//Ep:06  
  
useEffect(()=>{},[]) hook:  
>It usually use to make external call/API call.  
>It calls after initial render of the component.  
>It takes two arguments :  
1>a callback()  
2>[] -> dependency array(it’s a optional argument)

>case 1: if don’t pass anything as a second argument, useEffect() will call after every render.  
>case 2: if we pass an empty array, i.e there is no dependency, it will only call once after initial render.  
>case 3: if pass any state variable as dependency, then useEffect() will call after initial render, + every time when state variable gets updated/changed.  
e.g, useEffect(()=>{},[Restro])  
  
  
  
useEffect(()=>{

 const timer= setInterval(()=>{console.log("profile compDidMount")},1000)

//this return use for unmounting the func. component while changing page(precisely component as comp gets change in the same page)

 return()=>{

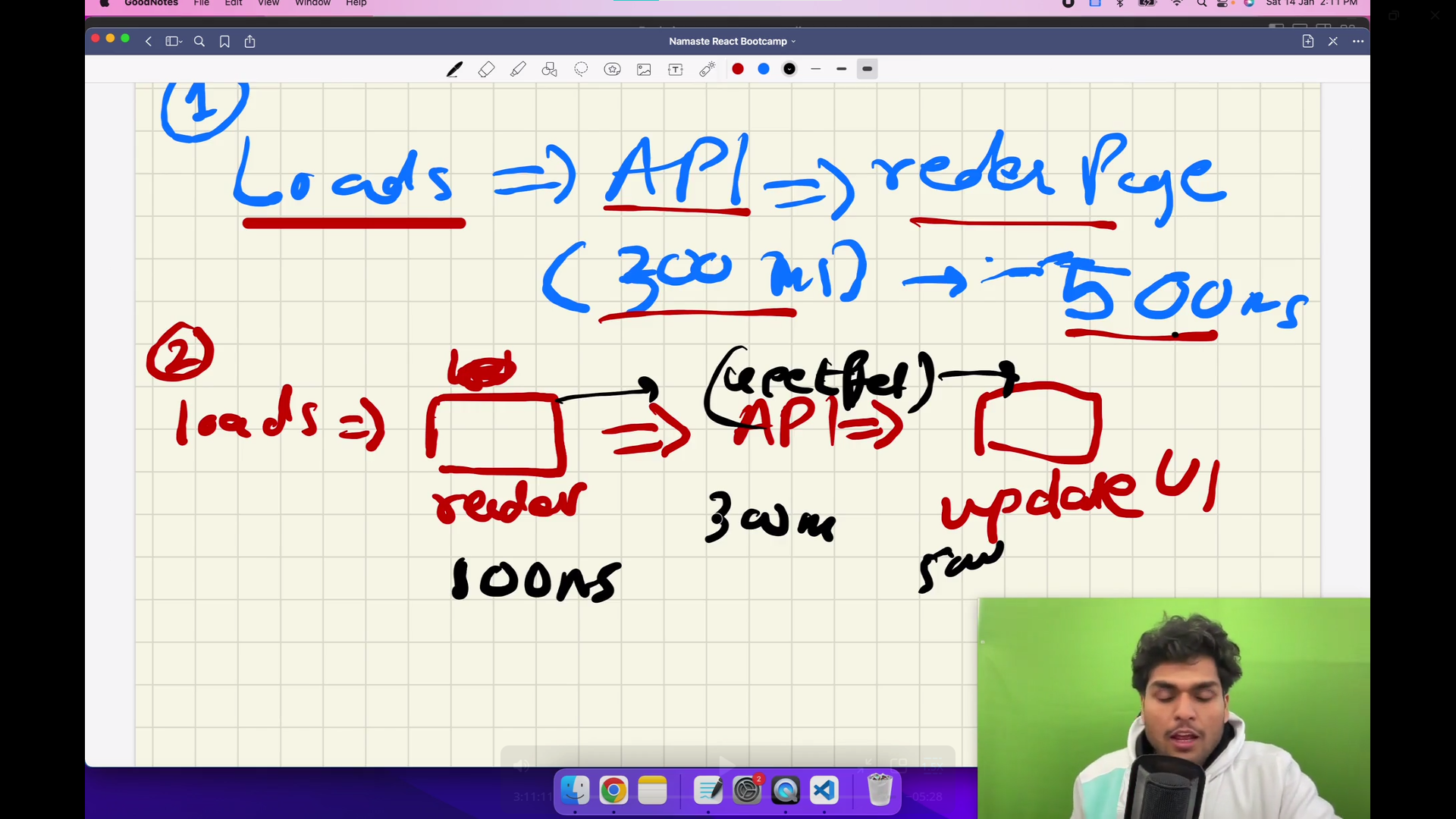
        //before unmounting it's good to clear unnecessary thing which could make app slower and unscalable

