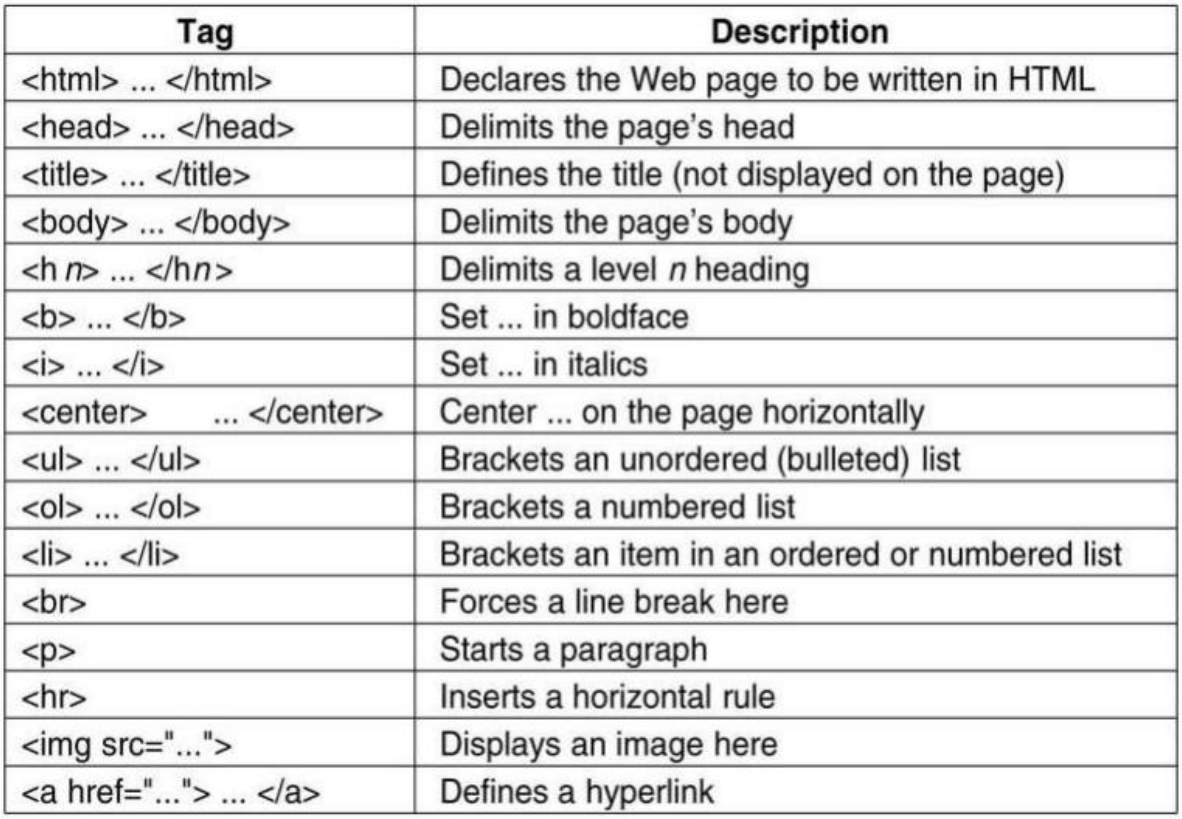
Experiment-1

Aim: Exercise on basic HTML tags.

Description:



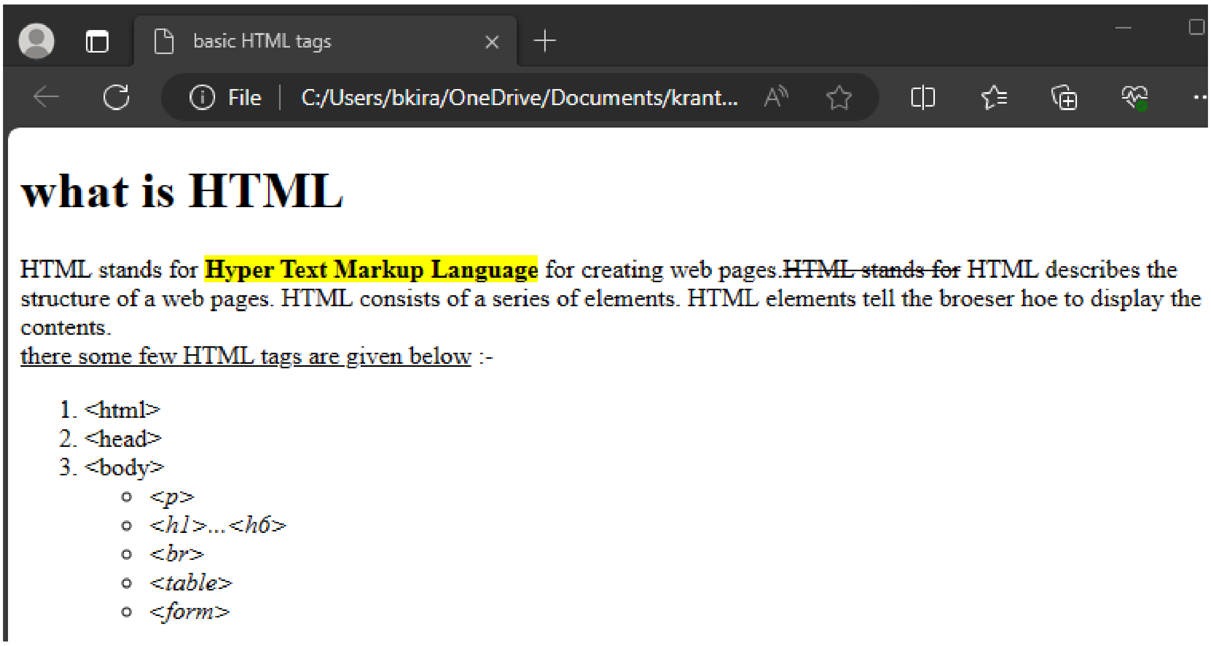
Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h *n*> ... </h*n*>	Delimits a level *n* heading
<b> ... </b>	Set ... in boldface
<i> ... </i>	Set ... in italics
<center> ... </center>	Center ... on the page horizontally
<ul> ... </ul>	Brackets an unordered (bulleted) list
<ol> ... </ol>	Brackets a numbered list
<li> ... </li>	Brackets an item in an ordered or numbered list
<br>	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a horizontal rule
<img src="...">	Displays an image here
<a href="..."> ... </a>	Defines a hyperlink

Program:

```
<!doctype html>
 <html>
  <head>
   <title>basic HTML tags</title>
  </head>
  <body>
   <h1>what is HTML</h1>
   <p>HTML stands for <b><mark>Hyper Text Markup Language</mark></b>
  for creating web pages.<strike>HTML stands for</strike> HTML describes the
  structure of a web pages. HTML consists of a series of elements. HTML
  elements tell the broeser hoe to display the contents.<br><u>there some few
  HTML tags are given below</u> :-
     <ol type="1">
       <li>&lthtml&gt</li>
```

```
        <li>&lthead&gt</li>
        <li>&ltbody&gt</li>
         <ul>
          <li><i>&ltp&gt</i></li>
          <li><i>&lth1&gt...&lth6&gt</i></li>
          <li><i>&ltbr&gt</i></li>
          <li><i>&lttable&gt</i></li>
          <li><i>&ltform&gt</i></li>
         </ul>
      </ol>
   </p>
  </body>
 </html>
```

output:



what is HTML

HTML stands for **Hyper Text Markup Language** for creating web pages.~~HTML stands for~~ HTML describes the structure of a web pages. HTML consists of a series of elements. HTML elements tell the broeser hoe to display the contents.
there some few HTML tags are given below :-

1. <html>
2. <head>
3. <body>
   - *<p>*
   - *<h1>...<h6>*
   - *
*
   - *<table>*
   - *<form>*

Result: Hence, exercise on basic HTML tags completed successfully.

Experiment-2

Aim: Design a HTML page using suitable table and attributes.

Description:

The <table> tag in HTML is used to create a table structure on a webpage. Tables are often used to present data in rows and columns. The <table> element serves as the container for the table's structure, and other related tags like <tr>, <th>,and <td> are used to define the table's content and layout.

Table Attributes

The <table> tag itself has several useful attributes to modify the table's appearance and behaviour:

1. **border**

   This attribute specifies the width of the table border. If set, it creates a visible border around the table.

2. **rowspan**

   - The **rowspan** attribute allows a cell to span multiple rows in a table.
   - The value of rowspan indicates how many rows the cell should occupy. It is typically used in the <td> or <th> elements to combine cells vertically.

3. **colspan**

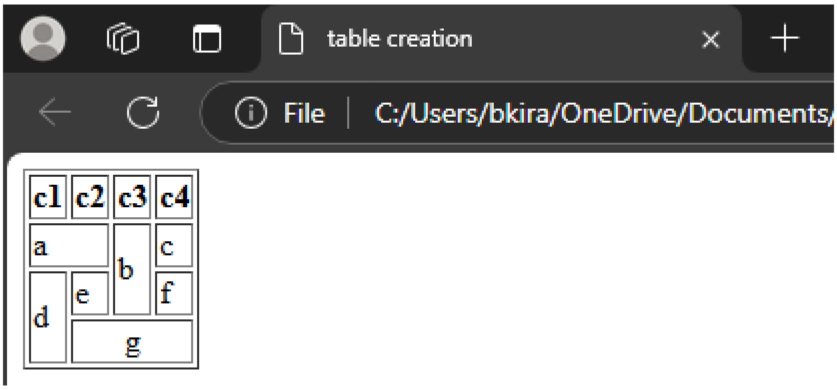
   - The **colspan** attribute allows a cell to span across multiple columns.
   - The value of colspan indicates how many columns the cell should occupy. It is also used in <td> or <th> elements to combine cells horizontally.

