LAB MANUAL

EASWARI ENGINEERING COLLEGE

DEPARTMENT OF INFORMATION TECHNOLOGY

IT 2357 – WEB TECHNOLOGY LABORATORY

LAB MANUAL

III YEAR –B.TECH (IT)

JAN 2015 to MAY 2015

PREPARED BY

Mr.M.VIVEKANANDAN, AP/IT
Mrs.K.KOQUILAMBALLE, AP/IT

APPROVED BY

HOD

SYLLABUS

LIST OF EXPERIMENTS

1. Client Side Scripts for Validating Web Form Controls using DHTML
2. Write programs in Java to create applets incorporating the following features
   Create a color palette with matrix of buttons
   Set background and foreground of the control text area by selecting a color from color palette.
   In order to select Foreground or background use check box control as radio buttons
   To set background images
3. Write programs in Java using Servlets:
   a. To invoke servlet from HTML form
4. Write programs in Java using Servlets:
   a. To invoke servlet from Applets
5. Write a Programs to load the text file on over the image using AJAX
6. Write a servlet program to add to numbers using servlet
7. *Write a JSP program to retrieve the data from the database (use your own database).*
8. 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.
9. *Write a program to link CSS with XML Document.*
10. Create a web page with all types CSS Selectors and Positioning
11. Program to validate the xml document and count the number of specified tags in xml document using DOM and SAX
12. Program to validate the xml document and count the number of specified tags in xml document using SAX
13. Create a web page with the following using HTML
    a. To embed an imag
    b.
    c. e map in a web page
    d. To fix the hot spots
    e. Show all the related information when the hot spots are clicked.
14. Create a web page with all types of Cascading style sheets.
15. Programs using XML – Schema – XSLT/XSL

EX NO:

* IMAGE MAPS*

* AIM :

To 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.

* TAGS USED WITH SYNTAX :

- **Image Tag : To insert an image in our web page .**
  - **Syntax :** <img src=”” height=”” width=””></img>
- **Hot Spot : To create an hotspot at the necessary position on the image**

  Syntax :
  - **Circle :**

<area shape="circle" coords="" href=" .html" alt=""> </area>
  - **Rectangle :**

<area shape="rect" coords="" href=" .html" alt=""> </area>
  - **Poly :**

<area shape="poly" coords="" href=" .htm" alt=""> </area>

* PROCEDURE :

- Embed an ‘India map’ image using Image Tag .
- Fix hot spot with different styles .
- Create a web pages related to the hotspot .
- Open a browser and display the web page .

* PROGRAM CODING :

```
<html>
<head>
<title>Image Map</title>
</head>
<body>
<img src="india-map.jpg" usemap=#i></img>
<map name=i>
<area shape=circle alt="Haryana" coords="149,151,20"
href="http://en.wikipedia.org/wiki/Haryana"></area>
<area shape=rect alt="Rajasthan" coords="59,170,141,230"
href="http://en.wikipedia.org/wiki/Rajasthan"></area>
<area shape=circle alt="Gujarat" coords="55,273,30"
href="http://en.wikipedia.org/wiki/Gujarat"></area>
<area shape=poly alt="Himachal Pradesh" coords="171,217,120,280,219,275"
href="http://en.wikipedia.org/wiki/Himachal_Pradesh"></area>
</map>
</body>
</html>
```

*** OUTPUT :**

*** RESULT :**

Thus the image map is created and hot spot is fixed and related information is displayed in HTML document.

EX. NO:

* CASCADING STYLE SHEETS *

* AIM :

To demonstrate the use of different types of style sheets.

* TAGS USED WITH SYNTAX :

- **Frameset Tag** :
  - **Syntax :** <frameset cols or rows=50%,50%></frameset>
- **Frame Tag :**
  - **Syntax :** <frame src=" .html" name="f1">
- **Anchor Tag** :
  - **Syntax :**
- **Preformatted Text** :
  - **Syntax** : <pre></pre>
- **Paragraph Tag :** To place the text in a paragraph format .
  - **Syntax :**
- **Types of List :**
  - **Ordered List** :

    Syntax :

    …(list item)…
    …(list item)…

  - **Unordered List** :

    Syntax :

    …(list item)…
    …(list item)…

- **Definition List** :

  Syntax :

  <dl>
  <dt>…(definition term)...</dt>
  <dd>…(definition data)…</dd>
  </dl>

* ALGORITHM:

- Create a simple HTML page.
- Write internal CSS using <style> tag in the same page become internal CSS.
- Create another CSS file using <style> tag with extension .css file, which becomes external CSS file
- Include the external CSS file in your file using Link tag.
- Find the style changes on your page.
- Create an inline style sheet with style attribute with the following attribute
  - <h1 style=”color:red;font-size:30>Inline</h1>
- Create a CSS file and import it in the main program , to implement importing CSS .

* PROGRAM CODING :

- **main.html :**

```
<html>
<head>
<title>CSS</title>
<frameset cols=40%,60%>
<frame src=1.html name=f1></frame>
<frame src=qaz.jpg name=f2></frame>
</frameset>
</head>
</html>
```

- **1.html :**

```
<html>
<head>
<title>Source</title>
<style type=text/css>
a:visited{color:red}
a:link{color:blue}
a:active{color:green}
a:hover{color:black}
h2.rr{border-color:pink;border-style:outset;border-width:10;cursor:move}
h2.rr1{border-color:pink;border-style:outset;border-
width:10;cursor:progress}
h2.rr2{border-color:pink;border-style:outset;border-width:10;cursor:wait}
h2.rr3{border-color:pink;border-style:outset;border-
width:10;cursor:crosshair}
</style>
</head>
<body>
<img src="op.jpg" width=50 height=50></img><img src="op.jpg"
width=50 height=50></img><img src="op.jpg" width=50
height=50></img><img src="op.jpg" width=50 height=50></img><img
src="op.jpg" width=50 height=50></img><img src="op.jpg" width=50
height=50></img><img src="op.jpg" width=50 height=50></img><img
src="op.jpg" width=50 height=50></img><img src="op.jpg" width=50
height=50></img><img src="op.jpg" width=50 height=50></img>
<center><h1>~* Types of CSS *~</h1></center>
<img src="op.jpg" width=50 height=50></img><img src="op.jpg"
width=50 height=50></img><img src="op.jpg" width=50
height=50></img><img src="op.jpg" width=50 height=50></img><img
src="op.jpg" width=50 height=50></img><img src="op.jpg" width=50
height=50></img><img src="op.jpg" width=50 height=50></img><img
src="op.jpg" width=50 height=50></img><img src="op.jpg" width=50
height=50></img><img src="op.jpg" width=50 height=50></img>
<center>
<br>
<br>
<br>
<h2 class=rr><a href=internal.html target=f2>Internal Cascading Style
Sheet !</a></h2>
<h2 class=rr1><a href=external.html target=f2>External Cascading Style
Sheet !</a></h2>
<h2 class=rr2><a href=inline.html target=f2>Inline Cascading Style Sheet
!</a></h2>
<h2 class=rr3><a href=importing.html target=f2>Importing Cascading
Style Sheet !</a></h2>
<br>
<img src="az.jpg" height=100 width=400></img>
```

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

- **Internal CSS** :

```
<html>
<head>
<title>Internal CSS</title>
<style type=text/css>
pre.s1{background-image:url('12.gif');background-repeat:repeat-
x;background-position:center }
p.s2{color:Maroon;text-align:center;letter-spacing:1inch;word-
spacing:2inch;text-decoration:overline;text-shadow:green;direction:ltr}
</style></head>
<body>
<center>
<pre class=s1><h1><font face="Lucida Calligraphy">~ * Hai Friends !!!
* ~</font></h1></pre>
<h2><p class=s2><font face="Lucida Calligraphy">Welcome to My Page
!!!!!</font></p></h2>
<img src="baby2.gif"></img></center>
<h2><marquee><font face="Lucida Calligraphy" color=black>" *
Internal CSS - Class Selector - Background Property and Text Property ! *
"</font></marquee></h2>
</body>
</html>
```

- **External CSS** :

```
<html>
<head>
<title>External CSS</title>
<link rel=stylesheet type=text/css href=one.css>
<link rel=stylesheet type=text/css href=two.css>
</head>
<body>
<center>
<h1 id=style1>~ * Here is my another page !! * ~</h1>
<h2><marquee><font face="Lucida Calligraphy" color=black>" *
External CSS - ID Selector - Font Property and Border Property used here
! * "</font></marquee></h2>
<h2 id=style2> * Stars : *</h2>
<h3> A star is a massive, luminous sphere of plasma held together by
gravity. The nearest star to Earth is the Sun, which is the source of most of
```

the energy on the planet. Some other stars are visible from Earth during the night when they are not obscured by
atmospheric phenomena, appearing as a mass of fixed luminous points because of their immense distance. Historically, the most prominent stars on the celestial sphere were grouped together into constellations and asterisms, and the brightest and largest stars have gained proper names. Extensive catalogues of stars have been assembled by astronomers, which provide standardized star designations.
</h3>
<img src="aa.jpg" width=200 height=200></img><img src="4.jpg" width=300 height=200></img><img src="aa.jpg" width=200 height=200></img>

</body>
</html>

- **one.css :**

```
#style1{font-size:40;font-family:Algerian;font-style:normal;font-variant:normal;font-weight:bolder}
```

- **two.css :**

```
#style2{border-color:red green black cyan;border-style:dashed;border-width:10px}
```

- **Inline CSS :**

```
<html>
<head>
<title>Inline CSS</title>
<style type=text/css>
.a1{padding-bottom:50pixel;padding-top:50pixel;padding-left:50pixel;padding-right:50pixel}
.a2{margin-top:20pixel;margin-bottom:20pixel;margin-left:20pixel;margin-right:20pixel}
</style>
</head>
<body><center>
<h1 style="color:maroon;font-family:Lucida Calligraphy">~ * This is my 3rd Web Page ! * ~</h1>
<dl><p></center>
<h2><marquee><font face="Lucida Calligraphy" color=black>" * Inline CSS - Generic Selector - Padding Property and Margin Property ! * "</font></marquee></h2>
<center><h2><dt><font face="Lucida Calligraphy">* Flower : *</font></dt></h2></style></p>
```

<dd><h3 class=a1>A flower, sometimes known as a bloom or blossom, is the reproductive structure found in flowering plants (plants of the division Magnoliophyta, also called angiosperms).
</h3>
<h4 class=a2>Fertilization and dispersal :
Some flowers with both stamens and a pistil are capable of self-fertilization, which does increase the chance of producing seeds but limits genetic variation.
The extreme case of self-fertilization occurs in flowers that always self-fertilize, such as many dandelions. Conversely, many species of plants have ways of preventing self-fertilization. Unisexual male and female flowers on the same plant may not appear or mature at the same time, or pollen from the same plant may be incapable of fertilizing its ovules. The latter flower types, which have chemical barriers to their own pollen, are referred to as self-sterile or self-incompatible
</h4></dd>
</dl>
<img src="12.gif" height=300 width=200></img><img src="o.jpg" height=200 width=300></img><img src="12.gif" height=300 width=200></img>

</body>
</html>

- **Importing CSS :**

<html>
<head>
<title>Importing CSS</title>
<style type=text/css>
@import url('qwe.css');
*{font-family:Lucida Calligraphy}
</style></head>
<body>
<h1>~ * My 4th page ! * ~</h1>
<h1> * Origami : *</h1>
<h2><marquee>" * Importing CSS - Universal Selector - List Property and Shorthand Property used here ! * "</marquee></h2>

<h2><li class=q>There is much speculation about the origin of Origami.
<li class=q> It is not known when this practice started, but it seems to have become popular during the Sung Dynasty (905–1125 CE).

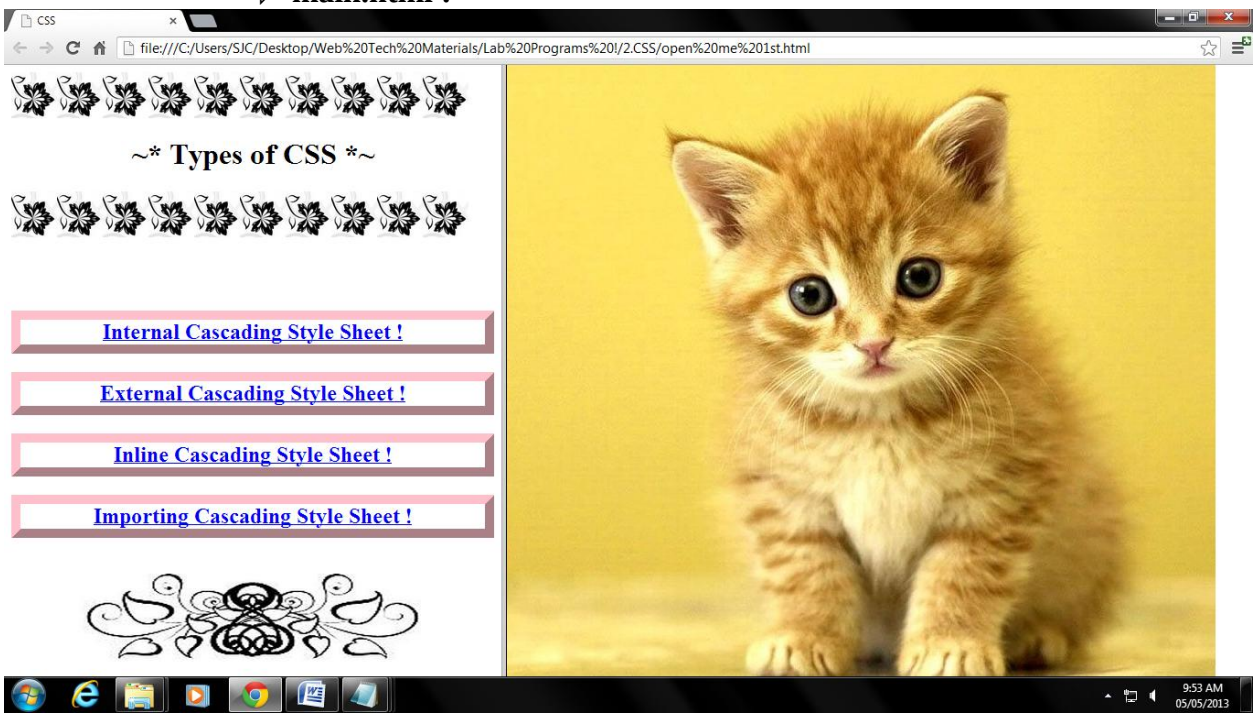

```
<li class=q> There is also evidence of a cut and folded paper box from
1440.</li>
</h2></center>
<img src="z.jpg" height=200 width=200></img><img src="n.jpg"
height=200 width=300></img><img src="x.jpg" height=200
width=200></img>
</body></html>
```

- **Qwe.css :**

```
<style type=text/css>
h1,h3{padding-left:100pixel}
li.q{list-style-image:url('op.jpg')}
</style>
```

* OUTPUT :

- **main.html :**

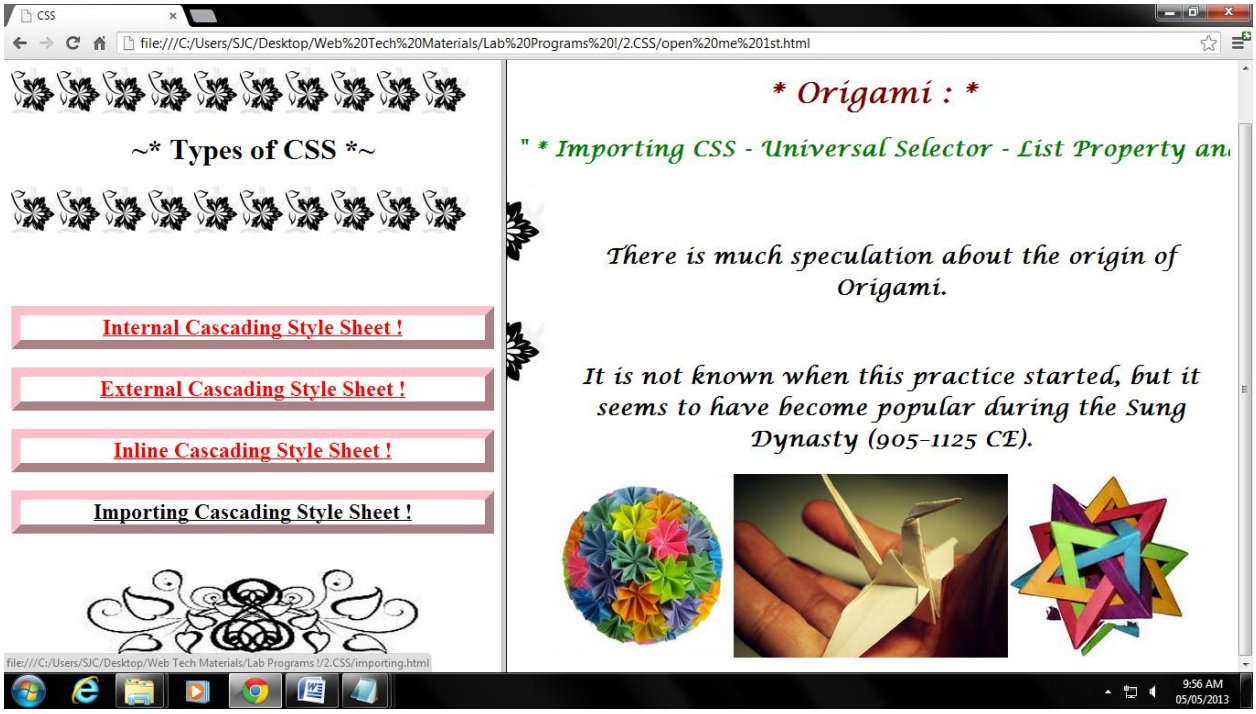


- **Internal CSS :**

↳ **External CSS :**

↳ **Inline CSS** :

↳ Importing CSS :



*** RESULT:**

Thus the different types of CSS was created and executed successfully.

EX NO:

* FORM VALIDATION *

* AIM :

To create a Client Side Scripts for Validating Web Form Controls using Java scripting .

* ALGORITHM :

- Create a form which include textbox,checkbox,radio button,select box,textarea reset button and a submit button.
- Validation script will ensure that the user enters their fields before the form is sent to the server.
- Open this page to see it in action.
- Try pressing the Submit button without filling anything in the text fields.
- The page consists of a JavaScript function called valied() that performs the form validation, followed by the form itself.

