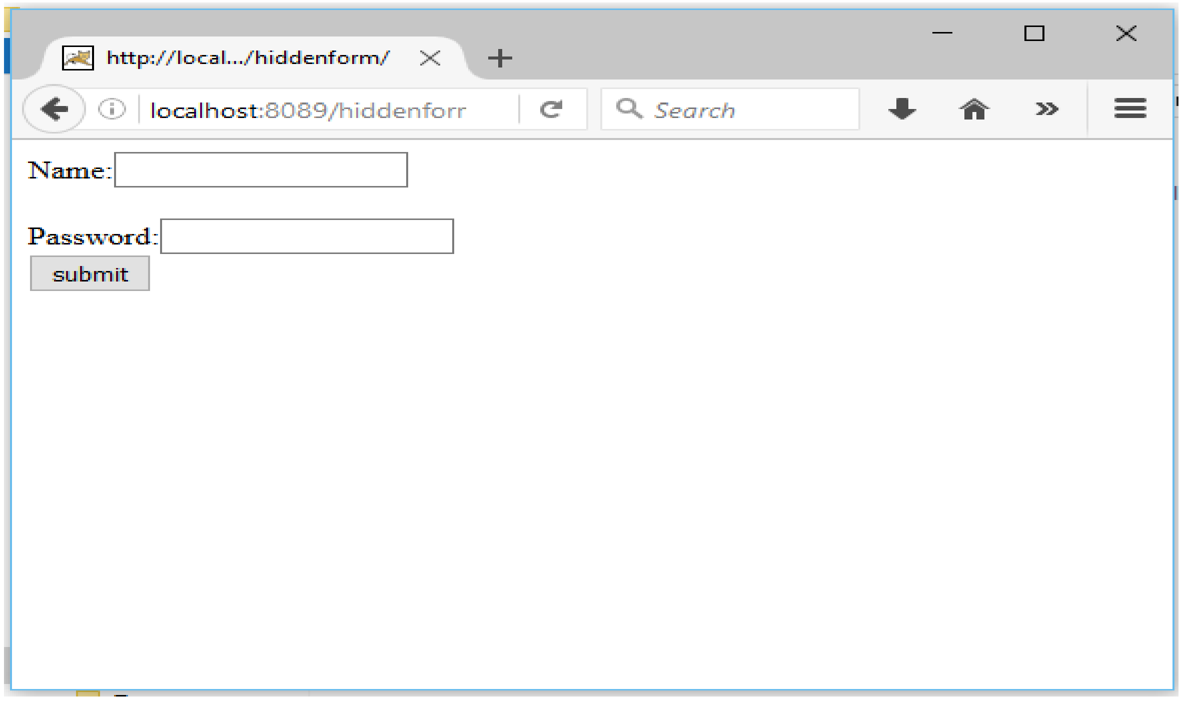
It will show the home page of Apache Tomcat Server.

Step 15: Execute your program as follows:

localhost:8089/project_folder_name/html_file_name.

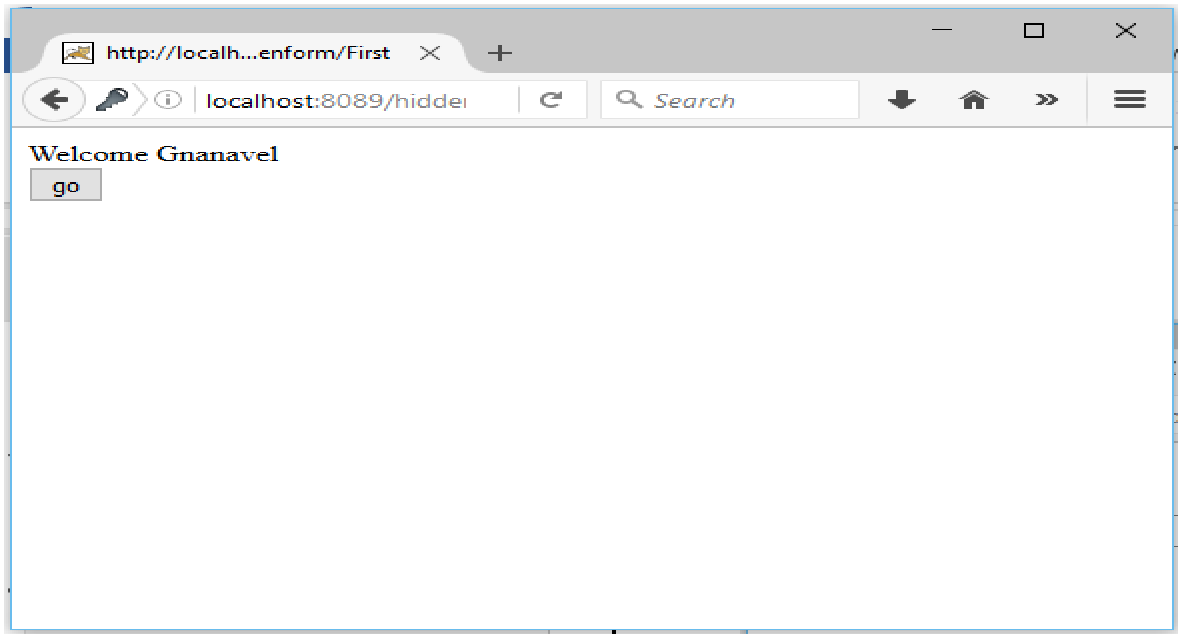
Example: localhost:8089/hiddenform/index.html

OUTPUT:



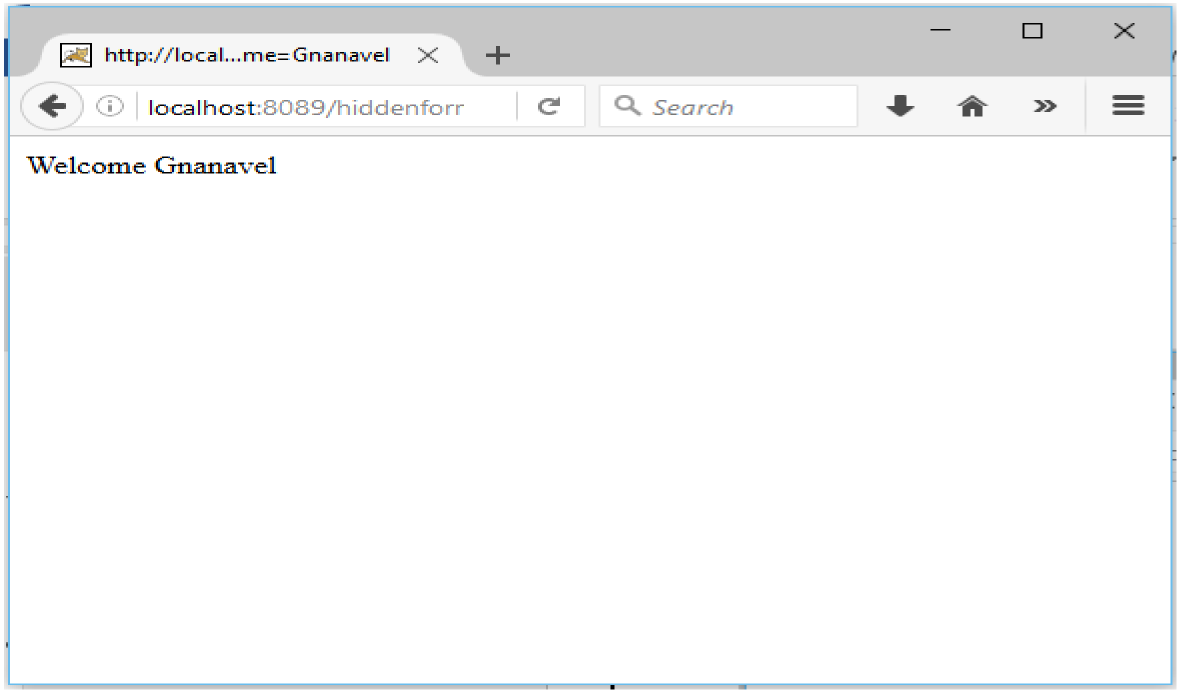
Name:

Password:



Welcome Gnanavel

go



Welcome Gnanavel

RESULT:

Thus the program for session tracking in servlets using hidden form fields was implemented and successfully executed.

EX.NO: 4 (iii)	**Session Tracking for a Hit Count**
DATE:	

AIM:

To write a java program to implement session tracking in servlets for a hit count.

ALGORITHM:

Step 1: Start the program.

Step 2: Import necessary packages to access the predefined classes and its methods.

Step 3: Define a class that extends HttpServlet and also define doGet() method by passing request and response object.

Step 4: Create an object for PrintWriter class.

Step 5: Create session object for HttpSession class.

Step 6: Hit count value is saved in client's session under the name "tracker.count.

Step 7: Display the count value.

Step 8: Stop the program.

PROGRAM:

SessionTracker.java

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class SessionTracker extends HttpServlet
{
        public void doGet(HttpServletRequest req, HttpServletResponse res)
                throws ServletException, IOException
        {
```

```
		res.setContentType("text/html");
		PrintWriter out = res.getWriter();

		// Get the current session object, create one if necessary
		HttpSession session = req.getSession();

		// Increment the hit count for this page. The value is saved
		// in this client's session under the name "tracker.count".
		Integer count = (Integer)session.getAttribute("tracker.count");
		if (count == null)
			count = new Integer(1);
		else
			count = new Integer(count.intValue() + 1);
		session.setAttribute("tracker.count", count);


	out.println("<HTML><HEAD><TITLE>SessionTracker</TITLE></HEAD
>");
		out.println("<BODY><H1>Session Tracking for a Hit Count</H1>");

		// Display the hit count for this page
		out.println("You've visited this page " + count +((count.intValue() ==
1) ? " time." : " times."));
		out.println("<P>");

		out.println("<H2>Here is your session data:</H2>");
		Enumeration e = session.getAttributeNames();
		while (e.hasMoreElements())
		{
			String name = (String) e.nextElement();
			out.println(name + ": " + session.getAttribute(name) +
"<BR>");
		}
		out.println("</BODY></HTML>");
	}
}
```

web.xml:

```
<web-app>
 <servlet>
<servlet-name> SessionTracker </servlet-name>
<servlet-class> SessionTracker </servlet-class>
</servlet>
<servlet-mapping>
<servlet-name> SessionTracker </servlet-name>
<url-pattern>/ SessionTracker </url-pattern>
</servlet-mapping>
</web-app>
```

EXECUTION:

Step 1: Save the servlet file as SessionTracker.java and xml file as web.xml

Step 2: Set class path for servlet-api.jar files as follows

i. My Computer→Properties→Advanced System Settings→Environment Variables→User variables→New
ii. Set Variable name as classpath and Variable value as your jar file location.
iii. Example C:\Program Files\Apache Software Foundation\Tomcat 7.0\lib\servlet-api.jar;
iv. Click ok→ok→ok→close.

Step 3: Open cmd prompt and goto your servlet program directory.

Step 4: Set path for javac compiler in cmd prompt as follows

```
set path=" C:\Program Files\Java\jdk1.7.0_79\bin";
```

Step 5: Compile your servlet program as follows

```
javac  SessionTracker.java
```

It generates Bytecode file (or Object file) as SessionTracker.class

Step 6: Goto C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps

Step 7: In the above webapps folder create your own new project folder. For example I created and named it as "hitcount".

Step 8: Goto C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\hitcount.

Step 9: In the above hitcount folder create one folder which is named as WEB-INF(should be capital letters) .

Step 10: Goto C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\hitcount\WEB-INF

Step 11: In the above WEB-INF folder create one folder which is named as classes (should be small letters).

Step 12: Now place your xml and servlet class file in the following directories.

a) Place web.xml in C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\hitcount\WEB-INF

b) Place SessionTracker.class in C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\hitcount\WEB-INF\classes

Step 13: Start the Apache Tomcat server as follows:

a) Goto C:\Program Files\Apache Software Foundation\Tomcat 7.0\bin.

b) Open Tomcat7w→Click start button.

Step 14: Open your browser and type in URL location as follows:

localhost:8089

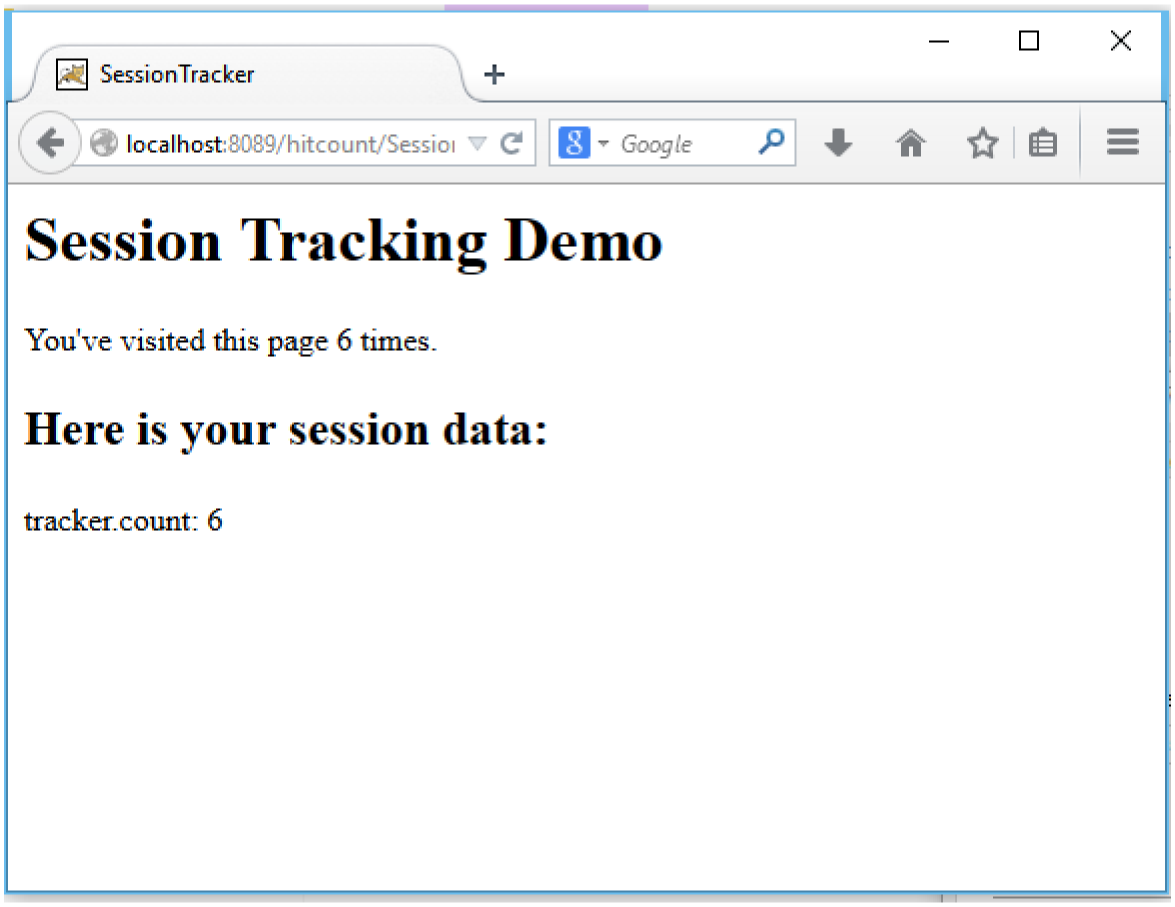
It will show the home page of Apache Tomcat Server.

Step 15: Execute your program as follows:

localhost:8089/project_folder_name/servlet_file_name.

Example: localhost:8089/hitcount/SessionTracker

OUTPUT:



Session Tracking Demo

You've visited this page 6 times.

Here is your session data:

tracker.count: 6

RESULT:

Thus the program for session tracking in Servlets for a hit count is implemented and successfully executed.

EX.NO: 5	Three-tier Application Using Servlets
DATE:	(On-line Examination)

AIM:

To write a program in Java to create three-tier applications using servlets for conducting on-line examination for displaying student mark list.

ALGORITHM:

HTML:

Step 1: Start the program.

Step 2: Using JavaScript validation check whether all the fields in the form are filled else give an alert to fill them.

Step 3: Using the form action attribute call the java program Exam.

Step 4: Call the field validation function on submit of the form.

Step 5: Display several questions and allow the user to choose the answer using the input type as radio buttons.

Step 6: Submit the answers using input type as submit and reset the answers using the input type reset.

SERVLET:

Step 1: Start the program.

Step 2: Import necessary packages to access the predefined classes and its methods.

Step 3: Define doPost() method by passing request and response object.

Step 4: Create an object for PrintWriter class.

Step 5: Connect the form to the database.

Step 6: Read user details from the form using getParameter() method.

Step 7: Evaluate the answers for each question and store the total value in the database along with the user details using insert query.

Step 7: Using select query display the table values.

Step 8: Stop the program.

PROGRAM:

home.html

```
<html>
<head>
<title>Online Examination</title>
<script language="javascript">

function validation(f1)
{
if(f1.seatno.value.length==0)
{
alert("Please,fill up the Seat Number");
f1.seatno.focus();
return false;
}
if(f1.name.value.length==0)
{
alert("Please,fill up the Name");
f1.name.focus();
return false;
}
return true;
}
</script>
</head

<body bgcolor=lightgreen>
<center>
<h1>OnLine Examination</h1>
</center>
<form action="Exam" method=post name="f1" onSubmit="return
validation(this)">
<table>
<tr>
<td><h3>Seat Number:</h3></td>
<td><input type="text" name="seatno"></td>
</tr>
<tr>
<td><h3>Name:</h3></td>
<td><input type="text" name="name" size="50"></td>
</tr>
<br/>
<tr>
<td><b>Total Marks:10(Each question carries equal marks) </b></td>
<td></td><td></td><td></td><td><b>Time: 15 Min.</b></td>
</tr>
</table>
<br/>
<b>1. Apache is an open source web server</b><br/>
```

