Saving data from a form

const [details, setDetails] = useState({

       name: '',

       otherName: '',

       otherName: '',

       })

const [readName, setName] = useState('')

const handleChange = (e) => {

            const { name, value } = e.target;

            setDetails((prev) => {

                return { ...prev, [name]: value };

            });

        };

const handleCalculate = (e) => {

    e.preventDefault();

   let readName;

   readName = details.name;

   setName(readName);

   console.log(details);

};

<form onSubmit={handleCalculate}>

<input type='text' name='name' onChange={handleChange} className='text-black'></input>

<button type='submit' className="w-[50%] ml-[25%] mr-[25%] mt-4 bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 border border-blue-700 rounded"> Submit </button>

</form>

Mapping an object using arrays

(objects cannot be mapped)

//object with form details in component

    const [details, setDetails] = useState({

        name: '',

        calories: 0,

        serving: 0,

    });

//In App.js use a state for destructuring object in an array

  const [row, setRow] = useState([]);

  const addProduct = (productInfo) => {

    setRow([...row, productInfo]);

  };

  return (

      <div>

        <Form addProduct={addProduct} />

        <ProductList row={row} />

      </div>

  );

}

//And we map using row state

export default function ProductList({ row }) {

  return (

          <div className='flex flex-col'>

          {row.map((product) => (

            <div key={uuid}>

              <p className='w-[25%]'>{product.name}</p>

                <p className='w-[25%]'>{product.calories}</p>

                <p className='w-[25%]'>{product.serving}</p>

              </div>

          ))}

          </div>

  )

}