VBA SOURCE CODE BOOK

Excel For Freelancers

This Automated Purchase Order Finds The Cheapest Vendors

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

MVP Microsoft® Most Valuable Professional

OUR COURSES & PRODUCTS

MENTORSHIP EXPRESS

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

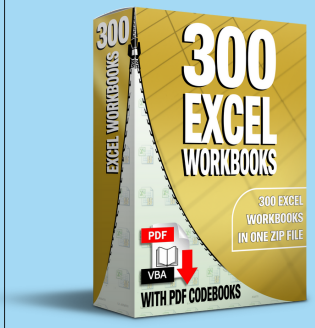
Click here to learn more

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.

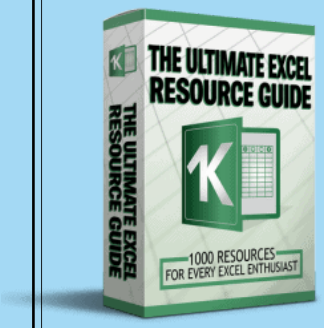
Click here to learn more



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".

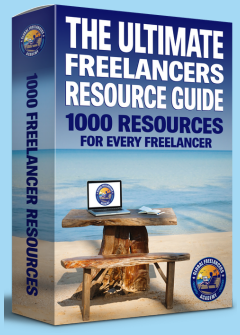
Click here to learn more

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

Table of Contents

```
1   Option Explicit
2
3   Private Sub Worksheet_Change(ByVal Target As Range)
4     'On Custom Field Type Change, Update Format
5     If Not Intersect(Target, Range("E8:E11" )) Is Nothing Then
6       Dim FieldType As String
7       FieldType = Target.Value
8       Select Case FieldType
9       Case Is = "Text"
10        Range("F"  & Target.Row).Value = "@"
11      Case Is = "Amount"
12        Range("F"  & Target.Row).Value = "$#,##0.00"
13      Case Is = "Percentage"
14        Range("F"  & Target.Row).Value = "#%"
15      Case Is = "Date"
16        Range("F"  & Target.Row).Value = "m/d/yyyy"
17      Case Else
18        Range("F"  & Target.Row).Value = "@"
19      End Select
20    End If
21
22    'On Custom Field Type Change, Update Format
23    If Not Intersect(Target, Range("F8:F11" )) Is Nothing Then
24      Dim RowNumb As Long
25      RowNumb = Target.Row
26      Select Case RowNumb
27      Case Is = 8
28        Products.Range("J11" ).NumberFormat = Target.Value
29      Case Is = 9
30        Products.Range("M11" ).NumberFormat = Target.Value
31      Case Is = 10
32        Products.Range("J13" ).NumberFormat = Target.Value
33      Case Is = 11
34        Products.Range("M13" ).NumberFormat = Target.Value
35      End Select
36    End If
37  End Sub
38
39
40
41
```

```
1   Option Explicit
2
```

```
1   Option Explicit
2
```

```
1   Option Explicit
2
3   Private Sub Worksheet_Change(ByVal Target As Range)
4     If Target.CountLarge > 2 Then Exit Sub
5     If Not Intersect(Target, Range("G3" )) Is Nothing Then Product_ListProducts 'List
      Unique Products on Search
6     If Not Intersect(Target, Range("J5" )) Is Nothing And Range("J5" ).Value <> Empty
      Then Products_GetUniqueSubCategories 'Get Unique Sub-Categories
7
8     'On Tier Vendor Change
9     If Not Intersect(Target, Range("M17" )) Is Nothing Then ProdPriceTiers_LoadTiers
10
11    'On Tier Qty Change, but not on Tier load
12    If Not Intersect(Target, Range("N19:N33" )) Is Nothing And Range("B15" ).Value =
      False Then
13      If IsNumeric(Target.Value) = True Then
14        Range("M"  & Target.Row + 1).Value = Target.Value + 1 'Increment Next Qty
15      End If
16    End If
17
18  End Sub
19
20  Private Sub Worksheet_SelectionChange(ByVal Target As Range)
21    If Target.CountLarge > 1 Then Exit Sub
22    If Not Intersect(Target, Range("D6:G9999" )) Is Nothing And Range("D"  & Target.Row)
      .Value <> Empty Then
23      Range("B2" ).Value = Range("C"  & Target.Row).Value
24      Product_Load 'Run Macro to load product
25    End If
26
27    'On Vendors With Price
