Q. What are the advantages and disadvantages of web applications?

Web Applications

Definitions:-

- Web applications are software programs that run on web servers and are accessed through web browsers over the internet or an intranet.
- They are designed to provide users with a rich, interactive and dynamic user interface to perform various tasks as to access and manage information.
- Web applications are built using various web technologies such as PHP, python, java, javascript, HTML, CSS and their programming languages.

Examples:

- Online Website Shopping
- Social media platforms
- Online-banking System

Advantages:

(1) Accessibility:

=> can be accessed from anywhere the world

=> as well as there is an internet connection and a web browser.

(2) Cost-effective:

=> less expensive to develop and maintain than traditional desktop applications because they can be accessed from a web browser without the need for software installation or updates.

(3) Cross-platform Compatibility:

=> WA are platform-independent and can be accessed from various devices such as desktop, laptops, tablets and smartphones, regardless of the OS.

(4) Scalability:

=> WA can be easily scaled up or down to accomodate changing

business needs and user demands.

(5) Collaborations

=> facilitate collaboration and communication among team members and users, allowing them to share and work on documents and data in real time.

Disadvantages:

(1) Internet dependency:

=> Web applications require an internet connection to function properly, and their performance and accessibility can be affected by internet speed, bandwidth and connectivity issues.

(2) Security risks:

=> web applications are vulnerable to security threats such as hacking, data breaches, and malware attacks and cyber attacks which can compromise sensitive data and information.

(3) Limited functionalities:

of features as desktop applications because they are constrained by web browser capabilities and technologies.

(4) User experience / performance issue

=> Web applications may suffer from performance issues such as slow loading times, latency, and bandwidth limitations, especially when dealing with large amounts of data.

business needs and user demands.

(5) Collaborations:

=> facilitate collaboration and communication among team members and users, allowing them to share and work on documents and data in real-time.

Disadvantages:

(1) Internet dependency:

=> Web applications require an internet connection to function properly, and their performance and accessibility can be affected by internet speed, bandwidth and connectivity issues.

(2) Security risks:

=> Web applications are vulnerable to security threats such as hacking, data breaches, and malware attacks and cyber attack which can compromise sensitive data and information.

(3) Limited functionalities:

=> Some web application may not [illegible] or features as desktop applications because they are constrained by web browser capabilities and technologies.

(4) User experience / performance issue:

=> Web applications may suffer from performance issues such as slow loading times, latency, and bandwidth limitations especially when dealing with large amounts of data.

Q How to access CSS from javascript explain with example?

⇒ Inorder to access CSS from JavaScript, we can use DOM to access modify CSS properties of an HTML element.

⇒ CSS is used to apply styles to HTML elements.

⇒ CSS defines how Html elements should be displayed, such as setting font size, color, margin, padding and many other styling properties.

⇒ DOM is used to access and modify CSS properties of an HTML element.

DOM:

⇒ DOM is an API for application programs to interact with the web-~~browser~~ documents. Using DOM along with javascript, we can create documents, navigate through the document structure, add or delete elements and their content.

⇒ Documents in the DOM have a tree like structure:

⇒ To access an HTML element in Javascript, we can use various DOM methods such as

- getElementById()
- getElementByClassName()
- getElementByTagName()

⇒ These methods return a reference to the HTML element, which can then use to modify its properties.

⇒ Different elements in a web document are treated as object in javascript and each object has properties and methods.

⇒ Using DOM, we can get the address of an HTML element in different ways:

Method 1:

To use the document objects forms array property along with the elements array property.

Example:

Consider this html code:

```
<html>
  <head> <title> Simple forms </title>
  </head>
  <body>
    <form action = " ">
```

Enter your name: <input type="text" />

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

- To obtain the textbox of this code, the following script is written:

var name = document.forms[0].elements[0].value;

Drawback:

⇒ it becomes difficult to obtain the address of the elements, when there are multiple forms in a document or if there are a large number of elements in a form or if new elements are added to a form.

Method 2:-

The second way is using the name attribute of the HTML elements.

Example:

<html>
<head><title>Simple form</title></head>
<body>
<form action="" name="fmmain">
Enter your name:
<input type="text" name="txtname" />
</form>
</body>
</html>

⇒ To obtain the address of the textbox,
var name = document.fmmain.txtname.value;

Cons:

- it does not work with a group of checkboxes or a group of radio buttons which will have the same value for their name attribute.
- Also XHTML 1.1 does not allow name attribute on form tag.

Method 3:

- By using the getElementById() method which was introduced in DOM 1.

Example:

```
var name = document.getElementById("txtname");
```

=> Since the value of id attribute can be different for checkboxes and radio buttons in a group, there will be no problem.

=> The id and name attribute can be used in combination for processing a group of checkboxes or radio buttons.

Example:

```
< form id = "genderGroup" >
    < input type = "radio" name = "gender"
value = "male" /> Male
< input type = "radio" name = "gender"
        value = "female" /> Female
</ form >
// Javascript code
var count = 0
var dem = document.getElementById("genderGroup");
for (inden = 0; inden < dem.genders.length, inden++)
    if (dem.genders[inden].checked)
        count++;
```

HTML code:

```
< div id = "myDiv" > Hello world </ div >
```

CSS code:

```
#myDiv {
    color : red;
    font-size : 24px;
}
```

Javascript code:

```
// Access the element
var myDiv = document.getElementById("myDiv");
// modify the css property
myDiv.style.color = "blue";
myDiv.style.fontsize = "30px";
```

- Once we have a reference to an HTML element, then modify the CSS properties using the style property of the element.
- The style property is an object that contains all the CSS properties defined for the element.
- Using javascript to modify the CSS properties of HTML elements dynamically can be useful for creating interactive & dynamic web pages.

for example,

we can change the background color of a button when the user hovers over it, to animate an element by changing its position, size, opacity over time.

