**App.js**

import './App.css';

import ClickCounter2 from "./Component2/ClickCounter2"

import ClickCounter3 from "./Component2/ClickCounter3";

function App() {

  return (

    <div className="App">

      <ClickCounter2/>

      <ClickCounter3/>

    </div>

  );

}

export default App;

**ClickCounter2.js**

import React, { Component } from 'react'

class ClickCounter2 extends Component {

    render() {

        const { countFromRenderProp,incrementMethodCall } = this.props;

        return (

            <div>

                <h1>Current Count :: {countFromRenderProp}</h1>

                <button onClick={incrementMethodCall}>Click Here</button>

            </div>

        )

    }

}

export default ClickCounter2

**ClickCounter3.js**

import React, { Component } from "react";

class ClickCounter3 extends Component {

  render() {

    const { countFromRenderProp,incrementMethodCall } = this.props;

    return (

      <div>

        <h1 onMouseOver={incrementMethodCall}>Current Count :: {countFromRenderProp}</h1>

      </div>

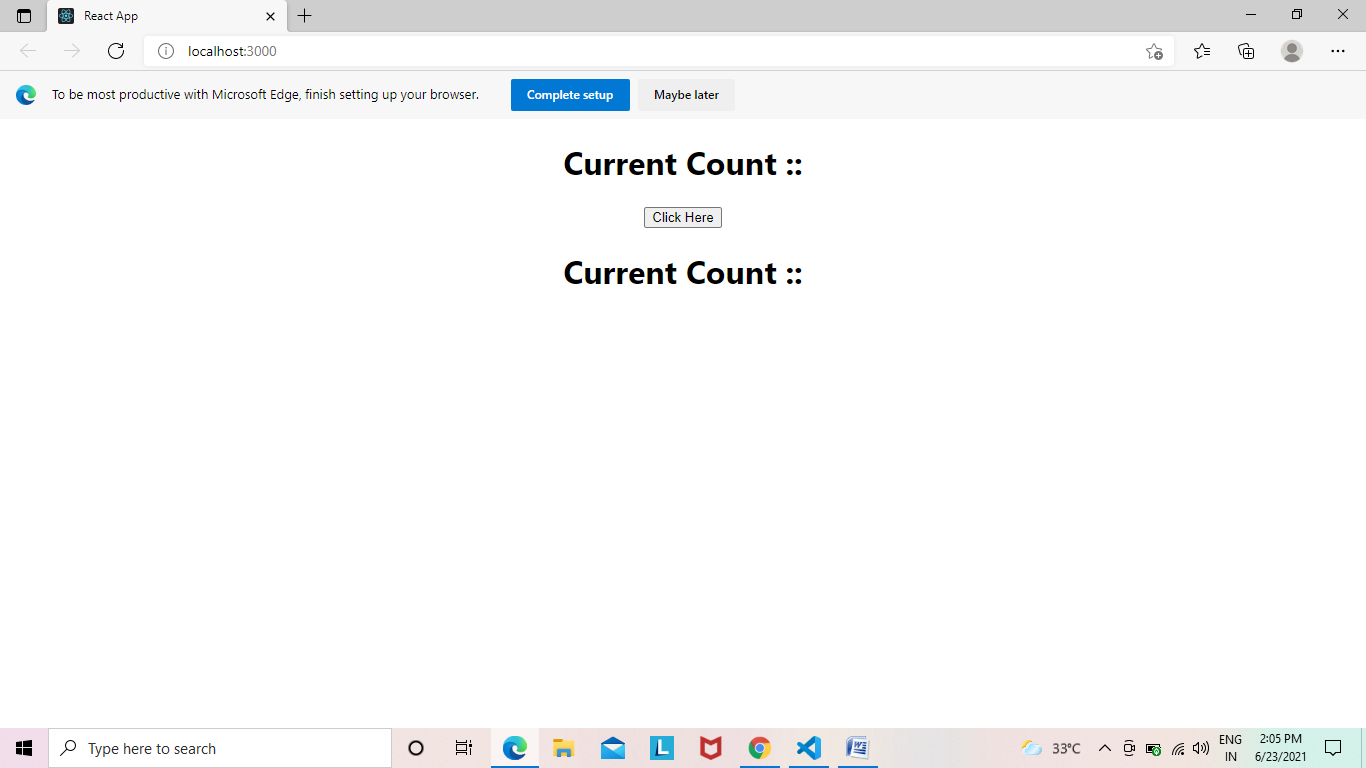
    );

  }

}

export default ClickCounter3;

