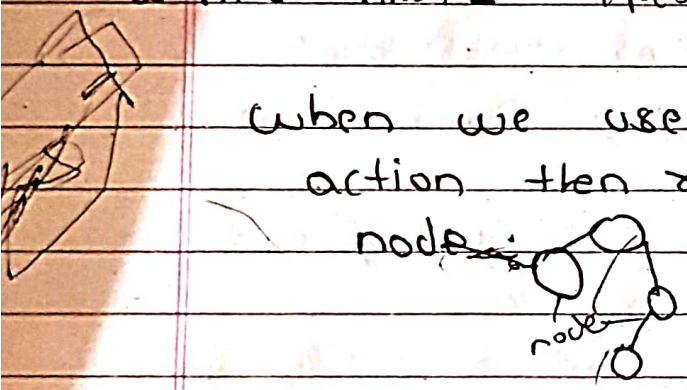
L-6

- Hook is normal function basically
- React is fast because it uses virtual Dom Reconciliation, diff algo, blunder.
- React does not track variable but it keep track of usestate variable.
- useState("", "") need initial value

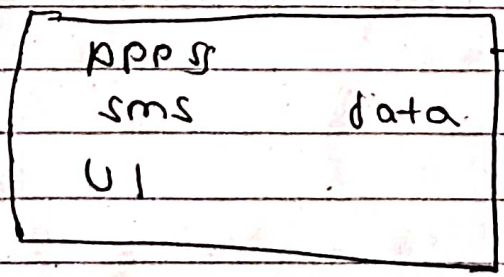
at time 41:00 L-6 → perform task,

when we usestate & perform any action then react will update that node



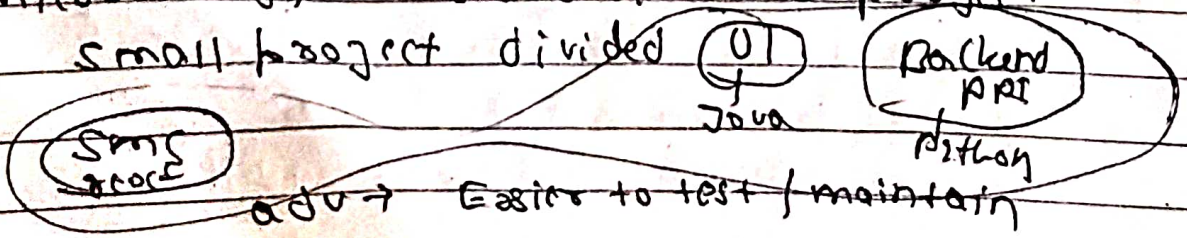
microseruices!

MONOLITH → Everything in one repo



In 1 project each & every action perform

microservig → Instead of one project small project divided



adv → Easier to test / maintain

Uber - math behind.

How these microservice connected to each other.

for example :-

Swiggy.com : 3000 → / (port no)
(domain name) : 4000 → /page
" : 5000 → /modification

they use same domain name & but different port no

Some company also change domain name.

Calling API

So, when we search it rerender component So, calling API is not good as because for small change in search it will render API again and again.

So, 2 ways

(i) Page loads ⇒ API (300ms) ⇒ render page (500ms)

(ii) Page loads ⇒ render initial page (100ms) ⇒ API (300ms) ⇒ update UI (50ms)

best

→ useEffect - to do this we have useEffect.

SAGAR
Page No
Date

"Hook which is JS

useEffect is a function useEffect() = {

}

we call useEffect by calling another function that another function is callback function

callback function → function which is not call immediately it will be called when useEffect wanted to be call.

useEffect(() => {

}, []);

this is dependencies.

we call useffect by two parameters one is callback & other is dependency array.

Do work
1:30

Suppose if we give searchtext in array

useEffect(() => {

}, [searchtext]);

here it depend on searchtext if State changes then useffect call.

Also Note ~~& initially~~ also called at begining after render other console.

- and if it is empty [] then after render other.. it will be called only once & it did not depend here on any one.

dep array [searchText] → once after initial render
+
every time after rerender

At 1:22 perform work

1:24 task

→ useEffect — if it is with empty dependency array - []
↳ it will ~~render~~ once after initial render
call

→ dep array [search] → once after intial render
+ everytime after rerender (searchtext change)

one ex — ~~Geography~~ useEffect is callback function so, it will render 1st.

