

<b>Ex. No. 1</b>	<b>CREATE A WEB PAGE WITH THE FOLLOWING USING HTML.</b>
	<ul style="list-style-type: none"> <li>• TO EMBED AN IMAGE MAP IN A WEB PAGE.</li> <li>• TO FIX THE HOT SPOTS.</li> <li>• SHOW ALL THE RELATED INFORMATION WHEN THE HOT SPOTS ARE CLICKED</li> </ul>

**Aim:**

To create a web page which includes a map and display the related information when a hot spot is clicked in the map.

**Procedure:**

- Create a html file with map tag
- Set the source attribute of the img tag to the location of the image and also set the use map attribute
- Specify an area with name, shape and href set to the appropriate values
- Repeat step 3 as many hot spots you want to put in the map
- Create html files for each and every hot spot the user will select

**Program:**

**ImageMap.html**

```

<HTML>
<HEAD>
<TITLE>Image Map</TITLE>
</HEAD>
<BODY>

<map name="metroid" id="metroid">
<area href='TamilNadu.html' shape='circle' coords='175,495,30' title='TamilNadu'/>
<area href = "Karnataka.html" shape = "rect" coords = "100,400,150,450" title =
"Karnataka" /> <area href = "AndhraPradesh.html" shape = "poly" coords = "150,
415, 175,348,265,360,190,420,190,440" title = "Andhra Pradesh" />
<area href = "Kerala.html" shape = "poly" coords =
"108,455,150,515,115,490,148,495,110,448,155,501" title
= "Kerala" /> </map>
</BODY>
</HTML>

```

**TamilNadu.html**

```

<HTML>
<HEAD>
<TITLE>About Tamil Nadu</TITLE>
</HEAD>
<BODY>

```

<CENTER><H1>Tamil Nadu</H1></CENTER> <HR>

<UL>

<LI>Area : 1,30,058 Sq. Kms.</LI>

<LI>Capital : Chennai</LI>

<LI>Language : Tamil</LI>

<LI>Population : 6,21,10,839</LI> </UL><hr>

<a href='ImageMap.html'>India Map</a>

</BODY>

</HTML>

## AndhraPradesh.html

<HTML>

<HEAD>

<TITLE>About Andhra Pradesh</TITLE> </HEAD>

<BODY>

<CENTER><H1>Andhra Pradesh</H1></CENTER> <HR>

<UL>

<LI>Area : 2,75,068 Sq. Kms</LI>

<LI>Capital : Hyderabad</LI>

<LI>Language : Telugu</LI>

<LI>Population : 7,57,27,541</LI>

</UL>

<hr>

<a href='ImageMap.html'>India Map</a>

</BODY>

</HTML>

## Karnataka.html

<HTML>

<HEAD>

<TITLE>About Karnataka</TITLE> </HEAD>

<BODY>

<CENTER><H1>Karnataka</H1></CENTER>

<HR>

<UL>

<LI>Area : 1,91,791 Sq. Kms</LI>

<LI>Capital : Bangalore</LI>

<LI>Language : Kannada</LI>

<LI>Population : 5,27,33,958</LI>

```
</UL>
<hr>
<a href='ImageMap.html'>India Map</a>
</BODY>
</HTML>
```

### **Kerala.html**

```
<HTML>
<HEAD>
<TITLE>About Kerala</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Kerala</H1></CENTER>
<HR>
<UL>
<LI>Area : 38,863 Sq. Kms.</LI>
<LI>Capital : Thiruvananthapuram</LI>
<LI>Language : Malayalam</LI>
<LI>Population : 3,18,38,619</LI>
</UL>
<hr>
<a href='ImageMap.html'>India Map</a>
</BODY>
</HTML>
```

### **Result:**

Thus the creation of a web page which includes a map and display the related in-formation when a hot spot is clicked in the map was executed successfully.

