Part 1:

What is useEffect Hook?

Its a hook in the react

When it will be called?

It depends on that we have passed in the dependency array.

If we have not passed anything in the dependency array, it will call whenever a component is re-render.

useEffect(()=>{

console.log("Use Effect hook called")

});

But we called with blank independency array then it will call only once at the time of initial loading.

useEffect(()=>{

console.log("Use Effect hook called")

}, []);

But when we pass any state variable in the dependency array,

const [name, setName] = useState('John Doe');

useEffect(()=>{

console.log("Use Effect hook called")

}, [name]);

It will only call when the value of name got change.

Part 3: React Router Dom ( for us only rrd)

Npm i react-router-dom

To create route

We have to write route configurations.

Import createBrowerRoute from RRD

We have to make route on root level file (App.js)

We have to pass list of ruts to createBroweserRoute function

Lets create

const appRouter = createBrowserRouter([

{

path:"/",

element:<AppLayout />

},

{

path:"/about",

element:<About />

}

]);

We need to provide this configuration to render.

We have to use RouterProvider

root.render( <RouterProvider route={appRouter} /> );

import { createBrowserRouter, RouterProvider } from 'react-router-dom';

So we can use a Router to show in the header line.

With the help of that router, we can manage the Error code.

Import { useRouteError } from react-router dom

To get a proper message

We can use all data from the routes

There are multiple data in which we can get its error code status and all other details

Now after adding page we are not getting header

Part 4: Children routes

As of now we are getting only content of about page without header

To show header on all apge

We have to add childerns to the routes.

We can load body dynamically with help of outlet

Import { Outlet } from react router dom

Lets see old code without proper routing and without common header

import React from 'react';

import ReactDOM from 'react-dom/client';

import Footer from './components/Footer';

import Header from './components/Header';

import HomeRestaurants from './components/HomeRestaurants';

import { createBrowserRouter, RouterProvider } from 'react-router-dom';

import About from './components/About';

import Contact from './components/Contact';

import Error from './components/Error';

const AppLayout = () =>{

return (

<div>

<Header />

<HomeRestaurants />

<Footer />

</div>

)

}

const appRouter = createBrowserRouter([

{

path:"/",

element:<AppLayout />,

errorElement:<Error />

},

{

path:"/about",

element:<About />,

errorElement:<Error />

},

{

path:"/contact",

element:<Contact />,

errorElement:<Error />

}

]);

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render( <RouterProvider router={appRouter} /> );

Now see the new code with Outlet and dynamic body with common header

import React from 'react';

import ReactDOM from 'react-dom/client';

import Footer from './components/Footer';

import Header from './components/Header';

import HomeRestaurants from './components/HomeRestaurants';

import { createBrowserRouter, Outlet, RouterProvider } from 'react-router-dom';

import About from './components/About';

import Contact from './components/Contact';

import Error from './components/Error';

const AppLayout = () =>{

return (

<div>

<Header />

<Outlet />

<Footer />

</div>

)

}

const appRouter = createBrowserRouter([

{

path:"/",

element:<AppLayout />,

errorElement:<Error />,

children:[

{

path:"/",

element:<HomeRestaurants />,

},

{

path:"/about",

element:<About />,

},

{

path:"/contact",

element:<Contact />,

}

]

},

]);

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render( <RouterProvider router={appRouter} /> );

So based on that we can have multiple designs

We can set different theme to the code and based on that we will add item to childer to that body will be part of its outlet.

Element will be replaced in the outlet in the component.

For navigation, we should not use an Anchor tag.

Because it will load the whole page again. Then there will not any sense to use react if you are reloading page again and again.

There is a feature in React for routing.

We can navigate to other page without reloading the page.

With help of Link component

Import { Link } from react router dom.

Link work same as Anchor tag

When we use anchor tag

<a href=”/about”>ABout</a>

In React

<Link to=’/about’’>About</Link>

2 types of routing in web app

1. Client-side routing
2. Server-side routing

Url: /resturent/:resId (here id is dynamic)

We can get this dynamic value in variable from react router dom.

Import { useParams } from ‘react-router-dom’

Const params = useParams();

console.log(params}

{ resId: 123 }

We can get directly get like

Const {resId} = useParams();