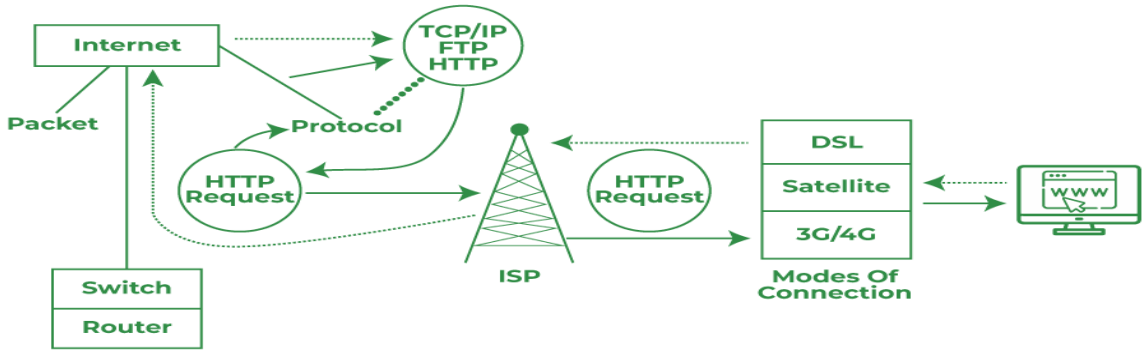
Assignment 1:

1. How internet works?

From opening a web browser to visiting a website, it all happens with specific methods. There are 5 easy steps.



1. Firstly, you'll be required to connect your system or PC with any router or modem to establish a connection. This connection is the base of the internet connection.

2. When you open the browser and start typing something like "www.google.com", your system will push a query command to your ISP (Internet Service Provider) that is connected with other servers that store and process data.

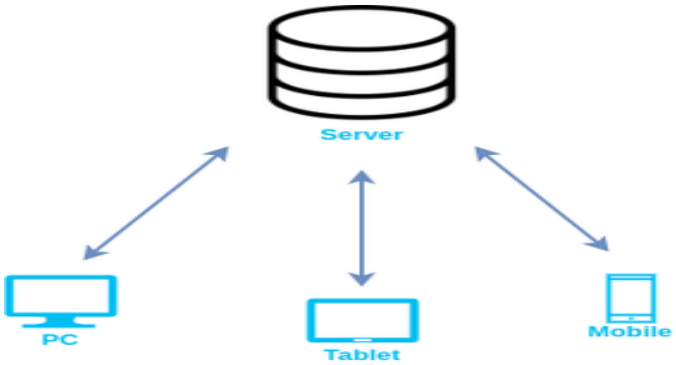
3. Now, the web browser will start indexing the URL that you've entered and will fetch the details in numeric format (in their language to identify the address (unique) that you're trying to reach.

4. Next, now your browser will start sending the HTTP request where you're trying to reach and send a copy of the website on the user's system. Note: The server will send data in the form of small packets (from the website to the browser)

5. Once all the data (of small packets) is received at the user's end (PC/Laptop), the browser will start arranging all those small packets and later will form a collective file (here, the browser will gather all the small packets and rearrange them just like a puzzle) and then you'll be able to see the contents of that website

2. How browser works?

A web browser helps us find information anywhere on the internet. It is installed on the client computer and requests information from the web server such a type of working model is called a client-server model. The browser receives information through HTTP protocol. In which transmission of data is defined. When the browser received data from the server, it is rendered in HTML to user-readable form and, information is displayed on the device screen.



3. What is Server?

A Server is a program or a device that provides functionality for called clients which are other programs or devices. This architecture is called the client-server model.

A single overall computation is distributed across multiple processes or devices. Servers can provide various functionalities called services. These services include sharing data or resources among multiple clients or performing computations for a client. Multiple clients can be served by a single server, and a single client can use multiple servers.

4. what are the types of server available?

The types of server available are:

1. Application Server

2. Catalog Server

3. Communication Server

4. Computing Server

5. Database Server

6. Fax Server

7. File Server

8. Game Server

9. Mail Server

10. Print Server

1 1. Proxy Server

12. Web Server

5. What is SEO? Importance of SEO?

SEO is the practice of increasing the quantity and quality of traffic to your website through organic search engine results. A higher rank when someone searches a term in your industry increases your brand's visibility online. The increase in visibility will drive more organic traffic to your site, and this, in turn, gives you more opportunities to convert qualified prospects into customers. When done correctly, SEO can help your brand stand above others as a trustworthy company and further improve the user's experience with your brand and website.

