EX NO: 1 Image Mapping using HTML

AIM:

To write the html code to perform the Image Mapping.

ALGORITHM:

Step 1: Start.
Step 2: Write the code for imagemap.html.
Step 3: Perform mapping using <MAP> tag.
Step 4: Insert Hyperlink using <A href>.
Step 5: Display Results.
Step 6: Stop.

PROGRAM:

Nether.html

```
<html>
<head>
<title>netherlands</title>
</head>
<body bgcolor="black">
<p align="Center">
</p>
<map name="netherlands">
<area href="C:\gro.html" shape="circle" coords="362,56,10">
<area href="C:\fry.html" shape="circle" coords="278,67,10">
<area href="C:\lim.html" shape="circle" coords="317,408,10">
<area href="C:\hol.html" shape="circle" coords="149,291,10">
</map>
<img src="C:\nether.gif"
usemap="#netherlands">
</body>
</html>
```

\fry.html

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

<img src="fry.gif">
A flag with pompeblêdden (lily leaves) was already
mentioned in the "Gudrunlied" from the 11th century.
This was also used by the Ommelanden. Since the
middle of the 19th century the flag is used by the
Fryske Biweging, and was accepted by the Deputed States (Provincial Legislature) in 1897.
During the festivities of the 40th and 50th reigning jubilee of Queen Wilhelmina in 1938 and
1948 it was used by each municipality with the municipal arms in the canton.
Finally it was officially adopted 9 July 1957 by
the States of Friesland, nr. 12 Prov./besluit 20.
The provincial paper, 1958, nr. 12, gives detailed
construction sheets.
</body>
</html>

gro.html

<html>
<head>
<title> groningen</title>
</head>
<body>
<img src="C:\gro.gif">
Originally a part of Frisia, Groningen became
a part of the Frankish Empire around 785.
Charlemagne assigned the Christianization of
this new possession to Ludger. In the 11th century,
the city of Groningen was a village in Drenthe that
belonged to the Bishopric of Utrecht, while most of
the province was in the diocese of Münster.
During the Middle Ages, central control was remote,
and the city of Groningen acted as a city state,
exerting a dominating influence on the surrounding
Ommelanden. Around 1500, Maximilian I, Holy Roman
</body>
</html>

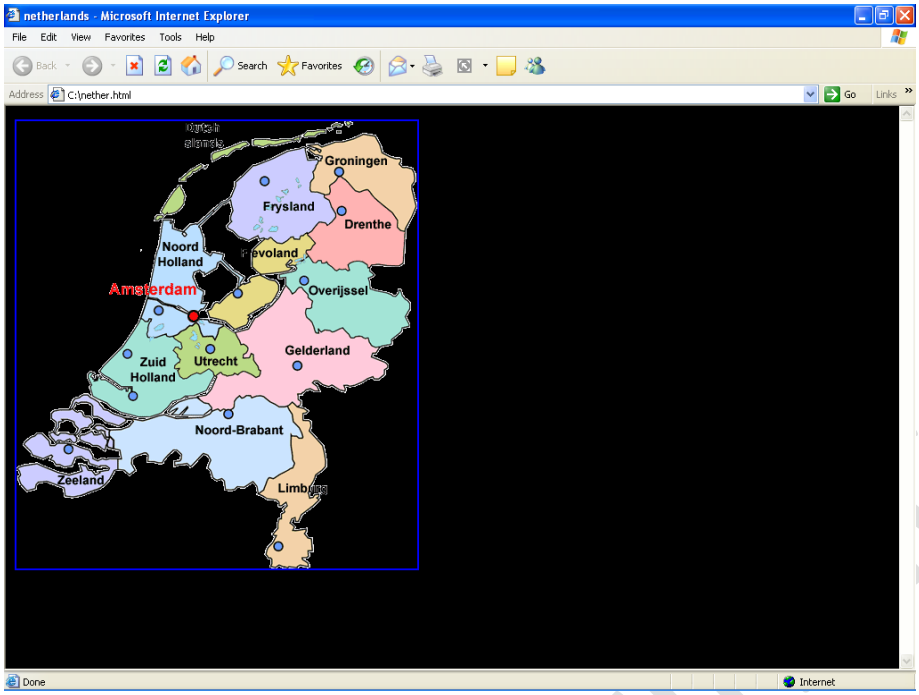
hol.html

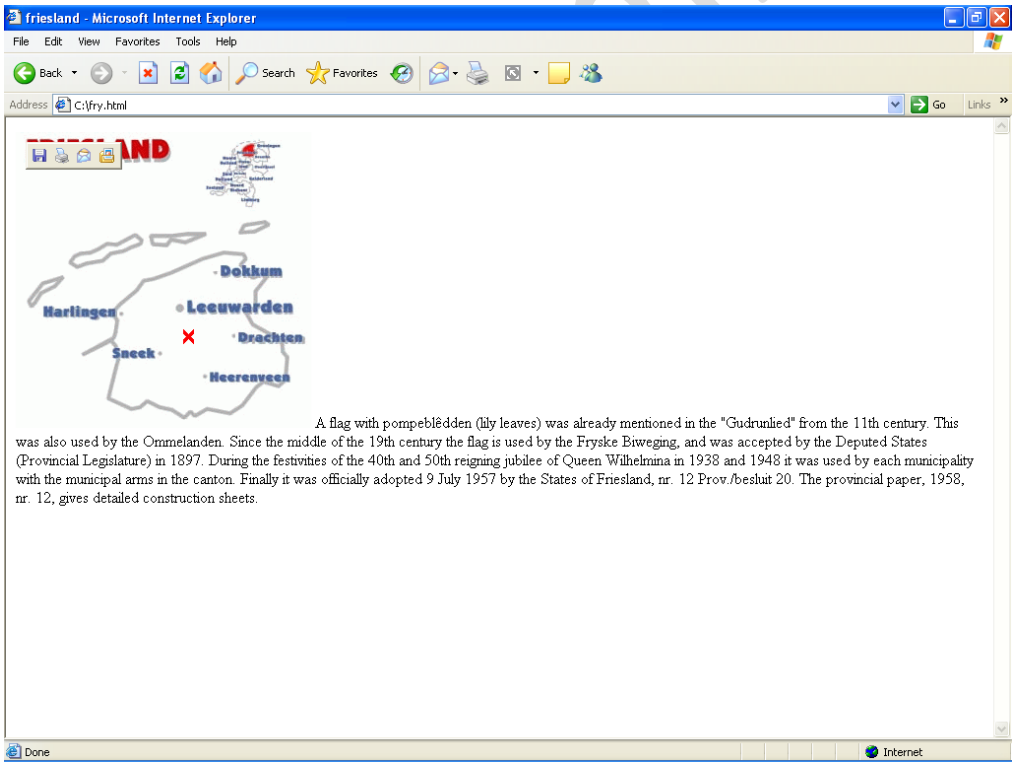
```
<html>
<head>
<title>zuid holland</title>
</head>
<body>
<img src="hol.gif">
Also spelled  Zuidholland , English  South
Holland  provincie, western Netherlands,
bordering the North Sea and adjoining the
provincies of Noord-Holland (north), Utrecht
and Gelderland (east), and Noord-Brabant and
Zeeland (south). Drained by the ramifications
of the Lek, Waal, and Maas (Meuse) rivers,
Zuid-Holland includes the islands of Dordrecht,
IJsselmonde, Hoeksche Waard, Voorne-Putten, and
 Goeree-Overflakkee.
</body>
</html>
```

