Practical No. 8

Develop a webpage for creating session and persistent cookies. Observe the effects with Browser cookie settings

Theoretical Background:

Cookies Basics

- A **cookie** is a small piece of information that a web site writes to user's hard disk when he visit the site.
- Cookies let you store user information in web pages.
- Cookies are data, stored in small text files, on your computer.
- When a web server has sent a web page to a browser, the connection is shut down, and the server forgets everything about the user.
- Cookies were invented to solve the problem "how to remember information about the user": • When a user visits a web page, his/her name can be stored in a cookie. • Next time the user visits the page, the cookie "remembers" his/her name.
- A JavaScript can be used to create cookies whenever someone visits the web page that contains the script.
- A JavaScript can also be used to read cookies stored on a user's computer, and it uses the information stored in a cookie to personalize the web page that a user visits.
- The text of a cookie must contain a name-value pair, which is a name and value of the information.
- There are two types of Cookies.
  - Session Cookies
  - Persistent Cookies

1. Session Cookies

- Also also known as an **in-memory cookie**
- It resides in memory for the length of the browser session.
- A browser session begins when the user starts the browser and ends when the user exits the browser.
- Even if the user surfs to another web site, the cookie remains in memory.
- Session cookie is automatically deleted when the user exits the browser application.

2. Persistent Cookies

- Also known as **transient cookie.**

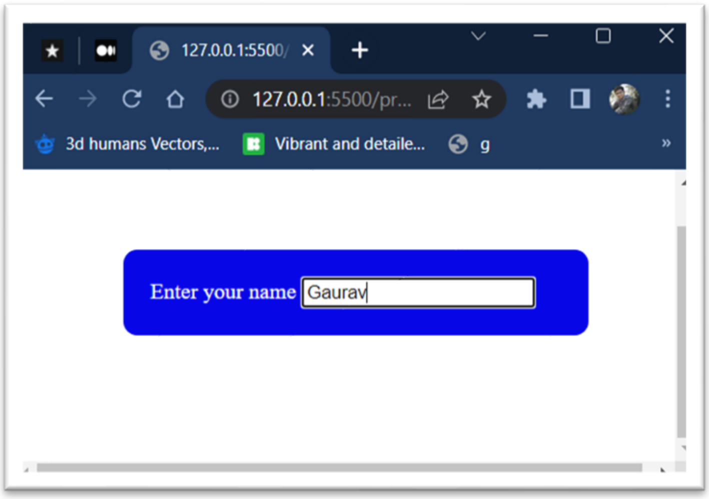
- A persistent cookie is a cookie that is assigned an expiration date.
- It is written to the computer's hard disk and remains there until the expiration date has been reached; then it's deleted.

CODE:

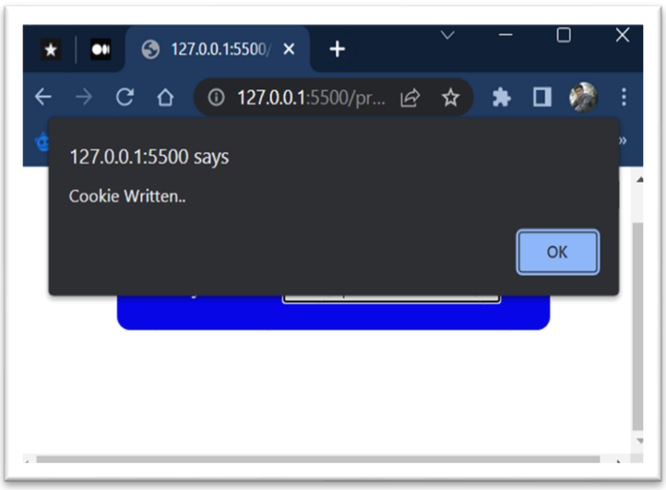
```
<html>
  <head>
    <style>
        body{
            width: 100%;
            height: 100vh;

          display: flex;
          align-items: center;
          justify-content: center;
            position: relative;
        }
      .Center{
        padding: 20px;
          width: 300px;
          height: auto;
          color: white;
          position: absolute;
          border-radius: 10px;
          background-color: rgb(0, 0, 229);
      }
    </style>
  </head>
  <body>
    <div class="Center">
      Enter your name
      <input
        type="text"
        id="myname"
        placeholder="Enter Name"
        onchange="createCookie()"
      />
    </div>
  </body>
  <script>
    function createCookie() {
      var name = document.getElementById("myname").value;
      document.cookie = "name=" + name + ";";
      alert("Cookie Written..");
    }
  </script>
```

```
</html>
```

Output:



Enter your name Gaurav



127.0.0.1:5500 says

Cookie Written..

