

# VBA SOURCE CODE BOOK

## MIND MAP TASK MANAGER - UPDATED

**MIND MAP TASK MANAGER**

Save New Delete Refresh Staff: Fred Fredders Freds House Remod [All Statuses] Start Date From: 5/1/2022 Due Date From: 7/30/2022

**Task Details:**

Select Project: Freds House Remodel  
Staff: Fred Fredders  
Task Name: Phone Meeting  
Status: Overdue  
Start Date: 5/28/2022  
Due Date: 6/6/2022  
Description: Site Scope Notes

**STAFF-TASK LIST**

Task Name	Project
Phone Meeting	Freds House Remodel
Project Scope	Johnson Shed Redo
Meeting Mtg	Freds House Remodel
Cust. Phone	Johnson Shed Redo
Site Design	Freds House Remodel
Phone Meeting	Johnson Shed Redo
Site Change	James New House
Design Screen	James New House
Site Scope	James New House
Dinner Meeting	James New House

**OVERDUE TASKS** Set All To Overdue

Task Name	Project	Due Date
Phone Meeting	Freds House Remodel	6/6/2022
App Scope	Fredders Kitchen	6/2/2022
Managers Meet	Lisa House Design	6/9/2022
TaskDes5	Garage Cleanout	6/5/2022
Site Design	Freds House Remodel	6/3/2022
Design Meeting	James New House	6/9/2022
Wire Frame Bull	Lisa House Design	6/10/2022
Prelim. Designs	Garage Cleanout	6/11/2022
Project Scope	Lisa House Design	6/3/2022
App Scope	Lisa House Design	6/3/2022
Client Email	Lisa House Design	6/3/2022
Project Scope	Lisa House Design	6/3/2022
App Scope	Lisa House Design	6/3/2022
Lunch Meeting	Lisa House Design	6/3/2022

**FEATURE, FIX, OR FOCUS**



*Excel For Freelancers*

Now You Can Create Mind  
Maps **AUTOMATICALLY** In Excel



[DOWNLOAD  
APPLICATION](#)



[VIEW  
TRAINING](#)

by: *Randy Austin*

# ABOUT THE AUTHOR

A two-time Microsoft MVP & lifetime Excel enthusiast, Randy Austin founded Excel For Freelancers in 2017. Excel For Freelancers quickly became the most prominent resource Excel for developers to learn how to turn their passion for Excel into profits by building & selling their own excel-based applications for passive & recurring income.

With over 300,000 YouTube subscribers, 15,000,000 video views, 200+ comprehensive training videos, and a thriving 50,000 member Facebook community, Excel For Freelancers has positioned itself as the #1 Excel developers resource in the world.

Get free content, training, and downloads just by clicking any of the free resources below:



[WEBSITE](#)



[YOUTUBE](#)



[FACEBOOK](#)



[TWITTER](#)



[DISCORD](#)



[INSTAGRAM](#)



[TELEGRAM](#)



[RUMBLE](#)



**Microsoft**<sup>®</sup>  
Most Valuable  
Professional



# OUR COURSES & PRODUCTS



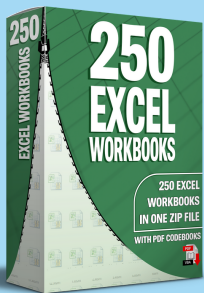
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[Click here to learn more](#)



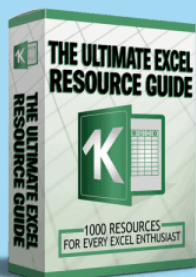
16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[Click here to learn more](#)



Incredible Package of 250 of my BEST Applications now with PDF VBA Codebooks packed into a SINGLE ZIP File which also includes the "250 Workbook Library".

[Click here to learn more](#)



With 1000 live links, continuously updating content, sort-able and filterable items, you will always have exactly what you need, when you need it.

[Click here to learn more](#)

Projects.....	2
VBAProject.....	2
Documents.....	2
Admin.....	2
(Declarations).....	2
Worksheet_SelectionChange [Sub ].....	2
MindMap.....	3
(Declarations).....	3
Worksheet_SelectionChange [Sub ].....	3
Projects.....	4
(Declarations).....	4
Staff.....	5
(Declarations).....	5
Tasks.....	6
(Declarations).....	6
ThisWorkbook.....	7
(Declarations).....	7
Modules.....	8
App_Macros.....	8
(Declarations).....	8
Admin_SetStatusColor [Sub ].....	8
BrowseForStaffPicFolder [Sub ].....	8
Icon_Add [Sub ].....	8
Icon_Clear [Sub ].....	8
MindMap_Macros.....	10
(Declarations).....	10
MindMap_CollapseTasks [Sub ].....	10
MindMap_HideProjects [Sub ].....	10
MindMap_LoadProjects [Sub ].....	10
MindMap_LoadStaff [Sub ].....	12
MindMap_LoadTasks [Sub ].....	14
MindMap_ViewTask [Sub ].....	17
Task_Macros.....	18
(Declarations).....	18
Task_CheckForOverdue [Sub ].....	18
Task_Delete [Sub ].....	18
Task_MarkAllAsOverdue [Sub ].....	18
Task_New [Sub ].....	18
Task_SavedMsg [Sub ].....	18
Task_SaveUpdate [Sub ].....	19
Task_SelectFromOverdueTasks [Sub ].....	20
Task_SelectFromStaffList [Sub ].....	20

```
1 Option Explicit
2
3 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
4     If Target.CountLarge > 1 Then Exit Sub
5     If Shapes("ColorPalette").Visible = True Then Shapes("ColorPalette").Visible =
6         msoFalse
7     If Shapes("AddIconBtn").Visible = True Then Shapes("AddIconBtn").Visible =
8         msoFalse
9     If Shapes("ClearIconBtn").Visible = True Then Shapes("ClearIconBtn").Visible =
10        msoFalse
11
12     'On Icon Type Selection, show icon buttons
13     If Not Intersect(Target, Range("E8:F12")) Is Nothing Then
14         Range("T1").Value = Target.Row 'Triggers cond. formatting
15         With Shapes("AddIconBtn")
16             .Left = Range("G" & Target.Row).Left
17             .Top = Range("G" & Target.Row).Top
18             .Visible = msoCTrue
19         End With
20         With Shapes("ClearIconBtn")
21             .Left = Range("G" & Target.Row).Left + 16
22             .Top = Range("G" & Target.Row).Top
23             .Visible = msoCTrue
24         End With
25     Else
26         Range("T1").ClearContents
27     End If
28
29     'Show color palette
30     If Not Intersect(Target, Range("C8:C12")) Is Nothing Then
31         With Shapes("ColorPalette")
32             .Left = Range("C" & Target.Row).Left
33             .Top = Range("C" & Target.Row + 1).Top
34             .Visible = msoCTrue
35             .ZOrder msoBringToFront
36         End With
37     End If
38 End Sub
39
40
```

```
1 Option Explicit
2
3 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
4     'Staff Task List Selection
5     If Not Intersect(Target, Range("D21:E999" )) Is Nothing And Range("C" & Target.Row)
6         .Value <> Empty Then
7             Range("B12" ).Value = Target.Row 'Selected Row
8             Task_SelectFromStaffList
9         End If
10
11     'Overdue Task List Selection
12     If Not Intersect(Target, Range("T5:V999" )) Is Nothing And Range("S" & Target.Row).
13     Value <> Empty Then
14         Range("B13" ).Value = Target.Row
15         Task_SelectFromOverdueTasks
16     End If
17 End Sub
```

