Web Engineering Suggestion Solve

4 set (1-33)

Introduction (1set)

1. **What is web page and website? Describe basic concept of static and dynamic website?**

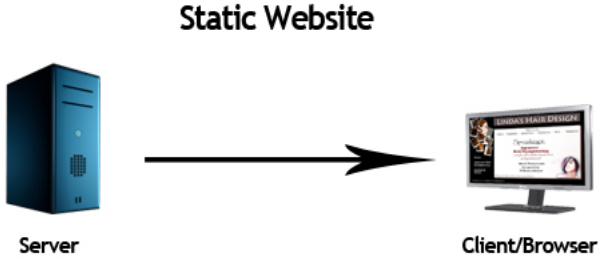
Ans: Web page: A webpage or web page is a document commonly written in HTML(Hypertext Markup Language) that is accessible through the internet or other networks using an internet browser. A web page is accessed by entering a url address and may content text, graphics and hyperlinks to other web pages and file.

Website: A site or website is a Central location of various web pages that are all related and can be accessed by visiting the home page of the website using a browser. For example, the google address URL(Uniform Resource Locator) is https://www.google.com

Static website

Static website is the basic type of website that is easy to create. You don't need the knowledge of web programming and database design to create a static website. Its web pages are coded in HTML.

The codes are fixed for each page so the information contained in the page does not change and it looks like a printed page.



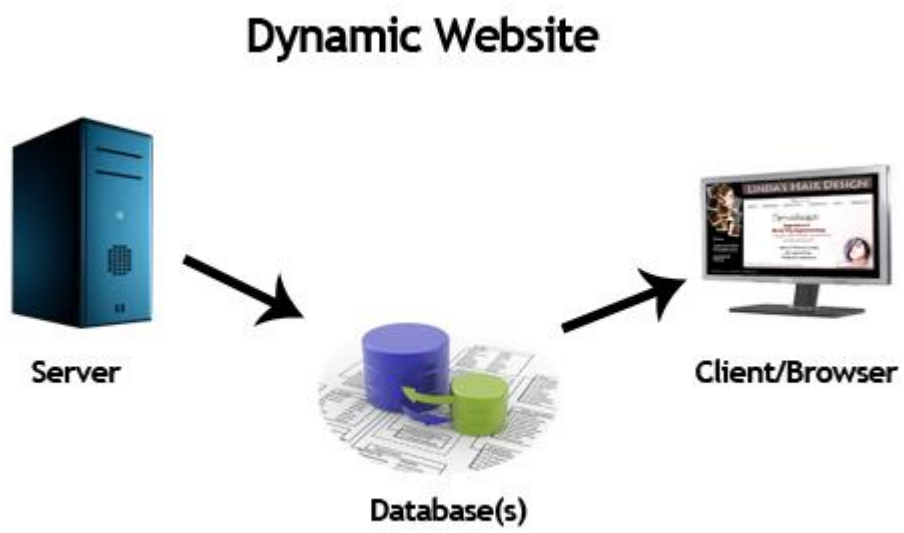
Dynamic website

Dynamic website is a collection of dynamic web pages whose content changes dynamically. It accesses content from a database or Content Management System (CMS). Therefore, when you alter or update the content of the database, the content of the website is also altered or updated.

Dynamic website uses client-side scripting or server-side scripting, or both to generate dynamic content.

Client side scripting generates content at the client computer on the basis of user input. The web browser downloads the web page from the server and processes the code within the page to render information to the user.

In server side scripting, the software runs on the server and processing is completed in the server then plain pages are sent to the user.



Or, Write down the basic concepts of making dynamic website.

Ans: The four components required to create a fully dynamic web page are
1) A web server
(such as Apache),
2) A server-side scripting language (PHP),
3) A database (MySQL), and
4) A client-side scripting language (JavaScript).

Web Server

A web server is computing system designed for exchanging information from server to browser or other client making http (the basic network protocol used to distribute information on the World Wide Web) request. Web server is consist of hardware computer, an operating system and various supporting applications to process network protocol requests. Most commonly Apache is used are server application system on top of Linux or windows server operating system.

Server-side scripting

Server-side scripting is a technique used in web development which involves employing scripts on a web server which produce a response customized for

each user's (client's) request to the website. The alternative is for the web server itself to deliver a static web page.

Database

Database are information house for any dynamic website, it is used to Store data in different tables and extract it dynamically on demand. Database is designed to keep all relevant information and website content in different tables and with the help of database query language we can read, insert or edit data very easily and efficiently, Most commonly MySql is used as database system as its free and widely tested and accepted. Even your favorite social network Facebook uses MySql to store and display data from all over the world.

Client-side scripting

The client-side environment used to run scripts is usually a browser. The processing takes place on the end users computer. The source code is transferred from the web server to the user's computer over the internet and run directly in the browser.

The scripting language needs to be enabled on the client computer. Sometimes if a user is conscious of security risks they may switch the scripting facility off.

When this is the case a message usually pops up to alert the user when script attempting to run.

Or, Differentiate between static and dynamic website.

Ans:

Static Website	Dynamic Website
Prebuilt content is same every time the page is loaded.	Content is generated quickly and changes regularly.
It uses the HTML code for developing a website.	It uses the server side languages such as PHP,SERVLET, JSP, and ASP.NET etc. for developing a webs
It sends exactly the same response for every request.	It may generate different HTML for each of the
The content is only changed when someone publishes and updates the file (sends it to the web server).	The page contains "server-side" code which allows the server to generate the unique content when the page is
Flexibility is the main advantage of static website.	Content Management System (CMS) is the main advantage of dynamic website.

Static sites are easy to develop and a bit experienced people can develop it.	Dynamic sites are not easy to develop because require qualify developers to create it, manage it, test it and maintain security of application and database.

2. Write down the concept of web application.

Ans: Web application: A web application is a computer program that utilizes web browsers and web technology to perform tasks over the Internet.

Web applications use a combination of server-side scripts (PHP and ASP) to handle the storage and retrieval of the information, and client-side scripts (JavaScript and HTML) to present information to users. This allows users to interact with the company using online forms, content management systems, shopping carts and more.

Web applications are usually coded in browser-supported language such as JavaScript and HTML as these languages rely on the browser to render the program executable. Some of the applications are dynamic, requiring server-side processing. Others are completely static with no processing required at the server.

The web application requires a web server to manage requests from the client, an application server to perform the tasks requested, and, sometimes, a database to store the information. Application server technology ranges from ASP.NET, ASP and ColdFusion, to PHP and JSP.

Here's what a typical web application flow looks like:

1. User triggers a request to the web server over the Internet, either through a web browser or the application's user interface
2. Web server forwards this request to the appropriate web application server
3. Web application server performs the requested task – such as querying the database or processing the data – then generates the results of the requested data
4. Web application server sends results to the web server with the requested information or processed data.

3. Explain the basic structure of a website.

Ans: A website's structure refers to how the website is set up, i.e. how the individual subpages are linked to one another.

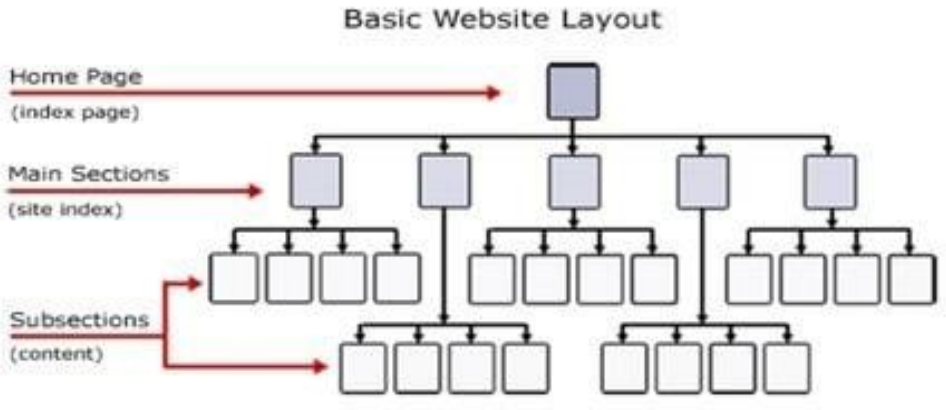
The basic structure of a website is shown below –

```
<!DOCTYPE html>
<html>
<head>
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <title>…..</title>
</head>
<body>
</body>
</html>
```

- <DOCTYPE! HTML> - A piece of code that tells the browser what kind of information it's getting, so that it can display your web page properly.
- <html>- All of your code must be between tags.
- <head>- Hold all the important behind-the-scenes information about your web page, like the title of your web page, and a links to a CSS style sheets.
- <title> - Stuff between title tags doesn't appear on the web page, but you'll see it in search engine results and browser tabs.
- <body> - Everything that displays in the browser -- header, navigation, images, content -- goes between the body tags.

It is particularly important that crawlers can find all subpages quickly and easily when websites have a large number of subpages. For this reason, a website's homepage needs to have links to the most important subpages. Files such as sitemap.xml and robots.txt also help the crawler do its job in this regard.

Ideal site structure of any site should resemble a pyramid with homepage at the top and categories with subcategories beneath it.



4. Describe the common features of a web server.

Ans: Some basic common features of web servers are :-
HTTP: Every web server program operates by accepting HTTP requests from the client, and providing an HTTP response to the client. The HTTP response usually consists of an HTML document, but can also be a raw file, an images, or some other type of document (defined by MIME-types); if some error is found in clients request or while trying to serve the request, a web server has to send an error response which may include some custom HTML or text messages to better explain the problem to end users.
Loggin: Usually web servers have also the capability of logging some detailed information, about client requests and server responses, to log files; this allows the webmaster to collect statistics by running log analyzers on log files.

