

# VBA SOURCE CODE BOOK



**How To Create A Complete  
Multi-User Store  
Management System In Excel**



**DOWNLOAD  
APPLICATION**



**VIEW  
TRAINING**

*by: Randy Austin*

# ABOUT THE AUTHOR

A 4-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 471,000 YouTube subscribers, 35,448,742 video views, 430+ comprehensive training videos, and a thriving 65,000 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®**  
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](#)

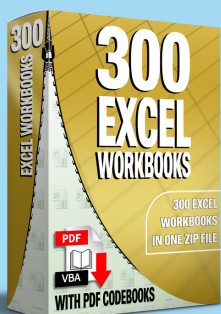
-



This incredible 13-hour freelancer masterclass will teach anyone how to be a successful freelancer with my proven 9-Phase 'Financial Freedom Roadmap' and includes 30+ downloads and exercises.

[Click here to learn more](#)

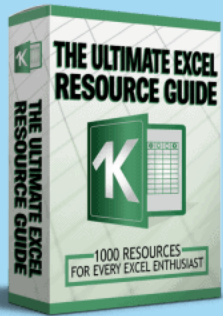
-



Incredible Package of 300 of my BEST Applications with PDF VBA Codebooks packed into a SINGLE ZIP File which also includes the "300 Workbook Library".

[Click here to learn more](#)

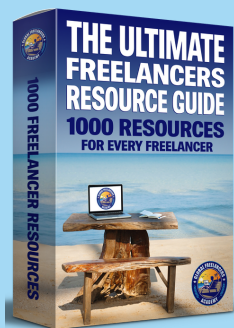
# OUR COURSES & PRODUCTS



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](#)

-



Freelancing essentials from freelance tools to freelance templates, in this Ultimate Freelancer's Resource Guide for freelancers at any stage in their career. With 1,000 Live Links, and a single click-to-update application, you will always have the most current and up-to-date information at your fingertips.

[Click here to learn more](#)

-



Revolutionize the way you work with Excel and take productivity to the next level with the Excel AI Toolpack - the FIRST AI tool designed for ANY Windows Desktop version of Excel. This incredible add-in combines FIVE powerful AI tools, transforming your Excel into an intelligent powerhouse!

[Click here to learn more](#)

Projects.....	3
VBAProject.....	3
Documents.....	3
ChartData.....	3
(Declarations).....	3
OrderItems.....	4
(Declarations).....	4
Worksheet_Change [Sub ].....	4
Orders.....	5
(Declarations).....	5
Worksheet_Change [Sub ].....	5
Products.....	6
(Declarations).....	6
Worksheet_Change [Sub ].....	6
StoreMgmt.....	7
(Declarations).....	7
Worksheet_Change [Sub ].....	7
Worksheet_SelectionChange [Sub ].....	9
ThisWorkbook.....	10
(Declarations).....	10
Users.....	11
(Declarations).....	11
Worksheet_Change [Sub ].....	11
Forms.....	12
LoginFrm.....	12
(Declarations).....	12
CancelBtn_Click [Sub ].....	12
OkBtn_Click [Sub ].....	12
ProductFrm.....	13
(Declarations).....	13
BrowseBtn_Click [Sub ].....	13
CancelBtn_Click [Sub ].....	13
SaveBtn_Click [Sub ].....	13
UserFrm.....	14
(Declarations).....	14
BrowseBtn_Click [Sub ].....	14
CancelBtn_Click [Sub ].....	14
SaveBtn_Click [Sub ].....	14
Modules.....	15
Admin_Macros.....	15
(Declarations).....	15
BrowseForMainSharedFolder [Sub ].....	15
BrowseForProductPicFolder [Sub ].....	15
BrowseForUserPicFolder [Sub ].....	15
MainMenuSelect [Sub ].....	15
CoAuthoring_Macros.....	17
(Declarations).....	17
SharedFolder_Check [Sub ].....	17
SyncTimer [Sub ].....	17
Workbook_CheckAndSync [Sub ].....	17
LoginOut_Macros.....	19
(Declarations).....	19
UserLogin [Sub ].....	19
UserLogin_Show [Sub ].....	19
UserLogout [Sub ].....	19
Product_Macros.....	21
(Declarations).....	21
Product_AddNew [Sub ].....	21
Product_AddPicture [Sub ].....	21
Product_Delete [Sub ].....	21
Product_Edit [Sub ].....	21
Product_RefreshList [Sub ].....	22
Product_SaveUpdate [Sub ].....	22
PurchOrder_Macros.....	23
(Declarations).....	23
PurchOrder_AddNew [Sub ].....	23

## Table of Contents

PurchOrder_Delete [Sub ].....	23
PurchOrder_Load [Sub ].....	23
PurchOrder_Print [Sub ].....	24
PurchOrder_SaveUpdate [Sub ] .....	24
SalesOrder_Macros .....	26
(Declarations) .....	26
SalesOrder_AddNew [Sub ] .....	26
SalesOrder_Delete [Sub ] .....	26
SalesOrder_Load [Sub ] .....	26
SalesOrder_Print [Sub ] .....	27
SalesOrder_ResetTotals [Sub ] .....	27
SalesOrder_SaveUpdate [Sub ] .....	27
User_Macros.....	29
(Declarations) .....	29
User_AddNew [Sub ] .....	29
User_AddPicture [Sub ] .....	29
User_Delete [Sub ] .....	29
User_Edit [Sub ].....	29
User_RefreshList [Sub ].....	30
User_SaveUpdate [Sub ] .....	30

1	Option Explicit
2	

```
1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     Dim SharedFold As String, UserName As String, FileName As String, FilePath As String
5     , FileType As String, OldValue As String
6     Dim SheetNm As String, CellAdd As String, CellVal As String, UserFolder As String,
7     CurrentUser As String
8     Dim UserRow As Long, LastUserRow As Long, ChangeRow As Long, TextFile As Integer
9     If Not Intersect(Target, Range("A4:K999999" )) Is Nothing Then 'Change this range
10    based on your data range or table
11        If Target.Count > 1 Then 'Changes to more than 1 cell at a time are not allowed
12            Application.EnableEvents = False
13            Application.Undo
14            Application.EnableEvents = True
15        End
16    End If
17    If StoreMgmt.Range("B35" ).Value = False Then 'Run When Sync is not active
18        SharedFolder_Check 'Check For Shared Folder
19        SharedFold = [SharedFolder] 'Set Shared Folder
20        CurrentUser = StoreMgmt.Range("B21" ).Value 'Current User Name
21        SheetNm = Target.Worksheet.Name 'Sheet Name
22        CellAdd = Target.Address 'Cell Address
23        CellVal = Target.Value 'Cell Value
24        LastUserRow = Users.Range("A9999" ).End(xlUp).Row 'Last User Row
25        If LastUserRow < 3 Then Exit Sub 'Exit On No Users
26        For UserRow = 4 To LastUserRow 'Loop Through All Users To Add Change to their
27        respective folders (except current user)
28            UserName = Users.Range("B" & UserRow).Value 'User Name
29            If CurrentUser = UserName Then GoTo NextUser 'No need to add changes into
30            current users folder
31            If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then MkDir (
32            SharedFold & "\User Changes" )
33            If Dir(SharedFold & "\User Changes\" & UserName, vbDirectory) = "" Then
34            MkDir (SharedFold & "\User Changes\" & UserName) 'Create user Folder if
35            it does not exist
36            FilePath = SharedFold & "\User Changes\" & UserName & "\" & SheetNm &
37            "--" & CellAdd & ".txt" 'Create Text File Name
38            TextFile = FreeFile 'Assign Unique & Free File #
39            On Error Resume Next
40            Open FilePath For Output As TextFile
41            Print #TextFile, CurrentUser 'Add Current User
42            Print #TextFile, CellVal 'Write Cell Value
43            On Error GoTo 0
44            Close #TextFile 'Close Text file
45            NextUser:
46        Next UserRow
47    End If
48 End If
49 End Sub
```



```

1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     Dim SharedFold As String, UserName As String, FileName As String, FilePath As String
5     , FileType As String, OldValue As String
6     Dim SheetNm As String, CellAdd As String, CellVal As String, UserFolder As String,
7     CurrentUser As String
8     Dim UserRow As Long, LastUserRow As Long, ChangeRow As Long, TextFile As Integer
9     If Not Intersect(Target, Range("A4:F99999" )) Is Nothing Then 'Change this range
10    based on your data range or table
11    If Target.Count > 1 Then 'Changes to more than 1 cell at a time are not allowed
12    Application.EnableEvents = False
13    Application.Undo
14    Application.EnableEvents = True
15    End
16    End If
17    If StoreMgmt.Range("B35" ).Value = False Then 'Run When Sync is not active
18    SharedFolder_Check 'Check For Shared Folder
19    SharedFold = [SharedFolder] 'Set Shared Folder
20    CurrentUser = StoreMgmt.Range("B21" ).Value 'Current User Name
21    SheetNm = Target.Worksheet.Name 'Sheet Name
22    CellAdd = Target.Address 'Cell Address
23    CellVal = Target.Value 'Cell Value
24    LastUserRow = Users.Range("A9999" ).End(xlUp).Row 'Last User Row
25    If LastUserRow < 3 Then Exit Sub 'Exit On No Users
26    For UserRow = 4 To LastUserRow 'Loop Through All Users To Add Change to their
27    respective folders (except current user)
28    UserName = Users.Range("B" & UserRow).Value 'User Name
29    If CurrentUser = UserName Then GoTo NextUser 'No need to add changes into
30    current users folder
31    If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then Mkdir (
32    SharedFold & "\User Changes" )
33    If Dir(SharedFold & "\User Changes\" & UserName, vbDirectory) = "" Then
34    Mkdir (SharedFold & "\User Changes\" & UserName) 'Create user Folder if
35    it does not exist
36    FilePath = SharedFold & "\User Changes\" & UserName & "\" & SheetNm &
37    "--" & CellAdd & ".txt" 'Create Text File Name
38    TextFile = FreeFile 'Assign Unique & Free File #
39    Open FilePath For Output As TextFile
40    Print #TextFile, CurrentUser 'Add Current User
41    Print #TextFile, CellVal 'Write Cell Value
42    Close #TextFile 'Close Text file
43    NextUser:
44    Next UserRow
45    End If
46    End If
47    End Sub

