

Write VBA macro

1. to check if the folder exists or not
2. to Get the data from file by importing file and copy data from selected range from first sheet
3. to Loop through each worksheet in the worksheets collection, insert new row and add list headers. Also, Format the list header

Solution:

Representation:

Working with Files, Folders			
Link	Topic Covered	What you Learn	Macro Name
DIR	Check if File or Folder Exists - DIR Function	<ul style="list-style-type: none">• Use the DIR function to check if a file or path exists• Create a new Folder (directory)	File_Exists() Path_exists()
SelectFile	User Selects a File - GetOpenFileName Method	Allow the user to pick a file from Excel's file open dialog box. The macro shows how to copy a specific range and import to this workbook.	Get_Data_From_File()
SelectFolder	Loop Inside a Folder - OpenFileDialog Property	Allow the user to pick a folder from Excel's file open dialog box. The macro shows how to loop through each file inside the folder.	Loop_Inside_Folder()

Check if File or Folder Exists - DIR Function

1. How to check if a file exists.
2. How to check if a folder exists.

"File_Exists" procedure shows how to check if a file name exists in the directory. "Path_exists" procedure shows how to check if a folder exists. If it does not exist, the macro will create it.

VBA Code:

Note: Dir path shall be your own directory structure path with filename

Option Explicit
<pre>Sub File_Exists() Dim FileName As String 'Update the path to a valid path on your PC FileName = VBA.FileSystem.Dir("D:\AllFiles\23-24\oddsem\3rdsem\DAE\VBA_Lab_Program\Test_data.xlsx") If FileName = VBA.Constants.vbNullString Then MsgBox "file does not exist." Else 'Update the path to a valid path on your PC Workbooks.Open "D:\AllFiles\23-24\oddsem\3rdsem\DAE\VBA_Lab_Program\" & FileName End If End Sub</pre>

Output: specified file name Test_data.xlsx pops up.

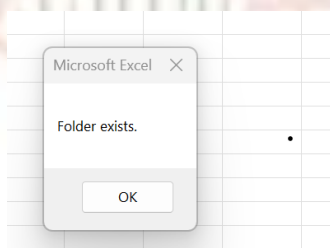
	A	B	C	D	E	F	G	H
1	Company ID	Sales Document	Document Date	Customer code	Article code	Quantity	Sales Value	
2	AT1030	84030	07-10-2018	8050	120	130	6960	
3	AT1030	84030	07-10-2018	8050	104	530	5160	
4	AT1030	84030	07-10-2018	8050	105	130	1200	
5	AT1030	84030	07-10-2018	8050	107	740	3450	
6	AT1030	84031	08-10-2018	8040	104	920	10440	
7	AT1030	84031	08-10-2018	8040	107	810	3800	

Now, Check for path:

```
Sub Path_Exists()
    Dim Path As String
    Dim Folder As String
    Dim Answer As VbMsgBoxResult
    'Update the path to a valid path on your PC
    Path = "D:\AllFiles\23-24\oddsem\3rdsem\DAE\VBA_Lab_Program\VBA2"
    Folder = Dir(Path, vbDirectory)

    If Folder = vbNullString Then
        Answer = MsgBox("Path does not exist. Would you like to create it?", vbYesNo, "Create Path?")
        Select Case Answer
            Case vbYes
                VBA.FileSystem.MkDir (Path)
            Case Else
                Exit Sub
        End Select
    Else
        MsgBox "Folder exists."
    End If
End Sub
```

Output:



b.

Representation:

User Selects a File - GetOpenFileName Method			
Note: In the sales reporting tool, you will learn how to use MultiSelect to import data from many files at once.			
1. Allow the user to select an Excel file			
2. Copy range A1:E20 of sheet 1 from the file selected and paste it starting from A10 below.			
<div style="border: 1px solid black; padding: 5px; display: inline-block;">Import Data</div>			
Hello World	Division	Category	Jan
	West	Telephone	700
	South	Telephone	700
	North	Telephone	980
	East	Telephone	900
	West	Telemarke	3800
	South	Telemarke	6980
	North	Telemarke	3200
	East	Telemarke	3825
	West	Technical	300
	South	Technical	730
	North	Technical	800
	East	Technical	800
	West	Supplies	2000
	South	Supplies	745
	North	Supplies	2410
	East	Supplies	3300
	West	Software	1150
	South	Software	1150
	North	Software	1150

