Vial Design Document: Calculator

Table of contents

Problem statement

Create a simple calculator with basic arithmetic functions.

Requirements

- The calculator should have a browser-based user interface (ie it can be opened using a web browser)
- The calculator should have a number pad with digits 0-9 and decimal point.
- The calculator should have buttons for addition, subtraction, multiplication, and division.
- The calculator should have a display that shows the input and the result of the calculation.
- The calculator should follow the order of operations (PEMDAS).
- The calculator should have a way to sign up with a username and password
- The calculator should have a way to log in with username and password

- The calculator should be usable with or without authentication
- The calculator should have a navigation bar or panel which displays the user's authentication status
- The calculator should have the following features
    - Memory functions (M+, M-, MR, MC)
    - Percentage function (%)
    - Square root function (√)
    - Exponential function (^)
    - History function

Proposed solution

1. User should be able to open the calculator in a web browser
2. User should be able to use a number pad with digits 0-9 and decimal point
3. User should be able to use buttons for addition, subtraction, multiplication, and division
    - + (addition)
    - - (subtraction)
    - × (multiplication)
    - ÷ (division)
4. User should be able to enter an input and the result of the calculation
    - User can use the delete/backspace key to delete one character at a time from the right
        1. Using a keyboard, backspace with the delete button
        2. Using a touchscreen, on a phone or tablet, tap into the display then use the virtual keyboard delete button
    - User can use their keyboard to type in:
        - 0-9

    - Decimal Point (.)
    - Parentheses ()
    - Exponent (^)
    - Multiplication (*)
    - Division (/)
    - Addition (+)
    - Subtraction (-)
    - Percentage (%)
    - pressing the enter key button will submit the user input
5. User should be able to enter a calculation and the result should follow the order of operations (PEMDAS) (Parentheses, Exponent, Multiplication, Division, Addition, Subtraction)
    - = (pressing the enter key or "=" button will submit the user input)
    - Parentheses "()"
    - Exponent (^)
    - Multiplication (×)
    - Division (÷)
    - Addition (+)
    - Subtraction (-)
6. User should be able to sign up with a username and password
    - Username Field
    - Password Field
    - Confirmation Button to submit the form and sign up account
7. User should be able to log in with username and password
    - Username Field
    - Password Field

- Confirmation Button to submit the form and login to their account

8. User should be able to use the calculator with or without authentication
9. User should see a navigation bar or panel which displays the user’s authentication status
   - Authentication Status is shown only when authenticated
   - User should see “Authenticated” in the top right below their username
10. User should be able to use operations such as:
    - Memory functions (M+, M-, MR, MC)
      - User starts session with a calculator memory at 0
      - Each time the user presses *M+* the number on the display is added to the number in the calculator memory
      - Each time the user presses *M-* the number on the display is subtracted from the number in the calculator memory
      - User can use *MR* button to recall the number in the calculator memory
      - User can use *MC* button to zero out the memory
    - Percentage function (%)
      - User can find the percentage of the number to the left of the percentage sign
        - example: 5% would return 0.05
    - Square root function (√)
      - User can find the square root of a number inside of the square root
        - example: √25 would return 5
    - Exponential function (^)
      - User can find the exponent of a number to the left of the exponent sign
        - example: 5^2 would return 25
    - History function

    - User can cycle through the history of previous memory calculations be pressing the H button
- AC function
    - User can clear the calculator’s input memory using the AC button

Architecture

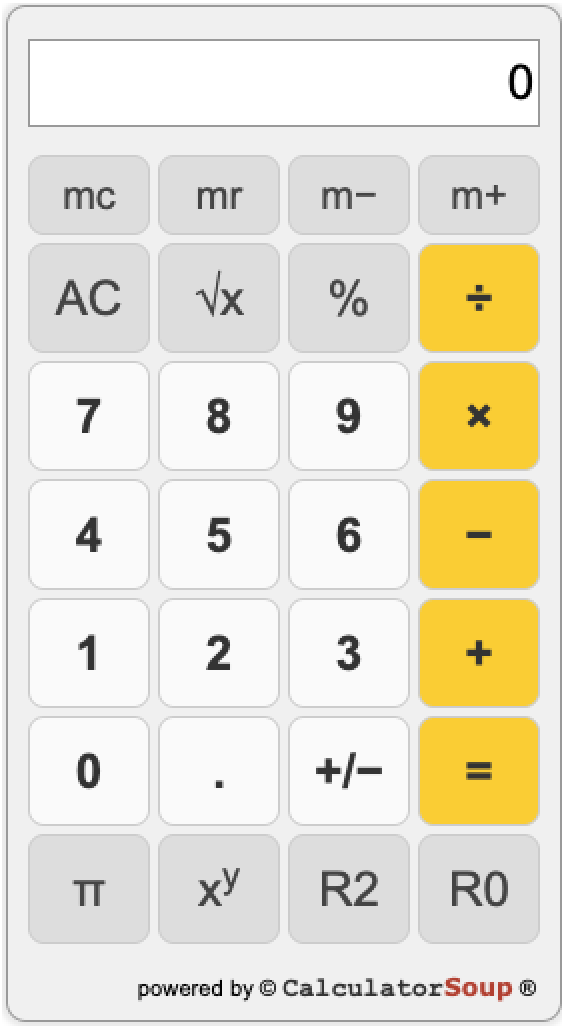
The calculator application will be designed using a Single Page Application (SPA) design.

- React Frontend Framework
    - why? fast, performant and easy to use
- Material UI (https://mui.com/material-ui/getting-started/overview/) for component library
    - easy to use and design an application
- Language: TypeScript
- Authenticated user is just stored using localStorage on frontend

Backend:

- Express/Node.js
- Knex for easier SQL querying
- Language: JavaScript
- User Passwords hashed using crypto library
- After signing up, user entity is created on backend

User Interface



A mock calculator design that contains most of the desired functionality is shown above.

The user interface of the calculator application will consist of the following elements:

- **Display**: The input display will show the current value of the calculation
    - User can type in all of the required user inputs detailed in the proposed solution
- **Buttons**: The buttons will allow the user to input numbers and perform operations as specified in the proposed solution
    - MC
    - MR
    - M-
    - M+

  - AC
  - √
  - %
  - ^
  - H (History)
  - Decimal (.)
  - Digits (0 - 9)
  - +
  - -
  - x
  - ÷
  - = (perform the calculation)
- Navigation bar contains
  - Logo on top left - clicking on this leads back to the main page
  - Authenticated Users are displayed with email/authentication status and have a logout button to clear their session
  - Login Button, Sign Up button

Success Criteria

How will you validate the solution is working correctly?

Testing with Jest

Addition
Subtraction

Multiplication

Division

Repeating Operations

Roots, Exponents and Power Functions

Order of Operations

Additional Tests

Percentage Operations

UI Testing
Functional requirements testing

Conclusion

Vial’s calculator application will be created using React, TypeScript and Material UI on frontend.

On the backend we used JavaScript, Node.js, Express, Knex with a basic PSQL database.

The user interface will consist of a user input for the calculator, calculator buttons and a navigation bar. Additional Screen will be created for the Login and Create an Account screens.

The calculator application will be user-friendly, fast, and FUN!