```
<input type="radio" name="group1" value="True">True
<input type="radio" name="group1" value="False">False
<br/>
<b>2. In Modern PC there is no cache memory.</b><br/>
<input type="radio" name="group2" value="True">True
<input type="radio" name="group2" value="False">False
<br/>
<b>3. Tim-Berner Lee is the originator of Java.</b><br/>
<input type="radio" name="group3" value="True">True
<input type="radio" name="group3" value="False">False
<br/>
<b>4.JPG is not a video file extension.</b><br/>
<input type="radio" name="group4" value="True">True
<input type="radio" name="group4" value="False">False
<br/>
<b>5. HTTP is a statefull protocol</b><br/>
<input type="radio" name="group5" value="True">True
<input type="radio" name="group5" value="False">False
<br/>
<center>
<input type = "submit" value="Submit">
<input type = "reset" value="Clear"><br><br>
</center>
</form>
</body>
</html>
```

Exam.java

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class Exam extends HttpServlet
{
        public void doPost(HttpServletRequest request, HttpServletResponse
        response) throws IOException, ServletException
        {
                PrintWriter out = response.getWriter();
                response.setContentType("text/html");
                int a1=0,a2=0,a3=0,a4=0,a5=0;
                Connection conn=null;
                ResultSet rs=null;
                Statement stmt=null;

                try
                {
                        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                        String url="jdbc:odbc:stu";
                        conn=DriverManager.getConnection(url,"","");
                        String myquery;
                        out.write("Connected with Database....");

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

                        String ans1=request.getParameter("group1");
                        if(request.getParameter("group1")!=null)
                        {
                                if(ans1.equals("True"))
                                        a1=1;
                                else
                                        a1=0;
                        }

                        String ans2=request.getParameter("group2");
```

```
if(request.getParameter("group2")!=null)
{
        if(ans2.equals("True"))
                a2=0;
        else
                a2=1;
}

String ans3=request.getParameter("group3");
if(request.getParameter("group3")!=null)
{
        if(ans3.equals("True"))
                a3=0;
        else
                a3=1;
}

String ans4=request.getParameter("group4");
if(request.getParameter("group4")!=null)
{
        if(ans4.equals("True"))
                a4=1;
        else
                a4=0;
}

String ans5=request.getParameter("group5");
if(request.getParameter("group5")!=null)
{
        if(ans5.equals("True"))
                a5=0;
        else
                a5=1;
}

int tot=a1+a2+a3+a4+a5;
if(seatno!="")
{
stmt=conn.createStatement();
```

```
        myquery="INSERT INTO STUDENTTABLE
    VALUES('"+seatno+"','"+name+"',"+tot+")";
        stmt.executeUpdate(myquery);
        stmt.close();
        }
        stmt=conn.createStatement();
        myquery="select * from StudentTable";
        rs=stmt.executeQuery(myquery);
        out.println("<html><head><title>Online
    Examination</title></head>");
        out.println(" <body bgcolor='pink'> <br/><br/>");
        out.println("<center><h1>Student Database</h1> <br/>");
        out.println("<table border='1' cellspacing='0'
    cellpadding='0'>");
        out.println("<tr><td><b>Seat
    No</b></td><td><b>Name</b></td><td><b>Mark</b></td></tr>");
        while(rs.next())
        {
            out.println("<tr>");
            out.println("<td>"+rs.getString(1)+"</td>");
            out.println("<td>"+rs.getString(2)+"</td>");
            out.println("<td>"+rs.getString(3)+"</td>");
            out.println("</tr>");
        }
        rs.close();
        stmt.close();
        conn.close();
        out.println("</table><br/><br/><h1>Thanks!....</h1></center>
    </body></html>");
    }
    catch (Exception e)
    {
        out.println("error");
    }
  }
}
```

web.xml:

```
<web-app>
 <servlet>
<servlet-name>Exam</servlet-name>
<servlet-class>Exam</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>Exam</servlet-name>
<url-pattern>/Exam</url-pattern>
</servlet-mapping>
<welcome-file-list>
<welcome-file>home.html</welcome-file>
</welcome-file-list>
</web-app>
```

EXECUTION:

Step 1: Open NetBeans IDE 7.2.

Step 2: Goto File→New Project→Choose categories as Java Web→Choose Projects as Web Application→Click Next→Type Project Name→Click Next→Select server as GlassFish Server 3.1.2.1→Click Next→Don't choose any Frameworks→click Finish.

Step 3: To create HTML file→Right click the project→New→HTML→Enter file name alone no need to enter file extension→Click Finish and type your html code here.

Step 4: To create servlet file→Right click the project→New→Servlet→Enter Class Name alone no need to enter file extension→Click Next→Select Add information to deployment descriptor (web.xml)→Click Finish and type your servlet code here.

Steps to Create Database:

Step 5: Open MS ACCESS →Click create table → Right click on Table1 and select Design view →Provide a Table name and Click OK →Now the various fields and datatype names will be asked→Provide them and click OK→Table is created→Save the access file with the database name.

Step 6: Goto Control panel →Administrative tools →Click ODBC Data Sources(32-bit) →Click Add and select the Microsoft Access driver(*.mdb,*.accdb) →Click Finish→An ODBC Microsoft Access Setup screen will be opened.

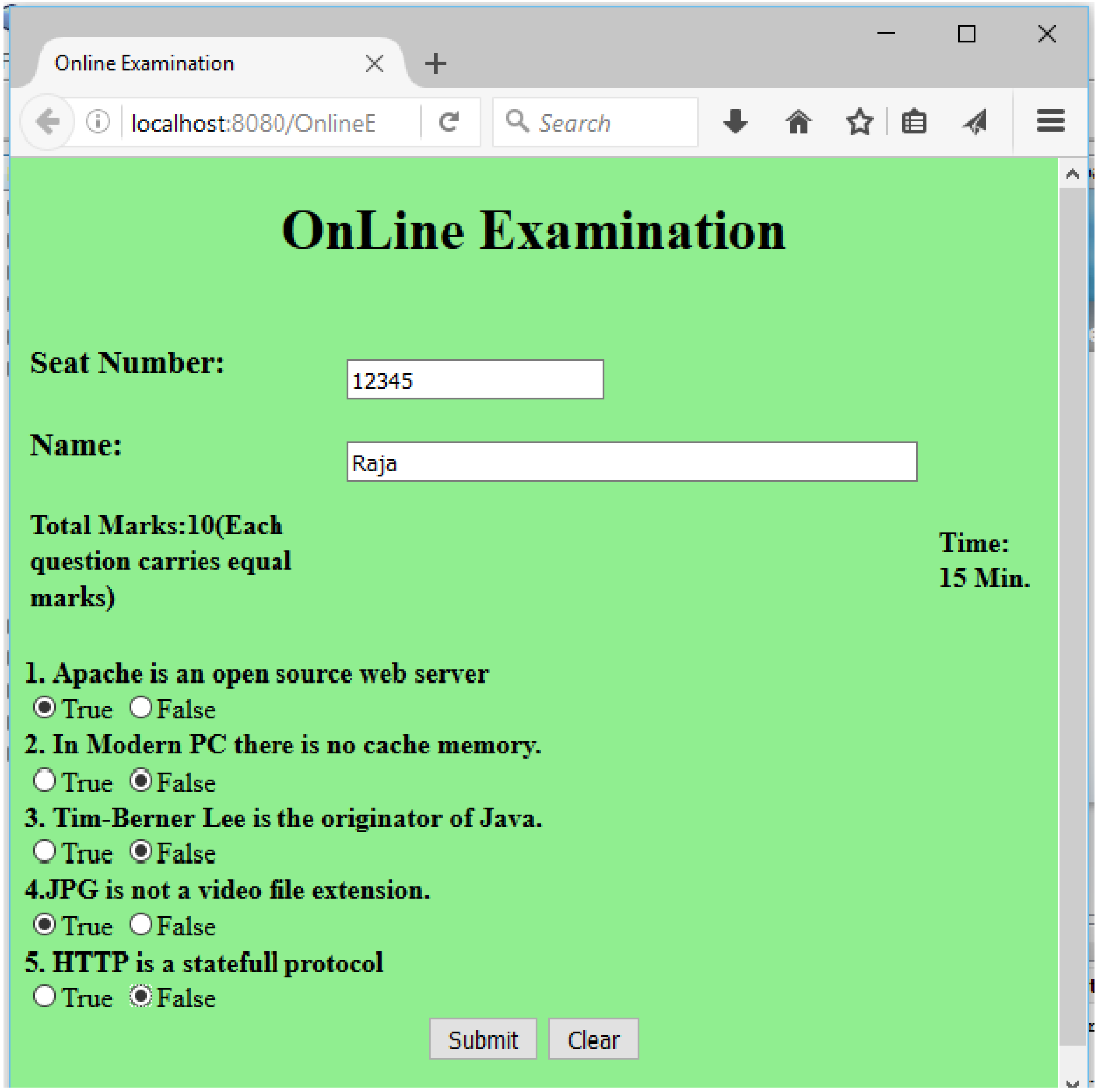
Step 7: Now in the ODBC Microsoft Access Setup screen provide a Data Source Name then under database Click Select→Select the directory in which the database is stored→Then the database in that directory will be displayed ,Click the database→select OK and finish the procedure.

Step 8: Now run the html file in NetBeans IDE 7.2.

Step 9: Fill the details in the form and click submit.

Step 10: Now open the database, the corresponding details will be stored in the database.

OUTPUT:



OnLine Examination

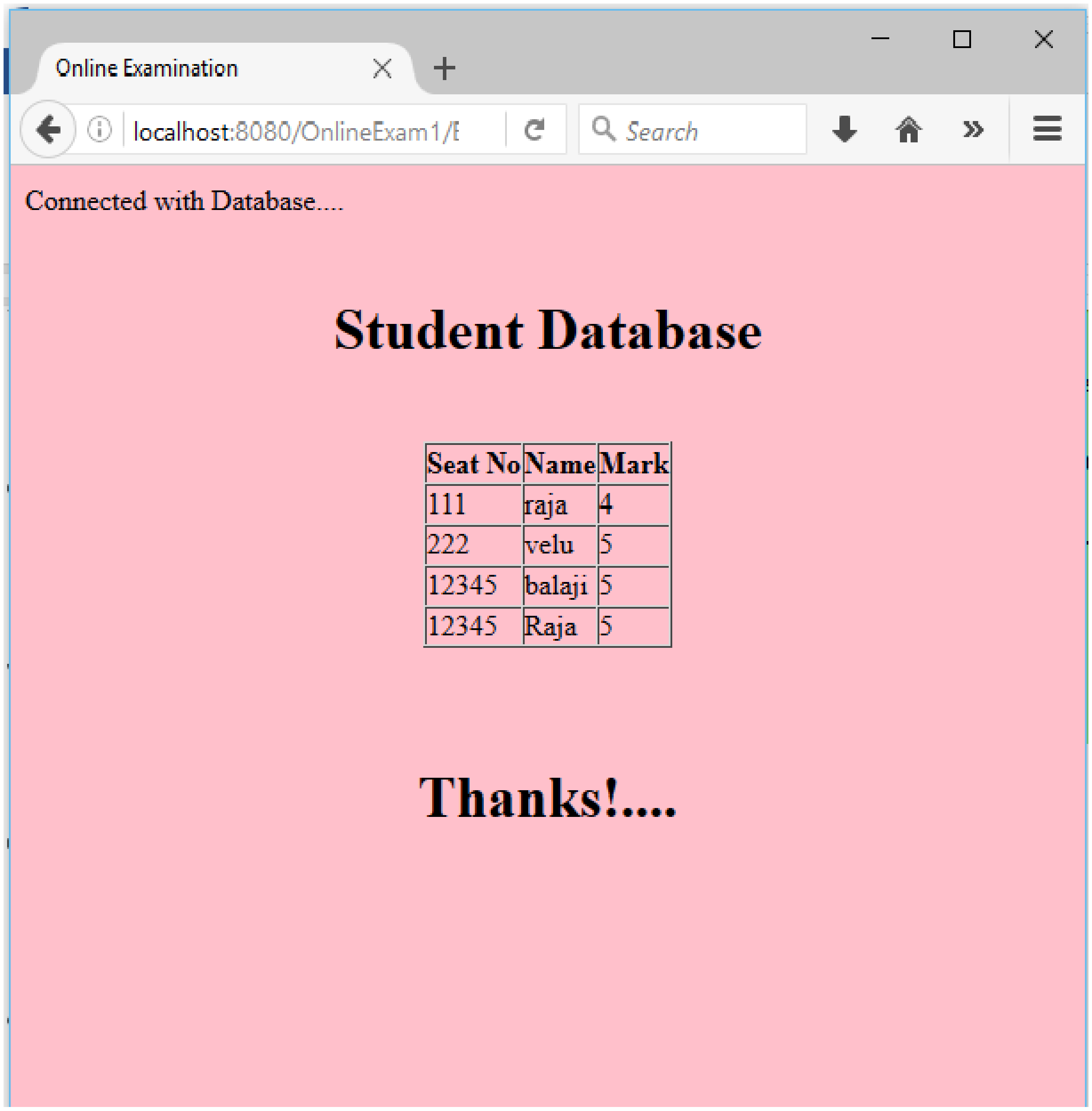
Seat Number: 12345

Name: Raja

Total Marks:10(Each question carries equal marks)

Time: 15 Min.

1. Apache is an open source web server
[✓ True] False
2. In Modern PC there is no cache memory.
True [✓ False]
3. Tim-Berner Lee is the originator of Java.
True [✓ False]
4.JPG is not a video file extension.
[✓ True] False
5. HTTP is a statefull protocol
True [✓ False]

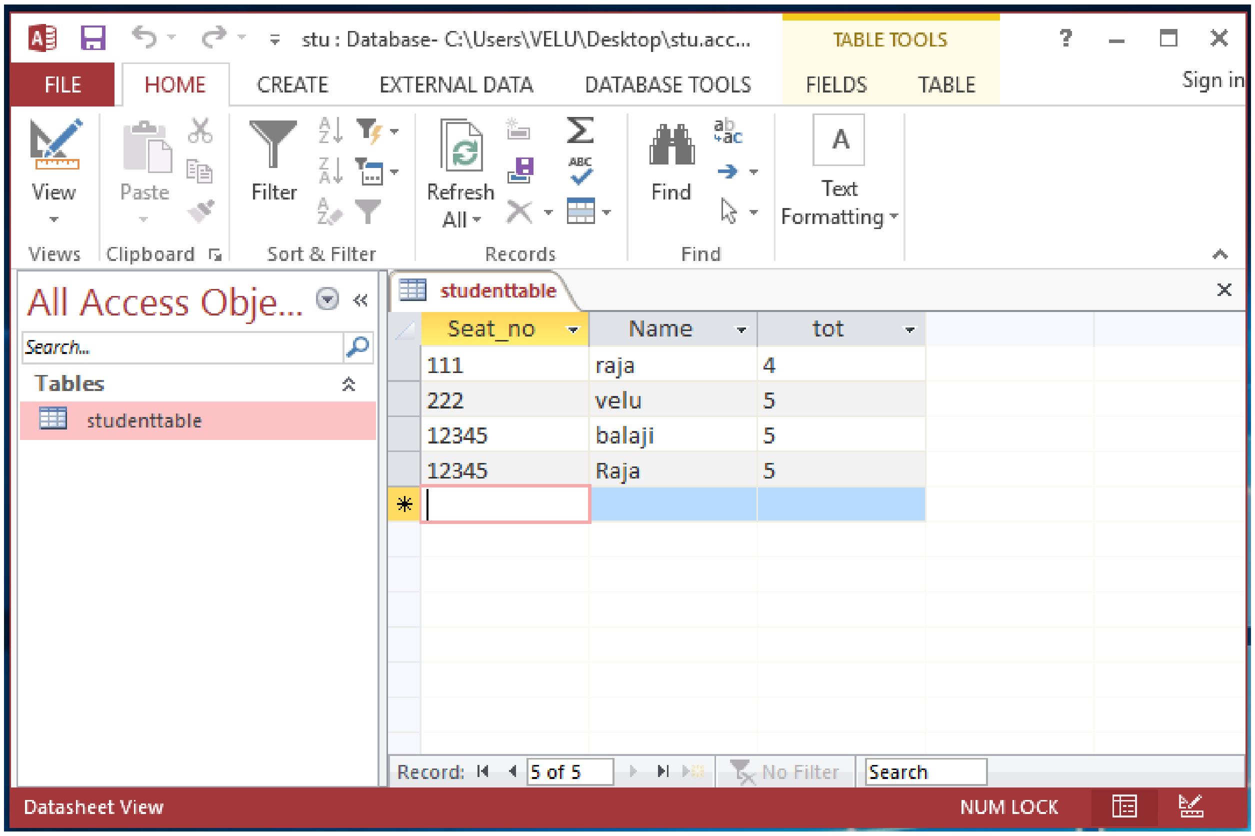


Connected with Database....

Student Database

Seat No	Name	Mark
111	raja	4
222	velu	5
12345	balaji	5
12345	Raja	5

Thanks!....



studenttable

Seat_no	Name	tot
111	raja	4
222	velu	5
12345	balaji	5
12345	Raja	5

RESULT:

Thus the program in Java to create three-tier applications using servlets for conducting on-line examination for displaying student mark list is executed successfully.

EX.NO: 6 (i)	**INSTALLING & CONFIGURING TOMCAT WEBSERVER**
DATE:	

AIM:

To Install & Configure TOMCAT web server.

PROCEDURE:

Step 1: Visit Apache Tomcat home page with a Web browser, and click the "Download" link under the "Tomcat 7.0.70 Released" section. You will see the "Tomcat 7 Downloads" page.

Step 2: Click "32-bit Windows zip" link under "Binary Distributions" section. You will see the download file save dialog box.(this supports jdk1.6 kit)

Step 3: Use the "Save file" option to save the download file "apache-tomcat-7.0.70-windows-x86.zip" to a temporary folder.

Step 4: Unzip "apache-tomcat-7.0.70-windows-x86.zip" to file installation folder “C:\ apache-tomcat-7.0.70”.

