

## IE 5390 – Assignment - 04

Name: Simran Abhay Sinha

### Lab 07

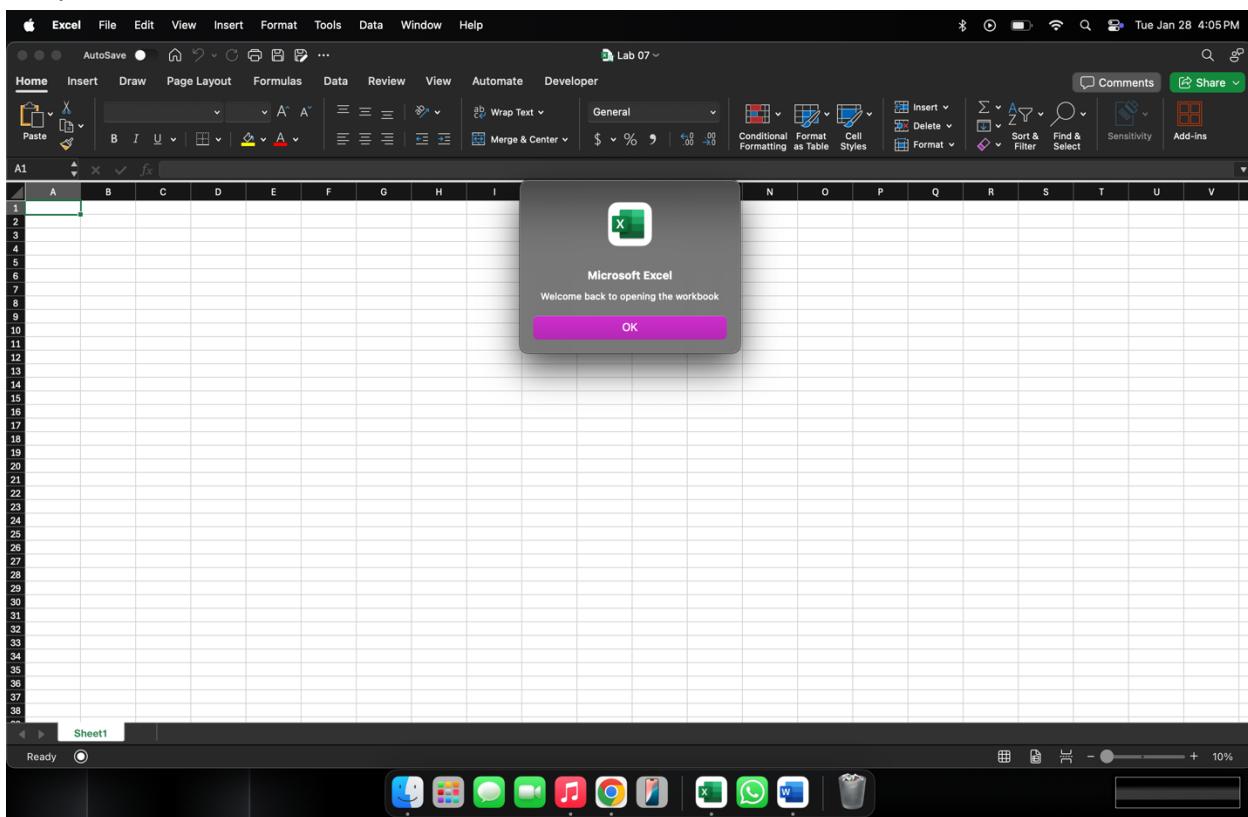
#### 1. Using Sheet1

##### A. Display a welcome message to user upon opening a workbook

Code:

```
Private Sub Workbook_Open()
'Lab 07 P1A
    MsgBox ("Welcome back to opening the workbook")
End Sub
```

Output:

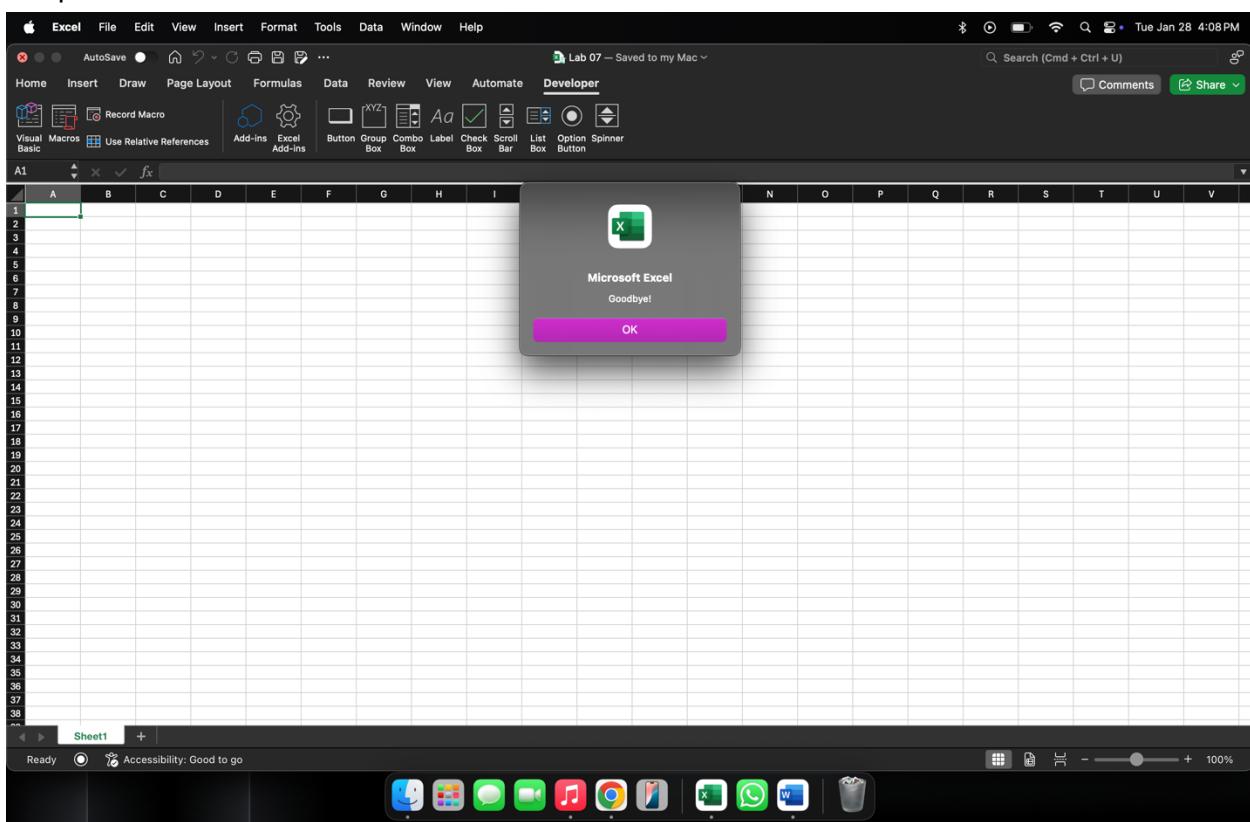


B. Display a reminder message to user before closing a workbook

Code:

```
WORKBOOK
Private Sub Workbook_BeforeClose(Cancel As Boolean)
'Lab 07 P1B
    MsgBox ("Goodbye!")
End Sub
```

Output:

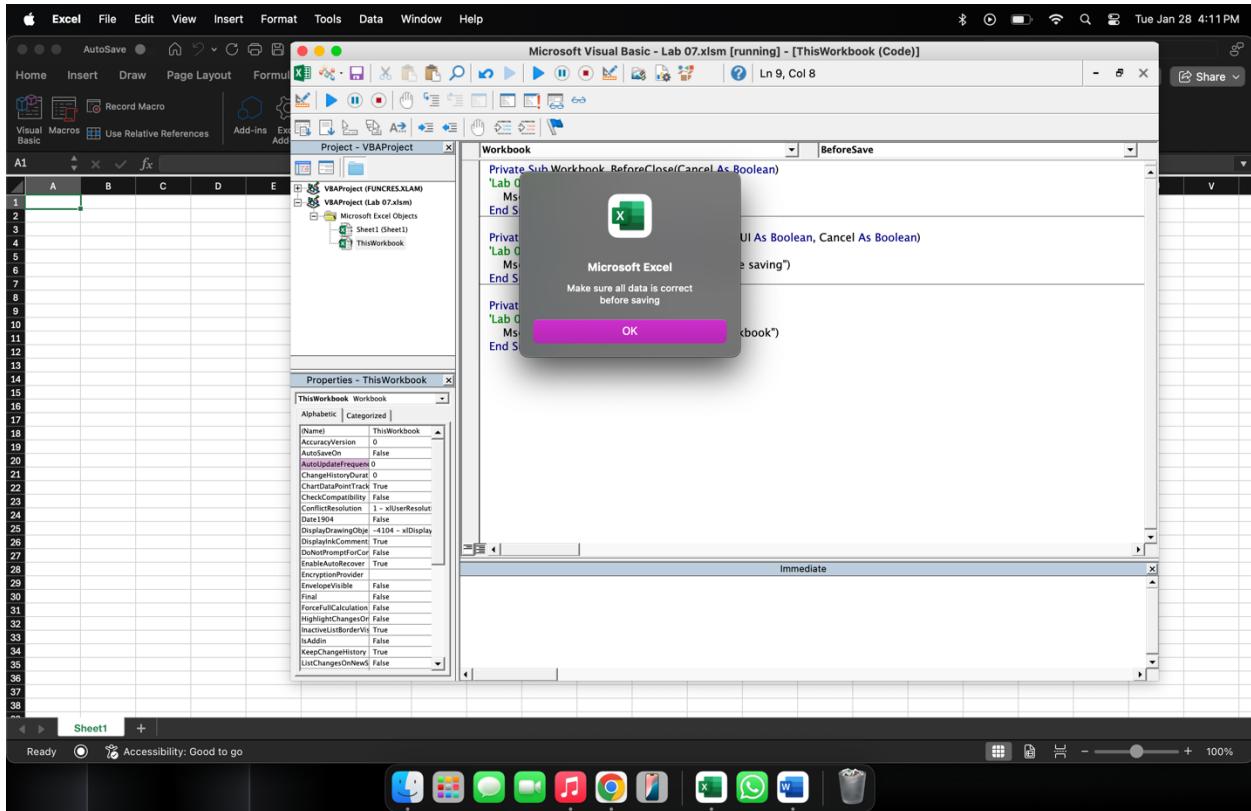


## C. Display a reminder message to user before saving a workbook

Code:

```
Private Sub Workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean)
'Lab 07 P1C
    MsgBox ("Make sure all data is correct before saving")
End Sub
```

Output:

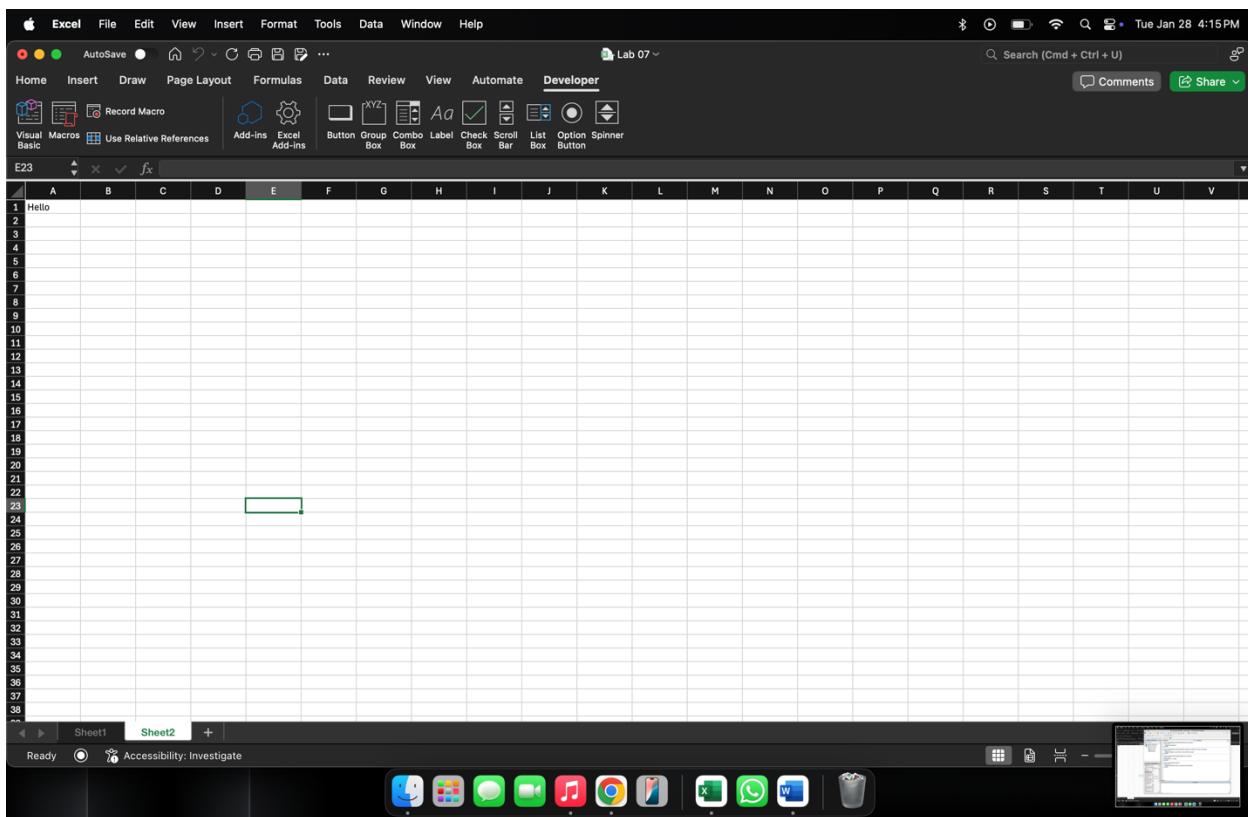


D. Add new sheet to workbook and insert “Hello” in cell A1

Code:

```
End Sub  
  
Private Sub Workbook_NewSheet(ByVal Sh As Object)  
'Lab 07 P1D  
Sh.Range("A1") = "Hello"  
End Sub
```

Output:



## 2. Using Sheet1

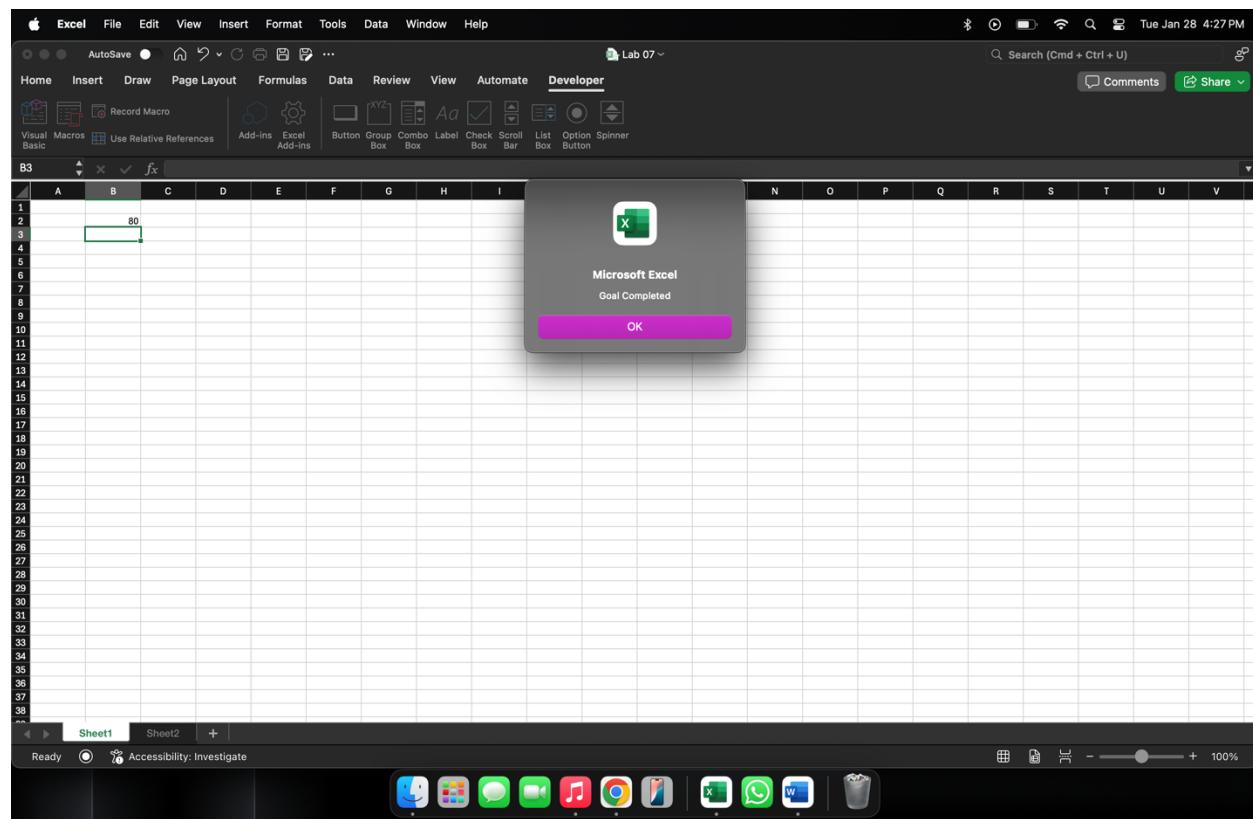
A. Write VBA code to check cell B2 value and display a message according to cell value:  $\geq 80$  prints “Task Completed”;  $< 80$  prints “Task Incomplete”

