**Advanced-Table-Component**

FetchCategoryValue(companyName: string): number[] {}

* Takes a companies name
* References variable: categorySelected
* Returns an array of numbers representative of the Category chosen

**Conditionals:**

1. Quarter v Year
2. yearRange + Ascending v Descending
3. Precise V Uniform
4. Quarter V Year

This perhaps will not be an issue. There are supposed to be 4 values per year – Q4 3 2 1. If we are checking to see if we are still within the boundaries of the year range we will keep pushing into the array new sets of results

1. yearRange + Ascending v Descending

Ascending and Descending, the sorting of the userList of Companies will be handled by a separate function whenever the Asc/Des Arrow key button is pressed. Likewise the yearList will be flipped as well.

Do we need to flip the yearList ?

It would make more sense to just flip the displayList and leave the yearList vanilla as a reference.

If we are starting from 2020 have results from 2000 and wish to set out yearRange to 2015 the following actions will take place:

Ascending

From 2000 to 2015

Descending

From 2020 to 2015

Remember, the userList of Companies we are pulling Results from is already sorted, so we just need to push values into the new array we are forming in this function.

An algorithm might be as follows: \*direction – descending if true

If(direction && companyResults.year >= yearRange.value || !direction && companyResults.year <= yearRange.value)

Summary of the above:

IF we are Descending and the current Results year is Greater or Equal to the Range we will push the value. Likewise, if we are Ascending and the current Results year is Less than or Equal to the Range we will push the value.

Once out of Range we break the loop.

1. Precise V Uniform

Problem. What if we have 2 companies. Company A’s results stop at 2021 while B’s carry on to 2022, or A’s stop at Q2 2022 and B’s continue to A4 2022 ?

When we display this data we want to make sure each period is displayed in its appropriate date. Therefore, we will require a conditional statement to check if we want precise or uniform.

For Precise we will have to compare the current year to the latest year in the yearList. Because the yearList has the latest year that any company can have it is better to check than the displayList.

Moreover, we do not need to worry about trailing years as an empty square is adequate. Our only concern is the leading years which a Company may not have data for but needs a 0 to act as a buffer.

**Problem**:

* If we are evaluating in Quarters Vs Years how can we tell how many 0’s to put in, especially if 2 Quarters of that year are occupied vs 3, or 1. 4
* And how to tell if we are ascending or descending ? Because we are only concerned about the leading data sets not the trailing. What is Company A’s data goes back to 1990 whilst B’s only to 2000, we will have to backfill B’s data to Precisely display the two data sets.
  + But does the Direction really matter if the UserList of Companies is already sorted for us. All we are really concerned with is filling in 0’s until we hit the first value of a matching year.

**Potential Solution**:

Separate Function to frontFill

Function fillUp(array: numbers, qtr as company.results[0].quarter) : array: number {

If(precise && quarter){

If(direction){

// Descending

Check qtr

} else

{

//Ascending

}

}

If(precise && year) // same as above

Else (uniform) // newest to oldest or vice versa , we do nothing here

}