* PROGRAM CODING :

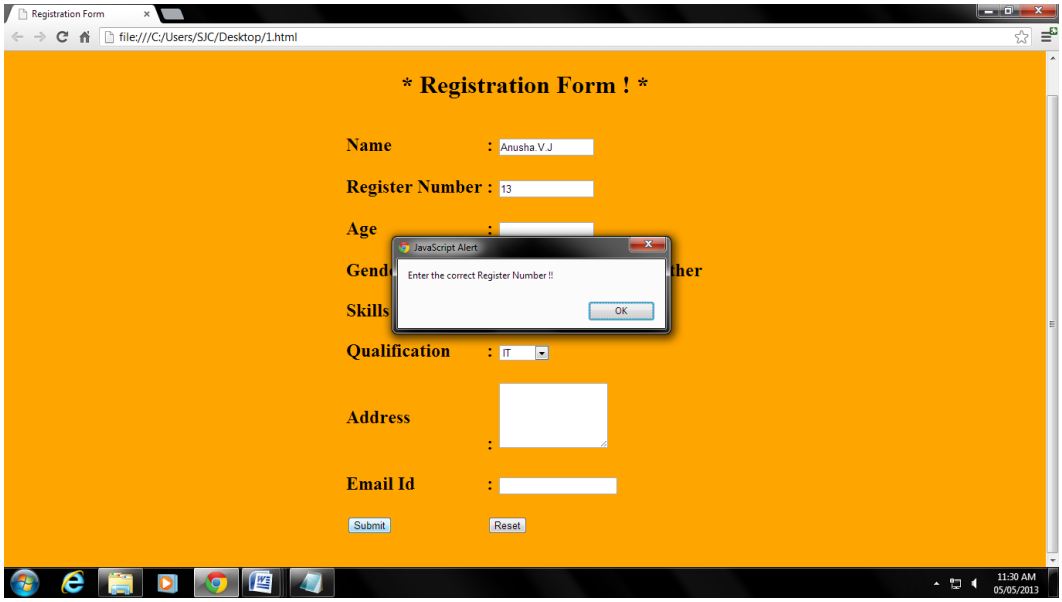
```
<html>
<head>
<title>Registration Form</title>
<script type=text/javascript>
function valied()
{
var name=document.f1.t1.value;
var regno=document.f1.t2.value;
var age=document.f1.t4.value;
if(name==null||name=="")
{
alert("Please Enter Your Name !");
return false;
}
if(regno>5||regno==null||regno=="")
{
alert("Enter the correct Register Number !!");
return false;
}
if(age<20||age>35||age==null||age=="")
{
alert("Enter your correct age !");
```

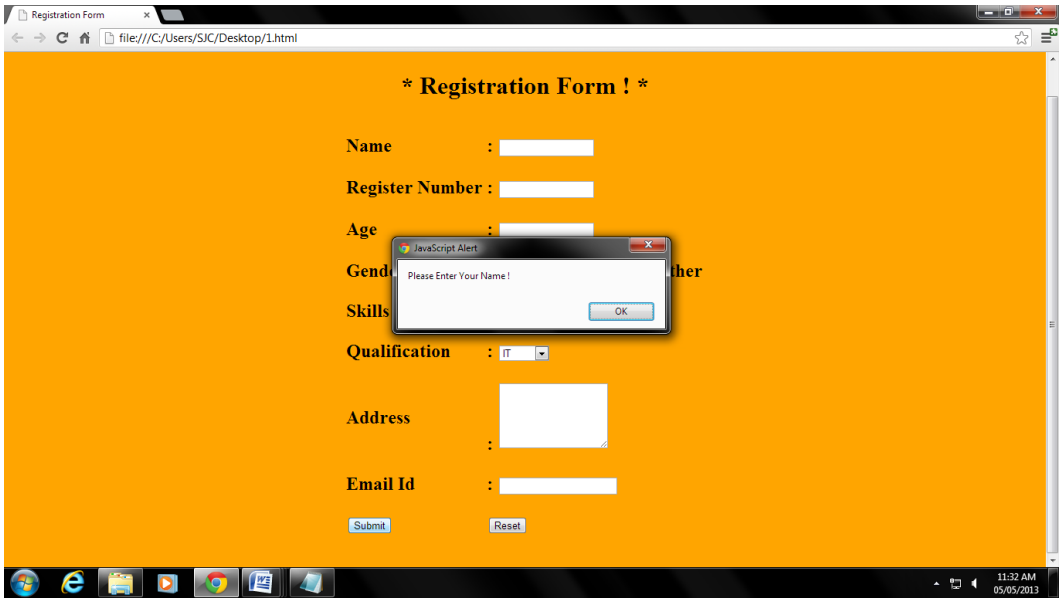
```
return false;
}
var index_at=str.indexOf("@")
var len=str.length
var index_dot=str.indexOf(".");
var emailID=document.f1.t5.value;
if((emailID.value==null)||(emailID.value==""))
{
alert("Please Enter ur email id");
emailID.focus()
return false
}
if(str.indexOf("@")==-1)
{
alert("Invalid email id")
return false
}
if(str.indexOf(".")==-1 ||str.indexOf(".")==0||str.indexOf(".")==index_at)
{
alert("invalid email id")
return false
}
if(str.indexOf("@",(index_at+1))!=-1)
{
alert("Invalid email id")
return false
}
if(str.indexOf("")!=-1)
{
alert("Invalid email id")
return false
}
if(!document.f1.r1.checked)
{
alert("Please select your Gender !");
return false;
}
return true;
}
</script>
</head>
<body bgcolor=orange>
<center>
<br>
<br>
<h1>* Registration Form ! *</h1>
```

```
<form name=f1 >
<br>
<table border=0 cellpadding=2>
<tr>
<td><h2>Name</h2></td>
<td><h2>: <input type=text size=15 name=t1></input></h2></td>
</tr>
<tr>
<td><h2>Register Number</h2></td>
<td><h2>: <input type=text size=15 name=t2></input></h2></td>
</tr>
<tr>
<td><h2>Age</h2></td>
<td><h2>: <input type=text size=15 name=t4></input></h2></td>
</tr>
<tr>
<td><h2>Gender</h2></td>
<td><h2>: <input type=radio name=r1>Male</input>
<input type=radio name=r1>Female</input>
<input type=radio name=r1>Other</input></h2></td>
</tr>
<tr>
<td><h2>Skills</td>
<td><h2>: <input type=checkbox name=c1>C++</input>
<input type=checkbox name=c1>Java</input>
<input type=checkbox name=c1>C</input></h2></td>
</tr>
<tr>
<td><h2>Qualification</h2></td>
<td><h2>:
<select>
<option>IT</option>
<option>ECE</option>
<option>EEE</option>
<option>CSE</option>
<option>EIE</option>
<option>CIVIL</option>
<option>MECH</option>
</select>
</h2><tr>
<td><h2>Address</h2></td>
<td><h2>: <textarea cols=15 rows=5></textarea></h2></td>
</tr>
<tr>
<td><h2>Email Id</h2></td>
<td colspan=2><h2>: <input type=text size=20 name=t5></h2></td>
```

```
</tr>
<tr>
<td><h1><input type=button value=Submit onclick=valied()></h1></td>
<td><h1><input type=reset value=Reset></h1></td>
</tr>
</table>
</form>
</center>
</body>
</html>
```

Output :





* RESULT:

Thus the form is created using HTML and validated using java script and executed successfully

.

EX NO:

* JAVA APPLET *

* AIM :

To write program in Java to create applets incorporating the following features:

- Create a color palette with matrix of buttons
- Set background and foreground of the control text area by selecting a color from color palette.
- In order to select Foreground or background use check box control as radio buttons
- To set background images

* PROCEDURE :

- Write an applet coding to change background , text color and image .
- Enter the required colors for background and text .
- Load an image source to the code to make it as the background image .
- Add the action listener function to make the web page listen to the commands .
- Write the applet code and specify the height and width of the window size .

* PROGRAM CODING :

- **Ex3.java :**

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class ex3 extends Applet implements ItemListener
{
int currcolor=5;
int flag=1;
String text="Click any of the button";
Button buttons[]=new Button[5];
String colors[]={"Red","Blue","Green","Yellow","Magenta"};
Image img;
```

```
CheckboxGroup cbg=new CheckboxGroup();
Checkbox box1=new Checkbox("Background color",cbg,true);
Checkbox box2=new Checkbox("Text color",cbg,false);
Checkbox box3=new Checkbox("Loading Image",cbg,false);
public void init()
{
for(int i=0;i<5;i++)
{
buttons[i]=new Button(" ");
add(buttons[i]);
}
buttons[0].setBackground(Color.red);
buttons[1].setBackground(Color.blue);
buttons[2].setBackground(Color.green);
buttons[3].setBackground(Color.yellow);
buttons[4].setBackground(Color.magenta);
add(box1);
add(box2);
add(box3);
box1.addItemListener(this);
box2.addItemListener(this);
box3.addItemListener(this);
}
public void itemStateChanged(ItemEvent ev)
{
if(box1.getState()==true)
flag=1;
else if(box2.getState()==true)
{
text="Default color is black";
flag=2;
}
else if(box3.getState()==true)
{
img=getImage(getDocumentBase(),"Sunset.jpg");
flag=3;
}
repaint();
}
public void paint(Graphics g)
{
if(flag==2)
{
g.drawString(text,30,100);
switch(currcolor)
{
```

```
case 0: g.setColor(Color.red);
break;
case 1: g.setColor(Color.blue);
break;
case 2: g.setColor(Color.green);
break;
case 3: g.setColor(Color.yellow);
break;
case 4: g.setColor(Color.magenta);
break;
case 5: g.setColor(Color.black);
break;
}
g.drawString(text,30,100);
}
else if(flag==1)
{
g.drawString(text,30,100);
switch(currcolor)
{
case 0: setBackground(Color.red);
break;
case 1: setBackground(Color.blue);
break;
case 2: setBackground(Color.green);
break;
case 3: setBackground(Color.yellow);
break;
case 4: setBackground(Color.magenta);
break;
case 5: setBackground(Color.black);
break;
}
}
else if(flag==3)
{
g.drawImage(img,20,90,this);
}
}
public boolean action(Event e,Object o)
{
for(int i=0;i<5;i++)
{
if(e.target==buttons[i])
{
currcolor=i;
```

```
text="You have chosen"+colors[i];
repaint();
}}
return false;
}}
```

↳ **main.html :**

```
/*<applet code=ex3 width=600 height=700>
</applet>*/
```

*** OUTPUT :**


app.html
file:///C:/Users/SJC/Desktop/Web%20Tech%20Materials/Lab%20Progra
Background color
Text color
Loading Image
You have chosenRed

*** RESULT:**

Thus the color palette is created and foreground, background color is changed using applet and executed successfull

EX NO:

* SERVLET PROGRAM *
* REGISTRATION FORM *

* AIM :

To create a servlet program to design a registration form and display the details in another page .

* ALGORITHM :

- Start the tomcat server.
- Now , type http://localhost:8080 in the link of internet explorer and press enter key .
- The home page of Tomcat server will be displayed.
- Create a folder reg in tomcat folder.
- Create an html document index.html with 4 text boxes to get input from the user and a button “Submit”.
- In the index.html, specify the folder and the java class name.
- Create a java code “reg.java” for server side operations to perform the display operation.
- Place the reg.class in the classes folder .
- Create an xml file for mapping.

* PROGRAM CODING :

Index.html :

```
<html>
<head>
<title>Registration Form</title>
</head>
<body>
<form action="http://localhost:8080/regform/a" method="get" name=f1>
<h1>Registration Form</h1>
<table border=0>
<tr>
<td>Name</td>
<td><input type=text size=15 name=t1></input></td>
</tr>
```

```
<tr>
<td>Register Number</td>
<td><input type=text size=15 name=t2></input></td>
</tr>
<tr>
<td>Age</td>
<td><input type=text size=15 name=t3></input></td>
</tr>
<tr>
<td>Email Id</td>
<td> <input type=text size=15 name=t4></input></td>
</tr>
<tr>
<td><input type=submit value=Submit></input></td>
<td><input type=reset value=Reset></input></td>
</tr>
</table>
</form>
</body>
</html>
```

- **web.Xml :**

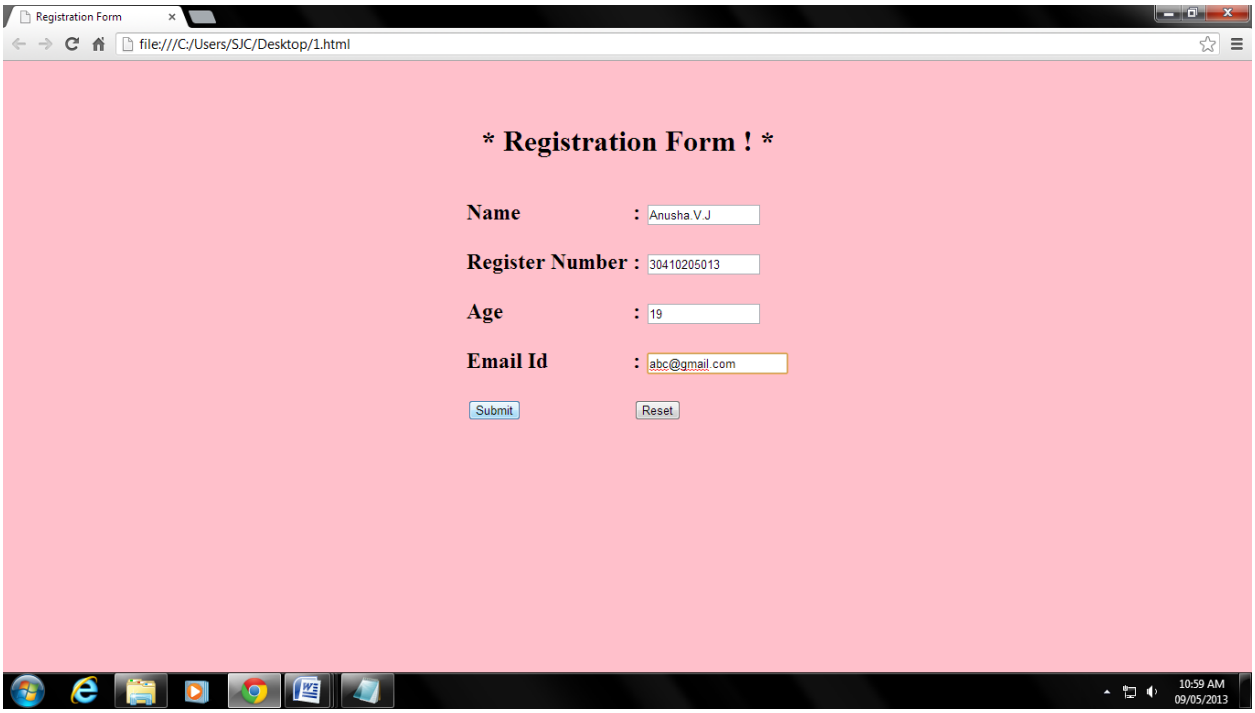
```
<web-app>
<servlet>
<servlet-name>a</servlet-name>
<servlet-class>a</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>a</servlet-name>
<url-pattern>/a</url-pattern>
</servlet-mapping>
</web-app>
```

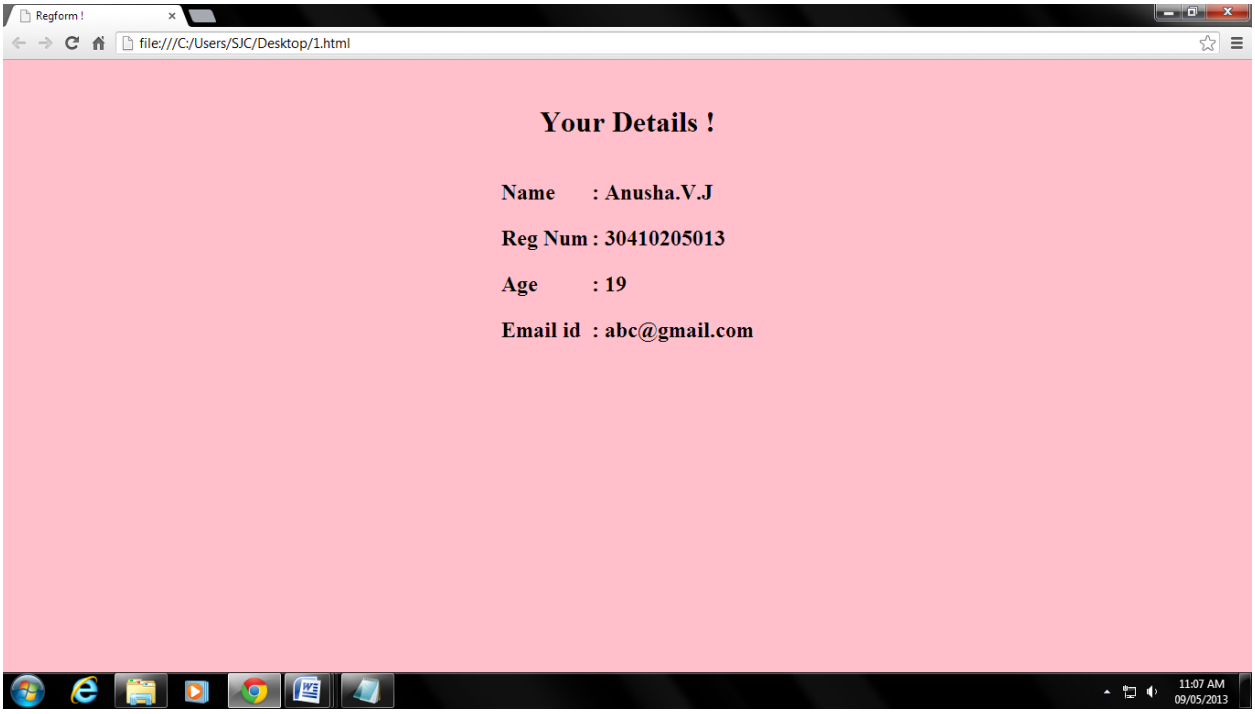
- **reg.java :**

```
import javax.servlet.*;
import java.io.*;
import java.util.*;
public class reg extends GenericServlet
{
public void service(ServletRequest rq,ServletResponse rs)throws
ServletException,IOException
```

```
{
String s1=rq.getParameter("t1");
String s2=rq.getParameter("t2");
String s3=rq.getParameter("t3");
String s4=rq.getParameter("t4");
PrintWriter out=rs.getWriter();
out.println(“<html>”);
out.println(“<head>”);
out.println(“<title>Reg</title>”);
out.println(“</head>”);
out.println(“<body>”);
out.println("Name = +s1”);
out.println("Reg Num = +s2”);
out.println("Age = +s3”);
out.println("Email ID = s4”);
out.close();
}
}
```

*** OUTPUT :**




Regform !
file:///C:/Users/SJC/Desktop/1.html
Your Details !
Name : Anusha.V.J
Reg Num : 30410205013
Age : 19
Email id : abc@gmail.com
11:07 AM
09/05/2013

* RESULT :

Thus a servlet code was created to display the registration form and the output was verified successfully .

EX.NO:

* INVOKE SERVLET FROM APPLET *

* AIM :

To create and implement a servlet program using applet to get and set the date, day, year and time.

* ALGORITHM:

- Start the invoking servlet from applet.
- Create the allpet_servlets.html.
- Create one text area for displaying the date and time and a button.
- Then create and save coding for daytimeapplet.java.
- Now save and run the program.

* PROGRAM CODING:

- **index.html :**

```
<html>
<head>
<title>Applet Servlet Communication Page</title>
</head>
<body>
<h3><hr width="100%">Applet Servlet Communication Page<hr width="100%"></h3>
<applet codebase="http://localhost:8080/echo" code="daytimeapplet.class" width=350
height=200>
<param name="servletURL" value="http://localhost:8080/echo/daytimeservlet">
</applet>
</body>
</html>
```

- **daytimeapplet.java :**

```
import javax.swing.*;



import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.util.*;
import java.net.*;
public class daytimeapplet extends java.applet.Applet implements ActionListener {
TextField dt = new TextField();
Label dl = new Label("The Daytime : ",Label.RIGHT);
Button db = new Button("Get the Time");
public void init()
{
setLayout(new BorderLayout());
add("West",dl);
add("Center",dt);
db.addActionListener(this);
add("South",db);
setVisible(true);
}
public void actionPerformed(java.awt.event.ActionEvent p1) {
dt.setText(getdate());
}
private String getdate()
{
String ts="";
String argstring="",urlString="";
```

```
try{
urlString = getParameter("servletURL");
URL url = new URL(urlString);
URLConnection con = url.openConnection();
con.setUseCaches(false);
InputStream in = con.getInputStream();
BufferedReader result = new BufferedReader(new InputStreamReader(in));
ts = result.readLine();
in.close();
return ts;
}
catch(Exception e)
{
System.out.println("Error\n"+e);
}
return "Security Imposed.. Unable to read the time from the server";
}




}
```

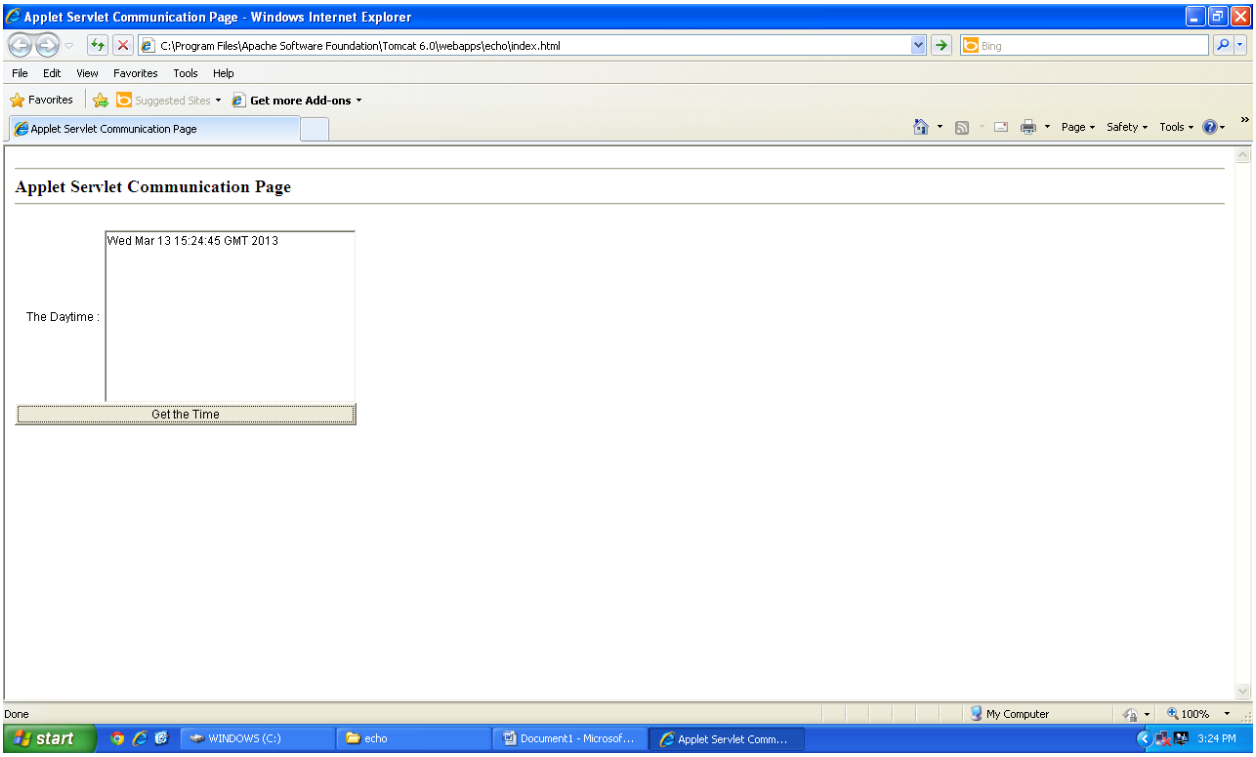
- **daytimeservlet.java :**

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class daytimeservlet extends HttpServlet
{
public void doGet(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException
{
res.setContentType("text/plain");
PrintWriter out = res.getWriter();
out.println(new java.util.Date().toString());
}
public void doPost(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException
{
doGet(req,res);
}
}
```

- **web.xml :**

```
<!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>
<!-- General description of your web application -->
<display-name>Echo Servlet</display-name>
<description>
Echo Servlet
</description>
<!-- define servlets and mapping -->
<servlet>
<servlet-name>daytimeservlet</servlet-name>
<servlet-class>daytimeservlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>daytimeservlet</servlet-name>
<url-pattern>/daytimeservlet</url-pattern>
</servlet-mapping>
</web-app>
```

* OUTPUT:



*** RESULT :**

Thus the program to invoke servlet from applet was executed successfully and verified for sample input and output.