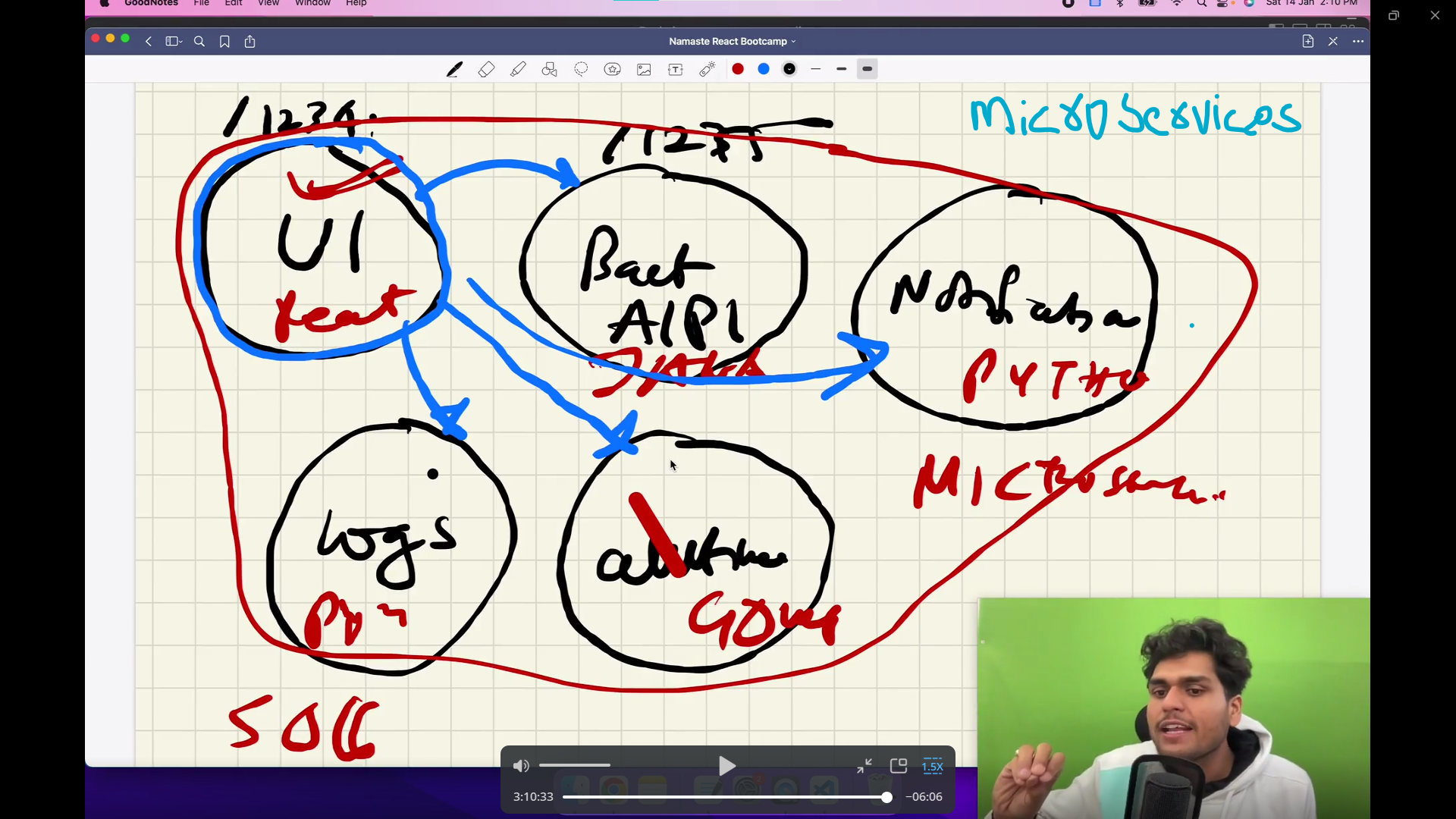
        clearInterval(timer)

