Creating "Enable Dark Mode" Button Using useState Hook | Complete React Course in Hindi #10

In this tutorial, we will be understanding the state by creating a new component called "**about**". So without further ado let's begin:

Create about.js:

Create "about.js" in your components folder and by using snippet 'rfc' we can easily generate a function-based component of React in our "about.js". Now we will enter some components, like the Accordion component, of bootstrap in return(). Earlier we have learned how to copy components from bootstrap. We are copying and Editing the code of the accordion component in our About.

In app.js:

Now we will render this "about.js" in the Home section of our application with the help of "app.js". So, comment out the "textform.js" and add the "about.js" file.

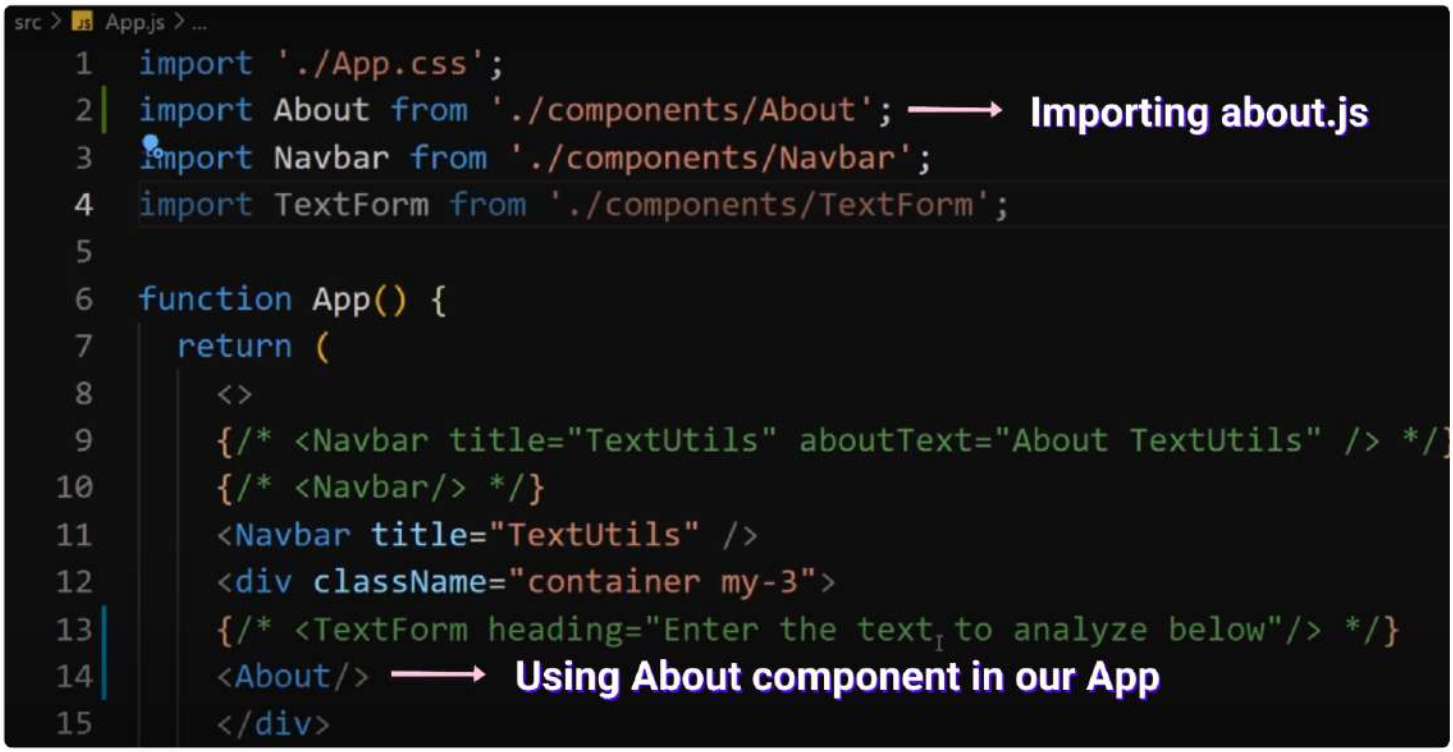

Figure1.1: Rendering About.js

Now we would like to add a button to our app. On clicking this button our app must enable or disable the dark mode. To do so we would be assigning function to this button later on.

