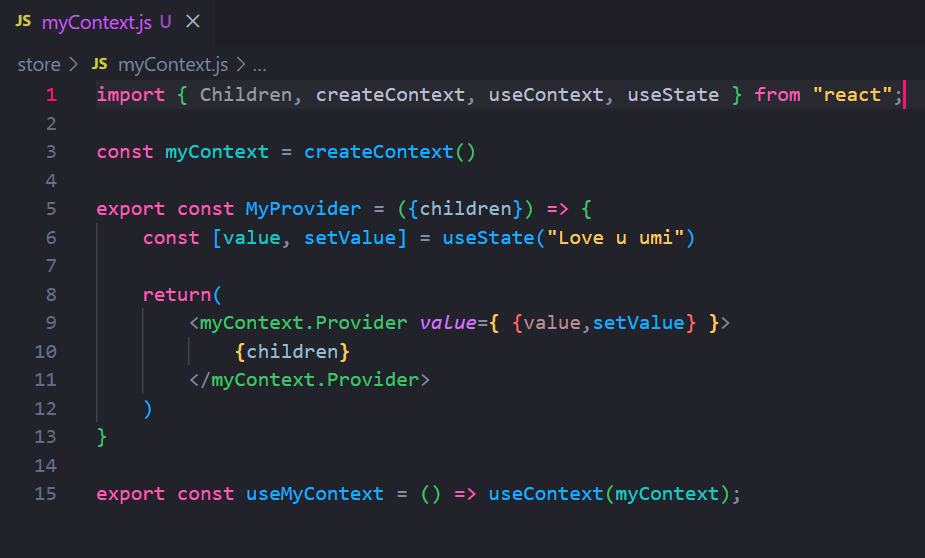
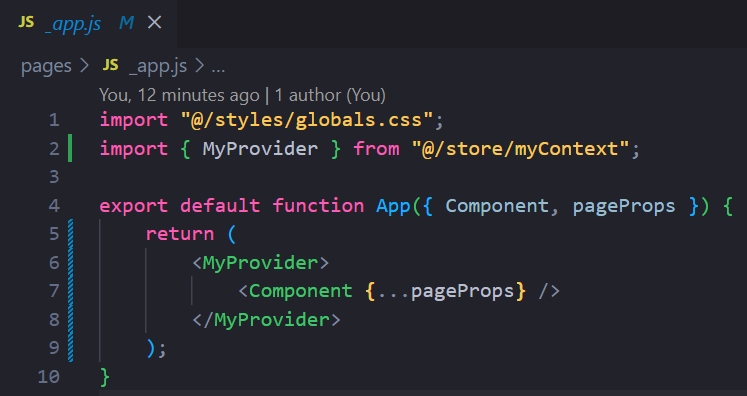
**Final ☺**

