

Let's discuss each one of them one by one

① Desktop application

- A type of software program that is installed and run directly on a personal computer rather than through a web browser.
- Needs to be installed on the OS (windows, macOS, -) and often built for specific OS. (windows app may not work on macOS)
- Faster performance because they run locally.
- It can work without requiring Internet access.
- It can use and access various system hardware like printer, camera, GPU, etc.
- Examples: VLC media player, microsoft word, visual studio, Adobe photo shop, etc.

② web applications

- A software program that runs on a web server and is accessed through web browser over the internet.
- User only need a browser, no setup required.
- Accessible from anywhere with internet.

- user just need to open a browser and enter a web address (URL) to access it.
- usually built using HTML, CSS and javascript for the frontend (what user see) and may use backend technologies like Node.js, Python, Java to handle logic, database and other tasks.
- Some modern web apps like Progressive web apps can also work offline.
- Examples: Amazon.com, PW.live, etc.

③ mobile application

- A software program designed specifically to run on mobile devices. One can download and install mobile application from an app store like Google play store or Apple App store.
- Some apps require constant internet access (youtube, instagram) but some can work offline (calculator, offline maps, etc).
- mobile apps are optimized for hardware and OS (iOS, Android) and they can access

device features like camera, microphone, GPS, accelerometer, etc.

→ Helpful in providing fast, responsive and user friendly interaction.

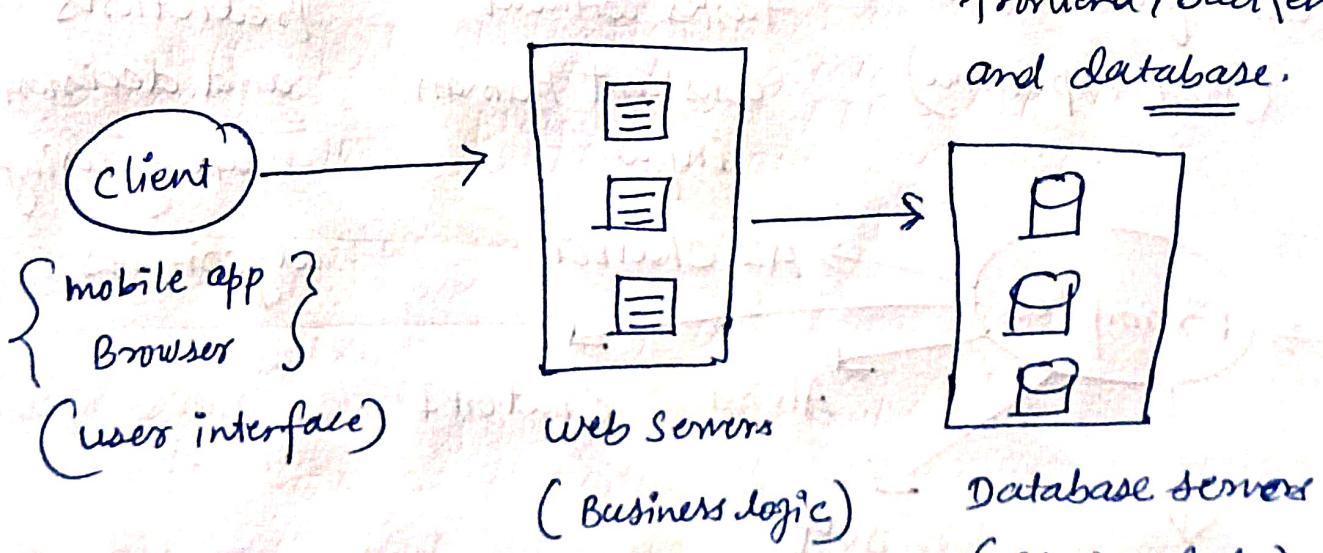
4. Distributed applications

→ A software system that runs on multiple separate computers connected by a network or internet.

One can scale it by adding more computers

one can distribute the incoming load on multiple computers

can be divided into various layer like frontend, backend and database.



