Sub RestructureFountainData2023()

Dim folderPath As String

Dim newFolderPath As String

Dim inputFile As String

Dim outputFile As String

Dim ws As Worksheet

Dim newWb As Workbook

Dim lastRow As Long

Dim i As Integer

Dim monthNames As Variant

Dim oldFilePath As String

Dim newFilePath As String

Dim r As Long

Dim activeFilePath As String

Dim activeWs As Worksheet

Dim activeWb As Workbook

' Define the folder path (Fountain 2023 folder on the desktop)

folderPath = "C:\Users\USER\Desktop\Fountain\2023\"

' Create a new folder called 2023new

newFolderPath = "C:\Users\USER\Desktop\Fountain\2023new\"

If Dir(newFolderPath, vbDirectory) = "" Then

MkDir newFolderPath

End If

' Define month names for file naming

monthNames = Array("jan", "feb", "mar", "apr", "may", "jun", \_

"jul", "aug", "sep", "oct", "nov", "dec")

' Open the "Active" file

activeFilePath = "C:\Users\USER\Desktop\Fountain\Active.xlsx"

If Dir(activeFilePath) = "" Then

MsgBox "Active file not found: " & activeFilePath, vbExclamation

Exit Sub

End If

Set activeWb = Workbooks.Open(activeFilePath)

Set activeWs = activeWb.Sheets(1)

' Disable screen updating and automatic calculations for performance

Application.ScreenUpdating = False

Application.Calculation = xlCalculationManual

' Loop through each month

For i = LBound(monthNames) To UBound(monthNames)

' Define the input file name (e.g., "aug2023.xlsx")

inputFile = monthNames(i) & "2023.xlsx"

outputFile = monthNames(i) & "2023\_restructured.xlsx"

oldFilePath = folderPath & inputFile

newFilePath = newFolderPath & outputFile

' Check if the input file exists

If Dir(oldFilePath) = "" Then

MsgBox "File not found: " & oldFilePath, vbExclamation

GoTo NextMonth

End If

' Open the input workbook

Workbooks.Open Filename:=oldFilePath

Set ws = ActiveWorkbook.Sheets(1)

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

' Create a new workbook

Set newWb = Workbooks.Add

With newWb.Sheets(1)

' Set headers

.Range("A1:D1").Value = Array("ID\_NUMBER", "NAME", "EMPLOYEE", "EMPLOYER")

' Copy names from Column B of the old file to Column B of the new file

ws.Range("B2:B" & lastRow).Copy Destination:=.Range("B2")

' Calculate EMPLOYEE (0.05 \* Salary in Column C) and place it in Column C

For r = 2 To lastRow

' Check if the value in Column C is numeric

If IsNumeric(ws.Cells(r, "C").Value) Then

.Cells(r, 3).Value = ws.Cells(r, "C").Value \* 0.05 ' Calculate Tier 2 contribution

Else

.Cells(r, 3).Value = 0 ' Set to 0 if the value is not numeric

End If

Next r

' Fill EMPLOYER with 0

.Range("D2:D" & lastRow).Value = 0

' Use XLOOKUP to extract ID\_NUMBER from the "Active" file

Dim lookupRange As Range

Set lookupRange = activeWs.Range("G:G") ' SSNIT numbers in Column G of the "Active" file

Dim idRange As Range

Set idRange = activeWs.Range("B:B") ' ID\_NUMBER in Column B of the "Active" file

For r = 2 To lastRow

On Error Resume Next

.Cells(r, 1).Value = Application.WorksheetFunction.XLookup(ws.Cells(r, "A").Value, lookupRange, idRange, "Not Found")

On Error GoTo 0

Next r

End With

' Save the new workbook

newWb.SaveAs Filename:=newFilePath

newWb.Close SaveChanges:=True

' Close the input workbook

ActiveWorkbook.Close SaveChanges:=False

NextMonth:

Next i

' Close the "Active" file

activeWb.Close SaveChanges:=False

' Re-enable screen updating and automatic calculations

Application.ScreenUpdating = True

Application.Calculation = xlCalculationAutomatic

MsgBox "Data restructuring for 2023 completed successfully!", vbInformation

End Sub

NEW CODE FOR ULTRA 3 VALIDATION

Sub RestructureFountainData2023()

Dim folderPath As String

Dim newFolderPath As String

Dim inputFile As String

Dim outputFile As String

Dim ws As Worksheet

Dim newWb As Workbook

Dim lastRow As Long

Dim i As Integer

Dim monthNames As Variant

Dim oldFilePath As String

Dim newFilePath As String

Dim r As Long

Dim activeFilePath As String

Dim activeWs As Worksheet

Dim activeWb As Workbook

Dim sumEmployee As Double ' Variable to store the sum of Column C

' Define the folder path (Fountain 2023 folder on the desktop)

folderPath = "C:\Users\USER\Desktop\Fountain\2023\"

