**Ex. No : 4 A**

**Date : 20.03.2025**

**Develop client-side application using React API**

**B. Application demonstrating React List**

**AIM:**

To Develop application to demonstrate react list and usage in map function.

**PROCEDURE:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **React** | **DESCRIPTION** |
| **1** | **useState()** | **Manages state inside a functional component (e.g., storing form data).** |
| **2** | **useEffect()** | **Runs side effects (e.g., fetching data, form validation on load).** |
| **3** | **onChange()** | **Handles user input changes in form fields.** |
| **4** | **onSubmit()** | **Triggers form submission and validation logic.** |
| **5** | **setState()** | **Updates the state of the component dynamically.** |
| **6.** | **preventDefault()** | **Prevents the default form submission behavior.** |
| **7** | **map()** | **Loops through an array (e.g., displaying validation errors).** |

**SOURCE CODE :**

App.js

import React, { useState } from "react";

import "./styles.css"

export default function Calculator() {

const [input, setInput] = useState("");

const [num1, setnum1] = useState("")

const [num2, setnum2] = useState("")

const [operation, setOperation] = useState("");

const handleClick = async (value) => {

if ( value == "Clear" ) {

setInput("")

} else if ( value == "=" ) {

try {

const response = await fetch("http://localhost:8080/calcDirectly", {

method: "POST",

headers: {"Content-Type": "application/json"},

body: JSON.stringify({

exp: input

})

})

const data = await response.json()

if(response.ok) setInput(data)

else setInput("Error")

} catch {

setInput("Error")

}

} else {

setInput(input + value)

}

};

const buttons = [

"1", "2", "3", "+",

"4", "5", "6", "-",

"7", "8", "9", "\*",

"/","0" ,"Clear", "=", "."

]

return <div className="Container pacifico-regular">

<div className="Calc">

<p className="title">Calculator</p>

<div className="Display">

<h1>

{input}

</h1>

</div>

<div className="ButtonContainer">

{

buttons.map((button) => {

return (

<div key={button} className={`button ${isNaN(button) ? "utiltity" : ""} `} onClick={() => handleClick(button)}>

<p>{button}</p>

</div>

)

})

}

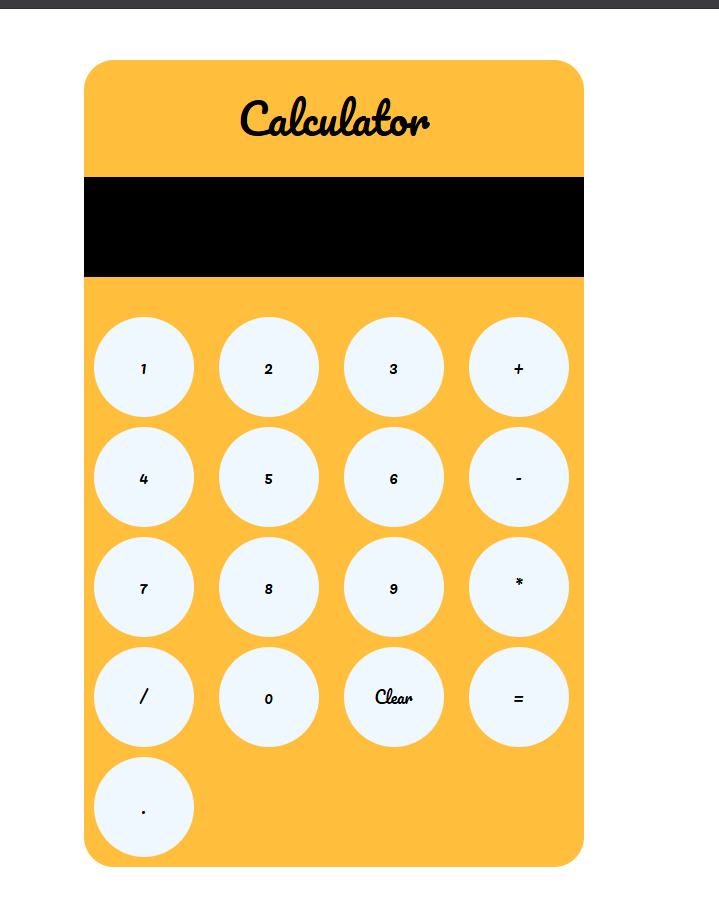
</div>

</div>

</div>

};

**OUTPUT 1 :**



**RESULT :**

The program has been successfully executed and the output is verified.