**Module 2 Challenge: Refactored VBA Code**

**By**

**Omar J. Nasir**

**Module2 Challenge**

**Deliverable2:**

**Overview of the Project**

The project is a learning exercise in determining the benefits and advantages of refactored code. Refactored Code is code, which has been improved from an earlier code to make it run more efficiently. The run times of the refactored code are tabulated.

The background of the project is to explore green energy stock performance by analyzing financial data using VBA. Steve, a fresh financial graduate, opens up a company to manufacture silicon wafers for solar panels. The company does great the first year but suffers a loss in the second year. So, it would be difficult to recommend its stock to a potential customer.

**Results**

**Graphical user interface, application, table, Excel

Description automatically generated**

The Code ran in 0.2 and 0.3 seconds and is a considerable improvement over the past.

The following is the code. It is also being loaded separately.

Attribute VB\_Name = "Module1"

Sub AllStocksAnalysisRefactored()

Dim startTime As Single

Dim endTime As Single

yearValue = InputBox("What year would you like to run the analysis on?")

startTime = Timer

'Format the output sheet on All Stocks Analysis worksheet

Worksheets("All Stocks Analysis").Activate

Range("A1").Value = "All Stocks (" + yearValue + ")"

'Create a header row

Cells(3, 1).Value = "Ticker"

Cells(3, 2).Value = "Total Daily Volume"

Cells(3, 3).Value = "Return"

'Initialize array of all tickers

Dim tickers(12) As String

tickers(0) = "AY"

tickers(1) = "CSIQ"

tickers(2) = "DQ"

tickers(3) = "ENPH"

tickers(4) = "FSLR"

tickers(5) = "HASI"

tickers(6) = "JKS"

tickers(7) = "RUN"

tickers(8) = "SEDG"

tickers(9) = "SPWR"

tickers(10) = "TERP"

tickers(11) = "VSLR"

'Activate data worksheet

Worksheets(yearValue).Activate

'Get the number of rows to loop over

RowCount = Cells(Rows.Count, "A").End(xlUp).Row

'1a) Create a ticker Index

tickerIndex = 0

'1b) Create three output arrays

Dim tickerVolumes(12) As Long

Dim tickerStartingPrices(12) As Long

Dim tickerEndingPrices(12) As Long

''2a) Create a for loop to initialize the tickerVolumes to zero.

For i = 0 To 11

tickerVolumes(i) = 0

Next i

'2b) Loop over all the rows in the spreadsheet.

For i = 2 To RowCount

'3a) Increase volume for current ticker

tickerVolumes(tickerIndex) = tickerVolumes(tickerIndex) + Cells(i, 8).Value

'3b) Check if the current row is the first row with the selected tickerIndex.

If Cells(i - 1, 1).Value <> tickers(tickerIndex) Then

tickerStartingPrices(tickerIndex) = Cells(i, 3).Value

End If

'3c) check if the current row is the last row with the selected ticker

'If the next row's ticker doesn't match, increase the tickerIndex.

If Cells(i + 1, 1).Value <> tickers(tickerIndex) Then

tickerEndingPrices(tickerIndex) = Cells(i, 6).Value

'3d Increase the tickerIndex.

tickerIndex = tickerIndex + 1

End If

Next i

'4) Loop through your arrays to output the Ticker, Total Daily Volume, and Return.

For i = 0 To 11

Worksheets("All Stocks Analysis").Activate

‘ the code will not work, if All Stock Analysis sheet is not there

Cells(4 + i, 1).Value = tickers(i)

Cells(4 + i, 2).Value = tickerVolumes(i)

Cells(4 + i, 3).Value = ((tickerEndingPrices(i) - tickerStartingPrices(i)) / tickerStartingPrices(i))

Next i

'Formatting

Worksheets("All Stocks Analysis").Activate

Range("A3:C3").Font.FontStyle = "Bold"

Range("A3:C3").Borders(xlEdgeBottom).LineStyle = xlContinuous

Range("B4:B15").NumberFormat = "#,##0"

Range("C4:C15").NumberFormat = "0.0%"

Columns("B").AutoFit

dataRowStart = 4

dataRowEnd = 15

For i = dataRowStart To dataRowEnd

If Cells(i, 3) > 0 Then

Cells(i, 3).Interior.Color = vbGreen

Else

Cells(i, 3).Interior.Color = vbRed

End If

Next i

endTime = Timer

MsgBox "This code ran in " & (endTime - startTime) & " seconds for the year " & (yearValue)

End Sub

The results of Return on Investment can also be calculated, without using the code. But for this project, since the learning objective was to familiarize oneself with VBA Code, so the code was written/developed.

**Summary**

Refactored code has advantages in computer time saved and it is more easy and simple to read. The other code is sometimes more complicated, because of loops and nested loops involved. So, when simplifying the code, a step wise refinement and decomposition was needed in the form of comments, which were provided.