Step 5: Try to start Tomcat server by running the "startup" command in a command line window:

```
C:\ >cd apache-tomcat-7.0.70\bin
C:\ >cd apache-tomcat-7.0.70\bin>startup
The CATALINA_HOME environment variable is not defined correctly
This environment variable is needed to run this program
```

Step 6: To fix the missing environment variables, CATALINA_HOME,JAVA_HOME & JRE_HOME,

- Click my computer->right click properties->Select Advance System Setting
- In this tab, Click Environment variable

- Click new in User variables for admin
- Enter variable name & Variable value

CATALINA_HOME	C:\apache-tomcat-7.0.70
JAVA_HOME	C:\Program Files\Java\
JRE_HOME	C:\Program Files\Java\jre7

Step 7: Configuring Tomcat

Open “C:\apache-tomcat-7.0.70\conf” folder

It consiste of the following xml files

Server.xml

Web.xml

Tomcat-user.xml

Context.xml

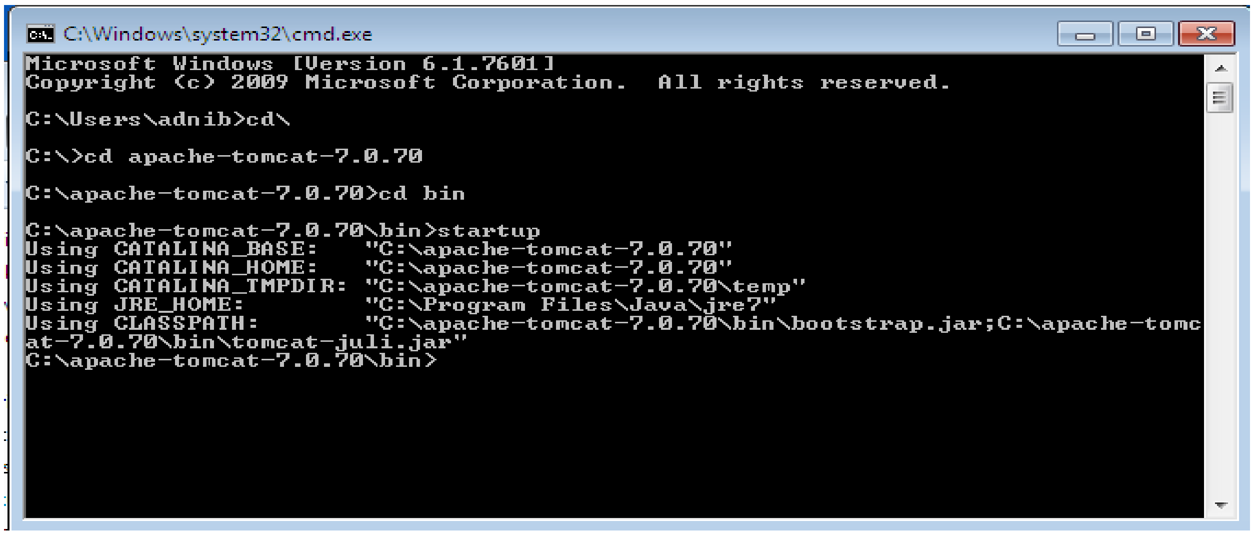
Open web.xml in notepad & modify default-listings-false to true

Open server.xml & change port number 8080 to anyother(8081)if any application access the same port

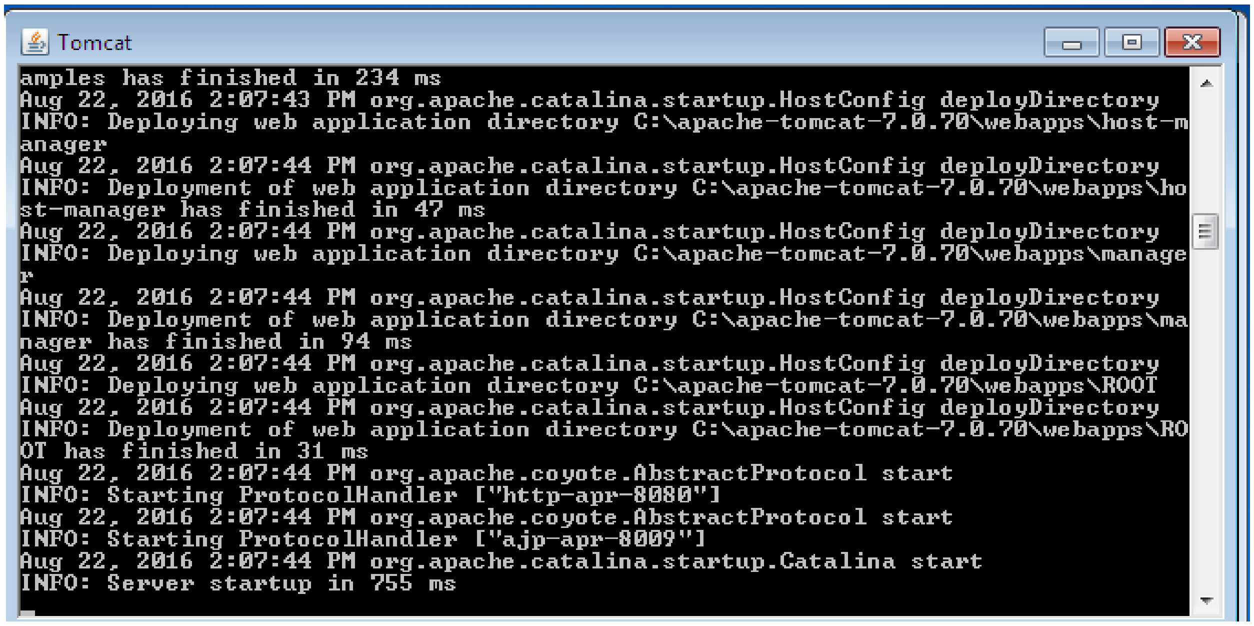
Open context.xml change reloadable attribute to true

Configuration procedure varies according to the version

Step 8: Try to start Tomcat server by running the "startup" command in a command line window:



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\adnib>cd\

C:\>cd apache-tomcat-7.0.70

C:\apache-tomcat-7.0.70>cd bin

C:\apache-tomcat-7.0.70\bin>startup
Using CATALINA_BASE:   "C:\apache-tomcat-7.0.70"
Using CATALINA_HOME:   "C:\apache-tomcat-7.0.70"
Using CATALINA_TMPDIR: "C:\apache-tomcat-7.0.70\temp"
Using JRE_HOME:        "C:\Program Files\Java\jre7"
Using CLASSPATH:       "C:\apache-tomcat-7.0.70\bin\bootstrap.jar;C:\apache-tomc
at-7.0.70\bin\tomcat-juli.jar"
C:\apache-tomcat-7.0.70\bin>
```



```
amples has finished in 234 ms
Aug 22, 2016 2:07:43 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory C:\apache-tomcat-7.0.70\webapps\host-m
anager
Aug 22, 2016 2:07:44 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deployment of web application directory C:\apache-tomcat-7.0.70\webapps\ho
st-manager has finished in 47 ms
Aug 22, 2016 2:07:44 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory C:\apache-tomcat-7.0.70\webapps\manage
r
Aug 22, 2016 2:07:44 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deployment of web application directory C:\apache-tomcat-7.0.70\webapps\ma
nager has finished in 94 ms
Aug 22, 2016 2:07:44 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory C:\apache-tomcat-7.0.70\webapps\ROOT
Aug 22, 2016 2:07:44 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deployment of web application directory C:\apache-tomcat-7.0.70\webapps\RO
OT has finished in 31 ms
Aug 22, 2016 2:07:44 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-apr-8080"]
Aug 22, 2016 2:07:44 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-apr-8009"]
Aug 22, 2016 2:07:44 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 755 ms
```

RESULT:

Thus the Tomcat server is installed and configured successfully.

EX.NO: 6 (ii)	**Convert static web pages into dynamic web pages using servlets and cookies**
DATE:	

AIM:

To convert the static web pages into dynamic web pages using servlets (or JSP) and cookies.

PROCEDURE:

Step-1: we will create a html form for entering the user name, password and card ID.
Step-2: From the above HTML form, the servlet program is invoked in which the validity of the user name, password and card id is checked. if it is a valid user then the welcome message will be displayed otherwise the “invalid user” message will be displayed. In this servlet we set the cookies in which the current user name is stored.
Step-3: compile the above servlet Login servlet.java and copy its class file in tomcats folder at c:\tomcatdirectory\webapps\examples\WEB-INF\classes. Then edit the web.xml in WEB-INF folder. We must store he user information such as user name, password and card id in the web.xml using init-param.
Step-4: On successful login , the information from the cookie is checked and shopping cart page for corresponding user can be displayed.
Step-5: Compile the above servlet LoginSuccess.java and copy its class file in the tomcat's folder at c:\tomcatdirectory\webapps\examples\WEB-INF\classes. Then edit the web.xml in WEB-INF folder.
Step-6: Start tomcat web server. Open the web browser and display the login form created in step1.

PROGRAM:

Index.jsp

```
<html>
    <head>
    <body>
        <form action="http://localhost:8084/ddd/LoginServlet" method="post">
            Enter username:
            <input type="text" value""name="user">
            <br>
             Enter Password:
            <input type="password" value""name="password">
            <br>
            Enter Card ID:
            <input type="text" value""name="cardID">
            <br>
              <br>    <br>    <br>
              <input type="submit" value="login">
        </form>
    </body>
```

Loginservlet.html

```
import java.io.*;
import java.net.*;

import javax.servlet.*;
import javax.servlet.http.*;
public class LoginServlet extends HttpServlet {
 protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {

            String usr=request.getParameter("user");
            String pwd=request.getParameter("password");
            String card=request.getParameter("cardID");
            boolean flag=true;

            String[] userID=getInitParameter("usernames").split(",");
            String[] password=getInitParameter("passwords").split(",");
            String[] cardids=getInitParameter("cardIDs").split(",");

            int i;
            for(i=0;i<userID.length;i++)
            {
      
if(userID[i].equals(usr)&&password[i].equals(pwd)&&cardids[i].equals(card))
                {
                    flag=false;
                    Cookie MyCookie=new Cookie("CurrentUser", usr);
                    MyCookie.setMaxAge(60*60);
                    response.addCookie(MyCookie);
                    response.sendRedirect("http://localhost:8084/ddd/LoginSuccess");
                  }
            }
                        if(flag==true)
            {
                out.print("Error");
                out.println("<h4>Invalid user,please try again by clicking following
link</h4>");
                out.println("<a href='http://localhost:8084/ddd/'>"+"LoginForm.html");
            }
        }
finally {
            out.close();
        }
    }
```

LoginSuccess.java

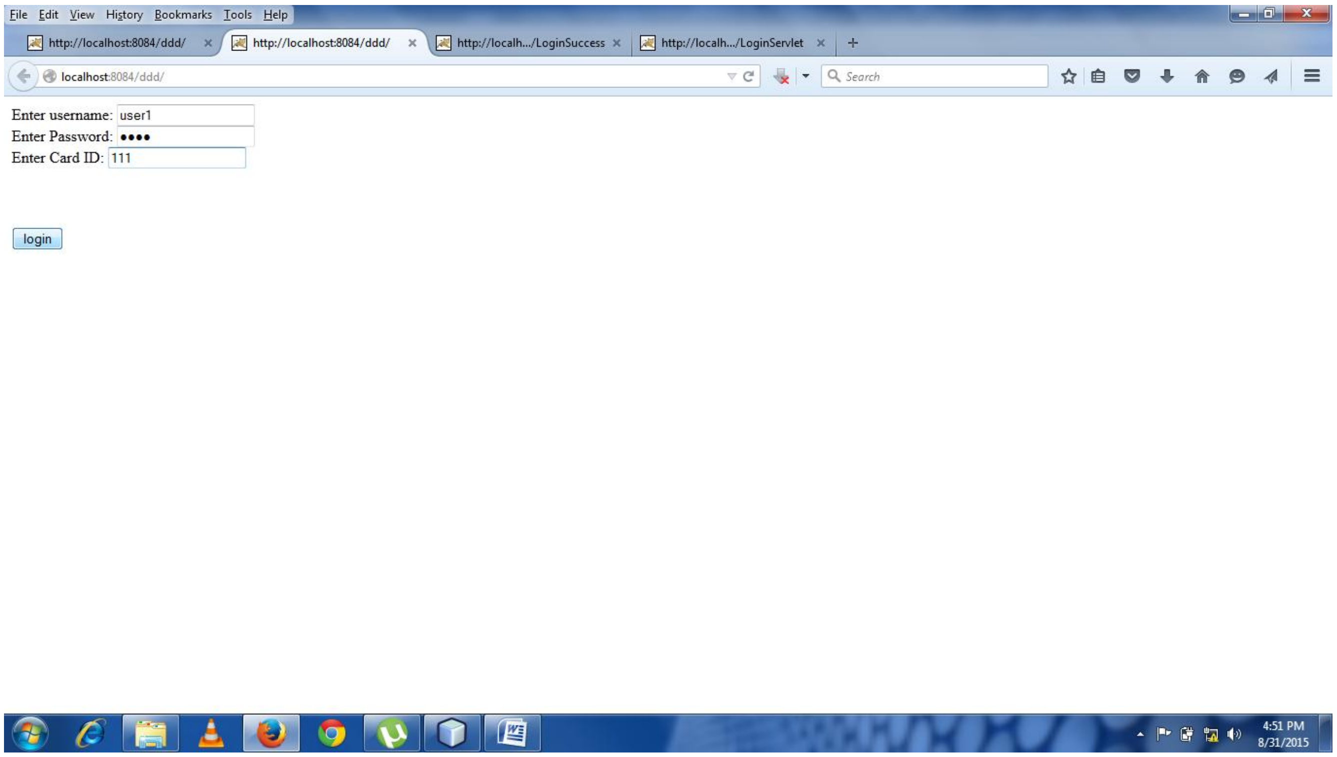
```
import java.io.*;
import java.net.*;

import javax.servlet.*;
import javax.servlet.http.*;
public class LoginSuccess extends HttpServlet {protected void
doGet(HttpServletRequest request, HttpServletResponse response)
   throws ServletException, IOException {
  Cookie[] my_cookies=request.getCookies();
  response.setContentType("text/html");
  PrintWriter out=response.getWriter();
  out.print("Login Success");
  out.println("<b>");
  String userName=null;
  if(my_cookies!=null)
  {
     for(Cookie cookie:my_cookies)
     {
        if(cookie.getName().equals("currentUser"))
           userName=cookie.getValue();
     }
  }
  out.print("<h3>Login Success!!!Welcome</h3>");
  out.print("<h2>This is a Shopping cart for"+userName+"</h2>");
  out.close();

     }
  }
```

Web.xml

```
   <servlet>
   <servlet-name>LoginServlet</servlet-name>
   <servlet-class>LoginServlet</servlet-class>
   <init-param>
         <param-name>usernames</param-name>
```

```
        <param-value>user1,user2,user3</param-value>
      </init-param>
    <init-param>
        <param-name>passwords</param-name>
        <param-value>pwd1,pwd2,pwd3</param-value>
        </init-param>
    <init-param>
        <param-name>cardIDs</param-name>
        <param-value>111,222,333</param-value>
    </init-param>
</servlet>
<servlet>
    <servlet-name>LoginSuccess</servlet-name>
    <servlet-class>LoginSuccess</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>LoginServlet</servlet-name>
    <url-pattern>/LoginServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>LoginSuccess</servlet-name>
    <url-pattern>/LoginSuccess</url-pattern>
</servlet-mapping>
<session-config>
    <session-timeout>
        30
    </session-timeout>
</session-config>
<welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
    </welcome-file-list>
</web-app>
```

Output:


File Edit View History Bookmarks Tools Help

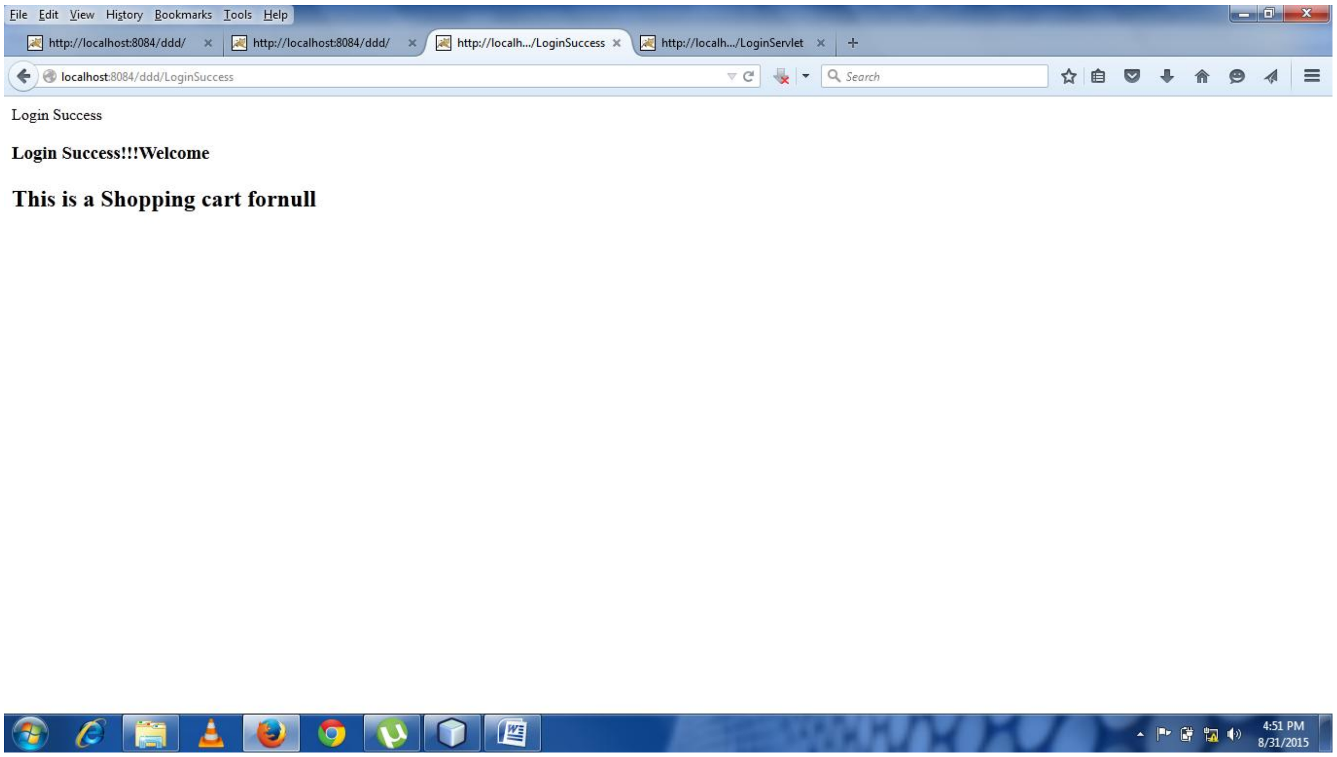
http://localhost:8084/ddd/ | http://localhost:8084/ddd/ | http://localh.../LoginSuccess | http://localh.../LoginServlet

localhost:8084/ddd/

Enter username: user1
Enter Password: ●●●●
Enter Card ID: 111

login

4:51 PM 8/31/2015

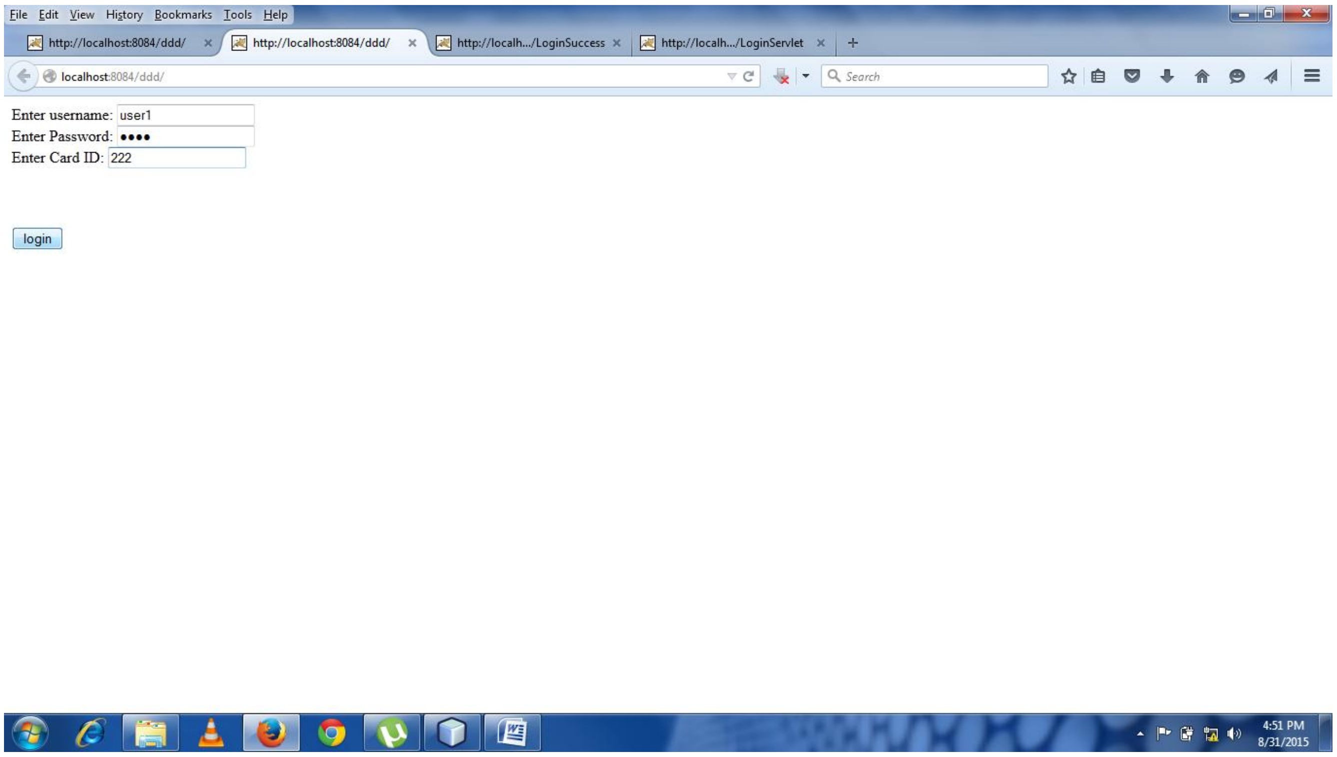

File Edit View History Bookmarks Tools Help

localhost:8084/ddd/LoginSuccess

Login Success

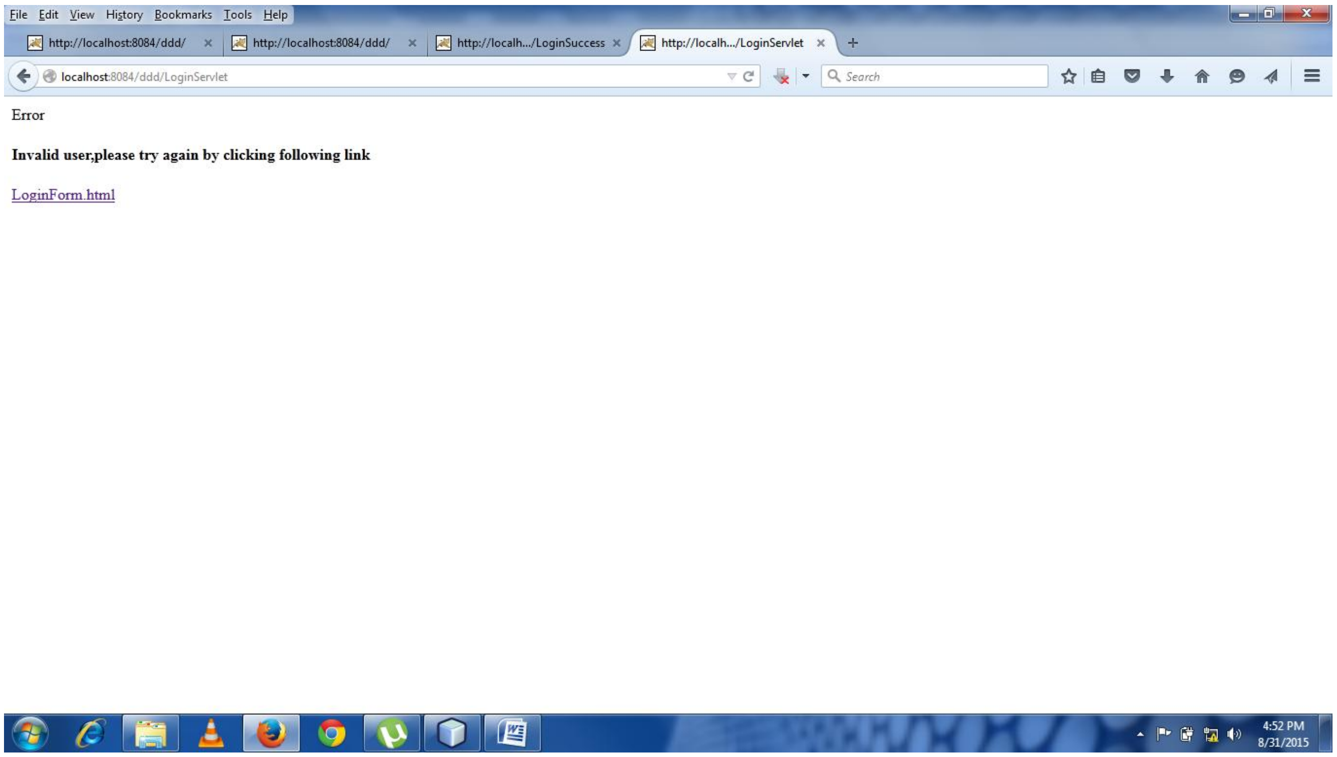
Login Success!!!Welcome

This is a Shopping cart fornull


File Edit View History Bookmarks Tools Help

localhost:8084/ddd/

Enter username: user1
Enter Password: ••••
Enter Card ID: 222


File Edit View History Bookmarks Tools Help
http://localhost:8084/ddd/
http://localhost:8084/ddd/
http://localh.../LoginSuccess
http://localh.../LoginServlet
localhost:8084/ddd/LoginServlet
Search
4:52 PM
8/31/2015

