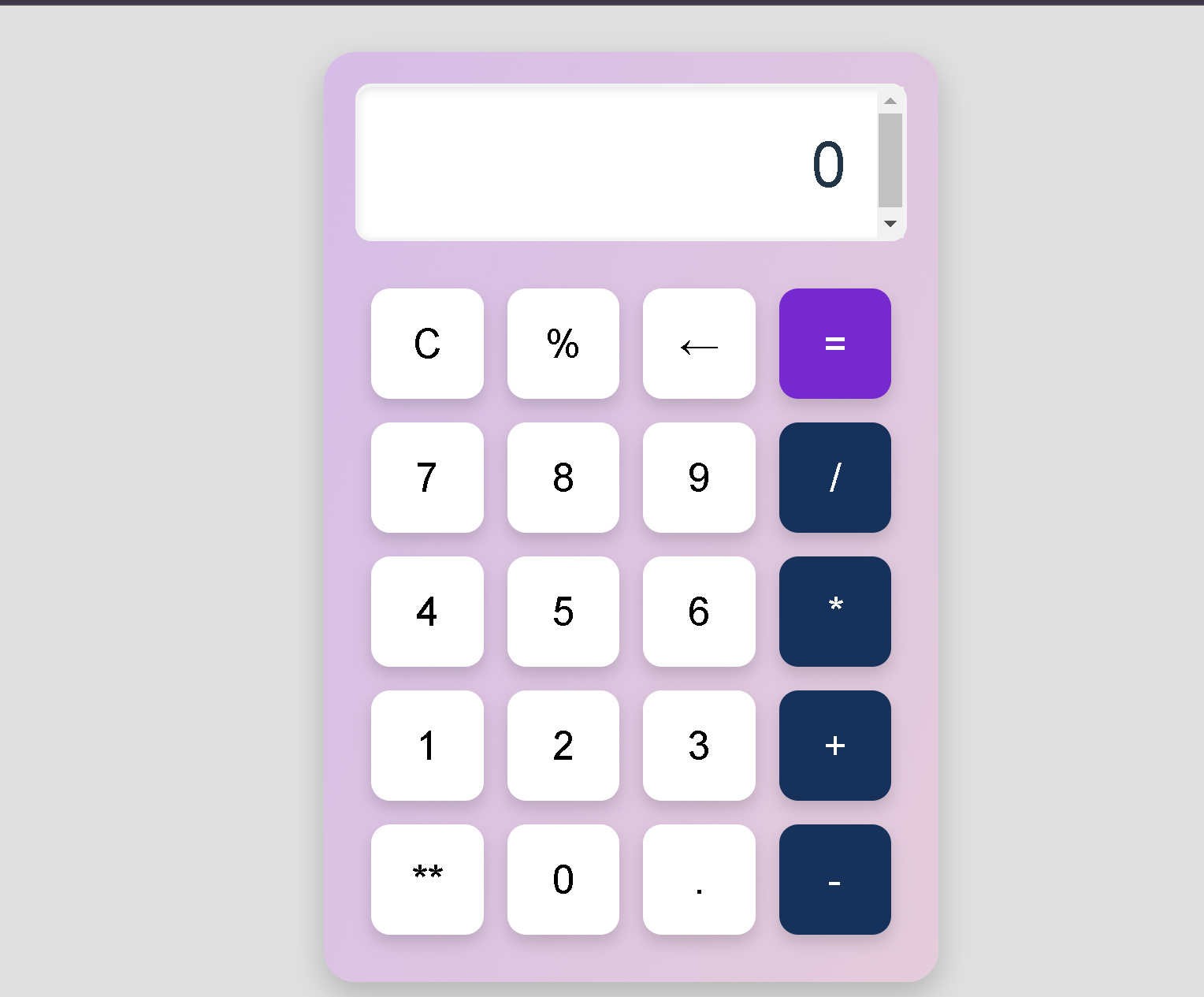
**Report**

1. **Task Description**

Create a simple calculator component that performs basic arithmetic operations and use reuable common component for the button

1. **Task Output Screenshot**



1. **Widget/Algorithm Used In Task**

* ***React Components:***

1. Calculator Component: In simple words, it acts like the main component where all the handling of the display of the calculator, button clicks, and calculations happen.
2. Button Component: A reusable component for each calculator button which accepts `value` and `handleClick` props .

* ***React Hooks:***

1. useState:  
   - display: Holds the last result displayed on the calculator display. Each time a button is clicked, this is updated along with holding the result of the computation in this state

* ***Methods:***

1. Click(val):  
   It processes the action of buttons on a click event.  
   If it is a click on `'='`, then call `calculateResult()` to compute the expression.  
   - For `'C'`: Calls `clearDisplay()` to clear the display.  
   - For `'←'`: Clears the last character.  
   - For all other values, append that value to the current display.
2. calculateResult():  
   Uses `eval()` to parse the currently entered mathematical expression in the display. If it returns an error, this function catches the error and resets the display to `'Error'`.
3. clearDisplay():  
   Clears the display by resetting the state to an empty string.

* ***Button Values :***

1. buttons Array:  
   It contains the values for the calculator buttons: digits (`0-9`), operators (`+`, `-`, `\*`, `/`, `%` etc.), special keys (`C`, `←`, `=`), and exponentiation (`\*\*`).

* ***HTML Structure:***

1. Display:  
   `<div className="display">`: It contains the current expression or result. When the display is empty, it defaults to `'0'`.
2. Buttons:  
   `<Button>`: Render each button according to the value from the `buttons` array.  
   - Each button will have an onclick event set to the `Click` function, which determines what the button does.

* ***CSS Styling (in `App.css`):***

1. **Calculator Layout:** .calculator-container: Flex column layout, 350px by 550px size, centre page, gradient background, rounded corners and shadow overlay applied
2. Display : .display: High font size to display an input/output data, White background with text aligned right and long inputs are scrollable.
3. Buttons Layout : .buttons: Grid layout with column count of 4 and spacing applied between buttons.
4. Button Styling :.calculator-button: Size set uniform and rounded corners are applied with subtle box shadows. Hover/active effects are applied.  
   Special button is "=", "/" and "\*" with background color different, which are PURPLE/BLUE Color. Hover effect is also kept upon that.
5. Body: Flex-centered layout, full viewport height with light gray background.