Web Development Project Report

Project Overview

Project Name:

Calculator Application

Description:

This web application is a calculator that allows users to perform basic arithmetic operations (addition, subtraction, multiplication, division) on numbers. It provides a user-friendly interface for entering digits and performing calculations.

Technologies Used

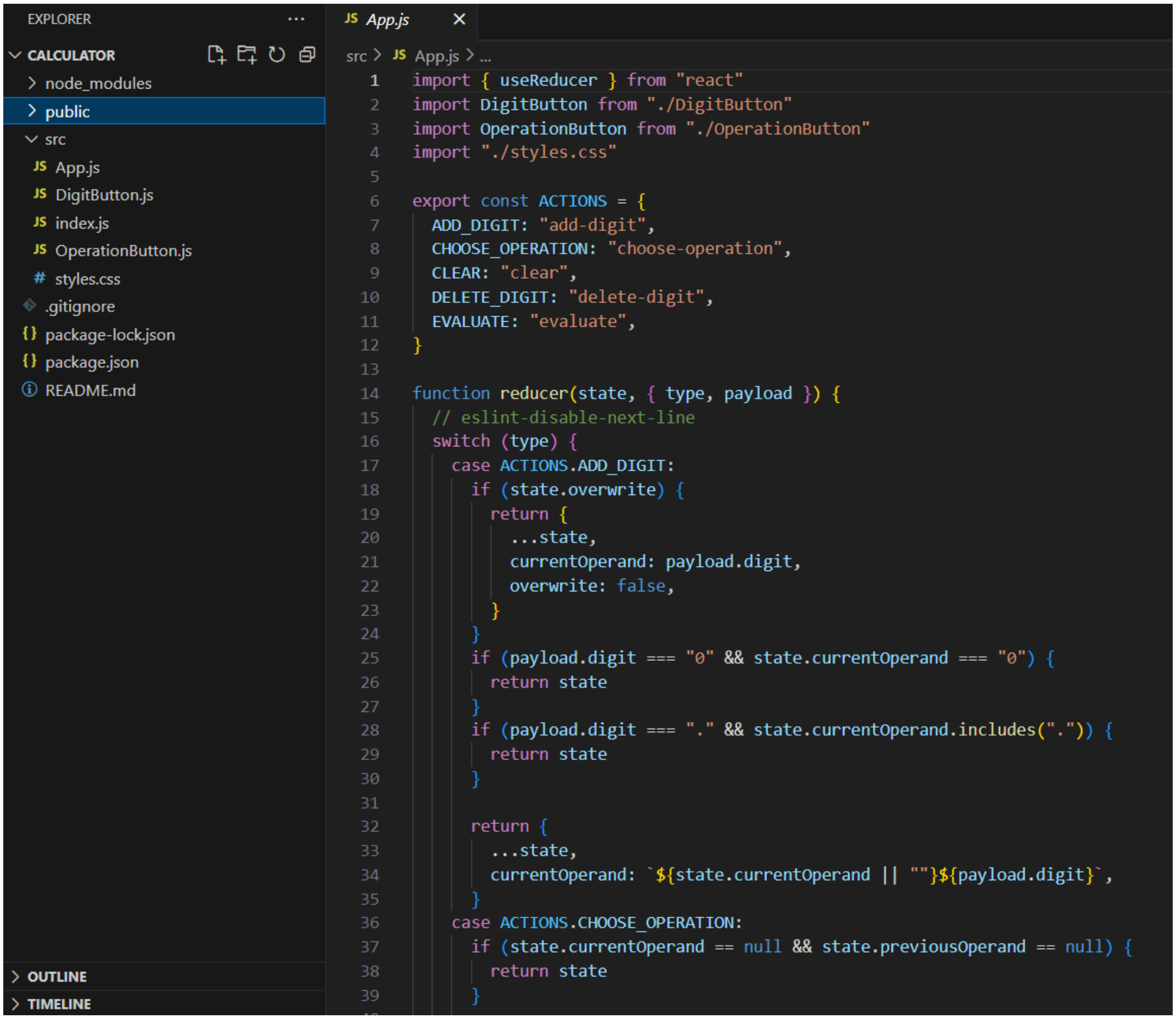
- Frontend Framework: React
- Styling: CSS (with some external styling libraries)
- Version Control: Git (with GitHub for repository hosting)
- Package Management: npm (Node Package Manager)
- Code Editor: Visual Studio Code

