<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>BIO DATA</title>

</head>

<body>

<h1 style="text-align: center;"> STUDENT BIO DATA</h1>

<marquee><img src="" alt="IMAGE OF STUDENT"></marquee>

<form>

<table>

<tr>

<td><b>REG.NO:</b></td>

<td><input type="text"></td>

</tr>

<tr>

<td><b>NAME :</b></td>

<td><input type="text"></td>

</tr>

<tr>

<td><b>EMAIL:</b></td>

<td><input type="email"></td>

</tr>

<tr>

<td><b>PHONE:</b></td>

<td><input type="number"></td>

</tr>

<tr>

<td><b>ADDRESS:</b></td>

<td><textarea id="address" type="textarea" required="true" placeholder="Address"></textarea></td>

</tr>

<tr>

<td><b>PIN CODE:</b></td>

<td><input type="number"></td>

</tr>

<tr>

<td><b>GENDER:</b></td>

<td>

<select name="GENDER">

<option value=" MALE">MALE</option>

<option value="FEMALE ">FEMALE</option>

<option value="OHTERS ">OTHERS</option>

</select>

</td>

</tr>

<tr>

<td><b>DOB:</b></td>

<td class="last" colspan=4><input id="db" type="datetime-local" required="true"></td>

</tr>

<tr>

<td><b>HOBBIES</b></td>

<td><label for="singing"><strong>singing</strong></label>

<input name="hobbies" id="singing" type="checkbox"></td>

<td><label for="playing"><strong>playing</strong></label>

<input name="hobbies" id="palying" type="checkbox"></td>

<td class="last"><label for="Dancing"><strong>Dancing</strong></label>

<input name="hobbies" id="Dancing" type="checkbox"></td>

</tr>

</table>

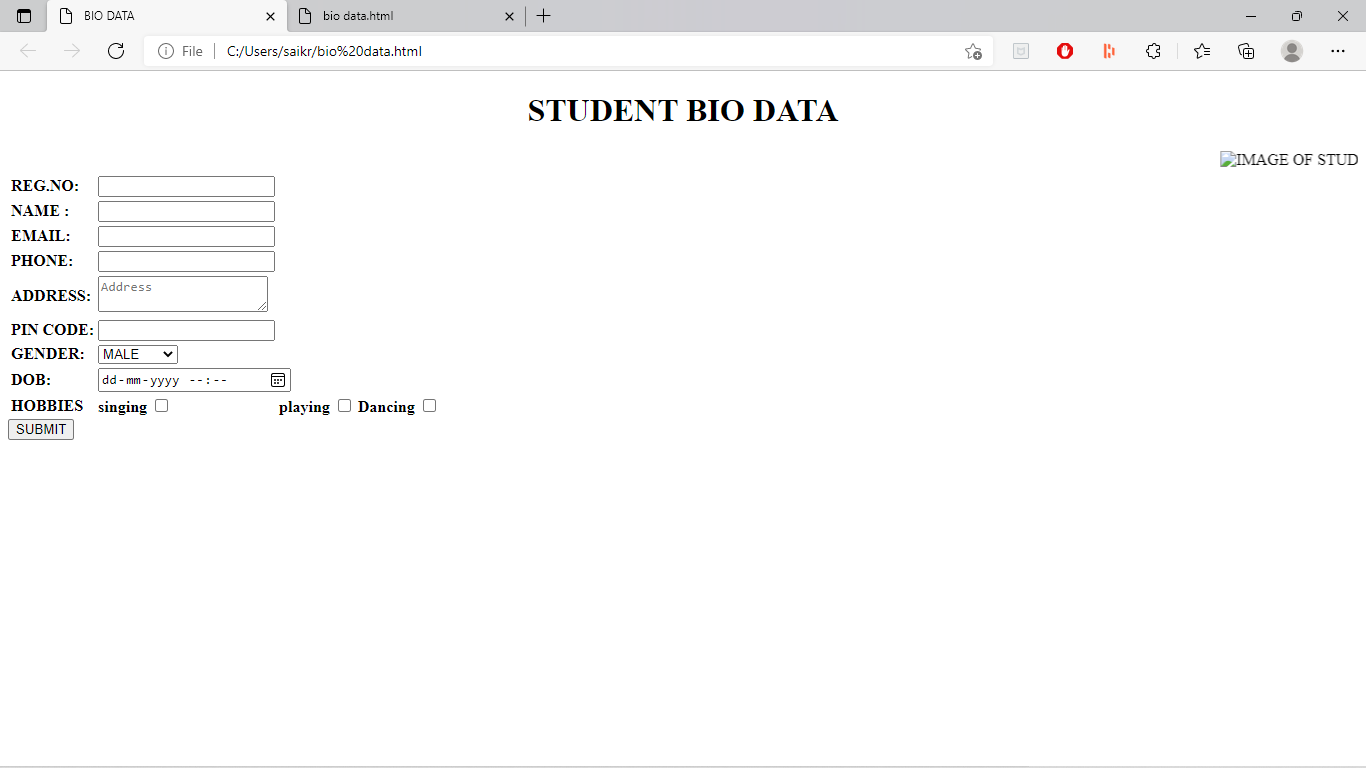
<input type="button" value="SUBMIT">

</form>

</body>

</html>

OUTPUT



2.Write the CSS rules for a simple static website.

* + 1. Rule for a background image is **left top** of the page, **tiling horizontally**. The image should **remain in place** when the user scrolls up or down.
    2. All paragraphs text **1.5 times** larger than the base font of the system and colors it **red** (inline, embedded and external style sheet).
    3. Rule for all H1 & H2 elements a **padding of 0.5em**, a grooved border style and a **margin of 0.5em**. (Box Model)

SOURCE CODE:

Source code:

<style>

body { background-image:url("Z:\Koala.jpg");

background-position:"top left ";

background-repeat:"repeat-x";

background-attachment:" fixed";

}

p{font-style:italic;

font-size:1.5em;

color:"red";

}

h1,h2{ padding:0.5em;

margin:0.5em;

border-style:groove;

}

</style>

<br />

<p>

Text is 1.5 times larger than the base font of the system, color isi red and font sytle is Italic

</p>

<h1>

Header 1 :: The text contains a padding of 0.5 em, margin 0.5 em and border style is groove

</h1>

<h2>

Header 2 :: The text contains a padding of 0.5 em, margin 0.5 em and border style is groove

</h2>

3.Write a script which reads a four digit integer entered by the user in a prompt dialog and encrypt it as replace each digit by the sum of that digit plus 7 modulus 10 and then swap the first digit with the third, and swap the second digit with the fourth. Then output the encrypted data . Write a separate script that inputs an encrypted integer and decrypts it to from the original number.

SOURCE CODE:

<

<html>

<head>

<title>ENCRPTION</title>

<script text="javascript">

num = window.prompt("enter the four digit number");

document.write("entered number is" + num + "<br>");

a = new Array(4);

for (i = 0; i <= 3; i++) {

n = num % 10;

a[i] = parseInt((n + 7) % 10)

num = num / 10

}

s = a[0]

a[0] = a[2]

a[2] = s

s = a[1]

a[1] = a[3]

a[3] = s

sum = 0

for (i = 3; i >= 0; i--) {

sum = sum \* 10 + a[i];

}

document.write("encrypted number is " + sum + "<br>")

for (i = 0; i < 4; i++)

a[i] = parseInt((a[i] + 3) % 10);

s = a[2]

a[2] = a[0]

a[0] = s

s = a[3]

a[3] = a[1]

a[1] = s

sum = 0

for (i = 3; i >= 0; i--)

sum = sum \* 10 + a[i];

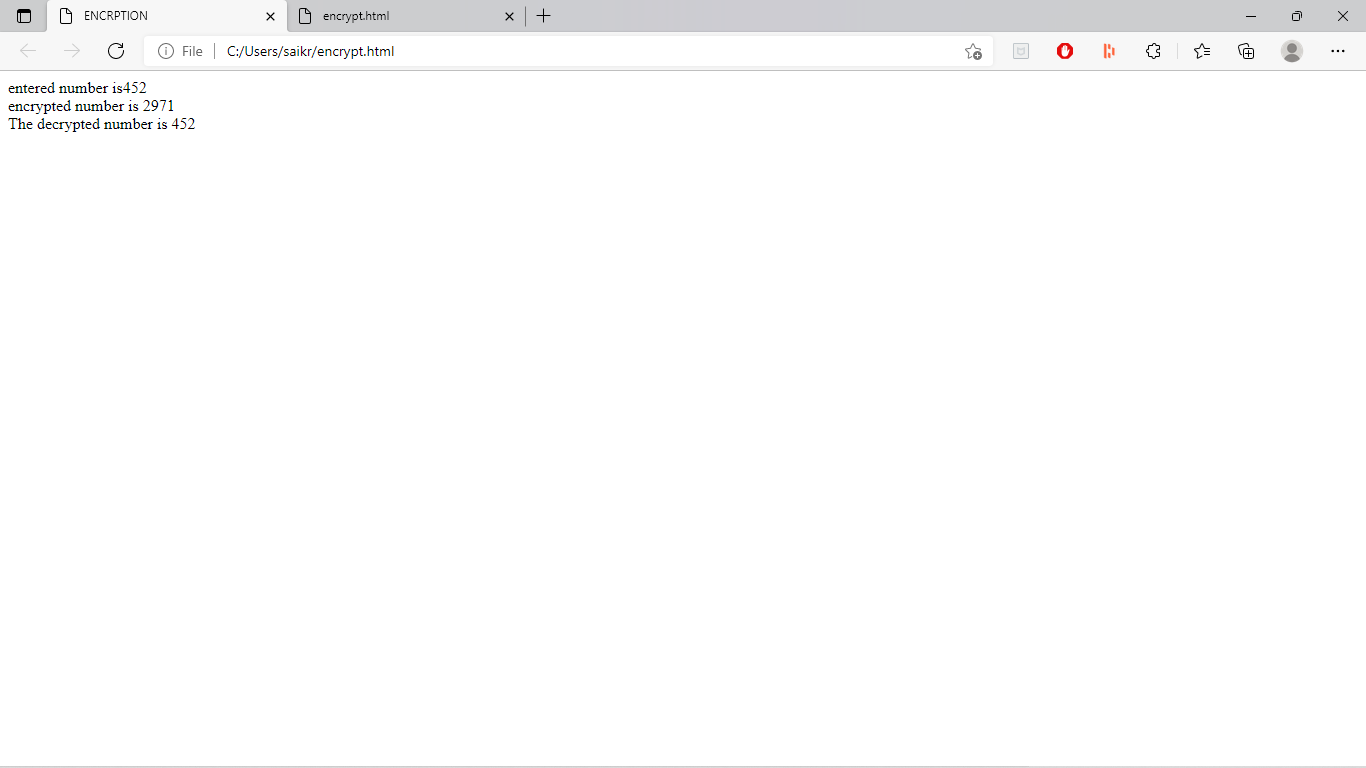
document.writeln("The decrypted number is " + sum);

