* Its is also used to same functionality to multiple component.
* Create example count on button click, hover and input .

Example :

Create Component for Hover count :

import React, { Component } from 'react'

export class HoverCounterex extends Component {

  constructor(props) {

    super(props)

    this.state = {

       count:0

    }

  }

  incrementCount=()=>{

      this.setState(prevState=>{

          return { count : prevState.count +1}

      })

    }

  render() {

    const { count } = this.state

    return <label onMouseOver={this.incrementCount}>Hovered {count} times</label>

  }

}

export default HoverCounterex

Create Component for Click Count

import React, { Component } from 'react'

 class ClickCounterex extends Component {

    constructor(props) {

      super(props)

      this.state = {

         count:0

      }

    }

    incrementCount=()=>{

        this.setState(prevState=>{

            return { count : prevState.count +1}

        })

    }

  render() {

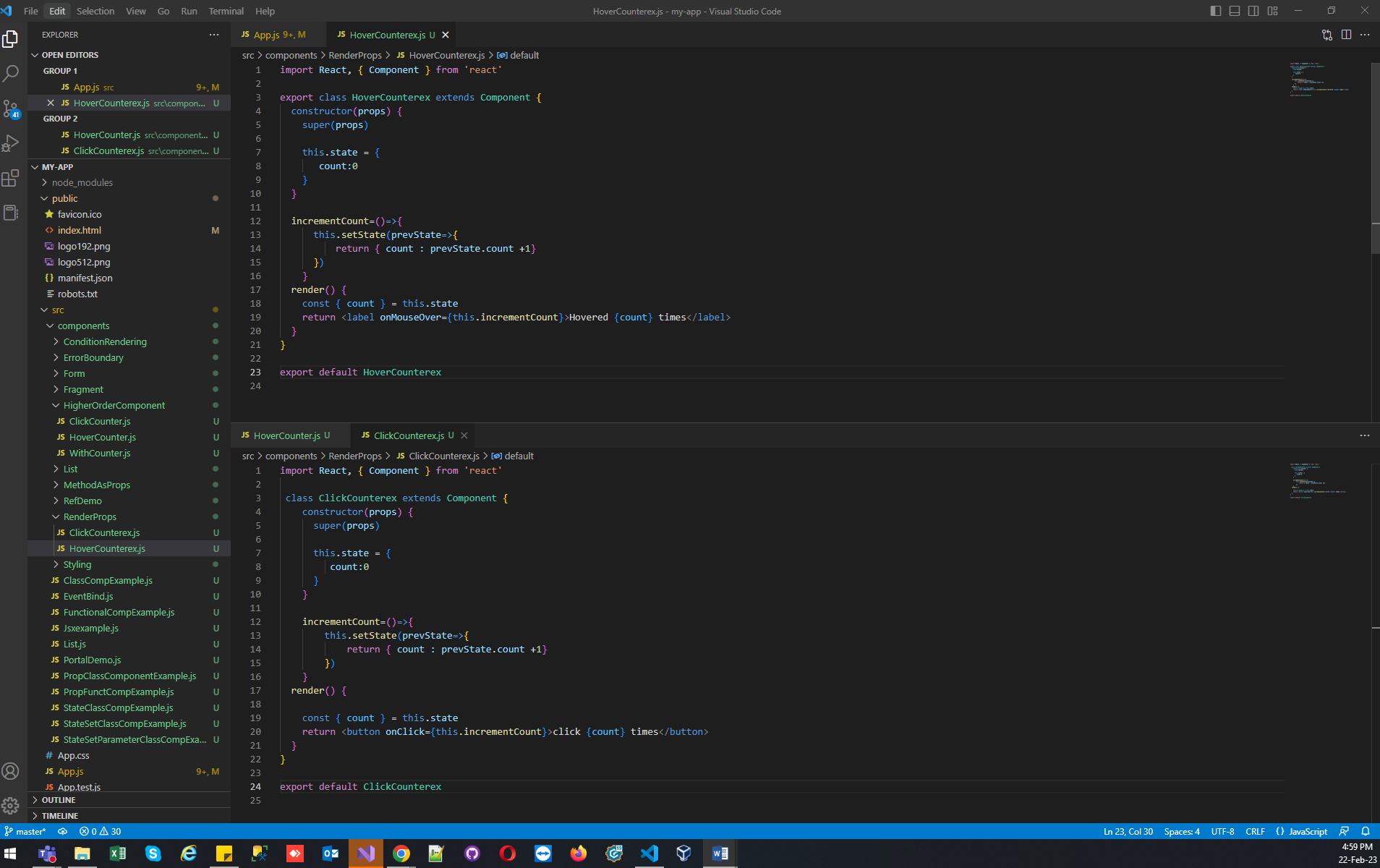
    const { count } = this.state

    return <button onClick={this.incrementCount}>click {count} times</button>

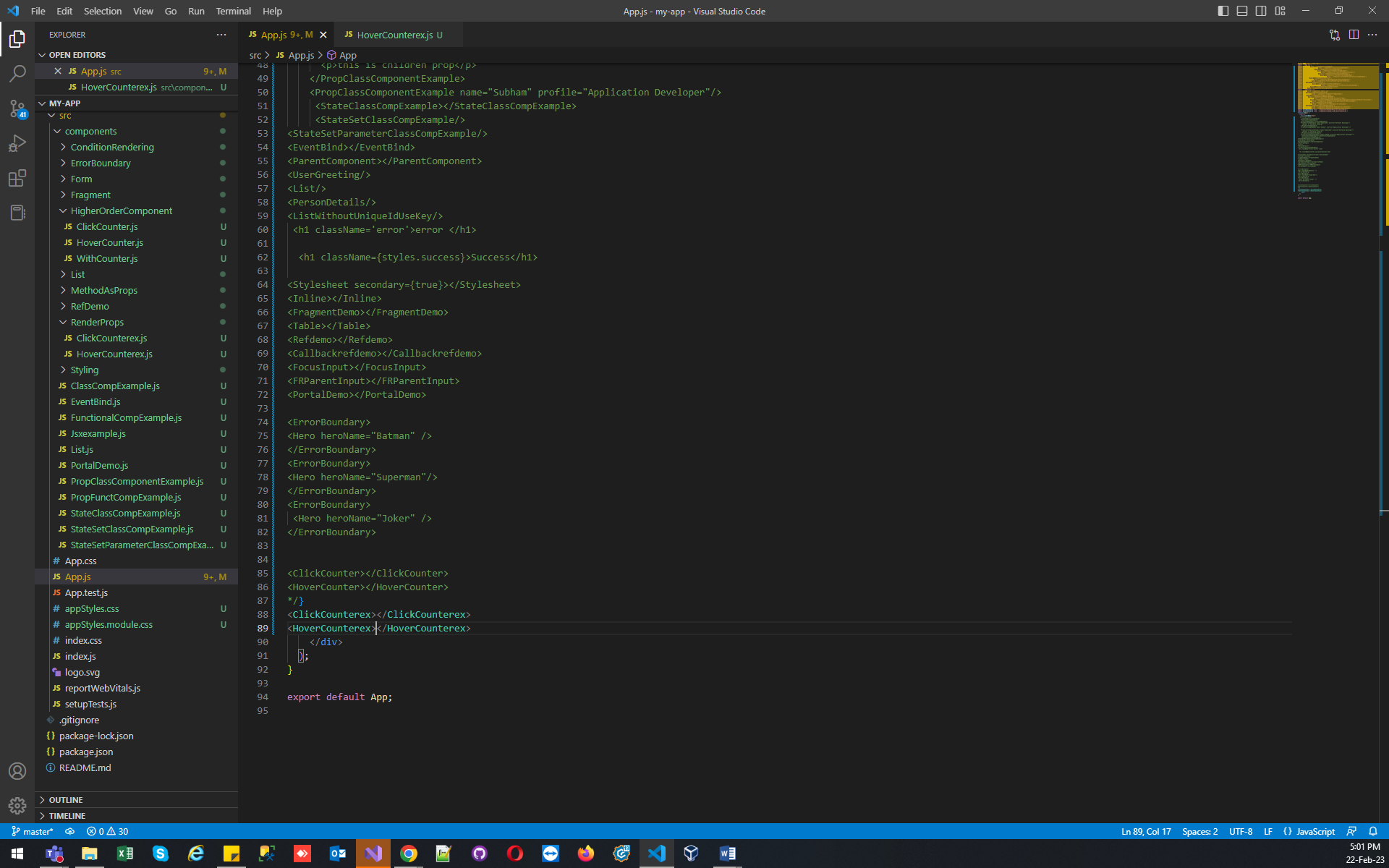
  }

}

export default ClickCounterex



In App.js Call both component :



Created new component User.js and share function :

import React, { Component } from 'react'

