"Full-stack web-development at a POPU glance"

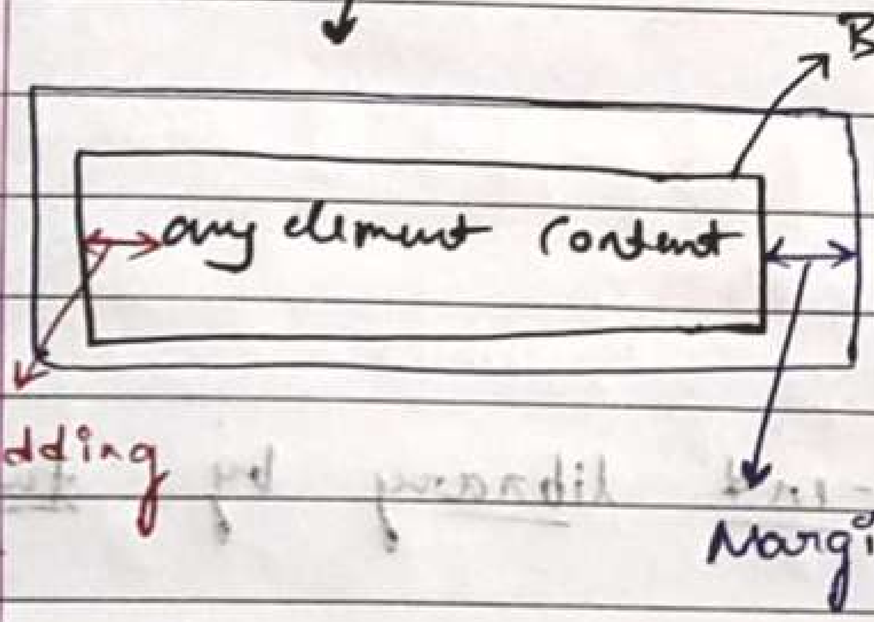
Page No.
Date: / /

(R) https://www.....

for favicon → go to favicon.cc website [create your own]

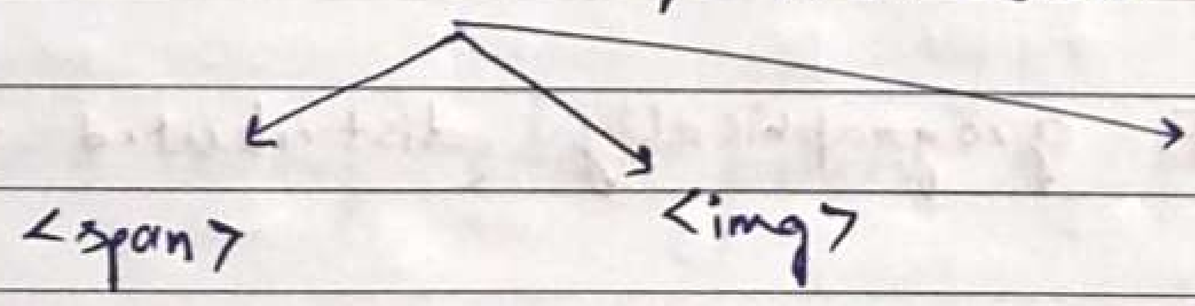
→ For colors → colorhunt.io

→ CSS Box Model → redmelon.net

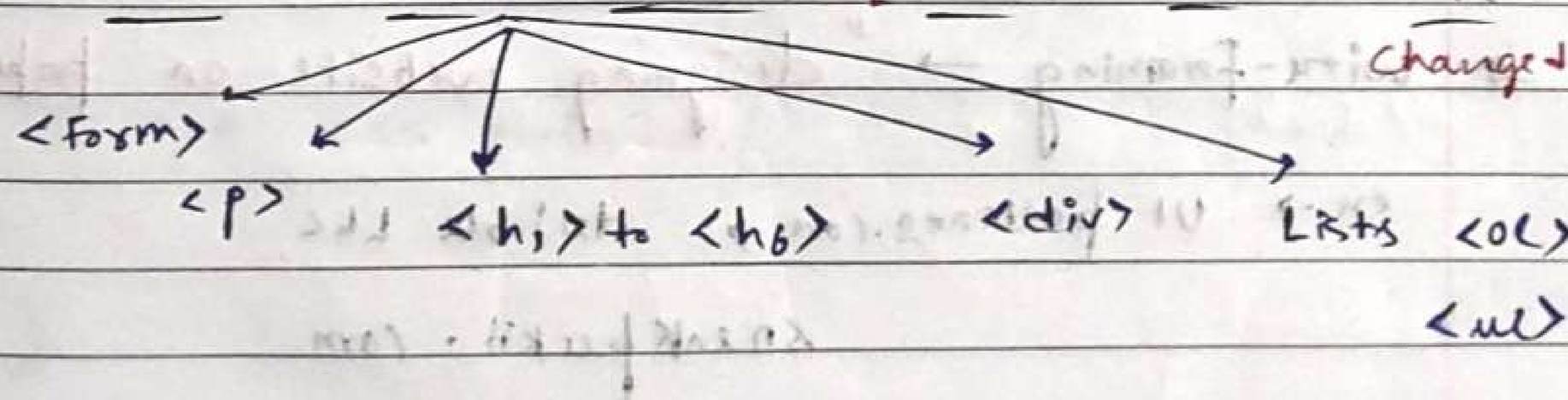


→ Common - Inline elements [those who lie inside block elements]

" we can't change their width



→ Common - block elements [whose width can be change]



* to make a block-element treated as inline

set in css display : inline block ;

for font-style → "Google fonts"

16 px = 1 em = 100% [font-size]

1 rem

↓

[root]

→ Bootstrap → front-end library by twitter

CdN → "Content delivery Network

↓

group of geographically distributed servers

that speed-up the delivery of web content"

wire-framing → "designing website on paper"

ex → UI patterns.com, dribble LLC

sneakpeekit.com

→ " css - z-index and stacking order "
↗ only works if elements are positioned

" To change the stacking order of elements "

if I do margin : 10px 20px 30px

top | lyt-Right | bottom

margin : 10px 20px 20px 30px

top | Right | bottom | lyt

margin : 10px 30px

top-bottom | lyt and right

media Query break points → for mobile Friendly ness

example →

@media (max-width : 800px) { ↗ for-devices below and equal to 800px

/* css rules */

}

POPU

Page No.
Date: / /

Web design

"Understanding Color theory"