Program:

```
<!doctype html>
<html>
 <head>
   <title>table creation</title>
```

```
</head>
<body>
  <table border="1" >
    <tr>
      <th>c1</th>
      <th>c2</th>
      <th>c3</th>
      <th>c4</th>
    </tr>
    <tr>
      <td  colspan="2" >a</td>
      <td  rowspan="2">b</td>
      <td>c</td>
    </tr>
    <tr>
      <td  rowspan="2">d</td>
      <td>e</td>
      <td>f</td>
    </tr>
    <tr>
      <td  align="center" colspan="3" >g</td>
    </tr>
  </table>
</body>
```

output:



<table>
<tr><th>c1</th><th>c2</th><th>c3</th><th>c4</th></tr>
<tr><td colspan="2">a</td><td rowspan="2">b</td><td>c</td></tr>
<tr><td rowspan="2">d</td><td>e</td><td>f</td></tr>
<tr><td colspan="3" align="center">g</td></tr>
</table>

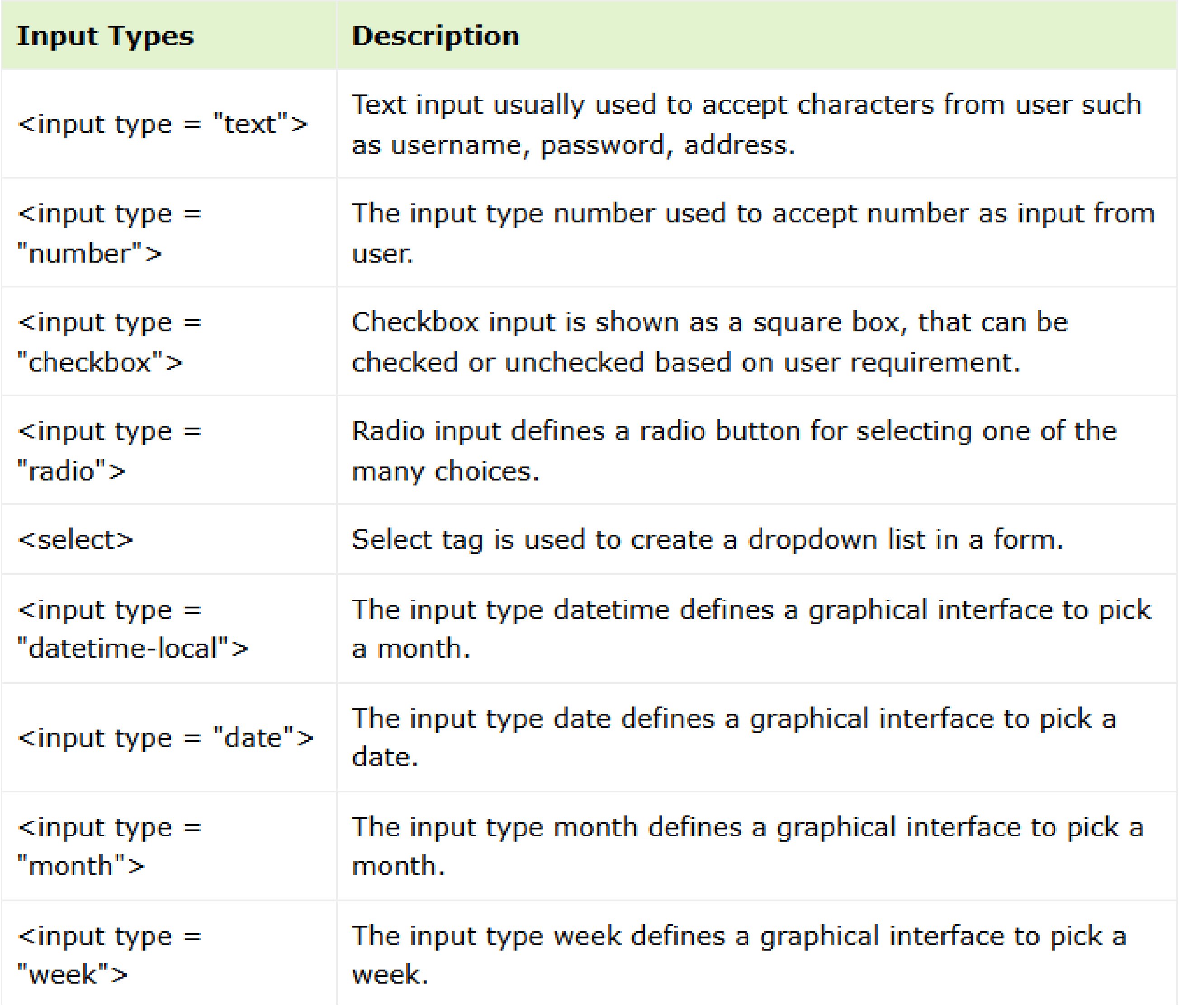
Result: Hence, design a HTML page using suitable table and attributes completed successfully.

Experiment-3

Aim: Design a HTML page with a form containing various controls.

Description:

The <form> tag in HTML is used to create an interactive form on a webpage, allowing users to submit data to a server for processing. Forms are essential for collecting input from users, such as registration details, feedback, search queries, and more. The form data can be sent to a server or processed on the client-side (depending on how the form is set up).



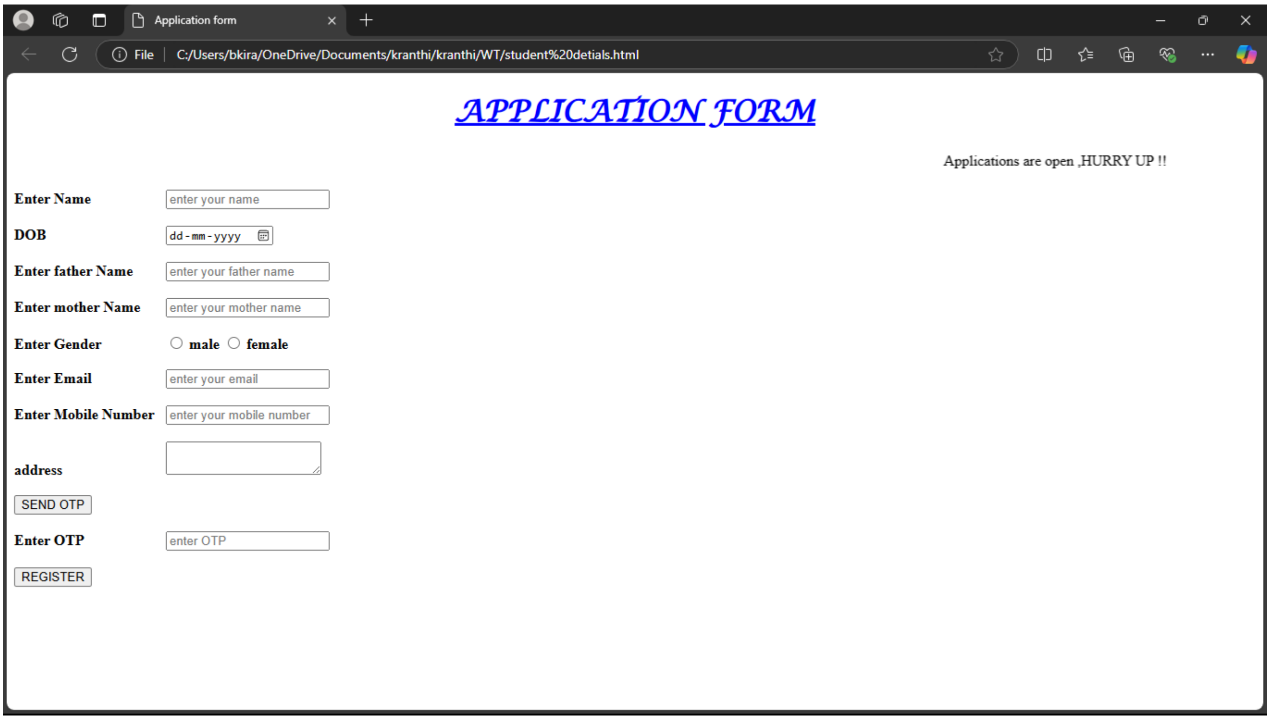
Input Types	Description
<input type = "text">	Text input usually used to accept characters from user such as username, password, address.
<input type = "number">	The input type number used to accept number as input from user.
<input type = "checkbox">	Checkbox input is shown as a square box, that can be checked or unchecked based on user requirement.
<input type = "radio">	Radio input defines a radio button for selecting one of the many choices.
<select>	Select tag is used to create a dropdown list in a form.
<input type = "datetime-local">	The input type datetime defines a graphical interface to pick a month.
<input type = "date">	The input type date defines a graphical interface to pick a date.
<input type = "month">	The input type month defines a graphical interface to pick a month.
<input type = "week">	The input type week defines a graphical interface to pick a week.

Program:

```
<!doctype html>
<html>
<head >
<title> Application form</title>
<h1 align="center" > <font  face="Lucida Calligraphy" color="BLUE"><u>
APPLICATION FORM </u></font></h1>
<marquee scrollamount="10"> Applications are open ,HURRY UP !!
</marquee>
<style type="text/css">
label
 {
   width:160px;
   display: inline-block;
 }
</style>
</head>
<body>
<DIV>
<form action="getmethod.php" method="GET">
<br>
<label for="label"><b>Enter Name</b></label>
<input type="text id="input" placeholder="enter your name"></input>
<br><br>
<label for="label"><b> DOB </b></label>
<input type="date" id="input" placeholder="enter date of birth"></input>
<br><br>
<label for="label"><b>Enter father Name</b></label>
<input type="text" id="input" placeholder="enter your father name"></input>
<br><br>
<label for="label"><b>Enter mother Name</b></label>
<input type="text" id="input" placeholder="enter your mother name"></input>
<br><br>
<label for="label"><b>Enter Gender</b></label>
<input type="radio" id="gender" > </input><b>male</b>
<input type="radio" id="gender"> </input><b>female</b>
<br><br>
<label for="label"><b>Enter Email</b></label>
```

```
<input type="email" id="input" placeholder="enter your email" ></input>
<br><br>
<label for="label"><b>Enter Mobile Number</b></label>
<input id="input" placeholder="enter your mobile number" ></input>
<br><br>
<label for="label"><b>address</b></label>
<textarea placeholder="address">  </textarea>
<br><br>
<input type="BUTTON" VALUE="SEND OTP"></input>
<br><br>
<label for="label"><b>Enter OTP</b></label>
<input id="input" placeholder="enter OTP" ></input>
<br><br>
<button>REGISTER</button>
</form>
</DIV>
</BODY>
</html>
```

output:


Application form
File
C:/Users/bkira/OneDrive/Documents/kranthi/kranthi/WT/student%20detials.html
SEND OTP
REGISTER

APPLICATION FORM

Applications are open ,HURRY UP !!

Enter Name	enter your name
DOB	dd-mm-yyyy
Enter father Name	enter your father name
Enter mother Name	enter your mother name
Enter Gender	○ **male** ○ **female**
Enter Email	enter your email
Enter Mobile Number	enter your mobile number
address	
Enter OTP	enter OTP

Result: Hence, design a HTML page with a form containing various controls completed successfully.

Experiment-4

Aim: Design a HTML page on frames.

