

بناء نظام لإدارة المخازن باستخدام برمجة الإكسل

م/أيمن السعيد عبدالواحد خشوعى

شركة ببنري كود سوفت ©

مدير عام

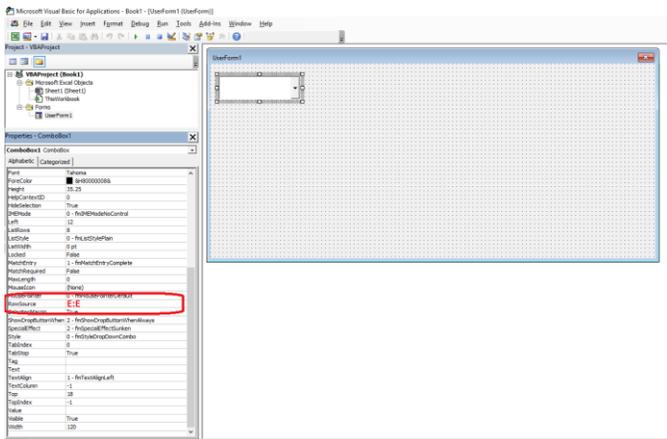
النزهة الجديدة ، القاهرة، جمهورية مصر العربية

التواصل: +٢٠١٠١٨٠٠٢٢٢٥

Admin@binarycodesoft.com

الفكرة الرئيسية

الفكرة الرئيسية للتطبيق هي تحويل ورقة عمل Excel إلى قاعدة بيانات للسماح بتعديل الحقول بسهولة بدلاً من استخدام ملفات قاعدة بيانات Access المكتملة. سنقوم بربط جميع حقول قاعدة البيانات باستخدام كلمة رئيسية فريدة (رقم الصف) والتي تحول علاقات الخلايا العادية في Excel إلى قاعدة بيانات ثابتة لقيم الأعمدة ذات الصف الواحد.



شكل ٢. التصميم الابتدائي للفورم الرئيسية

خطوات التصميم

انقر فوق علامة تبويب المطور وانقر فوق عرض الرمز ثم حدد إدراج (UserForm)

(أ) تغيير خاصية (الاسم) لنموذج المستخدم لتكون رئيسية.
(ب) تغيير خاصية (التسمية التوضيحية) للنموذج الرئيسي ليكون نظام إدارة المخزون Compny الخاص بي.

(ج) إضافة عنصر تحكم Combobox إلى النموذج الرئيسي.
(د) قم بتعيين خاصية (مصدر_الصف) في مربع التحرير والسرد لتكون العمود الذي يحتوي على القيم القابلة للتحديد.

(هـ) إضافة زر إلى ورقة العمل مع التسمية التوضيحية (إظهار).
(و) انقر نقرًا مزدوجًا فوق الزر لفتح نافذة التعليمات البرمجية وتعديلها

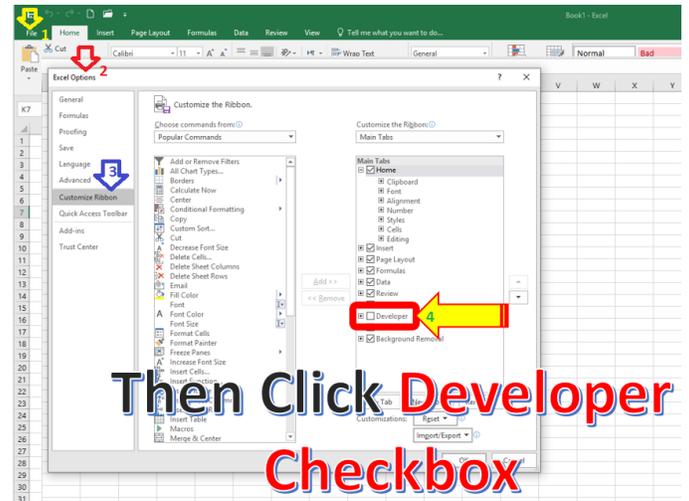
المُلخَص

نظام إدارة المخزون المستخدم للتعامل مع أصناف المخزون مثل الإصدار والإضافة والتحويل. هناك العديد من البرامج في السوق والتي تغطي الترخيص المجاني بعضًا منها. يمكنك الآن إنشاء برنامجك الخاص باستخدام هذه المقالة البسيطة بسهولة باستخدام بضعة أسطر من التعليمات البرمجية. حتى لو لم يكن لديك أي خبرة سابقة في الترميز.