#Authentication, optional authorization request (request of user name and password) before allowing access to some or all kinds of resources.
#Handling not only static content but also dynamic content by supporting one or more related interfaces (SSI, CGI, SCGI, FastCGI, JSP, PHP, ASP, ASP .NET, Server API such as NSAPI, ISAPI, etc).
#HTTPS support (by SSL or TLS) to allow secure (encrypted) connections to the server on the standard port 443 instead of usual port 80.
#Content compression (i.e. by gzip encoding) to reduce the size of the response (to lower bandwidth usage, etc.).
#Virtual hosting to serve many web sites using one IP address.
#Large file support to be able to serve files whose size is greater than 2GB on 32 bit OS.
#Bandwidth throttling to limit the speed of responsive in order to not saturate the network and to be able to serve more clients.

5. Write down the function of w3 system.

Ans: Www (World Wide Web) is a system of interlinked documents accessed via the internet. With the web browser, one can view with pages that may contain text images, videos and other multimedia and navigate between them via hyperlink.
Function of W3:
The terms internet and www are often used in everyday speech without much distinction. However the internet and the www are not one and the same. The internet is a global system of interconnected computer networks.
In contrast the web is one of the services that runs on the internet. It is a collection of text documents and other resources linked by hyperlink and urls usually accessed by a web browser from a web server.
Viewing a web page on the www normally begins either by typing the URL of the page into a web browser. The web browser then initially are services of communication message in order to fetch and display in. As an example consider accessing a page with the URL:

http:// example.org/wiki/worldwideweb

First the browser resolves the server name portion of the URL into an internet protocol address using the globally distributed database known as the honest this look up returns an ip address such as 208.80.152.2
Several applications are called web browsers that make it easy to access the www. Two of the most popular name Firefox and Microsoft Internet Explorer.

6. What are the importance of !doctype and <meta> tag in webpage.

Ans: Doctype**:** The Doctype declaration should be the very first statement of an HTML page, even before the <html> tag. DOCTYPE refers to a Document Type Definition **(**DTD**)**. The DTD identifies the particular version of HTML being used. It contains all the HTML elements and attributes and specifies the rules for the markup language so that browsers can render the page content correctly.

This is important because HTML has undergone several major revisions, older tags have been deprecated, and newer tags and features and rules have been introduced. However, the DOCTYPE declaration is not an HTML tag; it is merely an instruction to the web browser indicating what version of HTML the Web page is written in. If you do not declare your DOCTYPE there is a good chance your page will display inconsistently in different browsers.

There are a number of different DOCTYPES that can be used to declare the specific version of HTML or XHTML (Extensible HyperText Markup Language) used on a Web page. Such as HTML5 DOCTYPE, XHTML (E**x**tensible HyperText Markup Language), XHTML 1.0 DOCTYPE, HTML 4.01 DOCTYPE.

The basic structure is as follows:

```
<!DOCTYPE html>
<html>
<head>
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <title>…..</title>
</head>
<body>
</body>
</html>
```

The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parsable. Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.

Example 1 - Define keywords for search engines:

<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">

Example 2 - Define a description of your web page:

<meta name="description" content="Free Web tutorials on HTML and CSS">

Example 3 - Define the author of a page:

<meta name="author" content="Safin Shahri">

Example 4 - Refresh document every 30 seconds:

<meta http-equiv="refresh" content="30">

7. Write down the problems created by Hoax, Spyware, Virus and Pop-ups.

Ans:

A virus is a program or piece of code that is loaded onto your computer without your knowledge and runs against your wishes. It can also replicate themselves.

Spyware is a type of malware that is installed on a computer without the knowledge of the owner in order to collect the owner's private information.

A virus hoax is a false warning about a computer virus. Typically, the warning arrives in an e-mail note or is distributed through a note in a company's internal network.

Pop-ups are often forms of online advertising on the World Wide Web intended to attract web traffic or capture email addresses.[1] Pop-ups are generally new web browser windows to display advertisements.

HTML (1 set)

8. What is HTML attributes? Explain the HTML elements.

Ans: HTML attributes: An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag. All attributes are made up of two parts: a name and a value:

- The name is the property you want to set. For example, the paragraph element in the example carries an attribute whose name is align, which you can use to indicate the alignment of paragraph on the page.

- The value is what you want the value of the property to be set and always put within quotations. The below example shows three possible values of align attribute: left, center and right.

Syntex is as follow:

<tag attribute-name=”attribute –value”> content </tag>

<img src=”bird.jpg” alt=”photo of a bird” height=”400” width=”600” >

Multiple attributes are separated by space

HTML elements: An HTML element is defined by a starting tag. If the element contains other content, it ends with a closing tag, where the element name is preceded by a forward slash as shown below with few tags:

Start Tag Content End Tag

 This is paragraph content.

<h1> This is paragraph content. </h1>

 This is paragraph content.

9. What is advantage of HTML5 over HTML4? Discuss any two tag name with example that are not present in HTML 4.

Or, Give two tag name with example that are not present in HTML 4.

Ans: The following are the advantages of HTML5 over HTML4:

Cleaner markup/improved code: Html5 will enable the web with designers to use cleaner, neater code. We can remove div tags and replace them with semantic HTML5 elements.

Elegant forms: women pad designer to use fancier forms. There will be different types of imports, search and different fields for different purposes.

Consistency: As websites will adopt the new HTML5 element we will see more consistency in terms of HTML used to code a web page on one site compared to another.

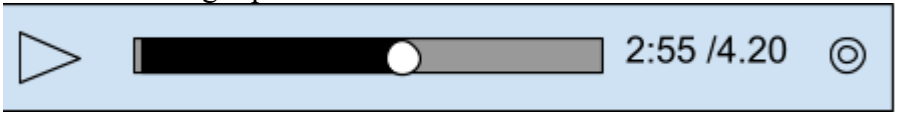
Supports reach media elements: HTML5 has an inbuilt capability to play audio and video and so we can bid goodbye to these plugin tags.

Offline application cache: HTML5 is first an offline application cache facility which will load the page the user has visited even if the user is temporarily offline. This feature will help the files to load much faster & reduces load on server.

Two tags name that are not present in html4

1. Audio: specifies audio content

<audio src=”/music.song.mp3” controls>

Output: 

2.Article: Specifies an articles

```
<article>
    <h2>This is a heading</h2>
    <p>This is paragraph</p>
</article>
```

Output: This is a heading
This is paragraph

10. What are the necessities of using HTML FORM? Explain HTML FORM tag with its different attributes.

Ans: Necessities of using HTML FORM:

HTML Forms are required when you want to collect some data from the site visitor. For example during user registration you would like to collect information such as name, email address, credit card, etc. A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application. Form Attributes Apart from common attributes, following is a list of the most frequently used form attributes:

Attribute Description:

action	Backend script ready to process your passed data.
method	Method to be used to upload data. The most frequently used are GET and POST methods.
target	Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
enctype	We can use the enctype attribute to specify how the browser encodes the data before it sends it to the server.

11. Describe basic tag of HTML.

Or, Explain the following tag with example

1. **<table>…….</table>**
2. **<img>…….</img>**
3. **<a href>…….**

Ans:

Tag	Description
<!DOCTYPE>	Defines the document type
<html>	Defines an HTML document
<head>	Contains metadata/information for the document
<title>	Defines a title for the document
<body>	Defines the document's body
<h1> to <h6>	Defines HTML headings
<p>	Defines a paragraph
 	Inserts a single line break
<hr>	Defines a thematic change in the content
<!--...-->	Defines a comment

<form>	Defines an HTML form for user input
<button>	Defines a clickable button
<img>	Defines an image
	Defines a section in a document
<header>	Defines a header for a document or section
<footer>	Defines a footer for a document or section

And so on.

1. Example of <table> tag

```
<!DOCTYPE html>
<html>
<body>
        <table>
            <tr>
                 <th>Month</th>
                <th>Savings</th>
            </tr>
            <tr>
                <td>January</td>
                <td>$100</td>
            </tr>
             <tr>
              <td>February</td>
```

```
    <td>$150</td>
  </tr>
 </table>
</body>
</html>
```

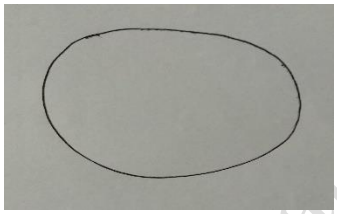
Output:

Month	Savings
January	$100
February	$150

2. Example of <img> tag

```
<!DOCTYPE html>
<html>
<body>

    <img src="circle.jpg" alt="it's a circle" width="200" height="150">

</body>
</html>
```

Output:



3.Example of <a href> tag

```
<!DOCTYPE html>

<html>
<body>

<h1>The a element</h1>

<a href="https://www.w3schools.com">Visit W3Schools.com! </a>
</body>
</html>
```

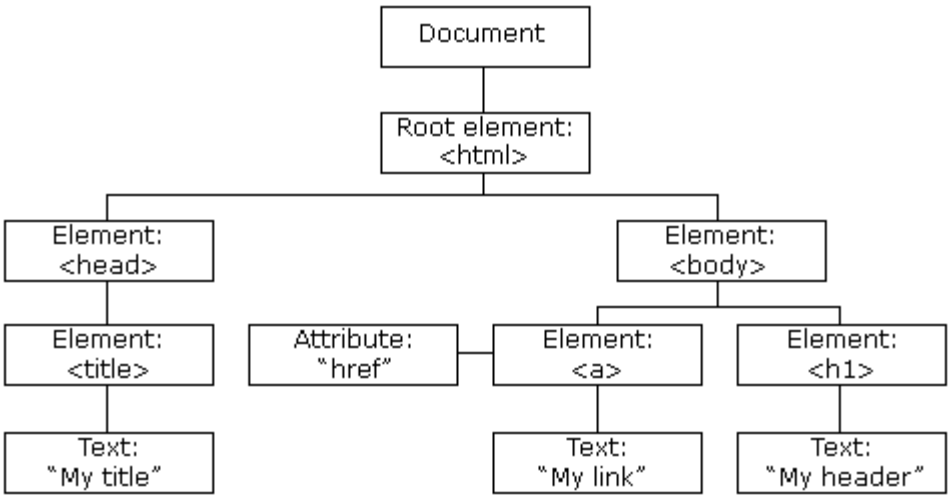
Output: Visit W3Schools.com!