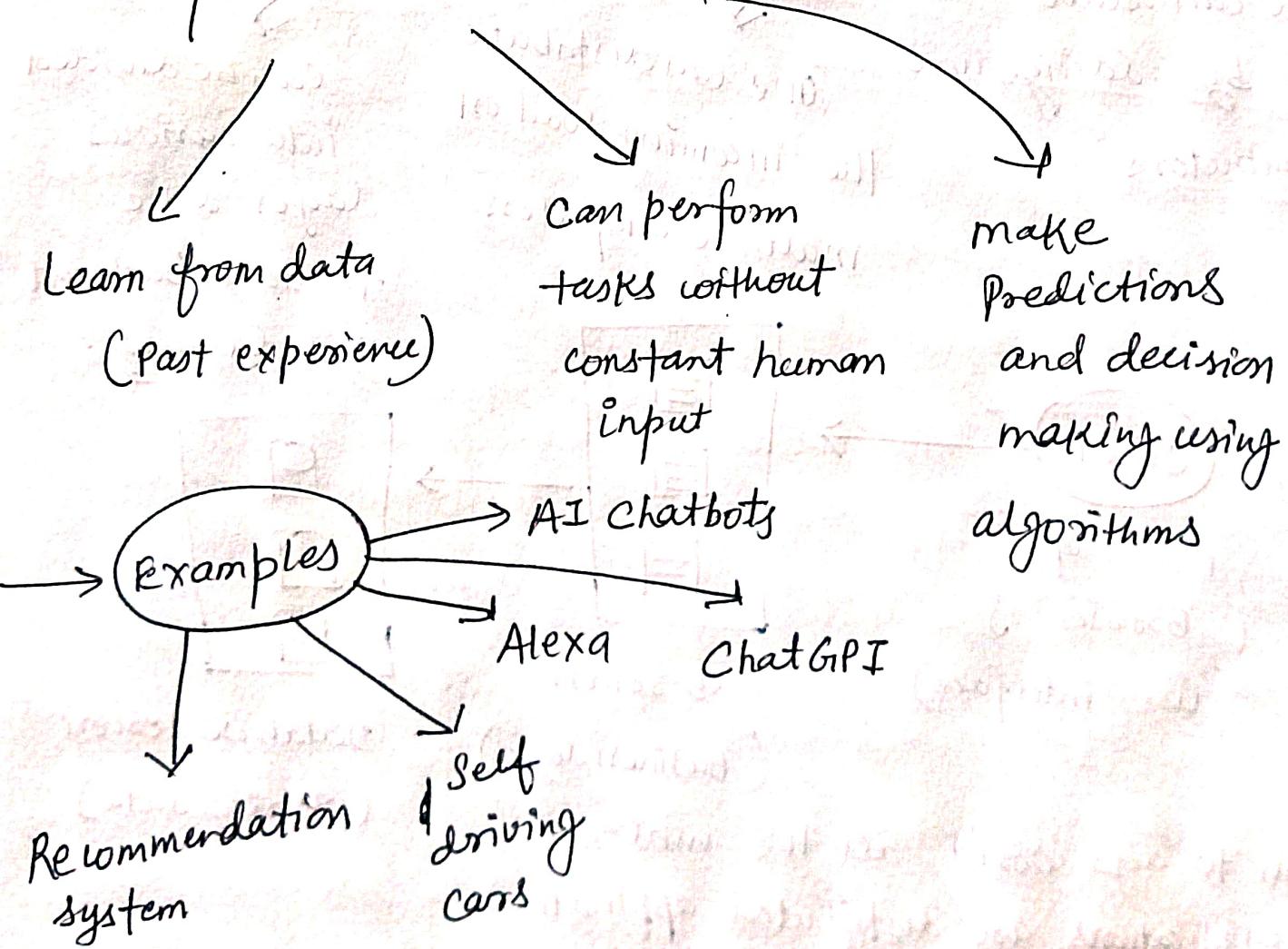
In today's world, all the modern applications are distributed applications.

(Amazon, Netflix, Facebook, Dropbox, Google Drive...etc.)

The goal of such applications is to provide best user experience in terms of performance, scale and reliability. That's why they have large pool of various servers (Data centres) distributed all across the world!

5. AI applications

→ use AI techniques like machine learning, deep learning, computer vision, etc. to mimic human intelligence and perform task that usually require human thinking and decision making.



