



ESTD. 2001

# **PRAITHYUSHA ENGINEERING COLLEGE**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

## **LAB MANUAL**

**for**

**CCS375 – WEB TECHNOLOGIES LABORATORY**

**(Regulation 2021 – V Semester)**

**ACADEMIC YEAR: 2023-2024**

**(Odd Semester)**

**PREPARED BY,  
Ms. SHIMONA E  
AP - CSE**

## INDEX

<b>S. No</b>	<b>NAME OF THE EXPERIMENTS</b>	<b>Page No</b>
<b>1.</b>	Create a webpage with the following using HTML. <ul style="list-style-type: none"><li>• To embed an image map in a webpage.</li><li>• To fix the hot spots.</li><li>• Show all the related information when the hotspots are clicked.</li></ul>	<b>1</b>
<b>2.</b>	Create a webpage with all types of Cascading style sheets	<b>8</b>
<b>3.</b>	Client Side Scripts for Validating Web Form Controls using DHTML.	<b>10</b>
<b>4.</b>	Installation of Apache Tomcat web server.	<b>15</b>
<b>5.</b>	Write programs in Java using Servlets:	
	<b>i.</b> To invoke servlets from HTML forms	<b>19</b>
	<b>ii.</b> Session Tracking	<b>22</b>
<b>6.</b>	Write programs in Java to create three-tier applications using JSP and Databases	
	<b>i.</b> For conducting on-line examination	<b>26</b>
	<b>ii.</b> For displaying student mark list. Assume that student information is available in a database which has been stored in a database server	<b>32</b>
<b>7.</b>	Programs using XML – Schema – XSLT/XSL	<b>35</b>

**CCS375  
OBJECTIVES**

**WEB TECHNOLOGIES LABORATORY**

- To understand different Internet Technologies
- To learn java-specific web services architecture
- To Develop web applications using frameworks

**LIST OF EXPERIMENTS**

1. Create a web page with the following using HTML.
  - To embed an image map in a web page.
  - To fix the hot spots.
  - Show all the related information when the hot spots are clicked.
2. Create a web page with all types of Cascading style sheets.
3. Client Side Scripts for Validating Web Form Controls using DHTML.
4. Installation of Apache Tomcat web server.
5. Write programs in Java using Servlets:
  - To invoke servlets from HTML forms.
  - Session Tracking.
6. Write programs in Java to create three-tier applications using JSP and Databases
  - For conducting on-line examination.
  - For displaying student mark list. Assume that student information is available in a database which has been stored in a database server.
7. Programs using XML – Schema – XSLT/XSL.

**TOTAL: 30 PERIODS**

**OUTCOMES:**

**At the end of the course, the student should be able to**

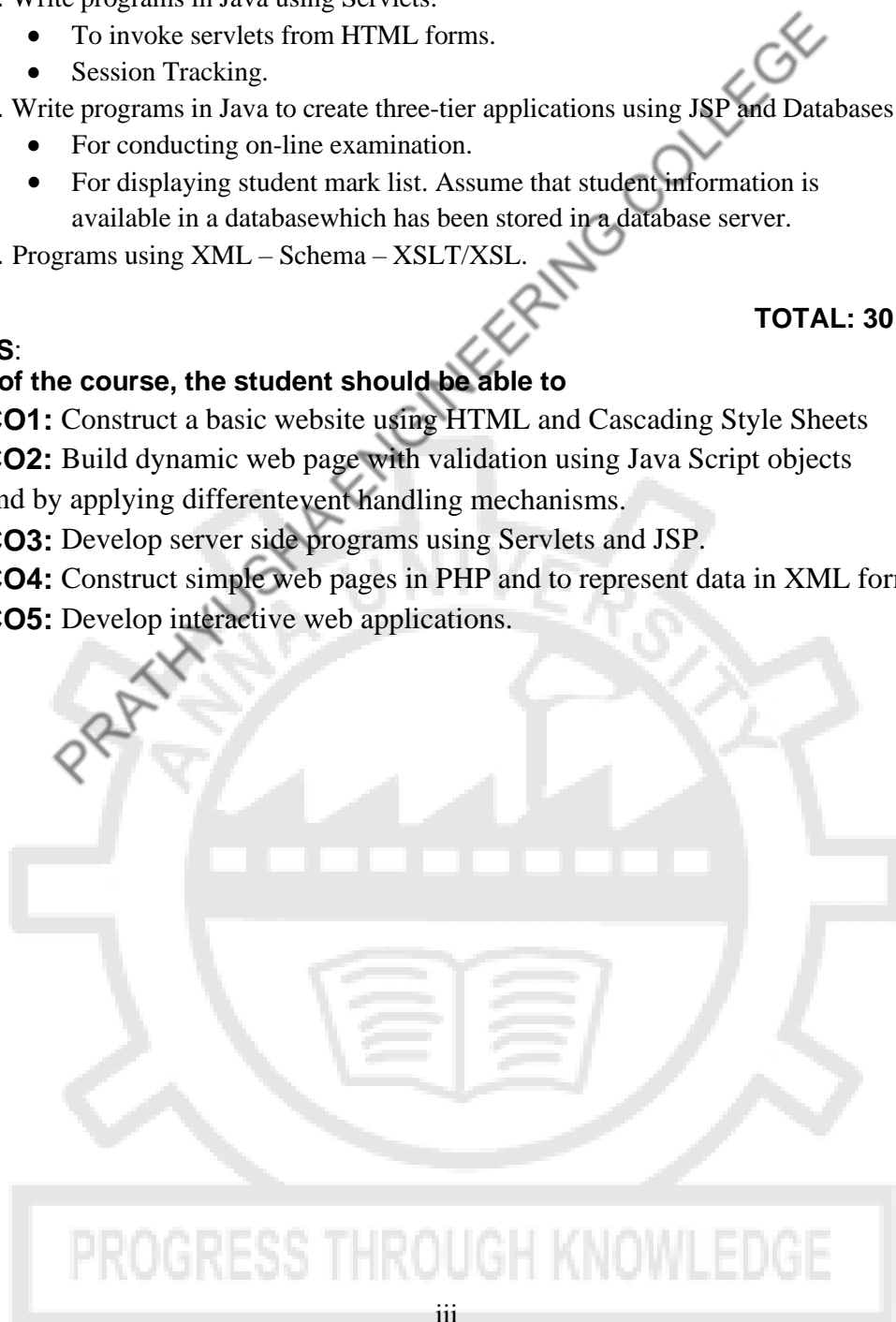
**CO1:** Construct a basic website using HTML and Cascading Style Sheets

**CO2:** Build dynamic web page with validation using Java Script objects and by applying different event handling mechanisms.

**CO3:** Develop server side programs using Servlets and JSP.

**CO4:** Construct simple web pages in PHP and to represent data in XML format.

**CO5:** Develop interactive web applications.



## EX.NO:1            CREATING A WEB PAGE USING IMAGE MAP

DATE :

**AIM:** To create a webpage with the following using HTML:

- i) To embed an image map in a webpage.
- ii) To fix the hotspots.
- iii) Show all the related information when the hot spots are clicked.

### ALGORITHM:

**Step 1:** Open notepad and type the HTML coding for homepage home. Html which has an image map using <MAP> tag and create some hotspots

**Step 2:** Hotspots are created by including a link at required coordinate position using <a> tag which directs to its corresponding web pages

**Step 3:** Write the coding for all the link web pages

**Step 4:** Run the home. Html in suitable web browser

**Step 5:** Display output.

### PROGRAM:

home.html

```
<html>
```

```
<head>
```

```
<title>Home-StatesofIndia!!!</title>
```

```
</head>
```

```
<body bgcolor="gold">
```

```
<h1><u><center>RepublicofIndia</center></u></h1>
```

```
<p>
```

```
IndiaistheSeventhLargestcountryintheworldbygeographicalarea,thesecondmostPopulouscountrywithover1.3billionpeople,IndiaisavastSouthAsiancountrywithdiverseterrain—fromHimalayanpeakstoIndianOceancoastline—
```

```
andhistoryreachingback5millennia..Indiaisafederalconstitutionalrepublicwithaparliamentarydemocracyconsistingof28statesand7UnionTerritories.</p>
```

```
<center>
```

```
<imgalign="center"width="275"height="290"alt="India"src="IndiaMap.jpg"usemap="#india"ismap="ismap">
```

```

<mapname="india">
<areashape="circle"coords="100,200,10"href="ANDHRAPRADESH.html"alt="Learnabout
andra">
<areashape="circle"coords="70,275,10"href="KERALA.html"alt="Learnaboutkerala">
<areashape="circle"coords="70,210,20"href="KARNATAKA.html"alt="Learnaboutkarnata
ka">
<areashape="circle"coords="100,250,20"href="TAMILNADU.html"alt="Learnabouttamiln
adu">
</map></center>
<h2>Features</h2>
<ul>
<li><b>Population</b>-133.92crores(2019).
<li><b>Capital</b>-NewDelhi
<li><b>LargestCity</b>-Mumbai
<li><b>Currency</b>-IndianRupee
<li><b>TimeFormat</b>-IST(UTC+5:30)
<li><b>NationalSport</b>-Hockey
<li><b>CurrentPM</b>-NarendraModi
<li><b>CurrentPresident</b>-PranabMukherjee
</li>
</ul>
<h2><b>Toviewdetailsofsouthernstatespleaseclickonthespecifiedareainthemap!</b>
</h2>
</body>
</html>