</script>

</head>

</html>

OUTPUT:



4.Write the following scripts using JavaScript

1. Search a number from a given list of numbers using **Binary Search**technique.
2. Sort given set of numbers using **Bubble Sorting** technique.

SOURCE CODE:

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Binary search</title>

<script>

a = new Array(1)

n = parseInt(window.prompt("enter the total numbers"));

for (i = 0; i < n; i++) {

a[i] = parseInt(window.prompt("enter the value of " + (i + 1) + " number "));

}

document.write("entered array of numbers-->>")

for (i = 0; i < n; i++) {

document.write(a[i])

}

a.sort();

document.write("<br>sorted array of numbers-->>")

for (i = 0; i < n; i++) {

document.write(a[i])

}

x = parseInt(window.prompt("enter the searching element"));

l = 0;

h = n - 1;

while (l <= h) {

mid = parseInt((l + h) / 2);

if (a[mid] == x) {

document.write("<br>entered element is found at" + (mid + 1))

break;

} else if (a[mid] > x) {

h = mid - 1;

} else

l = mid + 1;

}

if (l > h) {

document.write("<br>not found")

}

</script>

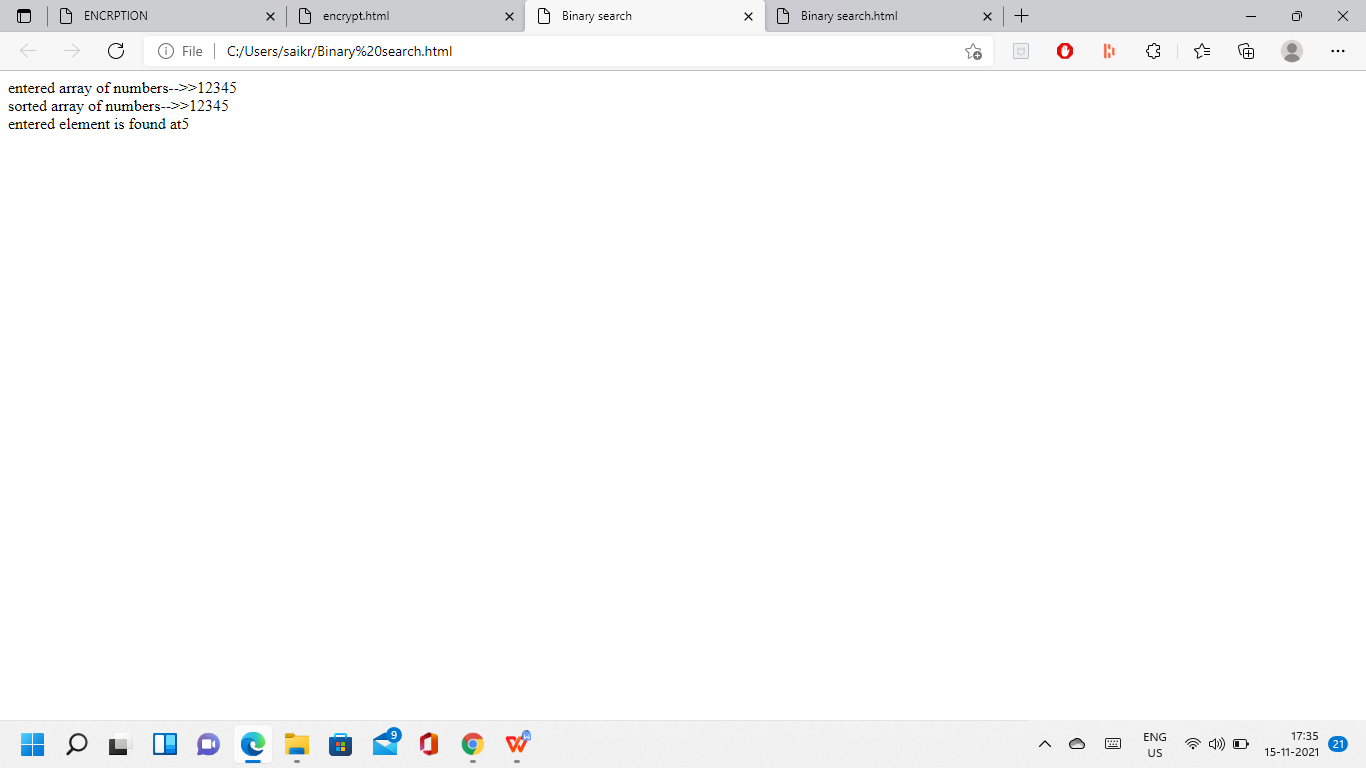
</head>

<body>

</body>

</html>

OUTPUT :



SOURCE CODE:

<script>

var a=new Array(10);

var n=parseInt(prompt("Enter no.of Elements:"));

for(var i=0;i<n;i++) //for(i in a) not working

a[i]=parseInt(window.prompt("Enter "+(i+1)+" Element:"));

document.write("<h1 align=center>Before Sorting elements are:");

for(var i in a)

document.write(a[i]+" ");

document.write("</h1>");

for(i in a)//for(var i=0;i<a.length;i++) --Sorting function

{

for(var j=0;j<a.length-i-1;j++)//for(j in a-i-1) not working

{

if(a[j]>a[j+1])

{

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

}

document.write("<h1 align=center>After Sorting elements are:");

for(var i in a)

document.write(a[i]+" ");

document.write("</h1>");

</script>

OUTPUT:

# Before Sorting elements are:5 1 2 8 4

# After Sorting elements are:1 2 4 5 8

5.Write scripts for the following using recursive functions

1. Write a function to power which takes two arguments **m** and **n**, and returns **mn** (consider m and n are integers**).**
2. Factorial & finding nth Fibonacci

SOURCE CODE:

<script>

var m,n,p=1;

m=window.prompt("Enter the value of m",0);

n=window.prompt("Enter the value of n",0);

power(m,n);

function power(m,n)

{

for(i=1;i<=n;i++)

p=p\*m;

}

document.write("<h1 align='center'>")

document.write(m+"<sup>"+n+"</sup> = "+p);

document.write("</h1>");

</script>

OUTPUT:

# power of a number

# 55 = 3125

SOURCE CODE:

<script>

var n;

n=parseInt(window.prompt("Enter a number"));

document.writeln("<h4>The Factorial numbers are<br></h4>");

document.write("<table style='border:1px solid red;'>");

for(var i=0;i<=n;i++)

document.write("<tr><td>"+i+"!</td><td>"+factorial(i)+"</td></tr>");

document.write("</table>");

document.writeln("<br><br><h4>The "+n+" fibonacci numbers are: </h4>");

for(i=0;i<n;i++)

{

f=fib(i);

document.writeln(f+" ");

}

function fib(n)

{

if(n==1||n==0)

return 1;

else

return fib(n-1)+fib(n-2);

}

function factorial(n)

{

if(n<=1)

return 1;

else

return n\*factorial(n-1);

}

</script>

OUTPUT:

#### The Factorial numbers are

0! 1

1! 1

2! 2

3! 6

4! 24

5! 120

#### The 5 fibonacci numbers are:

0 1 1 2 3

1. Write a script which reads two matrices from the prompt dialog, and do the matrix multiplication

SOURCE CODE:

<script>

var m1,n1,m2,n2;

m1=parseInt(window.prompt("enter no of rows of first matrix","0"));

n1=parseInt(window.prompt("enter no of colomns of first matrix","0"));

m2=parseInt(window.prompt("enter no of rows of second matrix","0"));

n2=parseInt(window.prompt("enter no of colomns of second matrix","0"));

if(n1!=m2)

window.alert("2 matrices r not compatable 4 multiplication");

else

{

a=new Array(m1);

for(var i=0;i<a.length;i++)

a[i]=new Array(n1);

b=new Array(m2);

for(var i=0;i<b.length;i++)

b[i]=new Array(n2);

c=new Array(m1);

for(var i=0;i<b.length;i++)

c[i]=new Array(n2);

read(a);

read(b);

for(i=0;i<m1;i++)

for(j=0;j<n2;j++)

{

c[i][j]=0;

for(k=0;k<n1;k++)

c[i][j]+=a[i][k]\*b[k][j];

}

display("first matrix",a);

display("second matrix",b);

display("result matrix",c);

}

function read(x)

{

for(i=0;i<x.length;i++)

for(j=0;j<x[i].length;j++)

x[i][j]=parseInt(window.prompt("enter["+i+1+","+j+1+"]element","0"));

}

function display(str,y)

{

document.write("<h2>"+str+"</h2>");

for(i=0;i<y.length;i++)

{

for(j=0;j<y[i].length;j++)

document.write(y[i][j]+" ");

document.writeln("<br/>");

}

}

</script>

OUTPUT:

## first matrix

1 2  
2 2

## second matrix

3 1  
2 2

## result matrix

7 5  
10 6

7.Write Scripts for the following events

1. Display digital clock using *onload* event.
2. To move text along with mouse pointer. The text will be inputted by a text field.