**Output:**

****

**App.js**

import './App.css';

import ClickCounter2 from "./Component2/ClickCounter2"

import ClickCounter3 from "./Component2/ClickCounter3";

import RenderProps from "./Component2/RenderProps";

function App() {

  return (

    <div className="App">

      <RenderProps

        name="Hello"

        render={(count,increment)=>(<ClickCounter2 countFromRenderProp={count} incrementMethodCall={increment}/>)}

      />

      <RenderProps

        render={(count, increment) => (

          <ClickCounter3

            countFromRenderProp={count}

            incrementMethodCall={increment}

          />

        )}/>

    </div>

  );

}

export default App;

**RenderProps.js**

import React, { Component } from "react";

class RenderProps extends Component {

  constructor(props) {

    super(props);

    this.state = {

      count: 0,

    };

  }

  increment = () => {

    this.setState((prevState) => {

      return { count: prevState.count + 1 };

    });

  };

  render() {

    const { name,render } = this.props;

    return <div>{this.props.render(this.state.count,this.increment)}</div>;

    return (

      <div>

        <h1>{name}</h1>

        {render(this.state.count,this.increment)}

      </div>

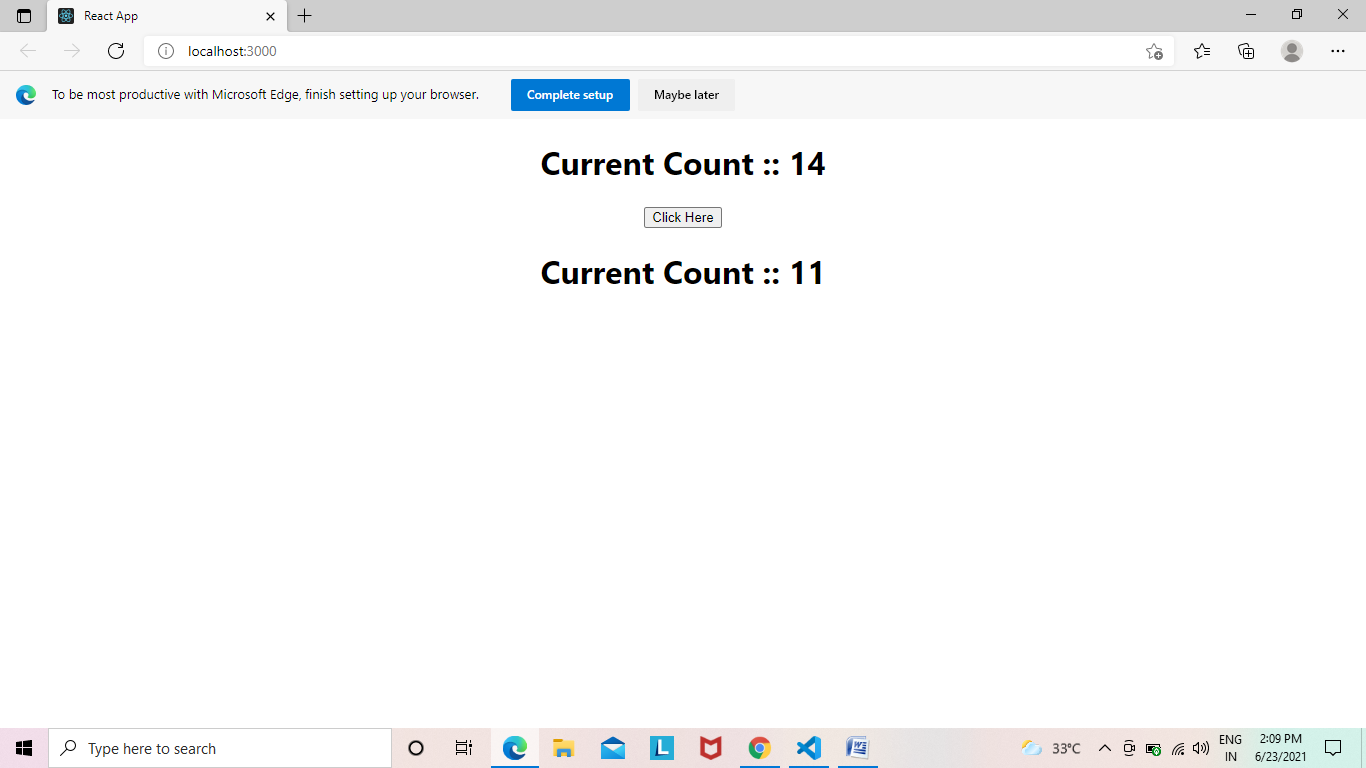
    );

  }

}

export default RenderProps;

**Output**

****