```

```

1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     Dim SharedFold As String, UserName As String, FileName As String, FilePath As String
5     , FileText As String, OldValue As String
6     Dim SheetNm As String, CellAdd As String, CellVal As String, UserFolder As String,
7     CurrentUser As String
8     Dim UserRow As Long, LastUserRow As Long, ChangeRow As Long, TextFile As Integer
9     If Not Intersect(Target, Range("A4:J99999" )) Is Nothing Then 'Change this range
10    based on your data range or table
11    If Target.Count > 1 Then 'Changes to more than 1 cell at a time are not allowed
12    Application.EnableEvents = False
13    Application.Undo
14    Application.EnableEvents = True
15    End
16    End If
17    If StoreMgmt.Range("B35" ).Value = False Then 'Run When Sync is not active
18    SharedFolder_Check 'Check For Shared Folder
19    SharedFold = [SharedFolder] 'Set Shared Folder
20    CurrentUser = StoreMgmt.Range("B21" ).Value 'Current User Name
21    SheetNm = Target.Worksheet.Name 'Sheet Name
22    CellAdd = Target.Address 'Cell Address
23    CellVal = Target.Value 'Cell Value
24    LastUserRow = Users.Range("A9999" ).End(xlUp).Row 'Last User Row
25    If LastUserRow < 3 Then Exit Sub 'Exit On No Users
26    For UserRow = 4 To LastUserRow 'Loop Through All Users To Add Change to their
27    respective folders (except current user)
28    UserName = Users.Range("B" & UserRow).Value 'User Name
29    If CurrentUser = UserName Then GoTo NextUser 'No need to add changes into
30    current users folder
31    If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then Mkdir (
32    SharedFold & "\User Changes" )
33    If Dir(SharedFold & "\User Changes\" & UserName, vbDirectory) = "" Then
34    Mkdir (SharedFold & "\User Changes\" & UserName) 'Create user Folder if
35    it does not exist
36    FilePath = SharedFold & "\User Changes\" & UserName & "\" & SheetNm &
37    "--" & CellAdd & ".txt" 'Create Text File Name
38    TextFile = FreeFile 'Assign Unique & Free File #
39    Open FilePath For Output As TextFile
40    Print #TextFile, CurrentUser 'Add Current User
41    Print #TextFile, CellVal 'Write Cell Value
42    Close #TextFile 'Close Text file
43    NextUser:
44    Next UserRow
45    End If
46    End If
47    End Sub

```

```

1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     Dim ProdName As String, ProdID As String, PicFolder As String, PicName As String,
5     PicPath As String
6     Dim ItemRow As Long, ProdRow As Long
7     If Target.CountLarge > 1 Then Exit Sub
8
9     '''DASHBOARD MACROS
10    If Not Intersect(Target, Range("H4" )) Is Nothing And Range("H4" ).Value <> Empty
11    And Range("B3" ).Value <> "" Then
12        Dim DtRngRow As Long
13        DtRngRow = Range("B3" ).Value
14        Range("B1" ).Value = True 'Set Date Range Change To True
15        Range("J4" ).Value = Range("BN" & DtRngRow).Value 'From Date
16        Range("L4" ).Value = Range("BO" & DtRngRow).Value 'To Date
17        Range("B1" ).Value = False 'Set Date Range Change To False
18    End If
19
20    '''SALES ORDER MACROS
21
22    'On Search Sales Order Change
23    If Not Intersect(Target, Range("X5" )) Is Nothing And Range("X5" ).Value <> Empty
24    Then
25        Range("B4" ).Value = Range("X5" ).Value 'Set Order ID
26        SalesOrder_Load 'Run macro to load sales order
27    End If
28
29    'On Bar Code Scanned Item /Product Name Change
30    If Not Intersect(Target, Range("P5,P6" )) Is Nothing Then
31        On Error Resume Next
32        If Target.Value = Empty Then Exit Sub
33        Shapes("ProdPic" ).Delete 'Remove any product pictures
34        On Error GoTo 0
35        'On Bar Code Change
36        If Not Intersect(Target, Range("P5" )) Is Nothing And Range("P5" ).Value <> Empty
37        Then
38            If Range("B6" ).Value = Empty Then
39                MsgBox "Please make sure to enter a correct bar code #"
40                Exit Sub
41            End If
42            ProdRow = Range("B6" ).Value 'Product Row
43            Range("P6" ).ClearContents
44        End If
45        'On Product Name Change
46        If Not Intersect(Target, Range("P6" )) Is Nothing And Range("P6" ).Value <> Empty
47        Then
48            If Range("B7" ).Value = Empty Then
49                MsgBox "Please make sure to enter a correct product name"
50                Exit Sub
51            End If
52            ProdRow = Range("B7" ).Value 'Product Row
53            Range("P5" ).ClearContents
54        End If
55        If ProdRow = 0 Then Exit Sub
56        'Look for existing Item
57        ProdID = Products.Range("A" & ProdRow).Value 'Product ID
58        'Show Picture (if any)
59        PicName = Products.Range("E" & ProdRow).Value 'Picture Name
60        PicFolder = [ProductFolder] 'Product Picture Folder

```

```

1 2
57 PicPath = PicFolder & "\" & PicName 'Picture Path
58 If PicName = Empty Or PicFolder = Empty Then GoTo NoPic
59 If Dir(PicPath, vbDirectory) = "" Then GoTo NoPic
60 Pictures.Insert(PicPath).Name = "ProdPic"
61 With Shapes("ProdPic" )
62     .LockAspectRatio = msoCTrue
63     If .Width > .Height Then .Width = 250 Else .Height = 150
64     .Left = Range("O9" ).Left + (Range("O:P" ).Width - .Width) / 2
65     .Top = Range("O9" ).Top + (Range("9:21" ).Height - .Height) / 2
66 End With
67 NoPic:
68 On Error Resume Next
69 ItemRow = Range("Q12:Q999" ).Find(ProdID, , xlFormulas, xlWhole).Row
70 On Error GoTo 0
71 If ItemRow <> 0 Then
72     Range("S" & ItemRow).Value = Range("S" & ItemRow).Value + 1 'Increment Qty
73     Up 1
74 Else 'New Item
75     ItemRow = Range("R99999" ).End(xlUp).Row + 1 'First Available Item Row
76     Range("T" & ItemRow & ":U" & ItemRow + 4).ClearContents 'Clear Totals
77     'Add Product Details
78     Range("Q" & ItemRow).Value = ProdID ' Product ID
79     Range("R" & ItemRow).Value = Products.Range("C" & ProdRow).Value 'Product
80     Name
81     Range("S" & ItemRow).Value = 1 'Set Default Qty
82     Range("T" & ItemRow).Value = Products.Range("G" & ProdRow).Value 'Sales
83     Price
84     Range("U" & ItemRow).Formula = "=S" & ItemRow & "*T" & ItemRow 'Total
85     Formula
86 End If
87
88 Range("B10" ).ClearContents 'Clear Selected Row
89 SalesOrder_ResetTotals
90 Target.Select
91 End If
92
93 ''''PURCHASE ORDER MACROS
94 'On Search Purchase Order Change
95 If Not Intersect(Target, Range("AI5" )) Is Nothing And Range("AI5" ).Value <> Empty
96 And Range("B14" ).Value <> Empty Then
97     Range("B12" ).Value = Range("AI5" ).Value 'Set Order ID
98     PurchOrder_Load 'Run macro to load sales order
99 End If
100
101 'On Purchase Order Item Name Change, but not on load
102 If Not Intersect(Target, Range("AA10:AA177" )) Is Nothing And Range("B16" ).Value =
103 False Then
104     If Range("AA" & Target.Row).Value <> Empty Then
105         ProdName = Target.Value
106         On Error Resume Next
107         ProdRow = Products.Range("Prod_Name" ).Find(ProdName, , xlValues, xlWhole).Row
108         'Prod Row
109         On Error GoTo 0
110         If ProdRow = 0 Then Exit Sub
111         Range("Z" & Target.Row).Value = Products.Range("A" & ProdRow).Value 'Prod.
112         ID
113         Range("AB" & Target.Row).Value = Products.Range("B" & ProdRow).Value
114         'UPC/SKU
115         Range("AC" & Target.Row).Value = Products.Range("D" & ProdRow).Value
116         'Description