SOURCE CODE:

<head>

<script>

function display()

{

var d=new Date()

var h=d.getHours()

var m=d.getMinutes()

var s=d.getSeconds()

var t= h+ ":"+ m+ ":" + s

//h1txt.innerText=t

document.getElementById("h1txt").innerText=t

}

function start()

{

setInterval("display()",1000)

// setInterval() method calls a function or evaluates an expression at specified intervals

// 1000 ms = 1 second.

}

</script>

</head>

<body onload="start()">

<h1 id="h1txt" style="text-align:center"> 00:00:00 </h1>

</body>

OUTPUT:

**17:56:15**

SOURCE CODE:

<html>

<head>

<script type="text/javascript">

function mpos()

{

X=event.offsetX;

Y=event.offsetY;

document.getElementById("h1text").style.left=X;

document.getElementsByTagName('h1')[0].style.top=Y;

document.getElementsByTagName('h1')[0].innerText=document.getElementById('t1').value

}

</script>

</head>

<body onmousemove="mpos()">

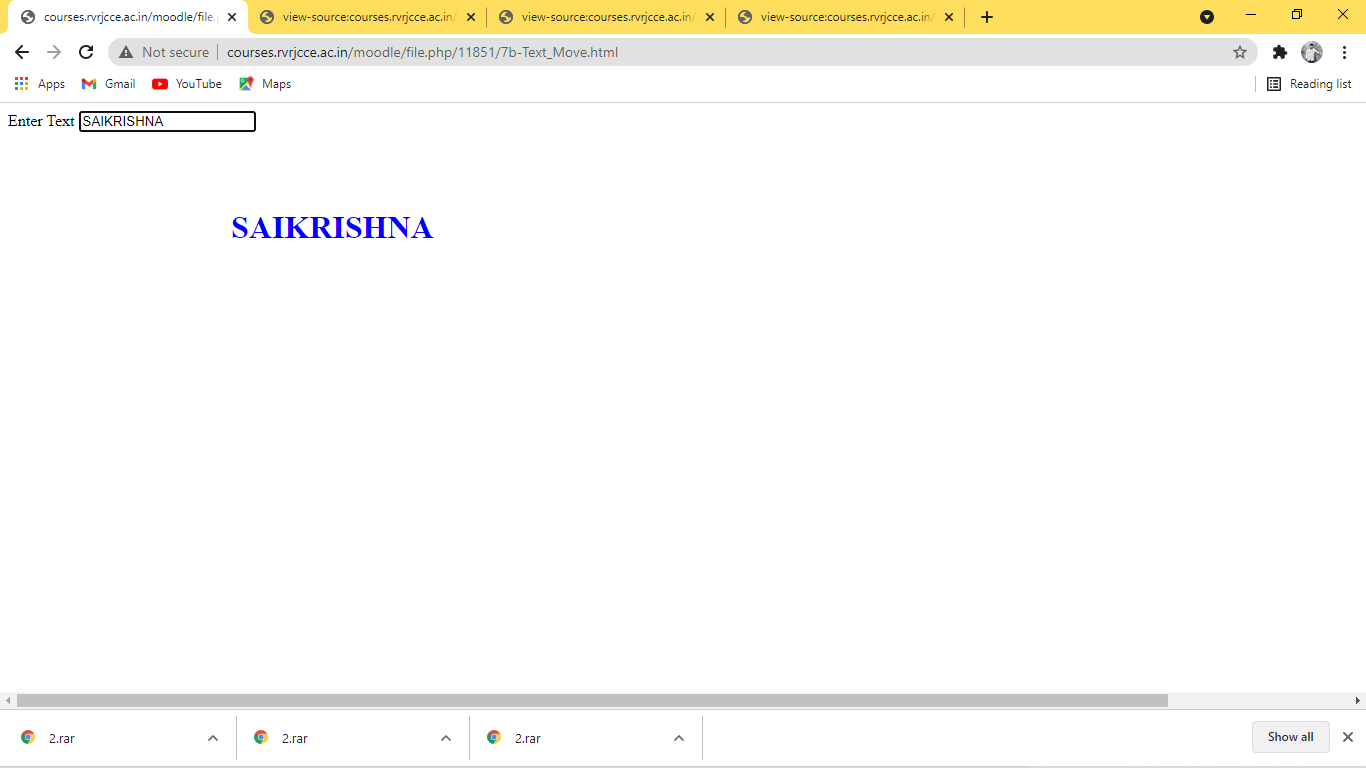
Enter Text <Input type="text" id="t1"/><br>

<h1 id="h1text" style="position:relative; color:blue; "> </h1>

</body>

</html>

OUTPUT:



8.Design a simple web page containing user’s login information. Validate that form by using following conditions.

* 1. Name should be entered.
  2. Password must be greater than or equal to 6 characters.
  3. Ignore the submit event.
  4. Prompt the user whether he wants to clear the data or not. If yes then only clear the login information from the text fields.

SOURCE CODE:

<script>

function validate()

{

//event.returnvalue=false;

if(document.f1.t1.value=="")

{

window.alert("Please enter the username");

return false;

}

else if(document.f1.p1.value=="")

{

window.alert("Please enter the Password");

return false;

}

else if(document.f1.p1.value.length<6)

{

window.alert("The password should be minimum 6 characters");

return false;

}

else

window.alert("You logged in successfully");

}

function Clear()

{

v=window.confirm("Do you want to clear the data");

if(v)

{

document.f1.t1.value="";

document.f1.p1.value="";

}

return false;

}

</script>

<form name="f1" onsubmit="return validate()" onreset="return Clear()">

<table>

<tr>

<td>Username :</td>

<td><input type="text" name="t1" size="20" /></td>

</tr>

<tr>

<td>Password :</td>

<td><input type="password" name="p1" size="20" /></td>

</tr>

<tr>

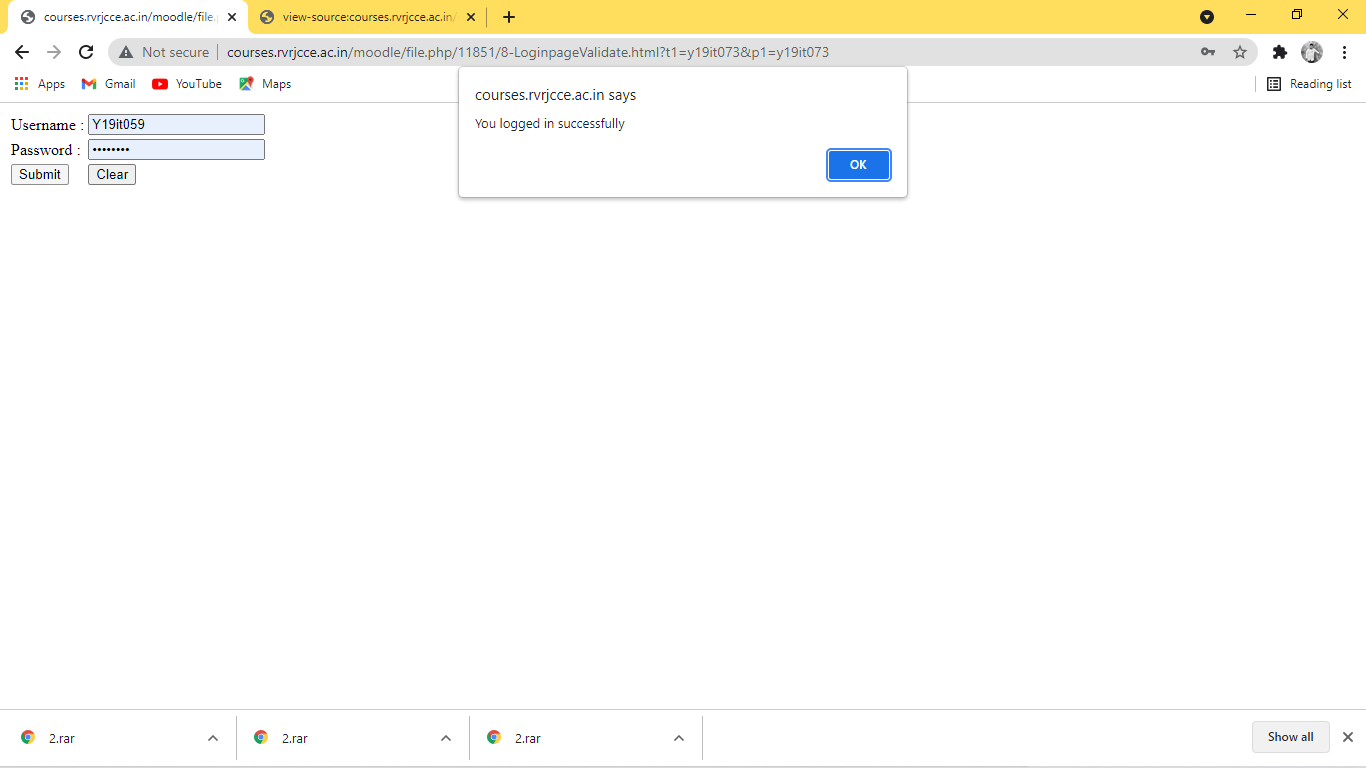
<td><input type="submit" value="Submit" /></td>

