ASSIGNMENT - 1

1. How internet works?

The internet is a global network of interconnected computers that communicate via standardized protocols. It uses TCP/IP (Transmission Control Protocol/Internet Protocol) to facilitate data transmission between devices.

2. How browser works?

A browser retrieves web pages from servers and renders them on a user's device. It interprets HTML, CSS, and JavaScript to display content and handle user interactions.

3. What is Server?

A server is a computer or system that provides resources, data, services, or functionality to other computers, known as clients, over a network.

4. What are the types of server available?

Servers can be categorized based on their functions:

- Web Server: Serves web pages (e.g., Apache, Nginx).
- Database Server: Manages database operations (e.g., MySQL, PostgreSQL).
- File Server: Stores and manages files (e.g., FTP servers).
- Application Server: Executes applications (e.g., Java EE servers like Tomcat).

5. What is SEO? Importance of SEO?

SEO (Search Engine Optimization) is the practice of optimizing web content to improve its visibility and ranking in search engine results. It's important for driving organic traffic to websites and increasing their online presence.

6. What is Accessibility?

Accessibility refers to designing products, devices, services, or environments that can be used by people with disabilities. In web context, it ensures websites are usable by everyone, including those with visual, auditory, motor, or cognitive impairments.

7. What is Markup Language?

A markup language is a system for annotating text to provide structure and formatting instructions. HTML (HyperText Markup Language) is the most common markup language used for creating web pages.

8. What is HTML?

HTML is a markup language used to create and structure web pages. It defines the structure and layout of content on the web.

9. What is browser engine?

A browser engine (layout engine) interprets HTML and CSS, and applies rendering rules to display web content. Examples include Blink (used by Chrome and Opera) and Gecko (used by Firefox).

10. What is rendering engine? Share the available rendering engines.

A rendering engine is a component of a browser that displays the requested content. Examples include Blink (Chrome, Opera), Gecko (Firefox), WebKit (Safari), and Trident (Internet Explorer).

11. What is JavaScript Engine? Share the available JS engines. Purpose of JS Engine?

A JavaScript engine executes JavaScript code in a browser. Examples include V8 (Chrome, Node.js), SpiderMonkey (Firefox), and JavaScriptCore (Safari). Its purpose is to interpret and execute JavaScript instructions.

12. How website works?

A website works by storing web pages on a server and delivering them to users via browsers. When a user requests a page, the server processes the request, retrieves the page, and sends it to the browser for display.

13. What is Data Structure?

A data structure is a way of organizing and storing data in a computer so that it can be accessed and modified efficiently. Examples include arrays, linked lists, trees, and graphs.

14. Explain Tree Data Structure?

A tree data structure consists of nodes where each node has zero or more child nodes. It's hierarchical, with a root node at the top and branches leading to leaf nodes. Used for organizing hierarchical data like file systems.

15. What is user agent? Share the list and its purpose?

A user agent is a software application that acts on behalf of a user. In web context, it's the string sent by a browser to identify itself to servers. Examples include:

- Chrome: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.104 Safari/537.36
- Firefox: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0) Gecko/20100101 Firefox/84.0
- Its purpose is to help servers deliver content tailored to different browsers and devices.

16. What is Hypertext?

Hypertext is text displayed on a computer or other electronic device with references (hyperlinks) to other text that the reader can immediately access.

17. What are HTML Tags?

HTML tags are keywords or commands surrounded by angle brackets (< >) that provide instructions for formatting and displaying content on a web page.

Examples include <html>, <head>, <body>, <p>, , etc.

18. What are HTML Attributes?

HTML attributes provide additional information about HTML elements. They are placed within the start tag and can modify an element's behavior or appearance.

19. What are HTML Elements?

HTML elements are building blocks of HTML pages. They consist of a start tag, content, and an end tag (where applicable), and describe the structure and semantics of content on a web page.

20. How do you convert elements to a tree?

Elements in an HTML document are converted into a tree-like structure known as the Document Object Model (DOM). This model represents the hierarchical structure of the HTML document, with each element as a node and relationships defined by parent-child connections.

21. What is DOCTYPE?

DOCTYPE (Document Type Declaration) is an instruction to the web browser about what version of HTML the page is written in. It helps browsers to render content correctly.

22. What are the ways we can save HTML file?

HTML files can be saved using various methods:

- Save from a text editor (e.g., Notepad, VS Code).
- Export from a web development tool (e.g., Dreamweaver).
- Save a webpage using a browser's "Save As" feature.

23.What is charset? Why do we need to use this?

Charset (character set) specifies the encoding used for interpreting characters in a document. It ensures that text displays correctly, especially when dealing with different languages and special characters.

24. What is metadata? What is the purpose of it?

Metadata is data that provides information about other data. In the context of web pages, metadata includes information like page title, author, description, and keywords. It helps search engines understand and index the content properly.

25. Explain Web Application Architecture?

Web Application Architecture refers to the structural design of web applications. It typically includes components such as:

- Client-side: Browser-based interface (HTML, CSS, JavaScript).
- Server-side: Application logic and database management (e.g., using MVC architecture).
- Database: Stores persistent data.
- Networking: Communication protocols (HTTP/HTTPS).