<b>Ex. No. 2</b>	<b>CREATE A WEB PAGE WITH ALL TYPES OF CASCADING STYLE SHEETS</b>

**Aim:**

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

**Procedure:**

1. Create a web page with frame sets consisting two frames
2. In the first frame include the links
3. In the second frame set display the web page of the link
4. Create a external style sheets
5. Create a embedded style sheets
6. Create a inline and internal style sheets and make it link to the external style sheets

**Program:**

**External.css**

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

**Style.html**

```
<html>
<head><h1><center>ALL STYLE SHEETS</center></h1>
<title>USE of STYLESHEETS
</title>
<link rel="stylesheet" href="External.css" type="text/css"> <!-- External Style Sheet
-->
<style type="text/css"> <!-- Internal Style Sheet -->
.S1{font-family:verdana; font-style:italic; color:red; text-align:center}
.S2{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> Anna University </b><br><br><br>
<li> V V College of Engineering, Nagercoil
<li> V V College of Engineering, Nellai
```

<li> V V College of Engineering, Tuticorin

</ol>

<p style="font-size:20pt;color:purple">

V V College of Engineering</p> <!-- InlineStyle Sheet -->

<p class="S2"> Run by Anna University, Chennai<br>

It is approved by AICTE.

<br>

</p>

<h2 class="S1"> V V College of Engineering</h2>

<br>

<font>Located in Tisaiyanvilai, Tirunelveli</font><br>

<br>

<font>

<h2>List of Courses offered</h2>

<ul>

<li>CSE</li>

<li>IT</li>

<li>ECE</li>

<li>EEE</li>

<li>MECH</li>

<li>Civil</li>

</ul>

</font>

<h3>Pass percentage in year 2017</h3>

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

<tr>

<th>Sl.No</th>

<th>Dept</th>

<th>Pass Percentage</th>

</tr>

<tr>

<td align="center">1</td>

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

<td align="center">80</td>

</tr>

<tr>

<td align="center">2</td>

```
<td align="center">ECE</td>
<td align="center">78</td>
</tr>
<tr>
<td align="center">3</td>
<td align="center">Mech</td>
<td align="center">75</td>
</tr>
</table>
</body>
</html>
```

**Result:**

Thus the creation of a web page that displays college information using various style sheet was successfully executed and verified.

<b>Ex. No. 3</b>	<b>CLIENT-SIDE SCRIPTS FOR VALIDATING WEB FORM CONTROLS USING HTML</b>

**AIM:**

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

**PROCEDURE:**

- Create the user login form
- Create the HTML file.
- Create the form and include the submit button.
- Create the home page, registration and user login, user profile page, books catalog, shopping cart, payment by credit card, order confirmation using various HTML tags.
- Close the HTML file.

**Home page:**

**Main.html:**

```
<frameset rows="25%, 75 %">
<frame src="top.html" name="top">
<frameset cols="25%,75%">
<frame src="left.html" name="left">
<frame src="right.html" name="right">
</frameset>
</frameset>
```

**Top.html:**

```
<html>
<body bgcolor="pink">
<br><br>
<marquee><h1 align="center"><b><u>ONLINE BOOKSTORAGE</u></b></h1></marquee>
</body>
</html>
```

**Right.html:**

```
<html>
<body bgcolor="pink">
<br><br><br><br><br>
<h2 align="center">
```

<b><p> welcome to online book storage. Press login if you are having id otherwise pressregistration.

</p></b></h2>

</body></html>

### **Left.html:**

<html>

<body bgcolor="pink">

<h3>

<ul>

<li><a href="login.html" target="right"><font color="black">LOGIN</font></a></li><br><br>

<li><a href="profile.html" target="right"><fontcolor="black"> USERPROFILE</font></a></li><br>

<br>

<li><a href="catalog.html" target="right"><fontcolor="black"> BOOKSCATALOG</font></a></li>

<br><br>

<li><a href="scart.html" target="right"><font color="black">

SHOPPINGCART</font></a></li><br>

<br>

<li><a href="payment.html" target="right"><fontcolor="black">

PAYMENT</font></a></li><br><br>

<li><a href="order.html" target="right"><font color="black"> ORDER

CONFIRMATION</font></a>

</li><br><br>

</ul>

</body>

</html>

### **Registration and user Login**

#### **Login.html:**

<html>

<body bgcolor="pink"><br><br><br>

<script language="javascript">

function validate()



```
{  
var flag=1;  
  
if(document.myform.id.value=="||  
document.myform.pwd.value=="")  
{  
flag=0;  
}  
  
if(flag==1)  
{  
alert("VALID INPUT");  
}  
else  
{  
alert("INVALID INPUT");  
document.myform.focus();  
}  
}  
  
</script>  
  
<form name="myform">  
  
<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" onClick="validate()">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
<input type="reset" value="clear" >  
  
</form>  
  
</body>  
  
</html>
```

