Model
Name
Email
age.

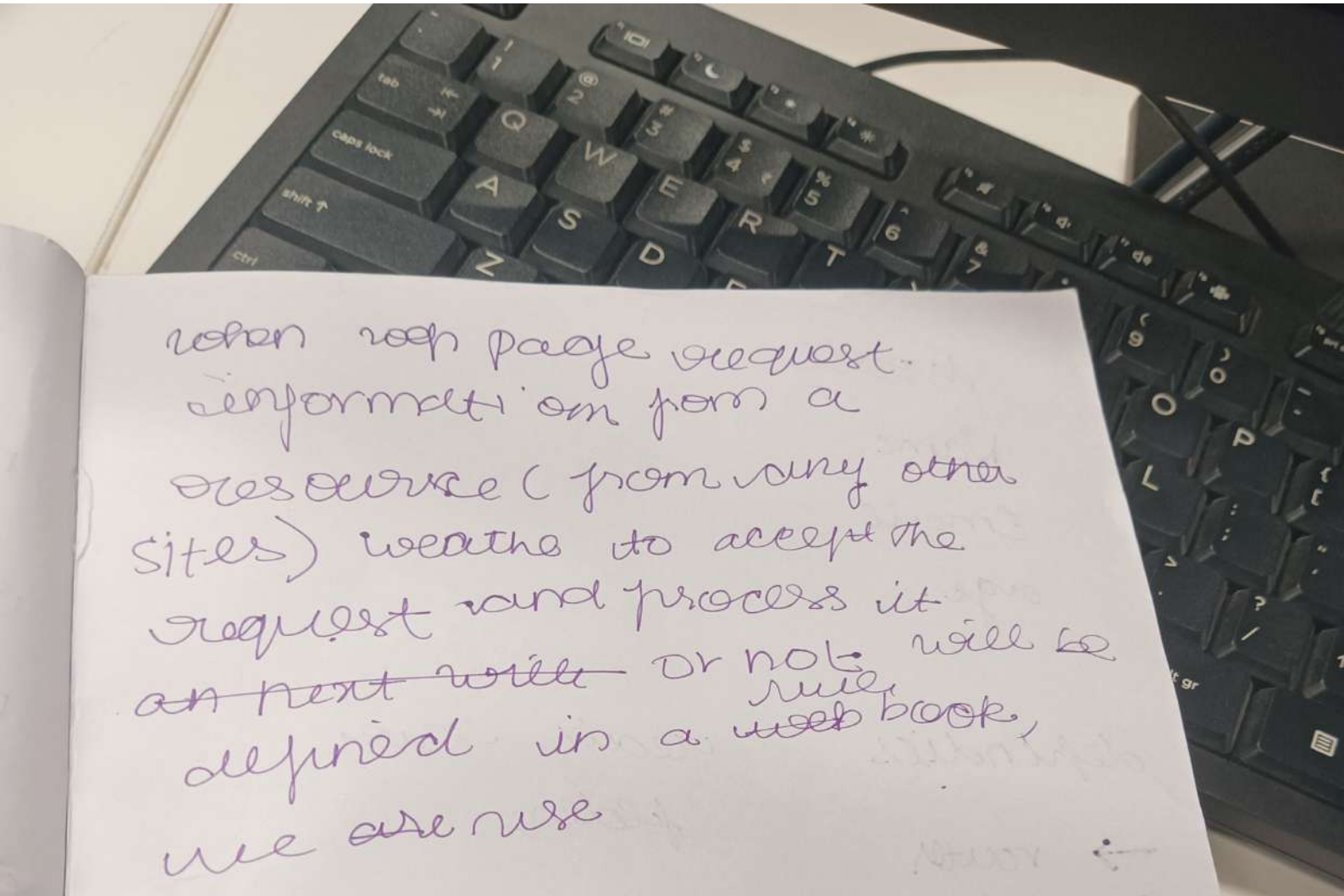
dependies

→ router
→ axios
→ express

create server folder

npm init -y
↓
create folder called server.

An index.js inside server folder write Backend code.



when web page request information from a resource (from any other sites) weather to accept the request and process it ~~on next with~~ or not. will be defined in a ~~web~~ rule book, we use use