```

```

108 1 2 3 Range("AD" & Target.Row).Value = 1 'Default Qty
109 Range("AE" & Target.Row).Value = Products.Range("F" & ProdRow).Value
      'Purchase Price
110 } Else 'Empty
111   Range("Z" & Target.Row & ":AE" & Target.Row).ClearContents
112 } End If
113 End If
114
115
116
117
118
119
120 End Sub
-----
121
122 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
123     '''PRODUCT ORDER MACROS
124     If Target.CountLarge > 1 Then Exit Sub
125     If Shapes("SelProdGrp" ).Visible = True Then Shapes("SelProdGrp" ).Visible =
msoFalse
126     If Shapes("SelUserGrp" ).Visible = True Then Shapes("SelUserGrp" ).Visible =
msoFalse
127     If Not Intersect(Target, Range("AL7:AT999" )) Is Nothing And Range("AL" & Target.
Row).Value <> Empty Then
128         Range("B18" ).Value = Target.Row
129         With Shapes("SelProdGrp" )
130             .Left = Range("AU" & Target.Row).Left
131             .Top = Range("AU" & Target.Row).Top
132             .Visible = msoCTrue
133         End With
134     End If
135
136     '''USER MACROS
137     If Not Intersect(Target, Range("AY7:BB999" )) Is Nothing And Range("AY" & Target.
Row).Value <> Empty Then
138         Range("B26" ).Value = Target.Row
139         With Shapes("SelUserGrp" )
140             .Left = Range("BC" & Target.Row).Left
141             .Top = Range("BC" & Target.Row).Top
142             .Visible = msoCTrue
143         End With
144     Else
145         'Range("B26").ClearContents
146     End If
147
148 End Sub

```

1 Option Explicit

2

```

1 Option Explicit
2
3 Private Sub Worksheet_Change(ByVal Target As Range)
4     Dim SharedFold As String, UserName As String, FileName As String, FilePath As String
5     , FileType As String, OldValue As String
6     Dim SheetNm As String, CellAdd As String, CellVal As String, UserFolder As String,
7     CurrentUser As String
8     Dim UserRow As Long, LastUserRow As Long, ChangeRow As Long, TextFile As Integer
9     If Not Intersect(Target, Range("A4:F99999" )) Is Nothing Then 'Change this range
10    based on your data range or table
11    If Target.Count > 1 Then 'Changes to more than 1 cell at a time are not allowed
12    Application.EnableEvents = False
13    Application.Undo
14    Application.EnableEvents = True
15    End
16    End If
17    If StoreMgmt.Range("B35" ).Value = False Then 'Run When Sync is not active
18    SharedFolder_Check 'Check For Shared Folder
19    SharedFold = [SharedFolder] 'Set Shared Folder
20    CurrentUser = StoreMgmt.Range("B21" ).Value 'Current User Name
21    SheetNm = Target.Worksheet.Name 'Sheet Name
22    CellAdd = Target.Address 'Cell Address
23    CellVal = Target.Value 'Cell Value
24    LastUserRow = Users.Range("A9999" ).End(xlUp).Row 'Last User Row
25    If LastUserRow < 3 Then Exit Sub 'Exit On No Users
26    For UserRow = 4 To LastUserRow 'Loop Through All Users To Add Change to their
27    respective folders (except current user)
28    UserName = Users.Range("B" & UserRow).Value 'User Name
29    If CurrentUser = UserName Then GoTo NextUser 'No need to add changes into
30    current users folder
31    If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then Mkdir (
32    SharedFold & "\User Changes" )
33    If Dir(SharedFold & "\User Changes\" & UserName, vbDirectory) = "" Then
34    Mkdir (SharedFold & "\User Changes\" & UserName) 'Create user Folder if
35    it does not exist
36    FilePath = SharedFold & "\User Changes\" & UserName & "\" & SheetNm &
37    "--" & CellAdd & ".txt" 'Create Text File Name
38    TextFile = FreeFile 'Assign Unique & Free File #
39    Open FilePath For Output As TextFile
40    Print #TextFile, CurrentUser 'Add Current User
41    Print #TextFile, CellVal 'Write Cell Value
42    Close #TextFile 'Close Text file
43    NextUser:
44    Next UserRow
45    End If
46    End If
47    End Sub

```

```
1  
2 Option Explicit  
3  
4 Private Sub CancelBtn_Click()  
5     Unload Me  
6 End Sub  
7  
8 Private Sub OkBtn_Click()  
9     UserLogin  
10 End Sub  
11
```



```
1  
2  
3 Option Explicit  
4  
5 Private Sub BrowseBtn_Click()  
6     Product_AddPicture  
7 End Sub  
8  
9 Private Sub CancelBtn_Click()  
10    Unload Me  
11 End Sub  
12  
13  
14 Private Sub SaveBtn_Click()  
15    Product_SaveUpdate  
16 End Sub
```

```
1  
2  
3 Option Explicit  
4  
5 Private Sub BrowseBtn_Click()  
6     User_AddPicture  
7 End Sub  
8  
9 Private Sub CancelBtn_Click()  
10    Unload Me  
11 End Sub  
12  
13  
14 Private Sub SaveBtn_Click()  
15    User_SaveUpdate  
16 End Sub
```

```
1 Option Explicit
2
3 Sub BrowseForMainSharedFolder()
4     Dim SharedFolder As FileDialog
5     Set SharedFolder = Application.FileDialog(msoFileDialogFolderPicker)
6     With SharedFolder
7         .Title = "Please select the main shared folder within Dropbox as instructed by
            your StoreMgmtistrator"
8         .AllowMultiSelect = False
9         If .Show <> -1 Then GoTo NotSelected
10        If InStr(.SelectedItems(1), "Dropbox" ) = 0 Then
11            MsgBox "Please make sure to select a folder within Drobpox"
12            Exit Sub
13        End If
14        StoreMgmt.Range("BI5" ).Value = .SelectedItems(1) 'Full Folder Path
15        NotSelected:
16    End With
17 End Sub
18
19
20 Sub BrowseForProductPicFolder()
21     Dim PicFolder As FileDialog
22     Set PicFolder = Application.FileDialog(msoFileDialogFolderPicker)
23     With PicFolder
24         .Title = "Browse Product Picture Folder"
25         .AllowMultiSelect = False
26         If .Show <> -1 Then GoTo NoSelection
27         StoreMgmt.Range("BI6" ).Value = .SelectedItems(1)
28     End With
29     NoSelection:
30 End Sub
31
32 Sub BrowseForUserPicFolder()
33     Dim PicFolder As FileDialog
34     Set PicFolder = Application.FileDialog(msoFileDialogFolderPicker)
35     With PicFolder
36         .Title = "Browse User Picture Folder"
37         .AllowMultiSelect = False
38         If .Show <> -1 Then GoTo NoSelection
39         StoreMgmt.Range("BI7" ).Value = .SelectedItems(1)
40     End With
41     NoSelection:
42 End Sub
43
44
45
46
47
48 Sub MainMenuSelect()
49     Dim GrpShp As Shape, MenuItem As String
50     Application.ScreenUpdating = False
51     With StoreMgmt
52         On Error Resume Next
53         MenuItem = Application.Caller
54         On Error GoTo 0
55         If MenuItem = "Login" Then MenuItem = "Sales" 'Set Default to Sales for
            initial login
56         .Shapes("Menu" ).Fill.Transparency = 1 '100%
57         .Shapes(MenuItem).Fill.ForeColor.RGB = RGB(52, 152, 219) 'Color Selected Button
58         .Shapes(MenuItem).Fill.Transparency = 0
59         .Range("B2" ).Value = MenuItem
```

