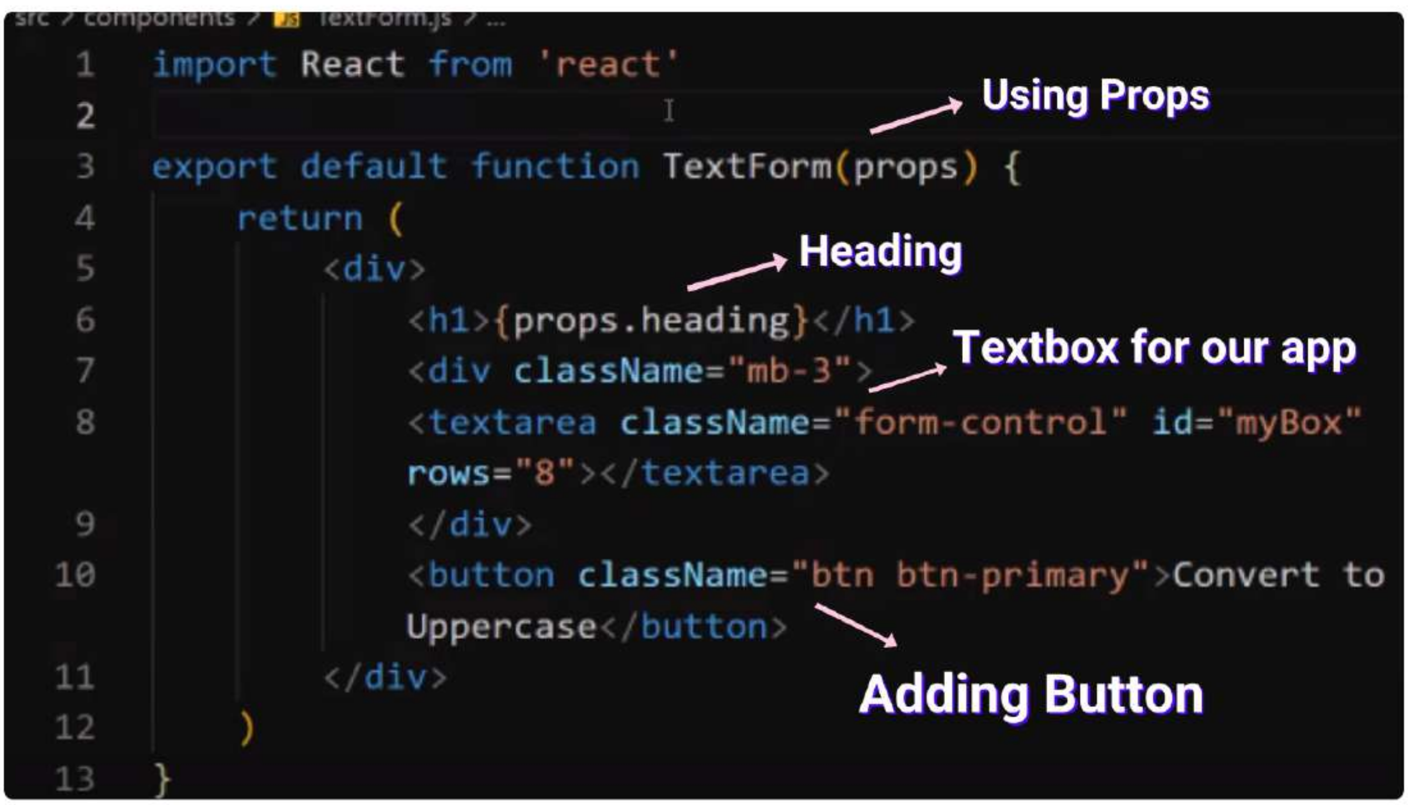
Understanding State & Handling Events in React | Complete React Course in Hindi #7

Till now we have created the navbar of our application and have learned about props. In this tutorial, we will understand State and Handling events. So, let's continue our journey of React:

Creating Textbox:

We are creating a new component named Textform.js in our Components folder. By using 'rfc' snippet you can generate a function-based component in "textform.js". Now inside this function, we will render our desired textbox. For example, we are using the textbox from Bootstrap. While copying anything from Bootstrap, Make sure to do the following changes:

- Close those tags which don't have a closing tag.
- Replace the 'class' keyword with 'className'.
- The code must be in one tag or use a JSX fragment.



src > components > textForm.js > ...

```
import React from 'react'

export default function TextForm(props) {
    return (
        <div>
            <h1>{props.heading}</h1>
            <div className="mb-3">
            <textarea className="form-control" id="myBox"
            rows="8"></textarea>
            </div>
            <button className="btn btn-primary">Convert to
            Uppercase</button>
        </div>
    )
}
```

Figure1.1: Our Textbox in textform.js

We would also like to attach some buttons in our app which will fire an event on clicking.

In textform.js:

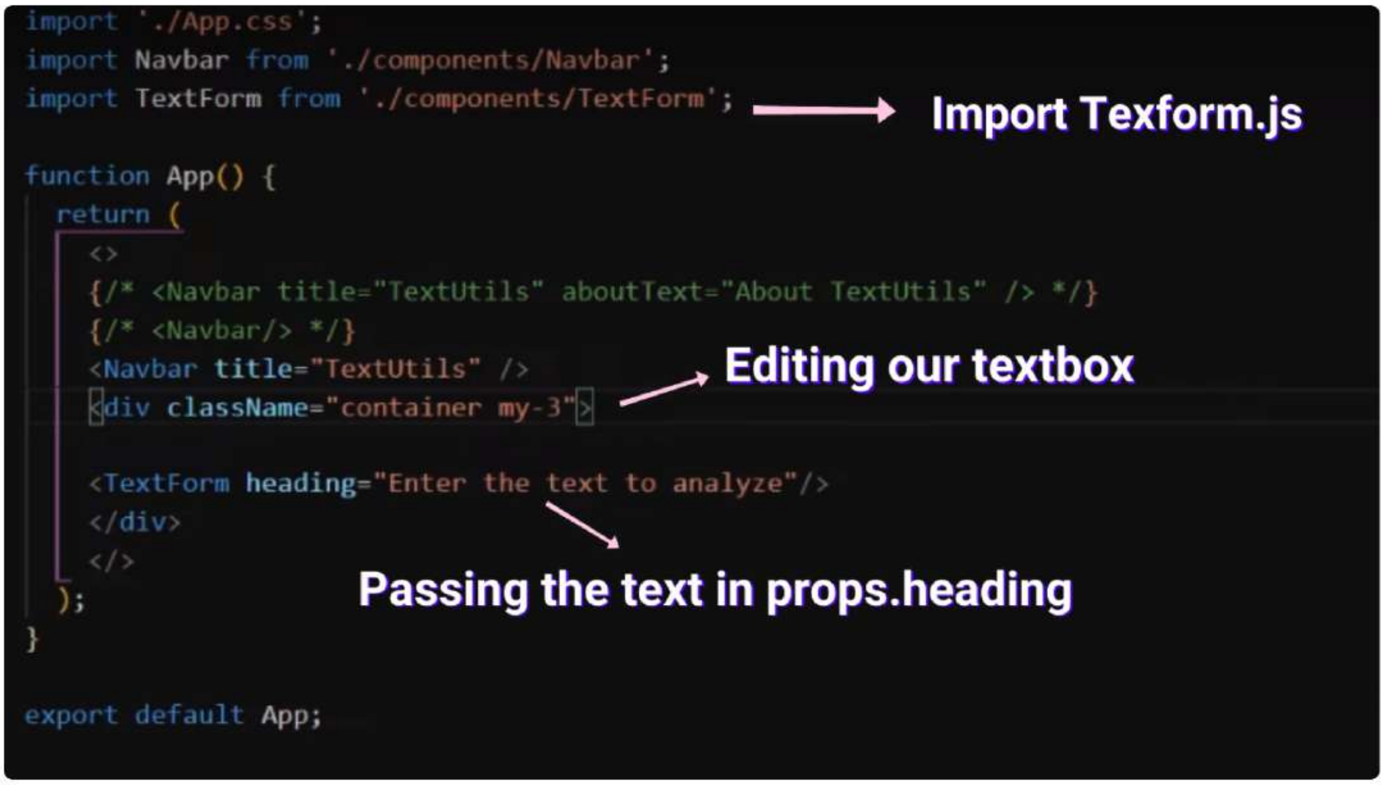
Adding Buttons: Creating a 'Convert to Uppercase button'. Adding the following code in textform.js

```
<button className="btn btn-primary">Convert to Uppercase</button>
```

