

~~Date  
26/06/93~~

## 5 Web Technologies

We have seen HTML

CSS

JavaScript

Node JS

Runtime of JS

Architecture

②

What is node JS.

Node.js is an open source runtime Environment that allows to run JavaScript code outside the web browser. It is mainly built for application programming interface (API).



Node.js allows to develop Server side and networking applications using JavaScript.

handle these services using Node.js'

Finally Node.js is a runtime environment that allows developers to use JavaScript for server side and networking applications. Its Event-driven, non-blocking I/O model, fast execution speed, extensive package ecosystem and Cross platform capability have contributed to its popularity in development community.

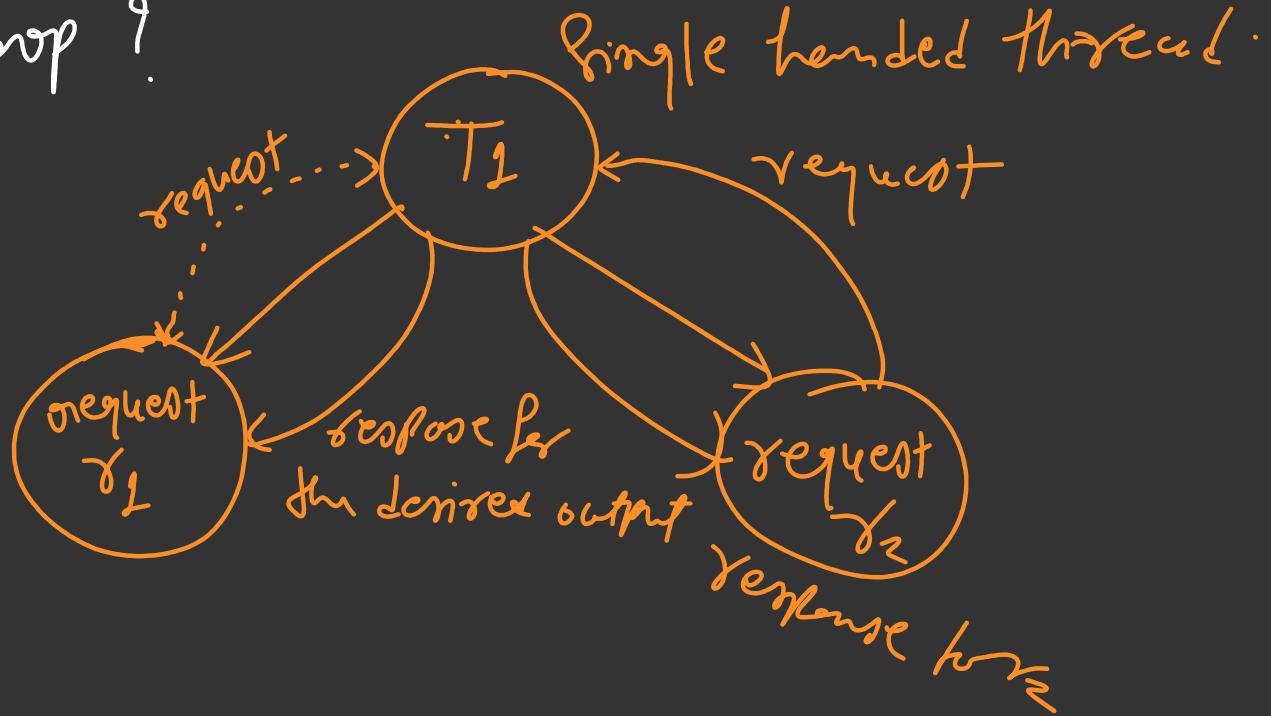
for example :