1 2

```
1 2
60 For Each GrpShp In .Shapes
61     If InStr(GrpShp.Name, "Grp" ) Then GrpShp.Visible = msoFalse 'Hide Group
        Shapes
62     Next GrpShp
63     .Range("E:BQ" ).EntireColumn.Hidden = True
64     .Shapes(MenuItem & "Grp" ).Visible = msoCTrue 'Unhide Tab Group
65     On Error Resume Next
66     .Shapes("ProdPic" ).Visible = msoFalse
67     On Error GoTo 0
68     Select Case MenuItem
69     Case Is = "Dashboard"
70         .Range("E:M" ).EntireColumn.Hidden = False
71     Case Is = "Sales"
72         .Range("N:Y" ).EntireColumn.Hidden = False
73         On Error Resume Next
74         .Shapes("ProdPic" ).Visible = msoCTrue
75         On Error GoTo 0
76     Case Is = "Purchases"
77         .Range("Z:AJ" ).EntireColumn.Hidden = False
78     Case Is = "Products"
79         .Range("AK:AW" ).EntireColumn.Hidden = False
80     Case Is = "Users"
81         .Range("AX:BF" ).EntireColumn.Hidden = False
82     Case Is = "Settings"
83         .Range("BG:BQ" ).EntireColumn.Hidden = False
84     End Select
85 End With
86 Application.ScreenUpdating = True
87 End Sub
```

```

1 Option Explicit
2 Dim SharedFold As String, Role As String, UserName As String
3
4 Sub SharedFolder_Check()
5     'Check For Correct Shared Folder
6     SharedFold = [SharedFolder] 'Set Shared Folder
7     If SharedFold = "" Or Dir(SharedFold, vbDirectory) = "" Or InStr(SharedFold,
8         "Dropbox" ) = 0 Then
9         MsgBox "Please select a correct dropbox folder"
10        BrowseForMainSharedFolder
11        SharedFold = [SharedFolder] 'Set Shared Folder
12        If SharedFold = "" Or Dir(SharedFold, vbDirectory) = "" Or InStr(SharedFold,
13            "Dropbox" ) = 0 Then End
14    End If
15    'Check For User Picture Folder
16    If Dir(SharedFold & "/User Pics" , vbDirectory) = "" Then Mkdir (SharedFold &
17        "/User Pics" ) 'Create User Picture folder if missing
18 End Sub
19
20 Sub SyncTimer()
21     Dim SyncFreq As Double
22     'SyncFreq = StoreMgmt.Range("").Value 'Sync Frequency
23     'Application.OnTime Now + SyncFreq, "Workbook_CheckAndSync" 'Run At Specified
24     Interval
25 End Sub
26
27
28
29
30
31 Sub Workbook_CheckAndSync()
32     Dim FileName As String, FilePath As String, FileText As String, OldValue As String,
33     TextArr() As String, NewValue
34     Dim SheetNm As String, CellAdd As String, CellVal As String, SentFrom As String,
35     Mesg As String, UserFolder As String, CurrentUser As String
36     Dim MsgNumb As Long, ChangeRow As Long, TextFile As Integer
37     Dim TopPos As Double, LeftPos As Double
38     With StoreMgmt
39         If .Range("B21" ).Value = Empty Then Exit Sub 'Exit On New User
40         SharedFolder_Check 'Run Macro to check for Correct shared folder
41         SharedFold = [SharedFolder] 'Set Shared Folder
42         CurrentUser = .Range("B21" ).Value 'Set User Name
43         If Dir(SharedFold & "\User Changes\" , vbDirectory) = "" Then Mkdir (SharedFold
44             & "\User Changes" )
45         UserFolder = SharedFold & "\User Changes\" & CurrentUser & "\" 'User Folder
46         If Dir(UserFolder, vbDirectory) = "" Then Mkdir (UserFolder) 'Create Folder if
47             it does not exist
48         MsgNumb = 1 'Set Intial Message Number
49         .Range("B35" ).Value = True 'Set File Sync To True
50         FileName = Dir(UserFolder & "*.txt" ) 'Check for any/all text files in folder
51         TextFile = FreeFile
52         Do While Len(FileName) > 0
53             FilePath = UserFolder & FileName
54             Open FilePath For Input As TextFile
55             FileText = Input(LOF(TextFile), TextFile)
56             Close TextFile
57             TextArr = Split(FileText, vbNewLine)

```

1 2 3

```
54      1 2 3      UserName = TextArr(0) 'User Who made change
55      1 2 3      NewValue = TextArr(1) 'Changed (new) Value
56      1 2 3      Close #1
57
58      'All Cell Changes
59      SheetNm = Left(FileName, InStr(FileName, "--" ) - 1) 'Extract Sheet Name
60      CellAdd = Replace(Right(FileName, Len(FileName) - InStr(FileName, "--" ) - 1),
61      ".txt" , "" ) 'Cell Address
62      ThisWorkbook.Sheets(SheetNm).Range(CellAdd).Value = NewValue 'Make Cell
63      Change
64      NextFile:
65      FileName = Dir() 'Clear File Name
66      Kill (FilePath) 'Delete File
67      Loop
68      .Range("B35" ).Value = False 'Set Syncing to False
69      'SyncTimer 'Run Sync Timer
70      End With
71      End Sub
```

```

1 Option Explicit
2 Dim UserRow As Long
3 Dim UserName As String, UserPW As String, PicName As String, PicFolder As String,
  PicPath As String
4 Dim GrpShp As Shape
5 Sub UserLogin()
6     With LoginFrm
7         UserName = .UserName
8         UserPW = .Password
9         If UserName = Empty Or UserPW = Empty Then
10            MsgBox "Please make sure to enter a Username and Password"
11            Exit Sub
12        End If
13        On Error Resume Next
14        UserRow = Users.Range("User_Name" ).Find(UserName, , xlValues, xlWhole).Row
15        On Error GoTo 0
16        If UserRow = 0 Then
17            MsgBox "Please enter a correct user name"
18            Exit Sub
19        End If
20        If Users.Range("C" & UserRow).Value <> UserPW Then
21            MsgBox "Password has not been entered correctly. Please try again"
22            Exit Sub
23        End If
24        Unload LoginFrm
25    End With
26
27    'User Logged in Successfully
28    With StoreMgmt
29        .Range("B21" ).Value = UserName 'Set Username
30        .Range("B22" ).Value = Users.Range("D" & UserRow).Value 'Set User Type
31        PicName = Users.Range("E" & UserRow).Value 'User Picture
32        PicFolder = [UserFolder]
33        PicPath = PicFolder & "\" & PicName 'User Picture Path
34        If PicName = Empty Or PicFolder = Empty Then GoTo NoPic
35        If Dir(PicPath, vbDirectory) = "" Then GoTo NoPic
36        .Shapes("Header" ).GroupItems("UserPic" ).Fill.UserPicture PicPath 'Add User
  Picture
37        NoPic:
38        .Range("B23" ).Value = Now 'Set Logged In Date/Time
39        .Shapes("Header" ).Visible = msoCTrue 'Unhide Header
40        .Shapes("Menu" ).Visible = msoCTrue 'Unhide Menu
41        .Range("C:D" ).EntireColumn.Hidden = False 'Unhide Column C
42        MainMenuSelect 'Run Macro To Load Sales
43        ActiveWindow.ScrollColumn = 1
44    End With
45 End Sub
46
47 Sub UserLogin_Show()
48     LoginFrm.Show
49 End Sub
50
51 Sub UserLogout()
52     With StoreMgmt
53         .Range("B21:B23" ).ClearContents 'Clear User Info
54         .Range("C:BZ" ).EntireColumn.Hidden = True
55         .Shapes("Menu" ).Visible = msoFalse
56         For Each GrpShp In .Shapes
57             If InStr(GrpShp.Name, "Grp" ) Then GrpShp.Visible = msoFalse 'Hide all group
  shapes
58         Next GrpShp