1 Option Explicit

2

1 Option Explicit

2



1 Option Explicit

2

1 Option Explicit

2

```

1 Option Explicit
2 Dim ShtName As String
3 Dim LastRow As Long, LastResultRow As Long, SelRow As Long
4 Dim IconFile As FileDialog
5
6
7 Sub Admin_SetStatusColor()
8     ActiveCell.Interior.Color = Admin.Shapes(Application.Caller).Fill.ForeColor.RGB
9     Admin.Shapes("ColorPalette").Visible = msoFalse
10 End Sub
11
12
13 Sub BrowseForStaffPicFolder()
14     Dim StaffFolder As FileDialog
15     Set StaffFolder = Application.FileDialog(msoFileDialogFolderPicker)
16     With StaffFolder
17         .Title = "Browse For Staff Folder"
18         .AllowMultiSelect = False
19         If .Show <> -1 Then GoTo NoSelection
20         Admin.Range("D3").Value = .SelectedItems(1)
21     End With
22     NoSelection:
23 End Sub
24
25
26
27
28 Sub Icon_Add()
29     With Admin
30         SelRow = .Range("T1").Value 'Selected Row
31         'Clear Any existing icon
32         On Error Resume Next
33         .Shapes("Icon" & SelRow - 7).Delete
34         On Error GoTo 0
35         Set IconFile = Application.FileDialog(msoFileDialogFilePicker)
36         With IconFile
37             .Title = "Please select an icon"
38             .Filters.Add "Picture Files", "*.jpg,*.png,*.gif, *.jpeg *.GIF", 1
39             .AllowMultiSelect = False
40             If .Show <> -1 Then GoTo NoSelection
41             Admin.Range("F" & SelRow).Value = Dir(.SelectedItems(1)) 'Set File Name
42             Admin.Pictures.Insert(.SelectedItems(1)).Name = "Icon" & SelRow - 7 'Create Icon
43         End With
44         With .Shapes("Icon" & SelRow - 7)
45             .Left = Admin.Range("E" & SelRow).Left + 1
46             .Top = Admin.Range("E" & SelRow).Top + 1
47             .Width = 12
48             .Height = 12
49             .Visible = msoCTrue
50         End With
51         NoSelection:
52     End With
53 End Sub
54
55 Sub Icon_Clear()
56     With Admin
57         SelRow = .Range("T1").Value 'Selected Row
58         'Clear Any existing icon
59         On Error Resume Next

```

```
60 | 1 2 | .Shapes("Icon" & SelRow - 7).Delete
61 | | On Error GoTo 0
62 | | End With
63 | | End Sub
```

```

1 Option Explicit
2 Dim MMShp As Shape
3 Dim LastRow As Long, LastResultRow As Long, ResultRow As Long, IconNumb As Long,
TaskListRow As Long
4 Dim StaffID As String, StaffNm As String, StaffPic As String, PicPath As String,
PicFolder As String
5 Dim MMTop As Double, MMLeft As Double
6
7 Sub MindMap_CollapseTasks()
8     Dim ProjID As String
9     With MindMap
10        ProjID = Replace(Application.Caller, "MMProjShrinkProj", "") 'Set Project ID
11        On Error Resume Next
12        .Shapes("TaskGrp-Proj" & ProjID).Delete
13        For Each MMShp In .Shapes
14            If InStr(MMShp.Name, "MMProj" & ProjID & "TaskConn") > 0 Then MMShp.Delete
'Delete Connectors
15            .Shapes("MMProjExpProj" & ProjID).Visible = msoCTrue
16            .Shapes(Application.Caller).Delete 'Remove Shrink Shape
17        Next MMShp
18    End With
19 End Sub
20
21 Sub MindMap_HideProjects()
22     With MindMap
23        .Range("J3").Value = "[All Projects]" 'Reset to All projects
24        For Each MMShp In .Shapes
25            On Error Resume Next
26            If InStr(MMShp.Name, "MMProj") > 0 Then MMShp.Delete 'Remove Project
27            If InStr(MMShp.Name, "TaskGrp") > 0 Then MMShp.Delete 'Remove Project
Groups of tasks
28            If InStr(MMShp.Name, "MMStaffShrinkStaff") > 0 Then MMShp.Delete 'Remove all
Shrink Staff/Project buttons
29            If InStr(MMShp.Name, "MMStaffExpStaff") > 0 Then MMShp.Visible = msoCTrue
30            On Error GoTo 0
31        Next MMShp
32    End With
33 End Sub
34
35
36 Sub MindMap_LoadProjects()
37     Dim ProjID As String, ProjNm As String
38     Dim ProjCount As Long
39     With MindMap
40        .Range("B12,C21:E9999").ClearContents 'Clear Task Fields
41        StaffID = Replace(Application.Caller, "MMStaffExpStaff", "") 'Get Staff ID
42        StaffNm = .Shapes("MMStaff" & StaffID).TextFrame2.TextRange.Text 'Get Staff
Name
43        .Range("H3").Value = StaffNm 'Set Staff Name Filter
44        .Range("J3").Value = "[All Projects]" 'Reset to All projects
'Remove all Mind Map Projects
45        For Each MMShp In .Shapes
46            On Error Resume Next
47            If InStr(MMShp.Name, "MMProj") > 0 Then MMShp.Delete 'Remove Project
48            If InStr(MMShp.Name, "MMStaffShrinkStaff") > 0 Then MMShp.Delete 'Remove all
Shrink Staff/Project buttons
49            If InStr(MMShp.Name, "TaskGrp") > 0 Then MMShp.Delete 'Remove Project
Groups of tasks
50            If InStr(MMShp.Name, "MMStaffExpStaff") > 0 Then MMShp.Visible = msoCTrue
51            On Error GoTo 0
52