Marks Obtained			Dated signature of Teacher
Process Related (15)	Product Related (35)	Total (50)	

Practical No. 9

Develop a webpage for placing the window on the screen and working with child window

Theoretical Background:

Window Object

- JavaScript's **"window"** object represents the browser window or, the frame that a document is displayed in.
- The properties of a particular instance of Window might include its size, the buttons, scrollbars, browser frame, position, and so on.
- The methods of the Window object include the creation and destruction of generic windows and the handling of special case windows such as alert, confirmation, and prompt dialogs.

Opening and Closing Window

- The Window object methods **open()** and **close( )** are used to create and destroy a window, respectively.
- When you open a window, you can set its URL, name, size, buttons, and other attributes, such as whether or not the window can be resized.
- **Syntax**

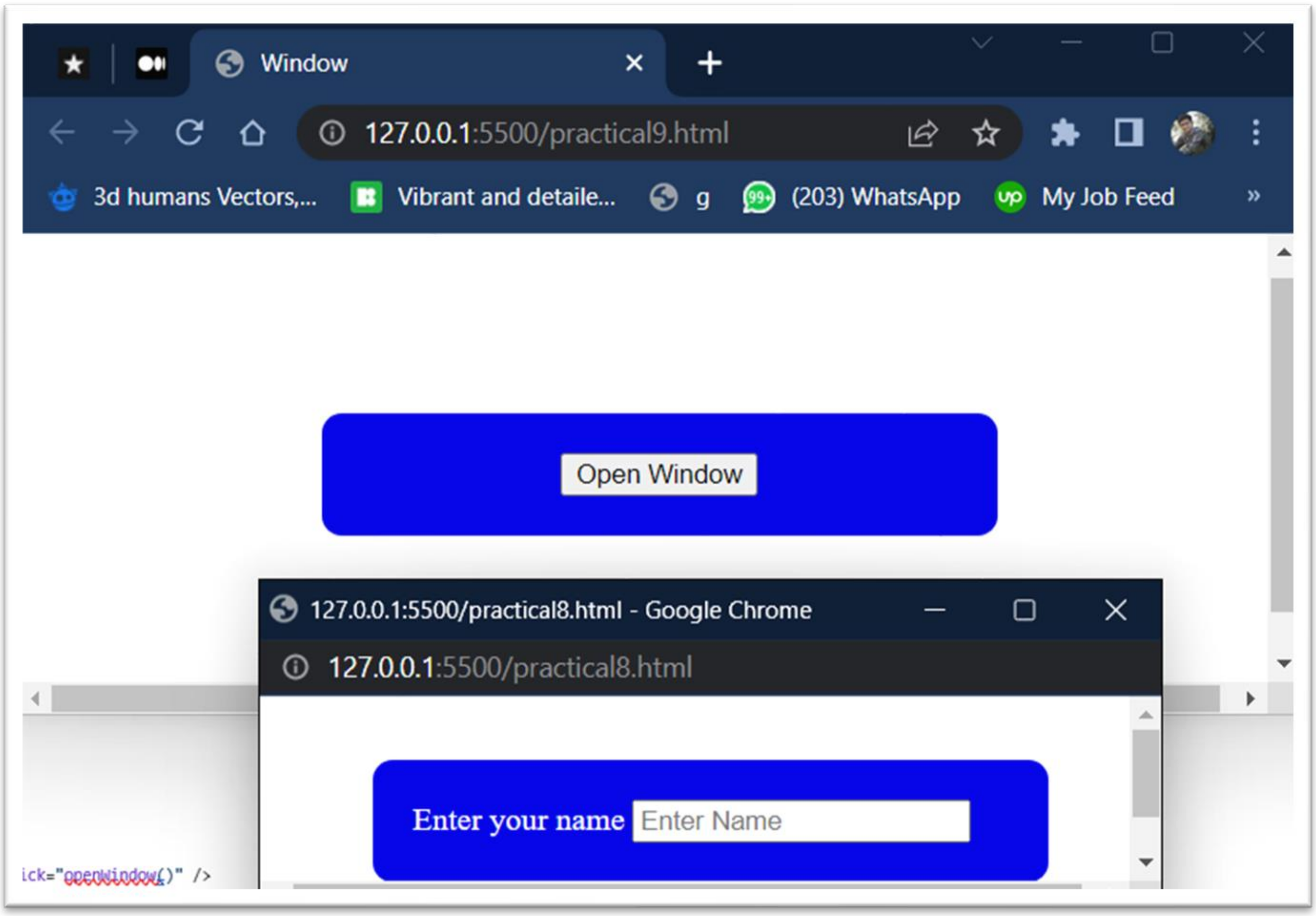
```
window.open (url, name, features)
```

- Where,
  - **url** is a URL that indicates the document to load into the window.
  - **name** is the name for the window (which is useful for referencing later on using the target attribute of HTML links).
  - **features** is a comma-delimited string that lists the features of the window.