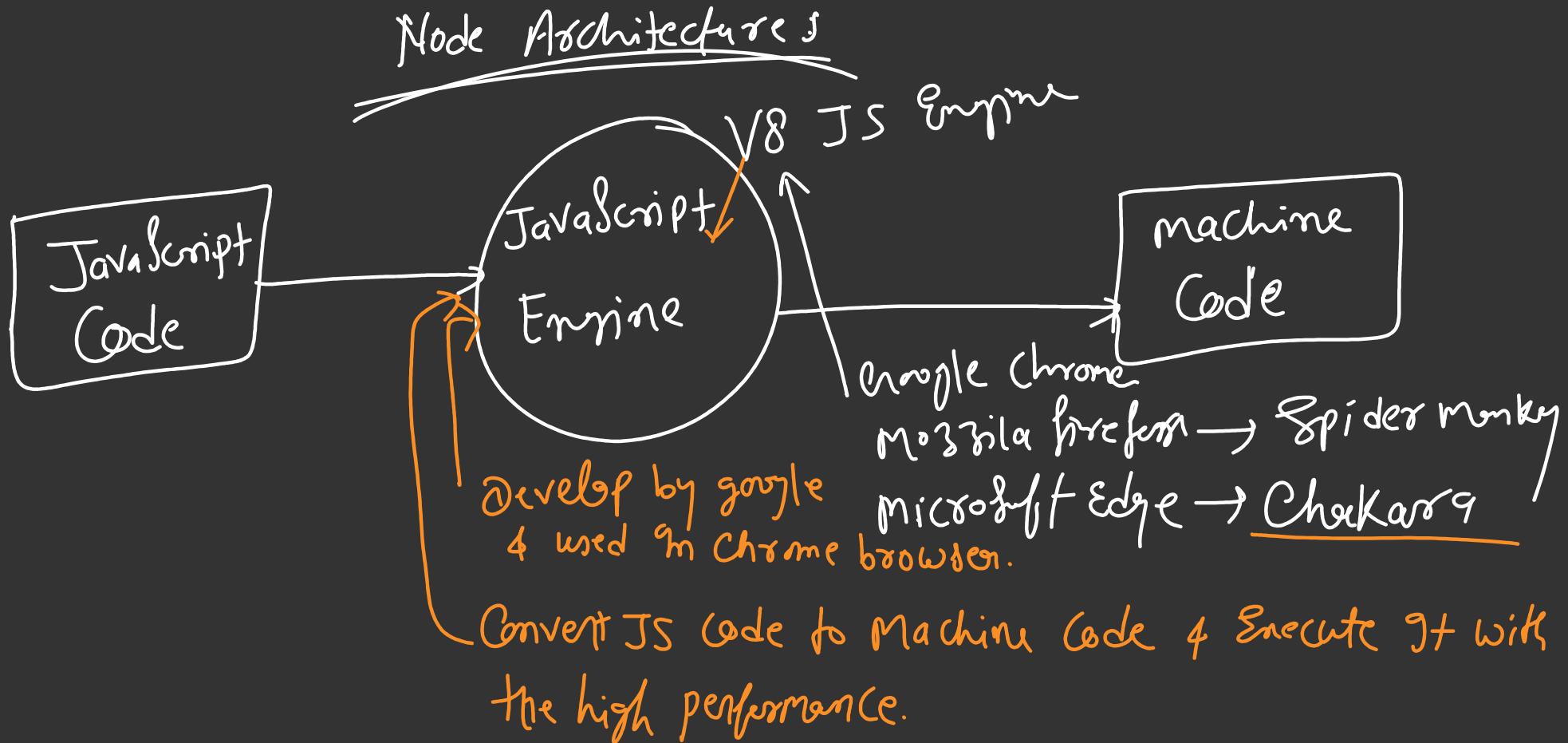
Application  
using node.js

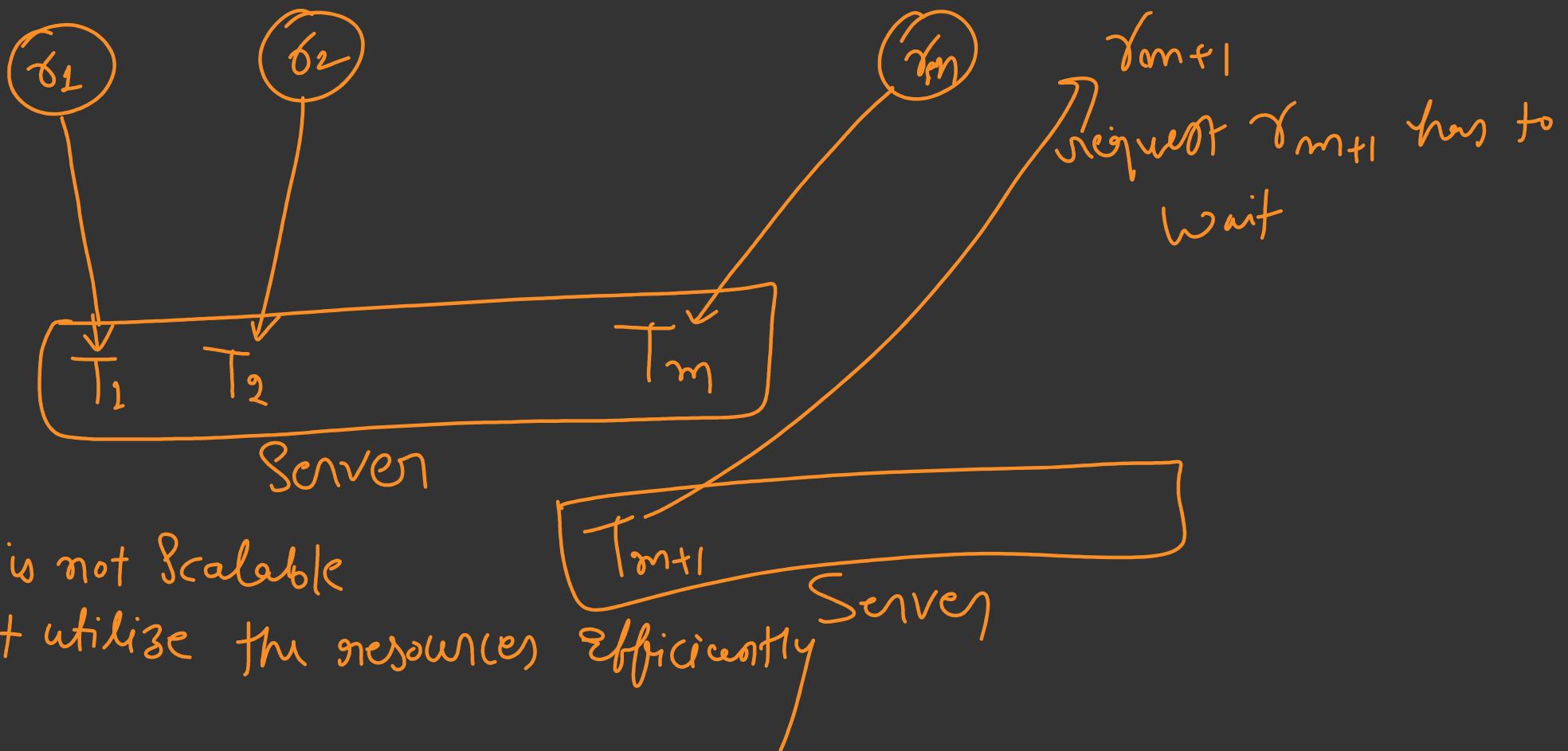
- c. Build twice as fast with fewer people.
- . 30% fewer line of code .js files.
- 2-3 times faster.

Event loop: It allows node.js to handle multiple concurrent requests efficiently. It continuously monitors the event queue and execute call back function when event occurs. It ensure that the execution remains non-blocking and asynchronous.

Why event loop ?

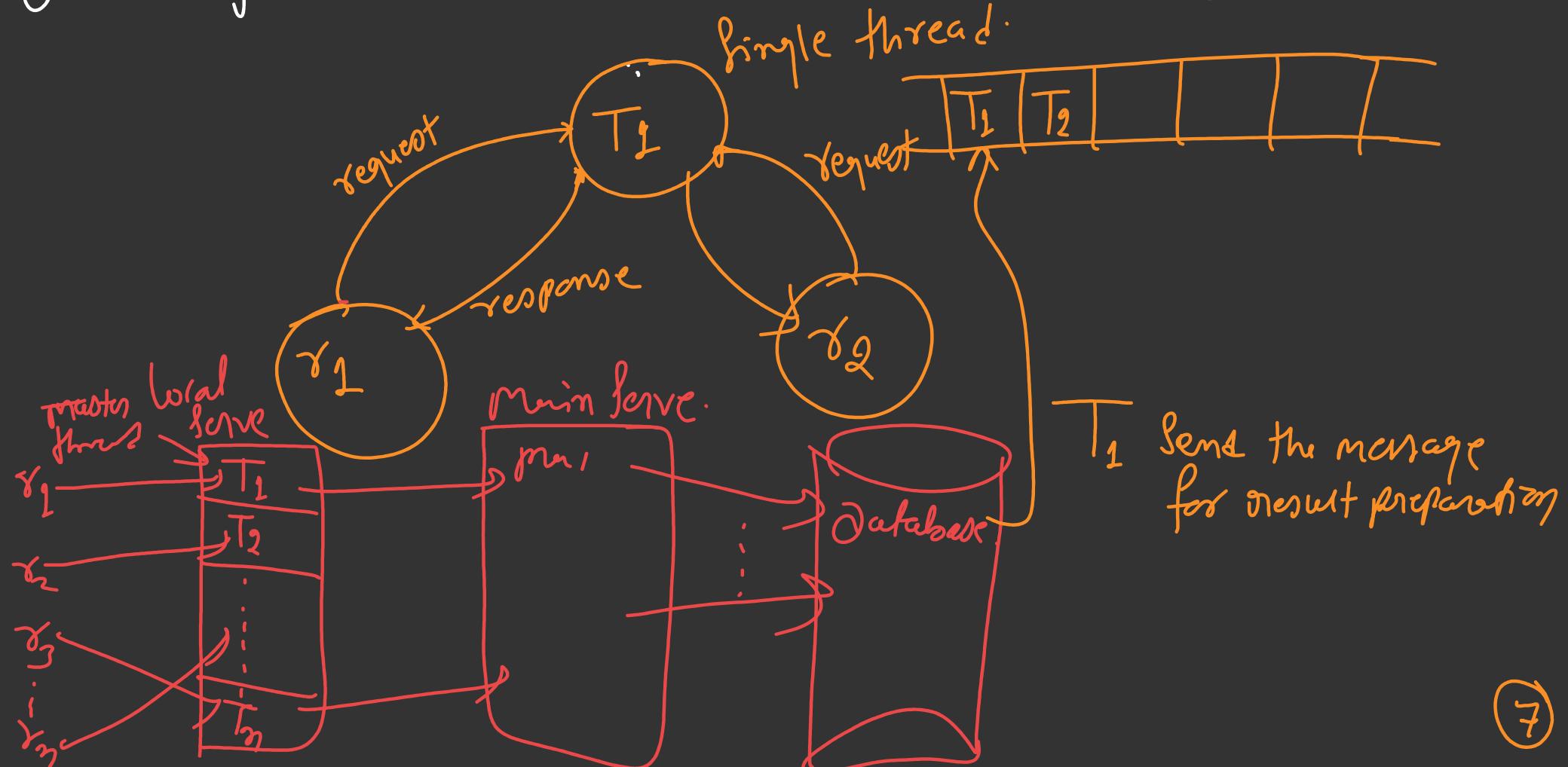






- It is not Scalable
- Not utilize the resources efficiently

Node.js Overcome this types of issues.  
Due to asynchronous nature of node applications by default.



