

Practical No-6.Create web page using Form Elements

Relevant Program Outcomes (POs)

- **Basic Knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problem.
- **Discipline Knowledge:** Apply computer programming knowledge to solve the computer group related problems.
- **Experiments and practices:** Plan to perform experiments and practices to use the result to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools within understanding of the limitations.
- **Individual and Teamwork:** Function effectively as a leader and the team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

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 end. Can 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 is. type 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

Maharashtra State board of Education

Client Side Scripting Languages (22519)

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 text field [with an initial value]

A multi-line text field

```
<textarea name="textarea" cols="24" rows="2">Hello</textarea>
```

A multi-line text field



A password field:

```
<input type="password" name="textfield3" value="secret" />
```

Client Side Scripting Languages (HTML)

A password field:

Buttons

A submit button send data

```
<input type="submit" name="submit" value="Submit" />
```

A reset button restores 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="pushMe" value="Push Me" />
```

Radio buttons

Radio buttons:

male
 female

```
<input type="radio" name="radiobutton" value="myValue1" />male<br>
```

```
<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 buttons 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:

name: used to reference this form element from JavaScript

Drop-down menu or list

A menu or list:

A menu or list:

Additional arguments:

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

A submit button [Submit]

A reset button [Reset]

A plain button [Push Me]

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>
```

coclusion:

In this practical we learn to create form using different form controls.

Client Side Scripting Languages (22519)

Questions:

1. Write HTML code to create Facebook Sign Up Form.
2. What are different attributes of form element?
3. Explain use of <optgroup> tag.

- 3) →
- 1) acceptcharset
 - 2) action
 - 3) autocomplete
 - 4) enctype
 - 5) method
 - 6) name
 - 7) novalidate
 - 8) referrer
 - 9) target

→ The <optgroup> tag is used to group related options in a <select> element (dropdown list). If you have a long list of options, groups of related options are easier to handle for a user.

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

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

Relevant Program Outcomes (POs)

- **Basic Knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problem.
- **Discipline Knowledge:** Apply computer programming knowledge to solve the computer group related problems.
- **Experiments and practices:** Plan to perform experiments and practices to use the result to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools within understanding of the limitations.
- **Individual and Teamwork:** Function effectively as a leader and the team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

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

Client Side Scripting Languages (22519)

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

2. ondblclick event

```
<html>
<head>
<script>
function myFunction() {
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>
function myFunction(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>
```

CONCLUSION:

In this practical we learn different events. we learn to implement form events.

Questions:

1. Write a note on event handler.
2. With the help of example explain how to change the option list dynamically.
3. With the help of example how to change(toggle) the labels dynamically.

1]

Ans :

- i] An event handler is a callback subroutine that handles inputs received in a program.
- ii] OnBlur - user removes input focus from form element.
- iii] onClick - user click on form element.
- iv] onChange - user changes values.
- v] onFocus - user gives form element input focus.
- vi] onLoad - user loads the page.
- vii] onselect - user select form element input field.

This are some event handler in javascript.

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

2] With the help of example explain how to change the option list dynamically

Ans. <!DOCTYPE html>

```

<html lang = "en">
<head>
    <title> Dynamically add remove options </title>
</head>
<body>
    <select id = "dynamic-select">
        <option value = "1"> one </option>
        <option value = "2"> two </option>
        <option value = "3"> three </option>
    </select>
    <button onclick = "getOption()"> get item </button>
    <button onclick = "addOption()"> add item </button>
    <button onclick = "removeOptions()"> remove item </button>
    <button onclick = "removeAllOption ()"> remove all </button>
<script>
    function getOption()
    {
        var select = document.getElementById("dynamic-select");
        if (select.options.length > 0)
    }

```

var option = select.options [select.selectedIndex];

alert ("Text : " + option.text + " In value " + option.value);

```
        } close
    }
}

function addoption () {
    var select = document.getElementById
    ('dynamic-select');
    select.options [select.options.length] = new
    option ('New Element', 'false', false);
}

function removeoption () {
    var select = document.getElementById
    ('dynamic-select');
    if (select.selectedIndex) {
        select.options [select.selectedIndex] = null;
    }
}

function removeAlloptions () {
    var select = document.getElementById
    ('dynamic-select');
    select.options.length = 0;
}

