**📑 Project Report: React Calculator**

**1. Introduction**

This project is a **simple React-based calculator** that allows users to perform basic arithmetic operations such as addition, subtraction, multiplication, and division. It demonstrates state management with React hooks and a clean, responsive user interface using CSS.

**2. Technologies Used**

* **React 18** – Component-based UI development
* **JavaScript (ES6+)** – Application logic
* **CSS** – Styling and layout
* **npm** – Package and dependency management

**3. Features**

* Displays entered input and results dynamically.
* Supports numbers, decimal points, and basic operators (+, -, \*, /).
* “Clear” button resets the input.
* Input validation with error handling for invalid expressions.
* Responsive UI styled with custom CSS.

**4. Code Highlights**

* **State Management**:
* const [input, setInput] = useState("");

Manages user input and updates dynamically.

* **Reusable Button Rendering**:  
  Calculator buttons are mapped from an array, reducing repetition:
* {["7","8","9","/","4","5","6","\*","1","2","3","-","0",".","=","+"].map((btn) => (
* <button key={btn} ...>{btn}</button>
* ))}
* **Expression Evaluation**:  
  Uses eval() for quick evaluation (suitable for demo purposes).
* **Error Handling**:  
  Catches invalid inputs and displays "Error".

**5. Limitations & Improvements**

* **Security**: eval() is not safe for production; a custom parser would be safer.
* **Features**: Could be extended with advanced functions (%, √, memory storage).
* **Testing**: Adding unit tests with Jest/React Testing Library would improve reliability.

**6. Conclusion**

This project demonstrates how to build a simple, interactive calculator in React. It applies fundamental concepts of **React state, event handling, and UI rendering**, making it a good starting point for beginners while offering room for enhancements.

This is the link to the repo

https://github.com/Akerele-Anuoluwapo/react-calculator2