T1 first take request  $r_1$  and Connect to the server for result  
gt doesn't wait for the response and serves the request  $r_2$   
When database prepare the results gt put to the memory <sup>in Event queue</sup>

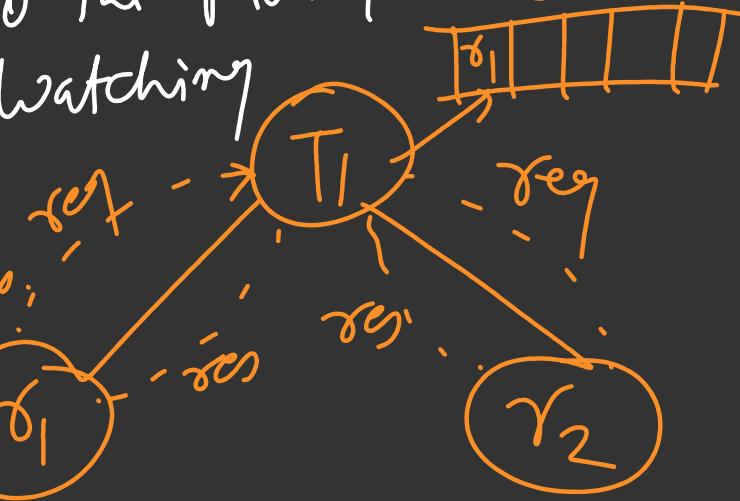
Event queue. and thread Continuously watching  
to the Event queue.

Node is the idea for I/O intensive apps;

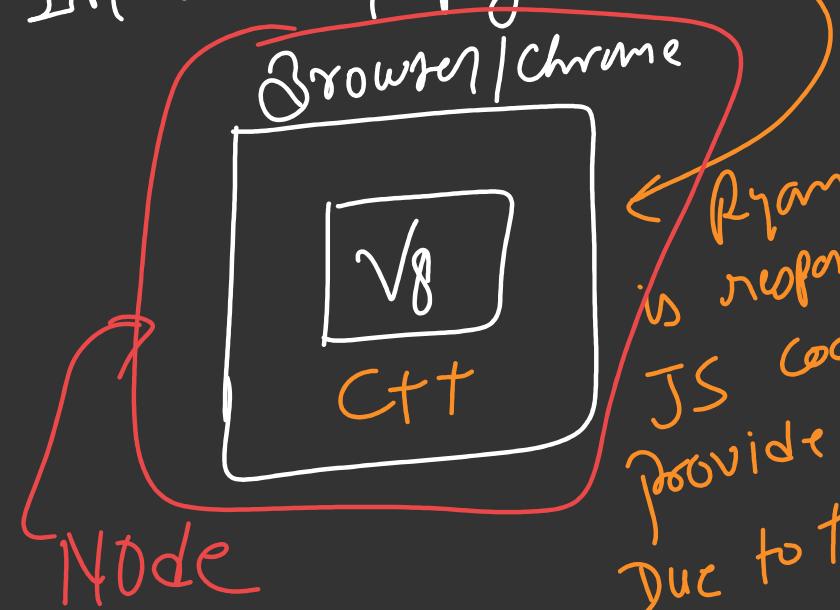
Node do not use the CPU intensive apps.

for example: `document.getElementById('');` ← JS

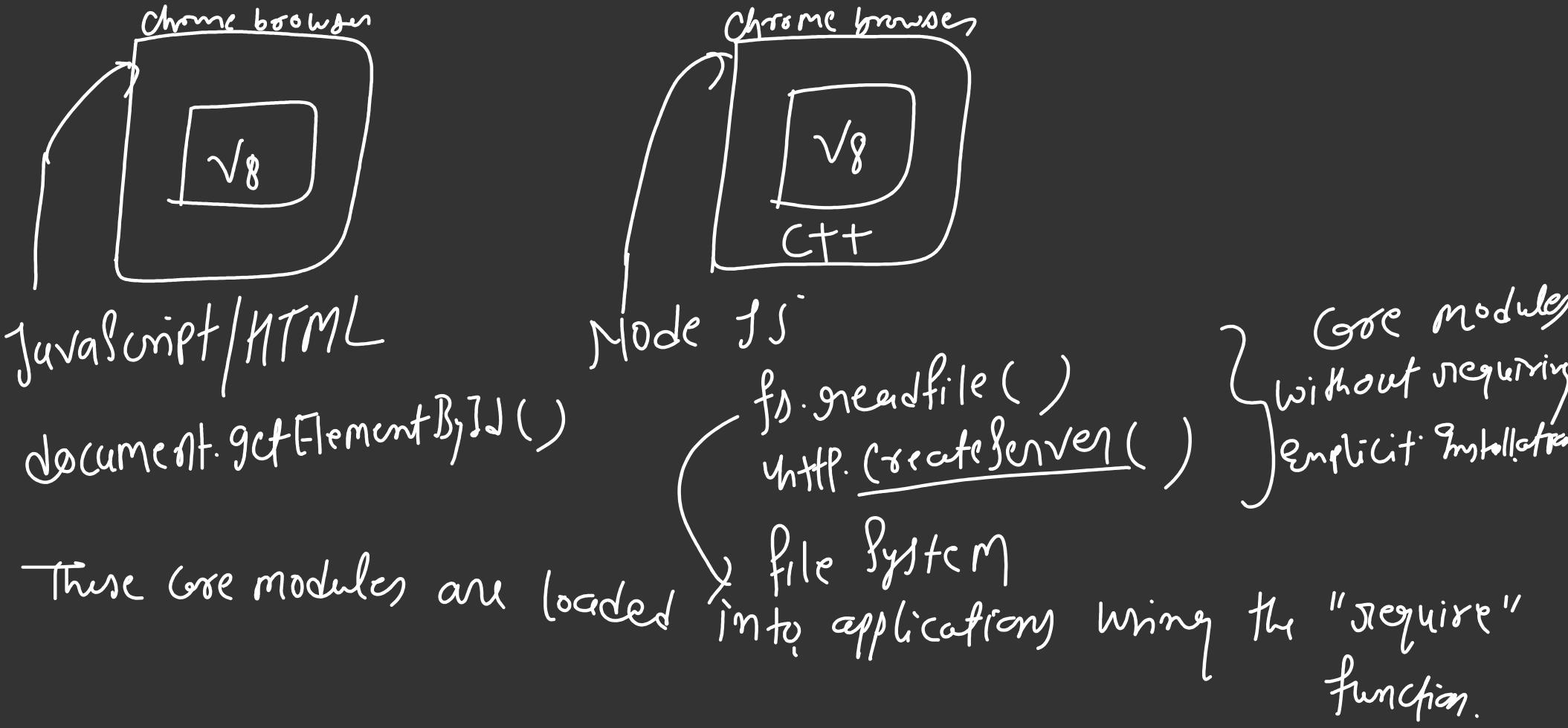
Window Object



In 2009, Ryan Dahl created the Node.js.