' Create a new folder called 2023new

newFolderPath = "C:\Users\USER\Desktop\Fountain\2023new\"

If Dir(newFolderPath, vbDirectory) = "" Then

MkDir newFolderPath

End If

' Define month names for file naming

monthNames = Array("jan", "feb", "mar", "apr", "may", "jun", \_

"jul", "aug", "sep", "oct", "nov", "dec")

' Open the "Active" file

activeFilePath = "C:\Users\USER\Desktop\Fountain\Active.xlsx"

If Dir(activeFilePath) = "" Then

MsgBox "Active file not found: " & activeFilePath, vbExclamation

Exit Sub

End If

Set activeWb = Workbooks.Open(activeFilePath)

Set activeWs = activeWb.Sheets(1)

' Disable screen updating and automatic calculations for performance

Application.ScreenUpdating = False

Application.Calculation = xlCalculationManual

' Loop through each month

For i = LBound(monthNames) To UBound(monthNames)

' Define the input file name (e.g., "aug2023.xlsx")

inputFile = monthNames(i) & "2023.xlsx"

oldFilePath = folderPath & inputFile

' Check if the input file exists

If Dir(oldFilePath) = "" Then

MsgBox "File not found: " & oldFilePath, vbExclamation

GoTo NextMonth

End If

' Open the input workbook

Workbooks.Open Filename:=oldFilePath

Set ws = ActiveWorkbook.Sheets(1)

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

' Create a new workbook

Set newWb = Workbooks.Add

With newWb.Sheets(1)

' Set headers

.Range("A1:D1").Value = Array("ID\_NUMBER", "NAME", "EMPLOYEE", "EMPLOYER")

' Copy names from Column B of the old file to Column B of the new file

ws.Range("B2:B" & lastRow).Copy Destination:=.Range("B2")

' Calculate EMPLOYEE (0.05 \* Salary in Column C) and place it in Column C

For r = 2 To lastRow

' Check if the value in Column C is numeric

If IsNumeric(ws.Cells(r, "C").Value) Then

.Cells(r, 3).Value = ws.Cells(r, "C").Value \* 0.05 ' Calculate Tier 2 contribution

Else

.Cells(r, 3).Value = 0 ' Set to 0 if the value is not numeric

End If

Next r

' Fill EMPLOYER with 0

.Range("D2:D" & lastRow).Value = 0

' Use XLOOKUP to extract ID\_NUMBER from the "Active" file

Dim lookupRange As Range

Set lookupRange = activeWs.Range("G:G") ' SSNIT numbers in Column G of the "Active" file

Dim idRange As Range

Set idRange = activeWs.Range("B:B") ' ID\_NUMBER in Column B of the "Active" file

For r = 2 To lastRow

' Check if the SSNIT number in Column A is empty

If ws.Cells(r, "A").Value = "" Then

' Option 1: Skip empty SSNIT numbers (leave ID\_NUMBER blank)

.Cells(r, 1).Value = "" ' Leave blank

' Option 2: Assign a default value for empty SSNIT numbers

'.Cells(r, 1).Value = "No SSNIT" ' Uncomment this line to use a default value

Else

' Perform XLOOKUP if SSNIT number is not empty

On Error Resume Next

.Cells(r, 1).Value = Application.WorksheetFunction.XLookup(ws.Cells(r, "A").Value, lookupRange, idRange, "Not Found")

On Error GoTo 0

End If

Next r

' Calculate the sum of Column C (EMPLOYEE)

sumEmployee = Application.WorksheetFunction.Sum(.Range("C2:C" & lastRow))

End With

' Define the output file name with the sum of Column C (e.g., "jun23\_500.34.xlsx")

outputFile = monthNames(i) & "23\_" & Format(sumEmployee, "0.00") & ".xlsx"

newFilePath = newFolderPath & outputFile

' Save the new workbook

newWb.SaveAs Filename:=newFilePath

newWb.Close SaveChanges:=True

' Close the input workbook

ActiveWorkbook.Close SaveChanges:=False

NextMonth:

Next i

' Close the "Active" file

activeWb.Close SaveChanges:=False

' Re-enable screen updating and automatic calculations

Application.ScreenUpdating = True

Application.Calculation = xlCalculationAutomatic

MsgBox "Data restructuring for 2023 completed successfully!", vbInformation

End Sub