Project Structure

The project is structured as follows:

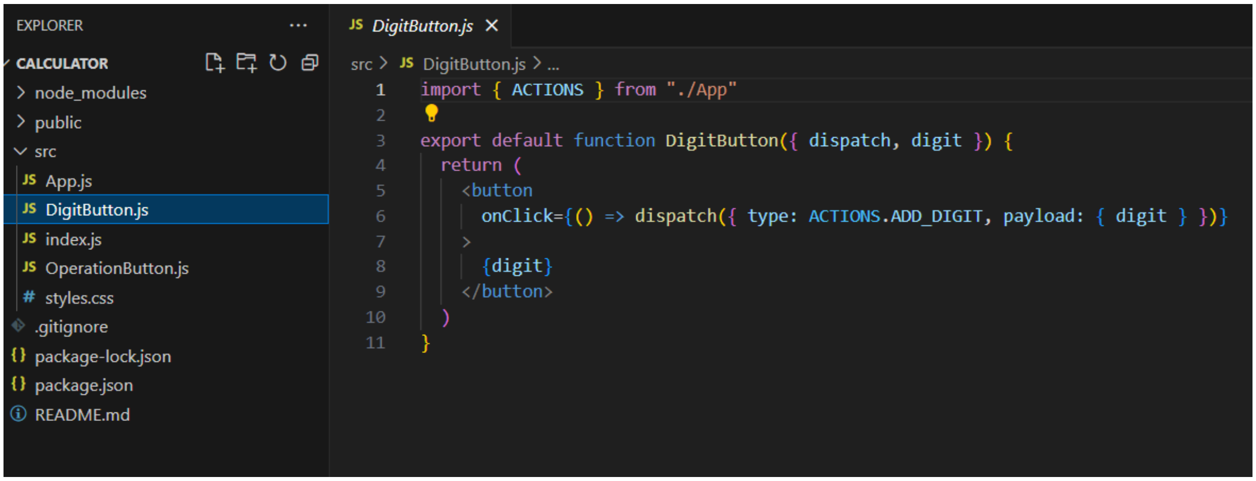
- `src/` (Source Code)

- `App.js`: Main component for the calculator application.



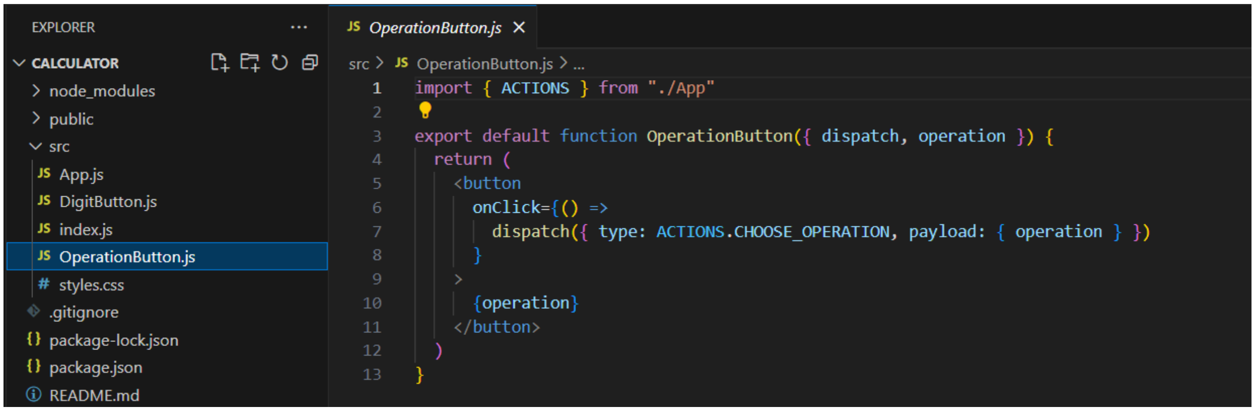
```
import { useReducer } from "react"
import DigitButton from "./DigitButton"
import OperationButton from "./OperationButton"
import "./styles.css"

export const ACTIONS = {
  ADD_DIGIT: "add-digit",
  CHOOSE_OPERATION: "choose-operation",
  CLEAR: "clear",
  DELETE_DIGIT: "delete-digit",
  EVALUATE: "evaluate",
}

function reducer(state, { type, payload }) {
  // eslint-disable-next-line
  switch (type) {
    case ACTIONS.ADD_DIGIT:
      if (state.overwrite) {
        return {
          ...state,
          currentOperand: payload.digit,
          overwrite: false,
        }
      }
      if (payload.digit === "0" && state.currentOperand === "0") {
        return state
      }
      if (payload.digit === "." && state.currentOperand.includes(".")) {
        return state
      }

      return {
        ...state,
        currentOperand: `${state.currentOperand || ""}${payload.digit}`,
      }
    case ACTIONS.CHOOSE_OPERATION:
      if (state.currentOperand == null && state.previousOperand == null) {
        return state
      }
```