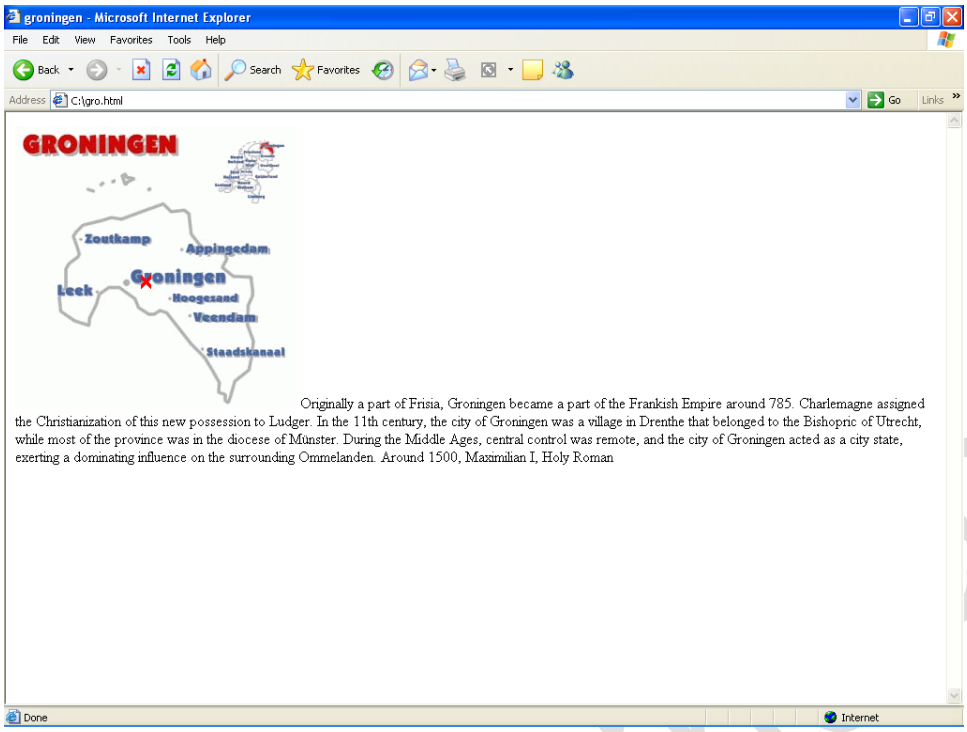
lim.html

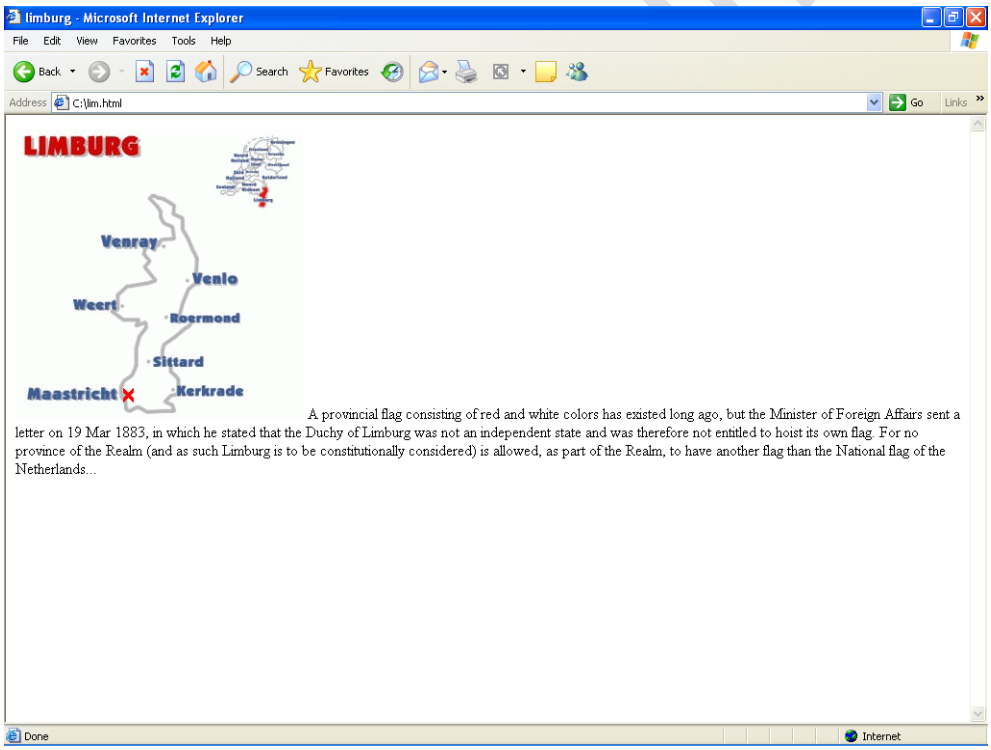
```
<html>
<head>
<title>limburg</title>
</head>
<body>
<img src="lim.gif">
A provincial flag consisting of red and white
colors has existed long ago, but the Minister of
Foreign Affairs sent a letter on 19 Mar 1883, in
which he stated that the Duchy of Limburg was not
 an independent state and was therefore not entitled
 to hoist its own flag. For no province of the Realm
 (and as such Limburg is to be constitutionally
considered) is allowed, as part of the Realm, to
have another flag than the National flag of the
Netherlands...
</body>
</html>
```

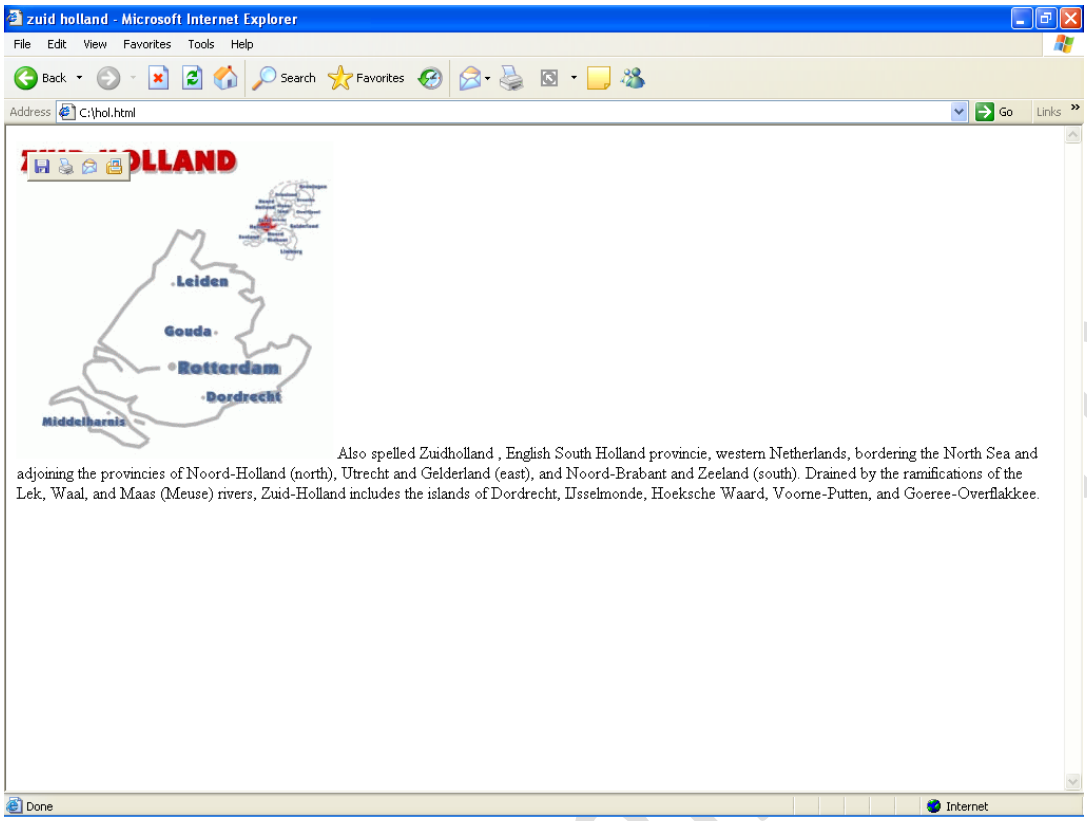
OUTPUT:


netherlands - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\nether.html
Go Links
Groningen
Frysland
Drenthe
Noord Holland
Flevoland
Amsterdam
Overijssel
Gelderland
Zuid Holland
Utrecht
Noord-Brabant
Zeeland
Done
Internet


friesland - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\fry.html
Go Links
Dokkum
Harlingen
Leeuwarden
Drachten
Sneek
Heerenveen
A flag with pompeblêdden (lily leaves) was already mentioned in the "Gudrunlied" from the 11th century. This was also used by the Ommelanden. Since the middle of the 19th century the flag is used by the Fryske Biweging, and was accepted by the Deputed States (Provincial Legislature) in 1897. During the festivities of the 40th and 50th reigning jubilee of Queen Wilhelmina in 1938 and 1948 it was used by each municipality with the municipal arms in the canton. Finally it was officially adopted 9 July 1957 by the States of Friesland, nr. 12 Prov./besluit 20. The provincial paper, 1958, nr. 12, gives detailed construction sheets.
Done
Internet


groningen - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\gro.html
Go Links
GRONINGEN
Zoutkamp
Appingedam
Leek
Groningen
Hoogezand
Veendam
Staadskanaal
Originally a part of Frisia, Groningen became a part of the Frankish Empire around 785. Charlemagne assigned the Christianization of this new possession to Ludger. In the 11th century, the city of Groningen was a village in Drenthe that belonged to the Bishopric of Utrecht, while most of the province was in the diocese of Münster. During the Middle Ages, central control was remote, and the city of Groningen acted as a city state, exerting a dominating influence on the surrounding Ommelanden. Around 1500, Maximilian I, Holy Roman
Done
Internet


limburg - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\lim.html
Go Links
LIMBURG
Venray
Venlo
Weert
Roermond
Sittard
Maastricht
Kerkrade
A provincial flag consisting of red and white colors has existed long ago, but the Minister of Foreign Affairs sent a letter on 19 Mar 1883, in which he stated that the Duchy of Limburg was not an independent state and was therefore not entitled to hoist its own flag. For no province of the Realm (and as such Limburg is to be constitutionally considered) is allowed, as part of the Realm, to have another flag than the National flag of the Netherlands...
Done
Internet



RESULT:

Thus image mapping has been performed using HTML.