```
Error

Invalid user,please try again by clicking following link

LoginForm.html
```

RESULT:

Thus the conversion of the static web pages into dynamic web pages using servlets and cookies has been executed successfully.

EX.NO: 7	**Convert static web pages into dynamic web pages using JSP** **(On-line Shopping)**
DATE:	

AIM:

To write an online book shopping application using JSP with database connectivity.

PROCEDURE:

Step 1: Create home, login, registration, profile, catalog and order html pages.

Step 2: Create jsp pages which does all business works on the server.

Step 3: Use appropriate database to store the details of the books.

Step 4: Create tables to store login details and books details.

Step 5: Connect the database using odbc.jdbc driver.

Step 6: Make changes in the control settings to enable database on your local machine.

PROGRAM:

main.html:

```
<html>
<body bgcolor="pink">
<br /><br /><br /><br /><br />
<h1 align="center"><U>ONLINE BOOK STORAGE</U></h1><br /><br /><br
/>
<h2 align="center"><pre>
<b>Welcome to online book storage.
Press LOGIN if you are having id
```

```
otherwise press REGISTRATION
</b></pre></h2>
<br /><br /><pre>
<div align="center"><a href="login.html">LOGIN</a> <a href="reg.html">
REGISTRATION</a></div></pre>
</body>
</html>
```

login.html:

```
<html>
<body bgcolor="pink"><br /><br /><br />
<form name="myform" method="post" action="login.jsp">
<div align="center"><pre>
LOGIN ID :<input type="text" name="id" /><br />
PASSWORD :<input type="password" name="pwd" /></pre><br /><br />
</div>
<br /><br />
<div align="center">
<input type="submit" value="ok"/>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset" value="clear" />
</div>
</form>
</body>
</html>
```

Reg.html:

```
<html>
<body bgcolor="pink"><br /><br />
<form name="myform" method="post" action="reg.jsp">
```

```
<div align="center"><pre>
NAME :<input type="text" name="name" /><br />
ADDRESS :<input type="text" name="addr" /><br />
CONTACT NUMBER :<input type="text" name="phno" /><br />
LOGIN ID :<input type="text" name="id" /><br />
PASSWORD :<input type="password" name="pwd" /></pre><br /><br />
</div>
<br /><br />
<div align="center">
<input type="submit" value="ok">
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset" value="clear" />
</div>
</form>
</body>
</html>
```

Profile.html:

```
<html>
<body bgcolor="pink"><br /><br /><br />
<form name="myform" method="post" action="profile.jsp">
<div align="center"><pre>
LOGIN ID :<input type="text" name="id" /><br />
</pre><br /><br />
</div>
<br /><br />
<div align="center">
<input type="submit" value="ok"/>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset" value="clear" />
```

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

catalog.html:

```
<html>
<body bgcolor="pink"><br /><br /><br />
<form method="post" action="catalog.jsp">
<div align="center"><pre>
BOOK TITLE :<input type="text" name="title" /><br />
</pre><br /><br />
</div>
<br /><br />
<div align="center">
<input type="submit" value="ok"
name="button1"/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<input type="reset" value="clear" name="button2"/>
</div>
</form>
</body></html>
```

order.html:

```
<html>
<body bgcolor="pink"><br /><br />
<form method="post" action="order.jsp">
<div align="center"><pre>
NAME :<input type="text" name="name" /><br />
PASSWORD :<input type="password" name="pwd" />
TITLE :<input type="text" name="title" /><br />
NO. OF BOOKS :<input type="text" name="no" /><br />
DATE :<input type="text" name="date" /><br />
```

```
CREDIT CARD NUMBER:<input type="password" name="cno" /><br
/></pre><br /><br />
</div>
<br /><br />
<div align="center">
<input type="submit" value="ok" name="button1"/>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset" value="clear"
name="button2"/>
</div>
</form>
</body>
</html>
```

login.jsp:

```
<%@page import="java.sql.*"%>
<%@page import="java.io.*"%>
<%
out.println("<html><body bgcolor='pink'>");
String id=request.getParameter("id");
String pwd=request.getParameter("pwd");
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con=DriverManager.getConnection ("jdbc:odbc:stu","","");
Statement stmt=con.createStatement();
String sqlstmt="select id,pwd from login where id='"+id+"' and pwd='"+pwd+"'";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
while(rs.next())
{
```

```
flag=1;
}
if(flag==0)
{
out.println("SORRY INVALID ID TRY AGAIN ID<br><br>");
out.println("<a href='login.html'>press LOGIN to RETRY</a>");
}
else
{
out.println("VALID LOGIN ID<br><br>");
out.println("<h3><ul>");
out.println("<li><a href='profile.html'><fontcolor='black'>USER
PROFILE</font></a></li><br><br>");
out.println("<li><a href='catalog.html'><fontcolor='black'>BOOKS
CATALOG</font></a></li><br><br>");
out.println("<li><a href='order.html'><fontcolor='black'>ORDER
CONFIRMATION</font></a></li><br><br>");
out.println("</ul>");
}
out.println("<body></html>");
%>
```

reg.jsp:

```
<%@page import="java.sql.*"%>
<%@page import="java.io.*"%>
<%out.println("<html><body bgcolor='pink'>");
String name=request.getParameter("name");
String addr=request.getParameter("addr");
String phno=request.getParameter("phno");
String id=request.getParameter("id");
String pwd=request.getParameter("pwd");
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con=DriverManager.getConnection ("jdbc:odbc:stu","","");
Statement stmt=con.createStatement();
String myquery="select id from login";
ResultSet rs=stmt.executeQuery(myquery);
int flag=0;
while(rs.next())
{if(id.equals(rs.getString(1)))
{flag=1;
}}if(flag==1)
{out.println("SORRY LOGIN ID ALREADY EXISTS TRY AGAIN WITH NEW
ID <br><br>");
out.println("<a href='reg.html'>press REGISTER to RETRY</a>");
}else
{Statement stmt1=con.createStatement ();
stmt1.executeUpdate ("insert into login values
('"+name+"','"+addr+"','"+phno+"','"+id+"','"+pwd+"')");
out.println ("YOU DETAILS ARE ENTERED <br><br>");
out.println ("<a href ='login.html'>press LOGIN to login</a>");}
out.println ("</body></html>");
%>
```

profile.jsp:

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

```
<%
out.println ("<html><body bgcolor='pink'>");
String id=request.getParameter("id");
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con=DriverManager.getConnection ("jdbc:odbc:stu","","");
Statement stmt=con.createStatement ();
String sqlstmt="select * from login where id='"+id+"'";
ResultSet rs=stmt.executeQuery (sqlstmt);
int flag=0;
while(rs.next())
{
out.println ("<div align='center'>");
out.println ("NAME :"+rs.getString(1)+"<br>");
out.println ("ADDRESS :"+rs.getString(2)+"<br>");
out.println ("PHONE NO :"+rs.getString(3)+"<br>");
out.println ("</div>");
flag=1;
}
if(flag==0)
{
out.println("SORRY INVALID ID TRY AGAIN ID <br><br>");
out.println("<a href='profile.html'>press HERE to RETRY </a>");
}
out.println ("</body></html>");
%>
```

catalog.jsp:

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

```
<%@page import="java.io.*"%>
<%out.println ("<html><body bgcolor='pink'>");
String title=request.getParameter ("title");
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con=DriverManager.getConnection ("jdbc:odbc:stu","","");
Statement stmt=con.createStatement ();
String sqlstmt="select * from book where title='"+title+"'";
ResultSet rs=stmt.executeQuery (sqlstmt);
int flag=0;
while(rs.next())
{
out.println ("<div align='center'>");
out.println ("TITLE :"+rs.getString(1)+"<br>");
out.println ("AUTHOR :"+rs.getString(2)+"<br>");
out.println ("VERSION:"+rs.getString(3)+"<br>");
out.println ("PUBLISHER :" +rs.getString(4)+"<br>");
out.println ("COST :" +rs.getString(5)+"<br>");
out.println ("</div>");
flag=1;
}
if(flag==0)
{
out.println("SORRY INVALID ID TRY AGAIN ID <br><br>");
out.println("<a href='catalog.html'>press HERE to RETRY </a>");
}
out.println ("</body></html>");
%>
```

order.jsp:

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

<%@page import="java.io.*"%>

<%out.println ("<html><body bgcolor='pink'>");

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

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

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

String count1=request.getParameter ("no");

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

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

int count=Integer.parseInt(count1);

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con=DriverManager.getConnection ("jdbc:odbc:stu","","");

Statement stmt=con.createStatement ();

String sqlstmt="select id,password from login";

ResultSet rs=stmt.executeQuery (sqlstmt);

int flag=0,amount,x;

while(rs.next())

{
```

```
if(id.equals(rs.getString(1))&& pwd.equals(rs.getString(2)))

{

flag=1;

}}

if(flag==0)

{

out.println("SORRY INVALID ID TRY AGAIN ID <br><br>");

out.println("<a href='order.html'>press HERE to RETRY </a>");

}

else

{

Statement stmt2=con.createStatement();

String s="select cost from book where title='"+title+"'";

ResultSet rs1=stmt2.executeQuery(s);

int flag1=0;

while(rs1.next())

{

flag1=1;

x=Integer.parseInt(rs1.getString(3));
amount=count*x;

out.println("AMOUNT :"+amount+"<br><br><br><br>");

Statement stmt1=con.createStatement ();

stmt1.executeUpdate ("insert into details
('"+id+"','"+title+"','"+amount+"','"+date+"','"+cno+"')");

out.println ("YOU ORDER HAS TAKEN<br>");

}

if(flag1==0)

{

out.println("SORRY INVALID BOOK TRY AGAIN <br><br>");

out.println("<a href='order.html'>press HERE to RETRY </a>");

}

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

%>
```

EXECUTION:

Step 1: Go to MS-access->select blank database.

Step 2: Give the database name and click create button.

Step 3: Select create table and design the required table.

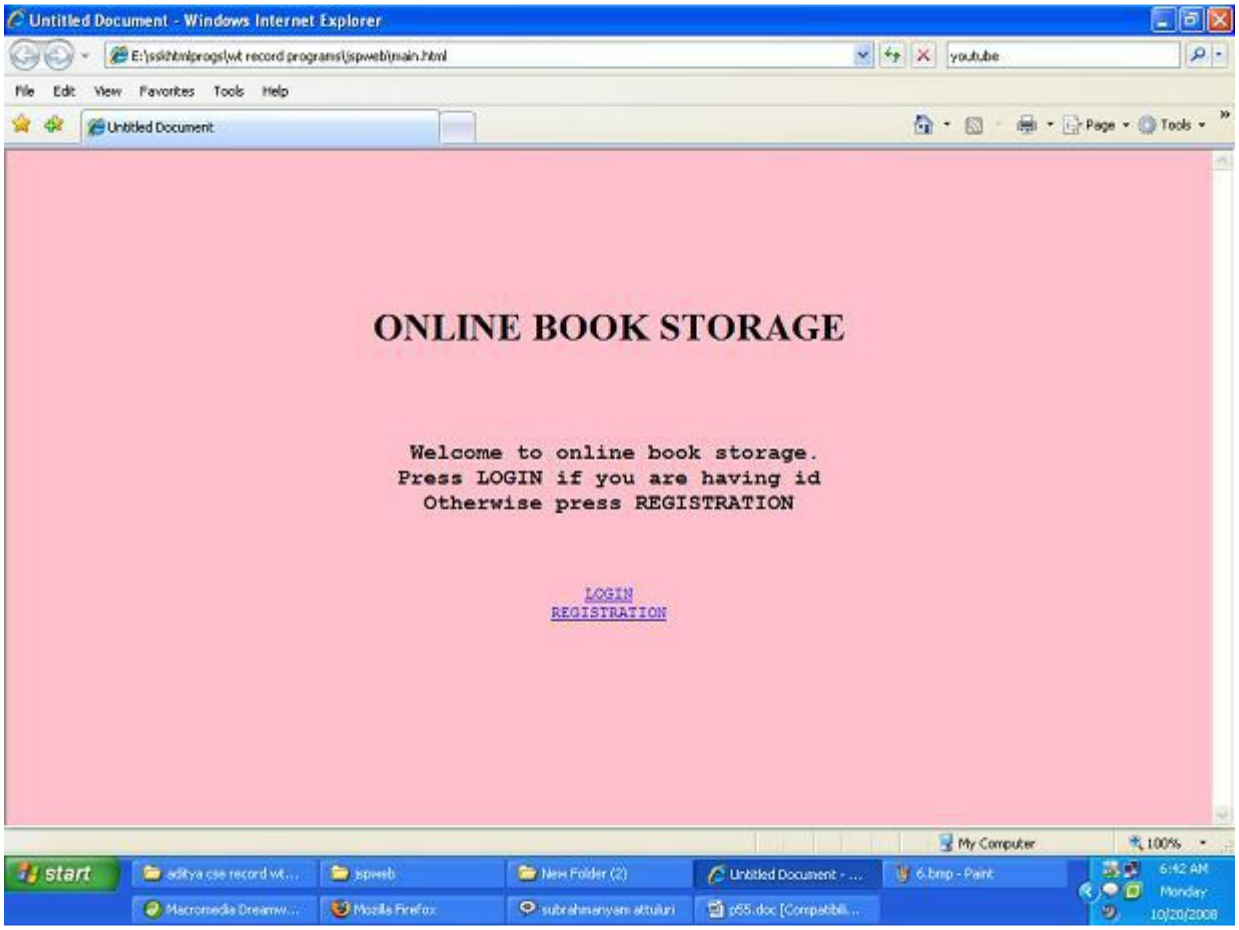
Step 4: Go to Control Panel->select Administrative tool-> select Data Source (ODBC).

Step 5: Select DNS tabs-> and click on ADD button-> select Microsoft access driver and select your database click finish.

Step 6: Save all the files in the web server (Eg:Tomcat) .

Step 7: Run the file on any web browser.