CODE:

```
<html>
  <head>
    <title>Window</title>
    <style>
      body {
        width: 100%;
        height: 100vh;
        display: flex;
        align-items: center;
        justify-content: center;
        position: relative;
      }
      .Center {
        padding: 20px;
        width: 300px;
        height: auto;
        color: white;
        position: absolute;
        border-radius: 10px;
        display: flex;
        justify-content: center;
        background-color: rgb(0, 0, 229);
      }
    </style>
  </head>
  <body>
    <div class="Center">
        <input type="button" value="Open Window" onclick="openWindow()" />
    </div>
  </body>
  <script>
    function openWindow() {
      myWindow = window.open(
        "practical8.html",
        "My Window",
        "top=700, left=800, width=450, height=100,status=1"
      );
    }
  </script>
</html>
```

OUTPUT:



Marks Obtained			Dated signature of Teacher
Process Related (15)	Product Related (35)	Total (50)	

Practical No. 10

Develop a webpage for validation of form fields using regular expressions

Theoretical Background :

Regular Expression

- A **Regular Expression** is an object that describes a pattern of characters.
- A Regular Expression (also “**regexp**”, or just “**reg**”) is a sequence of characters that forms a search pattern.
- They are useful for **searching** and **replacing** the characters in the string that match a pattern.
- For example, regular expressions can be used to validate form fields like email addresses and phone numbers, to perform all types of text search and text replace operations, for counting specific characters in a string.
- A regular expression is very similar to a mathematical expression, except it tells the browser how to manipulate text rather than numbers by using special symbols as operators.

Crating Regular Expression

- A regular expression could be defined by using two ways:

1) Using RegExp constructor

Syntax - var pattern = new RegExp(pattern [, flags]);**Example** - var re1 = new RegExp("xyz", "i");

2) Using literal

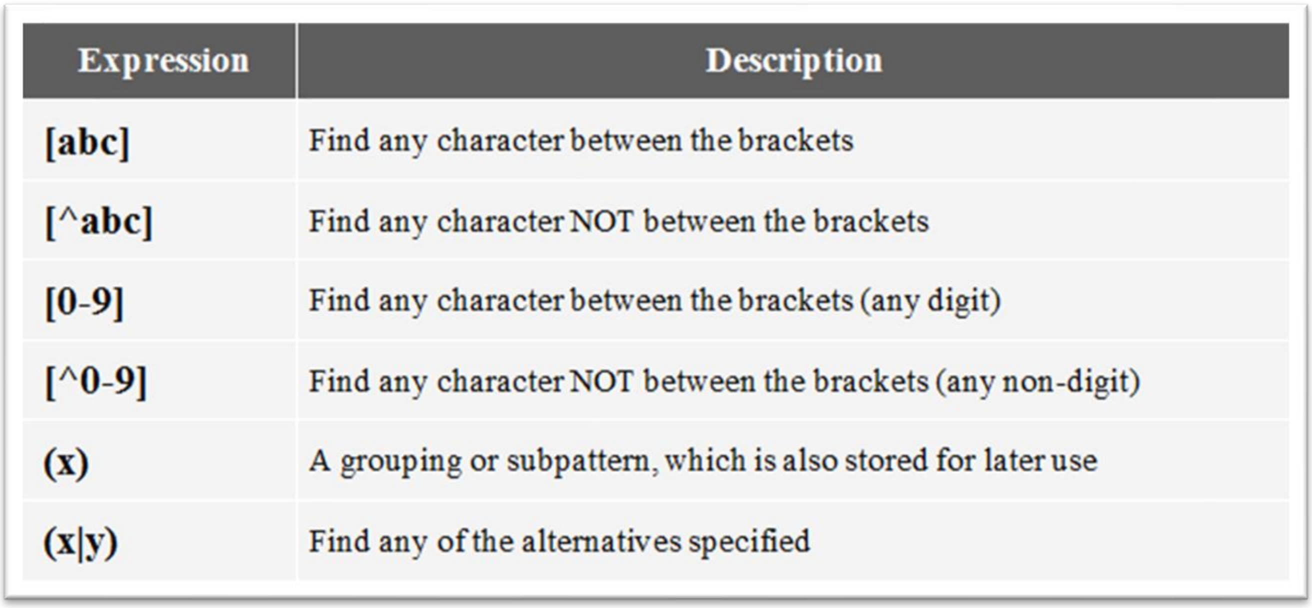
Syntax - var pattern = /pattern/flags;**Example** - var re2 = /xyz/;

Language of Regular Expression

- The language of regular expression consists of following:
  1) Character Classes (Brackets)
  2) Metacharacters
  3) Quantifiers

1)Character Classes (Brackets)

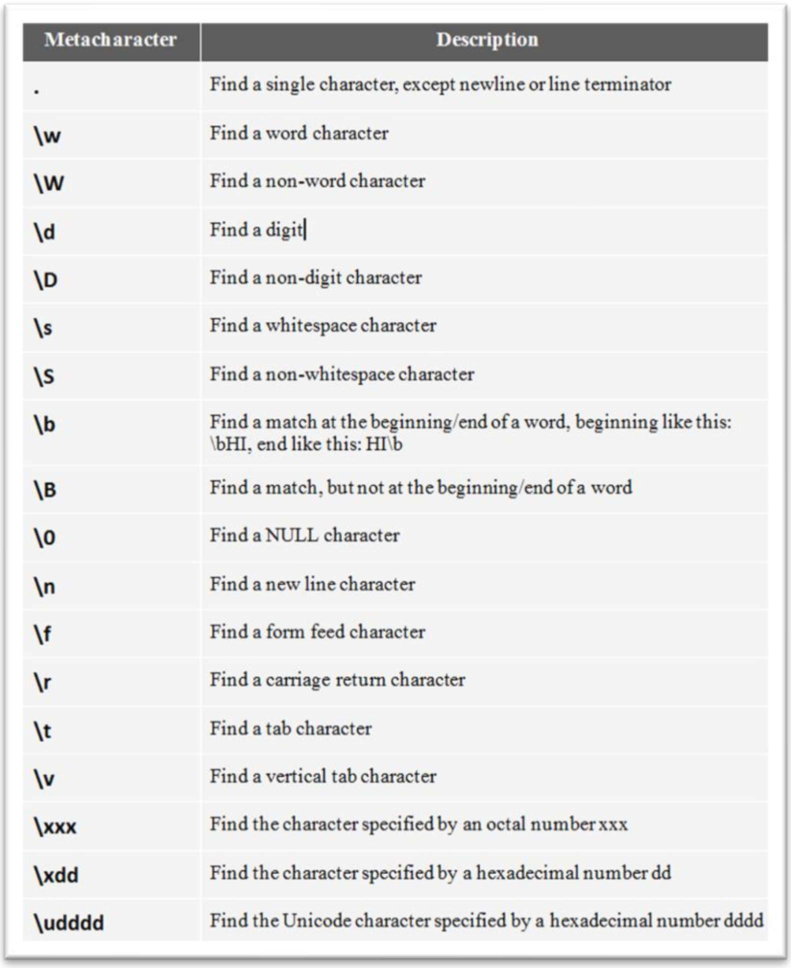
- You can group characters by putting them in Square brackets.
- Brackets are used to find a range of characters.
- […] matches any one of the characters between the brackets
- [^…] matches any one character, but not one of those inside the brackets
- (x|y) matches any of the alternatives specified.



Expression	Description
[abc]	Find any character between the brackets
[^abc]	Find any character NOT between the brackets
[0-9]	Find any character between the brackets (any digit)
[^0-9]	Find any character NOT between the brackets (any non-digit)
(x)	A grouping or subpattern, which is also stored for later use
(x|y)	Find any of the alternatives specified

2)Metacharacters

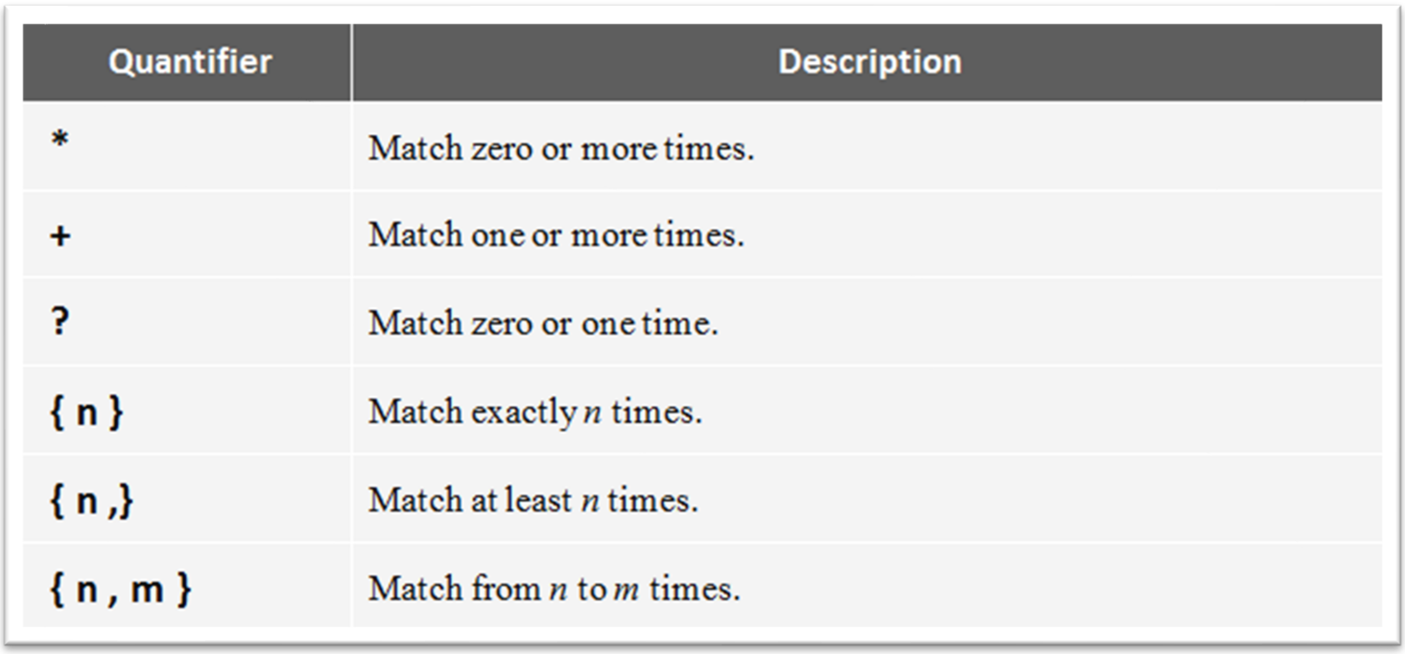
- **Metacharacters** are characters with a special meaning within the language of regular expression.



Metacharacter	Description
.	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 NULL 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

3)Quantifiers

- **Quantifiers** match a number of instances of a character, group, or character class in a string.
- The following table list the quantifiers:



Quantifier	Description
*	Match zero or more times.
+	Match one or more times.
?	Match zero or one time.
{ n }	Match exactly *n* times.
{ n ,}	Match at least *n* times.
{ n , m }	Match from *n* to *m* times.

RegExp class functions:

Here is a list of the methods associated with RegExp along with their description.

1) exec( ) method

- The exec method searches string for text that matches regexp. If it finds a match, it returns an array of results; otherwise, it returns null.
- **Syntax**

```
RegExpObject.exec( string );
```

- It returns the matched text if a match is found, and null if not.

2) test( ) method

- The test method searches string for text that matches regexp. If it finds a match, it returns true; otherwise, it returns false.
- **Syntax**

```
RegExpObject.test( string );
```

- It returns the matched text if a match is found, and null if not.

String class functions:

1) match(pattern)

Searches for a matching pattern. Returns an array holding the results, or null if no match is found

2) replace(pattern1, pattern2)

Searches for pattern1. If the search is successful pattern1 is replaced with pattern2

3) search(pattern)

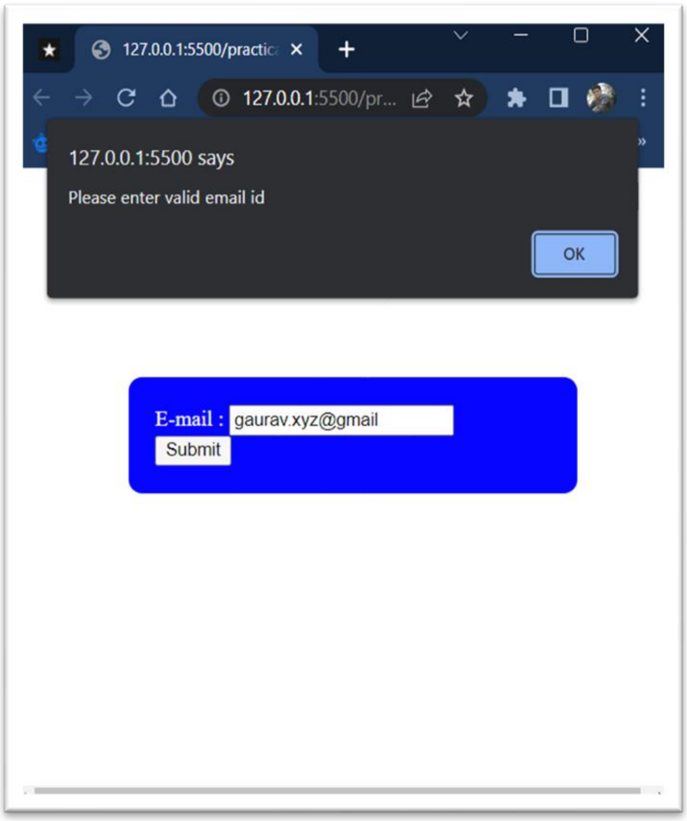
Searches for pattern in the string. If the match is successful, the index of the start of the match is returned. If the search fails, the function returns -1.

CODE:

```
<html>
  <head>
    <style>
      body {
        width: 100%;
        height: 80vh;

        display: flex;
        align-items: center;
        justify-content: center;
        position: relative;
      }
      .Center {
        padding: 20px;
        width: 300px;
        height: auto;
        color: rgb(255, 255, 255);
        position: absolute;
        border-radius: 10px;
        background-color: rgb(0, 0, 255);
      }
    </style>
  </head>
  <body>
    <div class="Center">
      E-mail : <input type="text" placeholder="Enter E-mail" id="email" /><br />
      <input type="button" value="Submit" onclick="checkEmail()" />
    </div>
  </body>
  <script>
    function checkEmail() {
      var email = document.getElementById("email").value;
      var regex = /^([a-zA-Z0-9_\.]+)@([a-zA-Z0-9_\.]+)\.([a-zA-Z]{2,5})$/;
      var res = regex.test(email);

      if (!res) {
        alert("Please enter valid email id");
      } else {
        alert("Thank you");
      }
    }
  </script>
</html>
```

OUTPUT:

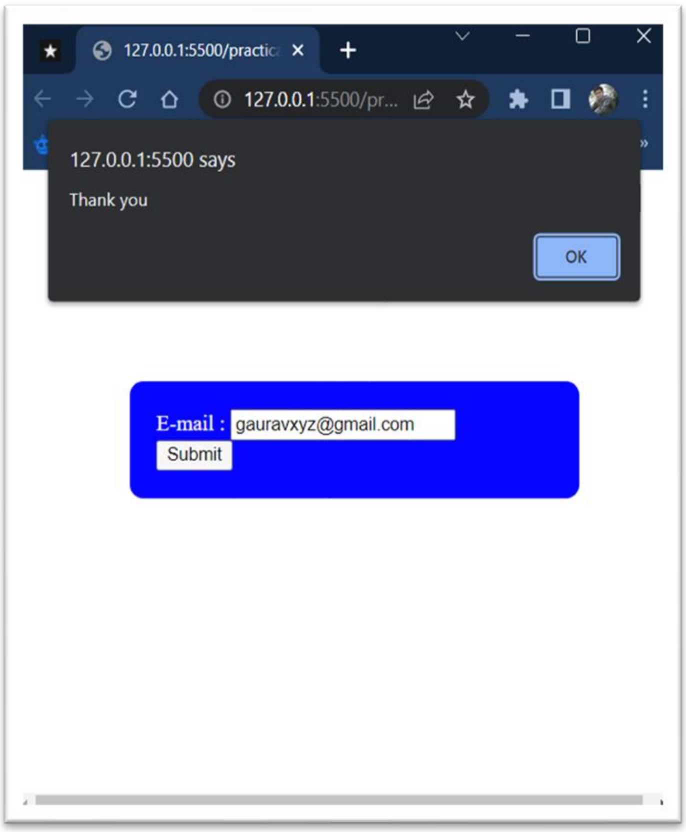
For Invalid E-mail Address



127.0.0.1:5500 says

Please enter valid email id

For valid E-mail Address



127.0.0.1:5500 says

Thank you

Marks Obtained			Dated signature of Teacher
Process Related (15)	Product Related (35)	Total (50)	

Practical No. 11

Develop JavaScript to implement Functions.

Theoretical Background :

Menus

- A **website menu** is a series of linked items that serve in navigating between the different pages or sections of a website.
- There are several kinds of menus, depending on the website's content and design.
- The main types of website menus are:

1. **Classic navigation menu:** This most widespread kind of menu is placed in the website's header, typically as a horizontal list.
2. **Sticky menu:** Also known as a fixed or floating menu, this menu stays put as visitors scroll down the site. These are ideal for long-scrolling pages.
3. **Dropdown menu:** A menu in which a list of additional items opens up once visitors click on — or hover over — one of the menu items. This option is suitable for websites with a lot of content.
4. **Sidebar menu:** A list of menu items that's located on the left or right side of a webpage.