12. What is HTML document tree? Explain the necessity of DOCTYPE in webpage design.

Ans: Each HTML document can actually be referred to as a document tree. We describe the elements in the tree like we would describe a family tree. There are ancestors, descendants, parents, children and siblings.

To understand the concept of a document tree as shown below-

```
<html>
    <head>
        <title>fabonnic series</title>
    </head>
<body>
    <h1>This is heading</h1>
    <p>This is paragraph</p>
    <a href="www.youtube.com">Youtube</a>
</body>
</html>
```



The necessity of DOCTYPE in webpage design are given below-

• The doctype declaration should be the very first thing in an HTML document, before the

tag.
• The <DOCTYPE> declaration tells the browsers what version of (X)HTML is being used so it will know how to display the page. The declaration may also provide a link to a text file (i.e. the loose DTD) which the browser uses to obtain further information.
• When performing HTML validation testing on a web page it tells the HTML (HyperText Markup Language) validator which version of (X)HTML standard the web page coding is supposed to comply with. When you validate your web page the HTML validator checks the coding against the applicable standard then reports which portions of the coding do not pass HTML validation (are not compliant).
• It tells the browser how to render the page in standards compliant mode.
• For example, specifying the doctype of your document allows you to use tools such as the Markup Validator to check the syntax of your (X)HTML. Such tools won't able to work if they do not know what kind of document you are using.

13. Write a code segment that will create a password field in a HTML form.

Ans:
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <form >

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

   </form>
   <div></div>
</body>
</html>
```

Output: Password :

14. What are the differences between HTML and XHTML? Discuss different types of links in HTML.

Ans: The differences between HTML and XHTML are as below-

Points	HTML	XHTML
Definition	Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications.	Extensible Hypertext Markup Language (XHTML) is a part of the family XML markup languages that mirrors the versions of HTML.
Base	HTML is based on SGML.	XHTML is XML based.
File Extension	HTML has file extensions of .html	XHTML has file extensions of .xhtml and .xht
Rules	HTML is strict to follow rules of structure.	XHTML is strict to follow rules of coding
Debugging	HTML is easier to debug.	XHTML is harder to debug.
Case Sensitivity	HTML is not case sensitive. The attributes and tags can either be lowercase or uppercase.	XHTML is case sensitive. The attributes and tags should be in lowercase.
Format	XHTML is well formatted and cleaner than HTML.	XHTML is well formatted and cleaner than HTML.
Origin	HTML was proposed by Tim Berners-Lee in 1987.	XHTML was proposed world Wide Web Consortium in 2000.
Developer	HTML is developed by W3C & WHATWG.	XHTML is developed by the World Wide Web Consortium.
Versions	HTML 2, HTML 3.2, HTML 4, HTML 5.	XHTML 1, XHTML 1.2, XHTML, XHTML 5.

CSS (1 set)

15. What is CSS? Write down the CSS syntax with example.

Ans: CSS: CSS stands for Cascading Style Sheets.CSS describes how HTML elements are to be displayed on screen, paper, or in other media.CSS saves a lot of work. It can control the layout of multiple web pages all at once. External style sheets are stored in CSS files.

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts −

- **Selector** − A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
- **Property** − A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be *color*, *border* etc.
- **Value** − Values are assigned to properties. For example, *color* property can have value either *red* or *#F1F1F1* etc.

You can put CSS Style Rule Syntax as follows −

16. Write down the different types of errors that may occur in CSS and explain how to handle those errors.

Ans:

Unknown Properties:

If an unknown property is encountered, a CSS-conforming user agent should ignore the declaratin. Given below-

h1 {color: red; trouble: right-here;}

The property "trouble" would simply be ignored and the rule would simply set the color. It does not matter what the position of the bogus property declaration is, the result should be the same as long as the declaration is otherwise well-formed.

h1 {trouble: right-here; color: red;}

Malformed Rules:

In the case where semicolons (;), colons (:), quotes ('or"), or curly braces ({ }) are misused, a browser should try to handle any unexpected characters and read the properties until a matching value can be found. As an example, consider the simple case of forgetting a semicolon:
h1 {color: red text-decoration: underline; font-style: italic;}
In this example there must be a semicolon after red otherwise the browser continue to parse the value of color as “red text-decoration: underline” before it sees a closing semicolon.
This error can be solved like this:
h1 {color red; text-decoration: underline; font-style: italic;}

Illegal or Unknown Property Values
CSS-conforming browsers must ignore a declaration with an illegal value. For example:
h1 {color: "green";}
Is incorrect, not because green is an illegal color, but because it is not the same as the keyword green when it is quoted. This should be below-
h1 {color: green;}

Unclosed Structures and End of File
A CSS browser should close all braces and quotes when it reaches the end of a style sheet. Otherwise it will show error.

```
1.  <style type="text/css"
2.          h1
3.          {
4.              color: green;
5.          }
6.      </style>
```

In this example, in line no.1 there is no closing braces. So this is an error.
To solve this error the opening style tag should be enclosed with a closing braces like as below-

```
1.  <style type="text/css">
2.          h1
3.          {
4.              color: green;
5.          }
6.      </style>
```

17. Why should you prefer CSS rather than using only HTML in web design?
Ans: The reasons are mentioned below why we use CSS:

BSS save time - You can write CSS once and then reuse the same sheet on multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages you want.

Pages load faster: If you are using CSS, you do not need to write HTML tag attributes everytime. Just write one css rule of a tag and apply it to all the occurrences of that tag. So less code means much faster download times.

Easy Maintenance - To make global change, simply change the style, and all elements in all the web pages will be uploaded automatically.

Superior styles to HTML - SS hasuch wider array of attributes than HTML so you can give a far better look to your HTML page in comparison to HTML attributes.

Multiple device Compatibility - Styleseat allows content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and smartphones or for printing.

Global web standards - Now HTML attributes are being deprecated and it is being recommended to use CSS. So it is a good idea to start using all the HTML pages to make them compatible with future browsers.

18. Discuss CSS inheritance with example.

Ans:

HTML elements can inherit CSS styles from their parent elements. This is called *CSS inheritance*. Here is a CSS inheritance example:

```
<div>
   <p>
      This text inherits the font-size of the parent div element.
   </p>
</div>
<style>
   div {
      font-family: Arial;
   }
</style>
```

In this example the div element has the font-family property set to Arial. This CSS property is inherited by the nested p element, so it will also have its CSS property font-family set to Arial.

HTML elements can inherit styles from more remote ancestors too, and not just from their parents. Look at this example:

```
<body>
   <div>
```

```
    <p>
      This text inherits the font-size of the parent div element.
    </p>
  </div>
  <style>
    body {
      font-family: Arial;
    }
  </style>
</body>
```

In this example the CSS property font-family is set on the body element. This CSS property is then inherited by the div and p elements.

Not all styles are inherited from a parent or ancestor element. Generally, styles that apply to text are inherited, whereas borders, margins and paddings and similar styles are not.

19. Discribe different CSS selector with example.

Or, Explain direct descendent selector with example.

Ans: CSS selectors are used to select the content you want to style. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute etc.
There are several different types of selectors in CSS.

1. CSS Element Selector
2. CSS Id Selector
3. CSS Class Selector

1) CSS Element Selector: The element selector selects the HTML element by name.

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

 p{
text-align: center;
color: blue;
}

 </style>
</head>

<body>

<p>This style will be applied on every paragraph.</p>
```

```
<p id="para1">Me too!</p>
<p>And me!</p>

 </body>
</html>
```

2) CSS Id Selector: The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element. It is written with the hash character (#), followed by the id of the element.
Let's take an example with the id "para1".

```
 <!DOCTYPE html>
<html>

 <head>
<style>

 #para1 {
text-align: center;

color: blue;

}

</style>
</head>

<body>
<p id="para1">Hello Javatpoint.com</p>

<p>This paragraph will not be affected.</p>
</body>

</html>
```

3) CSS Class Selector: The class selector selects HTML elements with a specific class attribute. It is used with a period character. (full stop symbol) followed by the class name.
Let's take an example with a class "center".

```
<!DOCTYPE html>
<html>

<head>
<style>

center {

text-align: center;

color: blue;
}
```

```
</style>
</head>

 <body>
<h1 class="center">This heading is blue and center-aligned.</h1>

<p class="center">This paragraph is blue and center-aligned.</p>
</body>

</html>
```

20. Compare different style sheet approaches.

Ans: There are three types of CSS which are given below:

- Inline CSS
- Internal or Embedded CSS
- External CSS
-

To add CSS styles to your website, you can use three different ways to insert the CSS. You can use an "External Stylesheet", an "Internal Stylesheet", or in "Inline Style". The benefit for using each depends on what you are doing with the Style. The following table explains the difference between them.