Ex. No: 2(a) APPLYING EMBEDDED CASCADING STYLE SHEETS TO
DATE: WEB PAGE

AIM:

To demonstrate the use of inline style sheets using DHTML.

PROGRAM:

```
<html>
<head>
<title>olympics</title>
<style type ="text/css">
h1
{
font_family=monotype corsiva;
font_size=34;
color=red;
}
</style>
</head>
<body bgcolor=white>
<center>
<h2><marquee width=100% behaviour=alternate>
all about olympics</marquee></h2>
</center>
<p>
<h1> MYTHS REGARDING OLYMPICS</h1>
<ul>

<li>there are many myths surrounding the orgin bof ancient olympic games ,the most popular of which identifies heracles as the creator of olympic games

<li>according to the legend ,heracles built tjhe olympic stadium and surrounding building as an honour to his father zeus,after completing his 12 labours.

<li> after he built the stadium he walked in a straightline for 400 studies and called this distance a "stadion" that later also became a unit of distance.

<li>this is also why a modern stadium track is 400m is circumference the distance a runner travels  in one lap.
```

another myth associates the 1st games with the ancient greek concept of ekechairia olympic twice.

</p><img
src="file:///C:/Documents%20and%20Settings/student/My%20Documents/My%20Pictures/fantasy_sports.jpg" width=200 height=200>
Opening ceremony climax with lighting of the cauldron.
</body>
</html>

OUTPUT:
MYTHS REGARDING OLYMPICS

- there are many myths surrounding the orgin bof ancient olympic games ,the most popular of which identifies heracles as the creator of olympic games
- according to the legend ,heracles built tjhe olympic stadium and surrounding building as an honour to his father zeus,after completing his 12 labours.
- after he built the stadium he walked in a straightline for 400 studies and called this distance a "stadion" that later also became a unit of distance.
- this is also why a modern stadium track is 400m is circumference the distance a runner travels in one lap.
- another myth associates the 1st games with the ancient greek concept of ekechairia olympic twice.

Opening ceremony climax with lighting of the cauldron.

RESULT:

Thus the program was executed successfully.

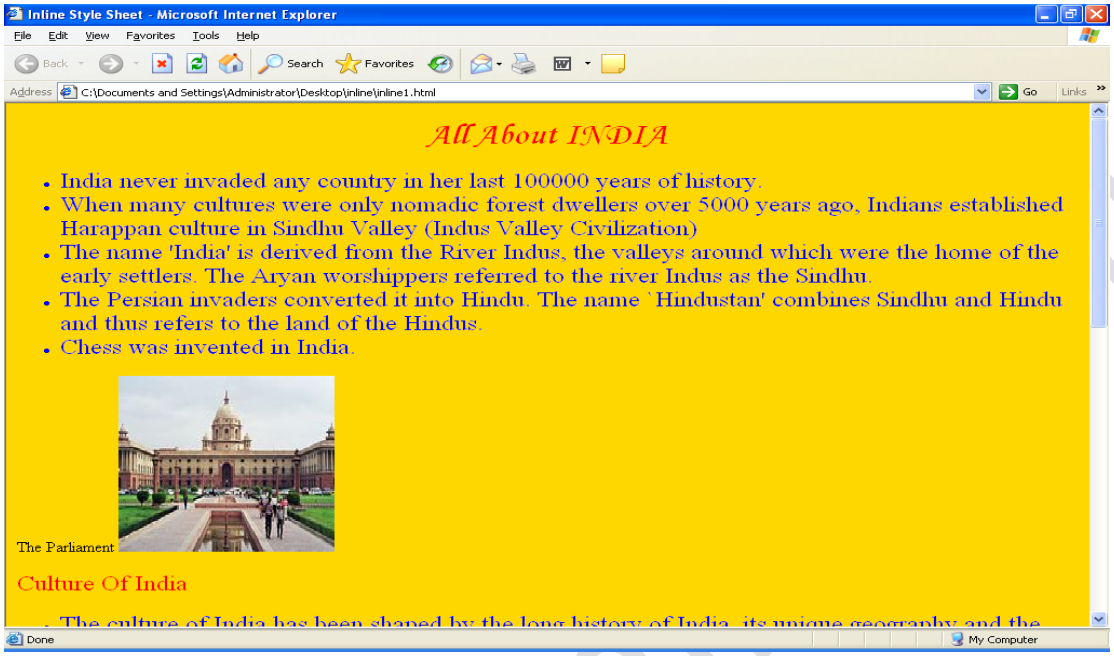
EX. NO: 2(b) APPLYING CASCADING STYLE SHEETS TO A WEB
DATE: PAGE

AIM:
To demonstrate the use of inline style sheets using DHTML.

PROGRAM:
```
<html>
<head>
<title>Inline Style Sheet</title>
<body bgcolor=gold>
<center>
<p style="font-family=Monotype Corsiva ;font.size=34;color=red">All About INDIA</p>
</center>
<p>
<ul><font size="5" face="Times New Roman" color=blue>
<li>India never invaded any country in her last 100000 years of history.
<li>When many cultures were only nomadic forest dwellers over 5000 years ago, Indians
established Harappan culture in Sindhu Valley (Indus Valley Civilization)
<li>The name 'India' is derived from the River Indus, the valleys around which were the home of
the early settlers. The Aryan worshippers referred to the river Indus as the Sindhu.
<li>The Persian invaders converted it into Hindu. The name `Hindustan' combines Sindhu and
Hindu and thus refers to the land of the Hindus.
<li>Chess was invented in India.
</font>
</ul>
<p> The Parliament
<img src="180px-NorthBlock.jpg" width=200 height=200>
</p>
<font size="5" face="Times New Roman" color=red>
Culture Of India</font>
<p>
<ul><font size="5" face="Times New Roman" color=blue>
<li>
The culture of India has been shaped by the long history of India, its unique geography and the
absorption of customs, traditions and ideas from some of its neighbors
as well as by preserving its ancient heritages, which were formed during the Indus Valley
Civilization and evolved further during the Vedic age, rise and decline of Buddhism, Golden age,
Muslim conquests and European colonization.
<li>India's great diversity of cultural practices, languages, customs, and traditions are examples
of this unique co-mingling over the past five millennia.
</font>
</ul>
</p>
<font size="6" face="Times New Roman" color=red>Indian Classical Dances</font></br>
<img src="Bharat.jpg" height=200 width=200>
```

```
<img src="kathakali.jpg" height=200 width=200>
<img src="manipuri.jpg" height=200 width=200>
<img src="kuchipudi.jpg" height=200 width=200>
<img src="mohiniattam.jpg" height=200 width=200>
<font size="5" face="Times New Roman" color=blue>
<ul>
The various dance forms of India are
<li> Bharatanatyam
<li>Odissi
<li>Kathakali
<li> MohiniAttam
</ul>
</font>
</body>
</html>
```

OUTPUT:



RESULT:

Thus the program was executed successfully.

EX NO: 3 Client side scripting to validate form object using DHTML

DATE:

AIM:

To perform client side scripts for validating web form controls using DHTML.

