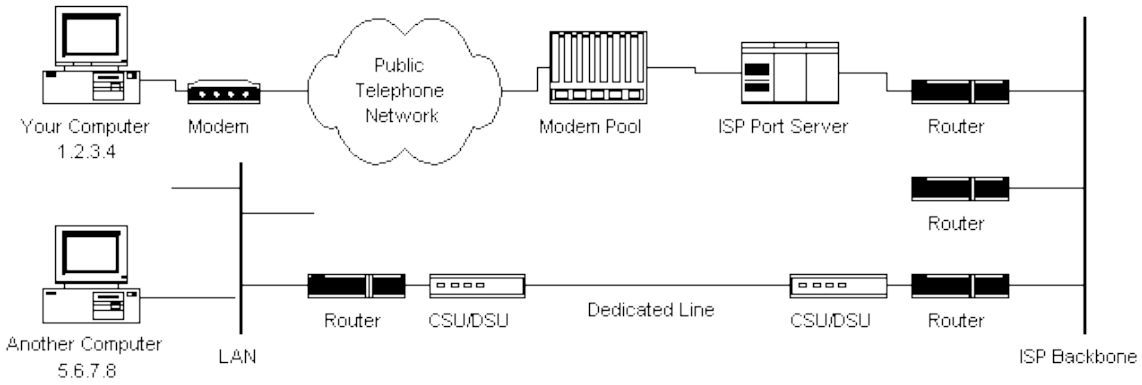
WEB TECHNOLOGIES S SHIVANI (23MCA28)

Assignmant-1

1. How internet works?

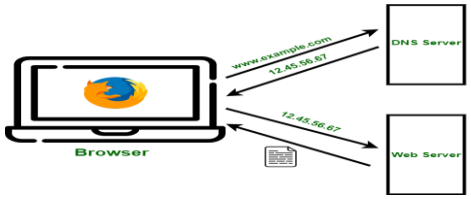
The internet function is an interconnected network of computers and devices worldwide. When you access a website or send data, your device connects to a local network or internet service provider, which then links to larger networks and internet backbones. Your data is broken into small packets, each routed independently across multiple networks using standardized protocols like TCP/IP. These protocols ensure packets reach their destination and are reassembled correctly. The Domain Name System (DNS) translates human-readable website names into IP addresses, allowing your device to locate the correct server. Web servers host content, which your browser requests and displays. This complex system enables near-instantaneous global communication and information sharing, forming the backbone of our digital world.



2. How browser works?

A web browser is a software application that acts as an intermediary between users and the vast world of online content. When a user enters a URL or clicks a link, the browser initiates a request to the appropriate web server.

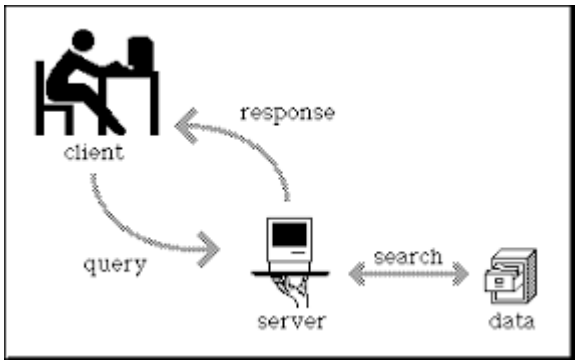
- Upon receiving the server's response, typically in the form of HTML, CSS, and JavaScript files, the browser begins its core functions. It parses the HTML to construct the Document Object Model (DOM), applies CSS styles to format the content, and executes JavaScript to add dynamic functionality.
- Throughout this process, the browser interprets and renders the code, transforming it into the visual and interactive web pages we see on our screens. Additionally, modern browsers handle various background tasks such as caching for faster load times, implementing security measures to protect users, and supporting plugins to extend functionality, all while providing a seamless and intuitive user interface for navigating the web.



3. What is server?

A server is a fundamental component of computer networks and the internet, acting as a central hub for providing services, resources, and functionality to other computers or devices, known as clients. In essence, it's a computer program or device designed to process requests and deliver data to other programs or devices over a network. Software runs specialized server software to handle client requests efficiently.