Example

```
<!Doctype html>
<head>
    <title> Accessing CSS from javascript
    </title>
    <style>
        #myElement {
            color: red;
        }
    </style>
</head>
<body>
    <div id="myElement"> Hello world! </div>
    <button onclick="changeColor()"> Change color
    </button>
    <script>
        function changeColor()
        {
            var elm = document.getElementById
            ("myElement");
            elm.style.color = "blue";
        }
    </script>
</body>
</html>
```

Q# Describe Javascript approaches for manipulation Html document content?

Manipulating:-

=> Manipulation is the act of making changes to something in order to alter its appearance, behavior or state.

=> Manipulation in web refers to modifying the content, layout or behavior of a webpage using programming languages such as javascript or CSS.

=> manipulating the content might involved:

- displayed on the page
- Adding elements
- Removing Elements
- Changing the attributes of existing elements

(1) Accessing Elements:-

=> Any element can be accessed in an HTML document by using the DOM.

=> The DOM is a hierarchical tree structure that represents the elements on an HTML document as objects.

=> You can use.

- document.getElementById()
- document.getElementsByClassName()
- document.getElementsByTagName()

are methods to access elements by their ID, class name or tag name respectively.

Manipulating Content:-

=> Once we have accessed an element, you can manipulate its content using the innerHTML property.

for example:

- to change the text of a paragraph element using the following code

```
document.getElementById("myParagraph").innerHTML = "New text";
```

Modifying Attributes:

=> To modify the attributes of an element using setAttribute() and removeAttribute() methods.

for example:

To change the src image attribute

of an image element using the following code:

```
document.getElementById("myImage").
        setAttribute("src",
        "new_image.png");
```

iii) Creating and Deleting Elements:

⇒ To create new elements using the createElement() method and them to the DOM using the appendChild() method.

for example:

⇒ To add a new paragraph element to a div element using the following code.

```
Var newParagraph = document.CreateElement("p")
var TextNode = document.createTextNode("New Paragraph")
newParagraph.appendChild(textNode);
document.getElementById("myDiv").appendChild
        (newparagraph);
```

⇒ for remove elements removeChild() is used.

for example:

To remove a paragraph element from a div element using the following code

```
var Parent = document.getElementById("myDiv");
var child = document.getElementById("myParagraph");
Parent.removeChild(child);
```

iv) Event Listeners:

⇒ To add event listeners to elements to respond to user actions such as mouse clicks, keyboard presses, and form submissions.

for example,

you can add a click event listener to a button element using the following code:

```
document.getElementById("myButton").addEventListener
        ("click",
        function(){ alert("Button clicked!");
        });
```

Conclusion:

⇒

- These are some ways in which the structure of an HTML document manipulated using JavaScript

Q. Write an Html and CSS to create one horizontal and one vertical menu using unordered list?

```
<!DOCTYPE html
<html>
  <title> Menu Example </title>
  <style>
    /* Horizontal Menu */
    .horizontal-menu
    { align-items: center;
      display : flex;
      list-style-type: none;
      margin : 0;
      padding : 0;
      background-color : #f1f1f1;
    }
    .horizontal-menu li
    {
      flex-grow : 1;
      text-align : center;
      padding : 14px 16px;
      border-right : 1px solid #ccc;
    }
    .horizontal-menu li : last-child
    {
      border-right : none;
    }
    . horizontal-menu li a
    {
      display : block;
      color : #666;
      text-align : center;
      text-decoration : none;
    }
    . horizontal-menu li a : hover
    {
      background-color : #ddd;
    }
    /* Vertical Menu */
    . vertical-menu {
      list-style-type : none;
      margin : 0;
      padding : 0;
      width : 200px;
      background-color : #f1f1f1;
    }
```

```
.vertical-menu li a
{
    display: block;
    color: #666;
    padding: 12px;
    text-decoration: none;
}
.vertical-menu li a:hover
{
    background-color: #ddd;
}
</style>
</head>
<body>
    <h2> Horizontal Menu </h2>
    <ul class="horizontal-menu">
    <li> a href="#">Home</a></li>
    <li> a href="#"> Products</a></li>
    <li> a href="#"> Services</a></li>
    <li> a href="#"> About us</a></li>
    <li> a href="#"> Contact us</a></li>
    </ul>
```

```
<h2> Vertical Menu </h2>
<ul class="vertical-menu">
    <li><a href="#"> Home</a></li>
    <li><a href="#"> Products</a></li>
    <li><a href="#"> Services</a></li>
    <li><a href="#"> About us</a></li>
    <li><a href="#"> Contact Us</a></li>
</ul>
</body>
</html>
```

Q#:
Javascript validation for re-type password.

```
function ValidatePassword()
{
    var Pass = document.getElementById("Pass
    var cfPass = document.getElementById("Confirm
    if (Pass != cfPass)
    {
        alert("Passwords do not match");
        return false;
    }
    return true;
}
```

Q. what is client and server side programming?

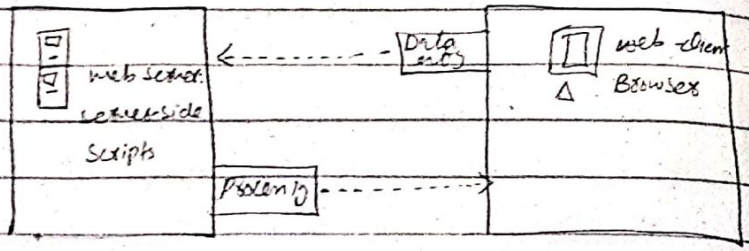
- Describe it in scenario of google search engine?

