Q.1 Attempt any FIVE of the following: [10]
Q.1(a) State the use of dot syntax in JavaScript with the help of suitable example. [2]
Ans.: In JavaScript, one can access properties using the dot notation. Dot notation allows us to
tell instance of a class to use one of the methods inside that class. For eg, car.name

Q.1(b) List and explain Logical operators in JavaScript. [2]
Ans.: Assume variable A holds 10 and variable B holds 20, then -

<table>
<thead>
<tr>
<th>Operator &amp; Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i) &amp; &amp; (Logical AND)</td>
</tr>
<tr>
<td>If both the operands are non-zero, then the condition becomes true.</td>
</tr>
<tr>
<td>Ex: (A &amp; &amp; B) is true.</td>
</tr>
<tr>
<td>(ii)</td>
</tr>
<tr>
<td>If any of the two operands are non-zero, then the condition becomes true.</td>
</tr>
<tr>
<td>Ex: (A</td>
</tr>
<tr>
<td>(iii) ! (Logical NOT)</td>
</tr>
<tr>
<td>Reverses the logical state of its operand. If a condition is true, then the Logical NOT</td>
</tr>
<tr>
<td>operator will make it false.</td>
</tr>
<tr>
<td>Ex: !(A &amp; &amp; B) is false.</td>
</tr>
</tbody>
</table>

Q.1(c) Write a JavaScript that identifies a running browser. [2]
Ans.:  
```html
<html>
  <head>
  </head>
  <body>
    <h2>Navigator Object</h2>
    <p id="p1">what is your Browser Application name?</p>
    <p id="p2">What is your Browser Application code name?</p>
    <script>
      document.getElementById("p1").innerHTML = "Browser Application name is: " + navigator.appName;
      document.getElementById("p2").innerHTML = "Browser Application code name is: " + navigator.appCodeName;
      document.getElementById("p2").innerHTML = "Browser version is: " + navigator.appVersion;
    </script>
  </body>
</html>
```

Q.1(d) Write a JavaScript that initializes an array called “Fruits” with names of five fruits. The script then displays the array in a message box. [2]
Ans.:  
```html
<html>
  <body>
    <h2>JavaScript Arrays</h2>
    <p>The length property returns the length of an array.</p>
    <script>
      var fruits = ["Banana", "Orange", "Apple", "Mango", "Pineapple"];
      alert(fruits);
    </script>
  </body>
</html>
```
Q.1(e) Give syntax of and explain the use of “with” statement/clause in JavaScript using suitable example.

Ans.: The 'with' statement adds the given object to the head of this scope chain during the evaluation of its statement body. If an unqualified name used in the body matches a property in the scope chain, then the name is bound to the property and the object containing the property. Otherwise a ReferenceError is thrown.

Syntax:
with (expression)
    statement

Example:
function f(foo, values)
{
    with (foo)
    {
        document.write(values);
    }
}

Q.1(f) Enlist and explain the use of any two Intrinsic JavaScript functions.

Ans.: abs()
    Returns the absolute value of a number.

    exp()
    Returns E^N, where N is the argument, and E is Euler's constant, the base of the natural logarithm.

Q.1(g) State and explain what is a session cookie?

Ans.: • A session cookie contains information that is stored in a temporary memory location and then subsequently deleted after the session is completed or the web browser is closed. This cookie stores information that the user has inputted and tracks the movements of the user within the website.
    • The main difference between a session and a cookie is that session data is stored on the server, whereas cookies store data in the visitor's browser. Sessions are more secure than cookies as it is stored in server.

Q.2 Attempt any THREE of the following:

Q.2(a) What are JavaScript Accessors (setters and getters)?

Ans.: • Getters and setters exist in most object-oriented programming languages, including JavaScript.
    • They are code constructs that help developers access the properties of objects in a secure way.
    • With getters, you can access ("get") the values of properties from external code, while setters let you change ("set") their values.
    • It gives simpler syntax.
    • It allows equal syntax for properties and methods.
    • It can secure better data quality
    • A JavaScript object can have multiple properties and methods that store static data and dynamic functionality. Properties are static key-value pairs, while methods are functions specific to the object.
For example, Car.color could be a property, while Car.drive() could be a method of the Car object. With a getter, you can access the color property of the Car object and with a setter, you can modify its value (for instance, from blue to black).