OUTPUT:

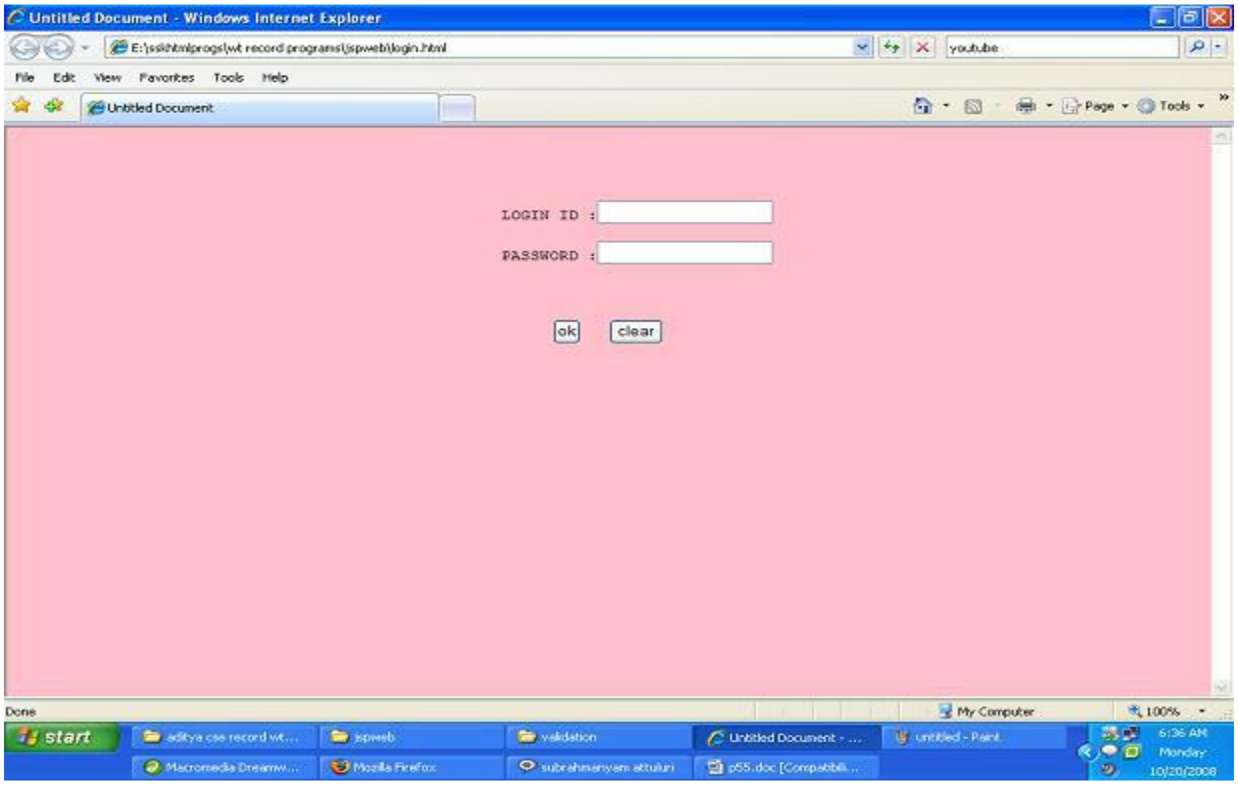


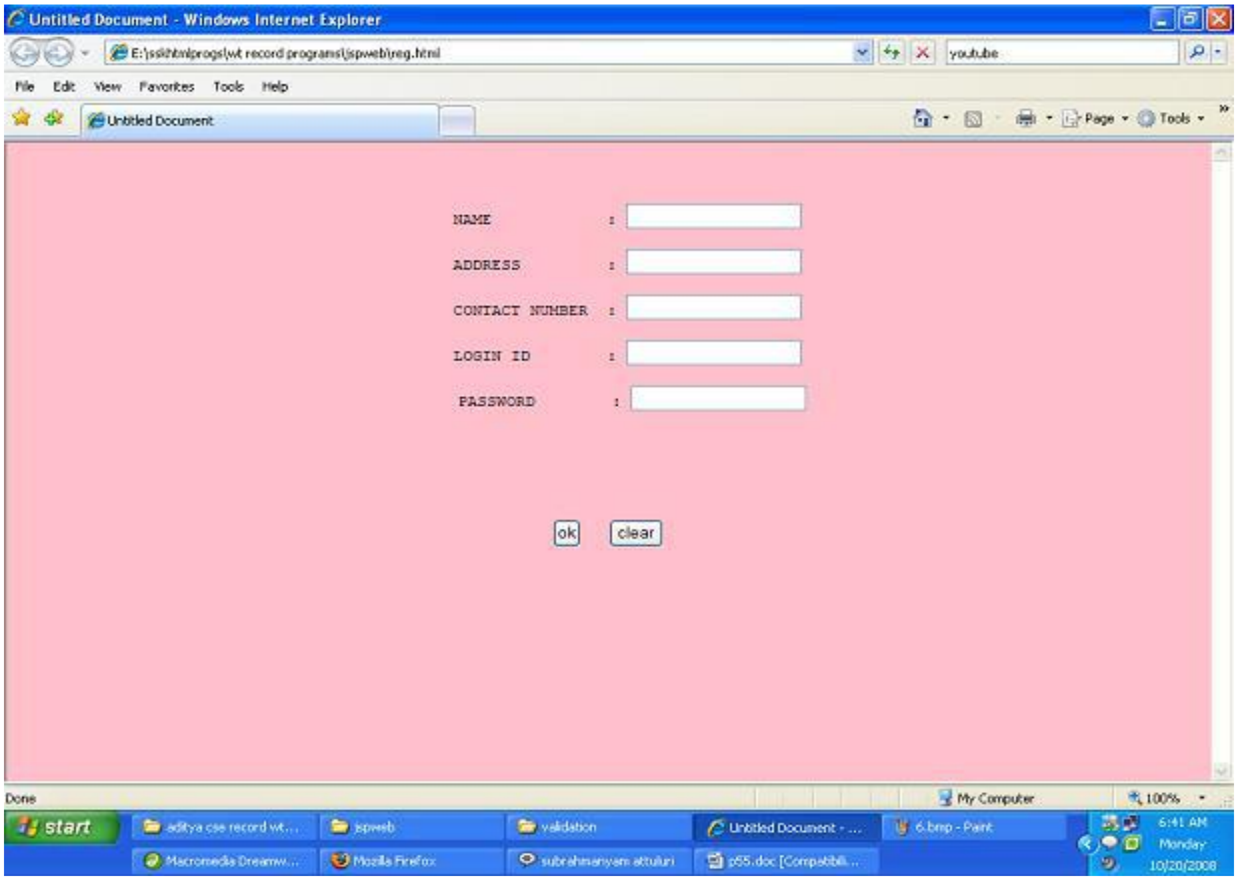
Untitled Document - Windows Internet Explorer
E:\sslhtmlprogs\wt record programs\jspweb\main.html
youtube
File Edit View Favorites Tools Help
Untitled Document
Page Tools
My Computer
100%
start
6:42 AM
Monday
10/20/2008

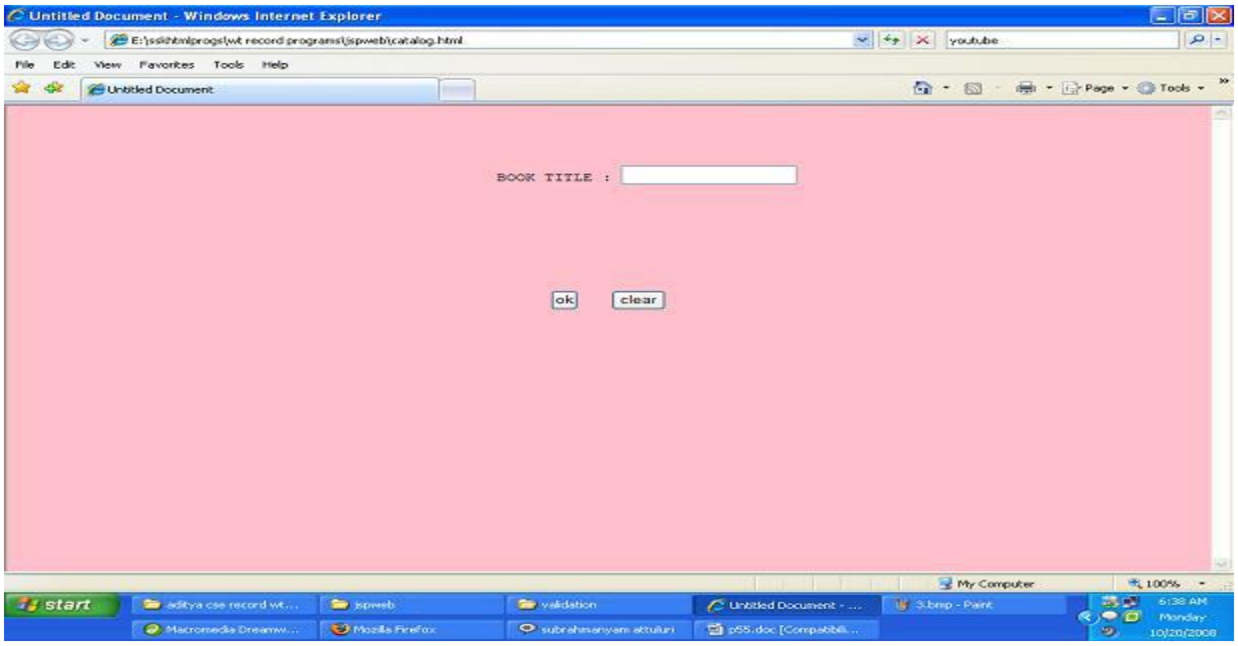
ONLINE BOOK STORAGE

```
Welcome to online book storage.
Press LOGIN if you are having id
Otherwise press REGISTRATION
```

LOGIN
REGISTRATION

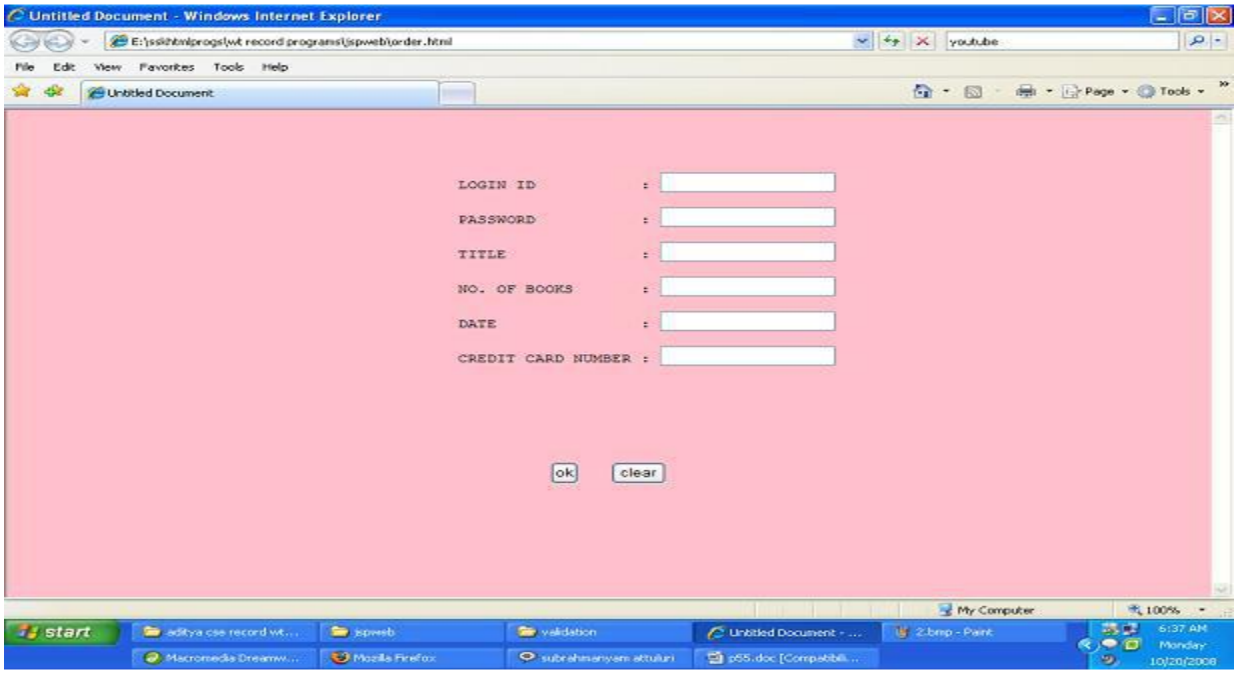

Untitled Document - Windows Internet Explorer
E:\sslhtmlprogs\wt record programs\jspweb\login.html
LOGIN ID :
PASSWORD :
ok
clear


Untitled Document - Windows Internet Explorer
E:\sslhtmlprogs\wt record programs\jspweb\reg.html
NAME :
ADDRESS :
CONTACT NUMBER :
LOGIN ID :
PASSWORD :
ok
clear


Untitled Document - Windows Internet Explorer

E:\sskhtmlprogs\wt record programs\jspweb\catalog.html

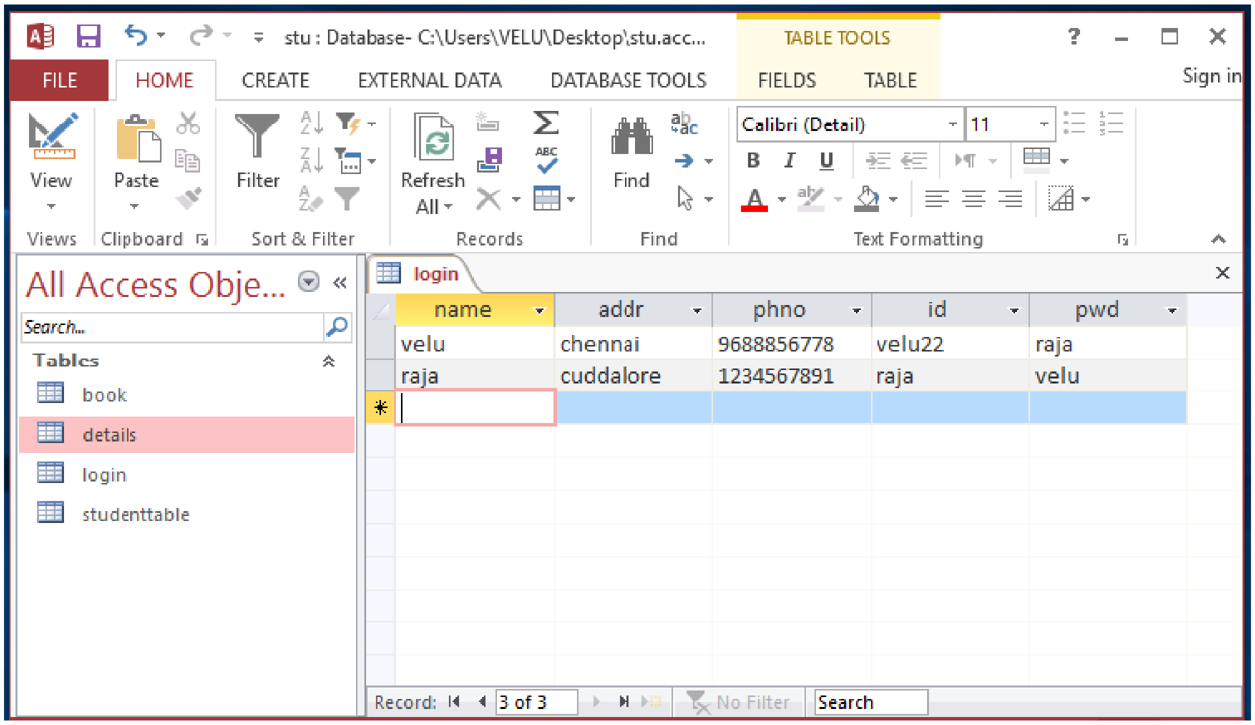
BOOK TITLE :


Untitled Document - Windows Internet Explorer

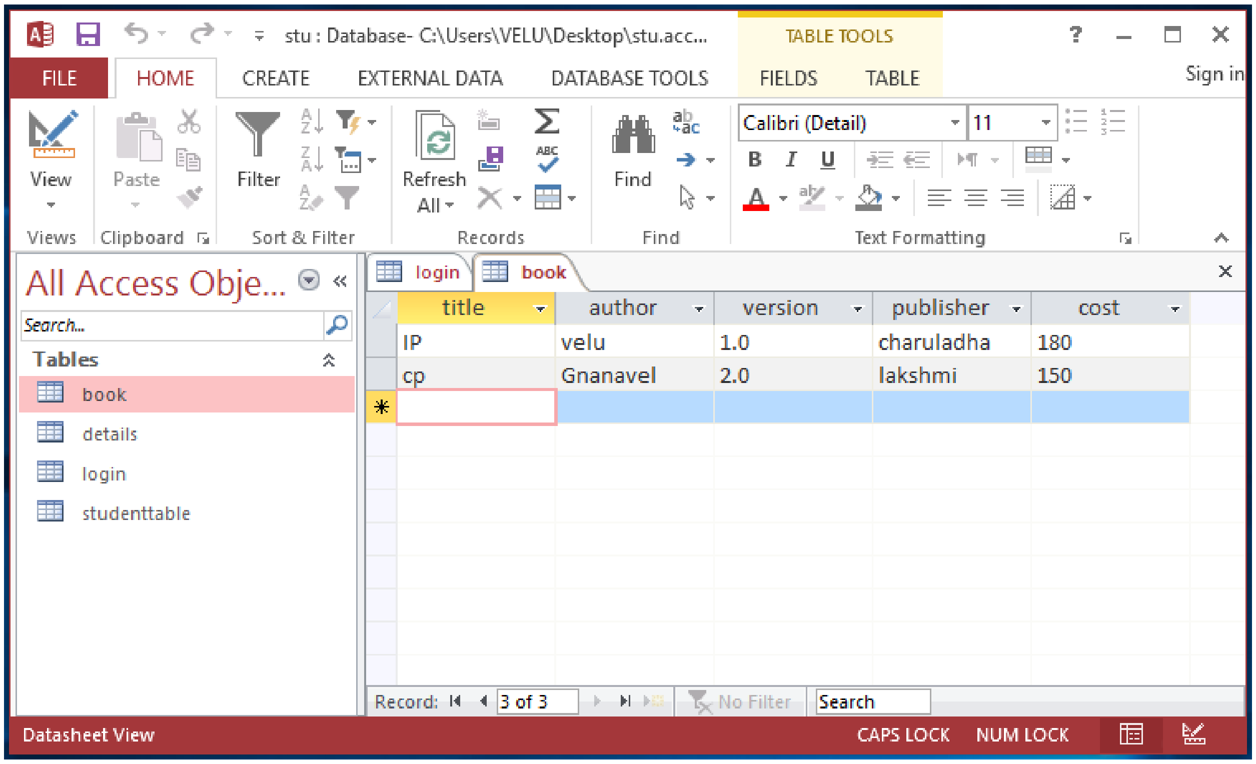
E:\sskhtmlprogs\wt record programs\jspweb\order.html