```

1 2 3

```

53 1 2 3 Next MMShp
54 'Run Project Filter
55 With Tasks
56 LastRow = .Range("A99999").End(xlUp).Row
57 If LastRow < 4 Then Exit Sub
58 .Range("A3:J" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
"O2:S4" ), CopyToRange:=.Range("AB2:AC2" ), Unique:=True
59 LastResultRow = .Range("AB99999").End(xlUp).Row
60 If LastResultRow < 3 Then
61 MsgBox "No projects exists for selected staff"
62 Exit Sub
63 End If
64 ProjCount = LastResultRow - 2 '# Of Projects
65 'Get Top Position
66 MMTop = MindMap.Shapes(Application.Caller).Top - ((50 * (ProjCount - 1)) / 2)
- 5 'Set top Position Split above & Below expander
67 MMLeft = MindMap.Shapes(Application.Caller).Left + 50 'Set left Position
68 If MMTop < MindMap.Range("J5").Top Then MMTop = MindMap.Range("J5").Top
'Set Minimum top Position
69
70 'Replace Staff Expander With staff Shrinker
71 MindMap.Shapes("StaffShrinkSample").Duplicate.Name = "MMStaffShrinkStaff" &
StaffID
72 MindMap.Shapes("MMStaffExpStaff" & StaffID).Visible = msoFalse 'Hide
Expander Shape
73 With MindMap.Shapes("MMStaffShrinkStaff" & StaffID) 'Place Shrink button in
place
74 .Left = MindMap.Shapes("MMStaffExpStaff" & StaffID).Left
75 .Top = MindMap.Shapes("MMStaffExpStaff" & StaffID).Top
76 .OnAction = "MindMap_HideProjects"
77 End With
78
79 For ResultRow = 3 To LastResultRow 'Loop Through Project Results
80 ProjID = .Range("AB" & ResultRow).Value 'Project ID
81 ProjNm = .Range("AC" & ResultRow).Value 'Project Name
82 MindMap.Shapes("MMSample").Duplicate.Name = "MMProj" & ProjID
83 With MindMap.Shapes("MMProj" & ProjID)
84 .Top = MMTop
85 .Left = MMLeft
86 .TextFrame2.TextRange.Text = ProjNm 'Set Project Name
87 .TextFrame.AutoSize = msoAutoSizeShapeToFitText
88 .Height = 18 'Set Height
89 End With
90
91 'Add Project Icon
92 On Error Resume Next
93 MindMap.Shapes("Icon2").Duplicate.Name = "MMProjIcon" & ProjID
94 With MindMap.Shapes("MMProjIcon" & ProjID)
95 .Left = MMLeft + 1 'Set Left Position
96 .Top = MindMap.Shapes("MMProj" & ProjID).Top + 1
97 .Height = 16
98 .Width = 16
99 End With
100 On Error GoTo 0
101
102 'Add in Staff Connector
103 MindMap.Shapes("ProjConnSample").Duplicate.Name = "MMProjConn" & ProjID
'Duplicate Staff Connector
104 With MindMap.Shapes("MMProjConn" & ProjID)
105 .ConnectorFormat.BeginConnect MindMap.Shapes("MMStaffShrinkStaff" &

```

```

1   2   3   4   5
106   StaffID), 7
107   .ConnectorFormat.EndConnect MindMap.Shapes("MMProj" & ProjID), 2
108   End With
109   'Add in project expanders
110   MindMap.Shapes("ProjExpSample" ).Duplicate.Name = "MMProjExpProj" & ProjID
111   'Duplicate Project Expander
112   With MindMap.Shapes("MMProjExpProj" & ProjID)
113     .Left = MindMap.Shapes("MMProj" & ProjID).Left + MindMap.Shapes(
114       "MMProj" & ProjID).Width + 8
115     .Top = MindMap.Shapes("MMProj" & ProjID).Top + 2.5
116     .OnAction = "MindMap_LoadTasks"
117   End With
118   'Add in Project Connector
119   MindMap.Shapes("ProjConnSample" ).Duplicate.Name = "MMProjStaffConn" &
120   ProjID 'Duplicate Project Connector
121   With MindMap.Shapes("MMProjStaffConn" & ProjID)
122     .ConnectorFormat.BeginConnect MindMap.Shapes("MMProj" & ProjID), 4
123     .ConnectorFormat.EndConnect MindMap.Shapes("MMProjExpProj" & ProjID), 3
124   End With
125   MMTop = MMTop + 50 'Increment Top Postion
126   Next ResultRow
127   'Get Unique Tasks Per Staff
128   .Range("A3:J" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
129     "O2:S4" ), CopyToRange:=.Range("AP2:AR2" ), Unique:=True
130   LastResultRow = .Range("AP99999" ).End(xlUp).Row
131   If LastResultRow < 3 Then Exit Sub
132   MindMap.Range("C21:E" & LastResultRow + 18).Value = .Range("AP3:AR" &
133     LastResultRow).Value
134   End With
135   End With
136   End Sub
137
138 Sub MindMap_LoadStaff()
139   With MindMap
140     .Range("B12,C21:E9999" ).ClearContents
141     .Range("H3" ).Value = "[All Staff]"
142     .Range("J3" ).Value = "[All Projects]"
143     .Range("L3" ).Value = "[All Statuses]"
144   'Remove all Mind Map Shapes
145   For Each MMShp In .Shapes
146     On Error Resume Next
147     If InStr(MMShp.Name, "MMStaff" ) > 0 Then MMShp.Delete 'Delete Staff Shapes
148     If InStr(MMShp.Name, "MMProj" ) > 0 Then MMShp.Delete 'Delete Project Shapes
149     If InStr(MMShp.Name, "TaskGrp" ) > 0 Then MMShp.Delete 'Delete any Task
150     Groups
151     If InStr(MMShp.Name, "Icon" ) > 0 Then MMShp.Delete 'Delete Icon shapes
152     On Error GoTo 0
153   Next MMShp
154   'Get Icons from Admin Page
155   For IconNumb = 1 To 5
156     If Admin.Range("E" & IconNumb + 7).Value <> "" Then
157       On Error Resume Next
158       Admin.Shapes("Icon" & IconNumb).Copy
159       .Activate