Code:

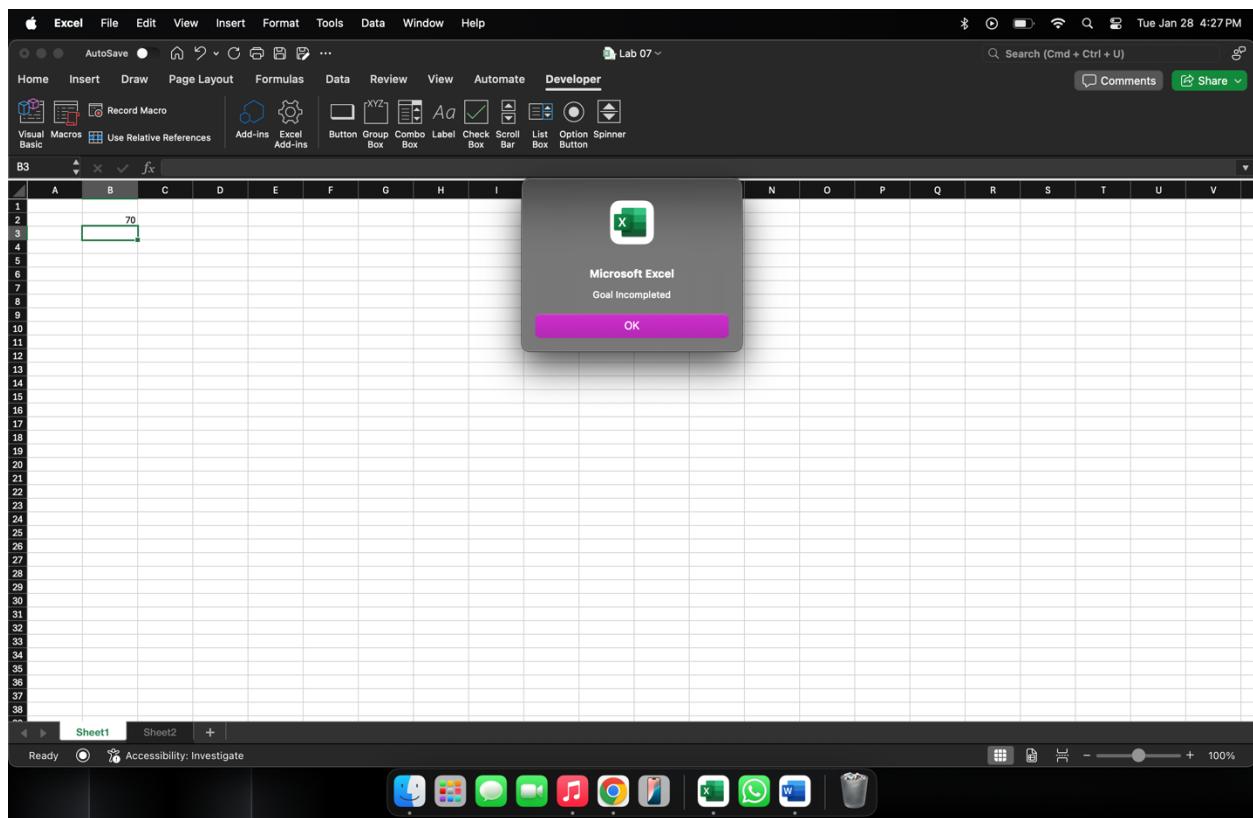
```
Worksheet
```

```
Private Sub Worksheet_Change(ByVal Target As Range)
'Lab 07 P2
If Target.Address = "$B$2" Then
    If Target.Value >= 80 Then
        MsgBox ("Goal Completed")
    End If
    If Target.Value < 80 Then
        MsgBox ("Goal Incompleted")
    End If
End If
End Sub
```

Output:



## IE5390 Structured Data Analytics



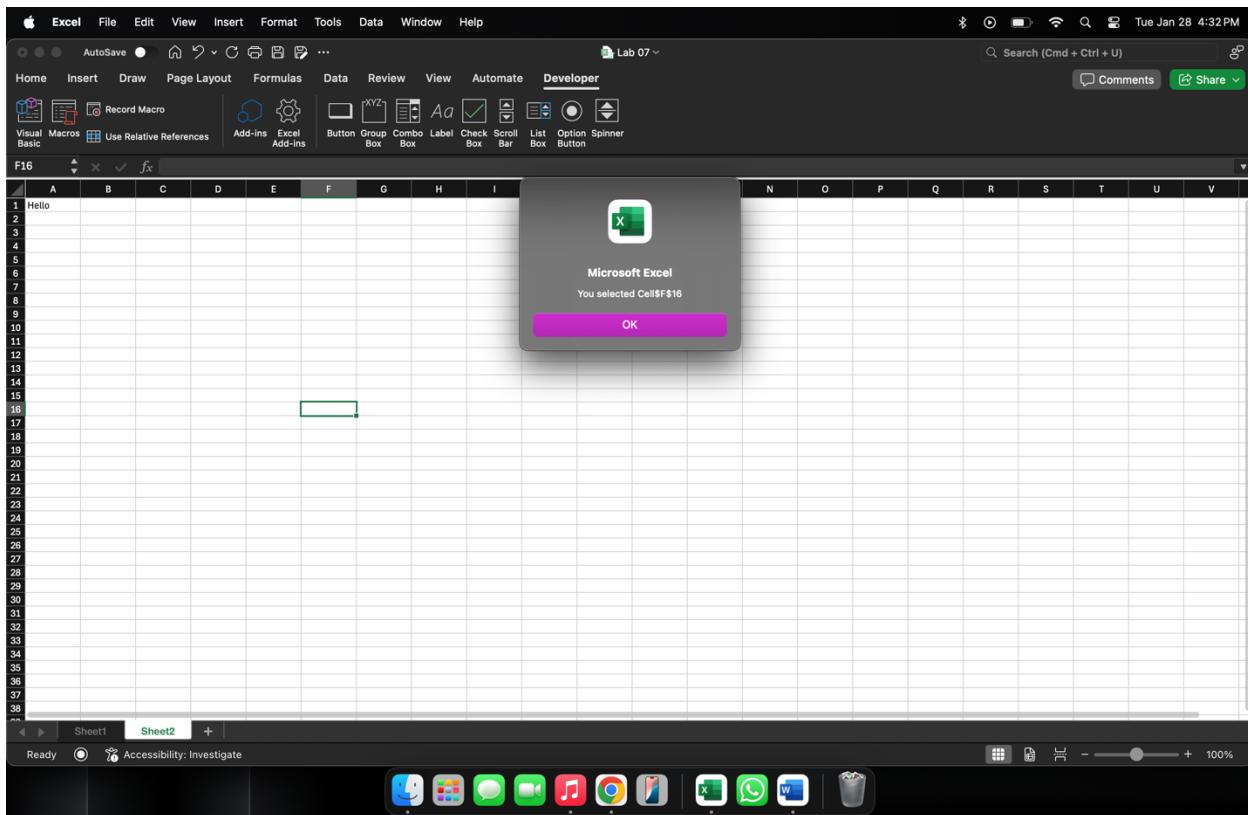
### 3. Using Sheet2

A. Display the address of a cell that the user selects, i.e. click it in Sheet2

Code:

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
'Lab 07 P3
    cell_addr = Target.Address
    MsgBox ("You selected Cell" & cell_addr)
End Sub
```

Output:



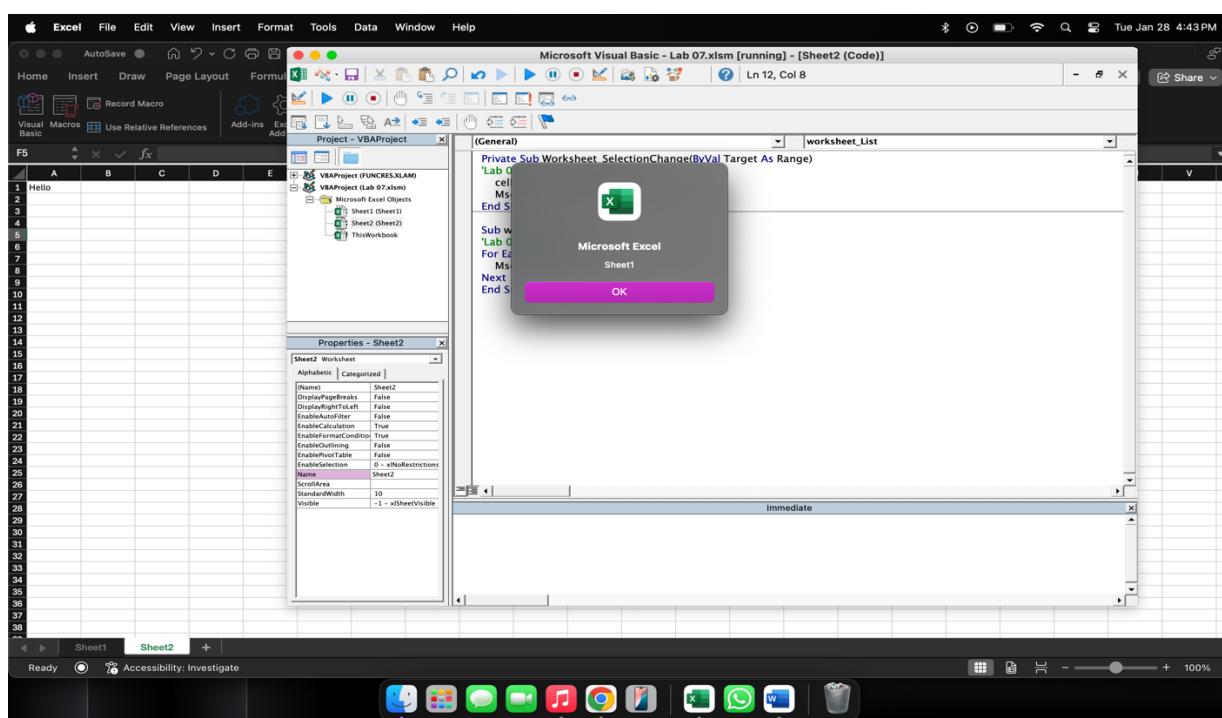
## 4. Using Sheet2, Do the following

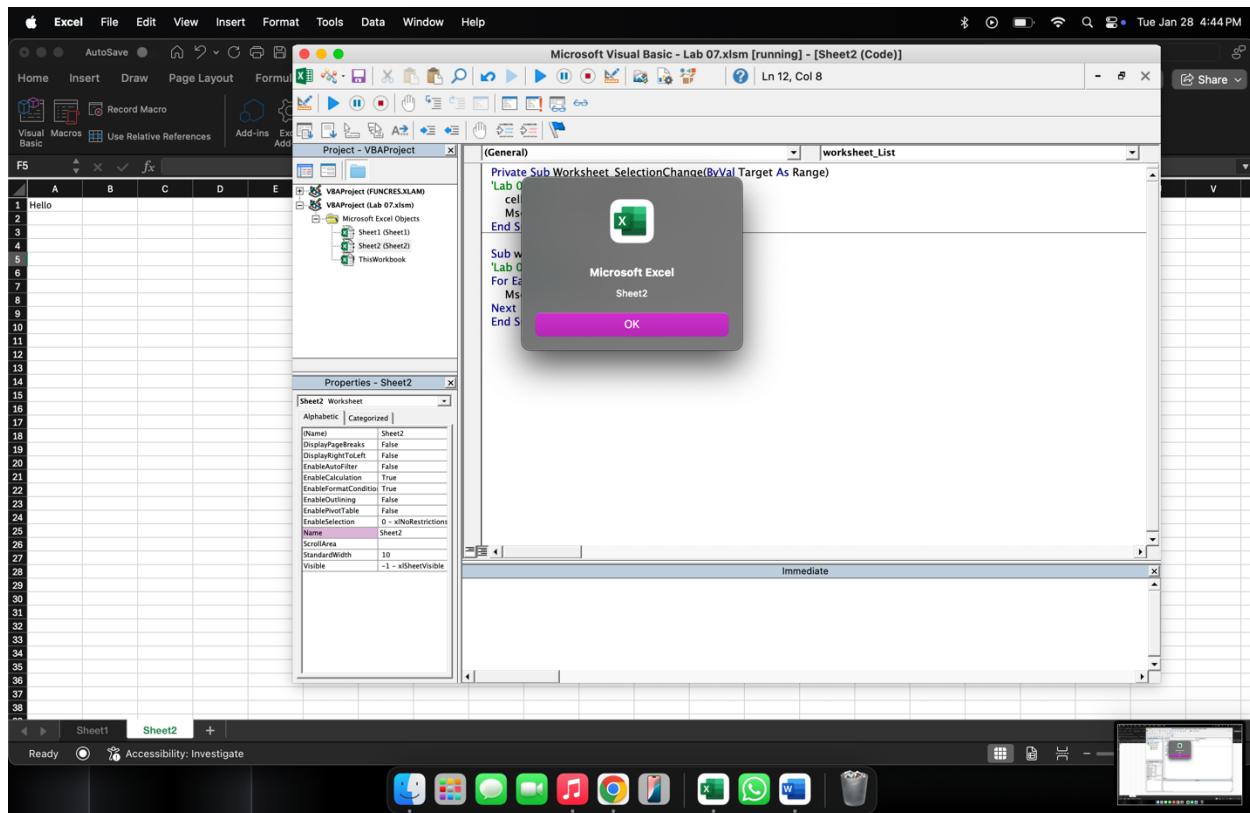
## A. List all worksheets in the active workbook

Code:

```
Sub worksheet_List()
'Lab 07 P4A
For Each ws In ActiveWorkbook.Worksheets
    MsgBox (ws.Name)
Next
End Sub
```

Output:



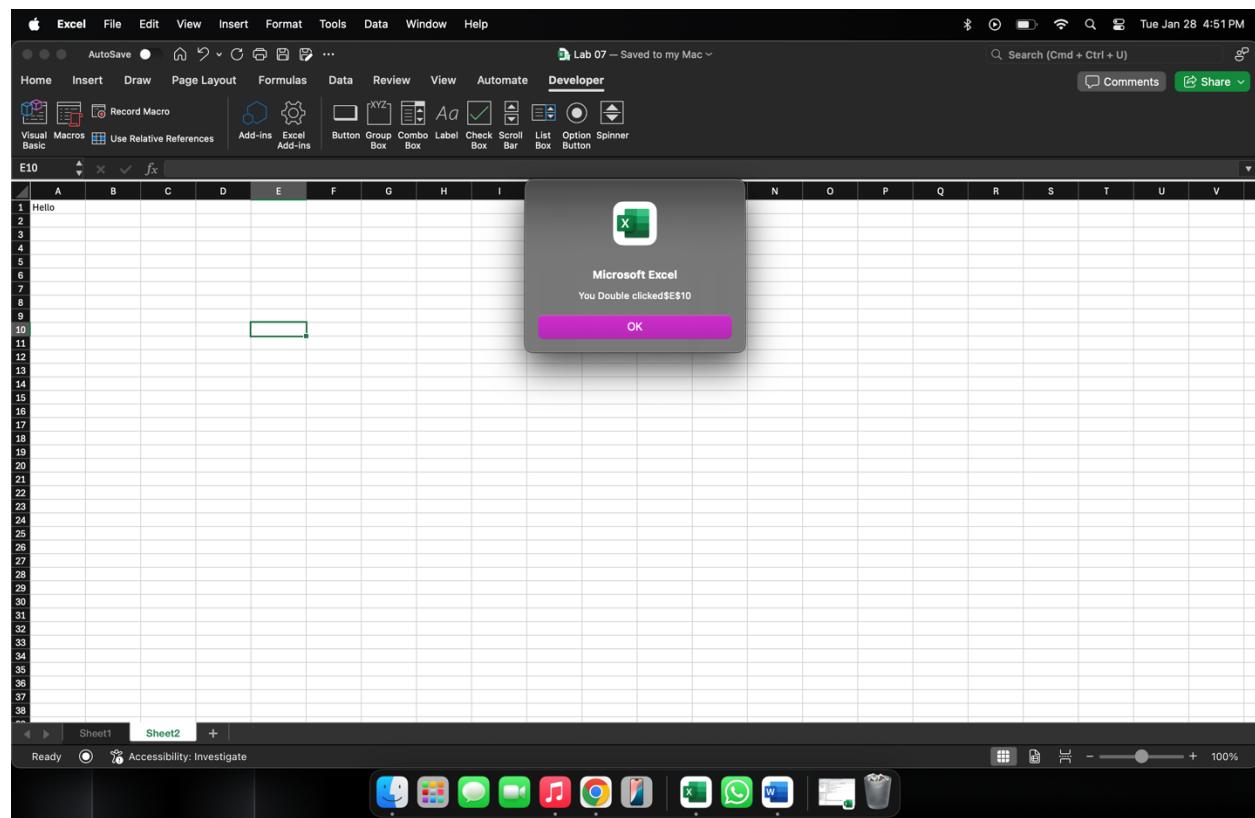


B. Show cell address when you double click it

Code:

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
'Lab 07 P4B
cell_addr = Target.Address
MsgBox ("You Double clicked" & cell_addr)
End Sub
```