Description:

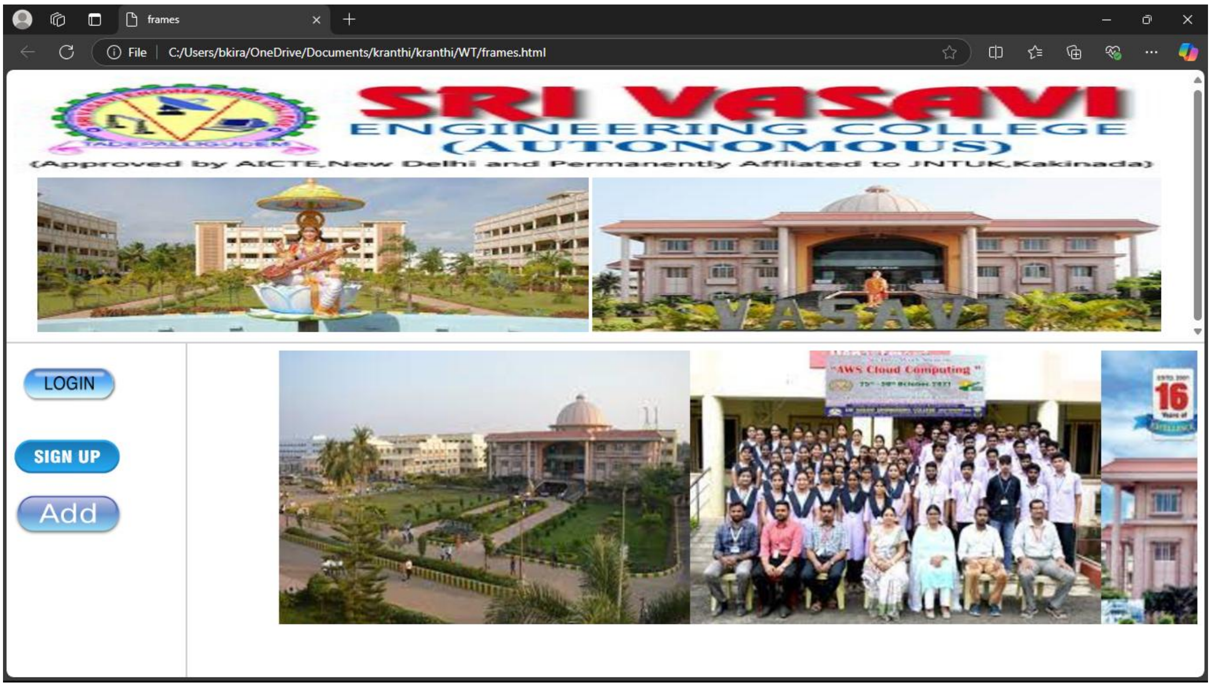
In HTML, **frames** were used to display multiple HTML documents in separate sections of the same browser window. Frames allowed web pages to be split into distinct areas, each capable of displaying different content, while keeping the same overall layout. This feature was commonly used in websites before modern responsive design and CSS layout techniques took over.

- **<frameset>**: Replaced the <body> element to define how the screen should be divided into frames.
- **<frame>**: Was used to display individual content within a frame.
- **<noframes>**: This element was used to provide alternative content for browsers that did not support frames.

Program:

```
<!doctype html>
<html>
 <head>
 <title>frames</title>
      <frameset border="2" rows="45%,*">
        <frame src="topimages.html">
      <frameset cols="15%,*">
        <frame src="hb.html">
        <frame src="home.html" name="replace">
      </frameset>
      </frameset>
 </head>
```

output:


frames
File | C:/Users/bkira/OneDrive/Documents/kranthi/kranthi/WT/frames.html
SRI VASAVI
ENGINEERING COLLEGE
(AUTONOMOUS)
(Approved by AICTE,New Delhi and Permanently Affiliated to JNTUK,Kakinada)
VASAVI
LOGIN
SIGN UP
Add
"AWS Cloud Computing"
16

Result: Hence, design a HTML page on frames completed successfully.

Experiment-5

Aim: Exercise on CSS.

Description:

CSS (Cascading Style Sheets) is a stylesheet language used to define the presentation (look and feel) of a web page written in HTML or XML. It controls the layout, colors, fonts, spacing, and other visual aspects of the elements on a webpage. By separating the content (HTML) from the presentation (CSS), CSS allows for cleaner code, easier maintenance, and better reusability.

Program:

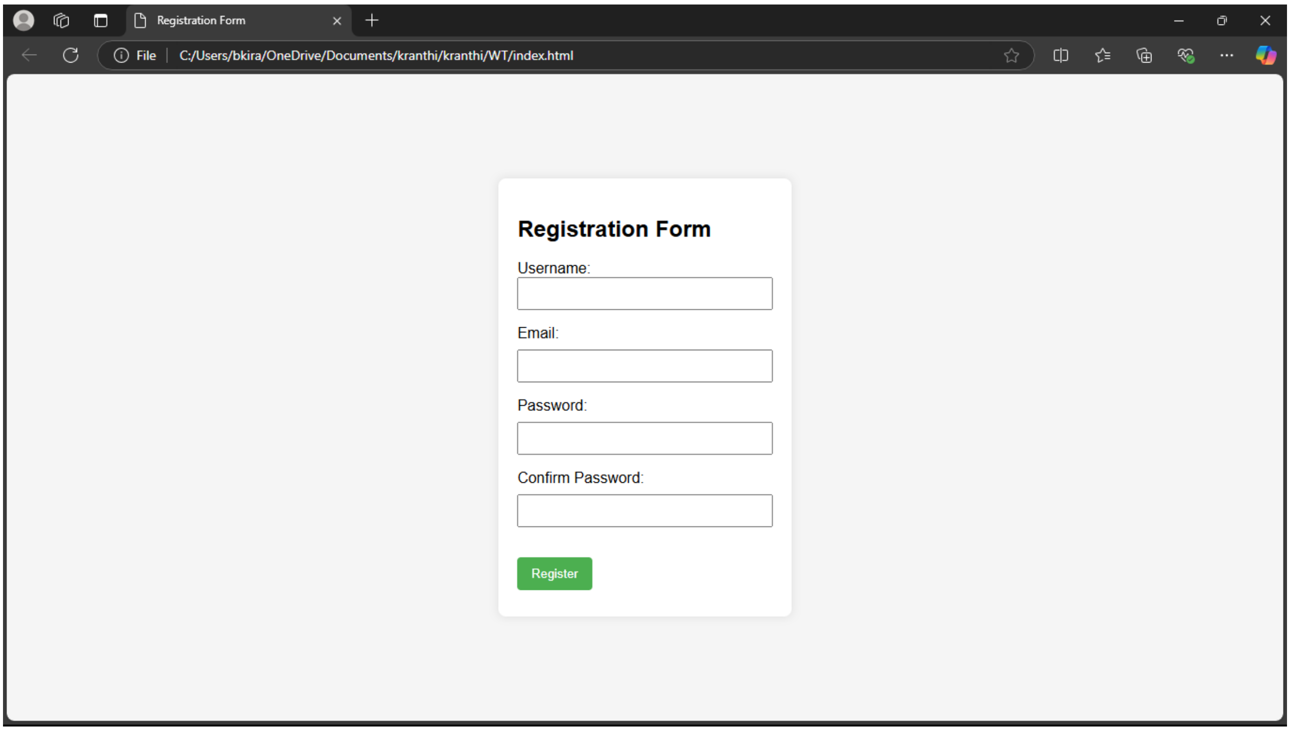
```
<!DOCTYPE html>
<html lang="en">
<head>
  <link rel="stylesheet" href="styles.css">
  <title>Registration Form</title>
</head>
<body>
  <div class="container">
    <form id="registrationForm" onsubmit="validateForm(); return false;">
      <h2>Registration Form</h2>

      <label for="username">Username:</labe0l>
      <input type="text" id="username" name="username" required>

      <label for="email">Email:</label>
      <input type="email" id="email" name="email" required>

      <label for="password">Password:</label>
      <input type="password" id="password" name="password" required>

      <label for="confirmPassword">Confirm Password:</label>
      <input type="password" id="confirmPassword" name="confirmPassword"
required>

      <p id="passwordError" class="error"></p>
      <button type="submit">Register</button>
    </form>
  </div>
</body>
</html>
```

CSS file:- styles.css

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  background-color: #f5f5f5;
}

.container {
  background-color: #fff;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

form {
  max-width: 400px;
  margin: 0 auto;
}

label {
  display: block;
  margin-bottom: 8px;
}
input {
  width: 100%;
  padding: 8px;
  margin-bottom: 16px;
  box-sizing: border-box;
}
```

```
button {
  background-color: #4caf50;
  color: #fff;
  padding: 10px 15px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

button:hover {
  background-color: #45a049;
}

.error {
  color: red;
}
```

output:


Registration Form
File
C:/Users/bkira/OneDrive/Documents/kranthi/kranthi/WT/index.html

Registration Form

Username:

Email:

Password:

Confirm Password:

Result: Hence, exercise on CSS completed successfully.

Experiment-6

Aim: Exercise on designing a XML document.

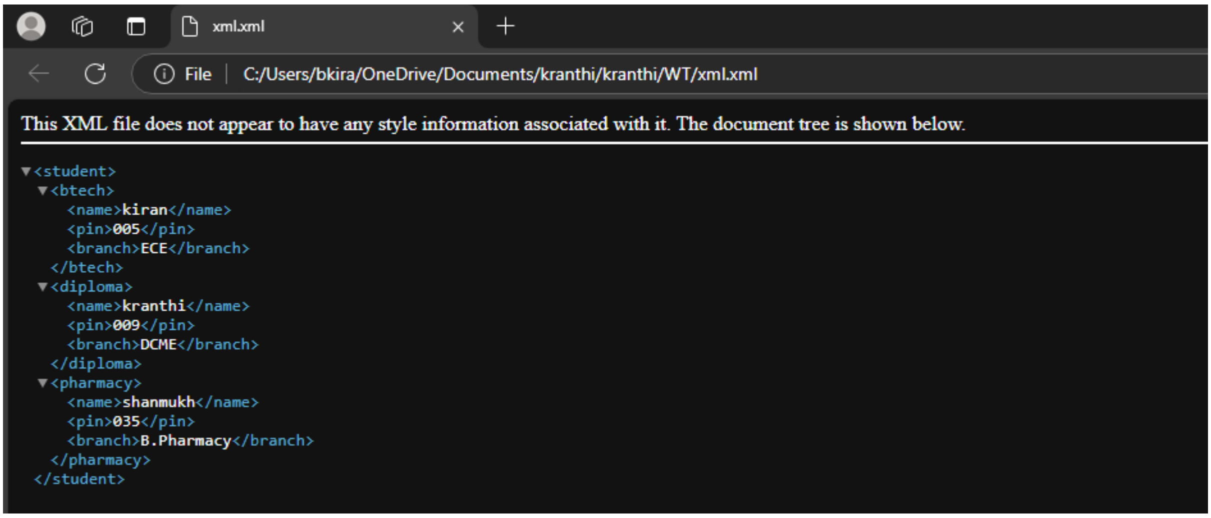
Description:

An **XML (Extensible Markup Language) document** is a text-based data format designed to store and transport data in a structured and self-descriptive way. XML is both human-readable and machine-readable, which makes it ideal for representing data across different systems, applications, and platforms. It allows you to define custom tags, making it flexible enough for a wide range of uses, from storing configuration data to exchanging information between web services.

Program:

```
<?xml version="1.0" encoding="utf-8"?>
<student>
 <btech>
  <name>kiran</name>
  <pin>005</pin>
  <branch>ECE</branch>
 </btech>
 <diploma>
  <name>kranthi</name>
  <pin>009</pin>
  <branch>DCME</branch>
 </diploma>
 <pharmacy>
  <name>shanmukh</name>
  <pin>035</pin>
  <branch>B.Pharmacy</branch>
 </pharmacy>
</student>
```

output:



xml.xml
File | C:/Users/bkira/OneDrive/Documents/kranthi/kranthi/WT/xml.xml

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<student>
  ▼<btech>
     <name>kiran</name>
     <pin>005</pin>
     <branch>ECE</branch>
   </btech>
  ▼<diploma>
     <name>kranthi</name>
     <pin>009</pin>
     <branch>DCME</branch>
   </diploma>
  ▼<pharmacy>
     <name>shanmukh</name>
     <pin>035</pin>
     <branch>B.Pharmacy</branch>
   </pharmacy>
 </student>
```

Result: Hence, exercise on designing a XML document completed successfully.