```

```

158     .Range("A" & 21 + IconNumb).Select
159     .Paste
160     On Error GoTo 0
161     End If
162     Next IconNumb
163
164     'Set Intial Top & Left Positions
165     MMLeft = .Range("F5" ).Left + 30 'Set Initial left position
166     MMTop = .Range("F5" ).Top 'Set Initial top positions
167
168     PicFolder = Admin.Range("PictureFolder" ) & "/" 'Set Picture folder
169     'Run Staff Filter
170     With Tasks
171         .Range("U3:W999,AP3:AR9999" ).ClearContents 'Clear Previous Results
172         LastRow = .Range("A99999" ).End(xlUp).Row
173         If LastRow < 4 Then Exit Sub
174         .Range("A3:J" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
175             "O2:S4" ), CopyToRange:=.Range("U2:V2" ), Unique:=True
176         LastResultRow = .Range("U99999" ).End(xlUp).Row
177         If LastResultRow < 3 Then Exit Sub
178         .Range("W3:W" & LastResultRow).Formula = .Range("W1" ).Formula 'Staff
179         Picture Formula
180         For ResultRow = 3 To LastResultRow
181             'Create Staff Shapes
182             StaffID = .Range("U" & ResultRow).Value 'Staff ID
183             StaffNm = .Range("V" & ResultRow).Value 'Staff Name
184             StaffPic = .Range("W" & ResultRow).Value 'Staff Name
185             MindMap.Shapes("MMSample" ).Duplicate.Name = "MMStaff" & StaffID
186             With MindMap.Shapes("MMStaff" & StaffID)
187                 .Left = MMLeft 'Set Left Position
188                 .Top = MMTop
189                 .TextFrame2.TextRange.Text = StaffNm 'Set Staff Name as text
190                 .TextFrame.AutoSize = msoAutoSizeShapeToFitText 'Size shape to fit the
191                 text
192                 .Height = 18 'Set Height
193             End With
194
195             'Add Staff Picture (if any)
196             If StaffPic <> "" Then 'Staff Picture Exists
197                 PicPath = PicFolder & StaffPic 'Set Picture Path
198                 If Dir(PicPath, vbDirectory) = "" Then GoTo BadPath
199                 MindMap.Shapes("StaffSample" ).Duplicate.Name = "MMStaffPic" & StaffID
200                 'Duplicate Sample Staff Pic
201                 With MindMap.Shapes("MMStaffPic" & StaffID)
202                     .Left = MMLeft 'Set Left Position
203                     .Top = MindMap.Shapes("MMStaff" & StaffID).Top
204                     .Fill.UserPicture (PicPath) 'Fills Shape with
205                     .Height = 18
206                     .Width = 18
207                 End With
208             Else 'No Staff Picture, use Icon
209                 BadPath:
210                 On Error Resume Next
211                 MindMap.Shapes("Icon1" ).Duplicate.Name = "MMStaffIcon" & StaffID
212                 With MindMap.Shapes("MMStaffIcon" & StaffID)
213                     .Left = MMLeft + 1 'Set Left Position
214                     .Top = MindMap.Shapes("MMStaff" & StaffID).Top + 1
215                     .Height = 16
216                     .Width = 16
217                 End With

```



```

1 2 3 4 5
214 On Error GoTo 0
215 End If
216 'Add in Staff Expander Buttons
217 MindMap.Shapes("StaffExpSample").Duplicate.Name = "MMStaffExpStaff" &
StaffID 'Duplicate Staff Expander
218 With MindMap.Shapes("MMStaffExpStaff" & StaffID)
219 .Left = MindMap.Shapes("MMStaff" & StaffID).Left + MindMap.Shapes(
"MMStaff" & StaffID).Width + 8
220 .Top = MindMap.Shapes("MMStaff" & StaffID).Top + 2.5
221 .OnAction = "MindMap_LoadProjects"
222 End With
223
224 'Add in Connector
225 MindMap.Shapes("StaffConnSample").Duplicate.Name = "MMStaffConn" &
StaffID 'Duplicate Staff Connector
226 With MindMap.Shapes("MMStaffConn" & StaffID)
227 .ConnectorFormat.BeginConnect MindMap.Shapes("MMStaff" & StaffID), 4
228 .ConnectorFormat.EndConnect MindMap.Shapes("MMStaffExpStaff" & StaffID)
, 3
229 End With
230 MMTop = MMTop + 50 'Increment Top Postion
231 Next ResultRow
232
233 'Get All Tasks List
234 .Range("A3:J" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
"O2:S4" ), CopyToRange:=.Range("AP2:AR2" ), Unique:=True
235 LastResultRow = .Range("AP99999" ).End(xlUp).Row
236 If LastResultRow < 3 Then Exit Sub
237 MindMap.Range("C21:E" & LastResultRow + 18).Value = .Range("AP3:AR" &
LastResultRow).Value
238 End With
239 End With
240 Task_CheckForOverdue 'Run macro to check for overdue
241 End Sub
242
243 Sub MindMap_LoadTasks()
244 Dim TaskID As String, TaskNm As String, TaskStatus As String, TaskColor As String,
ShrinkShape As String
245 Dim ProjID As String, ProjNm As String, TaskGrpString As String, TaskGrpArr() As
String
246 Dim TaskCount As Long, TaskRow As Long, LastProjRow As Long
247 Dim GroupCount As Long, ProjRow As Long, TotalTaskHeight As Long
248 Dim TaskGrp As Shape
249
250 With MindMap
251 'Set Project ID based on running macro
252 On Error Resume Next
253 If Application.Caller = "SaveBtn" Then 'Save Button Clicked
254 ProjID = .Range("B2" ).Value
255 Else 'Project expander button clicked
256 ProjID = Replace(Application.Caller, "MMPProjExpProj" , "" ) 'Get Project ID
257 End If
258 On Error GoTo 0
259 If ProjID = 0 Then ProjID = .Range("B2" ).Value
260 On Error Resume Next
261 .Shapes("TaskGrp-Proj" & ProjID).Delete 'Delete Project Group if it exists
262 On Error GoTo 0
263
264 ShrinkShape = "MMPProjShrinkProj" & ProjID 'Set Shrink Shape Name
265 On Error GoTo NoProj