EX NO :

* IMPLEMENTING XML USING CSS & XSL FILES *

*** AIM :**

To write programs using XML – using CSS and XSL files .

*** PROCEDURE:**

- Write 2 simple XML documents and save it with .xml extension .
- Now create a XSL Style sheet and save it as .xsl and create a CSS file and save it as .css .
- Insert .css in one of the XML document and execute it .
- Insert .xsl in another XML document and execute it .

*** PROGRAM CODING :**

- **one.css :**

```
title,country,company,price,year
{color:green;font-family:Lucida Calligraphy}
```

- **test-css.xml :**

```
<?xml-stylesheet type="text/css" href="one.css"?>
<catalog>
<cd><h2>
<title>Avathar</title>
<artist>Bob Dylan</artist>
<country>USA</country>
<company>Columbia</company>



<price>10.90</price>
<year>1985</year>
</h2> </cd>
<cd>
<title>Terminator</title>
<artist>Bonnie Tyler</artist>
<country>UK</country>
<company>CBS Records</company>
<price>9.90</price>
<year>1988</year>
</cd>
<cd>
<title>Greatest Hits</title>
<artist>Dolly Parton</artist>
<country>USA</country>
<company>RCA</company>
<price>9.90</price>
<year>1982</year>
</cd>
<cd>
<title>Eros</title>
<artist>Eros Ramazzotti</artist>
<country>EU</country>
<company>BMG</company>
<price>9.90</price>
<year>1997</year>
</cd>
</catalog>
```

- **two.xsl :**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
```

```
<center>
<h2>* My CD Collection *</h2>
<table border="1">
<tr bgcolor="purple">
<th>Title</th>
<th>Artist</th>
</tr>
<xsl:for-each select="catalog/cd">
<tr>
<td><xsl:value-of select="title"/></td>
<td><xsl:value-of select="artist"/></td>
</tr>
</xsl:for-each>
</table>
</center>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

↳ test.xml :

```
<?xml-stylesheet type="text/xsl" href="two.xsl"?>
<catalog>
<cd>
<title>Avathar</title>
<artist>Bob Dylan</artist>
<country>USA</country>
<company>Columbia</company>
<price>10.90</price>
<year>1985</year>
</cd>
<cd>
<title>Terminator</title>
<artist>Bonnie Tyler</artist>
<country>UK</country>
<company>CBS Records</company>



<price>9.90</price>
<year>1988</year>
```

```
</cd>
<cd>
<title>Greatest Hits</title>
<artist>Dolly Parton</artist>
<country>USA</country>
<company>RCA</company>
<price>9.90</price>
<year>1982</year>
</cd>
<cd>
<title>Eros</title>
<artist>Eros Ramazzotti</artist>
<country>EU</country>
<company>BMG</company>
<price>9.90</price>
<year>1997</year>
</cd>
</catalog>
```

*** OUTPUT :**

Avathar Bob Dylan *USA Columbia 10.90 1985 Terminator* Bonnie Tyler *UK CBS Records 9.90 1988 Greatest Hits* Dolly Parton *USA RCA 9.90 1982 Eros* Eros Ramazzotti *EU BMG 9.90 1997*

My CD Collection

Title	Artist
Avathar	Bob Dylan
Terminator	Bonnie Tyler
Greatest Hits	Dolly Parton
Eros	Eros Ramazzotti

*** RESULT :**

Thus the programs using XML was created and implemented successfully.

EX NO :

* AJAX *

*** AIM :**

To develop a server side scripting program using AJAX and mouse listener .

*** ALGORITHM :**

- Create an html document with loadXMLDoc() function to send HTTP request to the server and receive back a response containing XML document.

- Create an XMLHttpRequest object.
- Use ActiveXObject to construct an XMLHttpRequest instance for web browsers .
- The OnreadyStateChanges() function is triggered when the ready state changes.
- The open function is used to specify the request , url and whether the request should be handled asynchronously .

* PROGRAM CODING :

Index.html :

```
<html>
<head>
<style type="text/css">
img { float:Right; }
h1 {font-family:script;color:black}
</style>
<script type="text/javascript">
function a()
{
var xmlhttp;
if (window.XMLHttpRequest)



{// code for IE7+, Firefox, Chrome, Opera, Safari
xmlhttp=new XMLHttpRequest();
}
else
{// code for IE6, IE5
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.onreadystatechange=function()
{
if (xmlhttp.readyState==4 && xmlhttp.status==200)
{
document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
}
```

```
}
xmlhttp.open("GET","2.txt",true);
xmlhttp.send();
}
</script>
</head>
<body>
<img src="t.jpg" onmouseover="a()"></img>
<center>
<br>
<br>
<h1 id="myDiv"> ~ * Ninja ! * ~</h1>
</center></body>
</html>
```

2.txt :

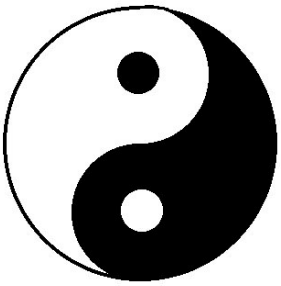
A ninja or shinobi was a covert agent or mercenary in feudal Japan who specialized in unorthodox warfare. The functions of the ninja included espionage, sabotage, infiltration, and assassination, and open combat in certain situations.[1] Their covert methods of waging war contrasted the ninja with the samurai, who observed strict rules about honor and combat.[2] The shinobi proper, a specially trained group of spies and mercenaries, appeared in the Sengoku or "warring states" period, in the 15th century.

* OUTPUT :

index.html

file:///C:/Users/SJC/Desktop/Ajx/index.html

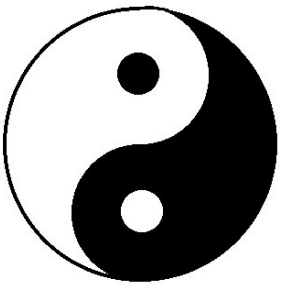
~ * Ninja ! * ~



1.html

file:///C:/Users/SJC/Desktop/Ajx/1.html

7:43 PM
07/04/2013



A ninja or shinobi was a covert agent or mercenary in feudal Japan who specialized in unorthodox warfare. The functions of the ninja included espionage, sabotage, infiltration, and assassination, and open combat in certain situations.[1] Their covert methods of waging war contrasted the ninja with the samurai, who observed strict rules about honor and combat.[2] The shinobi proper, a specially trained group of spies and mercenaries, appeared in the Sengoku or "warring states" period, in the 15th century,[3] but antecedents may have existed in the 14th century,[4] and possibly even in the 12th century (Heian or early Kamakura era).[5][6] In the unrest of the Sengoku period (15th–17th centuries), mercenaries and spies for hire became active in the Iga Province and the adjacent area around the village of Koga, and it is from their ninja clans that much of our knowledge of the ninja is drawn. Following the unification of Japan under the Tokugawa shogunate (17th century), the ninja faded into obscurity, being replaced by the Oniwabanshu body of secret agents.[7] A number of shinobi manuals, often centered around Chinese military philosophy, were written in the 17th and 18th centuries, most notably the Bansenshukai (1676).[8] By the time of the Meiji Restoration (1868), the tradition of the shinobi had become a topic of popular imagination and mystery in Japan. Ninja figured prominently in folklore and legend, and as a result it is often difficult to separate historical fact from myth. Some legendary abilities purported to be in the province of ninja training include invisibility, walking on water, and control over the natural elements. As a consequence, their perception in western popular culture in the 20th century was based more on such legend and folklore than on the historical spies of the Sengoku period.

7:44 PM
07/04/2013

*** CONCLUSION :**

Thus a server side scripting program using AJAX was developed and executed and the output was verified successfully

EX NO:

* WEB SERVICES *

*** AIM :**

To implement a case where we have two web Services- an airline service and a travel agent and the travel agent is searching for an airline. Implement this scenario using Web Services and Data base.

*** PROCEDURE :**

- Create a database in Access and name it as Airways.mdb . Inside the database , name a table Airways_table is created. Create a database DSN for it.
- Create a folder named AirReservation on C drive . Inside this folder create a folder named WEB_INF , inside that, create two folders namely src and classes. Inside these folders create one more folder named airinfo. Now open the src directory and inside which is named airinfo directory. In this directory create two JAVA files. The first java file defines the interface and another java program actually an airline service.
- Now open command prompt window and change it to WEB_INF. Then execute above program by using Javac.
- Now create a file named config.xml at the location C:\AirReservation .

- Create a jaxrpc-ri.wml in the folder C:\AirReservation WEB_INF.
- Write config.xml file inside the WEB_INF directory in this config.xml file the web service is specified.
- Open the command prompt window and change it to C:\your_tomcat_directory\AirReservationClient\WEB_INF\classes and issue the command javac myairreservationclient\AirReservation on client program java.
- Open the input form.html file and type the necessary inputs . Click submit and required form will be displayed.

* PROGRAM CODING :

Server : MyInterface.java :

```
package airinfo;
import java.rmi.RemoteException;
public interface MyInterface extends java.rmi.Remote
{
public String getInfo(String src,String dest)throws java.rmi.RemoteException;
}
TestDB.java:
package airinfo;
import java.sql.*;
public class TestDB implements MyInterface
{
public String getInfo(String src,String dest)
{
String str="";
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con=DriverManager.getConnection("jdbc:odbc:Airways","","");
Statement s=con.createStatement();
if(src.equals("Mumbai"))
```

```
{
if(dest.equals("Pune"))
s.executeQuery("SELECT * FROM Airways_table WHERE Start_from='Mumbai' AND
Dest='Pune'");
else if(dest.equals("Chennai"))
s.executeQuery("SELECT * FROM Airways_table WHERE Start_from='Mumbai' AND
Dest='Chennai'");
}
else if(src.equals("Pune"))
{
if(dest.equals("Mumbai"))
s.executeQuery("SELECT * FROM Airways_table WHERE Start_from='Pune' AND
Dest='Mumbai'");
else if(dest.equals("Chennai"))



s.executeQuery("SELECT * FROM Airways_table WHERE Start_from='Pune' AND
Dest='Chennai'");
}
else if(src.equals("Chennai"))
{
if(dest.equals("Pune"))
s.executeQuery("SELECT * FROM Airways_table WHERE Start_from='Chennai' AND
Dest='Pune'");
else if(dest.equals("Mumbai"))
s.executeQuery("SELECT * FROM Airways_table WHERE Start_from='Chennai' AND
Dest='Mumbai'");
}
ResultSet rs=s.getResultSet();
str+="<table border=1>";
while(rs.next())
{
str+="<tr><b>";
str+="<td>";
str+=rs.getString("name");
str+="</td>";
str+="<td>";
str+=rs.getString("Time");
str+="</td>";
str+="</tr></b>";
}
str+="</table>";
}catch(ClassNotFoundException ex)
{
```

```
System.out.println(ex);
}
catch(SQLException ex)
{
System.out.println(ex);
}
return str;
}
}
```

AirLineReservation.wsdl:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="AirLineReservation" targetNamespace="http://tempuri.org/wsdl"
xmlns:tns="http://tempuri.org/wsdl"



xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
<types/>
<message name="MyInterface_getInfo">
<part name="String_1" type="xsd:string"/>
<part name="String_2" type="xsd:string"/></message>
<message name="MyInterface_getInfoResponse">
<part name="result" type="xsd:string"/></message>
<portType name="MyInterface">
<operation name="getInfo" parameterOrder="String_1 String_2">
<input message="tns:MyInterface_getInfo"/>
<output message="tns:MyInterface_getInfoResponse"/></operation></portType>
<binding name="MyInterfaceBinding" type="tns:MyInterface">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/>
<operation name="getInfo">
<soap:operation soapAction=""/>
<input>
<soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"
namespace="http://tempuri.org/wsdl"/></input>
<output>
<soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"
namespace="http://tempuri.org/wsdl"/></output></operation></binding>
<service name="AirLineReservation">
<port name="MyInterfacePort" binding="tns:MyInterfaceBinding">
<soap:address
location="REPLACE_WITH_ACTUAL_URL"/></port></service></definitions>
```

jaxrpc-ri.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<webServices
xmlns="http://java.sun.com/xml/ns/jax-rpc/ri/dd"
version="1.0"
targetNamespaceBase="http://tempuri.org/wsdl"
typeNamespaceBase="http://tempuri.org/types"
urlPatternBase="/AirTicket">
<endpoint
name="AirInformation"
displayName="AirLine Reservation"
description="This service displays status of Airways"
interface="airinfo.MyInterface"



model="/WEB-INF/model.xml.gz"
implementation="airinfo.TestDB"/>
<endpointMapping
endpointName="AirInformation"
urlPattern="/air" />   </webServices>
```

web.xml:

```
<web-app
xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
version="2.4">
<display-name>AirLine Reservation</display-name>
<description>
This web page displays AirLine status
</description>
<session-config>
<session-timeout>60</session-timeout>
</session-config>
</web-app>
```

Client : AirRClientProg.java:

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

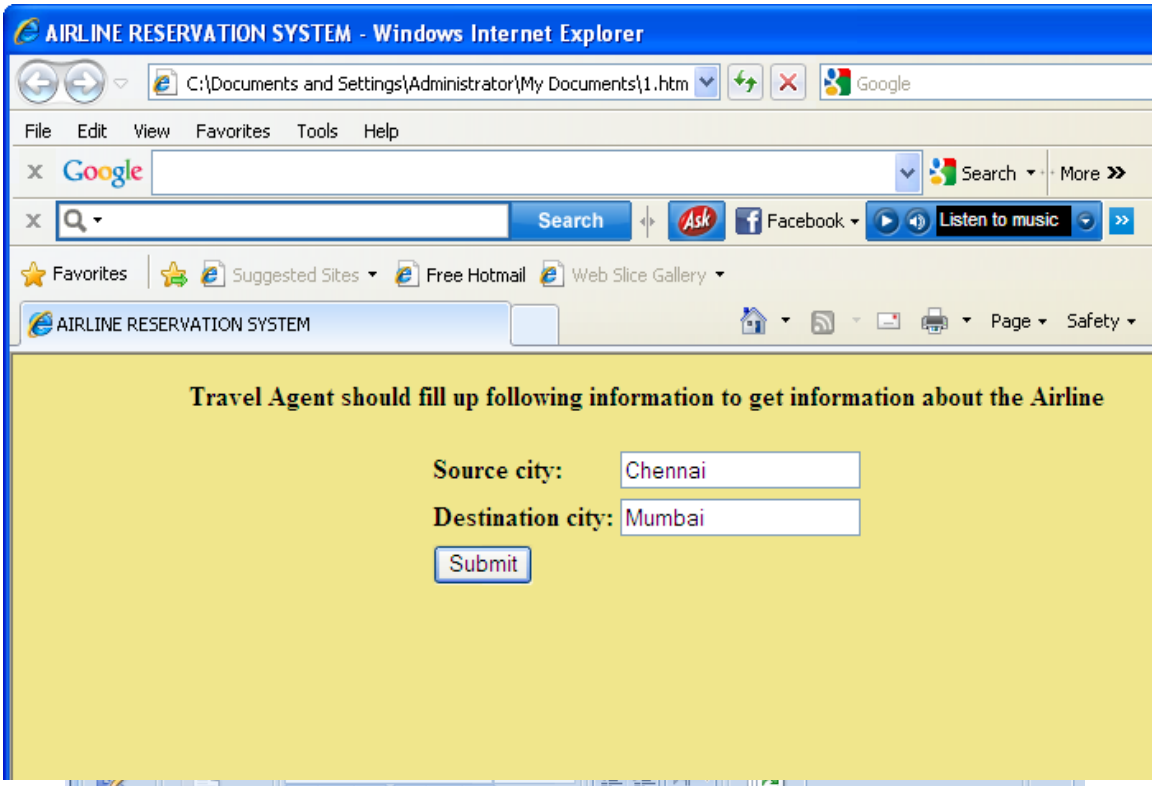
```
public class AirRClientProg extends HttpServlet
{
public void doGet(HttpServletRequest req,HttpServletResponse res)
throws IOException,ServletException
{
res.setContentType("text/html");
PrintWriter out=res.getWriter();
String s=req.getParameter("Src_name");
String d=req.getParameter("Dest_name");
AirLineReservation_Impl service =  new AirLineReservation_Impl();
MyInterface stub = service.getMyInterfacePort();
String response = stub.getInfo(s,d);
out.println("<html>");
out.println("<body bgcolor='khaki' >");



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

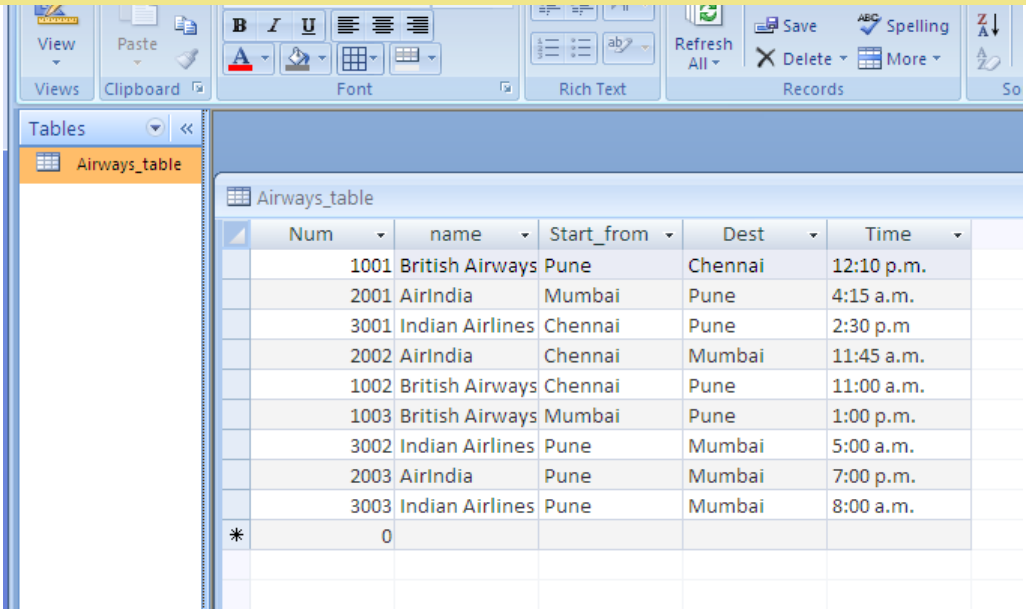
Inputform.html:

```
<html>
<head>
<title>AIRLINE RESERVATION SYSTEM</title>
</head>
<body bgcolor="khaki">
<center>
<h4>Travel Agent should fill up following information to get information about the
Airline</h4>
<form name="form1" method=GET action ="http://localhost:8080/servlets-
examples/servlet/AirRClientProg">
<table>
<tr>
<td><b>Source city: </b></td>
<td><input type="text" name="Src_name" size="20" value=""></td>
</tr>
<tr>
<td><b>Destination city: </b></td>
<td><input type="text" name="Dest_name" size="20" value=""></td>
</tr>
<tr>
<td><input type="submit" value="Submit"></td>
```

```
</tr>
</table>
</form>
</center>
</body>
</html>
```

*** OUTPUT :**





Tables

Airways_table

Airways_table

Num	name	Start_from	Dest	Time
1001	British Airways	Pune	Chennai	12:10 p.m.
2001	AirIndia	Mumbai	Pune	4:15 a.m.
3001	Indian Airlines	Chennai	Pune	2:30 p.m
2002	AirIndia	Chennai	Mumbai	11:45 a.m.
1002	British Airways	Chennai	Pune	11:00 a.m.
1003	British Airways	Mumbai	Pune	1:00 p.m.
3002	Indian Airlines	Pune	Mumbai	5:00 a.m.
2003	AirIndia	Pune	Mumbai	7:00 p.m.
3003	Indian Airlines	Pune	Mumbai	8:00 a.m.
0				

* RESULT:

Thus the web services for airline reservation was created and executed successfu

EX NO:

* POSITIONING *

* AIM :

Create a web page to display 3 different types of Positioning ,

- Relative Positioning .
- Absolute Positioning.
- Float Positioning .

* TAGS USED WITH SYNTAX :

- **Relative Positioning :**
  - **Syntax** : {position:relative;top:-10;width:10;height:5}
- **Absolute Positioning :**
  - **Syntax** : {position:absolute;top:-10;width:10;height:5}

- **Float Positioning :**

  - **Syntax** : {position:float;top:-10;width:10;height:5}

* ALGORITHM :

- Create a simple HTML page.
- Write relative positioning using <style> tag in the same page to create superscript and subscript .
- Create another file using absolute positioning within <style> tag with extension .
- Create a web page using float positioning to display text and image side by side .
- Create a main page to include all the above 3 web pages in it .

* PROGRAM CODING :

- **Main.html :**

