

Subject code & Name	:	CSE304 & Full Stack Development	Practical	:	5	Academic Year	:	2024-25
---------------------	---	---------------------------------	-----------	---	---	---------------	---	---------

Task - 5

Aim

Create a calculator React app involves several steps, from setting up the React environment to implementing the logic for basic arithmetic operations. Here's a detailed explanation of creating a simple calculator app with addition, subtraction, multiplication, and division capabilities.

Code

Calculator.js

```
import React, { useState } from 'react';
import './Calculator.css';

const Calculator = () => {
  const [input, setInput] = useState("");
  const [expression, setExpression] = useState("");

  const handleClick = (value) => {
    setInput((prev) => prev + value);
    setExpression((prev) => prev + value);
  };

  const handleClear = () => {
    setInput("");
    setExpression("");
  };

  const handleDelete = () => {
    setInput((prev) => prev.slice(0, -1));
    setExpression((prev) => prev.slice(0, -1));
  };

  const handleEqual = () => {
    try {
      setInput(eval(expression).toString()); // Use eval for simplicity; be cautious in production
      setExpression("");
    } catch (error) {
      setInput('Error');
      setExpression("");
    }
  };
};
```

```
return (  
  <div className="calculator">  
    <div className="display">  
      <span className="small-input">{expression}</span>  
      <span className="large-input">{input || '0'}</span>  
    </div>  
    <div className="buttons">  
      <button onClick={handleClear} className="clear">C</button>  
      <button onClick={() => handleClick('±')} className="operator">±</button>  
      <button onClick={() => handleClick('%')} className="operator">%</button>  
      <button onClick={() => handleClick('/')} className="operator">÷</button>  
  
      <button onClick={() => handleClick('7')}>7</button>  
      <button onClick={() => handleClick('8')}>8</button>  
      <button onClick={() => handleClick('9')}>9</button>  
      <button onClick={() => handleClick('*')} className="operator">×</button>  
  
      <button onClick={() => handleClick('4')}>4</button>  
      <button onClick={() => handleClick('5')}>5</button>  
      <button onClick={() => handleClick('6')}>6</button>  
      <button onClick={() => handleClick('-')} className="operator">-</button>  
  
      <button onClick={() => handleClick('1')}>1</button>  
      <button onClick={() => handleClick('2')}>2</button>  
      <button onClick={() => handleClick('3')}>3</button>  
      <button onClick={() => handleClick('+')} className="operator">+</button>  
  
      <button onClick={() => handleClick('0')} className="zero">0</button>  
      <button onClick={() => handleClick('.')}>.</button>  
      <button onClick={handleEqual} className="equal">=</button>  
      <button onClick={handleDelete} className="del">DEL</button>  
    </div>  
  </div>  
};  
  
export default Calculator;
```