Ryan Dahl embedded the C++ code which is responsible for JS engine to execute the JS code. It also has certain objects that provide runtime environment for JS code. Due to the above changes in V8 & Chrome the node JS has capabilities to execute the code on the server.



Install the node js in yr system.

- open the browser
- Type Nodejs.org
- Download the Nodejs.
- Install the Nodejs.

} Installation process.

• check the version of node js in yr system.

- - Directory/Path → node -v. <sup>for</sup> Windows.

↳ V18.1(1)

Write a program in Notepad, Save as xyz.js then run at terminal.

or CMD

- Connect node.js applications to the local server:  
i.e. localhost
- 1) Set up the local server      Express.js / Apache / WAMP / Microsoft 360  
  ↑ find out.
- 2) Create a node.js application : Develop node.js application using JS.
- 3) Define the Server to handle incoming requests. For that you can use the built-in 'http', 'https' modules in node.js to create the server.

```
const http = require('http'); // function for loading the core modules.  
// core module.  
const server = http.createServer((req, res) => // object.  
{  
    res.writeHead(200, { 'Content-Type': 'text/html' });  
    res.end('Hello world'); // executing the code at localhost  
}); // Assign the port number for executing the code at localhost  
const port = 3000; //  
server.listen(port, () => //  
{  
    console.log(`The server is running on port ${port}`);  
});
```

. Start the server → save yr file with myjs.js & navigate the directory where the file is located.

on a CMD

node filename.js

Access the node.js Server locally : once the Server is running we can access it in browser by entering the appropriate URL.

http://localhost:3000 /+ type in yr browser +/  
↑  
8080

CreateServer() method creates HTTP Server object on yr computer & creates a function

Date  
28/06/23

Last class : Node JS

- API
- Runtime Environment
- Installation of Node JS.
- Node architecture.
- functionalities of Node JS.

## { Modules In Node JS::

Node.js Modules are reusable pieces of code that can be imported and used in node.js applications. Modules allow developers to organize their code into separate files and use the functionality from one file to another file.

Two types of modules in Node.js.

- 1) Core modules
- 2) local modules

1) Core Modules: These modules are to be bundled with node.js and are available for use without any additional installation.

Core modules include the modules such as

{  
  'fs' file system  
  'http' (HTTP Server Client)  
  'Path' (Path of the file)

Core modules are typically accessed using the "require" function.

Final  
20 marks

Choice based  
T/F

30 marks

Programming

20 mark theory

```
const fs = require('fs'); // Core Module
fs.readFile("file.txt", 'utf-8', (err, data) =>
  if (err) // file path err
    throw err; // Encoding
  console.log(data); // data print
);
```

↳ Local Modules: These modules are created by the developer or third party.

↳ npm (Node Package Manager) ✓

↳ endpoints. SayHello = function( )

```
{ console.log("Hello");
```

```
}
```

```
const myModule = require('./modulePath');
```

```
myModule.sayHello();
```

node.js

endpoints.js

(19)

§ Node package manager: A NPM is a package manager for node.js to allows developers to discover, install and manage the external libraries.

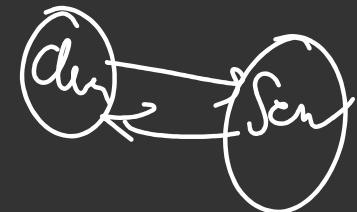
> Install NPM

> Install NPM Express.

{ n packages  
d package  
paid

§ Package.JSON (JavaScript Object Notation)

NPM init → Creating a package.



20

# >Create a web server:

Core Module  
function/method objects.

```
const http = require('http');
```

```
const server = http.createServer((req, res) =>
```

{

```
res.setHeader('Content-Type', 'text/plain');
```

```
res.write("Hello Class"); ✓ Hello class.
```

```
});
```

HTTP: localhost:3000  
Display the output

```
server.listen(3000, () =>
```

```
console.log("Server is listening on Port 3000");
```

Node.js

Server

filename.js

Execute

Node filename.js

21

file system in node JS ::

Read the file

`fs.read(path, option, callback)`

file path (absolute/relative)

Encoding

gt is a function (err,data)

) if error occurs gt will gives the  
Error message

Print/display the file data.

`fs.writeFileSync(file, data, option, callback)`

  ↑  
  data want to be write  
  Path      Encoding      function

file manipulation

`fs.rename(oldpath, newpath, callback);`

Date)  
03/07/23)

## { Web Technologies }

### Node JS

In Node JS, we have seen

#### 1) Node JS Basic Concepts

- V8 Chrome runtime Engine
- Custom with C++ + V8 browser
- Cross platform
- I/O operations.
- Non blocking I/O operations.

## modules of Node JS

→ Threaded I/O operations.

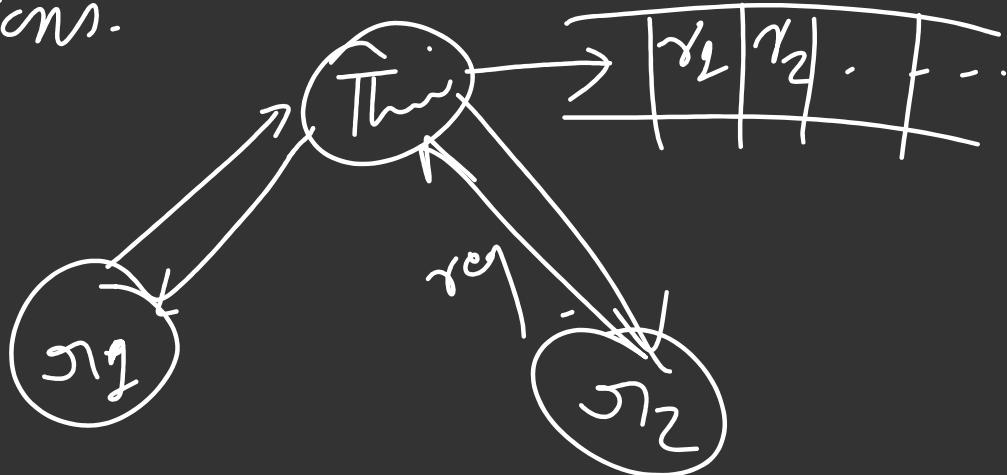
→ Core modules

→ local modules

→ Callback method

→ Required function for loading the Core modules.

→ Event loops to handling the requests using queue.