```

### Tamilnadu.html

```

<html>
<head><title>TamilNadu-India</title></head>
<bodybgcolor="palegreen">
<h1><center>TamilNadu</center></h1>
<h3>isoneofthe29statesofIndia.ItscapitalandlargestcityisChennai.TamilNadulies
inthesouthernmostpartoftheIndianPeninsulaand
ItisborderedbytheStatesofPuducherry,Kerala,Karnataka,AndhraPradesh
</h3><h3>
<ul>

```

```
<li>Districts<i>-37</i>
<li>CapitalCity<i>-Chennai</i>
<li>LargestCity<i>-Chennai</i>
<li>Governor<i>-BanwarilalPurohit</i>
<li>ChiefMinister<i>-Palanisamy</i>
<li>Population<i>-80,351,195</i>
<li>Touristspots<i>-
Mamallapuram,Ooty,Kodaikanal,Marina,MuduraiMeenakshiAmmanTem
ple,Thanjavuretc.,</i>
</ul>
<a href="Home.html">back</a>
</body>
</html>
```

## [andhrapradesh.html](#)

```
<html>
<head><title>AndhraPradesh-India</title></head>
<body bgcolor="tan">
<h1><center>AndhraPradesh</center></h1>
<h3>A.P.,isastatesituatedonthesoutheasterncoastofIndia.ItisIndia'sfourthl
argeststatebyareaandfifthlargestbypopulation.</h3>
<h3>
<ul>
<li>Districts<i>-13</i>
<li>CapitalCity<i>-Hyderabad</i>
<li>LargestCity<i>-Hyderabad</i>
<li>Governor<i>-BISWABHUSANHARICHARAN</i>
<li>ChiefMinister<i>-Y.S.JAGANMOHANREDDY</i>
<li>Population<i>-91,103,010</i>
<li>Touristspots<i>-
TirumalaTirupati,Guntur,GolcondaFort,Chandragiri,Arakuvalley,F
alaknumaPalaceetc.,</i>
</ul>
<a href="Home.html">back</a>
```

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

### **Karnataka.html**

```
<html>
<head><title>Karnataka-India</title></head>
<bodybgcolor="wheat">
<h1><center>Karnataka</center></h1>
<h3> <ul>
<li>Districts<i>-30</i>
<li>CapitalCity<i>-Bangalore</i>
<li>LargestCity<i>-Bangalore</i>
<li>Governor<i>-VajubhaiVala</i>
<li>ChiefMinister<i>-B.S. YEDIYURAPPA</i>
<li>Population<i>- 68,308,304</i>
<li>Touristspots<i>-GolGumbaz,MysorePalace,KeshavaTempleetc.,</i>
</ul>
</h3>
<a href="Home.html">back</a>
</body>
</html>
```

### **Kerala.html**

```
<html>
<head><title>Kerala-India</title></head>
<bodybgcolor="indianred">
<h1><center>Kerala</center></h1>
<h3>
<ul>
<li>Districts<i>-14</i>
<li>CapitalCity<i>-Thiruvananthapuram</i>
<li>LargestCity<i>-Thiruvananthapuram</i>
<li>Governor<i>-ARIFMOHAMMADKHAN</i>
<li>ChiefMinister<i>-PINARAYIVIJAYAN</i>
<li>Population<i>-34,545,868</i>
<li>Touristspots<i>-
```

EdakkalCaves,Palayur,KovalamBeach,Munnar,Kochi,Alapuzhaetc.,</i>

</ul>

<a href="Home.html">back</a>

</h3>

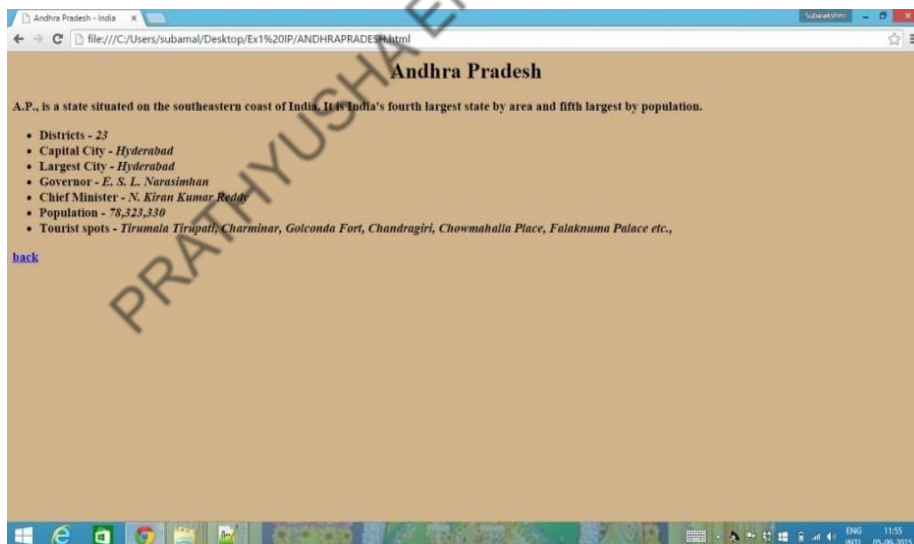
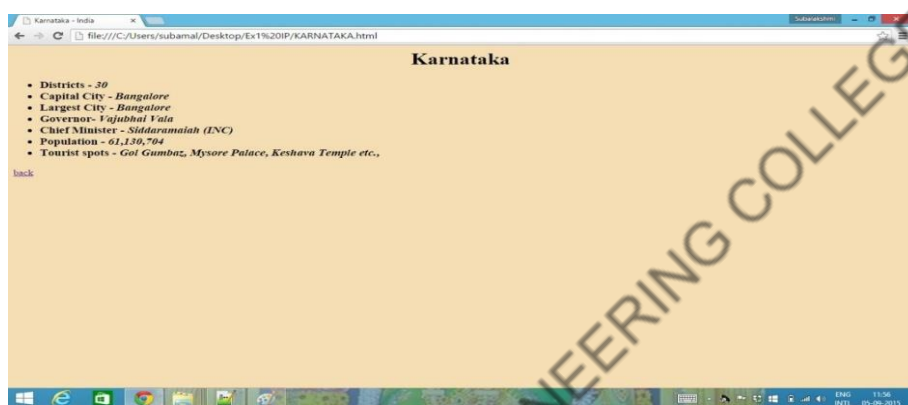
</body>

</html>

## OUTPUT:







**Result :**

Thus a web page with given specifications was created and its output was verified

## EX.NO: 2 CREATING A WEBPAGE WITH CASCADING STYLE SHEET

DATE :

### AIM:

To create a webpage with the following using html to embedded the style sheet

### ALGORITHM:

**Step1:** Create html file with the style tag, inside head tag.

**Step2:** Set the style such as font-family, font-size, color, left etc, for the heading

h1,h2,...h6 and respectively.

**Step3:** Close the head tag.

**Step4:** Specify the heading and information required inside the body tag.

**Step5:** Close the opened tag.

### PROGRAM:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
TRANSITION//EN" http://www.w3.org/TR/html1/DTD/html1.1.dtd>
```

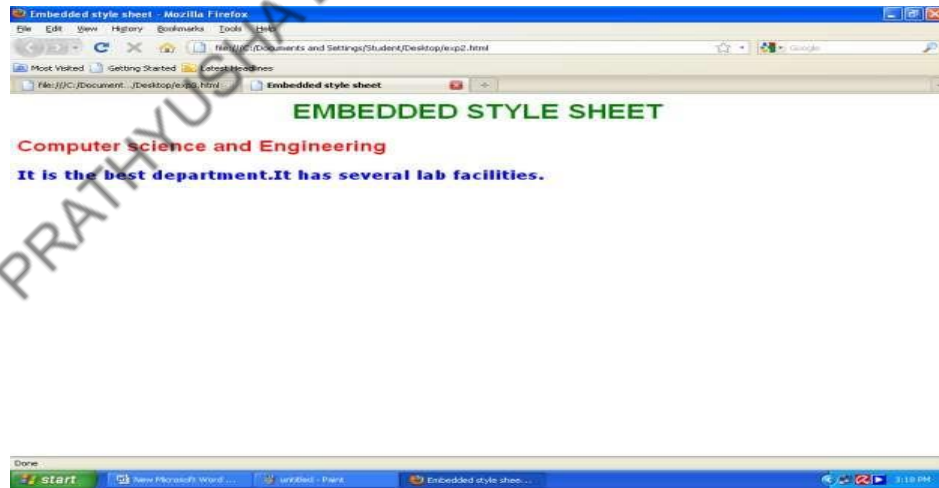
```
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<title>Embedded style sheet</title>  
<style type="text/css">h1  
{  
font-family:arial;  
color:green;  
}  
h2  
{  
font-family:arial;  
color:red; left:20px  
}  
h3  
{
```