Red → love, anger [car websites]

Yellow → joy, intellect, Attention

Green → freshness, Growth, Health

Blue → stability, trust [paypal]

purple → Feminie, Royalty, wrath

→ Combining → Color wheels by Adobe or Colorhunt.co

⇒ Flex-box → flexible-layout css model

```
<div class="container">
    <div class="box"></div>
    <div class="box"></div>
</div>
```

POPU

Page No.
Date :

- container {

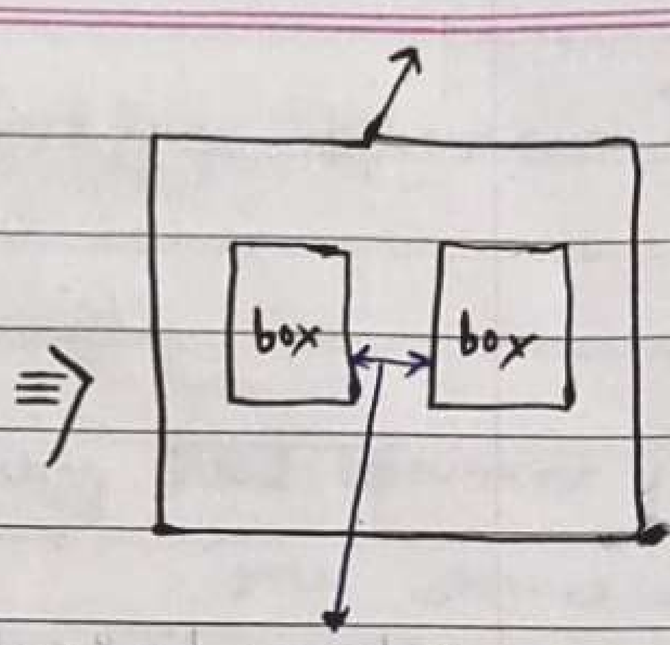
display : flex;

flex-direction : row;

gap : 30 px;

align-items : center;

}



or

- container {

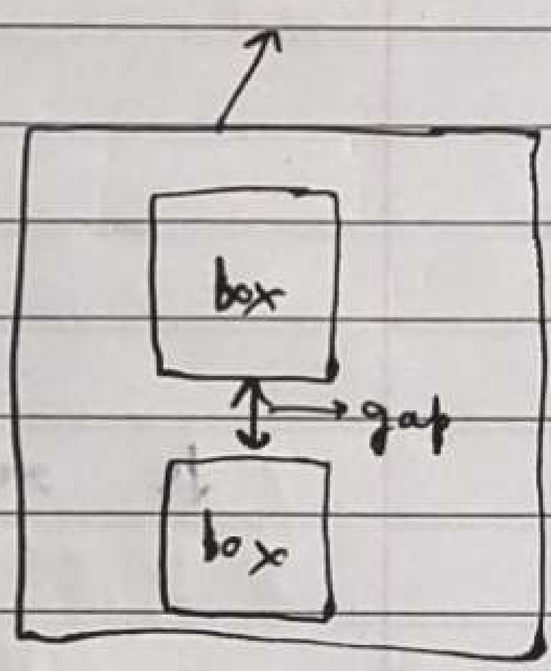
display : flex;

flex-direction : column;

justify-content : flex-end;

gap : 30px

}



by "Brendan Eich"

Page No.
Date: / /

JavaScript (1995)

to make web-page alive

chrono logically,

like Script → java Script → ECMA Script
(initially)
[when java was popular] [when became fully Independent language]

Js run on Any device with Js engine

V8 chrome, edge, Opera | spidermonkey | Nitro, Safari

"Limitation of Browser running Js"

POPU
Page No.
Date: / /

" 98% website use ds For webpage bthaviours "

alert (" hello ");

document object Model [representing html document in tree structure]