Now let's move towards developing a deep understanding of web technologies:

Frontend

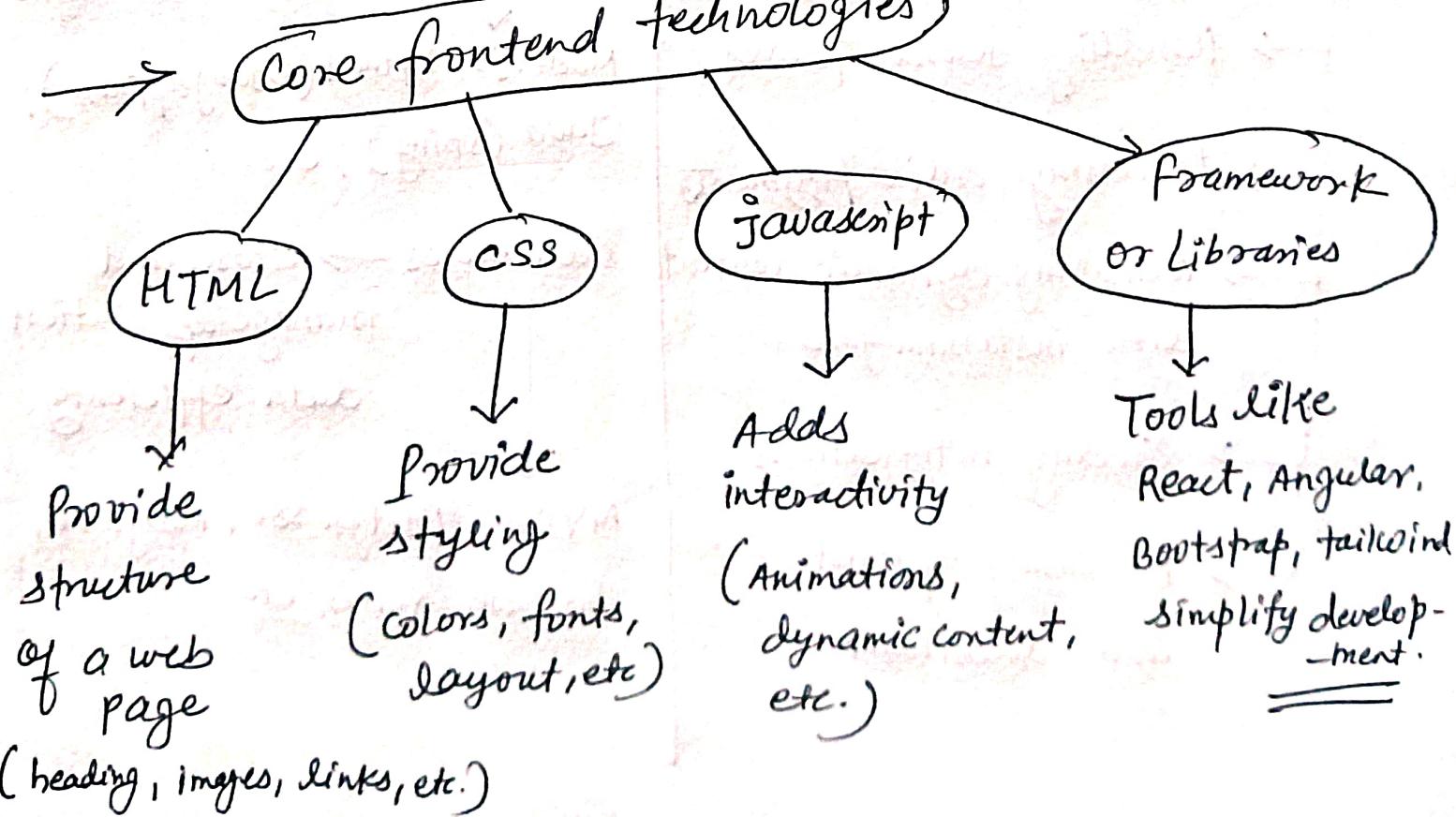
① Technology
(client side)

② Backend Technologies (server side)

① Frontend Technology

- A part of the web application that users see and interacts with directly in the browser.
- Display content, handle user interactions and communicate with backend to fetch and display the data.

Core frontend technologies



②

Backend Technology (server-side)

→ A part of web application that runs on the server. It is responsible for processing requests, managing databases and providing functionality to the frontend.