## User profile page

## Profile.html:

```
<html>

<body bgcolor="pink"><br><br>
```

```

<script language="javascript">
function validate()
{
var flag=1;

if(document.myform.name.value=="||document.myform.addr.value=="||
document.myform.phno.value=="||document.myform.id.value=="||
document.myform.pwd.value=="")
{
flag=0;
}
var str=document.myform.phno.value;

var x;
for(var i=0;i<str.length;i++)
{
x=str.substr(i,1)
if(!(x<=9))
{
flag=0;

break;
}
}
if(flag==1)
{
alert("VALID INPUT");
}
else
{
alert("INVALID INPUT");
document.myform.focus();
}}
</script>

<form name="myform">
<div align="center"><pre>

```





```
<script language="javascript">
function validate()
{
var flag=1;

if(document.myform.id.value=="||
document.myform.title.value=="||
document.myform.no.value=="||
document.myform.cost.value=="||
document.myform.date.value=="")
{
flag=0;
}
var str=document.myform.no.value;

var x;
for(var i=0;i<str.length;i++)
{
x=str.substr(i,1)
if(!(x<=9))
{
flag=0;

break;
}
}
str=document.myform.title.value;
var str1=document.myform.cost.value;
if(!((str=="c"&& str1==444) || (str=="jsp" && str1==555)))
{
flag=0;
}
if(flag==1)
{
alert("VALID INPUT");
}
else
{

```



```
{  
flag=0;  
}  
var str=document.myform.amount.value;  
var x;  
for(var i=0;i<str.length;i++)  
{  
x=str.substr(i,1);  
if(!(x<=9))  
{  
flag=0;  
break;  
}  
}  
str=document.myform.num.value;  
for(var i=0;i<str.lenght;i++)  
{  
x=str.substr(i,1);  
if(!(x<=9))  
{  
flag=0;  
break;  
}  
}  
if(flag==1)  
{  
alert("VALID INPUT");  
}  
Else  
{
```





<b>Ex. No. 5a</b>	<b>INVOKING SERVLETS FROM HTML FORM</b>

**Aim:**

To write a java program to invoke servlets from HTML form.

**Procedure:**

***client.html:***

- (1) Create a web page using HTML form that contains the fields such as label, text and one submit button.
- (2) Set the URL of the server as the value of form's action attribute.
- (3) Run the HTML program.
- (4) Submit the form data to the server.

***server.java:***

- (1) Define the class server that extends the property of the class GenericServlet.
- (2) Handle the request from the client by using the method service() of GenericServlet class.
- (3) Get the parameter names from the HTML form by using the method getParameterNames().
- (4) Get the parameter values from the HTML forms by using the method getParameter().
- (5) Send the response to the client by using the method of PrintWriter class.

**Program**

**Index.jsp**

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <form action="MyServlet">
      <strong> WELCOME TO J2ee </strong>
      <input type="submit" value="Call My Servlet"/>
    </form>
  </body>
</html>
```

**MyServlet.java**

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

```

import javax.servlet.*;
import javax.servlet.http.*;

public class MyServlet extends HttpServlet {
    public String msg;
    public void init()
    {
        msg="Hello Servlet programmers";
    }
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try
        {
            out.println(msg);
        }
        catch(Exception e)
        {
        }
        finally
        {
            out.close();
        }
    }
}

```

## Web.xml

```

<servlet>
    <servlet-name>MyServlet</servlet-name>
    <servlet-class>MyServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>MyServlet</servlet-name>
    <url-pattern>/MyServlet</url-pattern>
</servlet-mapping>
<session-config>

```

```
<session-timeout>
    30
</session-timeout>
</session-config>
```

**Result:**

Thus the java program to invoke servlets from HTML form has been executed successfully

<b>Ex. No. 5b</b>	<b>SESSION TRACKING</b>

**Aim:**

To write programs to illustrate session tracking using hidden fields.

**Algorithm:**

1. Write a html file contain two fields for entering user name and city and a submit button to send HTTP request to server.
2. Write the servlet program to capture this request and produces a response page that contains the user name and city as a hidden fields and one submit button.
3. The submit button of the response page sends HTTP request to the another servlet that receives the field values of the page and display the field values in the final response page.
4. Thus the user name and city informations are carried over the different access to the server from the single client.

**Program:**

**index.jsp**

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Session Tracking Using Hidden Form Fields</title>
  </head>
  <body>
    <form method=get action="Servlet1">
      Enter Name: <input type="text" name="userName"/><br/>
      Enter City: <input type="text" name="userCity"/><br/>
      <input type="submit" value="Submit"/><br/>
    </form>
  </body>
</html>
```