class User extends Component {

  render() {

    return (

      <div>

        {this.props.render(true)}

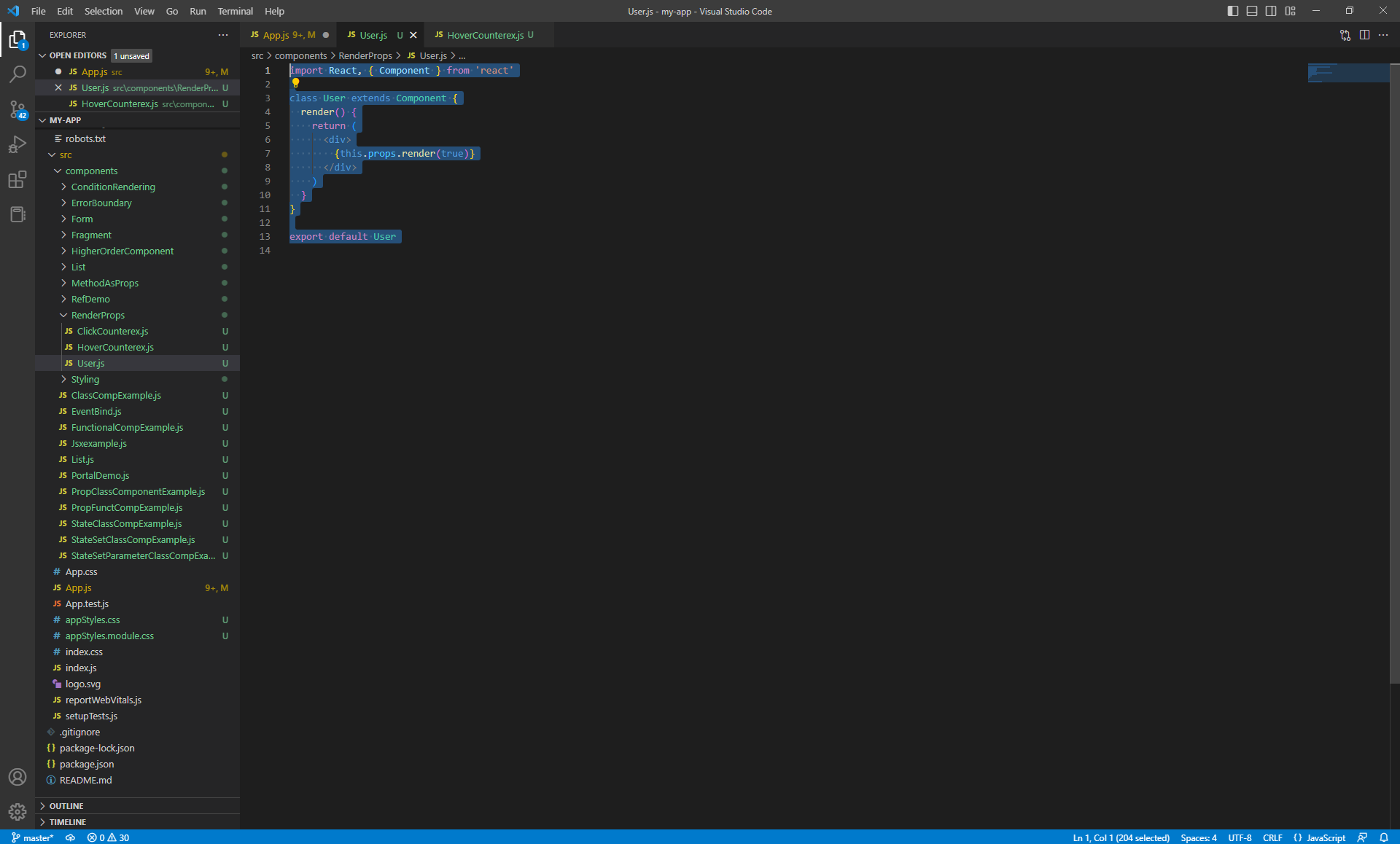
      </div>

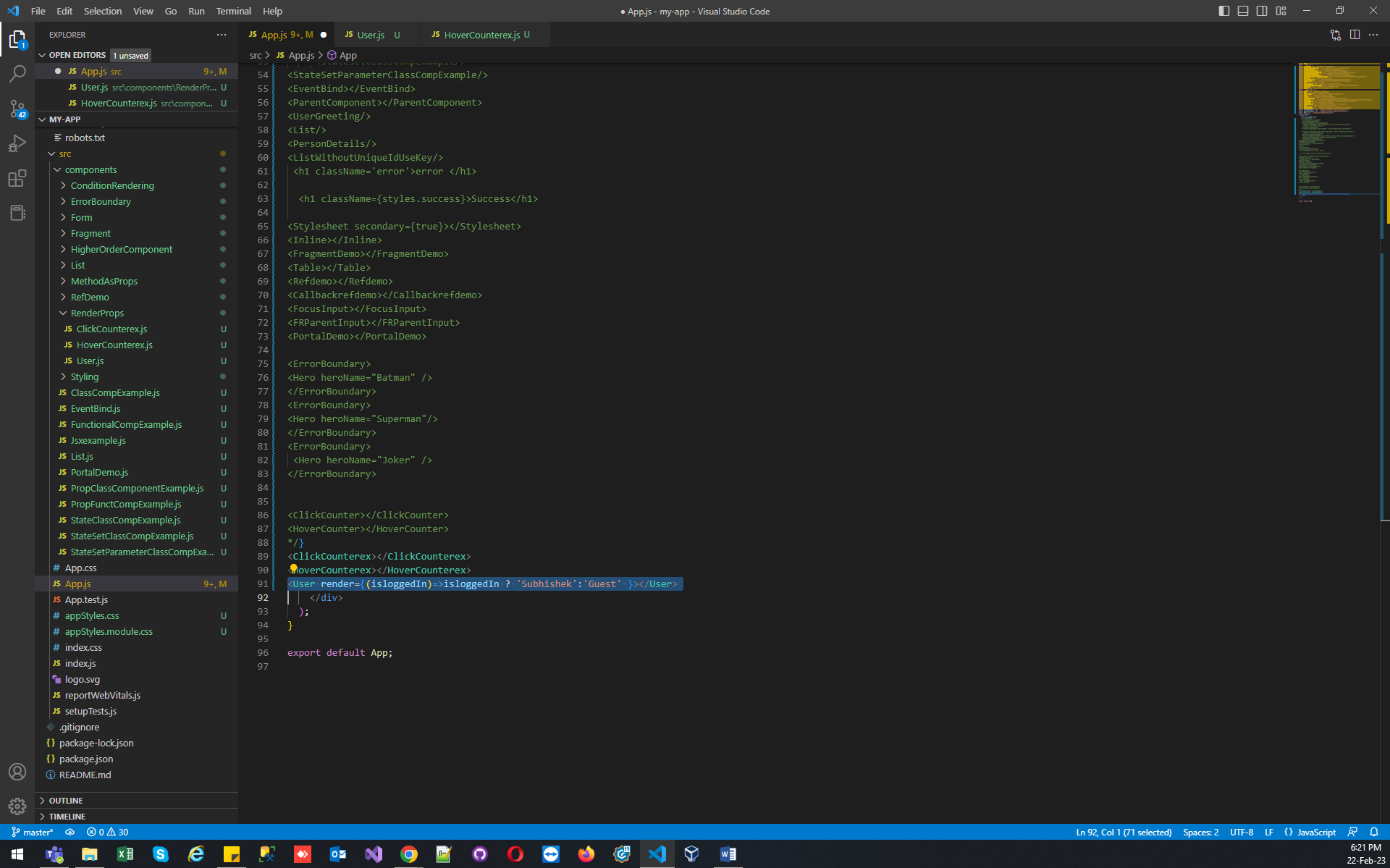
    )