<td><input type="reset" value="Clear" /></td></tr>

</table>

</form>

OUTPUT:



1. Write a script to dynamically change the color and font size of the *textarea*. Select color & size using select box.

SOURCE CODE:

<html>

<head>

<title>color change</title>

</head>

<body>

<form>

color<select name="colors" id="t1">

<option value="green">green</option>

<option value="red">red</option>

<option value="blue">blue</option>

<option value="yellow">yellow</option>

</select><br><br>

size<select name="size" id="t2">

<option value="30%">30%</option>

<option value="40%">40%</option>

<option value="80%">80%</option>

</select><br><br>

<textarea name="message" rows="10" cols="10" id="t3">

welcome to rvr

</textarea><br>

<br>

<button type="button" onclick="ok()">ok</button><br><br>

<input type="submit" value="Submit">

</form>

</body>

<script>

function ok()

{

x=document.getElementById("t3");

y=document.getElementById("t1");

z=document.getElementById("t2");

x.style.color=y.value;

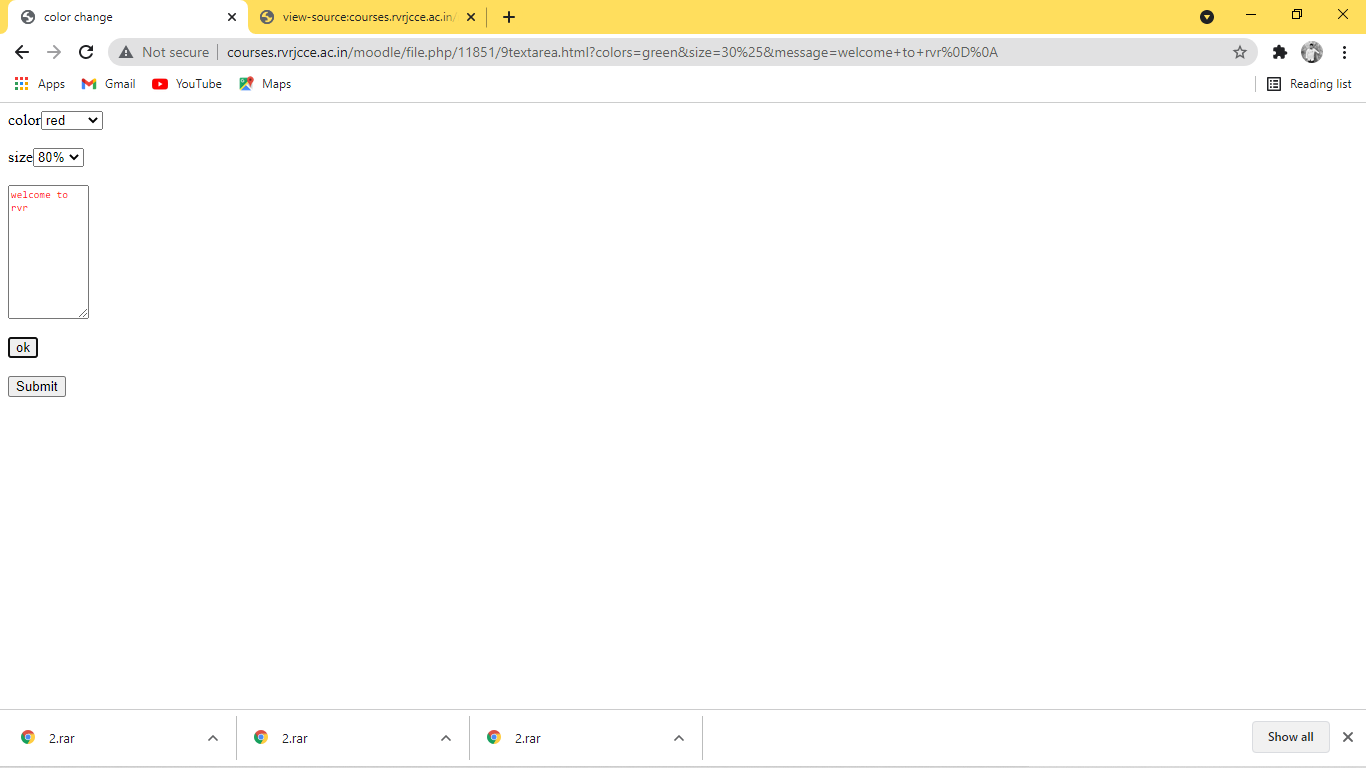
x.style.fontSize=z.value;

}

</script>

</html>

OUTPUT:



10.Write java script which uses random number generation to create sentences. Use four arrays of strings called **article**, **noun**, **verb**, and **preposition**. Create a sentence by selecting a word at random from each array in the following order.

**Article, noun, verb, preposition, article, and noun.**

As each word is picked, concatenate it to the previous words in the sentence. The words should be separated by spaces. When the final sentence is output, it should start with capital letter and end with period. The program should generate 20 sentences and output them to an HTML TEXTAREA.

The arrays should be filled as follows.

**Article** array contains “**the**”, “**a**”, “**one**”, “**some**”, and “**any**”.

**Noun** array contains “**boy**”, “**girl**”, “**dog**”, “**town**”, and “**car**”.

**Verb** array contains “**drove**”, “**jumped**”, “**ran**”, “**walked**”, and “**skipped**”.

**Preposition** array contains “**to**”, “**from**”, “**over**”, “**under**”, and “**on**”.

SOURCE CODE:

<html>

<head>

<body>

<form name="f1">

<textarea id="t1" rows="20" cols="35"> </textarea>

<input type="button" value="click me" onclick="generate()">

</form>

</body>

<script>

function generate()

{

var A=new Array("the","a","one","some");

var N=new Array("boy","girl","dog","town","car");

var V=new Array("drove","jumped","ran","walked","skipped");

var P=new Array("to","from","over","under","on");

var sent="";

var total="";

//var a = Math.random()\*4

//alert(a)

//alert(Math.round(a))

for(i=0;i<20;i++)

{

var w1=new String(A[Math.round(Math.random()\*4)]);

var w2=new String(N[Math.round(Math.random()\*4)]);

var w3=new String(V[Math.round(Math.random()\*4)]);

var w4=new String(P[Math.round(Math.random()\*4)]);

var w5=new String(A[Math.round(Math.random()\*4)]);

var w6=new String(N[Math.round(Math.random()\*4)]);

var l1=w1.substr(0,1)

var l2=w1.substr(1);

w1=l1.toUpperCase(l1)+l2

sent=w1+" "+w2+" "+w3+" "+w4+" "+w5+" "+w6+"."+"\n";

total=total+sent;

