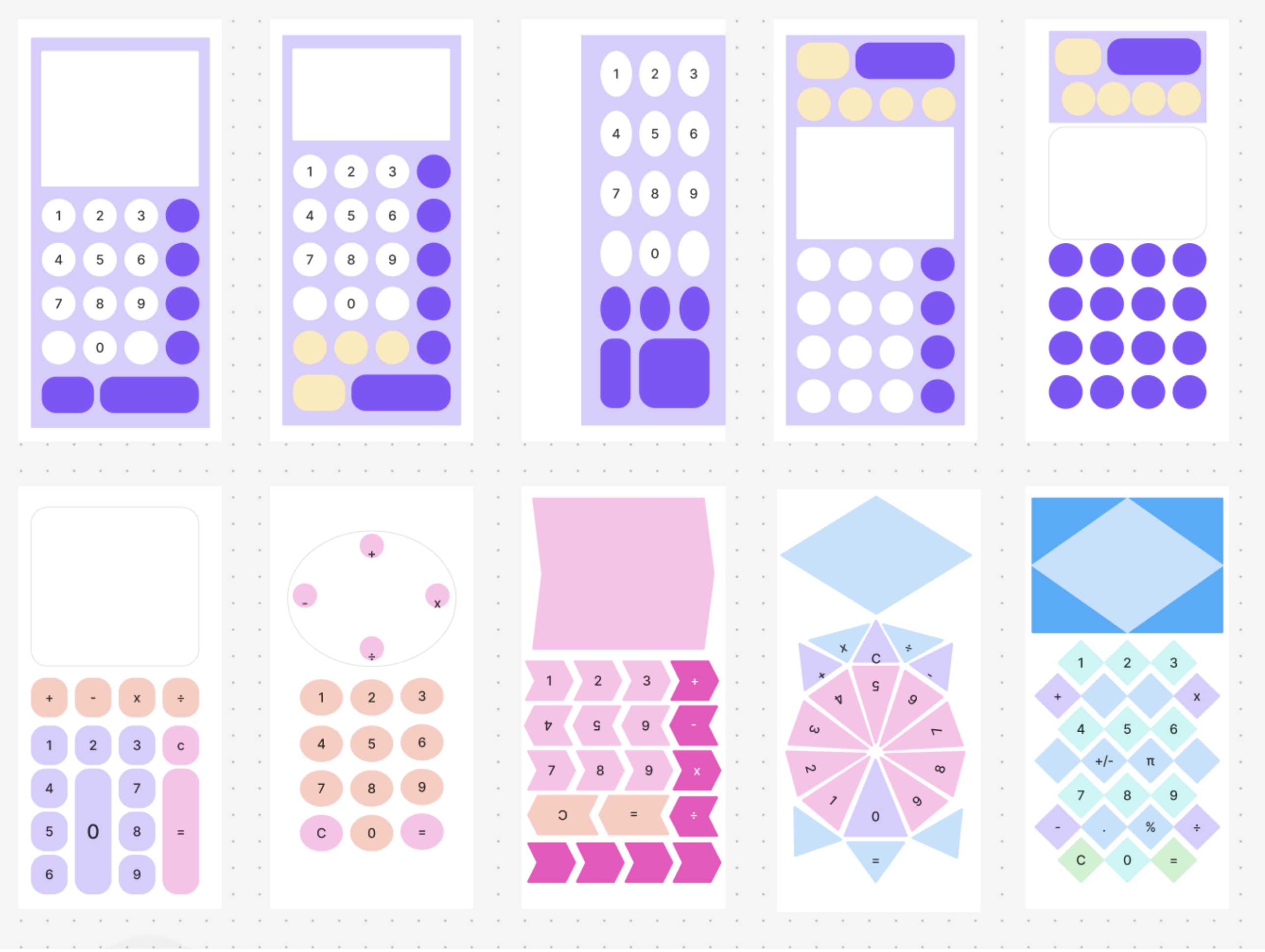
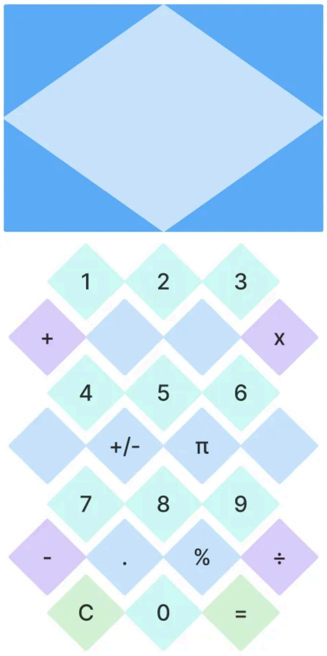


