Assignment 1

WEB TECHNOLOGIES

1. How internet works?

The internet works through a network of interconnected computers and servers globally.

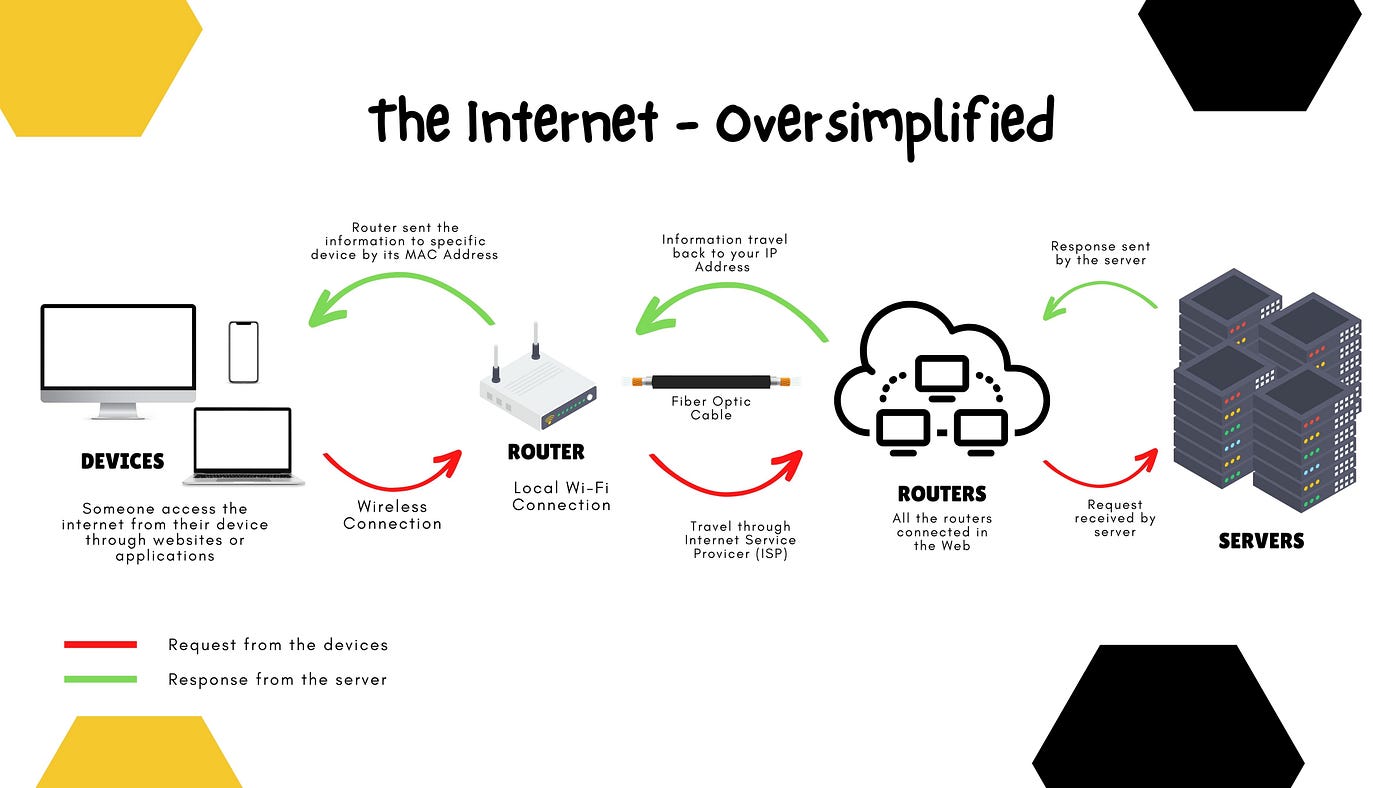
- Data Transmission: Information is broken down into packets.

- Routing: Packets travel through various routers across networks.

- Protocols: TCP/IP (Transmission Control Protocol/Internet Protocol) governs how data packets are transmitted and received.

- Addressing: Each device has an IP address (e.g., IPv4 or IPv6) for identification.

- Servers: Specialized computers host websites, services, and data, responding to requests from clients (e.g., browsers).



2. How browser works?

A web browser fetches and displays web pages. Key steps include:

- Request: Browser sends a request to a server for a web page.