Output:

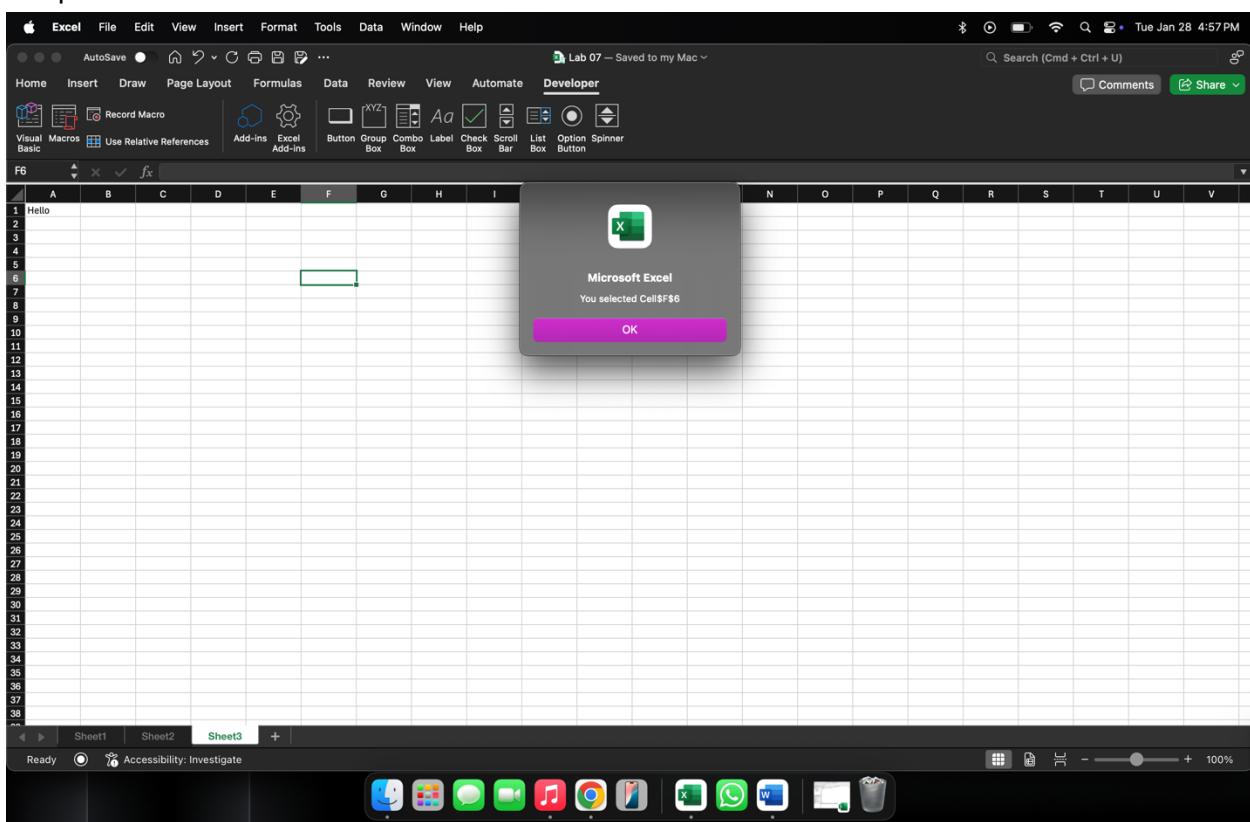


5. Using Sheet3, use the selection change event in an example

Code:

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
'Lab 07 P5
MsgBox ("You selected Cell" & Target.Address)
End Sub
```

Output:



## 6. Using Sheet3, Use the calculate event in an example

Code:

```
Worksheet
```

```

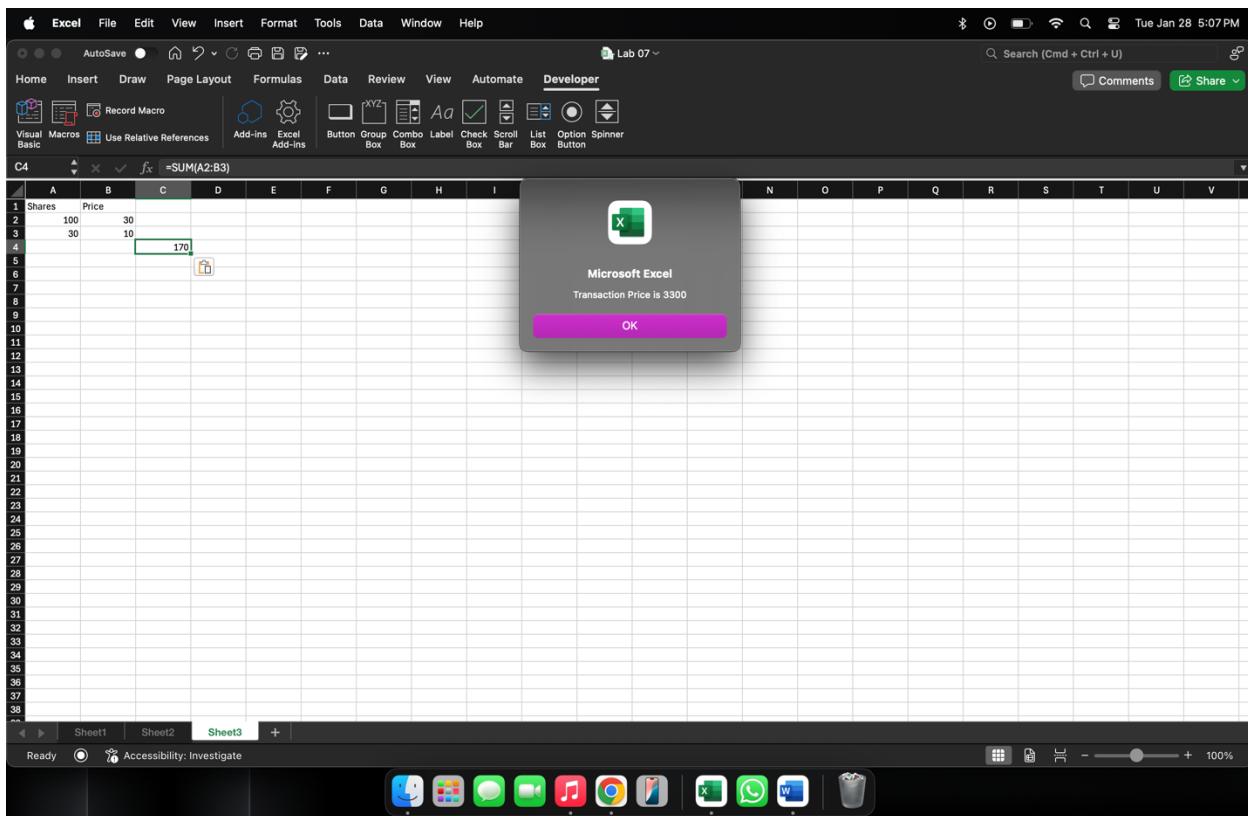
Private Sub Worksheet_Calculate()
'Lab 07 P6

'Initialize total value
totalValue = 0

'loop to find total price of transaction
For i = 1 To 2
    totalValue = totalValue + Cells(i + 1, 1) * Cells(i + 1, 2)
Next i
MsgBox ("Transaction Price is " & totalValue)
End Sub

```

Output:



7. Write VBA code that counts how many times a link has been clicked (followed) in a worksheet

Code:

```
ok x
Private Sub Worksheet_FollowHyperlink(ByVal Target As Hyperlink)
'Lab 7 Problem 7
'Get the clicked cell
Dim clickedCell As Range
Set clickedCell = Target.Range

' Check if "Click Count" column exists in column B
If Range("B1").Value <> "Click Count" Then
    Range("B1").Value = "Click Count"
End If

' Get the row of clicked link
Dim clickRow As Long
clickRow = clickedCell.Row

' Increment click count
If Range("B" & clickRow).Value = "" Then
    Range("B" & clickRow).Value = 1
Else
    Range("B" & clickRow).Value = Range("B" & clickRow).Value + 1
End If
End Sub
```

Output:

A	B	C
1 Shares	Clickcount	
2 100	30	
3 30	10	
4	170	
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18	<a href="#">Northeastern University Website</a>	
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		

## 8. Write VBA code to delete active worksheet

Code:

```

Sub DeleteActiveSheet()
'Lab 07 P8
Dim activeCellAddress As String

' Store the address of the active cell before deleting the sheet
activeCellAddress = ActiveCell.Address

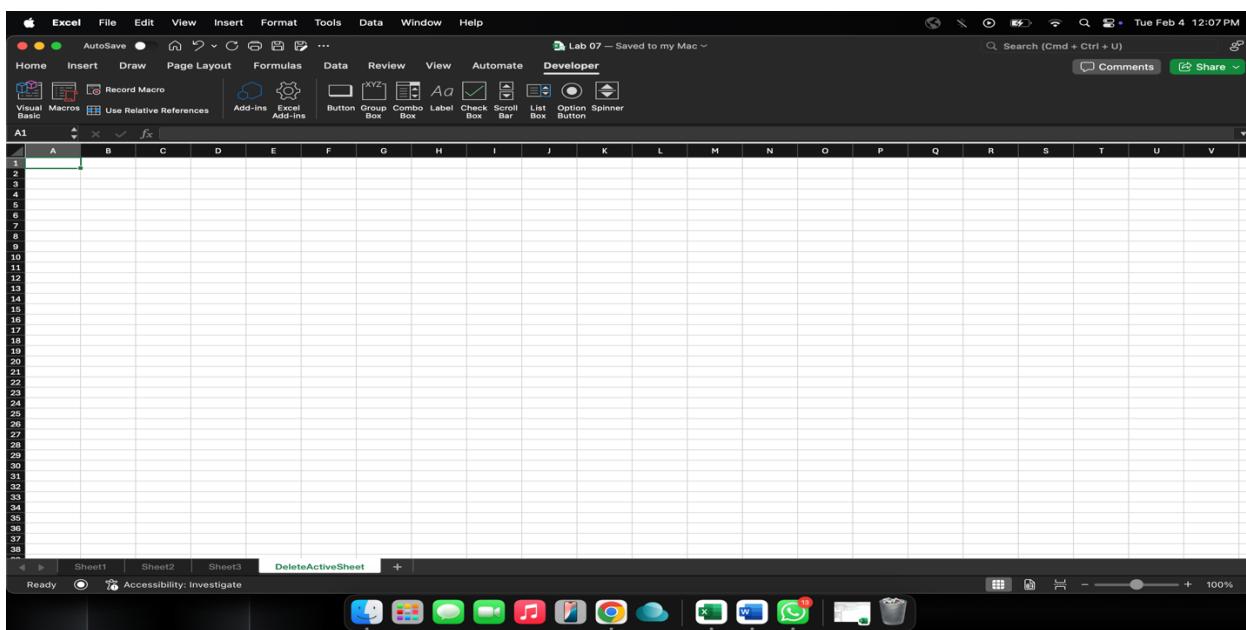
Application.DisplayAlerts = False ' Disable confirmation prompt
On Error Resume Next ' Handle potential errors

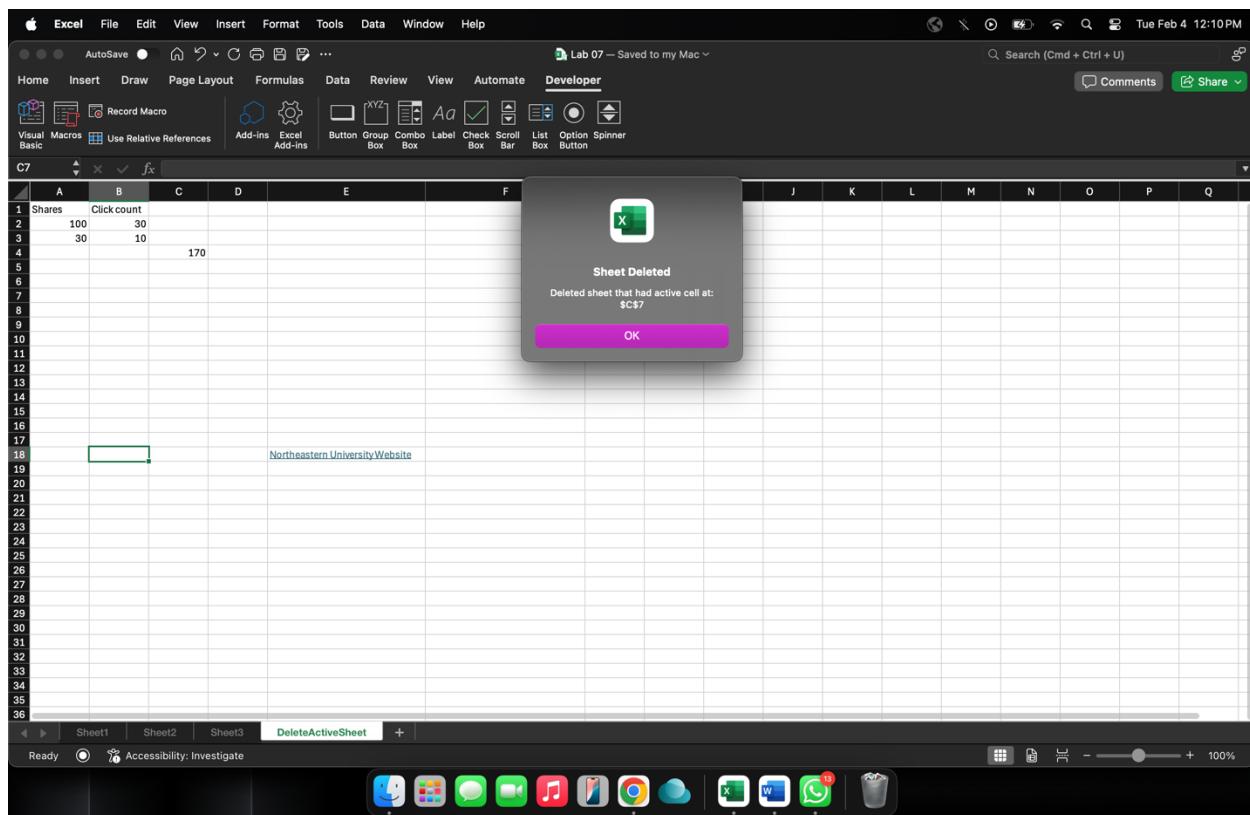
If ActiveSheet.Index > 1 Then ' Ensure at least one sheet remains
    ActiveSheet.Delete
    MsgBox "Deleted sheet that had active cell at: " & activeCellAddress, vbInformation, "Sheet Deleted"
Else
    MsgBox "Cannot delete the last remaining sheet.", vbExclamation, "Error"
End If

Application.DisplayAlerts = True ' Re-enable alerts
End Sub

```

## New Sheet: DeleteActiveSheet



**Output:**

Excel sheet and macro deleted

## 9. Gen AI Prompt Engineering

Using the Sample\_Sales\_Data.xlsx file, use VBA events to track changes and deletions in an Audit Log.

### I am using ChatGPT:

Prompt:

**File** Sample\_Sales\_Data.xlsx

You are tasked with creating an Audit Log in Excel to monitor changes in the workbook. This file contains:

- ▀ Region1 and Region2 sheets with sales data (date, product, and sales amount).
- ▀ An Audit Log sheet, initially empty, to track changes and deletions. The audit log must meet the following requirements:

1. Log Changes: Whenever a cell's value is changed in any sheet, record the following in an "Audit Log" sheet:
  - o Sheet Name
  - o Cell Address
  - o Old Value
  - o New Value
  - o Timestamp

2. Log Row Deletions: If any row is deleted in any sheet, record the following  
in the "Audit Log" sheet:  
o Sheet Name  
o Deleted Row Number(s)  
o Timestamp  
3. Exclude the Audit Log Sheet: The "Audit Log" sheet itself must not trigger any logging.  
You will use 3 workbook-level events: `workbook_SheetChange`, `workbook_sheetBeforeDeletion`, and `workbook_SheetChange`

GenAI:



I'll create a VBA script to track changes and deletions in the workbook as per your requirements. The script will use `Workbook_SheetChange` to log value changes and `Workbook_SheetChange` combined with `Workbook_SheetBeforeDelete` to track row deletions. I'll update the workbook with this functionality.

Let me know if you want any modifications or additional features.

Here's the VBA code you need to implement in the `ThisWorkbook` module of your Excel file. This will track changes and deletions while ensuring the "Audit Log" sheet itself is not monitored.

### Steps to Implement:

1. Open your Excel file (`Sample_Sales_Data.xlsxm`).
2. Press `ALT + F11` to open the **VBA Editor**.
3. Double-click **ThisWorkbook** under **Microsoft Excel Objects**.
4. Copy and paste the following code into the editor.
5. Save the workbook as a macro-enabled file (`.xslm`).

**Step 1: Store Old Value (Fix Selection Issue)**

