**Calculator Documentation**

**1. INTRODUCTION**

This document is to describe the design process for a calculator application. This document is intended to detail the:

* Objective of the Application
* Tools and Technologies used
* Design Steps
* Implementation

**2. OBJECTIVE**

The objective of this application is to perform simple arithmetic operations. Only one operation can be performed at a time. The calculator can evaluate addition, subtraction, multiplication, division, and modulo operations. If an expression has been evaluated, the user can select any of the operators to perform another calculation on the result of the previous calculation.

**3. TOOLS AND TECHNOLOGIES USED**

* Visual Studio Code
* Angular/TypeScript
* HTML
* CSS
* Bootstrap

**4. DESIGN STEPS**

This application was created by using Angular along with Bootstrap and CSS to provide responsiveness and styling.

Interpolation was used to bind data the user has input onto the calculator’s display. The screen will display what the user has clicked, and if it’s an invalid input, then a detailed error message will be shown.

Event binding was used for each click event by the user. Each button is bound to a function that will perform its respective task when clicked.

**5. IMPLEMENTATION**

Each button is bound to a click event, so when a button has been selected, its respective value will be stored as a string in a string array of size 3, so only one operator may be selected per operation. The expressionBuilder() function will maintain the values and operator that are chosen. This function is called when the user clicks on a digit or on “NEG”. This function also validates if the user has entered more than one decimal character and returns an error if there is more than one. The “NEG” button will convert the positive number to a negative. When the user clicks on “=”, the showResult() function will be called. This will first check to make sure all values are valid by calling on validateSelections(). If a value is missing, and error will be returned to the display. Once all the required values are chosen, showResult() will perform calculate() which checks the operator that was selected and perform the calculation. The result will then be displayed to the user via interpolation. The “AC” button will clear all values selected including the operator and the display screen will be cleared.