```
<html>
<head>
<title>Positions</title>
<frameset cols=40%,60%>
<frame src=2.html name=f1></frame>
<frame src=a.jpg name=f2></frame>
</frameset>
</head>
</html>
```

- **2.html :**

```
<html>
<head>
<title>Source</title>
<style type=text/css>
a:visited{color:red}
a:link{color:blue}
```

```
a:active{color:green}
a:hover{color:black}
h2.rr{border-color:pink;border-style:outset;border-width:10;cursor:move}
h2.rr1{border-color:pink;border-style:outset;border-width:10;cursor:progress}
h2.rr2{border-color:pink;border-style:outset;border-width:10;cursor:wait}
</style>
</head>
<body>
<img src="op.jpg" width=50 height=50></img><img src="op.jpg" width=50
height=50></img><img src="op.jpg" width=50 height=50></img><img src="op.jpg"
width=50 height=50></img><img src="op.jpg" width=50 height=50></img><img
src="op.jpg" width=50 height=50></img><img src="op.jpg" width=50
height=50></img><img src="op.jpg" width=50 height=50></img><img src="op.jpg"
width=50 height=50></img><img src="op.jpg" width=50 height=50></img>
<center><h1>~* Types of Position *~</h1></center>
<img src="op.jpg" width=50 height=50></img><img src="op.jpg" width=50
height=50></img><img src="op.jpg" width=50 height=50></img><img src="op.jpg"
width=50 height=50></img><img src="op.jpg" width=50 height=50></img><img
src="op.jpg" width=50 height=50></img><img src="op.jpg" width=50
height=50></img><img src="op.jpg" width=50 height=50></img><img src="op.jpg"
width=50 height=50></img><img src="op.jpg" width=50 height=50></img>



<center>
<br>
<br>
<br>
<h2 class=rr><a href=rel.html target=f2>Relative Positioning !</a></h2>
<h2 class=rr1><a href=abs.html target=f2 class=rr>Absolute Positioning !</a></h2>
<h2 class=rr2><a href=float.html target=f2 class=rr>Float Positioning !</a></h2>
<br>
<img src="az.jpg" height=100 width=400></img>
</center>
</body>
</html>
```

- **Relative Positioning :**

```
<html>
<head>
<title>Relative Positioning</title>
<style type=text/css>
.super{position:relative;top:-10px;width:10;height:5}
.sub{position:relative;bottom:-10px;width:10;height:5}
</style></head>
```

```
<body bgcolor=pink>
<center>
<br>
<br>
<br>
<br>
<h1>~ * Maths Formulas ! * ~</h1>
<br>
<h2>(a+b)<span class=super>2</span>=a<span class=super>2</span>+b<span
class=super>2</span>+2ab</h2>
<h2>(a-b)<span class=super>2</span>=a<span class=super>2</span>+b<span
class=super>2</span>-2ab</h2>
<br>
<h1>~ * Chemical Formulas ! * ~</h1>
<br>
<h2>2H<span class=sub>2</span>O+O<span class=sub>2</span>=2H<span
class=sub>2</span>O</h2>
</center>
</body>
</html>
```

- **Absolute Positioning** :

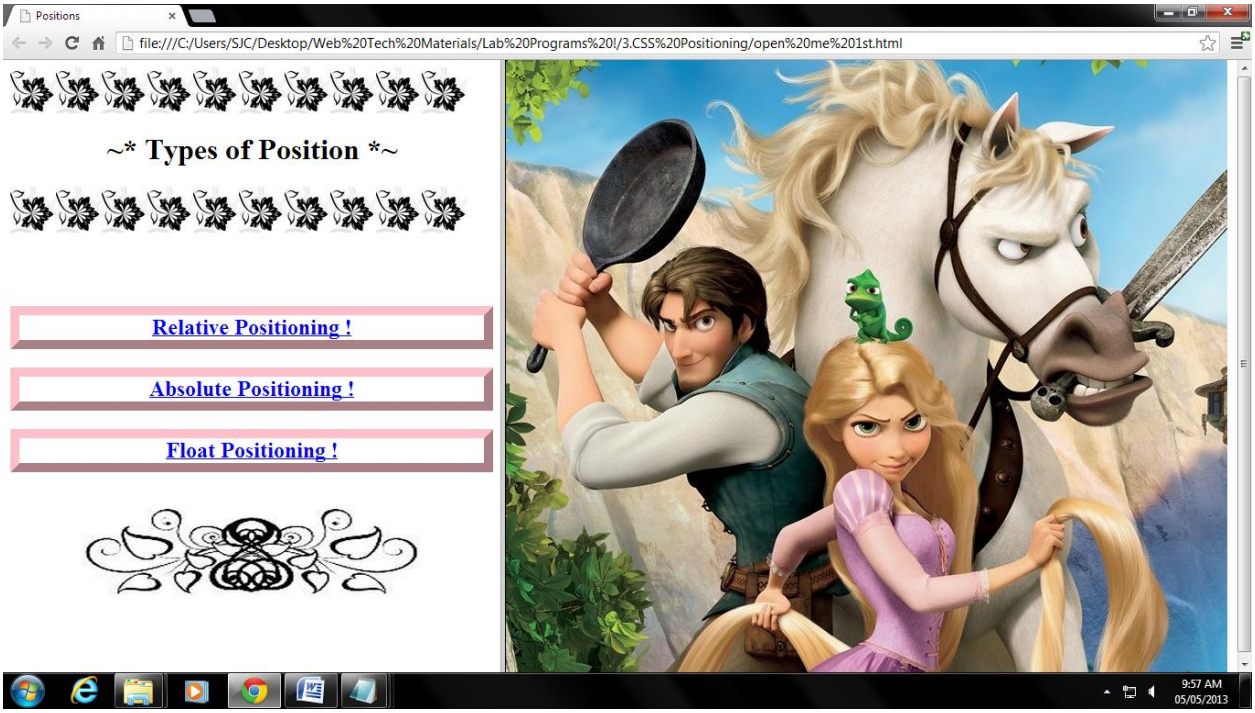
```
<html>
<head>
<title>Absolute Positioning</title>
<style type=text/css>
.abs{position:absolute;width:15;height:10}
</style>
</head>
<body>
<center>
<br>
<h1>~ * 7 Wonders of World ! * ~</h1></center>
<br>
<img src="7.jpg" height=300 width=750></img>
<br>
<h2><span class=abs>Pyramid_of_Giza Hanging_Gardens Statue_of_Zeus
Temple_of_Artemis Taj_Mahal Colossus_of_Rhodes
Lighthouse_of_Alexandria</span></h2>
</body>
</html>
```

- Float Positioning :

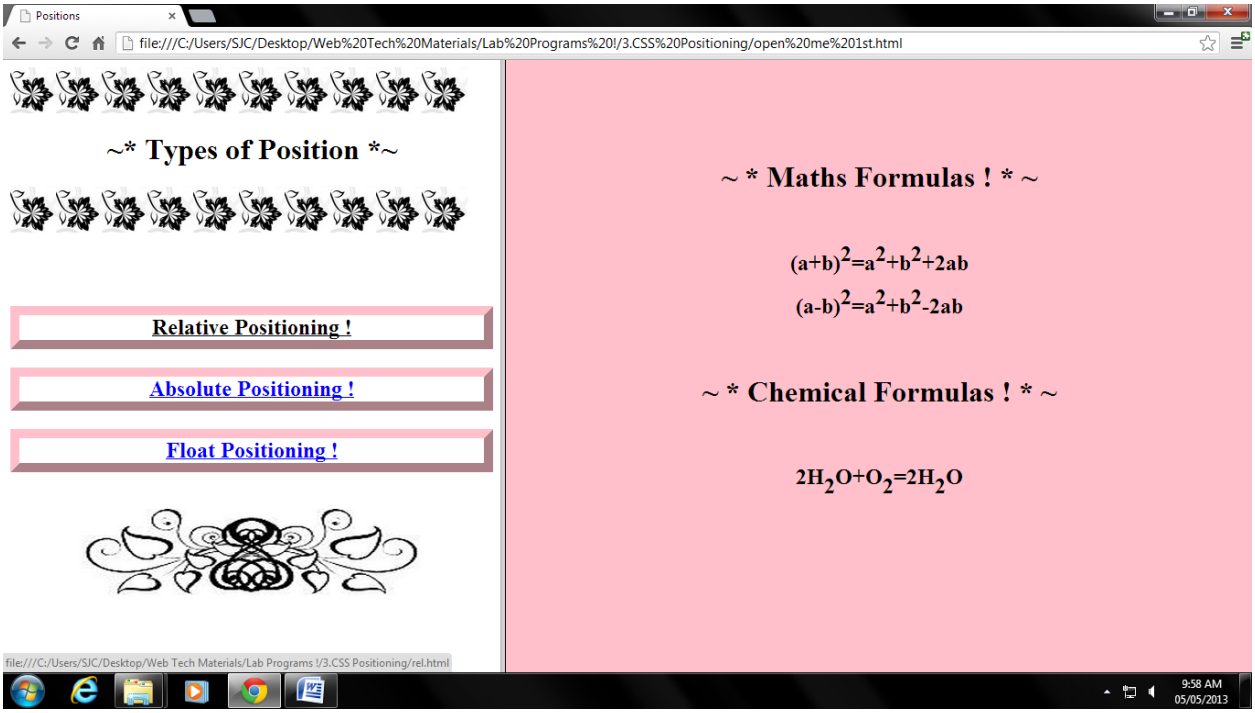
```
<html>
<head>
<title>Floating Point</title>
<style type=text/css>
img{float:right}
</style>
</head>
<body>
<br>
<center><h1>~ * Spidy * ~</h1></center>
<img src="w.jpg" height=400 width=500></img>
<h2>Spider-Man is a fictional character, a comic book superhero who appears in comic books published by Marvel Comics. Created by writer-editor Stan Lee and writer-artist Steve Ditko, he first appeared in Amazing Fantasy #15 (August 1962). Lee and Ditko conceived the character as an orphan being raised by his Aunt May and Uncle Ben, and as a teenager, having to deal with the normal struggles of adolescence in addition to those of a costumed crimefighter. Spider-Man's creators gave him super strength and agility, the ability to cling to most surfaces, shoot spider-webs using wrist-mounted devices of

his own invention which he called "web-shooters", and react to danger quickly with his "spider-sense", enabling him to combat his foes.
</h2>
</body>
</html>
```

* OUTPUT :

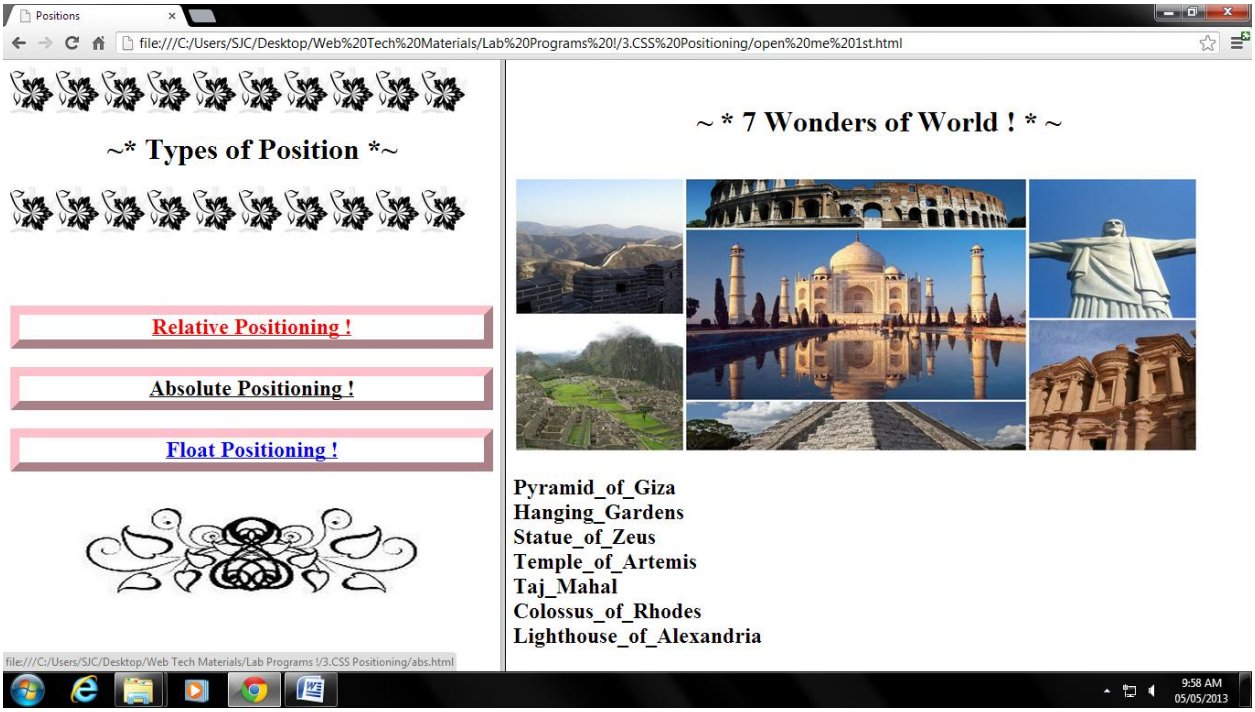
- **Main page :**



↳ Relative Positioning :



↳ **Absolute Positioning :**



↳ **Float Positioning :**

Positions

file:///C:/Users/SJC/Desktop/Web%20Tech%20Materials/Lab%20Programs%20!/3.CSS%20Positioning/open%20me%201st.html

~* Types of Position *~

Relative Positioning !

Absolute Positioning !

Float Positioning !

~ * Spidy * ~

Spider-Man is a fictional character, a comic book superhero who appears in comic books published by Marvel Comics. Created by writer-editor Stan Lee and writer-artist Steve Ditko, he first appeared in Amazing Fantasy #15 (August 1962). Lee and Ditko conceived the character as an orphan being raised by his Aunt May and Uncle Ben, and as a teenager, having to deal with the normal struggles of adolescence in addition to those of a costumed crimefighter. Spider-Man's creators gave him super strength and agility, the ability to cling to most surfaces, shoot spider-webs using wrist-mounted devices of his own invention which he called "web-shooters", and react to danger quickly with his "spider-sense", enabling him to combat his foes.

file:///C:/Users/SJC/Desktop/Web Tech Materials/Lab Programs !/3.CSS Positioning/float.html

9:58 AM
05/05/2013

*** RESULT:**

Thus all the 3 types of positioning were implemented in web pages and was executed successfully.

EX NO :

* WINDOW OBJECT *

*** AIM :**

To create a frameset , a textbox and two buttons and execute window object using this code .

*** ALGORITHM :**

- Create a frameset in which one part contains the main page of mainframe.html
- In that create a text area using <textarea> .
- Create two buttons “newframe.html” and “newwin.html”.
- The html code is written in the way that , if the new frame button is clicked , then the text entered in the textbox is shown in the next frame that is on the right side frame of the main screen.
- If the new window button is clicked , then the text entered in the text area will be shown in the new window.

*** PROGRAM CODING :**

- **1.html :**

<html>

```
<head>
<script type="text/javascript">
function Nextframe()
  {
  var w=window.parent.f2;
var code=t2.value;
w.document.write(code);
w.document.close();
  }
function Newframe()
{



var w=window.open();
w.document.open();
var code=t2.value;
w.document.write(code);
w.document.close();
}
</script>
</head>
<body>
<textarea cols=20 rows=20 name=t2></textarea>
<br>
<input type="button" value="New document" onclick="Nextframe()" />
<input type="button" value="New document" onclick="Newframe()" />
</body>
</html>
```

- **blank.html :**

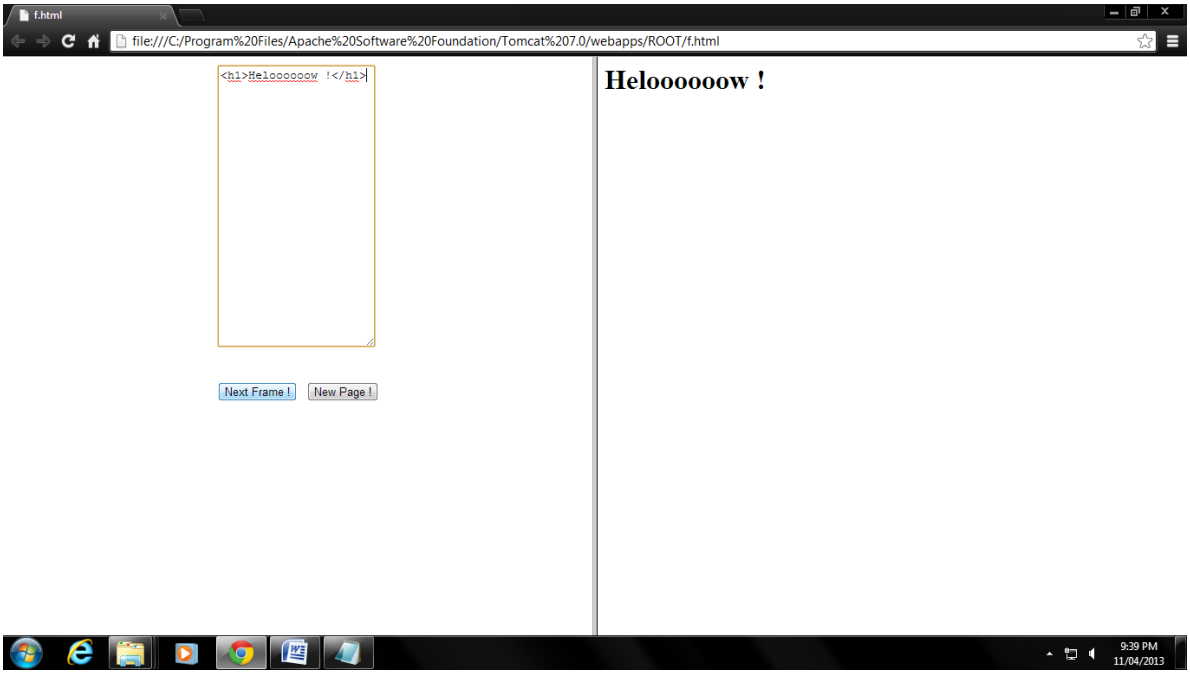
```
<html>
<body>
</body>
</html>
```

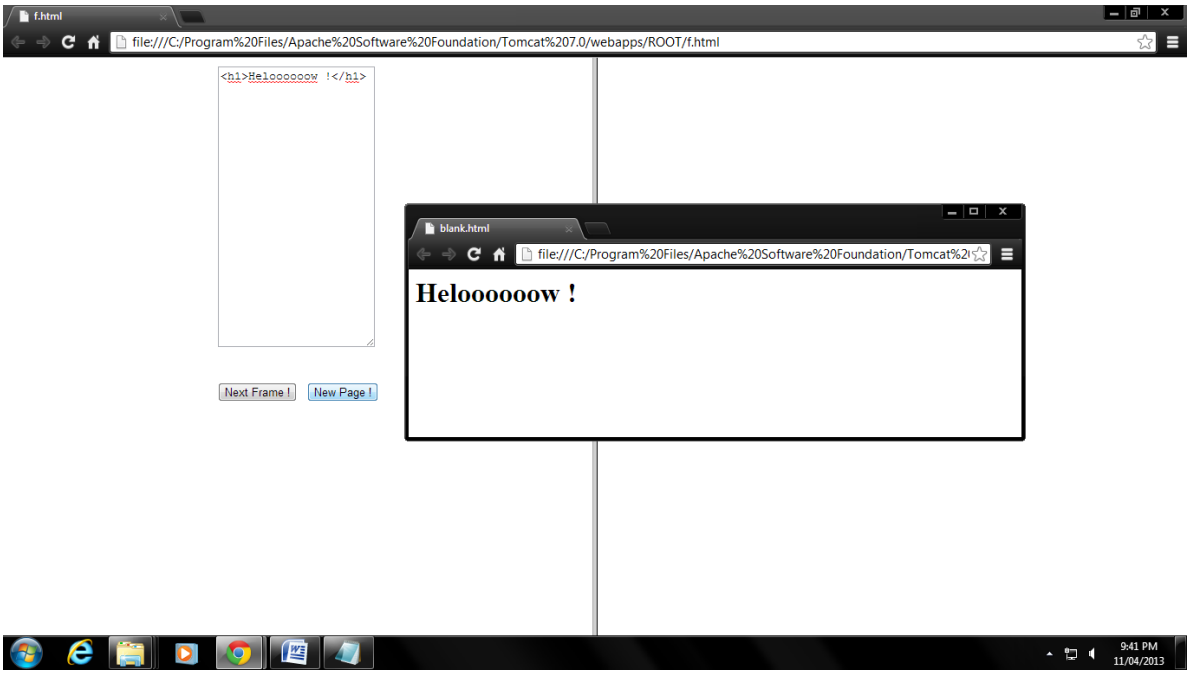
- **f.html :**