- 3) Set up & development of Node JS
  - ↳ how to install node JS
  - ↳ Adding dependency package JSON
    - ↳ Server Manager
  - ↳ File System
    - ↳ read a file
    - ↳ write a file

How to Send HTML as response

```

    const http = require('http');
    const hostname = 'localhost';
    const port = 3000;
    const server = http.createServer((req, res) => {
        res.statusCode = 200;
        res.setHeader('Content-Type', 'text/html');
        res.end(`<h1> DVCS </h1> <p> Dept of Comp Sc. </p>`);
    });
    server.listen(port, () => {
        console.log(`Server running at http://localhost:${port}`);
    });

```

Annotations:

- function**: core module
- core module**: allows node.js to transfer data over http/https protocol.
- hostname**: taking from local server
- port**: can create/run multiple servers on CMD
- server**: node.js

§ Suppose we create different routes over multiple responses.

```

server.listen(port number,
  hostname, () =>
  {
    console.log(`Server is
running at http://
${hostname}:${port}`);
  });

```

**CMD**  
node file1.js

**http://localhost:3001/home**

**http://localhost:3001/about**

Case "/home":  
 { res.writeHead(200);  
 res.end("<h1>
This is the home page for
DU </h1>");  
 } break;

Case "/about":  
 { res.writeHead(200);  
 res.end("<h1>
About the department
</h1>");  
 } break;

Default:  
 } break;

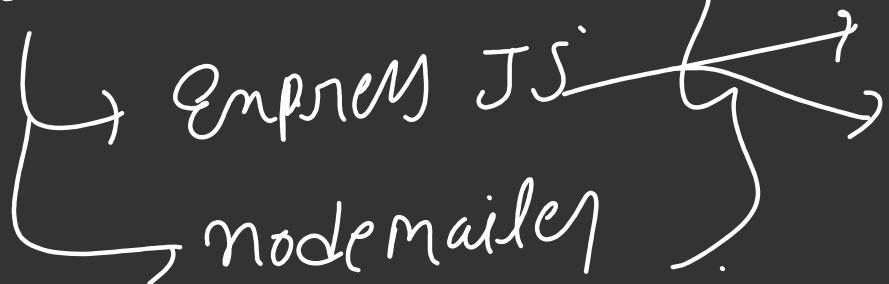
```

const http = require('http');
const hostname = 'localhost';
const port = 3001;
const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type',
    'HTML/Text');
  switch (req.url) {
    }
  }
);
server.listen(port, hostname, () => {
  console.log(`Server is
running at http://
${hostname}:${port}`);
});

```

§ Node Modules:: We have seen Node Modules fs, http.

other modules



Install the Express JS  
Install nodeMailer

# How to Create a Local module.

Reusability

```
import calc=calc
```

```
var dt=require
```

```
("./path of local  
module");
```

```
console.log(dt.calc  
(10,20,'+'));
```

```
Case '-':  
{ return a-b;  
break;
```

```
Case '*':  
{ return a*b;  
break;
```

```
};  
.  
default:  
{ return;
```

Const Calc(a,b,p)

```
{  
if (a||b||p)  $\frac{a+b}{3+4}$ 
```

return "Please provide  
all parameters";

```
};  
Switch(p)  
{
```

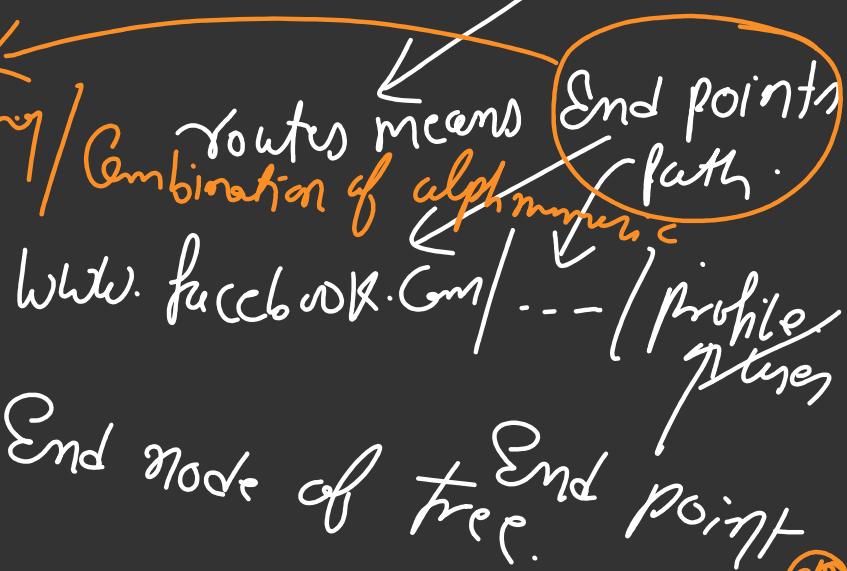
```
Case '+':  
return a+b;  
};  
break;
```

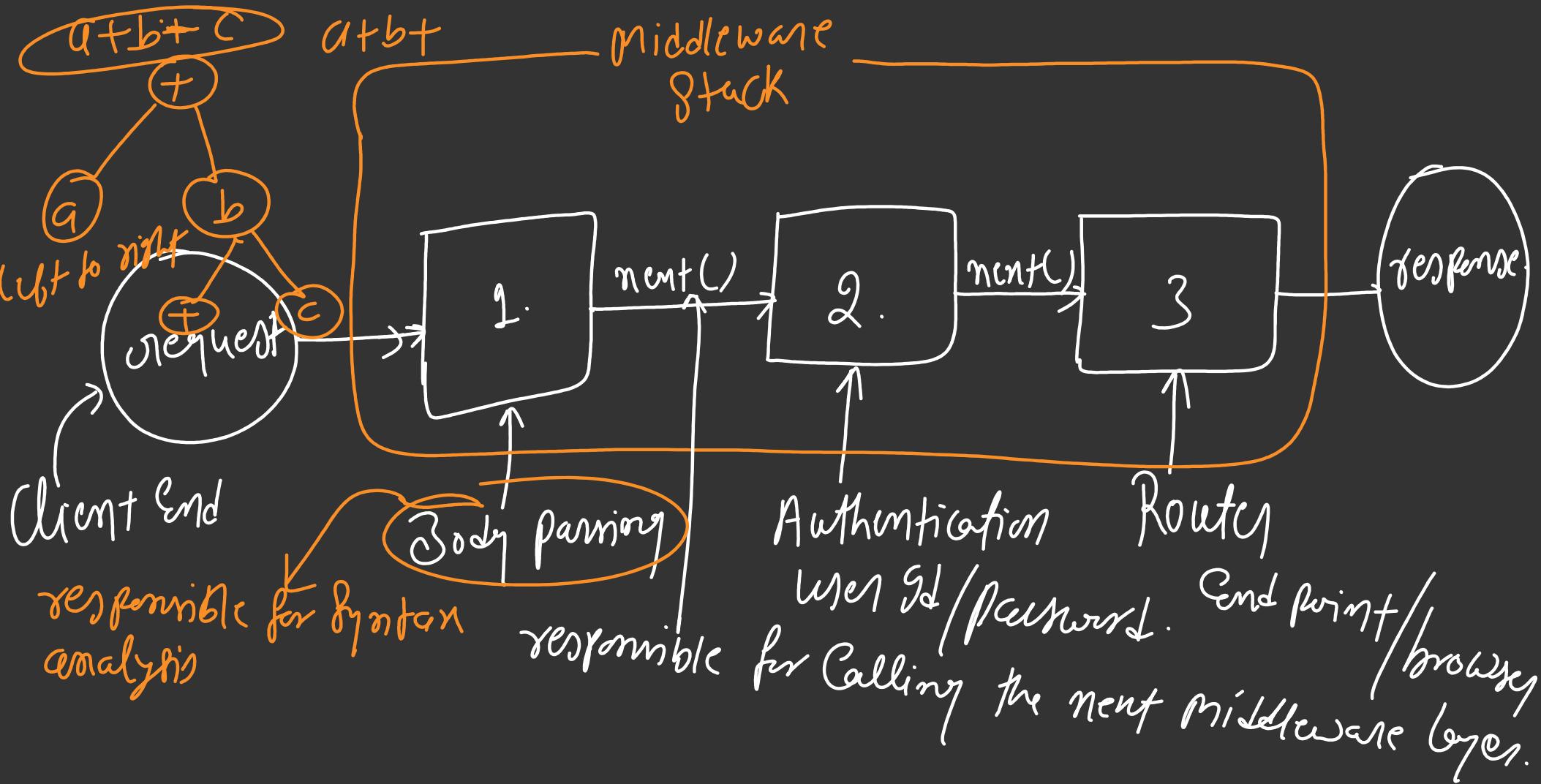
How package.json file looks like ? ← Assignment

