KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed University Established Under Section 3 of UGC Act, 1956)

Coimbatore - 641 021, India

FACULTY OF ARTS, SCIENCE AND HUMANITIES (FASH)

Department of CS,CA & IT

I M.Sc CS **I SEMESTER** **BATCH : 2017- 2019**

17CSP111 **WEB TECHNOLOGY – PRACTICAL** **4H – 2C**

Instruction Hours / week: L: 0 T: 0 P: 4 **Marks:** Int : **40** Ext : **60** Total: **100**

Course Objective

- To build web applications using ASP.NET and client side script technologies use with Microsoft's IIS.
- To build XML applications with DTD and style sheets.
- Students will gain the skills and project-based experience needed for entry into web application and development careers

Course Learning Outcome

- Students are able to develop a dynamic webpage by the use of java script.
- Students will be able to write a well formed / valid XML document.
- Students will be able to connect a java program to a DBMS and perform insert, update and delete operations on DBMS table.

Practical List

1. Using Java script change the font color on reloading a webpage.
2. Generate web page that represents clock-every 60 see the page updated with server current time Using JavaScript.
3. Design a form and validate it using JavaScript.
4. Write Database Access program using ASP.NET
5. Program to retrieve Cookies information using ASP.NET
6. Program to count web page hits using ASP.NET
7. Create a menu in XML.
8. Create a demo for XSLT.
9. Display XML information in Tree structure format.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed University Established Under Section 3 of UGC Act, 1956)

Coimbatore - 641 021, India

FACULTY OF ARTS, SCIENCE AND HUMANITIES (FASH)

Department of CS,CA & IT

I M.Sc CS I SEMESTER BATCH : 2017- 2019

WEB TECHNOLOGY – PRACTICAL (17CSP111)

Practical List

1. Using Java script change the font color on reloading a webpage.
2. Generate web page that represents clock-every 60 see the page updated with server current time Using JavaScript.
3. Design a form and validate it using JavaScript.
4. Write Database Access program using ASP.NET
5. Program to retrieve Cookies information using ASP.NET
6. Program to count web page hits using ASP.NET
7. Create a menu in XML.
8. Create a demo for XSLT.
9. Display XML information in Tree structure format.

CHANGING FONT COLOR

EX.NO : 1

AIM:

To create a program using JavaScript to change the font color on reloading a web page.

ALGORITHM:

STEP 1 : Start the process.

STEP 2 : Create the html tag and set language as Java Script.

STEP 3 : Declare and initialize array variable for color.

STEP 4 : Create the function changecolor().

STEP 5 : In function definition calculate the length of the color using length() method .

STEP 6 : To change the color using math.round(math.random()*1000) method.

STEP 7 : Using if statement change the font color of web page on reloading.

STEP 8 : call the function changecolor() and print the statement using document.Write()

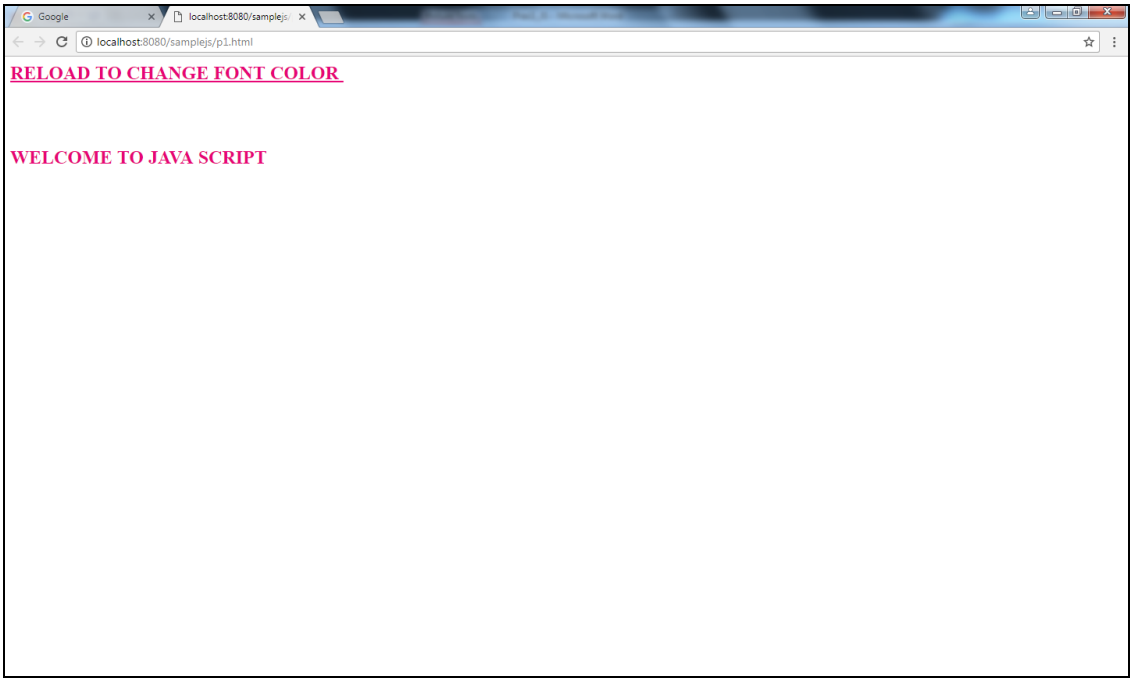
STEP 9 : Stop the process.

CODING:

```
<html>
<head>
<script language="javascript">
var hit=false
color=new Array(10)
functionchangecolor()
{
for(var i=0; i<color.length;i++)
{
while(hit==false)
{
start=(Math.round(Math.random()*1000))
if(start>255)
{
hit==false
continue
}
color[i]=start
hit=true
}
hit=false
}
x=color[0].toString(16)
if(x.length<2)
```

```
{
x=("0"+x)
}
y=color[1].toString(16)
if(y.length<2)
{
y=("0"+y)
}
z=color[2].toString(16)
if(z.length<2)
{
z=("0"+z)
}
return(' " ' + "#"  +x+y+z+  ' " ');
}
document.write("<font color="  +changecolor() +" ><h2><b><u> RELOAD TO
CHANGE FONT COLOR </u></b><BR/><BR/><BR/><BR/> WELCOME TO JAVA
SCRIPT <BR/></h2></font>");
</script>
</head>
</html>
```

OUTPUT:



RESULT:

Thus the above program has been executed successfully and the output was verified.

DISPLAYING DATE AND TIME

EX.NO : 2

AIM:

To write a program to displaying the system clock using JavaScript.

ALGORITHM:

STEP 1 : Start the process.

STEP 2 : Create the html tag and set language as Java Script.

STEP 3 : To create the function diplay_c() and display_ct().

STEP 4 : In function definition to get current date and time using refresh variable.

STEP 5 : Call the function display ct() in onload method.