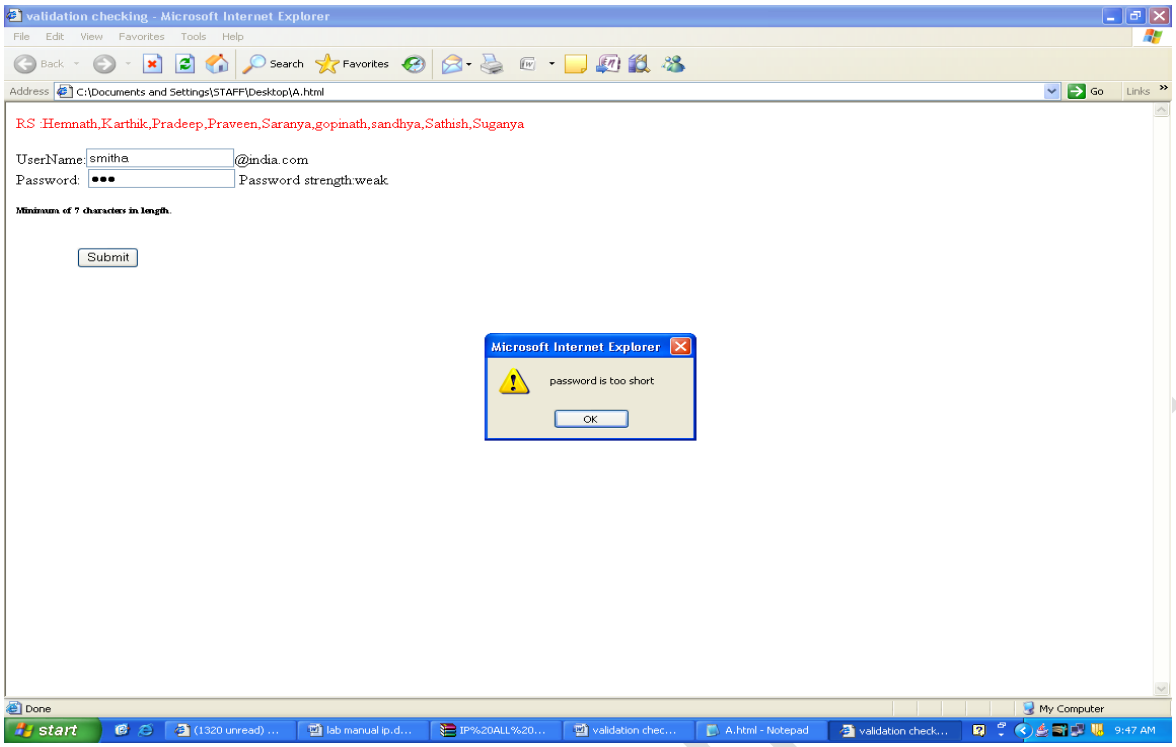
ALGORITHM:

Step1: Start.
Step2: Design an HTML file with user name and password which is to be validated.
Step3: Validate the password by checking if the password is of minimum 7 characters in length.
Step4: If step 3 fails an alert message is passed insisting the character should be minimum 7 characters in length.
Step5: Stop.

PROGRAM CODE:

```
<html>
<head>
<title> validation checking</title>
</head>
<script language="javascript">
function password strength(password)
{
var str1="strong";
var str2="weak";
if(password.length>6)
document.getElementById("password Description").inner HTML=str1;
else
document.getElementById("password description").inner HTML=str2;
}
function a()
{
var i=document.form1.uname.value;
var a=i.length;
var j=document.form1.pass.value;
var b=j.length;
if(a==0&&b==0)
window.alert("enter d username and password");
alse
{
```

```
if(a==0)
window.alert("enter username");
if(b==0)
window.alert("enter password");
}
if(a!=0&&b!=0&&b<7)
window.alert("password s short");
}
</script>
 <body>
<form name="form1">
<font color="red"><marquee>
web moderator:divya ram
</marquee></font>
<br>
<br>
username<input type="text" name="uname">@google.com
<br>
password;  
<input type="password" name="pass" onkeyup="password strength(this.value)">
<label password strength:</label>
<label id="password Description">password not entered</label>
<p><h6> minimum of 7 characters in length</h6>



</p>
<br>
    
    
  
<input type="button"value="submit"onclick="a()">
</form>
</body>
<html>
```



RESULT:

Thus, client side scripts for validating web form controls using DHTML was successfully tested and executed.

EX NO: 4 COLOR PALETTE WITH MATRIX OF BUTTONS

DATE:

AIM:

To write a program in Java Swing to perform the following actions:
a) Create a color palette with matrix of buttons
b) Set background and foreground of the text area by selecting a color from color palette.
c)In order to select foreground and background use check box control as radio buttons.

ALGORITHM:

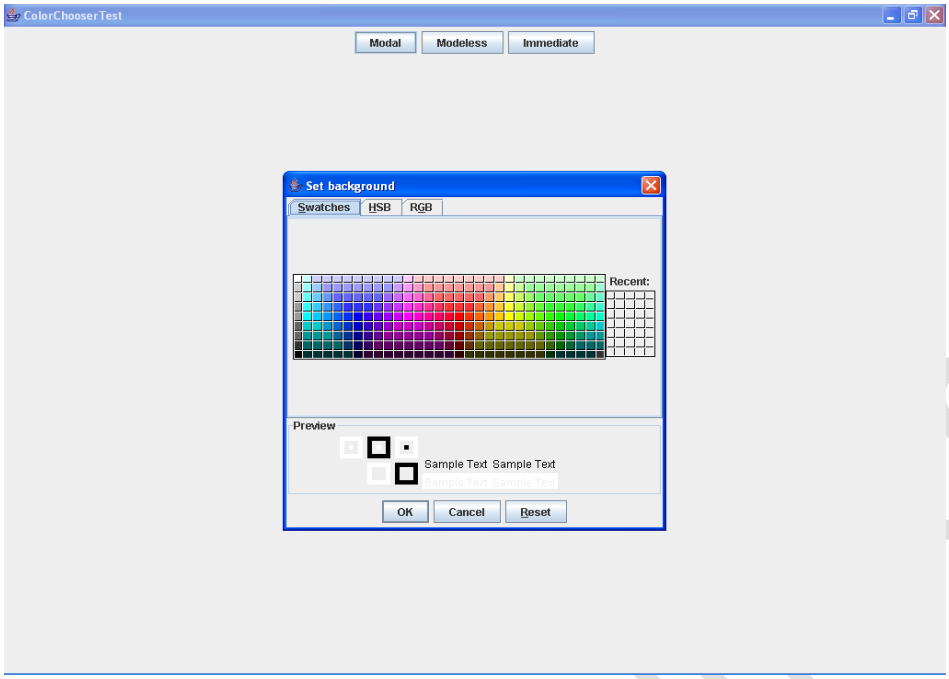
Step1: Start.
Step2: Create a color chooser frame from the respective class.
Step3: For the selected frame, background color and size are set.
Step4: The three buttons are created for modal, modeless and immediate dialog boxes.
Step5: Create panels for modal, modeless and immediate dialog boxes.
Step6: In action to be performed, according to the buttons clicked , the color palette appears.
Step7: The background or the foreground is selected and when any color is chosen from the palette ,the color is changed.
Step8: Stop.

PROGRAM CODE:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
class ColorChooserTest
{
public static void main(String[ ] args)
{
ColorChooserFrame frame=new ColorChooserFrame();
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.show();
}
}
class ColorChooserFrame extends JFrame
{
public ColorChooserFrame()
{
setTitle("ColorChooserTest");
setSize(WIDTH,HEIGHT);
```

```
ColorChooserPanel panel=new ColorChooserPanel();
Container contentPane=getContentPane();
contentPane.add(panel);
}
public static final int WIDTH=300;
public static final int HEIGHT=200;
}
class ColorChooserPanel extends JPanel
{
public ColorChooserPanel()
{
JButton modalButton=new JButton ("Modal");
modalButton.addActionListener(new ModalListener());
add(modalButton);
JButton modelessButton =new JButton ("Modeless");
modelessButton.addActionListener(new ModelessListener());
add(modelessButton);
JButton immediateButton =new JButton ("immediate");
immediateButton.addActionListener(new ImmediateListener());
add(immediateButton);
}
private class ModalListener implements ActionListener
{
public void actionPerformed(ActionEvent event)
{
Color defaultColor=getBackground();
Color selected =JColorChooser.showDialog(ColorChooserPanel.this,"Set background",
defaultColor);
setBackground(selected);
}
}
private class ModelessListener implements ActionListener
{
public ModelessListener()
{
chooser=new JColorChooser();
dialog=JColorChooser.createDialog(ColorChooserPanel.this,"Background Color", false,
chooser,new ActionListener()
{
public void actionPerformed(ActionEvent event)
```

