A Laboratory Manual for

Client Side Scripting Language (22519) Semester –V (CM)

Maharashtra State Board of Technical Education, Mumbai.

Maharashtra State

Board of Technical Education

Certificate

This is to certify that Mr./Ms. ……

Roll No……..of Third Semester of Diploma in Computer Technology of

Institute,……

(Code : ……..) has completed the term work satisfactorily in course

Client Side Scripting languages (22519) for the academic year …… to

…….. as Prescribed in the curriculum.

Place: …… **Enrollment No: ……..**

Date: ……. **Exam. Seat No: ……**

Subject Teacher **Head of Department** **Principal**

Seal of Institution

Program Outcomes (POs) to be achieved through Practical of this Course:-

PO 1.**Basic knowledge**: Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Computer engineering problem.

PO 2.**Discipline knowledge**: Apply Computer engineering discipline - specific knowledge to solve core computer engineering related problems.

PO 3.**Experiments and practice**: Plan to perform experiments and practices to use the results to solve broad-based Computer engineering problems.

PO 4.**Engineering tools**: Apply relevant Computer technologies and tools with an understanding of the limitations.

PO 5.**The engineer and society**: Assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to practice in field of Computer engineering.

PO 6.**Environment and sustainability**: Apply Computer engineering solutions also for sustainable development practices in societal and environmental contexts and demonstrate the knowledge and need for sustainable development.

PO 7**. Ethics**: Apply ethical principles for commitment to professional ethics, responsibilities and norms of the practice also in the field of Computer engineering.

PO 8.**Individual and team work**: Function effectively as a leader and team member in diverse/ multidisciplinary teams.

PO 9.**Communication**: Communicate effectively in oral and written form.

PO 10.**Life-long learning**: Engage in independent and life-long learning activities in the context of technological changes in the Computer engineering field and allied industry.

Content Page

List of Practical's and Progressive assessment Sheet

Sr.No	Title of practical	Date of performance	Date of submission	Assessment marks(25)	Dated sign of Teacher	Remarks
1	Write simple JavaScript with HTML for arithmetic expression evaluation and message printing.					
2	Develop JavaScript to use decision making and looping statements					
3	Develop JavaScript to implements Array functionalities					
4	Develop javascript to implement functions					
5	Develop javascript to implement Strings.					
6	Create web page using Form Elements					
7	Create web page to implement Form Events .Part I					
8	Create web page to implement Form Events .Part II					
9	Develop a webpage using intrinsic java functions					
10	Develop a webpage for creating session and persistent cookies. Observe the effects with browser cookies settings.					
11	Develop a webpage for placing the window on the screen and working with child window.					
12	Develop a web page for validation of form fields using regular expressions.					
13	Create web page with Rollovers effect.					
14	Develop a webpage for implementing Menus.					
15	Develop a webpage for implementing Status bars and web page protection.					
16	Develop a web page for implementing slideshow, banner.					
Total						

Practical No 1: Write simple JavaScript with HTML for arithmetic expression evaluation and message printing.

What is JavaScript?

- It is designed to add interactivity to HTML pages
- It is a scripting language (a lightweight programming language)
- It is an interpreted language (it executes without preliminary compilation)
- Usually embedded directly into HTML pages
- And, Java and JavaScript are different

What can a JavaScript Do?

- JavaScript gives HTML designers a programming tool:
    - simple syntax
- JavaScript can put dynamic text into an HTML page
- JavaScript can react to events
- JavaScript can read and write HTML elements
- JavaScript can be used to validate data
- JavaScript can be used to detect the visitor's browser
- JavaScript can be used to create cookies
    - Store and retrieve information on the visitor's computer

JavaScript How To

- The HTML <script> tag is used to insert a JavaScript into an HTML page

```
<script type="text/javascript">
document. write("Hello World!")
</script>
```

- Ending statements with a semicolon?
    - Optional; required when you want to put multiple statements on a single line

JavaScript can be inserted within the head, the body, or use external JavaScript file

How to handle older browsers?

<script type="text/javascript">

<!—

document.write("Hello World!")

// -->

</script>

JavaScript can "display" data in different ways:

- Writing into an HTML element, using innerHTML.

To access an HTML element, JavaScript can use the document.getElementById(id) method.
The id attribute defines the HTML element. The innerHTML property defines the HTML content.

- Writing into the HTML output using document.write().
- Writing into an alert box, using window.alert().
- Writing into the browser console, using console.log().

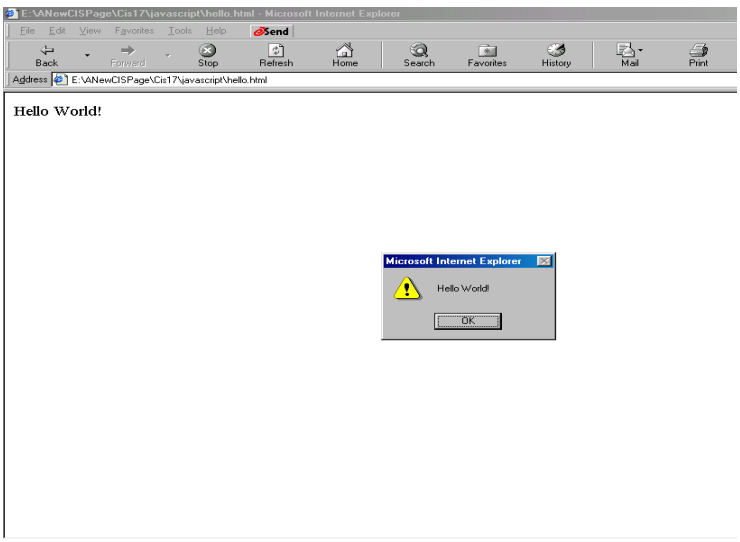
JavaScript Variables

- In a programming language, **variables** are used to **store** data values.
- JavaScript uses the var keyword to **declare** variables.
- An **equal sign** is used to **assign values** to variables.

JavaScript Arithmetic Operators

- Arithmetic operators are used to perform arithmetic on numbers:

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus (Remainder)
++	Increment
--	Decrement



1.Simple Java Script Program

<html>

<script language="JavaScript">

```
document. write("Hello World!");

alert("Hello World!");

</script>

</html>
```

2.Perform Multiplication of Two Numbers

```
<html>

<script language="JavaScript">

varans = 0;

varfirstnum = 0;

varsecondnum = 0;

firstnum = prompt("Enter the first number",0);

secondnum = prompt("Enter the second number",0);

ans = firstnum * secondnum;

document.write(ans);

</script>

</html>
```

Questions:

1. Which company developed JavaScript?
2. What are JavaScript Data Types?
3. How to declare variable in Javascript?
4. What are arithmetical operators?

Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	

Practical No2:-Develop JavaScript to use decision making and looping statements

Conditional Statements:

1. The if Statement

Use the if statement to specify a block of JavaScript code to be executed if a condition is true.

Syntax

```
if (condition) {
  // block of code to be executed if the condition is true
}
```

2.The else Statement

Use the else statement to specify a block of code to be executed if the condition is false.