To get the properties of the Car object, you need to call the getter methods.

Example:
```javascript
var obj = {
Ex: 'this is the value of foo',
getEx: function()
{
    return this.Ex;
},
setEx: function(val)
{
    this.Ex = val;
}
}

console.log(obj.getEx()); // "this is the value of foo"

obj.setEx('hello');

console.log(obj.getEx()); // "hello"
```

Q.2(b) Write a JavaScript program which compute, the average marks of the following students. Then, this average is used to determine the corresponding grade.

<table>
<thead>
<tr>
<th>Student Name</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advait</td>
<td>80</td>
</tr>
<tr>
<td>Anay</td>
<td>77</td>
</tr>
<tr>
<td>Manyata</td>
<td>88</td>
</tr>
<tr>
<td>Saanvi</td>
<td>95</td>
</tr>
<tr>
<td>Saachi</td>
<td>68</td>
</tr>
</tbody>
</table>

The grades are computed as follows:

<table>
<thead>
<tr>
<th>Range</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;60</td>
<td>F</td>
</tr>
<tr>
<td>&lt;70</td>
<td>D</td>
</tr>
<tr>
<td>&lt;80</td>
<td>C</td>
</tr>
<tr>
<td>&lt;90</td>
<td>B</td>
</tr>
<tr>
<td>&lt;100</td>
<td>A</td>
</tr>
</tbody>
</table>

Ans.: ```html
<!DOCTYPE html>
<html>
<head>
  <title>Compute the average marks and grade</title>
</head>
<script>
var students = [['Advait', 80], ['Anay', 77], ['Manyata', 88], ['Saanvi', 95], ['Saachi', 68]];
var Avgmarks = 0;
for (var i=0; i < students.length; i++)
{
    Avgmarks += students[i][1];
```
var avg = (Avgmarks/students.length);
}
document.write("Average grade: " + (Avgmarks)/students.length);

if (avg < 60)
{
    document.write("Grade: F");
}
else if (avg < 70)
{
    document.write("Grade: D");
}
else if (avg < 80)
{
    document.write("Grade: C");
}
else if (avg < 90)
{
    document.write("Grade: B");
}
else if (avg < 100)
{
    document.write("Grade: A");
}
</script>
</html>

Q.2(c) Write a JavaScript that displays all properties of window object. Explain the code.

Ans.: <html>
<body>
<script type="text/javascript">
function msg()
open("http://www.google.com");

</script>
</script>
document.writeln("<br/>screen.width: "+screen.width);
document.writeln("<br/>screen.height: "+screen.height);
</script>

var myWindow;
function openWin() {
    myWindow=window.open("", "myWindow", "width=200, height=100");
    myWindow.document.write("<p>This is new Window</p>");
}

function moveWin() {
    myWindow.moveTo(500, 100);
    myWindow.focus();
}
</script>
<input type="button" value="new window" onclick="msg()"/>
Q. 2(d) Write a JavaScript function that checks whether a passed string is palindrome or not.

Ans.: <html>
<head>
<title>Palindrome</title>
</head>
<body>
<script>
function palindrome()
{
    var str1 = new String("racecar");
    var b = str1.length;
    for (var i=0; i<b-1; i++)
    {
        if (str1[i]!=str1[b-1-i])
        {
            return false;
        }
    }
    return true;
}
var c=palindrome();
if (c == true)
{
    document.write("String is an palindrome");
}
else
{
    document.write("String is not a palindrome");
}
</script>
</body>
</html>

Q. 3 Attempt any THREE of the following: [12]

Q. 3(a) How to combine array elements into a string? [4]

Ans.: The array.join() method is an inbuilt function in JavaScript which is used to join the elements of an array into a string. The elements of the string will be separated by a specified separator and its default value is comma( , ).

Syntax:
array.join(separator);

Parameters:
separator: It is Optional i.e, it can be either used as parameter or not. Its default value is comma( , ).

Return Value:
It returns the String which contain the collection of array's elements.
Example:
var fruits = ["Banana", "Orange", "Apple", "Mango"];
var energy = fruits.join;

Output:
Banana,Orange,Apple,Mango

Example:
var fruits = ["Banana", "Orange", "Apple", "Mango"];
var energy = fruits.join(" and ");

Output:
Banana and Orange and Apple and Mango

Q.3(b) Write a JavaScript function to count the number of vowels in a given string.

Ans.:
```
<html>
<head>
<body>
<script>
    function getvowel()
    {
        var str = "ajay";
        var count=0;
        var string = str.toString();
        for(var i=0;i<string.length-1;i++)
        {
            if (string.charAt(i)=="a"||string.charAt(i)=="e"||string.charAt(i)=="i"||string.charAt(i)=="o"||string.charAt(i)=="u"||string.charAt(i)=="A"||string.charAt(i)=="E"||string.charAt(i)=="I"||string.charAt(i)=="O"||string.charAt(i)=="U")
                count+=1;
        }
        return count;
    }
</script>
</body>
</head>
</html>
```

Q.3(c) Write a JavaScript that find and displays number of duplicate values in an array.

Ans.:
```
function find_duplicate_in_array(arr1)
{
    var object = {};
    var result = [];