```
{
setBackground(chooser.getColor());
}
},null);
}
public void actionPerformed(ActionEvent event)
{
chooser.setColor(getBackground());
dialog.show();
}
private JDialog dialog;
private JColorChooser chooser;
}
private class ImmediateListener implements ActionListener
{
public ImmediateListener()
{
chooser=new JColorChooser();
chooser.getSelectionModel().addChangeListener(new ChangeListener()
{
public void stateChanged(ChangeEvent event)
{
setBackground(chooser.getColor());
}
});
dialog=new JDialog((Frame)null,false);
dialog.getContentPane().add(chooser);
dialog.pack();
}
public void actionPerformed(ActionEvent event)
{
chooser.setColor(getBackground());
dialog.show();
}
private  JDialog dialog;
private JColorChooser chooser;
}
}
```



RESULT:

Thus ,the program for implementing color palette has been executed successfully.

Ex.No: 5 **Implementing Colour picker applet to apply foreground and background colour**

DATE:

AIM: Execute programs in java to create Applets

ALGORITHM:

Step 1: Start

Step 2: A colorchooserApplet show six scroll bars that the user can manipulate to set the red,green,blue,hue,brightness and saturation components of a color.

Step 3: A color patch shows the selected color, and there are six labels that show the numerical values of all the components.

Step 4: RGB components are specified as integers in the range 0 to 255. HSB components are specified as float values in the range 0.0F to 1.0F.

Step 5: Stop

PROGRAM:

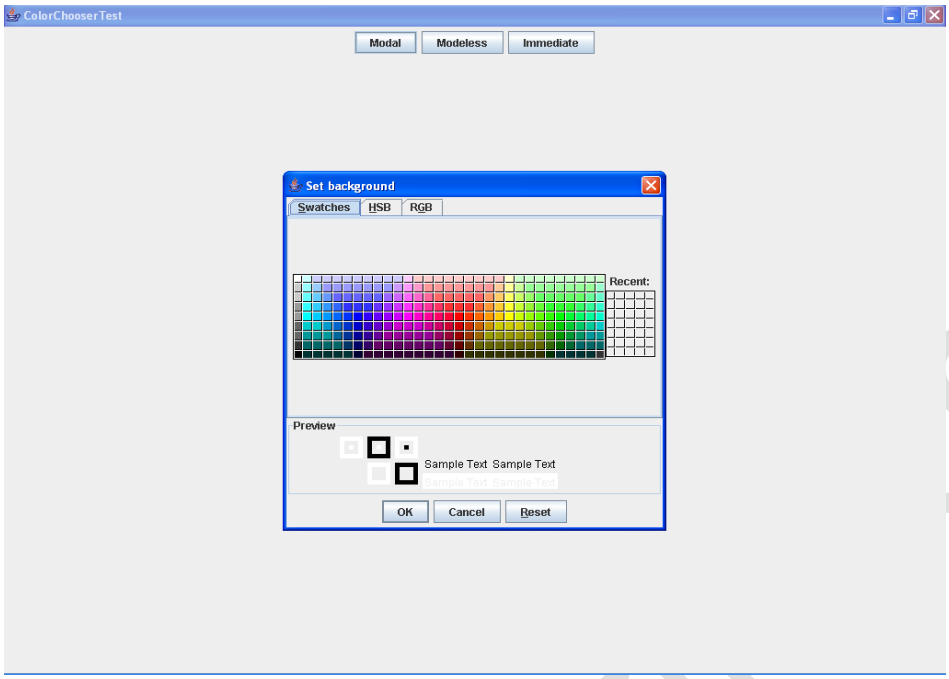
```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

public class ColorChooserApplet extends Applet implements AdjustmentListener {

private float[] hsb = new float[3];   // For holding HSB color components.

private int r = 0, g = 0, b = 0;      // The RGB color components.

private Scrollbar hueScroll, brightnessScroll, saturationScroll,       redScroll, greenScroll,
blueScroll;

private Label hueLabel, brightnessLabel, saturationLabel,  // Display            component values.
            redLabel, greenLabel, blueLabel;

private Canvas colorCanvas;  // Color patch for displaying the color.

public void init() {
```

```
Color.RGBtoHSB(0,0,0,hsb);  // Get HSB equivalent of RGB = (0,0,0);

/* Create Scrollbars with possible values from 0 to 255. */

hueScroll = new Scrollbar(Scrollbar.HORIZONTAL, (int)(255*hsb[0]), 10,  0, 265);
saturationScroll = new Scrollbar(Scrollbar.HORIZONTAL,          (int)(255*hsb[1]), 10, 0, 265);
brightnessScroll = new Scrollbar(Scrollbar.HORIZONTAL, (        int)(255*hsb[2]), 10, 0, 265);
redScroll = new Scrollbar(Scrollbar.HORIZONTAL, 0, 10, 0, 265);
greenScroll = new Scrollbar(Scrollbar.HORIZONTAL, 0, 10, 0, 265);
blueScroll = new Scrollbar(Scrollbar.HORIZONTAL, 0, 10, 0, 265);

/* Create Labels showing current RGB and HSB values. */

hueLabel = new Label(" H = " + hsb[0]);
saturationLabel = new Label(" S = " + hsb[1]);
brightnessLabel = new Label(" B = " + hsb[2]);
redLabel = new Label(" R = 0");
greenLabel = new Label(" G = 0");
blueLabel = new Label(" B = 0");

/* Set background colors for Scrollbars and Labels, so they don't
   inherit the gray background of the applet. */

hueScroll.setBackground(Color.lightGray);
saturationScroll.setBackground(Color.lightGray);
brightnessScroll.setBackground(Color.lightGray);
redScroll.setBackground(Color.lightGray);
greenScroll.setBackground(Color.lightGray);
blueScroll.setBackground(Color.lightGray);

hueLabel.setBackground(Color.white);
saturationLabel.setBackground(Color.white);
brightnessLabel.setBackground(Color.white);
redLabel.setBackground(Color.white);
greenLabel.setBackground(Color.white);
blueLabel.setBackground(Color.white);

/* Set the applet to listen for changes to the Scrollbars' values */

hueScroll.addAdjustmentListener(this);
saturationScroll.addAdjustmentListener(this);
brightnessScroll.addAdjustmentListener(this);
redScroll.addAdjustmentListener(this);
greenScroll.addAdjustmentListener(this);
blueScroll.addAdjustmentListener(this);
```

```
    /* Create a canvas whose background color will always be set to the
       currently selected color. */

    colorCanvas = new Canvas();
    colorCanvas.setBackground(Color.black);

    /* Create the applet format, which consists of a row of
       three equal-sized regions holding the Scrollbars,
       the Labels, and the color patch.  The background color
       of the applet is gray, which will show around the edges
       and between components. */

    setLayout(new GridLayout(1,3,3,3));
    setBackground(Color.gray);
    Panel scrolls = new Panel();
    Panel labels = new Panel();
    add(scrolls);
    add(labels);
    add(colorCanvas);

    /* Add the Scrollbars and the Labels to their respective panels. */

    scrolls.setLayout(new GridLayout(6,1,2,2));
    scrolls.add(redScroll);
    scrolls.add(greenScroll);
    scrolls.add(blueScroll);
    scrolls.add(hueScroll);
    scrolls.add(saturationScroll);
    scrolls.add(brightnessScroll);

    labels.setLayout(new GridLayout(6,1,2,2));
    labels.add(redLabel);
    labels.add(greenLabel);
    labels.add(blueLabel);
    labels.add(hueLabel);
    labels.add(saturationLabel);
    labels.add(brightnessLabel);

} // end init();


public void adjustmentValueChanged(AdjustmentEvent evt) {
        // This is called when the user has changed the values on
        // one of the scrollbars.  All the scrollbars and labels
        // and the color patch are reset to correspond to the new color.
    int r1, g1, b1;
```

```
    r1 = redScroll.getValue();
    g1 = greenScroll.getValue();
    b1 = blueScroll.getValue();
    if (r != r1 || g != g1 || b != b1) {  // One of the RGB components has changed.
      r = r1;
      g = g1;
      b = b1;
      Color.RGBtoHSB(r,g,b,hsb);
    }
    else {  // One of the HSB components has changed.
       hsb[0] = hueScroll.getValue()/255.0F;
       hsb[1] = saturationScroll.getValue()/255.0F;
       hsb[2] = brightnessScroll.getValue()/255.0F;
       int rgb = Color.HSBtoRGB(hsb[0],hsb[1],hsb[2]);
       r = (rgb >> 16) & 0xFF;
       g = (rgb >> 8) & 0xFF;
       b = rgb & 0xFF;
    }
    redLabel.setText(" R = " + r);
    greenLabel.setText(" G = " + g);
    blueLabel.setText(" B = " + b);
    hueLabel.setText(" H = " + hsb[0]);
    saturationLabel.setText(" S = " + hsb[1]);
    brightnessLabel.setText(" B = " + hsb[2]);
    redScroll.setValue(r);
    greenScroll.setValue(g);
    blueScroll.setValue(b);
    hueScroll.setValue((int)(255*hsb[0]));
    saturationScroll.setValue((int)(255*hsb[1]));
    brightnessScroll.setValue((int)(255*hsb[2]));
    colorCanvas.setBackground(new Color(r,g,b));
    colorCanvas.repaint();  // Tell the system to redraw the canvas in its new color.
  } // end adjustmentValueChanged