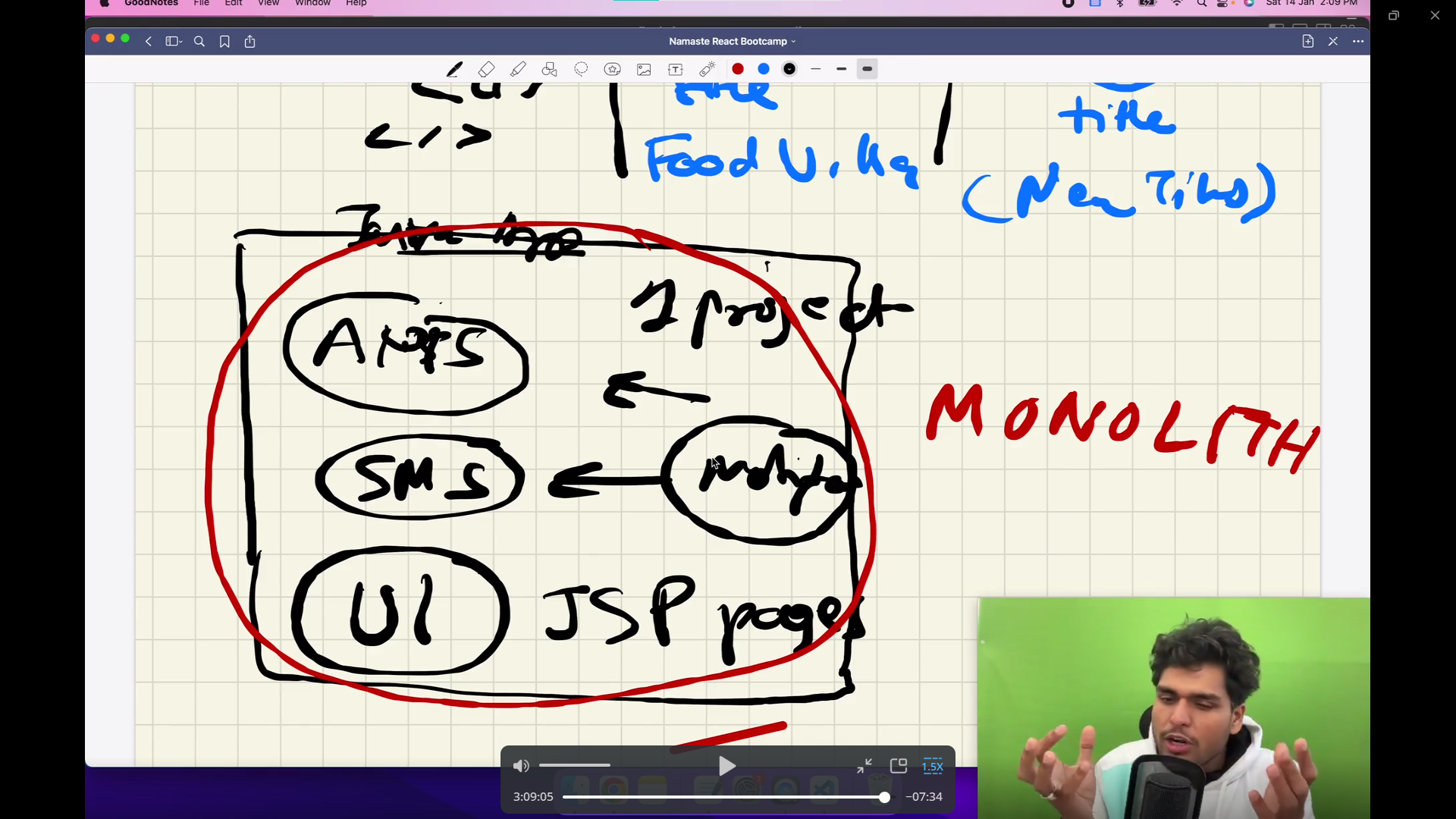
        console.log("unmounting functional comp")

 }

},[])

Note:  
>we can use several useEffect() inside a functional component.  
  
// async function to call api using fetch() method:  
const callApi= async()=>{  
const data= await fetch(“swiggy\_url”)   
// to get data in json format:  
const json= data.json()}  
  
  
  
  
// how app should load ?  
> the second way is correct way to load app :  
>i.e, load=>Render UI(dummy UI/ shimmer UI)=>Api call => UI update(render after getting data)  
  
  
  
  
  
//MicroServices :  
>separation of concern  
>it follows single responsibility principle(when each and every service has its own job without interfering others)

>this services are independent on tech stack  
  
  
  
//Monolithc  
>dependent on tech stack



>Api call in js , can be done by fetch, promises, AJAX.  
>fetch() is provided by browser.  
  