# Express JS framework for Node JS

↑  
predefined programming tools.

- It is a popular framework to develop web applications in node.
- It is used to develop web applications such as web servers, streaming engines etc.
- Express JS creates a simple web server with different routes.
- Middleware used in Express.js





```
const express = require('express');
```

```
const app = express();
```

```
app.get('/', (req, res) => { res.end("Hello"); })
```

To handle GET request and app.post to handling the post requests.

Install Express JS → Assignment (last date 05/07/23)

## Why Express JS ?

Node.js is JavaScript runtime Environment that is fast but it does not support request handling methods or serving files. So this where Express JS Comes into picture.

→ Why not support the request handling methods.  
not good middleware.

→ Additional features like abstraction & simplify the error.

try  
{

Catch block.

What is Express JS?

Express.js is a framework for Node.js designed for building APIs, Web Applications and mobile apps.

Express JS is high performance, fast, unopinionated and lightweight.

Abstraction.

It is used as a server side scripting language.

Active Community  
route efficiency  
Non blocking  
Design is simple.

open architecture  
Abstraction  
middleware  
good database  
mapping  
templates.

Features of Express JS.

- 1) Fast and accurate server side development
  - With the help of Express JS, Node JS can save the lots of time.
  - Using the tools as framework.

2) Middleware

→ Already discussed.

3) Routing: Already discussed.

4) Debugging

Express JS → middleware layers.

We can directly reach to the error point with the help of middleware.

Date  
07/05/23)

## § Web Technologies §

Last class: Express.js

Difference b/w Node.js & Express.js:

- 1) Node.js used for building both front-end and back-end of web applications, While Express.js used for building the backend application.
- 2) Node.js is runtime while Express.js is a framework.
- 3) Node.js written in C/C++, JavaScript while Express.js written in JavaScript
- 4) Node.js for client & server while Express.js is for Server only.

:

(38)

For database

Install database mysql  $\Rightarrow$  npm install mysql.

Const mysql = require('mysql');

Create a Connection.

Const Connection = mysql.CreateConnection({

host: 'localhost',

user: 'username',

password: 'password'

database: 'name of your database'

questions

marks (2-3)

- 1) What is the role of Event loop in node.js?
- 2) Explain the Concept of Single threaded nature of node.js. How does it handle the multiple requests simultaneously?
- 3) How does node.js supports handling of requests & responses?

5 Marks

- i) Write a code to Create an HTTP Server that listen on port no 3000 and return the text "helloworld".
- ii) Write a code to handle the form submission in node.js application using Express.js framework. Capture the <sup>form</sup> details and display it in the Console.
- iii) Read the file
- iv) Error handling and logging in node.js.

41

objective questions (10' marks)

Which of the following best describe the node.js

- 1) A web server
- 2) A Server side run time Environment for javascript
- 3) A programming language.
- 4) A front-end framework.

Identify the Error in following code.

Const fs = require('fs'); (10 marks)

fs.readFile('file.txt', (err, data) =>

{  
    if (Error)  
        {  
            Console.Error("Error  
                        Handling  
                        Request");  
        }  
    Else  
        {  
            Console.log("File Content", data);  
        }  
}

42

Date  
10/07/23

## { Web Security }

Hackers : Hackers are programmers with having advanced knowledge of operating system & programming language as well as CMD.

⇒ At the time of COVID-19.

→ Create a fraud registration domain.

Like fake E-commerce shops.

Duplicate <sup>↑</sup> Shops created by hackers



Phishing Emails & text messages.

↳ fraud Email & messages.

↳ digital spam.

Fake mobile apps:

↳ install unauthorized apps.

Data breaching:

↳ Transfer over the network.

- Social media Scams:-
- ↳ Demands for Money to send the unauthorized monies.
  - ↳ offering the fraud healthcare Schemes.
- Hackers always gives the threat.
- Threat:- It is a Malicious act that seeks to damage the data, steal the data etc.
- Eves Computer Viruses. AIIMS

Hacker Can hacked the website.

↳ Trust  
→ user (N)  
→ profit down

Who will provide the security.

↳ Money  
↳ Software Engineers Provide the Security

Web Security ::

- Deny effort
- Configuration of web server.
- good password Policies.
- Security Codes.

Robust

Why web security requires?

- ↳ prevent from hackers.
- ↳ denial of services.

From where these security threats come.

- ↳ Hostile Nation States.
  - ↳ Unfriendly nations.
- Terrorists group.

→ Corporate Spies & Organised Crime organisation.  
    ↳ Act of Stealing the trade secrets.  
    ↳ Attack on Core Algorithm.

→ Hackers  
    → Accept the challenges.  
    → Attack on Security layers  
    → Black mail.

## Examples of security threats-

- Malware
  - Corrupting the data over the Internet.
  - Control the system.
- Spyware →
  - Attack on real time information
  - Bank details etc.
- Phishing attacks.
  - Dominos debt hack

→ Ransomware →  
↳ Denying the access to the Computer system.

Date  
12/7/23

## { Web Security }

→ Security

- types of Attacks.
- family of Antivirus → Examples.
- Security issues
- Prevention on the Same.

# Basic Security Model:

WWW (World Wide Web) → Collection of pages

GET

POST

URL

Images  
Video  
Audio

FORM

FN1	LN1	Add1	:	Link1
#PH2	LN2	Add2	:	Link2
:	:	:	:	Link3
:	:	:	:	
;	;	;	;	

Frame	Uname	photo
Education Details.		
Address.		
:		
:		
Signature.		
Submit		

Database

# World Wide Web (WWW)

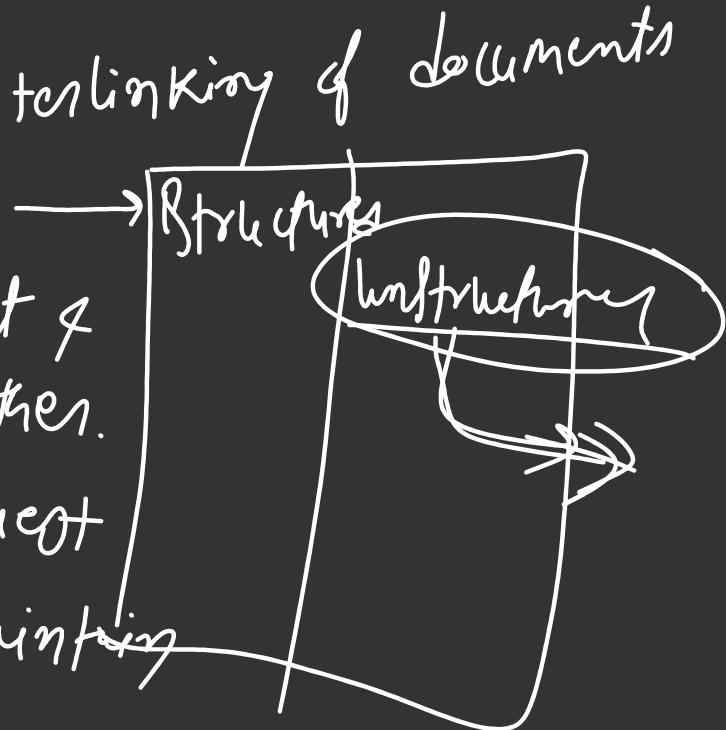
Allows multimedia documents to be shared b/w  
machines

from  
text  
Audio  
video

→ Resembles a Cob-web due to interlinking of documents  
which is Complex & unstructured.

HTTP: It is a method used at Client &  
Server side to interact each other.

HTTP is stateless which means Every request  
is treated discreetly and Server does not maintain  
the logical path.

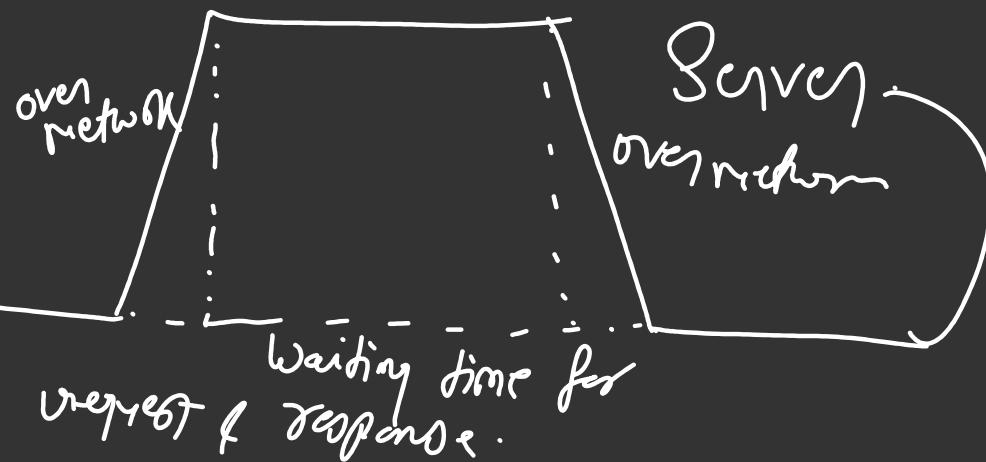


## HTTP Connections:-

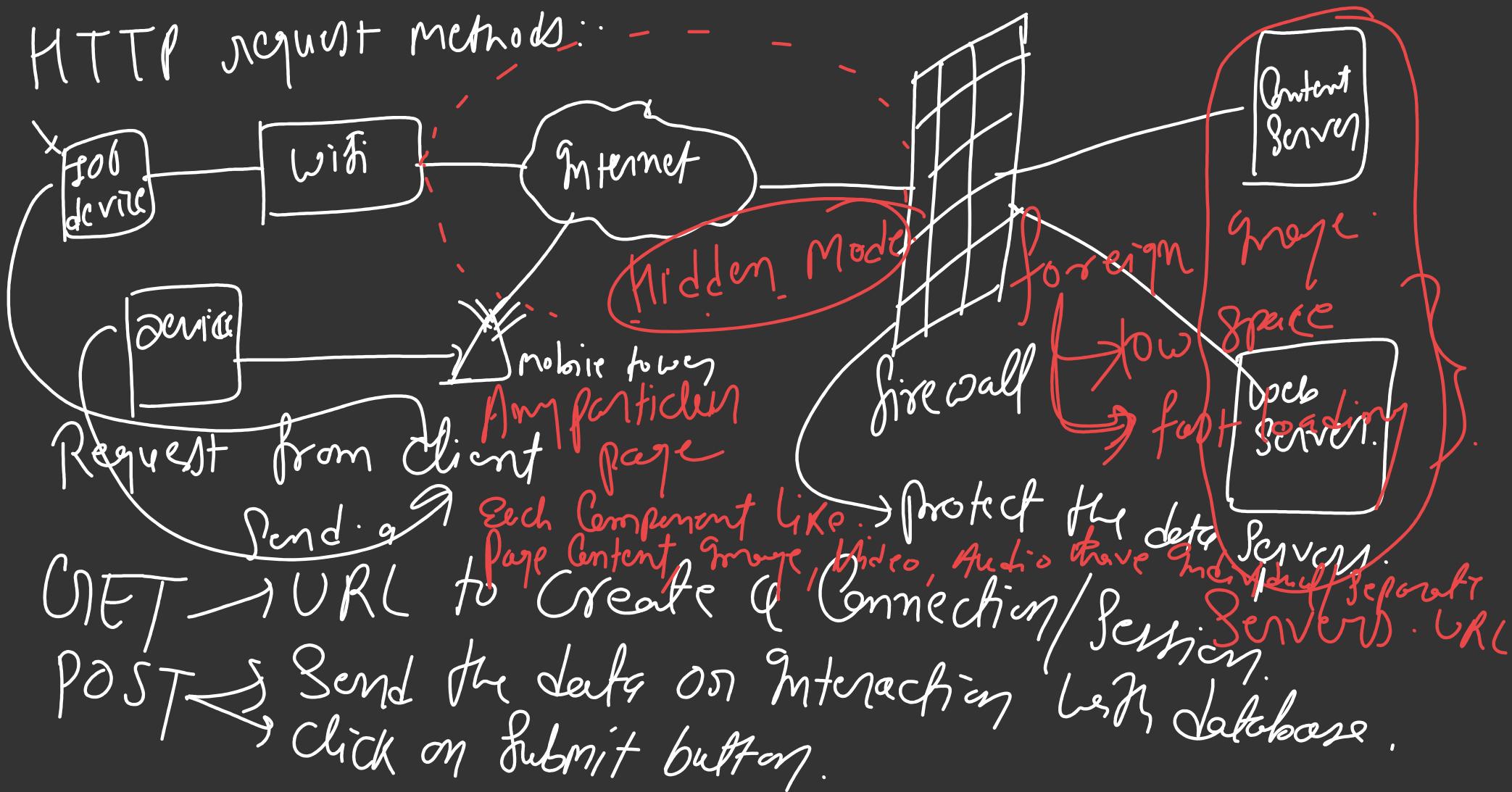
- Client open Socket Connection to HTTP Server  
*↳ Socket programming* ↳ Server →
- Client Sends the request to the Server
- Server Sends back the response to the Client
- Server Close the connection.

----- Request/response

Client



# HTTP request methods:



Date  
26/07/23

## { Web Security }

What are the Cookies?