}

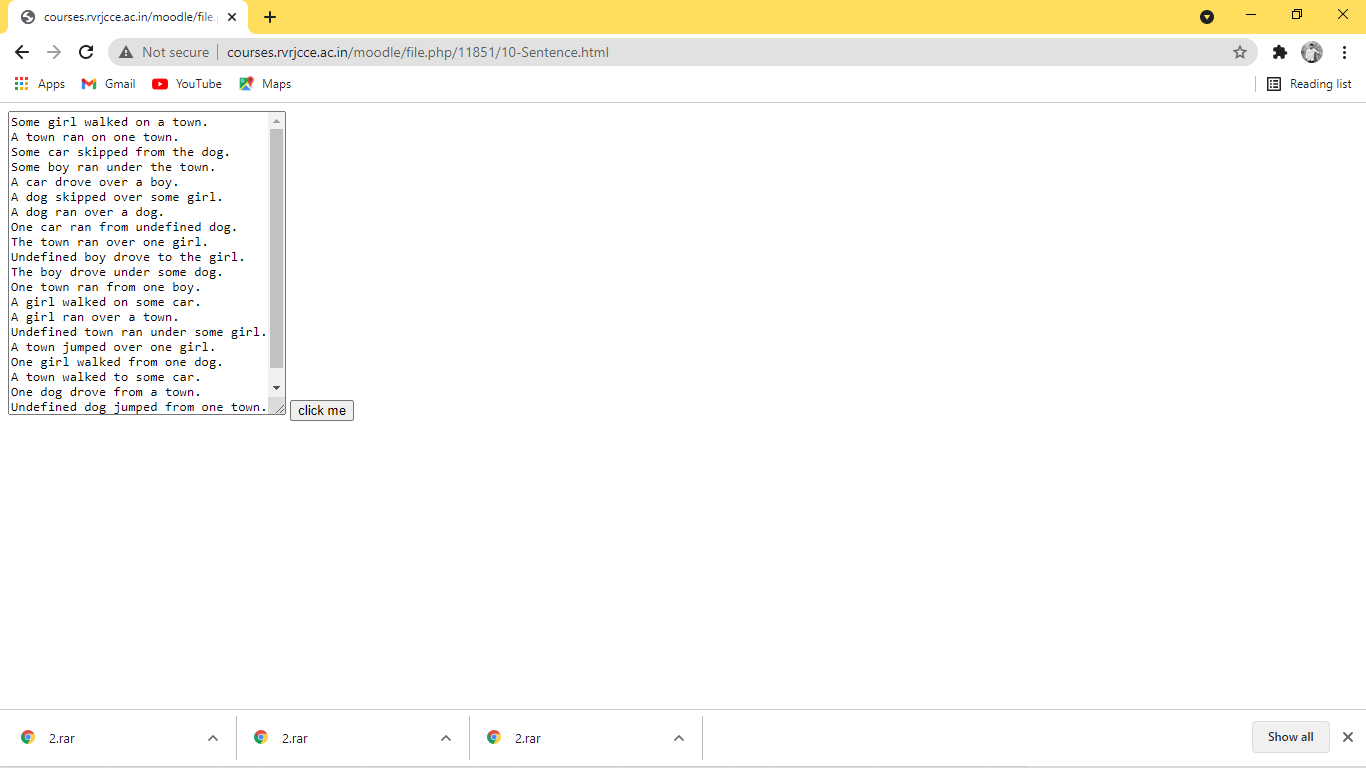
document.f1.t1.value=total;

}

</script>

</html>

OUTPUT:



11.Display all anchor elements in a given page into a XHTML table using DOM collections.

SOURCE CODE:

<html>

<head>

<title>app</title>

</head>

<body onload="alllinks()">

<p>hi this is to display all links in a program

<table style="border:1px red solid;">

<tr> <td><a href="[http://www.yahoo.com](http://www.yahoo.com/)">yahoo</a></td></br> </tr>

<tr> <td><a href="[http://www.google.com](http://www.google.com/)">google</a></td></br> </tr>

<tr> <td><a href="[http://www.gmail.com](http://www.gmail.com/)">gmail</a></td></br> </tr>

</table>

ending of the program </p>

<a href="[http://www.gmail.com](http://www.gmail.com/)">gmail</a>

<area shape="circle" coords="73,168,32" href="[DOM-1.html](http://courses.rvrjcce.ac.in/moodle/file.php/11851/DOM-1.html)"> </area>

<p id="para"> </p>

<script>

function alllinks()

{

/\*var x=document.images;

var y=document.anchors;

var z=document.forms; \*/

var l=document.links;

var txt = " "

for(i=0;i<l.length;i++)

txt +=l[i].href + " <br> "

document.getElementById('para').innerHTML = txt

}

</script>

</body>

</html>

OUTPUT:

hi this is to display all links in a program

[yahoo](http://www.yahoo.com/)

[google](http://www.google.com/)

[gmail](http://www.gmail.com/)

ending of the program

[gmail](http://www.gmail.com/)

http://www.yahoo.com/  
http://www.google.com/  
http://www.gmail.com/  
http://www.gmail.com/  
http://courses.rvrjcce.ac.in/moodle/file.php/11851/DOM-1.html

12.Display XML data in the XHTML table (Data Island).

SOURCE CODE:

<html>

<body>

<xml id="students">

<?xml version ="1.0"?>

<class>

<student>

<sno>123</sno>

<sname>abc</sname>

</student>

<student>

<sno>124</sno>

<sname>xyz</sname>

</student>

<student>

<sno>125</sno>

<sname>pqr</sname>

</student>

</class>

</xml>

<table datasrc="#students" border="2">

<tr>

<th><div DATAFLD="sno"></div></td>

<th><div DATAFLD="sname"></div></td>

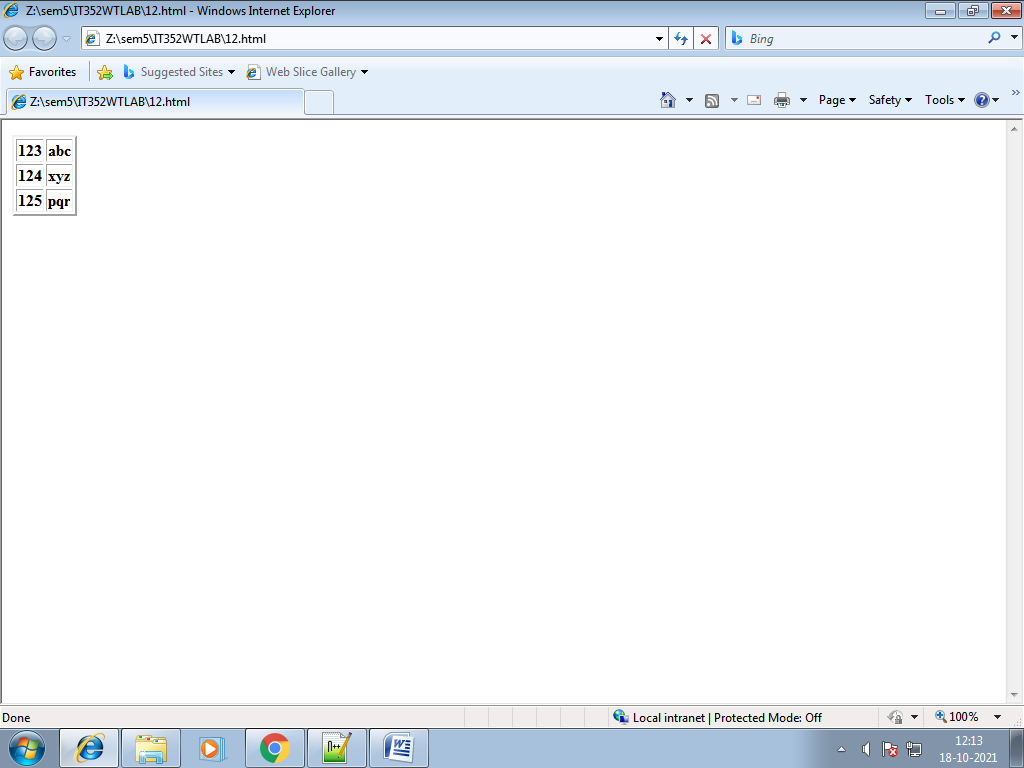
</tr>

</table>

</body>

</html>

OUTPUT :



1. Validate XML documents using DTD & XML Schema (XSD).

SOURCE CODE:

<?xml version="1.0"?>

<!DOCTYPE student

[

<!ELEMENT student (sname,address,marks)>

<!ELEMENTsname (#PCDATA)>

<!ELEMENT address (#PCDATA)>

<!ELEMENT marks (#PCDATA)>

]>

<student>

<sname>ram</sname>

<address>gnt</address>

<marks>70</marks>

</student>

13.XML

<?xml version="1.0"?>

<student xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="13.xsd">

<sname>ram</sname>

<address>gnt</address>

<marks>70</marks>

</student>

13.XSD

<?xml version="1.0"?>

<xs:schemaxmlns="http://www.w3.org/2001/XMLSchema">

<xs:element name="student">

<xs:complex Type>

<xs:sequence>

<xs:element name="sname" type="xs:string">

<xs:element name="address" type="xs:string">

<xs:element name="marks" type="xs:integer">

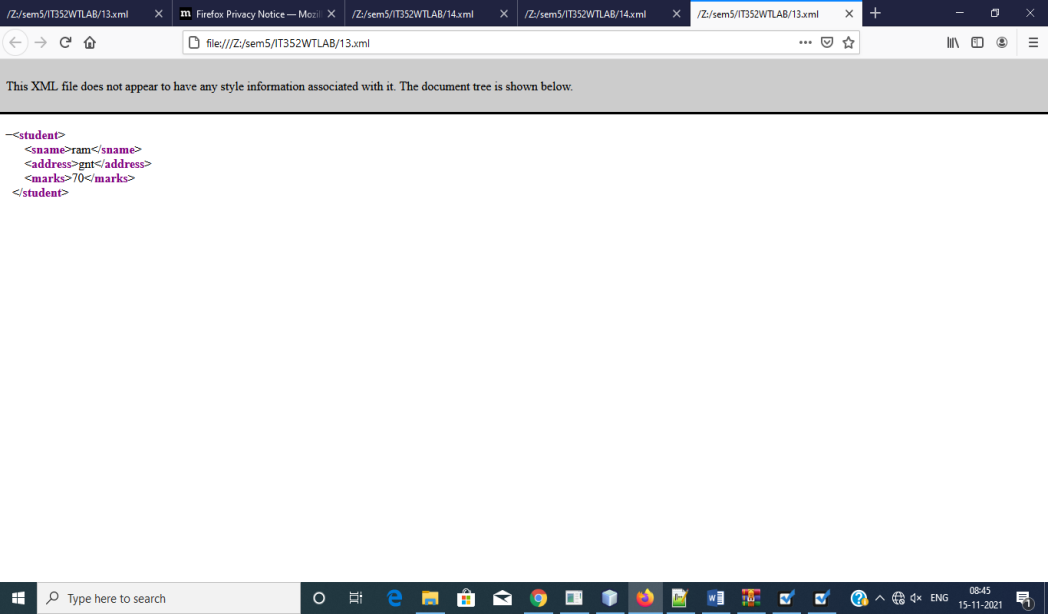
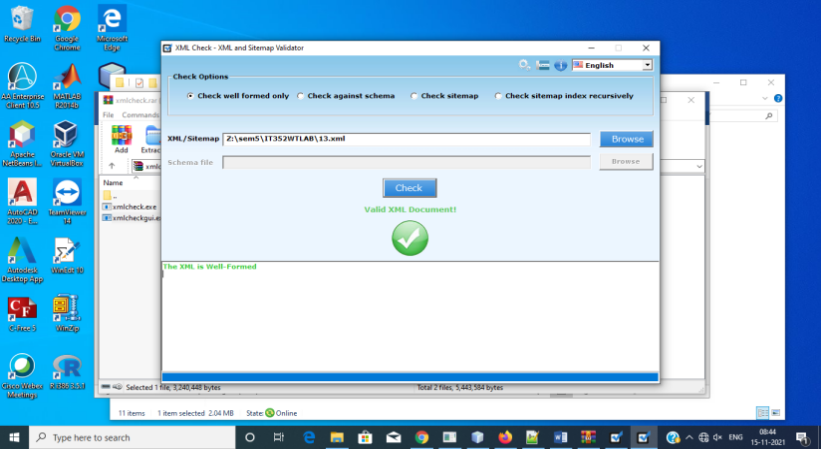
</xs:sequence>