Syntax

```
if (condition) {
  // block of code to be executed if the condition is true
} else {
  // block of code to be executed if the condition is false
}
```

3.The else if Statement

Use the else if statement to specify a new condition if the first condition is false.

Syntax

```
if (condition1) {
  // block of code to be executed if condition1 is true
} else if (condition2) {
  // block of code to be executed if the condition1 is false and condition2 is true
} else {
  // block of code to be executed if the condition1 is false and condition2 is false
}
```

4.TheSwitch Statement

Use the switch statement to select one of many code blocks to be executed.

Syntax

```
switch(expression) {
  case x:
    // code block
    break;
```

```
  case y:
    // code block
    break;
  default:
    // code block
}
```

JavaScript Loops

1. for loop
Loops are handy, if you want to run the same code over and over again, each time with a different value.

Syntax:-
for (initialization condition; testing condition; increment/decrement)
{
statement(s)
}
Or for objects
for (variableName in Object)
{
statement(s)
}

2. do while:
do while loop is similar to while loop with only difference that it checks for condition after executing the statements, and therefore is an example of Exit Control Loop.

Syntax:
do
{
statements..
}while (condition);

3. While loop
A while loop is a control flow statement that allows code to be executed repeatedly based on a given Boolean condition. The while loop can be thought of as a repeating if statement.

Syntax :
while (boolean condition)
{
loop statements...
}

Programs:
1.for loop

```
<script type = "text/javascript">
// JavaScript program to illustrate for loop
   varx;

   // for loop begins when x=2
   // and runs till x <=4
   for(x = 2; x <= 4; x++)
   {
      document.write("Value of x:"+ x + "<br />");
   }

< /script>
```

2.for..in loop

```
<script type = "text/javaScript">
// JavaScript program to illustrate for..in loop

   // creating an Object
   varlanguages = { first : "C", second : "Java",
                  third : "Python", fourth : "PHP",
                  fifth : "JavaScript"};

   // iterate through every property of the
   // object languages and print all of them
   // using for..in loops
   for(itr inlanguages)
   {
      document.write(languages[itr] + "<br>");
   }

< /script>
```

3.do ..while loop

```
<script type = "text/javaScript">
// JavaScript program to illustrate do-while loop

   varx = 21;

   do
   {
      // The line while be printer even
      // if the condition is false
      document.write("Value of x:"+ x + "<br />");

      x++;
   } while(x < 20);
```

< /script>

4.while loop

```
<script type = "text/javaScript">
// JavaScript program to illustrate while loop

   varx = 1;

   // Exit when x becomes greater than 4
   while(x <= 4)
   {
      document.write("Value of x:"+ x + "<br />");

      // increment the value of x for
      // next iteration
      x++;
   }

< /script>
```

5.if…else

```
<script type = "text/javaScript">

// JavaScript program to illustrate If-else statement

vari = 10;

if(i < 15)
  document.write("10 is less than 15");
else
  document.write("I am Not in if");

< /script>
```

6.switch case

```
<script type = "text/javascript">

   // JavaScript program to illustrate switch-case

   vari = 9;

   switch(i)
   {
     case0:
```

```
        document.write("i is zero.");
        break;
    case1:
        document.write("i is one.");
        break;
    case2:
        document.write("i is two.");
        break;
    default:
        document.write("i is greater than 2.");
  }
</script>
```

Questions

1. Is JavaScript case sensitive? Give an example?

2.What Boolean operators can be used in JavaScript?

Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	

Practical No-3.Develop JavaScript to implements Array functionalities

What is an Array?

An array is a special variable, which can hold more than one value at a time.

Creating an Array

Using an array literal is the easiest way to create a JavaScript Array.
Syntax:

var *array_name* = [*item1*, *item2*, ...];

JavaScript Array directly (new keyword)

The syntax of creating array directly is given below:
var arrayname=new Array();

Here, new keyword is used to create instance of array.

Eg :-1

```
<html>
<body>
<script>
var i;
varemp = new Array();
emp[0] = "Arun";
emp[1] = "Varun";
emp[2] = "John";

for (i=0;i<emp.length;i++){
document.write(emp[i] + "<br>");
}
</script>
</body>
</html>
```

2.

```
<html>
<body>
<script>
varemp=["Sonoo","Vimal","Ratan"];
for (i=0;i<emp.length;i++){
document.write(emp[i] + "<br/>");
}
```

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

JavaScript Array Methods

find()	It returns the value of the first element in the given array that satisfies the specified condition.
findIndex()	It returns the index value of the first element in the given array that satisfies the specified condition.
indexOf()	It searches the specified element in the given array and returns the index of the first match.
lastIndexOf()	It searches the specified element in the given array and returns the index of the last match.
pop()	It removes and returns the last element of an array.
push()	It adds one or more elements to the end of an array.
reverse()	It reverses the elements of given array.
shift()	It removes and returns the first element of an array.
sort()	It returns the element of the given array in a sorted order.

Questions:

1. What is array?
2. How to defined array?

Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	

Practical No-4.Develop javascript to implement functions

Function

JavaScript functions are used to perform operations. We can call JavaScript function many times to reuse the code.

Advantage of JavaScript function

There are mainly two advantages of JavaScript functions.

1. Code reusability: We can call a function several times so it save coding.
2. Less coding: It makes our program compact. We don't need to write many lines of code each time to perform a common task.

JavaScript Function Syntax

```
function function_Name([arg1, arg2, ...argN])
{
//code to be executed
}
```

JavaScript Functions can have 0 or more arguments.

Example

```
<html>
<body>
<script>
functionmsg()
{
alert("hello! this is message");
}
</script>
<input type="button" onclick="msg()" value="call function"/>
</body>
</html>
```

JavaScript Function Arguments

We can call function by passing arguments. Let's see the example of function that has one argument.

```
<html>
<body>
<script>
functiongetcube(number)
{
alert(number*number*number);
}
```

```
</script>
<form>
<input type="button" value="click" onclick="getcube(4)"/>
</form>
</body>
</html>
```

Function with Return Value

We can call function that returns a value and use it in our program. Let's see the example of function that returns value.

```
<html>
<body>
<script>
functiongetInfo(){
return"hello javatpoint! How r u?";  }
</script>
<script>
document.write(getInfo());
</script>
</body>
</html>
```

JavaScript Function Object

In JavaScript, the purpose of Function constructor is to create a new Function object. It executes the code globally. However, if we call the constructor directly, a function is created dynamically but in an unsecured way.

Syntax

new Function ([arg1[, arg2[, ...argn]],] functionBody)

Parameter-arg1, arg2, ... , argn - It represents the argument used by function.
functionBody - It represents the function definition.

```
<!DOCTYPE html>
<html>
<body>
<script>
var add=new Function("num1","num2","return num1+num2");
document.writeln(add(2,5));
</script>
</body>
</html>
```

Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	

Practical No-5.Develop javascript to implement Strings.