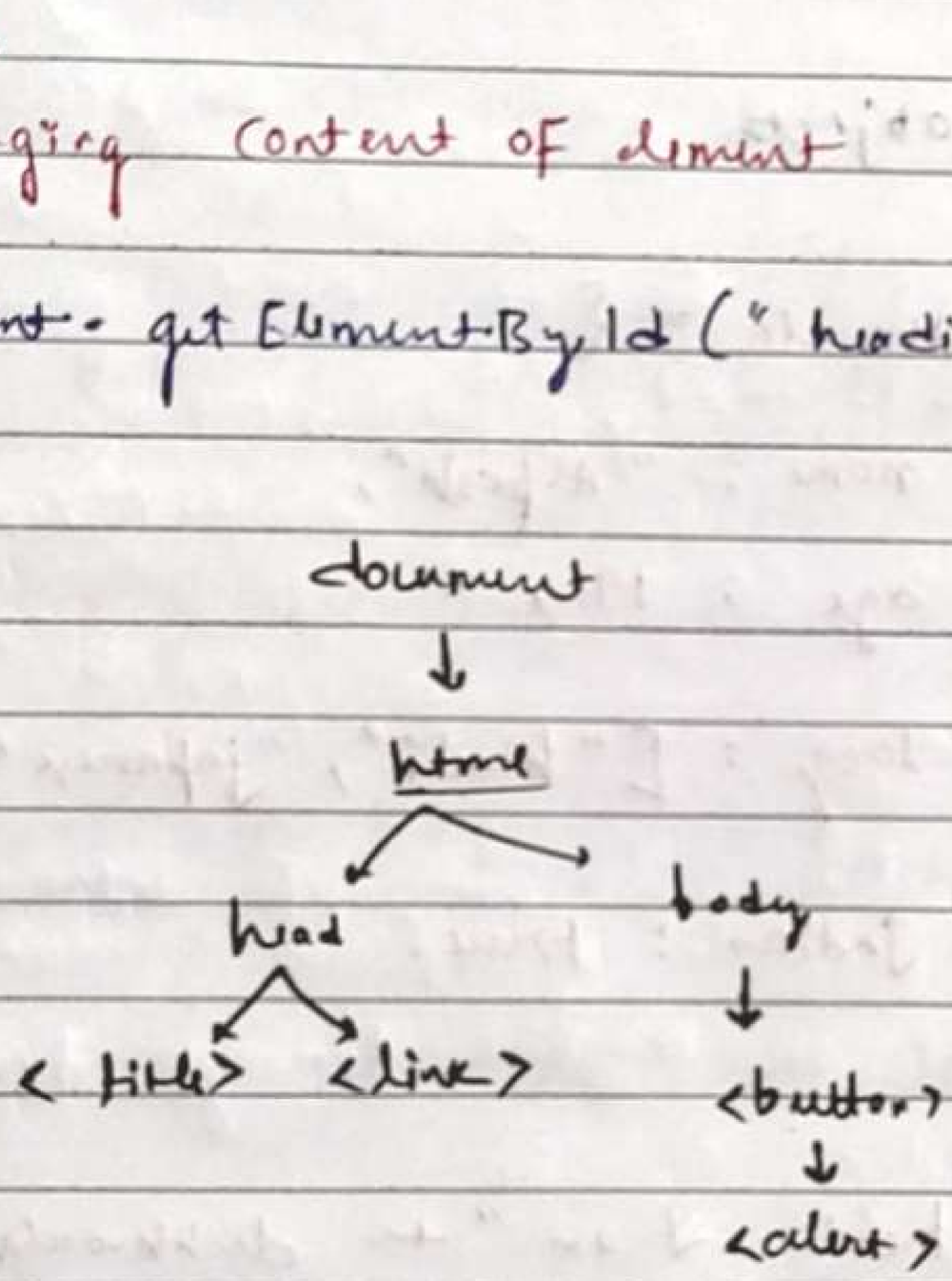
↗

DOM - Manipulation

↓

changing content of elment

document. getElementById ("heading"). innerHTML = " Hi ";



⇒ Adding Event listeners

document. getElementby Id (" button "). add Event Listener ("click", handle click);

function handle click () {

// code to handle that event

}

→ Playing Audio

```
Var audio = new Audio ("sounds/raju.mp3");

audio.play();
```

Javascript objects

```
var doodhwala = {

    name : "Alfresh",

    age : 19,

    lang : ["hindi", "japanese"],

    Indian : true,

}

alert ("hello I am " + doodhwala.name);
```

- - - ~ x ~ ~ ~ ~ ~

jquery → library of js to ease js code

Manipulating styles

```
$("h1").css ("color", "red");

$("h1").addClass ("raju");

$('h1').removeClass ("  ");
```

POPU

```
$("h1").hasClass("reign");
```

✓

return true or false

text - Manipulation

```
$("button").text("Submit");
```

or

```
$("button").html("Send");
```

Manipulating attributes

```
$("a").attr("href", "https://myntra.com");
```

Adding event listeners with jQuery

```
$("button").click(function(){
    alert("Ye Kya kiya");
});
```

Adding and Removing elements

```
$("h1").before("<button> Send </button>");
```

" web Aimations

```
append();
prepend();
.remove();
```

```
$("h1").animate({opacity:0.5});
```

only CSS attributes with Numerical value

```
$(document).ready(function(){
    alert("Yes"); });
```

Command - line : way of interacting with machine trough text / terminal

① Bash : Bourne Again shell

↓

Command line Interpreter for UNIX based systems

" Back-end - web - development "

⇓

" SERVER SIDE OF ANY WEB Application "

" runtime environment of Javascript "

↗ (browser ds engine into C++ code)

Node.js [ebay, uber, twitter, Netflix]

↓

To write code that interact directly with computer hardware

Ex: " to create desktop Application "

→ " the heavy code execution is on Server side Not just on browser "

Node → Non-blocking Input/output Model / platform

Node REPL

↓

read-evaluate print loop [to execute our code in byte-size chunks

Command : node to exit press : " Ctrl + D "

npm → [node package manager]

$ npm init

=

node install --save superhero

module name

index.js

POPU

Page No.
Date: / /

var superhero = require ("padman");

console.Log (superhero.random());

→ Express Js

framework to easily write server side

Code exceptionally for Node environment

is se hm server side application bana

sKte h

// how to creat your own server

$ mKdir my-app

$ cd my-app

$ touch server.js → write some code

$ node init

$ npm install express

$ node server.js

POPU

Page No.
Date: / /

Nodemon → node package that Automatically restarts server on every update

Server - App code

```
const body-parser = require ("body-parser");

app.use ( body-parser.urlencoded ("{ extended : true "));


app.listen ( 3000 , function(){

                // code });
```