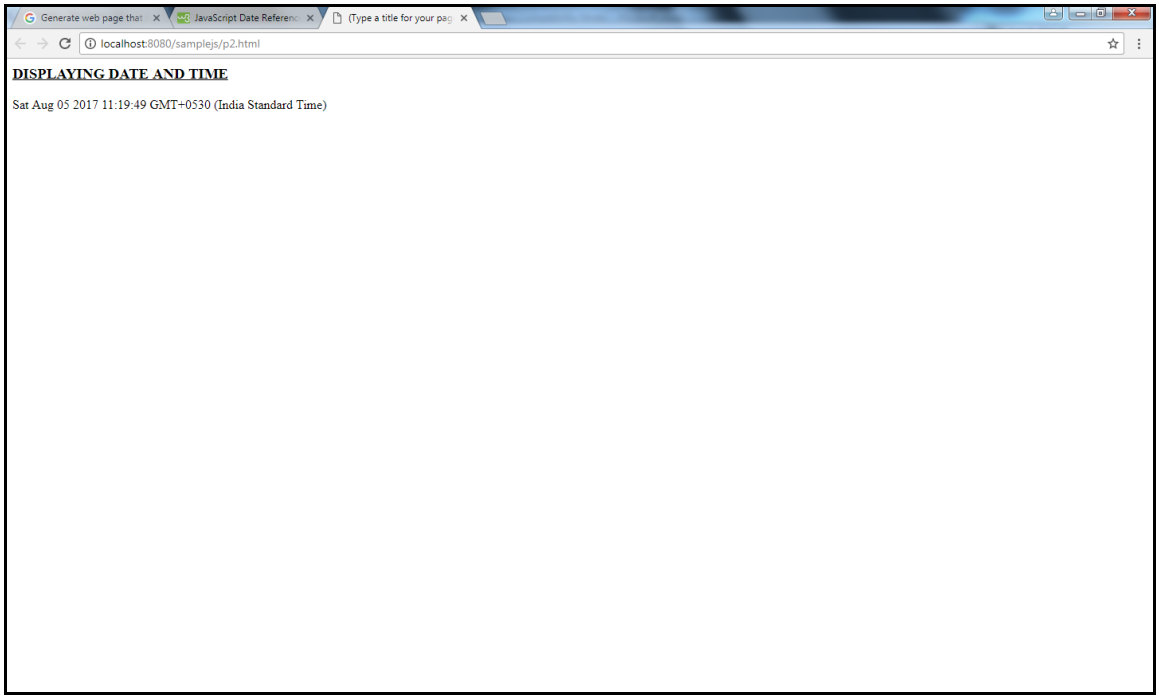
STEP 6 : Save and execute the program.

STEP 7 : Stop the process.

CODING:

```
<html>

<head>

<script type="text/javascript">
functiondisplay_c()
{
var refresh=1000;
mytime=setTimeout('display_ct()',refresh)
}

functiondisplay_ct()
 {
varstrcount
var x = new Date()
document.getElementById('ct').innerHTML = x;
tt=display_c();
}

</script>

</head>

<H3><B><U>DISPLAYING DATE AND TIME</B></U></H3>

<body onload=display_ct();>

<span id='ct' ></span>

</body>

</html>
```

OUTPUT:



RESULT:

Thus the above program has been executed successfully and the output was verified.

FORM VALIDATION

EX.NO : 3

AIM:

To design a form and validate it using JAVASCRIPT.

ALGORITHM:

STEP 1 : Start the process.

STEP 2 : Create the html tag and set language as Java Script.

STEP 3 : To create the function alpha(),num() and validate().

STEP 4 : In function definition, to validate the form using a=/^[a-z A-Z]+$1 for alpha , n=/^[0-9]+$1 for num and email validation to check indexOf(@).

STEP 6 : Inside the body tag to design the form and call the functions.

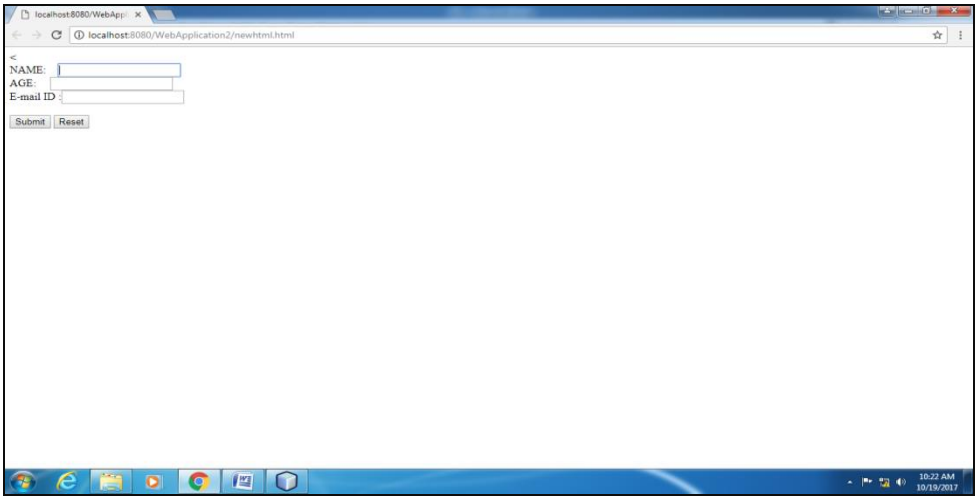
STEP 7 : Save and execute the program.

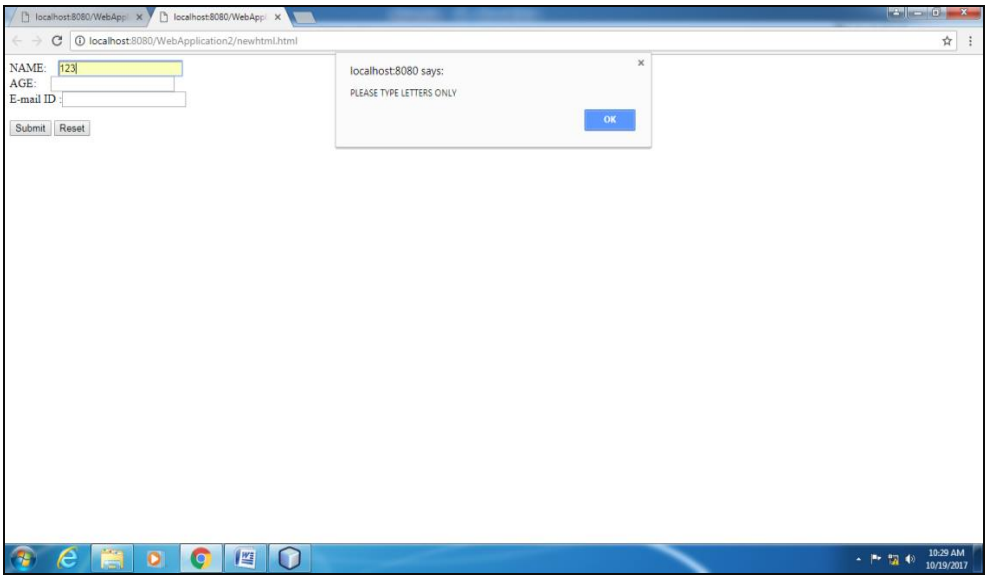
CODING:

```
<html>
<script type="text/javascript">
function alpha(e,msg){
var a=/^[a-zA-Z]+$/;
if(e.value.match(a))
{
return true;
}
else
{
alert(msg);
e.focus();
return false;
}
}
functionnum(e,msg)
{
var n=/^[0-9]+$/;
if(e.value.match(n))
{
return true;
}
else
{
alert(msg);
e.focus();
return false;
}
}
function validate()
{
var at=document.getElementById("email").value.indexOf("@");
submitOK="true";

if(at==-1)
```

```
{
alert("Not a valid e-mail!");
submitOK=="false";
}
if(submitOK=="false")
{
return false;
}
}
</script>
<form action="p3.html" onsubmit="return validate()">
 NAME: &amp;nbsp;&amp;nbsp;&amp;nbsp;<input type='text' id='letters' size="20"
onchange="alpha(document.getElementById('letters'),'PLEASE TYPE LETTERS ONLY')"/>
<BR>AGE: &amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;<input type='text' id='numbers' size="20"
onchange="num(document.getElementById('numbers'),'PLEASE TYPE NUMBERS ONLY')"/>

<br>
E-mail ID :<input type="text" id="email" size="20"><br/>
<br/>
<input type="submit" value="Submit">
<input type="Reset" value="Reset">
</body>
</html>
```