- `DigitButton.js`: Component for rendering numeric buttons.



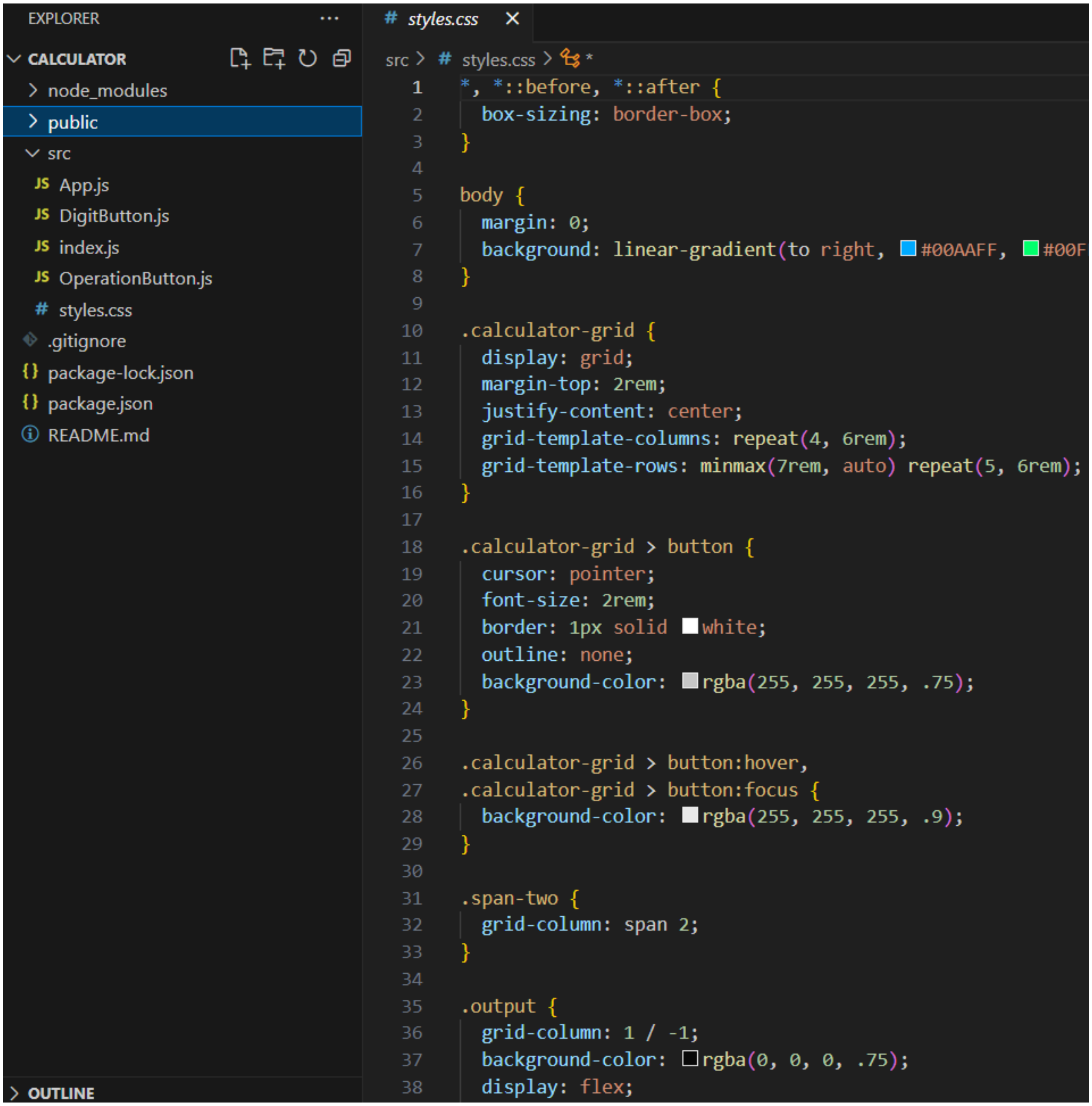
```
import { ACTIONS } from "./App"

export default function DigitButton({ dispatch, digit }) {
  return (
    <button
      onClick={() => dispatch({ type: ACTIONS.ADD_DIGIT, payload: { digit } })}
    >
      {digit}
    </button>
  )
}
```