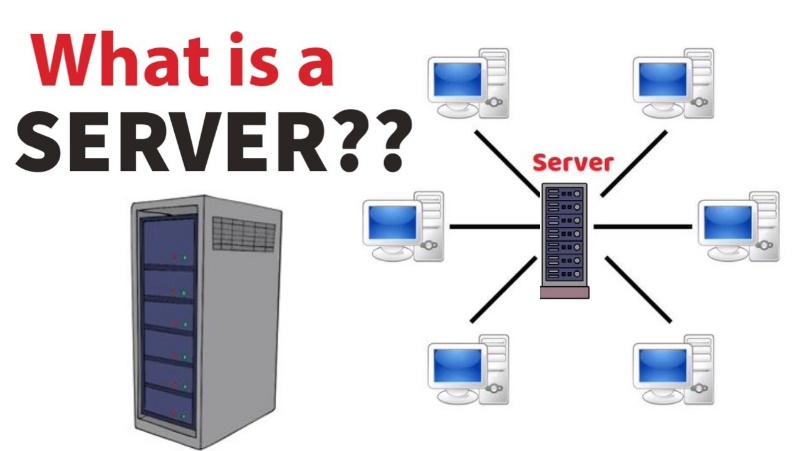
- Response: Server sends back HTML, CSS, JavaScript, and media files.

- Rendering: Browser parses HTML (structure), CSS (style), and JavaScript (behavior).

- Display: Content is rendered visually on the screen, enabling user interaction.

3. What is Server?

A server is a computer or software that provides services, data, resources, or functionality to other computers (clients) over a network.



4. What are the types of server available?

Types of servers include:

- Web Server: Delivers web pages (e.g., Apache, Nginx).

- Database Server: Manages databases (e.g., MySQL, PostgreSQL).

- File Server: Stores and manages files (e.g., FTP servers).

- Mail Server: Handles email communication (e.g., SMTP, IMAP servers).

- Application Server: Executes applications and provides data processing services.

5. What is SEO? Importance of SEO?

SEO (Search Engine Optimization) is the process of optimizing websites to rank higher in search engine results pages (SERPs). Importance:

- Visibility: Higher rankings drive more organic traffic.

- Credibility: Top results are seen as more trustworthy.

- ROI: Effective SEO can lead to higher conversion rates and ROI.

6. What is Accessibility?

Accessibility ensures that websites, applications, and tools are usable by people with disabilities. It involves:

- Perception: Content should be perceivable (e.g., screen readers for visually impaired).

- Operation: Interfaces should be operable (e.g., keyboard navigation).

- Understanding: Information should be understandable (e.g., clear language).

- Robustness: Content should be robust enough to work with assistive technologies.

7. What is Markup Language?

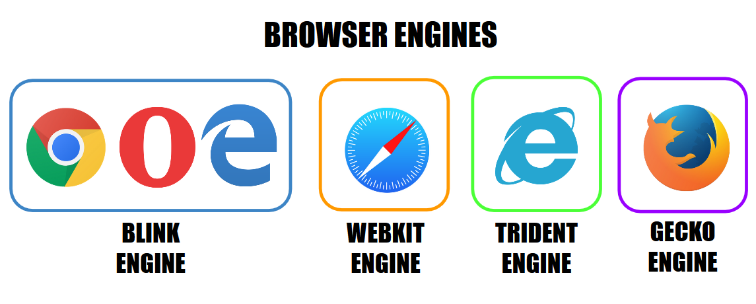
A markup language is a system for annotating text to define its structure, presentation, or semantics. HTML (HyperText Markup Language) is a primary example, used for creating and structuring web pages.

8. What is HTML?

HTML (HyperText Markup Language) is the standard markup language for creating web pages and web applications. It defines the structure of content using a variety of tags and attributes.

9. What is browser engine?

A browser engine (also known as a layout engine or rendering engine) is responsible for rendering web pages. It interprets HTML, CSS, and JavaScript and displays the parsed content on the screen. Examples include Blink, WebKit, and Gecko.



