

1. Create folder under components by name Context



2. Create file index.jsx under Context folder and add following code:

```
App.jsx U    JS ButtonContextComponent.jsx U    JS index.jsx U X    JS TextContextComponent.jsx U
src > context > JS index.jsx > ...
1  import { createContext } from "react";
2
3  export const GlobalContext = createContext(null);
4
5  export default function GlobalState({ children }) {
6    |   return <GlobalContext.Provider value={{}}>{children}</GlobalContext.Provider>;
7  }
```

3. Wrap the context in the root component Main.jsx:
Replace line `createRoot(document.getElementById("root")).render(<App/>);` with

```
import GlobalState from "./context/index.jsx";
createRoot(document.getElementById("root")).render(
  <GlobalState>
    <App />
  </GlobalState>
);
```

```
JS main.jsx U X    JS App.jsx U    JS ButtonContextComponent.jsx U
src > JS main.jsx
1  import { createRoot } from "react-dom/client";
2  import App from "./App.jsx";
3  import "./index.css";
4  import GlobalState from "./context/index.jsx";
5
6  createRoot(document.getElementById("root")).render(
7    <GlobalState>
8      <App />
9    </GlobalState>
10 );
```

4. Create 2 files in components folder as ContextButtonComponent.jsx and ContextTextComponent.jsx,

ContextTextComponent.jsx:

```
main.jsx U JS App.jsx U JS ButtonContextComponent.jsx U
src > components > JS TextContextComponent.jsx > ...
1 import React from "react";
2 import { useContext } from "react";
3 import { GlobalContext } from "../context";
4
5 export default function TextContextComponent() {
6
7   return (
8     <div>
9       <h2>
10        Incredible INDIA
11      </h2>
12    </div>
13  );
14 }
```

ContextButtonComponent.jsx:

```
main.jsx U JS App.jsx U JS ButtonContextComponent.jsx U X
src > components > JS ButtonContextComponent.jsx > ButtonContextCor
1 import React from "react";
2 import { useContext } from "react";
3 import { GlobalContext } from "../context";
4
5 export default function ButtonContextComponent() {
6   return (
7     <div>
8       <button>Change Theme</button>
9     </div>
10  );
11 }
```

5. Pass values in Context: Observe line 9 and 12 in following code

```
App.jsx U JS ButtonContextComponent.jsx U JS index.jsx U X JS TextContextComponent.jsx U
src > context > JS index.jsx > ...
1 import { useState } from "react";
2 import { createContext } from "react";
3
4 export const GlobalContext = createContext(null);
5
6 export default function GlobalState({ children }) {
7   const [theme, setTheme] = useState("light");
8   const handleChangeThemeOnButtonClick = () => {
9     setTheme(theme === "light" ? "dark" : "light");
10  };
11  return (
12    <GlobalContext.Provider value={{ theme, handleChangeThemeOnButtonClick }}>
13      {children}
14    </GlobalContext.Provider>
15  );
16 }
```

6. Update ContextTextComponent.jsx component:

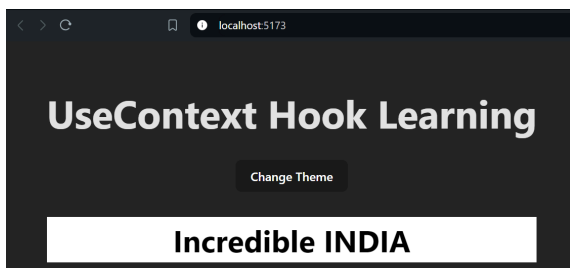
```
App.jsx U JS ButtonContextComponent.jsx U JS index.jsx U JS TextContextComponent.jsx U
src > components > JS TextContextComponent.jsx > TextContextComponent
1 import React from "react";
2 import { useContext } from "react";
3 import { GlobalContext } from "../context";
4
5 export default function TextContextComponent() {
6   const { theme } = useContext(GlobalContext);
7   return (
8     <div>
9       <h2>
10         style={{
11           fontSize: theme === "light" ? "40px" : "70px",
12           color: theme === "light" ? "black" : "white",
13           backgroundColor: theme === "light" ? "white" : "black",
14         }}
15       >
16         Incredible INDIA
17     </h2>
18   </div>
19 );
20 }
```

7. Update ContextButtonComponent.jsx component:

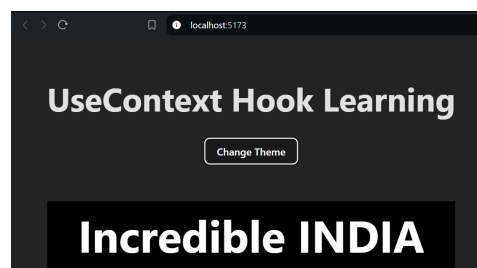
```
App.jsx U JS ButtonContextComponent.jsx U X JS index.jsx U JS TextContextComponent.jsx U
src > components > JS ButtonContextComponent.jsx > ...
1 import React from "react";
2 import { useContext } from "react";
3 import { GlobalContext } from "../context";
4
5 export default function ButtonContextComponent() {
6   const { handleChangeThemeOnButtonClick } = useContext(GlobalContext);
7   return (
8     <div>
9       <button onClick={handleChangeThemeOnButtonClick}>Change Theme</button>
10     </div>
11 );
12 }
```

8. Result:

Before button click:



On button click:



Code:

Main.jsx:

```
import { createRoot } from "react-dom/client";
import App from "./App.jsx";
import "./index.css";
import GlobalState from "./context/index.jsx";

createRoot(document.getElementById("root")).render(
  <GlobalState>
    <App />
  </GlobalState>
);
```

App.jsx:

```
import "./App.css";
import ButtonContextComponent from "./components/ButtonContextComponent";
import TextContextComponent from "./components/TextContextComponent";

function App() {
  return (
    <>
      <h1>UseContext Hook Learning</h1>
      <ButtonContextComponent />
      <TextContextComponent />
    </>
  );
}

export default App;
```

src\components\ButtonContextComponent.jsx:

```
import React from "react";
import { useContext } from "react";
import { GlobalContext } from "../context";

export default function ButtonContextComponent() {
  const { handleChangeThemeOnButtonClick } = useContext(GlobalContext);
  return (
    <div>
      <button onClick={handleChangeThemeOnButtonClick}>Change Theme</button>
    </div>
  );
}
```

src\components\TextContextComponent.jsx:

```
import React from "react";
import { useContext } from "react";
import { GlobalContext } from "../context";

export default function TextContextComponent() {
  const { theme } = useContext(GlobalContext);
  return (
    <div>
      <h2
        style={{
          fontSize: theme === "light" ? "40px" : "70px",
          color: theme === "light" ? "black" : "white",
          backgroundColor: theme === "light" ? "white" : "black",
        }}
      >
        Incredible INDIA
      </h2>
    </div>
  );
}
```

src\context\index.jsx:

```
import { useState } from "react";
import { createContext } from "react";

export const GlobalContext = createContext(null);

export default function GlobalState({ children }) {
  const [theme, setTheme] = useState("light");
  const handleChangeThemeOnButtonClick = () => {
    setTheme(theme === "light" ? "dark" : "light");
  };
  return (
    <GlobalContext.Provider value={{ theme, handleChangeThemeOnButtonClick }}>
      {children}
    </GlobalContext.Provider>
  );
}
```