Here are the key reasons why SEO is essential:

1. Increased Visibility and Traffic

2. Credibility and Trust

3. Better User Experience

4. Cost-Effectiveness

5. Competitive Advantage

6. Insights into Customer Behavior

7. Local SEO for Targeted Traffic

8. Long-Term Strategy

9. Improved ROI

10. Brand Awareness

6. What is Accessibility?

Accessibility is the practice of making your websites usable by as many people as possible.

7. What is Markup Language?

Markup languages are computer languages that are used to structure, format, or define relationships between different parts of text documents with the help of symbols or tags inserted in the document. These languages are more readable than usual programming languages with strict syntax.

8. What is HTML?

Hypertext Markup Language (HTML) is a markup language used to create and link webpages. It defines the basic structure of a web page and contains meta-data about the page and a series of elements to be displayed on the web page. It uses predefined tags such as <table>, <form>, etc. to render different elements on the webpage. Each element requires a starting and ending tag with content inside it.

A marked-up document written in HTML is displayed by a web browser that interprets different tags and accordingly formats and structures the content of the document before displaying it. It can be written in a plain text editor and can be associated with styling sheets such as CSS (Cascading Style Sheets) and scripting languages such as JavaScript.

Example: Implementation of the basic structure of HTML code.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Example Code</title>
</head>
<body>
  <h1>Aneena Thomas</h1>
  <p>From Department of CS</p>
</body>
</html>
```

9. What is browser engine?

Browser engine as the heart of the web browser. It is the essential software that acts as a bridge between the web page's code (HTML, CSS, JavaScript) and the visual experience you see on your screen.

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

Rendering engine is responsible for displaying the requested web resources by parsing the contents. By default, it can parse html, xml, and images. It uses different plugins and/or extensions to display other type of data such as flash, PDF, etc.

There are different rendering engines such as Gecko, WebKit, and Trident. Most widely used rendering engine is WebKit or its variant version. Gecko and WebKit are open source rendering engines while Trident is not. Firefox uses Gecko, Safari uses WebKit, Internet Explorer uses Trident, Chrome and Opera uses Blink, which is a variant of WebKit.

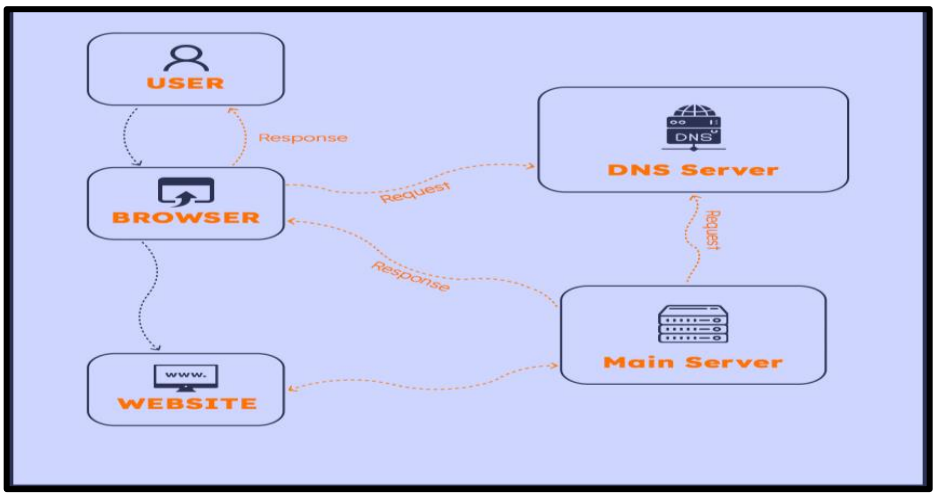
11. What is JavaScript Engine? share the available JS engine? Purpose of JS Engine?

A JavaScript engine is a computer program that executes JavaScript code and converts it into computer understandable language.

Browser	Name of Javascript Engine
Google Chrome	V8
Edge (Internet Explorer)	Chakra
Mozilla Firefox	Spider Monkey
Safari	Javascript Core Webkit

The JavaScript engine is simply a computer program that interprets JavaScript code. The engine is responsible for executing the code. Every major browser has a JavaScript engine that executes JavaScript code. The most popular one is the Google Chrome V8 engine.