: Assume yourself as client while entering a query in search bar and google as the server side.

Client-side Programming:

=> it refers to the code that runs in a users web browser

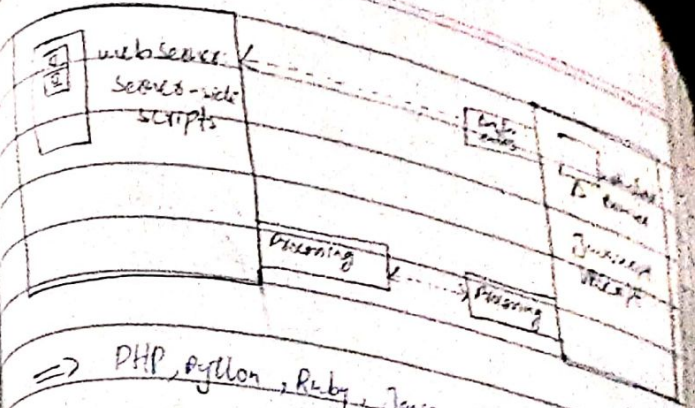
=> Can be embedded the code in html document.



- HTML, javascript & css are client-side PL

Server-side Programming:

=> it refers to the code that runs on the server that hosts the website.



=> PHP, Python, Ruby, Java, Node.js, ASP are server-side PL.

=> In the scenario of using Google search engine:-

=> When a user enters a query in the search bar and hits the enter button, the client-side programming sends an HTTP request to the server-side programming running on Google's Server.

=> The server-side programming then processes the request by searching through its index of web pages to find relevant results based on the user query.

=> Once the server-side programming has found the relevant results,

it formats the data into HTML and sends it back to the client-side programming, which displays the results in the user's web browser.

=> As the client, responsibility is that for sending the initial HTTP request by entering a query into the search bar.

=> The client side programming in this case is javascript that runs in the web-browser, which handle user interactions with the webpage and sends http requests to the server-side programming.

=> once the server-side programming processes the request and sends back the results, the client-side programming then handles the display of the results in the user's web-browser.

=> On the server side, Google uses a variety of programming languages and technologies to handle the processing of search requests, including python, C++ and java.

=> The server-side programming is responsible for handling millions of search queries per second and returning the relevant results to the user.

=> This involves complex algorithms for ranking search results, as well as distributed computing technologies for handling the massive scale of the search engine.

=> In Conclusion, Overall client-side and server side programming work together to create a seamless user experience on the web. Client-side programming handles user interaction with the webpage and sends requests to the server-side programming, which then processes the requests and return data to be displayed by the client-side programming in the users web-browser.

Q. Write the main responsibilities of a browser and its functionality?

Definition:

"A web browser is a software application that is used to access and navigate the internet."

=> its main responsibility is to provide an interface for users to view and interact with the web pages.

=> it also have several other functionalities are:

Retrieving web pages:-

- The browser's main function is to retrieve web pages from web servers over the internet using the HTTP or HTTPS protocols.

Rendering web pages:

- The browser renders the HTML, CSS and javaScript code of web pages to display them in a readable and interactive format.
- This involves interpreting the code and rendering it as text, images, and other media.

Managing tabs:-

=> The browser allows users to open multiple tabs and switch between them, making it easy to navigate between different websites and pages.

Managing bookmarks:-

=> The browser allows users to save bookmarks of their favorite websites, making it easy to revisit them in the future.

=> A bookmark is a saved link to a specific webpage or website that allows you to quickly and easily return to that page later.

=> Bookmarks can be organized into folders or categories.

Supporting plugins and extensions:

=> Browsers allow users to install plugins and extensions that add additional functionality to the browser such as blockers or password managers.

Managing Cookies and cache:

=> Browsers store cookies

small text files that are used to store information about a user's preferences or login credentials for websites.

=> The browser also stores temporary copies of web pages on its cache to speed up page load times.

Provide security features:

=> The browser includes features such as phishing protection, pop-up blockers, and warnings for potentially dangerous websites to help keep users safe while browsing the web.

=> And secure connections using HTTPS.

Providing developer tools:

=> Browsers provide developer tools that allow web developers to debug and optimize their websites, as well as test and experiment with new features.

Managing History:

=> it also keeps a history of the user's browsing activity, making it easy to return to previously visited web pages.

Download files:

=> A browser allows users to download files from the web, such as documents, images and videos.

=> it provides a download manager that allows users to track the progress of their downloads and manage their files.

Form fillings

=> A browser can automatically fill in forms with the user's personal information, such as their name, address and email addresses. it can also save form data for future use.

Conclusion:

=> The main responsibility of a browser is to provide an interface for users to access and navigate the internet, while also providing several additional functionalities that make it easier and more secure to use the web.

Q. what is TCP/IP and how it works?

TCP/IP:

TCP/IP is a suit of networking protocols that operate at different levels to enable communication between devices over a network.

TCP/IP works:

(1) Data is broken into packets:

- when a device wants to send data over the network, it breaks the data into smaller packets.
- Each packet includes a header with information such as the source and destination addresses.

(2) Packets are routed to their destination:

⇒ once the packets are created they are sent over the network using the IP protocol.

⇒ IP is responsible for routing the packets to their destination based on the destination address in the packet header.

