

# Final 😊

## Basic Context :-

```
JS myContext.js U X
store > JS myContext.js > ...
1 import { Children, createContext, useContext, useState } from "react";
2
3 const myContext = createContext()
4
5 export const MyProvider = ({children}) => {
6   const [value, setValue] = useState("Love u umi")
7
8   return(
9     <myContext.Provider value={ {value,setValue} } >
10      {children}
11    </myContext.Provider>
12  )
13 }
14
15 export const useMyContext = () => useContext(myContext);
```

```
JS _app.js M X
pages > JS _app.js > ...
You, 12 minutes ago | 1 author (You)
1 import "@styles/globals.css";
2 import { MyProvider } from "@store/myContext";
3
4 export default function App({ Component, pageProps }) {
5   return (
6     <MyProvider>
7       <Component {...pageProps} />
8     </MyProvider>
9   );
10 }
```

```
JS index.js M X
pages > JS index.js > ...
You, 1 second ago | 1 author (You)
1 import { useMyContext } from "@store/myContext";
2
3 export default function Home() {
4   const { value, setValue } = useMyContext()
5
6   return (
7     <div>
8       <h1>Hello I am using Context saxy</h1>
9       <h2>Current value is: {value} </h2>
10
11       <button onClick={()=>setValue("Iy myContext)}> Update Context </button>
12     </div>
13   );
14 }
15
```

val update

## Theme Context :-

```
JS myContext.js U X
store > JS myContext.js > [MyProvider]
14
15 const ThemeContext = createContext()
16
17 export const ThemeProvider = ( {children} ) => {
18   const [theme, setTheme] = useState('light')
19
20   return(
21     <ThemeContext.Provider value={ {theme, setTheme} } >
22       {children}
23     </ThemeContext.Provider>
24   )
25 }
26
27 export const useMyContext = () => useContext(myContext);
28 export const useThemeContext = () => useContext(ThemeContext)
```

```
JS _app.js M X
pages > JS _app.js > ...
You, 2 hours ago | 1 author (You)
1 import "@styles/globals.css";
2 import { MyProvider, ThemeProvider } from "@store/myContext";
3
4 export default function App({ Component, pageProps }) {
5   return (
6     <ThemeProvider>
7       <MyProvider>
8         <Component {...pageProps} />
9       </MyProvider>
10    </ThemeProvider>
11  );
12 }
13
```

```
JS index.js M X
pages > JS index.js > ...
You, 2 hours ago | 1 author (You)
1 import { useMyContext, useThemeContext } from "@store/myContext";
2
3 export default function Home() {
4   const { value, setValue } = useMyContext();
5   const { theme, setTheme } = useThemeContext();
6
7   return (
8     <div className={theme}>
9       <h1>Hello, I am using Context!</h1>
10      <h2>Current value is: {value}</h2>
11
12      <button onClick={() => setValue("ly shah")}>Update Context</button>
13      <br />
14      <br />
15      <button onClick={() => setTheme(theme === "dark" ? "light" : "dark")}>
16        Toggle Theme
17      </button>
18    </div>
19  );
20 }
```

# 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("A u B")

  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 light dark")}>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})
```