  
// Q. why do we need state variable?

>React doesn’t track normal js variable, if we use it in component and update its value , it won’t sync with UI component so no change will be reflected respective of variable values, however unlike normal js state variable synchronise with component and when state variable gets updated ,it re-renders the component where it declared.  
  
>when we need to update UI , it must synchronise with state variable.  
  
// Q. why React is fast?  
>Bcoz it has reconciliation process which consist diff algo, and when anything updated in DOM diff algo keep track on two state of virtual DOM (i.e older and newer)  
  
>note:  
>Diff algo written in React-core library, however Dom update happens by React-dom.  
  
// Q CORS policy?  
>when we get such error like origin mismatched , it generally block by Browser, It does’nt allow to call API from one origin to another origin.  
e.g(like from localhost to someone else origin e.g swiggy)

// Q. how to handle in useEffect()?  
//Q. not render component or early return ?  
//Q. Js expression and statement?  
//Q. JSx curly brackets?  
  
//Ep -07:  
  
//for routing install the packg.  
  
>npm i react-router-dom  
  
>when we’ve to develop routes , we’ve to create routing configuration, using createBrowserRouter.  
>configuration tells what will happen on a specific route/path.  
  
  
// React Route/path configuration:  
//createBrowserRouter([{path:””,element:<>,children:[{},{}]}]):  
> It’s a component use to create router.  
>it’s provided by “react-router-dom”, need to import it .  
 >It contain list of path(e.g, “/”, ”/about” etc.), where each path associated with an object I,e{path, element,errorElement}  
>where element takes tag/component,  
>errorElement, takes error component(error.js), it handles error.

const appRouter= createBrowserRouter([  
{

path:"/",

element:<App/>,

children:[

{

path:"/",

element:<Body/>

},

{

path:"/about",

element:<AboutUs/>

},

{

path:"/contact",

element:<ContactUs/>

},

{

path:"/resturant/:id",

element:<ResturantMenu/>

},

],errorElement:<Error/>,

}])

//<Outlet/>  
>it’s a “react-router-dom” component.  
>It filled by ‘children’ configuration.  
>all children go into the outlet acc. to the route/path  
>sometimes we’ve to load component acc. to routing, for that we use outlet, conditional/dynamic routing. (e.g if I want to keep header ,footer in all the pages just want to change content of body portion in different page I would use outlet with some configuration. )   
  
//Nested routing:  
defining ‘children’ route inside any route  
//useRouteError:  
>it’s a hook, provided by “react-router-dom”, usually use in error component i.e(Error.js page), to get exact routing error due to URL/path.  
  
//RouterProvider:  
>it’s a component of “react-routet-dom”  
>It provides router to render

>i.e it provides routing configuration to the app.  
root.render(<RouterProvider  router={appRouter}/>)

Imp\*> we shouldn’t use <a> tag to navigate pages/routing to other pages in react component, as it re-load whole page every time it gets clicked , instead use <Link> tag as it doesn’t reload whole page, it only re-render component .

//<Link> :  
>It’s a component of “react-router-dom”  
>It’s a wrapper over <a> tag, bts react-router-dom to keep track on link tag.  
>it is use to routing path without re-loading page( to achieve SPA)  
  
<Link to =”/about”>About us</Link>  
  
//SPA:  
>React is a single page application, that’s the another reason that it’s fast  
>it doesn’t reload whole page after initial loading when we navigate to different pages to pages, instead change components which doesn’t need to re-load whole page.  
  
Two types of Routing in web Apps:  
1>Client side routing:It means not loading anything from server side instead load page from provided components.  
2>Server side routing  
  
Dynamic Routing:  
//useParams  
>It’s a comp. of “react-router-dom”  
>use to get/read dynamic values of url/path parameter.  
// by using ':' we can define dynamic parameter in a url. here ':id' is a dynamic parameter , nd we can extract its value using 'useParams' hook

{

path:"/resturant/:id",

element:<ResturantMenu/>

},

// {} -> it use to extract values  
  
>while using map(), always give key to parent element.  
  
IMP\*  
//Q. use formic library for login-forms  
//Q. why CDN is best place to host images?  
// Q. how swiggy Api is related to client routing, as we are calling external call/ aPi call?