- `OperationButton.js`: Component for rendering operation buttons.



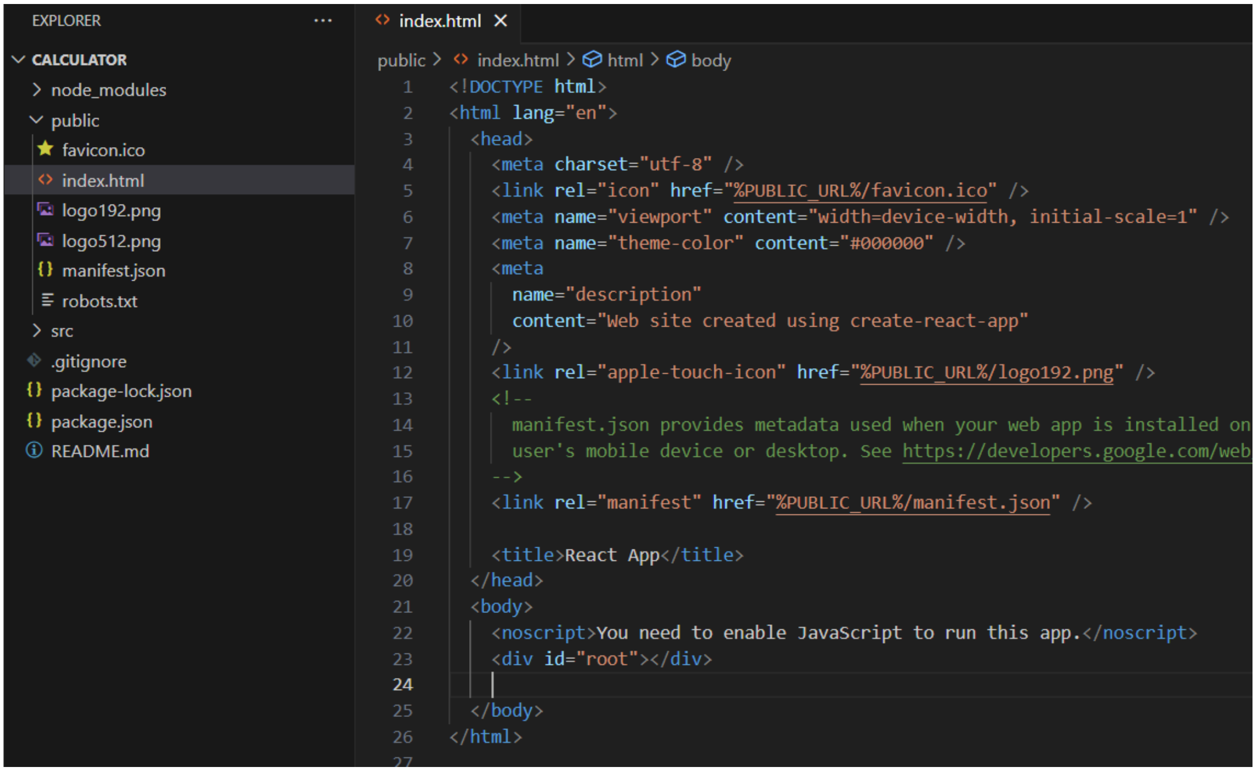
```
import { ACTIONS } from "./App"

export default function OperationButton({ dispatch, operation }) {
  return (
    <button
      onClick={() =>
        dispatch({ type: ACTIONS.CHOOSE_OPERATION, payload: { operation } })
      }
    >
      {operation}
    </button>
  )
}
```