  }

}

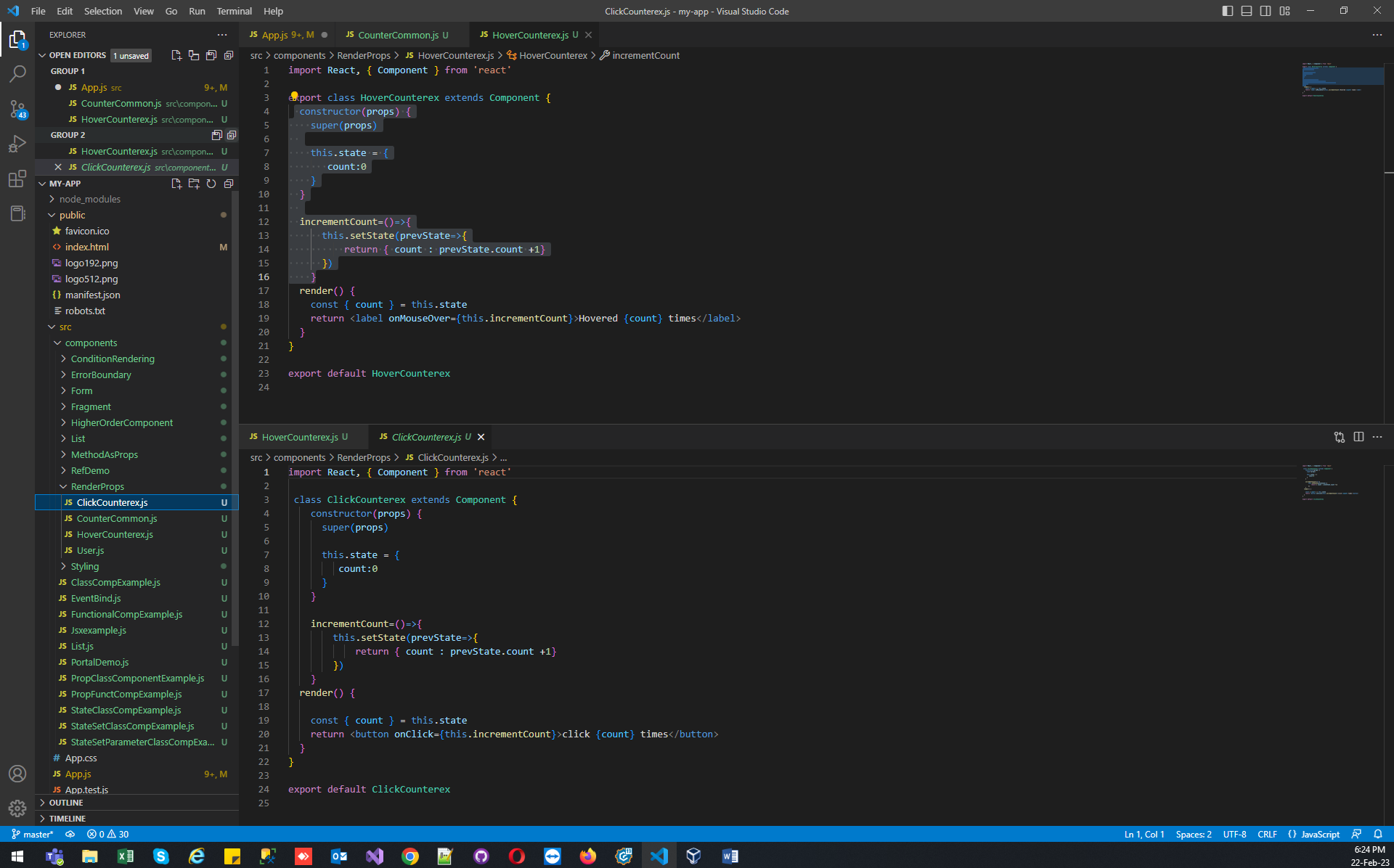
export default User



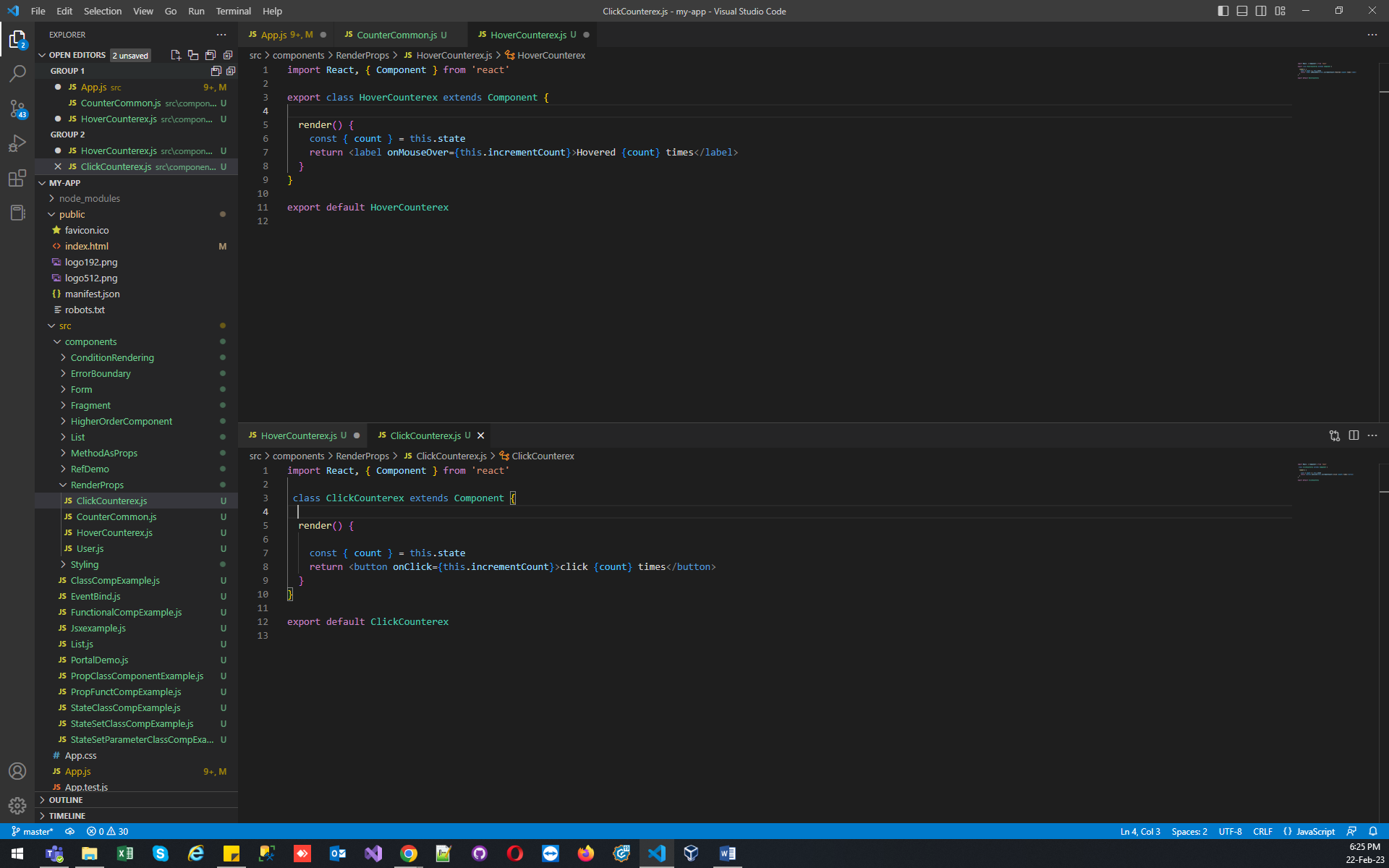