(3) Packets are reassembled:

⇒ once the packets reach their destination they are reassembled into the original data by the receiving device TCP ensures that all the packets arrive in the correct order and without errors.

(4) Communication is established

⇒ Before data transfer can begin, the two devices must establish a connection using the TCP protocol.

This involves a three-way handshake where the devices exchange messages to establish the connection and agree on certain parameters such as the maximum size of the data packets.

(5) Data transfer occurs:

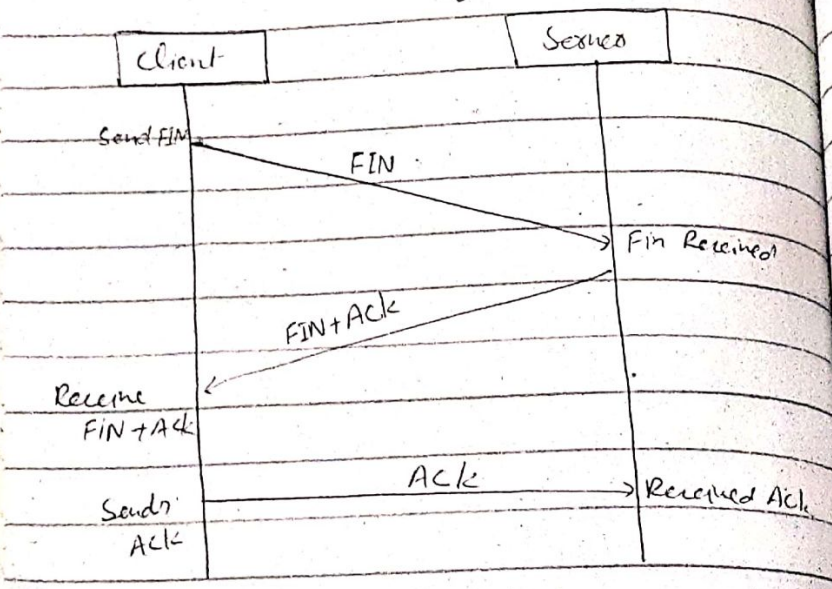
⇒ once the connection is established, data transfer can occur.

⇒ TCP breaks the data into packets which are sent over the network using the IP protocol. The receiving device reassembles the packets into the original data.

(6) Connection is closed.

⇒ Once the data transfer is complete, the two devices use the TCP protocol to close the connection.

⇒ This involves a three way handshake to ensure that all data has been successfully transferred and the connection can be safely closed.



for establishing use:

SYN instead of FIN

Layers in TCP/IP model

Applink layer	HTTP, FTP, Telnet, SMTP, SSH
Transport Layer	TCP & UDP
Internet	IP
Network	Ethernet / Wifi

Terms:

Packet:

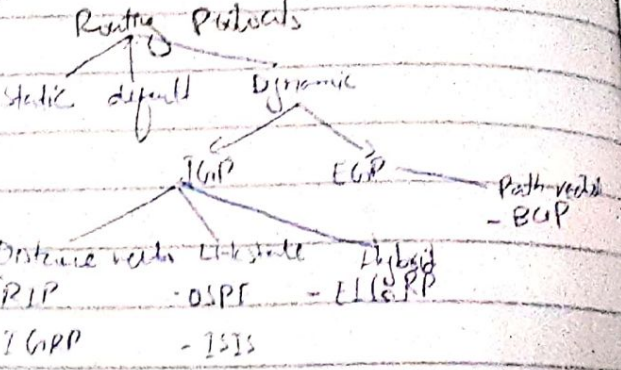
- A packet is a unit of data transmitted over a network.
- It contains two parts:
  - header (control information for routing)
  - Payload (Actual data)
- Packets are used to efficiently route information between devices over a network.
- They allow for reliable transmission

