Technical Design Document for Calculator Implementation
(By: Justin Mulroney for Vial)

Introduction:
This technical design document outlines the architecture, functionality, and implementation of a basic calculator application with a simple temporary login. The application should support the standard arithmetic operations such as addition, subtraction, multiplication, and division as well as additional features found on calculators and allow the user to login or signup.

Architecture:
The calculator application will be designed as a single-page web application using HTML, CSS, and JavaScript (React) . The user interface will be designed using HTML and CSS, while the JavaScript will handle the logic and calculations.

Functionality:
The calculator will have a user interface consisting of a display area and a set of buttons for each of the arithmetic operations, along with buttons for decimal point, clear, and equals. When the user presses a button, the corresponding action will be triggered and displayed on the calculator display.

Implementation:
The calculator will be implemented using JavaScript functions to handle the arithmetic operations. The following functions will be created:
• add(x, y): takes two numbers as input and returns their sum.
• subtract(x, y): takes two numbers as input and returns their difference.
• multiply(x, y): takes two numbers as input and returns their product.
• divide(x, y): takes two numbers as input and returns their quotient.
• operate(operator, x, y): takes an operator (+, -, *, or /) and two numbers as input and returns the result of the corresponding arithmetic operation.
• power(x,y): takes two numbers as inputs and raises the first number to the power of the second
• square root(x): takes a number and returns the square root
• percent (x) : takes a number and returns the percentage version of it
• M+ (x): takes the current number being typed and adds it to the memory value
• M- (x): takes the current number being typed and subtracts it from the memory value
• MR: displays the current value in memory to the screen
• MC: clears the value in the memory
• history: displays all past calculations
• C: clears all values on the display
• CE: deletes the most recent character in the display

Conclusion:
The implementation of a basic calculator requires a solid understanding of JavaScript functions, HTML, and CSS. With the implementation described above, users will be able to perform standard arithmetic operations on the calculator application as well as sign in, signup and log out.