Now we have created our textform. So let's render it in our application with the help of app.js.

In app.js:

We will be importing textform.js in app.js and return it inside our app by using a function-based component.



```
import './App.css';
import Navbar from './components/Navbar';
import TextForm from './components/TextForm';

function App() {
  return (
    <>
    {/* <Navbar title="TextUtils" aboutText="About TextUtils" /> */}
    {/* <Navbar/> */}
    <Navbar title="TextUtils" />
    <div className="container my-3">

    <TextForm heading="Enter the text to analyze"/>
    </div>
    </>
  );
}

export default App;
```

Figure1.2: Importing and Using Textform

Here, we have rendered the textform in our application and have successfully passed a heading in it, using props.
Our app is looking something like this,

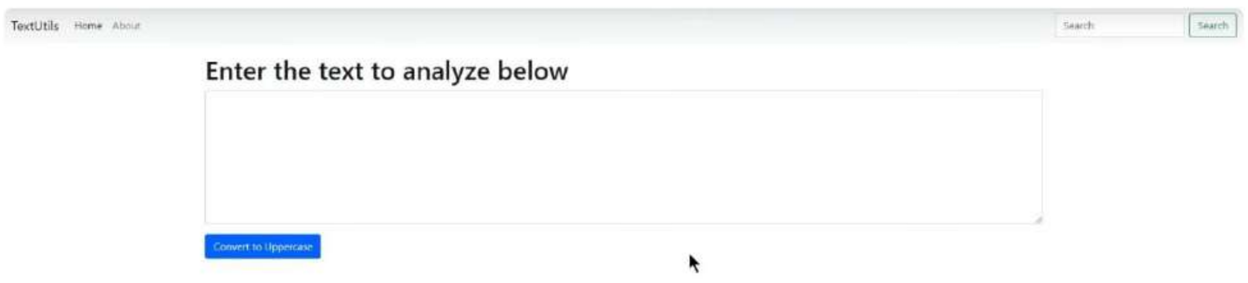


Figure: Our Text Editor application

Understanding State:

State: A state depicts the local information of a Component. For example: If we write something in our text box it will be the state of our object for that specific point in time. The state is the changeable variable of our app. Whenever the state object changes, the component re-renders itself.

Hook in React:

Hook allows you to use state and other react features with a function-based component, that is without writing a class.
useState: It is the type of hook in react which allows us to use state variables in the function-based components.

Steps to use state are:
Firstly, import the state to your react app by using the below command in app.js.

```
import React, { useState } from 'react';
```

To use state, firstly enter the following code inside your function-based component.

```
const [count, setCount] = useState(0);
```