</xs:complex Type>

</xs:element>

</xs:schema>

OUTPUT:



1. Display XML data in the XHTML table using XSL.

SOURCE CODE:

<?xml version="1.0"?>

<?xml-stylesheet type="text/xsl" href="14.xsl"?>

<students-info>

<student gender="m">

<name>xyz</name>

<rollno>04</rollno>

<address>

<city>tnl</city>

<contactno>755215</contactno>

</address>

</student>

<student gender="f">

<name>abc</name>

<rollno>02</rollno>

<address>

<city>gnt</city>

<contactno>755215</contactno>

</address>

</student>

<student gender="f">

<name>mno</name>

<rollno>03</rollno>

<address>

<city>gnt</city>

<contactno>755215</contactno>

</address>

</student>

</students-info>

14.XSL

<?xml version="1.0"?>

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

<xsl:template match="/">

<html>

<body>

<h3 align="center">Student Info</h3>

<table border="3" align="center">

<trbgcolor="yellow">

<td>roll no.</td>

<td>name</td>

<td>gender</td>

<td>city</td>

</tr>

<xsl:for-each select="students-info/student">

<xsl:choose>

<xsl:when test=" @gender = 'm' ">

<trbgcolor="red">

<td ><xsl:value-of select="rollno"/></td>

<td><xsl:value-of select="name"/></td>

<td><xsl:value-of select="@gender"/></td>

<td><xsl:value-of select="address/city"/></td>

</tr>

</xsl:when>

<xsl:otherwise>

<trbgcolor="green">

<td><xsl:value-of select="rollno"/></td>

<td><xsl:value-of select="name"/></td>

<td><xsl:value-of select="@gender"/></td>

<td><xsl:value-of select="address/city"/></td>

</tr>

</xsl:otherwise>

</xsl:choose>

</xsl:for-each>

</table>

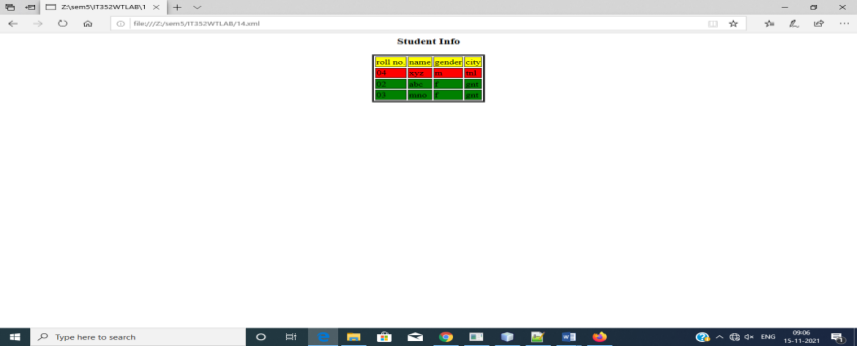
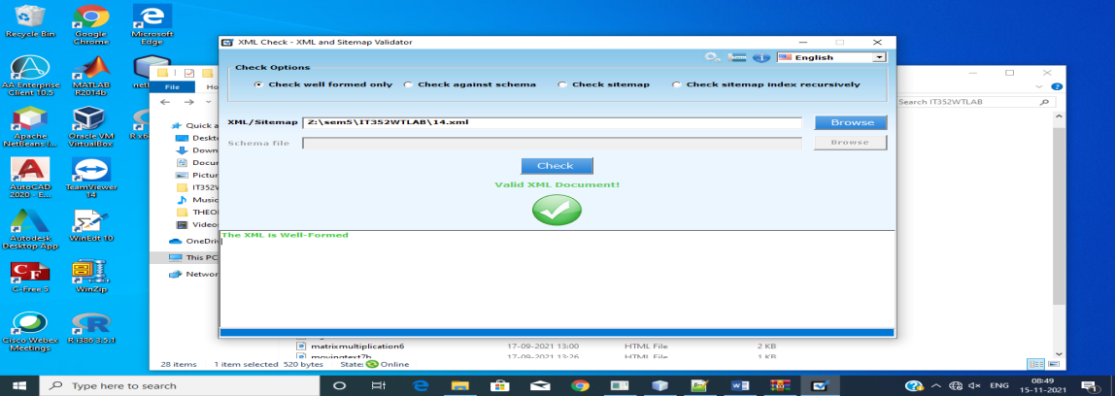
</body>

</html>

</xsl:template>

</xsl:stylesheet>

OUTPUT:



1. Write a program to demonstrate Generic and HTTP servlets.

SOURCE CODE:

GENERIC SERVLET

INDEX.JSP

<%--

Document : index

Created on : 1 Nov, 2021, 9:57:36 AM

Author : y19it59

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

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

<title>JSP Page</title>

</head>

<body>

<center>

<form name="form1" method="post" action="gservlet1">

<table>

<tr>

<td>

name:<input type="text" name="ename"><br>

</td>

<td>

password:<input type="text" name="ephone">

</td>

</tr>

</table>

<input type=submit value="Submit">

</form>

</center>

</body>

</html>

GSERVLET.JAVA

import java.io.\*;

import javax.servlet.\*;

import java.util.\*;

public class gservlet1 extends GenericServlet {

public void service(ServletRequest request, ServletResponse response)

throws ServletException, IOException {

PrintWriterpw=response.getWriter();

Enumeration e=request.getParameterNames();

while(e.hasMoreElements()){

String pname=(String)e.nextElement();

pw.print(pname+" = ");

String pvalue=request.getParameter(pname);

pw.println(pvalue);

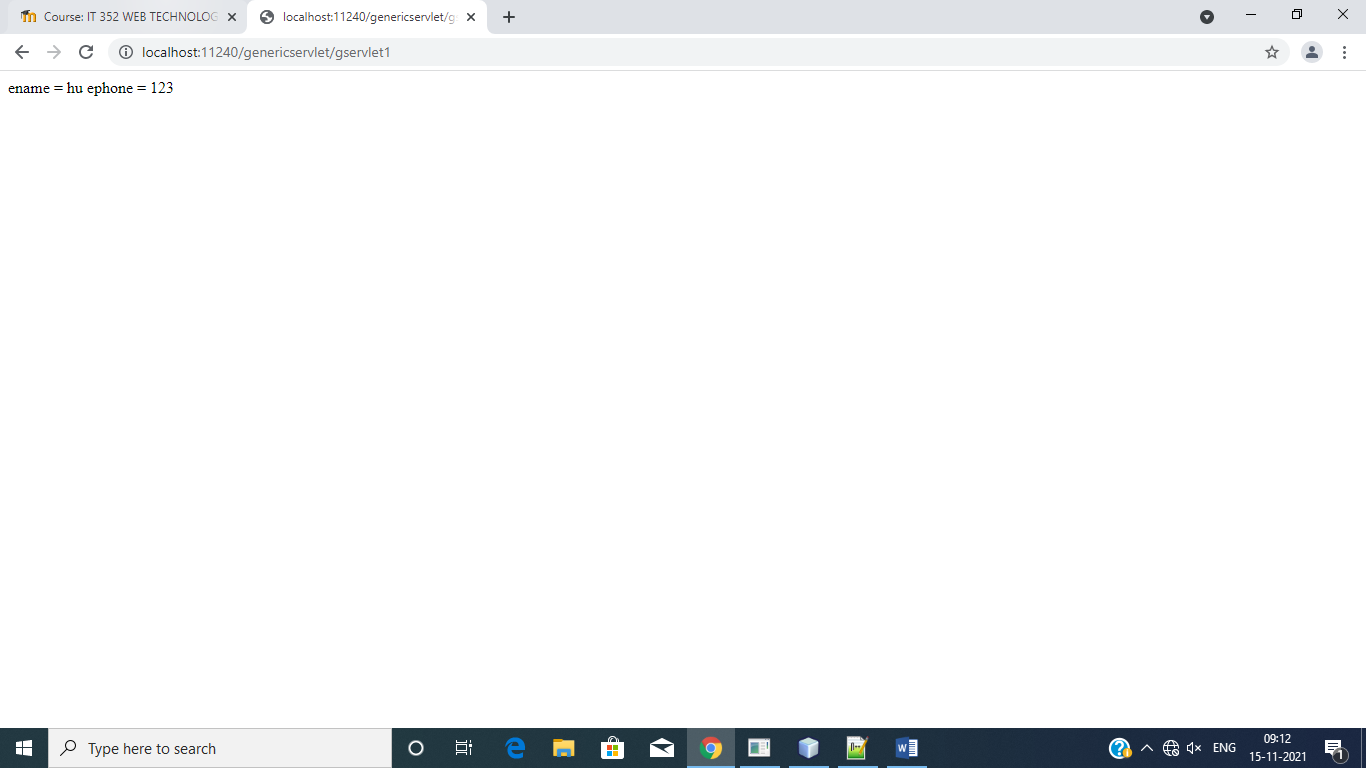
}

pw.close();