Experiment-7

Aim: Exercise on JavaScript functions.

Description:

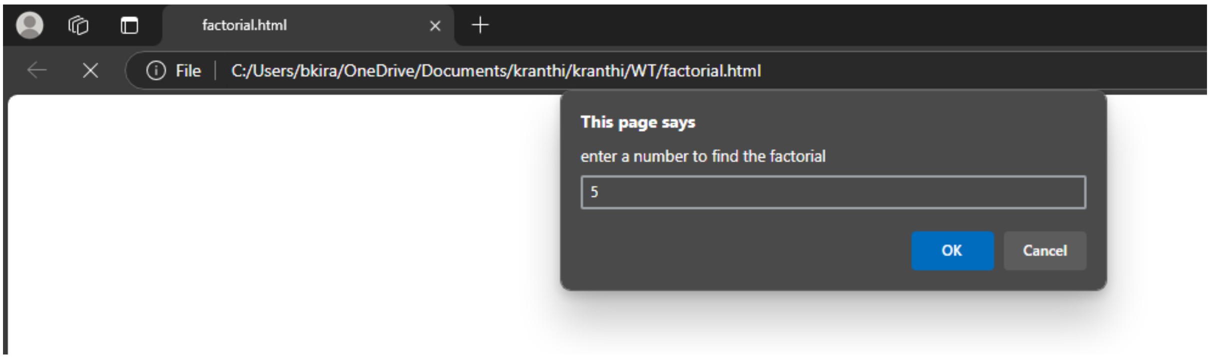
In JavaScript, a **function** is a block of reusable code that performs a specific task. Functions allow you to organize and modularize code, making it easier to maintain, debug, and reuse. They are one of the fundamental building blocks of JavaScript programming.

A JavaScript function can take inputs (parameters), execute code, and return an output (value). Functions can be used to perform calculations, manipulate data, handle events, and much more.

Program:

```
<html>
 <head>
  <script  type="text/javascript">
   function factorial(n)
    {
     let fact=1;
     let i=n;
     for(i=n;i>0;i--)
      {
       fact=fact*i;
      }
     return fact;
    }
   let num=Number(prompt("enter a number to find the factorial"));
   fact=factorial(num);
   document.write("factorial of "+num+" is "+fact);
  </script>
 </head>
</html>
```

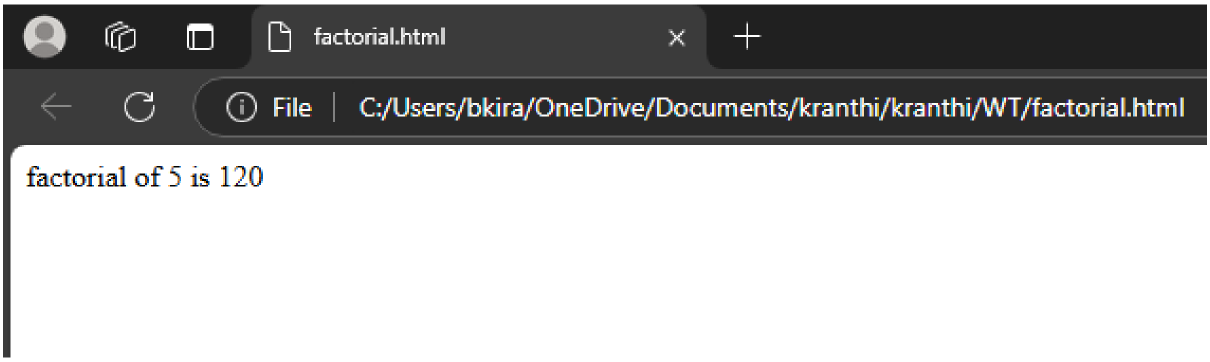
output:


factorial.html
File
C:/Users/bkira/OneDrive/Documents/kranthi/kranthi/WT/factorial.html
OK
Cancel

This page says

enter a number to find the factorial

5


factorial.html
File
C:/Users/bkira/OneDrive/Documents/kranthi/kranthi/WT/factorial.html

factorial of 5 is 120

Result: Hence, exercise on JavaScript functions completed successfully.

Experiment-8

Aim: Exercise on JavaScript arrays.

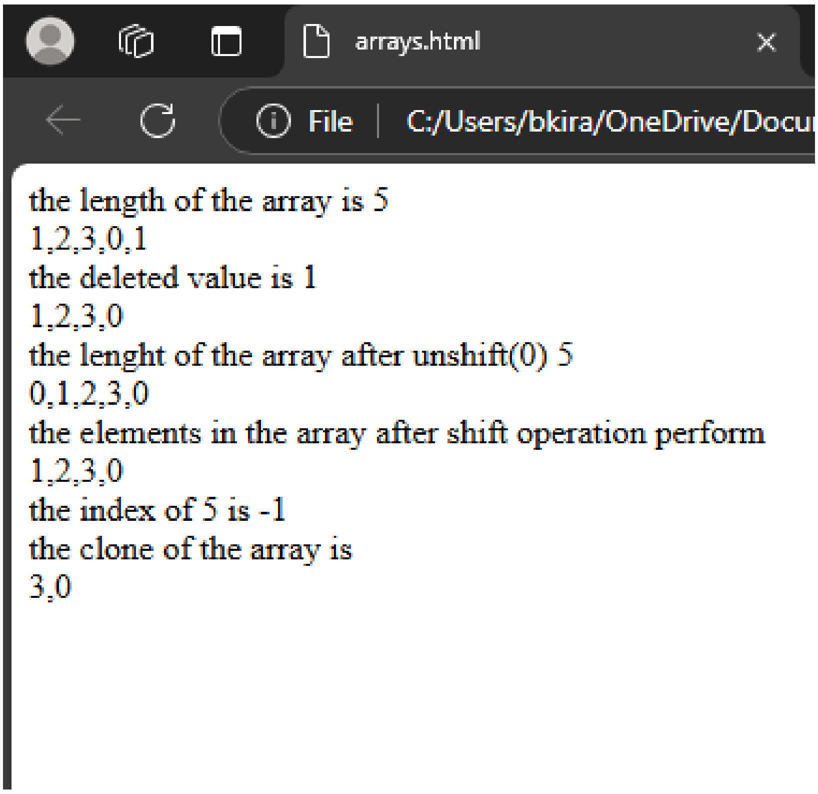
Description:

In JavaScript, an **array** is a special type of object that is used to store multiple values in a single variable. Arrays allow you to group related data together, which makes it easier to work with collections of items such as lists, sequences, or collections of objects. Arrays are one of the most commonly used data structures in JavaScript.

Program:

```
<script type="text/javascript">
 let n=Number(prompt("enter number of element in the array"))
 let arr=[];
 let i=0;
  for(i=0;i<n;i++)
   {
    arr.push(Number(prompt("enter the value into array")));
   }
  document.write("the length of the array is "+arr.length+"<br>")
  document.write(arr+"<br>");
  document.write("the deleted value is "+arr.pop()+"<br>");
  document.write(arr+"<br>");
  let value=Number(prompt("enter the value to unshift "));
  arr.unshift(value);
  document.write("the lenght of the array after unshift("+value+")
"+arr.length+"<br>");
  document.write(arr+"<br>");
  arr.shift();
  document.write("the elements in the array after shift operation perform<br>");
  document.write(arr+"<br>");
  let f=Number(prompt("enter value to find "));
  document.write("the index of "+f+" is "+arr.indexOf(f));
  let b=Number(prompt("enter begining to create clone of the array "));
  let e=Number(prompt("enter ending to create clone of the array "));
  let array=arr.slice(b,e);
  document.write("<br>the clone of the array is<br>");
  document.write(array);
</script>
```

output:



```
the length of the array is 5
1,2,3,0,1
the deleted value is 1
1,2,3,0
the lenght of the array after unshift(0) 5
0,1,2,3,0
the elements in the array after shift operation perform
1,2,3,0
the index of 5 is -1
the clone of the array is
3,0
```

Result: Hence, exercise on JavaScript arrays completed successfully.

Experiment-9

Aim: Write a program on mouse events using jquery.

Description:

Mouse events in jQuery are a set of methods that allow you to handle user interactions using a mouse. These events are triggered when a user interacts with HTML elements through actions like clicking, hovering, or dragging. jQuery simplifies handling these events by providing an easy-to-use API.

Common Mouse Events in jQuery

1. click

- Triggered when a mouse button is clicked on an element.

2. dblclick

- Triggered when an element is double-clicked.

3. mousedown

- Triggered when a mouse button is pressed down on an element.

4. mouseup

- Triggered when a mouse button is released after being pressed down.

5. mousemove

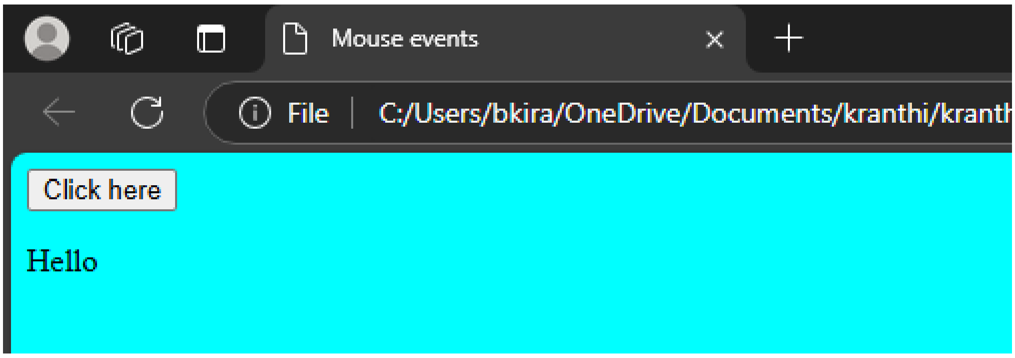
- Triggered when the mouse pointer moves over an element.