```

font-family:arial;
color:blue;
}
p
{
font-size:14pt;
font-family:verdana
}
</style>
</head>
<body>
<h1>
<center>This is created using embedded style sheet
</center>
</h1>
<h2>This line is alligned left and red colored;
</h2>
<p>
The embedded style sheet is the most commonly used style sheet
This paragraph is return in verdana font with font size of 14.
</p>
<h3>
This is a blue <a href="colorname.html">colored</a> line
</h3>
</body>
</html>

```

**OUTPUT:**



**RESULT:** Thus creation of an webpage using cascading style sheet has been developed successfully.

## **EX.NO :3 CLIENT SIDE SCRIPTS for validating web form control using DHTML**

**DATE:**

**AIM:**

To develop a program for validating web form control using DHTML.

**ALGORITHM:**

**Step1:**Start the program.

**Step2:** Define the title within the tag.

**Step3:** Give the script type within the script tag.

**Step4:** Validate each and every column as the box with the if condition.

**Step5:** If empty value are given or the block term are next then it is verified with certain condition.

**Step6:**If values is empty then a message is been displayed.

**Step7:** Form is designed with GUI tool is description.

**Step8:** All buttons are processed accordingly.

**Step9:** Stop the program.

**PROGRAM:**

```
//Webforms.html
<html>
<head>
<script
type='text/javascript'>
function formValidator()
{
var
firstname=document.getElementById('firstname');
var lastname=document.getElementById('lastname');
var addr=document.getElementById('addr');
var zip=document.getElementById('zip');
var
Countries=document.getElementById('Countries');
var username=document.getElementById('username');
var email=document.getElementById('email');
var dd=document.getElementById('dd');
var
mm=document.getElementById('mm');
var yyyy=document.getElementById('yyyy');
var
comment=document.getElementById('comment');var
password=document.getElementById('password');
if(isAlphabet(firstname,"Please enter only letters for your First name"))
{
if(isAlphabet(lastname,"Please enter only letters for your Last name"))
{
if(isNumeric(dd,"Please enter a date"))
{
if(madeSelection(mm,"Please Choose"))
{
```

```

if(isNumeric(yyyy,"Please enter a year"))
{
if(isAlphanumeric(addr,"Enter Numbers and letters only for address"))
{
if(isNumeric(zip,"please enter a valid zip code"))
{
if(madeSelection(Countries,"Please Choose"))
{
if(lengthRestriction(username,6,8))
{
if(isAlphanumeric(password,"Enter Numbers and letters only for password"))
{
if(emailValidator(email,"Please enter a valid email address"))
{
if(notEmpty(comment,"Please fill the comment"))
{
document.write("<b><i>Thank's for submitting your
details</i></b>");alert("Successful Entry!!");
return true;
}}}}}}}}}}}}
return false;
}
function notEmpty(elem,helperMsg)
{
if(elem.value.length==0)
{
alert(helperMsg)
;elem.focus();
return false;
}
return true;
}
function isNumeric(elem,helperMsg)
{
var numericExpression=/^[0-9]+$/;
if(elem.value.match(numericExpression))
{
return true;
}
else
{
alert(helperMsg)
;elem.focus();
return false;
}
}
function isAlphabet(elem,helperMsg)
{
var alphaExp=/^[a-zA-Z]+$/;
if(elem.value.match(alphaExp)
)
{

```

```

return true;
}
else
{
alert(helperMsg);
elem.focus();
;return false;
}
}
function isAlphanumeric(elem,helperMsg)
{
var alphaExp=/^[0-9, a-z a-z, 0-9, A-Z A-Z, - 0-9 .
]+$/;if(elem.value.match(alphaExp))
{
return true;
}
else
{
alert(helperMsg)
;elem.focus();
return false;
}
}
function lengthRestriction(elem,min,max)
{
var unput=elem.value;
if(unput.length>=min&&unput.length<=max)
{
return true;
}
else
{
alert("Please enter between "+min+" and "+max+"
characters");elem.focus();
return false;
}
}
function madeSelection(elem,helperMsg)
{
if(elem.value=="Please Choose")
{
alert(helperMsg)
;elem.focus();
return false;
}
else
{
return true;
}
}
function emailValidator(elem,helperMsg)
{

```

```

var emailExp=/^[0-9 a-z . a-z 0-9]+\@[a-z]+\.[a-
z]{2,4}$/;if(elem.value.match(emailExp))
{
return true;
}
else
{
alert(helperMsg)
;elem.focus();
return false;
}
}
</script>
<h1><center><b><font color="#347235">Please
Enter YourDetails</font></b></center></h1>
</head>
<body bgcolor="LIGHTGREEN">
<hr>
<form onsubmit='return formValidator()' height="50%">
<table height="50%" border="3pt" align="center">
<tr><td><b><font color="#347235">First
Name:</font></b></td><td><inputtype='text'
id='firstname'/></td></tr><br />
<tr><td><b><font color="#347235">Last
Name:</font></b></td><td><inputtype='text' id='lastname'/></td></tr><br
/>
<tr><td><b><font color="347235">Date of
Birth(dd/mm/yyyy):</font></b></td><td><input
type='text' id='dd' />
<select id='mm'>
<option>Please Choose</option>
<option value="1">Jan</option>
<option value="2">Feb</option>
<option value="3">Mar</option>
<option value="4">Apr</option>
<option value="5">May</option>
<option value="6">Jun</option>
<option value="7">Jul</option>
<option value="8">Aug</option>
<option value="9">Sep</option>
<option value="10">Oct</option>
<option value="11">Nov</option>
<option value="12">Dec</option>
</select>
<input type='text' id='yyyy' /></td></tr><br />
<tr><td><b><font
color="#347235">Address:</font></b></td><td><inputtype='text'
id='addr'/></td></tr><br />
<tr><td><b><font color="#347235">Zip
code:</font></b></td><td><inputtype='text' id='zip'/></td></tr><br />
<tr><td><b><font

```

```

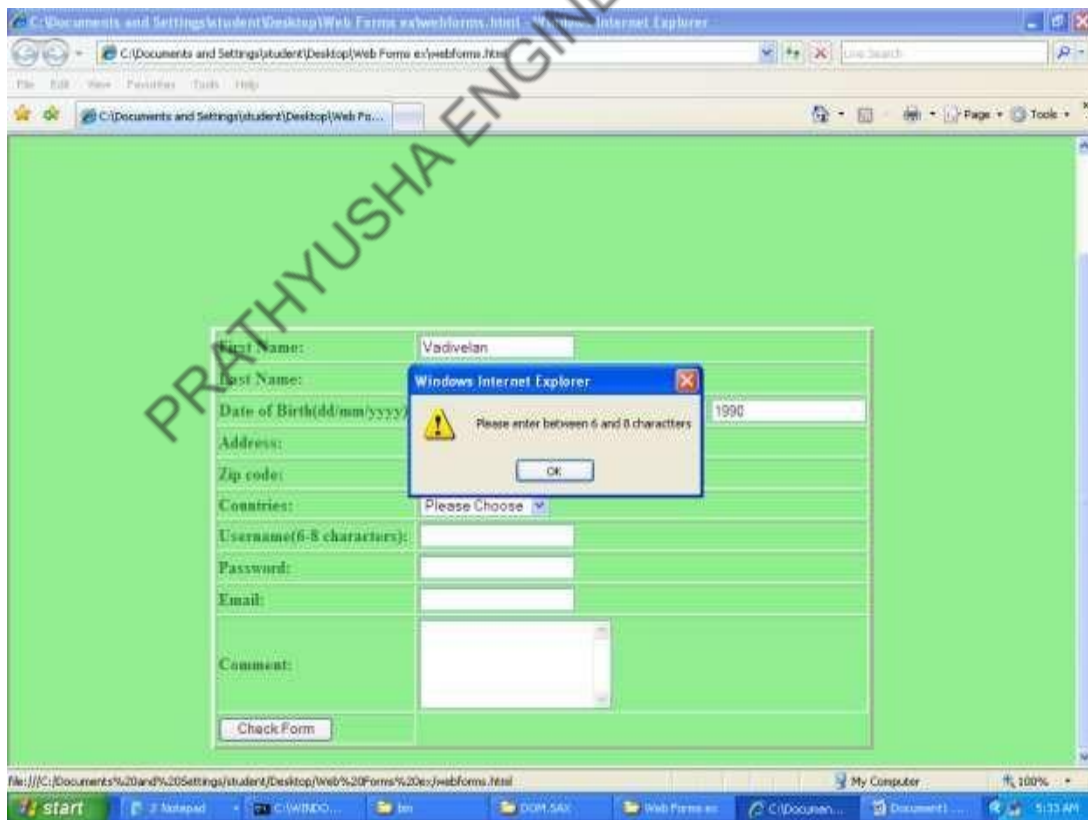
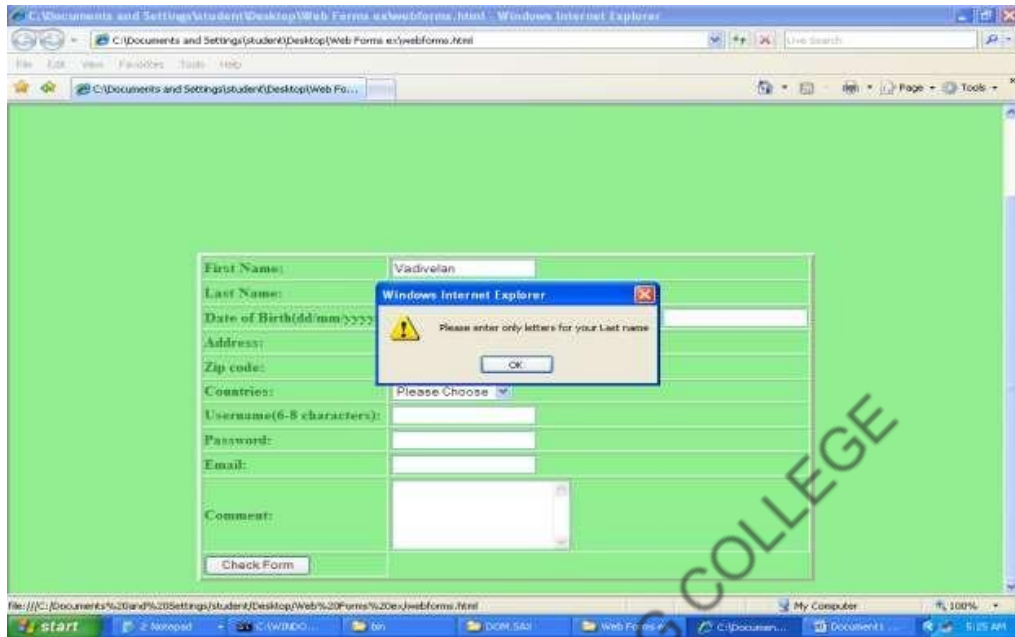
        color="#347235">Countries:</font></b></td><td><selectid='Countries'>
<option>Please Choose</option>
<option value="United Kingdom">United Kingdom</option>
<option value="Afghanistan">Afghanistan</option>
<option value="America">America</option>
<option value="India">India</option>
<option value="Tanzania">Tanzania</option>
<option value="Zimbabwe">Zimbabwe</option>
<option value="Switzerland">Switzerland</option>
</select></td></tr><br />
        <tr><td><b><font color="#347235">Username(6-8
characters):</font></b></td><td><input type='text' id='username'
/></td></tr><br />
        <tr><td><b><font
color="#347235">Password:</font></b></td><td><inputtype='password'
id='password' /></td></tr><br />
        <tr><td><b><font
color="#347235">Email:</font></b></td><td><inputtype='text'
id='email' /></td></tr><br />