→ npm i axios

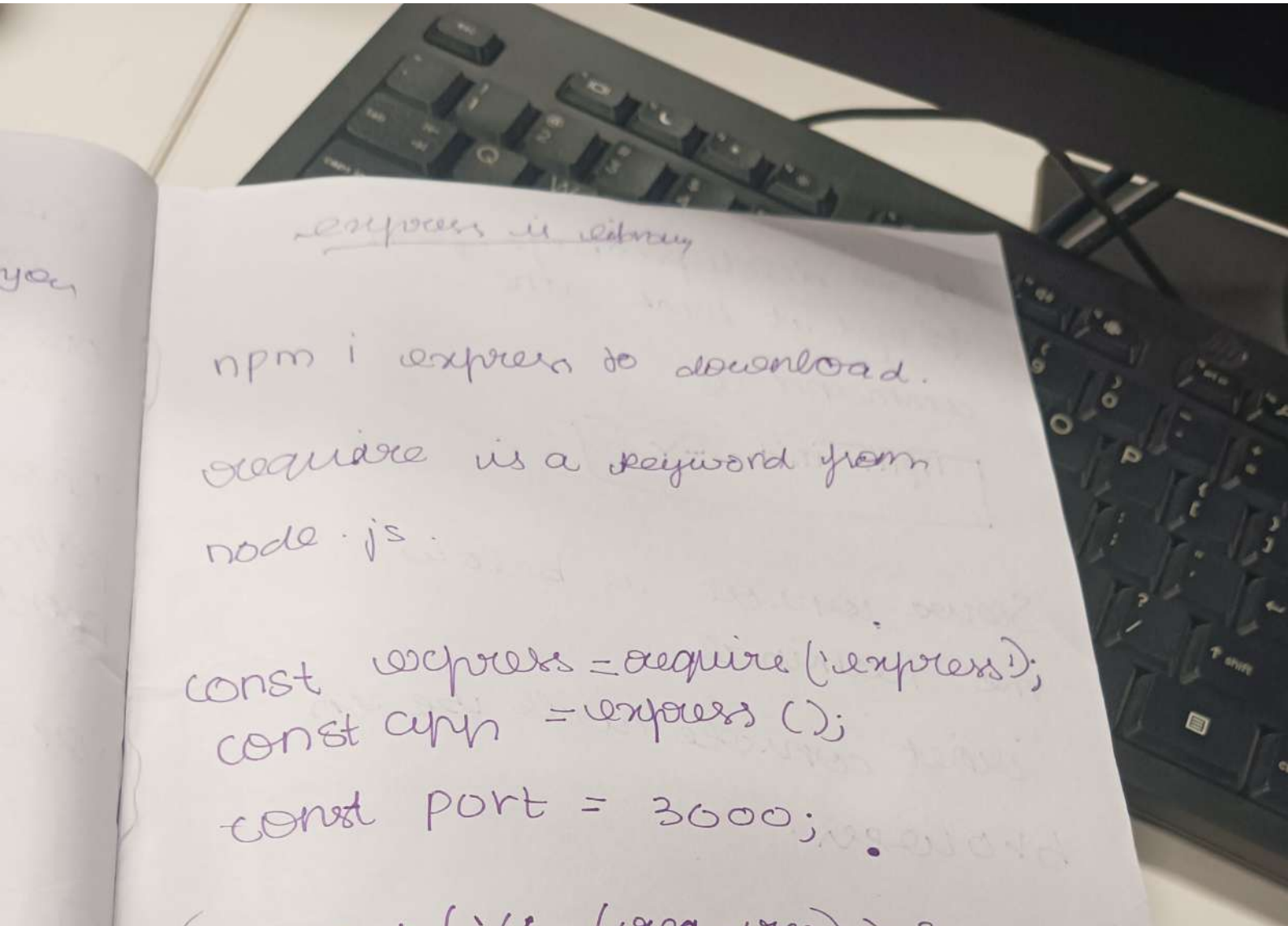
npm i cors.

If we delete package.json to get it back the command is

```
npm init -y
```

Server console is below the terminal

Client console will be in browser.



express is library

npm i express to download.

require is a keyword from node.js.

```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
    res.send('Hello, Node.js!');
});
```

Node.js

runtime environment that allows you to run javascript on the server side.

NOTE:

Maintain split terminals (2) in VS code in order to use client and server

run command

→ Start the server first

node server.js → file name

→ command to run client

npm start.

Create a db

computers

one collection

laptops

Insert

name, model, color, status, price

available or not available

vender name vender price

(1) list out particular model laptops.

(2) change it status vice versa

(3) list of all the

SQL - record
Mongo - Document

SQL - Table
Mongo - collection
collections are stored in DB
Mongo will have multiple DB

we can create DB
using command prompt
and with the help of
compass

Mongo-DB

→ NO-SQL
→ unstructured data) to process
→ file format in json files
↓
Java script object.

compass
→ Its an interface
→ helps us to reach mongo db server which means using compass we will fetch datas from server.

Mongosh

Mongo shell is replaced by Mongosh

Data modelling

Schema
→ It is an actual blueprint of DB which you created by giving the format using data modelling.

Redux

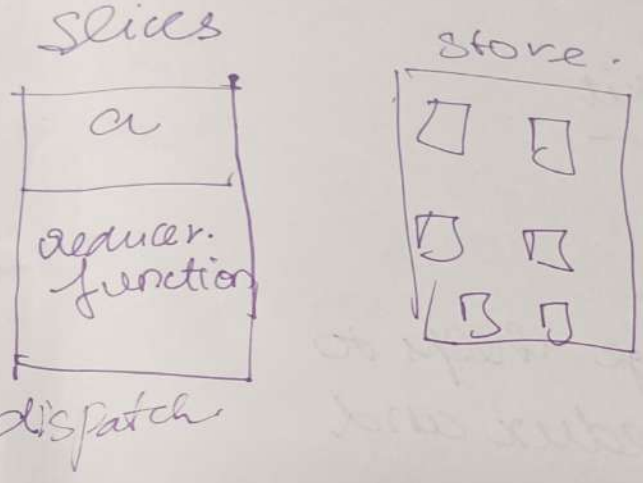
Installation

npm i @reduxjs/toolkit react-redux

redux js/tool kit
library.

react-redux
↳ Package helps to connect redux and react component

get the password from user as input
if the password is correct display
the component



Point - 2

useReducer returns array with two values like useState

first is that initial count and second is dispatch function

we call them as State and dispatch.