# **Basic Context :-**





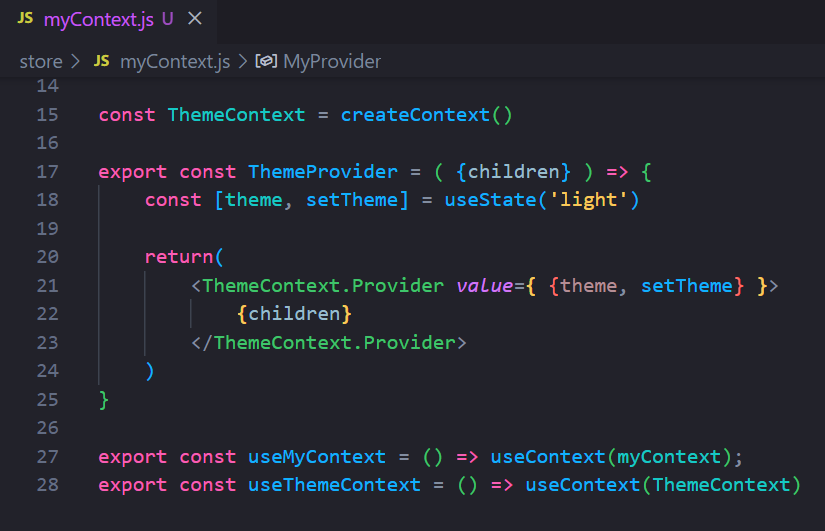




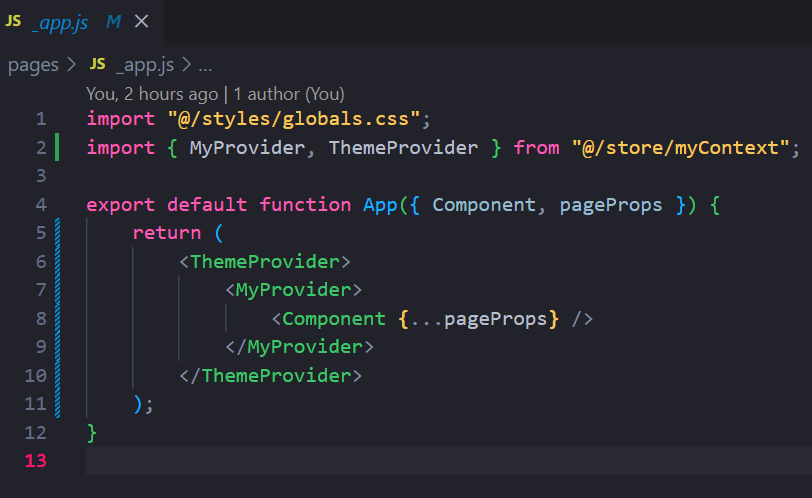




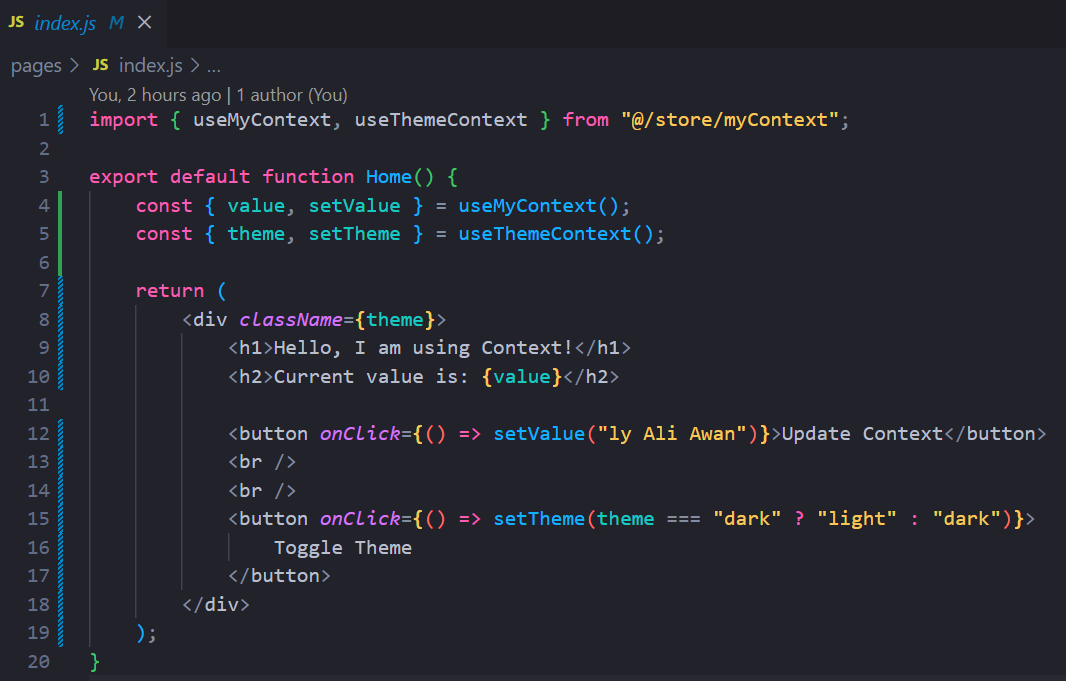
# **Theme Context :-**













**Context.js**

import { Children, createContext, useContext, useState } from "react";

const NotifContext = createContext({

    isNotifOpen : false,

    notifContent : null,

    openNotif : (content) => {},

    closeNotif : () => {}

})

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

    const [isNotifOpen, setNotif] = useState(false)

    const [notifContent, setNotifContent] = useState(null)

    const openHandler = (notifContent) => {

        setNotifContent(notifContent)

        setNotif(true)

