**Full stack development**

Full stack development is the process of developing both the frontend and backend of applications. Any application has a frontend (user-facing) component and a backend (database and logic) component.

**Frontend**

Front-end development is the development of visual and interactive elements of a website that users interact with directly. It's a combination of HTML, CSS and JavaScript, where HTML provides the structure, CSS the styling and layout, and JavaScript the dynamic behaviour and interactivity

**Backend**

The backend is the server side of the website. It manages data and ensures everything on the frontend works properly.

**Server:** It is a virtual Machine which is built in cloud

**How we can go to server ?**

Medium+ address

**IP Address:**

* the unique identifying number assigned to every device connected to the internet.
* Attackers can easily attack
* Instead of IP Address we use texts we can purchase text it is also called as Domain or DNS Server

Eg: godaddy.com

**Application:** set of programming Languages.

**Types of Applications:**

**Standalone Application:**

a type of software program that is designed to run on a single computer or local machine of the user, without the need for a server or internet connection.

**web application:**

A web application is software that runs in your web browser

**Mobile Applications:**

Applications designed for mobile devices like phones and tablets

**Desktop Applications:**

Software installed and run on a desktop

**Cloud:**

a term used to describe a global network of servers, each with a unique function.

Eg: AWS, google cloud.

**Libraries:**

Collecting the pre-defined code which is reuse

**Framework:**

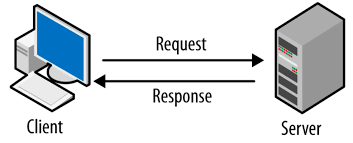
Collection of predefined Libraries is known as Framework.

Eg: React, Dijango & Spring boot.

**API**

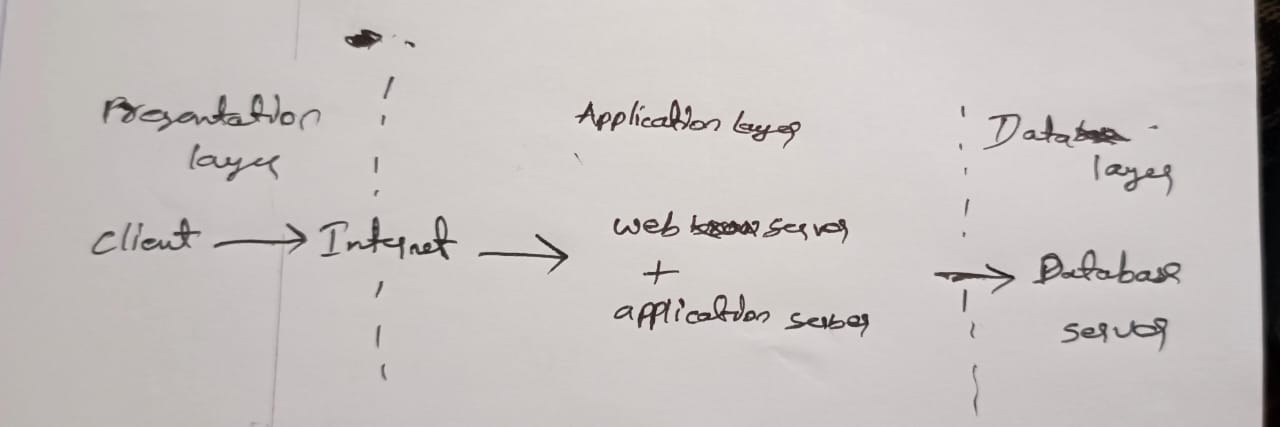
API stands for Application Programming Interface

The working of an API can be clearly explained with a few simple steps. Think of a client-server architecture where the client sends the request via a medium to the server and receives the response through the same medium.



**client-server model:**

The client-server model, or client-server architecture, is a distributed application framework dividing tasks between servers and clients



**Presentation layer:** handles the user interface (UI) part of the web application.

**Application layer:** also known as the logic tier or middle tier, is the heart of the application. It processes user input, manipulates data, and executes business rules.

**Database Layer:** where the information that is processed by the application is stored and managed. This can be a relational database management system such as PostgreSQL, MySQL