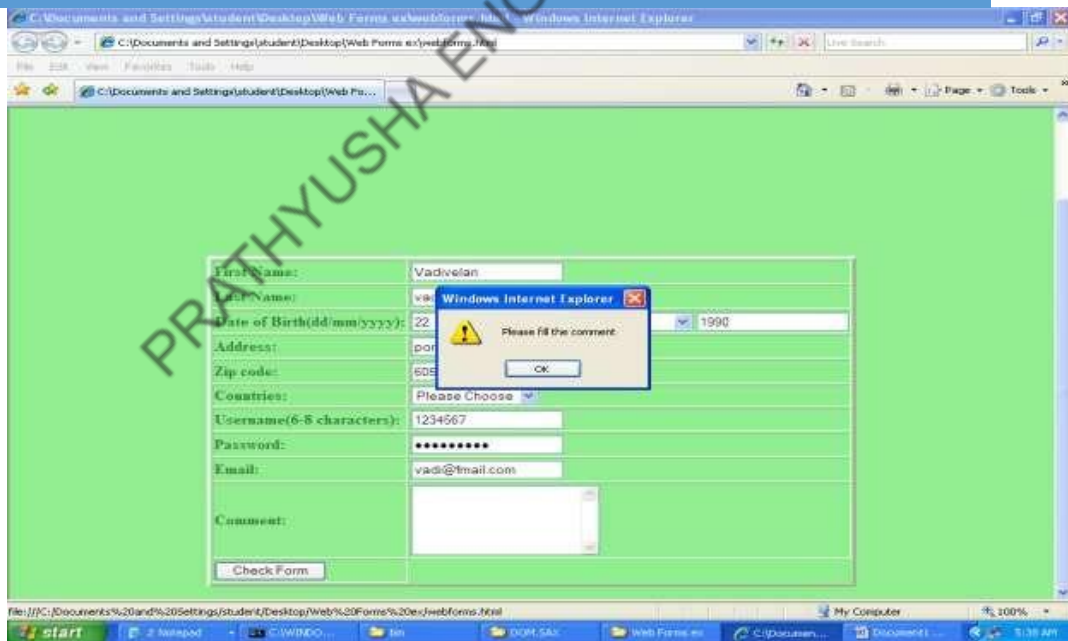
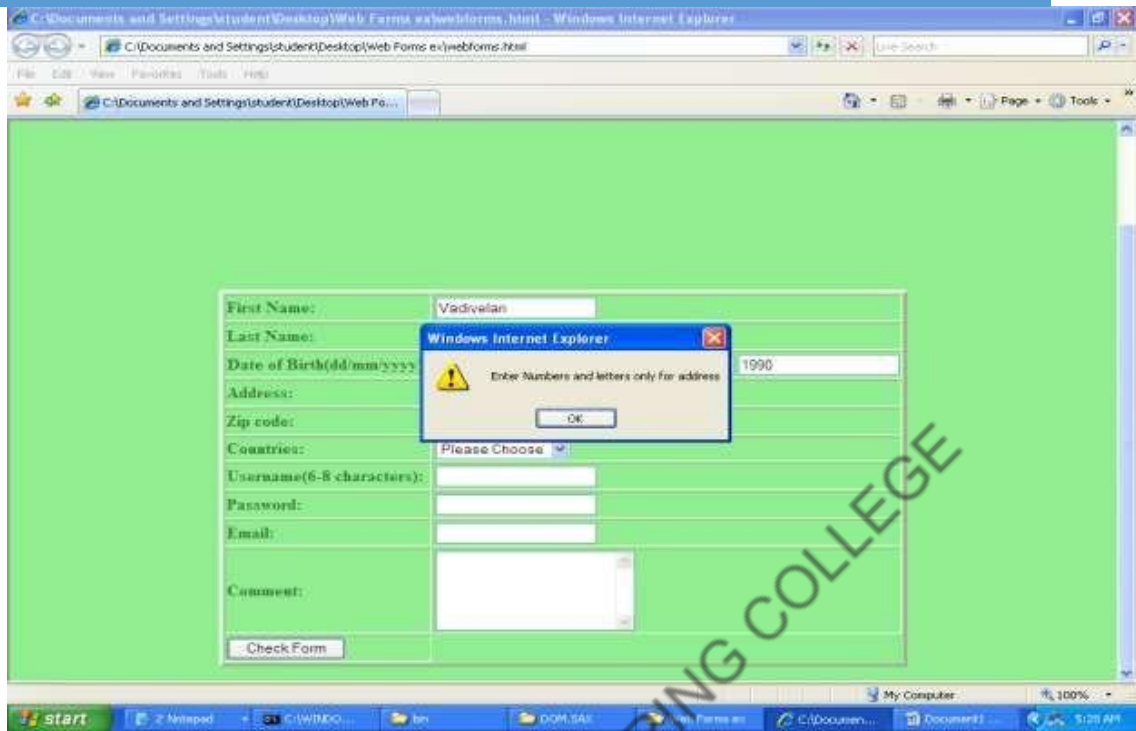
        <tr><td><b><font
color="#347235">Comment:</font></b></td><td><textarea
id='comment'cols="20"rows="5"
name="Address"></textarea></td></tr><br />
<tr><td><input type='submit' value='Check Form' /></td></tr>
</table>
</form>
</body></html>

```



**OUTPUT:**





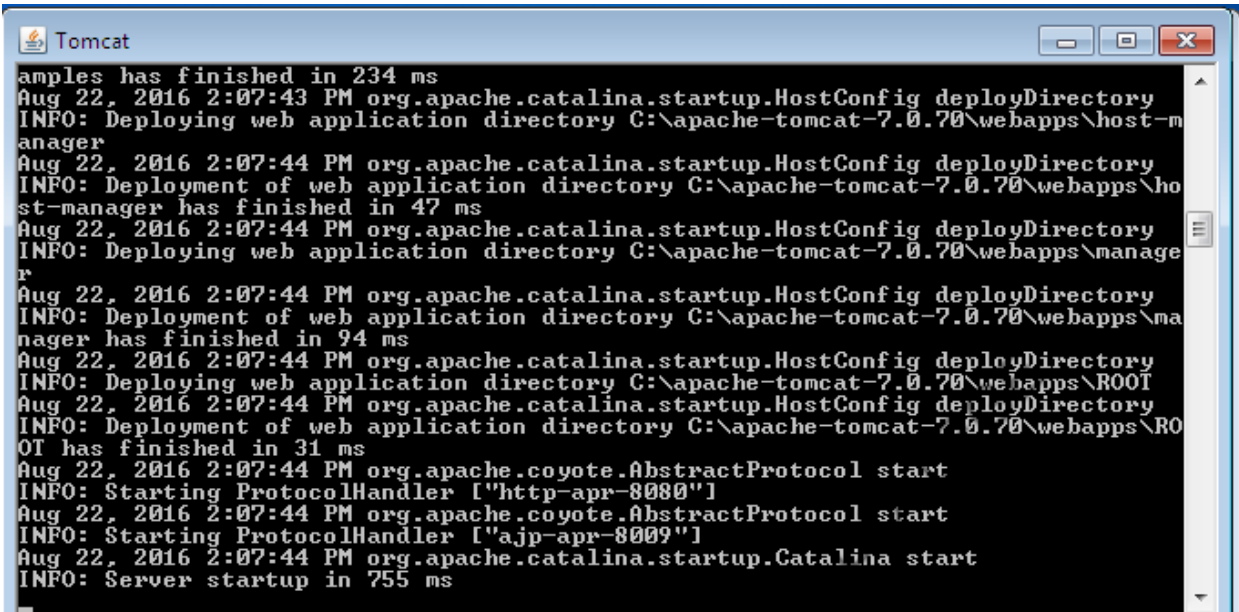
**RESULT :**

Thus developing client side scripts for validating web form controls using DHTML has been verified.

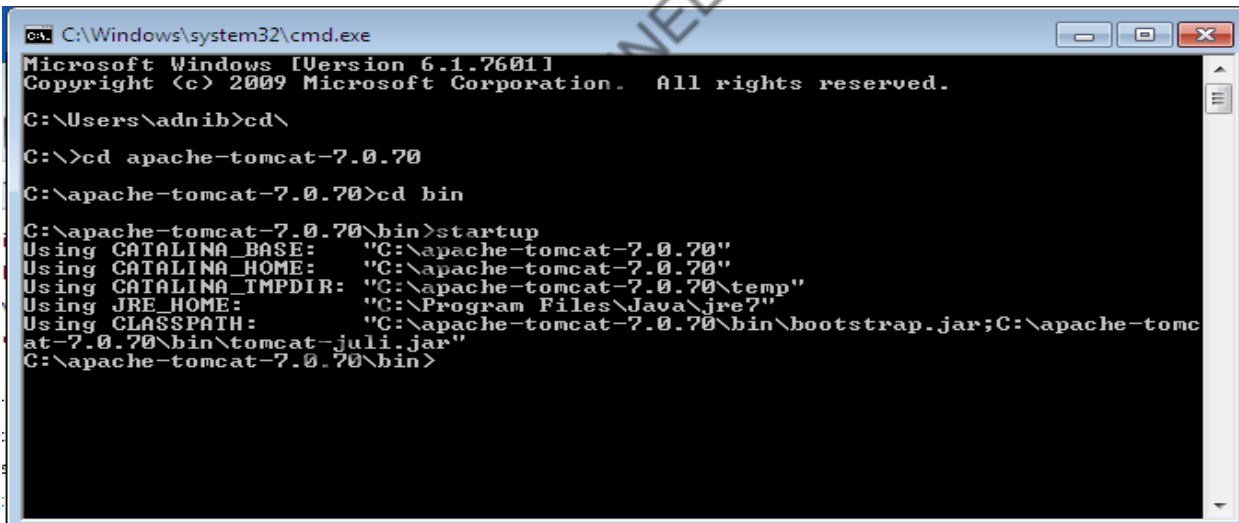


**Step 8:** Try to start Tomcat server by running the "startup" command in a command Line window:

**OUTPUT:**

A screenshot of a Tomcat application window showing a log of startup events. The log includes messages about deploying web applications (host-manager, manager) and starting protocol handlers (http-apr-8080, ajp-apr-8009). The server startup is completed in 755 ms.