port Number

where application will start

or

process.env.PORT

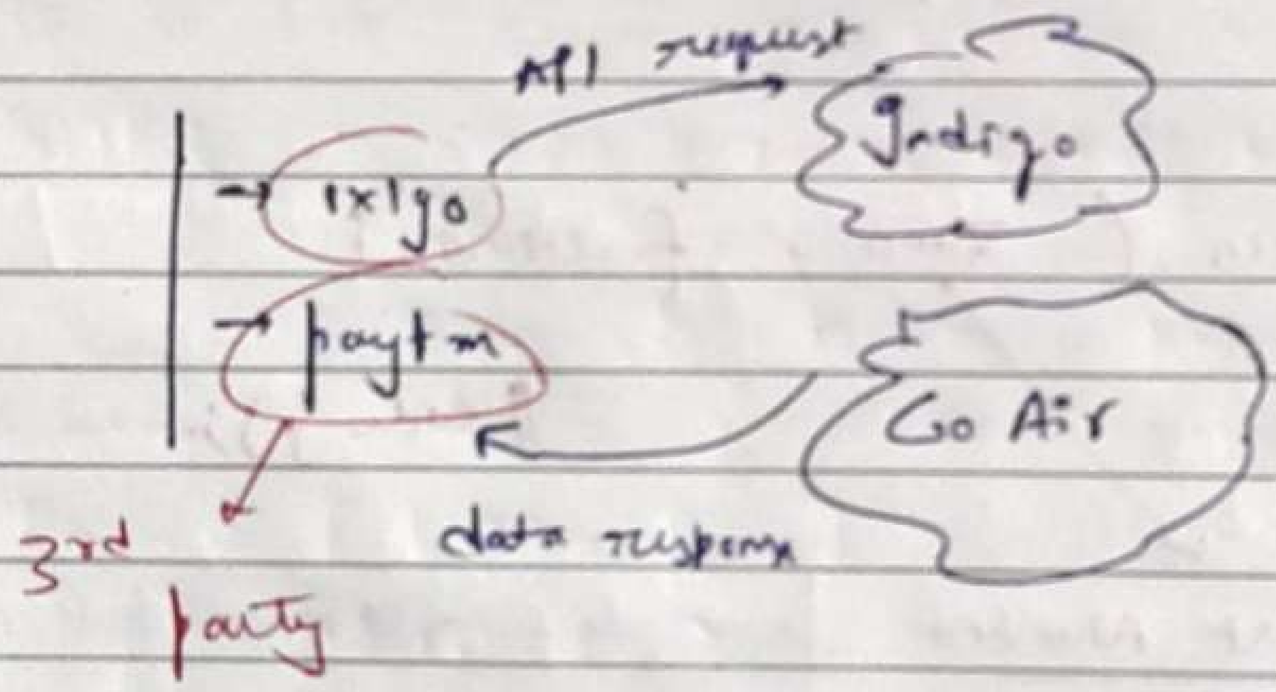
POPU

Page No.
Date: / /

API : Application programming Interface

set of tools and functions to expose the data of a system to the requested client

"flight ticket booking"



ex→ weather API, crime data API

API end point : https://jokes.com/joke

API path : ...../joke/coder

API parameter : ...../joke/programming?

contains = debugging

POPU

API Authentication and postman

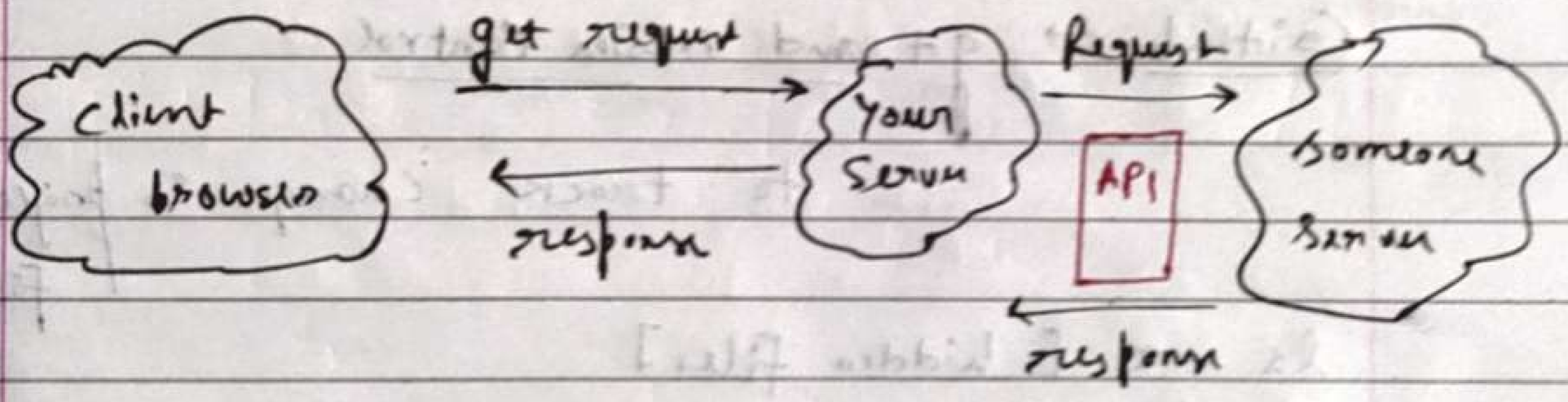
Weather API → the data we get in JSON format

XML or html Format

JSON → javaScript Object Notation

↓

For little space and human readable response



→ making request to API using Node HTTPS Module

```
Const  https = require ("https");

app.get ( function(){
        https.get ("link", function (response){
                console.log (response);
        });
});
```

POPU
Page No.
Date: / /

"status - codes"

100 - 199 [Informational response]

200 - 299 [Successful]

300 - 399 [Redirect]

400 - 499 [Client Side errors]

500 - 599 [Server Side error]

Github → git and version control

to track changes in project files

ls -a [hidden files]