Simply — useEffect ~~first~~ render itself after rendering other ~~code~~ sequentially line.
1st - render
2nd - useEffect

At 1:29 work

— Browser doesnot allow us to fetch any other domain data from localhost
~~SQ — SSD use to~~ So we use any extension like here CORS

At 1:40 → chrome extension

— browser-fetch-item-watch → click on network
click disable cache
click Fetch/XH & select Name & click
(Filter on Headers)

— find optional chaining

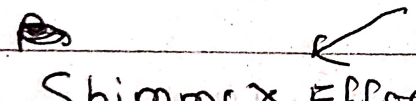
~~At — 1:53 task balance~~

~~always~~ then useEffect render

9:08
work

9:11 → postloader → Spinningloading → bad user experience

So, empty box of loading is good UI experience.



Prod it ✓ Shimmer Effect

Shimmer UI Code put in by using Conditional Rendering.

→ Shimmer code how to

Basic — ~~testt~~ - Installation - Revision

— 1st of all

after installing node JS

npm ~~install~~ init

it will ask few questions

Test: Test ;

GitHub repo: yourschool link

License: (ISC)

name: anything

Keywords: - nothing

Entry point: - APP.JS

node module & package.json

— This will install node module & package.json

Lock file install

npm install -D parcel } — package-lock.json file

npm install react

npm install react-dom

npx parcel index.html

2:40 HW

② JS Expression & Statement diffrent

{

// Here we can write any JS Expression only.

Let a = 10 — ✕ No you can't this is statement

console.log(a) - ✓

}

but trick

↓

((a=10), console.log(a))

console.log is Expression

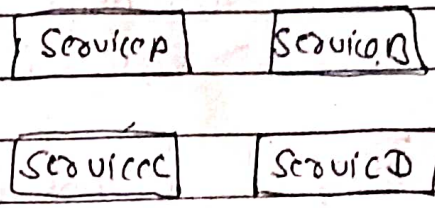
Assignment:-6

Page No.
Date

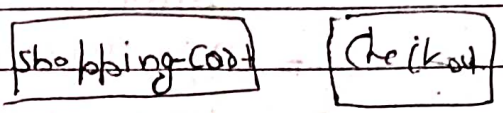
SAGAR

Q) What is a microservice

- Split application into smaller independent services



- Split based on business functionality
- Separation of Concerns:
  1 Service for 1 Specific Job
  or
- Self-contained & Independent



- Developed, deployed and scaled separately.

Services can be built and deployed separately

- Each microservice has its own version

- Communication b/w microservices:

- Each service has its own API
→ They can talk to each other, by sending requests to respective API endpoint
  - This is synchronous communication
  - wait for the response

- Second way to transfer.
- Communication via message (message broker)
- Common distribution pattern
  Publish/subscribe & point to point messaging.

3rd way : by service mesh like Istio

Downside :- Configure communication b/w services

~~Each microservice~~ - more difficult to monitor with multiple instance of each & service distributed across servers.

2) what is monolith architecture :-

- All Components are a part of single unit.
- Everything is developed, deployed and scaled as 1 unit
- App must be written with 1 tech stack
- Team need to be careful to not affect each other work
- 1 single artifact, so you must redeploy the entire application on each update.

Challenges.

- Application is too large & Complex
- Parts are more tangled into each other
- you can only scale the entire app instead of a specific service
- difficulty if services need different dependency version.

3) why do we need a useEffect Hook

useEffect is React hook that lets you synchronize a component with an external system.

Synchronize a component with external

System means :- Effects let you run some code after rendering so that you can synchronize your component with some system outside of React.

its usage
- Connecting to an external system
- Wrapping Effects in custom Hooks
- Controlling a non-React widget
- Fetching data with Effects

useEffect (setup, dependencies)
↳ it is call back function

First Normal code run then it render.

dependencies → explained in previous pages.

What is Optional Chaining?

Optional Chaining is feature that allow you to access property of an object or element of array without having to check whether the object or array is null or undefined first. it is represented by the ?. operator and can be used to concisely access

deeply nested properties without having to write a long chain of if statements to check for null or undefined values.