OUTPUT:





RESULT:

Thus the above program has been executed successfully and the output was verified.

RETRIVE COOKIES INFORMATION

EX.NO : 4

AIM:

To write a program for retrieve cookies information using Asp.net

ALGORITHM:

STEP 1 : Start the process.

STEP 2 : In default asp.net file, to design the form using four text boxes as book name, author name, publisher name and due date.

STEP 3 : Create two buttons of add and save.

STEP 4 : Click the add button and write coding for book details.

STEP 5 : Click the save button and redirect the ASP.net file.

STEP 6 : In default 2.asp file , declare the necessary variables.

STEP 7 : Using request, response object for retrieve the cookies information.

STEP 8 : Save and execute the program.

STEP 9 : Stop the process.

CODING:

Default.aspx.vb

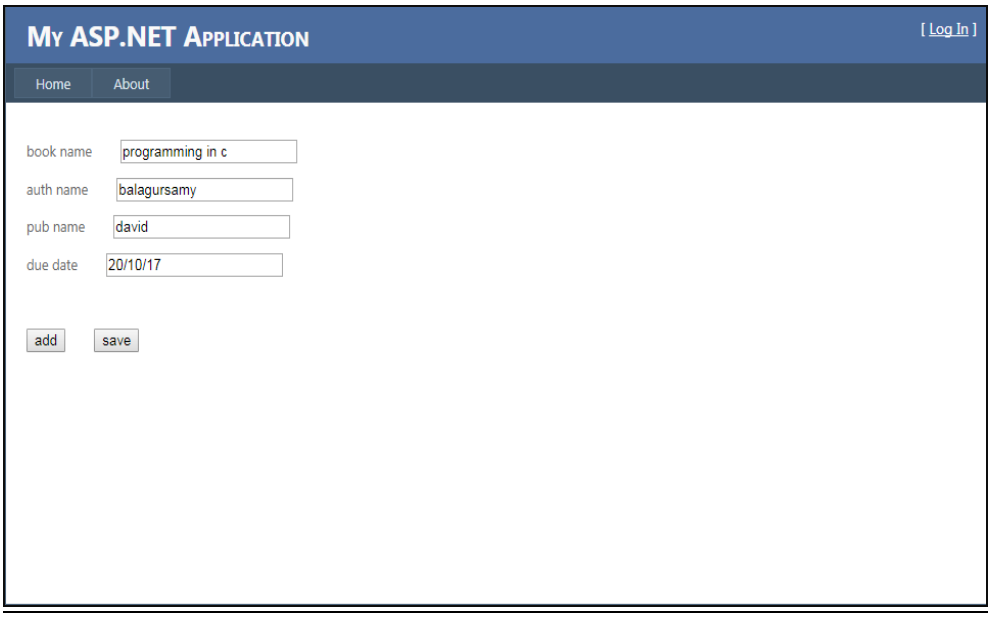
```
Partial Class _Default
   Inherits System.Web.UI.Page
 Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
      Response.Cookies("book")("Book.Name") = TextBox1.Text
      Response.Cookies("book")("Au.Name") = TextBox2.Text
      Response.Cookies("book")("Publisher") = TextBox3.Text
      Response.Cookies("book")("Due date") = TextBox4.Text
      Response.Cookies("book").Expires = Now.AddDays(1)
   End Sub
Protected Sub Button2_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button2.Click
      Response.Redirect("Default2.aspx")
   End Sub
End Class
```

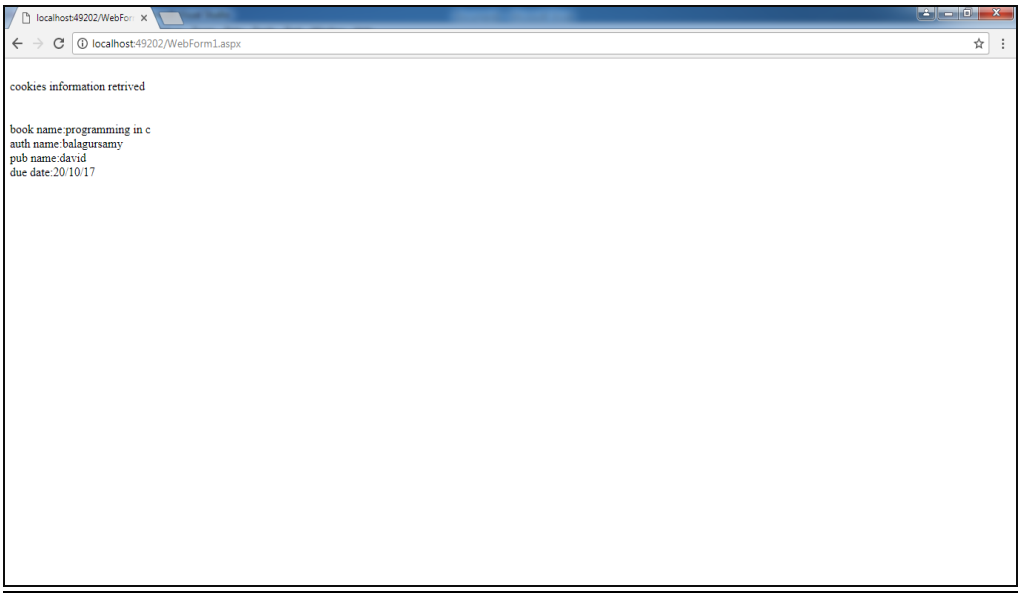
WebForm1.aspx.vb

```
Partial Class Default2
   Inherits System.Web.UI.Page
  Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
      Dim cook As String
      Dim an As String
      Dim pub As String
      Dim exp As String
```

```
        Response.Write("<br>")
        Response.Write("Cookies information retrived")
        Response.Write("<br>")
        Response.Write("</br>")
        If Not Request.Cookies("book")("Book.Name") Is Nothing Then
            Cook = Request.Cookies("book")("Book.Name")
            an = Request.Cookies("book")("Au.Name")
            pub = Request.Cookies("book")("Publisher")
            exp = Request.Cookies("book")("Due date")
            Response.Write("<br>")
            Response.Write("BookName:" & Cook)
            Response.Write("<br>")
            Response.Write("Au.Name:" & an)
            Response.Write("<br>")
            Response.Write("Publisher:" & pub)
            Response.Write("<br>")
            Response.Write("Due date:" & exp)
            Response.Write("<br>")
        End If
    End Sub
End Class
```

OUTPUT:





RESULT:

Thus the above program has been executed successfully and the output was verified.