Different CSS style linking	

Internal Style sheet	An internal stylesheet holds the CSS code for the web page in the head section of the particular file. This makes it easy to apply styles like classes or id's in order to reuse the code. The downside of using an internal stylesheet is that changes to the internal stylesheet only affect the page the code is inserted into.Example:
	`<!DOCTYPE html>`
	`<html>`
	`<head>`
	`<style>`
	`body {`
	`  background-color: blue;`
	`}`
	`h1 {`
	`  color: red;`
	`  padding: 60px;`
	`}`
	`</style>`
	`</head>`
	`<body>`
	
	`<h1>This is Heading</h1>`
	`<p>This is our paragraph.</p>`

	</body> </html>

External Stylesheet	The External Stylesheet is a .css file that you link your website to. This makes it so that whatever you change in the .css sheet, will affect every page in your website. This prevents you from having to make many code changes in each page. This is for "global" site changes. Example:
	
	<!DOCTYPE html>
	<html>
	<head>
	<link rel="stylesheet" type="text/css" href="style.css" />
	</head>
	<body>
	  This is Heading
	 This is our paragraph.
	</body>
	</html>
	
	style.css file is as below-
	
	body {
	background-color: blue;
	}
	h1 {
	color: red;
	padding: 60px;
	
	}

Inline Styles	The Inline style is specific to the tag itself. The inline style uses the HTML "style" attribute to style a specific tag. This is not recommended, as every CSS change has to be made in every tag that has the inline style applied to it. The Inline style is good for one individual CSS change that you do not use repeatedly through the site. <html> <body style="background-color:black;"> This is Heading

	Something usefull here. </body> </html>

External Stylesheet is the best CSS approach. The External Stylesheet is a .css file that you link your website to. This makes it so that whatever you change in the .css sheet, will affect every page in your website. This prevents you from having to make many code changes in each page. This is for "global" site changes.

21. Explain different ways of inserting style sheets on a web page.

Ans: To add CSS styles to your website, you can use three different ways to insert the CSS. You can use an "External Stylesheet", an "Internal Stylesheet", or in "Inline Style". The benefit for using each depends on what you are doing with the Style. The following table explains the difference between them.

Internal Style sheet

An internal stylesheet holds the CSS code for the web page in the head section of the particular file. This makes it easy to apply styles like classes or id in order to reuse the code. The downside of using an internal stylesheet is that changes to the internal stylesheet only affect the page the code is inserted into.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: blue;
}
h1 {
    color: red;
    padding: 60px;
}
</style>
</head>
<body>

<h1>This is Heading</h1>
<p>This is our paragraph.</p>

</body>
</html>
```

External Style sheet

The External Stylesheet is a .css file that you link your website to. This makes it so that whatever you change in the .css sheet, will affect every page in your website. This prevents you from having to make many code changes in each page. This is for "global" site changes.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
      <h1> This is Heading </h1>
      <p>This is our paragraph.</p>
</body>
</html>
```

style.css file is as below-

```
body {
    background-color: blue;
}
h1 {
    color: red;
    padding: 60px;

    }
```

Inline Styles sheets

The Inline style is specific to the tag itself. The inline style uses the HTML "style" attribute to style a specific tag. This is not recommended, as every CSS change has to be made in every tag that has the inline style applied to it. The Inline style is good for one individual CSS change that you do not use repeatedly through the site.

```
<html>

<body style="background-color:black;">

    <h1 style="color:white;padding:30px;"> This is Heading </h1>

    <p style="color:white;">Something usefull here.</p>

</body>
</html>
```

JavaScript (1 set)

22. What is scripting language? Write down the differences between Programming language and Scripting language.

Ans: Scripting language: A scripting or script language is a programming language for a special run-time environment that automates the execution of tasks; the tasks could alternatively be executed one-by-one by a human operator. Scripting languages are often interpreted, rather than compiled.Scripting language, which can be embedded within HTML commonly are used to add functionality to a web page, such as different menu styles or graphics displays or to serve dynamic advertisements.

Differences between Programming language and Scripting language are given below:

Factor	Scripting Language	Programming Language
Type of language	Interpreter based	Compiler based
Usage	To combine existing components	To develop from scratch
Running	Inside other program (dependent)	Independent of a parent program
Conversion	High level instructions converted to machine language	Full program converted to machine lang one time
Design	Makes coding simple and fast	Gives full use of language

Compilation	No need to compile	Needs to compile first
Coding type	It is a small piece of code	It is a full code of a program
Time to develop	Less time as required less code	More time as you need to write the full c
Complexity	Easy to write and use	Difficult
Interpretation	It is interpreted in another program	Stand-alone compile result, no need to b interpreted by another program
Length	Only a few and short lines of coding	Numerous lines for every function
Hosting requirement	Requires a host	Self-executable, no host needed
Support	Limited or no support to user interface design, data types, and graphic design	Rich support to user interface design, da and graphic design

Cost	Low maintenance	High maintenance
Example	JavaScript, PHP, Ruby, Perl, VB Script, etc.	C, C++, Java, Pascal, C#, VB, Basic, CO etc.

23. Write down the common mistakes in JavaScript.

Ans: Accidentally Using the Assignment Operator
JavaScript programs may generate unexpected results if a programmer accidentally uses an assignment operator (=), instead of a comparison operator (==) in as if statement.
var x = 0;
if (x ==10)
This if statement returns false (as expected) because x is not equal to 10.

Expecting Loose Comparison

In regular comparison, data type does not matter. This if statement true
var x = 10;

var y = 10;
In strict comparison, data type does matter. This if statement returns false:
var x = 10;
var y = "10";
if (x === y)

Confusion Addition & Concatenation
Addition is about adding numbers.
Concatenation is about adding strings.
In JavaScript both operations use the same + operator.
var x = 10 +5; // the result in x is 15
var x = 10 +"5"; // the result in is "105"

Misunderstanding Floats

All numbers in JavaScript are stored as 64-bits Floating point numbers (Floats).

```
var x = 0.1;
var y = 0.2;
var z = x+y;   // the result in z wili not be 0.3
```

To solve the problem above, it helps to multiply and divide

```
var z = (x*10+Y*10)/ 10;     // z will be 0.3
```

Breaking a JavaScript String

JavaScript will allow you to break a statement into two lines:

Example.l

```
var x =

"Hello World";
```

But, breaking a statement in the middle of a string will not work:

Example 2

```
var x "Hello
World!";
```

Expecting Block Level Scope

JavaScript does not create a new scope for each code block.

It is true in many programming languages, but not true in JavaScript.

This code will display the value of i (10), even OUTSIDE the for loop block

```
for (var i = 0; i < 10; i++) {
// some code
return i;
```

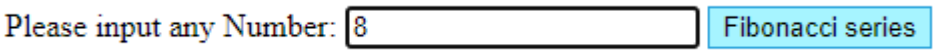
24. Write a JavaScript program to calculate the Fibonacci series.

Ans:

```
<!DOCTYPE html>

<html>

<head><title>fabonnic series</title>

<script>

function fib(){

   var A = 0;

   var B = 1;

   var C;

   var N = document.getElementById("number").value;
```

```
    document.write(A+"<br />");
    document.write(B+"<br />");
for(var i=3; i <= N;i++)
    {
        C = A + B;
        A = B;
        B = C;
document.write(C+"<br />");
    }
}
</script>
</head>
<body>
 Please input any Number: <input id="number">
<button type="button" onclick="fib()">Fibonacci series</button>
</body>
</html>
```

Output:



0
1
1
2
3
5
8
13

25. Write a JavaScript program to find the factorial of a given number.

Ans:

```
<!doctype html>
```

```
<html>
<head>
<script>
function show(){
var i, no, fact;
fact=1;
no=Number(document.getElementById("num").value);
        for(i=1; i<=no; i++)  {

        fact= fact*i;
        }
document.getElementById("answer").value= fact;
}
</script>
</head>
<body>
Enter Num: <input id="num">
<button onclick="show()">Factorial</button>
<input id="answer">
</body>
</html>
```

Output:

Enter Num: 5 Factorial 120

26. Describe with example, the syntax of declaring a variable, Array and object in JS.

Ans: Declaring variable

All JavaScript variables must be identified with unique names.
These unique names are called identifiers.
Identifiers can be short names (like x and y) or more descriptive names (age, sum, Volume).
The general rules for constructing names for variables (unique identifiers) are:
Names can contain letters, digits, underscores, and dollar signs.
Names must begin with a letter
• Names can also begin with S and _ (but we will not use it in this tutorial)
• Names are case sensitive (y and Y are different variables)
• Reserved words (tike JavaScript keywords) cannot be used as names
Variables are declared with the var keyword as follows

```
var x = 5;

var y = 5;

var z = x+y;
```

Declaring Array
In JavaScript, array is a single variable that is used to store different elements. It is often used when we want to store list of elements and access them by a single variable.

Unlike most languages where array is a reference to the multiple variable, in JavaScript array is a single variable that stores multiple elements-
Syntax:
var array_name = [iteml, item2,…]
var cars = ["Saab", "Volvo", "BMW"];

Declaring Object

When a JavaScript variable is declared with the keyword " new ", the variable is created as an object:

```
var x = new String(); // Declares x as a String object.
var y = new Number(); // Declares y as a Number object.
var z = new Boolean(); // Declares z as a Boolean object.
```

27. Explain the way JS handle array with examples.

Ans: Array: An array is an object that can store a collection of items. Arrays become really useful when we need to store a large amounts of data of the same type.

