# Feature Planning Document for React Calculator

**Project:** React Calculator ([https://github.com/ahfarmer/calculator](https://github.com/ahfarmer/calculator))
**By:** Mai Waheed Abdelghany
**Date:** 13/5/2025

## Feature 1: Scientific Mode

### User Story

As a user, I want to enable scientific mode so that I can perform advanced calculations like trigonometric and logarithmic functions.

### Technical Requirements

- Add toggle button for enabling/disabling scientific mode

- Render additional buttons (sin, cos, tan, log, √, ^)

- Handle scientific operations in logic

### Files Likely to be Modified

- App.js – handles the isScientificMode state, toggling, and scientific calculation logging

- ButtonPanel.js – conditionally renders scientific buttons

- calculate.js – processes scientific operations

### Potential Challenges

- Ensuring calculations are accurate and do not break when input is incomplete

- Correct angle unit handling (degrees vs. radians)

- Preventing layout issues from extra buttons

## Feature 2: Dark Mode Toggle

### User Story

As a user, I want to switch between light and dark mode to make the calculator easier on my eyes depending on lighting conditions.

### Technical Requirements

- Implement theme toggle button

- Maintain theme state in localStorage

- Update root class for theming

### Files Likely to be Modified

- App.js – manages isDarkMode state and theme toggling

- App.css – defines dark-theme and light-theme styles

- Ensuring consistency of dark/light theme across all components

- Making sure the stored preference applies correctly on reload


# Feature 3: Calculation History

**User Story**

As a user, I want to see a history of my previous calculations so I can refer back to them easily.

**Technical Requirements**

- Store history as an array of string entries

- Display history entries in a scrollable list

- Provide a clear history button

**Files Likely to be Modified**

- App.js – appends entries to history, displays them, and clears on button click

- App.css – styles the history section

- calculate.js – ensures output is correctly returned for saving

**Potential Challenges**

- Deciding when to add items to history (e.g., only on = or function buttons)

- Handling long histories (e.g., max items or scroll)