28    If Not Intersect(Target, Range("I19:J34" )) Is Nothing And Range("I"  & Target.Row).
      Value <> Empty Then
29      'Range("B14").Value = Target.Row 'Set Selected Row
30      Range("M17" ).Value = Range("I"  & Target.Row).Value 'Set Vendor Name
31    End If
32
33
34
35  End Sub
```

```
1   Option Explicit
2
```

```
1   Option Explicit
2
3   Private Sub Worksheet_Change(ByVal Target As Range)
4     If Target.CountLarge > 1 Then Exit Sub
5     'On Item Change but not on PO Load
6     If Not Intersect(Target, Range("G7:G37" )) Is Nothing And Range("B5" ).Value = False
      Then
7       Dim ProdRow As Long
8       If Target.Value <> Empty Then 'Selected Items
9         On Error Resume Next
10        ProdRow = ProductsDB.Range("Prod_Name" ).Find(Target.Value, , xlValues,
          xlWhole).Row
11        On Error GoTo 0
12        If ProdRow = 0 Then Exit Sub
13        Application.ScreenUpdating = False
14        Range("F"  & Target.Row).Value = ProductsDB.Range("A"  & ProdRow).Value
          'Product ID
15        Range("H"  & Target.Row).Value = ProductsDB.Range("C"  & ProdRow).Value 'Prod
          SKU/UPC
16        Range("I"  & Target.Row).Value = ProductsDB.Range("G"  & ProdRow).Value
          'Default Purch Qty
17        Range("J"  & Target.Row).Value = ProductsDB.Range("H"  & ProdRow).Value 'UOM
18        Range("K"  & Target.Row).Value = ProductsDB.Range("I"  & ProdRow).Value 'Base
          Price
19        Range("AA7:AT7" ).Copy
20        Range("AA"  & Target.Row & ":AT"  & Target.Row).PasteSpecial xlPasteFormulas
21        Application.CutCopyMode = False
22        Application.ScreenUpdating = True
23        Range("G"  & Target.Row + 1).Select 'Select Item Below
24      Else
25        Range("F"  & Target.Row & ":K"  & Target.Row).ClearContents 'Clear Item
          Details (not DB Row)
26
27      End If
28    End If
29
30    'On PO # Search
31    If Not Intersect(Target, Range("E2" )) Is Nothing And Range("E2" ).Value <> Empty
      Then
32      Range("B2" ).Value = Range("E2" ).Value 'Set PO ID
33      PO_Load 'Run Macro to Load PO
34    End If
35  End Sub
36
```

```
1   Option Explicit
2
```

```
1   Option Explicit
2
```

```
Option Explicit


Sub AddVendorPricing()
  Dim VendNumb As Long, LastProdRow As Long, VendCol As Long
  With PurchOrder
    VendNumb = Replace(Application.Caller, "AddVend" , "" )
    If VendNumb = 0 Then Exit Sub
    VendCol = 14 + ((VendNumb - 1) * 2)
    LastProdRow = .Range("G35" ).End(xlUp).Row
    If LastProdRow = 6 Then Exit Sub 'Exit on No Product
    .Range("K7:K"  & LastProdRow).Value = Range(.Cells(7, VendCol), .Cells(
    LastProdRow, VendCol)).Value 'Copy Over vendor Prices
    .Range("H2" ).Value = .Cells(4, VendCol).Value 'Set Vendor Name
  End With
End Sub

Sub DisplayTop5Vendors()
  Dim LastRow As Long, LastCol As Long, LastVendRow, VendRow As Long, VendCol As Long
  Dim PriceRng As Range
  With PurchOrder
    .Range("N3:W4,N8:W37,BA2:CI50" ).ClearContents 'Clear Previous Data
    LastRow = .Range("G37" ).End(xlUp).Row
    If LastRow = 6 Then
      MsgBox "Please make sure to add at least one product before displaying top 5
      vendors"
      Exit Sub
    End If
    Application.ScreenUpdating = False
    LastVendRow = VendorDB.Range("A9999" ).End(xlUp).Row