```
<html>
<head>
<frameset cols="50%,50%">
<frame src="1.html" name="f1"/>
<frame src="blank.html" name="f2"/>
</frameset>
</head>
```

```
</html>
```

*** OUTPUT :**


f.html
file:///C:/Program%20Files/Apache%20Software%20Foundation/Tomcat%207.0/webapps/ROOT/f.html
<h1>Heloooooow !</h1>
Next Frame !
New Page !
Heloooooow !
9:39 PM
11/04/2013



*** RESULT :**

To create a frameset , a textbox and two buttons and execute window object using this code .

EX NO :

* DROP DOWN MENU USING DHTML EVENT LISTENERS * * AND EVENT PROPAGATIONS *

* AIM :

To create a drop down menu using DOM for event listeners and event propagation by implementation tables.

* ALGORITHM :

- The drop down menu for event listeners and event propagation is implemented by using tables and by using visibility property , the menu can be hidden or shown .
- The style tag create table with font properties and to add hyper references underline , use text decoration :none.
- Create the style of table menus dropdown. To make the dropdown table merge with the main table , use position : absolute and by default keep visibility as hidden .
- Inside the script tag create two function to show the menu and hide the menu by visibility property .
- In the body create a table and call the show and hide functions for onmouseover and onmouseout.
- Add all the html files created , to the main file .
- Run the program , place the cursor over the

* PROGRAM CODING :

- **dropdown.html :**

```
<html>
<head>
<style>
body{font-family:Lucida Calligraphy}
```

```
table{font-size:90%;border-color:pink;border-style:outset;border-width:10}
a{color:black;text-decoration:none;font:bold}
a:hover{color:green}
td.menu{background:pink;border-color:pink;border-style:outset;border-width:10}
table.menu
{
font-size:100%;
position:absolute;
visibility:hidden;
border-color:pink;
border-style:outset;
border-width:10
}
h1{display:inline}
img{float:left}
img.t{float:right}
</style>
<script type="text/javascript">
function showmenu(elmnt)
{
document.getElementById(elmnt).style.visibility="visible";
}
function hidemenu(elmnt)
{
document.getElementById(elmnt).style.visibility="hidden";
}
</script>
</head>
<body>
<center>



<img src="t1.jpg" height=150 width=100></img>
<h1>~ * CSS Properties & Positioning * ~</h1>
<img class=t src="t2.jpg" height=150 width=100></img>
<p>* Place the Mouse over the options to see the drop down menus ! *</p>
<table width="100%">
<tr bgcolor="pink">
```

```
<td class="menu" onmouseover="showmenu('* Event Listener ! *')"
onmouseout="hidemenu('* Event Listener ! *')">
<a href="/default.asp"><h2>* Event Listener ! *</h2></a>
<table class="menu" id="* Event Listener ! *" width="120">
<tr><td class="menu"><a href="ml.html"><h4>Mouse !</h4></a></td></tr>
<tr><td class="menu"><a href="bl.html"><h4>Button !</h4></a></td></tr>
<tr><td class="menu"><a href="wl.html"><h4>Window !</h4></a></td></tr>
<tr><td class="menu"><a href="kp.html"><h4>KeyPress !</h4></a></td></tr>
<tr><td class="menu"><a href="tbl.html"><h4>Textbox !</h4></a></td></tr>
</table>
</td>
<td class="menu" onmouseover="showmenu('* Event Propagation ! *')"
onmouseout="hidemenu('* Event Propagation ! *')">
<a href="/default.asp"><h2>* Event Propagation ! *</h2></a>
<table class="menu" id="* Event Propagation ! *" width="120">
<tr><td class="menu"><a href="eb.html"><h4>Event Bubbling !</h4></a></td></tr>
<tr><td class="menu"><a href="ec.html"><h4>Event Capturing !</h4></a></td></tr>
</table>
</td>
</table>
</td>
</tr>
</table>
</body>
</html>
```

ml.html :

```
<html>
<head><script type="text/javascript">
function cod(event)



{
var x=event.screenX;
var y=event.screenY;
window.alert("X="+x+"Y="+y);
}
</script>
```

```
</head>
<body>
<script type="text/javascript">
window.document.addEventListener("mouseout",cod,false);
</script>
<center>
<h1> * Mouse Listener ! * </h1>
<h2>Click somewhere in the document to get that point's X and Y co ordinates !</h2>
</center></body>
</html>
```

bl.html :

```
<html>
<head>
<script type="text/javascript">
function cod()
{
var x=event.clientX;
var y=event.clientY;
window.alert("X="+x+" Y="+y);
}
</script>
</head>
<body>
<center>
<h1>* Button Listener ! *</h1>
<br>
<br>
<br>
<input id="msgbutton" type="button" value="Button Listener - OnClick"></input>



<script type="text/javascript">
var button=document.getElementById("msgbutton");
button.addEventListener("click",cod,false);
</script>
</center>
</body>
```

```
</html>
```

wl.html :

```
<html>
<head>
<script type="text/javascript">
function cod(event)
{
window.alert("Window resized");
}
</script>
</head>
<body>
<script type="text/javascript">
window.addEventListener("resize",cod,false);
</script>
<p>
window resize  listener
</p>
</body>
</html>
```

kp.html :

```
<html>
<head>
<script type="text/javascript">
function cod()
{
window.alert("keypress event");



}
</script>
</head>
<body>
<script type="text/javascript">
window.addEventListener("keypress",cod,false);
```

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

↳ tbl.html :

```
<html>
<head>
<script type="text/javascript">
function cod(event)
{
var t=event.currentTarget;
window.alert(t.value);
window.alert("Text Changed");
}
</script>
</head>
<body>
<center>
<h1>* Text box Listener ! *</h1>
<br>
<i>Enter Some text in the text box !</i>
<br>
<br>
<input id="t1" type="text" size="15"></input>
<script type="text/javascript">
var txt = document.getElementById("t1");
txt.addEventListener("change",cod,false);
</script>
</body>
</html>
```

↳ ec.html :

```
<!DOCTYPE HTML>
<html>
<body>
<center>
```

```
<h1>* Event Capturing ! *</h1>
<br>
<link type="text/css" rel="stylesheet" href="example.css">
<div class="d1">1
<div class="d2">2
<div class="d3">3
</div>
</div>
</div>
<script>
var divs = document.getElementsByTagName('div')
for(var i=0; i<divs.length; i++) {
divs[i].addEventListener("click", highlightThis, true)
divs[i].addEventListener("click", highlightThis, false)
}
function highlightThis() {
this.style.backgroundColor='yellow'
alert(this.className)
this.style.backgroundColor = ''
}
</script>
</center>
</body>
</html>
```

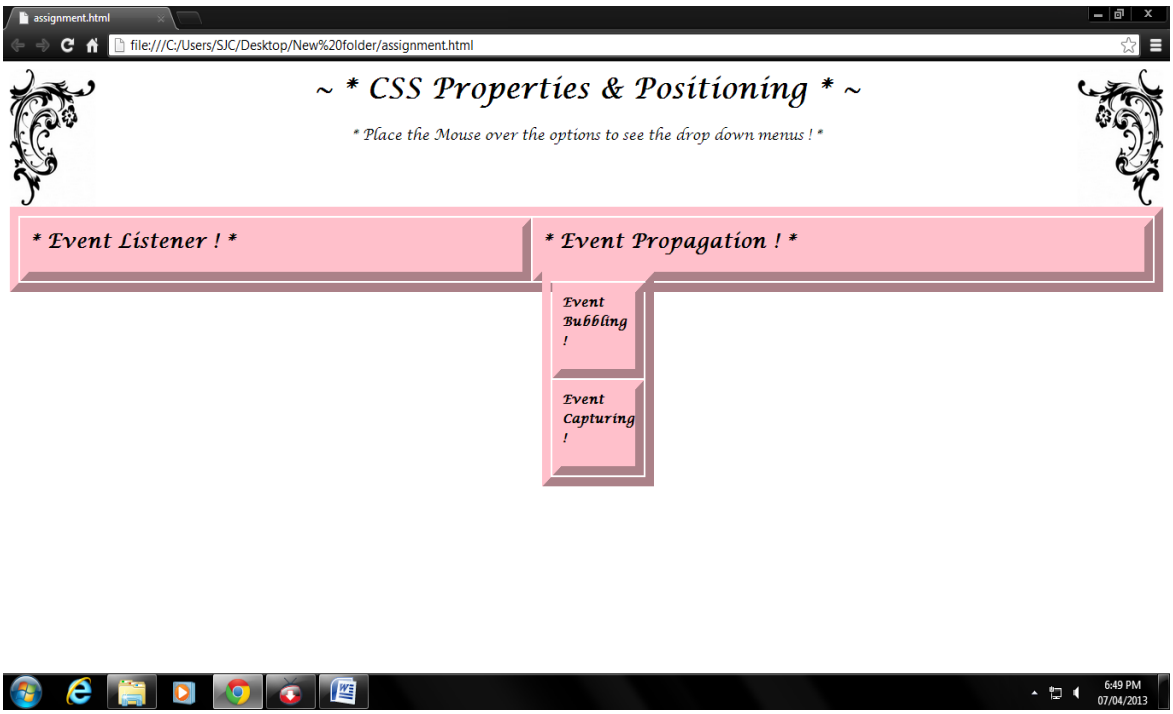
eb.html :

```
<HTML  onClick="f4()">
<HEAD>
<TITLE>Event Bubbles</TITLE>
<SCRIPT LANGUAGE="JavaScript">



function f1() {
alert("Event is now at button")
}
function f2() {
alert("Event is now at form")
}
```