Handle business logic

Store, retrieve and manage data in databases

Handle authentication and authorization

Example of backend tasks

- Storing user accounts and password securely.
- Handling search queries
- Processing online payments
- managing dynamic content like Instagram feed.
- Sending notifications
- Processing upload and download

Code backend technologies

Programming → used to write server-side logic
↓
Node.js, Python (Django/Flask), Java (Spring), etc.

Databases → store and manage application data efficiently.
↓

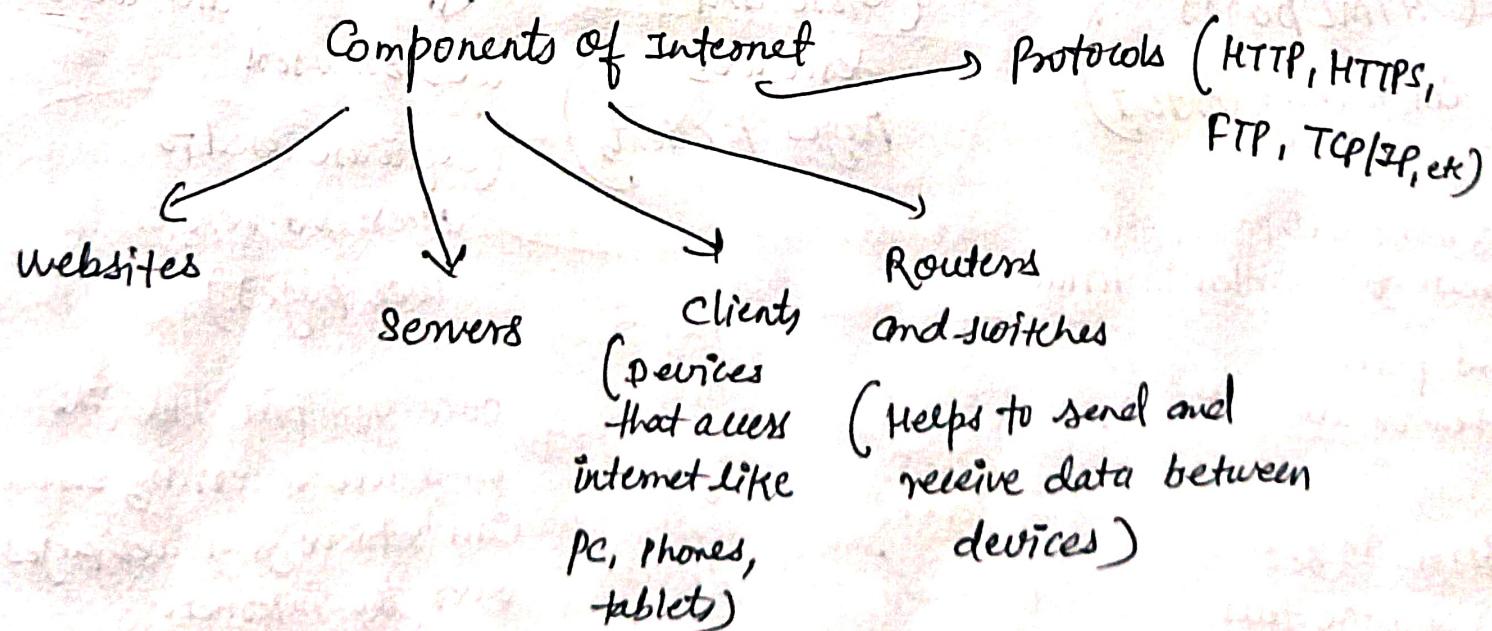
MySQL, PostgreSQL, MongoDB, firebase, Redis, etc.

Server/Hosting → Hosts applications and serve client requests

Apache, Nginx, AWS, Azure, Google cloud, etc.

What is Internet?

- A large network of connected computers and devices that communicate with each other using network protocols.
- Help to share information, access services and communicate across the world.
- When you open YouTube on your computer, your device connects to YouTube's servers over the internet. Then video data is transmitted from the server to your device for playback.



What is web?

