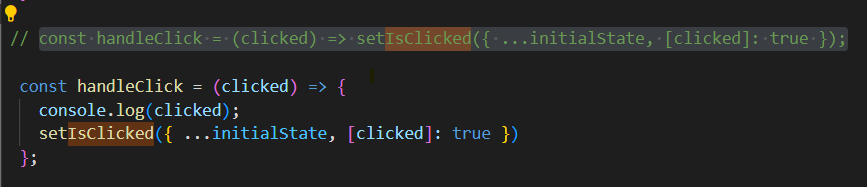
App.JS:

1.Have a Navbar ,SideBar(It typically has our NavLink or Link which helps us to Navigate to different components)

2.Have a Body where components are loaded based on routes.

In Your Contxt Api you can have different states(useState),your context can pass state updating functions to different components,Those components can call your state updating function based on requirement when useState in ContextAPI is loaded all the components using that state will be reevaluated

NavBar.js,ThemeSettings.js



Understand that when you want to have your key dynamic you can enclose it with square brackets

SideBar

When a particular Link is clicked NavLink Provides an isActive property which helps us to change the styles of the Link



<https://ultimatecourses.com/blog/active-navlink-inline-styles-with-react-router>

UseCOntext

import React, { createContext, useContext, useState } from 'react';

const StateContext = createContext();

const initialState = {

  chat: false,

  cart: false,

  userProfile: false,

  notification: false,

};

export const ContextProvider = ({ children }) => {

  const [screenSize, setScreenSize] = useState(undefined);

  const [currentColor, setCurrentColor] = useState('#03C9D7');

  const [currentMode, setCurrentMode] = useState('Light');

  const [themeSettings, setThemeSettings] = useState(false);

  const [activeMenu, setActiveMenu] = useState(true);

  const [isClicked, setIsClicked] = useState(initialState);

  const setMode = (e) => {

    setCurrentMode(e.target.value);

    localStorage.setItem('themeMode', e.target.value);

  };

  const setColor = (color) => {

    setCurrentColor(color);

    localStorage.setItem('colorMode', color);

  };

 // const handleClick = (clicked) => setIsClicked({ ...initialState, [clicked]: true });

  const handleClick = (clicked) => {

    console.log(clicked);

    setIsClicked({ ...initialState, [clicked]: true })

  };

  return (

    // eslint-disable-next-line react/jsx-no-constructed-context-values

    <StateContext.Provider value={{ currentColor, currentMode, activeMenu, screenSize, setScreenSize, handleClick, isClicked, initialState, setIsClicked, setActiveMenu, setCurrentColor, setCurrentMode, setMode, setColor, themeSettings, setThemeSettings }}>

      {children}

    </StateContext.Provider>

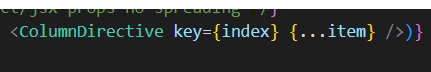
  );

};

export const useStateContext = () => useContext(StateContext);

Very cool pattern where with the help of useStateContext method you can extract all the context members

Ecommerce ,Orders,Employees,Customers its all about bringing data and looping through map function



Item object has keys and values and now they are passed as props to COlumnDirective,This is one shortcut way of sending props to a component

Calendar

Ref helps us to directly interact with DOM