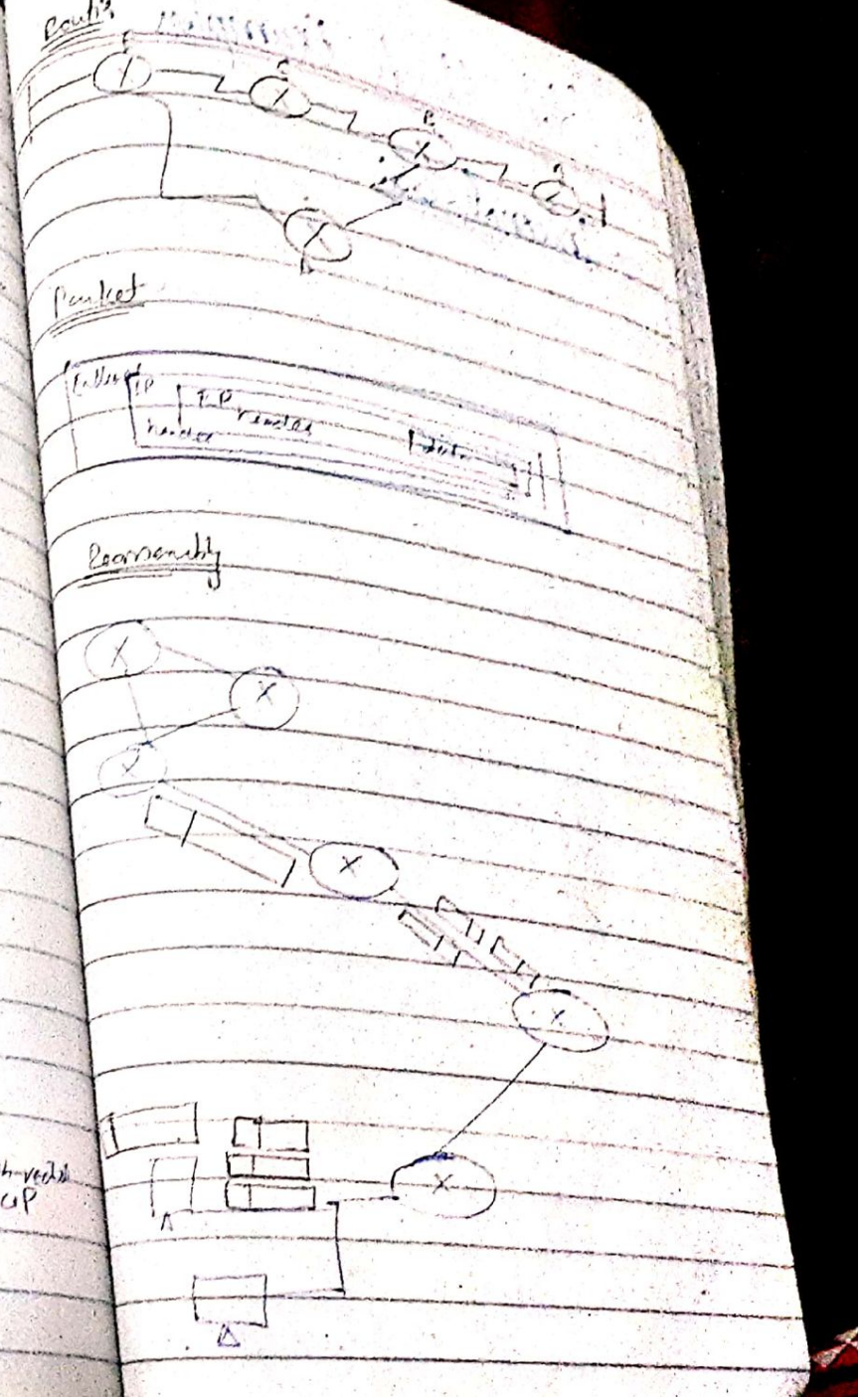
Reassembly:

- The process of reconstructing higher layer PDU from a collection of smaller lower-level PDU

Routing

- A routing is a process of selecting path along which the data can be transferred from source to the destination.
- Routing is performed by a device that is called router.
- There are three types of routing
  - Static
  - Dynamic
  - Default
- There are different types of routing protocols:
  - RIP < RIPv1 / RIPv2
  - Interior Gateway Protocol
  - Exterior Gateway Protocol
  - EIGRP
  - OSPF
  - BGP





Style Sheets Examples

– Internal CSS:

```
<html>
  <head>
    <title> Internal CSS </title>
    <style>
      .my{
        text-align: center;
        }
      .HEW{
        color: #009900;
        font-size : 50px;
        font-weight: bold;
        font-style : italic;
      }
      .geeks{
        font-size : 20px
        display : block;
        }
    </style>
  </head>
<body>
    <div class="my">
      <div class="HEW">Hello!</div>
      <div class="geeks">
        A Computer Science </div>
    </div>
  </body>
</html>
```

Example of External Stylesheet.

geeks.css

```
body {
    background-color : Powderblue;
    }
  .HEW{
      color : #009900;
      font-size: 50px;
      font-weight : bold;
    }
  #2{
      font-style : bold;
      font-size : 20px;
    }
```

Priorities of CSS:

- The Priorities of CSS are in the following orders.
  - Inline
  - Internal / Embedded
  - External / Emb
- Multiple Style sheets can be defined on one page.
- As inline has the highest priority, any style that are defined by internal and external sheets are overridden by inline styles.
- Internal or Embedded has second in priority list and overrides the styles in the external sheet.
- External Style sheet has the least priority.

if there are no styles defined either in inline or internal style sheet then external style sheet rules are applied for the HTML pages.

Difference blw &

- and are HTML elements used for structuring and styling content.

<Span>

<Span> is an inline level element used to apply styling or markup to specific portions of text within a large block of content. It is often used for text manipulation and styling with CSS.

eg. This is a blue text.

<Div>

<Div> is a block-level element used as a container and organize other HTML elements. It creates a logical division or sections for layout and visual effects.

eg

<h2> Welcome </h2>

 This is the content

⇒ Both elements have different default behavior and are commonly used in conjunction with CSS to achieve desired styling and layout effects on webpages.

Padding VS Margins

o Padding controls the space within the element.	o Margin control the space outside the element.
- it refers to Positioning	it refers to spacing between adjacent elements.

Q. PP) Write a Javascript program that take a city name from user and delete city from dropdown list, if found otherwise display message not found?

```
<!DOCTYPE html>
<html>
<head>
<title> Delete city from dropdown List</title>
</head>
<body>
<select id="cityList">
<Option value="New York"> New York </option>
<Option value="Tokyo"> Tokyo </option>
<Option value="Paris"> Paris </option>
<Option value="Sydney"> Sydney </option>
</select>
<script>
// Get the dropdown list element
var cityList = document.getElementById
("cityList");
// prompt the user to enter a city
name
```

```
var cityName = prompt("Enter the name of
the city you want to delete:");

// Search for the city in the list
var cityFound = false;
for (var i = 0; i < cityList.options.length; i++)
{ // Checks if the current option match
  if (cityList.options[i].value == cityName)
  {
    cityList.remove(i);
    cityFound = true;
    break;
  }
}
// Display message if the city not
found
  if (!cityFound)
  {
    alert("City not found in the
          list");
  }
</script>
</body>
</html>
```