WEBPAGE HITS

EX.NO:5

AIM:

To write program for to count Webpage hits using Asp.net

ALGORTHM:

STEP 1 : Start the process.

STEP 2 : To create a form and insert three textboxes as pp,ss, pvs.

STEP 3 : On page load assign the Application varibles to respective textboxes.

STEP 4 : To right click the file name and open file name->Add->New item->Global.aspx.

STEP 5 : And write the coding for Application. Start.,Session. Start.,Session. End. and Application_BeginRequest. to count Webpage hits.

STEP 6 : Save and execute the program.

STEP 7 : Stop the process.

CODING:

Default.aspx

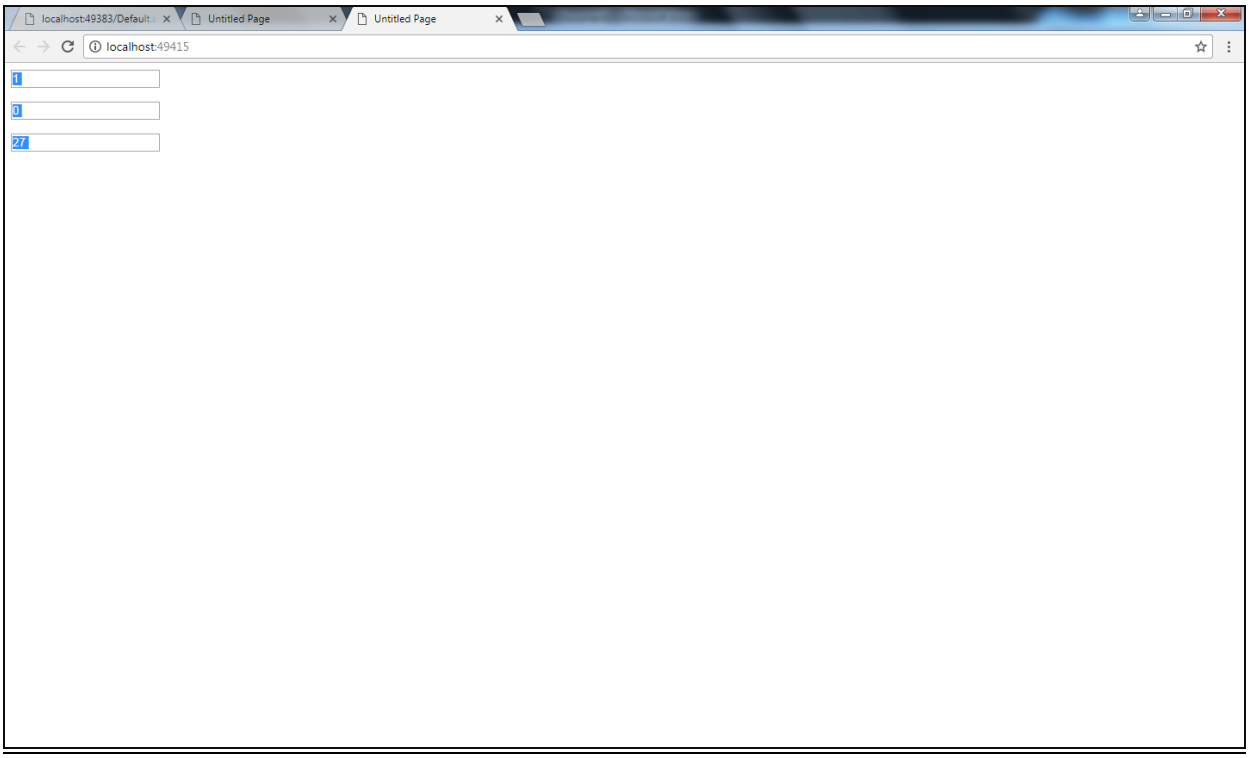
```
Partial Public Class _Default
   Inherits System.Web.UI.Page

   Protected Sub Page_Load(ByVal sender As Object, ByVal e As System. EventArgs) Handles
Me.Load
      TextBox1.Text = Application("PP")
      TextBox2.Text = Application("SS")
      TextBox3.Text = Application("PVS")
   End Sub

End Class
```

Global_asax

```
Imports System.Web.SessionState

Public Class Global_asax
   Inherits System.Web.HttpApplication

   Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
      ' Fires when the application is started

      Application("pp") = 0
      Application("ss") = 0
      Application("pvs") = 0

   End Sub

   Sub Session_Start(ByVal sender As Object, ByVal e As EventArgs)
      ' Fires when the session is started
      Application.Lock()
      Application("pp") = Application("pp") + 1
      Application.UnLock()

   End Sub

   Sub Application_BeginRequest(ByVal sender As Object, ByVal e As EventArgs)
      ' Fires at the beginning of each request
      Application.Lock()
```

```
        Application("pvs") = Application("pvs") + 1
        Application.UnLock()

    End Sub

    Sub Session_End(ByVal sender As Object, ByVal e As EventArgs)
        ' Fires when the session ends
        Application.Lock()
        Application("ss") = Application("ss") + 1
        Application.UnLock()

    End Sub

End Class
```

OUTPUT:



RESULT:

Thus the above program has been executed successfully and the output was verified.

MENU IN XML

EX.NO:6

AIM:

To create a Menu in XML

ALGORITHM:

STEP 1 : Start the process.

STEP 2 : Create an XML file with root element named as Karpagam.

STEP 3 : Create child nodes karpagamarts, karpagameng,karpagaminfotech.

STEP 4 : In every child nodes contains the collegename and departmebts.

STEP 5 : Specify the attributes and their contents of the elements.

STEP 6 : Create DTD for the XML file and XML should conform the DTD

STEP 7: Execute the xml file for getting the result.

STEP 8: Stop the process.