git states

git init // initialise empty git repo

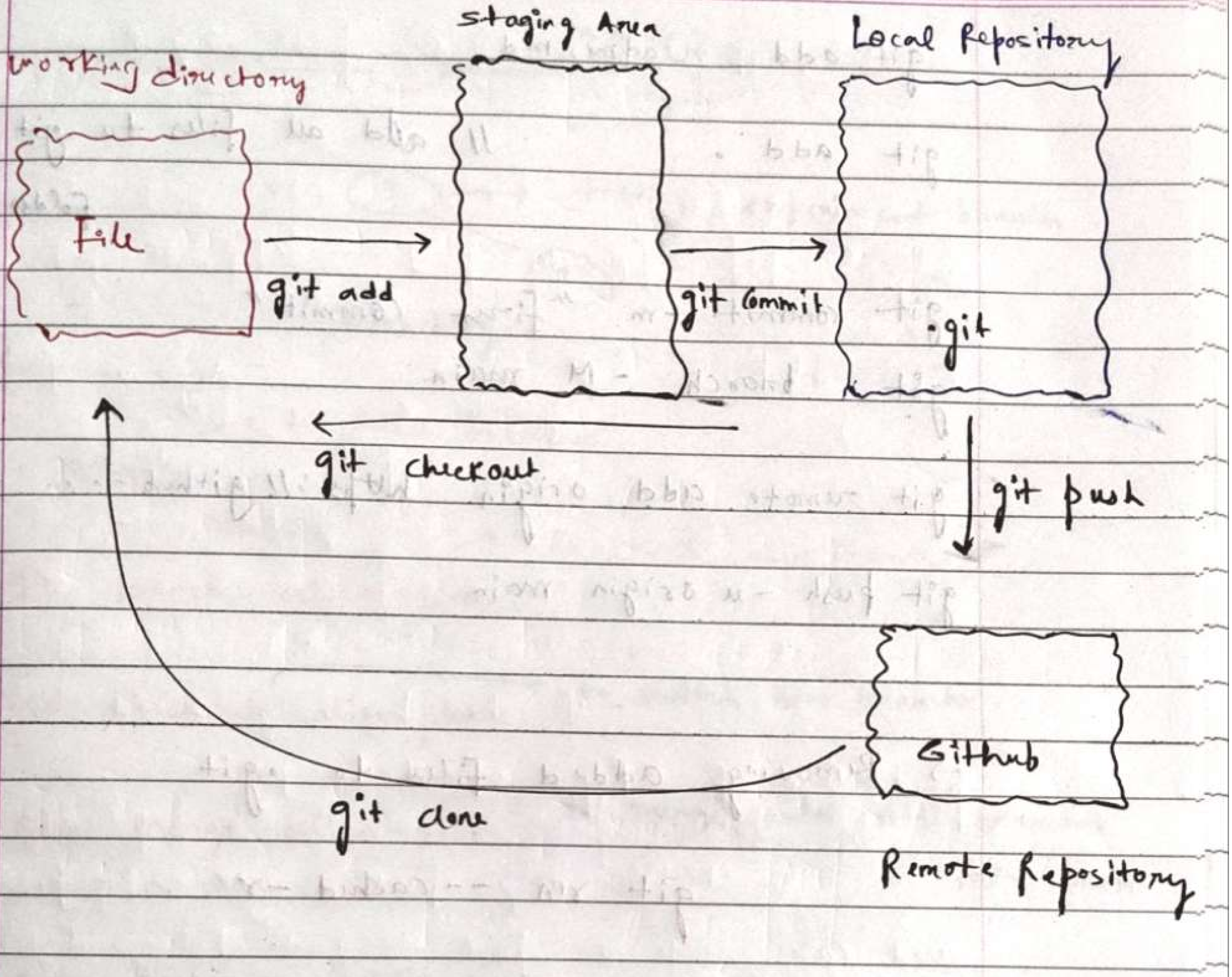
git add chapter1.txt

↑ Flag

git commit -m " complete chapter 1"

git log

helps to split a large change into multiple commits



→ if I made changes to chapter1.txt

I can see them

git diff chapter1.txt

to restore original file

git checkout chapter1.txt

echo "# repo-name" >> ReadMe.md

git init

POPU
Page No.
Date: / /

git add readme.md.

git add . // add all files to .git folder

git commit -m "first commit"

