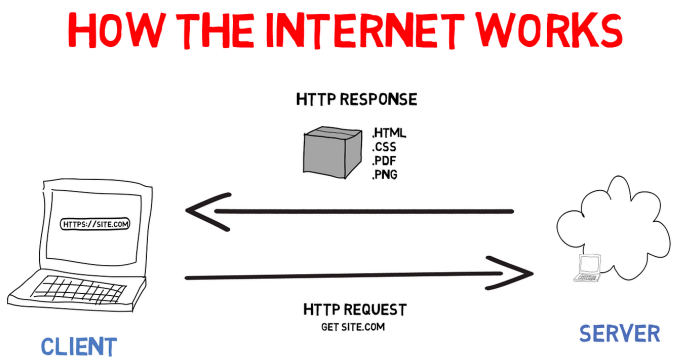
1. Introduction to Full Stack Development

The internet is a vast network that connects computers and devices all around the world. It allows us to access websites, send emails, stream videos, and much more. Here's a simplified explanation of how the internet works:



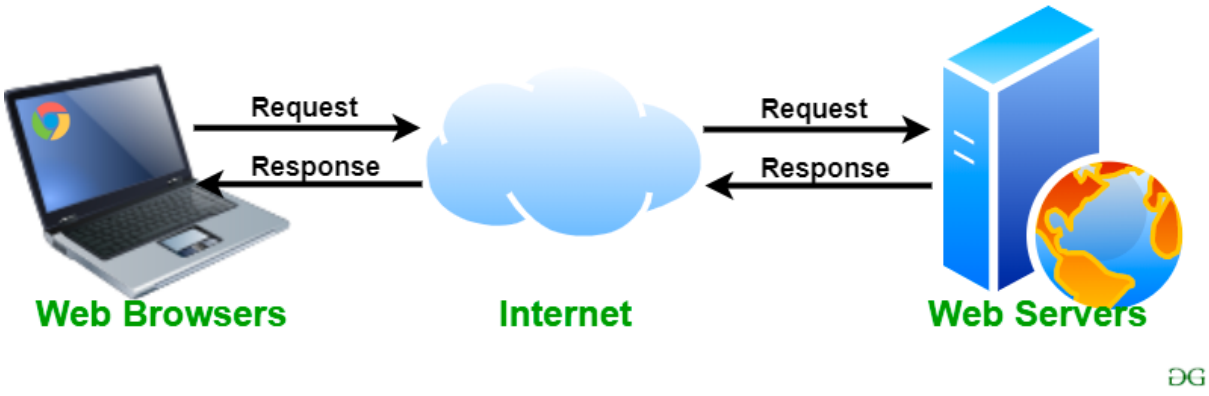
Client-server architecture is a common model for designing and implementing web applications, where multiple computers or devices work together to provide services or resources. In this architecture, there are two main components: the client and the server.

1. **Client**: The client is typically a user's device, such as a computer, smartphone, or tablet. It interacts with the user and sends requests to the server to access data, services, or resources. The client can be a web browser, a mobile app, or any application that communicates with the server.
2. **Server** The server is a powerful computer or system that stores and manages data, services, or resources. It responds to client requests by processing the requests and sending back the requested information. Servers can be dedicated machines or cloud-based services.

The communication between the client and server follows a request-response model. The client sends a request to the server, and the server processes the request and sends a response back to the client.

Example: When you write www.google.com on your browser then you as a client sending request to google server and the server will send you the required resources like html, css, js files in order to show webpage and your browsers will decode these received files and show to you.

