There are 4th approach to bind event handler in React :

**First Approach : Bind in render**

* Binding event in the render with arrow function but this approach isn’t good some cases .
* You don’t wantb to use because of performance implication

**Practical :**

import React,{Component} from "react";

class EventBindingInRender extends Component{

//------Constructor Set Message--------

    constructor(){

        super()

        this.state={

            message:'Hello Subhi'

        }

    }

//------Define Method-------

    clickHandler(){

        this.setState({

            message: 'Byeeee Subhi'

        })

    }

    render (){

        return(

            <div>

                    <div>{this.state.message}</div>

                      {/\*First approach to binding in render section to the button\*/}

                    <button onClick={this.clickHandler.bind(this)}>click</button>

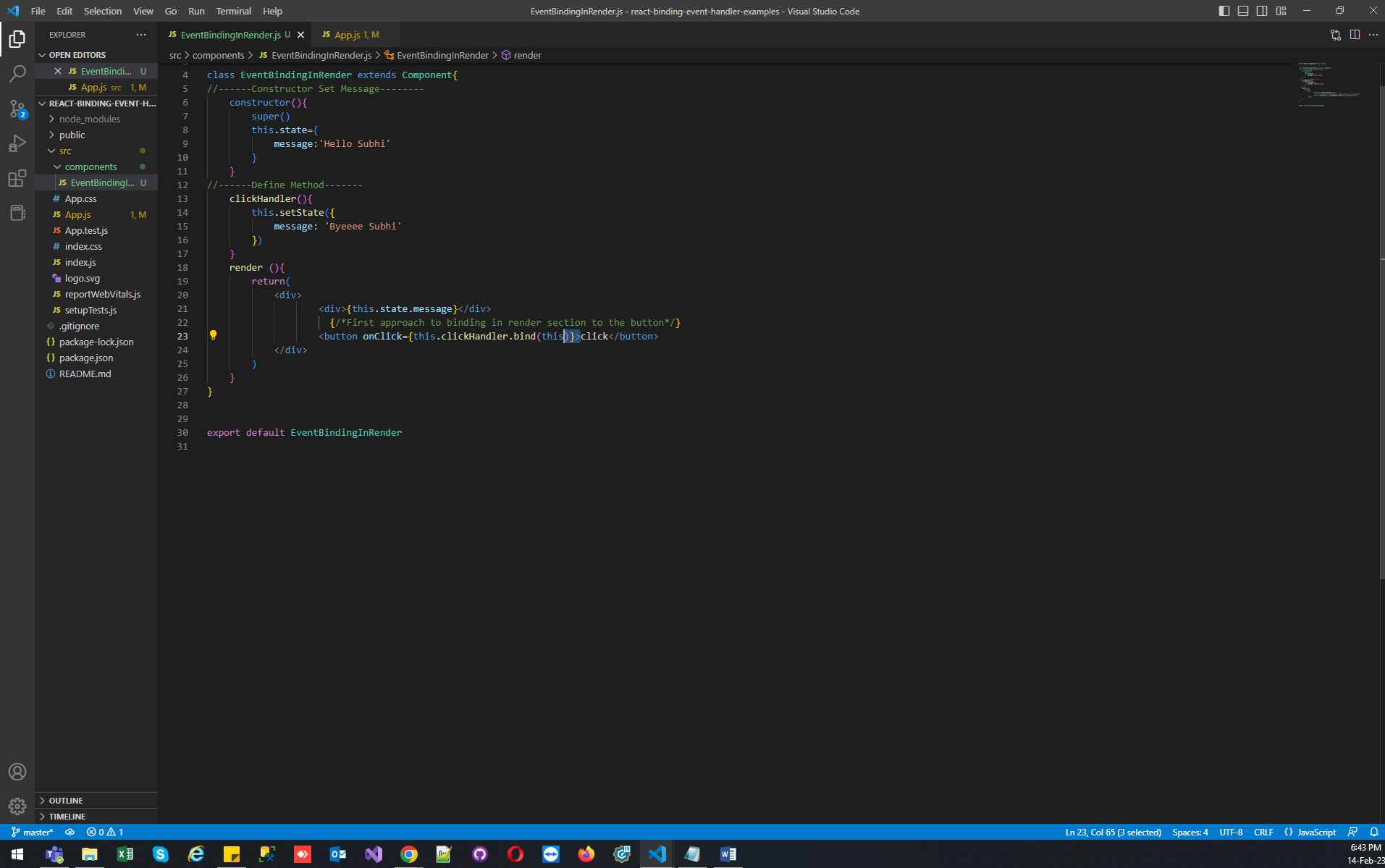
            </div>

        )

    }

}

export default EventBindingInRender



**Second Approach : Bind in render with arrow function**

* Binding event in the render but this approach isn’t good some cases ..
* It is also similar to the approach first, one is probably the easiest way to pass parameters if your code doesn’t involve three rendering nested children components this approach is still a viable option.

**Practical :**

import React,{ Component } from "react";

class EventBindingInRenderWithArrowFunction extends Component{

//Set the default message

constructor(props){

    super(props)

    this.state={

        message:'Hello Subhishek'

    }

}

 //Define Method

 clickHandler()

    {

    this.setState({

        message:'good bye Subhishek!!!'

    })

