Why This Way

# Tech stack choice

## Back-End – Web API

ASP.NET C# is my preferred technology as I’m very fluent with it. With Visual Studio it’s easy to create a project from scratch. Plus, the community’s content on this language is huge. With default built-in Kestrel when building it, this Web API can be executed as standalone, instead of hosting on other platforms (E.g., IIS).

## Front-End – Web UI

I chose Next.JS as my Web UI, it’s a react based framework having amazing packaging of good stuffs to produce high-quality Web UI. It is well documented, with a decent size of community around it, hence, it will be a good starting point as it also will be highly scalable.

# Solutioning

Delving into the problem of “converts numerical input into words”, I was looking for a pattern, by identifying a pattern, its easier to come up with a solution.

From my analysis, the pattern can be described visually as below.

A screenshot of a game

Description automatically generated

First there are two parts, dollar, and cent.

## Cent part

Cent is simple, just up to 2 digits. If we can solve the dollar part, this part will be just matter of plug the solution in.

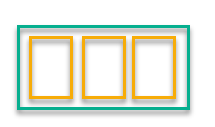
## Dollar part

For Dollar though, theoretically it can go up to infinity digits. However, because the mission is to turns the numbers into words, we need to have names for positions of each digit or digits’ group.

Looking at the pattern, if we traverse from the last digit to the first digit, for every 3 digits, it belongs to a group that has a name. E.g., First 3 digits is HUNDRED, second 3 digits is THOUSAND and so on.

## 3-Digits Problem

From here if we can solve the 3 digits wording problem, we will have a good start.



First digit in 3 digits, if it a zero we don’t have to spell it out, if it more than 0, for example 1, it will spell as “ONE” then suffix with “HUNDRED”, “ONE HUNDRED”. For second and third digit, it will be a 2-Digits problem, if it less than 20, it will have its own special name, for example 17 will be “SEVENTEEN”, for 9 it will be “NINE”, if is it more than 19 and can be divided by 10 without remainder it also have its special name, for example, 30 is “THIRTY”, 80 is “EIGHTY”, and if it division by 10 has a remainder that remainder will be append the its name bridge with a dash, for example 34 divide by 10 with remainder of 4, its spell out as “THIRTY-FOUR”.

So, in totality.

1. Extract first digit, if it more than 0, spell the number and append with “HUNDRED”
2. Extract the last 2 digits, if it is less than 20, spell out its special name, else divide by 10 to get the remainder, then. Spell out the second digit special name append with remainder name bridge by a dash.
3. Join the name in Step 1 and 2 with the word “AND”. For example, 125 is “ONE HUNDRED **AND** TWENTY-FIVE”.
4. This sample solution can be applied to Cent part, except Cent will always be 2-digits problem.

## Bigger Number

A black and yellow rectangles with white text

Description automatically generated

Next, let’s introduce the THOUSAND group into the game. THOUSAND group is just another 3-digits group, it’s simply gotten the name of the 3-digits then append word “THOUSAND” to it, then append it with HUNDRED group bridge with the word “AND”.

For example,1245.

A black and yellow rectangles with white and purple rectangles

Description automatically generated

001 is in THOUSAND group and 245 in HUNDRED group. THOUSAND group will produce “ONE” append it with the word “THOUSAND” and HUNDRED group will produce “TWO HUNDRED AND FORTY-FIVE”. Notice that only HUNDRED group doesn’t need append with word “HUNDRED”.

Join both groups with word “AND”, we will get “ONE **THOUSAND** AND **TWO HUNDRED AND FORTY-FIVE**”.

Another pattern here is, direction is from left to right when spelling out the group’s words and join the groups with word “AND”.

## Combine Dollar and Cent

To combine dollar and cent, is a final touch up.

A screenshot of a game

Description automatically generated

Get wording from all 3-digits groups from dollar part, append with word “DOLLARS” (or “DOLLAR” for value 1 or less) then append with wording 2-digits group from cent part then append with word “CENTS” (or “CENT” for value 1 or less).