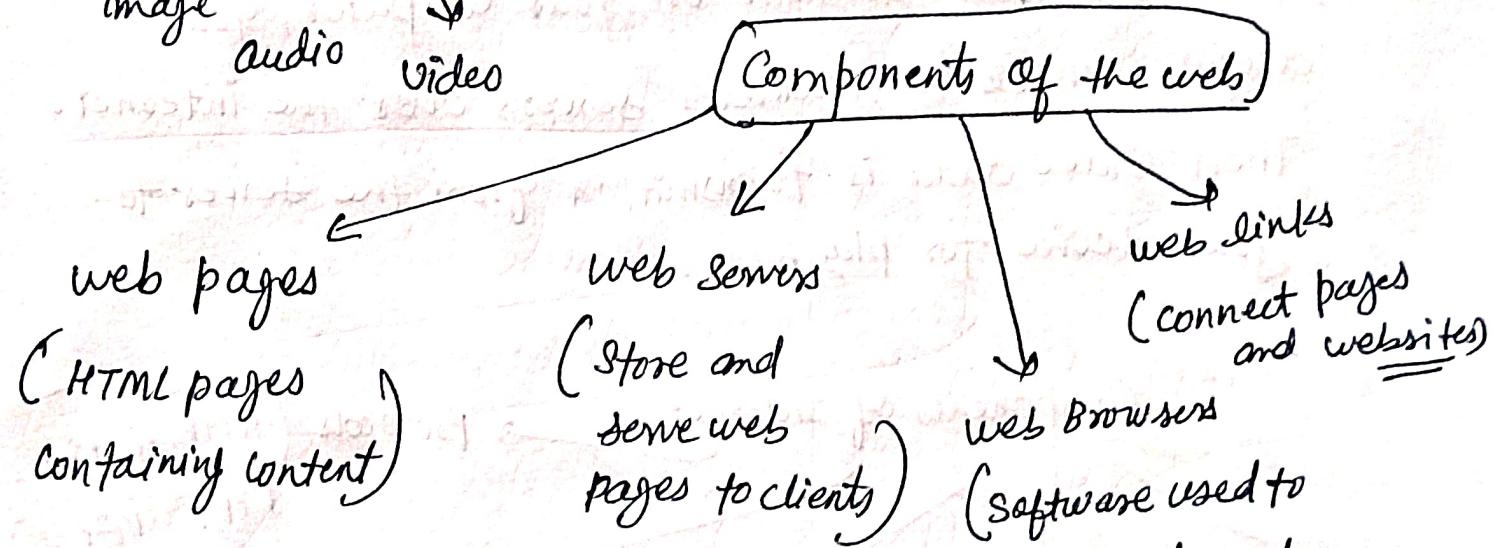
→ A system of interlinked documents and resources that are accessed via the Internet using web browsers.

use hyperlinks to connect pages and resources

Each resource on the web has a unique web address i.e. URL (Uniform resource locator)

use HTTP and HTTPS protocol for secure communication

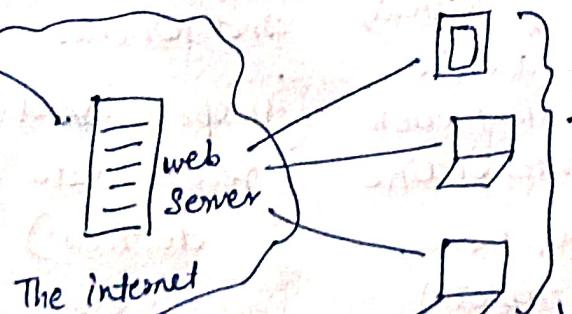
Text
Image
audio
video



To make web pages, you create file written in HTML

and place them on web server.

HTML in your web page tells the browser what it needs to know to display your page