```
function f3() {
alert("Event is now at  BODY ")
}
function f4() {
alert("Event is now at HTML")
}
</SCRIPT>
</HEAD>
<BODY onClick="f3()">
<CENTER>
<H1>* Event Bubbles ! *</H1>
<br>
<br>
<FORM onClick="f2()">
<INPUT TYPE="button" VALUE="Click Here"
onClick="f1()">
</FORM>
</CENTER>
</BODY>
</HTML>
```

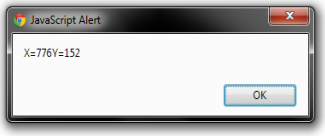
*** OUTPUT :**


assignment.html
file:///C:/Users/SJC/Desktop/New%20folder/assignment.html
~ * CSS Properties & Positioning * ~
* Place the Mouse over the options to see the drop down menus ! *
* Event Listener ! *
* Event Propagation ! *
Mouse !
Button !
Window !
KeyPress !
Textbox !
file:///C:/default.asp
6:49 PM
07/04/2013


assignment.html
file:///C:/Users/SJC/Desktop/New%20folder/assignment.html
~ * CSS Properties & Positioning * ~
* Place the Mouse over the options to see the drop down menus ! *
* Event Listener ! *
* Event Propagation ! *
Event Bubbling !
Event Capturing !
6:49 PM
07/04/2013

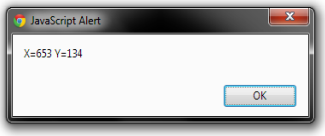
* Mouse Listener ! *

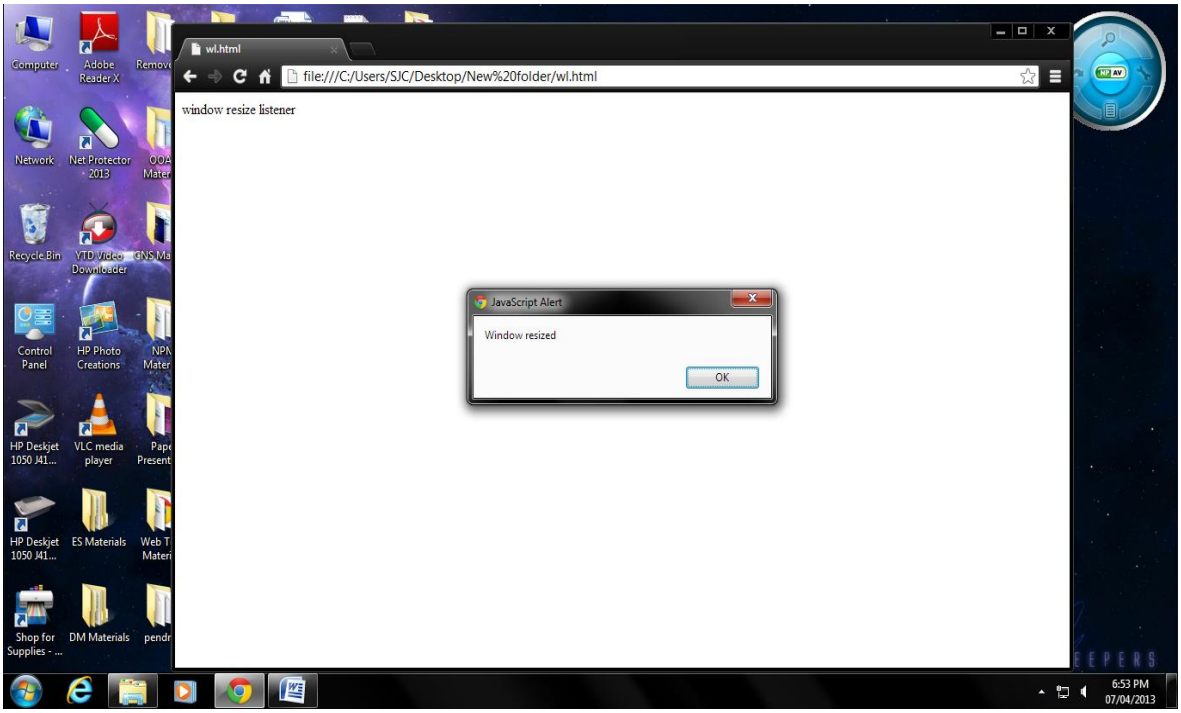
Click somewhere in the document to get that point's X and Y co ordinates !



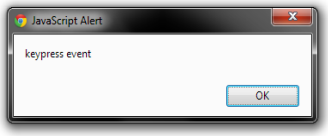
* Button Listener ! *

Button Listener - OnClick

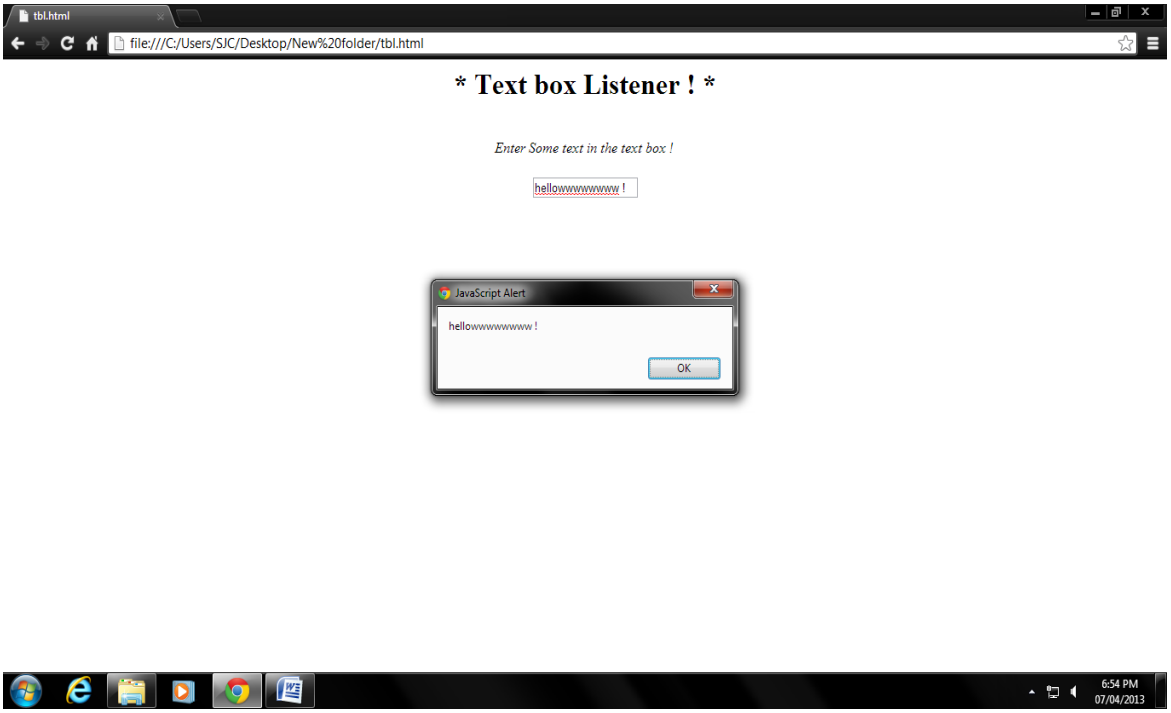


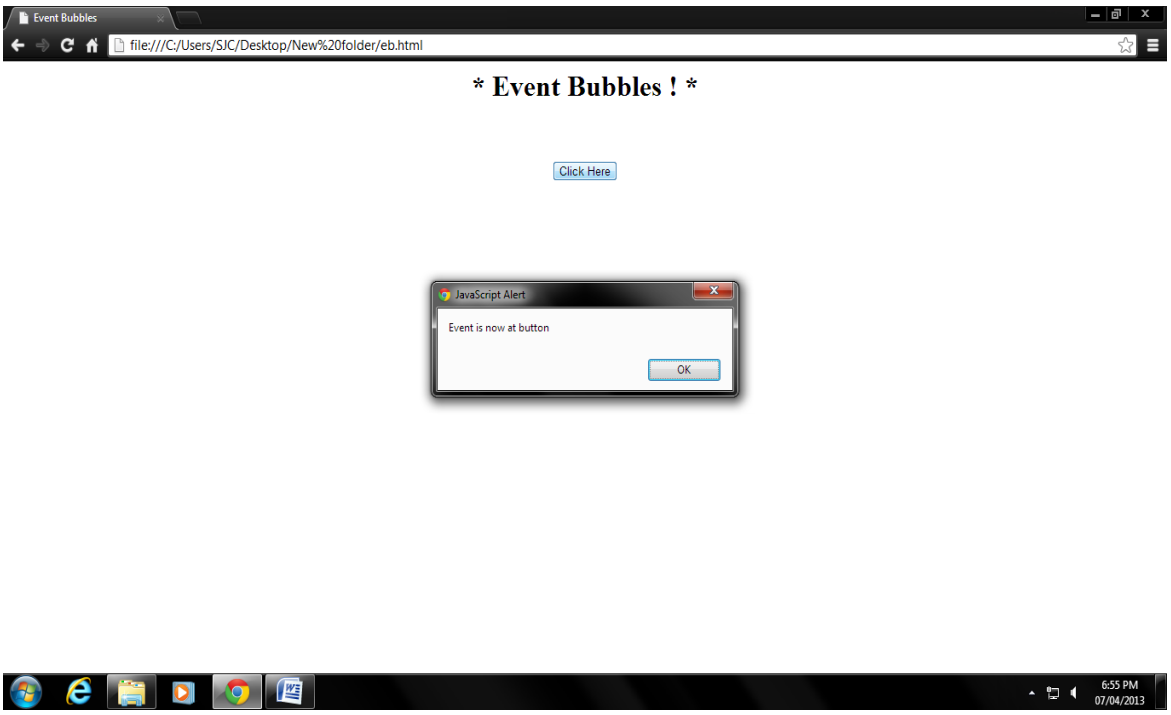

wl.html
file:///C:/Users/SJC/Desktop/New%20folder/wl.html
window resize listener
JavaScript Alert
Window resized
OK
6:53 PM
07/04/2013

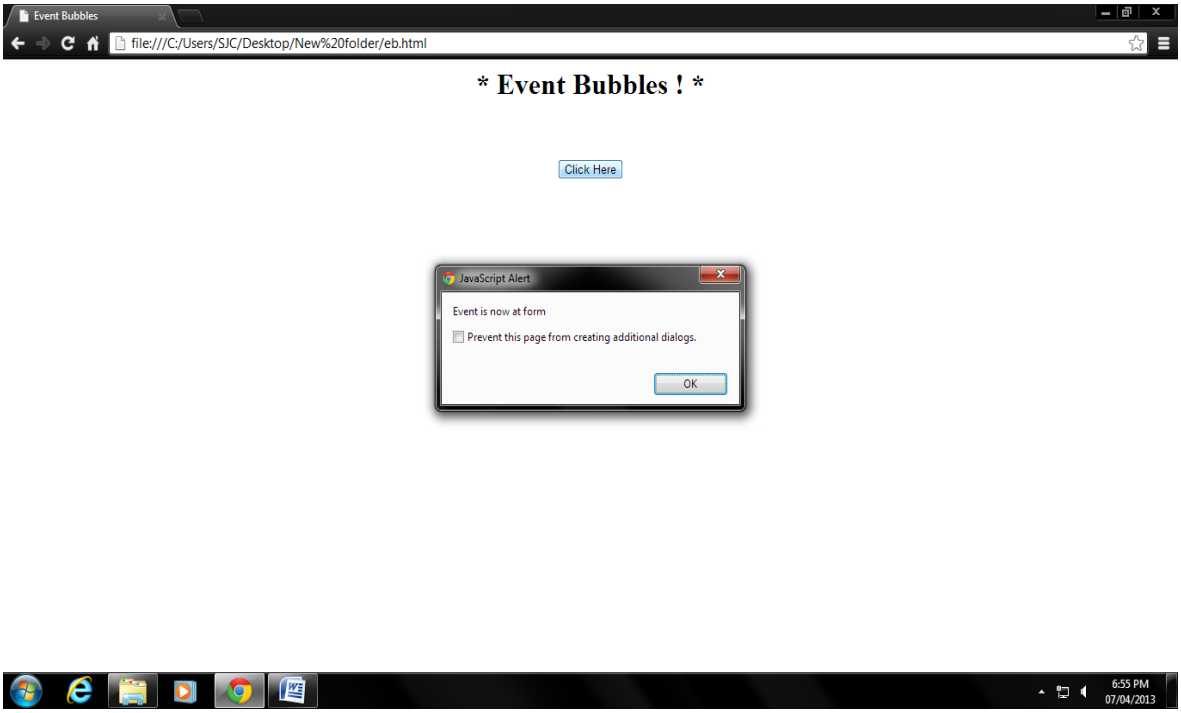

kp.html
file:///C:/Users/SJC/Desktop/New%20folder/kp.html

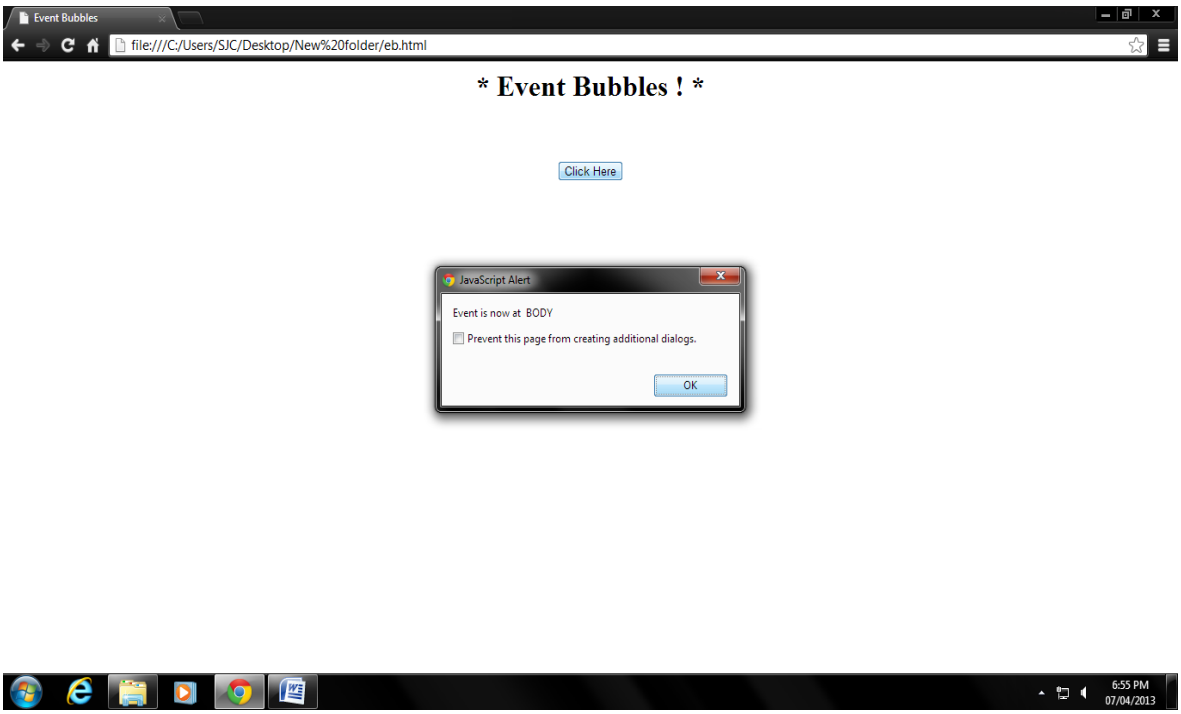

JavaScript Alert
keypress event
OK

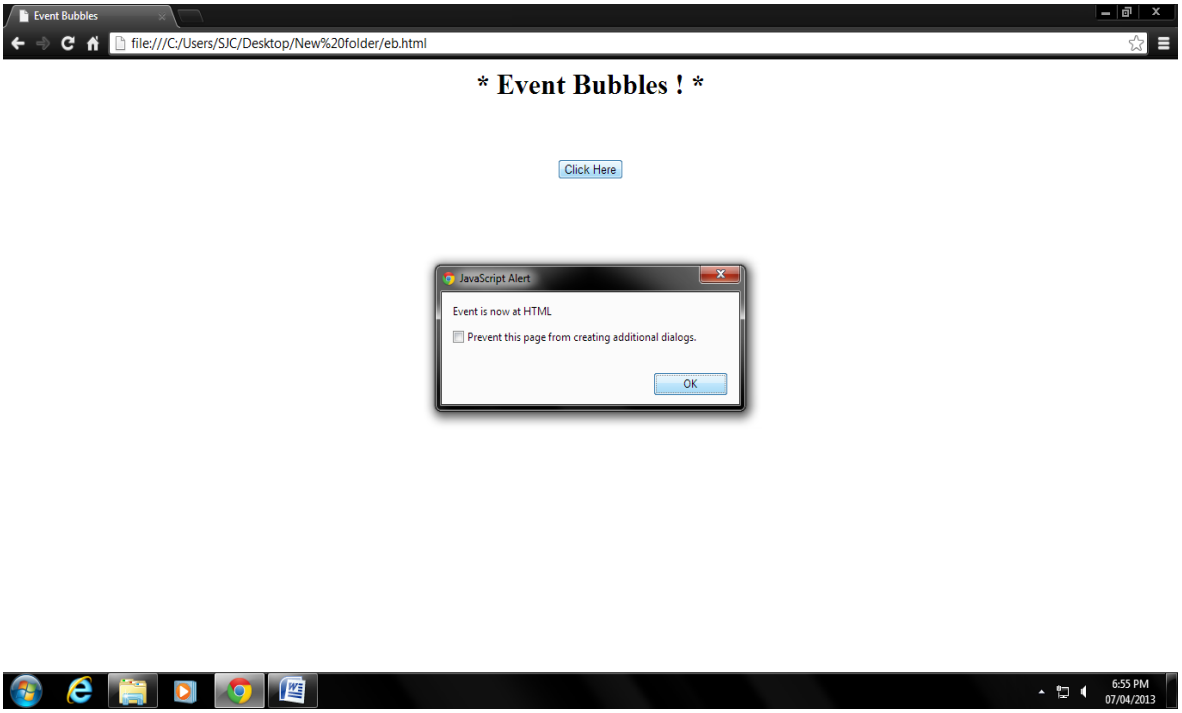

6:53 PM
07/04/2013

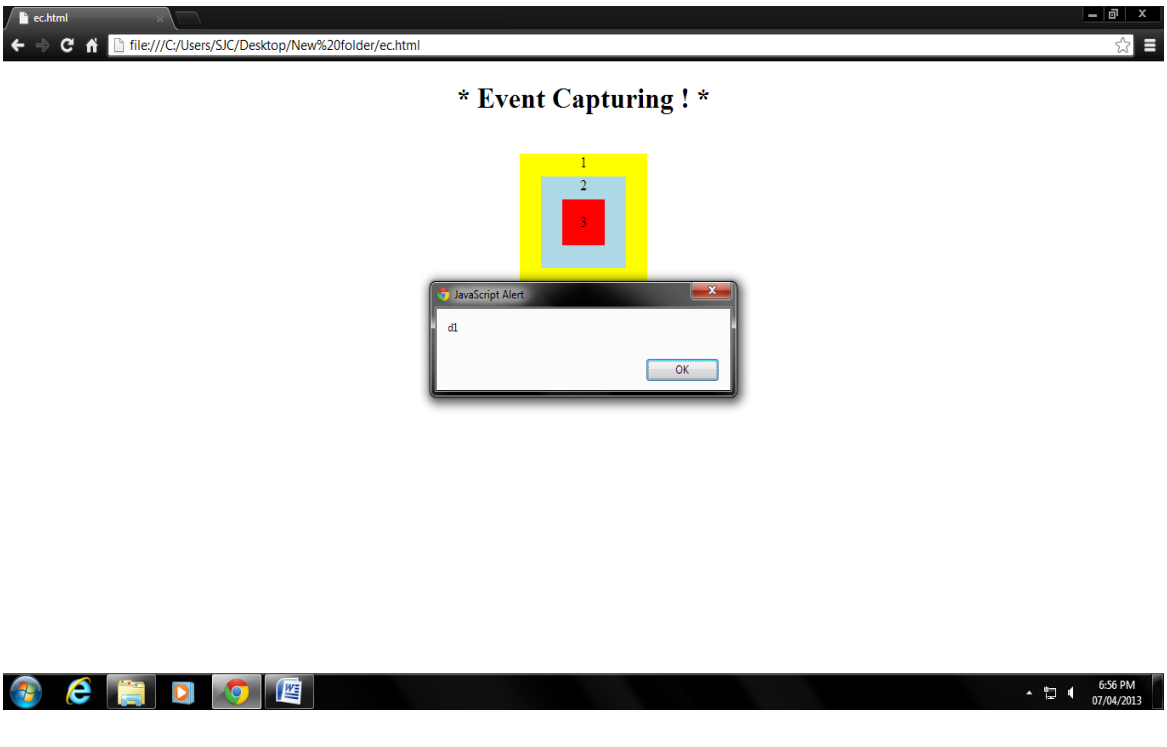

tbl.html
file:///C:/Users/SJC/Desktop/New%20folder/tbl.html
* Text box Listener ! *
Enter Some text in the text box !
hellowwwwwww !
JavaScript Alert
hellowwwwwww !
OK
6:54 PM
07/04/2013

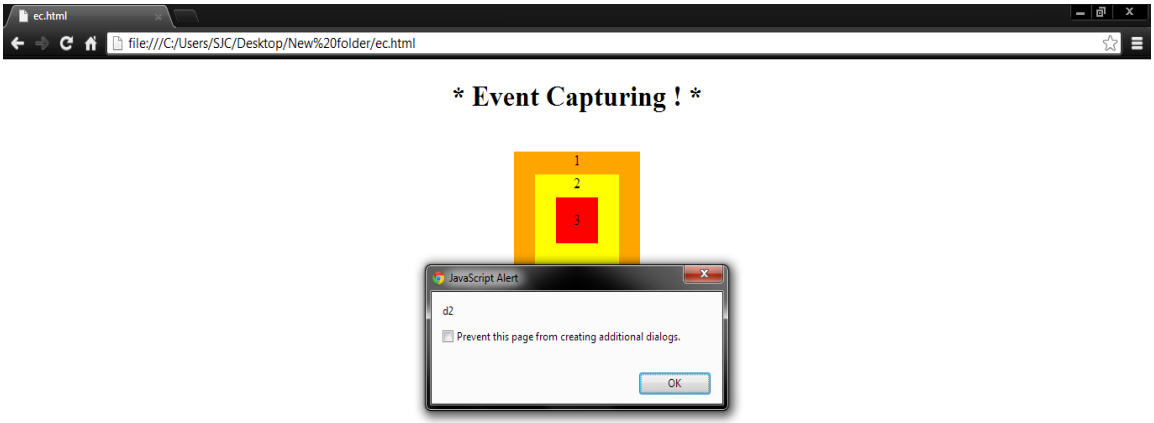

Event Bubbles
file:///C:/Users/SJC/Desktop/New%20folder/eb.html
* Event Bubbles ! *
Click Here
JavaScript Alert
Event is now at button
OK
6:55 PM
07/04/2013

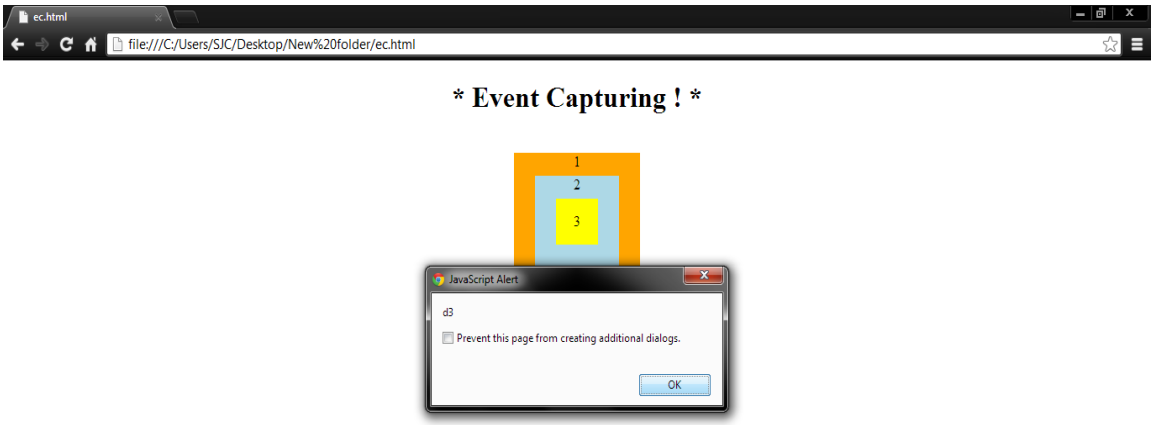

Event Bubbles
file:///C:/Users/SJC/Desktop/New%20folder/eb.html
* Event Bubbles ! *
Click Here
JavaScript Alert
Event is now at form
Prevent this page from creating additional dialogs.
OK
6:55 PM
07/04/2013


Event Bubbles
file:///C:/Users/SJC/Desktop/New%20folder/eb.html
* Event Bubbles ! *
Click Here
JavaScript Alert
Event is now at BODY
Prevent this page from creating additional dialogs.
OK
6:55 PM
07/04/2013


Event Bubbles
file:///C:/Users/SJC/Desktop/New%20folder/eb.html
* Event Bubbles ! *
Click Here
JavaScript Alert
Event is now at HTML
Prevent this page from creating additional dialogs.
OK
6:55 PM
07/04/2013


ec.html
file:///C:/Users/SJC/Desktop/New%20folder/ec.html
* Event Capturing ! *
1
2
3
JavaScript Alert
d1
OK
6:56 PM
07/04/2013


ec.html