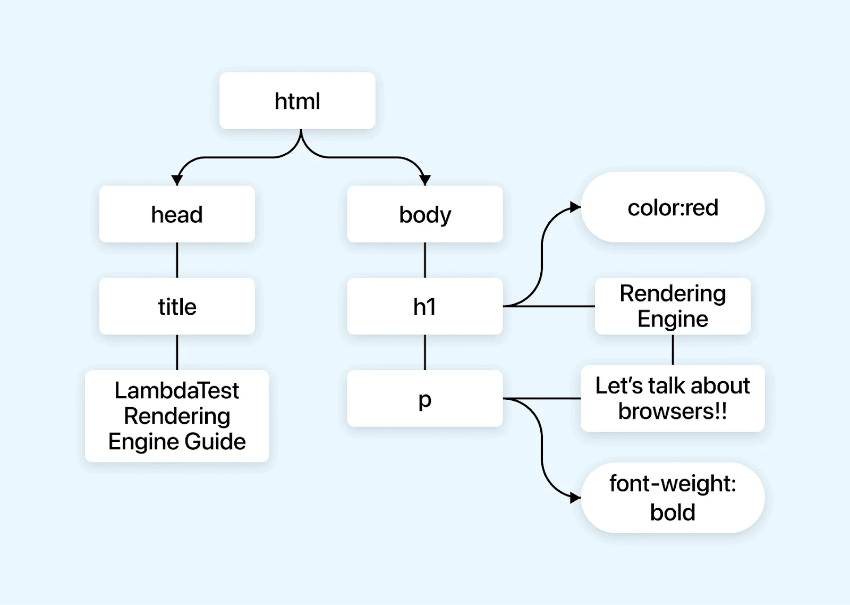
10. What is rendering engine? Share the available rendering engines.

A rendering engine is part of a browser or application responsible for rendering content into a visual format. Examples include:

- Blink: Used by Google Chrome and Microsoft Edge.

- WebKit: Used by Safari.

- Gecko: Used by Mozilla Firefox.



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

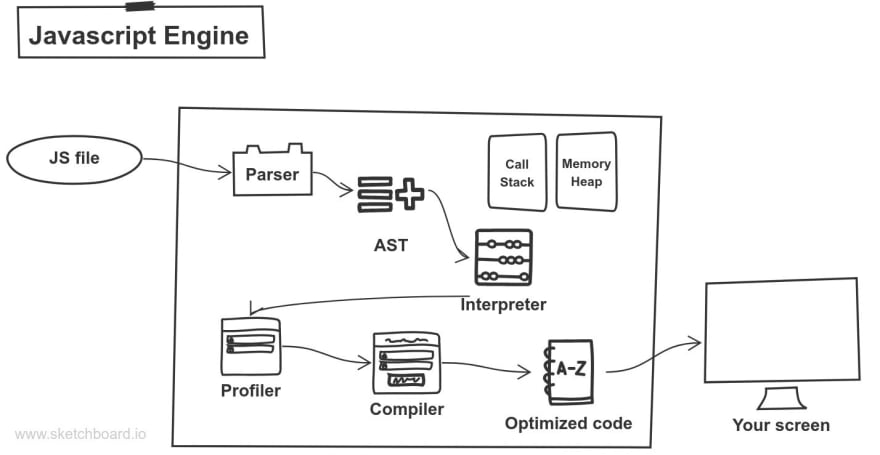
A JavaScript engine executes JavaScript code in a web browser or server-side environment. Examples include:

- V8: Used by Google Chrome and Node.js.

- SpiderMonkey: Used by Mozilla Firefox.

- JavaScriptCore: Used by Safari.

Purpose: To interpret and execute JavaScript code, enabling dynamic functionality on web pages.



12. How website works?

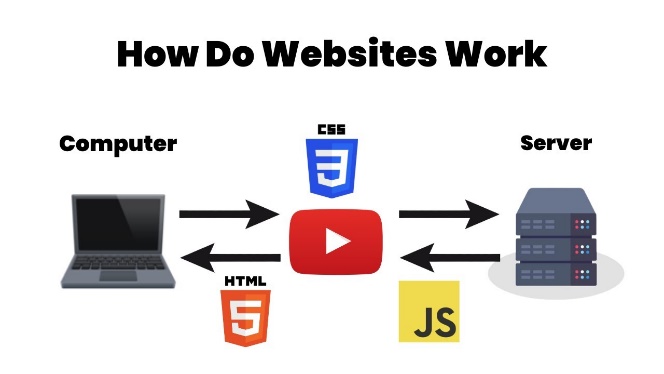
Websites work through the interaction of clients (web browsers) and servers:

- Request: Browser requests a web page.