12. How website works?

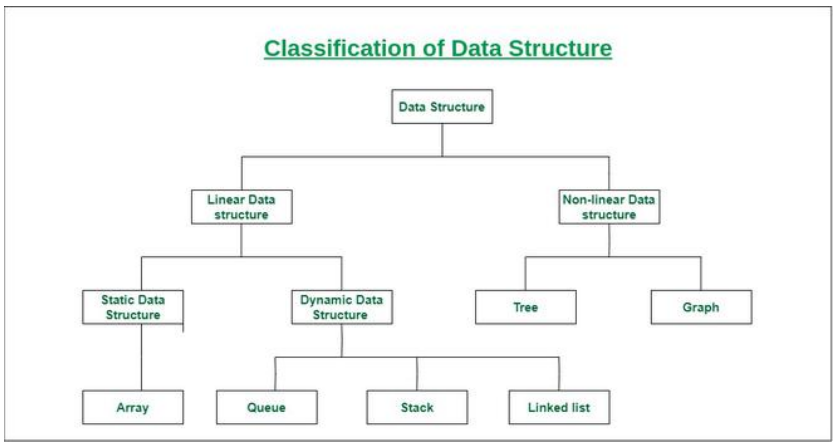


The process is quite simple. Firstly, you enter a domain name or a website address in the search bar. Then the browser passes the request to DNS Server. DNS server acts as an address directory. It converts the human-readable address to a machine-readable address i.e. the IP address of the Website address to a machine-readable address i.e. the IP address of the website.

Then it passes the request to the main server or the server where your site is stored. Then the server provides the response to the browser and now you are able to access the website. The whole process takes hardly 1 or 2 seconds.

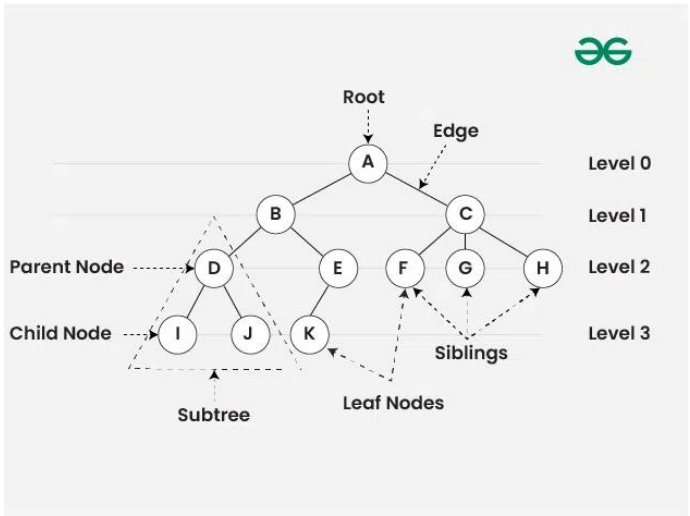
13. What is Data Structure?

A data structure is a storage that is used to store and organize data. It is a way of arranging data on a computer so that it can be accessed and updated efficiently.A data structure is not only used for organizing the data. It is also used for processing, retrieving, and storing data.The most commonly used data structures, including arrays, linked lists, stacks, queues, trees, and graphs.



14. Explain Tree Data Structure?

Tree data structure is a specialized data structure to store data in hierarchical manner. It is used to organize and store data in the computer to be used more effectively. It consists of a central node, structural nodes, and sub-nodes, which are connected via edges. We can also say that tree data structure has roots, branches, and leaves connected.



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

he User-Agent (UA) string is contained in the HTTP headers and is intended to identify devices requesting online content. The User-Agent tells the server what the visiting device is (among many other things) and this information can be used to determine what content to return. Of course this requires using a device detection solution which translates UAs into understandable software and hardware information.

On the Web, a user agent is a software agent responsible for retrieving and facilitating end-user interaction with Web content.

List of user agent are:

1. Android User Agents

2. iPhone User Agents

3. MS Windows User Agents

4. Tablet User Agents

5. Desktop User Agents

6. Set Top Box User Agents

7. Games Console User Agents

8. Bots and Crawlers User Agents

9. E-Readers User Agents

16. What is Hypertest?

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. "Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, , , <img>, <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <search>, <output>, <progress>, <video>, <ul>, <ol>, <li> and many others.

17. What is HTML Tags?

HTML tags are like keywords which defines that how web browser will format and display the content. With the help of tags, a web browser can distinguish between an HTML content and a simple content. HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

When a web browser reads an HTML document, browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents and render their properties. Each HTML tags have different properties.

Syntax

<tag> content </tag>

Ex:

```
<!DOCTYPE html>
<html>
 <head>
  <title>First HTML Code</title>
 </head>
 <body>
  <h2>Welcome To GFG</h2>
  <p>Hello Geeks</p>
 </body>
```

</html>

18. What is HTML Attributes?

HTML attributes are special words which provide additional information about the elements or attributes are the modifier of the HTML element.Each element or tag can have attributes, which defines the behaviour of that element.Attributes should always be applied with start tag.The Attribute should always be applied with its name and value pair.The Attributes name and values are case sensitive, and it is recommended by W3C that it should be written in Lowercase only.You can add multiple attributes in one HTML element, but need to give space between two attributes.

Syntax:

<element attribute_name="value">content</element>

Example:

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
    <h1> This is Style attribute</h1>
   <p style="height: 50px; color: blue">It will add style property in element</p>
    <p style="color: red">It will change the color of content</p>
</body>
</html>
```

19. What is HTML Elements?

An HTML Element is a collection of start and end tags with the content inserted between them. HTML elements are building blocks of web pages, representing different types of content such as headings, paragraphs, links, and images.

Ex:

 Hello world!!!

Here Hello world!!! Is the HTML element

20. How do convert elements to tree?

To convert elements into a tree structure, you'll need to understand both the elements and the tree data structure. Here's a step-by-step guide to achieve this:

Understand the Elements: The elements could be anything that you want to organize into a hierarchical structure, such as a list of items, a flat structure with parent-child relationships, etc.

Define the Tree Structure: Decide on the type of tree you need. For example, you might need a binary tree, a multi-way tree, or a generic tree.

Create Nodes: Each element will be converted into a node. A node typically contains data and pointers (or references) to its children.

Build the Tree:Identify the Root: Determine which element will be the root of the tree.

Add Children: Recursively or iteratively add children to each node according to the hierarchical relationships among the elements.

21. What is DOCTYPE?

HTML <!DOCTYPE> tag is used to inform the browser about the version of HTML used in the document. It is called as the document type declaration (DTD). <!DOCTYPE > is not a tag/element, it just an instruction to the browser about the document type. It is a null element which does not contain the closing tag, and must not include any content within it. The <!DOCTYPE> declaration refers Document Type Declaration (DTD) in HTML 4.01; because HTML 4.01 was based on SGML. But HTML 5 is not SGML based language.

Syntax:

<!DOCTYPE html>

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

HTML file can be done in several ways, depending on your needs and the tools you have available. Here are some common methods:

1. Manually Save from a Browser

Using the "Save As" Option:

Open the web page in your browser.

Right-click on the page and select "Save As..." or use the browser menu to select "File" -> "Save Page As..."

Choose the desired location and format (typically "Web Page, Complete" to save the HTML file along with all assets like images, or "Web Page, HTML only" for just the HTML).

2. Using a Text Editor or IDE

Creating a New HTML File:

Open a text editor or Integrated Development Environment (IDE) (e.g., Notepad, VSCode, Sublime Text).

Write or paste your HTML code.

Save the file with a .html extension by selecting "File" -> "Save As..." and choosing "All Files" in the save dialog, then typing yourfile.html as the filename.

3. Programmatically Saving an HTML File

Using JavaScript,python and Node.js

4. Using Command Line Tools

curl or wget (for downloading HTML content from a URL)

curl https://example.com -o my_page.html

wget -O my_page.html https://example.com

5. Using a Web Development Framework

If you're working within a web development framework (like Django, Flask, Rails, etc.), you might save HTML content as part of a template rendering process.