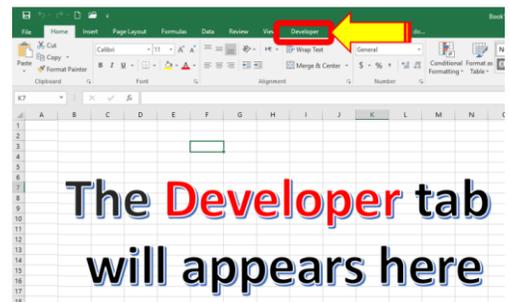
مقدمة

في البداية ، سنضيف علامة تبويب المطور إلى Microsoft Excel لتمكين إضافة رموز Visual Basic:

- 1- قائمة ملف.
- 2- اختيار الخيارات.
- 3- تخصيص Ribbon
- 4- خانة اختيار المطور.



شكل ١. إضافة علامة المطور إلى قائمة النوافذ في إكسل



```

Private Sub ComboBox1_Change()
Dim fvalue As Variant
Dim lRow As Long
Dim FindRow As Long
'Dim lCol As Long

'Find the last non-blank cell in column A(1)
lRow = Cells(Rows.Count, 1).End(xlUp).Row
On Error Resume Next

With Sheet1.Range("E:E")
Set fvalue = .Find(What:=ComboBox1.Value, _
LookIn:=xlValues, lookat:=xlWhole, MatchCase:=False)
FindRow = .Find(What:=ComboBox1.Value, _
LookIn:=xlValues, lookat:=xlWhole,
MatchCase:=False).Row
End With

If fvalue Is Nothing Then
MsgBox "This value does not exist"
'Me.cmdContact.value = "Cash Client"
Exit Sub
End If

With Me
.txtTotal.value = fvalue.Offset(, 2).value
.ItemNumber_TextBox.Text = fvalue.Offset(, 0).Value

.Description_En_txtBox.Text = fvalue.Offset(, -3).Value
.Description_Ar_txtBox.Text = fvalue.Offset(, -2).Value

.Size_txtBox.Text = fvalue.Offset(, -1).Value

.Qty_txtBox.Text = fvalue.Offset(, 1).Value

.Rack_txtBox.Text = fvalue.Offset(, 2).Value

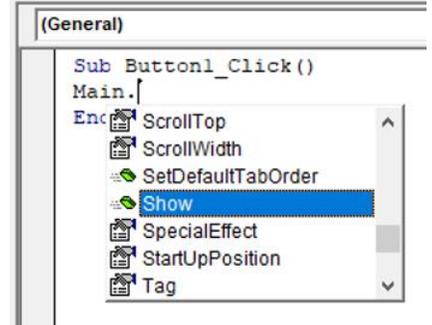
.Note_txtBox.Text = fvalue.Offset(, 3).Value

.row_number = FindRow

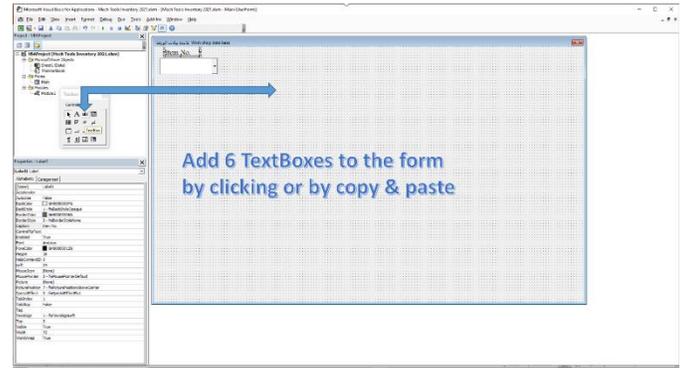
End With

End Sub
    
```