}

}

OUTPUT:



HTTP SERVLET:

SOURCE CODE:

INDEX.JSP

%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

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

<title>JSP Page</title>

</head>

<body>

<center>

<form name="form1" action="h1servlet" method="get">

color:<select name="color" size="1">

<option value="Red">Red</option>

<option value="Green">Green</option>

<option value="Blue">Blue</option>

<input type="submit" value="Submit">

</select>

</form>

</center>

</body>

</html>

HSERVLET.JAVA

mportjava.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class h1servlet extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

String color=request.getParameter("color");

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

String str="<b style='color:"+color+"'>The selected color is:";

out.println(str);

out.println(color);

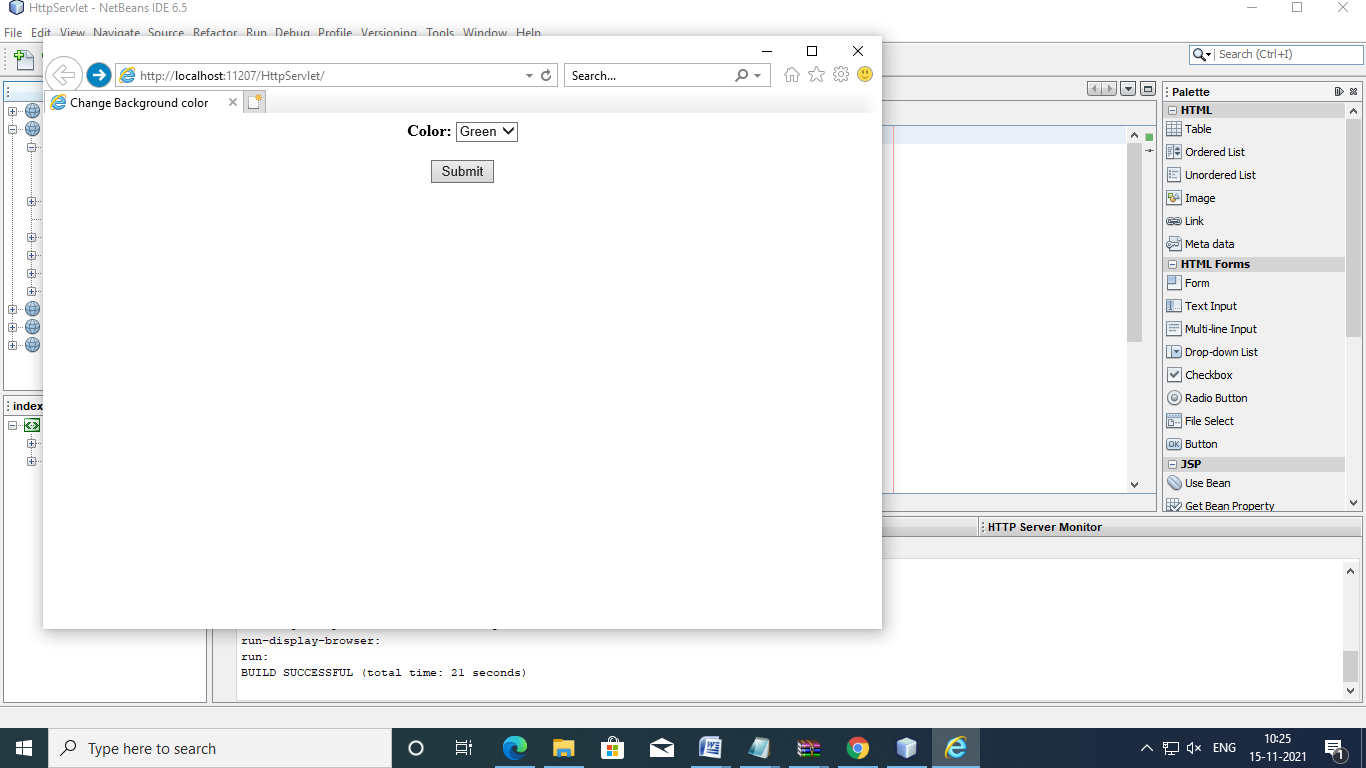
out.close();

} finally {

}

}

OUTPUT:



1. Write a program to demonstrate cookies.

SOURCE CODE:

INDEX.JSP

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

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

<title>JSP Page</title>

</head>

<body>

<center>

<form action="servlet1" method="post">

name:<input type="text" name="username"><br>

<input type="submit" value="go">

</form>

</center>`

</body>

</html>

SERVLET1.JAVA

import java.io.\*;

import javax.servlet.ServletException;

import javax.servlet.http.\*;

public class servlet1 extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

String n=request.getParameter("username");

out.print("welcome "+n);

Cookie ck=new Cookie("uname",n);

response.addCookie(ck);

out.print("<form action='servlet2' method='post'>");

out.print("<input type='submit' value='go'>");

out.print("</form>");

out.close();

} finally {

}

}

SERVLET2.JAVA

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

/\*\*

\*

\* @author y19it059

\*/

public class servlet2 extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

Cookie ck[]=request.getCookies();

out.println("hello "+ck[0].getValue());

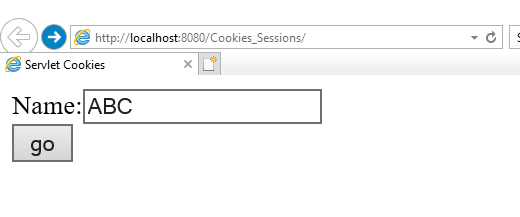
out.close();

} finally {

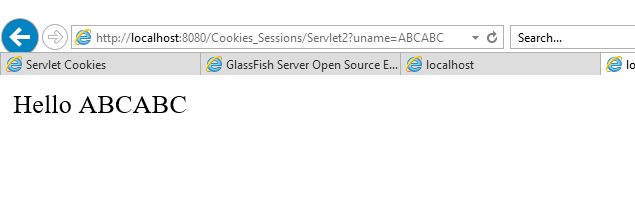
}

}

OUTPUT:







1. Write a program to demonstrate Sessions using HTTP servlets.\

SOURCE CODE:  
SESSIONS:

INDEX.JSP:

<%--

Document : index

Created on : 8 Nov, 2021, 9:31:46 AM

Author : y19it59

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

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

<title>JSP Page</title>

</head>

<body>

<form action="session1" method="get">

name:<input type="text" name="username"/><br>

<input type="submit" value="go"/>

</form>

</body>

</html>

SESSION1.JAVA:

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

/\*\*

\*

\* @author y19it59

\*/

public class session1 extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

String n=request.getParameter("username");

out.print("welcome "+n);

HttpSession session=request.getSession();

session.setAttribute("uname",n);

out.print("<br><br><a href='session2'>visit</a>");

out.close();

} catch(Exception e){System.out.println(e);}

}

SESSION2.JAVA:

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

/\*\*

\*

\* @author y19it59

\*/

public class session2 extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

HttpSession session=request.getSession(false);

String n=(String)session.getAttribute("uname");

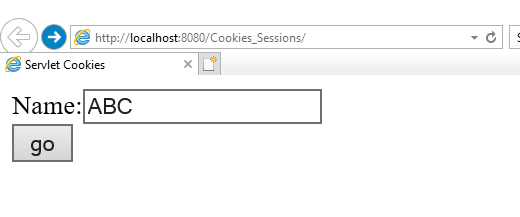
out.print("hello "+n);

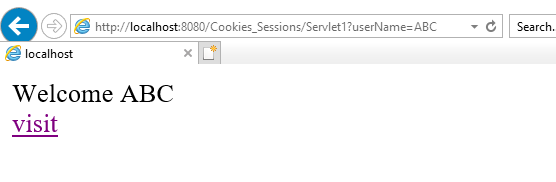
out.close();

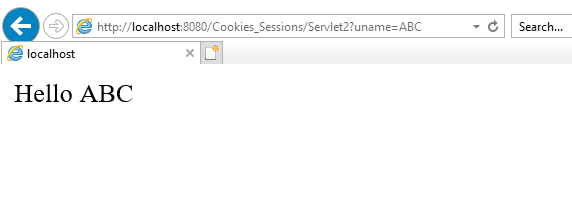
}catch(Exception e){System.out.println(e);}

}

OUTPUT:







18.Create a Java web application to validate user login details by implementing the following.