    }

    const closeHandler = () => {

        setNotif(false)

    }

    const context = {

        isNotifOpen,

        notifContent,

        openNotif : openHandler,

        closeNotif : closeHandler

    }

    return(

        <NotifContext.Provider *value*=**{**context**}**>

**{**children**}**

        </NotifContext.Provider>

    )

}

const myContext = createContext()

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

    const [value, setValue] = useState("Love u umi")

    return(

        <myContext.Provider *value*=**{** {value,setValue} **}**>

**{**children**}**

        </myContext.Provider>

    )

}

const ThemeContext = createContext()

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

    const [theme, setTheme] = useState('light')

    return(

        <ThemeContext.Provider *value*=**{** {theme, setTheme} **}**>

**{**children**}**

        </ThemeContext.Provider>

    )

}

export const useMyContext = () => useContext(myContext);

export const useThemeContext = () => useContext(ThemeContext)

export const useNotifContext = () => useContext(NotifContext)

**App.js:**

import "@/styles/globals.css";

import { MyProvider, ThemeProvider, NotifProvider } from "@/store/myContext";

export default function App({ Component, pageProps }) {

    return (

        <NotifProvider>

            <ThemeProvider>

                <MyProvider>

                    <Component **{**...pageProps**}** />

                </MyProvider>

            </ThemeProvider>

        </NotifProvider>

    );

}

**Index.js:**

import { useMyContext, useThemeContext, useNotifContext } from "@/store/myContext";

export default function Home() {

    const { value, setValue } = useMyContext();

    const { theme, setTheme } = useThemeContext();

    const { isNotifOpen, notifContent, openNotif, closeNotif } = useNotifContext();

    return (

        <div *className*=**{**theme**}**>

            <h1>Hello, I am using Context!</h1>

            <h2>Current value is: **{**value**}**</h2>

            <button *onClick*=**{**() => setValue("ly Ali Awan")**}**>Update Context</button>

            <br />

            <br />

            <button *onClick*=**{**() => setTheme(theme === "dark" ? "light" : "dark")**}**>

                Toggle Theme

            </button>

            <br />

            <br />

            <button *onClick*=**{**() => openNotif("hello")**}**>Open</button>

**{**isNotifOpen && (

                <div

*style*=**{**{

                        position: "fixed", top: "50%",left: "50%", transform: "translate(-50%, -50%)", padding: "20px",

                        background: "rgba(0, 0, 0, 0.5)", border: "1px solid black", zIndex: 999 }**}**>

                    <p>**{**notifContent**}**</p>

                    <button *onClick*=**{**closeNotif**}**>Close notif</button>

                </div>

            )**}**

        </div>

    );

}

**Login via Next Auth**

import CredentialsProvider from 'next-auth/providers/credentials';

import NextAuth from 'next-auth';

import { MongoClient } from 'mongodb';

export default NextAuth({

    session:{jwt:true},

    providers:[

        CredentialsProvider({

            name:'credentials',

            async authorize(cred){

                const client=await MongoClient.connect("mongodb+srv://hira:123456.@cluster0.k927h.mongodb.net/coll?retryWrites=true&w=majority&appName=Cluster0")

                const db=client.db();

                const user= await db.collection("users").findOne({email:cred.email})

                if(!user)

                {

                    throw new Error('no user found');

                }

                else{

                    if(cred.password === user.password)

                    {

                        return {email:user.email}

                    }

                    else

                    {

                        throw new Error('wrong password');

                    }

                }

            }

        })

    ]

})

**Signup**

import { MongoClient } from "mongodb";

export default async function handler(req, res) {

    if(req.method==='POST')

    {

    const email=req.body.email;

    const pass=req.body.password;

    console.log(email,pass);

    const client=await MongoClient.connect("mongodb+srv://hira:123456.@cluster0.k927h.mongodb.net/coll?retryWrites=true&w=majority&appName=Cluster0")

    const db=client.db();

    db.collection("users").insertOne({

        email:email,

        password:pass

    })

    res.status(200).json({ message:"created user" });

  }

}

const result=await signIn('credentials',{redirect:false,email:email,password:pass})