file:///C:/Users/SJC/Desktop/New%20folder/ec.html
* Event Capturing ! *
1
2
3
JavaScript Alert
d2
Prevent this page from creating additional dialogs.
OK
6:56 PM
07/04/2013


ec.html
file:///C:/Users/SJC/Desktop/New%20folder/ec.html
* Event Capturing ! *
1
2
3
JavaScript Alert
d3
Prevent this page from creating additional dialogs.
OK
6:56 PM
07/04/2013

*** RESULT :**

Thus a program was written to create a drop down menu using DOM for event listeners and event propagation by implementation tables.

EX NO :

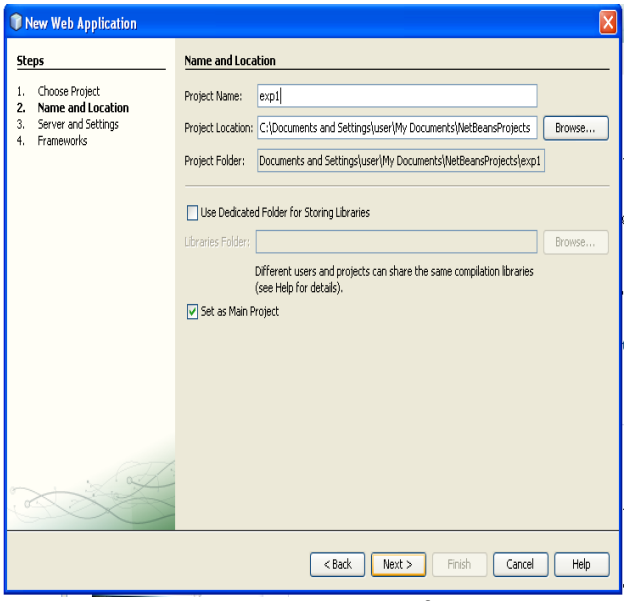
* ARITHMETIC OPERATION USING NET BEANS *

* AIM :

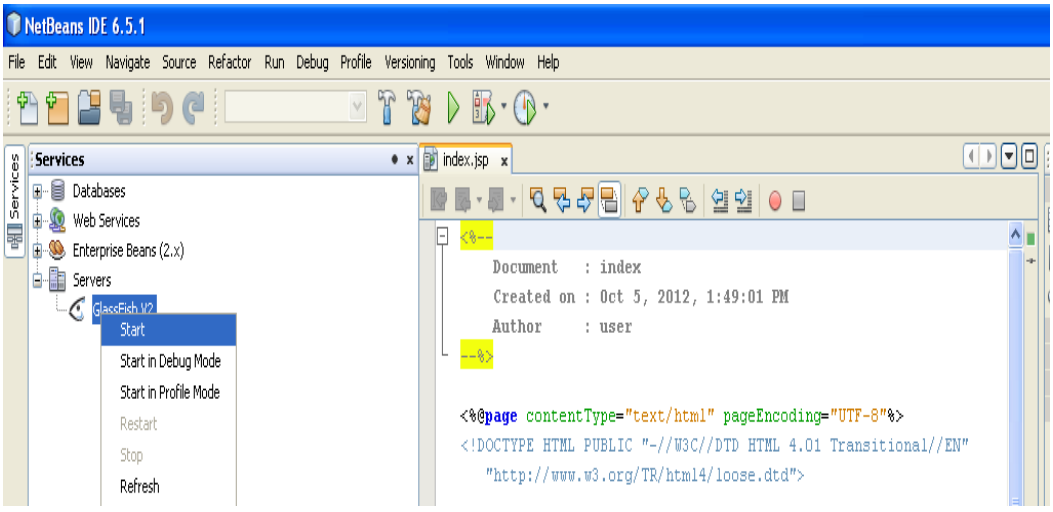
To create an addition application using Net Beans.

* PROCEDURE:

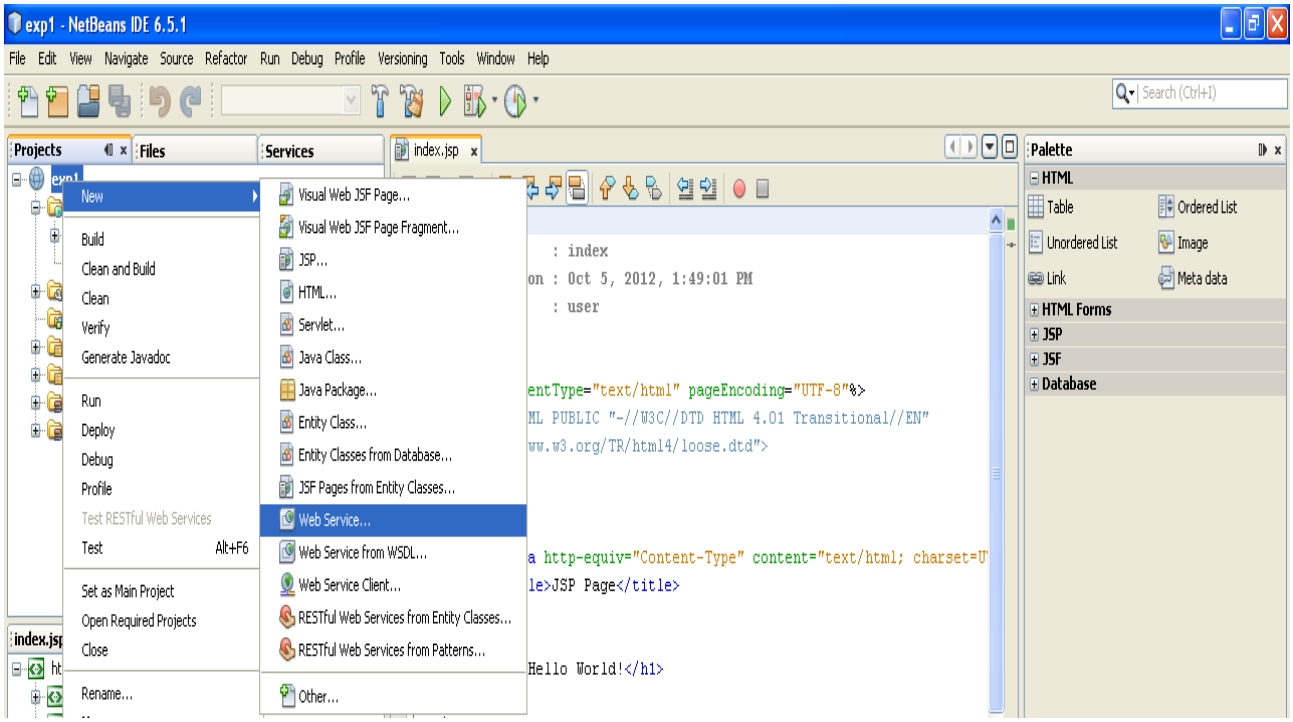
- Open Glassfish IDE 6.5.1-> File-> New Project-> Java Web(Categories)-> Web Application(Project)-> Next-> Give project name as exp1->next->finish.



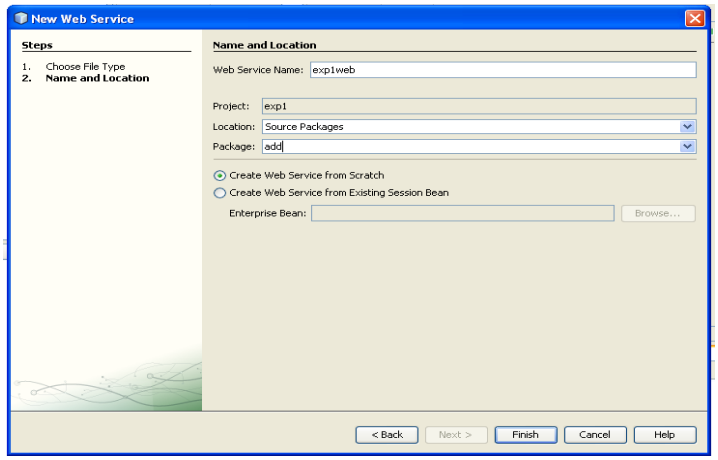
- Go to services, select server->Glassfish v2 and right click to start the server.



- Right click the project exp1->new-> right click->web service

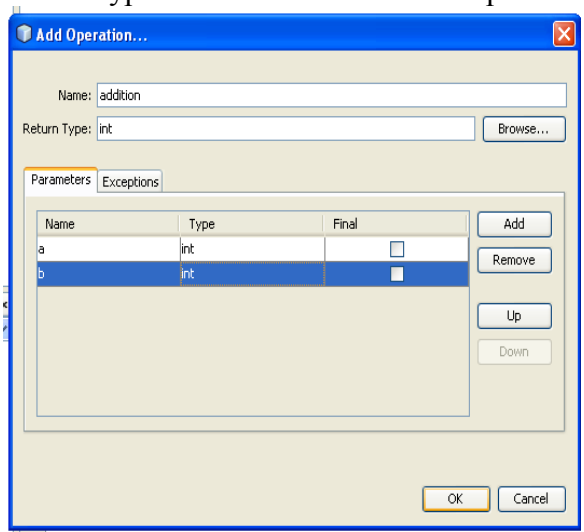


- Give web service name as exp1web and Give package name as add-> next.

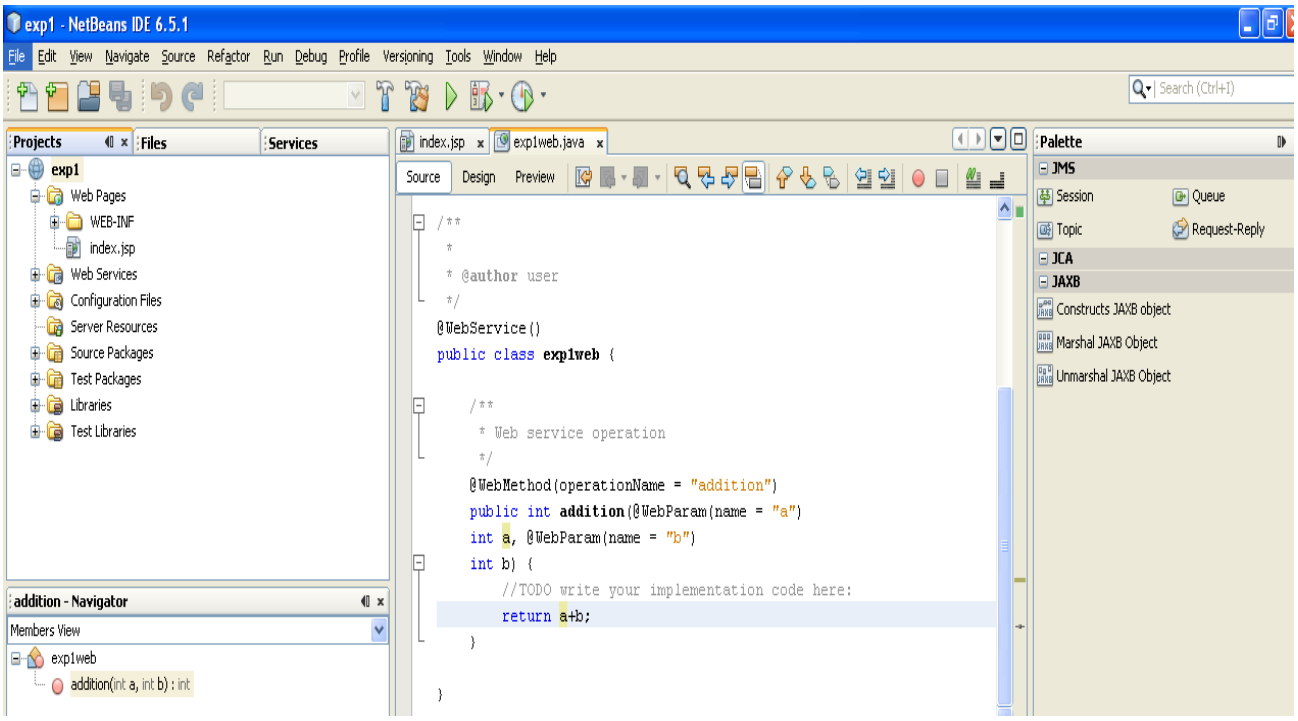


as

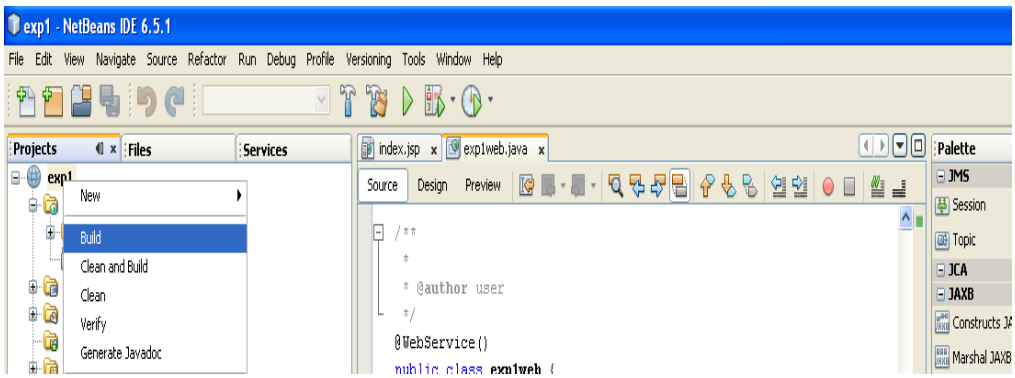
- Go to design tab->give method name addition->return type as int->Add->give ‘a’ as parameter type int ‘a’ and ‘b’ as another parameter type int->ok.



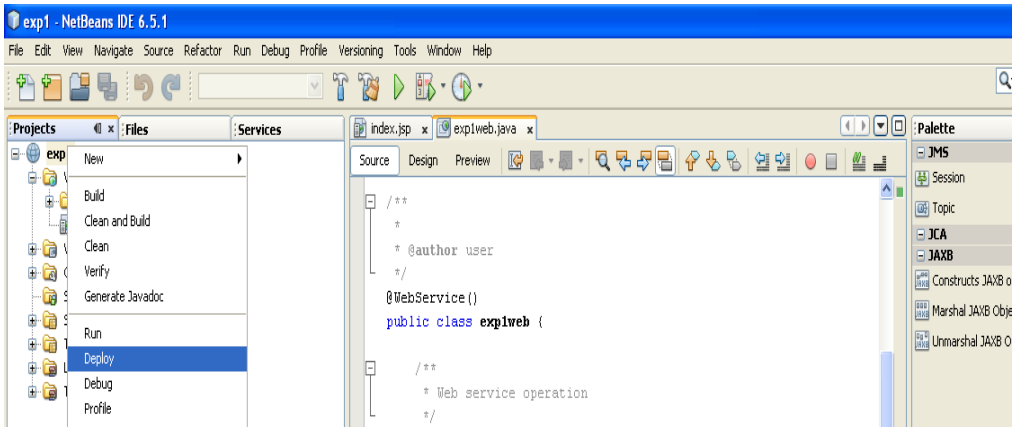
- Click Source tab->change the return value as a+b and save the project.



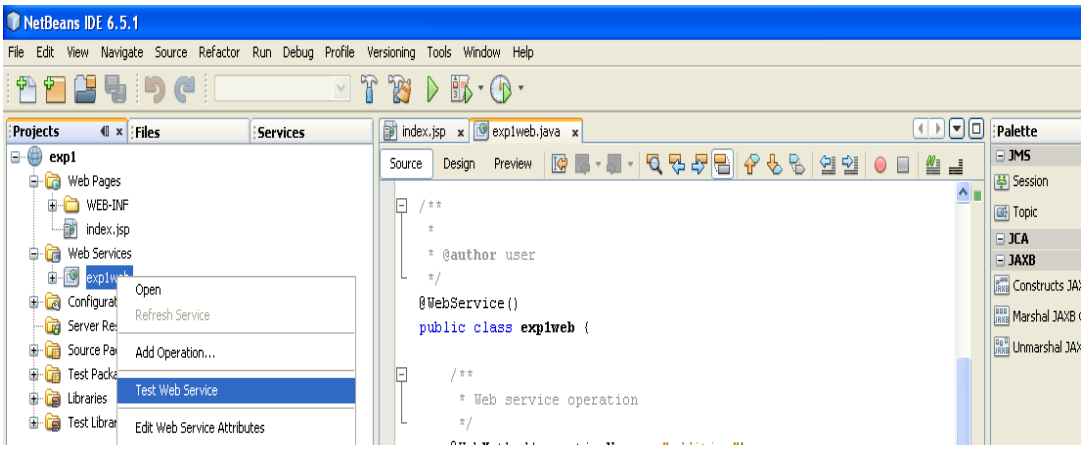
- Right click project exp1->build



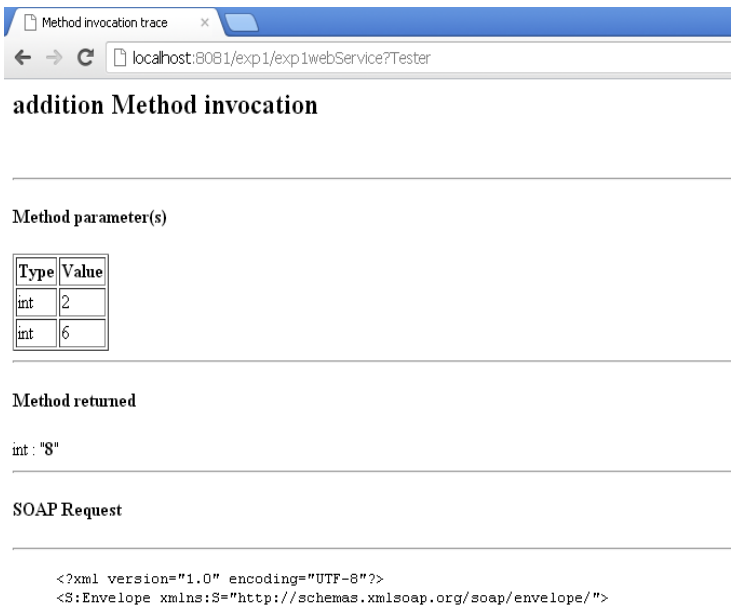
- Right click project exp1->deploy



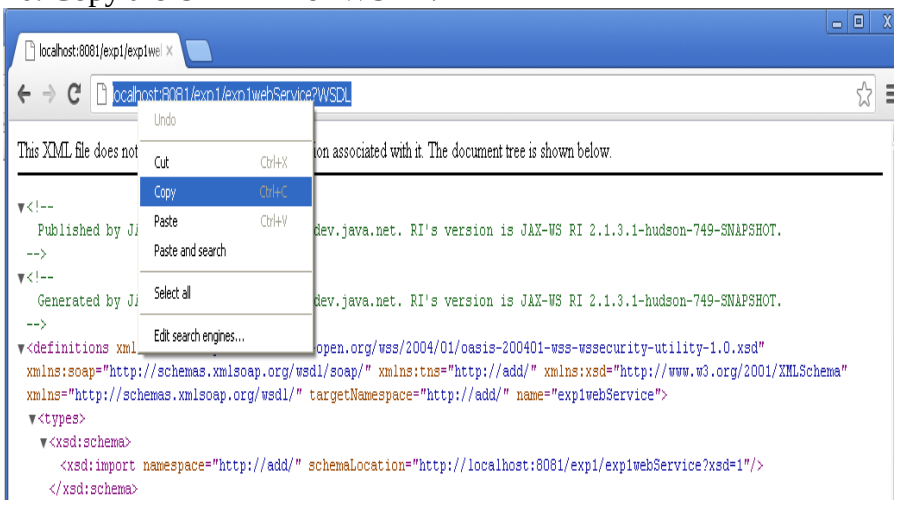
Click the '+' of webservice->right click exp1web->test webservice->output is displayed and enter the values in textbox.



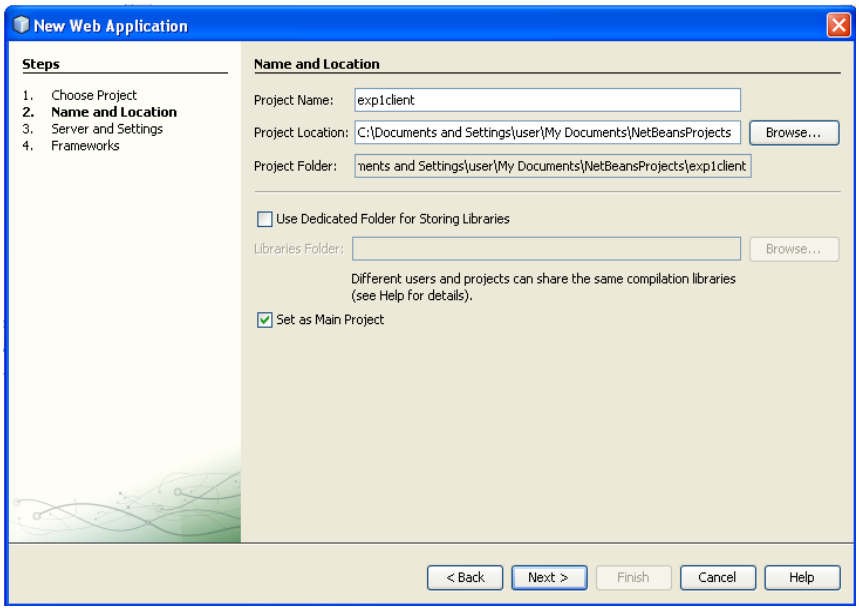
Output of Test



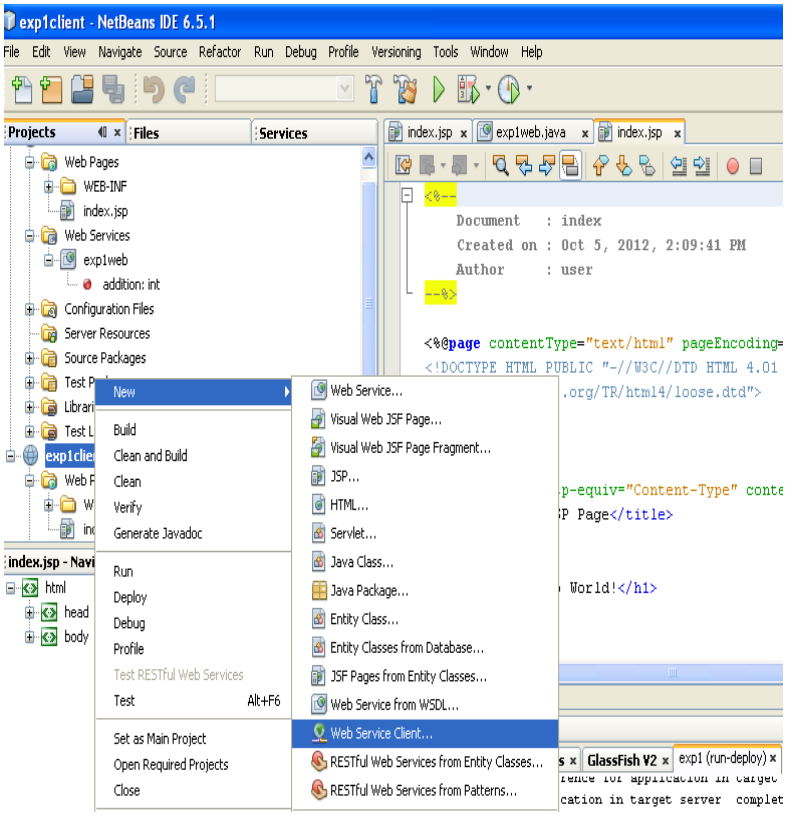
10. Copy the URL link of WSDL.



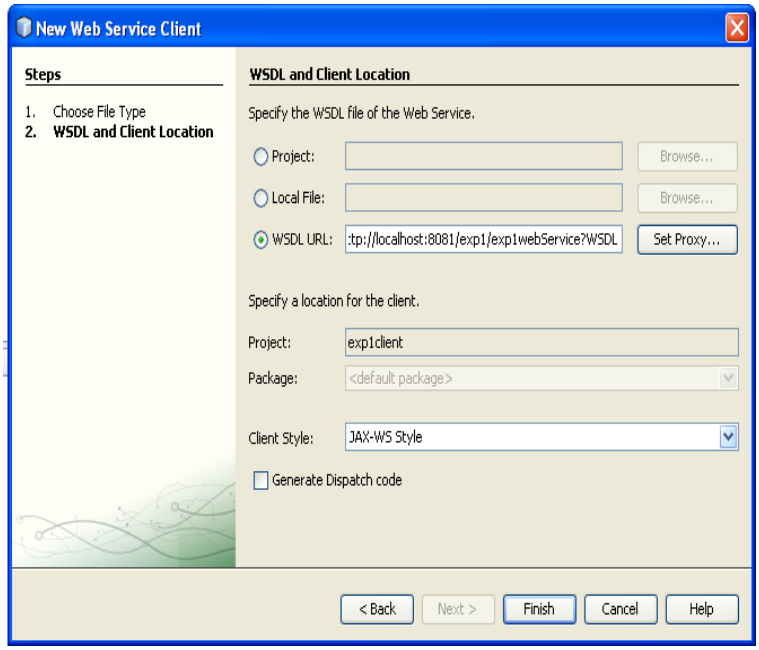
- In glassfish IDE 6.5.1->file->new project->java web (categories)->web application(project)->next->give project name as exp1client->next->finish



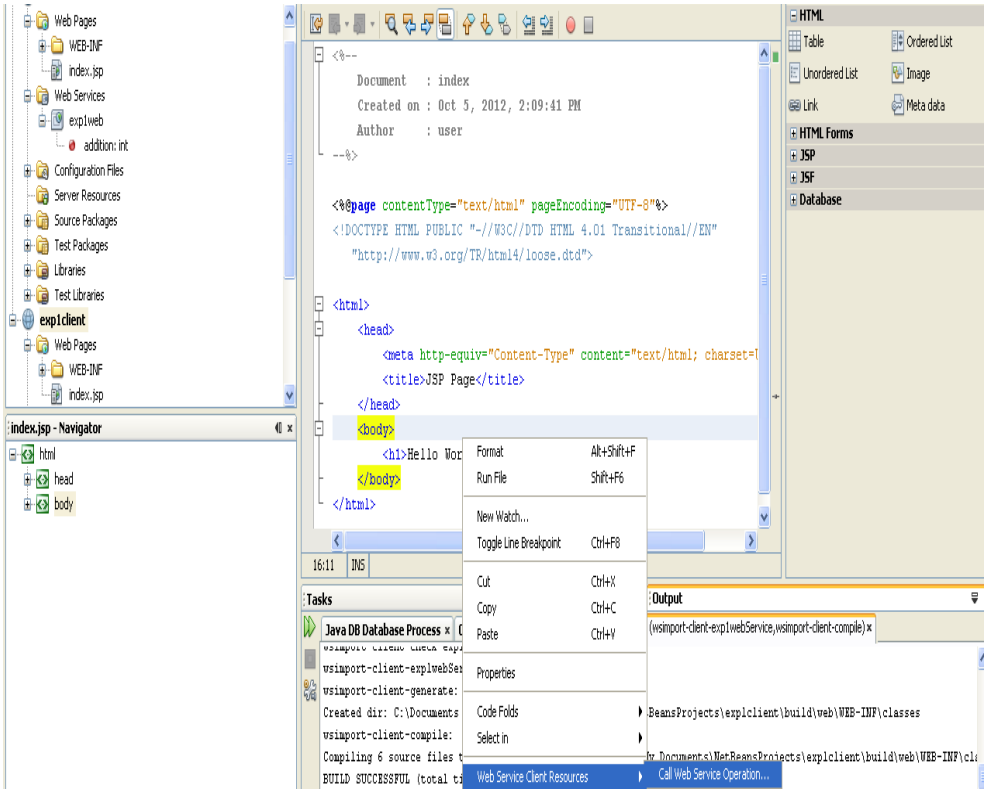
- Right click the project exp1client->new-> right click->web service client



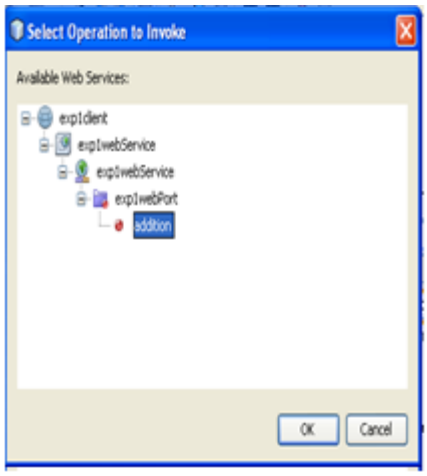
- Choose the WSDL URL and paste the URL link.



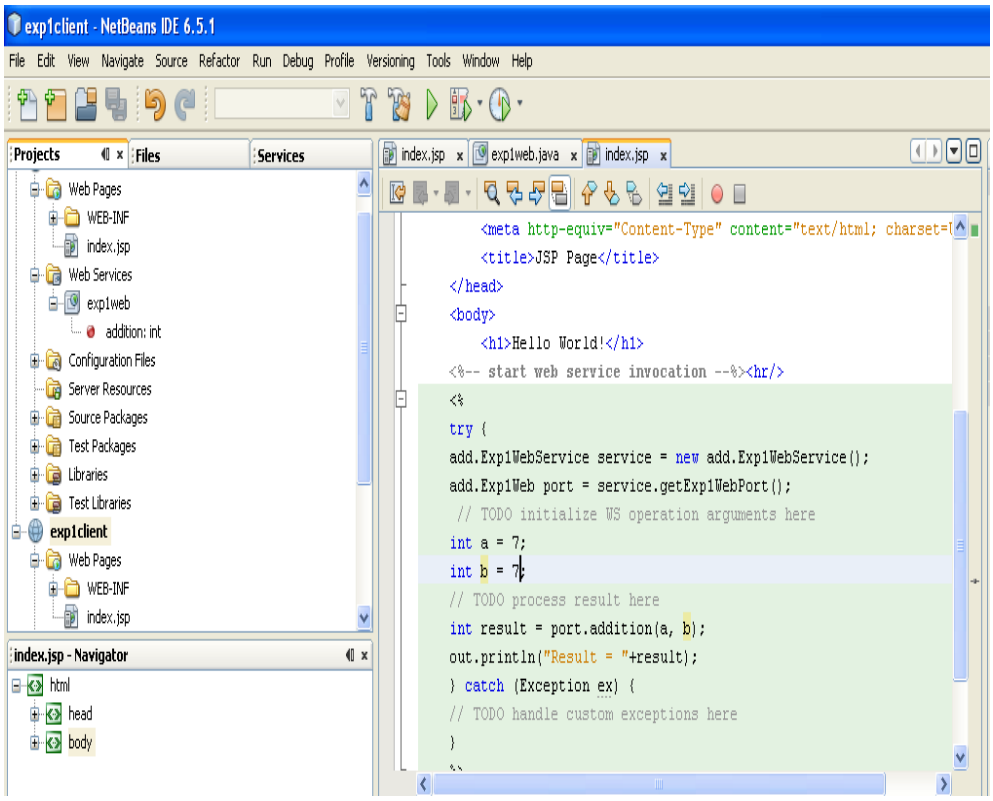
- In index.jsp, right click inside body tag and select web service client resource->call web service operations.



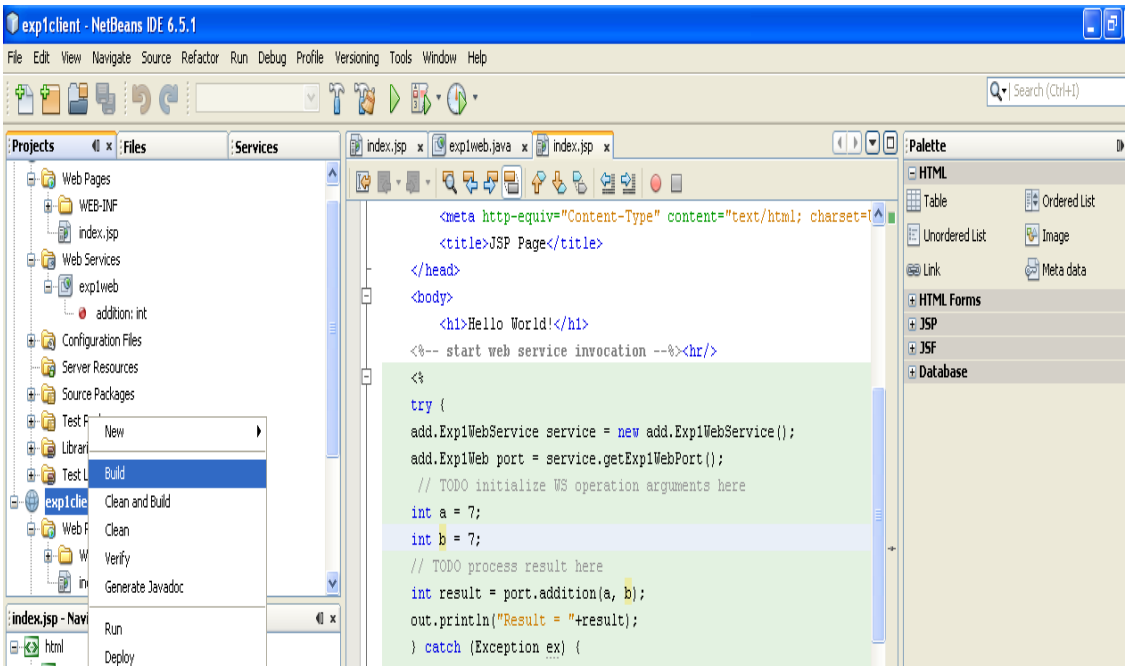
- Select the web service exp1web->addition(method)->ok.



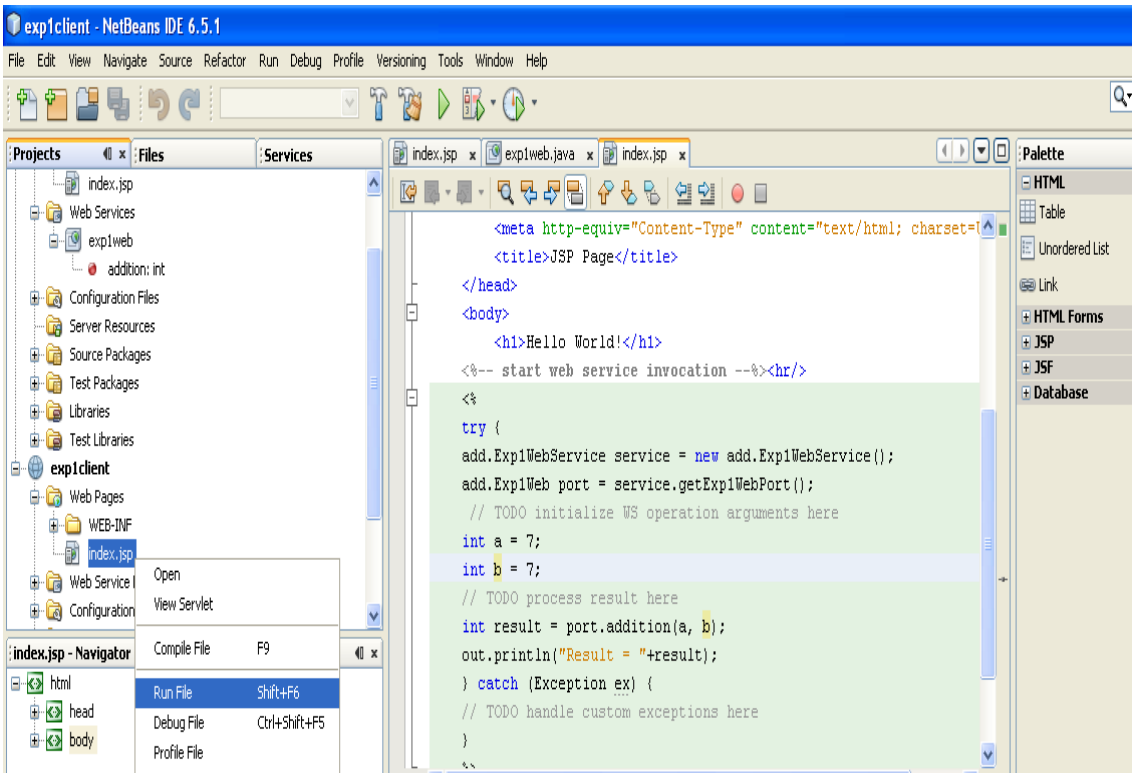
- Change the value of parameters in generated code and save the project.



- Right click the project exp1client->Build.



- Right click index.jsp ->Run File



* PROGRAM CODING :

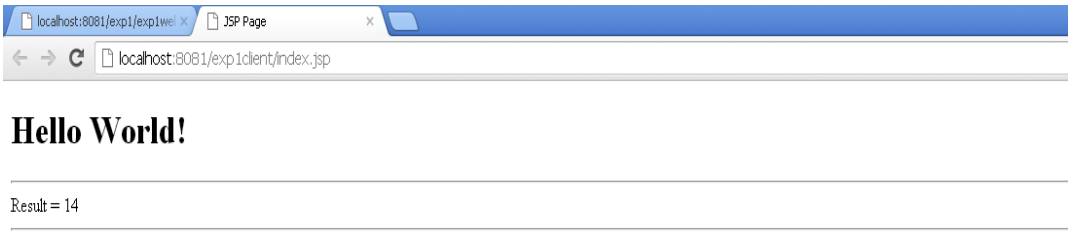
↳ EXP1WEB.JAVA