JavaScript String

The JavaScript string is an object that represents a sequence of characters.
There are 2 ways to create string in JavaScript

1. By string literal
2. By string object (using new keyword)

1) By string literal

The string literal is created using double quotes. The syntax of creating string using string literal is given below:
var stringname="string value";

Example:

```
<!DOCTYPE html>
<html>
<body>
<script>
varstr="This is string literal";
document.write(str);
</script>
</body>
</html>
```

2) By string object (using new keyword)

The syntax of creating string object using new keyword is given below:
var stringname=new String("string literal");
Here, new keyword is used to create instance of string.

Example

```
<!DOCTYPE html>
<html>
<body>
<script>
varstringname=new String("hello javascript string");
document.write(stringname);
</script>
</body>
</html>
```

JavaScript String Methods

charAt()	It provides the char value present at the specified index.
charCodeAt()	It provides the Unicode value of a character present at the specified index.
concat()	It provides a combination of two or more strings.
indexOf()	It provides the position of a char value present in the given string.

lastIndexOf() It provides the position of a char value present in the given string by searching a character from the last position.
search() It searches a specified regular expression in a given string and returns its position if a match occurs.
match() It searches a specified regular expression in a given string and returns that regular expression if a match occurs.
replace() It replaces a given string with the specified replacement.
substr() It is used to fetch the part of the given string on the basis of the specified starting position and length.
substring() It is used to fetch the part of the given string on the basis of the specified index.
toLowerCase() It converts the given string into lowercase letter.
toUpperCase() It converts the given string into uppercase letter.
toString() It provides a string representing the particular object.
valueOf() It provides the primitive value of string object.

Example

```
<!DOCTYPE html>
<html>
<body>
<script>
varstr="javascript";
document.write(str.charAt(2));
var s1="javascript ";
var s2="concat example";
var s3=s1.concat(s2);
document.write(s3);
var s1="javascript from javatpointindexof";
var n=s1.indexOf("from");
document.write(n);
var s1="javascript from javatpointindexof";
var n=s1.lastIndexOf("java");
document.write(n);
var s1="JavaScript toLowerCase Example";
var s2=s1.toLowerCase();
document.write(s2);
var s1="JavaScript toUpperCase Example";
var s2=s1.toUpperCase();
document.write(s2);
</script>
</body>
</html>
```

Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	

Practical No-6.Create web page using Form Elements

What are forms?

<form> is just another kind of XHTML/HTML tag. Forms are used to create (rather primitive) GUIs on Web pages. Usually the purpose is to ask the user for information. The information is then sent back to the server. A form is an area that can contain form elements

The syntax is:

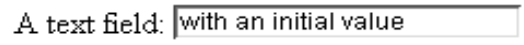
<form***parameters***>***...form elements...***</form>
Form elements include: buttons, checkboxes, text fields, radio buttons, drop-down menus, etc.
The arguments to form tell what to do with the user input
action="***url***" (required) :-Specifies where to send the data when the Submit button is clicked
method="get" (default):-Form data is sent as a URL with ?form_data info appended to the endCan be used *only* if data is all ASCII and not more than 100 characters
method="post" :-Form data is sent in the body of the URL request. Cannot be bookmarked by most browsers
target="***target***" :-Tells where to open the page sent as a result of the request.***target*****=** _blank means open in a new window. ***target*****=** _top means use the same window

The <input> tag

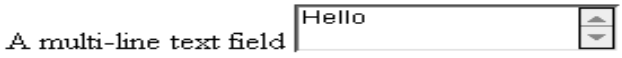
Most, but not all, form elements use the input tag, with a **type="..."** argument to tell which kind of element it istype can be text, checkbox, radio, password, hidden, submit, reset, button, file, or image
Other common input tag arguments include:
name: the name of the element
id: a unique identifier for the element
value: the "value" of the element; used in different ways for different values of type
readonly: the value cannot be changed
disabled: the user can't do anything with this element
Other arguments are defined for the input tag but have meaning only for certain values of type

Text input

A text field:
<input type="text" name="textfield" value="with an initial value" />



A multi-line text field
<textarea name="textarea" cols="24" rows="2">Hello</textarea>



A password field:
<input type="password" name="textfield3" value="secret" />

A password field: ••••••

A submit button: Submit

A reset button: Reset

A plain button: Push Me

Buttons

A submit button:send data
<input type="submit" name="Submit" value="Submit" />
A reset button:restore all form elements to their initial state
<input type="reset" name="Submit2" value="Reset" />
A plain button:take some action as specified by JavaScript
<input type="button" name="Submit3" value="Push Me" />

Radio buttons:
male
female

Radio buttons

Radio buttons:
<input type="radio" name="radiobutton" value="myValue1" />male
<input type="radio" name="radiobutton" value="myValue2" checked="checked" />female

If two or more radio buttons have the same name, the user can only select one of them at a time. This is how you make a radio button "group".
If you ask for the value of that name, you will get the value specified for the selected radio buttonas with checkboxes, radio buttons do not contain any text.

Labels

A label tag will bind the text to the control
<label><input type="radio" name="gender" value="m" />male</label>

Checkboxes

A checkbox: <input type="checkbox" name="checkbox" value="checkbox" checked="checked">
type: "checkbox"
name: used to reference this form element from JavaScript
value: value to be returned when element is checked

A checkbox: ☑

Drop-down menu or list

A menu or list:<select name="select">
<option value="red">red</option>
<option value="green">green</option>
<option value="BLUE">blue</option>
</select>

A menu or list: red

Additional arguments:
size: the number of items visible in the list (default is "1")
multiple

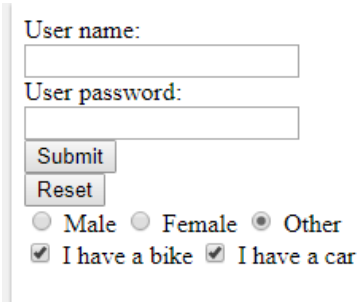
if set to "true" (or just about anything else), any number of items may be selected
if omitted, only one item may be selected
if set to "false", behavior depends on the particular browser

Additional input types:-

```
<input type="color">
<input type="date">
<input type="email">
<input type="file">
<input type="hidden">
<input type="image">
<input type="month">
<input type="number">
<input type="range">
<input type="search">
<input type="time">
<input type="url">
<input type="week">
```

Example

```
<html><body>
<form action="">
User name:<br>
<input type="text" name="userid"><br>
User password:<br>
<input type="password" name="psw"><br>
<input type="submit" value="Submit"><br>
<input type="reset"><br>
<input type="radio" name="gender" value="male" checked> Male
<input type="radio" name="gender" value="female"> Female
<input type="radio" name="gender" value="other">Other<br>
<input type="checkbox" name="vehicle1" value="Bike"> I have a bike
<input type="checkbox" name="vehicle2" value="Car"> I have a car <br>
</form>
</body></html>
```



Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	

Practical No-7: Create web page to implement Form Events. Part I

What is an Event?

JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page.

When the page loads, it is called an event. When the user clicks a button, that click too is an event. Other examples include events like pressing any key, closing a window, resizing a window, etc.

Developers can use these events to execute JavaScript coded responses, which cause buttons to close windows, messages to be displayed to users, data to be validated, and virtually any other type of response imaginable.