git branch -M main

git remote add origin https://github.... .git

git push -u origin main

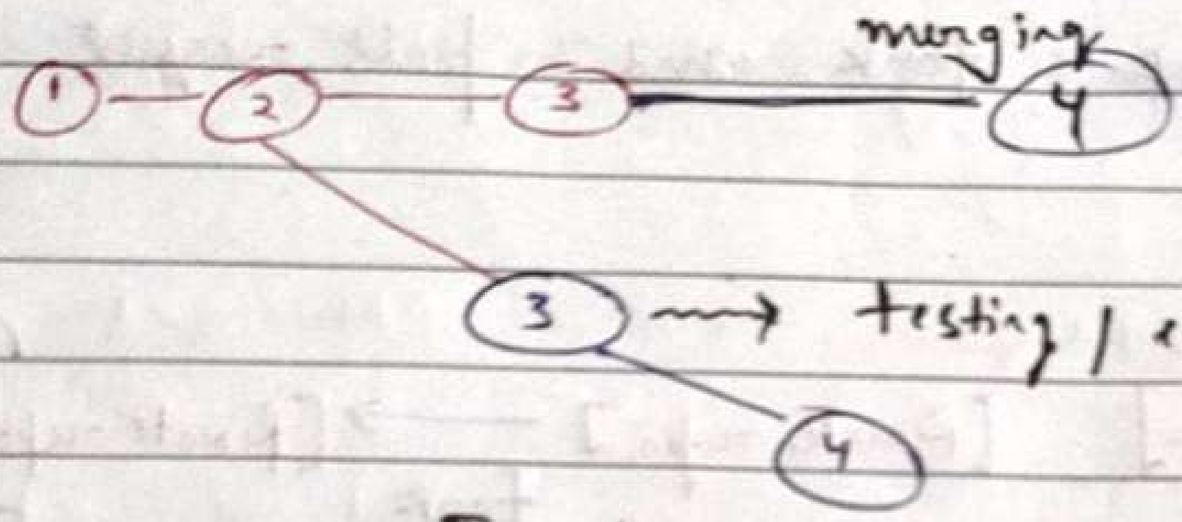
→ Removing added files to .git

git rm --cached -r

using .gitignore → All keys, password, aws keys

git clone url → downloading a repo into local storage

⇒ Branching and merging



Branching

git branch → to check branches

git branch alien-plot → to create new branch

git checkout alien-plot → to switch b/w branches

git merge alien-plot → to merge both / All branches into Main

git push -u origin master

pushing everything into master branch of repository

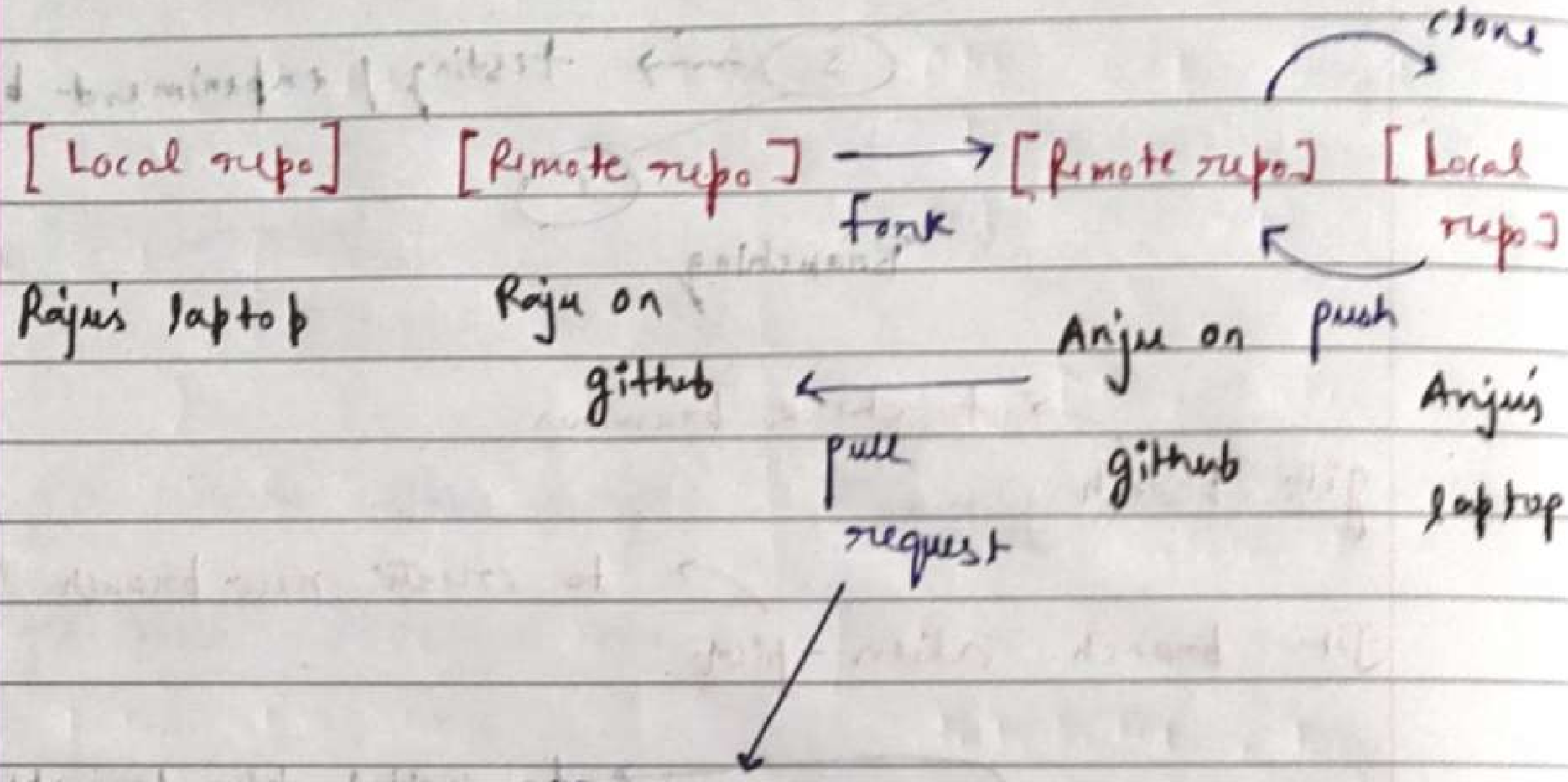
* You can see graph at github → repository

Insights → network

POPU

Page No.
Date: / /

working of fork and pull request



if she has only read privellege of Raju remote repo

she can only fork, work on rep. and can made a pull request else if she has Read and write. she could have made push request

Now After pull req. if Raju trusts Anju [as she is his heart] then

↓

he can approve pull request

POPU
Page No.
Date: / /

EJS → embedded javaScript templata

↓

if we want to make changes only in one html file

↓

<h1> hello, today is <%= Day %> </h1>

EJS

to write pure javaScript code

<% ..... %>

to include pages

<% — include ("header") — %>

⇒ MVC [model - view (controller) framework

↓

Architectural / design pattern

Support test driven development (TDD)

POPU

Page No.
Date: / /

Database → organised collection of structured information

SQL based	or	No SQL based
		↓
		Not only SQL
→ Old firms		mongo DB, redis
→ relational		→ Non-relational
→ table structure		distributed, document structure
→ requires a schema		Scalable, flexible, Non-flexible to changes
→ great with relation-ships		Not great
→ scales vertically		scales horizontally distributed

⇒ SQL database

```
Creat Table products (
    id INT NOT NULL,
    name STRING,
    price MONEY,
    PRIMARY key (id)
)
```

POPU
Page No.
Date: / /

insert into products
values (1, "Pen", 1.20)

Read operations

SELECT * FROM products;
↓
name, price
] for columns

SELECT * FROM products where id = 1
↓
capitalise
] For reading a row

update

update products
set price = 0.80
where id = 2

delete

delete FROM products
where id = 2

Adding column

Alter table products
Add stock INT

POPU

Page No.
Date : / /

relationships in SQL

create TABLE Custom (

....

);

select order.order_id, products.name

from orders

INNER JOIN products ON orders.productid
= products.id

Mongo DB →

to save and exit Vim

press esc + :wq !

install it in C: program files

open MongoDB shell through command

mongosh

Command line .

test > show dbs

> use shopdb

create operation

use shopdb

db.products.insertOne ({_id : 1 , name : "Pen" , price : 20})

db.products.find ( ) → query, projection

↓

OR

(

{name : "Pencil"}] query criteria

)

(

{name : 1 , address : 1}] projection

)

will show only name and address

POPU

Page No.
Date: / /

update

db. products . updateOne {{-id:1}, {$set:{stock : 32}}})

delete

db. products . deleteOne ({-id : 2})

Relationships

Working with . Native MongoDB driver (Node.js)

use Mongoose [ODM]

Node.js

object document Mapper

→ mkdir : fruitProject
touch app.js

npm init (-y) → yes to Everything

POPU

Page No.
Date: / /

Moongose → ODM (object document model)
Mapping

const moongoose = require ('moongosi');

moongoose.connect ("mongodb: 0.0.0.0: 27017 / fruitsB",
{ useNewURLParser : true });

const itemSchema = mongoose.schema ({
name: string
});

schema

const ITEM = mongoose.model ("item", itemschema);

const item1 = ITEM ({
name: Rajesh
});

ITEM. insertMany ([Array of objects]). then (function
(res) { })
. catch (function (err) {
console. Log (err);
});