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

Q3

With the help of example how to change the labels dynamically



```
<html>
  <script>
    function Test()
    {
      var x = document.getElementById("Test");
      if ((x.innerHTML).includes("*"));
      {
        x.innerHTML = x.innerHTML.slice(8);
      }
    }
  </script>
  <body>
    <label id="Test" for="new">My
first Javascript* </label>
    <button type="button"
      onclick ="Test()" id="new"
      name="new" > New
    </button>
    <p id="demo" </p>
  </body>
</html>
```

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

Relevant Program Outcomes (POs)

- **Basic Knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problem.
- **Discipline Knowledge:** Apply computer programming knowledge to solve the computer group related problems.
- **Experiments and practices:** Plan to perform experiments and practices to use the result to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools within understanding of the limitations.
- **Individual and Teamwork:** Function effectively as a leader and the team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

Load Events:-

1.onload event

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

2.upload event

```
<html>
<head>
<script>
function myFunction()
{
    alert("Thank you for visiting My page!");
}
</script>
</head>
<body onunload="myFunction()">
Maharashtra State board of Education
```

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

Key Events

1.onkeypress event

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

Client Side Scripting Languages (22519)

```
</html>
```

2.onkeyup event

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

Client Side Scripting Languages (22519)

```
</html>
```

2.onselect event

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

Client Side Scripting Languages (22519)

```
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>
function confirmInput()
{
  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>
```

CONCLUSION:

Hence from this practical we studied form events captured & event bubbling we learned what are mouse over & mouse out events using these events we created web pages.

Question:

1. Explain event capturing and event bubbling.
2. Illustrate the use of mouseover and mouseout events.
3. Explain with example how to read key strokes.

1

Ans.: Event bubbling and capturing are two ways to event propagation in HTML.

1] Bubbling : with Bubbling, the event is captured & handled by the innermost element, and then propagated to outer elements.

2] Capturing : with capturing, the event is first captured by outermost element & propagated to inner elements. Netscape browser has first introduced the concept of event capturing.

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

2]

Ans. 1] use of mouseover :-

The mouseover event is fired at an element when a pointing device such as a mouse or trackpad is used to move the cursor onto the element or one of child elements.

2) Use of mouseout :-

The mouseout event is fired at an element when a pointing device usually a mouse is used to move the cursor so that it is no longer contained within the elements or none of children.

3]

Ans. • keydown Event

```
<html>
<body>
<p> when any key is pressed it display
unicode </p>
<input type = "text" onkeydown = "myfunction">
<script>
function myfunction()
{
    var keycode (which is event)?
    event
    which key code
    alert ("The unicode key is"+key
    code)
}
```

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



Scanned with OKEN Scanner

<html>

• onkeypress Event

<html> <body> <input type="text" onkeypress="myfunction()"/>

<body> </body>

<p> A function is triggered when the user
is pressing a key in the input field</p>

<input type="text" onkeypress="myfunction()"/>

<script>

function myfunction()

{

 alert ("you pressed a key inside
 input field");

}

</script>

</body>

</html>

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

Relevant Program Outcomes (POs)

- **Basic Knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problem.
- **Discipline Knowledge:** Apply computer programming knowledge to solve the computer group related problems.
- **Experiments and practices:** Plan to perform experiments and practices to use the result to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools within understanding of the limitations.
- **Individual and Teamwork:** Function effectively as a leader and the team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

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

Client Side Scripting Languages (22519)

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
}
```

CONCLUSION:

Hence, in this practical we learned to develop a webpage using intrinsic java functions in javascript.

Questions:

1. Develop a javascript code to use image as button to submit and reset form.
2. Explain how to make elements disabled and readonly.

```

→ ] <!DOCTYPE html>
      <html>
        <head>
          <title> Intrinsic Functions </title>
        </head>
        <body>
          <form action = "destination.html"
                method = "GET">
            <fieldset>
              <legend> Image as Reset and Submit
                Button </legend>
              <input name = "Reset" type = "Image"
                    src = "data:image/png;">
              <input name = "Submit" type = "Image"
                    src = "https://www.pngmart.com/
                Button.png File.png">
            </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.**

Relevant Program Outcomes (POs)

- **Basic Knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problem.
- **Discipline Knowledge:** Apply computer programming knowledge to solve the computer group related problems.
- **Experiments and practices:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools within understanding of the limitations.
- **Individual and Teamwork:** Function effectively as a leader and the team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

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

Maharashtra State Board of Education

Client Side Scripting Languages (22519)

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";
```

Client Side Scripting Languages (22519)

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>
```

CONCLUSION:

In this practical we learn to create session cookie, persistent cookie.