HTML allows event handler attributes, **with JavaScript code**, to be added to HTML elements.

With single quotes:

<*element event*='***some JavaScript***'>

With double quotes:

<*element event*="***some JavaScript***">

Event &Event handlers for Form Elements.

Event	Occurs when...	Event Handler
click	User clicks on form element or link	onClick
change	User changes value of text, text area, or select element	onChange
focus	User gives form element input focus	onFocus
blur	User removes input focus from form element	onBlur
mouseover	User moves mouse pointer over a link or anchor	onMouseOver
mouseout	User moves mouse pointer off of link or anchor	onMouseOut
select	User selects form element's input field	onSelect
submit	User submits a form	onSubmit
resize	User resizes the browser window	onResize
load	User loads the page in the Navigator	onLoad
unload	User exits the page	onUnload

Commonly Used Events:

Object	Event	User Action
Text field	onBlur	Tab away from window, frame or form element. Use onBlur when you tab out of a field.
	onFocus	Tab to a window, frame or form element.
	onSelect	Select text. If a form has multiple choices, you must use onSelect.
	onChange	Change text and click.
Button	onClick	Mouse click.
Checkbox	onClick	Mouse click.
Radio button	onClick	Mouse click.
Link	onMouseOver	Move mouse pointer over.
	onClick	Mouse click.
Window	onLoad	When the browser finishes loading a window or all frames in a window.
	onUnload	When you exit a window.

Examples

Click Events:-

1. **onclick Event**

```
<html>
<head>
<script type = "text/javascript">
functionsayHello()
        {
alert("Hello World")
        }
</script>
</head>
<body>
<form><input type = "button" onclick = "sayHello()" value = "Say Hello" />
</form>
</body>
</html>
```

2.ondblclick event

```
<html>
<head>
<script>
functionmyFunction() {
document.getElementById("demo").innerHTML = "Hello World";
}
</script>
</head>
<body>
<p ondblclick="myFunction()">
Doubleclick this paragraph to trigger a function.</p>
<p id="demo"></p>
</body>
</html>
```

Mouse Events:-

1.onmouseover&onmouseout event

```
<!DOCTYPE html>
<html>
<body>
<h1 onmouseover="style.color='red'" onmouseout="style.color='black'">Mouse over this text</h1>
</body>
</html>
```

2.**onmouseup&onmousedown event**

```
<html>
<head>
<script>
functionmyFunction(elmnt, clr)
{                   elmnt.style.color = clr;
}
</script>
</head>
<body>
<p onmousedown="myFunction(this,'red')" onmouseup="myFunction(this,'green')">
hi how r u?
</p>
</body>
</html>
```

Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	

Practical No-8: Create web page to implement Form Events. Part II

Load Events:-

1.onload event

```
<html>
<head>
<script>
functionmyFunction() {
alert("Page is loaded");
}
</script>
</head>
<body onload="myFunction()">
<h2>Hello World!</h2>
</body>
</html>
```

2.unload event

```
<html>
<head>
<script>
functionmyFunction()
 {
alert("Thank you for visiting My page!");
}
</script>
</head>
<body onunload="myFunction()">
</body>
</html>
```

Key Events

1.onkeypress event

```
<html>
<head>
<script>
functionmyFunction() {
alert("You pressed a key inside the input field");
}
</script>
</head>
<body>
<input type="text" onkeypress="myFunction()">
</body>
```

```
</html>
```

2.onkeyup event

```
<html>
<head>
<script>
functionmyFunction() {
var x = document.getElementById("fname");
x.value = x.value.toUpperCase();
}
</script>
</head>
<body>
Enter your name: <input type="text" id="fname" onkeyup="myFunction()">
</body>
</html>
```

3.onkeydown event

```
<html>
<head>
<script>
functionmyFunction() {
alert("You pressed a key inside the input field");
}
</script>
</head>
<body>
<input type="text" onkeydown="myFunction()">
</body>
</html>
```

Other Events

1.onchange event

```
<html>
<head>
<script>
functionmyFunction() {
var x = document.getElementById("fname");
x.value = x.value.toUpperCase();
}
</script>
</head>
<body>
Enter your name: <input type="text" id="fname" onchange="myFunction()">
</body>
```

```
</html>
```

2.onselect event

```
<html>
<head>
<script>
functionmyFunction()
{
document.write("selected some text");
}
</script>
</head>
<body>
Some text: <input type="text" value="Hello world!" onselect="myFunction()">
</body>
</html>
```

3.onfocus event

```
<html>
<head>
<script>
functionmyFunction(x)
{
x.style.background = "yellow";
}
</script>
</head>
<body>
Enter your name: <input type="text" onfocus="myFunction(this)">
</body>
</html>
```

4.onblur event

```
<html>
<head>
<script>
functionmyFunction()
{
var x = document.getElementById("fname");
x.value = x.value.toUpperCase();
}
</script>
</head>
<body>
```

```
Enter your name: <input type="text" id="fname" onblur="myFunction()">
</body>
</html>
```

5.onreset event

```
<html>
<head>
<script>
function message() {
alert("This alert box was triggered by the onreset event handler");
}
</script>
</head>
<body>
<form onreset="message()">
  Enter your name: <input type="text" size="20">
<input type="reset">
</form>
</body>
</html>
```

6.onsubmit event

```
<html>
<head>
<script>
functionconfirmInput()
{
fname = document.forms[0].fname.value;
alert("Hello " + fname + "! You will now be redirected to My Page");
}
</script>
</head>
<body>
<form onsubmit="confirmInput()" action="https://google.com/">
 Enter your name: <input id="fname" type="text" size="20">
<input type="submit">
</form>
</body>
</html>
```

Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	

Question:
Design a form using different types of events.

Practical No:-9 .Develop a webpage using intrinsic java functions

Intrinsic Functions

- Intrinsic functions means the built in functions that are provided by JavaScript.
- The JavaScript provides the intrinsic functions for Submit or Reset button. One can use these functionalities while submitting the form or resetting the form fields.
- The **submit()** method of the form object can be used to send the form to the server in exactly same way as if the user has pressed the Submit button.

JavaScript Example

```
<!DOCTYPE html>
<html>
<body>
    <form name="myform">
    Roll Number:<input type="text" name="roll"/>
    <br/> <br/>
    Name :<input type="text" name="name"/>
    <br/><br/>
    <img src ="submit.gif" onlclik="javascript:document.forms.myform.submit()"/>
    <br/><br/>
    </form>
</body>
</html>
```

Disabling Elements

- We can restrict some fields on the form by using disabled.
- If disabled property of particular form elements is set true then user can not edit that element. Similarly on setting disabled property to false we can edit the field.

For Example

```
<!DOCTYPE html>
<html>
<head>
<script type ="text/javascript">
Function EnableFunction()
{
document.forms.myform.name.disabled=false
}
Function DisableFunction()
{
document.forms.myform.name.disabled=true
```

```
}
</script>
</head>
<body>
<form  name=”myform”>
      Username:<input type=”text”  name=”name”/>
      <br/> <br/>
        <input type=”button”  value=”Disable Name Field” onclick=” DisableFunction()”/>
       <br/><br/>
              <input type=”button”  value=”Enable  Name Field” onclick=” EnableFunction()”/>
       </form>   </body>     </html>
```

Read-Only Elements

- Sometimes we need to set some value to a field which user should not change.to restrict user from changing the value of perticular field we make that element readonly by setting readonly=true.
- If the readonly attribute is set false ,then anyone ,including the user entering information into the form ,can change the value of the elemet.
- Following example illustrates the use of readonly element

JavaScript Example

```
<!DOCTYPE  html>
<html>
<head>
<script type =”text/javascript”>
Function ReadOnlyFunction()
{          document.forms.myform.name.readOnly=true
}
</script>
</head>
<body>
<form  name=”myform”>
      Username: <input type=”text”  name=”name”/>
      <br/> <br/>
     <input type=”button”  value=”Read-only Name Field” onclick=” ReadOnlyFunction())”/>

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

Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	

Practical No:-10.Develop a webpage for creating session and persistent cookies. Observe the effects with browser cookies settings.

What are Cookies?

A cookie is a piece of data that is stored on your computer to be accessed by your browser. You also might have enjoyed the benefits of cookies knowingly or unknowingly.Cookies are data, stored in small text files, on your computer.

How It Works ?

Your server sends some data to the visitor's browser in the form of a cookie. The browser may accept the cookie. If it does, it is stored as a plain text record on the visitor's hard drive. Now, when the visitor arrives at another page on your site, the browser sends the same cookie to the server for retrieval. Once retrieved, your server knows/remembers what was stored earlier.

Cookies are a plain text data record of 5 variable-length fields −

- **Expires** − The date the cookie will expire. If this is blank, the cookie will expire when the visitor quits the browser.
- **Domain** − The domain name of your site.
- **Path** − The path to the directory or web page that set the cookie. This may be blank if you want to retrieve the cookie from any directory or page.
- **Secure** − If this field contains the word "secure", then the cookie may only be retrieved with a secure server. If this field is blank, no such restriction exists.
- **Name=Value** − Cookies are set and retrieved in the form of key-value pairs

Create a Cookie with JavaScript

You can create cookies using document.cookie property.

```
document.cookie = "cookiename=cookievalue"
```

You can even add expiry date to your cookie so that the particular cookie will be removed from the computer on the specified date. The expiry date should be set in the UTC/GMT format. If you do not set the expiry date, the cookie will be removed when the user closes the browser.

```
document.cookie = "cookiename=cookievalue; expires= Thu, 21 Aug 2014 20:00:00 UTC"
```

Storing Cookies

The simplest way to create a cookie is to assign a string value to the document.cookie object, which looks like this.

```
document.cookie = "key1 = value1;key2 = value2;expires = date";
```

Note − Cookie values may not include semicolons, commas, or whitespace. For this reason, you may want to use the JavaScript escape() function to encode the value before storing it in the cookie. If you do this, you will also have to use the corresponding unescape() function when you read the cookie value.

Example

```
<html>
<head>
<script type = "text/javascript">
functionWriteCookie() {
if(document.myform.customer.value == "" )
          {
alert("Enter some value!");
return;
           }
cookievalue = escape(document.myform.customer.value) + ";";
document.cookie = "name=" + cookievalue;
document.write ("Setting Cookies : " + "name=" + cookievalue );
         }
</script>
</head>
<body>
<form name = "myform" >
      Enter name: <input type = "text" name = "customer"/>
<input type = "button" value = "Set Cookie" onclick = "WriteCookie();"/>
</form>
</body>
</html>
```

Read a Cookie with JavaScript

You can access the cookie like this which will return all the cookies saved for the current domain

```
var x =  document.cookie
```

Reading a cookie is just as simple as writing one, because the value of the document.cookie object is the cookie. So you can use this string whenever you want to access the cookie. The document.cookie string will keep a list of name=value pairs separated by semicolons, where name is the name of a cookie and value is its string value.You can use strings' split() function to break a string into key and values

Example

```
<html>
<head>
<script type = "text/javascript">
functionReadCookie() {
varallcookies = document.cookie;
```

```
document.write ("All Cookies : " + allcookies );
        // Get all the cookies pairs in an array
cookiearray = allcookies.split(';');
        // Now take key value pair out of this array
for(var i=0; i<cookiearray.length; i++) {
name = cookiearray[i].split('=')[0];
value = cookiearray[i].split('=')[1];
document.write ("Key is : " + name + " and Value is : " + value);
          }
        }
</script>
</head>
<body>
<form name = "myform" action = "">
<p> click the following button and see the result:</p>
<input type = "button" value = "Get Cookie" onclick = "ReadCookie()"/>
</form>
</body>
</html>
```

Setting Cookies Expiry Date

You can extend the life of a cookie beyond the current browser session by setting an expiration date and saving the expiry date within the cookie. This can be done by setting the 'expires' attribute to a date and time.

Example

```
<html>
<head>
<script type = "text/javascript">
functionWriteCookie() {
var now = new Date();
now.setMonth(now.getMonth() + 1 );
cookievalue = escape(document.myform.customer.value) + ";"

document.cookie = "name=" + cookievalue;
document.cookie = "expires=" + now.toUTCString() + ";"
document.write ("Setting Cookies : " + "name=" + cookievalue );
        }
</script>
</head>
<body>
<form name = "myform" action = "">
      Enter name: <input type = "text" name = "customer"/>
<input type = "button" value = "Set Cookie" onclick = "WriteCookie()"/>
</form>
</body>
```

```
</html>
```

Delete a Cookie with JavaScript

To delete a cookie, you just need to set the value of the cookie to empty and set the value of expires to a passed date.

```
document.cookie = "cookiename= ; expires = Thu, 01 Jan 1970 00:00:00 GMT"
```

Example

```
<html>
<head>
<script type = "text/javascript">
functionWriteCookie() {
var now = new Date();
now.setMonth(now.getMonth() - 1 );
cookievalue = escape(document.myform.customer.value) + ";"

document.cookie = "name=" + cookievalue;
document.cookie = "expires=" + now.toUTCString() + ";"
document.write("Setting Cookies : " + "name=" + cookievalue );
        }
</script>
</head>
<body>
<form name = "myform" action = "">
      Enter name: <input type = "text" name = "customer"/>
<input type = "button" value = "Set Cookie" onclick = "WriteCookie()"/>
</form>
</body>
</html>
```

Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	

Practical No.11.Develop a webpage for placing the window on the screen and working with child window.

Window Object

The window object represents an open window in a browser. If a document contain frames (< iframe > tags), the browser creates one window object for the HTML document, and one additional window object for each frame.

Window open() Method

The open() method opens a new browser window, or a new tab, depending on your browser settings and the parameter values.

Syntax