Now Same function Move to CounterCommon.js From HoverCounterex.js and ClickCounterex.js

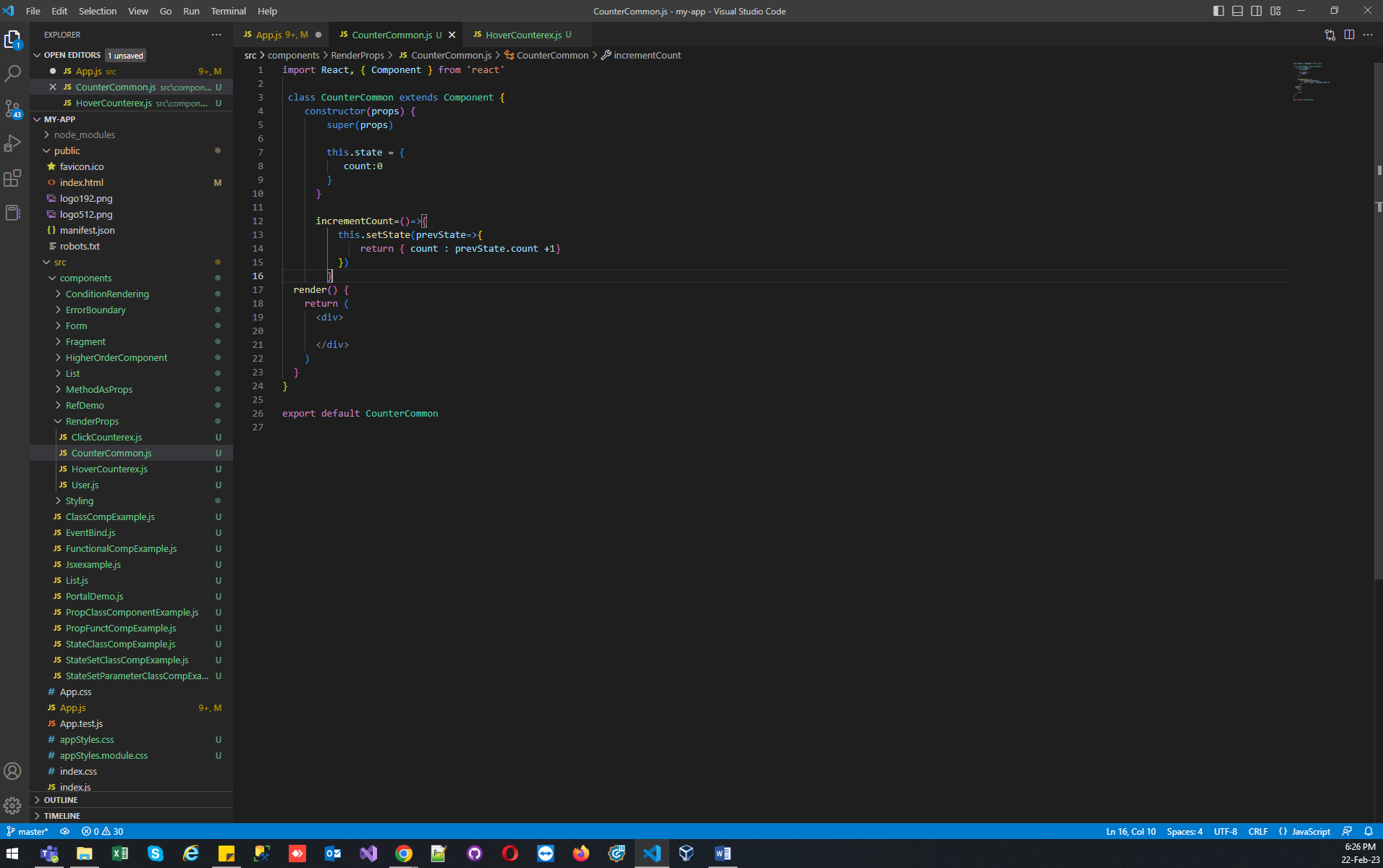
Before



After :