Program:

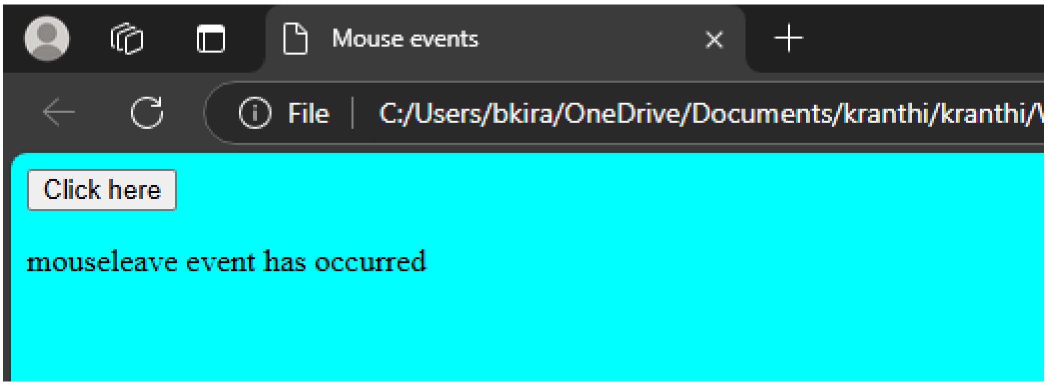
```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<title>Mouse events</title>
</head>
<body bgcolor="cyan">
<button id="btn">Click here</button>
```

```
<p id="key">Hello</p>
<script>
$("document").ready(function () {
$("#btn").click(clk);
$("#btn").dblclick(dou);
$("#btn").contextmenu(right);
$("#btn").mouseenter(enter);
$("#btn").mouseleave(leave);
function enter() {
$("#key").text( "mouseenter event has occurred");
}
function leave() {
$("#key").text( "mouseleave event has occurred");
}
function clk() {
$("#key").hide();
}
function dou() {
$("#key").show();
}
function right() {
$("#key").text( "Right click event has occurred");
}
});
</script>
</body>
</html>
```

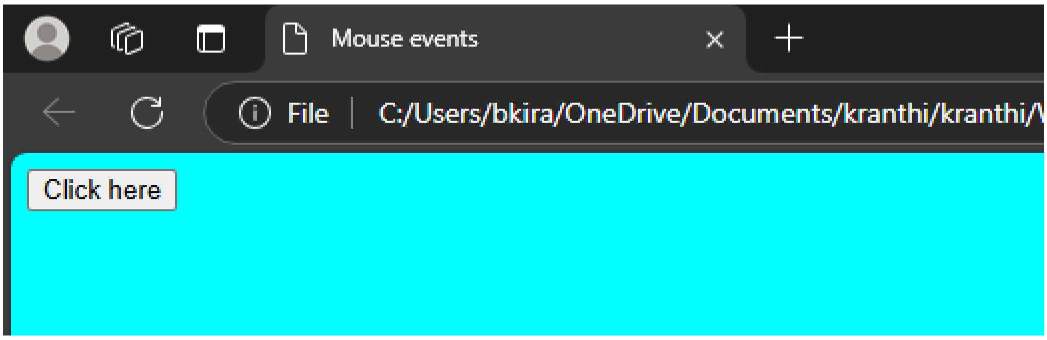
output:



Hello



mouseleave event has occurred



Result: Hence, exercise on mouse events using jquery completed successfully.

Experiment-10

Aim: Design a webpage to Design a webpage to apply the Effects of jquery to HTML elements.

Description:

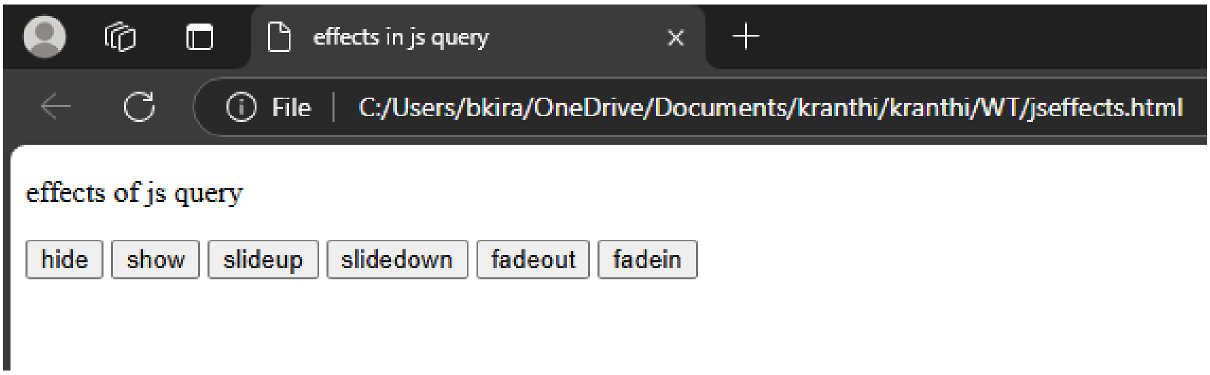
jQuery provides a variety of built-in **effects** that allow you to create animations and transitions to enhance the user interface. These effects are simple to implement, making it easy to create visually appealing interactions on web pages without extensive coding.

- **hide()**: Hides the selected element.
- **show()**: Displays the selected hidden element.
- **toggle()**: Toggles between show and hide.
- **fadeIn(speed)**: Gradually makes an element visible.
- **fadeOut(speed)**: Gradually hides an element.
- **slideDown(speed)**: Slides an element down to make it visible.
- **slideUp(speed)**: Slides an element up to hide it.

Program:

```
<!doctype html>
<html>
 <head>
  <title>effects in js query</title>
  <script src="jquery.js"></script>
  <script>
   $(document).ready(function ()
    {
          $("#b1").click(function()
           {
            $("p").hide();
           });
          $("#b2").click(function()
           {
            $("p").show();
           });
          $("#b3").click(function()
           {
           $("p").slideUp();
           });
          $("#b4").click(function()
           {
           $("p").slideDown();
           });
          $("#b5").click(function()
           {
           $("p").fadeOut(5000);
           });
          $("#b6").click(function()
           {
           $("p").fadeIn(5000);
           });
          });
  </script>
 </head>
 <body>
  <p>effects of js query</p>
  <button id="b1" >hide</button>
  <button id="b2" >show</button>
  <button id="b3" >slideup</button>
  <button id="b4" >slidedown</button>
  <button id="b5" >fadeout</button>
  <button id="b6" >fadein</button>
 </body>
</html>
```

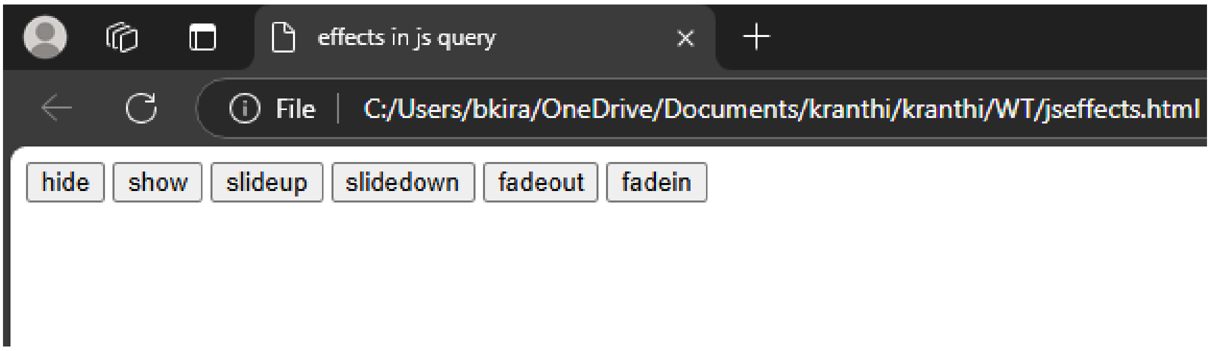
output:

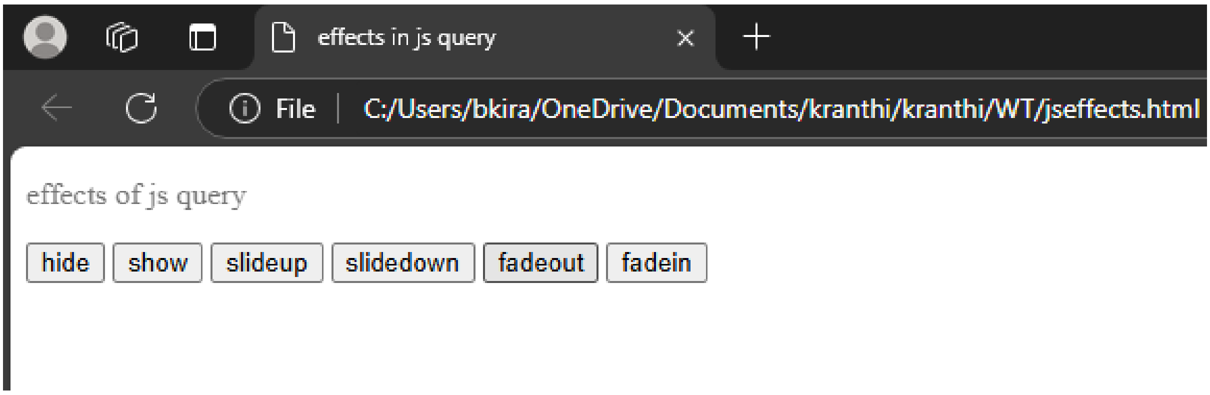


effects in js query

File | C:/Users/bkira/OneDrive/Documents/kranthi/kranthi/WT/jseffects.html

effects of js query


effects in js query
File
C:/Users/bkira/OneDrive/Documents/kranthi/kranthi/WT/jseffects.html
hide
show
slideup
slidedown
fadeout
fadein


effects in js query
File
C:/Users/bkira/OneDrive/Documents/kranthi/kranthi/WT/jseffects.html
hide
show
slideup
slidedown
fadeout
fadein

effects of js query

Result: Hence, design a webpage to Design a webpage to apply the Effects of jquery to HTML elements completed successfully.

Experiment-11

Aim: Exercise on changing background colour using css() function in jquery.