* Create one JSP (ex: login.jsp ) in which you will submit your “user name” and “password”.
* And create one servlet (ex: ControllerServlet) that will check if the password entered by the user is correct or not. If the username & password entered by the user is correct then your controller servlet will redirect the client request to another servlet (ex: ValidUserServlet) which will display welcome message. If the password entered by the user is wrong then the request will be forwarded to your login.JSP page (Use Send Redirect in servlet).

SOURCE CODE:

Login.jsp

<%--

Document : index

Created on : 15 Nov, 2021, 9:29:31 AM

Author : y19it59

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

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

<title>JSP Page</title>

</head>

<body>

<form action="Valid" method="post">

user:<input type="text" name="user">

passsword:<input type="text" name="password">

<input type="submit" value="ok">

</form>

</body>

</html>

VALID.JAVA

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class Valid extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

try {

String u=request.getParameter("user");

String p=request.getParameter("password");

if ((u.equals("rvr") &&p.equals("jc"))){

response.sendRedirect("Welcome");

}

else

response.sendRedirect("login.jsp");

} finally {

out.close();

}

}

}

WELCOME.JAVA

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\*

\* @author y19it59

\*/

public class Welcome extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

try {

out.println("hi");

} finally {

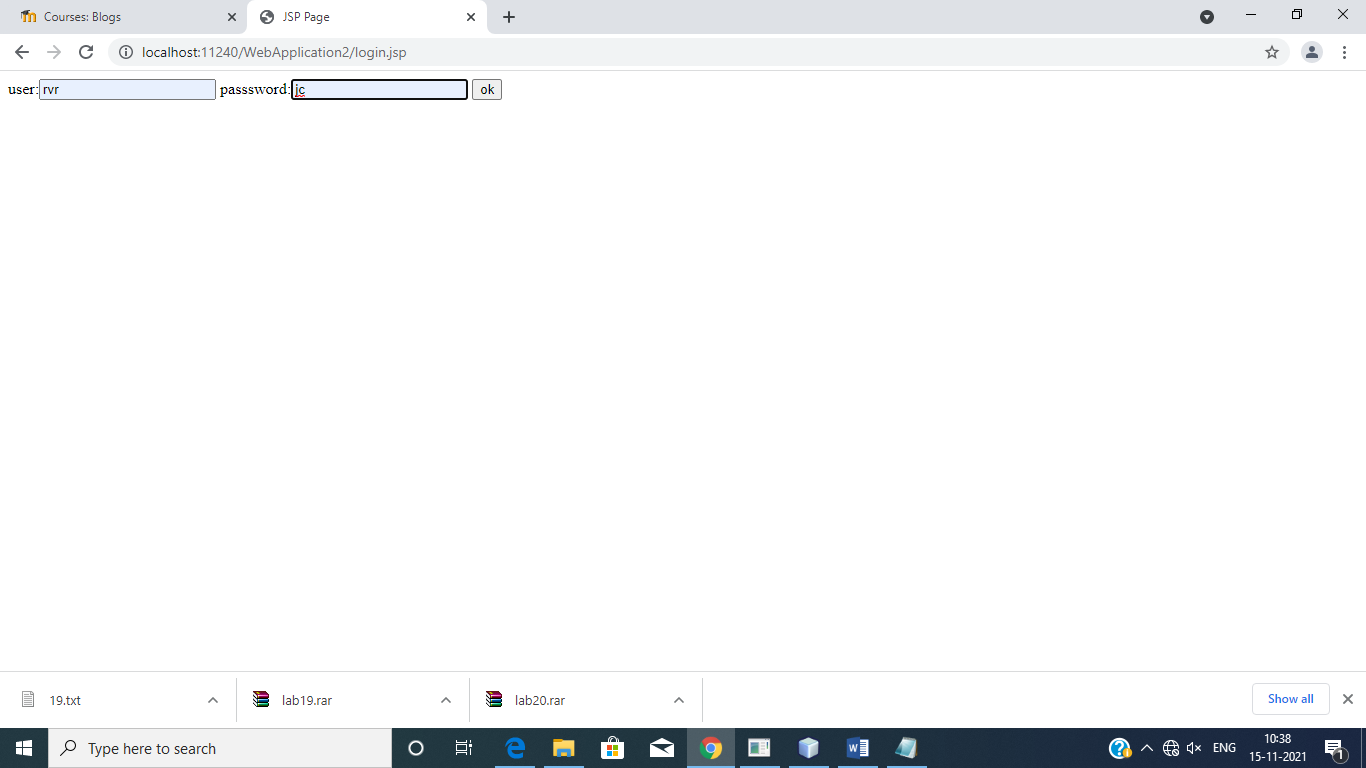
out.close();

}

}

}

OUTPUT:



hi

19.Make a “Course registration” form using JSP, that collects a first name, last name, contact no, email address & course name.

* Create 4 text boxes for first name, last name, contact no, email address.
* Use 3 check boxes for multiple selection.
* Send the registration information to a servlet that displays it. (use HTTP POST method)

:

SOURCE CODE”:

INDEX.JSP

<%--

Document : index

Created on : 15 Nov, 2021, 10:26:14 AM

Author : y19it59

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

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

<title>JSP Page</title>

</head>

<body>

<form method="get" action="Register">

fname:<input type="text" name="fname"><br><br>

lname:<input type="text" name="lname"><br><br>

contact:<input type="number" name="contact"><br><br>

email:<input type="email" name="email"><br><br>

select courses

<input type="checkbox" name="rvr" value="os">os<br><br>

<input type="checkbox" name="rvr" value="ds">ds<br><br>

<input type="checkbox" name="rvr" value="oops">c++<br><br>

<input type="submit" value="ok">

</form>

</body>

</html>

REGISTER.JAVA

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.\*;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class Register extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

try {

String f=request.getParameter("fname");

String l=request.getParameter("lname");

String c=request.getParameter("contact");

String e=request.getParameter("email");

String a[]=request.getParameterValues("rvr");

out.println(f);

out.println(l);

out.println(c);

out.println(e);

for(inti=0;i<a.length;i++)

{

out.println(a[i]);

}

} finally {

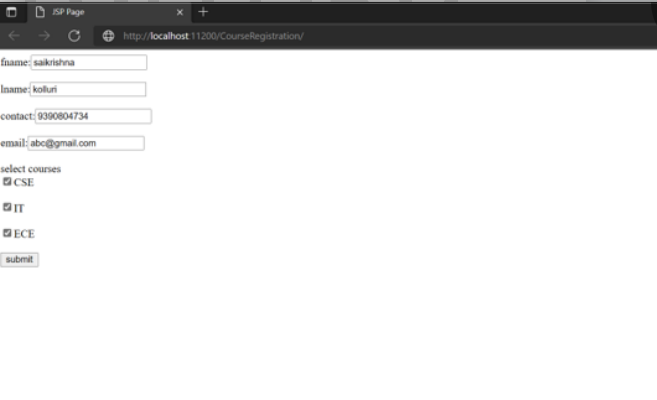
out.close();

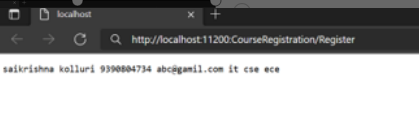
}

}

}

OUTPUT:





1. Create a bean to maintain student information and display the information in JSP page.

SOURCE CODE:

INSERT.JSP

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

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

<title>JSP Page</title>

</head>

<body>

<%

String u=request.getParameter("name");

String i=request.getParameter("id");

String p=request.getParameter("per");

String a=request.getParameter("addr");

%>

<jsp:useBean id="std" class="p1.Student" scope="session">

<jsp:setProperty name="std" property="stdname" value="<%=u%>"/>

<jsp:setProperty name="std" property="stdid" value="<%=i%>"/>

<jsp:setProperty name="std" property="stdper" value="<%=p%>"/>

<jsp:setProperty name="std" property="stdaddr" value="<%=a%>"/>

<jsp:getProperty name="std" property="stdname"/>

<jsp:getProperty name="std" property="stdid"/>

<jsp:getProperty name="std" property="stdper"/>

<jsp:getProperty name="std" property="stdaddr"/>

</jsp:useBean>

</body>

</html>

STD.JSP

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

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

<title>JSP Page</title>

</head>

<body>

<h1>Hello World!</h1>

<form method="post" action="insert.jsp">

<h1>User login page</h1>

name<input type="text" name="name"><br>

id<input type="text" name="id"><br>

percentage<input type="text" name="per"><br>

address<input type="text" name="addr"><br>

<input type="submit" value="submit">

<input type="reset" value="reset">

</form>

</body>

</html>

STUDENT.JAVA

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package p1;

/\*\*

\*

\* @author y19it59

\*/

public class Student {

public String stdname;

public String stdid;

public String stdper;

public String stdaddr;

public Student()

{

}

public void setStdname(String n)

{

stdname=n;

}

public void setStdid(String i)

{

stdid=i;

}

public void setStdper(String p)

{

stdper=p;

}

public void setStdaddr(String a)

{

stdaddr=a;

}

public String getStdname()

{

return stdname;

}

public String getStdid()

{

return stdid;

}

public String getStdper()

{

return stdper;

}

public String getStdaddr()

{

return stdaddr;

}

}

OUTPUT:

