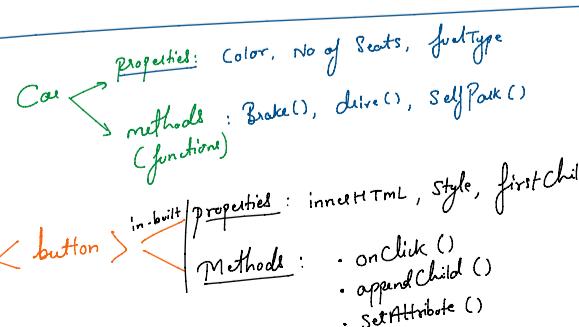
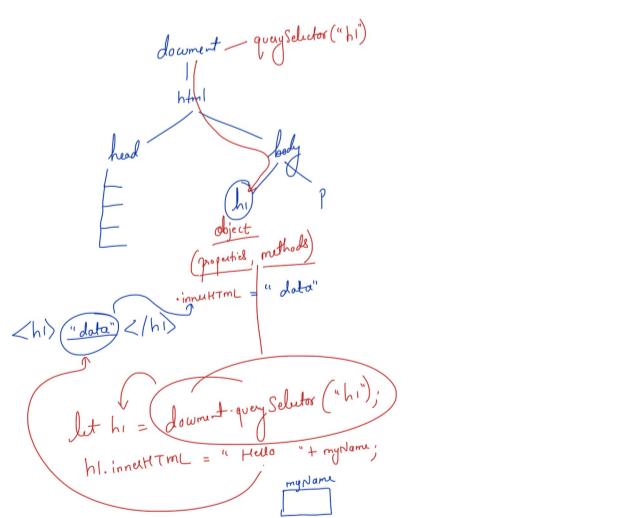


**DOM**: HTML file la Tree structure madhe convert karun zo tree aaplyala milto,

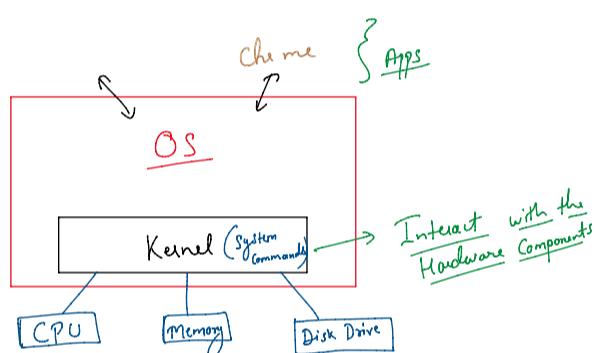
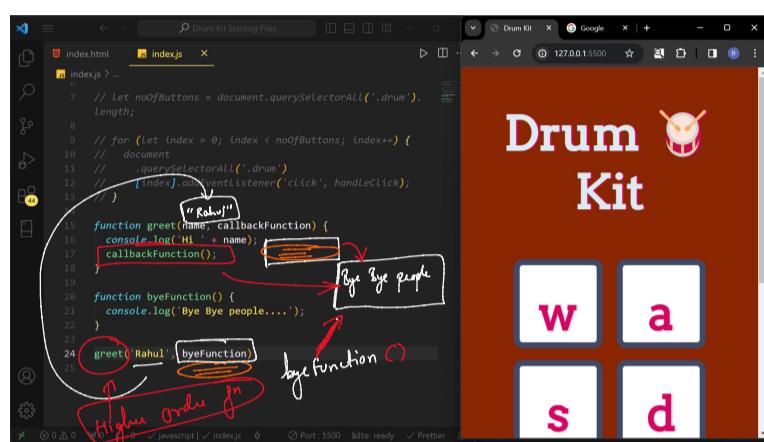
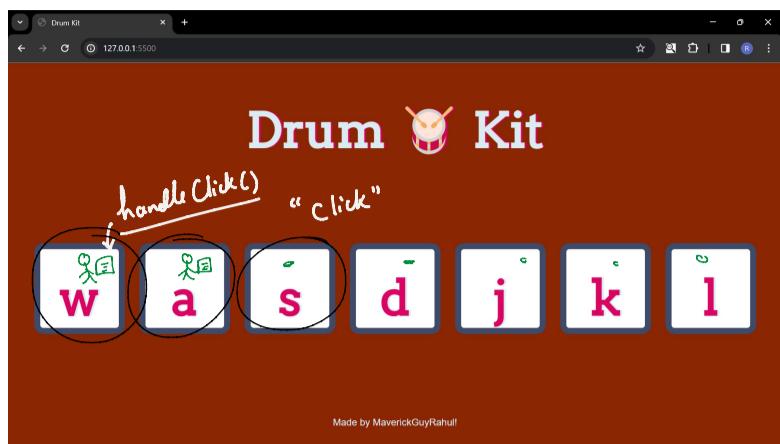
tyaas DOM tree asa mhantha zaato.

Usage: JS Can directly access, modify & delete the Dom objects  
So, Dom helps you to change HTML file structure using Javascript Code.

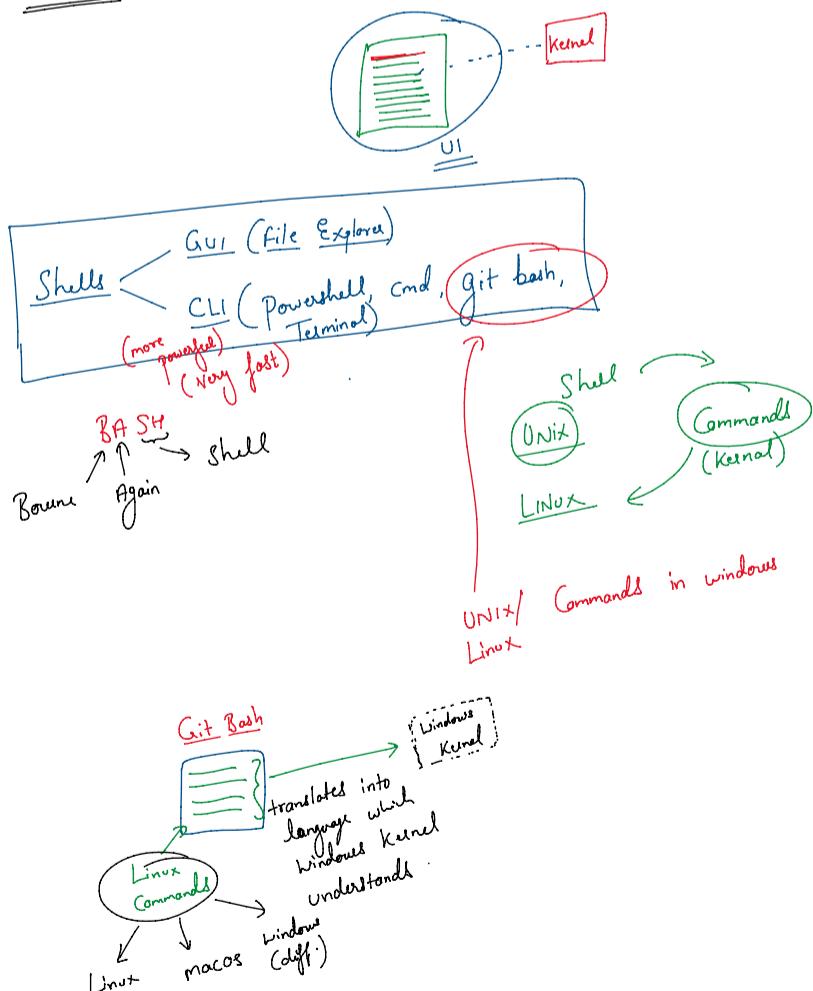
### Explanation

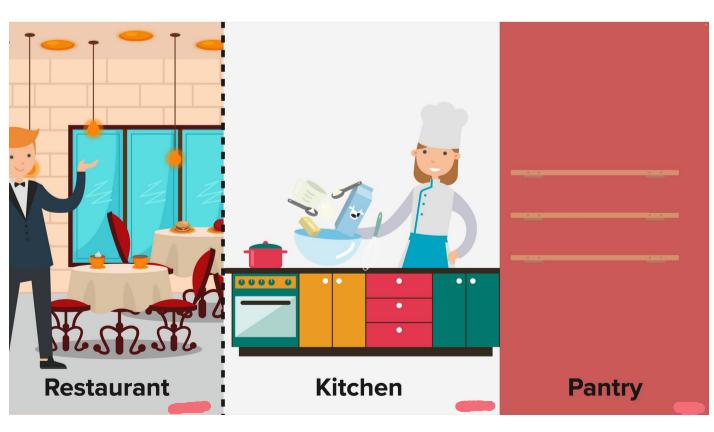
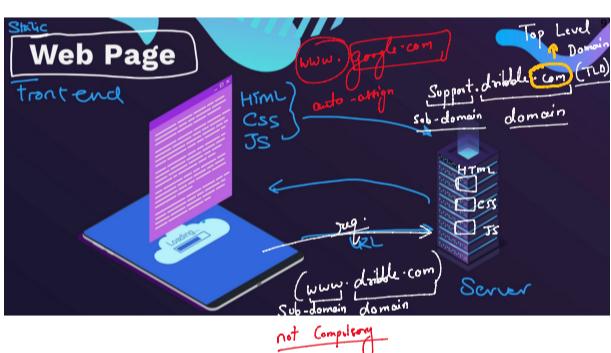
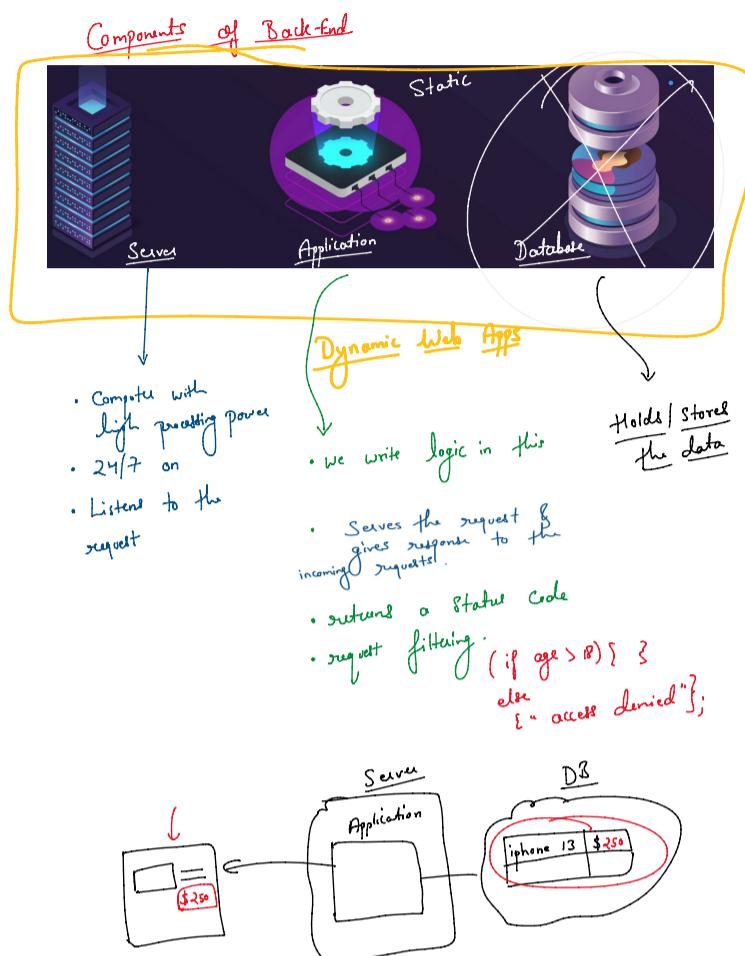
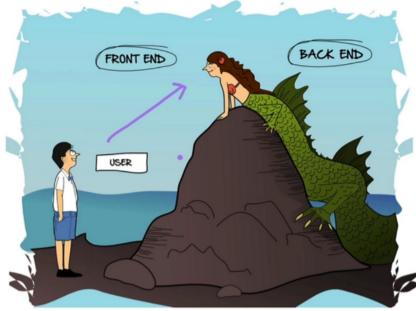


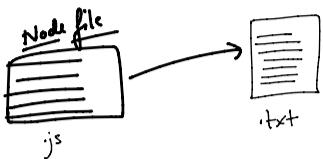
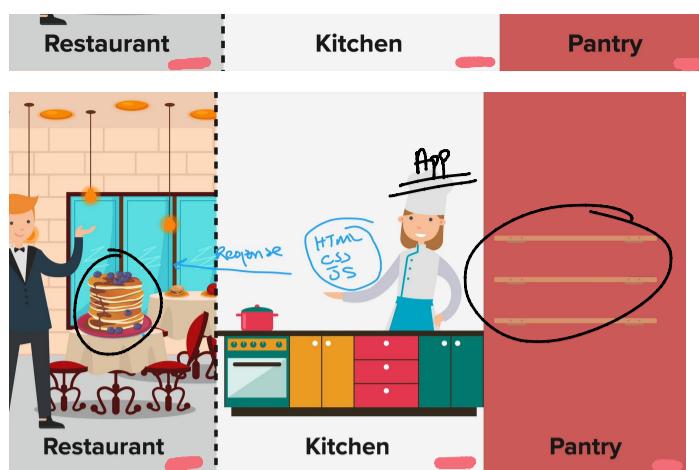
## Selection of HTML elements with JavaScript Code



Shell acts as a UI to communicate with the Kernel.







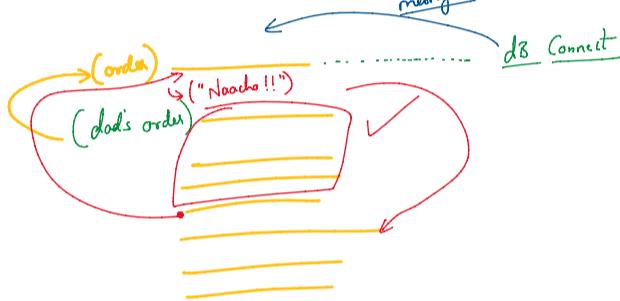
(APIs), request, Debug

Debug:  
An app which can understand the code of a particular lang. & can run it

Used to build scalable Applications.

JS runtime: Asynchronous, event-driven

Synchronous?  
Sequential flow of program (top to bottom)



Non-blocking I/O (Input /output)

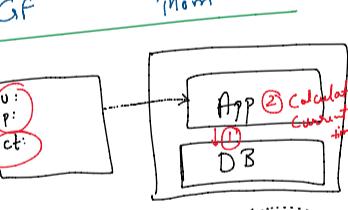
[reading file/network requests, talking to the DB]

ammo: 10

Single Threaded Architecture



Mem



① 3 Secs

② 1 Sec

Get

Status

• TLD - ① Purpose of the website/Web App

② Type of the Web App

.in	- Indian web-site/App
.com	- business (Commercial purpose)
.us	
.gov	- govt website/App
.org	- NGO

Domain Name Sellers

- GoDaddy
- Wix

• Node REPL:

Read

End

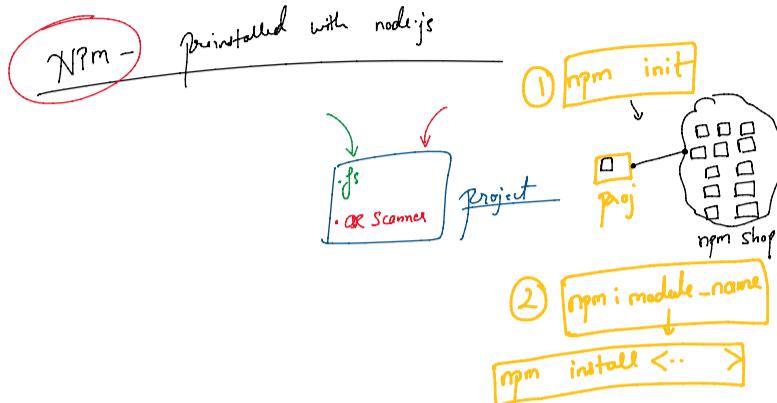
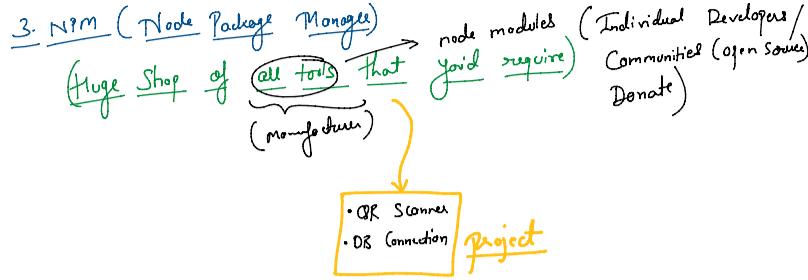
Print

(Type terminal - drag & drop)

2. Node Modules → pre-installed (js)

- a set of folders
- Containing js files
- which have some pre-written code.
- for a specific type of task.
- .. i. write into a text

- Can have some pre-written code.
  - for a specific type of task.
- File System to Access to read/write into a text file.



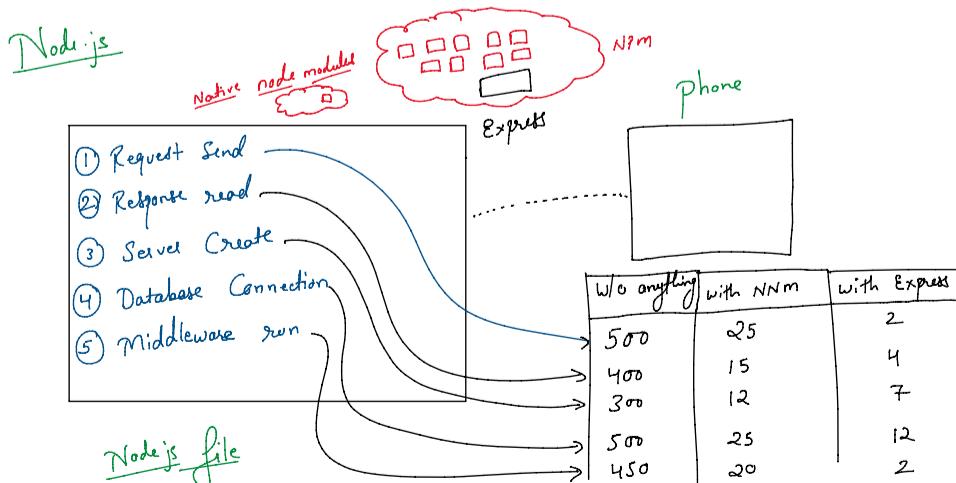
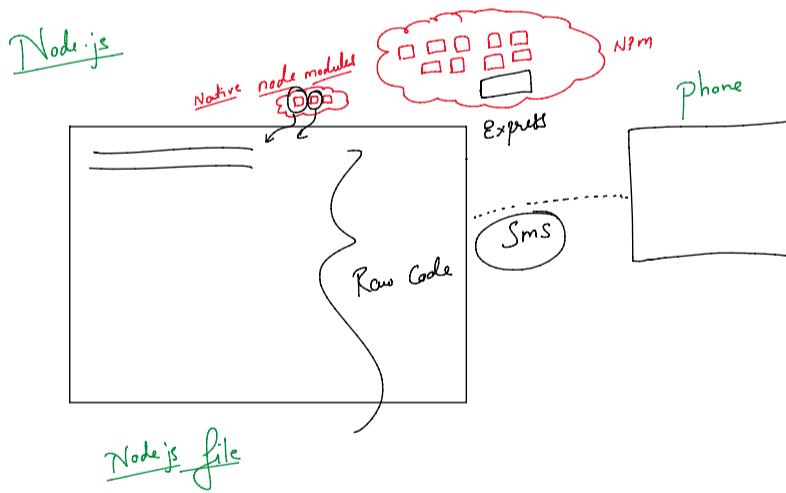
Ways of writing is importing a module

- ① CJS → Common JavaScript [Const express = require('express')]
- ② ESM → Ecma Script Modules [import express from "express"]

```

File Edit Selection View Go ... < > P SillyName Module
index.js
1 // const generateName = require('sillyname');// old way of importing a node module
2
3 import generateName from 'sillyname';
4 let sillyName = generateName();
5
r:\APTECH_Batches\2. MERN Stack Development\BATCH 2\Back-End Development\NPM\SillyName Module\index.js:
SyntaxError: Cannot use import statement outside a module
at internalCompileFunction (node:internal/vm:77:18)
at wrapSafe (node:internal/modules/cjs/loader:1288:20)
at Module._compile (node:internal/modules/cjs/loader:1340:27)
at Module._extensions..js (node:internal/modules/cjs/loader:1435:10)
at Module.load (node:internal/modules/cjs/loader:1323:32)
at Module._load (node:internal/modules/cjs/loader:1023:12)
at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:135:12)
at node:internal/main/run_main_module:28:49
Node.js v20.10.0

```



- Benefits of Express
1. Reduces the lines of code
  2. Discourages repetition of code in Apps.

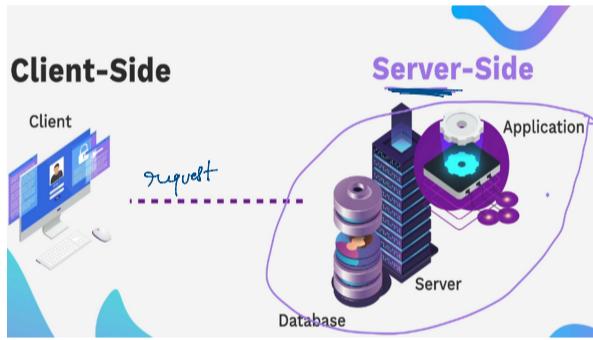
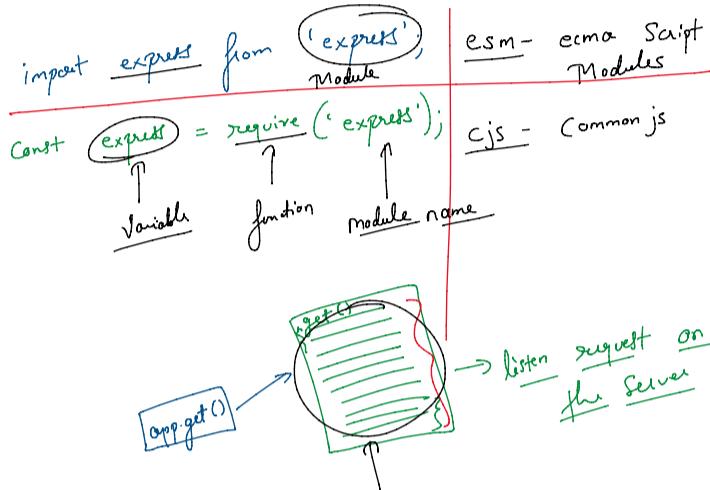
1. Reduces repetition of code
  2. Discourages repetition of code
  3. Eases web Dev creating Web Apps.
  4. Increases speed of creating Web Apps.
  5. It has pre-written code for doing the majority of tasks which any web app has to perform.
  6. Optimized code by running code commonly / regular.
- It is the fastest possible running code

### Node

```
index.js
1. index.js -- Web Development Projects
2. BackEnd > 0.0 m to Express > index.js
3. import http from "http";
4. import url from "url";
5. const server = http.createServer((req, res) => {
6.   if (parsedUrl.pathname === "/" && req.method === "GET") {
7.     res.writeHead(200, { "Content-Type": "text/html" });
8.     res.end(`<h1>Welcome to the homepage!</h1>`);
9.   } else if ("about" === parsedUrl.pathname && req.method === "GET") {
10.     res.writeHead(200, { "Content-Type": "text/html" });
11.     res.end(`<h1>About us</h1>`);
12.   } else {
13.     res.writeHead(404, { "Content-Type": "text/html" });
14.     res.end(`<h1>Page not found</h1>`);
15.   }
16. });
17. server.listen(3000, () => {
18.   console.log("Server running at http://localhost:3000/");
19. });
20.
21.
22.
```

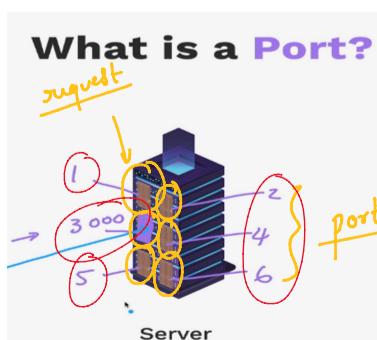
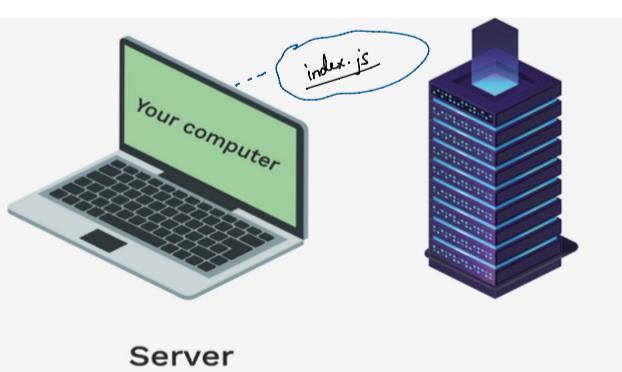
### Node + Express

```
index.js
1. index.js -- Web Development Projects
2. BackEnd > 0.0 m to Express > index.js
3. import express from "express";
4. const app = express();
5. app.get("/", (req, res) => {
6.   res.send("<h1>Welcome to the homepage!</h1>");
7. });
8. app.get("/about", (req, res) => {
9.   res.send("<h1>About us</h1>");
10. });
11. app.use((req, res) => {
12.   res.status(404).send("<h1>Page not found</h1>");
13. });
14. app.listen(3000, () => {
15.   console.log("Server running at http://localhost:3000/");
16. });
17.
18.
19.
20.
21.
22.
```



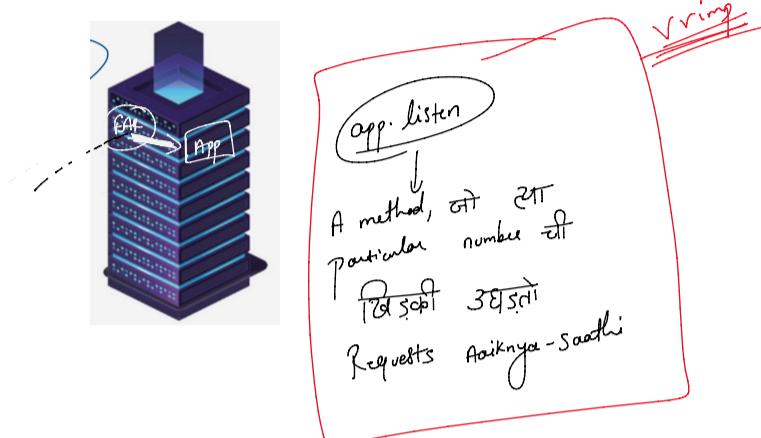
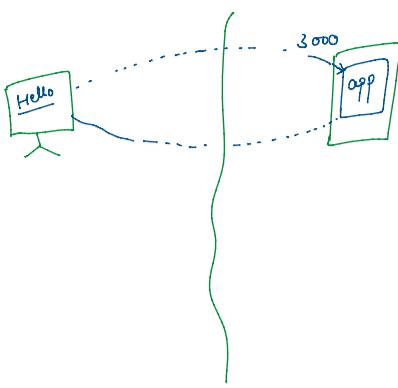
How to Create Express Server:

- ① Create a project dir. — `'express'` '`'expressLearning'`
- ② a `"index.js"` file
- ③ Initialize NPM
- ④ Install the `'express'` module
- ⑤ Change the settings to make project recognize esm way of code.

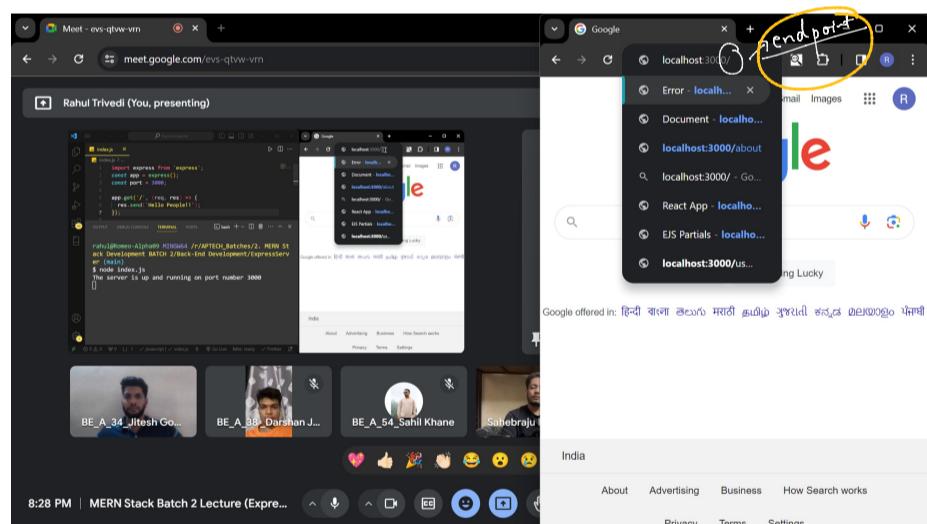


`netstat -an | find "LISTENING"`

netstat -an | find "LISTENING"



- HTTP - Hyper Text Transfer Protocol
  - ↳ GET - Give me something (HTML file + CSS file, text)
  - ↳ POST - Sending you a resource (email/password, PDF file / image)
  - ↳ PUT - replace resource
  - ↳ PATCH - partially update resource
  - ↳ DELETE - delete a resource



A screenshot of a terminal window titled 'ExpressServer'. It shows the following Node.js code in a file named 'index.js':

```
1 import express from 'express';
2 const app = express();
3 const port = 3000;
4
5 app.get('/', (req, res) => {
6   res.send('Hello People!');
7 });
8
9 app.listen(port, () => {
10   console.log(`The server is up and running on port number ${port}`);
11});
```

The terminal output shows the server starting successfully on port 3000. Handwritten annotations include 'Address' pointing to the port number '3000' and a box around the 'Hello People!' response message.

Response Code :

7 100 - 199 : Hold on I'm giving you the resource

- HTTP Status Codes
- 1] 100 - 199 : Hold on I'm giving you the resource
  - 2] 200 - 299 : Here you go (I have done your work successfully)
  - 3] 300 - 399 : Go away [I'm redirecting you somewhere else]
  - 4] 400 - 499 : You messed up
  - 5] 500 - 599 : I messed up

### Middlewares :



① Authentication : check whether the user is authenticated

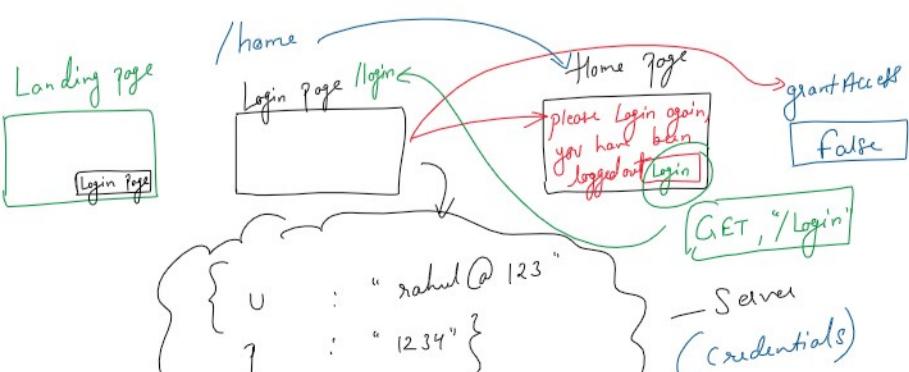
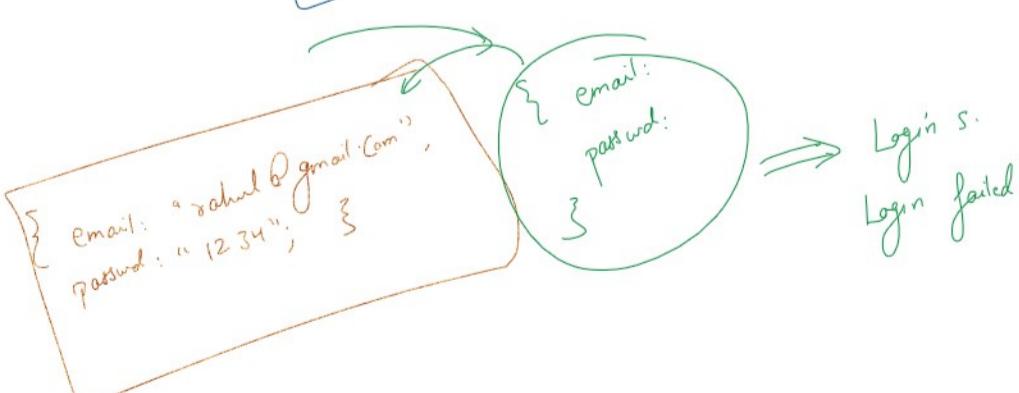
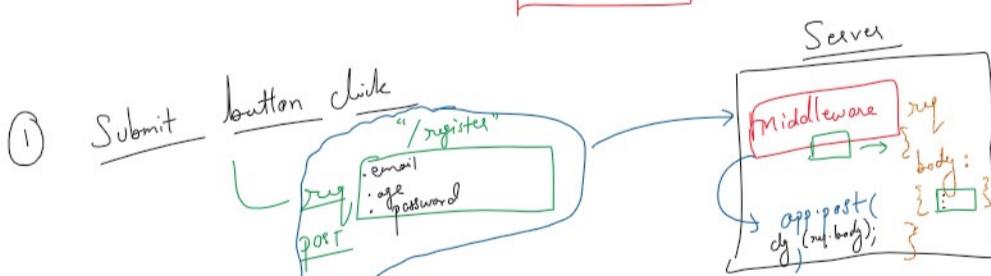
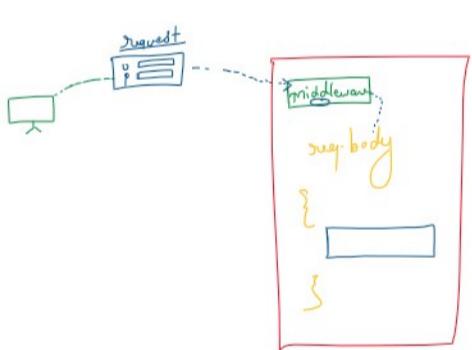
② Check Vulnerabilities :

③ Correct Handler

④ Logs the Content of requests

### Middlewares in npm :

① body-parser :



`{`  
`u : "rahul"`  
`p : "1234"`  
`}`

— Server  
(credentials)

```

if (credentials.p === req.body.password)
{
  grantAccess = true
}
else
{
  grantAccess = false;
}
next()

```

button click

\* Background processes that happen when Data Comes from Backend to Frontend \*

① `res.render("index", {you:})`

EJS uses locals  
Call it only this much

locals:  
{}  
{}  
{}

Assignments:

① **There are 9 letters in your name.**

RAMUL      TRADE      OK

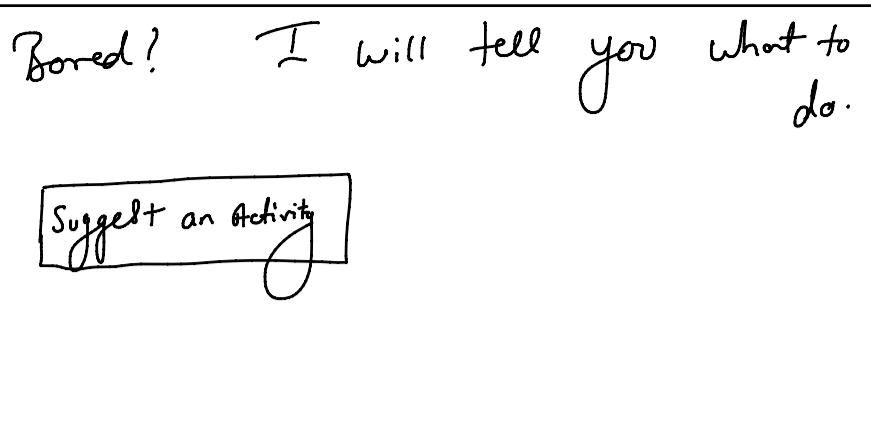
② UI: Wanna know what to do today?

Select a Day	Submit
Monday	Just relax & keep going
Tuesday	Hanuman chalisa padho

Server side:      object / Switch Case:

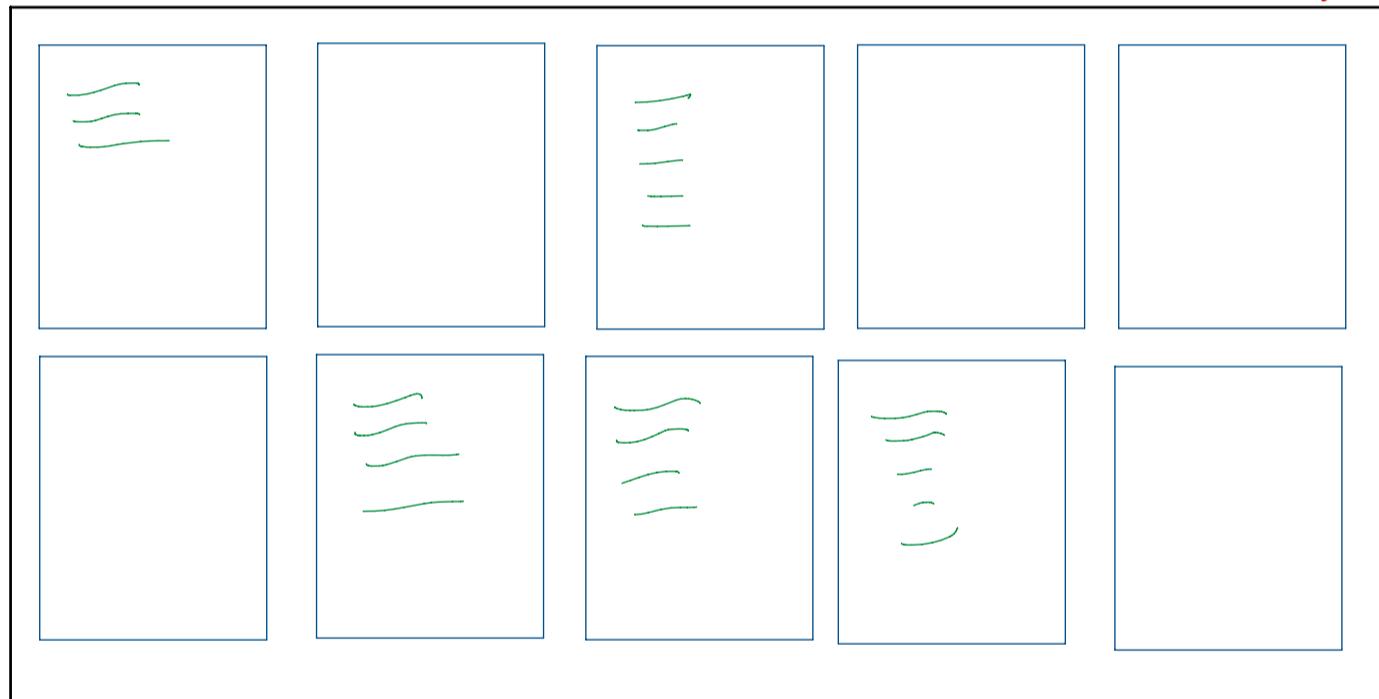
③ Random Activity → Kal

UI



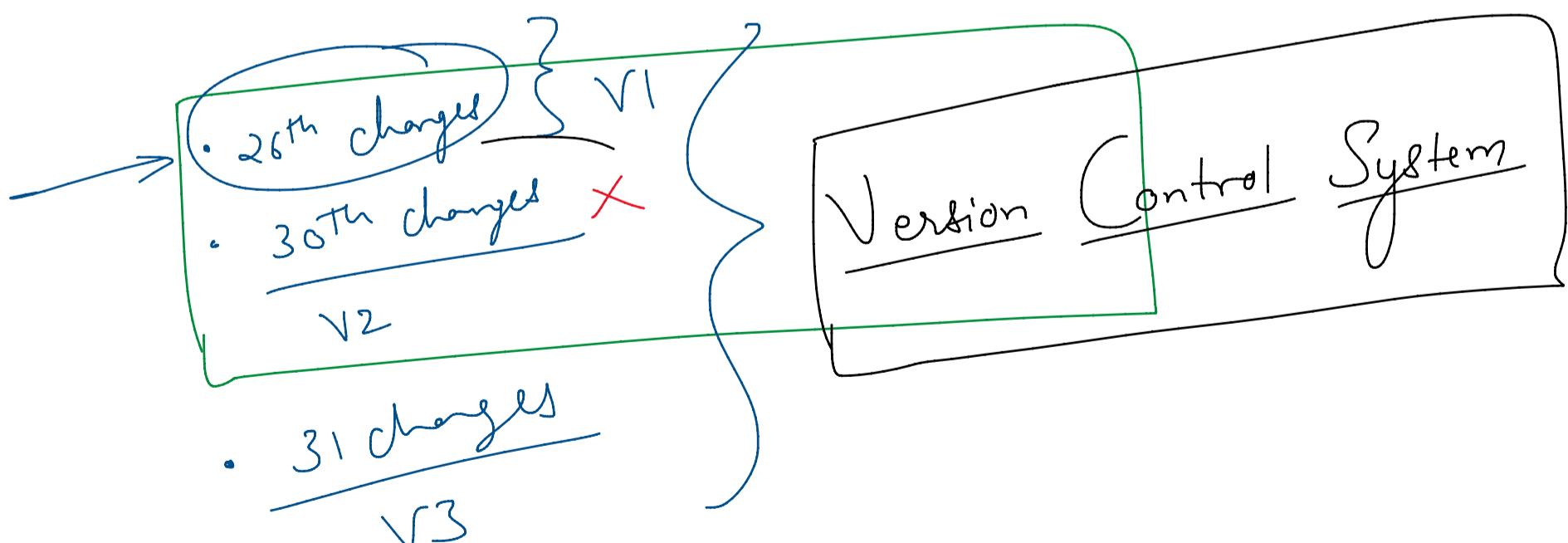
{ activity Title : "play Ludo",  
noOfPeople : 4,  
type : "entertainment"  
,  
{  
}

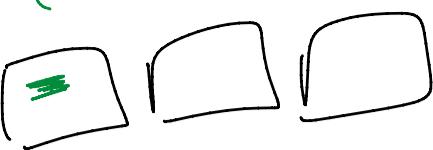
① git init  
② repository



notes down the details of changes made

- file dir - loc
- changed loc
- file name
- prev state
- current state

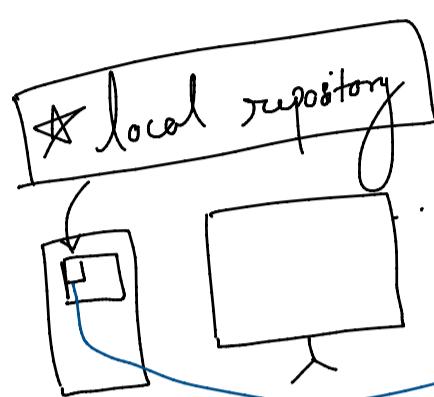




## Working Directory

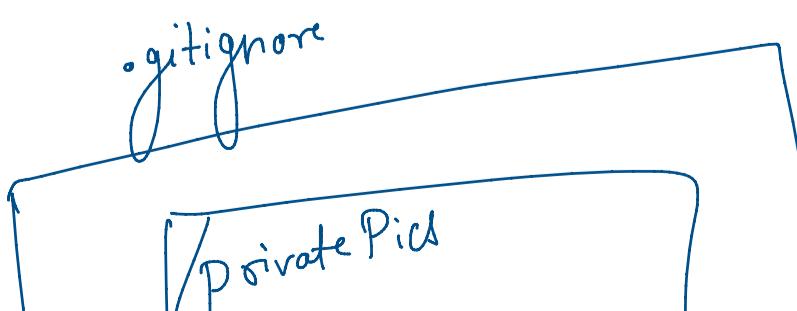
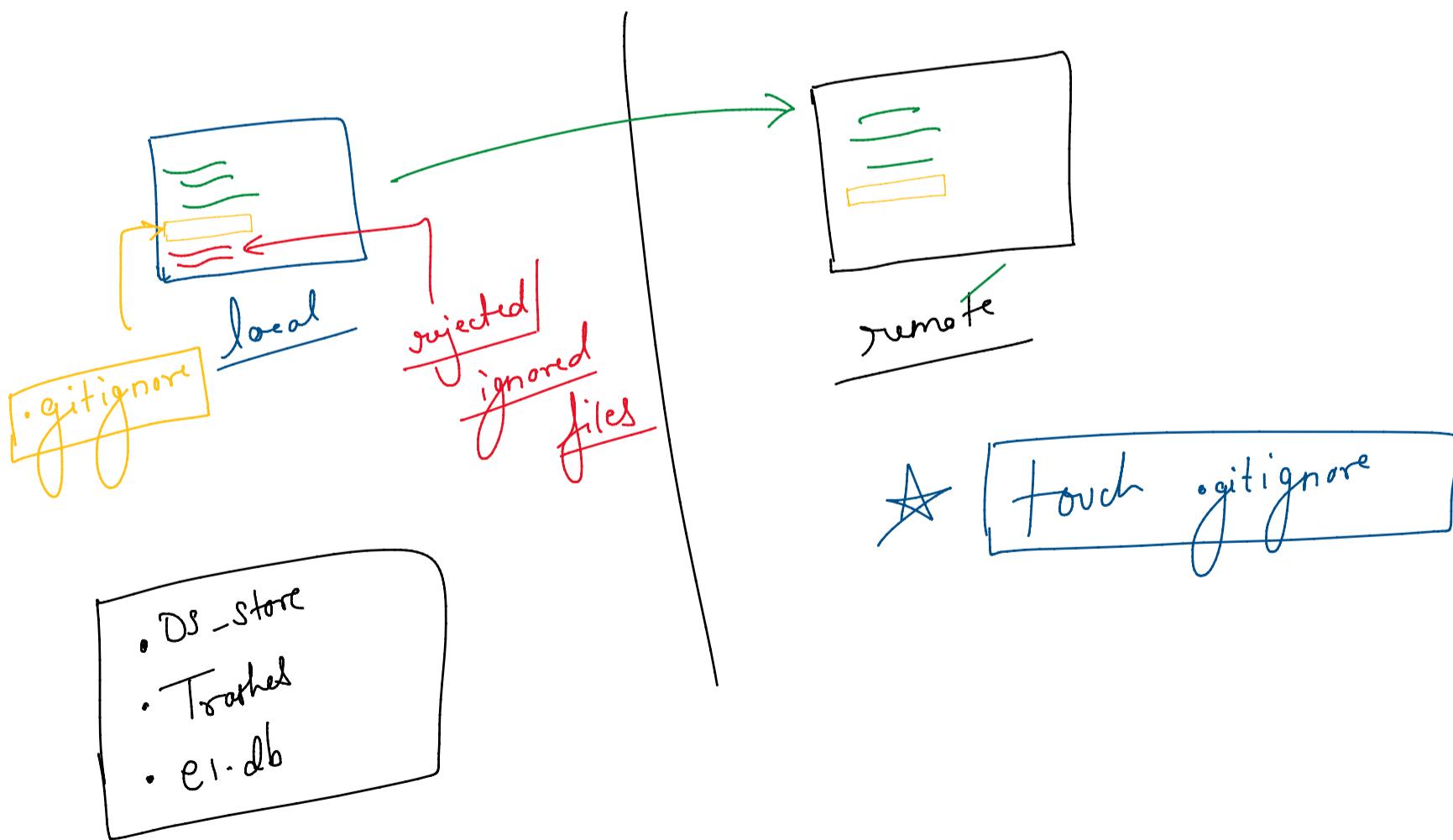
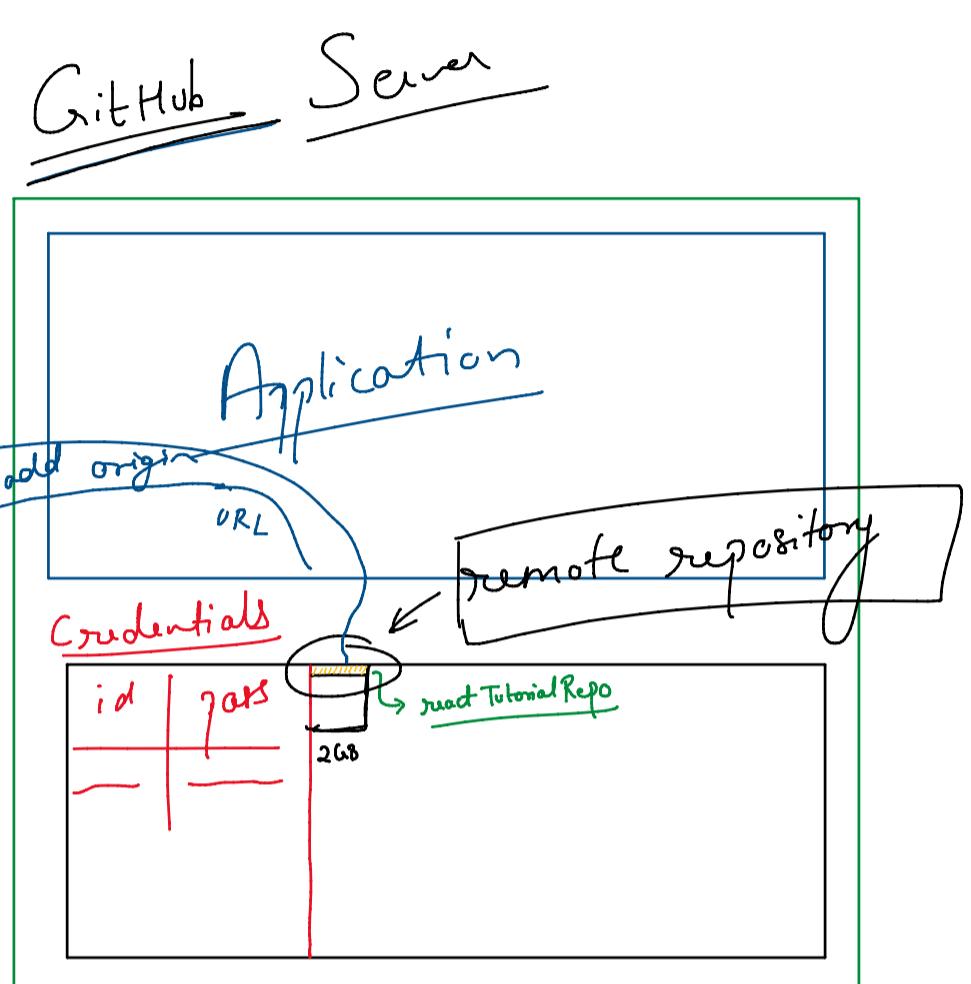


## Staging Area



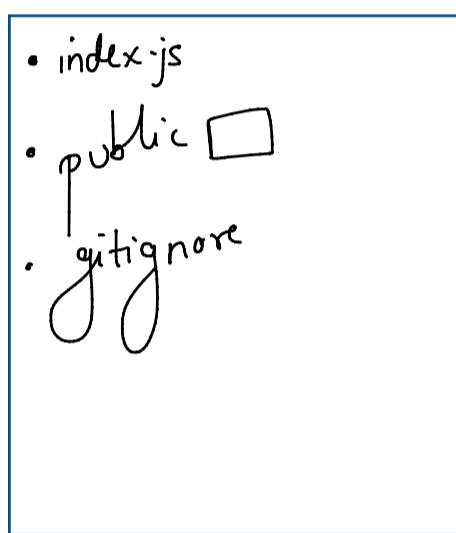
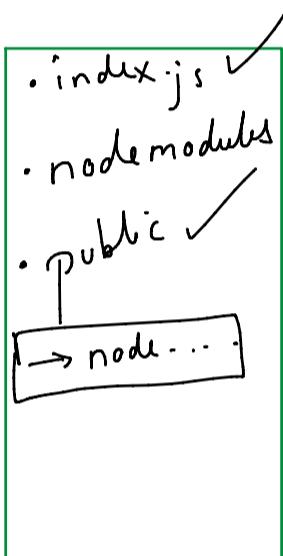
Drive

tunnel  
git remote add origin



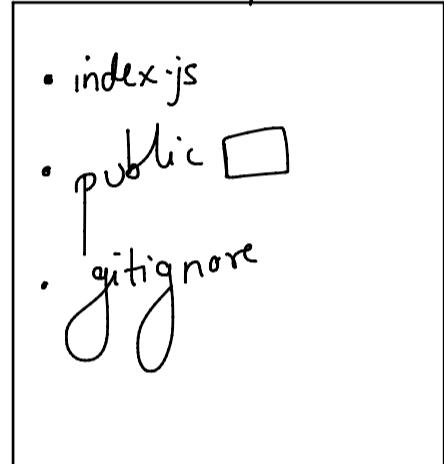
/Private Repo

## Example



cloning

example folder

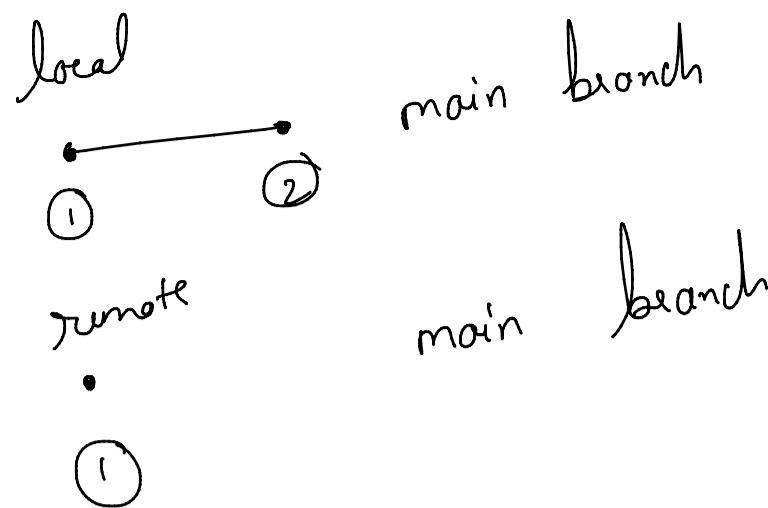


local repo

remote repo

The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows files in the GITLEARNING folder: test1.txt, test2.txt, test3.txt.
- TERMINAL:** Displays the commit history of the test2.txt file:
  - commit 6e0dba5dfa19a88f7cff99d0185c4107b7c3dea2 (HEAD -> main)  
Author: Rahul Trivedi <rahul.t2001@gmail.com>  
Date: Sat Mar 23 21:10:38 2024 +0530 (2)
  - test2 and test3 - both files modified
  - commit 68217026b53666ac712bfeaeec0f562d840eb56 (origin/main)  
Author: Rahul Trivedi <rahul.t2001@gmail.com>  
Date: Wed Mar 20 21:02:54 2024 +0530 (1)
  - made some changes in test2.txt
  - commit c81fdeade0c9ca538754bff7bb96ec9ab2f9d5f8  
:(
- STATUS BAR:** Shows file status (main), line count (Ln 5, Col 15), and encoding (UTF-8).



```

File Edit Selection View Go ... ← → Core Java
src > StreamAPI > StreamPartOne.java > StreamPartOne > main(String[])
6 import java.util.function.Predicate;
7 import java.util.stream.Stream;
8
9 public class StreamPartOne {
10     public static void main(String[] args) {
11         List<Integer> nums = Arrays.asList(3, 9, 6, 1, 5, 10, 12, 54, 27, 33);
12
13         Stream<Integer> s1 = nums.stream();
14
15         Predicate<Integer> pred = number -> {
16             return number % 2 == 0;
17         };
18
19         s1.filter(pred);
20
21         // s1.forEach(number -> System.out.println(number));
22
23         // s1.forEach(number -> System.out.println(number));
24     }
25
26 }

```

S |

3 | 9 | 6 | 1 | 5 | 10 | 12 | 54 | 27 | 33

fn

f

3, 9,

6 | 10 | 12 | 54

T

Ln 16, Col 36 Spaces: 4 UTF-8 CRLF { } java Go Live kite: ready Prettier

```

File Edit Selection View Go ... ← → Core Java
src > StreamAPI > StreamPartOne.java > StreamPartOne > main(String[])
9 public class StreamPartOne {
10     public static void main(String[] args) {
11         List<Integer> nums = Arrays.asList(3, 9, 6, 1, 5, 10, 12, 54, 27, 33);
12
13         Stream<Integer> s1 = nums.stream();
14
15         Predicate<Integer> pred = number -> {
16             return number % 2 == 0;
17         };
18
19         Stream<Integer> s2 = s1.filter(pred);
20
21         s2.map(number -> number * 2);
22
23         // s1.forEach(number -> System.out.println(number));
24
25         // s1.forEach(number -> System.out.println(number));
26     }
27
28 }

```

S<sub>2</sub>

6 | 10 | 12 | 54

number

doubling

12, 20, 24, 108

Ln 21, Col 38 Spaces: 4 UTF-8 CRLF { } java Go Live kite: ready Prettier

File Edit Selection View Go ... ← → Core Java

StreamPartOne.java 1 ×

```
src > StreamAPI > StreamPartOne.java > StreamPartOne > main(String[])
 9  public class StreamPartOne {
10      public static void main(String[] args) {
11          s3.reduce(0, (c, e) -> {
12              System.out.println("c : " + c);
13              System.out.println("e : " + e);
14              return c + e;
15      });
16  }
```

12, 20, 24, 108 S2

C : 0 , e : 12 add: 12

C : 12 , e : 20 add: 32

C : 32 , e : 24 add: 56

C : 56 , e : 108 add: 164

Run: StreamPartOne

OUTPUT DEBUG CONSOLE TERMINAL PORTS COPILOT VOICE

```
$ /usr/bin/env C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe -XX:+ShowCodeDetailsInExceptionMessages -cp R:\0.\Vishaka's\Java Course\Batch\Core Java\bin StreamAPI.StreamPartOne
c : 0
e : 12
c : 12
e : 20
c : 32
e : 24
c : 56
e : 108
```

rahul@Romeo-Alpha09 MINGW64 /r/0. Vishaka's Java Course Batch/Core Java (main)
\$

Ln 25, Col 44 Spaces: 4 UTF-8 CRLF {} java Go Live kite: ready Prettier

*Diagram:* A hand-drawn diagram showing a rectangular box labeled "Integer" with an arrow pointing from it to the word "Value". Above the box, the word "returned" is written.