Description:

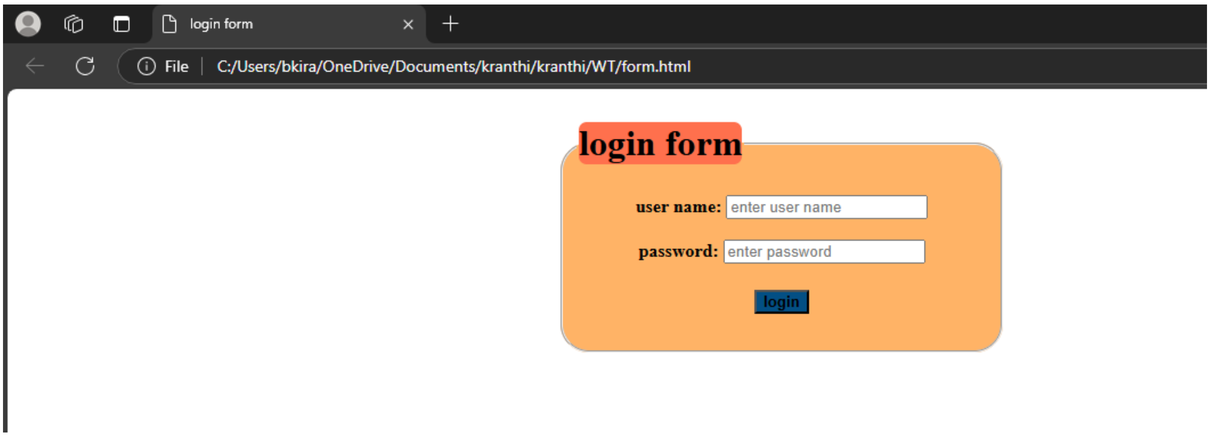
In jQuery, **functions** are reusable blocks of code that execute specific tasks when invoked. Functions in jQuery are similar to JavaScript functions but are often used in conjunction with jQuery methods to manipulate HTML elements, handle events, or create effects.

Program:

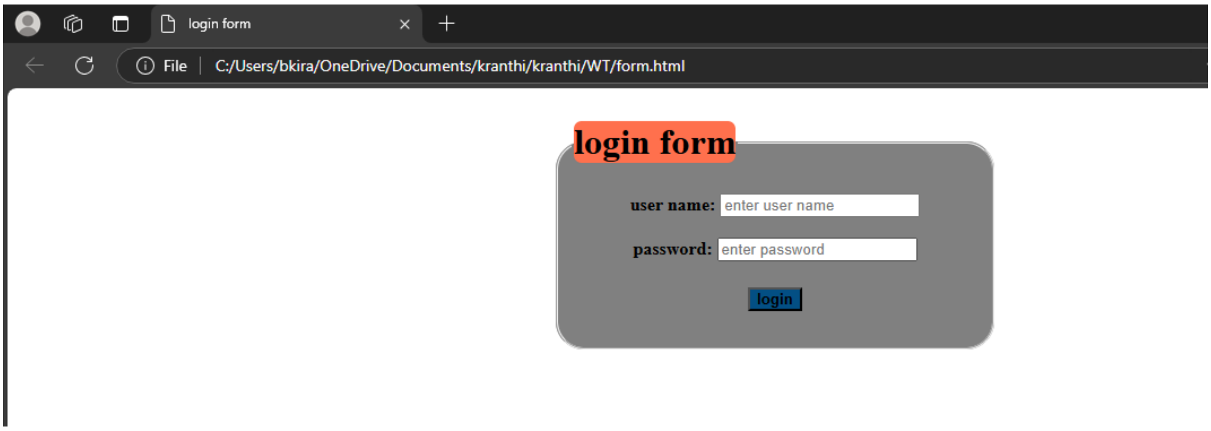
```
<!doctype html>
<html>
 <head>
   <title>login form</title>
   <link href="formbg.css" type="text/css" rel="stylesheet">
   <style type="text/css">
    #login
     {
       border-radius: 25px;
       background-color:#ffb366
     }
   </style>
   <script src="jquery.js"></script>
   <script>
     $(document).ready( function()
            {
             $("#login").mouseenter(enter);
          function enter()
                     {
                      $("#login").css( "background-color",'gray');
                     }
           $("#login").mouseleave(leave);
          function leave()
                     {
                      $("#login").css( "background-color",'#ffb366');
```

```
                    }
            });
    </script>
  </head>
  <body>
    <div id="formbg">
     <form action="/action_page.php" method="get" align="center">
       <fieldset id="login">
         <legend align="left"> <h1 style="background-color:#ff704d;border-
radius:7px;">login form</h1></legend>
          <b><label for="uname">user name: </label></b>
          <input type="text" id="uname" placeholder="enter user name">
          <br><br>
          <b><label for="pwd">password: </label></b>
          <input type="text" id="pwd" placeholder="enter password">
          <br>
          <h5> <input type="button" value="login" style="background-
color:#034f84;font-weight:bold"></h5>
        </fieldset><br>
     </form>
    </div>
  </body>
</html>
```

output:


login form
C:/Users/bkira/OneDrive/Documents/kranthi/kranthi/WT/form.html
login form
user name: enter user name
password: enter password
login

when mouse entered to the login form:-



Result: Hence, exercise on changing background colour using css() function in jquery completed successfully.

Experiment-12

Aim: Write a JavaScript program using Responsive Slides jquery plugin-(download from responsiveslides.com).

Description:

The **Responsive Slides** jQuery plugin is a lightweight, user-friendly, and feature-rich solution for creating responsive, mobile-friendly sliders and carousels. This plugin is ideal for displaying image slideshows, content slides, or featured images on a webpage while ensuring compatibility across devices.

Program:

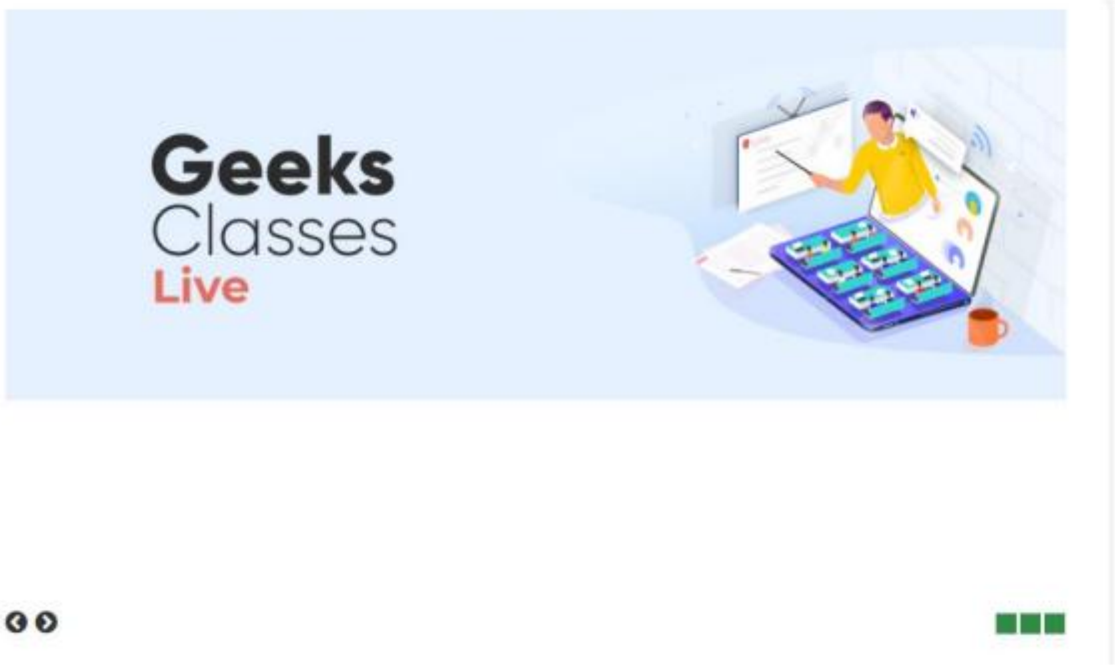
```
<!Doctype html>
<html>
<head>
<meta charset="utf-8">
<title>SlidesJS Example</title>
<meta name="description"
content="SlidesJS is a simple slideshow plugin for jQuery. ">
<!-- SlidesJS Required (if responsive):
Sets the page width to the device width. -->
<meta name="viewport"
content="width=device-width">
<!-- End SlidesJS Required -->
<!-- CSS for slidesjs.com example -->
<link rel="stylesheet"
href=
"https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">
<!-- End CSS for slidesjs.com example -->
<!-- SlidesJS Optional: If you'd like to use this design -->
<style>
#slides {
display: none
}
#slides .slidesjs-navigation {
margin-top: 3px;
}
```

```
#slides .slidesjs-previous {
margin-right: 5px;
float: left;
}
#slides .slidesjs-next {
margin-right: 5px;
float: left;
}
.slidesjs-pagination {
margin: 6px 0 0;
float: right;
list-style: none;
}
.slidesjs-pagination li {
float: left;
margin: 0 1px;
}
.slidesjs-pagination li a {
display: block;
width: 13px;
height: 0;
padding-top: 13px;
background-image: url(
https://media.geeksforgeeks.org/wp-
content/uploads/20201213110552/logo.png);
background-position: 0 0;
float: left;
overflow: hidden;
}
.slidesjs-pagination li a.active,
.slidesjs-pagination li a:hover.active {
background-position: 0 -13px
}
.slidesjs-pagination li a:hover {
background-position: 0 -26px
}
#slides a:link,
#slides a:visited {
color: #333
```

```
}
#slides a:hover,
#slides a:active {
color: #9e2020
}
.navbar {
overflow: hidden
}
</style>
<!-- SlidesJS Required: These styles are required
if you'd like a responsive slideshow -->
<style>
#slides {
display: none
}
.container {
margin: 0 auto
}
/* For tablets & smart phones */
@media (max-width: 767px) {
body {
padding-left: 20px;
padding-right: 20px;
}
.container {
width: auto
}
}
/* For smartphones */
@media (max-width: 480px) {
.container {
width: auto
}
}
/* For smaller displays like laptops */
@media (min-width: 768px) and (max-width: 979px) {
.container {
width: 724px
}
```

```
}
/* For larger displays */
@media (min-width: 1200px) {
.container {
width: 1170px
}
}
</style>
<!-- SlidesJS Required: -->
</head>
<body>
<!—Slides JS Required: Start Slides -->
<!-- The container is used to define the width of the slideshow -->
<div class="container">
<div id="slides">
<img src= "image_path" alt="">
<img src= " image_path " alt="" style="size:20px">
<img src= " image_path " alt="">
<a href="#" class="slidesjs-next slidesjs-navigation">
<i class="fa fa-chevron-circle-left"></i></a>
<a href="#" class="slidesjs-previous slidesjs-navigation">
<i class="fa fa-chevron-circle-right"></i></a>
</div>
</div>
<!-- End SlidesJS Required: Start Slides -->
<!-- SlidesJS Required: Link to jQuery -->
<script src="http://code.jquery.com/jquery-1.9.1.min.js"> </script>
<!-- End SlidesJS Required -->
<!-- SlidesJS Required: Link to jquery.slides.js -->
<script src= "https://cdnjs.cloudflare.com/ajax/libs/slidesjs/3.0/jquery.slides.js">
</script>
<!-- End SlidesJS Required -->
<!-- SlidesJS Required: Initialize SlidesJS with a jQuery doc ready -->
<script>
$(function() {
$('#slides').slidesjs({
width: 940,
height: 528,
navigation: false
});
});
</script>
<!-- End SlidesJS Required -->
</body>
</html>
```

output:



Result: Hence, to write a JavaScript program using Responsive Slides jquery plugin.