    .Range("BA2:BB"  & LastVendRow - 2).Value = VendorDB.Range("A4:B"  & LastVendRow)
    .Value 'Bring Over Vendors & Vendor ID
    LastCol = LastVendRow + 23
    Set PriceRng = .Range(.Cells(5, 27), .Cells(LastRow, LastCol))
    PriceRng.Copy
    .Range("BC2" ).PasteSpecial xlPasteValues, , , Transpose:=True
    With .Sort
      .SortFields.Clear
      .SortFields.Add Key:=PurchOrder.Range("BC2" ), SortOn:=xlSortOnValues, Order:=
      xlDescending, DataOption:=xlSortNormal  'Sort By Most Matching Products
      .SortFields.Add Key:=PurchOrder.Range("BD2" ), SortOn:=xlSortOnValues, Order:=
      xlAscending, DataOption:=xlSortNormal  'Sort By Lowest Cost
      .SetRange Range(PurchOrder.Cells(2, 53), PurchOrder.Cells(LastVendRow + 2,
      LastRow + 50)) 'Set Range
      .Apply 'Apply Sort
    End With

    'Add Top 5 Vendors
    For VendRow = 2 To 6
      VendCol = 14 + ((VendRow - 2) * 2)
      .Cells(3, VendCol).Value = .Range("BA"  & VendRow).Value 'Vendor ID
      .Cells(4, VendCol).Value = .Range("BB"  & VendRow).Value 'Vendor Name
    Next VendRow
    .Range("N1:W1" ).Copy 'Copy Formulas
    .Range("N7:W"  & LastRow).PasteSpecial xlPasteFormulas
    Application.CutCopyMode = False
    .Range("H2" ).Select
    Application.ScreenUpdating = True
  End With
End Sub
```

```
Option Explicit
Dim PicPath As String, PicFolder As String
#If VBA7 And Win64 Then
Private Declare PtrSafe Function URLDownloadToFile Lib "urlmon"  _
Alias "URLDownloadToFileA"  (ByVal pCaller As Long, _
ByVal szURL As String, ByVal szFileName As String, _
ByVal dwReserved As LongPtr, ByVal lpfnCB As Long) As Long
#Else
Private Declare Function URLDownloadToFile Lib "urlmon"  _
  Alias "URLDownloadToFileA"  ( _
    ByVal pCaller As Long, _
    ByVal szURL As String, _
    ByVal szFileName As String, _
    ByVal dwReserved As Long, _
    ByVal lpfnCB As Long _
  ) As Long
#End If

Sub AddProductPicture()
  Dim ProdPic As FileDialog
  Set ProdPic = Application.FileDialog(msoFileDialogFilePicker)
  PicFolder = [ProductPicFolder]
  If PicFolder = ""  Or Dir(PicFolder, vbDirectory) = ""  Then
    BrowseForProdPicFolder
    PicFolder = [ProductPicFolder]
    If PicFolder = ""  Or Dir(PicFolder, vbDirectory) = ""  Then Exit Sub
  End If
  With ProdPic
    .Title = "Select Item Picture"
    .Filters.Add "Select Item Picture" , "*.jpg, *.png, *.jpeg, *.giff" , 1
    If .Show <> -1 Then GoTo NoSelection
    PicPath = PicFolder & "\"  & Dir(.SelectedItems(1))
    If .SelectedItems(1) = PicPath Then GoTo SkipCopy
    If Dir(PicPath, vbDirectory) <> ""  Then Kill (PicPath)
    FileCopy .SelectedItems(1), PicPath 'Copy to new location
    SkipCopy:
  End With
  Products.Range("B9" ).Value = Dir(PicPath) 'Set Picture Name
  Product_ShowPicture
  Product_SaveUpdate
  NoSelection:
End Sub


Sub BrowseForProdPicFolder()
  Dim ProdPicFolder As FileDialog
  Set ProdPicFolder = Application.FileDialog(msoFileDialogFolderPicker)
  With ProdPicFolder
    .Title = "Browse For Product Picture Folder"
    .AllowMultiSelect = False
    If .Show <> -1 Then GoTo NoSelection
    Admin.Range("D3" ).Value = .SelectedItems(1)
  End With
  NoSelection:
End Sub


Sub ClearProductPicture()
  Products.Range("B9" ).ClearContents
  On Error Resume Next
  Products.Shapes("ProdPic" ).Delete
```