Web Server

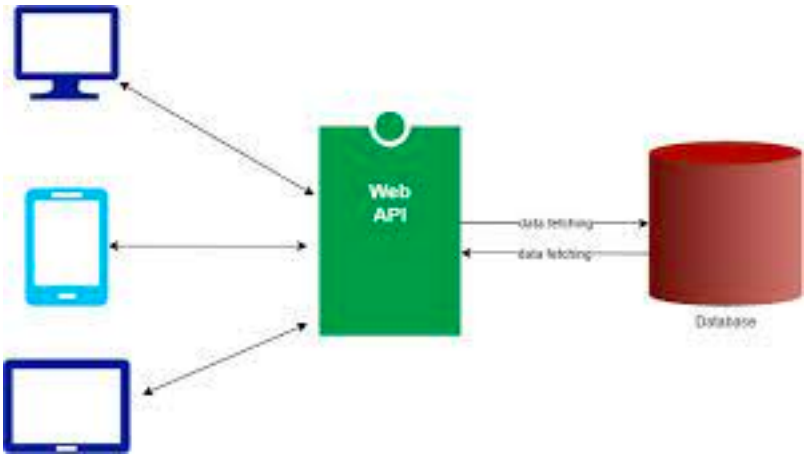


A web server is a software application that serves web pages and other web-related content to clients upon request. It acts as the intermediary between a client (usually a web browser) and the web application or website.

When a user requests a web page by typing a URL or clicking on a link, the web browser sends a request to the web server. The web server processes the request, retrieves the requested resources, and sends them back to the client as a response, which is then rendered and displayed by the web browser.

API (Application programmable interface)

API Act as an intermediate between web server and web client. API provides the secure way to web administrator to allow the client to access their services.



Example: In daily life when we go to hotel the Waiter is an API or we can say intermediate between customers and chef. Waiter will take our request and give it to the chef and once the chef is down making food (response) the waiter will take that response and give it to us. It ensures the authenticity of the communication between client and server.

API will take the request of client and validate it and then pass this request to server and server will do all the processing, once server generates the response the API will take that response and send it back the client.

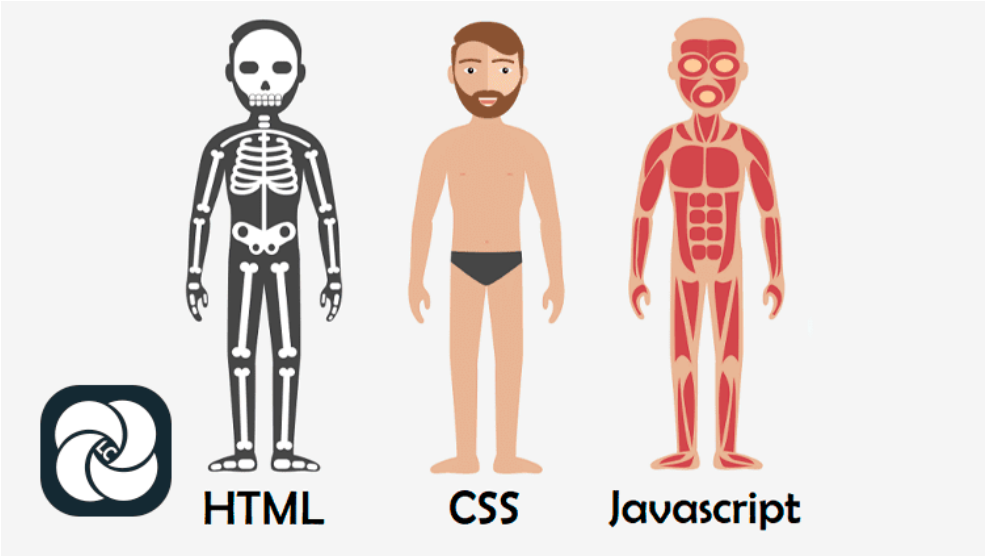
Front-End

Front-end refers to the part of a website or web application that users interact with directly. It involves the design, layout, and presentation of the user interface. Front-end developers use HTML, CSS, and JavaScript to create the visual and interactive elements that users see and interact with in their web browsers. They focus on ensuring a smooth and intuitive user experience.

Back-end:

Back-end refers to the behind-the-scenes technology that powers a website or web application. It encompasses the server-side logic, databases, and all the processes that enable the functionality of the front-end. Back-end developers work with programming languages like Python, Java, Ruby, or PHP, as well as frameworks and tools, to build and maintain the server, handle data storage and retrieval, implement business logic, and manage user authentication and security.

