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

import { Suspense, lazy } from 'react';

import React from 'react';

const Dashboard = lazy(() => import('./components/Dashboard'));

const Landing = lazy(() => import('./components/Landing'));

function App() {

  return (

    <BrowserRouter>

      <AppBar />

      <Suspense fallback={<div>Loading...</div>}>

        <Routes>

          <Route path="/dashboard" element={<Dashboard />} />

          <Route path="/" element={<Landing />} />

        </Routes>

      </Suspense>

    </BrowserRouter>

  );

}

function AppBar() {

  const navigate = useNavigate();

  return (

    <div>

      <button onClick={() => navigate('/')}>Landing Page</button>

      <button onClick={() => navigate('/dashboard')}>Dashboard</button>

    </div>

  );

}

export default App;

**WITHOUT USING PROP DRILLING/Context API**

function App(){

  const [count,setCount]=useState(0);

  return(

    <div>

      <Count count={count} setCount={setCount }/>

    </div>

  )

}

function Count({count,setCount}){

  return (

    <div>

      {count}

      <Buttons count={count} setCount={setCount}/>

    </div>

  )

}

function Buttons({count,setCount}){

  return (

    <div>

      <button onClick={()=>{

        setCount(count+1)

      }}>Increase</button>

      <button onClick={()=>{

      setCount(count-1)

      }}>Decrease</button>

    </div>

  )

}

export default App;

with contextApi

import { useContext, useState } from "react"

import { CountContext } from "./context";

function App(){

  const [count,setCount]=useState(0);

  return (

    <div>

      <CountContext.Provider value={count}>

      <Count count={count} setCount={setCount}/>

      </CountContext.Provider>

    </div>

  )

}

function Count({setCount}){

  return (

    <div>

      <CountRender/>

      <Button setCount={setCount}/>

    </div>

  )

}

function CountRender(){

  const count =useContext(CountContext);

  return (

    <div>

      {count}

    </div>

  )

}

function Button({setCount}){

  const count =useContext(CountContext);

  return(

    <div>

      <button onClick={()=>{

        setCount(count+1)

      }}>Increase</button>

      <button onClick={()=>{

        setCount(count-1)

      }}>Decrease</button>

    </div>

  )

}

export default App

==================================================================================