**Diy 75:**

Fetch from api and display it in table and implement search functionality

function App() {

  useEffect(()=>{

    fetch('https://jsonplaceholder.typicode.com/users')

      .then(response => response.json())

      .then(json => console.log(json))

  },[])

  return (

    <>

    <div>

    </div>

    </>

  )

}

  const [data,setData]=useState([])

console.log(Data) after useeffect

setdata in useeffect

function App() {

  const [data,setData]=useState([])

    useEffect(() => {

    fetch('https://jsonplaceholder.typicode.com/users')

      .then(result => result.json())

      .then((resp )=> {

        console.log("result",resp)

        setData(resp)

      })

  }, [])

  console.log(data)

  return (

    <>

    <div>

    </div>

    </>

  )

}

Execute and Check data in console .

Creating table

    <>

    <div>

      <h1>get api call</h1>

      <table>

        <tr>

          <td>id</td>

          <td>name</td>

          <td>email</td>

          <td>phone</td>

        </tr>

      </table>

    </div>

    </>

        <table border="1" >

          <thead>

          <tr>

            <td>id</td>

            <td>name</td>

            <td>email</td>

            <td>phone</td>

          </tr>

          </thead>

          <tbody>

          {

            data.map((item) => {

              return(

              <tr key={item.id}>

                <td>{item.id}</td>

                <td>{item.name}</td>

                <td>{item.email}</td>

                <td>{item.phone}</td>

              </tr>

              )

            })

          }

          </tbody>

        </table>

Got data displayed in table

**Search functiolality:**

  const [search,setSearch]=useState('')

**create a input field above the table tag**

        <input type="text"  placeholder='search here'/>

        <table border="1" >

**Add onchange handler**

        <input type="text" onChange={(e)=> setSearch(e.target.value)} placeholder='search here'/>

  const [search,setSearch]=useState('')

  console.log(search)

**console.log(search) – to check**

**applying filter:**

**to search by name:**

            data.filter((item)=>{

             return  search.toLowerCase == ' ' ? item : item.name.toLowerCase().includes(search)

            }).map((item) => {

              return(

              <tr key={item.id}>

                <td>{item.id}</td>

                <td>{item.name}</td>

                <td>{item.email}</td>

                <td>{item.phone}</td>

              </tr>

              )

            })

          }

**Search by id:**

            return search.toString === '' ? item :item.id.toString().includes(search)

**diy 76:**

**react router dom :**

npm I react-router-dom package

**react is single page application – no reload on navigating from home to about etc.**

**app.js**

function App() {

  return (

    <>

    <div>

      <h1>welcome to react router dom tutorial</h1>

    </div>

    </>

  )

}

export default App

**index.js or main.jsx**

**import browserrouter**

import { BrowserRouter }  from 'react-router-dom'

**then wrap <App/> inside <BrowserRouter>**

import { BrowserRouter }  from 'react-router-dom'

ReactDOM.createRoot(document.getElementById('root')).render(

  <React.StrictMode>

    <BrowserRouter>

    <App />

    </BrowserRouter>

  </React.StrictMode>,

)

**Then all routes we need to specify inside <Routes> component wrapper**

**App.js**

import {Routes,Route} from 'react-router-dom'

**create a pages folder in src**

**and create pages like** [**home.jsx,about.jsx,product.js**](http://home.jsx,about.jsx,product.js) **inside it**

**import these pages into app.js**

import Home from './pages/home';

import About from './pages/about';

import Products from './pages/products';

**now adding route paths**

    <div>

      <h1>welcome to react router dom tutorial</h1>

      <Routes>

        <Route path='/'  element={<Home />} />

        <Route path='/about'  element={<About />} />

        <Route path='/products'  element={<Products />} />

      </Routes>

    </div>

**Now check go to / endpoint , /about , /contact endpoint**

**Now instead of gng to url , can have buttons or links like navigation tab**

**Create src> components folder**

**Inside it create Navbar.jsx component**

import React from 'react'

const Navbar = () => {

  return (

    <div>

      <ul>

        <li>Home</li>

        <li>About</li>

        <li>Products</li>

      </ul>

    </div>

  )

}

export default Navbar

**now to add <a> anchor tag, we can use link tag**

**instead of <a href=””> we use link ta g <Link to=’/’ >**

import { Link } from 'react-router-dom'

            <div>

            <ul>

                <li>

                    <Link to='/'>Home</Link>

                </li>

                <li>

                    <Link to='/about'>About</Link>

                </li>

                <li>

                    <Link to='/products'>Products</Link>

                </li>

            </ul>

        </div>

    )

}

**Then call navbar component in app.js**

**App.js**

import Navbar from './components/Navbar';

function App() {

  return (

    <>

    <div>

            <header>

        <Navbar />

      </header>

      <h1>welcome to react router dom tutorial</h1>

      <Routes>

        <Route path='/'  element={<Home />} />

        <Route path='/about'  element={<About />} />

        <Route path='/products'  element={<Products />} />

      </Routes>

    </div>

    </>

  )

}

export default App

**add styles to ul and li**

**add these styles in App.css or Navbar.css**

ul{

  display: flex;

  list-style: none;

  justify-content: right;

  padding-right: 30px;

}

li{

  margin-right: 20px;

  font-size: 20px;

  cursor: pointer;

}

**Now check navigation**

**Instead of Link , can use NavLink – inspect and check NavLink will accept class attribute and id attribute to style it**

**//instead of routes can use now createBrowserrouter – from react 6.4**

**Index.js**

Replace import {BrowserRouter} line with below code

import {

createBrowserRouter,

RouterProvider,

} from "react-router-dom";

**Instead of routes can wrap direct into routetag**

import Home from './pages/home';

import About from './pages/about';

import Products from './pages/products';

import Navbar from './components/Navbar';

import {

  createBrowserRouter,

  createRoutesFromChildren,

  createRoutesFromElements,

  Route,

  RouterProvider,

} from "react-router-dom";

const router=createBrowserRouter(

  createRoutesFromElements(

    <Route path='/' element={<App />}>

              <Route index  element={<Home />} />

        <Route path='about'  element={<About />} />

        <Route path='products'  element={<Products />} />

    </Route>

  )

)

ReactDOM.createRoot(document.getElementById('root')).render(

  <React.StrictMode>

    <RouterProvider router={router} />

  </React.StrictMode>,

)

**App.js**

import Navbar from './components/Navbar';

function App() {

  return (

    <>

    <div>

      <header>

        <Navbar />

      </header>

      <h1>welcome to react router dom tutorial</h1>

    </div>

    </>

  )

}

export default App

**now to tell where to display – use outlet**

**app.js**

import { Outlet } from 'react-router-dom';

    <div>

      <header>

        <Navbar />

      </header>

      <h1>welcome to react router dom tutorial</h1>

      <Outlet />

    </div>

    </>