Ex-

```
const user = {
    name: "singh",
    address: {
    street: "ok",
    },
};
```

console.log(user.address.zodno.housno)

I know both does not exist & this will throw error

So now using Chaining

console.log(user.address?.zodno?.housno)

if it exist then show otherwise undefined

it will give actual undefined not error that we want.

Optional chaining can also be used to access elements of array in Javascript. it works in a similiar way to access properties of an object but using the ?.[] operator instead of the ?. operator

Ex-

```
Const users: = [
    {name: "John", age: 30},
    {name: "Rohan", age: 25}
];

Console.log(users[3]?.name);
```

2. ~~Uses of if else statement~~

2. What is shimmer UI
2. What is difference between JS Expression and JS statement.

Expression:- piece of code that always produces ~~some~~ something or

Statement:- an instruction to perform a specific action Ex for Sum;

Conditional Statement
tells whether piece of code should run or not
if else switch case

( ) ? TRUE : FALSE

Page No.
Date

2. What is Conditional Rendering, explain with a code example.

Conditional rendering | if condition

old way:-

```
const [loggedIn, setLoggedIn] = useState(true)

if (loggedIn) {
    return (
        <div>
            <h1>welc</h1>
        </div>
    )
} else {
    return (
        <div>
            <h1>welc</h1>
        </div>
    )
}

return (
    <div>
        {loggedIn ? <h1>welc</h1> : <h1>welc</h1>}
    </div>
)
```

for more than 2 condition i.e (>= else if

```
const [loggedIn, setLoggedIn] = useState(2);

return (
  <div>
  { loggedIn == 1 ?
    <h1> WLC 1 </h1>
    : loggedIn == 2 ? <h1> WLC 2 </h1>
    : <h1> WLC user 3 </h1> }

    </div>
  )

}
```

→ what is CORS?

CORS is mechanism which uses additional HTTP headers to tell the browsers whether a specific web app can share resource with another web app but both web app should have different origin. if they same origin they can share resource very easily

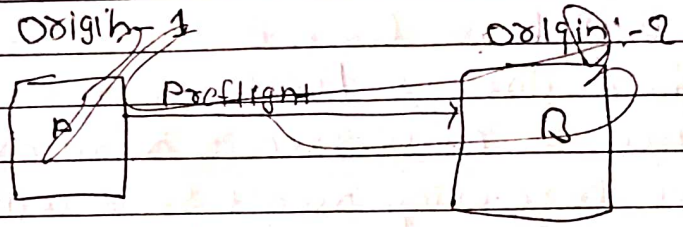
→ older browse Not allow this

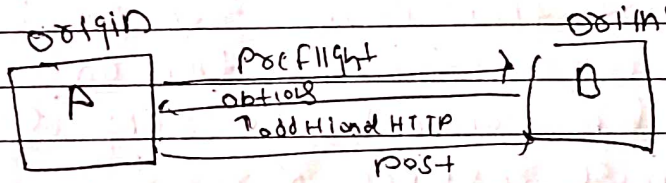
if any website like

aksay.saini.com access/get data from

google.com/gd/data ⊗ different domain not allowed
api.akash.in ⊗ Not even subdomain
akash.saini.is50 ⊗ Not even to different port
http ... ⊗ Not to request to from https to http.

Now modern browser allow this all & have CORS standard mechanism





Not for all
for their criteria
• Set

what is async & await

async fn - it will return promise
→ it is used to define an asynchronous function. An asynchronous function always return a Promise.

The await is used to pause the execution of an async function until a promise settled. it can only be used on async function

When await is used with Promise it suspends the execution of function until the Promise is resolved & then it returns resolved value & if promise is rejected, it throws an error.

Q What is use of Const Json = await data.Json in getRestaurants()

→ ~~it will first fetch d~~ when ~~after~~ data is fetch ~~it waits~~ Since ~~to~~ Const Json = await data.Json we here using await so, it will Check whether data fetching is completed or not if if not then it will run other code & if it is completed then data will be converted to JSON & Promise resolved & return.

SAGAR
Page No.
Date

What is Shimmer Effect?

Basically we can say Shimmer Effects are loading indicator used when fetching data from a datasource.
it is visual technique used to add a touch of interactivity and engagement to ~~carbon~~ webpage.
mainly used as loading indicator-it gives users a visual indication that something is happing in background, reducing the perception of delay & providing sense of progress.