Web Technology

Basic Web Technology Terminologies:

1. IP Address (Internet Protocol Address):
Definition: An IP address is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication. It serves two main functions: host or network interface identification and location addressing.

2. Public IP Address:
Definition: A public IP address is an address assigned to a device on the internet. It is globally unique and allows devices to communicate across the internet. Websites, servers, and routers typically have public IP addresses.

3. Private IP Address:
Definition: A private IP address is used within a private network and is not directly accessible from the internet. Devices within a local network, such as computers, printers, or smartphones, use private IP addresses for communication.

4. Domain Name:
Definition: A domain name is a human-readable address that corresponds to the IP address of a server on the internet. It provides a way to access websites without needing to remember numerical IP addresses.

5. URL (Uniform Resource Locator):
Definition: A URL is a specific type of Uniform Resource Identifier (URI) that provides a means to locate and retrieve resources on the internet. It typically includes the protocol (e.g., http or https), domain name, and path.

6. Web Server:
Definition: A web server is software or hardware that stores, processes, and delivers web pages to users. It responds to requests from web browsers by providing the requested web pages.

7. Web Browser:
Definition: A web browser is a software application used to access and view web pages. Examples include Chrome, Firefox, Safari, and Edge.

8. Router:
Definition: A router is a networking device that forwards data packets between computer networks. In a home network, a router often assigns local IP addresses to devices and manages internet connectivity.

9. DNS (Domain Name System):

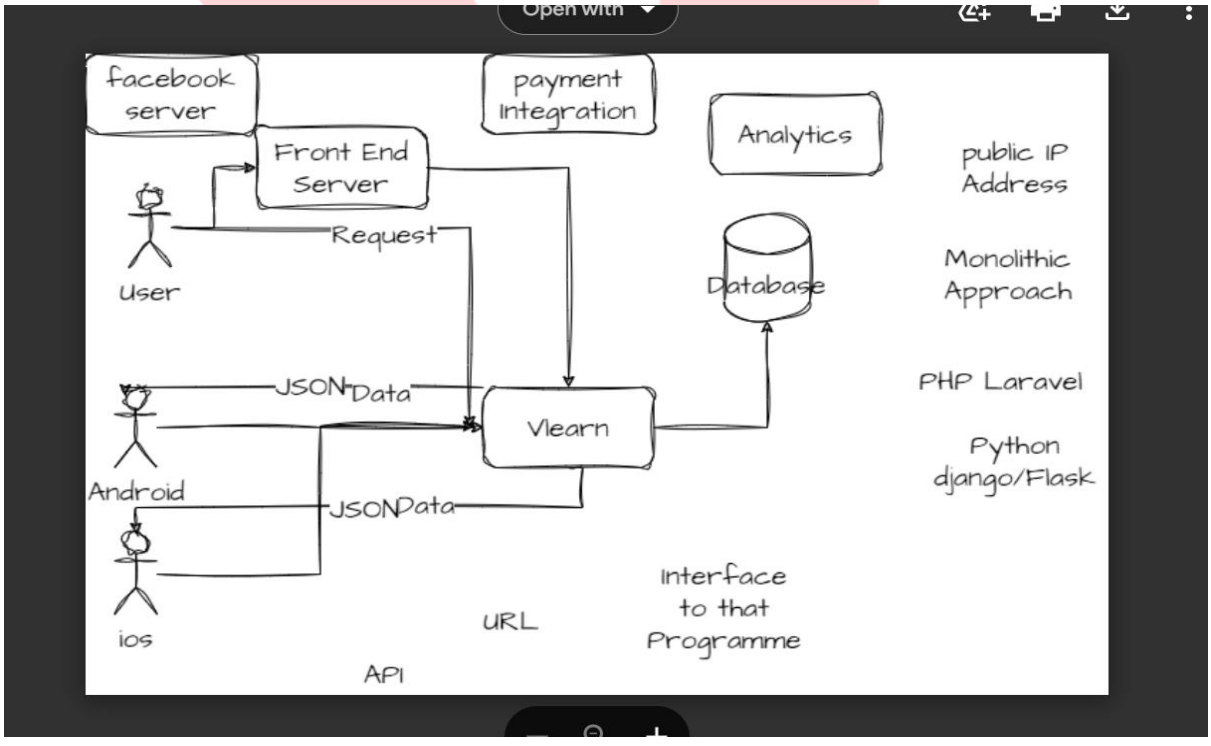
Definition: DNS is a system that translates human-readable domain names (e.g., www.example.com) into IP addresses. It helps users access websites using memorable domain names.

10. HTTP (Hypertext Transfer Protocol) and HTTPS:
Definition: HTTP is a protocol used for transmitting hypertext documents on the World Wide Web. HTTPS (HTTP Secure) is a secure version of HTTP that encrypts data between the web browser and server.

11.Client:
Definition: A client is a user's device or software application that requests and displays web pages. It interacts with servers to access and consume web content.

12.Responsive Design:
Definition: Responsive design is an approach to web design that ensures a webpage's layout adjusts dynamically based on the user's device, providing an optimal viewing experience across various screen sizes.

How Does Web Technology Work?



It shows a simplified overview of how web technology works, focusing on the interaction between a web server and a user. Here's a breakdown of the steps involved:

User Requests a Web Page: The process starts when a user enters a URL (web address) into their web browser. The browser then sends an HTTP request to the web server specified in the URL.

Server Processes the Request: The web server receives the request and locates the requested web page files on its storage system.

Server Sends Response: The server processes the web page files, which may involve interpreting server-side scripting languages like PHP or Python. The processed web page data is then packaged into an HTTP response message.

Response Sent to Browser: The server sends the HTTP response message back to the user's web browser.

Browser Renders the Web Page: The web browser receives the response message and extracts the web page data. It then uses the HTML code in the data to structure the content of the page.

Browser Applies Styling: The browser also uses the CSS code in the data to style the web page, applying fonts, colors, layouts, and other visual elements.

Page Displayed to User: Finally, the web page is rendered on the user's screen, and they can interact with its content, such as clicking links, filling out forms, or watching videos.
The image also mentions some additional details about the web technology stack:

Front End vs Back End: The diagram distinguishes between the front-end and back-end of a web application. The front-end refers to the parts of the website that users see and interact with, such as the HTML, CSS, and JavaScript code. The back-end refers to the server-side code and databases that power the website and handle its logic.

Databases: Web applications often store data in databases, such as MySQL or PostgreSQL. The diagram shows the database connected to the server, indicating that the server can access and manipulate the data stored there.

APIs: APIs (Application Programming Interfaces) allow different software components to communicate with each other. The diagram mentions APIs for Facebook and payment integration, suggesting that the website may use these APIs to provide certain functionalities.

This is a simplified overview of how web technology works, and there are many other details and complexities involved in real-world web applications. However, I hope this explanation gives you a basic understanding of the process from a user requesting a web page to seeing it displayed on their screen.