1

```
    On Error GoTo 0
End Sub


Sub GetPictures()
    Dim ProdRow As Long
    With ProdDB
        For ProdRow = 645 To 694
            .Activate
            .Range("B"  & ProdRow).Select
            GetProductPicture
        Next ProdRow
    End With
End Sub

Sub GetProductPicture()
    Dim FileName As String, PicPath As String, PicURL As String, ProdName As String
    Dim Prompt As String, SecretKey As String, Model As String, Temp As String, httpBody
     As String, Response As String
    Dim ProdRow As Long
    Dim oXMLHTTP As Object
    With ProdDB
        If .Range("H2" ).Value = ""  Then
            MsgBox "Please make sure to select a product in order to generate a picture"
            Exit Sub
        End If
        ProdRow = .Range("H2" ).Value 'Product ROw
        ProdName = .Range("B"  & ProdRow).Value 'Product Name
        If ProdName = ""  Then
            MsgBox "Please make sure to select a product in order to generate a picture"
            Exit Sub
        End If

        If [ProductPicFolder] = ""  Then
            MsgBox "Please make sure to set a picture folder in the Admin screen"
            BrowseForProdPicFolder
            If [ProductPicFolder] = ""  Then Exit Sub
        End If
        If Dir([ProductPicFolder], vbDirectory) = ""  Then
            MsgBox "Please make sure to set a picture folder in the Admin screen"
            BrowseForProdPicFolder
            If [ProductPicFolder] = ""  Or Dir([ProductPicFolder], vbDirectory) = ""  Then
             Exit Sub
        End If

        SecretKey = Admin.Range("K8" ).Value 'API Key
        Model = Admin.Range("K9" ).Value  'Model
        Temp = Admin.Range("K10" ).Value  'Temperature
        If Model = ""  Then Model = "text-davinci-003"  'Set Default Model
        If Temp = ""  Then Temp = "0.2"  'Set Default Temp
        Prompt = ProdName & " with white background"
        'Set Prompt to meal name
        Set oXMLHTTP = CreateObject("MSXML2.ServerXMLHTTP" )
        oXMLHTTP.Open "POST" , "https://api.openai.com/v1/images/generations" , False
        'Set Open Posts For Text
        httpBody = "{" "model" ": " "dall-e-3" ", " "prompt" ": " ""  & Prompt & "" "}"
        oXMLHTTP.setRequestHeader "Content-Type" , "application/json"  'Set Header Type
        oXMLHTTP.setRequestHeader "Authorization" , "Bearer "  & SecretKey 'Set Header
        Authorization
        Debug.Print httpBody
        On Error GoTo TimeOut
```

```
119         DoEvents
120         oXMLHTTP.send httpBody
121         On Error GoTo 0
122         Response = oXMLHTTP.responseText 'Extract Response
123         If InStr(Response, "You didn't provide an API key" ) > 0 Then
124           MsgBox "No API Key was added. You can find your API key at "  & vbCrLf &
              "https://platform.openai.com/account/api-keys."
125           Exit Sub
126         End If
127         If InStr(Response, "Incorrect API key provided" ) > 0 Then
128           MsgBox "Incorrect API key provided. You can find your API key at "  & vbCrLf &
               "https://platform.openai.com/account/api-keys."
129           Exit Sub
130         End If
131         If InStr(Response, "maximum context length" ) > 0 Then
132           MsgBox "You have reached the maximum content length. Please clear the history
              above to move forward"
133           Exit Sub
134         End If
135         Debug.Print Response
136         PicURL = Mid(Response, InStr(Response, "url" ) + 7, InStr(Response, "}" ) - InStr
            (Response, "url" ) - 13)
137         FileName = ProdName & ".jpg"    'Set Unique File Name
138         PicPath = [ProductPicFolder] & "\"  & FileName 'Full File Path
139         If Dir(PicPath, vbDirectory) <> ""  Then Kill (PicPath) 'Delete any existing
            picture under the same name
140         URLDownloadToFile 0, PicURL, PicPath, 0, 0
141         .Range("F"  & ProdRow).Value = FileName
142
143
144         Exit Sub
145         TimeOut:
146         MsgBox "The request has timed out or server is busy. Please try again"