```
package add;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
@WebService()
public class exp1web
{
@WebMethod(operationName = "addition")
public int addition(@WebParam(name = "a")
int a, @WebParam(name = "b")
int b)
{return a+b;    }
}
```

↳ Index.jsp:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<h1>Hello World!</h1>
try {add.Exp1WebService service = new add.Exp1WebService();
add.Exp1Web port = service.getExp1WebPort();
int a = 7;
int b = 7;
int result = port.addition(a, b);
out.println("Result = "+result);
}
catch (Exception ex) {  }
</body></html>
```

*** OUTPUT :**



*** RESULT :**

Thus an addition application was created using net beans and the output was verified.

EX NO:

* SERVLET PROGRAM *
* ADDITION OF 2 NUMBERS *

*** AIM :**

To create a servlet program to send request from the client side to add two numbers.

* ALGORITHM :

- Start the tomcat server.
- Now , type http://localhost:8080 in the link of internet explorer and press enter key .
- The home page of Tomcat server will be displayed.
- Create a folder add in tomcat folder.
- Create an html document index.html with 2 text boxes to get input from the user and a button “Submit”.
- In the index.html, specify the folder and the java class name.
- Create a java code “add.java” for server side operations to perform the addition operation.
- Place the add.class in the classes folder .
- Create an xml file for mapping.

* PROGRAM CODING :

index.html :

```
<html>
<head>
<title>servlet</title>
</head>
<body>
<form name=f1 action="http://localhost:8080/addit/addit" method="get">
<center>
<br>
<br>
<h1>Addition</h1>
<h2>Num1 :<input type=text size=10 name=t1></h2>
<h2>Num2 :<input type=text size=10 name=t2></h2>
<input type=submit value=Add></input>
</center>
</form>
</body>
</html>
```

- **web.xml :**

```
<web-app>
<display-name>welcome to tomcat</display-name>
<description>
welcome to tomcat</discription>
<servlet>
<servlet-name>add</servlet-name>
<servlet-class>add</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>add</servlet-name>
<url-pattern>/add</url-pattern>
</servlet-mapping>
</web-app>
```

- **add.java :**

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class add extends HttpServlet
{
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws IOException, ServletException
{
response.setContentType("text/html");
PrintWriter out = response.getWriter();
int s1=Integer.parseInt(request.getParameter("t1"));
int s2=Integer.parseInt(request.getParameter("t2"));
out.println("<html>");
out.println("<head>");
out.println("<title>Example For Servlet</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1>Addition</h1>");
out.println("Sum = "+(s1+s2));
out.println("</body>");
out.println("</html>");}    }
```

*** OUTPUT :**

Addition

Num1 : 11

Num2 : 13

Add

Addition

Sum = 24

* RESULT :

Thus a servlet code was created to add 2 numbers and the output was executed successfully.

EX NO:

* SERVLET PROGRAM *
* SESSION TRACKING *

* AIM :

To create a servlet program to find the hit rate of a web page .

* ALGORITHM :

- Start the tomcat server.
- Now , type http://localhost:8080 in the link of internet explorer and press enter key .
- The home page of Tomcat server will be displayed.
- Create a folder st in tomcat folder.
- Create an html document index.html with a button “Submit”.
- In the index.html, specify the folder and the java class name.

- Create a java code “st.java” for server side operations to perform the display operation.
- Place the st.class in the classes folder .
- Create an xml file for mapping.

* PROGRAM CODING :

- **index.html :**

```
<html>
<head>
<title>Session Tracking</title>
</head>
<body>
<center>
<h1>Session Tracking</h1>
<form action="http://localhost:8080/session/s" method="get">
<br>
<input type="submit" value=submit></input>
</form>
</center>
</body>
</html>
```

- **web.xml :**

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

- **s.java :**

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

```
import javax.servlet.*;
import javax.servlet.http.*;
public class s extends HttpServlet
{
public void doGet(HttpServletRequest rq,HttpServletResponse rs)throws
IOException,ServletException
{
rs.setContentType("text/html");
PrintWriter out=rs.getWriter();
HttpSession sn=rq.getSession();
String head;
Integer cnt=(Integer)sn.getAttribute("cntobj");
if(cnt==null)
{
cnt=new Integer(0);
head="You are Accessing for the 1st time !";
}
else
{
head="Welcome once again !";
cnt=new Integer(cnt.intValue()+1);
}
sn.setAttribute("cntobj",cnt);
out.println("<html>");
out.println("<head>");
out.println("<title>Hellowww !</title>");



out.println("</head>");
out.println("<body bgcolor=yellow>");
out.println("<center>");
out.println("<h1>Your Access : </h1>"+cnt);
out.println("</center>");
out.println("</body>");
out.println("</html>");
}
}
```

*** OUTPUT :**

Session Tracking

submit

Ur access rate is
22

* RESULT :

Thus a servlet code was created to track the hit rate of a web page and the output was verified successfully .

EX.NO:

* INVOKE SERVLET USING COOKIES *

* AIM :

To create and implement a servlet program using cookies to get the value for the cookie.

* ALGORITHM:

- Initially create folders corresponding to the cookie in webapps.
- Then create java codes for getcookieservlet and save in .java in the required location.
- Now extract the jar file in the location where you have saved the java code.
- Configure the tomcat file and start the process.
- Set the url as "http://localhost:8080/cok/mycookieservlet".
- Finally save and run the program.

* PROGRAM CODING:

- **index.html :**

```
<html>
<head>
<title>Demo of Cookie</title>
</head>
<body bgcolor=lightblue>
<center>
<form name="form1" method="post" action="http://localhost:8080/cok/mycookieservlet">
<br>
<h1>Cookie !</h1>
<br>
<h2> Enter the value for my Cookie: </h2>
<input type=text name="txt_data" size =20 value=" ">
<br>
<br>
<input type=submit  value="Submit">
</form>
</body>
</html>
```

- **getCookieservlet.java :**

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class getCookieServlet extends HttpServlet
{
 public void doGet(HttpServletRequest req,HttpServletResponse res)
 throws ServletException,IOException
{
Cookie[ ] my_cookies=req.getCookies();
res.setContentType("text/html");
PrintWriter out=res.getWriter();
out.println("<b>");
int n=my_cookies.length;
for(int i=0;i<n;i++)
{
String name=my_cookies[i].getName();
String value=my_cookies[i].getValue();
out.println("name= "+name);
out.println("and value= "+value);
}
out.close();
```

```
}
}
```

- **mycookieservlet.java :**

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class mycookieservlet extends HttpServlet
{
public void doPost(HttpServletRequest req,HttpServletResponse res)
throws ServletException, IOException
{
String txt_data = req.getParameter("txt_data");
Cookie cookie = new Cookie("My_Cookie", txt_data);
res.addCookie(cookie);
res.setContentType("text/html");
PrintWriter out = res.getWriter();
out.println("<h2>MyCookie has been set to: ");
out.println(txt_data);



out.println("<br><br><br>");
out.println("This page shows that the cookie has been added");
out.close();
}
}
```

- **web.xml**

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

* OUTPUT :

Cookie

Enter the value for Cookie:

qwerty

Submit

MyCookie has been set to : qwerty

This page shows that the cookie has been added !

*** RESULT :**

Thus the program to invoke the servlet from cookies was executed successfully and the output was verified.

EX NO :

* JAVA OBJECTS AS FILES USING SERIALIZATION *

* AIM :

To Write a program for storing the Java Objects as files using serialization.

* ALGORITHM :

- Save some data in a file . Now write a code to retrieve datas from that file using a jsp code.
- Notice that for a class to be serialized successfully, two conditions must be met :
  - The class must implement the java.io.Serializable interface.
  - All of the fields in the class must be serializable. If a field is not serializable, it must be marked transient.
- Now run the java code as usual , javac sy.java / java sy .

* PROGRAM CODING :

sy.java :

```
import java.io.*;
import java.util.*;
class My implements Serializable
{
String name;
int roll;
public My(String n,int r)
{
name=n;
roll=r;
}
public void disp()



{
System.out.print("name:"+name+"  "+"roll no."+roll);
}
}
class sy
{
public static void main(String arg[]) throws ClassNotFoundException,IOException
```

```
{
My obj=new My("shiva",144);
obj.disp();
ObjectOutputStream o=new ObjectOutputStream(new FileOutputStream("2.txt")); //Serialize
o.writeObject(obj);
o.close();
ObjectInputStream i=new ObjectInputStream(new FileInputStream("2.txt"));
My x1=(My)(i.readObject());
x1.disp();
}
}
```

* OUTPUT :

```
C:\Program Files\Java\jdk1.5.0\bin>java sy
name:shiva  roll no.144name:shiva  roll no.144
```

* RESULT :

Thus a java code was written to save objects as a file and retrieve those objects and the output was executed successfully .

EX NO :

* JSP PROGRAM – RETRIEVE DATAS FROM * DATABASE *

* AIM :

To write a jsp program for retrieving datas from database.

* ALGORITHM :

- Write a jsp code to display the details from a database named StudentDB1.

- The database contains name , reg number and marks of a student .
- Place the jsp code in the ROOT directory of Tom Cat server. And place the database in the C: drive .
- In the Jsp code mention the connectivity as “jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ=c:\\StudentDB1.mdb".
- Create a table and display the details of the database in that table format.
- Start the tomcat server and type this command in the url “http://localhost:8080/disp.jsp”.

* PROGRAM CODING :

disp.jsp :

```
<%@ page import="java.sql.*" %>
<%@ page import="java.io.*" %>
<%
try{
Connection con = null;



Statement s=null;
ResultSet rs=null;
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
String myDB ="jdbc:odbc:Driver={Microsoft Access Driver
(*.mdb)};DBQ=c:\\StudentDB1.mdb";
con = DriverManager.getConnection(myDB,"","");
s= con.createStatement();
rs=s.executeQuery("select * from StudentTable");
%>
<h1>Students Marksheet</h1>
<h1>Name of the College:Easwari Engineering College</h1>
<table border="2" cellspacing="0" cellpadding="0">
<tr>
<td><b>Seat_No</b></td>
<td><b>Name</b></td>
<td><b>Marks</b></td>
</tr>
```

```
<%
while(rs.next())
{
%>
<tr>
<td><%=rs.getString("Seat_no")%></td>
<td> <%=rs.getString("Name")%></td>
<td> <%=rs.getInt("Marks")%></td>
</tr>
<%
}
%>
</table>
<%
}catch(Exception e){out.print(e);}
%>
```

- **StudentDB1 :**

StudentTable

Seat_No	Name	Marks
5013	Anusha.V.J	10
5023	Ghayathri.V	10

*** OUTPUT :**

Students Marksheet

Name of the College:Easwari Engineering College

Seat_No	Name	Marks
5013	Anusha.V.J	10
5023	Ghayathri.V	10

*** RESULT :**

Thus a jsp program was written for restoring datas from database.