CODING:

MENU.DTD

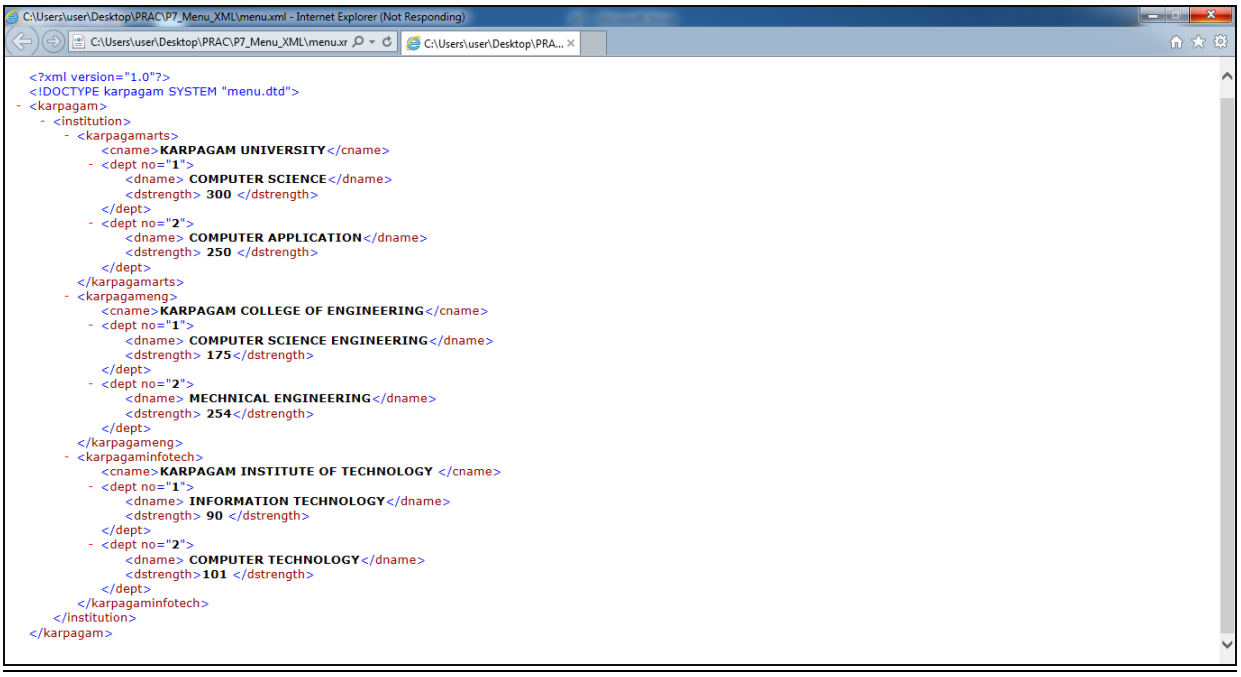
```
<!ELEMENTkarpagam (institution)>

<!ELEMENT institution (karpagamarts,karpagameng,karpagaminfotech)>

<!ELEMENTkarpagamarts (cname,dept+)>

<!ELEMENTcname (#PCDATA)>

<!ELEMENTdept (dname,dstrength)+>

<!ELEMENTdname (#PCDATA)>

<!ELEMENTdstrength (#PCDATA)>

<!ATTLIST dept no CDATA #REQUIRED>

<!ELEMENTkarpagameng (cname,dept+)>

<!ELEMENTcname (#PCDATA)>

<!ELEMENTdept (dname,dstrength)+>

<!ELEMENTdname (#PCDATA)>

<!ELEMENTdstrength (#PCDATA)>

<!ATTLIST dept no CDATA #REQUIRED>

<!ELEMENTkarpagaminfotech (cname,dept+)>

<!ELEMENTcname (#PCDATA)>

<!ELEMENTdept (dname,dstrength)+>

<!ELEMENTdname (#PCDATA)>

<!ELEMENTdstrength (#PCDATA)>

<!ATTLIST dept no CDATA #REQUIRED>
```

MENU.XML:

```
<?xml version="1.0"?>
<!DOCTYPEkarpagam SYSTEM "menu.dtd">
<karpagam>
<institution>
<karpagamarts>
<cname>KARPAGAM UNIVERSITY</cname>
<dept no="1">
<dname> COMPUTER SCIENCE</dname>
<dstrength> 300 </dstrength>
</dept>
<dept no="2">
<dname> COMPUTER APPLICATION</dname>
<dstrength> 250 </dstrength>
</dept>
</karpagamarts>
<karpagameng>
<cname>KARPAGAM COLLEGE OF ENGINEERING</cname>
<dept no="1">
<dname> COMPUTER SCIENCE ENGINEERING</dname>
<dstrength> 175</dstrength>
</dept>
<dept no="2">
<dname> MECHNICAL ENGINEERING</dname>
```

```
<dstrength> 254</dstrength>
</dept>
</karpagameng>
<karpagaminfotech>
<cname>KARPAGAM INSTITUTE OF TECHNOLOGY </cname>
<dept no="1">
<dname> INFORMATION TECHNOLOGY</dname>
<dstrength> 90 </dstrength>
</dept>
<dept no="2">
<dname> COMPUTER TECHNOLOGY</dname>
<dstrength>101 </dstrength>
</dept>
</karpagaminfotech>
</institution>
</karpagam>
```

OUTPUT:



RESULT:

Thus the above program has been executed successfully and the output was verified.

STUDENT DTABASE

EX.NO:7

AIM:

To write a database access program using Asp.net

ALGORITHM:

STEP 1 : Start the process.

STEP 2 : Create a new standard asp project.

STEP 3 : Design the form with necessary details label, textbox and buttons as name, id, result.

STEP 4 : And gets tools->connect to database insert necessary fields in access and insert, values into the data base form.

STEP 5 : Given the source code for insert, update, clean and delete.

STEP 6 : Perform the above operation on students information system.

STEP 7 : Save and execute the program.

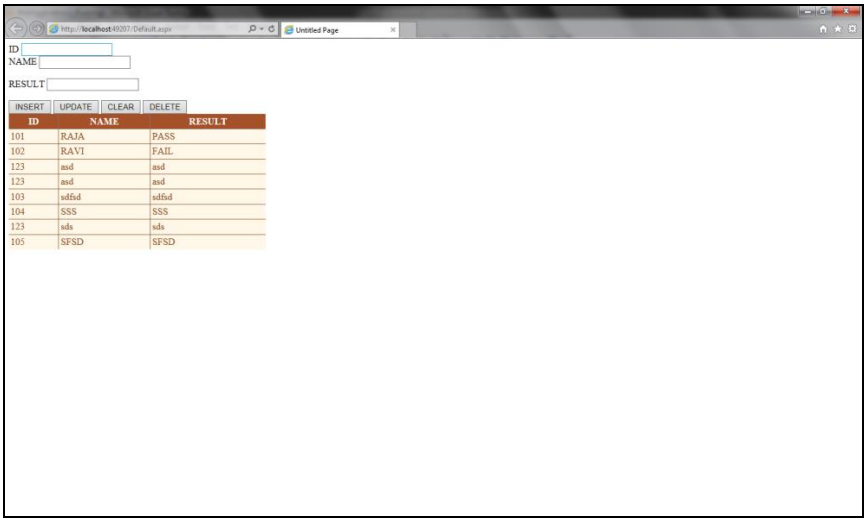
STEP 8 : Stop the process.