```

```
59 | 1 2 .Shapes("Header" ).Visible = msoFalse  
60 |     .Shapes("StartGrp" ).Visible = msoCTrue  
61 |     End With  
62 | End Sub
```



```

1 Option Explicit
2 Dim ProdRow As Long, SelRow As Long, ProdCol As Long, LastRow As Long, LastResultRow As Long
3 Dim ProdFld As control
4
5 Sub Product_AddNew()
6     StoreMgmt.Range("B18").ClearContents 'Clear Selected Product Row
7     ProductFrm.Show
8 End Sub
9
10 Sub Product_AddPicture()
11     Dim ProdPic As FileDialog
12     Dim PicName As String, PicPath As String, PicFolder As String
13
14     If [ProductFolder] = "" Then
15         MsgBox "Please select a folder for product pictures"
16         BrowseForProductPicFolder
17         If [ProductFolder] = "" Then Exit Sub
18     End If
19     PicFolder = [ProductFolder] 'Set Picture Folder
20     Set ProdPic = Application.FileDialog(msoFileDialogFilePicker)
21     With ProdPic
22         .Title = "Select Product Picture"
23         .Filters.Add "Select Picture", "*.jpg, *.png, *.jpeg, *.gif", 1
24         If .Show <> -1 Then GoTo NoSelection
25         PicPath = .SelectedItems(1) 'Entire Picture Path
26     End With
27     PicName = Dir(PicPath) 'Set Picture Name
28     ProductFrm.Field4.Value = PicName 'Set Picture Name
29     'Copy Product Picture to picture folder
30     On Error Resume Next
31     FileCopy PicPath, PicFolder & "\" & PicName
32     On Error GoTo 0
33     NoSelection:
34 End Sub
35
36 Sub Product_Delete()
37     If MsgBox("Are you sure you want to delete this product?" , vbYesNo, "Delete Product" ) = vbNo Then Exit Sub
38     If StoreMgmt.Range("B18").Value = Empty Then Exit Sub
39     SelRow = StoreMgmt.Range("B18").Value 'Selected Row
40     ProdRow = StoreMgmt.Range("AU" & SelRow).Value 'Product Row
41     For ProdCol = 2 To 10
42         Products.Cells(ProdRow, ProdCol).ClearContents
43     Next ProdCol
44     StoreMgmt.Range("B18").ClearContents
45     Product_RefreshList
46 End Sub
47
48
49 Sub Product_Edit()
50     With StoreMgmt
51         If .Range("B18").Value = Empty Then
52             MsgBox "Please select on a product to edit"
53             Exit Sub
54         End If
55
56         SelRow = .Range("B18").Value 'Product Row
57         ProdRow = .Range("AU" & SelRow).Value 'Product Row
58         With ProductFrm
59             For ProdCol = 2 To 7

```

1 2 3 4

```

1 2 3 4
60 Set ProdFld = .Controls("Field" & ProdCol - 1)
61 ProdFld.Value = Products.Cells(ProdRow, ProdCol).Value 'Add Product Data
62 Next ProdCol
63 .Show 'Display User Form
64 End With
65 End With
66 End Sub

67
68 Sub Product_RefreshList()
69 With Products
70 StoreMgmt.Range("AL7:AU9999" ).ClearContents
71 LastRow = .Range("A9999" ).End(xlUp).Row
72 If LastRow < 4 Then Exit Sub
73 .Range("A3:J" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
74 "M2:M3" ), CopyToRange:=.Range("P2:Y2" ), Unique:=True
75 LastResultRow = .Range("P9999" ).End(xlUp).Row
76 If LastResultRow < 3 Then Exit Sub
77 StoreMgmt.Range("AL7:AU" & LastResultRow + 4).Value = .Range("P3:Y" &
78 LastResultRow).Value
79 End With
80 End Sub

81 Sub Product_SaveUpdate()
82 'Check For Required field
83 With ProductFrm
84 If .Field2.Value = Empty Then
85 MsgBox "Please make sure to add a Product Name before saving"
86 Exit Sub
87 End If
88 If StoreMgmt.Range("B18" ).Value = Empty Then 'New Product
89 ProdRow = Products.Range("A9999" ).End(xlUp).Row + 1 'First Avail. Row
90 Products.Range("A" & ProdRow).Value = StoreMgmt.Range("B19" ).Value 'Next
91 Product ID
92 Products.Range("J" & ProdRow).Value = "=Row()"
93 Else 'Existing Prod.
94 SelRow = StoreMgmt.Range("B18" ).Value 'Selected Row
95 ProdRow = StoreMgmt.Range("AU" & SelRow).Value 'Product Row
96 End If
97 For ProdCol = 2 To 7
98 Set ProdFld = .Controls("Field" & ProdCol - 1)
99 Products.Cells(ProdRow, ProdCol).Value = ProdFld.Value
100 Next ProdCol
101 End With
102 Unload ProductFrm
103 Product_RefreshList
104 End Sub

```

```

1 Option Explicit
2 Dim OrderRow As Long, OrderCol As Long, ItemRow As Long, LastRow As Long, LastItemRow
   As Long, ItemDBRow As Long, ItemDBCol As Long
3 Dim LastResultRow As Long, ResultRow As Long
4 Dim OrderType As String, NameType As String
5 Sub PurchOrder_AddNew()
6     With StoreMgmt
7         .Range("B12,B15,AB6,Z10:AE177,AI5,AG10:AG177" ).ClearContents
8         .Range("B12" ).Value = .Range("B17" ).Value 'Next Order ID
9         .Range("AB8" ).Value = Now 'Set Current & Time
10        .Range("AB6" ).Select
11    End With
12 End Sub
13
14 Sub PurchOrder_Delete()
15     With StoreMgmt
16         If MsgBox("Are you sure you want to delete this order?" , vbYesNo, "Delete
           Order" ) = vbNo Then Exit Sub
17         If .Range("B13" ).Value = Empty Then GoTo NotSaved
18         OrderRow = .Range("B13" ).Value
19         For OrderCol = 2 To 6
20             Orders.Cells(OrderRow, OrderCol).ClearContents
21         Next OrderCol
22         'Delete Order Items
23         With OrderItems
24             LastRow = .Range("A99999" ).End(xlUp).Row
25             If LastRow < 4 Then Exit Sub
26             .Range("A3:K" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
               "Y2:Y3" ), CopyToRange:=.Range("AB2:AI2" ), Unique:=True
27             LastResultRow = .Range("AC99999" ).End(xlUp).Row
28             If LastResultRow < 3 Then Exit Sub
29             For ResultRow = 3 To LastResultRow
30                 ItemDBRow = .Range("AI" & ResultRow).Value 'Item DB Row
31                 For ItemDBCol = 2 To 11
32                     .Cells(ItemDBRow, ItemDBCol).ClearContents 'Clear Contents
33                 Next ItemDBCol
34             Next ResultRow
35         End With
36         NotSaved:
37         PurchOrder_AddNew
38     End With
39 End Sub
40
41 Sub PurchOrder_Load()
42     With StoreMgmt
43         .Range("B15,AB6,Z10:AE177,AI5,AG10:AG177" ).ClearContents
44         If .Range("B13" ).Value = Empty Then
45             MsgBox "Please enter a correct Order # to load"
46             Exit Sub
47         End If
48         OrderRow = .Range("B13" ).Value
49         OrderType = Orders.Range("B" & OrderRow).Value
50         If OrderType = "Sale" Then
51             MsgBox "The order # entered is a Sales Order. Please select on Sales & enter
               the order # again"
52             Exit Sub
53         End If
54         .Range("AB6" ).Value = Orders.Range("C" & OrderRow).Value 'Supplier Name
55         .Range("AB8" ).Value = Orders.Range("D" & OrderRow).Value 'Order Date & Time
56     End With
57     'Load Items