```
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
```

A screenshot of a Windows command prompt window showing the steps to start the Tomcat server. The user navigates to the bin directory and runs the startup command, which displays environment variables like CATALINA\_BASE, CATALINA\_HOME, and CLASSPATH.

```
C:\Windows\system32\cmd.exe
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>
```

**RESULT:**

Thus the Tomcat server is installed and configured successfully

**DATE:**

**AIM:**

To write a html program for invoking servlet using html.

**ALGORITHM:**

**Step1:** In html program, define the html, head and title tag.

**Step2:** Then the title is Student Information Form and close the title and head tag.

**Step3:** Define the body tag inside the body tag create form and table simultaneously.

**Step4:** The table consists of following information Roll no, Student name, Address, Phone no and total marks.

**Step5:** In the servlet program, import the summary package and create a own servlet class extends with generic servlet.

**Step6:** In the service method defined to request and response.

**Step7:** Create the object and for print writer and get writer() value.

**Step8:** The enumeration object get the servlet request parameter.

**Step9:** Create objects for string method and it is displayed another object value received get parameter of name received and displayed the value received value.

**PROGRAM:**

**//index.jsp**

```
<html>
<head>
<title>Processing get requests with data</title>
</head>
<body>
    <form action = "Servlet3" method = "get">
        <b><p><label>Enter Your name Please!!
    <br />
    <input type = "text" name = "firstname" />
    <input type = "submit" value = "Submit" />
    </label></p></b>
    </form>
</body>
</html>
```

**//Servlet3.java**

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

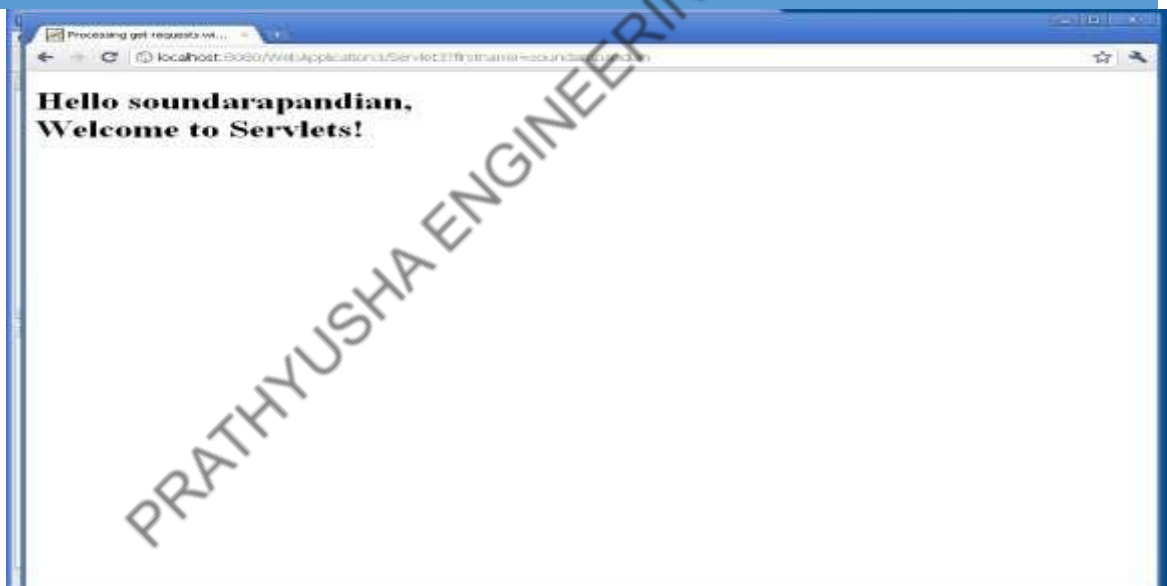
```

public class Servlet3 extends HttpServlet {
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
    String firstName = request.getParameter( "firstname"
);response.setContentType( "text/html" );
    PrintWriter out = response.getWriter();
    // send XHTML document to client
    // start XHTML
    documentout.println(
    "<html>" );
    // head section of
    documentout.println(
    "<head>" );
    out.println("<title>Processing get requests with data</title>"
);out.println( "</head>" );
    // body section of
    documentout.println(
    "<body>" );
    out.println( "<h1>Hello " + firstName + ",<br />"
);out.println( "Welcome to Servlets!</h1>" );
    out.println( "</body>" );
    // end XHTML document
    out.println( "</html>" );
    out.close(); // close stream to complete the page
    }