If you have a list of items (a list of car names, for example), storing the Cars in single variables could look like this:

var carl = "Saab";
var car2 = "Volvo";
var car3 = "BMW";
However, what if you want to loop through the cars and find a specific one? And what if you had not 3 cars, but 300?
The solution is an array!
An array can hold many values under a single name, and you can access the values by referring to an index number.

Creating an Array

Using an array literal is the easiest way to create a JavaScript Array.

Syntax:

var array_name = [iteml, item2,…….];
Example
var cars = ["Saab", "Volvo". "BMW"];

Access the Elements of an Array
You access an array element by referring to the index number.
This statement accesses the value of the first element in cars:
var name = cars[O];

Changing an Array Element

This statement changes the value of the first element in cars:
cars[O] = "Opel";

28. What is events? Describe different events used in JS.

Or, Describe the following events in JS with example.

i) Input event

ii) Mouse event

iii) Load event

Ans: Event: The change in the state of an object is known as an **Event**. In html, there are various events which represents that some activity is performed by the user or by the browser. When javascript code is included in HTML, js react over these events and allow the execution.

1. Input event:

onblur - When a user leaves an input field
onchange - when a user changes the content of an input field
onchange - when a user selects a dropdown value
onfocus - I will when and input field gets focus
onselect - when input text is selected
onsubmit - when a user clicks the submit button
onreset - when a user clicks the reset button
onkeydown - when a user is pressing/ holding down a key
onkeypress - when a user is pressing or holding down a key
onkeyup - When a user releases a key
onkeydown vs onkeyup – both

2. Mouse event:

onmouseover/onmouseout – When the mouse passes over an element
onmousedown/onmouseup – When pressing/releasing a mouse button
onmousedown – When mouse is clicked: Alert which element
onmousedown – When mouse is clicked: Alert which button
onmousemove/onmouseout – When moving the mouse pointer over/out of an image
onmousemove/onmouseout – When moving the mouse over/out of an image

3. Load event

onload- When the page has been loaded
onload – When an image has been loaded
onerror – When an error occurs when loading an image
onunloaded – When the browser closes the document
onresize – When the browser window is resized

29. Explain document.getElementById() function with example.

Ans:

The document object represents the whole html document.When html document is loaded in the browser, it becomes a document object. It is the root element that represents the html document. It has properties and methods. By the help of document object, we can add dynamic content to our web page.

We can access and change the contents of document by its various methods and getElementById() is one of them.

The getElementById() is a DOM method used to return the element that has the ID attribute with the specified value. This is one of the most common methods in the HTML DOM and is used almost every time we want to manipulate an element on our document. This method returns null if no elements with the specified ID exists. The ID should be unique within a page. However, if more than one element with the specified ID exists, it returns the last element in the javascript code.

Syntax:
document.getElementById(element_ID)

Parameter: This function accepts single parameter *element_ID* which is used to hold the ID of element.
Return Value: It returns the object of given ID. If no element exists with given ID then it returns null.

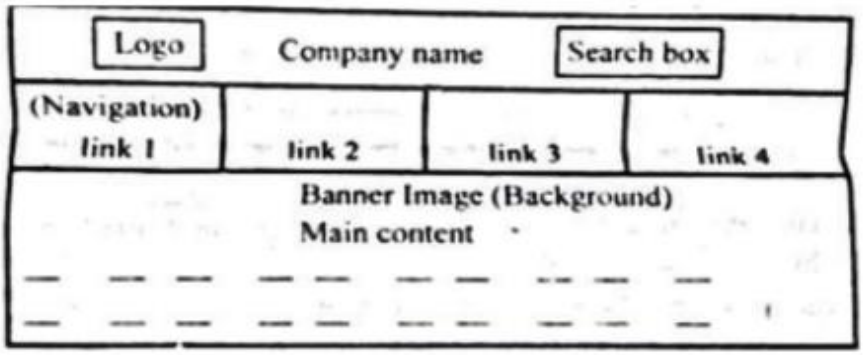
For an example:

```
<html>

<body>

<p id="element">GetElementById</p>

<script>

   var s = document.getElementById("element").innerHTML;

   document.write(s);

</script>

</body>

</html>
```

Output

```
GetElementById
GetElementById
```

30) Create an HTML webpage to show the following output:



Ans:

HTML Code(index.html):

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="utf-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <title> Webpage </title>
   <link rel="stylesheet" href="style.css">
</head>
<body>
   <header>
        <div class="row head">
       <ul class="head-1">
          <li><a href="#"><img src="logo.svg" alt="!" class="logo"></a></li>
          <li><a href="#" class="company-1"> Web Master </a></li>
          <li>
             <form action="" class="search-form">
                <input type="text" placeholder="search">
                <button>search</button>
```

```
                </form>
            </li>
        </ul>
    </div>
        <nav class="menu">
         <ul>
            <li> <a href="#"> Home</a></li>
            <li><a href="#"> About</a></li>
            <li><a href="#"> Gallary</a></li>
            <li><a href="#"> Contact</a></li>
         </ul>
        </nav>
    </header>
    <section>
      <div class="row">
          <div class="hero-text-box">
            <h2>Hi there. We build awesome websites and mobile apps.</h2>
            <a href="#contact" class="btn btn-hero">Work With Us</a>
          </div>
      </div>
    </section>
</body>
</html>
```

CSS Code (style.css):

```
*{
    margin: 0px;
    padding: 0px;
```

```
    list-style: none;
    text-decoration: none;
}
.head{
    width: 100%;
    background: blue;
    overflow: auto;
    height: 50px;
}
.head-1 {
    padding-top: 10px;
}
.head ul li{
    float: left;
}
.logo{
    float: left;
    height: 30px;
    width: 150px;
    text-align: center;
    padding-left: 100px;

}

.head .company-1{
    color: #fff;
    padding-left: 250px;
    font-size: 20px;
```

```
    font-family: sans-serif;
    text-decoration: none;
    text-align: center;
 }
.search-form{
    margin-right: 100px;
    padding-left: 220px;
    padding-top: .5rem;
}
input [type= text] {
    padding: 7px;
    border: none;
    font-size: 16px;
    font-family: sans-serif;
}
 button {
    float: right;
    background: orange;
    color: white;
    cursor: pointer;
    position: relative;
    font-family: sans-serif;
    border: none;
    font-size: 16px;
}
.menu{
    width: 100%;
    background: #142b47;
```

```
    overflow: auto;
}
.menu ul{
    margin: 0px;
    padding: 0px;
    list-style: none;
    line-height: 60px;
    float: center;
    padding-left: 60px;

}
.menu li{
    float: left;
    padding-left: 80px;
}
.menu ul li a {
    background: #142b47;
    text-decoration: none;
    width: 200px;
    display: block;
    text-align: center;
    color: #fff;
    font-size: 1.8rem;
    font-family: sans-serif;
    letter-spacing: 0.5px;
}
section {
    background: url(nature.jpg);
```

```
    background-repeat: no-repeat;
    background-size: cover;
    background-position: center center;
    background-attachment: fixed;
    height: 85vh;
    position: relative;
}
h1 {
     font-size: 42px;
     color: #fff;
 }
.btn {
    background: #333;
    color: #fff;
    text-decoration: none;
    text-transform: uppercase;
    padding: 15px 30px;
    display: inline-block;
    margin-top: 30px;
    font-size: 16px;
    text-align: center;
}
.btn-hero {
    background: #eb7d4b;
    border-color: #c86a40;
}
```

31) Write down the CSS and HTML code to design the following menu:

Home	Department	Contact
	CSE	
	BBA	
	T H M	

Or, Write down the CSS and HTML code to design the following menu:

Home	Department	Contact
	CSE	
	BBA	

Ans: **HTML Code (index.html):**

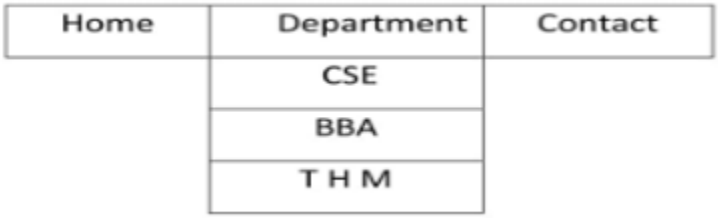
```
<!DOCTYPE html>
<html lang="en">
<head>

   <link rel="stylesheet" href="style.css">
</head>
<body>

 <nav class="nav main-nav">

      <ul>
         <li><a href="Index.html">Home</a></button></li>
           <div class="dropdown">
            <li><a href="Index.html">Department</a></button></li>
            <div class="dropdown-content">
               <a href="#">CSE</a>
               <a href="#">BBA</a>
                <a href="#">T H M</a>    [31 er 'or' a ei line ta hobe na. baki shob thik ache.]
            </div>
          </div>
          <li><a href="contact.html">Contact</a></li>

      </ul>
   </nav>

</body>
</html>
```

CSS code (style.css):

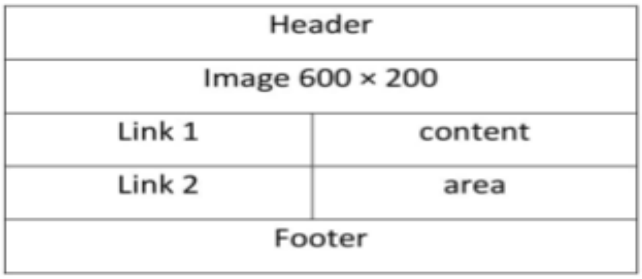
```
nav li {
   display: inline;
```

```
    font-size: 30px;
}
.nav a {
    display: inline-block;
    padding: 12px 16px;
    color: black;
    text-decoration: none;
}
.main-nav {
 text-align: center;
 font-size: 1em;
}
  .dropdown {
    position: relative;
    display: inline-block;
  }