    }

    render (){

        return(

            <div>

                <div>{this.state.message}</div>

                {/\*2nd approach to binding in render with arrow function\*/}

                <button onClick={()=>this.clickHandler()}>click</button>

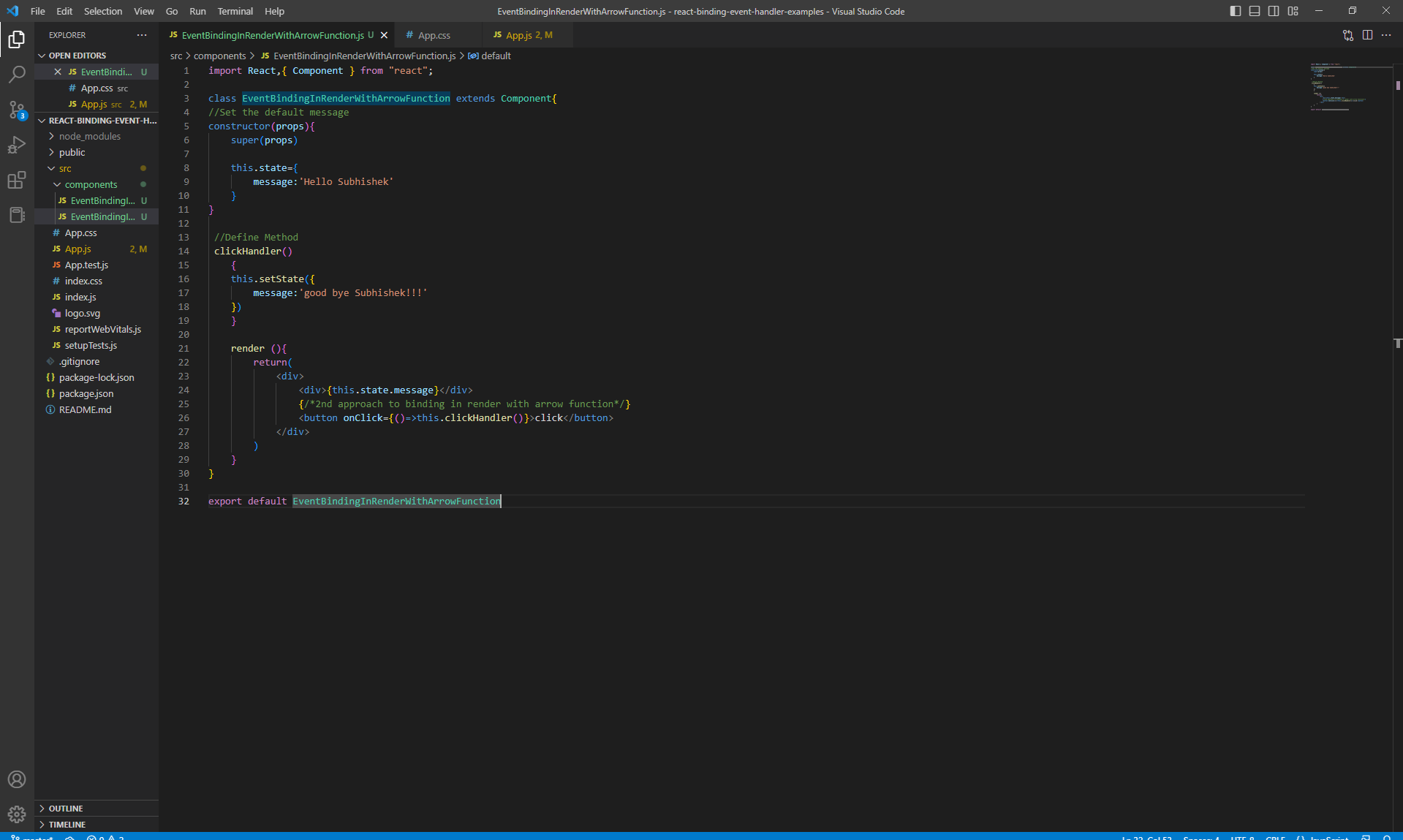
            </div>

        )

    }

}

export default EventBindingInRenderWithArrowFunction



**Third Approach : Class constructor**

* Binding event in the class constructor. this approach also suggest react official documentation.

**Practical :**

import React,{ Component } from "react";

class EventBindingWithClassConstructors extends Component{

//Set the default message

constructor(props){

    super(props)

    this.state={

        message:'Hello Subhisheks'

    }

    //3rd approach to bind event with class Constructor

    this.clickHandler=this.clickHandler.bind(this)

}

 //Define Method

 clickHandler()

    {

    this.setState({

        message:'good bye Subhishek!!!'

    })