```

```

1 2
266 ProjNm = .Shapes("MMProj" & ProjID).TextFrame2.TextRange.Text 'Get Project Name
267 .Range("J3" ).Value = ProjNm 'Set Project Name Filter
268 End With
269 'Run Taskect Filter
270 With Tasks
271 LastRow = .Range("A99999" ).End(xlUp).Row
272 If LastRow < 4 Then Exit Sub
273 .Range("A3:J" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
"O2:S4" ), CopyToRange:=.Range("AL2:AN2" ), Unique:=True
274 LastResultRow = .Range("AL99999" ).End(xlUp).Row
275 If LastResultRow < 3 Then
276 MsgBox "No Tasks exists for selected Project"
277 Exit Sub
278 End If
279 Application.ScreenUpdating = False
280 With .Sort 'Sort based on Status (custom List)
281 .SortFields.Clear
282 .SortFields.Add Key:=Tasks.Range("AN3" ), SortOn:=xlSortOnValues, Order:=
xlAscending, CustomOrder:="To Do, In Progress, Overdue, Completed" ,
DataOption:=xlSortNormal 'Sort By Custom Status Order
283 .SetRange Tasks.Range("AL3:AN" & LastResultRow)
284 .Apply
285 End With
286
287 TaskCount = LastResultRow - 2 '# Of Tasks
288 'Get Top Position
289 MMTop = MindMap.Shapes("MMProjExpProj" & ProjID).Top - ((50 * (TaskCount - 1)) /
2) 'Set top Position Split above & Below expander
290 MMLeft = MindMap.Shapes("MMProjExpProj" & ProjID).Left + 80 'Set left Position
291 If MMTop < MindMap.Range("J5" ).Top Then MMTop = MindMap.Range("J5" ).Top 'Set
Minimum top Position
292 'Replace Project Expander With Project Shrinker
293 MindMap.Shapes("ProjShrinkSample" ).Duplicate.Name = "MMProjShrinkProj" & ProjID
294 MindMap.Shapes("MMProjExpProj" & ProjID).Visible = msoFalse 'Hide Expander
Shape
295 With MindMap.Shapes("MMProjShrinkProj" & ProjID)
296 .Left = MindMap.Shapes("MMProjExpProj" & ProjID).Left
297 .Top = MindMap.Shapes("MMProjExpProj" & ProjID).Top
298 .OnAction = "MindMap_CollapseTasks"
299 End With
300 For ResultRow = 3 To LastResultRow 'Loop Through Task Results
301 TaskID = .Range("AL" & ResultRow).Value 'Task ID
302 TaskNm = .Range("AM" & ResultRow).Value 'Task Name
303 TaskStatus = .Range("AN" & ResultRow).Value 'Task Status
304 On Error Resume Next
305 TaskRow = Admin.Range("Status" ).Find(TaskStatus, , xlValues, xlWhole).Row
306 On Error GoTo 0
307 If TaskRow <> 0 Then TaskColor = Admin.Range("C" & TaskRow).Interior.Color
'Set Task Color
308
309 'Set Main Task Shape
310 MindMap.Shapes("MMSample" ).Duplicate.Name = "MMTask" & TaskID
311 With MindMap.Shapes("MMTask" & TaskID)
312 .Left = MMLeft 'Set Left Position
313 .Top = MMTop
314 .TextFrame2.TextRange.Text = TaskNm
315 .TextFrame.AutoSize = msoAutoSizeShapeToFitText

```

```

1 2 3 4
316 .Height = 18 'Set Height
317 .Width = .Width + 18 'Add Space for View Task Icon
318 TaskGrpString = TaskGrpString & "MMTask" & TaskID & ","
319 End With
320
321 'Add Task Icon
322 On Error Resume Next
323 MindMap.Shapes("Icon3").Duplicate.Name = "MMTaskIcon" & TaskID
324 With MindMap.Shapes("MMTaskIcon" & TaskID)
325 .Left = MMLeft + 3 'Set Left Position
326 .Top = MindMap.Shapes("MMTask" & TaskID).Top + 4
327 .Height = 16
328 .Width = 16
329 .Line.Visible = msoCTrue
330 .Line.ForeColor.RGB = TaskColor 'Set border Color based on Status
331 .Fill.ForeColor.RGB = TaskColor
332 TaskGrpString = TaskGrpString & "MMTaskIcon" & TaskID & ","
333 End With
334 On Error GoTo 0
335
336 'Add View Task Icon
337 On Error Resume Next
338 MindMap.Shapes("Icon4").Duplicate.Name = "MMTaskViewIcon" & TaskID
339 With MindMap.Shapes("MMTaskViewIcon" & TaskID)
340 .Left = MMLeft + MindMap.Shapes("MMTask" & TaskID).Width - 17 'Set Left
    Position
341 .Top = MindMap.Shapes("MMTask" & TaskID).Top + 3
342 .Height = 13
343 .Width = 13
344 .OnAction = "MindMap_ViewTask"
345 TaskGrpString = TaskGrpString & "MMTaskViewIcon" & TaskID & ","
346 End With
347 On Error GoTo 0
348
349 'Add in Task Connector
350 MindMap.Shapes("TaskConnSample").Duplicate.Name = "MMProj" & ProjID &
    "TaskConn" & TaskID 'Duplicate Task Connector
351 With MindMap.Shapes("MMProj" & ProjID & "TaskConn" & TaskID)
352 .ConnectorFormat.BeginConnect MindMap.Shapes("MMProjShrinkProj" & ProjID),
    7
353 .ConnectorFormat.EndConnect MindMap.Shapes("MMTask" & TaskID), 2
354 End With
355 MMTop = MMTop + 25 'Increment Top Postion
356 Next ResultRow
357
358 'Group Task Shapes
359 TaskGrpString = Left(TaskGrpString, Len(TaskGrpString) - 1) 'Remove Last Comma
360 TaskGrpArr = Split(TaskGrpString, ",") 'Create Task Group Array
361 MindMap.Shapes.Range(TaskGrpArr).Group.Name = "TaskGrp-Proj" & ProjID
362
363 'Space out Task Groups accordingly
364 LastProjRow = .Range("AB99999").End(xlUp).Row
365 If LastProjRow < 4 Then GoTo EndMacro 'Exit on 1 or less projects visible
366 MMTop = MindMap.Range("N5").Top 'Set intial top Position
367 For ProjRow = 3 To LastProjRow
368 ProjID = .Range("AB" & ProjRow).Value 'Set Project ID
369 On Error Resume Next
370 Set TaskGrp = MindMap.Shapes("TaskGrp-Proj" & ProjID)
371 On Error GoTo 0
372 If Not TaskGrp Is Nothing Then 'Group Exists