147       End With
148     End Sub
149
150     Sub Product_ShowPicture()
151       PicFolder = [ProductPicFolder] 'Product Picture Folder (if any)
152       With Products
153         On Error Resume Next
154         Products.Shapes("ProdPic" ).Delete 'Delete any picture if it exists
155         On Error GoTo 0
156         PicPath = PicFolder & "\"  & .Range("B9" ).Value 'Picture path
157         On Error Resume Next
158         If Dir(PicPath, vbDirectory) = ""  Then Exit Sub
159         On Error GoTo 0
160         .Pictures.Insert(PicPath).Name = "ProdPic"
161         With .Shapes("ProdPic" )
162           .LockAspectRatio = msoCTrue
163           If .Width > .Height Then .Width = 85 Else .Height = 70
164           .Left = Products.Range("O3" ).Left + (Products.Range("O:P" ).Width - .Width) /
               2  'Center Picture Horizontally
165           .Top = Products.Range("O3" ).Top + (Products.Range("3:7" ).Height - .Height) /
               2 'Center Picture Vertically
166         End With
167       End With
168     End Sub
```

```
Option Explicit
Dim LastRow As Long, LastResultRow As Long

Sub ProdPriceTiers_ListVendors()
  Products.Range("H19:J34" ).ClearContents 'Clear Previous Vendors
  With ProdPricingDB
    .Range("O3:Q999" ).ClearContents 'Clear Previous Results
    LastRow = .Range("A99999" ).End(xlUp).Row
    If LastRow < 4 Then Exit Sub
    'Get Unique Vendors & Add Emails
    .Range("A3:D"  & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
    "K2:K3" ), CopyToRange:=.Range("O2:P2" ), Unique:=True
    LastResultRow = .Range("O99999" ).End(xlUp).Row
    If LastResultRow < 3 Then Exit Sub
    .Range("Q3:Q"  & LastResultRow).Formula = .Range("Q1" ).Formula 'Bring Down
    Vendor Email Formula
    Products.Range("H19:J"  & LastResultRow + 16).Value = .Range("O3:Q"  &
    LastResultRow).Value 'Bring over vendors
  End With
End Sub

Sub ProdPriceTiers_LoadTiers()
  Products.Range("M19:Q34" ).ClearContents 'Clear Previous Prices
  With ProdPricingDB
    LastRow = .Range("A99999" ).End(xlUp).Row
    If LastRow < 4 Then Exit Sub
    'Get Unique Vendors & Add Emails
    .Range("A3:J"  & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
    "K2:L3" ), CopyToRange:=.Range("S2:X2" ), Unique:=True
    LastResultRow = .Range("S99999" ).End(xlUp).Row
    If LastResultRow < 3 Then Exit Sub
    If LastResultRow < 4 Then GoTo SkipSort
    'Sort By Tier
    With .Sort
      .SortFields.Clear
      .SortFields.Add Key:=ProdPricingDB.Range("S3" ), SortOn:=xlSortOnValues, Order
      :=xlAscending, DataOption:=xlSortNormal  'Sort
      .SetRange ProdPricingDB.Range("S3:X"  & LastResultRow) 'Set Range
      .Apply 'Apply Sort
    End With
    SkipSort:
    Products.Range("M19:Q"  & LastResultRow + 16).Value = .Range("T3:X"  &
    LastResultRow).Value 'Bring over Pricing Tier Details
  End With
End Sub

Sub ProdPriceTiers_SaveTiers()
  Dim LastTierRow As Long, TierRow As Long, TierDBRow As Long
  With Products
    If .Range("B3" ).Value = Empty Then
      MsgBox "Please save the product before adding any pricing tiers"
      Exit Sub
    End If
    If .Range("B13" ).Value = Empty Then
      MsgBox "Please select a Vendor before saving any pricing tiers"
      .Range("M17" ).Select
      Exit Sub
    End If
    LastTierRow = .Range("N36" ).End(xlUp).Row
    If LastTierRow < 19 Then Exit Sub 'Exit on no tier data
    For TierRow = 19 To LastTierRow
1 2 3
```

```
                If .Range("Q"  & TierRow).Value = Empty Then 'New Tier DB Row
                  TierDBRow = ProdPricingDB.Range("A99999" ).End(xlUp).Row + 1 'First Avail
                  Row
                  ProdPricingDB.Range("A"  & TierDBRow).Value = .Range("B2" ).Value 'Set
                  Product ID
                  ProdPricingDB.Range("C"  & TierDBRow).Value = .Range("B13" ).Value 'Set
                  Vendor ID
                  ProdPricingDB.Range("I"  & TierDBRow).Value = "=Row()"  'Set DB Row
                  .Range("Q"  & TierRow).Value = TierDBRow 'Set DB Row
                Else 'existing Tier DB Row