Code Shifted To CounterCommon.js :



In App.js Pass function as prop using render and comments both component now we will access using CounterCommon.js:

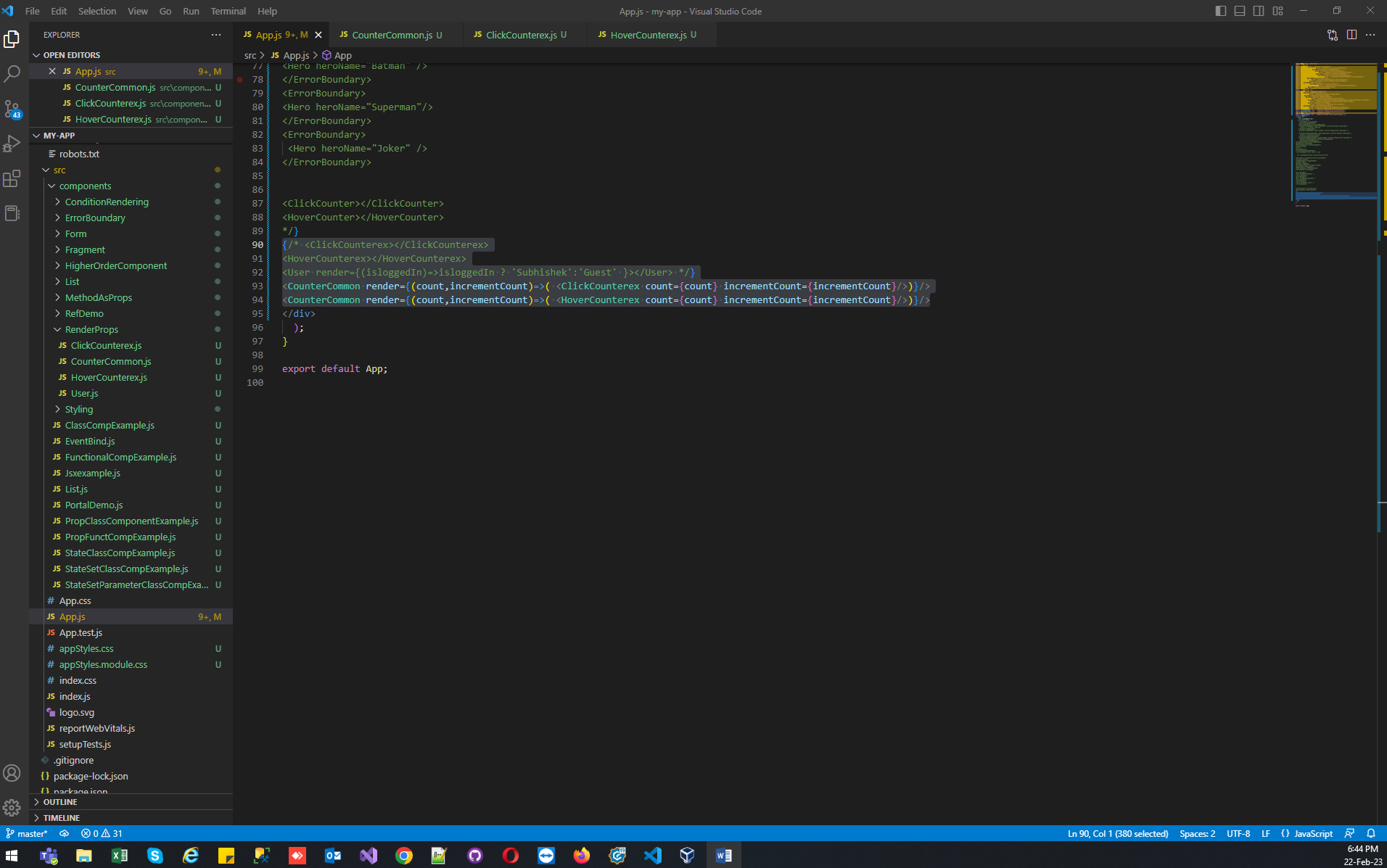
{/\* <ClickCounterex></ClickCounterex>

<HoverCounterex></HoverCounterex>

<User render={(isloggedIn)=>isloggedIn ? 'Subhishek':'Guest' }></User> \*/}

<CounterCommon render={(count,incrementCount)=>( <ClickCounterex count={count} incrementCount={incrementCount}/>)}/>

<CounterCommon render={(count,incrementCount)=>( <HoverCounterex count={count} incrementCount={incrementCount}/>)}/>



CounterCommon.js :

import React, { Component } from 'react'

 class CounterCommon extends Component {

    constructor(props) {

        super(props)

        this.state = {

           count:0

        }

      }

      incrementCount=()=>{

          this.setState(prevState=>{

              return { count : prevState.count +1}

          })