    public String
    getServletInfo() {
        return "Short
        description";
    }
}

```

**OUTPUT:**



**RESULT:** Thus the invocation of servlet from HTML form has been developed successfully.

**DATE:**

**AIM:**

To write a html program for invoking servlet using html.

**ALGORITHM:**

**STEP 1:** Remove a specific attribute You can delete the value associated with a specific key by

calling the `public void removeAttribute(String name)` function.

**STEP 2:** Delete your whole session. To delete an entire session, use the `public void invalidate()` function.

**STEP 3:** Setting Session Timeout You may set the timeout for a session separately by calling the

`public void setMaxInactiveInterval(int interval)` function.

**STEP 4:** Log the user out On servers that support servlets 2.4, you may use the `logout` method to

log the client out of the Web server and `invalidate` all of the users' sessions.

**STEP 4:** web.xml Configuration If you're using Tomcat, you may set the session timeout in the `web.xml` file, in addition to the ways listed above.

**PROGRAM:**

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

// Extend HttpServlet class
public class GfgSession extends HttpServlet {

    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException
    {

        // Create a session object if it is already not
        // created.
        HttpSession session = request.getSession(true);

        // Get session creation time.
        Date createTime
            = new Date(session.getCreationTime());

        // Get last access time of this web page.
        Date lastAccessTime
            = new Date(session.getLastAccessedTime());

        String title = "Welcome Back to geeksforgeeks";
        Integer visitCount = new Integer(0);
        String visitCountKey = new String("visitCount");
        String userIDKey = new String("userID");
        String userID = new String("GFG");
```



```

// Check if this is new comer on your web page.
if (session.isNew()) {
    title = "Welcome to GeeksForGeeks";
    session.setAttribute(userIDKey, userID);
}
else {
    visitCount = (Integer)session.getAttribute(
        visitCountKey);
    visitCount = visitCount + 1;
    userID
        = (String)session.getAttribute(userIDKey);
}
session.setAttribute(visitCountKey, visitCount);

// Set response content type
response.setContentType("text/html");
PrintWriter out = response.getWriter();

String docType
    = "<!doctype html public "-//w3c//dtd html 4.0 "
      + "transitional//en">\n";

out.println(
    docType + "<html>\n"
    + "<head><title>" + title + "</title></head>\n"
    +
    "<body bgcolor = \"#f0f0f0\">\n"
    + "<h1 align = \"center\">" + title + "</h1>\n"
    + "<h2 align = \"center\">Gfg Session Information</h2>\n"
    + "<table border = \"1\" align = \"center\">\n"
    +
    "<tr bgcolor = \"#949494\">\n"
    + " <th>Session info</th><th>value</th>"
    + "</tr>\n"
    +
    "<tr>\n"
    + " <td>id</td>\n"
    + " <td>" + session.getId() + "</td>"
    + "</tr>\n"
    +
    "<tr>\n"
    + " <td>Creation Time</td>\n"
    + " <td>" + createTime + " </td>"
    + "</tr>\n"
    +
    "<tr>\n"
    + " <td>Time of Last Access</td>\n"

```

```

+ " <td>" + lastAccessTime + "</td>"
+ "</tr>\n"
+
" <tr>\n"
+ " <td>User ID</td>\n"
+ " <td>" + userID + "</td>"
+ "</tr>\n"
+
" <tr>\n"
+ " <td>Number of visits</td>\n"
+ " <td>" + visitCount + "</td>"
+ "</tr>\n"
+ "</table>\n"
+ "</body>"
+ "</html>");
}
}
File: web.xml

```

- XML

```

<web-app>
<servlet>
  <servlet-name>GfgSession</servlet-name>
  <servlet-class>GfgSession</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>GfgSession</servlet-name>
  <url-pattern>/GfgSession</url-pattern>
</servlet-mapping>
</web-app>

```

Compile the servlet SessionTrack described above and add it to the web.xml file. When you run <http://localhost:8080/SessionTrackingGfg/GfgSession> for the first time, you should get the following result:

**Output:**



If we try to run the same servlet again, we will get the following result.



**RESULT:**

Thus the invocation of servlet in session tracking been developed successfully.

**EX.NO : 6 A**

**ONLINE EXAMINATION**

**DATE:**

**AIM:**

To write a java servlet program to conduct online examination and to display studentmark list available in a database.

**ALGORITHM:**

**Step1:** Create a html file with form tag.

**Step2:** The form tag action="http://localhost:8080/example/servlet/exam"

**Step3:** Create a two textbox(name & seat number).

**Step4:** The 5 question are defined into true or false model and close the all tags.

**Step5:** Import the necessary packages and declare class, class name in exam.

**Step6:** Declare the connection, statement and result set object.

**Step7:** Use the deposit () for check the connection in JDBC:ODBC driver.

**Step8:** The data are inserting into corresponding table.

**Step9:** The execute update () are update the database.

**Step10:** Display the table in after html file compilation.

**PROGRAM:**

**//index.jsp**

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Welcome to Online Examination!!!!</title>
</head>
<body>
Welcome to Online Examination!!!!
<form action="exam" method="get">
    <label><p> Enter Your name Please!!<br/> <input
    type="text" name="name"/>
<br/>
<input type="submit" name="SUBMIT"/>
</p></label>
</form>
</body>
</html>
```

```

//exam.java
import java.io.IOException;import
java.io.PrintWriter;
import javax.servlet.ServletException; import
javax.servlet.annotation.WebServlet;import
javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest; import
javax.servlet.http.HttpServletResponse;public class
exam extends HttpServlet {
    protected void doGet(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out=response.getWriter();
    String
    name=request.getParameter("name");
    out.println("<html>");
        out.println("<head>");
        out.println("<title>Online Examination</title>");
        out.println("</head>");
        out.println("<body bgcolor=PINK>");
        out.println("<h2 align=center>Online Examination</h2><hr>");
        out.println("<h3 align=center> Welcome
Mr."+name+"</h3><hr>");out.println("<h4><u>Terms and
Conditions:</u></h4>"); out.println("<ul type=disc>");
        out.println("<li>The Paper consists a set of five questions.</li>");
        out.println("<li>Every question consists of two options.</li>");
        out.println("<li>All must be answered</li></ul><hr>");
        out.println("<center><h5><u>Your
Questions</u></h5></center>");out.println("<hr>");
        out.println("<form method=get action=exam2>");
        out.println("<p>1.Operating System is a ..... </p>");
        out.println("<input type=radio name=q1
value=0>Hardware");out.println("<br>");
        out.println("<input type=radio name=q1
value=1>Software");out.println("<hr>");
        out.println("<p>2.Developer of C Language is ..... </p>");
        out.println("<br>");
        out.println("<input type=radio name=q2 value=0>Dennis
Richee");out.println("<br>");
        out.println("<input type=radio name=q2 value=1>James
Thompson");out.println("<hr>");
        out.println("<p>3.Which of the following is a multitasking,multi
user,multiprocessing);
        out.println("<p>OS..... </p>");
        out.println("<br>");
        out.println("<input type=radio name=q3 value=0>MS
DOS");out.println("<br>");
        out.println("<input type=radio name=q3 value=1>Windows
NT");out.println("<hr>");

```

```

        out.println("<p>4.Father of Computers is..... </p>");
        out.println("<br>");
        out.println("<input type=radio name=q4 value=1>Charles
        babbage");out.println("<br>");
        out.println("<input type=radio name=q4 value=0>Charles
        Dickson");out.println("<hr>");
        out.println("<p>5.What is the current generation of computers
        ?</p>");out.println("<br>");
        out.println("<input type=radio name=q5
        value=0>Fifth");out.println("<br>");
        out.println("<input type=radio name=q5
        value=1>Sixth");out.println("<hr>");
        out.println("<input type=submit
        value=Done>");out.println("</form>");
        out.println("</body>");
        out.println("</html>");
    }
    public String getServletInfo()
    { return "A Servlet of the
    user";
    }
}

```

**//exam2.java**

```

import
java.io.IOException;import
java.io.PrintWriter;
import javax.servlet.ServletException;
import
javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import
javax.servlet.http.HttpServletResponse;public
class exam2 extends HttpServlet {
    protected void doGet(HttpServletRequest request,
    HttpServletResponse
    response)
        throws ServletException,
        IOException {int count=0;j;
        response.setContentType("text/html
        ");PrintWriter
        out=response.getWriter(); String
        q1=request.getParameter("q1");
        String
        q2=request.getParameter("q2");
        String

```

```

q3=request.getParameter("q3");
String
q4=request.getParameter("q4");
String
q5=request.getParameter("q5");
if(q1.equals("1"))
{
    count=count+1;
}
if(q2.equals("1"))
{
    count=count+1;
}
if(q3.equals("1"))
{
    count=count+1;
}
if(q4.equals("1"))
{
    count=count+1;
}
if(q5.equals("1"))
{
    count=count+1;
}

out.println("<html>");
out.println("<head><title>Examination
Results</title></head>");out.println("<body>");
out.println("<h2 align=center>Online Examination</h2><hr>");
out.println("<h3>Number of Questions answered
correctly:</h3>"+count);if(count>=3)
{
out.println("<hr><h3>Congrats!!! You Have
Passed!!!</h3><hr>");
out.println("<h4><b>Try Other Tests!!</b></h4>");
}

else
{
out.println("<hr><h3>Sorry!!! You Have
Failed!!!</h3><hr>");out.println("<h4><b>Try
Again:</b></h4>");
}