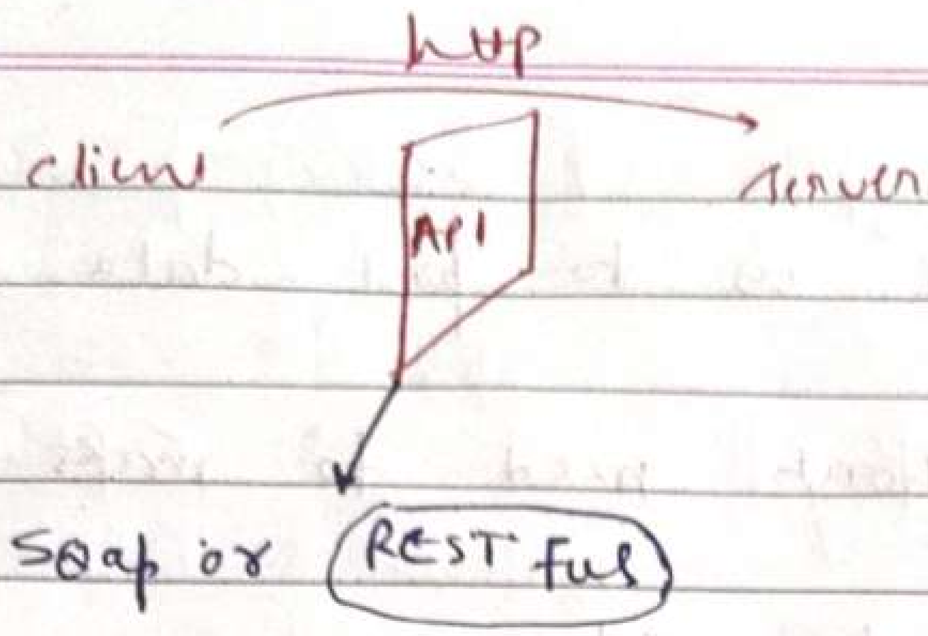
mongoose → findByIdAnd Remove (item id). then ().
. catch ();

→ Mongoose findOne And Update()

```
fruit.findOneAndUpdate({ conditions },
{ updates },
function (err, results){ }
);
```

```
{ $pull : { field : { _id : value } } }
```

REST full API



GraphQL or FALCOR

http verbs → get, post, put, patch, delete

Put vs patch

↳ updating only a portion needed to be updated

updating by replacing entire object

→ use GUI (studio 3T) of MongoDB

→ body-parser deprecated!

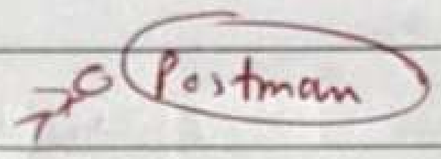
then do, app.use (express.urlencoded (

{ extended : true });

app.use (express.json())

var postdata = req.body || JSON.parse (req.body).

→ postman helps us to post data to server /database without need to make html form at client side



→ patch request lets you modify the existing document

→ put request → let you replace entire object with new one

get → to get the route of express

post → to post the document

delete → to delete the document.

POPU

" Authentication and Security "

to uniquely identify user on our database | Restrict Access

level-1> username and password [just plain text stored]

level-2>

npm : mongoose.encrypt

Const encrypt = require ("mongoose.encryption")

userSchema.plugin (encrypt: { secret: process.env.SECRET,

encryptedFields: ["password"]});

this thing stored in .env file

environment variable

to use this

at the top of app.js

require ("dotenv").config();

use gitignore to hide .env

POPU

Page No.
Date: / /

level-3) encryption [hashing password]

password + key —Cipher method→ Ciphertext

hashing

pwd —hashing→ key

key ←— can't go back

* 20 billion hash keys can be checked / second by most advanced hackers

for this require (md5 module)

and at

```
const newUser = USER({
    email: ....
    password: md5(req.body.password)
});
```

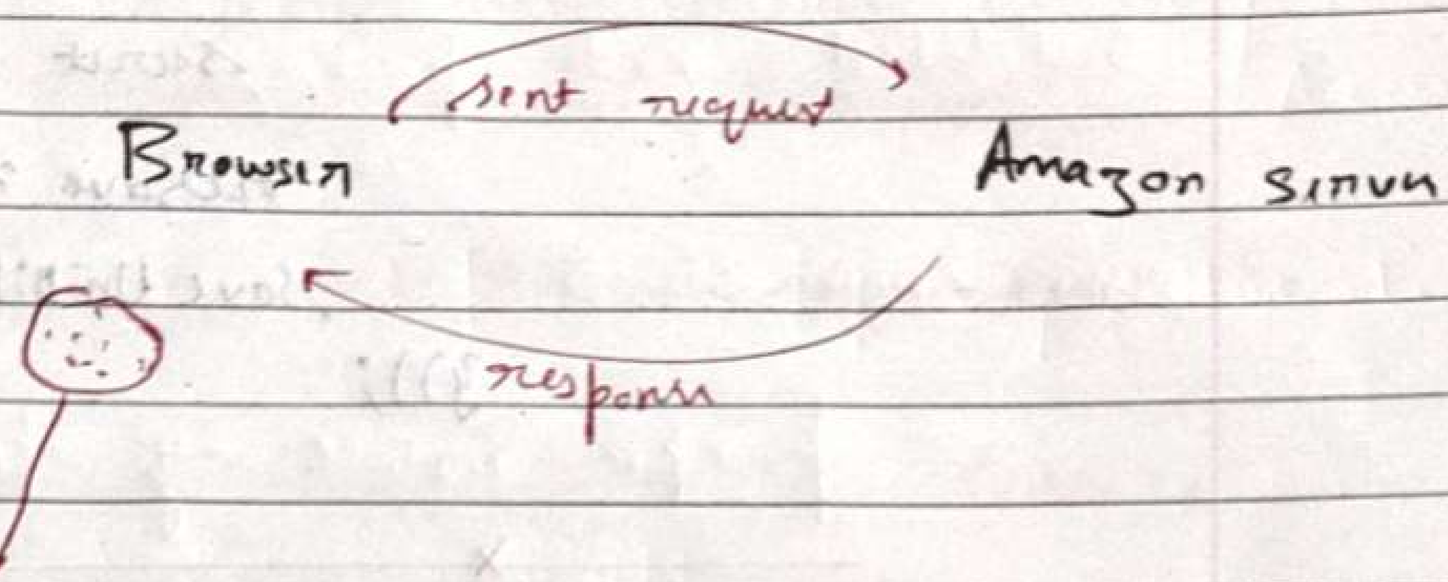
level-4) "hashing and salting"