"Get_Data_From_File" Method, brings up the Open File dialog box which prompts the user to select a file. This macro copies cells A1:E20 of the first tab and imports them here. (Click me)

BONUS sub-procedure
"Bonus_Get_Data_From_File_InputBox" allows the user to input the name of the tab they would like to import (it looks for a partial match so the user doesn't have to input the entire tab name) - (click me)

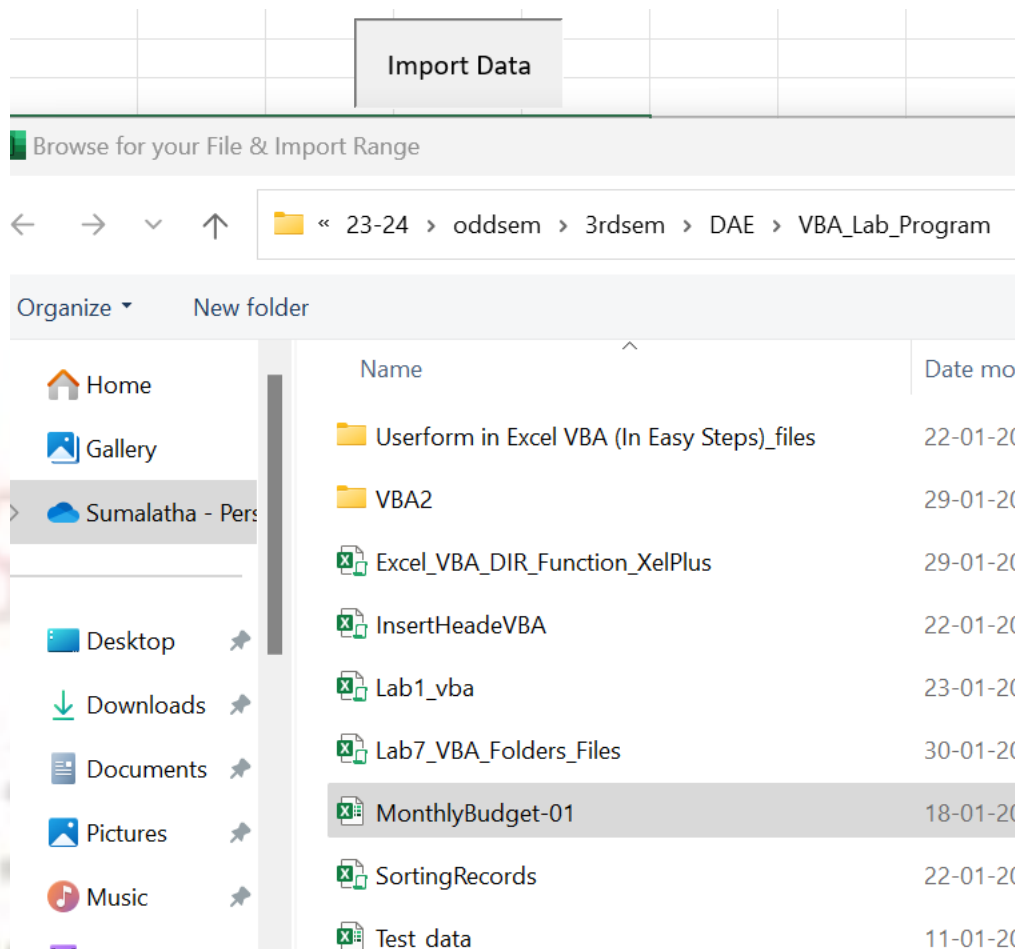
VBA Code:

```
Sub Get_Data_From_File()

    'Note: In the Regional Project that's coming up we learn how to import data from multiple Excel workbooks
    ' Also see BONUS sub procedure below (Bonus_Get_Data_From_File_InputBox()) that expands on this by including an input box
    Dim FileToOpen As Variant
    Dim OpenBook As Workbook
    Application.ScreenUpdating = False
    FileToOpen = Application.GetOpenFilename(Title:="Browse for your File & Import Range", FileFilter:="Excel Files (*.xls*),*xls*")
    If FileToOpen <> False Then
        Set OpenBook = Application.Workbooks.Open(FileToOpen)
        'copy data from A1 to E20 from first sheet
        OpenBook.Sheets(1).Range("A1:E20").Copy
        ThisWorkbook.Worksheets("SelectFile").Range("A10").PasteSpecial xlPasteValues
        OpenBook.Close False
    End If
    Application.ScreenUpdating = True
End Sub
```

Output:

Click "Import Data" and select any file



You can see that excel data is pasted to A1 to E20 range

c.