QUESTIONS:

1. Explain persistent and non-persistent cookies.
2. Develop a web page to create a session cookie.
3. Explain different components of browser.

1] Ans.:

1] Persistent cookies are permanent cookies. They are stored as a text file in the hard disk on computer.

2] Non-Persistent cookies are in-memory cookies which are added to the memory of browser but not recorded in any file & does not store in client hard disk.

2]

Ans.:

```

<html>
<head>
<script>
    function createcookie()
    {
        var x = document.getElementById('myname')
        .value
        document.cookie = "name=" + x + ";"
    }
</script>
</head>
<body>
    Enter your Name
    <input type="text" id="myname"
    onchange = "createcookie()" />
</body>
</html>

```

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

3]

Ans

Components of web browser.

1) User Interface : It is an environment allowing users to use certain features.

2) Browser Engine : The bridge connects the interface & the engine.

3) Networking : The protocol provides an URL & manages all sorts of safety, privacy & communication.

4) Data storage : The cookie store information as the data store is an uniform layer that the browsers use.

5) Javascript Interpreter : It allows conversion of javascript code in a document & executes it.

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

Relevant Program Outcomes (POs)

- **Basic Knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problem.
- **Discipline Knowledge:** Apply computer programming knowledge to solve the computer group related problems.
- **Experiments and practices:** Plan to perform experiments and practices to use the result to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools within understanding of the limitations.
- **Individual and Teamwork:** Function effectively as a leader and the team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

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:
Maharashtra State board of Education

Client Side Scripting Languages (22519)

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

Client Side Scripting Languages (22519)

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:

Client Side Scripting Languages (22519)

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>
```

CONCLUSION:

In this practical we learn method of window object & use them in program.

Questions:

1. Develop a program to access height and width properties of browser window.
2. Explain different methods to use scroll a window.
3. Explain timer & interval related methods.

Ans :

```

<!DOCTYPE html>
<html>
<body>
<p> Click the button to display this frame
height & width </p>
<button onclick = "myfunction()"> Try it </button>
<p id = "demo"></p>
<script>

    function myfunction()
    {
        var w = window.innerWidth;
        var h = window.innerHeight;
        document.getElementById("demo").innerHTML = "Width:" + w + " Height:" + h;
    }

```

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

Destination.html

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

This is destination file

```
</body>
```

```
</html>
```

2]

→ 1) Disabling Elements :-

An element can be disabled in HTML by setting disable property to true & enabled again by setting disabled = false. We can disable some elements to restrict data entry into those elements. Such disabled elements will be displayed on form but users are not able to enter information in these elements.

+ 2) Read only Elements :-

An element can be restricted by the user from changing the value of element by setting its readonly property to true. If we want user to enter value in that elements then we can set its readonly property to false.

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

Relevant Program Outcomes (POs)

- **Basic Knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problem.
- **Discipline Knowledge:** Apply computer programming knowledge to solve the computer group related problems.
- **Experiments and practices:** Plan to perform experiments and practices to use the result to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools within understanding of the limitations.
- **Individual and Teamwork:** Function effectively as a leader and the team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

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 replace 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.

::3schools 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 :

Maharashtra State board of Education

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

Client Side Scripting Languages (22519)

\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: \bH\l, end like this: H\l\b
\B	Find a match, but not at the beginning/end of a word
\o	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!");
```

CONCLUSION:

In this method Practical, we learn about Regular Expressions & their method.

Questions:

1. Develop a web page for validation of form fields using regular expressions.
2. Develop JavaScript Code to match string by using exec() Function of RegExp.
3. Explain with example how to validate an Email ID.
- 4 How to replace text using regular Expression? Explain with example.

