Calculator With Mutiple Inputs

import React from 'react';

import './App.css';

//import Parent from './component/Parent';

//import Child from './component/child';

//import Child1 from './component/child1';

let a=prompt('Enter 1st Number')

//let b=prompt('Enter 2nd Number')

let tot=a

function App() {

  function add() {

    let b=prompt('Enter next number')

    tot=Number(tot)+Number(b)

    document.getElementById("result").innerHTML=tot

  }

  function sub(){

    let b=prompt('Enter next number')

    tot=Number(tot)-Number(b)

    document.getElementById("result").innerHTML=tot

  }

  function mul(){

    let b=prompt('Enter next number')

    tot=Number(tot)\*Number(b)

    document.getElementById("result").innerHTML=tot

  }

  function div(){

    let b=prompt('Enter next number')

    tot=Number(tot)/Number(b)

    document.getElementById("result").innerHTML=tot

  }

  function equal() {

    document.getElementById("result").innerHTML=tot

    alert(`The Total is ${tot}`)

  }

  return (

    <div>

      <h1 align='center'>

      <button className='button' onClick={add}>+</button><br></br>

      <button className='button' onClick={sub}>-</button><br></br>

      <button className='button' onClick={mul}>\*</button><br></br>

      <button className='button' onClick={div}>/</button><br></br>

      <button className='button' onClick={equal}>=</button><br></br>

      <p id='result'></p>

      </h1>

    </div>

  );

}

export default App

Evalute the Expression

Parent.js

import React from "react";

class Parent extends React.Component{

    constructor(){

        super()

        this.state={

            message:''

        }

    }

    change=(event)=>{

        this.setState({

            message:event.target.value

        })

    }

    submit=event=>{

        alert(eval(this.state.message))

    }

    render(){

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

            <div className="App">

                <label>Expression</label>

                <input type="text" value={this.state.message} onChange={this.change}/>

                <br></br>

                <br></br>

                <button type='submit'>Submit</button>

            </div>

        </form>

        );

    }

}

export default Parent

Ref

Parent.js

import React from "react";

class Parent extends React.Component{

    constructor(){

        super()

        this.inputref=React.createRef()

    }

    componentDidMount(){

        this.inputref.current.focus()

    }

    render(){

        return <input type='text' ref={this.inputref}/>

    }

}

export default Parent

Ref – this.cd method

Parent.js

import React from "react";

class Parent extends React.Component{

    constructor(){

        super()

        this.cd=null

        this.setcd=element=>{

        this.cd=element

    }

    }

    componentDidMount(){

        if(this.cd){

            this.cd.focus()

        }

    }

    render(){

        return <input type='text' ref={this.setcd}/>

    }

}

export default Parent

Ref-Method:3-Parent call child to focus

Parent.js

import React from "react";

import Child from "./child";

class Parent extends React.Component{

    constructor(){

        super()

        this.inputref=React.createRef()

    }

    handle=()=>{

        this.inputref.current.focusInput()

    }

    render(){

        return<><Child ref={this.inputref}/>

        <button onClick={this.handle}>Click Here</button></>

    }

}

export default Parent

child.js

import React from "react";

class Child extends React.PureComponent{

   constructor(){

    super()

    this.inputref=React.createRef()

   }

   focusInput(){

    this.inputref.current.focus()

   }

   render()

   {

    return <input type='text' ref={this.inputref}/>

   }

}

export default Child;

Ref – Method:3 Alter one-Forward Ref

Parent.js

import React from "react";

import Child from "./child";

class Parent extends React.Component{

    constructor(){

        super()

        this.inputref=React.createRef()

    }

    handle=()=>{

        this.inputref.current.focus()

    }

    render(){

        return<><Child ref={this.inputref}/>

        <button onClick={this.handle}>Click Here</button></>

    }

}

export default Parent

Child.js

import React from "react";

const Child=React.forwardRef((props,ref)=>{

    return<><input type='text' ref={ref}/></>

})

export default Child;