Approach Overview

Before writing any code, I wrote out numbers in English to understand a pattern in how numbers are said phonetically in order to leverage it in my code. This showed me that numbers can be grouped in threes and handled separately (e.g., 125 => ONE HUNDRED AND TWENTY-FIVE, && 125000 => ONE HUNDRED AND TWENTY-FIVE THOUSAND). This pattern could then be leveraged by looping through from the end of the user input to the start of the string (`i = userInput.Length; i > 0; i-= 3`) to handle each chunk separately.

A dictionary was used to equate single (0-9) and double digit (10 - 19, 20, 30,...,90) numbers to define their English counterparts. A list was also used to handle the chunk position words so that they can be appended into the correct position in the processed output string:

- Digits 1-999: Position Word = “”
- Digit 10^3 - 10^5: Position Word = “THOUSAND”
- Digit 10^6 - 10^8: Position Word = “MILLION”
- Digit 10^9 - 10^11: Position Word = “BILLION”
- Digit 10^12 - 10^14: Position Word = “TRILLION”

Decimal inputs were split into 2 separate strings and processed separately so the strings “DOLLARS” and “CENTS” could be appended to the end of each before adding them back together with an “AND” string separating them. Whereas, an integer string input was handled by itself with “DOLLARS” appended to the end of it.

To ensure a correct input is passed to the server side logic, a event listener was added to the convert button that checks the following:

- If there’s leading zeros in both integer string or decimal string
    - Removes all leading zeros if the input is an integer string.
    - Removes all but 1 zero before the decimal point in a decimal string input.
- If there’s white spaces present => removes them.
- Appends a ‘0’ if there's only one digit after the decimal
- If the user input matches the acceptable pattern
    - If not, it provides the following on the UI: "Please enter a valid number.”
- If the amount of digits before the decimal point is > 15 (accepted: 1 - 999 999 999 999 999)
    - If not, it provides the following error message on the UI: "Input exceeds the maximum allowed digits before the decimal."