1 2 3
4 5 6
7 8 9
0
+ - x ÷
c
=
C
+/-
π
%
.



J: Hi! I need some code for a functioning online calculator. It is a simple calculator, with Arabic numerals 1-0, calculate and clear buttons, a display box, and addition, subtraction, multiplication, and division capabilities. There are additional functions to calculate with decimals, percentages, negative numbers, and pi. I've attached a mock-up for how to style the calculator. Is this something you can do?

I'll help you create a React component for a calculator based on your mockup. This will include all the functionality you've described along with styling to match your design.

I've created a fully functional calculator component that matches your mockup. Here are the key features:

1. Basic Operations:
    - Addition, subtraction, multiplication, division
    - Support for decimal numbers
    - Percentage calculation
    - Positive/negative toggle
    - Pi constant
    - Clear function
2. Display Features:
    - Shows current input or result
    - Handles overflow with scrolling
    - Updates dynamically as you type
3. Styling:
    - Matches your mockup's color scheme
    - Responsive grid layout

    - Hover effects for better user interaction
    - Color-coded buttons by function type

To use this component, you just need to import and render it in your React application. The calculator uses Tailwind CSS for styling, matching the colors from your mockup.

Would you like me to:

- Add any additional mathematical functions?
- Modify the styling or layout?
- Add keyboard support for number input?
- Include error handling for division by zero?

J: Great start! If I can ask for a couple of layout changes - can we place the add and subtract buttons where the pi and negative number buttons currently are, so those functions are all grouped together? We can move the = button below all of those functions, and place the decimal button to that space instead, to keep function types grouped together. Would we also be able to add the keyboard support for number input and error handling for dividing by zero?

I'll reorganize the layout and add those additional features.

I've made the following changes and improvements:

1. Layout Changes:
    - Moved the arithmetic operators (×, ÷, -, +) to be grouped together on the right side
    - Grouped the special function buttons (+/-, ., %) together
    - Repositioned the = button to the bottom right
    - Kept π with the other special functions
2. Added Keyboard Support:
    - Number keys (0-9)
    - Operators (+, -, *, /)
    - Enter or = for calculating
    - Decimal point (.)
    - Escape key for clear
3. Added Error Handling:
    - Division by zero now shows an error message
    - Added error state management
    - Error messages appear above the display
    - Clear button (or Escape key) removes errors
4. Additional Improvements:
    - Added floating-point precision handling to avoid long decimal results
    - Improved state management for error conditions

    - Added hover effects for better user feedback
    - Maintained consistent styling with your original design

The calculator now supports both mouse clicks and keyboard input, making it more user-friendly. Would you like me to make any additional adjustments to the layout or functionality?

J: Amazing! One more change to the layout - can we switch the locations of the decimal and pi buttons?

I'll swap the positions of the decimal and pi buttons.

I've swapped the positions of the decimal (.) and pi (π) buttons. The decimal button is now in the bottom row next to the 0, and the pi button is in the row above with the other special function buttons. All functionality remains the same. Is there anything else you'd like me to adjust?

J: No more changes, this is perfect! Thank you so much.

You're welcome! The calculator is now fully functional with a clean, organized layout that groups similar operations together. It includes all the features you requested:

- Basic arithmetic operations
- Special functions (π, %, +/-)
- Decimal support
- Error handling for division by zero
- Keyboard input support
- Clear organization with similar functions grouped together

The component is ready to be integrated into your React application. Just import it and you're good to go! Let me know if you need any help implementing it or if you want to make any changes in the future.