- Network role acts as a central point for storing, managing, and distributing information or services.



4. What are the types of server available?

1. Web servers
2. Application servers
3. Database servers
4. File servers
5. Mail servers
6. Print servers
7. Game servers
8. DNS servers
9. FTP servers
10. Proxy servers

5. What is SEO? Importance of SEO?

SEO stands for Search Engine Optimization. It's a set of practices designed to improve a website's visibility and ranking in search engine results pages (SERPs). SEO involves optimizing website content, structure, and external factors to make it more attractive to search engines like Google. This includes keyword optimization, improving site speed, creating quality content, and building backlinks.

1. Increased visibility: Helps websites rank higher in search results, attracting more potential visitors.
2. Organic traffic: Drives free, targeted traffic to your website.
3. Credibility: Higher search rankings often equate to greater trust from users.
4. User experience: Good SEO practices often improve overall website usability.

6. What is Accessibility?

Web accessibility is the practice of designing and developing websites and web applications to be usable by everyone, regardless of their abilities or disabilities. It involves creating digital content that can be easily perceived, operated, and understood by all users. This approach incorporates various techniques such as using proper HTML structure, providing text alternatives for non-text content, ensuring keyboard navigation, maintaining adequate color contrast, and making content compatible with assistive technologies, web accessibility not only benefits users with disabilities but often improves the overall user experience for all, resulting in more intuitive and adaptable web interfaces.

7. What is markup Language?

A markup language is a way to add special labels or tags to a document that tell computers how to display or process the text. It's like giving instructions to a computer about what each part of a document is and how it should look. For example, you might use tags to say "this is a heading" or "this text should be bold." Common markup languages include HTML for websites and Markdown for simple text formatting. These languages help organize information and make it easier for both people and computers to understand the structure of a document.

Examples of markup language:

- HTML (Hypertext Markup Language): Used for creating web pages
- XML (Extensible Markup Language): A versatile language for storing and transporting data
- Markdown: A lightweight markup language for creating formatted text using plain-text syntax.

8. What is HTML?

HTML stands for Hypertext Markup Language. It's the standard language used to create web pages and is a fundamental technology of the World Wide Web. HTML is a way to structure content on the internet. It uses tags (special code words in angle brackets) to tell web browsers how to display text, images, and other content on a webpage. For example:

- Tags like <h1> and <p> define headings and paragraphs
- <img> is used to insert images
- <a> creates links to other pages or websites

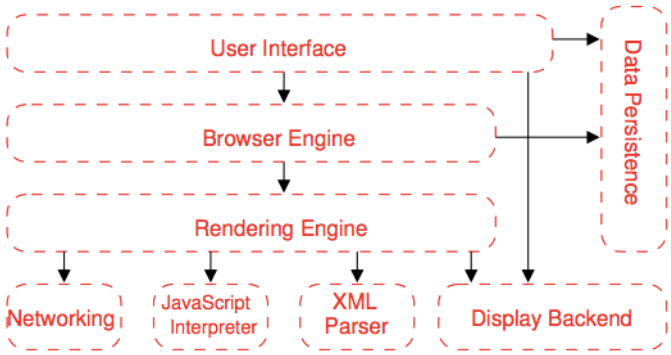
HTML provides the basic structure of web pages, which can then be styled with CSS and made interactive with JavaScript. It's relatively easy to learn and is essential for anyone interested in web development or design.

9. What is browser Engine?

A browser engine, also known as a rendering engine or layout engine, is a core component of web browsers that processes HTML, CSS, and other web content to display web pages on a user's screen. It interprets and renders web content into a visual or audio format.

1. Main functions:
    - Parsing HTML and CSS
    - Layout calculation
    - Content rendering
    - JavaScript execution (often via a separate JavaScript engine)

Browser engines determine how accurately and quickly web pages are displayed, affecting user experience and web developers' ability to create consistent experiences across different browsers. Browser engines are crucial for interpreting web standards and ensuring that websites function as intended across different browsers and devices.