- `styles.css`: CSS file for styling the application.



```css
*, *::before, *::after {
  box-sizing: border-box;
}

body {
  margin: 0;
  background: linear-gradient(to right, #00AAFF, #00F
}

.calculator-grid {
  display: grid;
  margin-top: 2rem;
  justify-content: center;
  grid-template-columns: repeat(4, 6rem);
  grid-template-rows: minmax(7rem, auto) repeat(5, 6rem);
}

.calculator-grid > button {
  cursor: pointer;
  font-size: 2rem;
  border: 1px solid white;
  outline: none;
  background-color: rgba(255, 255, 255, .75);
}

.calculator-grid > button:hover,
.calculator-grid > button:focus {
  background-color: rgba(255, 255, 255, .9);
}

.span-two {
  grid-column: span 2;
}

.output {
  grid-column: 1 / -1;
  background-color: rgba(0, 0, 0, .75);
  display: flex;
```

- `public/`: Contains the HTML file (`index.html`) and other static assets.



```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
      user's mobile device or desktop. See https://developers.google.com/web
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />

    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>

  </body>
</html>
```

Features

1.Basic Arithmetic Operations:
  - Addition, subtraction, multiplication, and division operations are supported.

2.Clear and Delete Functionality:
  - Users can clear the display or delete the last entered digit.

3.Decimal Point Handling:
  - The application ensures that only one decimal point can be entered in a number.

4.Error Handling:
  - The application is designed to handle edge cases and prevent invalid input.

5.State Management:
  - State is managed using the `useReducer` hook, ensuring predictable and controlled updates.

6.Memory Functions:
  -Memory Store (MS)
  -Memory Recall (MR)
  -Memory Clear (MC)

7.Advanced Scientific Functions:

- Trigonometric functions (sin, cos, tan)
- Exponential (exp)
- Natural logarithm (ln)
- Common logarithm (log10)
- Square root (sqrt)

Code Structure

- Component-Based Architecture:
  The application is built using a component-based architecture. Each component handles a specific aspect of the UI (e.g., buttons, display).

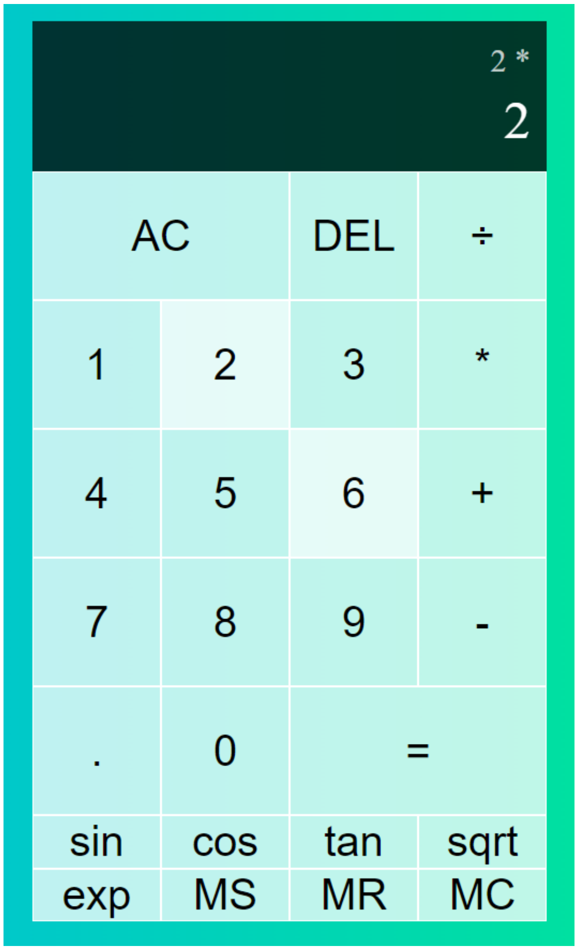
- Reducer for State Management:
  State is managed using a reducer function, which processes actions and returns a new state based on the action type.