```

```

1
58 With OrderItems
59 LastRow = .Range("A9999" ).End(xlUp).Row
60 If LastRow < 4 Then Exit Sub
61 .Range("A3:K" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
"Y2:Y3" ), CopyToRange:=.Range("AB2:AJ2" ), Unique:=True
62 LastResultRow = .Range("AC9999" ).End(xlUp).Row
63 If LastResultRow < 3 Then Exit Sub
64 StoreMgmt.Range("B16" ).Value = True 'Set PO Load To True
65 For ResultRow = 3 To LastResultRow
66 ItemRow = .Range("AJ" & ResultRow).Value 'Item Row
67 StoreMgmt.Range("Z" & ItemRow & ":AE" & ItemRow).Value = .Range("AB" &
ResultRow & ":AG" & ResultRow).Value 'Product ID, Name, Qty & Amount
68 StoreMgmt.Range("AG" & ItemRow).Value = .Range("AI" & ResultRow).Value 'DB
Row
69 Next ResultRow
70 StoreMgmt.Range("B16" ).Value = False 'Set PO Load To False
71 End With
72 End Sub

73
74 Sub PurchOrder_Print()
75 With StoreMgmt
76 LastRow = .Range("AA9999" ).End(xlUp).Row
77 .PageSetup.PrintArea = "$AA$5:$AF$" & LastRow
78 .PageSetup.Orientation = xlLandscape
79 .PrintOut , , , , True, , , False
80 End With
81 End Sub

82
83
84 Sub PurchOrder_SaveUpdate()
85 With StoreMgmt
86 If .Range("AB6" ).Value = Empty Then
87 MsgBox "Please enter a supplier before saving this Purchase Order"
88 Exit Sub
89 End If
90
91 LastItemRow = .Range("AA177" ).End(xlUp).Row
92 If LastItemRow < 10 Then
93 MsgBox "Please make sure to enter at least one product before saving"
94 Exit Sub
95 End If
96 If .Range("B13" ).Value = Empty Then 'New Order
97 OrderRow = Orders.Range("A176" ).End(xlUp).Row + 1
98 .Range("B12" ).Value = .Range("B17" ).Value 'Next Order ID
99 Orders.Range("A" & OrderRow).Value = .Range("B17" ).Value 'Next Order ID
100 Else 'Existing Order
101 OrderRow = .Range("B13" ).Value 'Existing Order DB Row
102 End If
103 Orders.Range("B" & OrderRow).Value = "Purchase" 'Set Order Type To Sale
104 Orders.Range("C" & OrderRow).Value = .Range("AB6" ).Value 'Supplier
105 Orders.Range("D" & OrderRow).Value = Now 'Order Time & Date
106 Orders.Range("E" & OrderRow).Value = .Range("AF8" ).Value 'Purchase Order Total

107
108 'Save/Udpate Items
109 For ItemRow = 10 To LastItemRow
110 If .Range("AG" & ItemRow).Value = Empty Then 'New Item DB
111 ItemDBRow = OrderItems.Range("A9999" ).End(xlUp).Row + 1 'First availRow
112 OrderItems.Range("A" & ItemDBRow).Value = .Range("B12" ).Value 'Purch.
Order ID
113 OrderItems.Range("B" & ItemDBRow).Value = "Purchase" 'Set Order Type To

```

```
1 2 3 4
Purchase
114 OrderItems.Range("J" & ItemDBRow).Value = ItemRow 'Set Item Row
115 OrderItems.Range("K" & ItemDBRow).Value = "=Row()"
116 .Range("AG" & ItemRow).Value = ItemDBRow
117 Else 'Existing
118     ItemDBRow = .Range("AG" & ItemRow).Value 'Existing Order Items DB Row
119 End If
120 OrderItems.Range("C" & ItemDBRow).Value = .Range("Z" & ItemRow).Value
    'Product ID
121 OrderItems.Range("D" & ItemDBRow).Value = .Range("AA" & ItemRow).Value
    'Product Name
122 OrderItems.Range("E" & ItemDBRow).Value = .Range("AB" & ItemRow).Value 'UPC
123 OrderItems.Range("F" & ItemDBRow).Value = .Range("AC" & ItemRow).Value
    'Prod Desc.
124 OrderItems.Range("G" & ItemDBRow).Value = .Range("AD" & ItemRow).Value
    'Item Qty
125 OrderItems.Range("H" & ItemDBRow).Value = .Range("AE" & ItemRow).Value
    'Price
126 OrderItems.Range("I" & ItemDBRow).Value = .Range("AF" & ItemRow).Value
    'Total
127 Next ItemRow
128 NoItems:
129     MsgBox "Purchase Order Saved"
130 End With
131 End Sub
```

```

1 Option Explicit
2 Dim OrderRow As Long, OrderCol As Long, ItemRow As Long, LastRow As Long, LastItemRow
   As Long, ItemDBRow As Long, ItemDBCol As Long
3 Dim LastResultRow As Long, ResultRow As Long
4 Dim OrderType As String, NameType As String
5 Sub SalesOrder_AddNew()
6     With StoreMgmt
7         .Range("B4,E5:E7,P5:P7,X5,X6,Q12:V99999" ).ClearContents
8         .Range("B4" ).Value = .Range("B17" ).Value 'Next Order ID
9         .Range("B9" ).Value = Now 'Set Current & Time
10        On Error Resume Next
11        .Shapes("ProdPic" ).Delete 'Delete any product picture
12        On Error GoTo 0
13        .Range("P5" ).Select
14    End With
15 End Sub
16
17 Sub SalesOrder_Delete()
18     With StoreMgmt
19         If MsgBox("Are you sure you want to delete this order?" , vbYesNo, "Delete
   Order" ) = vbNo Then Exit Sub
20         If .Range("B5" ).Value = Empty Then GoTo NotSaved
21         OrderRow = .Range("B5" ).Value
22         For OrderCol = 2 To 6
23             Orders.Cells(OrderRow, OrderCol).ClearContents
24         Next OrderCol
25
26         'Delete Order Items
27         With OrderItems
28             LastRow = .Range("A99999" ).End(xlUp).Row
29             If LastRow < 4 Then Exit Sub
30             .Range("A3:K" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
   "N2:N3" ), CopyToRange:=.Range("Q2:V2" ), Unique:=True
31             LastResultRow = .Range("Q99999" ).End(xlUp).Row
32             If LastResultRow < 3 Then Exit Sub
33             For ResultRow = 3 To LastResultRow
34                 ItemDBRow = .Range("U" & ResultRow).Value 'Item DB Row
35                 For ItemDBCol = 2 To 11
36                     .Cells(ItemDBRow, ItemDBCol).ClearContents 'Clear Contents
37                 Next ItemDBCol
38             Next ResultRow
39         End With
40         NotSaved:
41         SalesOrder_AddNew
42     End With
43 End Sub
44
45 Sub SalesOrder_Load()
46     With StoreMgmt
47         .Range("E5:E7,X5,X6,P5:P7,Q12:V99999" ).ClearContents
48         On Error Resume Next
49         .Shapes("ProdPic" ).Delete 'Remove any product pictures
50         On Error GoTo 0
51         If .Range("B5" ).Value = Empty Then
52             MsgBox "Please enter a correct Order # to load"
53             Exit Sub
54         End If
55         OrderRow = .Range("B5" ).Value
56         OrderType = Orders.Range("B" & OrderRow).Value
57         If OrderType = "Purchase" Then
58             MsgBox "The order # entered is a Purchase Order. Please select on Purchases &

```

1 2 3

```

1 2 3
    enter the order # again"
59     Exit Sub
60     End If
61     Application.ScreenUpdating = False
62     .Range("P7" ).Value = Orders.Range("C" & OrderRow).Value 'Customer Name
63     .Range("B9" ).Value = Orders.Range("D" & OrderRow).Value 'Order Date & Time
64     .Range("X6" ).Value = Orders.Range("F" & OrderRow).Value 'Payment Amount
65     End With
66     'Load Items
67     With OrderItems
68         LastRow = .Range("A99999" ).End(xlUp).Row
69         If LastRow < 4 Then Exit Sub
70         .Range("A3:K" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
71             "N2:N3" ), CopyToRange:=.Range("Q2:V2" ), Unique:=True
72         LastResultRow = .Range("R99999" ).End(xlUp).Row
73         If LastResultRow < 3 Then
74             Application.ScreenUpdating = True
75             Exit Sub
76         End If
77         For ResultRow = 3 To LastResultRow
78             ItemRow = .Range("V" & ResultRow).Value 'Item Row
79             StoreMgmt.Range("Q" & ItemRow & ":T" & ItemRow).Value = .Range("Q" &
80                 ResultRow & ":T" & ResultRow).Value 'Product ID, Name, Qty & Amount
81             StoreMgmt.Range("V" & ItemRow).Value = .Range("U" & ResultRow).Value 'DB
82             Row
83         Next ResultRow
84         StoreMgmt.Range("U12:U" & LastResultRow + 9).Formula = StoreMgmt.Range("U4" ).
85         Formula 'Bring down Total Formulas
86         SalesOrder_ResetTotals
87     End With
88     Application.ScreenUpdating = True
89     StoreMgmt.Range("P5" ).Select
90 End Sub
91
92 Sub SalesOrder_Print()
93     With StoreMgmt
94         LastRow = .Range("R99999" ).End(xlUp).Row + 5
95         .PageSetup.PrintArea = "$R$5:$U$" & LastRow
96         .PageSetup.Orientation = xlPortrait
97         .PrintOut , , , True, , , False
98     End With
99 End Sub
100
101 Sub SalesOrder_ResetTotals()
102     With StoreMgmt
103         LastRow = .Range("R99999" ).End(xlUp).Row
104         If LastRow < 12 Then Exit Sub
105         .Range("T" & LastRow + 1 & ":U" & LastRow + 5).ClearContents 'Clear any
106         previous totals
107         .Range("TotalRange" ).Copy
108         .Range("T" & LastRow + 1).PasteSpecial xlPasteFormulas
109         Application.CutCopyMode = False
110     End With
111 End Sub
112
113 Sub SalesOrder_SaveUpdate()
114     With StoreMgmt
115         LastItemRow = .Range("R99999" ).End(xlUp).Row
116         If LastItemRow < 12 Then
117             MsgBox "Please make sure to enter at least one product before saving"
118             Exit Sub
119         End If
120     End With
121 End Sub
1 2 3