```
vba
Dim OldValue As String ' Store the old value before a change occurs

Private Sub Workbook_SheetSelectionChange(ByVal Sh As Object, ByVal Target As Range)
    ' Prevent logging for the Audit Log sheet
    If Sh.Name = "Audit Log" Then Exit Sub

    ' Store old value as a string to prevent TypeMismatch errors
    If Not IsEmpty(Target.Value) Then
        On Error Resume Next
        OldValue = CStr(Target.Value)
        On Error GoTo 0
    Else
        OldValue = "[Blank]"
    End If
End Sub
```

**Step 2: Log Changes & Row Deletions Correctly**

```
vba
Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As Range)
    Dim wsAudit As Worksheet
    Set wsAudit = ThisWorkbook.Sheets("Audit Log")

    ' Prevent logging changes in the Audit Log sheet itself
    If Sh.Name = "Audit Log" Then Exit Sub

    Dim LogRow As Range
    Set LogRow = wsAudit.Cells(wsAudit.Rows.Count, 1).End(xlUp).Offset(1, 0)

    ' Log value changes
    LogRow.Value = Sh.Name ' Sheet Name
    LogRow.Offset(0, 1).Value = Target.Address ' Cell Address
    LogRow.Offset(0, 2).Value = OldValue ' Old Value
    LogRow.Offset(0, 3).Value = Target.Value ' New Value
    LogRow.Offset(0, 4).Value = Now ' Timestamp
End Sub
```

**Step 3: Log Actual Row Deletions ( Workbook\_SheetBeforeDelete )**

```
vba
Private Sub Workbook_SheetBeforeDelete(ByVal Sh As Object)
    Dim wsAudit As Worksheet
    Set wsAudit = ThisWorkbook.Sheets("Audit Log")

    ' Prevent logging the deletion of the Audit Log sheet itself
    If Sh.Name = "Audit Log" Then Exit Sub

    Dim LogRow As Range
    Set LogRow = wsAudit.Cells(wsAudit.Rows.Count, 1).End(xlUp).Offset(1, 0)

    ' Log row deletion
    LogRow.Value = Sh.Name ' Sheet Name
    LogRow.Offset(0, 1).Value = "Entire Sheet Deleted"
    LogRow.Offset(0, 2).Value = "[N/A]" ' Old Value
    LogRow.Offset(0, 3).Value = "Deleted Sheet"
    LogRow.Offset(0, 4).Value = Now ' Timestamp
End Sub
```

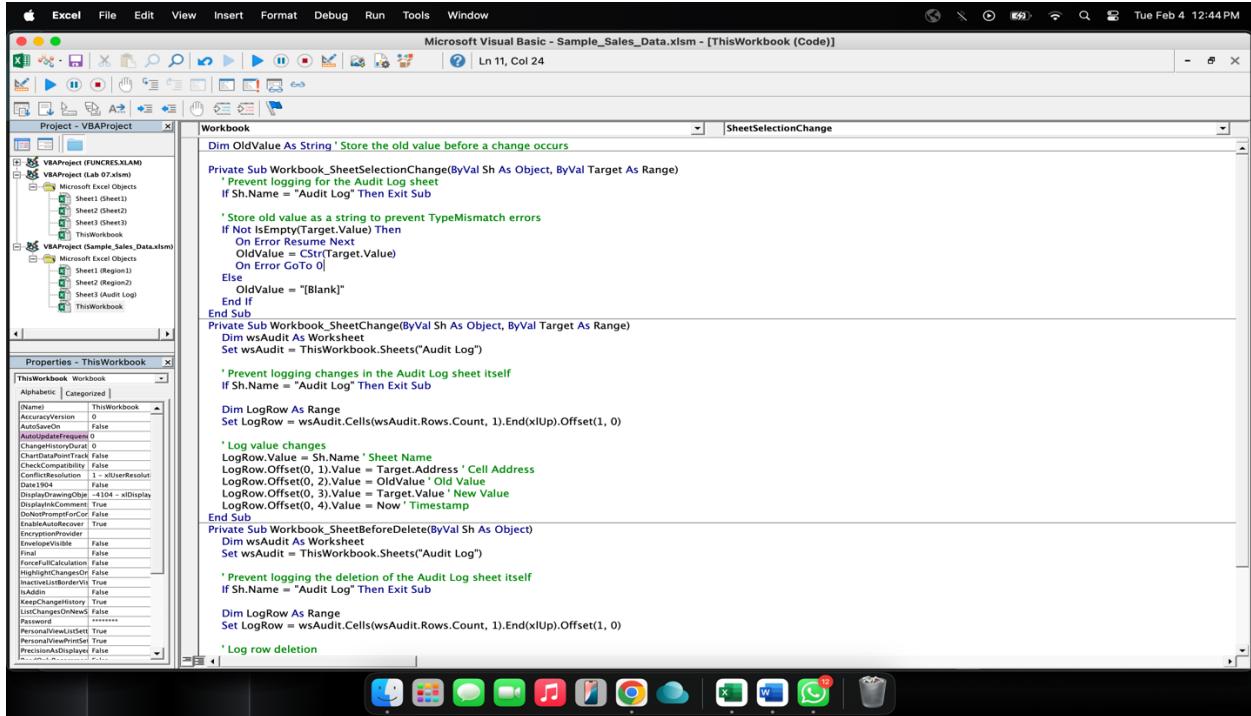


### Final Steps to Apply Fixes

1. **Open VBA Editor ( ALT + F11 )**
2. **Go to ThisWorkbook Module**
3. **Delete your old code and paste the new code.**
4. **Save as .xlsm**
5. **Test the fixes:**
  - Change a value → Check **Audit Log**.
  - Delete a row → Check **Audit Log**.
  - Delete a sheet → Check **Audit Log**.

## Excel Implementation

Code:



```

Dim OldValue As String ' Store the old value before a change occurs
Private Sub Workbook_SheetSelectionChange(ByVal Sh As Object, ByVal Target As Range)
    ' Prevent logging for the Audit Log sheet
    If Sh.Name = "Audit Log" Then Exit Sub

    ' Store old value as a string to prevent TypeMismatch errors
    If Not IsEmpty(Target.Value) Then
        On Error Resume Next
        OldValue = CStr(Target.Value)
        On Error GoTo 0
    Else
        OldValue = "[Blank]"
    End If
End Sub

Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As Range)
    Dim wsAudit As Worksheet
    Set wsAudit = ThisWorkbook.Sheets("Audit Log")

    ' Prevent logging changes in the Audit Log sheet itself
    If Sh.Name = "Audit Log" Then Exit Sub

    Dim LogRow As Range
    Set LogRow = wsAudit.Cells(wsAudit.Rows.Count, 1).End(xlUp).Offset(1, 0)

    ' Log value changes
    LogRow.Value = Sh.Name ' Sheet Name
    LogRow.Offset(0, 1).Value = Target.Address ' Cell Address
    LogRow.Offset(0, 2).Value = OldValue ' Old Value
    LogRow.Offset(0, 3).Value = Target.Value ' New Value
    LogRow.Offset(0, 4).Value = Now ' Timestamp
End Sub

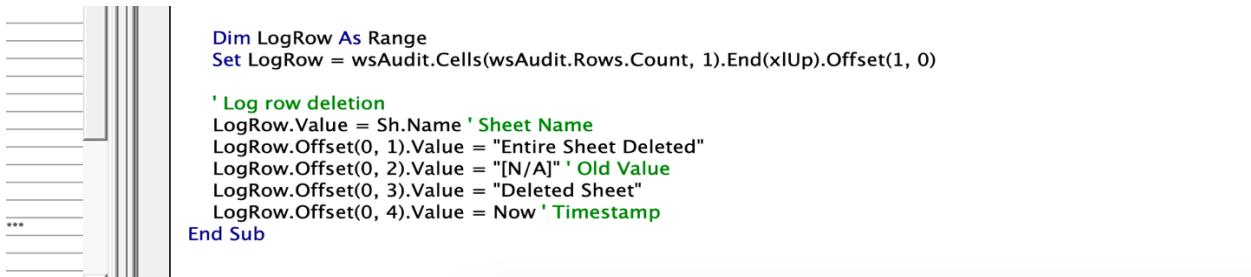
Private Sub Workbook_SheetBeforeDelete(ByVal Sh As Object)
    Dim wsAudit As Worksheet
    Set wsAudit = ThisWorkbook.Sheets("Audit Log")

    ' Prevent logging the deletion of the Audit Log sheet itself
    If Sh.Name = "Audit Log" Then Exit Sub

    Dim LogRow As Range
    Set LogRow = wsAudit.Cells(wsAudit.Rows.Count, 1).End(xlUp).Offset(1, 0)

    ' Log row deletion
    LogRow.Value = Sh.Name ' Sheet Name
    LogRow.Offset(0, 1).Value = "Entire Sheet Deleted"
    LogRow.Offset(0, 2).Value = "[N/A]" ' Old Value
    LogRow.Offset(0, 3).Value = "Deleted Sheet"
    LogRow.Offset(0, 4).Value = Now ' Timestamp
End Sub

```



```

Dim LogRow As Range
Set LogRow = wsAudit.Cells(wsAudit.Rows.Count, 1).End(xlUp).Offset(1, 0)

' Log row deletion
LogRow.Value = Sh.Name ' Sheet Name
LogRow.Offset(0, 1).Value = "Entire Sheet Deleted"
LogRow.Offset(0, 2).Value = "[N/A]" ' Old Value
LogRow.Offset(0, 3).Value = "Deleted Sheet"
LogRow.Offset(0, 4).Value = Now ' Timestamp
End Sub

```

**Output:**

The screenshot shows a Microsoft Excel window titled "Sample\_Sales\_Data". The ribbon is visible at the top with tabs like Home, Insert, Draw, Page Layout, Formulas, Data, Review, View, Automate, and Developer. The Developer tab is selected. The status bar at the bottom right shows "Tue Feb 4 12:41 PM". The main area displays an audit log table with the following data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Sheet Name	Cell Address	Old Value	New Value	Timestamp										
2	Region2	\$C\$3	300	400	2/4/25 12:40										
3	Region2	\$C\$6	280	200	2/4/25 12:40										
4	Region2	\$B\$6	Gadget Y	Gadget YA	2/4/25 12:41										
5	Region1	\$A\$5:\$C\$5	Gadget Z	1/5/25	2/4/25 12:41										
6	Region1	\$B\$10	Widget A	Widget M	2/4/25 12:41										
7	Region1	\$C\$10	130	30	2/4/25 12:41										
8	Region1	\$A\$3	1/2/25	5/20/25	2/4/25 12:41										
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															
21															
22															
23															
24															
25															
26															

The ribbon tabs shown are Home, Insert, Draw, Page Layout, Formulas, Data, Review, View, Automate, and Developer. The Developer tab is selected. The status bar at the bottom right shows "Ready" and "Accessibility: Good to go". The zoom level is set to 150%.

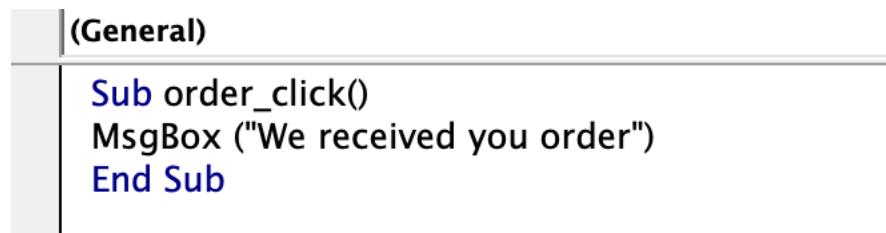
## Lab 08

### 1. Using Sheet1

#### A. Create VBA controls using two methods

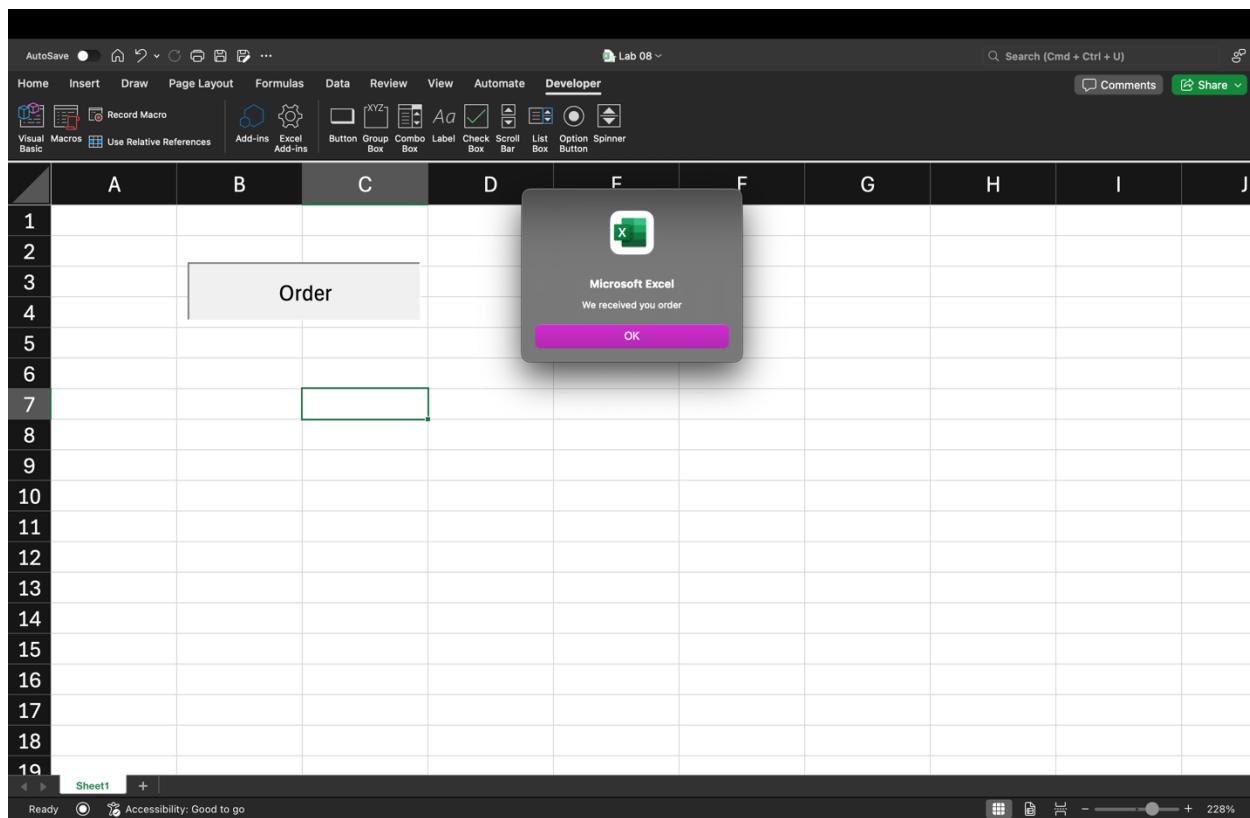
##### i. Embedded button control

Code:



Output:

The screenshot shows a Microsoft Excel spreadsheet titled 'Lab 08'. The ribbon is visible with the 'Developer' tab selected. The worksheet contains a single button labeled 'Order' located at cell B3. The button has a green border and is highlighted with a green selection box. The rest of the spreadsheet is empty with a light gray background.



ii. Controls of User form. Create these specific controls: button, textbox, radio button, checkbox, spinner control (button)

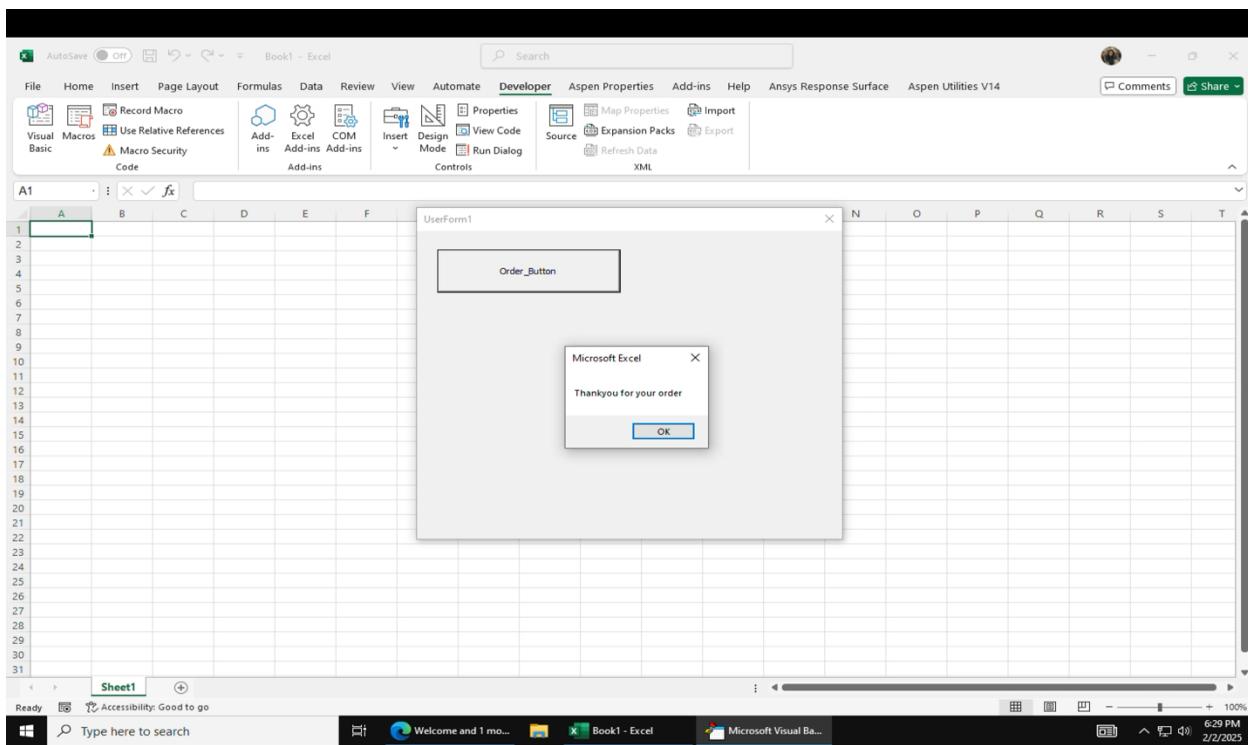
**Button:**

Code:

```

Private Sub Order_button_Click()
'Lab 09 PA-1
MsgBox ("We received your order")
End Sub

```

**Output:****TextBox:**

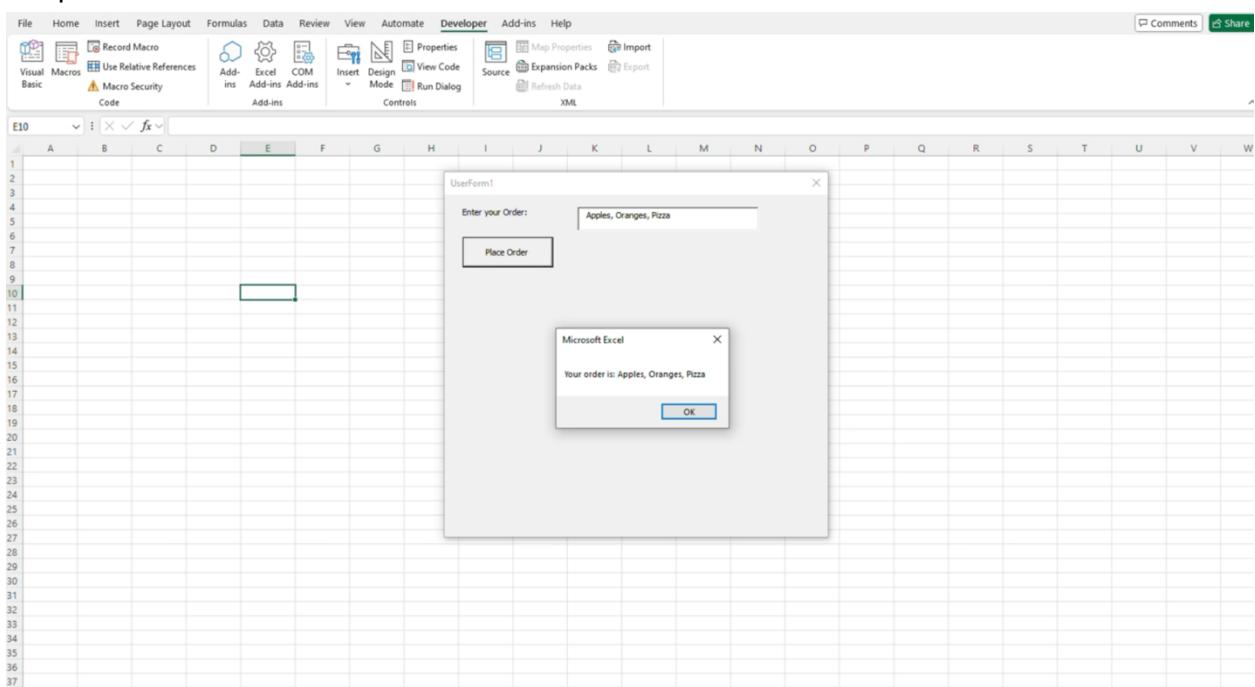
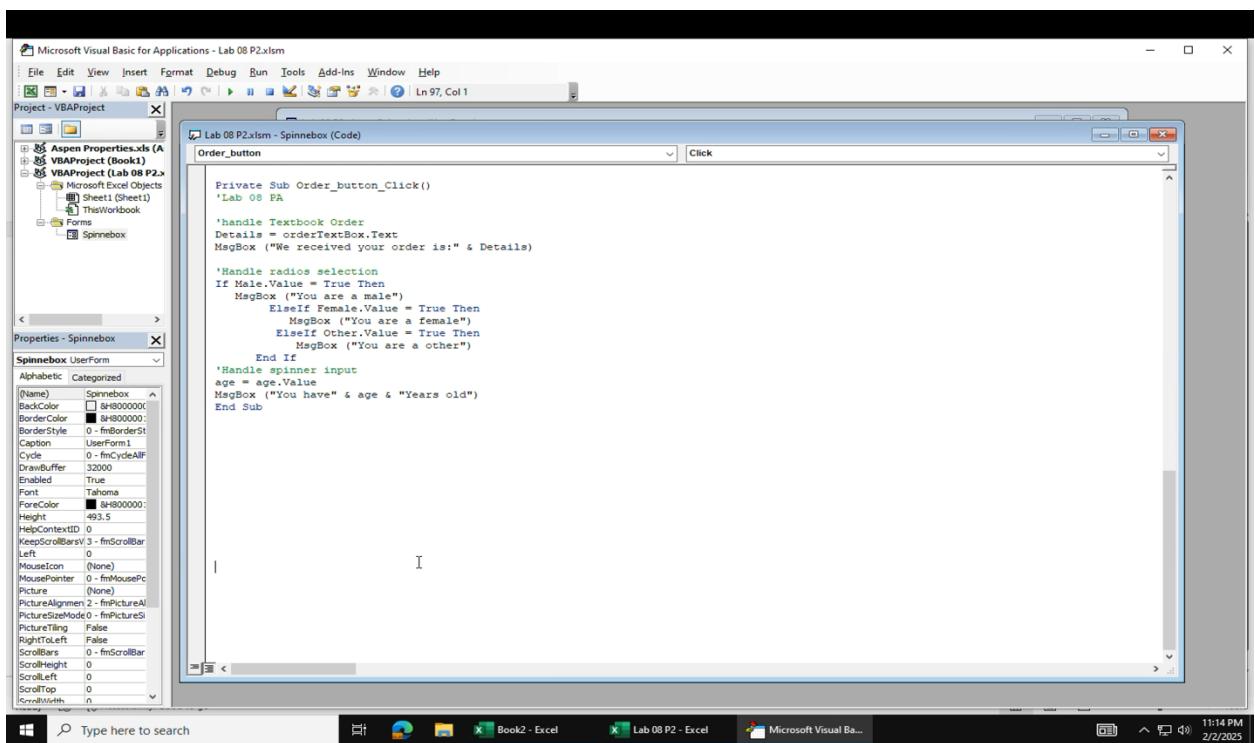
Code:

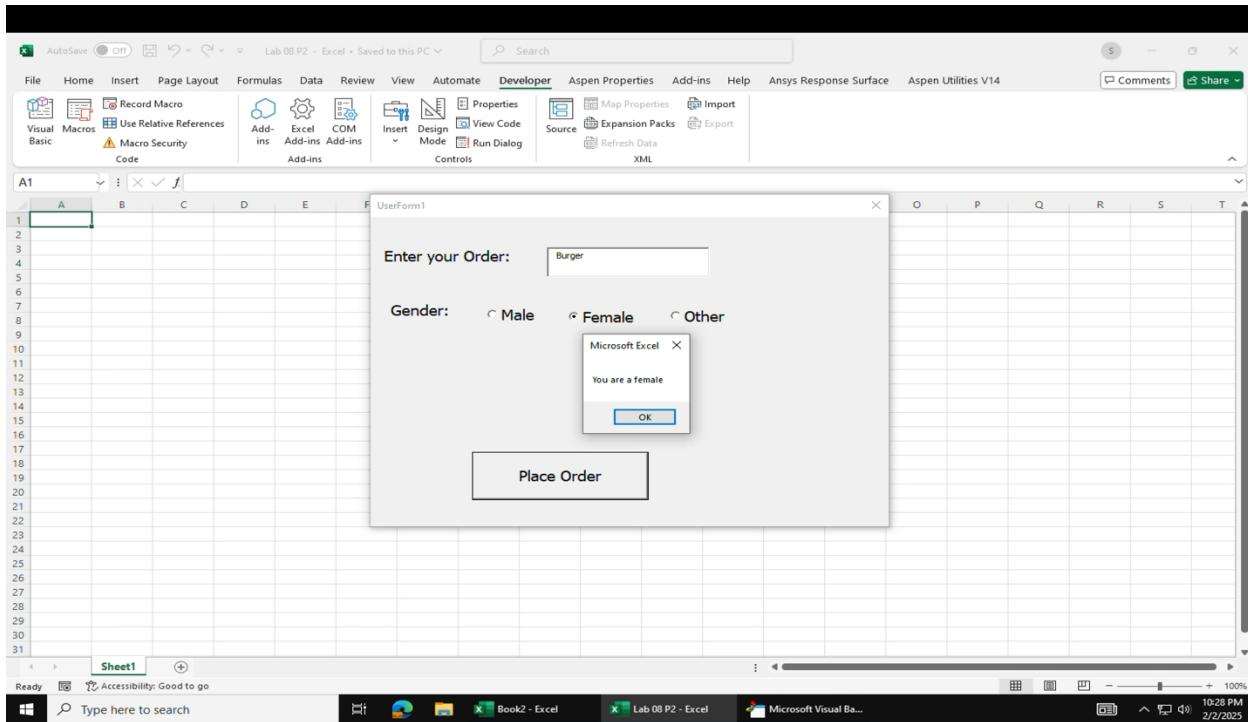
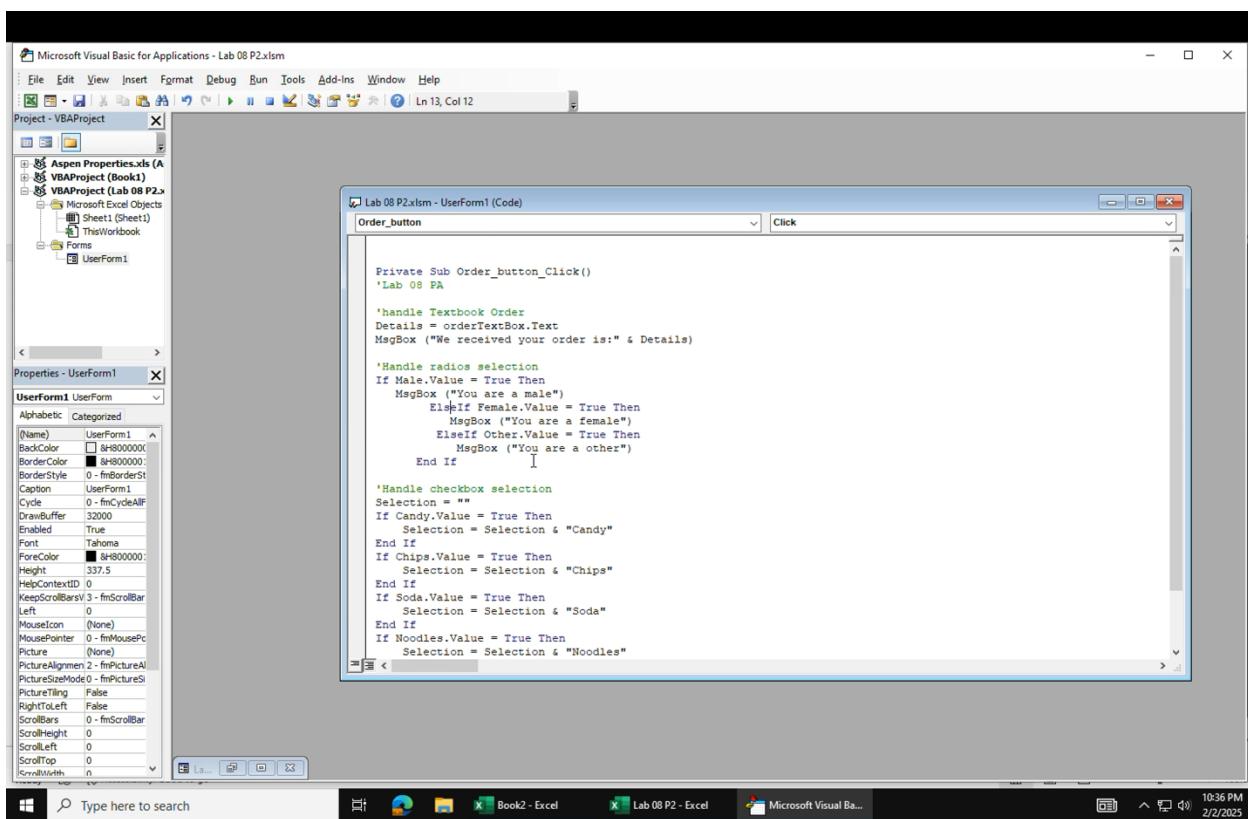
```
Private Sub Order_button_Click()
    'Lab 08 PA