    }

    render (){

        return(

            <div>

                <div>{this.state.message}</div>

               {/\*3rd approach to bind event with class Constructor ---React official site\*/}

             <button onClick={this.clickHandler} >Click </button>

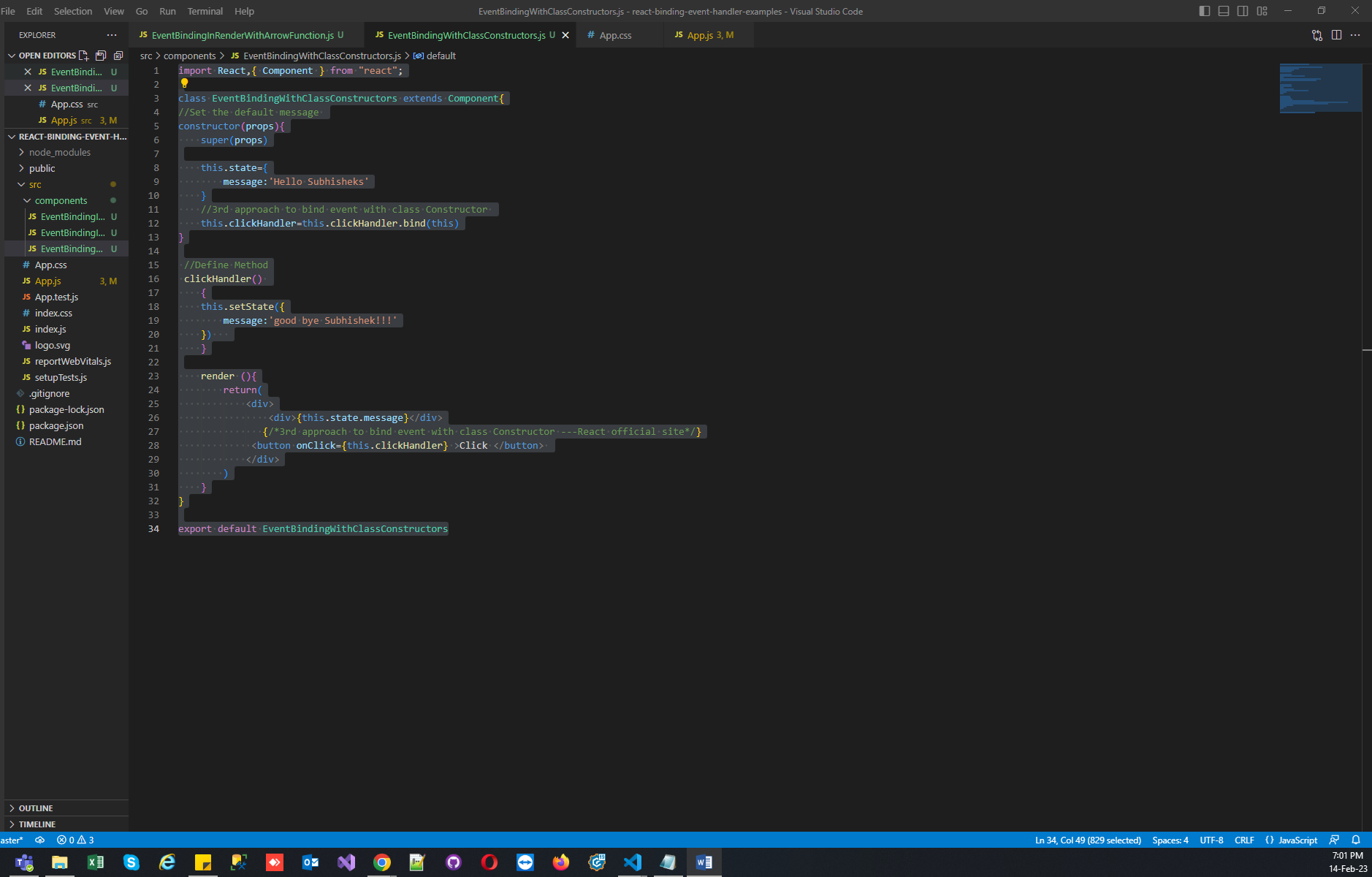
            </div>

        )

    }

}

export default EventBindingWithClassConstructors



**Fourth Approach : Class property**

* Binding event in the class property as arrow function. this approach also suggest react official documentation.
* React app doesn’t support the class library property approach so there is nothing wrong using this approach. In fact when the class property approach becomes an accepted feature it would be the go to approach

**Practical :**

import React,{ Component } from "react";

class EventBindingWithClassProperty extends Component{

//Set the default message

constructor(props){

    super(props)

    this.state={

        message:'Hello Subhishek'

    }

}

 //Define Method

 clickHandler=() => {

    this.setState({

        message:'good byee shukla!!!'

    })