Cookies are the files created by websites that you visit.  
They make online experience easier.

for example: Site can keep you signed in, remember your site preferences and give you locally relevant contents.

There are two types of cookies.

i) first party cookies → created by the site you visit.

ii) Third party cookies → created by the site visit by others.  
like images, text, ads. etc.

To See the Cookies:: (Google Chrome)

Open your browser

↳ At top right, Click on more settings.

↳ click privacy & Security

↳ cookies and other site data.

↳ click all site data & permissions .

Allow all cookies .

first party cookies. Third party cookies  
allow / delete.

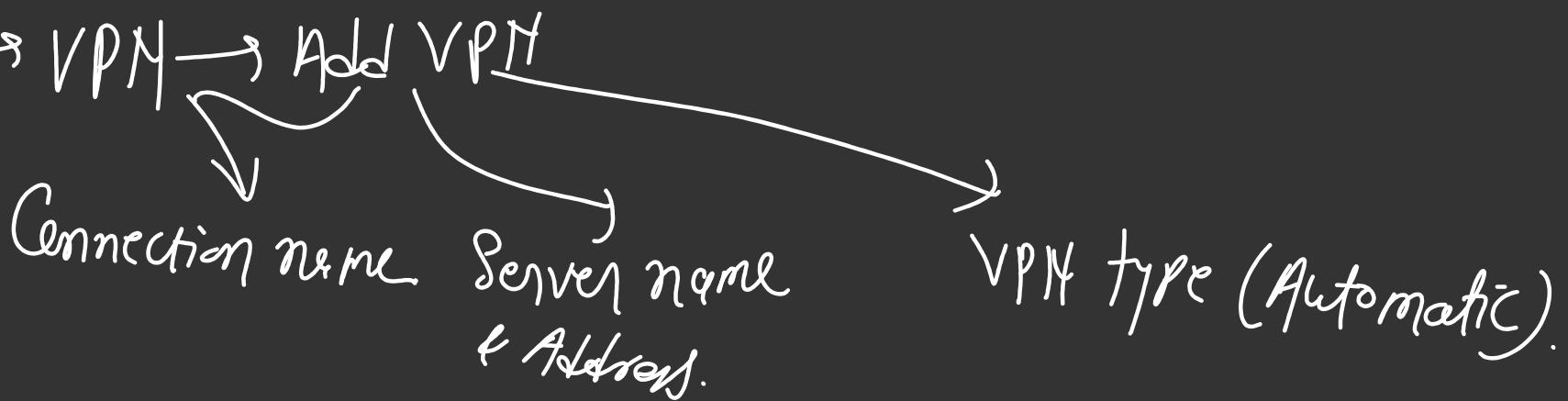
{ Anonymous Access :-

↳ use the web proxy to hide your IP address.  
or use the VPN to encrypt everything.

How to do it ?

Start → Setting → network & internet.

→ VPN → Add VPN



## { Authentication of WebRTC :-

Authentication : The act of Confirming the Identity of a potential user.

i.e. Authentication is the first step in any Cryptographic Solution. Because unless we know who is communicating so there is no point to communication.

How to implement the Authentication ?

Authentication may be implemented using Credentials, Each of which is Composed of a user ID & Password.

Now you have to prove that who are you.

I) what you know.

WU ID / password.

II) where are you

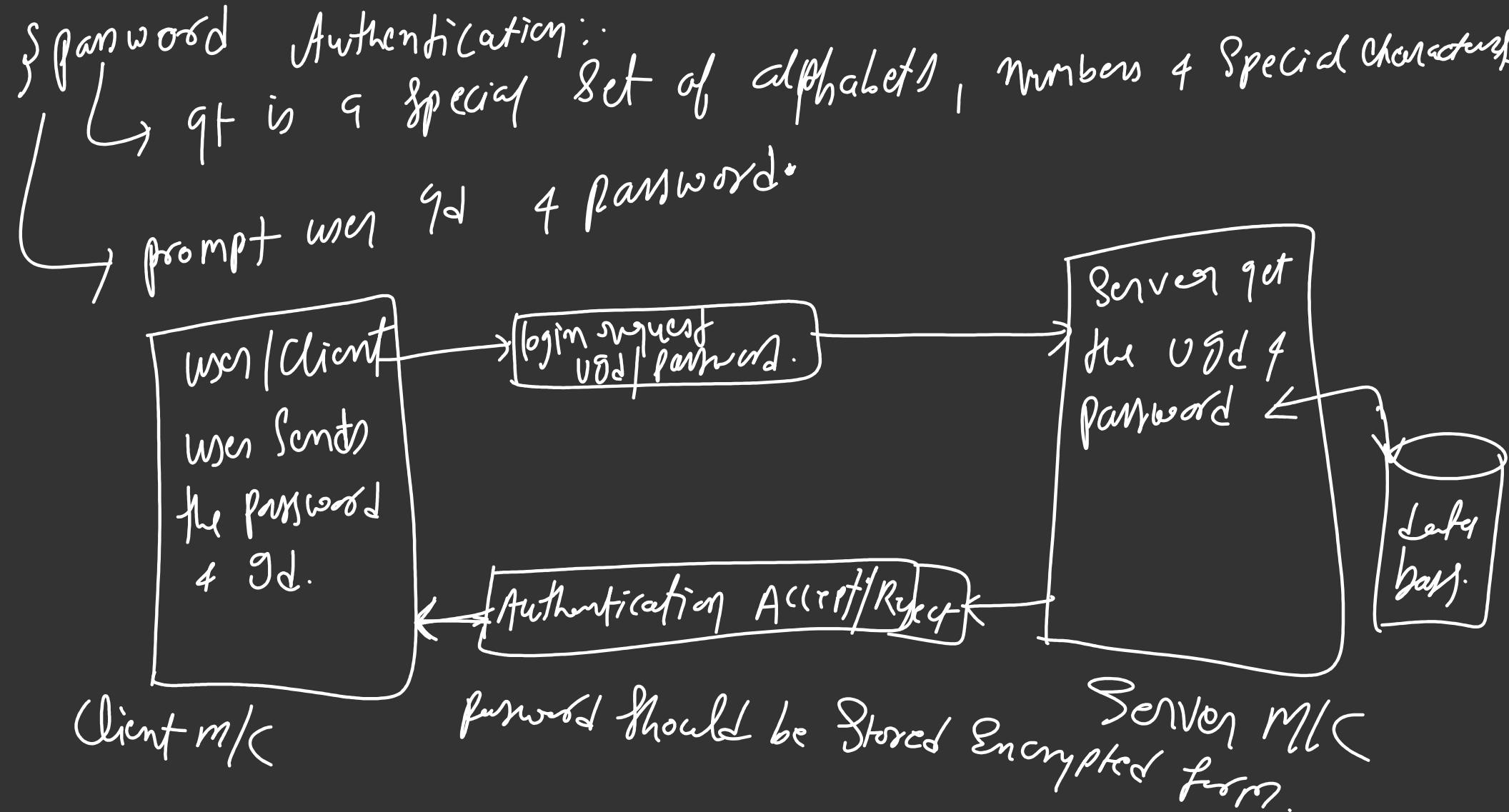
IP Address.

III) what you are

Biometric / Camera

IV) what you have.

Secure tokens / Smart Cards / ATM cards.



## § Centralized Authentication :-