1) Pulldown Menu

- A web site is normally a contents a collection of web pages. A site visitor navigates from one page to another.
- If a menu of these web pages is created then it becomes easy for a visitor to select appropriate web page.
- The <**select**> element is used to create a pulldown menu.
- The <**option**> tags inside the <**select**> element define the options in the list.

2) Context Menu

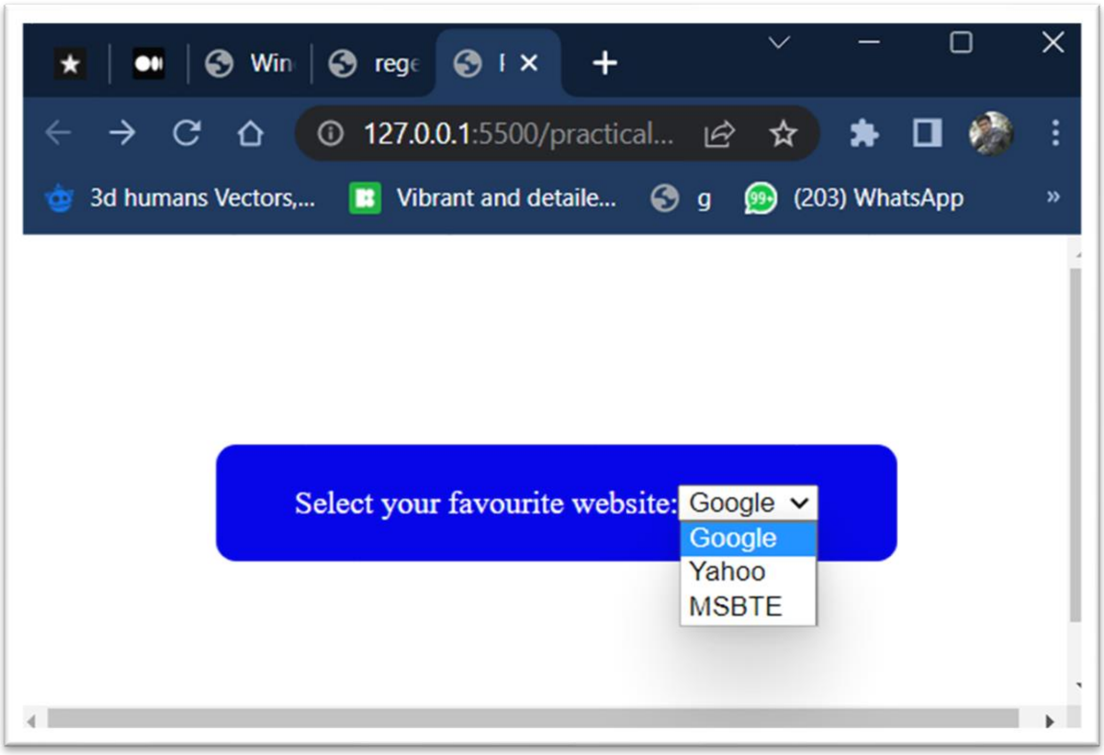
- The context menu appears on the web page when user clicks right mouse button on anywhere on the screen.
- For this we have to determine the mouse click position and set the context menu on that position.

CODE:

```
<html>
  <head>
    <title>Pulldown Menu Example</title>
    <style>
```

```
      body {
        width: 100%;
        height: 100vh;

        display: flex;
        align-items: center;
        justify-content: center;
        position: relative;
      }
      .Center {
        padding: 20px;
        width: 300px;
        height: auto;
        color: white;
        position: absolute;
        border-radius: 10px;
        display: flex;
        justify-content: center;
        background-color: rgb(0, 0, 229);
      }
    </style>
    <script>
      function displayPage(ch) {
        page = ch.options[ch.selectedIndex].value;
        if (page != "") {
          window.location = page;
        }
      }
    </script>
  </head>
  <body>
    <div class="Center">
      Select your favourite website:
      <select name="mymenu" onchange="displayPage(this)">
        <option value="https://www.google.com">Google</option>
        <option value="https://www.yahoo.com">Yahoo</option>
        <option value="https://www.msbte.org.in">MSBTE</option>
      </select>
    </div>
  </body>
</html>
```

OUTPUT:



Select your favourite website: Google

Google
Yahoo
MSBTE

Marks Obtained			Dated signature of Teacher
Process Related (15)	Product Related (35)	Total (50)	

Practical No. 12

Develop a webpage for implementing Slideshow and Banner

Theoretical Background :

Banner Advertisements

- The **banner advertisement** is the hallmark of every commercial web page.
- It is typically positioned near the top of the web page, and its purpose is to get the visitor's attention.
- All banner advertisements are in a file format such as a GIF, JPG, TIFF, or other common graphic file formats.
- Some are Flash movies that require the visitor to have a browser that includes a Flash plug-in.
- Following are the steps to insert banner advertisement in webpage.