```
window.open(URL, name, specs, replace)
```

Parameter Description

URL Optional. Specifies the URL of the page to open. If no URL is specified, a new window/tab with about:blank is opened

name Optional. Specifies the target attribute or the name of the window. The following values are supported:

- _blank - URL is loaded into a new window, or tab. This is default
- _parent - URL is loaded into the parent frame
- _self - URL replaces the current page
- _top - URL replaces any framesets that may be loaded
- name - The name of the window

specs Optional. A comma-separated list of items, no whitespaces. The following values are supported:

channelmode=yes|no|1|0	Whether or not to display the window in theater mode. Default is no. IE only
directories=yes|no|1|0	Obsolete. Whether or not to add directory buttons. Default is yes. IE only
fullscreen=yes|no|1|0	Whether or not to display the browser in full-screen mode. Default is no. A window in full-screen mode must also be in theater mode. IE only
height=pixels	The height of the window. Min. value is 100
left=pixels	The left position of the window. Negative values not allowed
location=yes|no|1|0	Whether or not to display the address field. Opera only
menubar=yes|no|1|0	Whether or not to display the menu bar
resizable=yes|no|1|0	Whether or not the window is resizable. IE only
scrollbars=yes|no|1|0	Whether or not to display scroll bars. IE, Firefox & Opera only
status=yes|no|1|0	Whether or not to add a status bar

titlebar=yes|no|1|0	Whether or not to display the title bar. Ignored unless the calling application is an HTML Application or a trusted dialog box
toolbar=yes|no|1|0	Whether or not to display the browser toolbar. IE and Firefox only
top=pixels	The top position of the window. Negative values not allowed
width=pixels	The width of the window. Min. value is 100

replace Optional. Specifies whether the URL creates a new entry or replaces the current entry in the history list. The following values are supported:
true - URL replaces the current document in the history list
false - URL creates a new entry in the history list

```
<!DOCTYPE html>
<html>
<body>
<p>Click the button to open an about:blank page in a new browser window that is 200px wide and 100px tall.</p>
<button onclick="myFunction()">Try it</button>
<script>
function myFunction() {
  var myWindow = window.open("", "", "width=200,height=100");
}
</script>
</body>
</html>
```

Window.close()

This method is used to close the window which are opened by *window.open()* method.

Syntax

window.close()

Window print() Method

The print() method prints the contents of the current window.The print() method opens the Print Dialog Box, which lets the user to select preferred printing options.

window.print();

- **The resizeBy()** method resizes a window by the specified amount, relative to its current size.

Syntax:

resizeBy(width, height)

- **The moveBy()** method moves a window a specified number of pixels relative to its current coordinates.

Syntax:

window.moveBy(x, y)

- **The resizeTo()** method resizes a window to the specified width and height.

Syntax:

window.resizeTo(width, height)

- **The scrollBy()** method scrolls the document by the specified number of pixels.

Syntax
window.scrollBy(xnum, ynum)

- **The setInterval() method** calls a function or evaluates an expression at specified intervals (in milliseconds).The setInterval() method will continue calling the function until clearInterval() is called, or the window is closed.The ID value returned by setInterval() is used as the parameter for the clearInterval() method.

Tip: 1000 ms = 1 second.
Tip: To execute a function only once, after a specified number of milliseconds, use the setTimeout() method.

Syntax:
setInterval(function, milliseconds, param1, param2, ...)

Parameter	Description
function	Required. The function that will be executed
milliseconds	Required. The intervals (in milliseconds) on how often to execute the code. If the value is less than 10, the value 10 is used
param1, param2, ...	Optional. Additional parameters to pass to the function

- **The setTimeout() method** calls a function or evaluates an expression after a specified number of milliseconds.

Syntax:
*setTimeout(*function, milliseconds, param1, param2, ...*)*
Parameter Values

Parameter	Description
function	Required. The function that will be executed
milliseconds	Optional. The number of milliseconds to wait before executing the code. If omitted, the value 0 is used
param1, param2, ...	Optional. Additional parameters to pass to the function

Example

```
<html>
<body>
<p>Click the button to open a new window and close the window after three seconds (3000 milliseconds)</p>
<button onclick="openWin()">Open "myWindow"</button>
<script>
function openWin() {
  var myWindow = window.open("", "myWindow", "width=200, height=100");
  myWindow.document.write("<p>This is 'myWindow'</p>");
  setTimeout(function(){ myWindow.close() }, 3000);
}
</script>
</body>
</html>
```

Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	

Practical No.12. Develop a web page for validation of form fields using regular expressions.

JavaScript Regular Expression

A regular expression is a sequence of characters that forms a search pattern.
The search pattern can be used for text search and text replaces operations.

What Is a Regular Expression?

A regular expression is a sequence of characters that forms a search pattern.
When you search for data in a text, you can use this search pattern to describe what you are searching for.
A regular expression can be a single character, or a more complicated pattern.
Regular expressions can be used to perform all types of text search and text replace operations.

Syntax

/pattern/modifiers;

Example
var patt = /w3schools/i;
Example explained:
/w3schools/i is a regular expression.
w3schools is a pattern (to be used in a search).
i is a modifier (modifies the search to be case-insensitive).

Using String Methods

In JavaScript, regular expressions are often used with the two string methods: search() and replace().

The search() method :

uses an expression to search for a match, and returns the position of the match.

The replace() method

returns a modified string where the pattern is replaced.

Using String search() With a String

The search() method searches a string for a specified value and returns the position of the match:

Example
Use a string to do a search for "W3schools" in a string:
var str = "Visit W3Schools!";
var n = str.search("W3Schools");

Using String search() With a Regular Expression

Example

Use a regular expression to do a case-insensitive search for "w3schools" in a string:
var str = "Visit W3Schools";
var n = str.search(/w3schools/i);
The result in n will be:
6

Using String replace() With a String
The replace() method replaces a specified value with another value in a string:
var str = "Visit Microsoft!";
var res = str.replace("Microsoft", "W3Schools");

Use String replace() With a Regular Expression

Example
Use a case insensitive regular expression to replace Microsoft with W3Schools in a string:
var str = "Visit Microsoft!";
var res = str.replace(/microsoft/i, "W3Schools");
The result in res will be:
Visit W3Schools!

Regular Expression Modifiers

Modifiers can be used to perform case-insensitive more global searches:

Modifier	Description
i	Perform case-insensitive matching
g	Perform a global match (find all matches rather than stopping after the first match)
m	Perform multiline matching

Regular Expression Patterns

Brackets are used to find a range of characters:

Expression	Description
[abc]	Find any of the characters between the brackets
[^abc]	Find any character NOT between the brackets
[0-9]	Find any of the digits between the brackets
[^0-9]	Find any character NOT between the brackets (any non-digit)
(x|y)	Find any of the alternatives separated with |

Meta characters are characters with a special meaning:

.	Find a single character, except newline or line terminator
\w	Find a word character

\W	Find a non-word character
\d	Find a digit
\D	Find a non-digit character
\s	Find a whitespace character
\S	Find a non-whitespace character
\b	Find a match at the beginning/end of a word, beginning like this: \bHI, end like this: HI\b
\B	Find a match, but not at the beginning/end of a word
\0	Find a NUL character
\n	Find a new line character
\f	Find a form feed character
\r	Find a carriage return character
\t	Find a tab character
\v	Find a vertical tab character
\xxx	Find the character specified by an octal number xxx
\xdd	Find the character specified by a hexadecimal number dd
\udddd	Find the Unicode character specified by a hexadecimal number dddd

Quantifiers

Quantifier	**Description**
n+	Matches any string that contains at least one n
n*	Matches any string that contains zero or more occurrences of n
n?	Matches any string that contains zero or one occurrences of n
n{X}	Matches any string that contains a sequence of X n's
n{X,Y}	Matches any string that contains a sequence of X to Y n's
n{X,}	Matches any string that contains a sequence of at least X n's
n$	Matches any string with n at the end of it
^n	Matches any string with n at the beginning of it
?=n	Matches any string that is followed by a specific string n
?!n	Matches any string that is not followed by a specific string n

RegExp Object Properties

Property	**Description**
constructor	Returns the function that created the RegExp object's prototype
global	Checks whether the "g" modifier is set
ignoreCase	Checks whether the "i" modifier is set
lastIndex	Specifies the index at which to start the next match
multiline	Checks whether the "m" modifier is set
source	Returns the text of the RegExp pattern

RegExp Object Methods

Method	Description

compile()	Deprecated in version 1.5. Compiles a regular expression
exec()	Tests for a match in a string. Returns the first match
test()	Tests for a match in a string. Returns true or false
toString()	Returns the string value of the regular expression

Using test()

The following example searches a string for the character "e":

Example