    arr1.forEach(function (item) {
        if(object[item])
            object[item] = 0;
        else
            object[item] += 1;
    });
```
for (var prop in object) {
    if (object[prop] >= 2) {
        result.push(prop);
    }
}
return result;

Q.3(d) Write a function that prompts the user for a color and uses what they select to set the background color of the new webpage opened.

Ans.: Demo.html

```html
<!DOCTYPE html>
<html>
<head>

<title>Change the Background Color with JavaScript</title>
<script>
// Function to change webpage background color
function changeBodyBg() {
    window.open("Yellow.html");
}

function changeBodyBgR() {
    window.open("Red.html");
}

</script>
</head>
<body>
<h1 id="heading">This is a heading</h1>
<p>This is a paragraph of text.</p>
<hr>
<div>
    <label>Change Webpage Background To:</label>
    <button type="button" onclick="changeBodyBg()">Yellow</button>
    <button type="button" onclick="changeBodyBgR()">Red</button>
</div>
<br>
</body>
</html>

Yellow.html

<!DOCTYPE html>
<html>
<head>
<body style="background-color:yellow;">
    <h1 id="heading">Yellow</h1>
</body>
</html>
Red.html

```html
<!DOCTYPE html>
<html>
<body style="background-color:red;">
  <h1 id="heading">Red</h1>
</body>
</html>
```

Q.4 Attempt any THREE of the following: [12]
Q.4(a) State what is a regular expression? Explain its meaning with the help of a suitable example.
Ans.: A regular expression is very similar to a mathematical expression, except a regular expression tells the browser how to manipulate text rather than numbers by using special symbols as operators.

For example, the browser might be told to determine whether a specific character exists in one or more lines of text. Likewise, the browser might be told to replace all occurrences of a word with another word. This and more can be accomplished by writing a regular expression.

Unlike a mathematical expression, a regular expression begins and ends with a slash (/). You place the special symbols that make up the regular expressions between slashes. You'll notice that a pair of square brackets ([ ]) appears following the first slash. This tells the browser to search the text for characters that appear within the brackets.

Example:

```html
<html>
<head>
<title>Simple Regular Expression</title>
<script>
function RegExpExpression()
{
 var name='Bob'
 re = /[bB]/
 if (re.test(name))
 {
 alert('Found')
 }
 else
 {
 alert('Not Found')
 }

</script>
</head>
<body>
<form action="" method="post">
<p>
<input name="Run RegExp" value="Run RegExp Expression" type="button" onclick="RegExpExpression()"/>
</p>
</form>
</body>
</html>
```
In this expression, two characters are within the square brackets: a b and a t, which tells the browser to determine whether the text includes a b or a t, or both. That’s the regular expression.

The regular expression is assigned to the re variable. We don’t use quotation marks, which would tell the browser that the special symbols of the regular expression is part of a string, which it isn’t.

The test() method is called and passed the variable name that contains the string Bob. The test() method is one of several methods of the regular expression object.

The browser evaluates Bob using the regular expression. A true is returned if either a b or a t or both are found in the name Bob; otherwise a false is returned. Depending on this result, the appropriate alert dialog box is displayed on the screen.

**Q.4(b) With the help of the example explain how to replace text using regular [4] expression?**

**Ans.:** We can also use a regular expression to replace portions of the text by using the replace() method.

The replace() method requires two parameters: a regular expression and the replacement text.

how the replace() method works:
First, we create a regular expression that identifies the portion of the text that you want replaced.