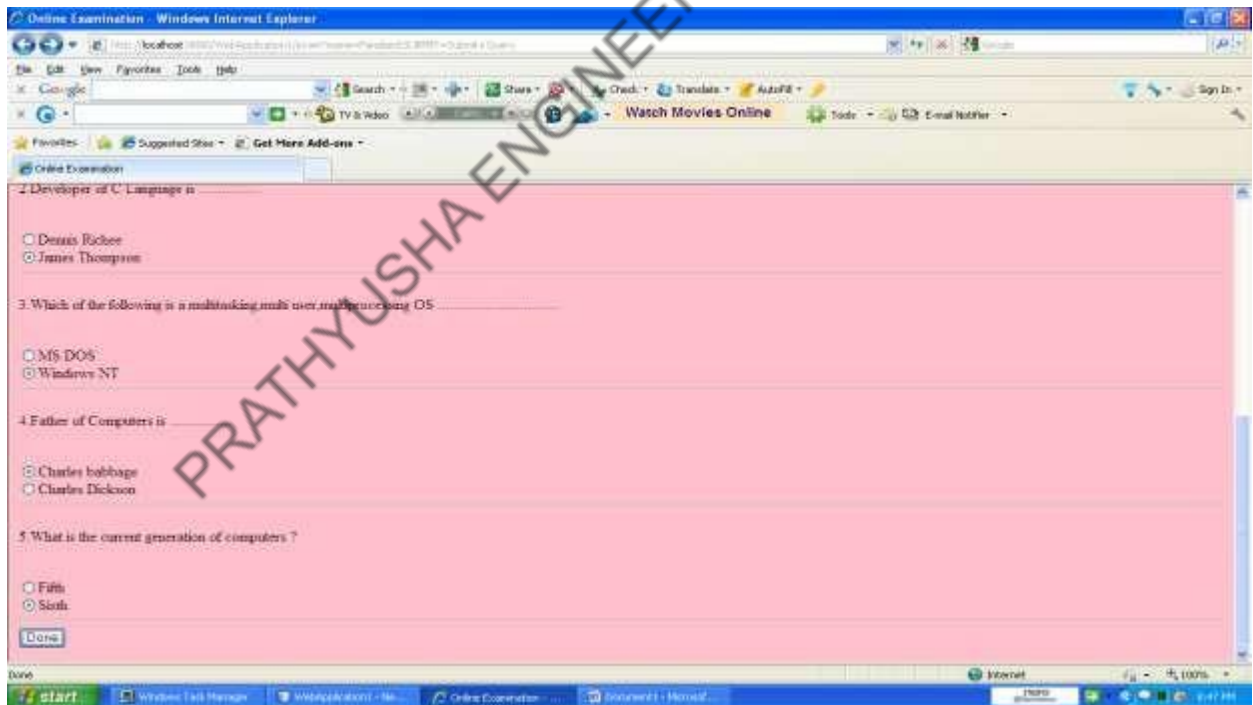
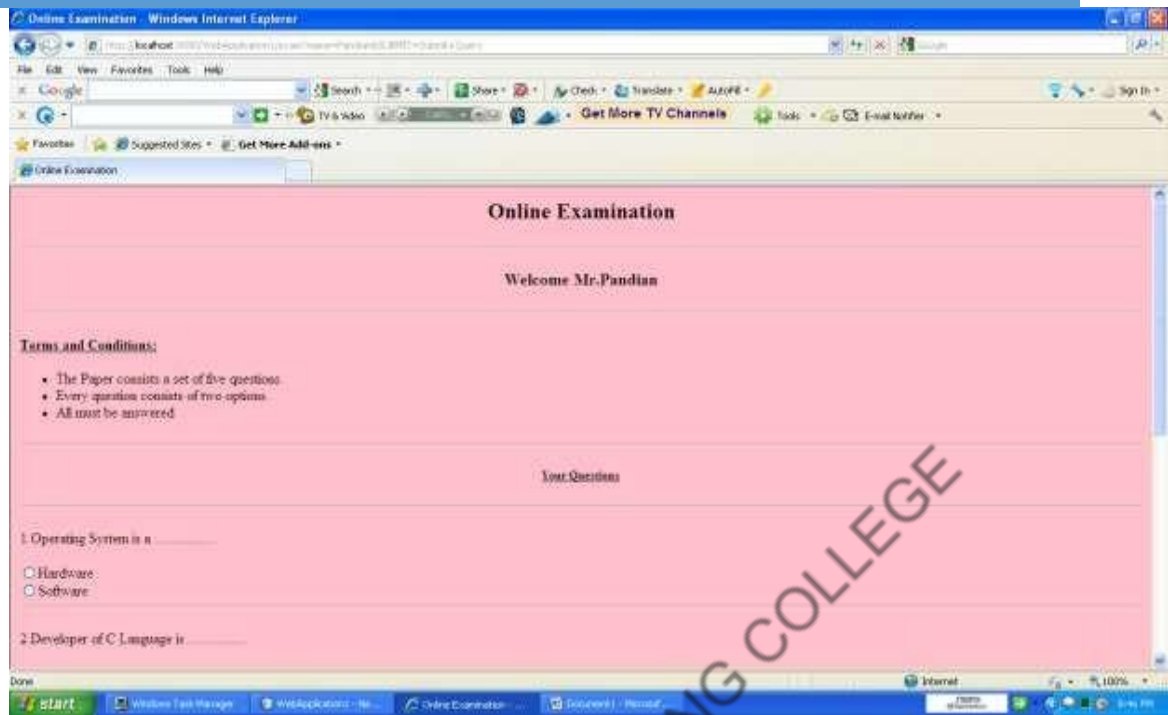
```

```
        out.println("</body>");
        out.println("</html>");
    }
    public String getServletInfo() {
        return "A Servlet of the
        User";
    }
}
```

### **OUTPUT:**







**RESULT:** Thus the development of program in java to create three tire application using servlet has been verified successfully.

**EX.NO : 6 B**

**DISPLAYING STUDENT MARKLIST USING JSP**

**DATE:**

**AIM:**

To create a three tier application for displaying student mark list using JSP and database.

**ALGORITHM:**

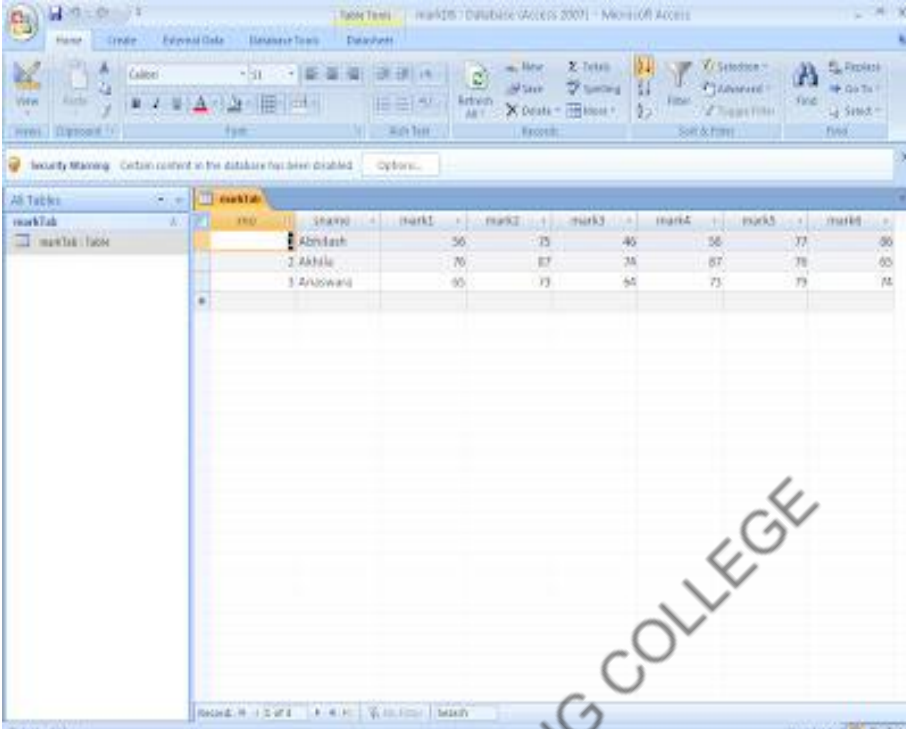
1. Design the HTML page (stud.html) with the following
  - a) Create a form to get the input (Register Number) from the user.
  - b) Set the URL of the server (marklist.jsp) as the value of the action attribute.
  - c) Use submit button to invoke the server and send the form data to the server.
2. Create the JSP file with the following
  - a) Read the parameter value (Register Number) from the form by using the method `getParameter()`.
  - b) Server retrieves the details from the database table with respect to the form input.
  - c) Server displays the mark list to the client as the response.

**marklist.jsp:**