LOGIN ID	:
PASSWORD	:
TITLE	:
NO. OF BOOKS	:
DATE	:
CREDIT CARD NUMBER	:

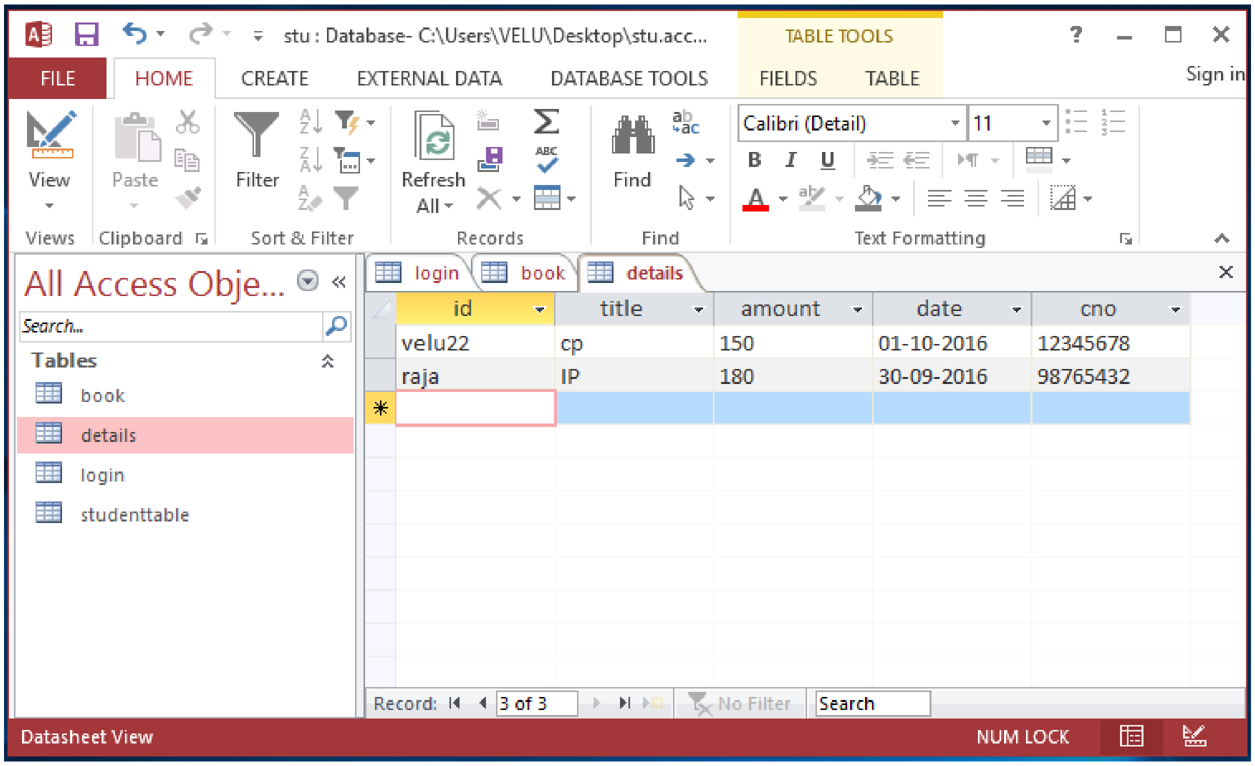
TABLES:


stu : Database- C:\Users\VELU\Desktop\stu.acc...
TABLE TOOLS
FILE HOME CREATE EXTERNAL DATA DATABASE TOOLS FIELDS TABLE
Sign in
View Paste Filter Refresh All Find
Calibri (Detail) 11
Views Clipboard Sort & Filter Records Find Text Formatting
All Access Obje...
Search...
Tables
book
details
login
studenttable
login
Record: 3 of 3 No Filter Search

name	addr	phno	id	pwd
velu	chennai	9688856778	velu22	raja
raja	cuddalore	1234567891	raja	velu
*				


stu : Database- C:\Users\VELU\Desktop\stu.acc...
TABLE TOOLS
FILE HOME CREATE EXTERNAL DATA DATABASE TOOLS FIELDS TABLE
Sign in
View Paste Filter Refresh All Find
Calibri (Detail) 11
Views Clipboard Sort & Filter Records Find Text Formatting
All Access Obje...
Search...
Tables
book
details
login
studenttable
login
book
Record: 3 of 3 No Filter Search
Datasheet View
CAPS LOCK NUM LOCK

title	author	version	publisher	cost
IP	velu	1.0	charuladha	180
cp	Gnanavel	2.0	lakshmi	150
*				



id	title	amount	date	cno
velu22	cp	150	01-10-2016	12345678
raja	IP	180	30-09-2016	98765432

RESULT:

Thus the online book shopping application using JSP with database connectivity was coded and executed successfully.

EX.NO: 8	**Retrieving Information From XML Document**
DATE:	

AIM:

To create and save an XML document at the server, which contains 10 users Information. Write a Program, which takes user Id as an input and returns the User details by taking the user information from the XML document.

ALGORITHM:

XML FILE:

Step 1: Start the xml program.

Step 2: Within the <userdata>tag, define 10 users information using user defined tags.

Step 3: The <user> tag is used to define rollno, name, phno and address of particular user.

Step 4: Stop the program.

HTML FILE:

Step 1: Start the html program.

Step 2: Within the body tag define <script> element for javascript code.

Step 3: Assign ActiveXObject("Microsoft.XMLHTTP") to xmldoc variable. Internet Explorer uses the ActiveXObject("Microsoft.XMLHTTP") to create an instance of XMLHttpRequest object, other browsers use theXMLHttpRequest() method.

Step 4: Using load() method load the xml file in your browser.

Step 5: documentElement property returns the root node of the document to ele variable.

Step 6: Use prompt box to read user input.

Step 7: The childNodes property returns a collection of a node's child nodes, as a NodeList object. The nodes in the collection are sorted as they appear in the source code and can be accessed by index numbers. The index starts at 0.

Step 8: Using for() loop display user details.

Step 9: Stop the program.

PROGRAM:

data.xml

```
<?xml version="1.0"?>
<userdata>
<user1>
        <rollno>561</rollno>
        <name> chandu</name>
        <phno>9989891510</phno>
        <address>chennai</address>
</user1>
<user2>
        <rollno>540</rollno>
        <name> karteek</name>
        <phno>9701443556</phno>
        <address>chennai</address>
</user2>
<user3>
        <rollno>525</rollno>
        <name> giri</name>
        <phno>9897895301</phno>
        <address>trichy</address>
</user3>
<user4>
        <rollno>526</rollno>
        <name>gopi</name>
        <phno>9999789540</phno>
        <address>salem</address>
</user4>
```

```
<user5>
        <rollno>513</rollno>
        <name> manoj</name>
        <phno>9989233331</phno>
        <address>madurai</address>
</user5>
<user6>
        <rollno>514</rollno>
        <name> balaji</name>
        <phno>9999789560</phno>
        <address>trichy</address>
</user6>
<user7>
        <rollno>567</rollno>
        <name>kiran </name>
        <phno>9999178957</phno>
        <address>thanjavur</address>
</user7>
<user8>
        <rollno>518</rollno>
        <name> sekhar</name>
        <phno>789580</phno>
        <address>neyveli</address>
</user8>
<user9>
        <rollno>517</rollno>
        <name>chaitu</name>
        <phno>789590</phno>
        <address>coimbatore</address>
</user9>
<user10>
        <rollno>595</rollno>
        <name> sravan</name>
        <phno>9000789500</phno>
        <address>cuddalore</address>
</user10>
</userdata>
```

Information Retrieval

data.html

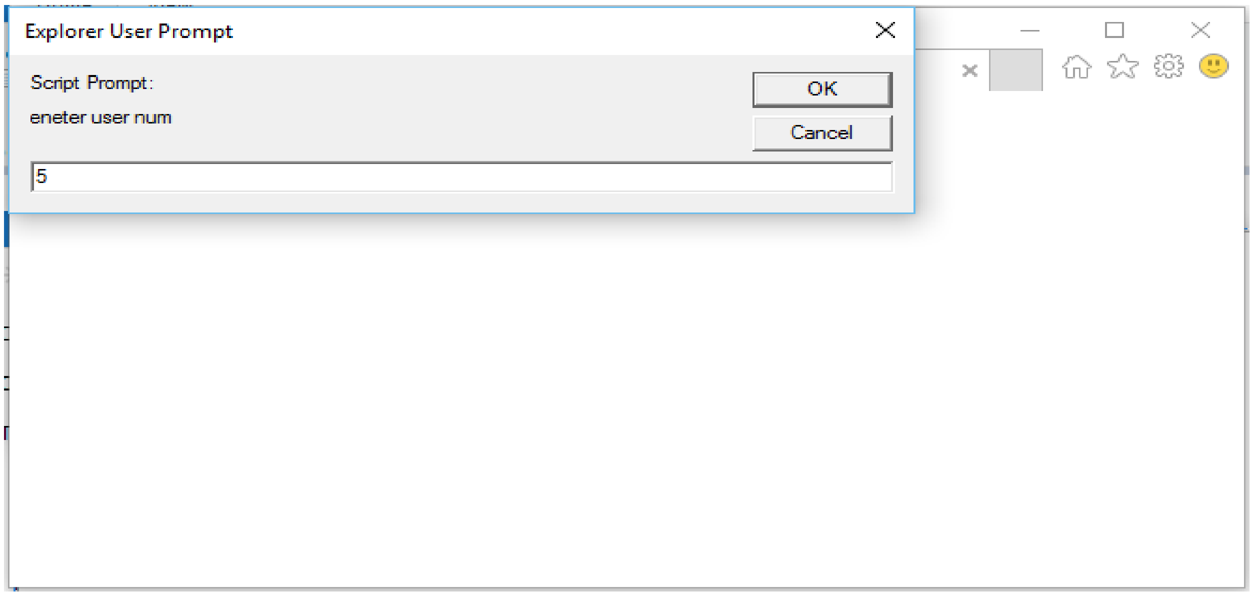
```
<html>
<head>
<title>user profile example</title>
</head>
<body>
<script type="text/JavaScript">
var xmldoc=new ActiveXObject("Microsoft.XMLDOM");
xmldoc.load("data.xml");
var ele=xmldoc.documentElement;
var y=window.prompt("eneter user num",1);
var node=ele.childNodes.item(y-1);
for(var i=0;i<node.childNodes.length;i++)
{
var child=node.childNodes.item(i);
var val=child.firstChild;
document.write("<h2>"+child.nodeName+":"+val.nodeValue);
}
</script>
</body>
</html>
```

EXECUTION:

Step 1: Save xml file with file extension as .xml and html file with file extension as .html in one directory.

Step 2: Run the html only in internet explorer web browser.

OUTPUT:

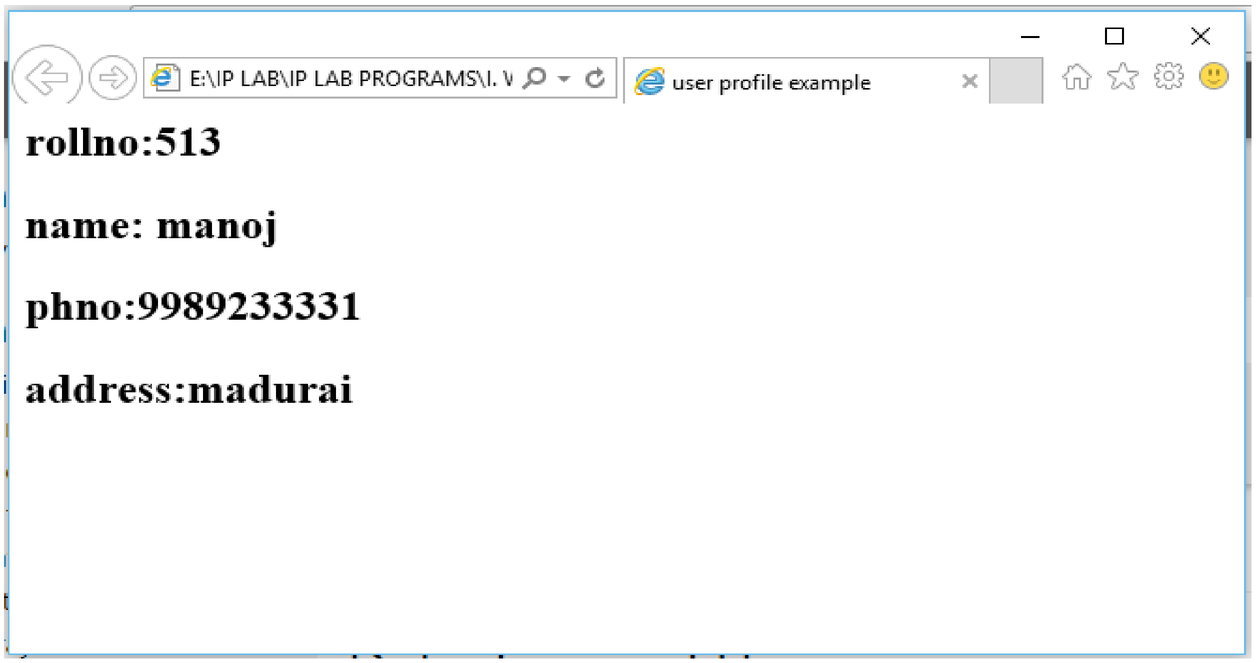


Explorer User Prompt

Script Prompt:

eneter user num

5



```
rollno:513

name: manoj

phno:9989233331

address:madurai
```

RESULT:

Thus the program, which takes user Id as an input and returns the User details by taking the user information from the XML document is executed successfully.

EX.NO: 9 (i)	Validate the form using PHP regular expression
DATE:	

AIM:

To write a PHP program to validate the form using PHP regular expression.

ALGORITHM:

Step 1: Start the program.

Step 2: Design an html form with the following form fields name, E-mail, website, comment, gender and submit button.

Step 3: Validate all the fields in html form using preg_match() function which is defined in PHP regular expression.

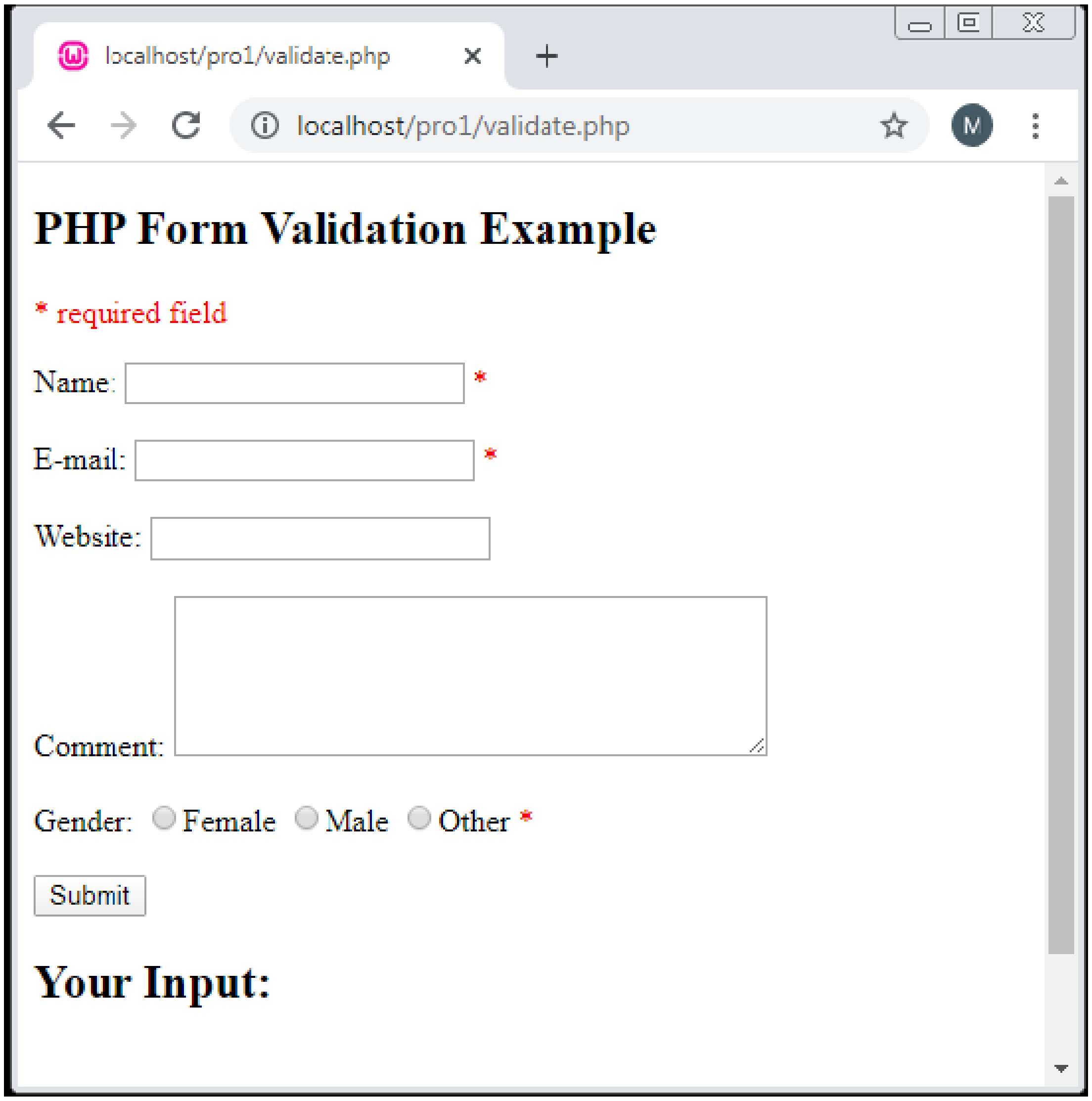
Step 4: Stop the program.