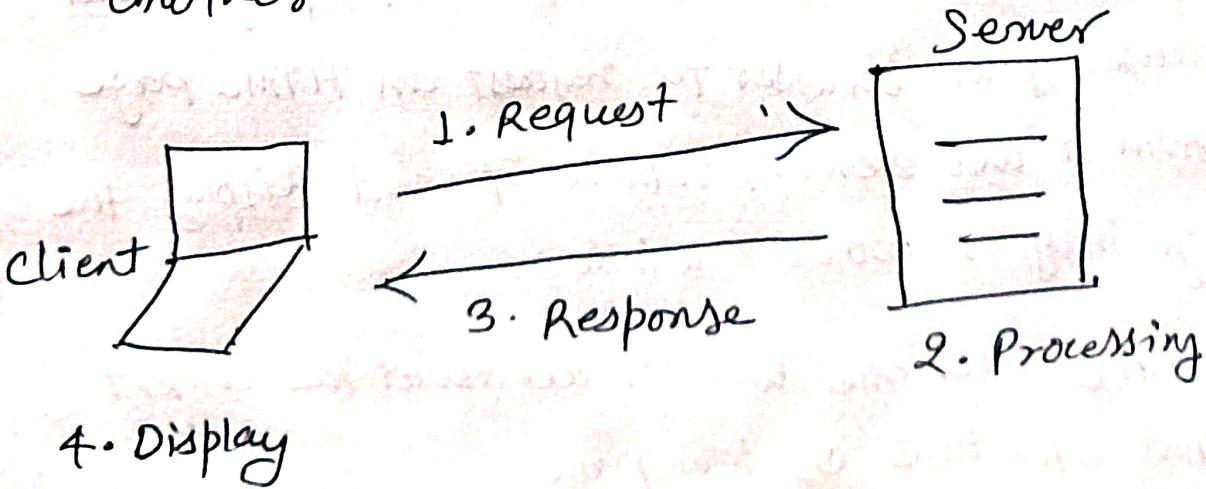
once you put your files on a web server, any browser can retrieve your web pages over the Internet.

There are a lot of PCs and devices connected to Internet all running web browsers.

Client-Server Architecture

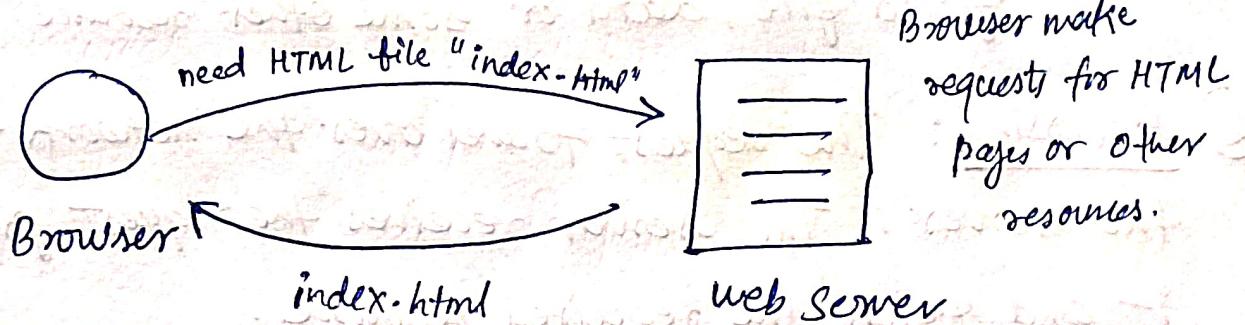
- Two main components: Client and Server.
- Client send a request for resources (HTML, CSS, Images) and server responds with the appropriate response.

- ① Request: Client (browsers, mobile app) send a request for a file, data or some other resource.
- ② Processing: The request travel over the network to the server. The server receives the request, process it and fetch the required resource.
- ③ Response: After processing server sends a response back to client. This response can be requested data, file or error message.
- ④ Display: The client receives the response and displays the details to user or perform another action based on response.



what does the web server do?

- Always waiting for requests from web browsers.
- Requests for web pages, images, videos, etc.
- When server gets a request for any of these resources, the server finds the resource and then send it back to the browser



- Web server is just a computer connected to the internet waiting for requests from browsers.
(If server has those ~~no~~ resources, server locate it and sends it to the browser)