Consider any work sheet with data.

VBA Code:

```

Sub AddFormatHeader()
'
' AddFormatHeader Macro
' To insert header content
'
' Keyboard Shortcut: Ctrl+Shift+L
'

Windows("InsertHeaderVBA.xlsm").Activate
ActiveCell.Offset(9, 8).Range("A1").Select
Sheets("SouthRecords").Select
Windows("Book1").Activate
ActiveCell.Offset(-25, -10).Range("A1").Select
ActiveCell.FormulaR1C1 = "Region"
ActiveCell.Offset(0, 1).Range("A1").Select
ActiveCell.FormulaR1C1 = "Ca"
ActiveCell.Select
ActiveCell.FormulaR1C1 = "Category"
ActiveCell.Offset(0, 1).Range("A1").Select
ActiveCell.FormulaR1C1 = "Jan"
ActiveCell.Offset(0, 1).Range("A1").Select
ActiveCell.FormulaR1C1 = "Feb"
ActiveCell.Offset(0, 1).Range("A1").Select
ActiveCell.FormulaR1C1 = "Mar"
ActiveCell.Offset(0, 1).Range("A1").Select
Windows("InsertHeaderVBA.xlsm").Activate
Windows("Book1").Activate
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .Color = 49407
    .TintAndShade = 0
    .PatternTintAndShade = 0
End With
ActiveCell.Offset(1, 2).Range("A1:D22").Select
Selection.Style = "Currency"
End Sub

```

Output:

	A	B	C	D	E	F	G
1							
2							
3	Region	Category	Jan	Feb	Mar	Total	
4	South	Technical Support	₹ 800.00	₹ 650.00	₹ 700.00	₹ 2,150.00	
5	South	Telephone	₹ 900.00	₹ 850.00	₹ 850.00	₹ 2,600.00	
6	South	Copying	₹ 4,850.00	₹ 3,200.00	₹ 1,155.00	₹ 9,205.00	
7	South	Overhead	₹ 1,250.00	₹ 1,250.00	₹ 1,250.00	₹ 3,750.00	
8	South	Software	₹ 2,025.00	₹ 2,200.00	₹ 1,650.00	₹ 5,875.00	
9	South	Maintenance	₹ 1,350.00	₹ 1,500.00	₹ 1,700.00	₹ 4,550.00	
10	South	Supplies	₹ 3,300.00	₹ 3,500.00	₹ 3,700.00	₹ 10,500.00	
11	South	Telemarketing	₹ 3,825.00	₹ 3,725.00	₹ 3,750.00	₹ 11,300.00	
12	South	Contractors	₹ 8,900.00	₹ 10,315.00	₹ 5,250.00	₹ 24,465.00	
13	South	Consultants	₹ 6,250.00	₹ 6,000.00	₹ 6,500.00	₹ 18,750.00	
14	South	Rent	₹ 8,000.00	₹ 8,000.00	₹ 8,000.00	₹ 24,000.00	
15	South	Miscellaneous	₹ 11,500.00	₹ 12,500.00	₹ 12,500.00	₹ 36,500.00	
16	South	Advertising	₹ 12,250.00	₹ 12,250.00	₹ 12,750.00	₹ 37,250.00	
17	South	Clerical Support	₹ 25,000.00	₹ 24,000.00	₹ 26,390.00	₹ 75,390.00	
18	South	Maintenance	₹ 1,350.00	₹ 1,500.00	₹ 1,700.00	₹ 4,550.00	
19	South	Supplies	₹ 3,300.00	₹ 3,500.00	₹ 3,700.00	₹ 10,500.00	
20	South	Telemarketing	₹ 3,825.00	₹ 3,725.00	₹ 3,750.00	₹ 11,300.00	
21	South	Contractors	₹ 8,900.00	₹ 10,315.00	₹ 5,250.00	₹ 24,465.00	
22	South	Consultants	₹ 6,250.00	₹ 6,000.00	₹ 6,500.00	₹ 18,750.00	
23	South	Telephone	₹ 900.00	₹ 850.00	₹ 850.00	₹ 2,600.00	
24	South	Copying	₹ 4,850.00	₹ 3,200.00	₹ 1,155.00	₹ 9,205.00	
25	South	Overhead	₹ 1,250.00	₹ 1,250.00	₹ 1,250.00	₹ 3,750.00	
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							

< >

SouthRecords

Run Macro

+