.dropdown-content {
    display: none;
    position: absolute;
    padding:  12px 16px;
    font-size: 30px;
    text-decoration: center;
    z-index: 1;
  }

.dropdown:hover .dropdown-content {
    display: block;
  }
```

Output:



32) Write down the CSS and HTML code to design the following webpage :-



Ans:

HTML code (index.html):

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="utf-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <title> Webpage </title>
   <link rel="stylesheet" href="style.css">
</head>
<body>
   <header>
      <nav class="menu">
         <ul>
            <li> <a href="#"> Home</a></li>
            <li><a href="#"> About</a></li>
            <li><a href="#"> Gallary</a></li>
            <li><a href="#"> Contact</a></li>
         </ul>
      </nav>
   </header>
  <div class="row">
```

```
   <img src="nature.jpg" alt="!" class="image">
 </div>
<section>
   <div class="left">
      <div>
         <ul>
            <li> <a href="#" class="link_1">Link 1</a></li>
            <li> <a href="#" class="link_2">link 2</a></li>
          </ul>
      </div>
    </div>
    <div class="right">
       <div class="right_1">
         <p>content</p>
       </div>

       <div>
         <p class="right_2">Area</p>
       </div>
    </div>
 </section>
<footer class="footer-section">
 <div class="row">
   <ul>
    <li><a href="#">Facebook</a></li>
    <li><a href="#">Twitter</a></li>
    <li><a href="#">Google+</a></li>
    <li><a href="#">LinkedIn</a></li>
```

```
      <li><a href="#">Dribble</a></li>
      <li><a href="#">GitHub</a></li>
    </ul>
  </div>
</footer>
</body>
</html>
```

CSS code (style.css):

```
*{
    margin: 0;
    padding: 0;
    list-style: none;
    text-decoration: none;
}
.menu{
    width: 100%;
    background: #142b47;
    overflow: auto;
}
.menu ul{
    margin: 0px;
    padding: 0px;
    list-style: none;
    line-height: 60px;
    padding-left: 60px;
    }
```

```
.menu li{
    float: left;
    padding-left: 80px;
}
.menu ul li a {
    background: #142b47;
    text-decoration: none;
    width: 200px;
    display: block;
    text-align: center;
    color: #fff;
    font-size: 1.8rem;
    font-family: sans-serif;
    letter-spacing: 0.5px;
}
.image{
    width: 100%;
    height: 390px;
}
section{
    height: 110px;
}
.row{
    padding: 0px !important;
    margin: 0px !important;
}
.left{
    height: 100px;
```

```
    width: 50%;
    background: #fff;
    float: left;
}
.left ul{
    list-style: none;
}
.left ul li{
    padding: 10px
}
.left ul li a{
    text-decoration: none:;
    color: #000;
    font-size: 30px;
}
.link_1 {
    padding-left: 200px;
    padding-right: 355px;
    text-align: center;
    border: 2px solid;
    border-color: #000;
    padding-top: 3px;
    padding-bottom: 10px;
}

.link_2 {
    padding-bottom: 8px;
    padding-top: 7px;
```

```
    padding-left: 200px;
    padding-right: 363px;
    text-align: center;
    border: 2px solid;
    border-color: #000;
}

.right{
    background: #fff;
    width: 50%;
    height: 100px;
    float: right;
    margin-top: 5px;
}
.right_1 {
    padding-top: 6px;
    padding-left: 310px;
    padding-right: 200px;
    text-align: center;
    border: 2px solid;
    border-color: #000;
    color: #000;
    font-size: 30px;
    padding-bottom: 7px;
}

.right_2 {
    padding-top: 5px;
```

```
    padding-left: 310px;
    padding-right: 200px;
    text-align: center;
    border: 2px solid;
    border-color: #000;
    color: #000;
    font-size: 30px;
    padding-bottom: 8px;
}
.footer-section {
    background: #344b8e;
    padding: 20px 0;
    text-align: center;
    border-top: 10px rgba(255, 255, 255, 0.3);
}
.footer-section li {
    display: inline-block;
    margin: 0 20px;
}
.footer-section li a {
    color: #fff;
    text-decoration: none;
    font-size: 20px;
}
```

33. Design a simple webpage with html and css code which contains:

- An image of 300*300 pixel

-Three navigation buttons such as: Home, about us, contact us.

-Page background color is: green.

Ans:

```
<!DOCTYPE html>
<html>
<head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <title> Simple Webpage </title>
        <link rel="stylesheet" href="style.css">
</head>
<body>
<header>
       <nav>
        <div class="row">
           <ul class="main-nav">
              <li><a href="# ">Home</a></li>
              <li><a href="# ">About Us</a></li>
              <li><a href="#" >Contact Us</a></li>
              </ul>
        </div>
      </nav>
  </header>
<section>
          <div class="row">
        <img src="safin.jpg" alt="!" class="image" height="300" width="300">
```

```
        </div>
    </section>
  </body>
</html>
```

CSS code (style.css):

```
        body{
            background: green;
        }
        ul li {
    list-style-type: none;
   }
    ul li a {
    color: #fff;
    text-decoration: none;
        }
    .main-nav {
    padding-left: 40%;
    margin-top: 40px;
    }
.main-nav li {
    display: inline-block;
    margin-left: 40px;
}
.image{
            padding-left: 40%;
            padding-top: 10%;
}
```

PHP & MySql (2 set)

34. What is PHP? How is it related to HTML?

PHP: PHP stands for Hypertext Preprocessor. PHP is a general-purpose scripting language especially suited to web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994. The PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

PHP is a server-side scripting language. That means a PHP script is executed on the server, the output is built on the server, and the result is sent as HTML to the client browser for rendering. It's natural to mix PHP and HTML in a script, but as a beginner, it's tricky to know how to combine the PHP code with the HTML code.

When it comes to using PHP in HTML, there are two different approaches. The first is to embed the PHP code in your HTML file itself with the .html extension—this requires a special consideration, which we'll discuss in a moment. The other option, the preferred way, is to combine PHP and HTML tags in .php files.
Since PHP is a server-side scripting language, the code is interpreted and run on the server side. For example, if you add the following code in your index.html file, it won't run out of the box.

```
<!DOCTYPE html>
<html>
<head>
<title>Embed PHP in a .html File</title>
</head>
<body>
<h1><?php echo "Hello World" ?></h1>
</body>
</html>
```

First of all, don't worry if you haven't seen this kind of mixed PHP and HTML code before, as we'll discuss it in detail throughout this article. The above example outputs the following in your browser:

```
1
<?php echo "Hello World" ?>
```

So as you can see, by default, PHP tags in your .html document are not detected, and they're just considered plain text, outputting without parsing. That's because the server is usually configured to run PHP only for files with the .php extension.
If you want to run your HTML files as PHP, you can tell the server to run your .html files as PHP files, but it's a much better idea to put your mixed PHP and HTML code into a file with the .php extension.

35. Write down the difference between GET and POST.

Differences between GET and POST are given below:-

	GET	POST
History	Parameters remain in browser history because they are part of the URL.	Parameters are not saved in browser history.
Bookmarked	Can be bookmarked.	Cannot be bookmarked.
BACK button/resubmit behavior	GET requests are re-executed but may not be re-submitted to the server if the HTML is stored in the browser cache.	The browser usually alerts the user that data will need to be re-submitted.
Encoding type	application/x-www-form-urlencoded	Multipart/form-data or application/x-www-form-urlencoded Use multipart encoding for binary data.
Parameters	Can send but the parameter data is limited to what we can stuff into the request line (URL). Safest to use less than 2K of parameters, some servers handle up to 64K	Can send parameters, including uploading files, to the server.
Hacked	Easier to hack for script kiddies.	More difficult to hack.
Restrictions on form data type	Yes, only ASCII characters are allowed.	No restrictions. Binary data is also allowed.
Security	GET is less secure compared to POST because data sent is part of the URL. So it's saved in browser history and server logs in plaintext.	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs.
Restrictions on form data length	Yes, since form data is in the URL and URL length is restricted. A safe URL length limit is often 2048 characters but varies by browser and web server.	Not restricted.
Usability	GET method should not be used when sending passwords or other sensitive information.	POST method used when sending passwords or other sensitive information.

Visibility	The GET method is visible to everyone (it will be displayed in the browser's address bar) and has limits on the amount of information to send.	POST method variables are not displayed in the URL.
Cached	Can be cached.	Not cached.

36. How is the array declared in PHP? Also explain various types of arrays with proper examples.

Ans:

Declaration array in PHP: An array is a data structure that stores one or more similar type of values in a single value. For example if you want to store 100 numbers then instead of defining 100 variables its easy to define an array of 100 length.

In PHP, the array() function is used to create an array: array();

Syntax for an array
array (value1, value2, value3, ……);

Example:-

```
<?php
$cars = array("VOLVO", "BMW", "TOYOTA");
echo "I like".$cars[0] ", " $cars[1] "and" $cars[2] "." ;
?>
```

Result : I like VOLVO, BMW and TOYOTA.

In PHP there are three types. They are--

- **Indexed arrays**: These are arrays where each element is referenced by a numeric index, usually starting from zero. For example, the first element has an index of 0, the second has an index of 1 as so on.