                  TierDBRow = .Range("Q"  & TierRow).Value
                End If
                ProdPricingDB.Range("B"  & TierDBRow).Value = .Range("J3" ).Value 'Product
                Name
                ProdPricingDB.Range("D"  & TierDBRow).Value = .Range("M17" ).Value 'Vendor
                Name
                ProdPricingDB.Range("E"  & TierDBRow).Value = TierRow - 18 'Set Tier #
                ProdPricingDB.Range("F"  & TierDBRow).Value = .Range("M"  & TierRow).Value
                'From #
                ProdPricingDB.Range("G"  & TierDBRow).Value = .Range("N"  & TierRow).Value
                'To #
                ProdPricingDB.Range("H"  & TierDBRow).Value = .Range("P"  & TierRow).Value
                'Price
              Next TierRow
            End With
            ProdPriceTiers_ListVendors 'Reload Product Price Tiers
        End Sub
```

```
Option Explicit
Dim LastRow As Long, LastResultRow As Long, ProdRow As Long, ProdCol As Long, SelRow As Long
Dim ProdFolder As String, PicPath As String
Sub Product_AddNew()
  Products.Range("B2,B14,J3,M3,J5,M5,J7,M7,B9,J9,M9,J11,M11,J13,M13,H19:J34,M19:Q34" ).ClearContents
  On Error Resume Next
  Products.Shapes("ProdPic" ).Delete  'Delete any picture if it exists
  On Error GoTo 0
  Products.Range("J3" ).Select
End Sub
Sub Product_ClearFilter()
  Products.Range("E3,G3" ).ClearContents
End Sub

Sub Product_Delete()
  With Products
    If MsgBox("Are you sure you want to delete this Product?" , vbYesNo, "Delete Product" ) = vbNo Then Exit Sub
    If .Range("B3" ).Value = Empty Then GoTo NotSaved
    ProdRow = .Range("B3" ).Value 'Products Row
    ProductsDB.Range(ProdRow & ":"  & ProdRow).EntireRow.Delete 'Delete DB Row
    NotSaved:
    Products_GetUniqueCategories 'Run Macro To Update Unique Categories
    Products_GetUniqueSubCategories  'Run Macro To Update Unique Sub-categories
    Product_AddNew 'Run macro to create new Products
    Product_ListProducts 'Update Products list
  End With
End Sub


Sub Product_ListProducts()
  Products.Range("C6:G99999" ).ClearContents 'Clear Existing Products
  With ProductsDB
    LastRow = .Range("A99999" ).End(xlUp).Row
    If LastRow < 4 Then Exit Sub
    .Range("A3:F"  & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range("Z2:Z3" ), CopyToRange:=.Range("AB2:AF2" ), Unique:=True
    LastResultRow = .Range("AB99999" ).End(xlUp).Row
    If LastResultRow < 3 Then Exit Sub
    If LastResultRow < 4 Then GoTo NoSort  'No sort on only 1 Product
    'Sort Alphabetally based on product name
    With .Sort
      .SortFields.Clear
      .SortFields.Add Key:=ProductsDB.Range("AC3" ), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal  'Sort
      .SetRange ProductsDB.Range("AB3:AF"  & LastResultRow) 'Set Range
      .Apply 'Apply Sort
    End With
    NoSort:
    Products.Range("C6:G"  & LastResultRow + 3).Value = .Range("AB3:AF"  & LastResultRow).Value 'Bring over data
  End With
End Sub

Sub Product_Load()
  With Products
    .Range("B14,J3,M3,J5,M5,J7,M7,B9,J9,M9,J11,M11,J13,M13,H19:J34,M19:Q34" ).ClearContents
```

```
55          If .Range("B3" ).Value = Empty Then
56            MsgBox "Please select a correct Product to load"
57            Exit Sub
58          End If
59          ProdRow = .Range("B3" ).Value  'Products Row
60          For ProdCol = 2 To 14
61            .Range(ProductsDB.Cells(1, ProdCol).Value).Value = ProductsDB.Cells(ProdRow,
              ProdCol).Value 'Bring over data
62          Next ProdCol
63        End With
64        Product_ShowPicture
65        ProdPriceTiers_ListVendors 'Refresh Product Financial Items
66      End Sub
67
68      Sub Product_SavedMsg()
69        With Products.Shapes("ProductSavedMsg" )
70          Dim i As Long, Delay As Double, StartTime As Double
71          .Visible = msoCTrue
72          For i = 1 To 75
73            .Fill.Transparency = i / 75
74            Delay = 0.005
75            StartTime = Timer
76            Do
77              DoEvents
78            Loop While Timer - StartTime < Delay