```
var patt = /e/;
patt.test("The best things in life are free!");
```

Since there is an "e" in the string, the output of the code above will be:

true

You don't have to put the regular expression in a variable first. The two lines above can be shortened to one:

```
/e/.test("The best things in life are free!");
```

Using exec()

The exec() method is a RegExp expression method.

It searches a string for a specified pattern, and returns the found text as an object.

If no match is found, it returns an empty (null) object.

The following example searches a string for the character "e":

Example 1

```
/e/.exec("The best things in life are free!");
```

1. Develop a web page for validation of form fields using regular expressions.

Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	

Practical.No.13.Create web page with Rollovers effect.

Rollover means a webpage changes when the user moves his or her mouse over an object on the page. It is often used in advertising. There are two ways to create rollover, using plain HTML or using a mixture of JavaScript and HTML. We will demonstrate the creation of rollovers using both methods.

Creating Rollovers using HTML

The keyword that is used to create rollover is the <onmousover> event. For example, we want to create a rollover text that appears in a text area. The text *"What is rollover?"* appears when the user place his or her mouse over the text area and the rollover text changes to *"Rollover means a webpage changes when the user moves his or her mouse over an object on the page"* when the user moves his or her mouse away from the text area.

```
<HTML>
<head></head>
<Body>
<textarea rows=”2” cols=”50” name=”rollovertext” onmouseover=”this.value=’What is rollover?'”
onmouseout=”this.value=’Rollover means a webpage changes when the user moves his or her mouse
over an object on the page'”></textarea>
</body>
</html>
```

We create a rollover effect that can change the color of its text using the style attribute.

```
<p
onmouseover=”this.style.color=’red'”
onmouseout=”this.style.color=’blue'”>
Move the mouse over this text to change its color to red. Move the mouse away to
change the text color to blue.
</p>
```

This example shows how to create rollover effect that involves text and images. When the user places his or her mouse pointer over a book title, the corresponding book image appears.

```
<html>
<head>
<title>Rollover Effect</title>
</head>
<body>
<table>
<tbody>
<trvalign=”top”>
<td width=”50”>
<a><imgsrc=”vb2010book.jpg” name=”book”></a>
</td>
<td><img height=”1” width=”10”></td>
```

```
<td><a onmouseover=”document.book.src=’vb2010book.jpg'”><b>Visual Basic 2010 Made
Easy</b></a>
<br>
<a onmouseover=”document.book.src=’vb2008book.jpg'”><b>Visual Basic 2008 Made Easy</b></a>
<br>
<a onmouseover=”document.book.src=’vb6book.jpg'”><b>Visual Basic 6 Made Easy</b></a>
<br>
</td>
</tr>
</tbody>
</table>
</body>
</html>
```

Creating Rollovers Using JavaScript

Though HTML can be used to create rollovers, it can only performs simple actions. If you wish to create more powerful rollovers, you need to use JavaScript. To create rollovers in JavaScript, we need to create a JavaScript function.

In this example, we have created an array *MyBooks* to store the images of three book covers. Next, we create a *ShowCover(book)* to display the book cover images on the page. Finally, we call the *ShowCover* function using the *onmouseover* event.

```
<!DOCTYPE html PUBLIC “-//W3C//DTD XHTML 1.0 Transitional//EN”>
<html xmlns=”http://www.w3.org/1999/xhtml”>
<head>
<script language=”Javascript”>
MyBooks=new Array(‘vb2010book.jpg’,’vb2008book.jpg’,’vb6book.jpg’)
book=0
function ShowCover(book){document.DisplayBook.src=MyBooks[book]
}</script></head>
<body>
<body>
<P align=”center”><imgsrc=”vb2010book.jpg” name=”DisplayBook”/><p>
<center>
<table border=0>
<tr>
<td align=center><a onmouseover=”ShowCover(0)”><b>Visual Basic 2010 Made Easy</b></a><br>
<a onmouseover=”ShowCover(1)”><b>Visual Basic 2008 Made Easy</b></a><br>
<a onmouseover=”ShowCover(2)”><b>Visual Basic 6 Made Easy</b></a><br>
</td>
</tr>
</table>
</body>
</html>
```

Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	

Practical.No.14. Develop a webpage for implementing Menus

The <select> element is used to create a drop-down list. The <option> tags inside the <select> element define the available options in the list.

Example

```
<html>
<body>
<select>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="opel">Opel</option>
  <option value="audi">Audi</option>
</select>
</body>
</html>
```

Dynamically Changing menu

To create an interdependent select list, where selecting the options of one select element changes the options of another with corresponding content.

Example

```
<html>
   <head>
   <script language="javascript" type="text/javascript">
   function dynamicdropdown(listindex)
   {
      switch (listindex)
      {
      case "manual" :
         document.getElementById("status").options[0]=new Option("Select status","");
         document.getElementById("status").options[1]=new Option("OPEN","open");
         document.getElementById("status").options[2]=new Option("DELIVERED","delivered");
         break;
      case "online" :
         document.getElementById("status").options[0]=new Option("Select status","");
         document.getElementById("status").options[1]=new Option("OPEN","open");
         document.getElementById("status").options[2]=new Option("DELIVERED","delivered");
         document.getElementById("status").options[3]=new Option("SHIPPED","shipped");
         break;
      }
      return true;
   }
   </script>
```

```
   </head>
   <title>Dynamic Drop Down List</title>
   <body>
   <div class="category_div" id="category_div">Source:
      <select id="source" name="source" onchange="javascript:
dynamicdropdown(this.options[this.selectedIndex].value);">
      <option value="">Select source</option>
      <option value="manual">MANUAL</option>
      <option value="online">ONLINE</option>
      </select>
   </div>
   <div class="sub_category_div" id="sub_category_div">Status:
      <script type="text/javascript" language="JavaScript">
      document.write('<select name="status" id="status"><option value="">Select
status</option></select>')
      </script>

      <select id="status" name="status">
         <option value="open">OPEN</option>
         <option value="delivered">DELIVERED</option>
      </select>

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

Q. Develop a web page which validating menu Selection.

Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	

Practical.No.15. Develop a webpage for implementing Status bars and web page protection.