PROGRAM:

```
<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>

<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
  if (empty($_POST["name"])) {
    $nameErr = "Name is required";
  } else {
    $name = test_input($_POST["name"]);
    // check if name only contains letters and whitespace
    if (!preg_match("/^[a-zA-Z ]*$/",$name)) {
      $nameErr = "Only letters and white space allowed";
    }
  }

  if (empty($_POST["email"])) {
    $emailErr = "Email is required";
  } else {
    $email = test_input($_POST["email"]);
    // check if e-mail address is well-formed
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
      $emailErr = "Invalid email format";
    }
  }

  if (empty($_POST["website"])) {
    $website = "";
  } else {
    $website = test_input($_POST["website"]);
    // check if URL address syntax is valid
    if  (!preg_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?=~_|!:,.;]*[-a-
z0-9+&@#\/%=~_|]/i",$website)) {
      $websiteErr = "Invalid URL";
    }
  }

  if (empty($_POST["comment"])) {
    $comment = "";
  } else {
    $comment = test_input($_POST["comment"]);
  }
  if (empty($_POST["gender"])) {
    $genderErr = "Gender is required";
  } else {
    $gender = test_input($_POST["gender"]);
  }
}

function test_input($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
}
?>

<h2>PHP Form Validation Example</h2>
<p><span class="error">* required field</span></p>
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
  Name: <input type="text" name="name">
  <span class="error">* <?php echo $nameErr;?></span>
  <br><br>
  E-mail: <input type="text" name="email">
  <span class="error">* <?php echo $emailErr;?></span>
  <br><br>
  Website: <input type="text" name="website">
  <span class="error"><?php echo $websiteErr;?></span>
  <br><br>
  Comment: <textarea name="comment" rows="5" cols="40"></textarea>
  <br><br>
  Gender:
  <input type="radio" name="gender" value="female">Female
  <input type="radio" name="gender" value="male">Male
  <input type="radio" name="gender" value="other">Other
  <span class="error">* <?php echo $genderErr;?></span>
  <br><br>
  <input type="submit" name="submit" value="Submit">
```

```
</form>

<?php
echo "<h2>Your Input:</h2>";
echo $name;
echo "<br>";
echo $email;
echo "<br>";
echo $website;
echo "<br>";
echo $comment;
echo "<br>";
echo $gender;
?>

</body>
</html>
```

OUTPUT:



PHP Form Validation Example

* required field

Name: *

E-mail: *

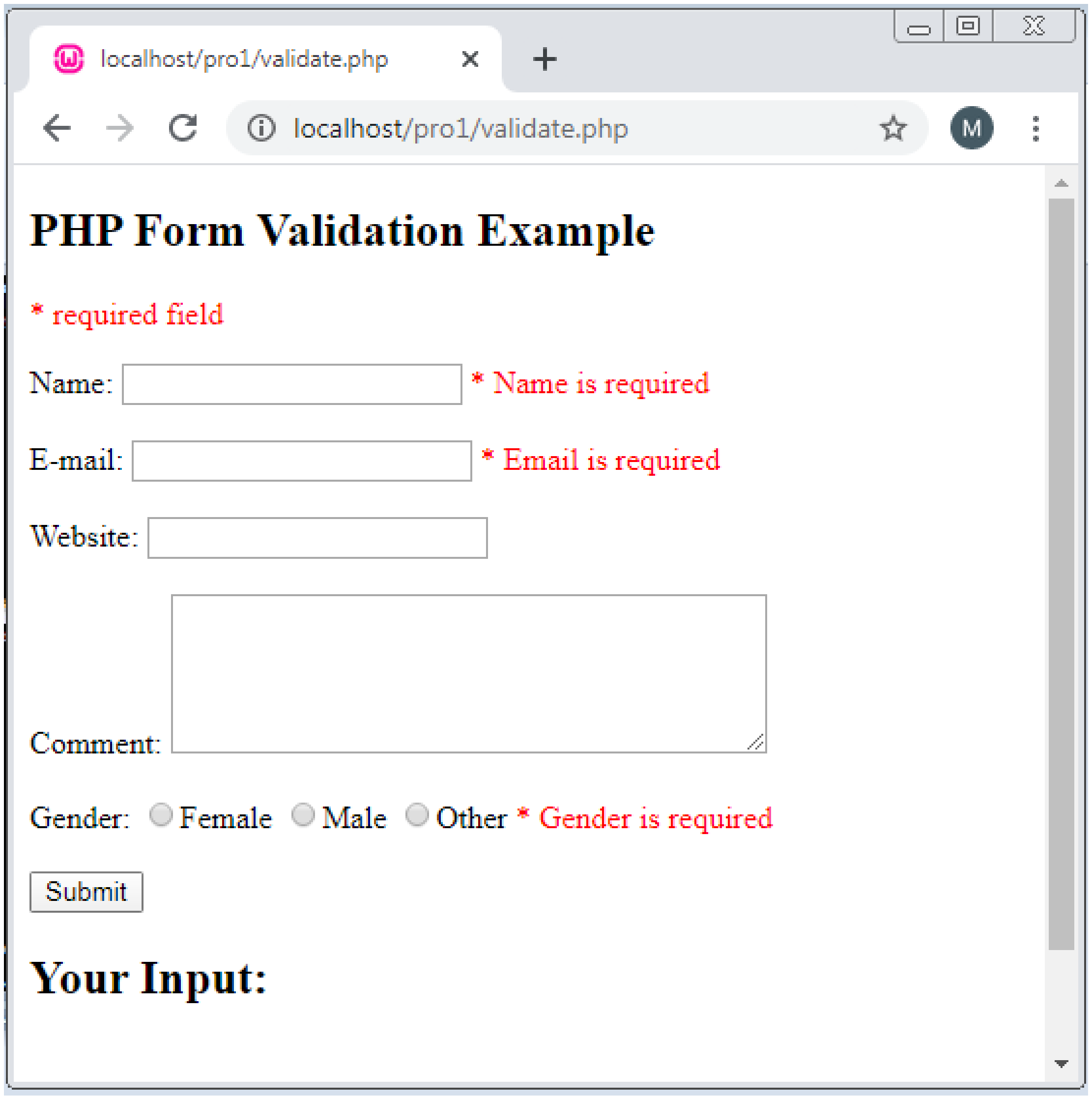
Website:

Comment:

Gender: Female Male Other *

Submit

Your Input:



localhost/pro1/validate.php

PHP Form Validation Example

* required field

Name: * Name is required

E-mail: * Email is required

Website:

Comment:

Gender: Female Male Other * Gender is required

Submit

Your Input:

RESULT:

Thus the PHP program for validating the form using PHP regular expression is executed successfully.

EX.NO: 9 (i)	**Displaying Employee Details Using PHP With Database**
DATE:	

AIM:

To write a PHP program to display employee details using mysql database.

ALGORITHM:

Step 1: Start the program.

Step 2: Assign localhost with port number 3306 to a variable $host.

Step 3: Assign the database name test to a variable $dbname.

Step 4: Pass the parameters $host, $user, $pass, $dbname in mysqli_connect() function to establish connection with database.

Step 5: If connection is failed, mysqli_connect_error() function displays error message.

Step 6: Assign database query statement to the variable $sql.

Step 7: Execute the query using mysqli_query() function.

Step 8: The method mysqli_num_rows($retval) returns the number of records present in the database.

Step 9: In while loop the method mysqli_fetch_assoc() is called to fetch the record from database and stored in $row.

Step 10: Display the records of each employee.

Step 11: Terminate the connection using mysqli_close().

Step 12: Stop the program.

PROGRAM:

```
<?php
$host = 'localhost:3306';
$user = ' ';
$pass = ' ';
$dbname = 'test';
$conn = mysqli_connect($host, $user, $pass, $dbname);
if(!$conn)
{
        die('Could not connect: '.mysqli_connect_error());
}
echo 'Connected successfully<br/>';
$sql = 'SELECT * FROM emp4';
$retval=mysqli_query($conn, $sql);
if(mysqli_num_rows($retval) > 0)
{
        while($row = mysqli_fetch_assoc($retval))
        {
                echo "EMP ID :{$row['id']}  <br> ".
                "EMP NAME : {$row['name']} <br> ".
                "EMP SALARY : {$row['salary']} <br> ".
                "--------------------------------<br>";
        }
}
else
{
        echo "0 results";
}
mysqli_close($conn);
?>
```

OUTPUT:

```
Connected successfully
EMP ID :1
EMP NAME : ratan
EMP SALARY : 9000
--------------------------------
EMP ID :2
EMP NAME : karan
EMP SALARY : 40000
--------------------------------
EMP ID :3
EMP NAME : jai
EMP SALARY : 90000
--------------------------------
```

TABLE

EMP ID	**EMP NAME**	**EMP SALARY**
1	Ratan	9000
2	Karan	40000
3	jai	90000

RESULT:

Thus the PHP program for displaying the employee details using mysql database is executed successfully.

EX.NO: 10	Creating a Web Service for Collecting People Opinion
DATE:	

AIM:

To create a webservice in java to collect people's opinion about a product to predict its sale.

PROCEDURE

1. Create project

Open Netbeans >> Select File >> New Project >> Java Web -> Web Application:

Click Next >> Enter Project name: 'WC1':

Click Next >> Select Server->Click Finish

2. Create WebService

Right click on the project 'wc1' >> New >> Web Service:

Enter Web service name: two-Enter Package: com.prgguru.example

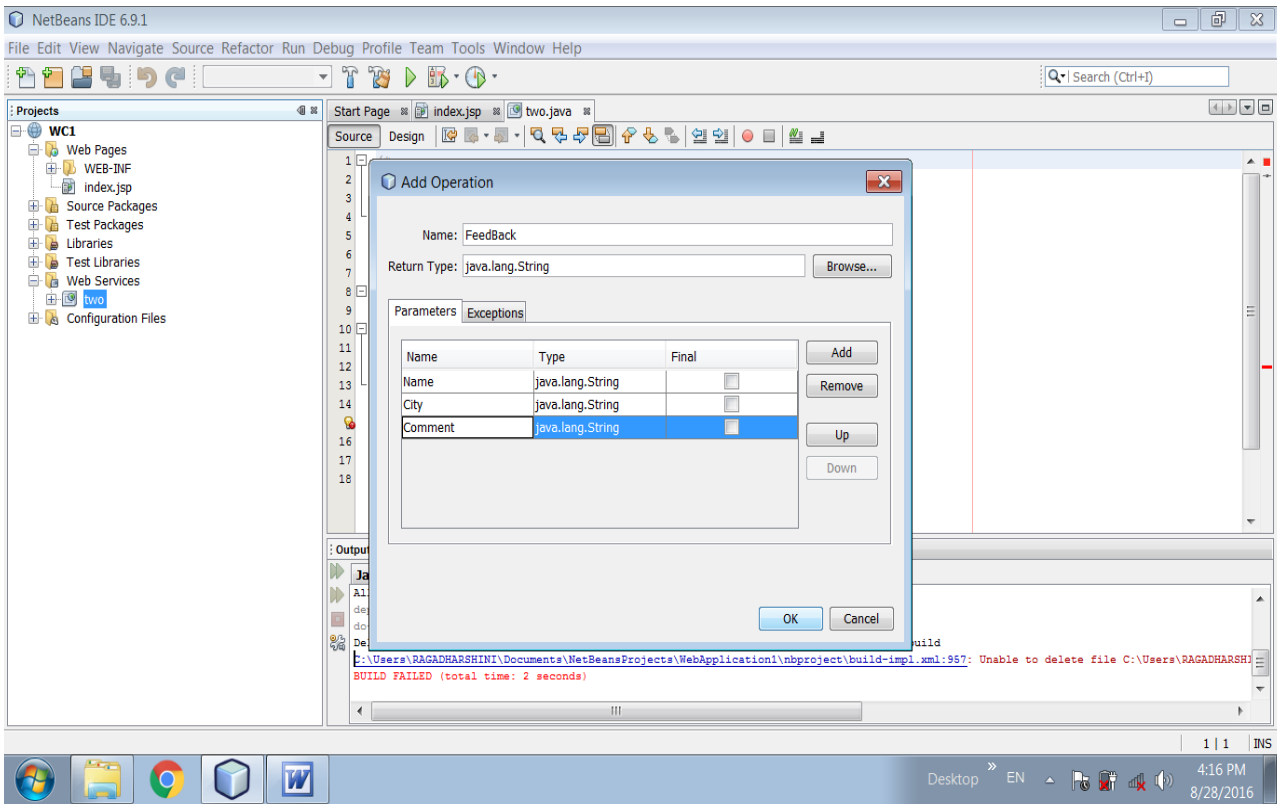
Click Finish

3. Adding an Operation to the Web Service

The NetBeans IDE provides a dialog for adding an operation to a web service. Right Click the 'two' webservice. Click Add Operation. The Add Operation dialog opens Add Operation dialog box, type FeedBack in Name and type String in the Return Type drop-down list.

4. In the lower part of the Add Operation dialog box, click Add and create a parameter of type int named Name,City,Comment as String type.

5. Click OK at the bottom of the Add Operation dialog box. You return to the editor.



Name: FeedBack
Return Type: java.lang.String

Name	Type	Final
Name	java.lang.String	☐
City	java.lang.String	☐
Comment	java.lang.String	☐

```
C:\Users\RAGADHARSHINI\Documents\NetBeansProjects\WebApplication1\nbproject\build-impl.xml:957: Unable to delete file C:\Users\RAGADHARSHI
BUILD FAILED (total time: 2 seconds)
```

6.Remove the default sayhello operation, by deleting the sayhello() method in the source code

7. In code window, type the coding for FeedBacK operation

Two.java

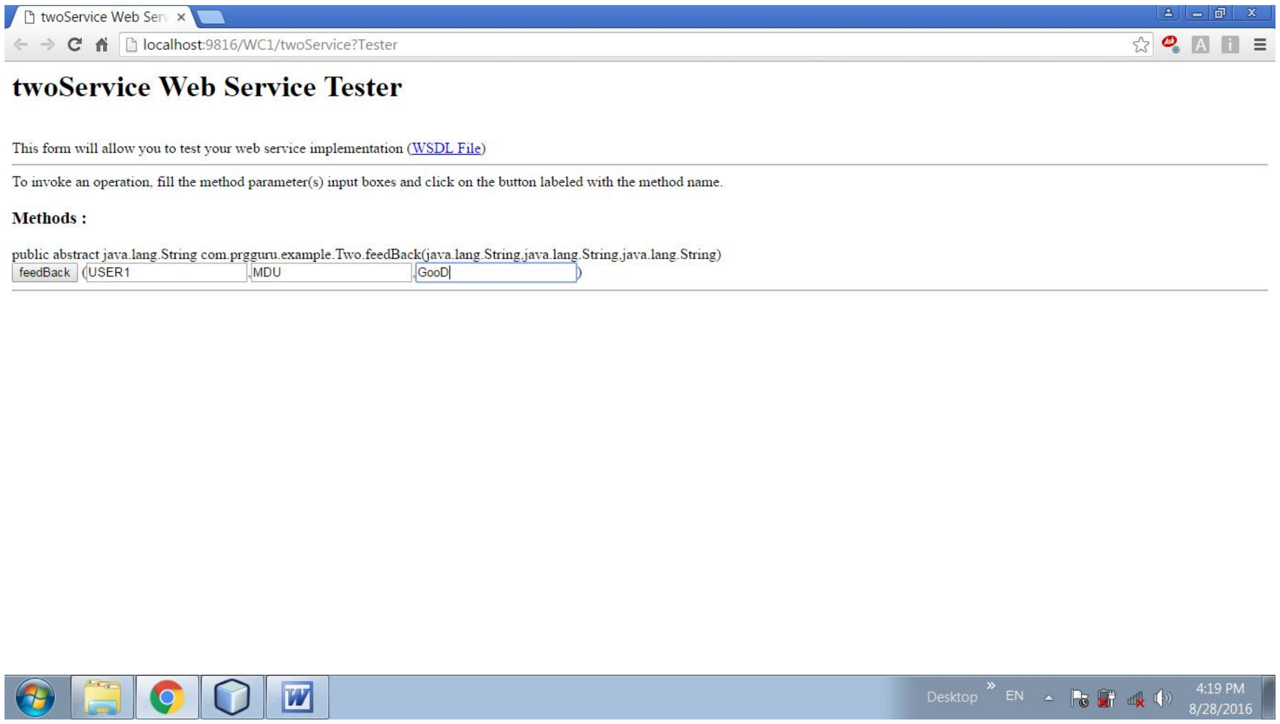
```
package com.prgguru.example;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
@WebService()
public class two
{
    @WebMethod(operationName = "FEEDBACK")
    public String FeedBack (@WebParam(name = "Name")
    String Name, @WebParam(name = "City")
    String City, @WebParam(name = "Comment")
```

```
        String Comment)
        {
            int G=0,B=0,P=0;
        if(Comment.equals("GooD"))
                G=G+1;
        else if (Comment.equals("Better"))
                B=B+1;
         else
                P=P+1;
    String s="<html><body>NAME :"+Name+ "<br> GooD= "+String.valueOf(G)+"Better= "+ String.valueOf(B)+"Poor= "+String.valueOf(P);
            return s;
        }
    }
```

8 Clean & Build the project

9 Deploy the project

10 Expand webservice->right click two webservice ->select Test webservice.Now the output is displayed on browser window.


twoService Web Serv ×

localhost:9816/WC1/twoService?Tester

twoService Web Service Tester

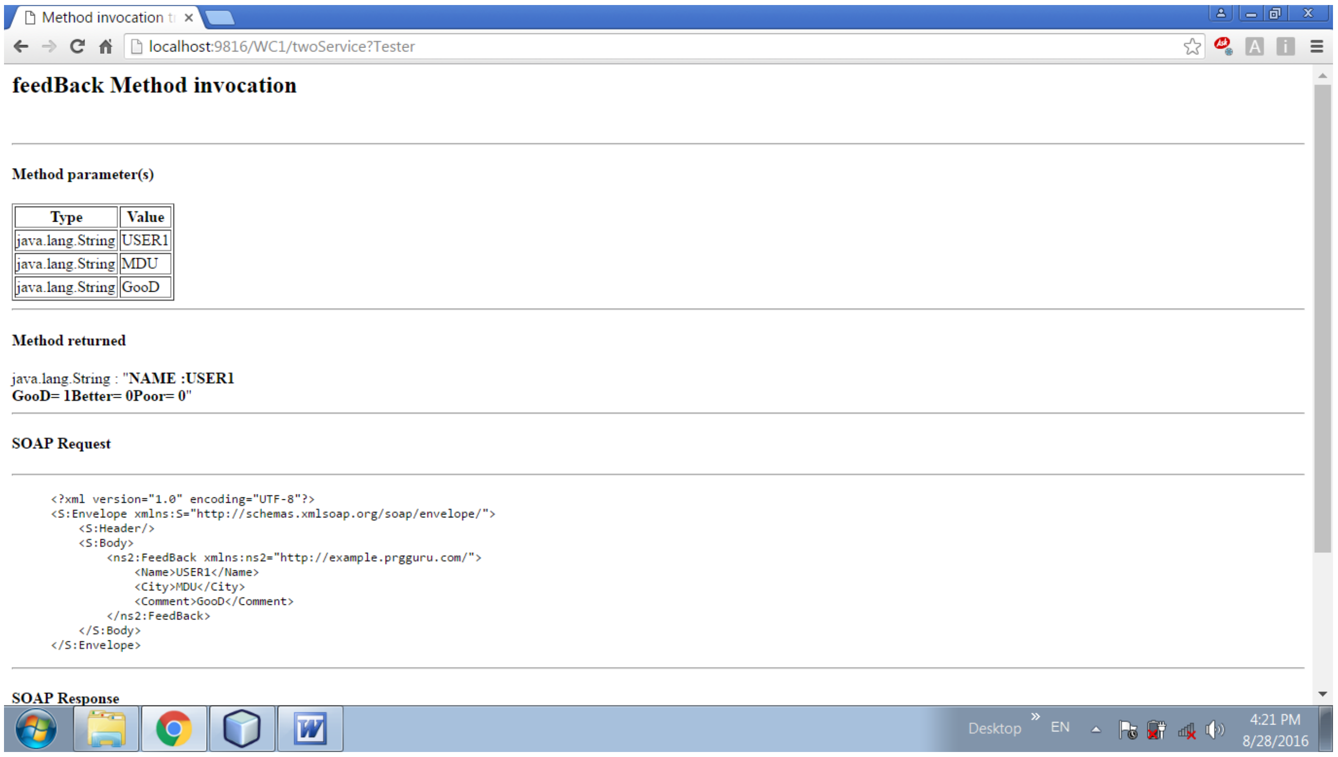
This form will allow you to test your web service implementation (WSDL File)