**Web.xml**

```
<servlet>
  <servlet-name>Servlet1</servlet-name>
  <servlet-class>Servlet1</servlet-class>
</servlet>
<servlet>
  <servlet-name>Servlet2</servlet-name>
  <servlet-class>Servlet2</servlet-class>
</servlet>
```

```

<servlet-mapping>
    <servlet-name>Servlet1</servlet-name>
    <url-pattern>/Servlet1</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>Servlet2</servlet-name>
    <url-pattern>/Servlet2</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>

```

### **Servlet1.java**

```

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

public class Servlet1 extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try
        {
            String n=request.getParameter("userName");
            String c=request.getParameter("userCity");

            out.println("<form action=Servlet2>");
            out.println("<input type='hidden' name='uname' value='"+n+"'>");
            out.println("<input type='hidden' name='ucity' value='"+c+"'>");
            out.println("<input type='Submit' value='Just Click Here'>");

```

```

out.println("</form>");
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
    out.close();
}
}

```

### **Servlet2.java**

```

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

public class Servlet2 extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try
        {
            String n=request.getParameter("uname");
            out.println("Hello "+n);
            String c=request.getParameter("ucity");
            out.println(" You are from "+c);
            out.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        out.close();
    }
}

```

**Result:**

Thus the Java Program for Session Tracking Using Hidden Form Fields has been executed successfully.

<b>Ex. No. 6</b>	<b>WRITE PROGRAMS IN JAVA TO CREATE THREE-TIER APPLICATIONS USING JSP AND DATABASES</b>
	<ul style="list-style-type: none"> <li>• FOR CONDUCTING ON-LINE EXAMINATION.</li> <li>• FOR DISPLAYING STUDENT MARK LIST. ASSUME THAT STUDENT INFORMATION IS AVAILABLE IN A DATABASE WHICH HAS BEEN STORED IN A DATABASE SERVER</li> </ul>

### **Aim:**

To write java servlet programs to conduct online examination and to display student mark list available in a database.

### **Procedure:**

#### **Client:**

- ☐ In index.html on the client side declare the contents that you like to transfer to the server using html form and input type tags.
- ☐ Create a submit button and close all the included tags.

#### **Server:**

- ☐ Import all necessary packages
- ☐ Define a class that extends servlet
- ☐ In the doPost() method, do the following:
  - i) Set the content type of the response to "text/html"
  - ii) Create a writer to the response
  - iii) Get a parameter from the request
  - iv) If its value is equal to right answer then add 5 to mark variable
  - v) Similarly repeat step
  - vi) for all parameters
  - vii) Display the result in an html format using the writer

#### **Student Mark List Database:**

- ☐ Import necessary java packages and javax packages and classes
- ☐ Create a class that extends HttpServlet and implements ServletException
- ☐ and IOException
- ☐ In the doGet() method, do the following:
  - i) Create a PrintWriter object
  - ii) Open a connection with the data source name
  - iii) Write a sql query and execute to get the resultset
  - iv) Display the resultset information in html form

### **Program:**

#### **index.jsp**

```
<html>
```

```
<head>
```

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



```

<title>Exam Form</title>
</head>

<body>

<form action="ExamServlet">
  <strong><p>QUESTIONS:</p></strong>
  1. JavaUses:
  <p>

    <input type = "radio" name = "ans1" value="inter">Interpreter
    <input type = "radio" name = "ans1" value="comp">Compiler
    <input type = "radio" name = "ans1" value="intercomp">Interpreter & Compiler
  </p>

  2. Net is an OS:
  <p>
    <input type = "radio" name = "ans2" value="yes">Yes
    <input type = "radio" name = "ans2" value="no">No

  </p>

  3. ASP Uses:
  <p>
    <input type = "radio" name = "ans3" value="vbs">VBScript
    <input type = "radio" name = "ans3" value="js">JavaScript
    <input type = "radio" name = "ans3" value="vbjs">VBScript & JavaScript
  </p>
    <input type="submit" value="Submit">
    <input type="reset" value="Reset">
  </form>
</body>
</html>

```

## web.xml

```

<servlet>
  <servlet-name>ExamServlet</servlet-name>
  <servlet-class>ExamServlet</servlet-class>

```

```

</servlet>
<servlet-mapping>

    <servlet-name>ExamServlet</servlet-name>
    <url-pattern>/ExamServlet</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>