Write a code to display different shapes using javascript?

```
<!DOCTYPE html>
<html>
<head>
  <title> Shapes </html>
  <style>
    canvas {
      border: 1px solid #000;
    }
  </style>
</head>
<body>
  <canvas id="myCanvas" width="400"
          height="400"></canvas>
<script>
// Get the canvas element...
const canvas = document.getElementById
               ('myCanvas');
const ctx = canvas.getContext('2d');
// function to draw circle
function drawCircle(x, y, radius)
{
```

```
ctx.beginPath();
ctx.fillStyle = "green";
ctx.arc(x, y, radius, 0, Math.PI * 2);
ctx.closePath();
ctx.stroke(); ctx.fill();
}
// draw the rectangle:-
function drawRectangle(x, y, width, height)
{
    ctx.fillStyle = 'blue';
    ctx.fill();
    ctx.beginpath();
    ctx.rect(x, y, width, height);
    ctx.closepath();
    ctx.stroke();
}
// draw the triangle:-
function drawTriangle(x1, y1, x2, y2, x3, y3)
{
    ctx.strokeStyle="#FF00FF";
    ctx.beginPath();
    ctx.moveTo(x1, y1);
    ctx.lineTo(x2, y2);
    ctx.lineTo(x3, y3);
    ctx.fileStyle = 'Red';
    ctx.closePath();
    ctx.stroke();
}
```

```
// Draw the line
function drawLine(x, y)
{
    ctx.beginPath();
    ctx.moveTo(x, y);
    ctx.lineTo(y, y);
    ctx.strokeStyle = "#FF00FF";
    ctx.stroke();
}
// Clear the canvas
ctx.clearRect(0, 0, canvas.width, canvas.height);
// Calling function
drawCircle(100, 100, 50);
drawRectangle(200, 100, 100, 50);
drawTriangle(300, 300, 350, 200, 400, 300);
drawLine(100, 300);
</script>
</body>
</html>
```

Description:-

- Code uses the HTML5 <canvas> element and the 2D rendering context getcontent("2d") to draw the desired

=> moveTo(), lineTo(), arc(), rect() are used to create different shapes.

=> strokeStyle property of javascript is used to set the color, gradient or pattern used for strokes (lines) when drawing shapes on canvas. It defines the color of the outline of the shape.

=> fillStyle used for filling the shapes. it set the color of interior of the shape.

<canvas> used to draw graphics on a web-page.

Q. What javascript command would you use to replace the current document with the main page of the oreilly.com website?

=> To replace the current document with the main page of the "oreilly.com" website using javascript by using the location.replace() method.

```
=> window.location.replace("https://www.oreilly.com");
```

=> This code will redirect the browser to the main page of oreilly.com website by replacing the document with the new URL specified as the argument of the replace() method.

=> The browser will navigate to "http://www.oreilly.com", loading the main page of the O'Reilly website.

Drawback of this method:

o Users will lose the current page's history, and pressing the back

button will not take them back to the previous page.

Overcome:

- if user want to be able to navigate back to the previous page by using window.location.assign().
- it has similar functionality but adds the new URL to the navigation history.

Example:

```
window.location.assign("https://www.oreilly.com");
```

Q. How can you let JavaScript deal gracefully with an error when it encounters?

⇒ To handle error gracefully in JavaScript, by using try-catch blocks to capture and handle the exceptions.

Example:

```
try {
    // code that throw error
    //.....
}
catch (error)
{
    // code to handle the error
    conside.log("An error occured:", error.message);
    // can perform additional actions
}
finally
{
    // code that will always execute
    , regardless of whether an error
    occured or not.
    //....
}
```

Description:-

⇒ The code that may throw an error is placed in try block. If an error occurs the execution jumps into corresponding catch block, where error are handled.

⇒ The error objects contain the error message.

Q. What delay between events should you set (in milliseconds) to achieve an animation rate of 50 frames per second?

=> To achieve an animation rate of 50 frames per second (FPS), by calculate the delay between each frame using the formula:

delay = 1000 / FPS

=>

Given: FPS = 50

delay = 1000/50 = 20 milliseconds

=> Therefore, to achieve an animation rate of 50 FPS, by set the delay of 20 milliseconds between each frame update.

=> This ensures that the animation runs smoothly at the desired frame rate

```
.myElement {
    animation-name : myAnimation;
    animation-duration: 5Qs ;
    animation-delay : 20s ;
}
```

Q. Give two cross-browser methods to display the URL assigned to the link with an id of this link?

=> There are the following cross-browser methods to display the URL assigned to a link with id:

Method 1:

- Using Javascript and innerHTML property

```
var link= document.getElementById("myId")
var url= link.href; // access the url of the link
console.log(url); // display the url
```

Method 2:

- Javascript and getAttribute().

```
var link= document.getElementById("myId")
var url= link.getAttribute("href");
console.log(url);
```

Q. What is the difference b/w a WAMP, a MAMP and a LAMP?

Web Stack:

=> A web stack or web application stack refers to a compilation of software that is together used to build websites or web applications.

=> To construct a stack basic requirements are.

- OS
- webserver
- Database
- Scripting/Programming

Local server:-

=> A local server is a server that is hosted locally on your machine or local computer.

=> with help of local server website can be test by the web developers many times before updating it to the web-server.

=> It saves time and easy to use.

=> XAMPP, WAMP, LAMP, MAMP are local servers that are used while developing PHP websites.

=> Using this bugs and errors are find before uploading to the main server.

=> The major difference between the OS.