  public Insets getInsets() {
       // The system calls this method to find out how much space to
       // leave between the edges of the applet and the components that
       // it contains.  I want a 3-pixel border at each edge.
     return new Insets(3,3,3,3);
  }
}  // end class ColorChooserApplet
```

OUTPUT



RESULT:

Thus the colorchooser program was successfully executed using java applets.

Ex.No:6 **Servlet Program using HTTP**

DATE:

AIM: To write a simple servlet program using HTTP in java.

ALGORITHM:

1. Create a servlet program using http.
2. Set classpath where servlet-api.jar file resides.
3. Compile the servlet program using javac programname.java
4. Place the class file …\Tomcat 5.5\webapps\ROOT\WEB-INF\classes\ folder.
5. modify the web.xml file using your servletClassName.
6. Invoke the class file using http://localhost:8080/servetClassName from your browser

PROGRAM:

Home.java

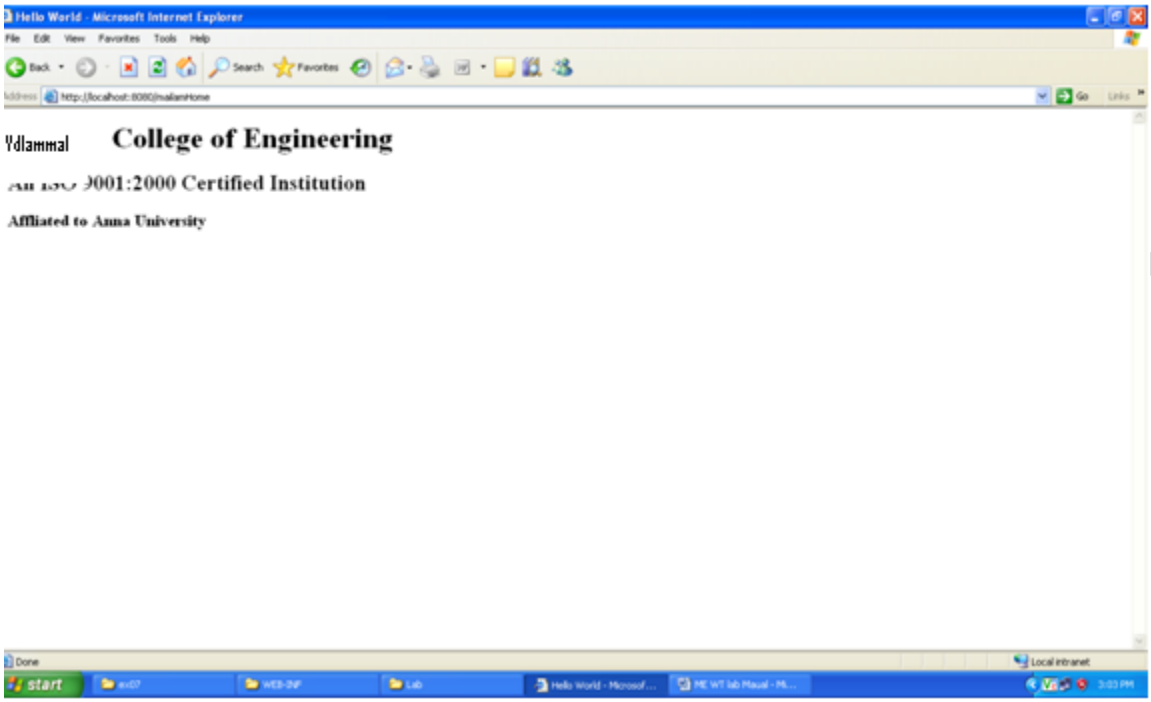
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class mailamHome extends HttpServlet
{
  public void doGet(HttpServletRequest request, HttpServletResponse response)
                        throws ServletException,IOException
  {
    response.setContentType("text/html");
    PrintWriter pw = response.getWriter();
    pw.println("<html>");
    pw.println("<head><title>Hello World</title></title>");
    pw.println("<body>");
    pw.println("<h1>Velammal College of Engineering</h1>");
    pw.println("<h2>An ISO 9001:2000 Certified Institution</h2>");
    pw.println("<h3>Affliated to Anna University</h3>");
```

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

web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--<!DOCTYPE web-app
 PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
 "http://java.sun.com/dtd/web-app_2_3.dtd"> -->
<web-app>
 <servlet>
  <servlet-name>Mailam</servlet-name>
  <servlet-class>mailamHome</servlet-class>
 </servlet>
 <servlet-mapping>
 <servlet-name>Mailam</servlet-name>
 <url-pattern>/mailamHome</url-pattern>
 </servlet-mapping>
</web-app>
```

OUTPUT:



RESULT:

Thus the given program was coded and executed successfully.

EX NO: 7 **Online Shopping using JSP with three tier architecture**

DATE:

AIM: To write an online book shopping application using JSP objects.

ALGORITHM:

1. Create home, login, registration, profile, catalog and order html pages.
2. Create jsp pages which does all business works on the server.
3. Use appropriate database to store the details of the books.
4. Create tables to store login details and books details.
5. Connect the database using odbc.jdbc driver.
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><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=”/tr/login.html”>LOGIN</a>
 href=”/tr/login.html”>REGISTRATION</a></div></pre>
 </body></html>
```

Login.html:

```
<html>
    <body bgcolor=”pink”><br><br><br>
    <form name="myform" method="post" action=/tr1/login.jsp">
    <div align="center"><pre>
```

```
LOGIN ID : <input type="passwors" name="pwd"></pre><br><br>
PASSWORD : <input type="password" name="pwd"></pre><br><br>
</div>
<br><br>
<div align="center">
  <inputtype="submit"value="ok"
onClick="validate()">     <input type="reset"
value="clear">
</form>
</body>
</html>
```

Reg.html:

```
<html>
<body bgcolor="pink"><br><br>
<form name="myform" method="post" action="/tr1/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">
    <inputtype="submit"value="ok"
  onClick="validate()">()">     <input type="reset"
  value="clear">
  </form>
  </body>
  </html>
```

Profile.html:

```
 <html>
<body bgcolor="pink"><br><br>
<form name="myform" method="post" action="/tr1/profile.jsp">
  <div align="center"><pre>
  LOGIN ID     : <input type="text" name="id"><br>
  </pre><br><br>
  </div>
  <br><br>
  <div align="center">
    <inputtype="submit"value="ok"
  onClick="validate()">()">     <input type="reset"
  value="clear">
  </form>
  </body>
  </html>
```

Catalog.html:

```
<html>
<body bgcolor="pink"><br><br><br>
<form method="post" action="/tr1/catalog.jsp">
  <div align="center"><pre>
  BOOK TITLE : <input type="text" name="title"><br>
  </pre><br><br>
  </div>
  <br><br>
  <div align="center">
    <inputtype="submit"value="ok"
  name="button1">     <inputtype="reset"value="clear"
  name="button2">
</form>
```

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

Order.html:

```
<html>
<body bgcolor="pink"><br><br><br>
<form method="post" action="/tr1/order.jsp">
 <div align="center"><pre>
 LOGIN ID        :<input type="text" name="id"><br>
 PASSWORD    : <input type="password" name="pwd"><br>
 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">     <input type="reset" value="clear"
 name="button2">
 </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");
Driver d=new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(d);
    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger"
);
Statement stmt=con.createStatement();
String sqlstmt="select id,password from login where id="+id+" and password="+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=\"/tr1/login.html\">press LOGIN to RETRY</a>");
 }
 else
 {
    out.println("VALID LOGIN ID<br><br>");
    out.println("<h3><ul>");
        out.println("<li><ahref=\"profile.html\"><fontcolor=\"black\">USER
  PROFILE</font></a></li><br><br>");

    out.println("<li><ahref=\"catalog.html\"><fontcolor=\"black\">BOOKS
  CATALOG</font></a></li><br><br>");
            out.println("<li><ahref=\"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");
     int no=Integer.parseInt(phno);
     Driver d=new oracle.jdbc.driver.OracleDriver();
     DriverManager.registerDriver(d);
     Connection con=
    DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
     Statement stmt=con.createStatement();
     String sqlstmt="select id from login";
     ResultSet rs=stmt.executeQuery(sqlstmt);
     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=\"/tr1/reg.html\">press REGISTER to RETRY</a>");
}
else
{
   Statement stmt1=con.createStatement ();
   stmt1.executeUpdate ("insert into login values ("+name+","+addr+","+no+","+id+","+pwd+")");
       out.println ("YOU DETAILS ARE ENTERED <br><br>");
       out.println ("<a href =\"/tr1/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");
    Driver d=new oracle.jdbc.driver.OracleDriver();
    DriverManager.regiserDriver(d);
   Connection con=
      DriverManager.getConnection
                ("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
          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=\"/tr1/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");
         Driver d=new oracle.jdbc.driver.OracleDriver ();
         DriverManager.regiserDriver (d);
         Connection con=
         DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
          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=\"/tr1/catalog.html\">press HERE to RETRY </a>");
}
out.println ("</body></html>");
%>
```

Order.jsp:

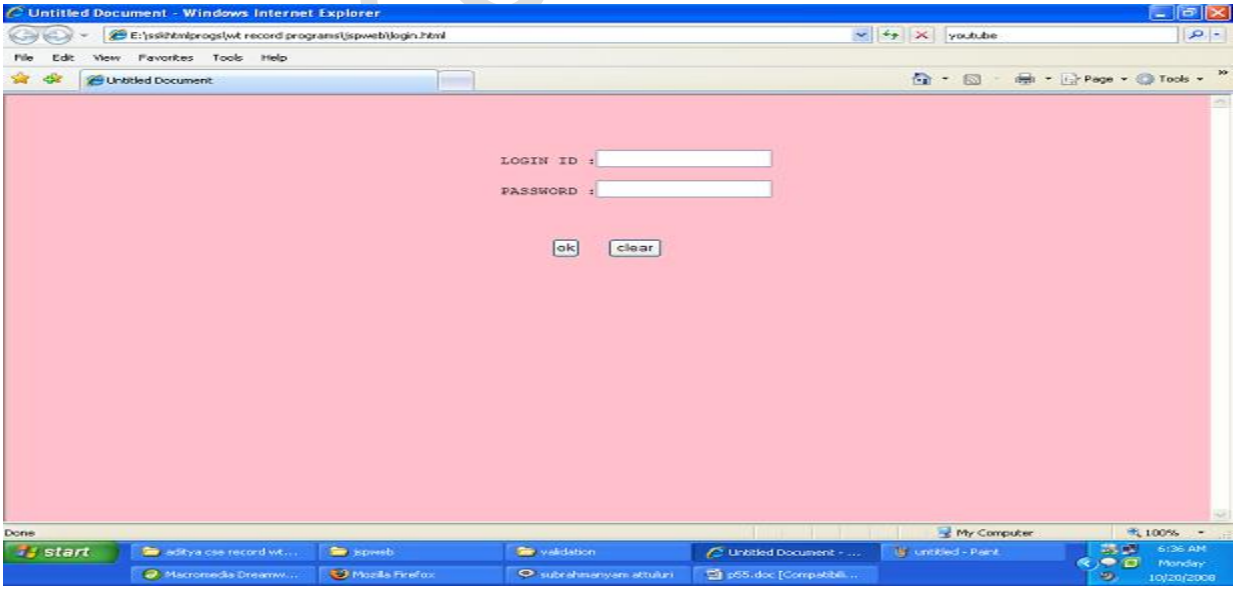
```
    <%@page import="java.sql.*"%>
    <%@page import="java.io.*"%>
    <%
        out.println ("<html><body bgcolor=\"pink\">");
        String id=request.getParameter ("id");
        String pwd=request.getParameter ("pwd");
        String title=request.getParameter ("title");
        String count1=request.getParameter ("no");
        String date=request.getParameter ("date");
        String cno=request.getParameter ("cno");
```

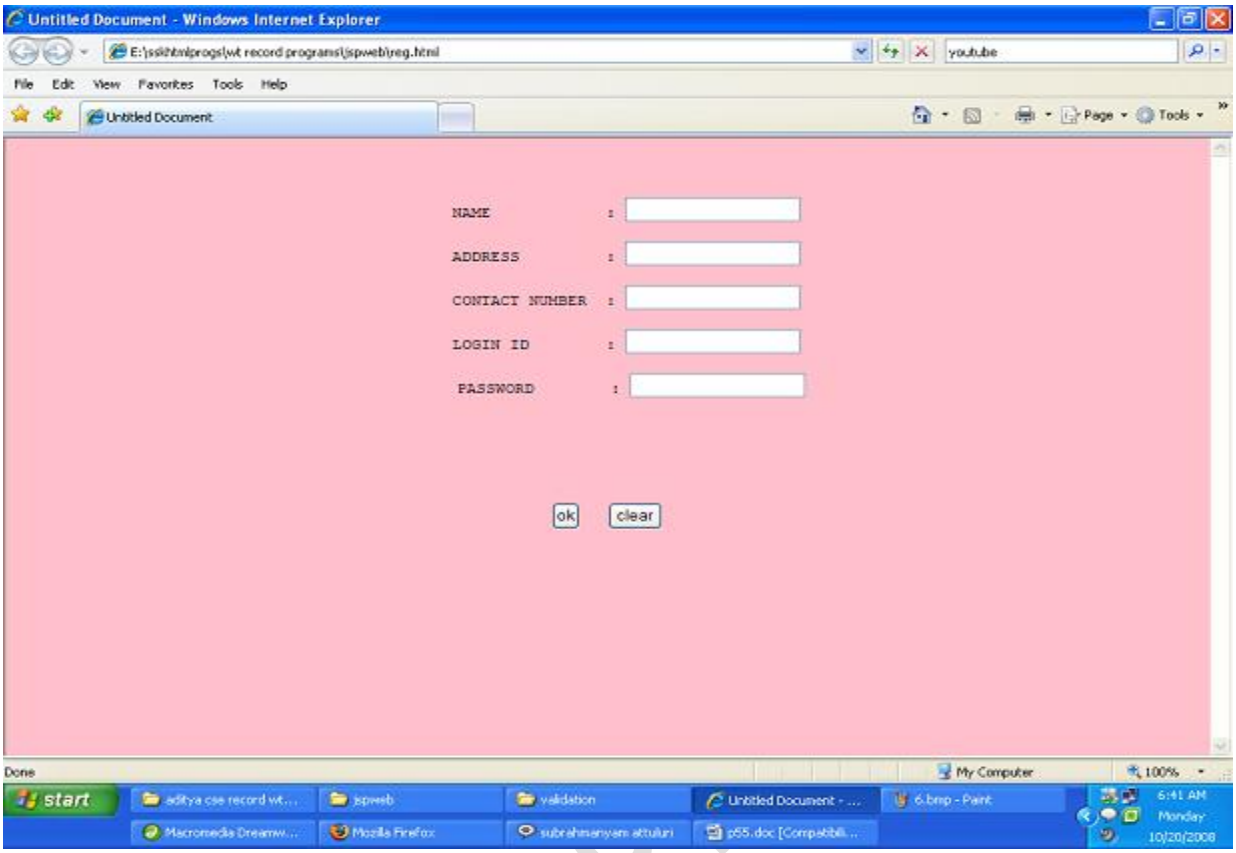
```
        int count=Integer.parseInt(count1);
        Driver d=new oracle.jdbc.driver.OracleDriver ();
        DriverManager.regiserDriver (d);
        Connection con=
      DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
       Statement stmt=con.createStatement ();
       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=\"/tr1/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(1));
```

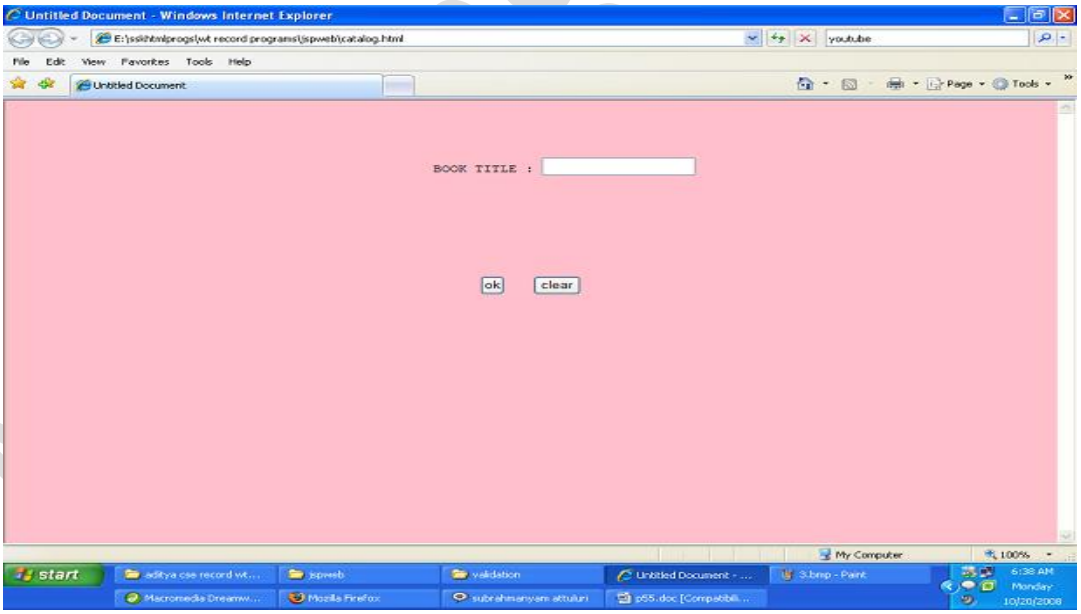
```
        amount=count*x;
        out.println("AMOUNT   :"+amount+"<br><br><br><br>");
        Statement stmt1=con.createStatement ();
        stmt1.executeUpdate ("insert into details
("+id+","+title+","+amount+","+date+","+cno+")");
        out.println ("YOU ORDER HAS TAKEN<br>");
}
if(flag1==0)
{
          out.println("SORRY INVALID BOOK TRY AGAIN  <br><br>");
          out.println("<a href=\"/tr1/order.html\">press HERE to RETRY </a>");
   }
}                    out.println ("</body></html>");%>
}
```