```

```

114     End If
115     Application.ScreenUpdating = False
116     If .Range("B5" ).Value = Empty Then 'New Order
117         OrderRow = Orders.Range("A99999" ).End(xlUp).Row + 1
118         .Range("B4" ).Value = .Range("B17" ).Value 'Next Order ID
119         Orders.Range("A" & OrderRow).Value = .Range("B17" ).Value 'Next Order ID
120     Else 'Existing Order
121         OrderRow = .Range("B5" ).Value 'Existing Order DB Row
122     End If
123     Orders.Range("B" & OrderRow).Value = "Sale" 'Set Order Type To Sale
124     Orders.Range("C" & OrderRow).Value = .Range("P7" ).Value 'Customer (if any)
125     Orders.Range("D" & OrderRow).Value = Now 'Order Time & Date
126     Orders.Range("E" & OrderRow).Value = .Range("B31" ).Value 'Order Total
127     Orders.Range("F" & OrderRow).Value = .Range("X6" ).Value 'Payment Amount
128
129     'Save/Udpate Items
130     For ItemRow = 12 To LastItemRow
131         If .Range("V" & ItemRow).Value = Empty Then 'New Item DB
132             ItemDBRow = OrderItems.Range("A9999" ).End(xlUp).Row + 1 'First avail Row
133
134             OrderItems.Range("A" & ItemDBRow).Value = .Range("B4" ).Value 'Order #
135             OrderItems.Range("B" & ItemDBRow).Value = "Sale" 'Set Order Type To
136             Sale
137             OrderItems.Range("J" & ItemDBRow).Value = ItemRow 'Set Item Row
138             OrderItems.Range("K" & ItemDBRow).Value = "=Row()" 'Database row formula
139
140             .Range("V" & ItemRow).Value = ItemDBRow
141         Else 'Existing
142             ItemDBRow = .Range("V" & ItemRow).Value 'Existing Order Items DB Row
143         End If
144         OrderItems.Range("C" & ItemDBRow).Value = .Range("Q" & ItemRow).Value
145         'Product ID
146         OrderItems.Range("D" & ItemDBRow).Value = .Range("R" & ItemRow).Value
147         'Product Name
148         OrderItems.Range("G" & ItemDBRow).Value = .Range("S" & ItemRow).Value 'Item
149         Qty
150         OrderItems.Range("H" & ItemDBRow).Value = .Range("T" & ItemRow).Value
151         'Price
152         OrderItems.Range("I" & ItemDBRow).Value = .Range("U" & ItemRow).Value
153         'Total
154     Next ItemRow
155     NoItems:
156     Application.ScreenUpdating = True
157     MsgBox "Sales Order Saved"
158 End With
159 End Sub

```



```

1 Option Explicit
2 Dim UserRow As Long, SelRow As Long, UserCol As Long, LastRow As Long, LastResultRow As Long
3 Dim UserFld As control
4
5 Sub User_AddNew()
6     StoreMgmt.Range("B26").ClearContents 'Clear Selected User Row
7     UserFrm.Show
8 End Sub
9
10 Sub User_AddPicture()
11     Dim UserPic As FileDialog
12     Dim PicName As String, PicPath As String, PicFolder As String
13
14     If [UserFolder] = "" Then
15         MsgBox "Please select a folder for User pictures"
16         BrowseForUserPicFolder
17         If [UserFolder] = "" Then Exit Sub
18     End If
19     PicFolder = [UserFolder] 'Set Picture Folder
20     Set UserPic = Application.FileDialog(msoFileDialogFilePicker)
21     With UserPic
22         .Title = "Select User Picture"
23         .Filters.Add "Select Picture", "*.jpg, *.png, *.jpeg, *.gif", 1
24         If .Show <> -1 Then GoTo NoSelection
25         PicPath = .SelectedItems(1) 'Entire Picture Path
26     End With
27     PicName = Dir(PicPath) 'Set Picture Name
28     UserFrm.Field4.Value = PicName 'Set Picture Name
29     'Copy User Picture to picture folder
30     On Error Resume Next
31     FileCopy PicPath, PicFolder & "\" & PicName
32     On Error GoTo 0
33     NoSelection:
34 End Sub
35
36 Sub User_Delete()
37     If MsgBox("Are you sure you want to delete this User?" , vbYesNo, "Delete User" ) = vbNo Then Exit Sub
38     If StoreMgmt.Range("B26").Value = Empty Then Exit Sub
39     SelRow = StoreMgmt.Range("B26").Value 'Selected Row
40     UserRow = StoreMgmt.Range("BC" & SelRow).Value 'User Row
41     For UserCol = 2 To 6
42         Users.Cells(UserRow, UserCol).ClearContents
43     Next UserCol
44     StoreMgmt.Range("B26").ClearContents
45     User_RefreshList
46 End Sub
47
48
49
50 Sub User_Edit()
51     With StoreMgmt
52         If .Range("B26").Value = Empty Then
53             MsgBox "Please select on a User to edit"
54             Exit Sub
55         End If
56         SelRow = .Range("B26").Value 'User Row
57         UserRow = .Range("BC" & SelRow).Value 'User Row
58         With UserFrm
59             For UserCol = 2 To 5

```

1 2 3 4

```

1 2 3 4
60     Set UserFld = .Controls("Field" & UserCol - 1)
61     UserFld.Value = Users.Cells(UserRow, UserCol).Value 'Add User Data
62     Next UserCol
63     .Show 'Display User Form
64     End With
65     End With
66     End Sub
67
68     Sub User_RefreshList()
69     With Users
70     StoreMgmt.Range("AY7:BC9999" ).ClearContents
71     LastRow = .Range("A9999" ).End(xlUp).Row
72     If LastRow < 4 Then Exit Sub
73     .Range("A3:F" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
74     "K2:K3" ), CopyToRange:=.Range("N2:R2" ), Unique:=True
75     LastResultRow = .Range("N9999" ).End(xlUp).Row
76     If LastResultRow < 3 Then Exit Sub
77     StoreMgmt.Range("AY7:BC" & LastResultRow + 4).Value = .Range("N3:R" &
78     LastResultRow).Value
79     End With
80     End Sub
81     Sub User_SaveUpdate()
82     'Check For Required field
83     With UserFrm
84     If .Field1.Value = Empty Then
85     MsgBox "Please make sure to add a User Name before saving"
86     Exit Sub
87     End If
88     If StoreMgmt.Range("B26" ).Value = Empty Then 'New User
89     UserRow = Users.Range("A9999" ).End(xlUp).Row + 1 'First Avail. Row
90     Users.Range("A" & UserRow).Value = StoreMgmt.Range("B27" ).Value 'Next User
91     ID
92     Users.Range("F" & UserRow).Value = "=Row()"
93     Else 'Existing User.
94     SelRow = StoreMgmt.Range("B26" ).Value 'Selected Row
95     UserRow = StoreMgmt.Range("BC" & SelRow).Value 'User Row
96     End If
97     For UserCol = 2 To 5
98     Set UserFld = .Controls("Field" & UserCol - 1)
99     Users.Cells(UserRow, UserCol).Value = UserFld.Value
100    Next UserCol
101    End With
102    Unload UserFrm
103    User_RefreshList
104    End Sub