Adding button:

To add button we will use:

```
<button type="button" className="btn btn-primary">Enable Dark Mode</button>
```

What our function must do?

On invoking the function, by clicking the button, we want that dark mode to get enabled and the text in our button to change from 'Enable dark mode' to 'Enable Light Mode' and vice versa. To do so follow the steps:
Firstly, we will add the style of dark mode to our application and then interchange it with light mode on invoking the function.

Adding Styles of Dark mode:

```
style = {mystyle}
```

where ‘myStyle’ is an object which we are using as a style variable. So let’s create a JavaScript object with the name of ‘myStyle’.

```
let myStyle = {
    color: 'white',
    backgroundColor: 'black'
}
```

Remember: In JS everything is in camel case

Using this Style:

We are using this style object in our three accordion buttons and body. After applying styles our app is appearing something like this:

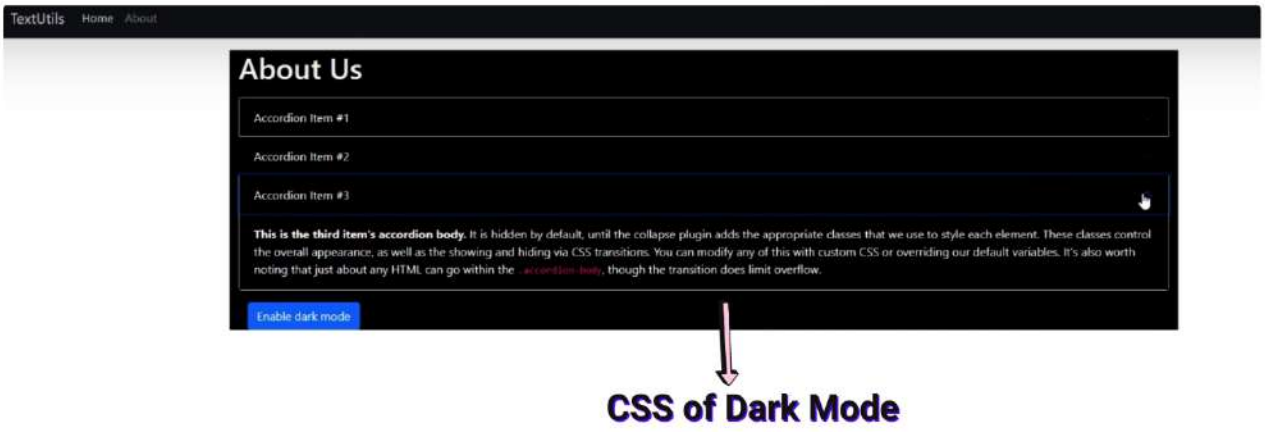

Figure1.2: Our Application

Style as a State variable:

Now we make this myStyle object as a state variable so we can use it to change the style of our application by clicking on the button. We would be using two-state variables:

1. **myStyle as a State variable**

```
const[mystyle, setMystyle] = useState({
    color: 'black',
    backgroundColor: 'white'
})
```

2. **Button Text as a State variable**

```
const[btnText, SetBtnText] = useState("Enable Dark mode")
```

Initially, these are the State of our application and we would replace them with setMyStyle and setBtnText.