79          Next i
80          .Visible = msoFalse
81        End With
82      End Sub
83
84
85
86      Sub Product_SaveUpdate()
87        With Products
88          If .Range("j3" ).Value = Empty Then  'Empty Fields
89            MsgBox "Please make sure to enter a Product Name before saving"
90            Exit Sub
91          End If
92          If .Range("B3" ).Value = Empty Then 'New Product
93            ProdRow = ProductsDB.Range("A99999" ).End(xlUp).Row + 1        'First Avail
              Row
94            .Range("B2" ).Value = .Range("B4" ).Value 'New Product ID
95            ProductsDB.Range("A"  & ProdRow).Value = .Range("B2" ).Value 'Set Products ID
96          Else 'Existing Products
97            ProdRow = .Range("B3" ).Value 'Existing Products Row
98          End If
99          For ProdCol = 2 To 14
100           ProductsDB.Cells(ProdRow, ProdCol).Value = .Range(ProductsDB.Cells(1, ProdCol)
              .Value).Value 'Save/Update Data
101         Next ProdCol
102         Product_ListProducts 'Update Products list
103         Products_GetUniqueCategories 'Run Macro To List Unique Categories
104         Products_GetUniqueSubCategories  'Run Macro To List Unique Sub-categories
105         Product_SavedMsg
106       End With
107     End Sub
108
109     Sub Products_GetUniqueCategories()
110       With ProductsDB
111         LastRow = .Range("A99999" ).End(xlUp).Row
```

```
112         If LastRow < 4 Then Exit Sub
113         On Error Resume Next
114         .Names("Criteria" ).Delete
115         On Error GoTo 0
116         .Range("D3:D"  & LastRow).AdvancedFilter xlFilterCopy, , CopyToRange:=.Range(
            "T2" ), Unique:=True
117       End With
118     End Sub
119
120     Sub Products_GetUniqueSubCategories()
121       With ProductsDB
122         LastRow = .Range("A99999" ).End(xlUp).Row
123         If LastRow < 4 Then Exit Sub
124         .Range("D3:E"  & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
            "V2:V3" ), CopyToRange:=.Range("X2" ), Unique:=True
125       End With
126     End Sub
```

```
Option Explicit
Dim PORow As Long, POCol As Long, POItemRow As Long, LastItemRow As Long, ItemRow As Long
Dim LastRow As Long, ResultRow As Long, LastPOItemRow As Long, LastItemResultRow As Long, ItemDBRow As Long


Sub PO_AddNew()
  With PurchOrder
    .Range("E2:E5,H2:I2,F7:K37,M7:M37,N3:W4,N7:W37" ).ClearContents 'Clear PO Fields

    .Range("AA8:AT37" ).ClearContents 'Clear Unused Formulas
    .Range("B2" ).Value = .Range("B4" ).Value 'Next PO #
    .Range("E3" ).Value = Date 'PO Time
    .Range("E4" ).Value = Admin.Range("C19" ).Value  'Set Initial PO Status
  End With
End Sub

Sub PO_Delete()
  With PurchOrder
    If MsgBox("Are you sure you want to delete this Purchase Order?" , vbYesNo, "Delete Purchase Order" ) = vbNo Then Exit Sub
    If .Range("B3" ).Value = Empty Then GoTo NotSaved
    PORow = .Range("B3" ).Value 'Set Purchase Order Row
    PurchOrderDB.Range(PORow & ":"  & PORow).EntireRow.Delete

    'Delete PO Items
    With POItemsDB
      LastRow = .Range("A99999" ).End(xlUp).Row
      If LastRow < 4 Then Exit Sub
      .Range("A3:J"  & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range("O2:O3" ), CopyToRange:=POItemsDB.Range("T2:AA2" ), Unique:=False
      LastItemResultRow = POItemsDB.Range("T99999" ).End(xlUp).Row  'Last Results Row
      If LastItemResultRow < 3 Then GoTo NoItems

      For ResultRow = 3 To LastItemResultRow
        ItemDBRow = .Range("Z"  & ResultRow).Value 'Set Purchase Order Database Row
        .Range("A"  & ItemDBRow & ":J"  & ItemDBRow).ClearContents 'Clear Fields (deleting creates issues with results
      Next ResultRow
      'Resort DB to remove spaces
      With .Sort
        .SortFields.Clear
        .SortFields.Add Key:=POItemsDB.Range("A4" ), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal  'Sort
        .SetRange POItemsDB.Range("A4:J"  & LastRow) 'Set Range
        .Apply 'Apply Sort
      End With
    End With
    NoItems:
    NotSaved:
    PO_AddNew 'Add New Purchase Order