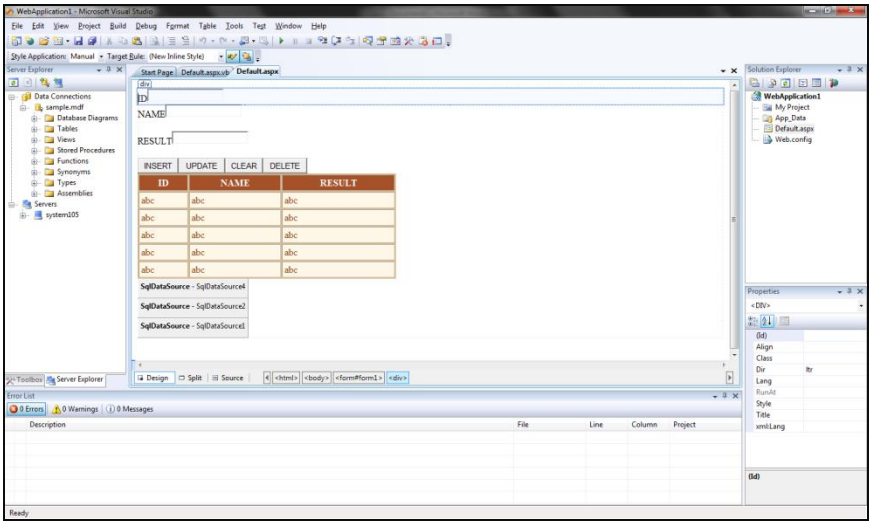
CODING:

```
Imports System.Data.SqlClient
Partial Class _Default
Inherits System.Web.UI.Page

   Dim con As SqlConnection
   Dim cmd As SqlCommand
   Dim dr As SqlDataReader

  Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
                                  Handles Me.Load
con = New SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\user\Documents\sample.mdf;Integrated
Security=True;Connect Timeout=30;User Instance=True")
      con.Open()
   End Sub

Protected Sub Button1_Click(ByVal sender As Object, ByVal e As EventArgs)
                                    Handles Button1.Click
      cmd = New SqlCommand(" insert into Table1 values ('" & TextBox1.Text & " ',
                '" & TextBox2.Text & "' , '" & TextBox2.Text & "') ", con)
      cmd.ExecuteNonQuery()
      MsgBox("Successfully inserted")
      cmd.Dispose()
      con.Close()
   End Sub

Protected Sub Button2_Click(ByVal sender As Object, ByVal e As EventArgs) Handles
Button2.Click
      Dim a As String
      a = TextBox2.Text
      cmd = New SqlCommand("Update Table1 set name= 'RAJA' where name=  ' "
                                    & a & "' ", con)
      cmd.ExecuteNonQuery()
      MsgBox("UPDATED")
      cmd.Dispose()
      con.Close()
   End Sub

  Protected Sub Button3_Click(ByVal sender As Object, ByVal e As EventArgs)
                                    Handles Button3.Click
      TextBox1.Text = ""
```

```
        TextBox2.Text = ""
        TextBox3.Text = ""
    End Sub

Protected Sub Button4_Click(ByVal sender As Object, ByVal e As EventArgs)
                                        Handles Button4.Click
        Dim a As String
        a = TextBox2.Text
        cmd = New SqlCommand("delete from Table1 where name=' " & a & "'", con)
        cmd.ExecuteNonQuery()
        MsgBox("deleted")
        cmd.Dispose()
        con.Close()
    End Sub

End Class
```

OUTPUT:





RESULT:

Thus the above program has been executed successfully and the output was verified.

XML STYLESHEET

EX.NO:8

AIM:

To create a demo for XSLT (XML Style Sheet)

ALGORITHM:

STEP 1 : Start the process.

STEP 2 : Create an XML file which contains reference for the XSLT file.

STEP 3 : In the XML file create necessary elements with their content and attributes.

STEP 4 : Create an XSLT file that sorts the contents of the XML file.

STEP 5 : In the XSLT match the root element with <xsl:template match=" ">tag.

STEP 6 : Use of <xsl:value-of select=" "> tag to retrieve the value from the specified element.

STEP 7 : Make use of the <xsl:for-each select=" "> to loop for every occurrence of an element.

STEP 8 : To sort the records make use<xsl:sort select=" " order=" "/> tag specify the order as ascending or desceding.

STEP 9 : Then close all the tags.

STEP 10 : Stop the process.

CODING:

STYLESHEET.XML

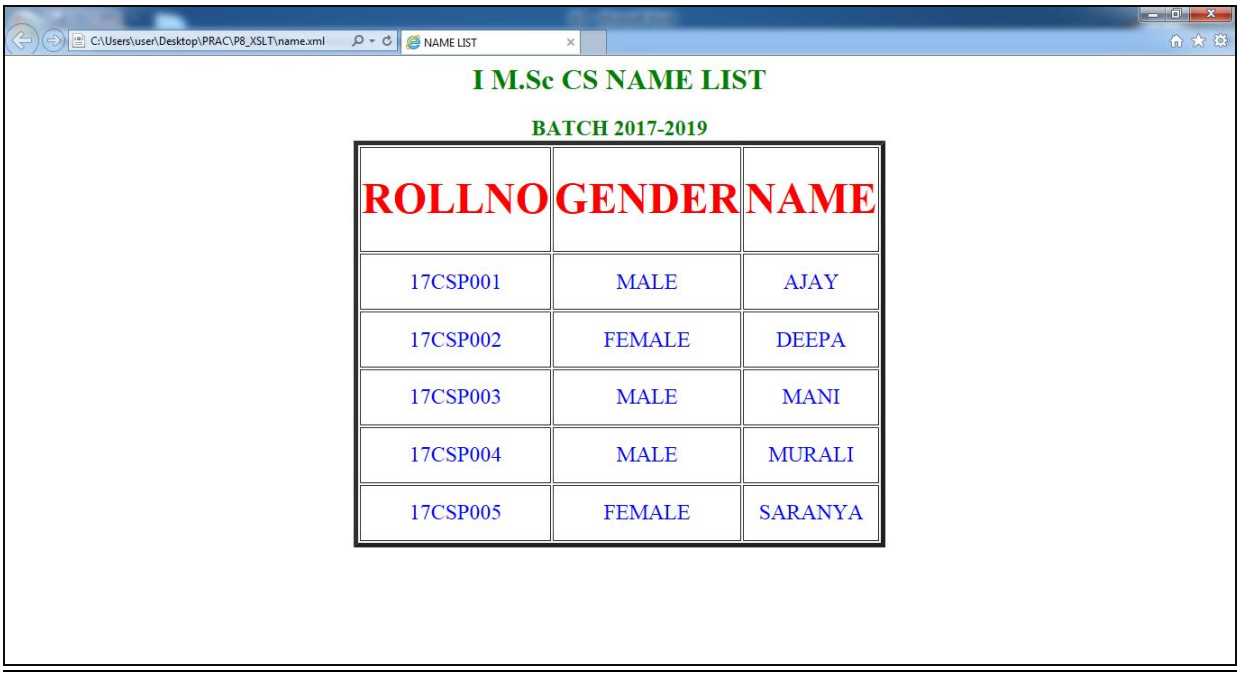
```
<?xml version="1.0"?>

<?xml-stylesheet type="text/xsl" href="namelist.xsl"?>

<stu>

<class>I M.Sc CS </class>

<batch>2017-2019</batch>

<stu1>

<s1 rollno="17CSP001" gender="MALE">AJAY</s1>

<s1 rollno="17CSP002" gender="FEMALE">DEEPA</s1>

<s1 rollno="17CSP003" gender="MALE">MANI</s1>

<s1 rollno="17CSP004" gender="MALE">MURALI</s1>

<s1 rollno="17CSP005" gender="FEMALE">SARANYA</s1>

</stu1>

</stu>
```