```

```

1 2 3 4
373 TotalTaskHeight = TotalTaskHeight + TaskGrp.Height 'Combine Heights of all
task groups
374 GroupCount = GroupCount + 1 'Count Groups
375 Set TaskGrp = Nothing 'Reset Task Group
376 End If
377 Next ProjRow
378 Set TaskGrp = Nothing 'Reset Task group
379 If GroupCount = 1 Then GoTo EndMacro 'No need to reset height on only 1 group
visible
380 For ProjRow = 3 To LastProjRow
381 ProjID = .Range("AB" & ProjRow).Value 'Set Project ID
382 On Error Resume Next
383 Set TaskGrp = MindMap.Shapes("TaskGrp-Proj" & ProjID)
384 On Error GoTo 0
385 If Not TaskGrp Is Nothing Then 'Group Exists
386 TaskGrp.Top = MMTop
387 MMTop = TaskGrp.Top + TaskGrp.Height + 30
388 Set TaskGrp = Nothing 'Reset Task Group
389 End If
390 Next ProjRow
391 EndMacro:
392 NoProj:
393 Application.ScreenUpdating = True
394 End With
395 End Sub
396 Sub MindMap_ViewTask()
397 Dim TaskID As String
398 Dim TaskRow As Long, TaskCol As Long
399 With MindMap
400 On Error Resume Next
401 TaskID = Replace(Application.Caller, "MMTaskViewIcon", "") 'Get Task ID
402 On Error GoTo 0
403 If TaskID = "" Then TaskID = .Range("B5").Value 'Set task ID on List
selection
404 .Range("B5").Value = TaskID
405 If .Range("B6").Value = "" Then
406 MsgBox "Please select a correct task to view"
407 Exit Sub
408 End If
409 TaskRow = .Range("B6").Value 'Task Row
410 For TaskCol = 2 To 10 'Load Task Details (But not Staff ID or Proj. ID which are
generated by VBA)
411 If TaskCol <> 3 And TaskCol <> 8 Then .Range(Tasks.Cells(1, TaskCol).Value).
Value = Tasks.Cells(TaskRow, TaskCol).Value
412 Next TaskCol
413 End With
414 End Sub
415
416
417

```

```

1 Option Explicit
2 Dim TaskRow As Long, TaskCol As Long, LastRow As Long, LastResultRow As Long
3 Dim ProjID As String, ProjName As String, TaskID As String
4 Dim ProjRow As Long, StaffRow As Long, SelRow As Long
5 Sub Task_CheckForOverdue()
6     MindMap.Range("S5:V999").ClearContents 'Clear Previous tasks
7     With Tasks
8         LastRow = .Range("A99999").End(xlUp).Row
9         If LastRow < 4 Then Exit Sub
10        .Range("A3:J" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
11            "AU2:AV3" ), CopyToRange:=.Range("AX2:BA2" ), Unique:=True
12        LastResultRow = .Range("AX99999").End(xlUp).Row
13        If LastRow < 3 Then Exit Sub
14        MindMap.Range("S5:V" & LastResultRow + 2).Value = .Range("AX3:BA" &
15            LastResultRow).Value 'Bring Over Overdue Tasks
16    End With
17 End Sub
18 Sub Task_Delete()
19     If MsgBox("Are you sure you want to delete this Task?" , vbYesNo, "Delete Task" ) =
20         vbNo Then Exit Sub
21     If MindMap.Range("B6").Value = Empty Then GoTo NotSaved
22     TaskRow = MindMap.Range("B6").Value 'Set Task Row
23     Tasks.Range(TaskRow & ":" & TaskRow).EntireRow.Delete
24     NotSaved:
25     MindMap_LoadStaff 'Refresh MindMap
26     Task_New 'Clear Fields
27 End Sub
28 Sub Task_MarkAllAsOverdue()
29     Dim MapRow As Long, TaskCount As Long
30     With MindMap
31         LastRow = .Range("T99999").End(xlUp).Row
32         If LastRow < 5 Then
33             MsgBox "There are no overdue tasks"
34             Exit Sub
35         End If
36         For MapRow = 5 To LastRow
37             TaskID = .Range("S" & MapRow).Value 'Task ID
38             On Error Resume Next
39             TaskRow = Tasks.Range("Task_ID").Find(TaskID, , xlValues, xlWhole).Row
40             On Error GoTo 0
41             If TaskRow <> 0 Then
42                 If Tasks.Range("E" & TaskRow).Value <> Admin.Range("B10").Value Then
43                     'Overdue Status
44                     Tasks.Range("E" & TaskRow).Value = Admin.Range("B10").Value 'Set To
45                     Overdue
46                     TaskCount = TaskCount + 1
47                 End If
48             End If
49         Next MapRow
50         MsgBox TaskCount & " tasks have been set to overdue"
51     End With
52 End Sub
53 Sub Task_New()
54     MindMap.Range("B5,E6:E16").ClearContents
55     MindMap.Range("E9").Value = "To Do" 'set intial Status
56     MindMap.Range("E6").Select 'Select intial cell
57 End Sub
58 Sub Task_SavedMsg()
59     With MindMap.Shapes("TaskSavedMsg" )