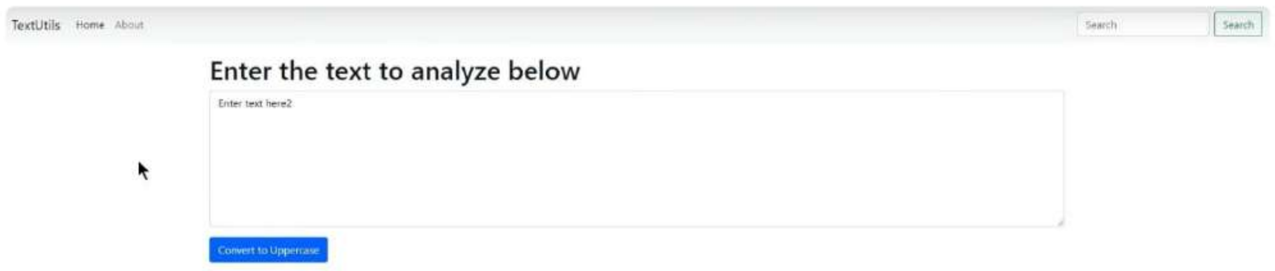
The count is the state variable, which contains the current state value. In our case it is text. So, we will replace count with text.

```
const [text, setText] = useState('Enter Text here2');
```

Here, The value of our text is 'Enter text here2'. Let's pass this value in our textbox and make sure to use the 'Onchange' event to enable text entry in your textbox. To do so change the code of your text area to:

```
<textarea className='form-Control' value={text} Onchange={handleOnchange} id='mybox' rows= '8
```

Result:



Enter the text to analyze below

Enter text here2

Figure1.4: Used State to enter text

Changing Values using SetText:

Now, To change the value in the textbox, we can use the set text function:

```
setText("Your Text here")
```

Remember you cannot change the value of the variable in React like text = “New Text”;

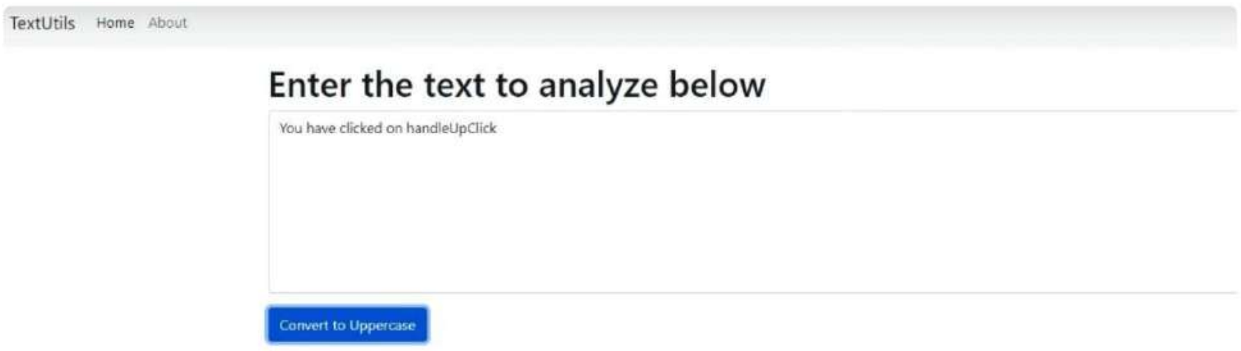
Button:

Now let's make our button functional by assigning a function to it. We are creating a function named handleUpClick, which on being clicked will change the text of the textbox to "You have clicked on handleUpClick".

Our function:

```
const handleUpClick = () => {
    console.log("Uppercase was clicked");
    SetText("You have clicked on handleUpClick")
}
```

Here, is the image showing you the results of the above function:



You have clicked on handleUpClick

Figure1.5: Displaying the result

You will notice that you cannot add more text in the textbox. This is due to the reason as we didn't assign a function to the 'Onchange' event.

Creating new function handleOnChange:

```
const handleOnChange = (event) => {
    console.log("On change");
    setText(event.target.value);
}
```

Result: After using the above function you will find out that you are able to add more text to your textbox.

Changing the text to Uppercase:

Now, we want that on clicking the button it changes the existing text value of the textbox to uppercase. So let's make changes to our existing handleUpClick function:

```
const handleUpClick = () => {
    console.log("Uppercase was clicked");
    let newText = text.toUpperCase();
    setText("new text")
}
```

Result: After this, you will be able to enter text in your document and by pressing the 'Convert to uppercase' you will be able to convert your text in uppercase.