Experiment-13

Aim: Install the following on local machine:

- apache web server
- mysql
- php and configure it to work with apche web server and mysql

Description:

1. Install Apache Web Server

Windows:

1. Download the Apache HTTP Server from the Apache Lounge.
2. Extract the downloaded ZIP file to a directory (e.g., C:\Apache24).
3. Open the conf folder inside the Apache directory and edit the httpd.conf file:
   - Look for the line starting with #ServerName and replace it with:
   - ServerName localhost:80
4. Install Apache as a Windows service:
   - Open a command prompt as Administrator.
   - Navigate to the bin folder of Apache (cd C:\Apache24\bin) and run:
   - httpd -k install
5. Start the Apache service:
   - Run:
   - httpd -k start
   - Alternatively, start it through the Windows Services app.

2. Install MySQL

Windows:

1. Download MySQL Community Server from the official website.
2. Run the installer and select **Server Only** or **Developer Default** during setup.
3. Configure MySQL:
   - Set the root password during installation.
   - Choose the port (default: 3306).
4. Start MySQL Service:
   - Open the MySQL Workbench to connect and manage the database.

3. Install PHP

Windows:

1. Download PHP from the official website.
   - Choose the **Thread Safe** version for Apache compatibility.
2. Extract PHP to a folder (e.g., C:\php).
3. Configure Apache to work with PHP:
   - Open the httpd.conf file in the Apache conf folder.
   - Add these lines at the end of the file:
   - LoadModule php_module "C:/php/php8apache2_4.dll"
   - AddType application/x-httpd-php .php
   - PHPIniDir "C:/php"
   - Restart Apache.

4. Configure PHP to Work with Apache and MySQL

1. Verify PHP is properly loaded by Apache:
   - For Windows, check httpd.conf for the correct LoadModule directive.
   - For Linux, confirm the libapache2-mod-php package is installed.

2. Connect PHP with MySQL:
   - Install the PHP MySQL extension if not already installed:
   - sudo apt install php-mysql
   - On Windows, ensure php.ini has the following line uncommented:
   - extension=mysqli

5. Test the Setup

Create a test PHP file in the Apache root directory:

Windows: C:\Apache24\htdocs\test.php

Linux: /var/www/html/test.php

```
<?php
$conn = new mysqli("localhost", "root", "your_password", "test_db");
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error); }
echo "Connected successfully to MySQL!";
?>
```

Open the file in your browser: Visit http://localhost/test.php.

Result: Hence, Installation process completed successfully.

Experiment-14

Aim: Exercise on PHP arrays.

Description:

An **array** in PHP is a data structure that allows you to store multiple values in a single variable. Arrays are a fundamental part of PHP and are versatile, enabling you to organize and manipulate data efficiently.

PHP supports three types of arrays:

1. **Indexed Array**:
   - Contains numeric keys (starting from 0 by default).
   - Keys can be assigned explicitly.
2. **Associative Array**:
   - Uses named keys instead of numeric keys.
3. **Multidimensional Array**:
   - An array containing one or more arrays.
   - Useful for representing complex data structures like tables.

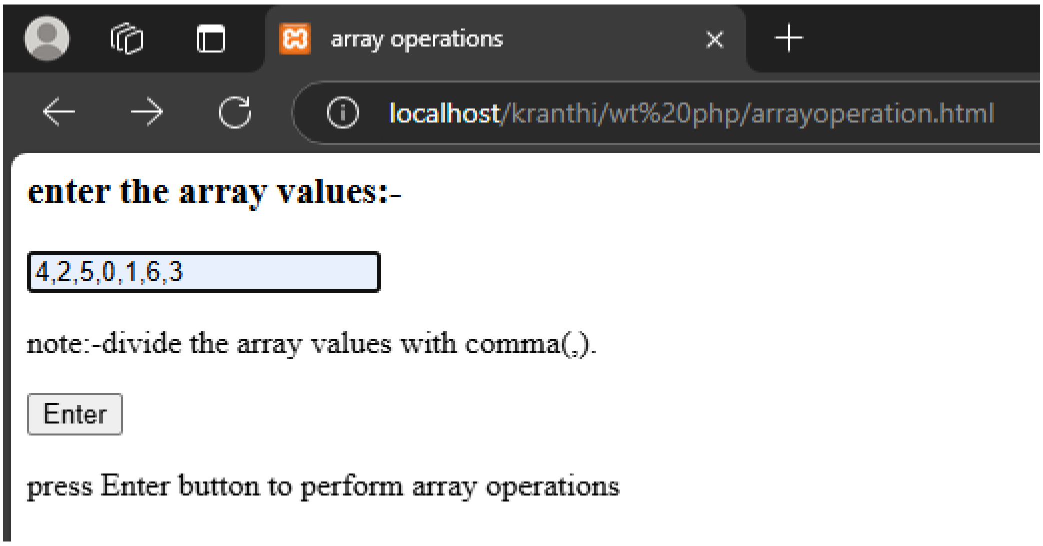
Program:

```
<html>
 <head>
  <title>array operations</title>
 </head>
 <body>
  <form action="arrayoperations.php" method="post">
   <h3><b>enter the array values:-</b></h3>
   <input type="text" name="elements"/>
   <p>note:-divide the array values with comma(,).</p>
   <input type="submit" value="Enter"/><br>
   <p>press Enter button to perform array operations</p>
  </form>
 </body>
</html>
```

PHP file:-

```
<?php
 $a=explode(',',$_POST["elements"]);
 $n=count($a);
 $array=implode(',',$a);
 $min=$a[0];
 $max=$a[0];
   for($i=0;$i<$n;$i++)
    {
      if($min >= $a[$i])
       {
        $min = $a[$i];
       }
      if($max <= $a[$i])
       {
        $max = $a[$i];
       }
     for($j=0;$j<$n-$i-1;$j++)
      {
        if($a[$j] > $a[$j+1])
         {
          $temp = $a[$j] ;
          $a[$j] = $a[$j+1] ;
          $a[$j+1] = $temp ;
         }
      }
    }
$arr=implode(',',$a);
echo"<b>Given Array:-</b> <input type=text value='$array'>";
echo"<br><br><b>sorted Array:-</b> <input type=text value='$arr'>";
echo"<br><br><b>minimum value in Array:-</b> <input type=text
value='$min'>";
echo"<br><br><b>maximum value in Array:-</b> <input type=text
value='$max'>";
?>
```

output:



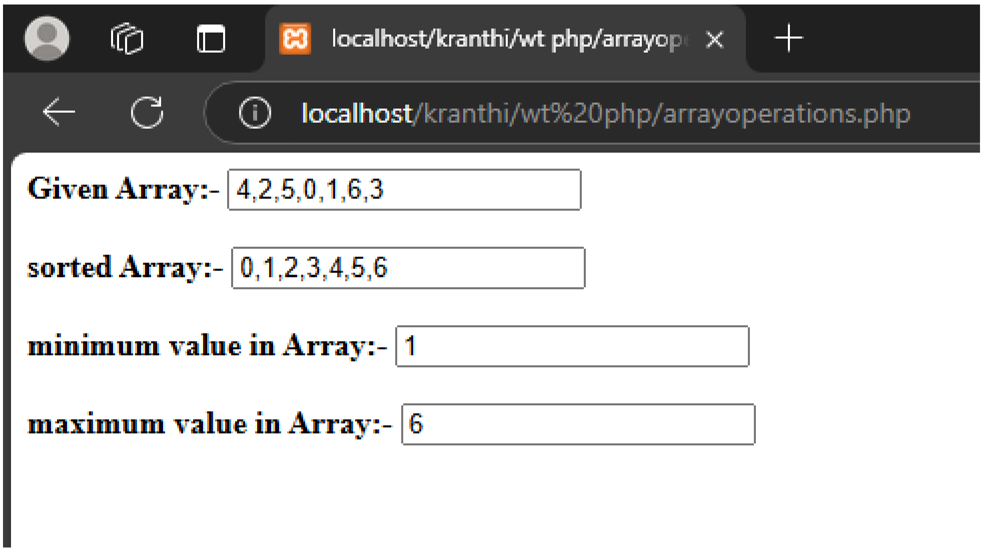
enter the array values:-

4,2,5,0,1,6,3

note:-divide the array values with comma(,).

Enter

press Enter button to perform array operations


localhost/kranthi/wt php/arrayop
localhost/kranthi/wt%20php/arrayoperations.php

Given Array:- 4,2,5,0,1,6,3

sorted Array:- 0,1,2,3,4,5,6

minimum value in Array:- 1

maximum value in Array:- 6

Result: Hence, exercise on PHP arrays completed successfully.

Experiment-15

Aim: Design a form and access the elements of form using PHP.

Description:

Forms are essential for collecting user input in web applications. PHP can handle the submitted form data for processing, such as saving it to a database or performing calculations.

1. **Create an HTML Form**
   - The form collects user input through input fields like text boxes, radio buttons, checkboxes, and dropdowns.
   - The action attribute specifies the PHP script to process the form.
   - The method attribute defines how data is sent to the server (GET or POST).
2. **Access Form Elements in PHP**
   - Use PHP's superglobals ($_GET, $_POST, or $_REQUEST) to retrieve the form data.