Then you determine the replacement text. Pass both of these to the replace() method, and the browser follows the direction given in the regular expression to locate the text. If the text is found, the browser replaces it with the new text that you provided.

A common problem is to replace all occurrences of one or more characters of a string. You do this by creating a regular expression and calling the replace() method; however, you’ll need to place the g special character at the end of the regular expression, which tells the browser to replace all occurrences of the regular expression in the string.

**Example:**
```html
<html>
<head>

<script>
function check()
{
    var abc = /-/g;
    var str = document.getElementById("txt").value;
    var xyz = str.replace(abc, "."DEFINED");

document.getElementById("demo1").innerHTML = xyz;
}
</script>
</head>
<body>
Enter text"<textarea id="txt" /></textarea>"
<input type="button" onclick="check()" value="Check">
```
Q. 4(c) Write the syntax of and explain use of following methods of JavaScript Timing [4]
Event. (i) setTimeout() (ii) setInterval()
Ans.: (i) setTimeout():
Syntax:
```
setTimeout(function, milliseconds, parameter1, ...)
```
Parameter: It accept some parameters which are specified below-
- function: It is the function that will be executed.
- milliseconds: It is the number of milliseconds to wait before executing the code. It is optional and its default value is zero(0).
- parameter1: It is additional parameters to pass to the function and it is optional.
- Return Value: It returns a number representing the ID value of the timer that is set.

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

Syntax:
```
setInterval(function, milliseconds, param1, param2, ...)
```
Parameters:
- 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

Q. 4(d) How to create banner ads in javascript? [4]
Ans.: Displaying banners ads is a common practice for showing advertisements on web pages to the visitors. You can create some animated effect using JavaScript, like rotating static banner ads at a certain time interval.

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. Let’s create four banner images and name them as banner1.jpg, banner2.jpg, banner3.jpg and banner4.jpg
The JavaScript starts by declaring an array to store the banner images using the new Array keywords, as follows:

```javascript
MyBanners = new Array("banner1.jpg","banner2.jpg","banner3.jpg","banner4.jpg")
```

Each element in the array is assigned with an index, starting from 0. In our example, banner1.jpg is assigned with index 0, banner2.jpg is assigned with index 1, banner3.jpg is assigned with index 2 and banner3.jpg is assigned with index 3.

To track the index of the current banner, we can assign the index to a variable. Here we use the variable banner and initialize it with a value 0 to load the first banner image.

Next, we create a function ShowBanners which will display all the banner images at fix interval. This can be achieved by incrementing the index values using the statement banner++ and when the index value is equal to the total number of elements in the array (denoted by MyBanners.length), the index value is set back to 0, which is the index of the first banner.

```javascript
function ShowBanners()
{
    if (banner == MyBanners.length)
    {
        banner = 0
    }
    document.ChangeBanner.src = MyBanners[banner]
    setTimeout("ShowBanners()", 5000)
}
```

The process is repeated at a fix time interval using the setTimeout() function. The setTimeout() function comprises two arguments, the first is the function to be activated, i.e., the ShowBanners() function and the second one is the duration measured in milliseconds, therefore 1000 is equivalent to 1 second so 5000 is equal to 5 seconds.

The final part of the JavaScript is to call out the ShowBanners() function using the onload method.

Example:
```html
<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>
</html>
```
Q.4(e) Write a program to demonstrate slide menu.  

Ans.:  
```html
<html>
<head>
</head>
<body>
<div id="navbar">
<a href="#home">Home</a>
<a href="#news">News</a>
<a href="#contact">Contact</a>
</div>
<div style="padding:15px 15px 2500px;font-size:30px">
<p>This example demonstrates how to slide down a navbar when the user starts to scroll the page.</p>
<p>Scroll down this frame to see the effect!</p>
<p>Scroll to the top to hide the navbar.</p>
</div>
<script>
// When the user scrolls down 20px from the top of the document, slide down the navbar
window.onscroll = function()
{
    scrollFunction();
};
function scrollFunction()
{
    if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20)
    {
        document.getElementById("navbar").style.top = "0";
    }
    else
    {
        document.getElementById("navbar").style.top = "-50px";
    }
}
</script>
</body>
</html>
```

Q.5 Attempt any TWO of the following:  

Q.5(a) Write HTML Script that displays textboxes for accepting Name, middle name,  
Surname of the user and a Submit button. Write proper JavaScript such that  
when the user clicks on submit button  
(i) all textboxes must get disabled and change the color to "RED", and with  
respective labels.  
(ii) Constructs the mailID as <name>.<surname>@msbte.com and displays mail ID  
as message. (Ex. If user enters Rajni as name and Pathak as surname mailID  
will be constructed as rajni.pathak@msbte.com).