Here the State will hold initial value and updated once you called dispatch function. and dispatch will trigger updated function.

events
→ on click
→ on abort
→ on mouse over
→ on change.

USE REDUCER

same as usestate to manage or update states that is data that is values of components.
difference is if you have complex states or complex things you use usereducer hook.

Step 1:
create +, - program using usestate

Step 2:
replace use state with use reducer.

note
2 arguments
→ Reducer function. (what you want)
→ Initial value of state

useEffect

Synchronizing a component with external system

After action monitoring being sideeffects happening in the functional components - its possible using use effect hook.

components

→ app
→ container
→ user

usecontext

without following the hierarchy Passing State from one component to another component in efficient way using the hook.

Two imp things

1. create context
2. use context

In the given example. create context will be done in app component and that will be used in user component using usecontext.

Props can be passed between components only by following the hierachy which means.

Parent
↓ to
child

To overcome this in terms of efficiency we are using hooks.

conclusion

If we want we state from one compo to another compo only way to achieve is by passing Props in hierachy.

This is not efficient, to make it efficient we have an exclusive hook called UseContext

lets say 4 have 5 components

C_1, C_2, C_3, C_4, C_5

$C_1 \rightarrow$ Props (Hi)
↓ Pass to
C_5

and add display messages in component 1, 2, 3, 4, 5 respectively by keeping component one as parent (C_1) rest all childress as per the order.

So C_1 child is C_2, child is C_3 - child is C_4 - child is C_5

every component should display its name as message. as component 1 like that and display them sir wise from h1 – h5

NOTE

{ } → ~~Only~~ whenever we use something in { } it can be either javascript object or react component

use effect

upon, the condition or action we apply in our functional components monitoring the impact or sideeffects can be done using use-effect hook.

Note

use effect accepts two arguments

→ ~~call back~~

→ dependency array.

call back function is like constructor in java.

spread operator

→ helps to combine the array.

```
// Initial arrays.
const array1 = [1, 2, 3];
const array2 = [4, 5, 6];
const combinedArray = [...array1, ...array2];

console.log('Array 1:', array1);
console.log('Array2: array2);
console.log('combined Array:', combined
                                Array);
```

Best example for usState

Counter clock.

Starting the intial state as zero we can implement it decrement it using use state hook.

These are components

```
{persons.map(person)=>(
<Rough name={person.name}
        age={person.age}/>
))}
</div>
);
}
```

Passing Props between components

Earlier in IT industry they were using class component reason being state concept not available with functional component

Now hooks is used to implement state in functional component

useState
useEffect
useRef
useReducer
useContext

Flipkart website

Homepage

Groceries mobile Fashion } These are components

subtitle

mobiles → component name

Props

name

version

Price

State

discount

```
function App(){
  const person = [
  {name: 'vishal', age: 19},
  {name: 'deek', age: 20},
  {name: 'nandha', age: 22},
];
return (
  <div>
```

Props and States
↳ Property

every component will have Props and States

Props

It wont change

ex:
Name "Tata's Bisleri"

States

It changes or user can change it

ex:
water bottle level in bottle.

Initial state - full

updated state - half

current state - empty

DOM

→ Document object model

React follows virtual dom once here unlike html once dom gets created, the changes or manipulation what we do. gets completed and only that part will be re rendered.

where as in HTML everytime we make change entire dom will be re rendered.

Web application created by react js, each and everything called as component.

Types of components

→ functional component

→ class component

any → will only work with true

race → work with fast also

React JS

→ It's a lib or framework of JS

ex:
websites
Netflix
amazon.

HTML

→ Youtube
→ wikipedia

Two important folders in react

→ Public folder
→ SRC folder

3 important files
→ Index.html
→ Index.JS
→ Index.CSS

NOTE:
As of now dont touch index file

Intially write the code in App.JS.

Promise unbuilt methods

when there is more than one promise in order to receive them we can use promise unbuilt method according to the requirement

Subtible→methods

→ Promise.all
→ Promise.any
→ Promise.allsettled
→ Promise.race.
→ Promise.all .
once it sees promise false- it will stop.

Promise.any
gives the shortest time promise,
Provided Status should be true

will displays any of these 3 states.
Promise.allsettled
→ fullfilled
→ rejected
→ Pending

write a promise called "andhra-BP"

Distance

andhra → A = 5000

andhra → B = 2000

andhra → C = 1000

or:

C reached

B reached

A reached.

Promise → JS objects

resolved (success)

Rejected (failure)

```
<html>
<script>
let reach A = new promise ((resolve, reject) => {
    let reached = true
  if (reached)
  Set Timeout (resolve, 1000, "A reached")
  else
   reject ("A not reached")
  })
```