}

    render (){

        return(

            <div>

                <div>{this.state.message}</div>

                {/\*4th approach to bind event class property as arrow function\*/}

                <button onClick={this.clickHandler} >Click </button>

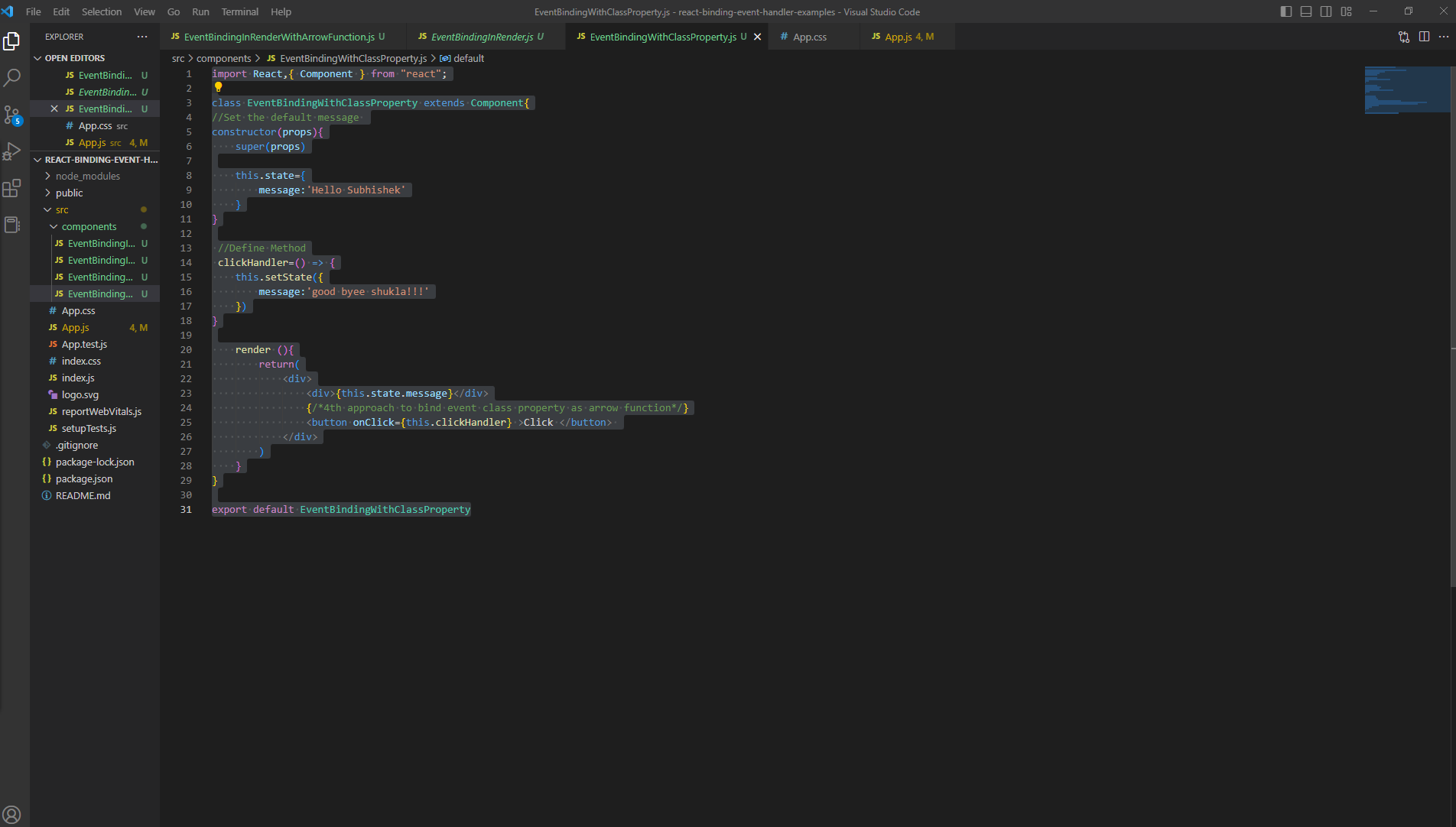
            </div>

        )

    }

}

export default EventBindingWithClassProperty



All In One :

import React, { Component } from 'react'

 class EventBind extends Component {

    constructor(props) {

      super(props)

      this.state = {

            message:'Hello'

      }

      //3rd approach to bind event with class Constructor ---React official site

      // this.clickHandler=this.clickHandler.bind(this)

    }

    //Define Method

//  clickHandler()

//     {

//     this.setState({

//         message:'good bye!!!'

//     })

//     }

//class property as arrow function

    clickHandler1=() => {

        this.setState({

            message:'good byee!!!'

        })

    }

  render() {

    return (

      <div>

        <div>{this.state.message}</div>

        {/\*First approach to binding in render\*/}

        {/\* <button onClick={this.clickHandler.bind(this)}>Click</button> \*/}

        {/\*2nd approach to binding in render with arrow function\*/}

        {/\* <button onClick={()=>this.clickHandler()} >Click </button>  \*/}

        {/\*3rd approach to bind event with class Constructor ---React official site\*/}

        {/\* <button onClick={this.clickHandler} >Click </button>  \*/}

        {/\*4th approach to bind event class property as arrow function\*/}

        <button onClick={this.clickHandler1} >Click </button>

      </div>

    )

  }

}

export default EventBind

