Here's the full code in a single document format that combines the HTML, JavaScript, and the logic for both the basic and scientific calculators within a Salesforce LWC (Lightning Web Component).

Full Code for CalculatorApp (LWC)

---

calculatorApp.html (Main Render Component)

html

Copy code

<!-- Main render component -->

```
<template>
    <lightning-card title="Render Demo">
        <div class="slds-m-around_medium">
            <lightning-button label="CALCULATOR" variant="brand"
onclick={handleBasicCalculatorClick}></lightning-button>
            <lightning-button label="SCIENTIFIC CALCULATOR"
variant="brand"
onclick={handleScientificCalculatorClick}></lightning-button>
        </div>
    </lightning-card>
</template>
```

<!-- Basic Calculator template -->

```
<template if:true={isBasicMode}>
    <!-- Calculator interface for basic arithmetic operations -->
    <div class="calculator">
        <h2>CALCULATOR</h2>
        <div>
            <!-- Add the display and buttons layout for standard
operations here -->
        </div>
    </div>
</template>
```

<!-- Scientific Calculator template -->

```
<template if:true={isScientificMode}>
    <!-- Scientific calculator interface with additional advanced functions -->
    <div class="scientific-calculator">
        <h2>SCIENTIFIC CALCULATOR</h2>
        <div>
            <!-- Add the display and buttons layout for scientific operations here -->
        </div>
    </div>
</template>
```

---

calculatorApp.js (JavaScript Logic)

javascript

Copy code

```
import { LightningElement } from 'lwc';
import basicTemplate from './calculator.html';
import scientificTemplate from './scientificCalculator.html';
import calculatorRender from './calculatorRender.html';

export default class CalculatorApp extends LightningElement {
    chosenTemplate = null;
    result = '';
    isBasicMode = false;
    isScientificMode = false;

    render() {
        return this.chosenTemplate === 'Basic' ? basicTemplate :
               this.chosenTemplate === 'Scientific' ? scientificTemplate :
               calculatorRender;
    }

    handleBasicCalculatorClick() {
        this.chosenTemplate = 'Basic';
        this.isBasicMode = true;
        this.isScientificMode = false;
        this.result = ''; // Reset on switching
    }

    handleScientificCalculatorClick() {
        this.chosenTemplate = 'Scientific';
        this.isScientificMode = true;
        this.isBasicMode = false;
        this.result = ''; // Reset on switching
    }

    handleButtonClick(event) {
        const value = event.target.label;
        this.result += value;
    }

    handleCalculate() {
        try {
            this.result = eval(this.result);
        } catch (error) {
            this.result = 'Error';
        }
    }

    handleClear() {
        this.result = '';
    }

    handleBackspace() {
        this.result = this.result.slice(0, -1);
    }

    handleSquare() {
        this.result = Math.pow(parseFloat(this.result), 2);
    }

    handleSqrt() {
        this.result = Math.sqrt(parseFloat(this.result));
    }

    handleSin() {
        this.result =
Math.sin(this.toRadians(parseFloat(this.result)));
    }

    handleCos() {
        this.result =
Math.cos(this.toRadians(parseFloat(this.result)));
    }

    handleTan() {
        this.result =
Math.tan(this.toRadians(parseFloat(this.result)));
    }

    handleLog() {
        this.result = Math.log10(parseFloat(this.result));
    }

    handleCbrt() {
        this.result = Math.cbrt(parseFloat(this.result));
    }

    handleClose() {
        this.chosenTemplate = null;
        this.result = '';
    }

    toRadians(value) {
        return (value * Math.PI) / 180;
    }
}
```

---

calculatorRender.html (Base Template for Rendering)

html

Copy code

```
<template>
    <!-- Base template layout for the calculator -->
    <lightning-card title="Calculator App">
        <div class="slds-m-around_medium">
            <lightning-button label="CALCULATOR" variant="brand"
onclick={handleBasicCalculatorClick}></lightning-button>
            <lightning-button label="SCIENTIFIC CALCULATOR"
variant="brand"
onclick={handleScientificCalculatorClick}></lightning-button>
        </div>
    </lightning-card>
</template>
```

Notes:

- This LWC project allows you to switch between a **Basic Calculator** and a **Scientific Calculator** using buttons.
- It handles operations like addition, subtraction, square roots, trigonometric functions (sin, cos, tan), logarithms, and cube roots.
- The `render` method in `calculatorApp.js` dynamically loads the appropriate template (`basic Template` or `scientific Template`) based on the mode selected.
- The `handleButtonClick` method handles the input for calculator buttons, and `handleCalculate` evaluates the mathematical expression entered.