```

### **ExamServlet.java**

```

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

public class ExamServlet extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        int s=0;
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        String result=request.getParameter("ans1");
        if(result.equals("intercomp"))
            s=s+5;
        result = request.getParameter("ans2");
        if(result.equals("no"))
            s=s+5;
        result = request.getParameter("ans3");

```

```
        if(result.equals("vbjs"))
            s=s+5;
        out.println("<html><body><p><center><h1><b>RESULT</b></h1><br><br><hr>" + s + "<hr><h2>T
        hanx for ur Participation</h2></center></p></body></html>");
        out.close();
    }
}
```

**Result:**

Thus to write java servlet programs to conduct online examination and to display student mark list available in a database was successfully executed and verified.

<b>Ex. No. 7</b>	<b>PROGRAMS USING XML – SCHEMA – XSLT/XSL.</b>

**AIM:**

To Create and Save an XML document at the server, which contains 10 users information. which takes user ID as input and returns the user details by taking the user information from XML Document.

**Procedure:**

**Login Page:**

**Std.html:**

```

<html>
<head>
<script>
function LoadXmlDoc(dname)
{
xmldoc=new ActiveXObject("Microsoft.XMLDOM");
xmldoc.async="false";
xmldoc.load(dname);
return xmldoc;
}
function validate()
{
var i,k,j=0;
xmldoc=LoadXmlDoc("student.xml");
var v1=myform.n2.value;
if(v1.length==0)
window.alert("enter the roll no.");
else
{
v1=parseInt(v1);
arr=xmldoc.getElementsByTagName("students");
for(i=0;i<arr.length;i++)
{
var txt=xmldoc.getElementsByTagName("rollno")[i].childNodes[0].nodeValue; if(txt==v1)
{
k=i;
j=1;
}
}
}
}

```

```

    }
    if(j==1)
    {
        nam=xmldoc.getElementsByTagName("name")[k].childNodes[0].nodeValue;
        rol=xmldoc.getElementsByTagName("rollno")[k].childNodes[0].nodeValue;
        per=xmldoc.getElementsByTagName("percentage")[k].childNodes[0].nodeValue; 47
        document.write("<body bgcolor='pink'>");
        document.write("<table border=1 align='center'><tr><th colspan='2'>USER
        DETAILS</th></tr>");
        document.write("<tr><th>Name::</th><td>" + nam + "</td></tr>"); document.write("<tr>
        <th>RollNumber::</th><td>" + rol + "</td></tr>");
        document.write("<tr><th>Percentage::</th>
        <td>" + per + "</td></tr>"); document.write("</table></body>");
    }
    else
        window.alert("roll number not found");
    }
    }
</script>
</head>
<body bgcolor="pink" text="red">
<form name="myform">
<table align="center">
<tr><td><B>RollNumber</B></td><td><input type="text" size=15
name="n2"></td></tr>
</table>
<br>
<center><input type=submit value="submit" name="b1" onClick="validate()"></center>
</form>
</body>
</html>

```

### Student.xml:

```

<?xml version="1.0" ?>
<cse>
<students>
<rollno>501</rollno>

```

```
<name>ABC</name>
<percentage>65%</percentage>
</students>
<students>
<rollno>502</rollno>
<name>DEF</name>
<percentage>67%</percentage>
</students>
<students>
<rollno>503</rollno>
<name>GHI</name>
<percentage>69%</percentage>
</students>
<students>
<rollno>504</rollno>
<name>JKL</name>
<percentage>65%</percentage>
</students>
<students>
<rollno>505</rollno>
<name>MNO</name>
<percentage>73%</percentage>
</students>
<students>
<rollno>506</rollno>
<name>PQR</name>
<percentage>74%</percentage>
</students>
<students>
<rollno>507</rollno>
<name>stu</name>
<percentage>65%</percentage>
</students>
<students>
<rollno>508</rollno>
<name>VWX</name>
<percentage>70%</percentage>
```

```
</students>
<students>
<rollno>509</rollno>
<name>YZ</name>
<percentage>72%</percentage>
</students>
<students>
<rollno>510</rollno>
<name>PQR</name>
<percentage>75%</percentage>
</students>
</cse>
```

**Result:**

Thus the creation of an XML document at the server by taking the user information from XML Document is executed successfully.

