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract java.lang.String com.prgguru.example.Two.feedBack(java.lang.String,java.lang.String,java.lang.String)

feedBack (USER1 ,MDU ,GooD)


Method invocation t ×

localhost:9816/WC1/twoService?Tester

feedBack Method invocation

Method parameter(s)

Type	Value
java.lang.String	USER1
java.lang.String	MDU
java.lang.String	GooD

Method returned

java.lang.String : "**NAME :USER1**
GooD= 1Better= 0Poor= 0"

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:FeedBack xmlns:ns2="http://example.prgguru.com/">
            <Name>USER1</Name>
            <City>MDU</City>
            <Comment>GooD</Comment>
        </ns2:FeedBack>
    </S:Body>
</S:Envelope>
```

SOAP Response

10. Procedure to create client application(Client : JSP Page in Web Application)

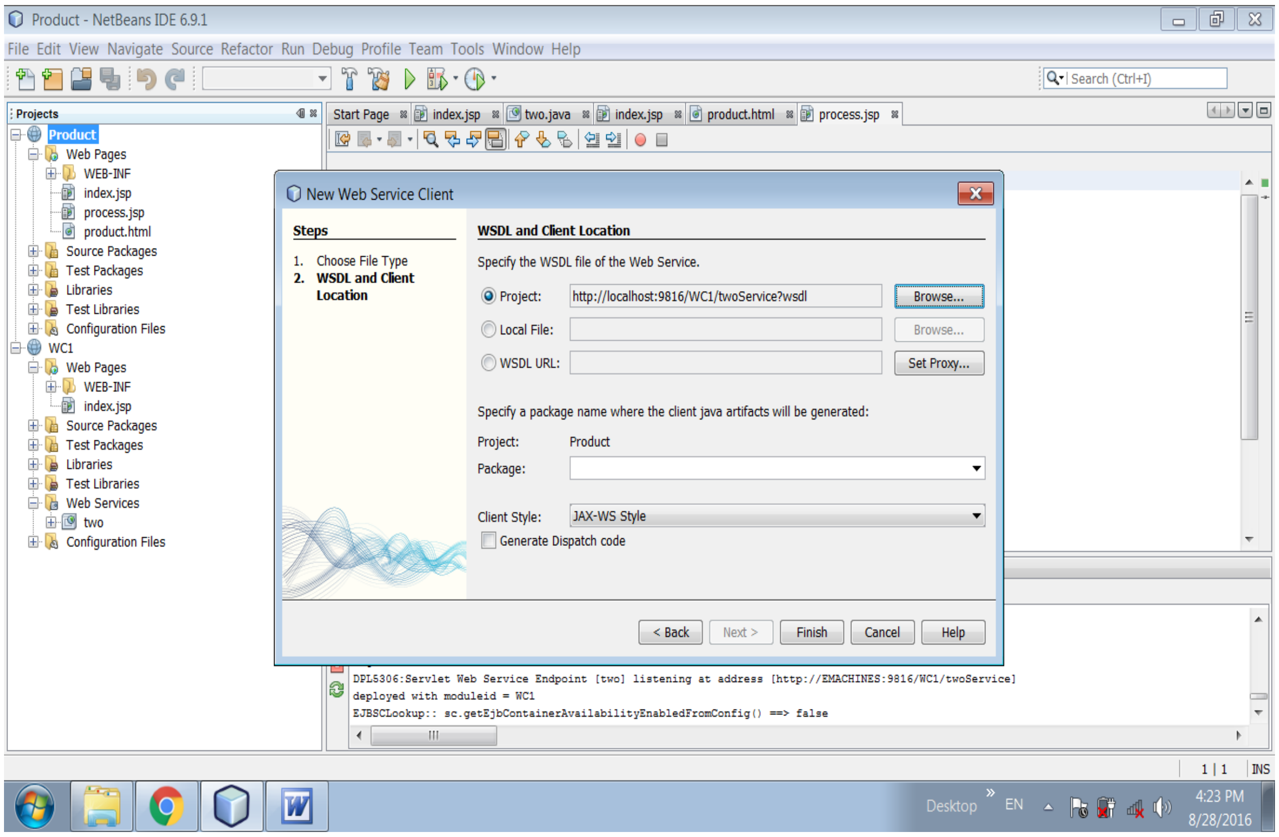
11. Choose File > New Project .Select Web Application from the Java Web category. Name the project **product**. Click Next and then click Finish.

12. Right-click the **project and** choose New > HTML file generate a feedback form

13. Right-click the **Web Pages** node and choose New > JSP

14. Right-click the **product project** and choose New > Web Service Client.

15. Select Project as the WSDL source. Click Browse. Browse to the WC1 web service and select two service in it, click OK.



WSDL and Client Location

Specify the WSDL file of the Web Service.

Project: http://localhost:9816/WC1/twoService?wsdl Browse...

Local File: Browse...

WSDL URL: Set Proxy...

Specify a package name where the client java artifacts will be generated:

Project: Product

Package:

Client Style: JAX-WS Style

Generate Dispatch code

```
DPL5306:Servlet Web Service Endpoint [two] listening at address [http://EMACHINES:9816/WC1/twoService]
deployed with moduleid = WC1
EJBSCLookup:: sc.getEjbContainerAvailabilityEnabledFromConfig() ==> false
```

16. Do not select a package name. Leave this field empty.

17. Leave the other settings at default and click Finish.

18. The Projects window displays the new web service client

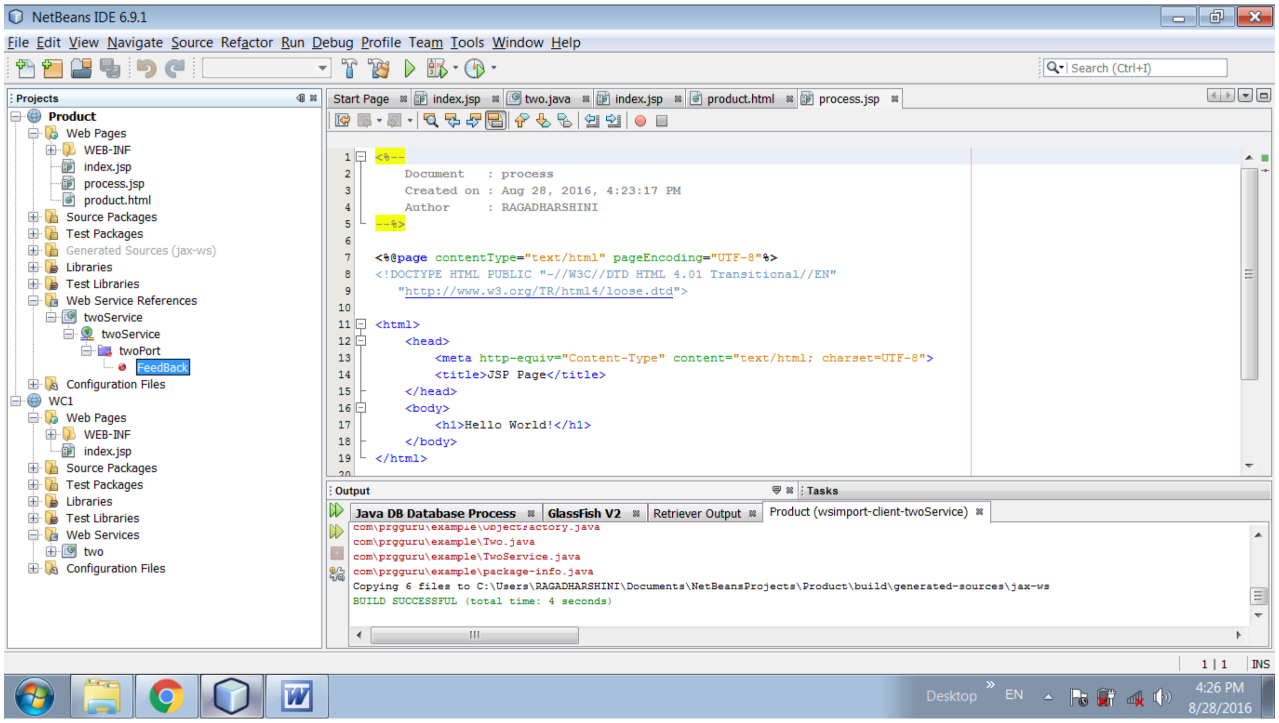
19. In the Web Service References node, expand the node that represents the web service. The FEEDBACK operation, which you will invoke from the client, is now exposed.Drag the FEEDBACK operation to the client's process.jsp page, and drop it below the H1 tags.

20. The code for invoking the service's operation is now generated in the process.jsp page.

21. Change the value for the input parameters.

22. Right-click the project and choose Run.

23. The server starts, if it wasn't running already. The application is built and deployed, and the browser opens, displaying the result:



```
<%--
    Document   : process
    Created on : Aug 28, 2016, 4:23:17 PM
    Author     : RAGADHARSHINI
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Hello World!</h1>
    </body>
</html>
```

```
com\prgguru\example\ObjectFactory.java
com\prgguru\example\Two.java
com\prgguru\example\TwoService.java
com\prgguru\example\package-info.java
Copying 6 files to C:\Users\RAGADHARSHINI\Documents\NetBeansProjects\Product\build\generated-sources\jax-ws
BUILD SUCCESSFUL (total time: 4 seconds)
```

<u>SAMPLE PROGRAM:</u>

Product.html

```
<html>
  <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
     <center>People's Opinion Form</center>
     <form         name="f1"         action="http://localhost:8080/product/process.jsp"
method="GET">
           Name:<input type="text" name="t1" ><br>
           City:<input type="text" name="t2" ><br>
           Opinion:
           <input type="radio" name="t3" value="GooD">Good
           <input type="radio" name="t3" value="Better">Better
           <input type="radio" name="t3"  value="Poor">Poor
           <input type="submit" value="SUBMIT" >
           <input type="reset" value="CLEAR" >
     </form>
  </body>
</html>
```

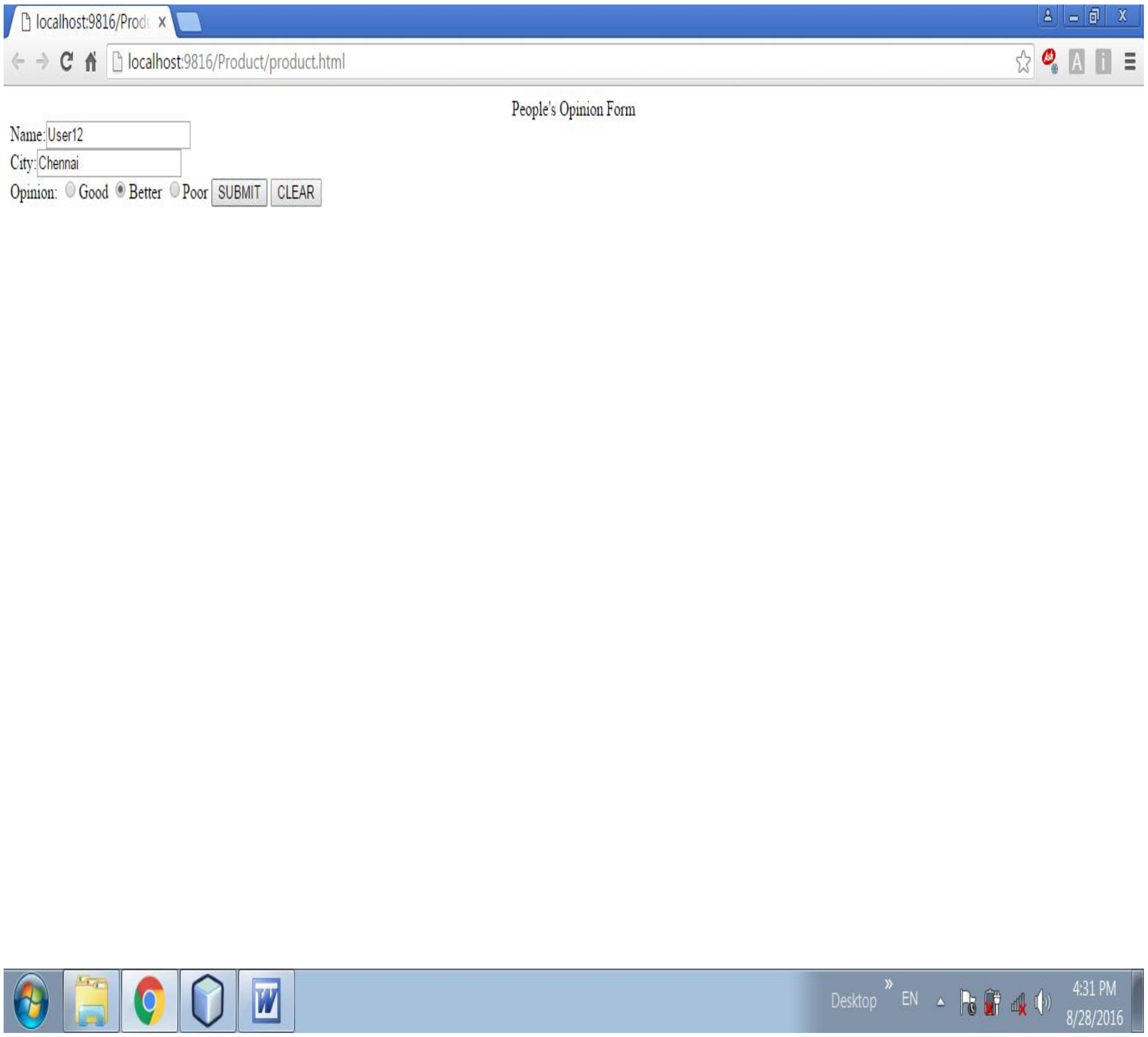
<u>Process.jsp</u>

(after adding the service code in this file by dragging)

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
      <body> <h1>RESULT</h1>
    <%-- start web service invocation --%><hr/>
    <%
    try {
         com.prgguru.example.TwoService          service          =          new
com.prgguru.example.TwoService();
         com.prgguru.example.Two port = service.getTwoPort();
          // TODO initialize WS operation arguments here
         java.lang.String name =request.getParameter("t1");
         java.lang.String city =request.getParameter("t2");
         java.lang.String comment = request.getParameter("t3");
         // TODO process result here
```

```
        java.lang.String result = port.feedback(name, city, comment);
        out.println("Result = "+result);
    } catch (Exception ex) {
        // TODO handle custom exceptions here
    }
    %>
    <%-- end web service invocation --%><hr/>
         </body>
</html>
```

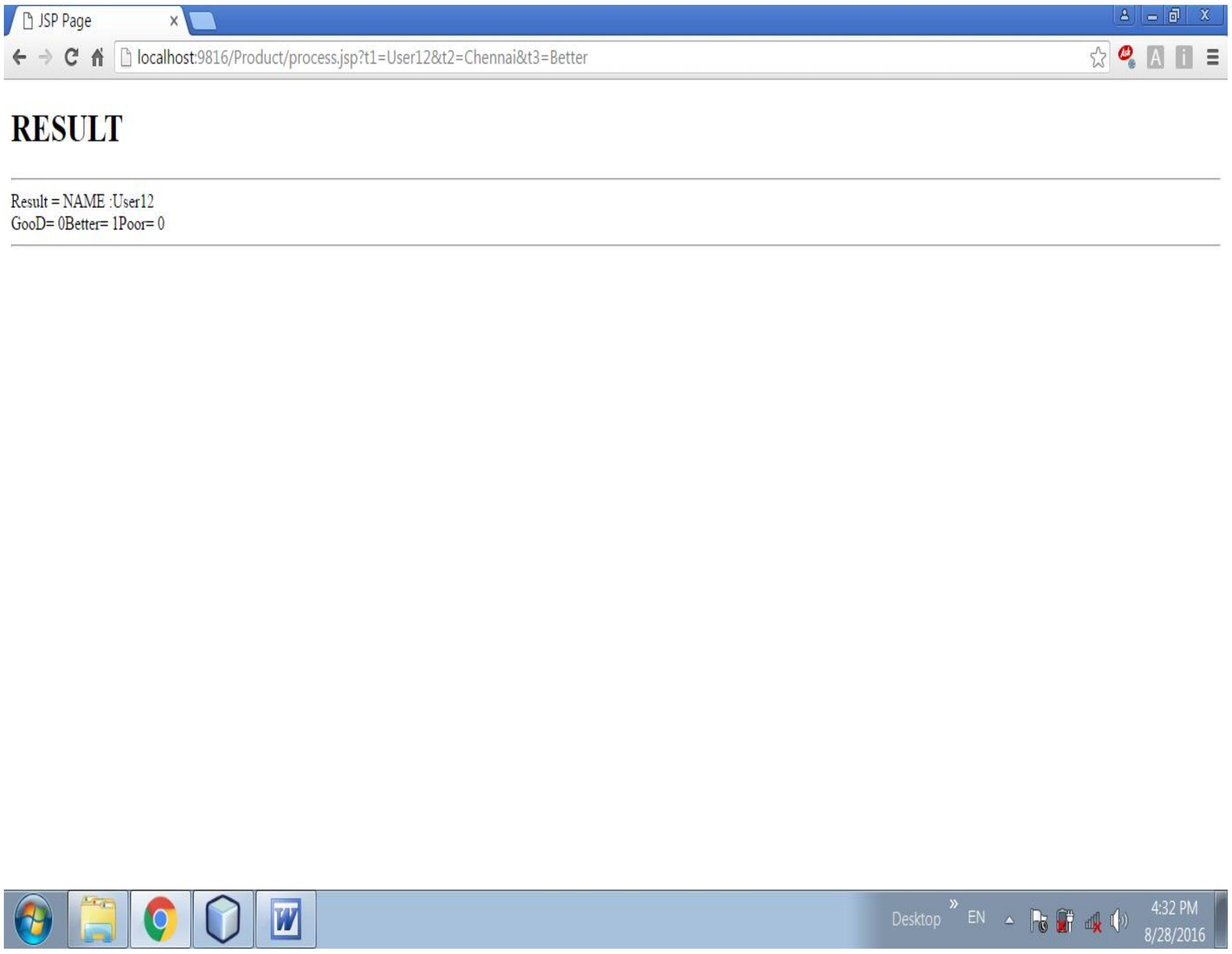
OUTPUT:



localhost:9816/Product/product.html

People's Opinion Form

Name: User12
City: Chennai
Opinion: ○ Good [✓ Better] ○ Poor SUBMIT CLEAR


JSP Page
localhost:9816/Product/process.jsp?t1=User12&t2=Chennai&t3=Better
Desktop
EN
4:32 PM
8/28/2016

```
RESULT

Result = NAME :User12
GooD= 0Better= 1Poor= 0
```

RESULT:

Thus the test program for web services java to collect people’s opinion about a product to predict its sale is also performed and the output is verified.