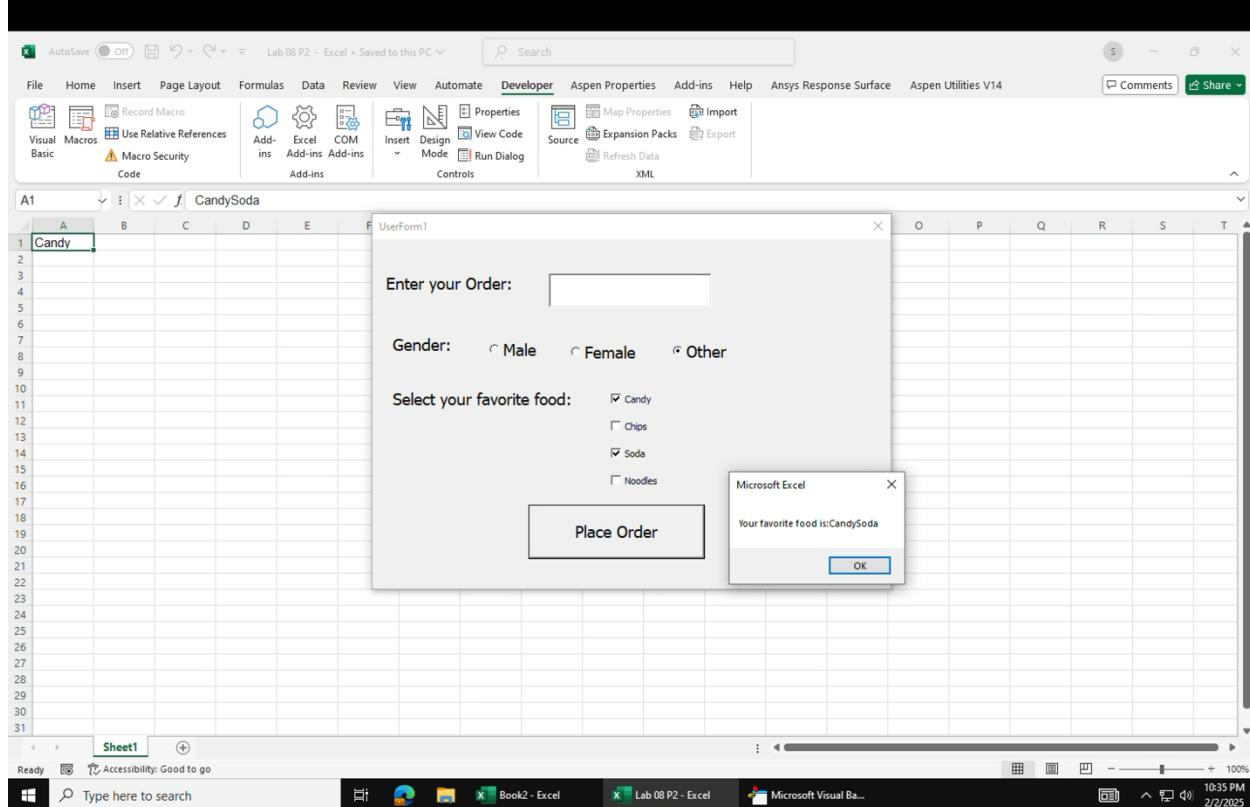
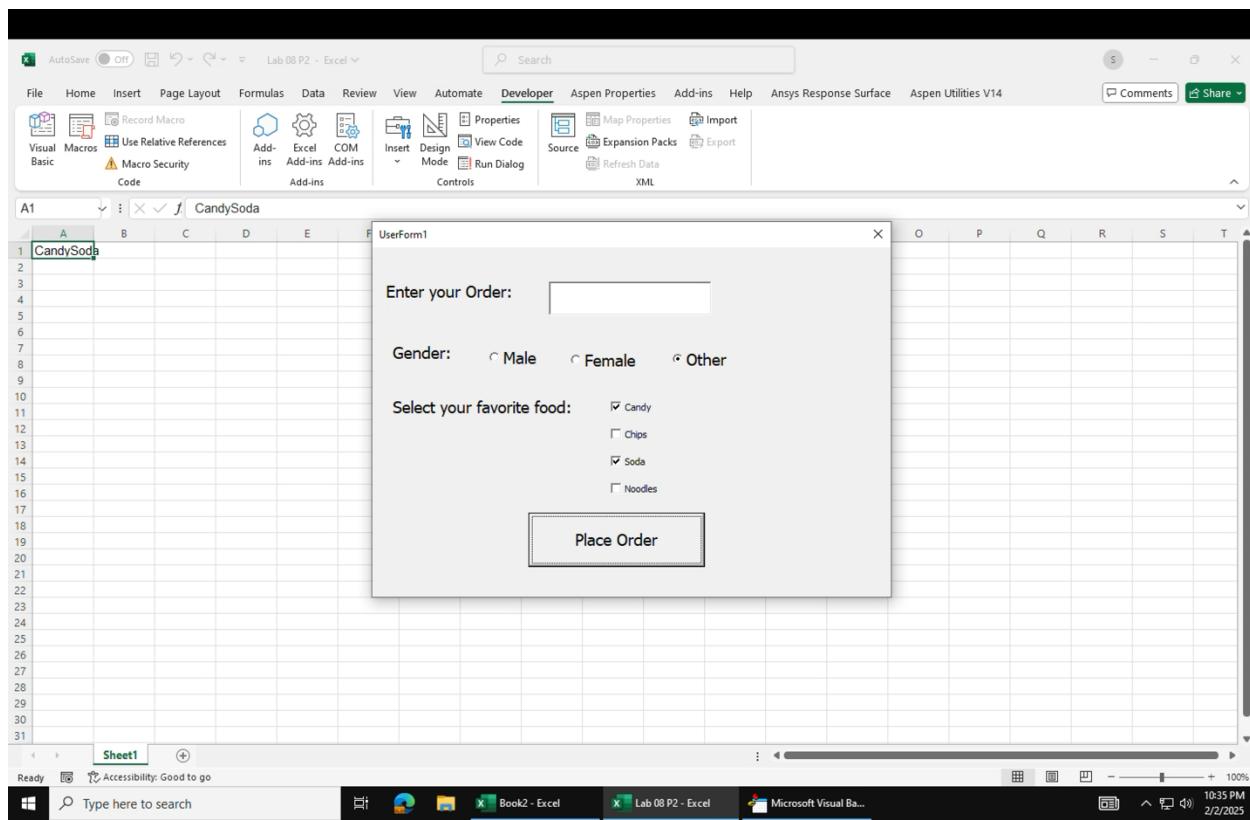
    'handle Textbook Order
    Details = orderTextBox.Text
    MsgBox ("We received your order is:" & Details)

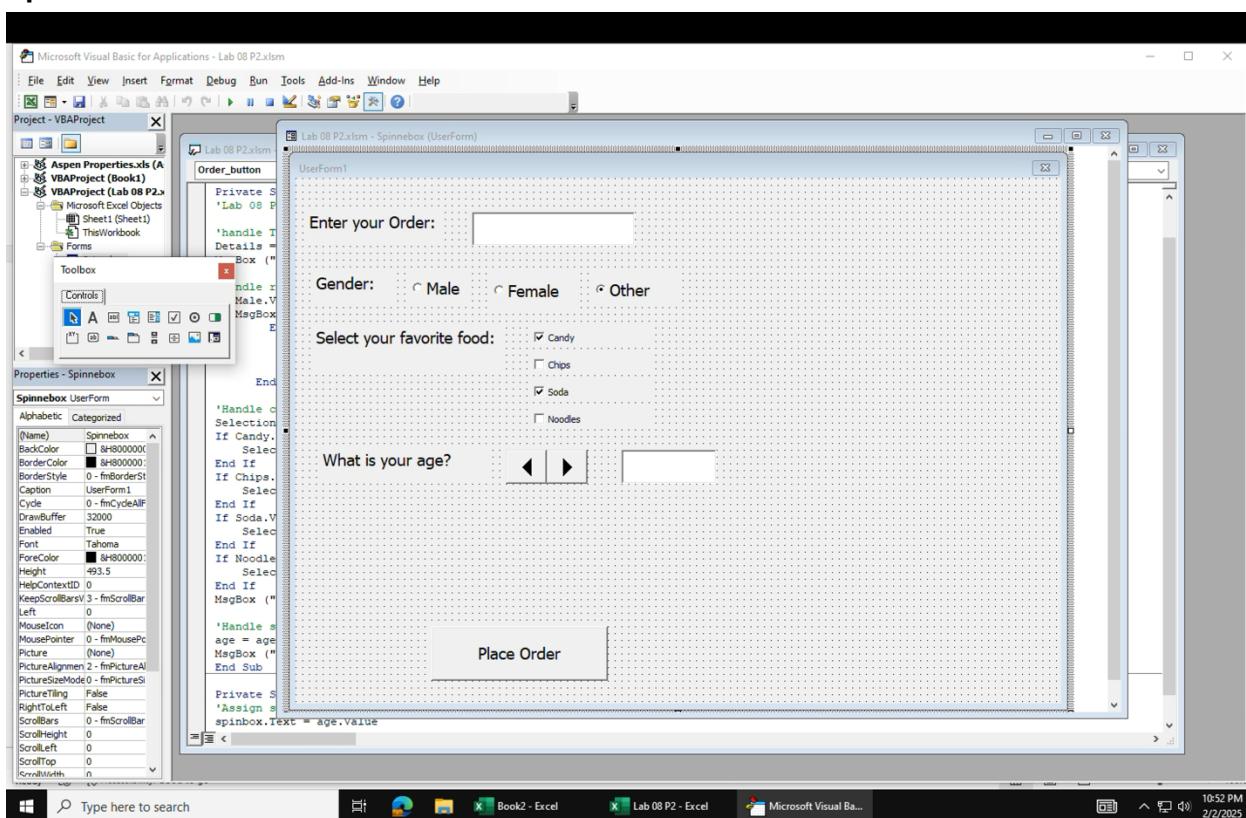
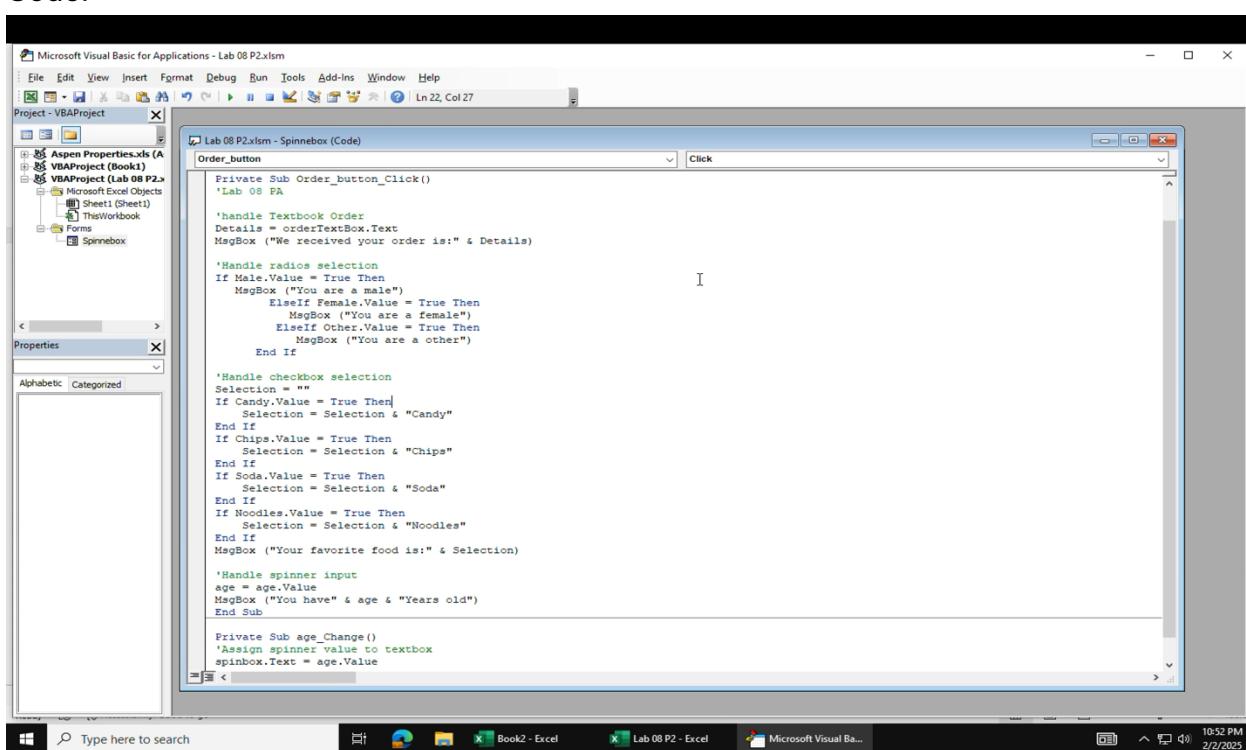
End Sub
```

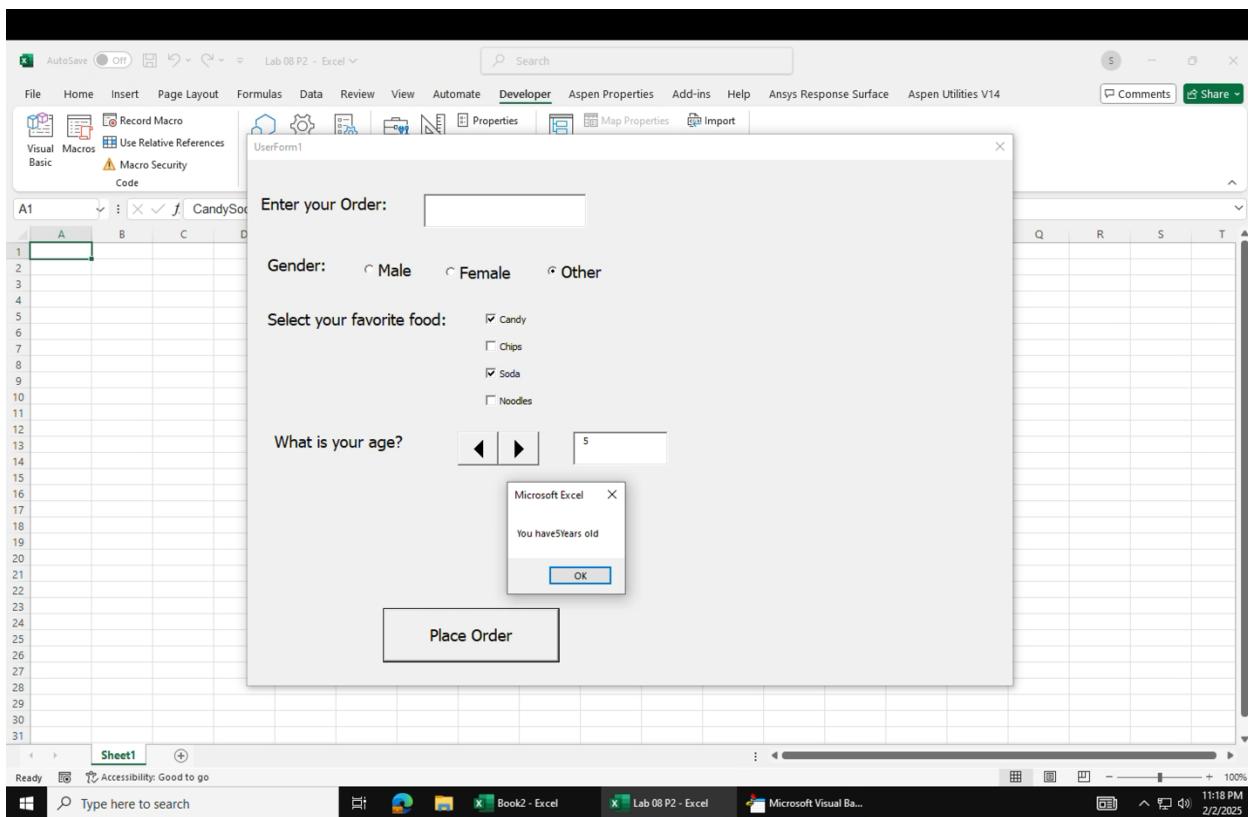
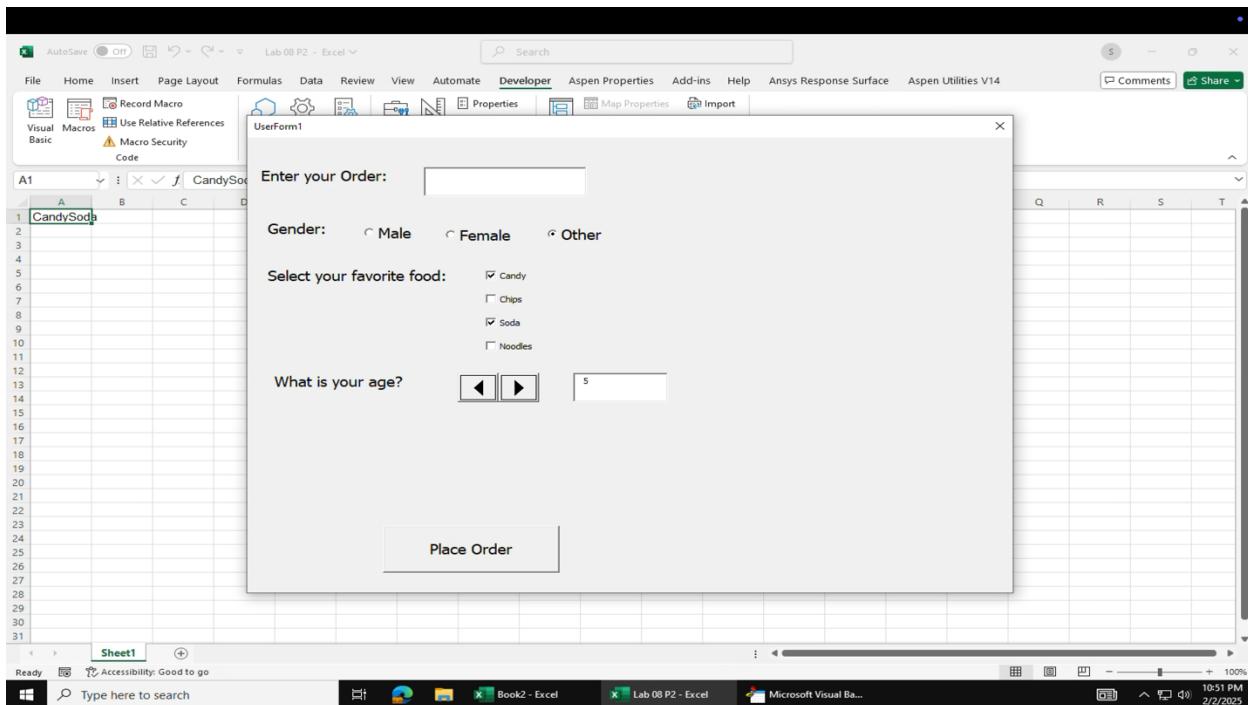
The image shows the Microsoft Visual Basic Editor (VBE) window. On the left is the Project Explorer, and on the right is the code editor. The code editor contains the provided VBA code for the 'Order\_button\_Click' subroutine.

**Output:****Radio Button:****Code:**

**Output:****Checkbox:****Code:**



**Spinner:****Code:**

**Output:**

## 2. Gen AI Prompt Engineering

Creating a Book Library Management System using VBA UserForm.

Problem Statement:

Create a library management system that allows:

Adding new books (Title, Author, ISBN, Publication Year)

Searching books by Title/Author/ISBN

Displaying all books in the library

Removing books from the library

To implement this problem:

1. Create a new Excel workbook
2. Add sheet named "Library" with headers: Title, Author, ISBN, Year
3. Create UserForm with:

- o TextBoxes: txtTitle, txtAuthor, txtISBN, txtYear, txtSearch
- o Buttons: btnAdd, btnSearch, btnRemove
- o ListBox: ListBox1. This list box shows list of books. When user selects one and click Remove button, the book is deleted from Excel file

**I am using Claude.Ai**

Prompt:

You have to create a library management system that allows: 1. Adding new books (Title, Author, ISBN, Publication Year) 2. Searching books by Title/Author/ISBN 3. Removing books from the library. To implement this problem: 1. Create a new Excel workbook 2. Add sheet named "Library" with headers: Title, Author, ISBN, Year 3. Create UserForm with: TextBoxes: txtTitle, txtAuthor, txtISBN, txtYear, txtSearch Buttons: btnAdd, btnSearch, btnRemove ListBox: lstBooks, This list box shows list of books. When user selects one and click Remove button, the book is deleted from Excel file. Please ensure all the operations are correctly mapped. If you cannot give me a downloadable excel file, give me details steps and VBA code to implement this

## Gen AI:

Claude Q Excel VBA Library Management System

I'll help you create a library management system using Excel VBA. I'll provide the detailed steps and code for implementation.

**Library Management System VBA Code**

Click to open code

Here are the steps to implement this library management system:

1. Create the Excel Workbook:
  - Open Excel and save a new workbook
  - The VBA code will automatically create the "Library" sheet with headers when the UserForm is initialized
2. Create the UserForm:
  - Press Alt + F11 to open the VBA editor
  - Right-click on your project in the Project Explorer
  - Select Insert > UserForm
  - Name it "frmLibrary"
3. Add Controls to the UserForm:
  - Add TextBoxes:
    - txtTitle (Label: "Title:")
    - txtAuthor (Label: "Author:")
    - txtISBN (Label: "ISBN:")
    - txtYear (Label: "Year:")
    - txtSearch (Label: "Search:")
  - Add Buttons:

Reply to Claude...

Claude 3.5 Sonnet Choose style ▾

**Library Management System VBA Code**