- Response: Server sends HTML, CSS, and JavaScript files.

- Rendering: Browser renders content visually.

- Interaction: Users interact with forms, links, and media.



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 connected by edges, where each node can have zero or more child nodes. It's hierarchical and is commonly used to represent hierarchical relationships such as in HTML DOM.

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

A user agent is a string that identifies the browser and operating system to web servers. Purpose:

- Content Negotiation: Servers send optimized content based on user agent.

- Feature Detection: Websites adapt features based on capabilities.

Example:

- Mozilla/5.0

16. What is Hypertext?

Hypertext is text that contains links to other texts, enabling non-linear navigation between related topics or documents.

17. What is HTML Tags?

HTML tags are keywords enclosed in angle brackets (<>) used to define the structure and elements of web pages. They indicate how browsers should format and display content.

18. What is HTML Attributes?

HTML attributes provide additional information about HTML elements. They modify the element's behavior or appearance. Examples include `src`, `href`, `class`, `id`, etc.

19. What is HTML Elements?

HTML elements are the building blocks of HTML pages. They represent various components such as headings (`<h1>`), paragraphs (`<p>`), images (`<img>`), links (`<a>`), forms (`<form>`), etc.

20. How do you convert elements to a tree?

HTML elements are parsed into a tree-like structure called the Document Object Model (DOM) by the browser's rendering engine. Each HTML element becomes a node in this tree, representing the structure of the web page.

21. What is DOCTYPE?

DOCTYPE (Document Type Declaration) specifies the type and version of HTML used in a web document. It ensures browsers render content correctly according to the specified HTML standard.

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

HTML files can be saved using:

- Text editors (e.g., Notepad++, Sublime Text)

- Integrated development environments (IDEs) (e.g., Visual Studio Code, Atom)

- Browser developer tools (e.g., Save Page As)

- Command-line tools (e.g., wget, curl)

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

Charset (character encoding) specifies how characters are represented in a web page. It ensures that text is displayed correctly, especially when using non-ASCII characters or different languages.

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

Metadata provides information about other data. In web development, metadata is often included in `<meta>` tags within the `<head>` section of HTML. It includes information such as page description, author, keywords, and viewport settings. Metadata helps browsers and search engines understand and present web pages appropriately.

25. Explain simple Web Application Architecture?

Web application architecture defines the interactions between applications, middleware systems, and databases to ensure multiple applications can work together. It's essentially the framework that maintains the structure and function of a web application. Here’s a simple breakdown of the main components and their roles:

1. Client-Side (Front-End)

This is the part of the web application that interacts with the user.

* Web Browser: The user's interface to interact with the web application.
* User Interface (UI): Designed with HTML, CSS, and JavaScript, it dictates the look and feel of the application.
* Client-Side Logic: This includes all the scripts and frameworks (e.g., React, Angular, Vue.js) that run on the user's device, providing dynamic content and user interaction.

2. Server-Side (Back-End)

This is the part of the web application that processes business logic and interacts with the database.

* Web Server: Handles HTTP requests from the client, serving web pages, and executing server-side code.
* Application Server: Executes the server-side code, processes user requests, and returns the appropriate response. Technologies like Node.js, Ruby on Rails, Django, and Spring Boot are often used.
* Server-Side Logic: This includes the business logic, authentication, and authorization processes, usually written in server-side languages like Java, Python, Ruby, or PHP.

3. Database

Stores and manages data needed by the application.

* Database Management System (DBMS): A system like MySQL, PostgreSQL, MongoDB, or SQLite that handles data storage, retrieval, and management.
* Database Queries: Written in SQL or other query languages, these are used to interact with the database to create, read, update, or delete data.

4. Communication

Defines how data moves between the client-side and server-side.

* HTTP/HTTPS Protocols: Used for communication between the client and server. HTTPS is the secure version, encrypting data transfer.
* API Endpoints: Specific paths on the server that handle requests and send responses, often designed using REST (Representational State Transfer) or GraphQL.

5. Middleware

Software that connects different parts of the web application and allows them to communicate.

* Authentication Middleware: Manages user authentication and authorization.
* Caching Middleware: Stores frequently accessed data in memory to improve performance.
* Logging Middleware: Keeps records of application events and user activities for monitoring and debugging.