```

**A**

ActiveWindow, 19  
 Add, 21, 29  
 Address, 4-6, 11  
 AdvancedFilter, 22-24, 26, 27, 30  
 AllowMultiSelect, 15  
 Application, 4-6, 11, 15, 16, 21, 27-29

**B**

BrowseBtn\_Click, 13, 14  
 BrowseForMainSharedFolder, 15, 17  
 BrowseForProductPicFolder, 15, 21  
 BrowseForUserPicFolder, 15, 29

**C**

Caller, 15  
 CancelBtn\_Click, 12-14  
 CellAdd, 4-6, 11, 17, 18  
 Cells, 21-23, 26, 29, 30  
 CellVal, 4-6, 11, 17  
 ChangeRow, 4-6, 11, 17  
 ClearContents, 7-9, 19, 21-23, 26, 27, 29, 30  
 Close, 4-6, 11, 17, 18  
 control, 21, 29  
 Controls, 22, 30  
 Copy, 27  
 CopyToRange, 22-24, 26, 27, 30  
 Count, 4-6, 11  
 CountLarge, 7, 9  
 CriteriaRange, 22-24, 26, 27, 30  
 CurrentUser, 4-6, 11, 17  
 CutCopyMode, 27

**D**

Delete, 7, 26  
 Dir, 4-6, 8, 11, 17-19, 21, 29  
 DtRngRow, 7

**E**

Empty, 7-9, 17, 19, 21-24, 26, 28-30  
 EnableEvents, 4-6, 11  
 EntireColumn, 16, 19  
 Explicit, 3-7, 10-15, 17, 19, 21, 23, 26, 29

**F**

Field1, 30  
 Field2, 22  
 Field4, 21, 29  
 FileCopy, 21, 29  
 FileDialog, 15, 21, 29  
 FileName, 4-6, 11, 17, 18  
 FilePath, 4-6, 11, 17, 18  
 FileText, 4-6, 11, 17  
 Fill, 15, 19  
 Filters, 21, 29  
 Find, 8, 19  
 ForeColor, 15  
 Formula, 8, 27  
 FreeFile, 4-6, 11, 17

**G**

GroupItems, 19  
 GrpShp, 15, 16, 19

**H**

Height, 8  
 Hidden, 16, 19

**I**

Input, 17  
 Insert, 8  
 InStr, 15-19  
 Intersect, 4-9, 11  
 ItemDBCcol, 23, 26  
 ItemDBRow, 23-26, 28  
 ItemRow, 7, 8, 23-28

**K**

Kill, 18

**L**

LastItemRow, 23, 24, 26-28  
 LastResultRow, 21-24, 26, 27, 29, 30  
 LastRow, 21-24, 26, 27, 29, 30  
 LastUserRow, 4-6, 11  
 Left, 8, 9, 18  
 LeftPos, 17  
 Len, 17, 18  
 LockAspectRatio, 8  
 LOF, 17  
 LoginFrm, 19

**M**

MainMenuSelect, 15, 19  
 MenuItem, 15, 16  
 Msg, 17  
 Mkdir, 4-6, 11, 17  
 MsgBox, 7, 15, 17, 19, 21-30  
 MsgNumb, 17  
 msoCTrue, 8, 9, 16, 19, 20  
 msoFalse, 9, 16, 19, 20  
 msoFileDialogFilePicker, 21, 29  
 msoFileDialogFolderPicker, 15

**N**

Name, 4-6, 8, 11, 16, 19  
 NameType, 23, 26  
 NewValue, 17, 18  
 NextFile, 18  
 NextUser, 4-6, 11  
 NoItems, 25, 28  
 NoPic, 8, 19  
 NoSelection, 15, 21, 29  
 NotSaved, 23, 26  
 NotSelected, 15  
 Now, 19, 23, 24, 26, 28

**O**

OkBtn\_Click, 12  
 OldValue, 4-6, 11, 17  
 Open, 4-6, 11, 17  
 OrderCol, 23, 26  
 OrderItems, 23-28  
 OrderRow, 23, 24, 26-28  
 Orders, 23, 24, 26-28  
 OrderType, 23, 26  
 Orientation, 24, 27  
 Output, 4-6, 11

**P**

PageSetup, 24, 27  
 Password, 19  
 PasteSpecial, 27  
 PicFolder, 7, 8, 15, 19, 21, 29  
 PicName, 7, 8, 19, 21, 29  
 PicPath, 7, 8, 19, 21, 29  
 Pictures, 8

Print, 4-6, 11  
 PrintArea, 24, 27  
 PrintOut, 24, 27  
 ProdCol, 21, 22  
 ProdFld, 21, 22  
 ProdID, 7, 8  
 ProdName, 7, 8  
 ProdPic, 21  
 ProdRow, 7-9, 21, 22  
 Product\_AddNew, 21  
 Product\_AddPicture, 13, 21  
 Product\_Delete, 21  
 Product\_Edit, 21  
 Product\_RefreshList, 21, 22  
 Product\_SaveUpdate, 13, 22  
 ProductFolder, 7, 21  
 ProductFrm, 21, 22  
 Products, 7-9, 21, 22  
 PurchOrder\_AddNew, 23  
 PurchOrder\_Delete, 23  
 PurchOrder\_Load, 8, 23  
 PurchOrder\_Print, 24  
 PurchOrder\_SaveUpdate, 24

**R**

Range, 4-9, 11, 15-19, 21-30  
 Replace, 18  
 ResultRow, 23, 24, 26, 27  
 RGB, 15  
 Right, 18  
 Role, 17  
 Row, 4-6, 8, 9, 11, 19, 22-24, 26-28, 30

**S**

SalesOrder\_AddNew, 26  
 SalesOrder\_Delete, 26  
 SalesOrder\_Load, 7, 26  
 SalesOrder\_Print, 27  
 SalesOrder\_ResetTotals, 8, 27  
 SalesOrder\_SaveUpdate, 27  
 SaveBtn\_Click, 13, 14  
 ScreenUpdating, 15, 16, 27, 28  
 ScrollColumn, 19  
 SelectedItems, 15, 21, 29  
 SelRow, 21, 22, 29, 30  
 SentFrom, 17  
 Shape, 15, 19  
 Shapes, 7-9, 15, 16, 19, 20, 26  
 SharedFold, 4-6, 11, 17  
 SharedFolder, 4-6, 11, 15, 17  
 SharedFolder\_Check, 4-6, 11, 17  
 SheetNm, 4-6, 11, 17, 18  
 Sheets, 18  
 Show, 15, 19, 21, 22, 29, 30  
 Split, 17  
 StoreMgmt, 4-6, 11, 15, 17, 19, 21-24, 26, 27, 29, 30  
 SyncFreq, 17  
 SyncTimer, 17

**T**

Target, 4-9, 11  
 TextArr, 17, 18  
 TextFile, 4-6, 11, 17  
 ThisWorkbook, 18  
 Title, 15, 21, 29  
 Top, 8, 9  
 TopPos, 17  
 Transparency, 15

**U**

Undo, 4-6, 11  
Unique, 22-24, 26, 27, 30  
Unload, 12-14, 19, 22, 30  
User\_AddNew, 29  
User\_AddPicture, 14, 29  
User\_Delete, 29  
User\_Edit, 29  
User\_RefreshList, 29, 30  
User\_SaveUpdate, 14, 30  
UserCol, 29, 30  
UserFld, 29, 30  
UserFolder, 4-6, 11, 17, 19, 29  
UserFrm, 29, 30  
UserLogin, 12, 19  
UserLogin\_Show, 19  
UserLogout, 19  
UserName, 4-6, 11, 17-19  
UserPic, 29  
UserPicture, 19  
UserPW, 19  
UserRow, 4-6, 11, 19, 29, 30  
Users, 4-6, 11, 19, 29, 30

**V**

Value, 4-9, 11, 15, 17-19, 21-30  
vbDirectory, 4-6, 8, 11, 17, 19  
vbNewLine, 17  
vbNo, 21, 23, 26, 29  
vbYesNo, 21, 23, 26, 29  
Visible, 9, 16, 19, 20

**W**

Width, 8  
Workbook\_CheckAndSync, 17  
Worksheet, 4-6, 11  
Worksheet\_Change, 4-7, 11  
Worksheet\_SelectionChange, 9

**X**

xlFilterCopy, 22-24, 26, 27, 30  
xlFormulas, 8  
xlLandscape, 24  
xlPasteFormulas, 27  
xlPortrait, 27  
xlUp, 4-6, 8, 11, 22-24, 26-28, 30  
xlValues, 8, 19  
xlWhole, 8, 19

# 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*