HTML, CSS , JS



HTML (Hypertext Markup Language):

HTML is the standard markup language used for creating the structure and content of web pages. It defines the elements and their organization on a web page using tags. HTML provides a way to format text, insert images, create links, and define the overall layout of a webpage.

CSS (Cascading Style Sheets):

CSS is a stylesheet language used to control the presentation and styling of HTML documents. It describes how elements on a web page should appear, including their colors, fonts, sizes, spacing, and positioning. CSS allows developers to separate the design aspects of a website from its structure, enabling greater flexibility and maintainability.

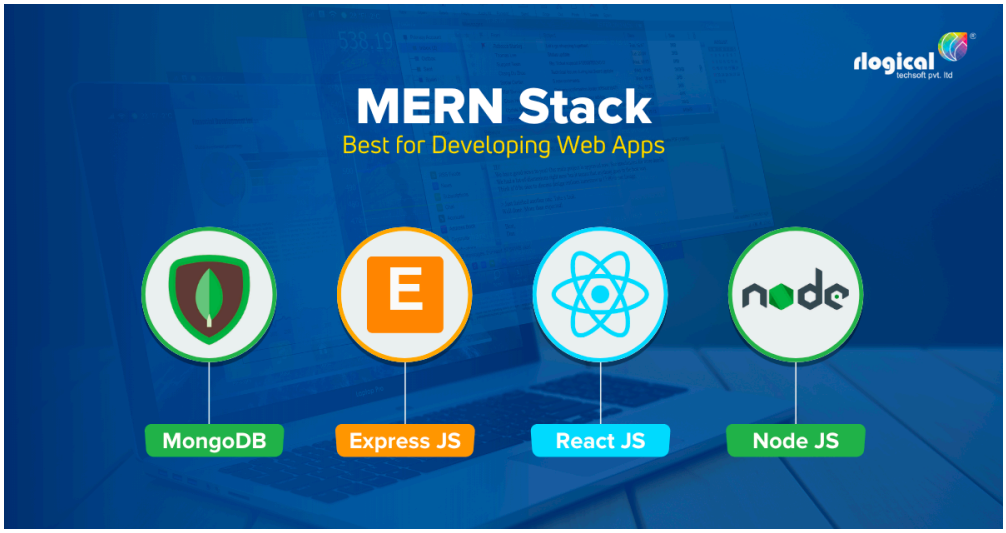
JavaScript:

JavaScript is a programming language that allows web developers to add interactivity and dynamic behavior to web pages. It runs on the client-side, meaning it executes in the user's web browser. JavaScript can be used to handle user interactions, manipulate and modify HTML and CSS dynamically, fetch data from servers, create animations, and perform various operations to enhance the user experience on a website.

Full Stack Development

Full-stack development refers to the ability to work on both the front-end and back-end aspects of a web application. MERN is an acronym that stands for MongoDB, Express.js, React, and Node.js, which are technologies commonly used together for full-stack development.

Here's a breakdown of the MERN stack:



1. **MongoDB**: MongoDB is a popular NoSQL database that stores data in a flexible, JSON-like format called BSON (Binary JSON). It provides scalability, flexibility, and easy integration with JavaScript-based applications.
2. **Express.js**: Express.js is a lightweight web application framework for Node.js. It simplifies the process of building robust and scalable web applications by providing a set of features and tools for handling routes, requests, and responses.
3. **React**: React is a JavaScript library for building user interfaces. It allows developers to create reusable UI components and efficiently manage the application's state. React uses a virtual DOM (Document Object Model) for efficient updates and rendering.
4. **Node.js**: Node.js is a JavaScript runtime environment that allows developers to run JavaScript on the server-side. It provides a non-blocking, event-driven architecture that enables building scalable and high-performance web applications.

Extra Study Resources:

1. History of Js:

2. What is Full Stack Development