	XAMPP	WAMP	MAMP	LAMP
o Supporting platform:	- Cross-platform for Linux, Windows, and Mac OS.	- for the Windows OS only.	- for the mac OS only.	- for the Linux OS only.
o Programming language	- Perl & PHP	- PHP	- PHP	- PHP, Perl & Python
o Database	- uses MariaDB developed by MySQL.	- RDBMS	- MySQL	- MySQL RDBMS
o server	o Apache	o Apache	o Apache	o Apache

Q. Elabore XAMPP?

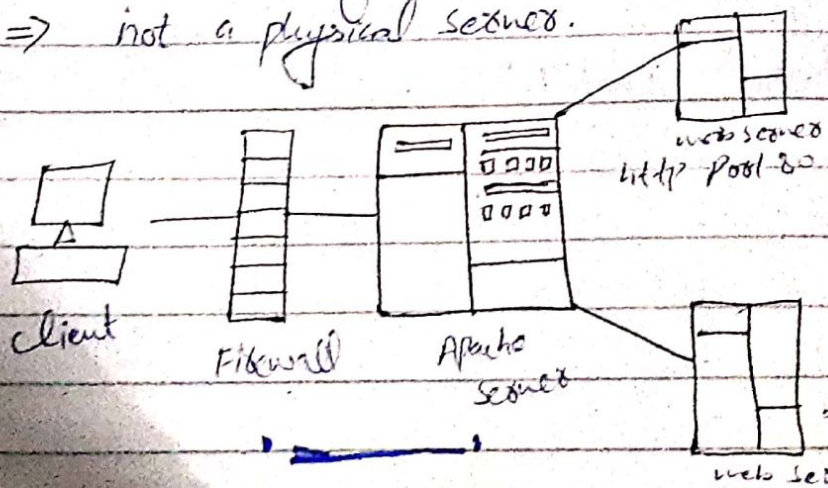
=> Stands for:

- Cross-platform
- Apache
- MariaDB (MySQL)
- PHP
- Perl

=> Simplest and light weight server for testing websites.
=> Opensource platform
=> it has X-os because it works in all major OS.
=> includes additional features like Filezilla, Linux, mail etc.
=> Download and installed easily.
=> unlimited Testing can be performed.

Apache server:

=> Free and open source software that allows users to deploy their websites on the internet.
oldest and most reliable web servers software maintained by the Apache Software Foundation, with the first version released in 1995.
=> cross-platform webserver
=> not a physical server.



2. Write a PHP Program that allows you to:-

(i) Create the text file "text-file".

```
<?php
$file = fopen("text.txt", "w"); // Open the file in write mode
if ($file)
{
fwrite($file, "This is my file.");
        // write the content.
fclose($file); // close the file
echo "The file 'text.txt' has been created";
}
else
{
  echo "Unable to create the file";
}
?>
```

(ii) Insert the text "Hello..." into the file.

```
<?PHP
$file = fopen("text.txt", 'a')
if ($file)
  {
```

```
fwrite ( $file, " Hello dear");
fclose ( $file);
}
else
{
 echo " Unable to open the file";
}
?>
```

(3) append the text "welcome".

Same like previous.

except:

```
fwrite ( $file , "welcome");
```

Difference between server-side and client side Scripting language?

Scripting language:-

⇒ A Scripting language is a programming language that employs a high-level control to interpret and execute one command at a time.

⇒ In general, scripting language are easier to learn and faster to code.

⇒ It is structured and compiled languages such as C++ and C.

Client-side Script	Server-side Script
o Web browser executes that resides at the user Computer.	o Web server executes that produces the page to be sent to the browser.
o Response from this script is faster because the scripts are processed on the local computer.	o Response from server-side script is slower because the scripts are processed remotely.

o Client side scripting cannot be used to connect to the database on the web server.	o Server side scripting is used to connect to the databases that resides on the web server.
o Cannot access to the file system that resides at the web servers.	o Can access the file system residing at the web servers.
o it is possible to be blocked by users.	o cannot be blocked by users.
o Users can view source code from the web browsers.	o Can not view source code from the web browsers by user.
o Can be used to reduce servers load.	o Allows website to be dynamic rather than static.
o HTML, CSS, JavaScript & more	o PHP, Ruby on Rails, ASP.NET Lasc, Python & mo

Insert data enlesed in form(having fields for First Name, Last Name, Genders, Date of Bisth, Usesname, Passwoed, Addsess) into "LoginOb" database by PHP code, also display all the content of "LoginDB" database.

```
<?php
// Get the form data
if (isset($_POST['submit']))
  {
$fname = $_POST['FirstName'];
$lname = $_POST['LastName'];
$genders = $_POST['genders'];
$DOB = $_POST['date-bisth'];
$unome = $_POST['usesname'];
$Passwoed = $_POST['Passwoed'];
$addsess = $_POST['addsess'];
} // Conned to the database
```

```
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "loginDB";
$conn = new mysqli($servername,
$username, $password, $dbname);
if ($conn->connect_error)
{
    die("Connection failed".
        $conn->connect_error);
}
// Insert the form data into
the database.
$sql = "INSERT INTO Users
    (first_name, last_name, gender,
dob, username, password, address)
    VALUES ('$fname', '$lname',
'$gender', '$DOB', '$uname', '$password',
'$address')";
if ($conn->query($sql) === TRUE)
    echo "Data inserted successfully";
else
echo "Error" .sql :<br>" . $conn->
```

```
// Display all content of the
loginDB database.
$sql = "SELECT * FROM users";
$result = $conn->query($sql);
if ($result->num_rows > 0)
{
echo "<table><tr>
    <th> ID </th>
    <th> First Name </th>
    <th> Last Name </th>
    <th> Gender </th>
    <th> Birth Date </th>
    <th> Username </th>
    <th> Address </th>
    </tr>";
while ($row = $result->fetch_assoc())
    {
echo "<tr><td>" . $row["id"] .
    "</td><td>" . $row["FirstName"] .
    "</td><td>" . $row["LastName"] .
    "</td><td>" . $row["gender"] .
    "</td><td>" . $row["date_birth"] .
```