  There are two ways to create indexed arrays:
  The index can be assigned automatically (index always starts at 0), like this:
  $cars = array("Volvo", "BMW", "Toyota");

  Or, the index can be assigned manually:
  $cars[0] = "Volvo";
  $cars[1] = "BMW";
  $cars[2] = "Toyota";

- **Associative arrays**: This type of array is also referred to as a hash or map with associative arrays each element is referenced by a starting index.
  For example, we might create an array element representing a customer's age & give it an index of “age”.
  Syntax for associative array
  array (key => value, key=> value, key => value, …..);

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
Or,
$age['Peter'] = "35";
$age['Ben'] = "37";
$age['Joe'] = "43";
```

- **Multidimensional arrays**: A multidimensional array is an array containing one or more arrays. PHP supports multidimensional arrays that are two, three, four, five, or more levels deep. However, arrays more than three levels deep are hard to manage for most people.

  We can store the data from the table above in a two-dimensional array, like this:

```
$cars = array (
  array("Volvo",22,18),
  array("BMW",15,13),
  array("Saab",5,2),
  array("Land Rover",17,15)
);
```

37. What are the techniques used for form validation and verification in PHP?

Ans:

Validation means check the input submitted by the user. There are two types of validation are available in PHP. They are as follows —

• Client-Side Validation — Validation is performed on the client machine web browsers.

• Server Side Validation — After submitted by data, The data has sent to a server and perform validation checks in server machine.

The HTML form we will be working at in these chapters, contains various input fields: required and optional text fields, radio buttons, and a submit button:

PHP Form validation example

***required field**

Name: ★

E-mail: ★

Website:

Comment:

Gender: ○ Male ○ Female ○ Other *

Submit

Your Input:

The validation rules for the form above are as follows:

Field	Validation Rules
Name	Required. + Must only contain letters and whitespace
Email	Required. + Must contain a valid email address (with @ and .)
Website	Optional. If present, it must contain a valid URL
Comment	Optional. Multi-line input field (textarea)
Gender	Required. Must select one

First we will look at the plain HTML code for the form:
Text Fields
The name, email, and website fields are text input elements, and the comment field is a textarea. The HTML code looks like this:
Name: <input type = "text" name = "name">
E-mail: <input type = "text" name = "email">

Website: <input type = "text" name = "website">
Comment: <textarea name = "comment" rows = "5" col= "30"> </textarea>

Radio Buttons
The gender fields are radio buttons and the html code looks like this:
Gender:

<input type = "radio" name = "gender" value = "female"> Male
<input type = "radio" name = "gender" value = "female"> Female
<input type = "radio" name = "gender" value = "female"> Other

The Form Element
The HTML code of the form looks like this:
<form method="post" action="<?php echo "htmlspecialchars($_SERVER["PHP_SELF"]);?>">
When the form is submitted, the form data is sent with method="post"

(38) What are the basic requirements to running a PHP program?

Ans:

To start using PHP, you can:

- Find a web host with PHP and MySQL support.
- Install a web server on your own computer, and the install PHP and MySQL.

Use a Web Host with PHP support:

If your server has activated support for PHP you do not need to do anything. Just create some .php files, place them in your web directory and the server will automatically parse them for you. You do not need to compile anything or install any extra tools. Because PHP is free, most web hosts offer PHP support. However, if your server does not support PHP, you must:

- Install a web server
- Install PHP
- Install a database, such as MySQL

(39) Describe following global variable in PHP:--

(i) S-GET

(ii) S-POST

(iii) S-REQUEST

Ans:

(i) $-GET

$_GET is a PHP super global variable which is used to collect form data after submitting an HTML form with method="get".

$_GET can also collect data sent in the URL.

Assume we have an HTML page that contains a hyperlink with parameters:

```
<html>
<body>
<a href="test_get.php?subject=PHP&web=W3schools.com">Test $GET</a>
</body>
</html>
```

When a user clicks on the link "Test $GET", the parameters "subject" and "web" are sent to "test_get.php", and you can then access their values in "test_get.php" with $_GET.

The example below shows the code in "test_get.php

```
<html>
<body>

<?php
echo "Study " . $_GET['subject'] . " at " . $_GET['web'];
?>

</body>
```

(ii) $-POST

$_POST is a PHP super global variable which is used to collect form data after submitting an HTML form with method="post". $_POST is also widely used to pass variables.

The example below shows a form with an input field and a submit button. When a user submits the data by clicking on "Submit", the form data is sent to the file specified in the action attribute of the <form> tag. In this example, we point to the file itself for processing form data. If you wish to use another PHP file to process form data, replace that with the filename of your choice. Then, we can use the super global variable $_POST to collect the value of the input field:

```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // collect value of input field
  $name = $_POST['fname'];
  if (empty($name)) {
    echo "Name is empty";
  } else {
    echo $name;
  }
}
?>

</body>
</html>
```

Output-

(iii) $-REQUEST

$_REQUEST is a PHP super global variable which is used to collect data after submitting an HTML form.

The example below shows a form with an input field and a submit button. When a user submits the data by clicking on "Submit", the form data is sent to the file specified in the action attribute of the <form> tag. In this example, we point to this file itself for processing form data. If you wish to use another PHP file to process form data, replace that with the filename of your choice. Then, we can use the super global variable $_REQUEST to collect the value of the input field:

```
<!DOCTYPE html>
<html>
<body>
```

```
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
   // collect value of input field
   $name = htmlspecialchars($_REQUEST['fname']);
   if (empty($name)) {
      echo "Name is empty";
   } else {
      echo $name;
   }
}
?>

</body>
</html>
```

(40) Write down the procedure of creating database and table in MySQL.

Ans:

Create a MySQL Database

The CREATE DATABASE statement is used to create a database in MySQL
The following examples create a database named "myDB";

```
< ?php
$servername = "localhost" ;
$username = "username" ;
$password = "password" ;

// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn-> connect_error) {
        die("Connection failed: " . $conn->connect_error);
        }
// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query ($sql) === TRUE) {
        echo "Database created successfully";
        }
        else {
        echo "Error creating database: " . $conn->error;
        }
$conn->close();
?>
```

Create a MySql Table

The CREATE TABLE statement is used to create a table in MySQL
We will create a table named "MyGuests", with five columns: "id", "firstname", "lastname", "email" and "reg_date":

```
CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP
)
```

```
< ?php

$servername = "localhost" ;
$username = "username";
$password = " password " ;
$dbname =  "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname) ;
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

// sql to create table
$sql = "CREATE TABLE myGuests (
id INT(6)  UNSIGNED  AUTO_INCREMENT  PRIMARY KEY,
firstname  VARCHAR(30)  NOT NULL,
lastname  VARCHAR(30)  NOT NULL,
email  VARCHAR(5Ø),
reg_date TIMESTAMP
) ";
if ($conn->query ($sql) === TRUE) {
echo "Table MyGuests created successfully";
}
else {
        echo "Error creating database: " . $conn->error;
        }

$conn-> close();
```

?>

41. Discuss the steps to establish a connection from PHP to MySQL with sample code.

Ans:

There are 6 steps to establish connection from PHP to MySql which is given below:-

Step1- Connecting to a MySQL database: We need our MySQL username, password & server address(most likely localhost on 127.0.0.1). Create a filemanager.php file & open & close the PHP code with tags before the html, we can put regular html after it e.g.

```
<?php
 $db = mysqli-connect(“localhost”,”username”,”password:);
 if(!$db){
          die(“Database connection failed miserable : mysqli- error());
         }
   <html>
      <head></head>
      <body>

      </body>
   </html>
```

Step 2- Selecting the database to use: We will need here is the database name if we don’t know the table name either get it through MySQL via the shell in php-myadmin.

Step3- Performing database query: The query is actually performed in the body of the html page. So additional php opening and closing tag will be required.

Step4- Put the data on the page: Here we are taking the result as an array & displaying it row after row in a loop. Instead of the numbers we can use the names of the field in the table to display the data in “double quotes”.

Step5-Closing off the connection: Closing the connection will require another set of opening and closing php tags after the closing html tag.

Step6- Staying the data surround: The data input call of php step 3 & 4 in a div gives it a class, style in css. We can wrap is row in a html tag such as a h2 & p tag by echoing before and after each row the make a distant selection that targets the h2 & the p tag.

Sample code :-

```
   <?php
 $db = mysqli-connect(“localhost”,”username”,”password:);
 if(!$db){
          die(“Database connection failed miserable : mysqli- error());
         }
    $db = select = mysqli-select-db(“databasename”,$db);
     If (!db-select){
```