I. Ans.: Web page for validation of form

```

<!DOCTYPE HTML>
<html>
<head>
<style>
legend {
    display: block;
    padding-left: 2px;
    padding-right: 2px;
    border: none;
}

</style>
<script type="text/javascript">
function validate() {
    var user = document.getElementById("c").value;
    var re = /^[7-9][0-9]{9}\$/;
    if (re.test(user)) {
        alert("done");
        return true;
    }
    else {
        alert ("Enter correct contact number");
        return false;
    }
}
</script>
<head>
<body>
<center>
<h1> Email Registration </h1>
<form>
<fieldset style="width: 300px;">
<legend> Registration </legend>

```

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

```
<table>
  <tr>
    <td> <input type="text" placeholder="firstname" maxlength="10" />
  </tr>
  <br> <br>
  <tr>
    <td> <input type="text" placeholder="lastname" maxlength="10" />
  </tr>
  <br> <br>
  <tr>
    <td> <input type="password" placeholder="password" />
  </tr>
  <br> <br>
  <tr>
    <td> <input type="password" placeholder="confirm" />
  </tr>
  <br> <br>
  <tr>
    <td> <input type="text" placeholder="contact" id="c" />
  </tr>
  <br> <br>
  <tr>
    <td> <label> Gender </label>
      <select id="gender">
```

```
<option value = "male"> male </option>
<option value = "female"> female </option>
<option value = "others"> others </option>
<select>
<tr>
<br> <br>
<br>
<input type = "submit" onclick = "validet()" value = "create" />
</tr>
</table>
</form>
</fieldset>
</center>
</body>
</html>
```

2] Develop Javascript code to match string by using exec() function of RegExp :

```
<html>
<head>
<script language = "javascript" type = "text/javascript">
function checkExp () {
    var x1 = /sneha/;
    var str = document.getElementById("tf1").val;
    var result = x1.exec(str);
    document.getElementById("tf2").innerHTML =
    = result;
```

3) Enter Text : <input type = "text" id = "tf1">
 </head>
 <body>
 <p id = "tf2"></p>
 </body>
 </html>

Explain with Example how to validate an Email ID.
 →

```

<html>
<title> Javascript email validate </title>
<body onload = 'document.form1.text1.focus ()'>
<h2> Enter email to validate </h2>
<form name = "form1" action = "#">
<input type = "text" name = "text1"/>
<li></li>
<li class = "validate" onclick = "validateEmail (document.form1.text1.value)"> 1. <input type = "submit" name = "validate" value = "Validate" onclick = "validateEmail (document.form1.text1.value)"></li>
</form>
<div>
<script src = "email-validation.js"></script>
</body>
</html>
  
```

```
function validateEmail(inputText)
```

```
{  
    var mailformat = /^[^w+([.-]?)w+([.-]?)w+).w{2,}:/;  
    //  
    if (inputText.value.match(mailformat))  
    {  
        alert (" Valid Email ");  
        document . form1 . text1 . focus ();  
        return true ;  
    }  
    else  
    {  
        alert (" You have entered invalid email address ");  
        document . form1 . text1 . focus ();  
        return false ;  
    }  
}  
  
q) How to replace text using regular Expression w  
→  
example  
<html>  
<head>  
<script language="javascript" type="text/javascript">  
function checkExp ()  
{  
  
    var x1 = / - /g;  
  
    var str = document.getElementById("tf1").ve  
    var result = str.replace(x1, "");  
    document.getElementById("tf2").innerHTML=
```



```
</script>
<head>
<body>
    Enter Text : < input type = "text" id = "tf1" >
    <input type = "button" value "Check" onclick =
        "Check EXPO" >
    <p id = "tf2" > </p>
    </body>
</html>
```

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

Relevant Program Outcomes (POs)

- **Basic Knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problem.
- **Discipline Knowledge:** Apply computer programming knowledge to solve the computer group related problems.
- **Experiments and practices:** Plan to perform experiments and practices to use the result to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools within understanding of the limitations.
- **Individual and Teamwork:** Function effectively as a leader and the team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

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 <onmouseover> 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 places 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 Maharashtra State board of Education

Client Side Scripting Languages (22519)

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>
<tr valign="top">
<td width="50">
<a></a>
</td>
<td><img height="1" width="10"></td>
```

Client Side Scripting Languages (22519)

```
<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>
```

CONCLUSION:

"In this practical we learned about Rollover effect.

Questions:
Write a program to create sample web page with Rollover effect.