JavaScript gives you the ability to modify the status bar. For example it can be useful to display information about a link, when the user moves his mouse over it or you can display a small amount of information about the page the user is on in the status bar. You can also trick people into clicking a link, so be careful how you use it. If you play too many tricks on your visitors, they might not come back.

Status Bar Example:

```
<html>

<head>

<title>JavaScript Status Bar</title></head>

<body onLoad="window.status='Welcome!';return true">

</body>

</html>
```

onLoad tells the browser that as soon as your page finished loading, it will display in your current window's status bar (window.status) the message "Welcome!". The return true is necessary because without, it won't work.

Status Bar Example:

```
<html>
<head>
<title>JavaScript Status Bar</title></head>
<body>
  <a href="http://www.htmlcenter.com"
   onMouseOver="window.status='HTMLcenter';return true"
   onMouseOut="window.status='';return true">
  HTMLcenter
  </a>
</body>
</html>
```

Our second script listening shows how to modify the status bar using onMouseOver and onMouseOut with links. When the user moves his mouse over the link, it will display "HTMLcenter" in the status bar. When he moves his mouse away from the link the status bar will display nothing.

You could also have another message displayed in the status bar, when the user moves his mouse cursor away from the link. You have to change the onMouseOut statement in the link to for example: onMouseOut="window.status='You moved your cursor away from the link.';return true".

Moving the message along the status bar

```
<html>
<head>
<title>Scrolling Text</title>
<script language="JavaScript">
var scrollPos = 0
var maxScroll = 100
var blanks = ""

function scrollText(text, milliseconds) {
  window.setInterval("displayText('"+text+"')", milliseconds)
}
function displayText(text) {
    window.defaultStatus = blanks + text
    ++scrollPos
    blanks += " "
    if(scrollPos > maxScroll) {
      scrollPos = 0
      blanks = ""
    }
}
</script>
</head>
<body onload="scrollText('Watch this text scroll!!!', 300)">
<p>Watch the text scroll at the bottom of this window!</p>
</body>
</html>
```

Protection web page

There are so many ways for users to get around this method of protection that it shouldn't even really be considered a method of protecting your data. Disable JavaScript. For this to work, JavaScript must be enabled on the browser. View the source code and locate the image or text they want to copy in the source code. Drag the image from the browser and drop it into the desktop, file manager, or another open program. Disable the ability for user to highlight text, copy, and paste it elsewhere.

Example

```
<html>
<head>
<script language="JavaScript">
   function function2() {
      alert("This image is copyrighted")
   }
</script>
</head>
```

```
<body oncontextmenu="function2()">
   <p>Right click in the image.</p>
   <img oncontextmenu="function2()"
       src="http://www.java2s.com/style/logo.png"
       alt="www.java2s.com"
       width="99"
       height="76">
</body>
</html>
```

If you want to disable the context menu, add the following code to the <body>:
oncontextmenu="function2(); return false;"

Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	

Practical No: 16 Develop a web page for implementing slideshow, banner.

Displaying banners ads is a common practice for showing advertisements on web pages to the visitors. Banners ads are normally created using standard graphic tools such as Photoshop, Paintbrush Pro, and other software. Banner ads can be static or animated. Animated images are animated GIF files or flash movies. Flash movies are created using Macromedia Flash and the browsers must have installed flash plugin to view the movies. On the other hand, you can create some animated effect using JavaScript, like rotating static banner ads at a certain time interval.

Creating Rotating Banner Ads

Rotating banners ads comprises several banner images that constantly rotate on a webpage at a fix time interval. You can create these banner images using standard graphics tools.

```
<html>
<head>
<script language="Javascript">MyBanners=new
Array('banner1.jpg','banner2.jpg','banner3.jpg','banner4.jpg')
banner=0
function ShowBanners()
{ if (document.images)
{ banner++
if (banner==MyBanners.length) {
banner=0}
document.ChangeBanner.src=MyBanners[banner]
setTimeout("ShowBanners()",5000)
}
}
</script>
<body onload="ShowBanners()">
<center>
<img src="banner1.jpg" width="900" height="120" name="ChangeBanner"/>
</center>
</body>
</html>
```

Creating Rotating Banner Ads with URL Links

Creating rotating banner images will provide the visitor to your webpage with some basic information. However, if you want the visitor to get more information by clicking on the banner images, you need to create rotating banner ads that contain URL links.

```
<html>
<head>
<script language="Javascript">MyBanners=new
Array('banner1.jpg','banner2.jpg','banner3.jpg','banner4.jpg')
```

```
MyBannerLinks=new
Array('http://www.vbtutor.net/','http://www.excelvbatutor.com/','http://onlinebizguide4you.com/','htt
p://javascript-tutor.net/')
banner=0
function ShowLinks(){
document.location.href="http://www."+MyBannerLinks[banner]
}function ShowBanners()
{ if (document.images)
{ banner++
if (banner==MyBanners.length) {
banner=0}
document.ChangeBanner.src=MyBanners[banner]
setTimeout("ShowBanners()",5000)
}
}
</script>
<body onload="ShowBanners()">
<center>
<a href="javascript: ShowLinks()">
<img src="banner1.jpg" width="900" height="120" name="ChangeBanner"/></a>
</center>
</body>
</html>
```

Slide Show

The JavaScript code for the slideshow is almost similar to the JavaScript code of the rotating banners but it gives control to the user to choose the banner ads he or she wants to see by clicking on the forward and backward buttons.

To create the JavaScript slideshow, first of all, you need to create a few banner images using some graphics tools, or you can snap some photos with your digital camera or your smartphone.

```
<html >
<head>
<script language=”Javascript”>
MySlides=new Array(‘banner1.jpg’,’banner2.jpg’,’banner3.jpg’,’banner4.jpg’)
Slide=0
function ShowSlides(SlideNumber){

{ Slide=Slide+SlideNumber
if (Slide>MySlides.length-1){
Slide=0
}
if (Slide<0) {
Slide=MySlides.length-1
}
document.DisplaySlide.src=MySlides[Slide]
}
}
```

```
</script>
</head>
<body>
<P align=”center”><img src=”banner1.jpg” name=”DisplaySlide” width=”900” height=”120” /><p>
<center>
<table border=0>
<tr>
<td align=center>
<input type=”button” value=”Back” onclick=”ShowSlides(-1)”>
<input type=”button” value=”Forward” onclick=”ShowSlides(1)”>
</td>
</tr>
</table>
</center>
</body>
</html>
```

Marks Obtained			Dated Signed of teacher
Process Related(35)	Product Related(15)	Total(50)	