Function:

We would like to change the style of our application on invoking a function, Suppose toogleStyle.
For that Create a function toggleStyle and use the if-else statement to define conditions for change.
Toggle style function :

```
const toogleStyle = () => {
    if (myStyle.color === "black") {
        setMyStyle({
            color: "white",
            backgroundColor: "black",
            border: "1px solid white",
        })
        setBtnText("Enable Light Mode")
    }
}
```

Explanation: The above code means if the color of text of myStyle object is black(light mode) then on clicking the button change myStyle with SetMyStyle, which is replace the color of text with white and background color with black and enable border (that is Dark Mode). We have also replaced the text of the button with setBtnText, which is 'enable light mode'.

```
else {
    setMyStyle({
        color: 'black',
        backgroundColor: 'white'
    })
    setBtnText("Enable Dark Mode")
}
```

Explanation: If the above-mentioned conditions are false then replace myStyle with setMyStyle, this time, which is text color being black and background color being white, and replace the text in button with “Enable Dark mode”.Actually, this is the same as the initial condition of our text, which is light mode is being enabled.
Hence we have successfully created our function and would like to assign it to our button.

Code of button:

```
<button onClick={toggleStyle} type="button" className="btn btn-primary"> {btntext}</button>
```

Hence, we have successfully created a button to enable dark mode in our Application.

Summary:

What we have done in our about.js:

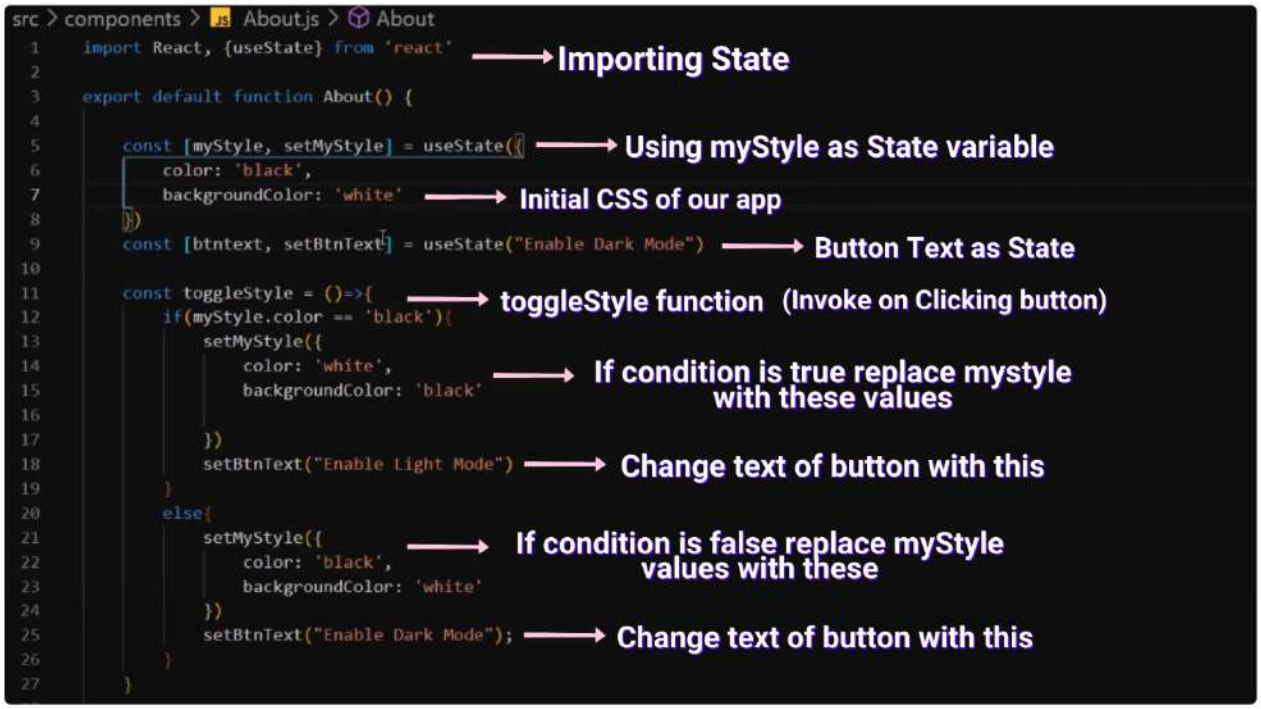

Figure1.3: A short summary

We have rendered about.js in the Home section just for the demo. So, Comment out the about component and again use text form component. Later on, we would be rendering this about.js to the About section of our application with the help of a router.