use "bcrypt" module

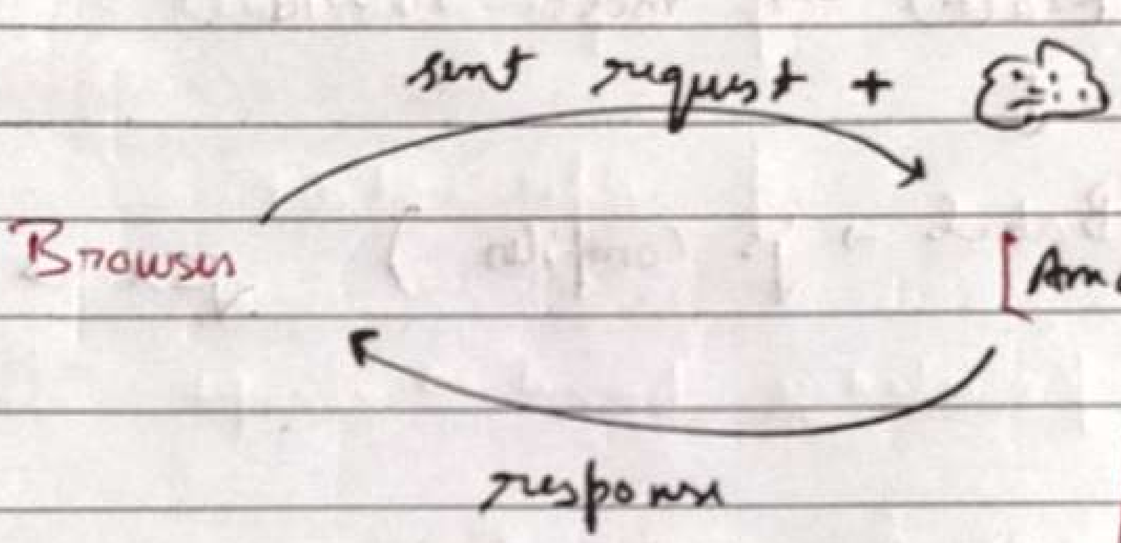
user password + Machine pwd —hashing→ hash key

→ cookies and session

today :



tomorrow :



→ use passport.js

require passport , passport-local , passport-local-mongoose

, express-session

POPU

Page No.
Date: / /

<code>

```
app.use(session({
    secret: "xyz",
    resave: false,
    saveUninitialized: false
}));
```

" Is library to build User Interfaces "

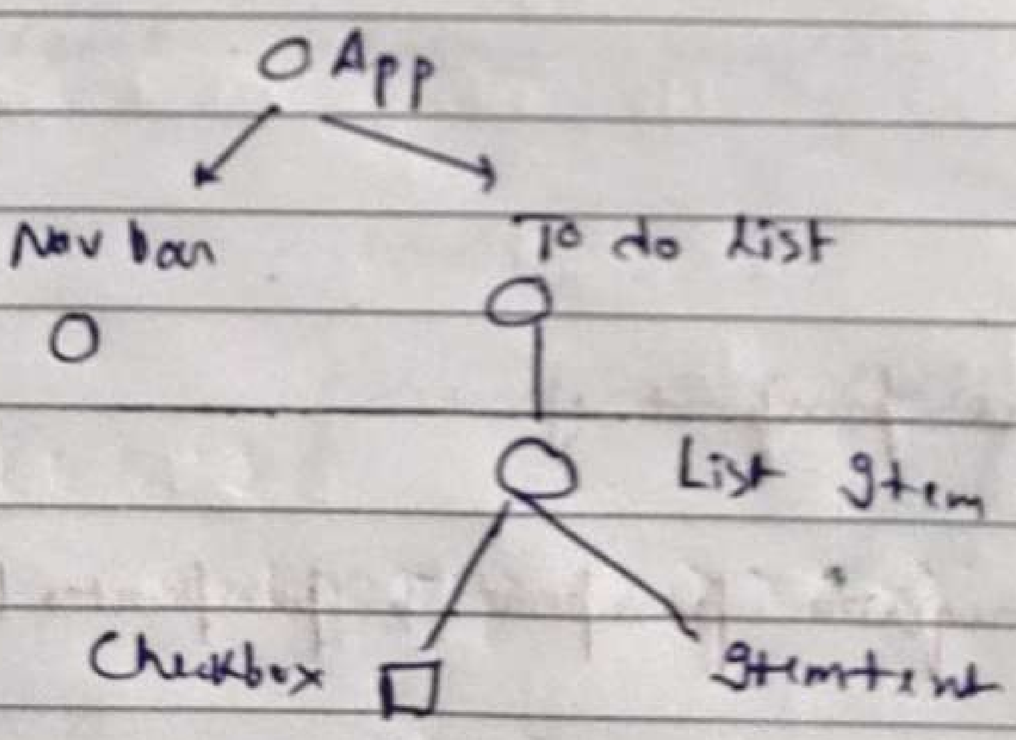
↑

(React) → most loved

↓

(Babel → js compiler) JSX → JS

Entire web App



POPU

Page No.
Date: / /

boiler-plate →

```
import React from "react";

import ReactDom from "react-dom";

ReactDom.render (<div>...</div>,

document.getElementById ("root"));
```

Importing and Exporting Model

Header.jsx

```
import ...

x

function Header() {

}

export Header;
```

→ when installing some internal repo with outdated node packages

gun commands →

$ npm install -g npm-check-updates

$ ncu -u

$ npm install

$ npm start

→ Knowledge bond → concept System organised collection of facts about a System

---

Arrow function

Var num = [1, 2, 3];

const NewNum = numbin.map (x => x * x);

or

(x, y) = { return x * y }

Set Interval (functionname, 1000);

helps to call a function repeatedly after specified time.