STYLESHEET.DTD

```
<?xml version="1.0"?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="stu">

<head>

<title> NAME LIST </title>

</head>

<body background="C:\Documents and Settings\All Users\Documents\My Pictures\Sample Pictures\Sunset.jpg">
```

```
<font color="green">
<h1 align="center"><b><xsl:value-of select="class"/>NAME LIST </b></h1>
<h2 align="center"><b>BATCH <xsl:value-of select="batch"/></b></h2></font>
<table border="5" align="center" width="500" height="450">
<tr>
<td align="center"><font color="red" size="10"><b> ROLLNO </b></font></td>
<td align="center"><font color="red" size="10"><b>GENDER </b></font></td>
<td align="center"><font color="red" size="10"><b>NAME </b></font></td>
</tr>
<xsl:for-each select="stu1/s1">
<xsl:sort select="@rollno" order="ascending"/>
<tr>
<td align="center"><font color="blue" size="5"><xsl:value-of select="@rollno"/></font></td>
<td align="center"><font color="blue" size="5"><xsl:value-of select="@gender"/></font></td>
<td align="center"><font color="blue" size="5"><xsl:value-of select="."/></font></td>
</tr>
</xsl:for-each>
</table>
</body>
</xsl:template>
</xsl:stylesheet>
```

OUTPUT:



RESULT:

Thus the above program has been executed successfully and the output was verified.

TREE STRUCTURE

EX.NO:9

AIM:

To display XML information in tree structure format

ALGORITHM:

STEP 1 : Start the process.

STEP 2 : Create an external DTD.

STEP 3 : Create an XML file with necessary elements, their content and attributes.

STEP 4 : In a HTML document using JavaScript create an ActiveXObject and Load the XML file.

STEP 5 : Get the root of XML, say b, and check if there are child nodes. If yes goto step6.

STEP 6 : Initialize i=0; repeat until i<b.childNodes.length,incrementing i by 1, to print the name of each child node using an ordered list.

STEP 7 : Initialize k=0; repeat until k<b.attibutes.length, incrementing k by 1, to print all the attributes of current node and its attribute value using an ordered list.

STEP 8 : Run the script to get the output in tree format.

STEP 9 : Stop the process.

CODING:

TREE.DTD

```
<!ELEMENT college (student,staff)>
<!ELEMENT student (name,class,dept,year)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT class (#PCDATA)>
<!ELEMENT dept (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT staff (name1,dept1,join)>
<!ELEMENT name1 (#PCDATA)>
<!ELEMENT dept1 (#PCDATA)>
<!ELEMENT join (#PCDATA)>
<!ATTLIST name value ID #REQUIRED>
<!ATTLIST class std ID #REQUIRED>
<!ATTLISTdeptdep ID #REQUIRED>
<!ATTLIST year yr CDATA #REQUIRED>
<!ATTLIST name first CDATA #REQUIRED>
<!ATTLIST name1 value ID #REQUIRED>
<!ATTLIST dept1 dep ID #REQUIRED>
```

TREE.XML

```
<?xml version="1.0"?>
<!DOCTYPE college SYSTEM "tree.dtd">
<college>
```

```
<student>
<name value="xyz" first="aaa">RAJA</name>
<class std="I">M.Sc</class>
<deptdep="computer">COMPUTER SCIENCE</dept>
<year yr="2017">FIRST YEAR</year>
</student>
<staff>
<name1 value="ABC">KAVITHA</name1>
<dept1 dep="cs">COMPUTER SCIENCE</dept1>
<join>20/06/2017</join>
</staff>
</college>
```

TREE.HTML

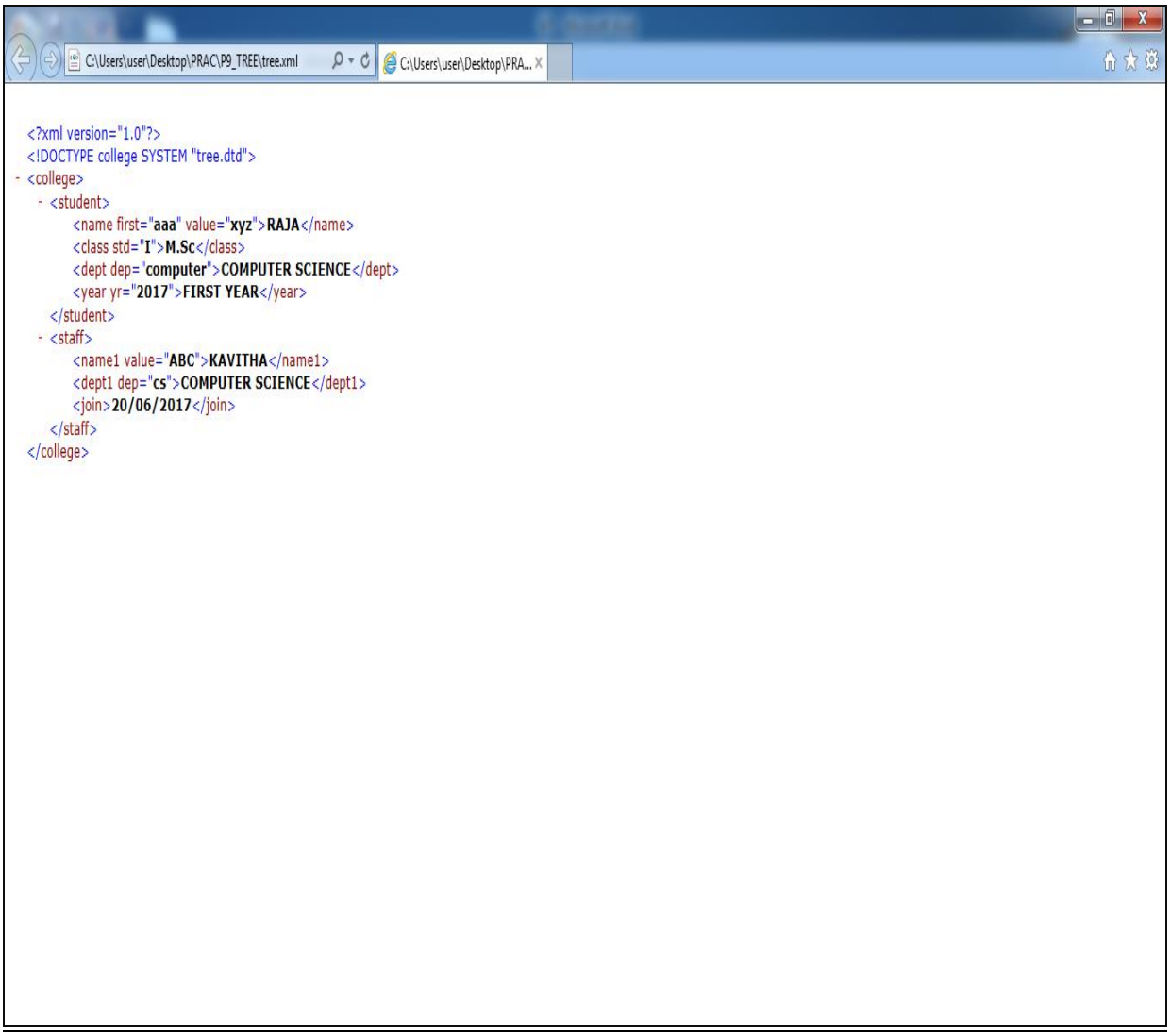
```
<!DOCTYPE html PUBLIC "-//w3c//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/tree.dtd">
<html>
<head>
<title> A DOM Example </title>
</head>
<body bgcolor="pink">
<script type="text/javascript" language="javascript">
varxmldocument=new ActiveXObject("Microsoft.XMLDOM");
xmldocument.load("tree.xml");
var element=xmldocument.documentElement;
```