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

A rendering engine is a core component of a web browser responsible for displaying web content on a user's screen. It interprets HTML, CSS, and other web technologies to render the visual or audio output of a web page. Available rendering engines:

1. Blink:
    - Used by: Google Chrome, Microsoft Edge, Opera
    - Developed by: Google (forked from WebKit)
2. Gecko:
    - Used by: Mozilla Firefox
    - Developed by: Mozilla Foundation
3. WebKit:
    - Used by: Apple Safari
    - Developed by: Apple (originally forked from KHTML)
4. Trident:
    - Used by: Internet Explorer (legacy)
    - Developed by: Microsoft

11. What is JavaScript engine? Share the available JS engine? Purpose of JDS engine?

A JavaScript engine is a program or interpreter that executes JavaScript code. It's a crucial component of web browsers and other JavaScript runtime environments. Purpose of JavaScript engines:

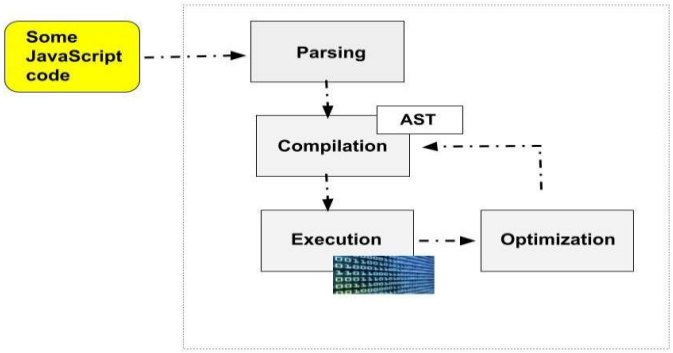
1. Parse JavaScript code
2. Compile code to machine language or bytecode
3. Optimize code execution
4. Manage memory allocation and garbage collection
5. Provide runtime environment for JavaScript execution

Available JavaScript engines:

1. V8:
    - Used by: Google Chrome, Node.js, Microsoft Edge
    - Developed by: Google
2. SpiderMonkey:
    - Used by: Mozilla Firefox
    - Developed by: Mozilla Foundation
3. JavaScriptCore (Nitro):
    - Used by: Apple Safari
    - Developed by: Apple
4. Chakra:
    - Used by: Internet Explorer, older versions of Microsoft Edge
    - Developed by: Microsoft
5. Hermes:
    - Used by: React Native for Android
    - Developed by: Facebook

The purpose of JavaScript engines is ;

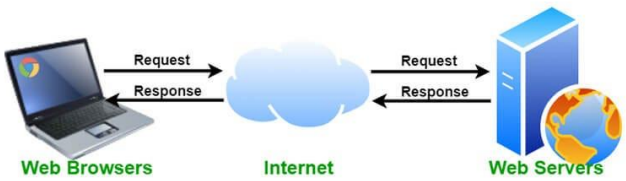
1. Performance optimization: Just-in-time (JIT) compilation and other techniques to speed up code execution.
2. Memory management: Efficient allocation and deallocation of memory resources.
3. Security: Sandboxing JavaScript execution to prevent malicious code from accessing system resources.
4. Compatibility: Implementing JavaScript language specifications to ensure consistent behaviour across platforms.
5. Feature support: Implementing new JavaScript features as they are added to the language specification.



12. How Website works?

A website works through a series of interactions between a user's device, a web browser, and web servers. When a user enters a URL or clicks a link, the browser sends a request to the appropriate web server.

- The server processes this request and sends back the necessary files, typically HTML, CSS, and JavaScript. The browser's rendering engine then interprets the HTML to create the Document Object Model (DOM), applies CSS styles, and executes JavaScript code.
- This process builds the visual structure, appearance, and interactive elements of the web page. The JavaScript engine handles any dynamic content or user interactions. Throughout this process, additional resources like images or videos may be fetched from the server.
- The final result is a fully rendered and functional web page displayed on the user's screen, ready for interaction. This entire process happens quickly, often in fractions of a second, creating a seamless browsing experience for the user.