        }

  render() {

    return (

      <div>

        {this.props.render(this.state.count,this.incrementCount)}

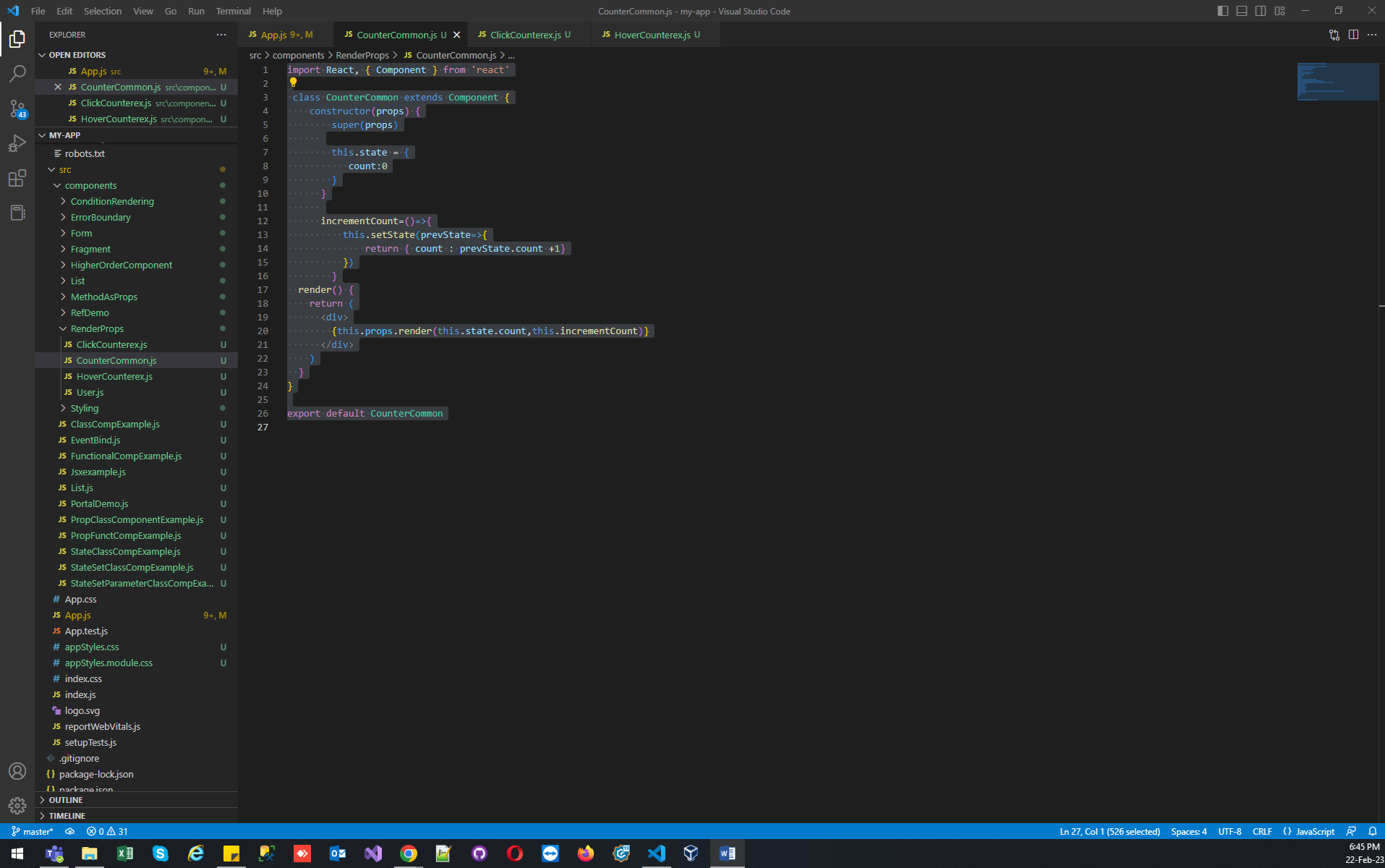
      </div>

    )