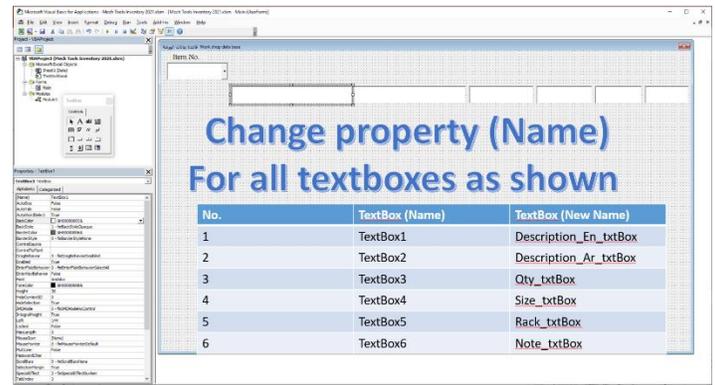
أكتب الآتى فى السطر المحدد وسيتم تكملة الكود بعد إضافة النقطة وحرف (S)
أختر أمر Show لعرض الفورم (النموذج) الرئيسى



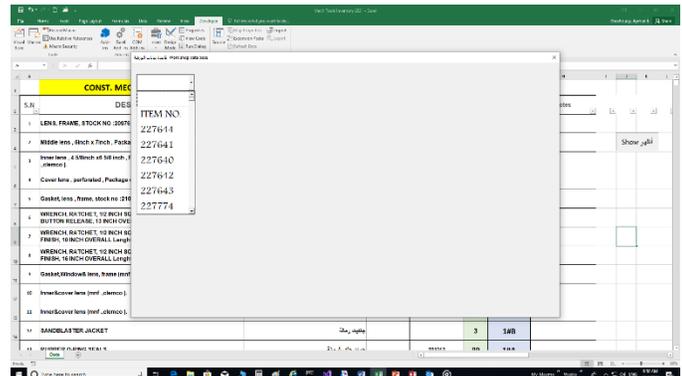
أضف الآن عنصر تحكم تسمية إلى النموذج وقم بتسميته row_number مع التسمية التوضيحية هي row:
الخطوة التالية هي إضافة 6 عناصر تحكم جديدة في مربع نص (بناءً على عدد الحقول التي تريد إظهارها في النموذج الرئيسى)



إضافة 6 عناصر TextBox



أضبط خصائص وأسماء العناصر الستة كما بالجدول



يمكنك وضع زر للإضافة وآخر للخصم من الكمية الموجودة بالمخزن كما هو موضح بالكود التالي كود الخصم(Issue)

```
Private Sub Issue_btn_Click()
Dim i As Integer
Dim IssueMyValue As Integer

Dim ws1 As Worksheet

Set ws1 = ThisWorkbook.Worksheets("Data")
IssueMyValue = InputBox("Issue this value from Qty", "Qty Control", 1)
If ws1.Range("F" & row_number.Caption).Value = 0 Then
i = MsgBox("Item is Nill", vbOKOnly, "Qty Control")
Exit Sub
ElseIf IssueMyValue > ws1.Range("F" & row_number.Caption).Value Then
i = MsgBox("Qty is too much" & vbNewLine & "Only " & ws1.Range("F" & _
row_number.Caption).Value & " Exist", vbOKOnly, "Qty Control")
Exit Sub
ElseIf IssueMyValue = ws1.Range("F" & row_number.Caption).Value Then
i = MsgBox("Qty will be (Nill)" & vbNewLine & "Qty --> 0 " & _
vbNewLine & " Continu???", vbOKCancel, "Qty Control")
If i = vbOK Then
ws1.Range("F" & row_number.Caption).Value = Range("F" & _
row_number.Caption).Value - IssueMyValue
Me.Qty_txtBox.Text = Range("F" & row_number.Caption).Value
Exit Sub
Else
Exit Sub
End If
Else
ws1.Range("F" & row_number.Caption).Value = Range("F" & _
row_number.Caption).Value + IssueMyValue
Me.Qty_txtBox.Text = Range("F" & row_number.Caption).Value
End If
End Sub
```