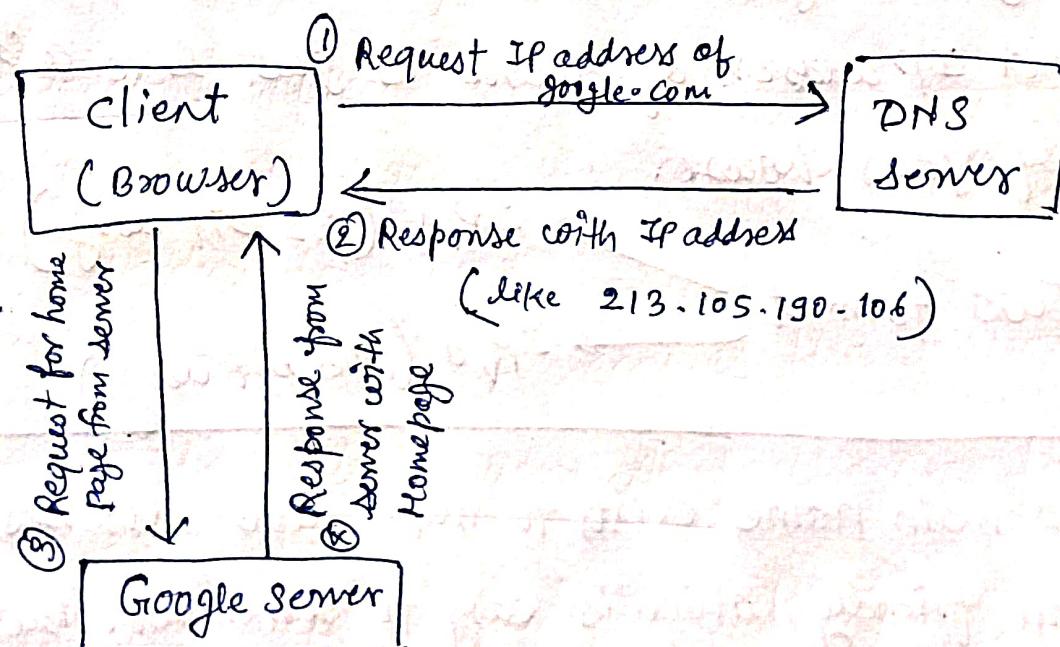
What does the web browser do?

- When you click on a link to visit a page, that causes your browser to request an HTML page from a web server, retrieve it and display the page in your browser window.
- HTML tells the browser all about the content and structure of the page.

when you open www.google.com.

- Browser (client) requests the Google homepage
- Google's web server processes your request and sends back the HTML, CSS and JS files.
- The browser renders the page for you to interact with.

But above steps are very abstract view. These are a lot of details behind the scene. Let's explore it!



Before explaining the above diagram, I would like to explain few terms for better understanding:

what is domain name?

- A human readable address used to access website on the internet.

→ It is easier to remember than the numerical IP addresses.

what is IP address?

IP → internet protocol

→ A unique id for each device connected to internet. Helps ~~and~~ devices to identify and communicate over the internet.

→ This is just like "home address" for devices on the internet. For example, if internet is a city then your device is a house in the city and IP address of device is just like address of your house.

⇒ Two format → IP_{v4} (32 bit long)
→ IP_{v6} (128 bit long)

So domain name exists to make web addresses human friendly, which is easy to remember. But to connect to web server to get the web page, browser need the IP address of web server.

⇒ So here comes DNS (Domain name system)

What is DNS?

- A system that translates human-readable domain names (www.google.com) into machine-readable IP addresses (like 142.250.190.78).
- Just like a Internet phone book.
- ~~Without DNS, you need to remember IP addresses of every web server of a website.~~

So here are the step by step explanation of what happens when you type an URL into browser?

- ① Browser checks cache. If requested web page is present in cache, browser load the page quickly without contacting the server.
- ② If requested web page is not present in browser cache, browser needs to find the IP address of the server hosting the website.
- ③ Browser sends a request to DNS server to get the IP address of server hosting the domain.
- ④ DNS server return the IP address of ~~domain~~ the server.

- ⑤ Now browser establishes a TCP connection with the server using IP address.
- ⑥ Now browser sends HTTP request to server, asking for web page.
- GET /index.html HTTP/1.1
Host: www.google.com
- HTTP request format
- ⑦ Server processes the request and retrieves static files (HTML, CSS, JS), execute server side logic, (if needed) and fetches data from database (if needed).
- ⑧ Now server sends an HTTP response back to the browser. It may contain status code (200 OK, 404 not found, etc) and content (HTML, CSS and JS).
- ⑨ Now browser parses the HTML content and constructs the DOM (Document Object Model), apply CSS to style the page, execute JavaScript for interactivity and fetch resources like images or videos. (Rendering process)
- ⑩ After rendering, the web page is fully visible and interactive.