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

HTML Charset is also called HTML Character Sets or HTML Encoding. It is used to display an HTML page properly and correctly because for displaying anything correctly, a web browser must know which character set (character encoding) to use.

Ex:

<meta charset="UTF-8">

24. What is meta data? what is the purpose of it?

Metadata is data that provides information about other data. In the context of web pages and digital content, metadata helps describe the content, structure, context, and properties of the data, making it easier to find, use, and manage. Here are some purposes of metadata:

Discovery: Helps in finding relevant information quickly and efficiently. Search engines use metadata to index web pages, making them discoverable through search queries.

Organizing and Classifying: Aids in organizing and classifying content. For example, a library catalog uses metadata to organize books by author, title, subject, etc.

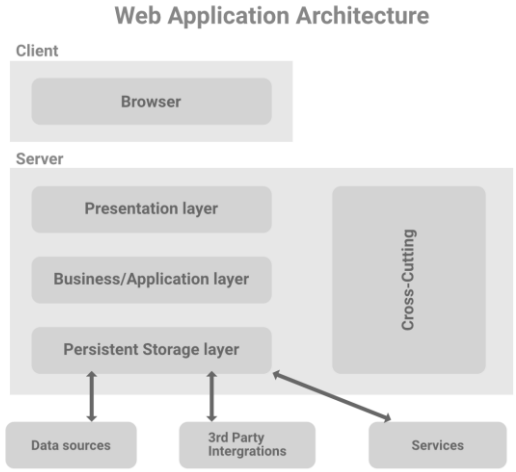
Resource Management: Supports the management and administration of resources. Metadata can indicate who created a file, who has access to it, and when it was last modified.

Interoperability: Ensures that information can be easily shared and understood across different systems. Standardized metadata formats facilitate the exchange of information between diverse systems.

Preservation: Assists in the preservation of digital content by providing information about the format, creation, and context of the content, which is essential for long-term storage and access.

Content Understanding: Enhances understanding of the content by providing context. For instance, metadata in an academic paper can include the abstract, keywords, and citation information.

25. Explain Web Application Architecture?



Web application architecture is the framework that defines the structure, components, and communication methods of a web application. It outlines how the various parts of a web application interact with each other and with external systems to deliver a cohesive experience to users. Here's an overview of the key components and common patterns in web application architecture:

Key Components of Web Application Architecture

1.Client-Side (Front-End):

Browser: The user's interface to interact with the web application.

HTML/CSS: Markup and styling of web pages.

JavaScript: Client-side scripting for dynamic behavior.

Frameworks/Libraries: Tools like React, Angular, or Vue.js for building rich user interfaces.

2.Server-Side (Back-End):

Web Server: Serves web pages to clients. Common web servers include Apache, Nginx, and IIS.

Application Server: Handles the business logic and application operations. Common languages and frameworks include Node.js, Django (Python), Spring (Java), and Ruby on Rails.

Database Server: Stores and manages data. Common databases include MySQL, PostgreSQL, MongoDB, and SQL Server.

3.APIs and Web Services:

RESTful APIs: Used for communication between client and server or between different services.

GraphQL: An alternative to REST for more efficient querying of data.

SOAP: A protocol for exchanging structured information in web services.

4.Security:

Authentication and Authorization: Ensuring users are who they say they are and have the right permissions (e.g., OAuth, JWT, SAML).

HTTPS: Secure communication over the internet.

Firewalls and Gateways: Protecting against unauthorized access.

5.Scalability and Load Balancing:

Load Balancers: Distribute incoming traffic across multiple servers.

Caching: Improves performance by storing frequently accessed data in memory (e.g., Redis, Memcached).

Microservices: An architectural style that structures an application as a collection of loosely coupled services.

Communication Flow:

1.Client Request: The user interacts with the client-side application (e.g., a web browser).

2.Server Processing: The client-side application sends a request to the server-side application through an API.

3.Data Retrieval: The server-side application processes the request, interacts with the database to retrieve or store data, and executes business logic.

4.Response: The server-side application sends the processed data back to the client-side application.

5.Client Rendering: The client-side application renders the data for the user