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

Write a php function to login existing users. (make database table and php code).

```
<!DOCTYPE html>
<head>
    <title> Login </title>
</head>
<body>
    <h2> Login </h2>
<form method="POST" action=
"login.php">
    <label for="username"> Username: </label>
<input type="text" name="username"
id="username" required> </br></br>
<label for="password"> Password: </label>
<input type="password" name="password"
id="password" required></br></br>
<input type="submit" name="submit"
value="Login">
</form>
</body>
</html>
```

Database setup:

```
CREATE TABLE users
(
id INT AUTO_INCREMENT PRIMARY
username VARCHAR(50) NOT NULL,
password VARCHAR(225) NOT NULL,
email VARCHAR(100) NOT NULL,
);
```

PhP code:

```
<?php
// Database Connection details
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "mydb";
$conn
// Function to authenticate user
function login($username, $passw
{
global $servername, $username,
$password, $dbname;
// Create database connection
$conn = new mysqli($servername,
$username, $password, $dbname);
// Check for connection error
if ($conn->connect_error)
{
die("Connection failed" . $conn->
connect_error);
}
// Sanitize the input data to
prevent SQL injection
$username = mysqli_real_escape_string
($conn, $username);
$password = mysqli_real_escape_string($conn,
$password);
// fetch user record from
data base.
$query = "SELECT *
FROM Users
WHERE username = '$username' LIMIT 1";
$result = $conn->query($query);
// Check if user record exist.
if ($result->num_rows == 1)
$user = $result->fetch_assoc();
```

```
// Verify the password
if (password_verify ($password,
$user['password']))
{
    return true;
}
}
// User not found or password
incorrect.
    return false;
}

if(isset($_POST['submit']))
{
$username = $_POST['username'];
$password = $_POST['Password'];
if (login ($username, $password))
{
    header ("Location: home.php");
    exit();
}
else
{
    echo "Invalid username or
    password";
}
}
?>
```

LIMIT 1: clause is used to restrict the result set to only one row. Commonly used in SQL queries to specify the maximum number of rows to return.

- it is used with a value of 1 to retrieve only the first matching row from the users table.
- By using LIMIT 1, ensure that if there are multiple rows with the same username, only the first row is returned.

mysqli_real_escape_string(): used to escape special characters in a string that will be used in an SQL query. Used to prevent SQL injection attacks.

- SQL injection is a security vulnerability where attackers can manipulate SQL queries by injecting malicious code into user inputs.

Design login form for a website contain only two fields username and password. Submit this form to login.php in a secure way.

```
<!Doctype html>
<html>
<head>
  <title> Login </title>
</head>
<body>
    <h1> Login form </h1>
  <form action="login.php" method="POST">
  <label for="username"> Username: </label>
<input type="text" id="username"
name="username" required>
<label for="password"> Password: </label>
<input type="password" id="password"
name="password" required>
<input type="submit" name="submit"
value="Login" > Login </br></br>
```

```php
<?php
$username = $_POST['username'];
$password = $_POST['password'];
// perform necessary validation checks.
if (empty($username) || empty($password)) {
    echo "Please enter both username and password";
    exit();
}
// connect to database:
// write code here .....
// .....

// query
$query = "SELECT *
          FROM users
          WHERE username = '$username'
          AND password = '$password'";
$result = $conn->query($query);

if ($result->num_rows == 1) {
    // success login
    // Redirect the user to page
    header("Location: secure-page.php");
    exit();
} else {
    echo "Invalid username or password";
}
$conn->close();
?>
```

Q. Write a program in php that validate the form with the following fields.

(a) Name (String)
(b) CNIC (XXXXX-XXXXXXX-X)
(c) email address.

```
<?php
function validateName($name)
{
if(!preg_match("/^[a-zA-Z]*$/",$name))
{
echo "Only letters and white space allowed";
}
elseif(empty($name))
{
echo "Name is required";
}
else
{
return true;
}
function validateCNIC($cnic)
{
if(empty($cnic))
{
echo "CNIC is required";
return false;
}
elseif(!preg_match("/^\d{5}-\d{7}-\d{1}$/",$cnic))
{
echo "Invalid CNIC";
return false;
}
else
{
return true;
}
}
function validateEmail($email)
{
if(empty($email))
{
echo "email is required";
return false;
}
elseif(!filter_var($email,FILTER_VALIDATE_EMAIL))
```

```
{
echo "Invalid email format";
return false;
}
else
{
return true;
}
}
// check after if submitted form
if ($_SERVER["REQUEST_METHOD"]=="POST
{
$name= $_POST["name"];
$cnic= $_POST["cnic"];
$email= $_POST["email"];
$validName = validateName($name);
$validcnic = validateCNIC($cnic);
$validemail = validateEmail($email);
if($validName && $validcnic && $valide
{
echo "form is submitted";
}
else
{
if(!$validName)
echo "Invalid name";
if(!$validcnic)
echo "Invalid CNIC";
if(!$validemail)
echo "Invalid email";
}
}
?>
```