1) Create banner advertisement using a graphics tool such as PhototShop, Paint, etc.

2) Create an <img> element in web page with height and width to display banner advertisement.

3) Build JavaScript that loads and display banner advertisements.

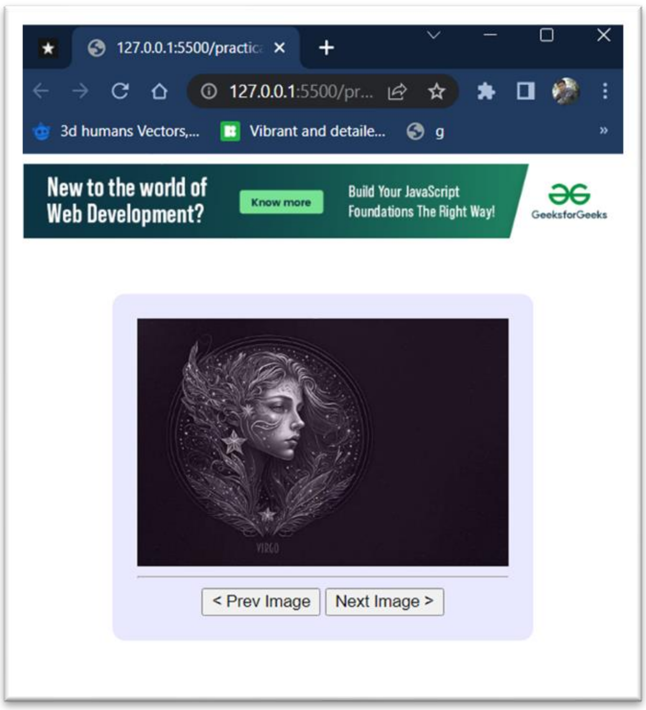
SlideShow

- A slideshow is similar in concept to a banner advertisement in that a slideshow rotates multiple images on the web page.
- A slideshow gives the visitor the ability to change the image that's displayed: the visitor can click the Forward button to see the next image and the Back button to see the previous image.
- Creating a slideshow for your web page is a straightforward process.
- The <body> tag contains an <img> tag that is used to display the image on the web page.

CODE:

```
<html>
  <head>
    <style>
      main {
        width: 100%;
        height: 80vh;

        display: flex;
        align-items: center;
        justify-content: center;
        position: relative;
      }
      .Center {
        padding: 20px;
        width: 300px;
        height: auto;
        color: white;
        position: absolute;
        border-radius: 10px;
        background-color: rgb(230, 230, 252);
      }
    </style>
  </head>
  <body>
    <center>
      <a
        href="https://practice.geeksforgeeks.org/courses/javascript?utm_source=gfg&utm_med
ium=in-article&utm_campaign=JS-foun"
      >
        <img
          width="100%"
          src="https://media.geeksforgeeks.org/wp-content/post-ads-banner/2022-05-25-23-
21-01-Web_Development_In-Article_Ad.webp"
        />
      </a>
    </center>
    <main>
      <div class="Center">
        <center>
          <img
            id="imageId"
            src="https://cdn.pixabay.com/photo/2022/11/26/11/45/virgo-7617694__340.jpg"
            width="300"
            height="200"
          />
          <br />
          <hr />
          <input type="button" value="< Prev Image" onclick="previousImage()" />
          <input type="button" value="Next Image >" onclick="nextImage()" />
        </center>
      </div>
    </main>
  </body>
  <script>
    var images = [
      "https://cdn.pixabay.com/photo/2022/11/26/11/45/virgo-7617694__340.jpg",
      "https://cdn.pixabay.com/photo/2022/11/19/15/50/holly-7602422__340.jpg",
      "https://cdn.pixabay.com/photo/2022/11/26/11/45/taurus-7617693__340.jpg",
      "https://cdn.pixabay.com/photo/2022/11/26/11/45/scorpio-7617691__340.jpg",
      "https://cdn.pixabay.com/photo/2022/11/25/10/29/cosmos-7615933__340.jpg",
    ];
    var count = 0;
    function previousImage() {
      if (count != 0) count--;
      var id = document.getElementById("imageId");
      id.src =images[count];
    }
    function nextImage() {
      if (count != 4) count++;
      var id = document.getElementById("imageId");
      id.src = images[count];
    }
  </script>
</html>
```

OUTPUT:



Marks Obtained			Dated signature of Teacher
Process Related (15)	Product Related (35)	Total (50)	