Final Calculator

import React from "react";

class Parent extends React.Component{

    constructor(){

        super()

        this.state={

            prevmsg:''

        }

    }

    change=(event)=>{

        this.setState((prev)=>({

            prevmsg:prev.prevmsg+event.target.value

        }))

    }

    delchange=(event)=>{

        let em=""

        this.setState((prev)=>({

            prevmsg:this.state.prevmsg.replace(prev.prevmsg,em)

        }))

    }

    eldelchange=(event)=>{

        this.setState((prev)=>({

            prevmsg:this.state.prevmsg.slice(0,-1)

        }))

    }

    tchange=(event)=>{

        this.setState({

            prevmsg:event.target.value

        })

    }

    submit=(event)=>{

        event.preventDefault()

        let ans=eval(this.state.prevmsg)

        this.setState((prev)=>({

            prevmsg:ans

        }))

    }

    render(){

        return(<form onSubmit={this.submit}>

            <div className="App"><br></br>

            <h1>My Calculator</h1><br></br>

                <textarea rows={4} cols={40} value={this.state.prevmsg} onChange={this.tchange} /><br></br>

                <button className="button" type="button" value={" "}  onClick={this.eldelchange}>Del</button>&nbsp;

                <button className="button" type="button" value={" "}  onClick={this.delchange}>X</button>&nbsp;

                <button className="button" type='button' onClick={this.submit}>=</button>

                <br></br>

                <button className="button" type="button" value={1}  onClick={this.change}>1</button>&nbsp;

                <button  className="button" type="button" value={2}  onClick={this.change}>2</button>&nbsp;

                <button  className="button" type="button" value={3}  onClick={this.change}>3</button>&nbsp;

                <br></br>

                <button className="button" type="button" value={4}  onClick={this.change}>4</button>&nbsp;

                <button className="button" type="button" value={5}  onClick={this.change}>5</button>&nbsp;

                <button className="button" type="button" value={6}  onClick={this.change}>6</button>&nbsp;

                <br></br>

                <button className="button" type="button" value={7}  onClick={this.change}>7</button>&nbsp;

                <button className="button" type="button" value={8}  onClick={this.change}>8</button>&nbsp;

                <button className="button" type="button" value={9}  onClick={this.change}>9</button>&nbsp;

                <br></br>

                <button className="button" type="button" value={'+'}  onClick={this.change}>+</button>&nbsp;

                <button className="button" type="button" value={0}  onClick={this.change}>0</button>&nbsp;

                <button className="button" type="button" value={'-'}  onClick={this.change}>-</button>&nbsp;

                <br></br>

                <button className="button" type="button" value={'\*'}  onClick={this.change}>\*</button>&nbsp;

                <button className="button" type="button" value={'/'}  onClick={this.change}>/</button>&nbsp;

                <button className="button" type="button" value={'%'}  onClick={this.change}>%</button>&nbsp;

                <br></br>

                <br></br>

            </div>

        </form>

        );

    }

}

export default Parent

CSS

.button {

  position: relative;

  background-color: #04AA6D;

  border: none;

  font-size: 25px;

  color: #FFFFFF;

  padding: 20px;

  width: 100px;

  text-align: center;

  margin-top: 2px;

  -webkit-transition-duration: 0.4s; /\* Safari \*/

  transition-duration: 0.4s;

  text-decoration: none;

  overflow: hidden;

  cursor: pointer;

}

.button:after {

  content: "";

  background: #90EE90;

  display: block;

  position: absolute;

  padding-top: 300%;

  padding-left: 350%;

  margin-left: -20px!important;

  margin-top: -120%;

  opacity: 0;

  transition: all 0.8s

}

.button:active:after {

  padding: 0;

  margin: 0;

  opacity: 1;

  transition: 0s

}

React Portal

Parent.js

import React from 'react';

import  ReactDOM  from'react-dom';

function parent()

{

    return ReactDOM.createPortal(<h1>welcome</h1>,

    document.getElementById('childroot'))

}

export default parent

Index.html

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <meta name="theme-color" content="#000000" />

    <meta

      name="description"

      content="Web site created using create-react-app"

    />

    <link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

    <!--

      manifest.json provides metadata used when your web app is installed on a

      user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/

    -->

    <link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

    <!--

      Notice the use of %PUBLIC\_URL% in the tags above.

      It will be replaced with the URL of the public folder during the build.

      Only files inside the public folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will

      work correctly both with client-side routing and a non-root public URL.

      Learn how to configure a non-root public URL by running npm run build.

    -->

    <title>React App</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div id="root"></div>

    <div id="childroot"></div>

    <!--

      This HTML file is a template.

      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.

      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run npm start or yarn start.

      To create a production bundle, use npm run build or yarn build.

    -->

  </body>

</html>

Error Handling

Child.js

Import React from 'react'

Class Child extends React.Component{

    constructor(){

        super()

        this.state={

            hasError:false

        }

    }

    Static getDerivedStateFromError(){

        return{

            hasError:true

        }

    }

    Component Did Catch(error,info){

        console.log(error)

        console.log(info)

    }

    render(){

        if(this.state.hasError){

            return<h1>Something went wrong</h1>

        }

        returnthis.props.children

    }

}

exportdefaultChild

Parent.js

importReactfrom'react'

//import  ReactDOM  from 'react-dom';

functionparent(props){

    if(props.name==='hello'){

        thrownewError("No hello to be passed")

    }

    return<h1>{props.name}</h1>

}

exportdefaultparent

App.js

importReactfrom'react';

importParentfrom'./component/Parent';

import'./App.css';

importChildfrom'./component/Child';

functionApp()

{

  return(

  <divclassName='App'>

   <Child>

<Parentname="hi"/>

   </Child>

   <Child>

<Parentname="welcome"/>

   </Child>

  <Child>

<Parentname="hello"/>

  </Child>

  </div>

 );

}

exportdefaultApp

Higher Order Components

import React from "react";

//import Child from "./child";

class Parent extends React.Component{

    constructor(){

        super()

        this.state={

            count:0

        }

    }

    inc=()=>{

        this.setState(Prev=>{

            return{

                count:Prev.count+1

            }

        })

    }

    render(){

        return<>

        <button onClick={this.inc}>Clicked{this.state.count}Times</button></>

    }

}

export default Parent

NewComponent=update(orginalComponent)

App.js

import React from 'react';

import './App.css';

import Parent from './component/Parent';

import Child from './component/child';

function App() {

  return (

    <div className='App'>

      <Parent name='Hello'/>

      <Child />

    </div>

  );

}

export default App

Parent.js

import React from "react";

import Update from "./child1"

class Parent extends React.Component{

    render(){

        return<>

        <button onClick={this.props.increment}>Clicked {this.props.count} Times {this.props.name}</button></>

    }

}

export default Update(Parent,9)

Child.js

import React from "react";

import Update from "./child1"

class Parent extends React.Component{

    render(){

        return<>

        <button onMouseOver={this.props.increment}>Clicked {this.props.count} Times {this.props.name}</button></>

    }

}

export default Update(Parent,10)

Child1.js

import React from "react";

const Update = (Orginal,value)=>{

    class Newcomp extends React.Component{

        constructor(){

            super()

            this.state={

                count:0

            }

        }

        inc=()=>{

            this.setState(Prev=>{

                return{

                    count:Prev.count+value

                }

            })

        }

        render(){

            return<>

            <Orginal count={this.state.count}

            increment={this.inc}

            {...this.props}/></>

        }

    }

    return Newcomp

    }

    export default Update