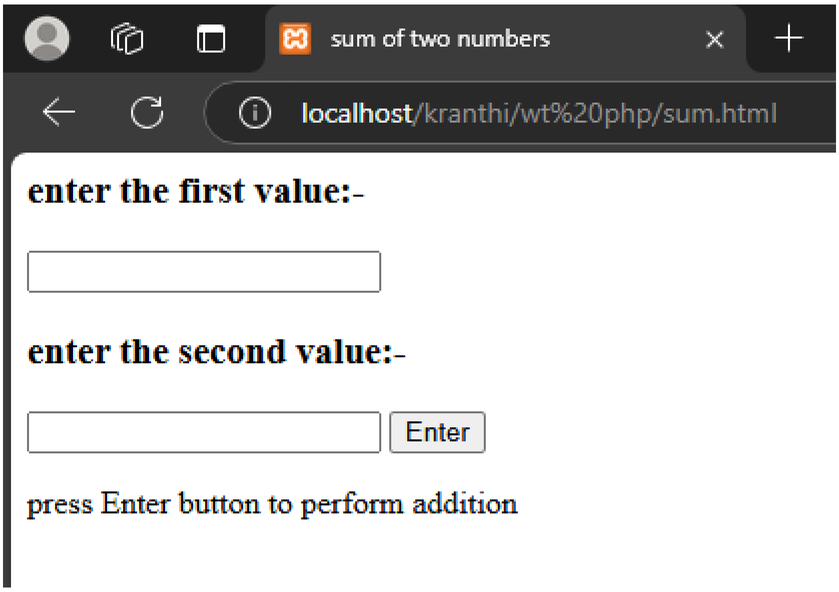
Program:

```
<html>
 <head>
  <title>sum of two numbers</title>
 </head>
 <body>
  <form action="sum.php" method="post">
   <h3><b>enter the first value:-</b></h3>
   <input type="number" name="val1"/>
   <h3><b>enter the second value:-</b></h3>
   <input type="number" name="val2"/>
   <input type="submit" value="Enter"/><br>
   <p>press Enter button to perform addition</p>
  </form>
 </body>
</html>
```

PHP file:-

```
<?php
 $n1=$_POST["val1"];
 $n2=$_POST["val2"];
 $sum=$n1+$n2;
 echo"<h2>sum of ".$n1." and ".$n2." is ".$sum."</h2>";
?>
```

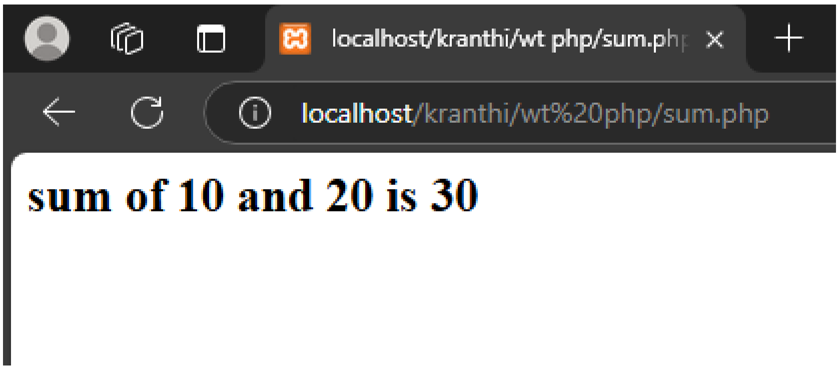
output:



enter the first value:-

enter the second value:-

press Enter button to perform addition


localhost/kranthi/wt php/sum.php
localhost/kranthi/wt%20php/sum.php

sum of 10 and 20 is 30

Result: Hence, design a form and access the elements of form using PHP completed successfully.

Experiment-16

Aim: Write PHP program to perform various operations on a database table using functions.

Description:

Database operations like inserting, updating, retrieving, and deleting data can be efficiently handled using PHP. Functions can encapsulate these operations, making the code reusable and modular.

Steps for Database Operations

1. **Establish a Connection to the Database**:
   - Use PHP's mysqli or PDO extension to connect to the database.
2. **Create a Database Table**:
   - Ensure a table is available for performing operations.
3. **Define Functions for CRUD Operations**:
   - **C**reate: Insert data into the table.
   - **R**ead: Fetch data from the table.
   - **U**pdate: Modify existing data.
   - **D**elete: Remove records.
4. **Close the Database Connection**:
   - Properly close the connection after operations.

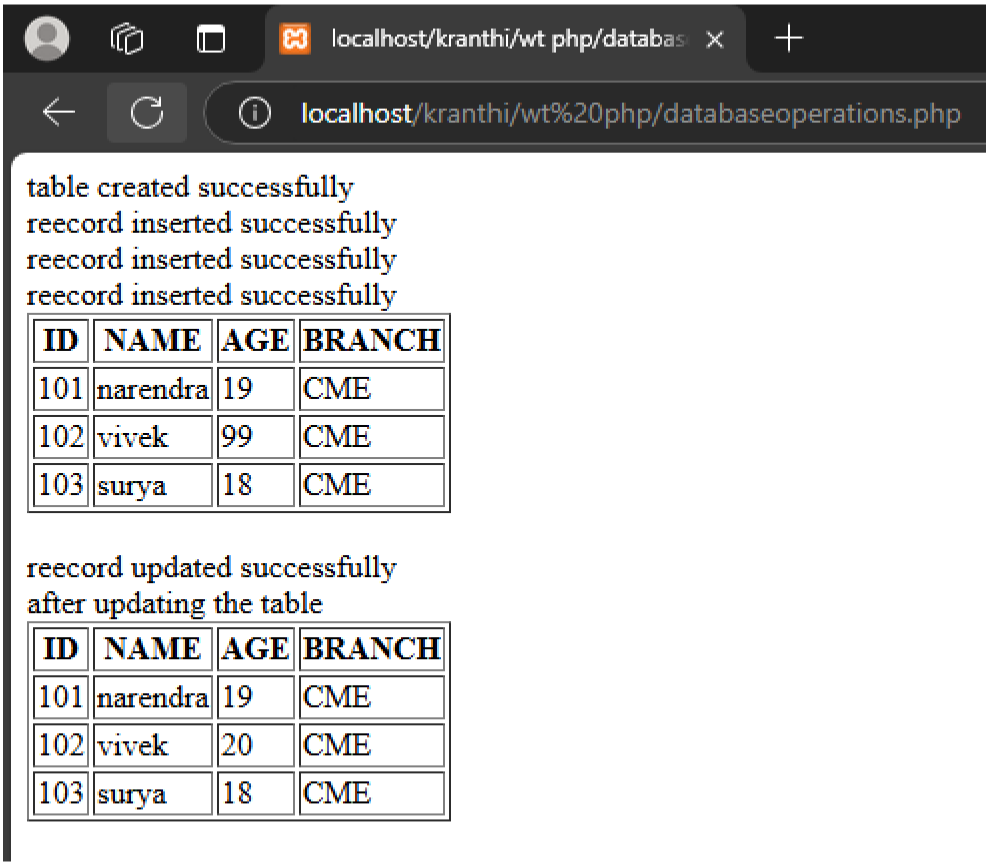
Program:

```
<?php
 $host="localhost";
 $user="root";
 $pass="";
 $dbname="kranthi";
 $conn=new mysqli($host,$user,$pass,$dbname);
 if($conn->connect_error) {
    die("could not connect:".$conn->connect_error);
  }
```

```
$sql='create table students(id int(10),name varchar(30),age int(10),branch
varchar(30))';
  if($conn->query($sql)) {
          echo"table created successfully";
  }
  else{
          echo"could not connect".$conn->error($conn);
  }
$sql='insert into students (id,name,age,branch)
values(101,"narendra",19,"CME")';
  if($conn->query($sql)) {
          echo"<br>reecord inserted successfully";
  }
  else {
          echo"could not insert:".$conn->error($conn);
  }
$sql='insert into students (id,name,age,branch)
 values(102,"vivek",99,"CME")';
  if($conn->query($sql)) {
          echo"<br>reecord inserted successfully";
  }
  else {
          echo"could not insert:".$conn->error($conn);
  }
 $sql='insert into students (id,name,age,branch)
 values(103,"surya",18,"CME")';
  if($conn->query($sql)) {
          echo"<br>reecord inserted successfully";
  }
  else {
          echo"could not insert:".$conn->error($conn);
  }
 $sql="select * from students";
 $result=$conn->query($sql);
 if($result->num_rows > 0) {
   echo"<table border=1>
      <tr>
       <th>ID</th>
       <th>NAME</th>
      <th>AGE</th>
      <th>BRANCH</th>
     </tr>";
            while($row = $result->fetch_assoc()) {
              echo"<tr>
                   <td>$row[id]</td>
                   <td>$row[name]</td>
                   <td>$row[age]</td>
                   <td>$row[branch]</td>
                  </tr>";
              }
  echo"</table>";
 }
else {
  echo"No data found";
 }
$sql='update students set age=20 where name="vivek" ';
 if($conn->query($sql)) {
        echo"<br>reecord updated successfully";
 }
 else  {
        echo"could not update:".$conn->error($conn);
 }
echo"<br>after updating the table";
$sql="select * from students";
$result=$conn->query($sql);
if($result->num_rows > 0)  {
  echo"<table border=1>
     <tr>
      <th>ID</th>
      <th>NAME</th>
      <th>AGE</th>
      <th>BRANCH</th>
     </tr>";
            while($row = $result->fetch_assoc())  {
              echo"<tr>
                   <td>$row[id]</td>
                   <td>$row[name]</td>
                   <td>$row[age]</td>
```

```
                        <td>$row[branch]</td>
                       </tr>";
               }
   echo"</table>";
  }
 else {
   echo"No data found";
  }
 $conn->close();
 ?>
```

output:


localhost/kranthi/wt php/databas
localhost/kranthi/wt%20php/databaseoperations.php

```
table created successfully
reecord inserted successfully
reecord inserted successfully
reecord inserted successfully
```

ID	NAME	AGE	BRANCH
101	narendra	19	CME
102	vivek	99	CME
103	surya	18	CME

```
reecord updated successfully
after updating the table
```

ID	NAME	AGE	BRANCH
101	narendra	19	CME
102	vivek	20	CME
103	surya	18	CME

Result: Hence, to perform various operations on a database table using functions completed successfully.

Experiment-17

Aim: Write a PHP program to set a cookie.

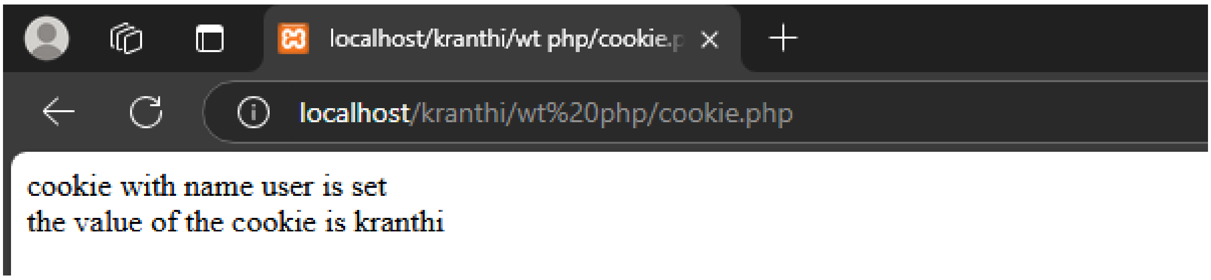
Description:

A **cookie** is a small piece of data stored on the client's computer by a web server. Cookies are used to store information about a user's session or preferences for use in future requests to the same website.

Program:

```
<?php
 $name="user";
 $value="kranthi";
 setcookie("$name","$value",time()+86400);
 ?>
 <html>
 <body>
   <?php
     if(!isset($_COOKIE["$name"]))  {
            echo"cookie with name ".$name." is not set";
           }
          else  {
            echo"cookie with name ".$name." is set";
                  echo"<br>the value of the cookie is ".$value;
           }
        ?>
 </body>
</html>
```

output:



localhost/kranthi/wt php/cookie.p
localhost/kranthi/wt%20php/cookie.php

```
cookie with name user is set
the value of the cookie is kranthi
```

Result: Hence, to set a cookie completed successfully.