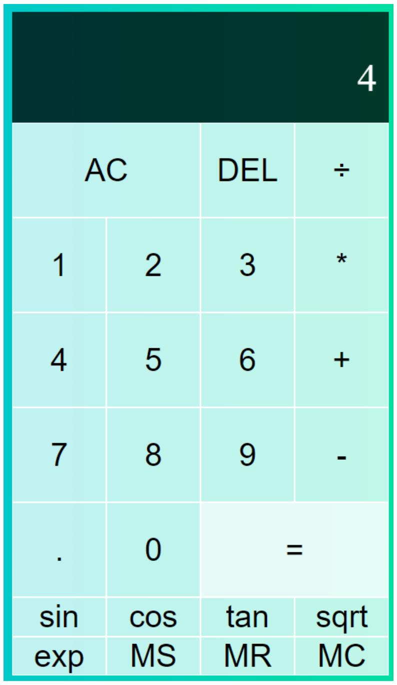
- Spread Operator for Immutability:
  The spread operator (`...`) is used to create new state objects, ensuring immutability and predictable state updates.

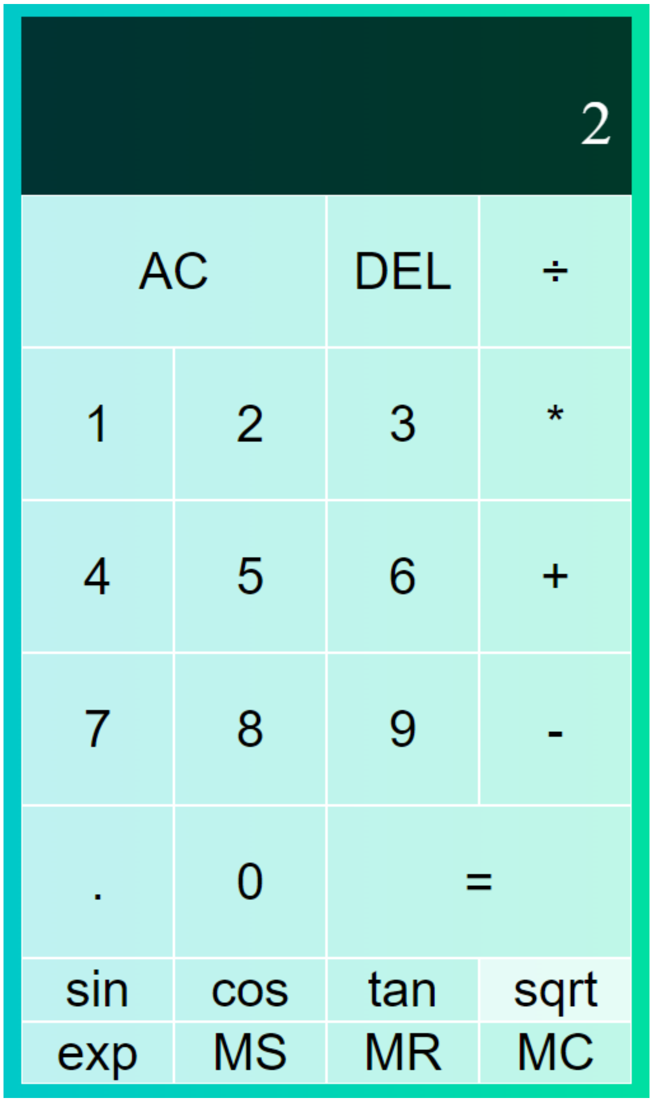
Future Improvements

1. Unit Conversion:
   a. Implement features for unit conversions (e.g., length, weight, temperature).
2. Graphing Functions:
   a. Add the ability to graph mathematical functions.
3. Accessibility Improvements:
   a. Enhance accessibility features to make the calculator usable for individuals with disabilities.

ScreenShot

1. 
2 *
2
AC
DEL
÷
1
2
3
*
4
5
6
+
7
8
9
-
.
0
=
sin
cos
tan
sqrt
exp
MS
MR
MC

2. 
4
AC
DEL
÷
1
2
3
*
4
5
6
+
7
8
9
-
.
0
=
sin
cos
tan
sqrt
exp
MS
MR
MC

3. 

Conclusion

The Extended Scientific Calculator project showcases an advanced level of proficiency in React and modern web development practices. The inclusion of a wide range of mathematical functions and memory operations extends its functionality, making it a versatile tool for various mathematical computations.