```
<% @ page contentType="text/html" language="java" import="java.sql.*"%>
<html>
<head>
<title>Three Tier Application</title>
<style type="text/css">
  body { color:blue;font-family:courier;text-align:center}
</style></head><body>
<h2>EXAMINATION RESULT</h2><hr/>
<%
String str=request.getParameter("regno");
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con=DriverManager.getConnection("jdbc:odbc:markDS");
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("SELECT*FROM markTab WHERE rno="+str);
while(rs.next())
{
%>
Register No:<%=rs.getObject(1)%><br/>
Name:<%=rs.getObject(2)%><br/>
<table border="1">
<th>SUBJECT</th><th>Mark</th>
<tr><td>Network Programming and Management</td><td><%=rs.getObject(3)%></td></tr>
<tr><td>Object Oriented Analysis and Design</td><td><%=rs.getObject(4)%></td></tr>
<tr><td>Cryptography and Network Security</td><td><%=rs.getObject(5)%></td></tr>
<tr><td>Embedded Systems</td><td><%=rs.getObject(6)%></td></tr>
<tr><td>Web Technology</td><td><%=rs.getObject(7)%></td></tr>
<tr><td>Software Requirement and Engineering</td><td><%=rs.getObject(8)%></td></tr>
</table>
<% }
%>
<br/>
<a href="stud.html">Back</a>
</body></html>
```



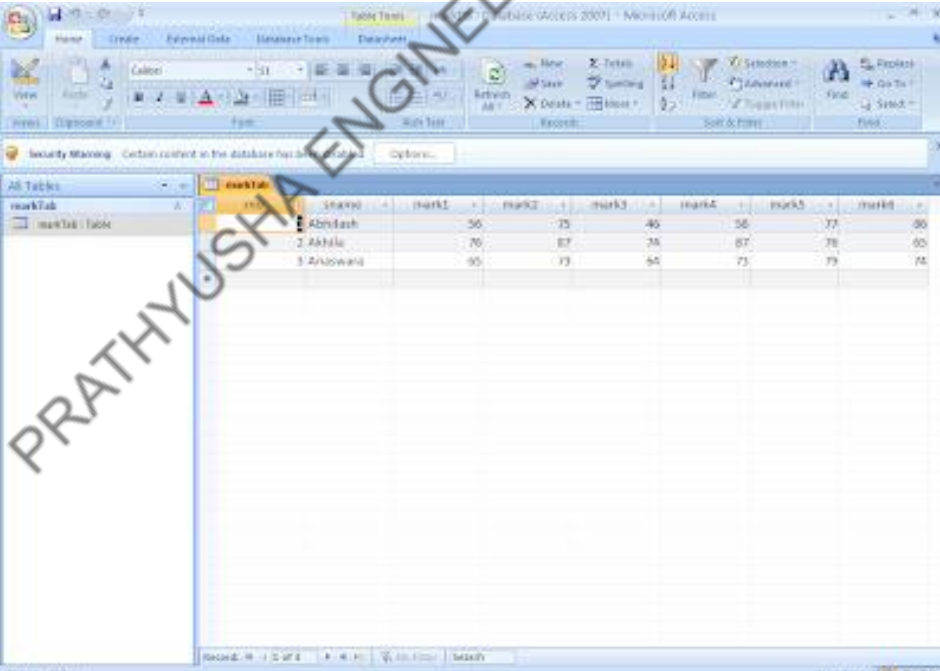
## OUTPUT SCREENSHOTS



Security Warning: Certain content in the database has been disabled. Options...

id	name	mark1	mark2	mark3	mark4	mark5	mark6
1	Abhishah	56	75	46	56	77	80
2	Akhile	70	87	24	87	78	65
3	Anuswand	95	73	64	73	79	74

Record: 1 of 3 | Page: 1 of 1 | Search



Security Warning: Certain content in the database has been disabled. Options...

id	name	mark1	mark2	mark3	mark4	mark5	mark6
1	Abhishah	56	75	46	56	77	80
2	Akhile	70	87	24	87	78	65
3	Anuswand	95	73	64	73	79	74

Record: 1 of 3 | Page: 1 of 1 | Search

**RESULT:** Thus the creation of a three tier application for displaying student mark list using JSP and database has been verified successfully.

**EX.NO : 7**

## **XML SCHEMA FOR STUDENT DETAILS**

**DATE:**

### **AIM:**

To write a program for implementing student information using XML & XSL.

### **ALGORITHM:**

**Step1:**The XML document reference to the XSL document.

**Step2:** The create the student information in the student tag and insert the same information about the student.

**Step3:**Close all opened tags.

**Step4:**In XSL document create a html file include the student information in table format.

**Step5:**Close the necessary tags.

### **PROGRAM:**

**//student.xml**

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="student.css"?>
<!DOCTYPE student SYSTEM "student.dtd">
<students>
<student>
<sno>801041</sno>
<sname>S.Soundarapandian</sname>
<dob>05/081991</dob>
<address>Neyveli</address>
<m1>80</m1>
<m2>90</m2>
<m3>95</m3>
</student>
<student>
<sno>801049</sno>
<sname>R.Vadivelan</sname>
<dob>22/07/1990</dob>
<address>Pondicherry</address>
<m1>90</m1>
<m2>95</m2>
<m3>80</m3>
</student>
<student>
<sno>801037</sno>
<sname>R.Satheesh</sname>
<dob>21/01/1991</dob>
<address>Kanyakumari</address>
<m1>80</m1>
<m2>90</m2>
<m3>95</m3>
</student>
</students>
```

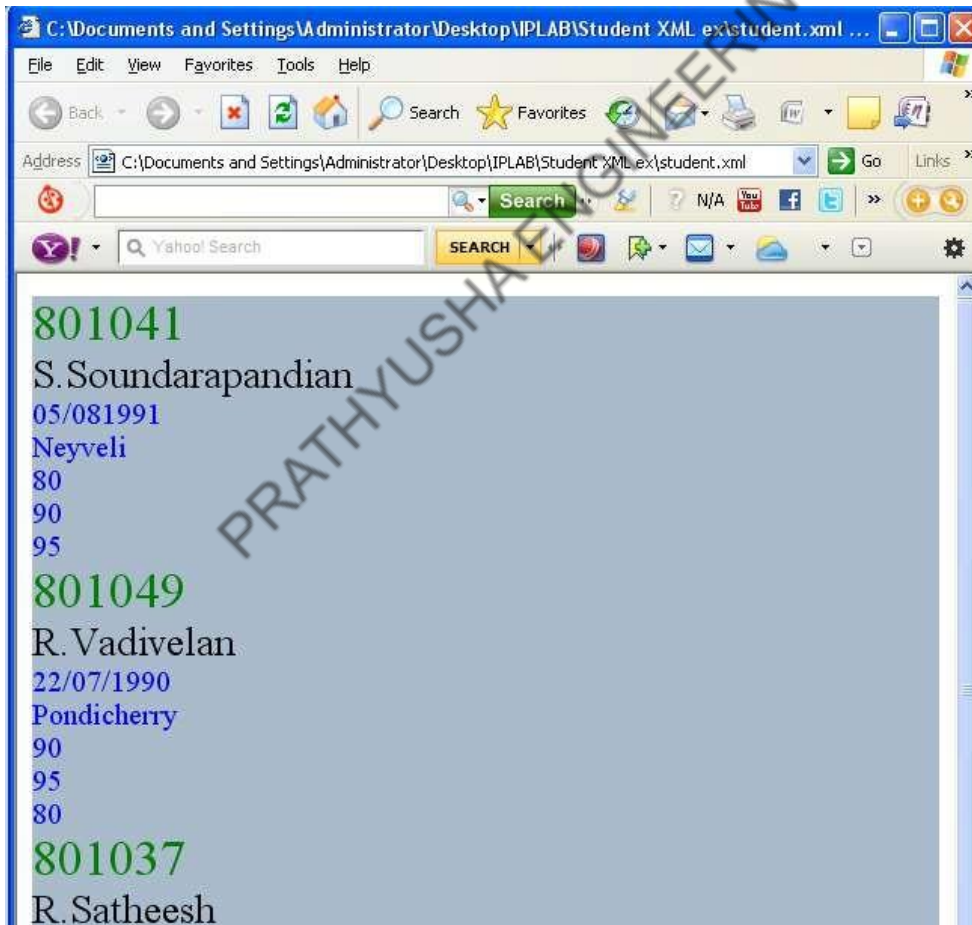
### //student.css

```
Student { background.color:#aabbcc;width:100%; } Sno {
display:block; color:GREEN; font.size:25pt; } Sname {
display:block; color:BLACK; font.size:20pt; } Dob {
display:block; color:BLUE; font.size:15pt; } Address {
display:block; color:BLUE; font.size:15pt; } m1 {
display:block; color:BLUE; font.size:15pt; }
m2 { display:block; color:BLUE; font.size:15pt; } m3 {
display:block; color:BLUE; font.size:15pt;}
```

### //student.dtd`

```
<?xml version="1.0"?>
<!ELEMENT students (student+)>
<!ELEMENT student (sno,sname,dob,address,m1,m2,m3)>
<!ELEMENT sno (#PCDATA)>
<!ELEMENT sname (#PCDATA)>
<!ELEMENT dob (#PCDATA)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT m1 (#PCDATA)>
<!ELEMENT m2 (#PCDATA)>
<!ELEMENT m3 (#PCDATA)>
```

### OUTPUT:



**RESULT:** Thus the creation of XSL document using Xml has been verified successfully