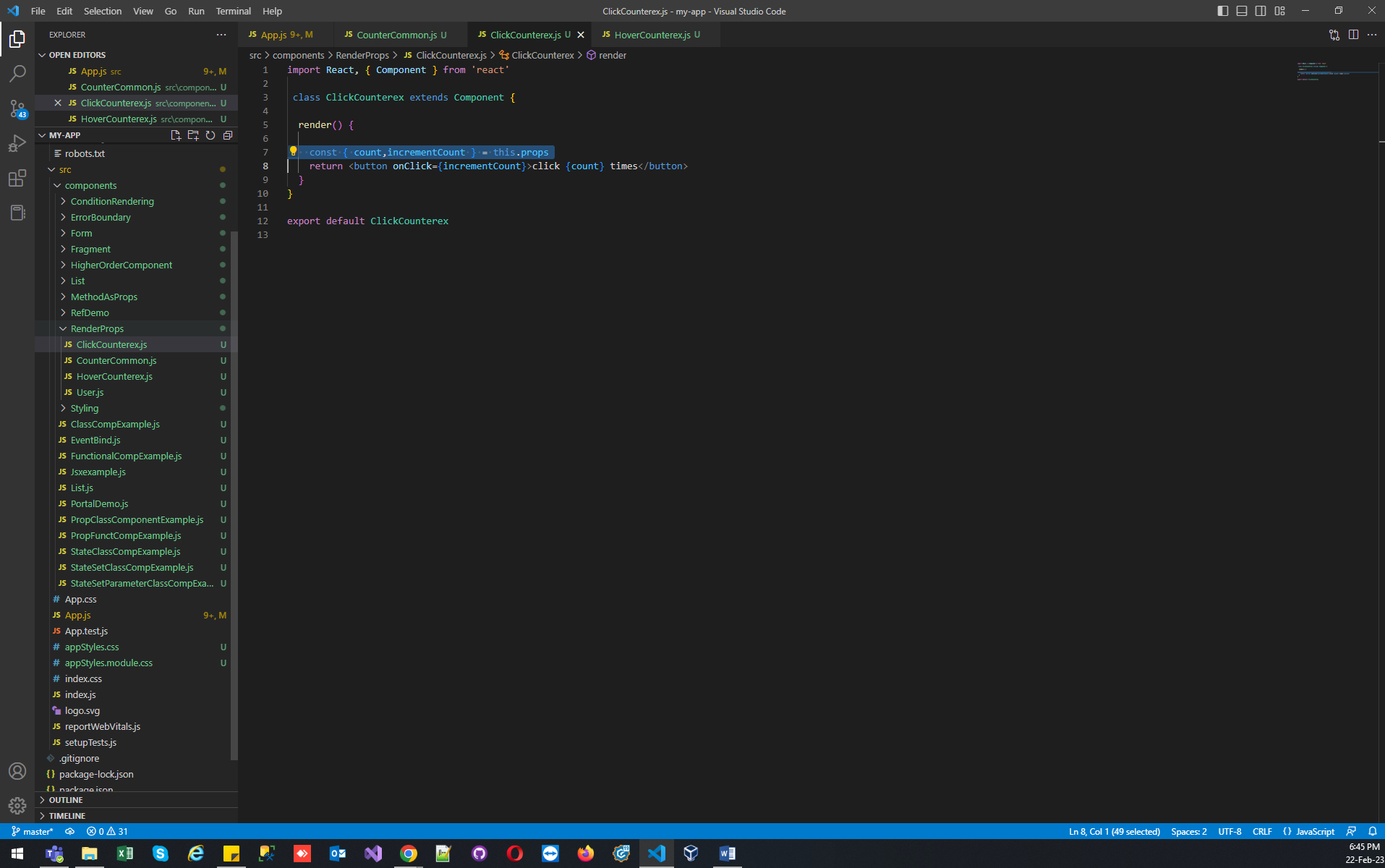
  }

}

export default CounterCommon



ClickCounterex.js :



import React, { Component } from 'react'

 class ClickCounterex extends Component {

  render() {

    const { count,incrementCount } = this.props

    return <button onClick={incrementCount}>click {count} times</button>

  }

}

export default ClickCounterex

HoverCounterex.js :

import React, { Component } from 'react'

export class HoverCounterex extends Component {

  render() {

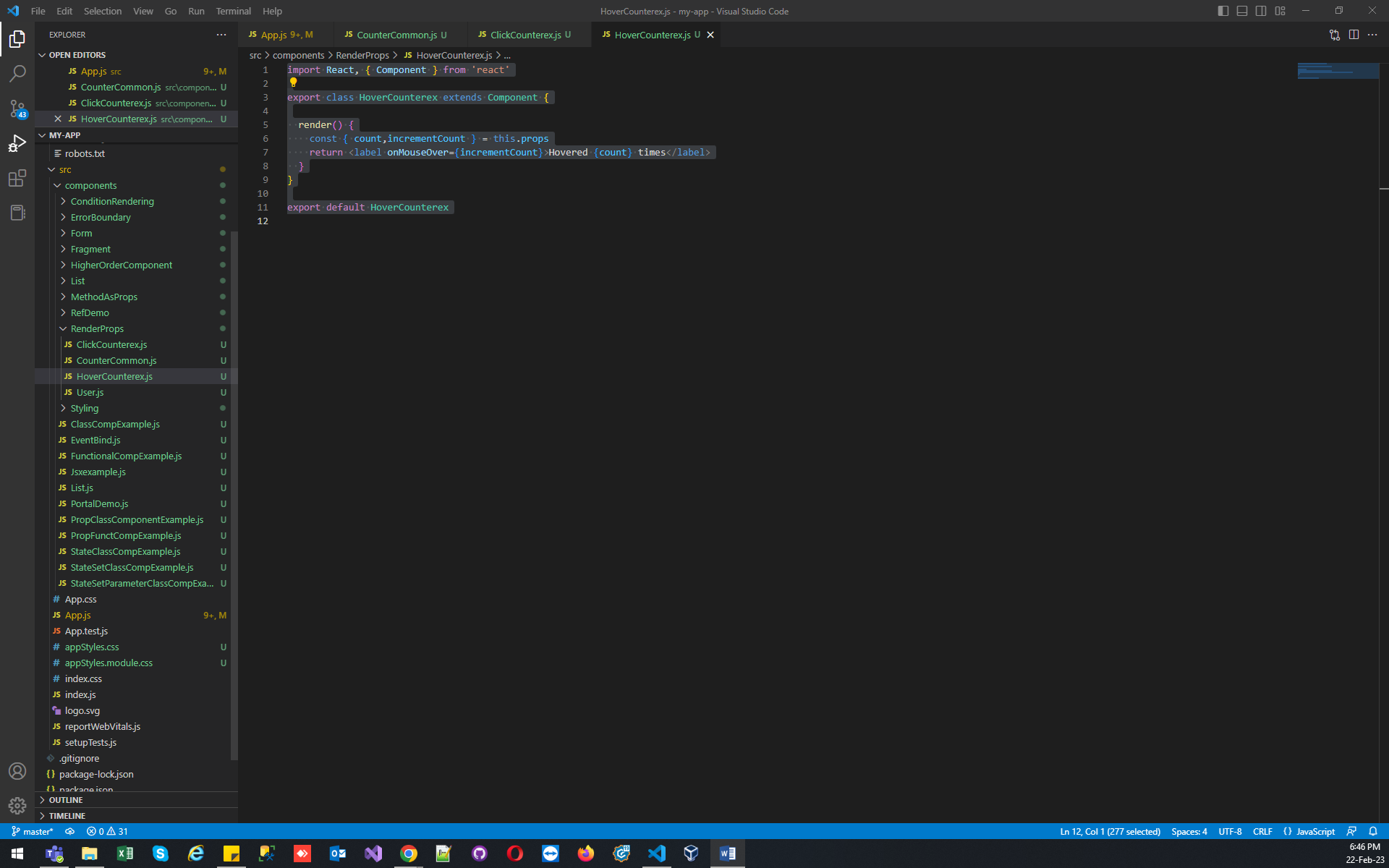
    const { count,incrementCount } = this.props

    return <label onMouseOver={incrementCount}>Hovered {count} times</label>

  }

}

export default HoverCounterex



Output :