OUTPUT:


Untitled Document - Windows Internet Explorer
E:\sskhtmlprogs\wt record programs\jspweb\main.html
youtube
File Edit View Favorites Tools Help
Untitled Document
Page Tools
ONLINE BOOK STORAGE
Welcome to online book storage.
Press LOGIN if you are having id
Otherwise press REGISTRATION
LOGIN
REGISTRATION
My Computer
100%
start


Untitled Document - Windows Internet Explorer
E:\sskhtmlprogs\wt record programs\jspweb\login.html
youtube
File Edit View Favorites Tools Help
Untitled Document
Page Tools
LOGIN ID :
PASSWORD :
ok
clear
Done
My Computer
100%
start


Untitled Document - Windows Internet Explorer
File Edit View Favorites Tools Help
Untitled Document
NAME :
ADDRESS :
CONTACT NUMBER :
LOGIN ID :
PASSWORD :
ok
clear
Done
My Computer
start


Untitled Document - Windows Internet Explorer
File Edit View Favorites Tools Help
Untitled Document
BOOK TITLE :
ok
clear
My Computer
start

RESULT:

Thus the given program was coded and executed successfully.

EX.NO: 8 **Applying styles to XML**

DATE:

AIM: Programs using XML-Schema-XSLT/XSL

ALGORITHM:

Step 1: Start the Program
Step 2: Create a root process for food
Step 3: Create a style for XSLT with focus on each item
Step 4: Output the items
Step 5: Stop

PROGRAM:

XML File

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<breakfast_menu>
	<food>
		<name>Belgian Waffles</name>
		<price>$5.95</price>
		<description>two of our famous Belgian Waffles with plenty of real maple syrup</description>
		<calories>650</calories>
	</food>
	<food>
		<name>Strawberry Belgian Waffles</name>
		<price>$7.95</price>
		<description>light Belgian waffles covered with strawberries and whipped cream</description>
		<calories>900</calories>
	</food>
	<food>
		<name>Berry-Berry Belgian Waffles</name>
		<price>$8.95</price>
		<description>light Belgian waffles covered with an assortment of fresh berries and whipped cream</description>
		<calories>900</calories>
```

```
        </food>
        <food>
                <name>French Toast</name>
                <price>$4.50</price>
                <description>thick slices made from our homemade sourdough
bread</description>
                <calories>600</calories>
        </food>
        <food>
                <name>Homestyle Breakfast</name>
                <price>$6.95</price>
                <description>two eggs, bacon or sausage, toast, and our ever-popular hash
browns</description>
                <calories>950</calories>
        </food>
</breakfast_menu>
```

XSLT File

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<html xsl:version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.w3.org/1999/xhtml">
  <body style="font-family:Arial;font-size:12pt;background-color:#EEEEEE">
    <xsl:for-each select="breakfast_menu/food">
      <div style="background-color:teal;color:white;padding:4px">
        <span style="font-weight:bold"><xsl:value-of select="name"/></span>
        - <xsl:value-of select="price"/>
      </div>
      <div style="margin-left:20px;margin-bottom:1em;font-size:10pt">
        <xsl:value-of select="description"/>
        <span style="font-style:italic">
          <xsl:value-of select="calories"/> (calories per serving)
        </span>
      </div>
    </xsl:for-each>
  </body>
</html>
```

OUTPUT:

Belgian Waffles - $5.95
two of our famous Belgian Waffles with plenty of real maple syrup*650 (calories per serving)*
Strawberry Belgian Waffles - $7.95
light Belgian waffles covered with strawberries and whipped cream*900 (calories per serving)*
Berry-Berry Belgian Waffles - $8.95
light Belgian waffles covered with an assortment of fresh berries and whipped cream*900 (calories per serving)*
French Toast - $4.50
thick slices made from our homemade sourdough bread*600 (calories per serving)*
Homestyle Breakfast - $6.95
two eggs, bacon or sausage, toast, and our ever-popular hash browns*950 (calories per serving)*

RESULT:

Thus Programs using XML-Schema-XSLT/XSL was developed and successfully executed.

Ex.No. 9 **Advanced Java Script program to get text**

DATE:

AIM:

To Execute a Program using AJAX to get text

ALGORITHM:

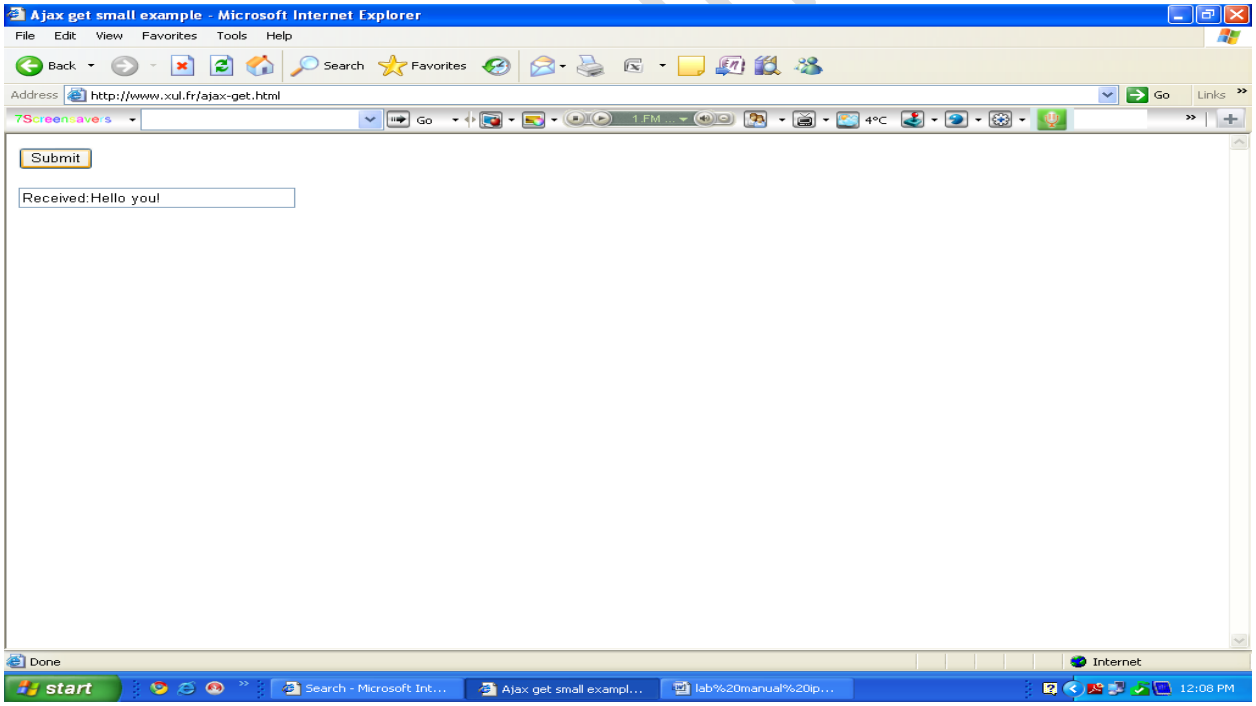
Step 1: Start
Step2: Create a text box and submit button of event handling submitform()
Step 3: Create new ActiveXobject of XMLHTTP under exception
Step 4: if ready state is four then get the message else error message
Step 5: stop

PROGRAM:

```
<html>
<head>
<script>
function submitForm()
{
   var xhr;
   try {  xhr = new ActiveXObject('Msxml2.XMLHTTP');   }
   catch (e)
   {
      try {   xhr = new ActiveXObject('Microsoft.XMLHTTP');    }
      catch (e2)
      {
        try {  xhr = new XMLHttpRequest();     }
        catch (e3) {  xhr = false;   }
      }
    }

   xhr.onreadystatechange  = function()
   {
       if(xhr.readyState  == 4)
       {
           if(xhr.status  == 200)
              document.ajax.dyn="Received:"  + xhr.responseText;
           else
             document.ajax.dyn="Error code " + xhr.status;
       }
   };

  xhr.open(GET, "data.txt",  true);
  xhr.send(null);
```

```
}
</script>
</head>

<body>
   <FORM method="POST" name="ajax" action="">
      <INPUT type="BUTTON" value="Submit"  ONCLICK="submitForm()">
      <INPUT type="text" name="dyn"  value="">
   </FORM>
 </body>
</html>
```

OUTPUT:



RESULT:

Thus the AJAX program was successfully tested and executed.