  End With
End Sub


Sub PO_Load()
  With PurchOrder
    If .Range("B3" ).Value = Empty Then
```

```
    1 2 3
53          MsgBox "Please enter a correct PO #"
54          Exit Sub
55        End If
56        Application.ScreenUpdating = False
57        PORow = .Range("B3" ).Value 'PO Row
58        .Range("E2:E5,H2:I2,F7:K37,M7:M37,N3:W4,N7:W37" ).ClearContents 'Clear PO Fields
59        .Range("B5" ).Value = True 'Set PO Load To True
60        For POCol = 2 To 5
61          .Range(PurchOrderDB.Cells(1, POCol).Value).Value = PurchOrderDB.Cells(PORow,
            POCol).Value 'Load PO Details
62        Next POCol
63      End With
64
65      With POItemsDB
66
67        'Load In PO Items
68        LastPOItemRow = .Range("A99999" ).End(xlUp).Row   'Last Item Row
69        If LastPOItemRow < 3 Then GoTo NoItems
70        .Range("A3:J"  & LastPOItemRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.
          Range("O2:O3" ), CopyToRange:=POItemsDB.Range("T2:AA2" ), Unique:=False
71        LastItemResultRow = POItemsDB.Range("T99999" ).End(xlUp).Row  'Last Results Row
72        If LastItemResultRow < 3 Then GoTo NoItems
73        For ResultRow = 3 To LastItemResultRow
74          POItemRow = POItemsDB.Range("AA"  & ResultRow).Value 'Purchase Order Item Row
75          PurchOrder.Range("F"  & POItemRow & ":K"  & POItemRow).Value = .Range("T"  &
            ResultRow & ":Y"  & ResultRow).Value 'Item Name, Desc, Qty, Price
76          PurchOrder.Range("M"  & POItemRow).Value = .Range("Z"  & ResultRow).Value
            'Item Database Row
77        Next ResultRow
78        'Update Pricing Formulas
79        If LastItemResultRow = 3 Then GoTo SingleItem 'Don't copy formula on single item
80        PurchOrder.Range("AA7:AT7" ).Copy
81        PurchOrder.Range("AA8:AT"  & LastItemResultRow + 4).PasteSpecial xlPasteFormulas
82        Application.CutCopyMode = False
83        SingleItem:
84        NoItems:
85        PurchOrder.Range("B5" ).Value = False 'Set PO Load To False
86        PurchOrder.Range("E2" ).Select
87        Application.ScreenUpdating = True
88      End With
89    End Sub
90
91    Sub PO_Print()
92      PurchOrder.PrintOut , , , , True, , , , False
93    End Sub
94
95
96
97    Sub PO_SaveUpdate()
98
99      With PurchOrder
100       'Determine if New or Existing PO
101       If .Range("B3" ).Value = ""  Then 'New PO
102         PORow = PurchOrderDB.Range("A99999" ).End(xlUp).Row + 1  'First Avail. Row
103         .Range("B2" ).Value = .Range("B4" ).Value 'Next PO #
104         PurchOrderDB.Range("A"  & PORow).Value = .Range("B2" ).Value 'PO #
105       Else ' Existing PO
106         PORow = .Range("B3" ).Value  'Purchase Order Row
    1 2 3
```

```
107         End If
108         For POCol = 2 To 6
109           PurchOrderDB.Cells(PORow, POCol).Value = .Range(PurchOrderDB.Cells(1, POCol).
              Value).Value 'Save PO Details
110         Next POCol
111          'Add/Update PO Item
112         LastItemRow = .Range("G37" ).End(xlUp).Row  'Last Item Row
113         If LastItemRow < 7 Then GoTo NoItems
114         For ItemRow = 7 To LastItemRow
115           If .Range("M"  & ItemRow).Value <> Empty Then 'Existing Row
116             POItemRow = .Range("M"  & ItemRow).Value 'Existing PO Item Row
117           Else 'New Row
118             POItemRow = POItemsDB.Range("A9999" ).End(xlUp).Row + 1  'First Avail Item
                Row
119             POItemsDB.Range("A"  & POItemRow).Value = .Range("B2" ).Value 'PO Number
120             POItemsDB.Range("I"  & POItemRow).Value = ItemRow 'PO Row
121             POItemsDB.Range("J"  & POItemRow).Value = "=Row()"  'Database Row
122             .Range("M"  & ItemRow).Value = POItemRow
123           End If
124           POItemsDB.Range("B"  & POItemRow & ":H"  & POItemRow).Value = .Range("F"  &
              ItemRow & ":L"  & ItemRow).Value 'Item, SKU, Qty, UOM, Price, Total
125         Next ItemRow
126         NoItems:
127       End With
128     End Sub
```

W

X

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