```

```

1 2
56 Dim i As Long, Delay As Double, StartTime As Double
57 .Visible = msoCTrue
58 For i = 1 To 100
59     .Fill.Transparency = i / 100
60     Delay = 0.009
61     StartTime = Timer
62     Do
63         DoEvents
64     Loop While Timer - StartTime < Delay
65 Next i
66 .Visible = msoFalse
67 End With
68 End Sub
69 Sub Task_SaveUpdate()
70     With MindMap
71         If .Range("B1" ).Value < 4 Then
72             MsgBox "Please make sure to add a Project, Staff, Task Name and Status before
73                 saving this task"
74             Exit Sub
75         End If
76         'Check For New Project
77         If .Range("B3" ).Value = "" Then
78             If MsgBox("This project has not previously been added. Would you like to save
79                 it & continue?" , vbYesNo, "New Project" ) = vbNo Then Exit Sub
80             ProjRow = Projects.Range("A9999" ).End(xlUp).Row + 1
81             Projects.Range("A" & ProjRow).Value = .Range("B4" ).Value 'Next Project ID
82             Projects.Range("B" & ProjRow).Value = .Range("E6" ).Value 'Project Name
83             'Sort Projects By Name
84             With Projects
85                 LastRow = .Range("A9999" ).End(xlUp).Row
86                 .Range("N3:O999" ).ClearContents 'Clear Previous Results
87                 .Range("N3:N" & LastRow).Value = .Range("B3:B" & LastRow).Value 'Copy
88                 Over Projects for sorting
89                 With .Sort
90                     .SortFields.Clear
91                     .SortFields.Add Key:=Projects.Range("N3" ), SortOn:=xlSortOnValues,
92                     Order:=xlAscending, DataOption:=xlSortNormal 'Sort Alphabetically
93                     .SetRange Projects.Range("N3:N" & LastRow)
94                     .Apply
95                 End With
96                 .Range("O3:O" & LastRow).Value = .Range("N3:N" & LastRow).Value 'Copy
97                 Projects for Additional List
98             End With
99         End If
100         'Check For New Staff
101         If .Range("B9" ).Value = "" Then
102             If MsgBox("This Staff Name has not previously been added. Would you like to
103                 save it & continue?" , vbYesNo, "New Staff" ) = vbNo Then Exit Sub
104             StaffRow = Staff.Range("A9999" ).End(xlUp).Row + 1
105             Staff.Range("A" & StaffRow).Value = .Range("B10" ).Value 'Next Staff ID
106             Staff.Range("B" & StaffRow).Value = .Range("E7" ).Value 'Staff Name
107             'Sort Staff By Name
108             With Staff
109                 LastRow = .Range("A9999" ).End(xlUp).Row
110                 .Range("Q3:R999" ).ClearContents 'Clear Previous Results
111                 .Range("Q3:Q" & LastRow).Value = .Range("B3:B" & LastRow).Value 'Copy
112                 Over Staff for sorting
113             End With
114         End If
115     End With
116 End Sub
1 2 3 4 5

```

```

109     .SortFields.Clear
110     .SortFields.Add Key:=Staff.Range("Q3" ), SortOn:=xlSortOnValues, Order:=
xlAscending, DataOption:=xlSortNormal 'Sort Alphabetically
111     .SetRange Staff.Range("Q3:Q" & LastRow)
112     .Apply
113     End With
114     .Range("R3:R" & LastRow).Value = .Range("Q3:Q" & LastRow).Value 'Copy
Staff for Additional List
115     End With
116     End If
117
118     'Check For New or Existing Task
119     If .Range("B6" ).Value = "" Then 'New task
120         TaskRow = Tasks.Range("A99999" ).End(xlUp).Row + 1 'First Avail. Task Row
121         .Range("B5" ).Value = .Range("B7" ).Value 'Set Task ID
122         Tasks.Range("A" & TaskRow).Value = .Range("B5" ).Value 'Task ID
123     Else 'Existing Task
124         TaskRow = .Range("B6" ).Value 'Existing Task Row
125     End If
126     For TaskCol = 2 To 10
127         Tasks.Cells(TaskRow, TaskCol).Value = .Range(Tasks.Cells(1, TaskCol).Value).
Value 'Save/Update Data
128     Next TaskCol
129     End With
130     MindMap_LoadTasks 'Run macro to load tasks
131     Task_SavedMsg
132 End Sub
133
134 Sub Task_SelectFromOverdueTasks()
135     With MindMap
136         SelRow = .Range("B13" ).Value 'Selected Row
137         TaskID = .Range("S" & SelRow).Value 'Task ID
138         .Range("B5" ).Value = TaskID
139         ProjName = .Range("U" & SelRow).Value 'Project Name
140         On Error Resume Next
141         ProjRow = Projects.Range("Proj_Name" ).Find(ProjName, , xlValues, xlWhole).Row
142         On Error GoTo 0
143         If ProjRow = 0 Then Exit Sub
144         ProjID = Projects.Range("A" & ProjRow).Value 'Project ID
145         .Range("B2" ).Value = ProjID 'Set Selected Project ID
146         MindMap_LoadTasks 'Run Macro to Load tasks
147         MindMap_ViewTask 'Run Macro to view task
148     End With
149 End Sub
150
151
152 Sub Task_SelectFromStaffList()
153     With MindMap
154         SelRow = .Range("B12" ).Value 'Selected Row
155         TaskID = .Range("C" & SelRow).Value 'Task ID
156         .Range("B5" ).Value = TaskID
157         ProjName = .Range("E" & SelRow).Value 'Project Name
158         On Error Resume Next
159         ProjRow = Projects.Range("Proj_Name" ).Find(ProjName, , xlValues, xlWhole).Row
160         On Error GoTo 0
161         If ProjRow = 0 Then Exit Sub
162         ProjID = Projects.Range("A" & ProjRow).Value 'Project ID
163         .Range("B2" ).Value = ProjID 'Set Selected Project ID
164         MindMap_LoadTasks 'Run Macro to Load tasks
165         MindMap_ViewTask 'Run Macro to view task

```

166	1	End With
167	2	End Sub



**A**

Activate, 12  
 ActiveCell, 8  
 Add, 8, 15, 19, 20  
 Admin, 8, 12, 13, 15, 18  
 Admin\_SetStatusColor, 8  
 AdvancedFilter, 11-15, 18  
 AllowMultiSelect, 8  
 Application, 8, 10, 11, 14, 15, 17  
 Apply, 15, 19, 20  
 AutoSize, 11, 13, 15