```
        die("Database selection also failed miserably:" mysql-error());
       }
      ?>
  <html>
     <head></head>
     <body>
      <div class="css style">
        <?php
          $result = mysqli-query("select*FORM mytable",$db);
          if(!$result){
                    die("Database query failed:"mysql-error());
                   }
          While ($row=mysql-fetch-array($result)))
                {
                  echo "<h2>";
                  echo  $row[1] . " ";
                  echo "</h2>";
                  echo "<p>";
                  echo  $now[2] . " ";
                  echo "</p>";
                 }
          </div>
     </body>
  </html>
  <?php
    mysql-close($db);
   ?>
```

(42) What are the pros and cons of MYSQL compared to other relational database management Systems used in web applications?

Ans:

Advantages:-

- It is easy to use.
- It includes solid data security layers that project sensitive data from includes.
- It's inexpensive.
- MySQL is fast and scalable.
- It manages memory very well.
- It supports novell cluster services.
- It runs on many operating systems.
- It supports several development interfaces.

Disadvantages:-

- It's got a few stability issues.
- It suffers from relatively poor performance sealing.
- Development is not community driven and hence has lagged.
- It's functionality tends to be heavily dependent on add-ons.
- Developers may find some its limitations to be frustrating.

(43) Explain the following functions with syntax, purpose and example:-- (i) My sqli----connect ( ) (ii) My sqli---close ( ). (iii) My sqli---query ( )

Ans: (i) mysql-connect ( ):- The mysql-connect() function is used to open a new connection to the mysql server.

Syntax-

mysql-connect(host,uname,password,dbname,port,socket);

parameter

host - optional specifies a host name over IP address.
username- optional specifies the MySQL username.
password- optional specifies the MySQL password.
dbname- optional specifies the default database to be used.
port- optional specifies the port number to attempt to connect to the MySQL server.
socket - optional specifies the socket or named pipe to be end.

(ii) mysql-select-db( ) : The mysql-selectdb() function is used to change the default database for the connection.

Syntax-

mysql-connect(connection,dbname);

parameter

connection required specifies the mysql-connection to be used.
dbname- Required specifies the default database to be used.

(iii) mysql-query ( ): The mysql-query() function is used to perform a query against the database.

Syntax-

mysql-query(connection,query,resultmode);

parameter

connection - required specifying the mysql-connection to be used.
query - required specifies the query string.
resultmode - optional a constant either;
.MySQL-USE-RESULT (use this if we have to retrieve longer amount of data)
.MySQL-STORE-RESULT(This is default.)

(iv) mysql-close ( ) : The mysql-close() function is used to close a previously opened database connection.

Syntax-

mysql-close(connection);

parameter

connection required specifies the mysql-connection to close.

(44) What are the use of session and cookie variables in state management? Explain with example.

Ans:

Use of cookie Variable:- A cookie variable sent by the browser. A cookie is typically a small text file that the server embeds on the computer. A cookie must be assigned before any other output is client.

Syntax: setcookie(name,value,expire,path,domain,secure)

Example:
```
<?php
    $value =”my cookie value”;
    //send a cookie that expires in 24 hours.
    setcookie(“cookie_name, $cookie_value, time()+3600*24);
    ? >
```

Use of session variable: A php session variable issued to store information about or change setting for a user session. Session variables hold information about one single user and are available to all pages in one application.

Example:- Starting a php session.

```
<? Php session_start();
 ?>
<html>
   <head>
   </head>
   <body>
   </body>
</html>
```

45. Write short notes:- (i) Session; (ii) Cookie; (iii) Request

Ans:

Session: An alternative way to make data accessible across the various pages of an entire website is to use a PHP Session.

A session creates a file in a temporary directory on the server where registered session variables and their values are stored. This data will be available to all pages on the site during that visit.

A session is started with the session_start() function.

Example:

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>
```

```
<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>

</body>
</html>
```

Output: Session variables are set.

Cookie: A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

A cookie is created with the setcookie() function.

Syntax

setcookie(*name, value, expire, path, domain, secure, httponly*);

Only the *name* parameter is required. All other parameters are optional. The setcookie() function must appear BEFORE the <html> tag.

The value of the cookie is automatically URLencoded when sending the cookie, and automatically decoded when received.

Example:

```
<!DOCTYPE html>
<?php
$cookie_name = "user";
$cookie_value = "Alex Porter";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
    echo "Cookie '" . $cookie_name . "' is set!<br>";
    echo "Value is: " . $_COOKIE[$cookie_name];
}
?>
```

</body>
</html>

Output: Cookie 'user' is set!
Value is: John Doe

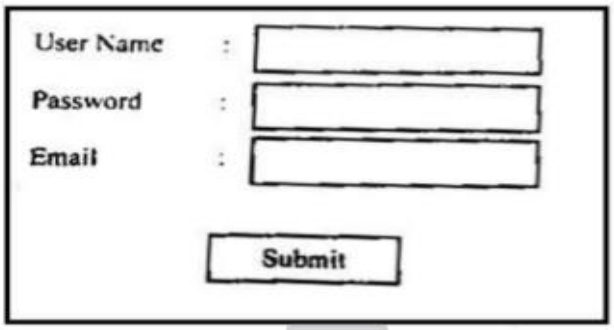
Request: PHP $_REQUEST is a PHP super global variable which is used to collect data after submitting an HTML form.

In PHP, it supports various request methods depends on which the capabilities and functionalities to be applied on request data before sending it to the server, will be varied. The list of PHP supported request methods are,

- GET
- POST
- PUT
- HEAD

$_GET and $_POST method are widely used.

46) Write down the HTML, PHP and SQL code to store the following information into a database :



Ans:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
```

```
<body>
   <form action="process.php" method="POST">
    <div>
       <label for="username">Username :</label>
       <input type="text" name="username">
    </div>


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


    <div>
       <label for="email">Email :</label>
       <input type="text" name="email">
    </div>

    <button type="submit">Submit</button>

   </form>

</body>
</html>
```

Output:

Username: [ ]

Password: [ ]

Email: [ ]

Submit

When a user clicks the submit button of the registration form in the above example, data inserted in the form is sent to process.php which connects to the mysql database server retrieves forms field using $_POST variable and executes the insert query to add the records. Here is he complete code of process.php :-

```
<?php
$link=mysqli_connect('localhost', 'root', '', 'user');

$username = mysqli_real_escape_string($link, $_POST['username']);
```

```
$password = mysqli_real_escape_string($link, $_POST['password']);
$email = mysqli_real_escape_string($link, $_POST['email']);
$sql = "INSERT INTO user (username, password, email) VALUES ('$username', '$password', '$email')";

if(mysqli_query($link,$sql)){
   echo "Records added successfully";
}

else{
   echo "ERROR:Could not find database";
}

mysqli_close($link);

?>
```

47. Write PHP code to store data in student table and retrieve data from student table – using PHP Form.

Ans:

```
<?php

$host="localhost";

$dbUser="root";

$dbPass=" ";

$dbName="students";

$connect=mysqli_connect($host,$dbUser,$dbPass,$dbName);

if($connect==false){

echo"Not connected".mysqli_error($connect);

}

?>

<!Doctype HTML>

<html>

<head>

<title>Login Form</title>

</head>
```

```
<body>
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>" method="POST">
Name<input type="text" name="name"><br>
Roll<input type="number" name="roll"><br>
Class<input type="number" name="class"><br>
Sec<input type="text" name="sec"><br>
<input type="submit" name="submit">
</form>
<?php
if(isset($_POST['submit'])){
$name=$_POST['name'];
$roll=$_POST['roll'];
$class=$_POST['class'];
$sec=$_POST['sec'];
$matchQuery="INSERT INTO students_table (Name,Roll,Class,Sec)
values('$name','$roll','$class','$sec')";
$run=mysqli_query($connect,$matchQuery);
$result=mysqli_query($connect,"SELECT * From students_table");
if (mysqli_num_rows($result)>0) {
?>
  <table>
<tr>
   <td>Name</td>
   <td>Roll</td>
   <td>Class</td>
   <td>Sec</td>
  </tr>
<?php
$i=0;
```

```
while($row = mysqli_fetch_array($result)) {
?>
<tr>
    <td><?php echo $row["Name"]; ?></td>
    <td><?php echo $row["Roll"]; ?></td>
    <td><?php echo $row["Class"]; ?></td>
    <td><?php echo $row["Sec"]; ?></td>
</tr>
<?php
$i++;
}
?>
</table>
 <?php
}
else{
    echo "No result found";
}
}
?>
</body>
</html>
```

48.

48) Write down simple PHP and MySQL code for adding and retrieving data in the following table :-

Department
CSE
BBA

Ans:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Department Info</title>
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet">
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/2.6.0/umd/popper.min.js" ></script>
   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" ></script>
</head>
</head>
<body>
   <nav class="navbar navbar-expand-md bg-dark navabar-dark">
      <a class="navbar-brand" href="#">CRUD OPERATION THIRD ANSWER</a>
      <button class="navabr-toogler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">
        <span class="navbar-toogle-icon"></span>
```

```
    </button>
  </nav>
  <div class="container-fluid mt-2">
   <div class="col-md-10">
     <h2 class="text-center">FILL THE FORM</h2>
     <hr>
   </div>
  </div>
  <div class="row">
   <div class="col-md-4">
     <h3 class="text-center text-info">ADD RECORD</h3>
     <form action="" method="post">
       <div class="form-group">
        DEPARTMENT: <input type="text" name="department" value="" class="form-control" placeholder="enter your department name" required>
       </div>
       <div class="form-group">
           <input type="submit" name="submit" class="btn btn-primary btn-block">
       </div>
     </form>
   </div>
  </div>
   <div class="col-md-6">
     <h3 class="text-center text-info" >RECORDS PRESENT IN THE DATABASE</h3>
      <table class="table table-bordered table-hover mt-4">
        <thead>
          <tr class="text-center">
            <th> DEPARTMENT </th>
          </tr>
```

```
            </thead>
            <tbody>
            <?php

              include 'dbcon.php';
              $select="SELECT * from department" ;

              $query=mysqli_query($con,$select);
              while($result=mysqli_fetch_assoc($query)){
                 ?>
                 <tr class="text-center">
                  <td> <?php echo $result ['department'];?>
                  </td>
                 </tr>
                 <?php
              }
            ?>
            </tbody>
          </table>
     </div>
    </div>
</div>
</body>
</html>

<?php

<?php
```

```
include 'dbcon.php';
if(isset($POST['submit'])){
    $department-$POST['department'];
    $insert="INSERT INTO department(department) VALUES('$department')";

    $query=mysqli_query($con,$insert);

    if($query){
    ?>
      <script>
         alert("inserted sucessfully");
      </script>
      <?php
    }else{

       ?>
       <script>
          alert("insertion unsuccessul.")
       </script>
       <?php
    }
}

?>
```

Output:

Solved By: Safin Shahri

Special Thanks to Md. Sefayat Shameem, Mridul Talha & Omra Haque Anika