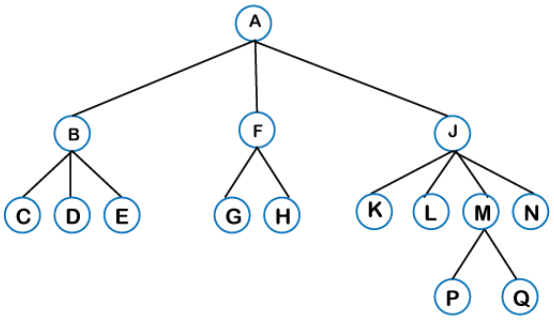
13. What is Data structure?

Data structures are organized methods for storing and managing data in computer programs and systems. They provide efficient frameworks for data access, manipulation, and retrieval. Common types include arrays, linked lists, stacks, queues, trees, graphs, and hash tables, each suited for specific tasks and scenarios. Properly selected data structures enable developers to optimize algorithms, solve complex computational problems, and create more efficient software. They are fundamental to computer science and form the backbone of many applications, from simple list management to complex database systems and network algorithms. Understanding data structures is crucial for writing efficient code and designing scalable solutions in software development.

14. Explain tree data structure?

A tree is a hierarchical data structure composed of nodes connected by edges. It starts with a root node at the top and branches out to child nodes below, forming a structure resembling an inverted tree. Each node can have zero or more child nodes, and every node has exactly one parent node. Nodes without children are called leaf nodes. Trees are widely used in computer science for organizing and representing hierarchical data, such as file systems, organization charts, or HTML DOM. They offer efficient searching, insertion, and deletion operations,

making them valuable for various applications, including binary search trees, decision trees in machine learning, and parsing in compilers.



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

A User Agent is a string of information that web browsers and other applications send to websites and servers to identify themselves. It typically includes details about the browser, operating system, and sometimes the device being used.

list of common components found in User Agent strings and their purposes:

1. Browser name and version: Identifies the specific web browser being used.
2. Operating system: Indicates the OS running on the user's device.
3. Device type: Can specify if it's a mobile device, tablet, or desktop computer.
4. Rendering engine: Identifies the browser's core software for rendering web pages.
5. Browser language: Shows the language settings of the browser.

16. What is HyperTest?

HyperTest typically refers to a high-performance test automation platform designed for executing large-scale test suites quickly and efficiently. It's often associated with software testing and quality assurance processes. Key features of HyperTest platforms often include:

1. Parallel execution: Ability to run multiple tests simultaneously across different environments.
2. Cross-browser testing: Support for running tests on various web browsers.
3. Cloud-based infrastructure: Allowing tests to be run on remote servers for scalability.
4. Integration capabilities: Often can integrate with popular testing frameworks .

17. What is HTML Tags?

HTML tags are the building blocks of HTML (Hypertext Markup Language) used to structure and format content on web pages. They are enclosed in angle brackets and typically come in pairs: an opening tag and a closing tag. HTML tags allow browsers to interpret and display web content correctly, ensuring proper structure and formatting of web pages

-Tags define the structure and meaning of content within a web page.

-Structure: Most tags have an opening <tag> and closing </tag> format.

- <p> for paragraphs
- <h1> to <h6> for headings
- <a> for links
- <img> for images
- for content division

-Self-closing tags: Some tags don't require a closing tag, such as
 for line breaks or <img>.

18. What is HTML Attributes?

HTML attributes provide additional information about HTML elements and modify their behaviour. They are always specified in the opening tag of an element. Purpose: Attributes define properties or characteristics of HTML elements. HTML attributes allow developers to customize elements, link to resources, and enhance functionality and accessibility of web pages.

-Structure: Attributes are written as name-value pairs: name="value"

- class: Specifies one or more class names for styling
- id: Provides a unique identifier for an element
- src: Specifies the source URL for elements like images
- href: Defines the URL for links
- style: Applies inline CSS styles

- -Boolean attributes: Some attributes don't require a value, like "disabled" or "checked"
- Global attributes: Can be used on any HTML element, such as "title" or "lang"
- Event attributes: Used for JavaScript events, like "onclick" or "onload"

19. What is HTML Elements?

HTML elements are the components that make up an HTML document. They consist of a start tag, content, and an end tag (in most cases). HTML elements provide the structure for web content, allowing browsers to render pages correctly and search engines to understand content.