Ans.:  
```html
<html>
<head>
</head>
<body>
</body>
</html>
```
Q.5(b) Write a webpage that displays a form that contains an input for Username and password. User is prompted to enter the input and password and password becomes value of the cookie. Write The JavaScript function for storing the cookie. It gets executed when the password changes.

Ans.: <html>
<body>
<script>
function validateform(){
    var name=document.myform.name.value;
    var password=document.myform.password.value;
    if (name==null || name==""){
        alert("Name can’t be blank");
        return false;
    }else if(password.length<6)
    {
        alert("Password must be at least 6 characters long.");
        return false;
    }
}
</script>
<body>
Q.5(c) Write a script for creating following frame structure:

<table>
<thead>
<tr>
<th>FRAME 1</th>
<th>FRAME2</th>
<th>FRAME3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FRUITS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FLOWERS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CITIES</td>
<td></td>
</tr>
</tbody>
</table>

Fruits, Flowers and Cities are links to the webpage fruits.html, flowers.html, cities.html respectively. When these links are clicked corresponding data appears in "FRAME3".

Ans.:<html>
<head>
  <title>Iframe Demo</title>
</head>
<body>
<table border="1">
<tr>
  <td align="center" colspan="2">
    FRAME 1
  </td>
</tr>
<tr>
  <td>
    FRAME 2
  </td>
</tr>
<tr>
  <td>
    FRAME 3<br>
    <iframe name="mainframe"></iframe>
  </td>
</tr>
<tr>
  <td>
    <ul>
      <li>
        <a href="fruits.html" target="mainframe">FRUITS</a>
      </li>
      <li>
        <a href="flowers.html" target="mainframe">FLOWERS</a>
      </li>
      <li>
        <a href="cities.html" target="mainframe">CITIES</a>
      </li>
    </ul>
  </td>
</tr>
</table>
</body>
</html>
Q.6 Attempt any TWO of the following: [12]

Q.6(a) Write HTML Script that displays dropdownlist containing options New Delhi, Mumbai, Bangalore. Write proper JavaScript such that when the user selects any options corresponding description of about 20 words and image of the city appear in table which appears below on the same page.

Ans.:<html>
<body>
<form>
<select name="Select City">
<option value="New Delhi">New Delhi</option>
<option value="Mumbai">Mumbai</option>
<option value="Bangalore">Bangalore</option>
</select>
<div class="new delhi">You have selected New Delhi so i am here</div>
</form>
</body>
</html>

Q.6(b) Write a JavaScript function to hide email addresses to protect from unauthorized user.

Ans.: protect_email = function (user_email) {
    var avg, splitted, part1, part2;
    splitted = user_email.split("@");
    part1 = splitted[0];
    avg = part1.length / 2;
    part1 = part1.substring(0, (part1.length - avg));
    part2 = splitted[1];
    return part1 + "...@" + part2;
};
console.log(protect_email("robin_singh@example.com"));

Output:
robin...@example.com

Q.6(c) Create a slideshow with the group of four images, also simulate the next and previous transition between slides in your JavaScript.

Ans.:<html>
<head>
<title>Banner Ads</title>
<script type="text/javascript">
var pics=new Array('picture(1).jpg', 'picture(2).jpg', 'picture 3.jpg', 'picture 4.jpg');
var count=0;
function SlideShow(status)
{
    if(document.images)
```javascript
{ 
count=count+status;
if(count>(pics.length-1))
{
count=0;
}
if(count<0)
{
count=pics.length-1;
}
document.img1.src=pics[count];
}
</script>
</head>
<body>
<img src="picture(1).jpg" width="200" alt="" height="300" name="img1"/>
<br>
<input type="button" value="Next" onclick="SlideShow(1)"/>
<input type="button" value="Back" onclick="SlideShow(-1)"/>
</body>
</html>

□ □ □ □ □