```
<html>
<head> Rollover Effect </head>
<body>
<textarea rows = "2" cols = "50"
name = "rollover-text">
onmouseover = "this.value = 'Rollover means,
a webpage changes when the user moves his
or her mouse over an object on the page.'>
</textarea>
</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

Relevant Program Outcomes (POs)

- **Basic Knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problem.
- **Discipline Knowledge:** Apply computer programming knowledge to solve the computer group related problems.
- **Experiments and practices:** Plan to perform experiments and practices to use the result to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and Teamwork:** Function effectively as a leader and the team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

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)
Maharashtra State board of Education
```

Client Side Scripting Languages (22519)

```
{  
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>
```

CONCLUSION:

In this practical, we learn how to develop a web page for implementing menus, like context menu, tree & so on.

QUESTIONS:

1. Develop a program to create popup menu.
2. Explain Folding tree menu.
3. Describe context menu in detail.

]]<!DOCTYPE html>
<html>
<head>
<meta name="viewport"
content="width=device-width,initial-
scale=1">
</head>
<body style="text-align:center">
<h2> Popup </h2>

```

<div class = "popup" onclick = "myFunction()> Click me to
toggle the popup.

<span class = "popuptext" id = "mypopup">
A simple popup ! </span>

</div>

<script>

function myfunction()
{
var popup = document.getElementById(
"myPopup");
popup.classList.toggle("show");
}

</script>

</body>

</html>

```

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

②

Folding Tree menu is a graphical user interface control that represents hierarchical data in a tree structure. It provide great performance with its advanced feature like load on demand. Checkbox , support , multiple selection ,tree, navigation . drag & drop tree node editing & template support.

③

A context menu also known as contextual menu, Shortcut menu or popup menus) is the menu that appear when you right click & offers a set of choices that are available choices are usually actions specifically related to the selected object.

```
<html>
<head>
<script>
<!--
    function sayHello()
    {
        alert (" you right click Here");
    }
-->
</script>
</head>
</body>
<p oncontextmenu = "sayHello()"> Right Click </p>
</body>
```

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

Relevant Program Outcomes (POs)

- **Basic Knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problem.
- **Discipline Knowledge:** Apply computer programming knowledge to solve the computer group related problems.
- **Experiments and practices:** Plan to perform experiments and practices to use the result to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools within understanding of the limitations.
- **Individual and Teamwork:** Function effectively as a leader and the team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

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>
Maharashtra State board of Education
```

```

</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')
```

Client Side Scripting Languages (22519)

```
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()">  
</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]  
}  
}
```

Client Side Scripting Languages (22519)

```
</script>
</head>
<body>
<p align="center"></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>
```

CONCLUSION:

In this practical we learned, How to create banner, How to create slides how protection web pages.

QUESTIONS:

1. Develop a program to load and display banner in JavaScript.
2. Explain Slideshow with example.
3. How to protect webpages?

II
→

```
<html>
<head>
<script language="javascript">MyBanners =
new Array ("bg1.jpeg")
banner = 0
function ShowBanners ()
{
if (document.images)
{
banner++
if (banner == MyBanners.length)
{
banner = 0}
document.ChangeBanner.src = MyBanners[banner]
setInterval ("showBanner()", 1000)
}
}
```

```

</script>
<body onload = "ShowBanners()">
<center>
<img src = "bgi.jpeg" width="900"
height="120" name="ChangeBanner"/>
</center>
</body>
</html>

```

2] slideshow: A slideshow is similar to a banner advertisement in which multiple images rotates inside a `` element on the web page.

Creating a slideshow:

create a `` tag & two buttons for navigation between images.

A javascript function named as `slideShow()` takes status argument which can have two values 1 or -1.

If status is 1 then next image will display in `` element else previous image will display in `` elements.

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

Example:

```
<html>
<head>
<h1> Slide Show </h1>
<script language="javascript" type="text/javascript">
pics = new Array('bgi.jpeg')
Count = 0;
function slideshow(status)
{
    if (document.images)
    {
        Count = Count + status;
        if (Count > (Pic.length - 1))
        {
            Count = 0
        }
        if (Count < 0)
        {
            Count = Pic.length - 1;
        }
        document.images[Count].src = pic[Count];
    }
}
</script>
</head>
</body>

<br>
<input type="button" value="Next" onclick=
"Slideshow(1)">
```

```
<input type = "button" value = "Back" onclick = "slideshow"
</html>
```

3) Protect web pages:

We are aware that anyone with a computer knowledge can use a few mouse click to display your HTML code, including your javascript code on screen.

Although you cannot entirely prevent Potruders to view our web page but can take few steps to stop all but the best computer wizards from gaining access to your javascript.