Calculator.css

```
/* General body styling */
body {
  background-color: #f1f1f1;
  font-family: 'Helvetica Neue', sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
}

/* Calculator container */
.calculator {
  background-color: black;
  padding: 20px;
  border-radius: 30px;
  width: 360px;
  box-shadow: 0px 0px 25px rgba(0, 0, 0, 0.7);
  display: flex;
  flex-direction: column;
  align-items: center;
}

/* Display area (number input & result) */
.display {
  background-color: black;
  color: white;
  font-size: 72px;
  padding: 10px;
  text-align: right;
  width: 100%;
  border-radius: 10px;
  margin-bottom: 15px;
  height: 120px;
  box-shadow: inset 0 -4px 10px rgba(255, 255, 255, 0.2);
  display: flex;
  justify-content: flex-end;
  align-items: center;
  flex-direction: column;
}

.display .small-input {
  font-size: 24px;
```

```
    color: lightgray;
    margin-right: 10px;
    margin-bottom: 5px;
}

.display .large-input {
    font-size: 48px;
    color: white;
}

/* Button container */
.buttons {
    display: grid;
    grid-template-columns: repeat(4, 1fr);
    gap: 10px;
    width: 100%;
}

/* Button styling (default for numbers) */
button {
    background-color: #333;
    border: none;
    color: white;
    font-size: 28px;
    border-radius: 50%;
    padding: 25px;
    cursor: pointer;
    transition: background-color 0.2s ease, box-shadow 0.2s ease;
    box-shadow: 0 4px rgba(0, 0, 0, 0.2);
    display: flex;
    justify-content: center;
    align-items: center;
}

button:active {
    transform: scale(0.95);
}

button:hover {
    background-color: #4a4a4a;
}

/* Operator buttons (orange in iOS) */
.operator {
    background-color: #ff9500;
```

```
    color: white;
}

.operator:hover {
    background-color: #ffad33;
}

.operator:active {
    background-color: #e08a00;
}

/* Clear button (light gray) */
.clear {
    background-color: #a5a5a5;
    color: black;
}

.clear:hover {
    background-color: #b8b8b8;
}

/* Del button (light gray like clear) */
.del {
    background-color: #a5a5a5;
    color: black;
}

.del:hover {
    background-color: #b8b8b8;
}

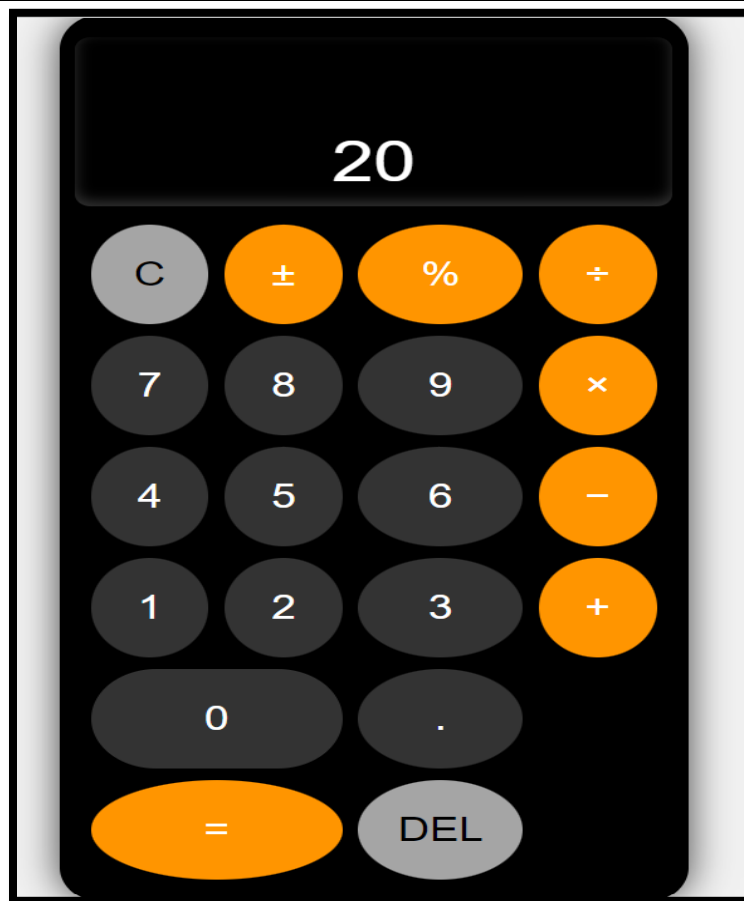
/* Equals button (orange) */
.equal {
    background-color: #ff9500;
    color: white;
    grid-column: span 2;
}

.equal:hover {
    background-color: #ffad33;
}

.equal:active {
    background-color: #e08a00;
}
```

```
/* Zero button (spans two columns) */  
.zero {  
  grid-column: span 2;  
  border-radius: 50px;  
}  
  
/* Button press effect */  
button:active {  
  box-shadow: 0 0 5px rgba(0, 0, 0, 0.3);  
  background-color: #505050;  
}
```

Output



5.1 Calculator App

Signature :