المصادر

- Excel VBA Macro Programming by Richard Shepherd [1]
- VBA and Macros for Microsoft Excel (Business Solutions) by Bill Jelen and Tracy Syrstad [2]
- Office VBA Macros You Can Use Today by Juan Pablo Gonzalez, Cindy Meister, Suat Ozgur, Bill Dilworth. [3]

تلخيص

لتطوير Microsoft Excel هذه الورقة هي عينة من كيفية استخدام تطبيقات كاملة باستخدام عدد قليل من أسطر التعليمات البرمجية. يمكنك العثور على مزيد من التفاصيل حول هذه الأساليب في موقع الويب (<https://www.binarycodesoft.com>) روابط لمشاريعي المطورة المنشورة

<https://www.udemy.com/user/ayman-khoshouey-2/>

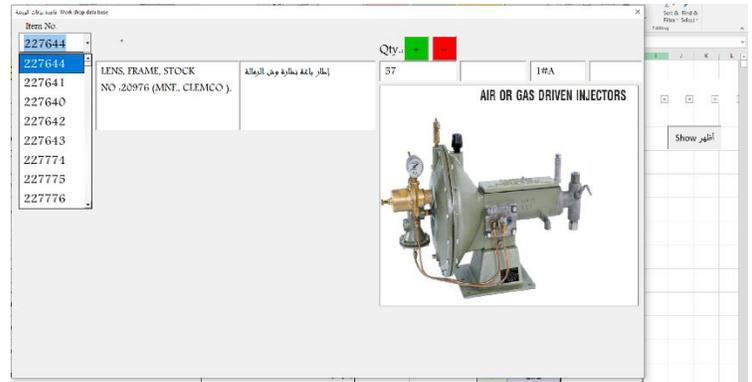
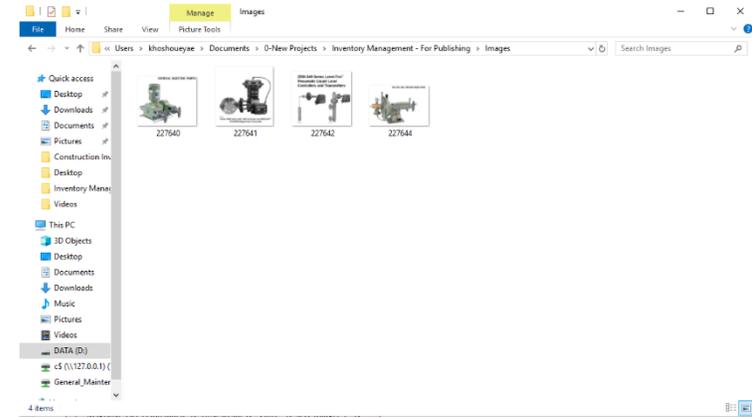
الكود الخاص بعرض صورة العنصر عند اختياره من القائمة

```
Dim Path As String
Path = Application.ActiveWorkbook.Path & "\Images"

Me.Image1.Picture = LoadPicture(Path & "\" & _
Me.ComboBox1.Text & ".jpg")
```

بخصوص وضع الصور في مجلد وتسمية كل صورة برقم العنصر الخاص بها. انظر الصورة أدناه

شكل ٤. وضع صور العناصر في مجلد واحد يسمى (Images) وكل صورة باسم رقم العنصر



شكل ٤. البرنامج عند اختيار عنصر وعرض الصورة الخاصة به

كود الإضافة (Add)

```
Private Sub Add_btn_Click()
Dim AddMyValue As Variant
Dim ws As Worksheet

Set ws = ThisWorkbook.Worksheets("Data")

AddMyValue = InputBox("Add this value to Qty", "Qty Control", 1)

ws.Range("F" & row_number.Caption).Value = Range("F" & _
row_number.Caption).Value + AddMyValue

Me.Qty_txtBox.Text = Range("F" & row_number.Caption).Value
End Sub
```