**B**

BadPath, 13  
 BeginConnect, 11, 12, 14, 16  
 BrowseForStaffPicFolder, 8

**C**

Caller, 8, 10, 11, 14, 17  
 Cells, 17, 20  
 Clear, 15, 19, 20  
 ClearContents, 2, 10, 12, 13, 18, 19  
 Color, 8, 15  
 ConnectorFormat, 11, 12, 14, 16  
 Copy, 12  
 CopyToRange, 11-15, 18  
 CountLarge, 2  
 CriteriaRange, 11-15, 18  
 CustomOrder, 15

**D**

DataOption, 15, 19, 20  
 Delay, 19  
 Delete, 8-10, 12, 14, 18  
 Dir, 8, 13  
 DoEvents, 19  
 Duplicate, 11-16

**E**

Empty, 3, 18  
 EndConnect, 12, 14, 16  
 EndMacro, 16, 17  
 EntireRow, 18  
 Explicit, 2-8, 10, 18

**F**

FileDialog, 8  
 Fill, 8, 13, 16, 19  
 Filters, 8  
 Find, 15, 18, 20  
 ForeColor, 8, 16  
 Formula, 13

**G**

Group, 16  
 GroupCount, 14, 17

**H**

Height, 8, 11, 13, 16, 17

**I**

i, 19  
 Icon\_Add, 8  
 Icon\_Clear, 8  
 IconFile, 8  
 IconNumb, 10, 12, 13  
 Insert, 8  
 InStr, 10, 12  
 Interior, 8, 15

Intersect, 2, 3

**K**

Key, 15, 19, 20

**L**

LastProjRow, 14, 16, 17  
 LastResultRow, 8, 10-15, 18  
 LastRow, 8, 10-15, 18-20  
 Left, 2, 8, 11-16  
 Len, 16  
 Line, 16

**M**

MapRow, 18  
 MindMap, 10-20  
 MindMap\_CollapseTasks, 10  
 MindMap\_HideProjects, 10  
 MindMap\_LoadProjects, 10  
 MindMap\_LoadStaff, 12, 18  
 MindMap\_LoadTasks, 14, 20  
 MindMap\_ViewTask, 17, 20  
 MMLLeft, 10, 11, 13, 15, 16  
 MMShp, 10-12  
 MMTop, 10-17  
 MsgBox, 11, 15, 17-19  
 msoAutoSizeShapeToFitText, 11, 13, 15  
 msoBringToFront, 2  
 msoCTrue, 2, 8, 10, 16, 19  
 msoFalse, 2, 8, 11, 15, 19  
 msoFileDialogFilePicker, 8  
 msoFileDialogFolderPicker, 8

**N**

Name, 8, 10-16  
 NoProj, 14, 17  
 NoSelection, 8  
 NotSaved, 18

**O**

OnAction, 11, 12, 14-16  
 Order, 15, 19, 20

**P**

Paste, 13  
 PicFolder, 10, 13  
 PicPath, 10, 13  
 Pictures, 8  
 ProjCount, 10, 11  
 Projects, 19, 20  
 ProjID, 10-12, 14-18, 20  
 ProjName, 18, 20  
 ProjNm, 10, 11, 14, 15  
 ProjRow, 14, 16-20

**R**

Range, 2, 3, 8, 10-20  
 Replace, 10, 14, 17  
 ResultRow, 10-16  
 RGB, 8, 16  
 Row, 2, 3, 11-16, 18-20

**S**

ScreenUpdating, 15, 17  
 SelectedItems, 8  
 SelRow, 8, 9, 18, 20  
 SetRange, 15, 19, 20  
 Shape, 10, 14  
 Shapes, 2, 8-18  
 Show, 8

ShrinkShape, 14  
 ShtName, 8  
 Sort, 15, 19  
 SortFields, 15, 19, 20  
 SortOn, 15, 19, 20  
 Split, 16  
 Staff, 19, 20  
 StaffFolder, 8  
 StaffID, 10-14  
 StaffNm, 10, 13  
 StaffPic, 10, 13  
 StaffRow, 18, 19  
 StartTime, 19

**T**

Target, 2, 3  
 Task\_CheckForOverdue, 14, 18  
 Task\_Delete, 18  
 Task\_MarkAllAsOverdue, 18  
 Task\_New, 18  
 Task\_SavedMsg, 18, 20  
 Task\_SaveUpdate, 19  
 Task\_SelectFromOverdueTasks, 3, 20  
 Task\_SelectFromStaffList, 3, 20  
 TaskCol, 17, 18, 20  
 TaskColor, 14-16  
 TaskCount, 14, 15, 18  
 TaskGrp, 14, 16, 17  
 TaskGrpArr, 14, 16  
 TaskGrpString, 14, 16  
 TaskID, 14-18, 20  
 TaskListRow, 10  
 TaskNm, 14, 15  
 TaskRow, 14, 15, 17, 18, 20  
 Tasks, 11, 13, 15, 17, 18, 20  
 TaskStatus, 14, 15  
 Text, 10, 11, 13, 15  
 TextFrame, 11, 13, 15  
 TextFrame2, 10, 11, 13, 15  
 TextRange, 10, 11, 13, 15  
 Timer, 19  
 Title, 8  
 Top, 2, 8, 11-17  
 TotalTaskHeight, 14, 17  
 Transparency, 19

**U**

Unique, 11-15, 18  
 UserPicture, 13

**V**

Value, 2, 3, 8, 10-20  
 vbDirectory, 13  
 vbNo, 18, 19  
 vbYesNo, 18, 19  
 Visible, 2, 8, 10, 11, 15, 16, 19

**W**

Width, 8, 11-14, 16  
 Worksheet\_SelectionChange, 2, 3

**X**

xlAscending, 15, 19, 20  
 xlFilterCopy, 11-15, 18  
 xlSortNormal, 15, 19, 20  
 xlSortOnValues, 15, 19, 20  
 xlUp, 11-16, 18-20  
 xlValues, 15, 18, 20  
 xlWhole, 15, 18, 20

**Z**

ZOrder, [2](#)

# Thank You!

This source code was created and made available to help you gain a better understanding of how VBA is used to create amazing Excel-based applications.

Thank you so much for your continued shares, likes and support. It really helps.



*Excel For Freelancers*