1. Structure: <tagname>Content goes here</tagname>

2. Purpose: Define the structure and content of web pages

3. Types: - Block-level elements (e.g., , <p>, <h1>) - Void elements (self-closing, e.g.,
, <img>)

- Inline elements (e.g., , <a>,)

4. Nesting: Elements can contain other elements

5. Common elements:

- <html>: Root element - <head>: Contains metadata

- <body>: Contains visible content
- <h1> to <h6>: Headings
- <p>: Paragraphs
- <a>: Links
- <img>: Images

6. Semantic elements: Describe the meaning of content (e.g., <article>, <nav>, <header>)

20. How do convert elements to tree?

Converting HTML elements to a tree structure is a common practice in web development, especially for parsing and manipulating the Document Object Model /

→ Start with the root element (usually <html>)
→ For each element:
→ Create a node in the tree
→ Add child nodes for its nested elements
→ Add attributes as properties of the node
→ Text content becomes leaf nodes

21. What is DOCTYPE?

DOCTYPE, which stands for Document Type Declaration, is a statement at the beginning of an HTML document that informs the web browser about the version of HTML being used. The DOCTYPE is crucial for ensuring consistent rendering across different browsers and maintaining web standards compliance.

- Tells the browser how to interpret the document

22. What are the ways we can save html files?

HTML files can be saved in various ways depending on your tools and needs. The simplest method is using a basic text editor like Notepad or TextEdit, saving the file with a .html extension. For more advanced features, code editors like VS Code or Sublime Text offer syntax highlighting and other helpful tools. Web browsers allow saving web pages directly, usually giving options for saving the complete page or just the HTML. it's important to use the correct file extension (.html or .htm) and choose UTF-8 encoding for best compatibility across different systems and browsers.

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

The charset(character set), is a element in HTML documents that specifies the character encoding used. It's used for ensuring proper text rendering and data interpretation across different systems and browsers. By declaring the charset, we tell browsers how to correctly interpret and display text characters, which is particularly important for multilingual support and preventing garbled text. The charset also maintains data integrity, ensures consistency in text representation, aids in search engine optimization, and facilitates accurate data transmission between servers and clients. In modern HTML, the charset is typically declared in the <head> section of the document using a meta tag. Properly specifying the charset is

fundamental to creating web content that displays correctly for all users, regardless of their language or system configuration.

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

Metadata in HTML is information about the web page that is not directly displayed to users but provides important details about the document to browsers, search engines, and other web services. It is typically included within the <head> section of an HTML document using various tags, most commonly the <meta> tag.

The purpose of metadata is

- it helps define the character encoding,
- provides a concise description of the page's content, specifies the author,
- controls how the page should be indexed or cached
- can even define how the page should be displayed when shared on social media platforms.

plays a important role in search engine optimization by providing search engines with structured information about the page's content and purpose. By effectively using metadata, web developers can enhance the functionality, accessibility, and discoverability of their web pages, ensuring they are properly interpreted by both machines and humans across various platforms and devices.

25. Explain Web Application Architecture?

Web Application Architecture is the structure and interaction of components in a web application. It has both client-side (front-end) and server-side (back-end) elements, along with databases, middleware, web servers, and application servers.

- The client-side handles user interface and browser-based logic, while the server-side manages application logic, data processing, and database interactions. Databases store application data, and middleware facilitates communication between different parts.
- Web servers handle HTTP requests and serve static files, while application servers execute server-side code and business logic. APIs enable component communication, and caching layers improve performance. Load balancers distribute traffic for scalability.

1. Client-side (Front-end):
    - User interface (HTML, CSS, JavaScript)
2. Server-side (Back-end):
    - Application logic
3. Database:
    - Stores application data
4. Middleware:
    - Facilitates communication between different parts of the application
5. Web server:
    - Handles HTTP requests and responses
6. Application server:
    - Executes server-side code

7. APIs:
    - Enable communication between different components or external services
8. Caching layer:
    - Improves performance by storing frequently accessed data
9. Load balancer:
    - Distributes incoming traffic across multiple servers for scalability