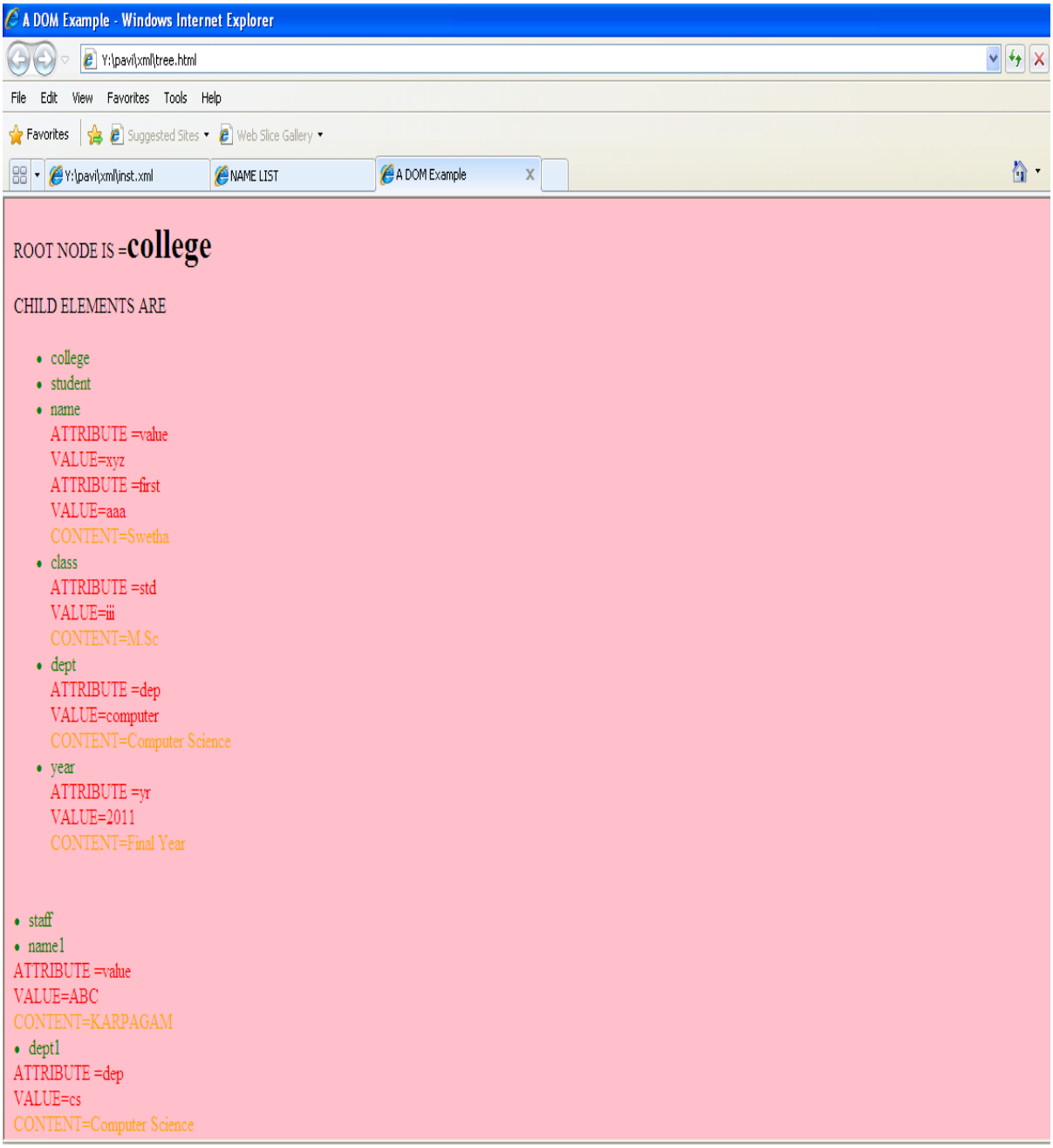
```
document.writeln("<p>ROOT NODE IS =<b><font size=6>" + element.nodeName+"</font>");
document.writeln("</b></p>CHILD ELEMENTS ARE <ul>");
var b;
var child;

if(element.childNodes.length !=0)
{
cur(element);
}
function cur(b)
{
document.write(" <font color=green>");
if(b.nodeName!="#text")
{
document.writeln("<li>"+b.nodeName);
}
else
{
document.write("<font color=blue>");
document.writeln("CONTENT="+b.nodeValue);
document.write("</font>");
}
document.write("</font");
```

```
if(b.childNodes.length !=0)
{
document.writeln("<ul type='square'>");
for(var i=0;i<b.childNodes.length;i++)
 {
document.write("<font color=red>");
for(var k=0;k<b.attributes.length;k++)
  {
document.writeln("<br>ATTRIBUTE ="+b.attributes(k).nodeName);
document.writeln("<br>VALUE="+b.attributes(k).nodeValue);
  }
document.write("</font>");
child=b.childNodes.item(i);
document.writeln("<Br>");
cur(child);
 }
document.writeln("</ul>");
}
else
return;
}
</script>
```

```
</body>

</html>
```

OUTPUT:





RESULT:

Thus the above program has been executed successfully and the output was verified.