```
' UserForm Code
Option Explicit

Private Sub UserForm_Initialize()
    ' Initialize headers in Excel if they don't exist
    If WorksheetExists("Library") = False Then
        ThisWorkbook.Sheets.Add.Name = "Library"
        With ThisWorkbook.Sheets("Library")
            .Range("A1").Value = "Title"
            .Range("B1").Value = "Author"
            .Range("C1").Value = "ISBN"
            .Range("D1").Value = "Year"
        End With
    End If

    ' Load existing books into listbox
    LoadBooks
End Sub

Private Sub btnAdd_Click()
    ' Input validation
    If txtTitle.Text = "" Or txtAuthor.Text = "" Or txtISBN.Text = "" Or txtYear.Text = "" Then
        MsgBox "Please fill in all fields!", vbExclamation
        Exit Sub
    End If

    ' Validate year
    If Not IsNumeric(txtYear.Text) Or Len(txtYear.Text) <> 4 Then
        MsgBox "Year must be a four-digit number.", vbInformation
        Exit Sub
    End If

```

Last edited 47 minutes ago

SS

Reply to Claude...

Claude 3.5 Sonnet Choose style ▾

**Library Management System VBA Code**

```
Region: Please enter a valid 4 digit year, yyyymmdd
Exit Sub
End If

' Find the next empty row
Dim ws As Worksheet
Set ws = ThisWorkbook.Sheets("Library")
Dim nextRow As Long
nextRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row + 1

' Add new book
With ws
    .Cells(nextRow, 1).Value = txtTitle.Text
    .Cells(nextRow, 2).Value = txtAuthor.Text
    .Cells(nextRow, 3).Value = txtISBN.Text
    .Cells(nextRow, 4).Value = txtYear.Text
End With

' Clear form and reload books
ClearForm
LoadBooks
MsgBox "Book added successfully!", vbInformation
End Sub

Private Sub btnSearch_Click()
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Library")
    Dim searchText As String
    searchText = LCase(txtSearch.Text)

```

Last edited 47 minutes ago

vba

Sub ShowLibraryForm()
 frmLibrary.Show
End Sub

Features implemented in the code:

- Automatic creation of the Library sheet with headers
- Input validation for all fields
- Search functionality across Title, Author, and ISBN
- Book removal with confirmation
- Automatic list refresh after adding/removing books
- Clear form after adding a book
- Error handling for missing worksheet

Reply to Claude...

Claude 3.5 Sonnet Choose style ▾

Claude Q Excel VBA Library Management System Add ListBook

- lstBooks (make it wide enough to display all book information)

4. Copy and paste the provided VBA code into the UserForm's code window

5. To show the form, add this code in a regular module:

```
vba
Sub ShowLibraryForm()
    frmLibrary.Show
End Sub
```

Features implemented in the code:

- Automatic creation of the Library sheet with headers
- Input validation for all fields
- Search functionality across Title, Author, and ISBN
- Book removal with confirmation
- Automatic list refresh after adding/removing books
- Clear form after adding a book
- Error handling for missing worksheet

Would you like me to explain any particular part of the code in more detail or help you with customizing any of the features?

Claude does not have the ability to run the code it generates yet.

Reply to Claude...  

Claude 3.5 Sonnet  Choose style

```
<- Library Management System VBA Code
    End If
    Next i
End Sub

Private Sub btnRemove_Click()
    If lstBooks.ListIndex = -1 Then
        MsgBox "Please select a book to remove!", vbExclamation
        Exit Sub
    End If

    ' Get the ISBN from the selected item
    Dim selectedISBN As String
    selectedISBN = Split(lstBooks.Value, " | ") (2)

    ' Find and delete the row with matching ISBN
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Library")
    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

    Dim i As Long
    For i = 2 To lastRow
        If ws.Cells(i, 3).Value = selectedISBN Then
            ws.Rows(i).Delete
            Exit For
        End If
    Next i

    ' Reload books

```

Last edited 47 minutes ago   

Claude Q Excel VBA Library Management System Add ListBook

- lstBooks (make it wide enough to display all book information)

4. Copy and paste the provided VBA code into the UserForm's code window

5. To show the form, add this code in a regular module:

```
vba
Sub ShowLibraryForm()
    frmLibrary.Show
End Sub
```

Features implemented in the code:

- Automatic creation of the Library sheet with headers
- Input validation for all fields
- Search functionality across Title, Author, and ISBN
- Book removal with confirmation
- Automatic list refresh after adding/removing books
- Clear form after adding a book
- Error handling for missing worksheet

Would you like me to explain any particular part of the code in more detail or help you with customizing any of the features?

Claude does not have the ability to run the code it generates yet.

Reply to Claude...  

Claude 3.5 Sonnet  Choose style

```
<- Library Management System VBA Code
    ' If search box is empty, show all books
    If searchText = "" Then
        LoadBooks
        Exit Sub
    End If

    ' Search through all rows
    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

    Dim i As Long
    For i = 2 To lastRow
        If LCase(ws.Cells(i, 1).Value) Like "*" & searchText & "*" Or _
            LCase(ws.Cells(i, 2).Value) Like "*" & searchText & "*" Or _
            LCase(ws.Cells(i, 3).Value) Like "*" & searchText & "*" Then
            lstBooks.AddItem ws.Cells(i, 1).Value & " | " & _
                ws.Cells(i, 2).Value & " | " & _
                ws.Cells(i, 3).Value & " | " & _
                ws.Cells(i, 4).Value
        End If
    Next i
End Sub

Private Sub btnRemove_Click()
    If lstBooks.ListIndex = -1 Then
        MsgBox "Please select a book to remove!", vbExclamation
        Exit Sub
    End If

```

Last edited 47 minutes ago   

**Claude** Q Excel VBA Library Management System Add ListBook

- lstBooks (make it wide enough to display all book information)

4. Copy and paste the provided VBA code into the UserForm's code window

5. To show the form, add this code in a regular module:

```
vba
Sub ShowLibraryForm()
    frmLibrary.Show
End Sub
```

Features implemented in the code:

- Automatic creation of the Library sheet with headers
- Input validation for all fields
- Search functionality across Title, Author, and ISBN
- Book removal with confirmation
- Automatic list refresh after adding/removing books
- Clear form after adding a book
- Error handling for missing worksheet

Would you like me to explain any particular part of the code in more detail or help you with customizing any of the features?

(C) Claude does not have the ability to run the code it generates yet.

Reply to Claude...

Claude 3.5 Sonnet Choose style

```
<- Library Management System VBA Code

Dim lastRow As Long
lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

Dim i As Long
For i = 2 To lastRow
    lstBooks.AddItem ws.Cells(i, 1).Value & " | " & _
        ws.Cells(i, 2).Value & " | " & _
        ws.Cells(i, 3).Value & " | " & _
        ws.Cells(i, 4).Value

Next i
End Sub

Private Sub ClearForm()
    txtTitle.Text = ""
    txtAuthor.Text = ""
    txtISBN.Text = ""
    txtYear.Text = ""
End Sub

Private Function WorksheetExists(sheetName As String) As Boolean
    Dim ws As Worksheet

    On Error Resume Next
    Set ws = ThisWorkbook.Sheets(sheetName)
    On Error GoTo 0

    WorksheetExists = Not ws Is Nothing
End Function
```

Last edited 48 minutes ago

Copy Download Publish

**Claude** Q Excel VBA Library Management System Add ListBook

- lstBooks (make it wide enough to display all book information)

4. Copy and paste the provided VBA code into the UserForm's code window

5. To show the form, add this code in a regular module:

```
vba
Sub ShowLibraryForm()
    frmLibrary.Show
End Sub
```

Features implemented in the code:

- Automatic creation of the Library sheet with headers
- Input validation for all fields
- Search functionality across Title, Author, and ISBN
- Book removal with confirmation
- Automatic list refresh after adding/removing books
- Clear form after adding a book
- Error handling for missing worksheet

Would you like me to explain any particular part of the code in more detail or help you with customizing any of the features?

(C) Claude does not have the ability to run the code it generates yet.

Reply to Claude...

Claude 3.5 Sonnet Choose style

```
<- Library Management System VBA Code

' Reload books
LoadBooks
MsgBox "Book removed successfully!", vbInformation
End Sub

Private Sub LoadBooks()
    lstBooks.Clear

    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Library")
    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

    Dim i As Long
    For i = 2 To lastRow
        lstBooks.AddItem ws.Cells(i, 1).Value & " | " & _
            ws.Cells(i, 2).Value & " | " & _
            ws.Cells(i, 3).Value & " | " & _
            ws.Cells(i, 4).Value

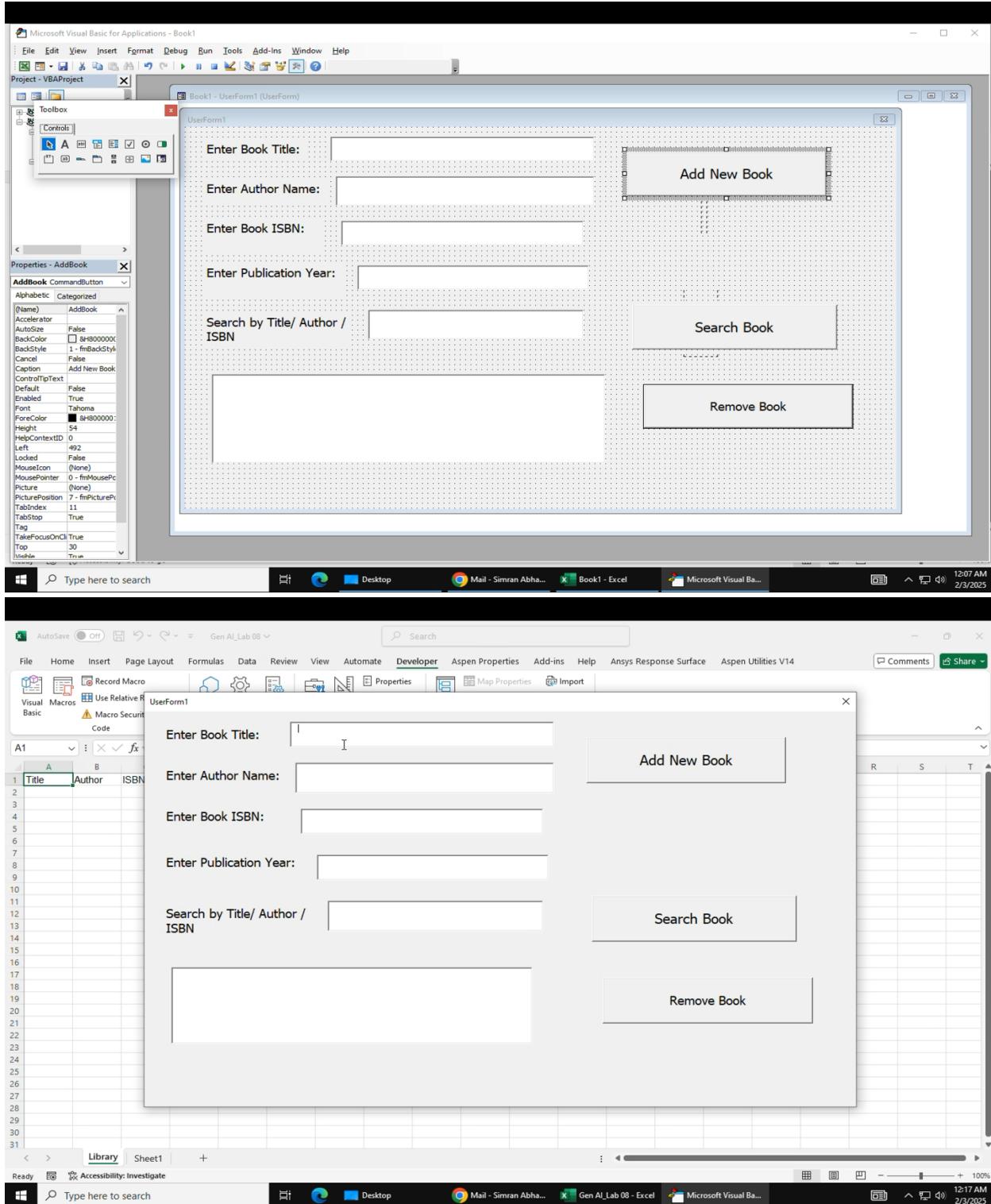
    Next i
End Sub

Private Sub ClearForm()
    txtTitle.Text = ""
    txtAuthor.Text = ""
    txtISBN.Text = ""
    txtYear.Text = ""
End Sub
```

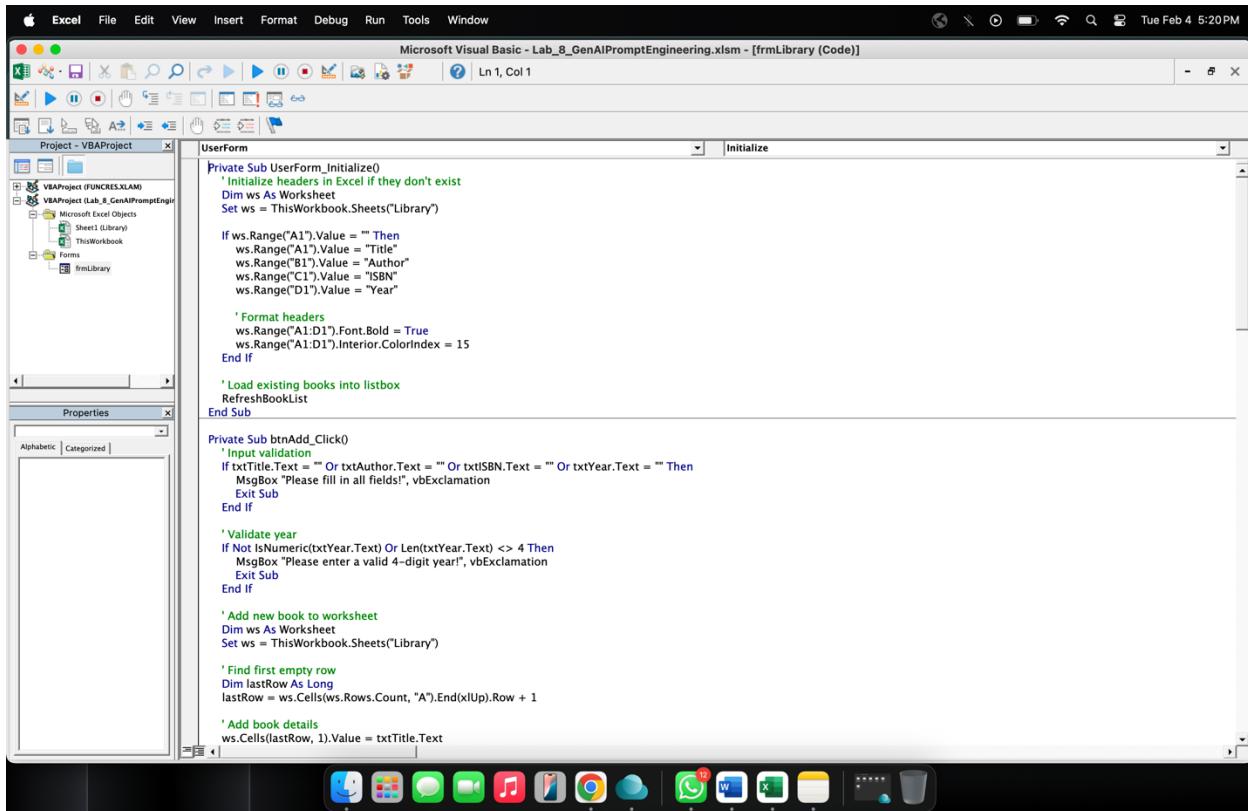
Last edited 48 minutes ago

Copy Download Publish

## Interface using GenAI



## Code:



Microsoft Visual Basic - Lab\_8\_GenAIPromptEngineering.xlsm - [frmLibrary (Code)]

```

Private Sub UserForm_Initialize()
    ' Initialize headers in Excel if they don't exist
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Library")

    If ws.Range("A1").Value = "" Then
        ws.Range("A1").Value = "Title"
        ws.Range("B1").Value = "Author"
        ws.Range("C1").Value = "ISBN"
        ws.Range("D1").Value = "Year"
    End If

    ' Format headers
    ws.Range("A1:D1").Font.Bold = True
    ws.Range("A1:D1").Interior.ColorIndex = 15

    ' Load existing books into listbox
    RefreshBookList
End Sub

Private Sub btnAdd_Click()
    ' Input validation
    If txtTitle.Text = "" Or txtAuthor.Text = "" Or txtISBN.Text = "" Or txtYear.Text = "" Then
        MsgBox "Please fill in all fields!", vbExclamation
        Exit Sub
    End If

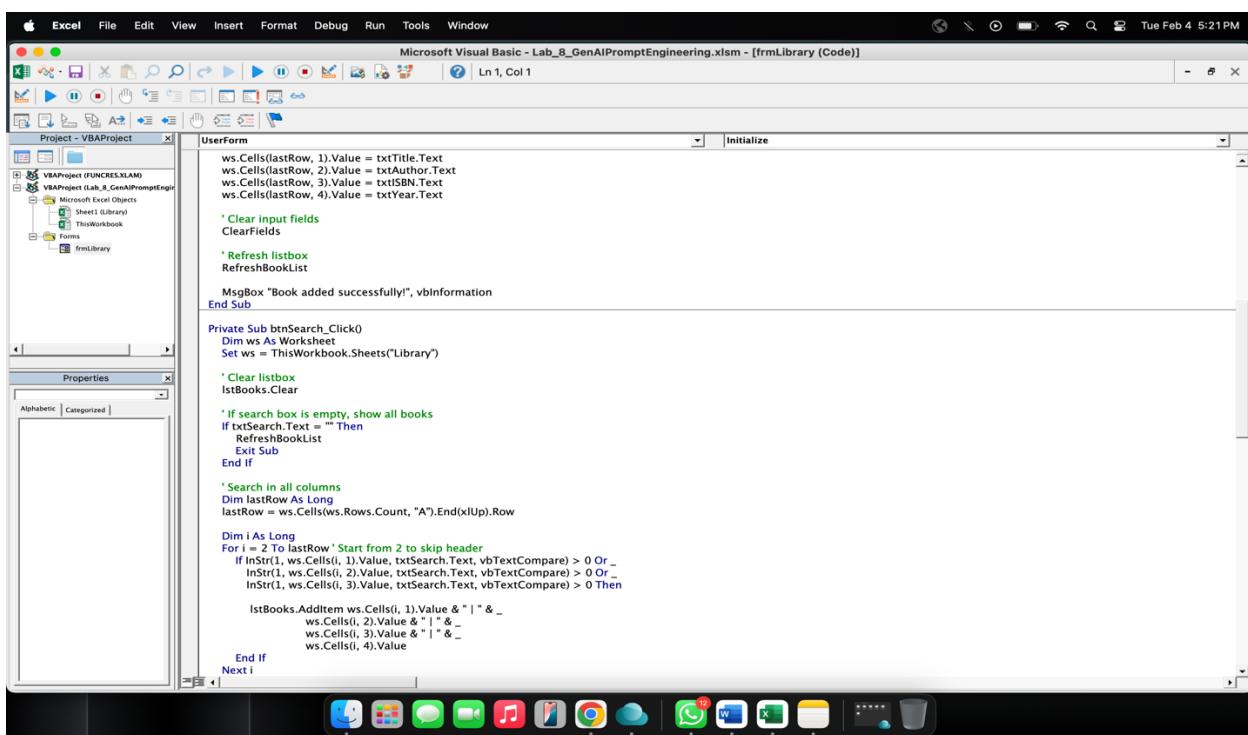
    ' Validate year
    If Not IsNumeric(txtYear.Text) Or Len(txtYear.Text) <> 4 Then
        MsgBox "Please enter a valid 4-digit year!", vbExclamation
        Exit Sub
    End If

    ' Add new book to worksheet
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Library")

    ' Find first empty row
    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row + 1

    ' Add book details
    ws.Cells(lastRow, 1).Value = txtTitle.Text

```



Microsoft Visual Basic - Lab\_8\_GenAIPromptEngineering.xlsm - [frmLibrary (Code)]

```

ws.Cells(lastRow, 2).Value = txtAuthor.Text
ws.Cells(lastRow, 3).Value = txtISBN.Text
ws.Cells(lastRow, 4).Value = txtYear.Text

' Clear input fields
ClearFields

' Refresh listbox
RefreshBookList

MsgBox "Book added successfully!", vbInformation
End Sub

Private Sub btnSearch_Click()
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Library")

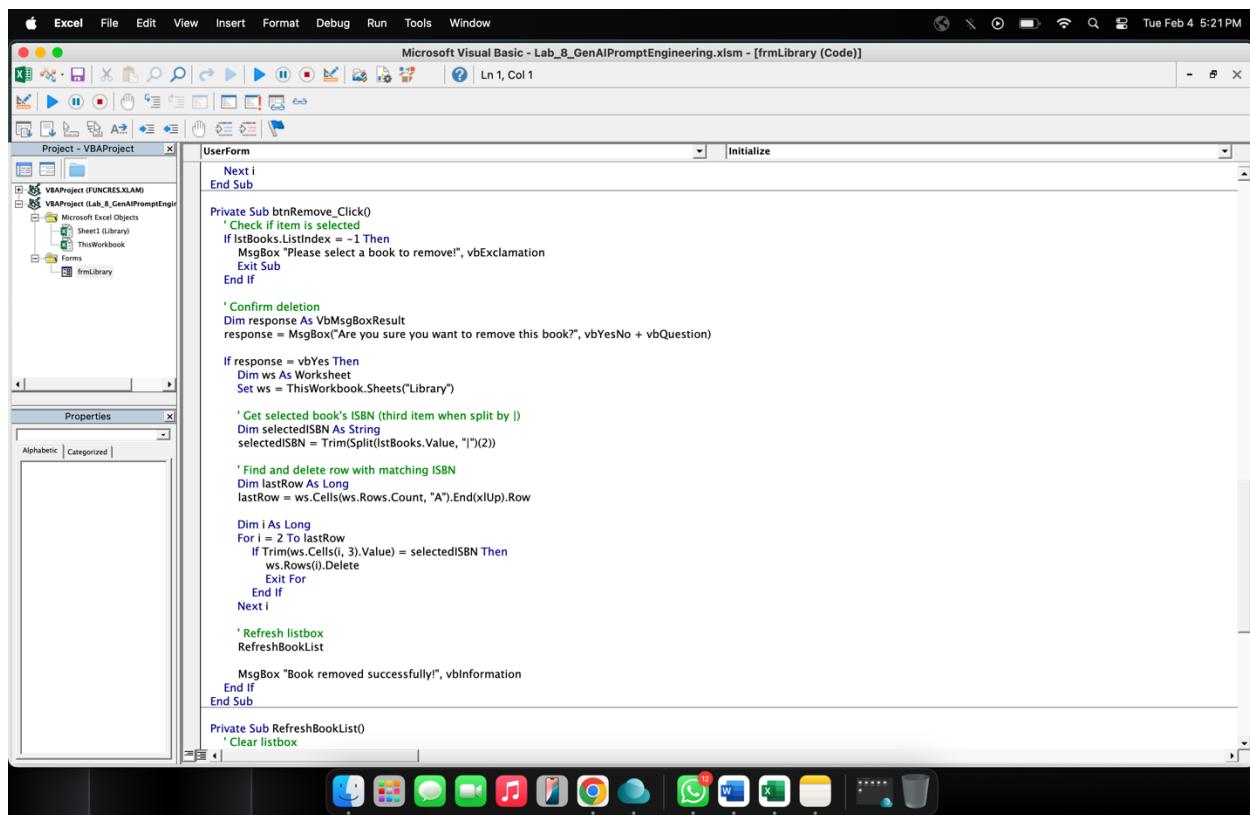
    ' Clear listbox
    lstBooks.Clear

    ' If search box is empty, show all books
    If txtSearch.Text = "" Then
        RefreshBookList
        Exit Sub
    End If

    ' Search in all columns
    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

    Dim i As Long
    For i = 2 To lastRow ' Start from 2 to skip header
        If InStr(1, ws.Cells(i, 1).Value, txtSearch.Text, vbTextCompare) > 0 Or _
           InStr(1, ws.Cells(i, 2).Value, txtSearch.Text, vbTextCompare) > 0 Or _
           InStr(1, ws.Cells(i, 3).Value, txtSearch.Text, vbTextCompare) > 0 Then
            lstBooks.AddItem ws.Cells(i, 1).Value & " | " & _
                           ws.Cells(i, 2).Value & " | " & _
                           ws.Cells(i, 3).Value & " | " & _
                           ws.Cells(i, 4).Value
        End If
    Next i

```



```

Private Sub btnRemove_Click()
    'Check if item is selected
    If lstBooks.ListIndex = -1 Then
        MsgBox "Please select a book to remove!", vbExclamation
        Exit Sub
    End If

    'Confirm deletion
    Dim response As VbMsgBoxResult
    response = MsgBox("Are you sure you want to remove this book?", vbYesNo + vbQuestion)

    If response = vbYes Then
        Dim ws As Worksheet
        Set ws = ThisWorkbook.Sheets("Library")

        'Get selected book's ISBN (third item when split by |)
        Dim selectedISBN As String
        selectedISBN = Trim(Split(lstBooks.Value, "|")(2))

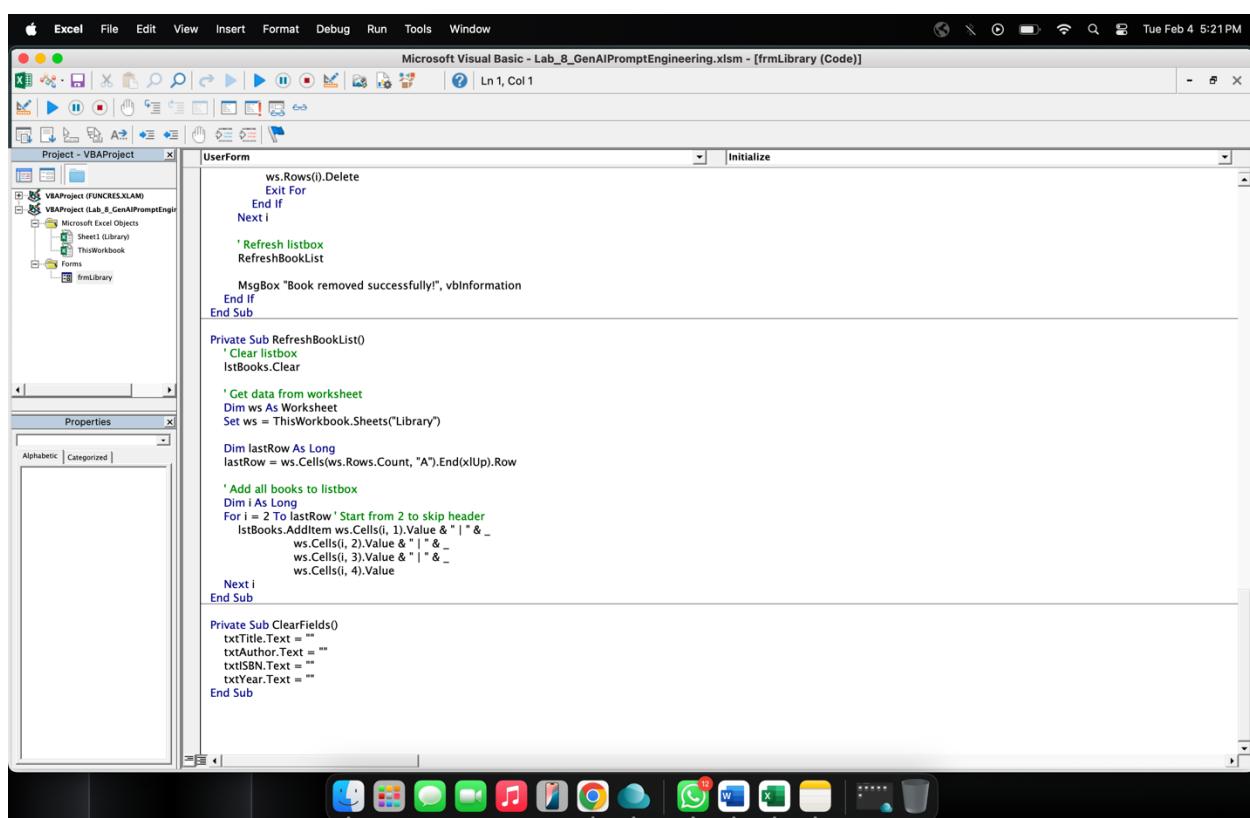
        'Find and delete row with matching ISBN
        Dim lastRow As Long
        lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

        Dim i As Long
        For i = 2 To lastRow
            If Trim(ws.Cells(i, 3).Value) = selectedISBN Then
                ws.Rows(i).Delete
                Exit For
            End If
        Next i

        'Refresh listbox
        RefreshBookList

        MsgBox "Book removed successfully!", vbInformation
    End If
End Sub

```



```

ws.Rows(i).Delete
Exit For
End If
Next i

' Refresh listbox
RefreshBookList

MsgBox "Book removed successfully!", vbInformation
End If
End Sub

Private Sub RefreshBookList()
    'Clear listbox
    lstBooks.Clear

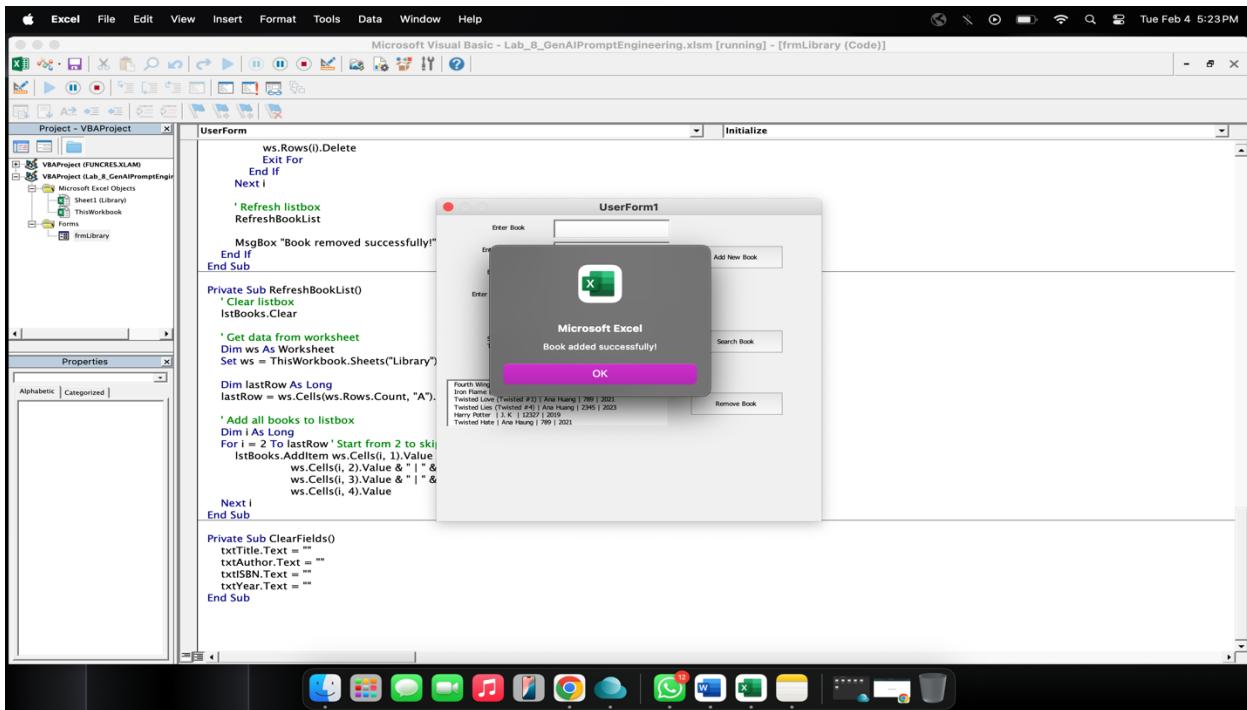
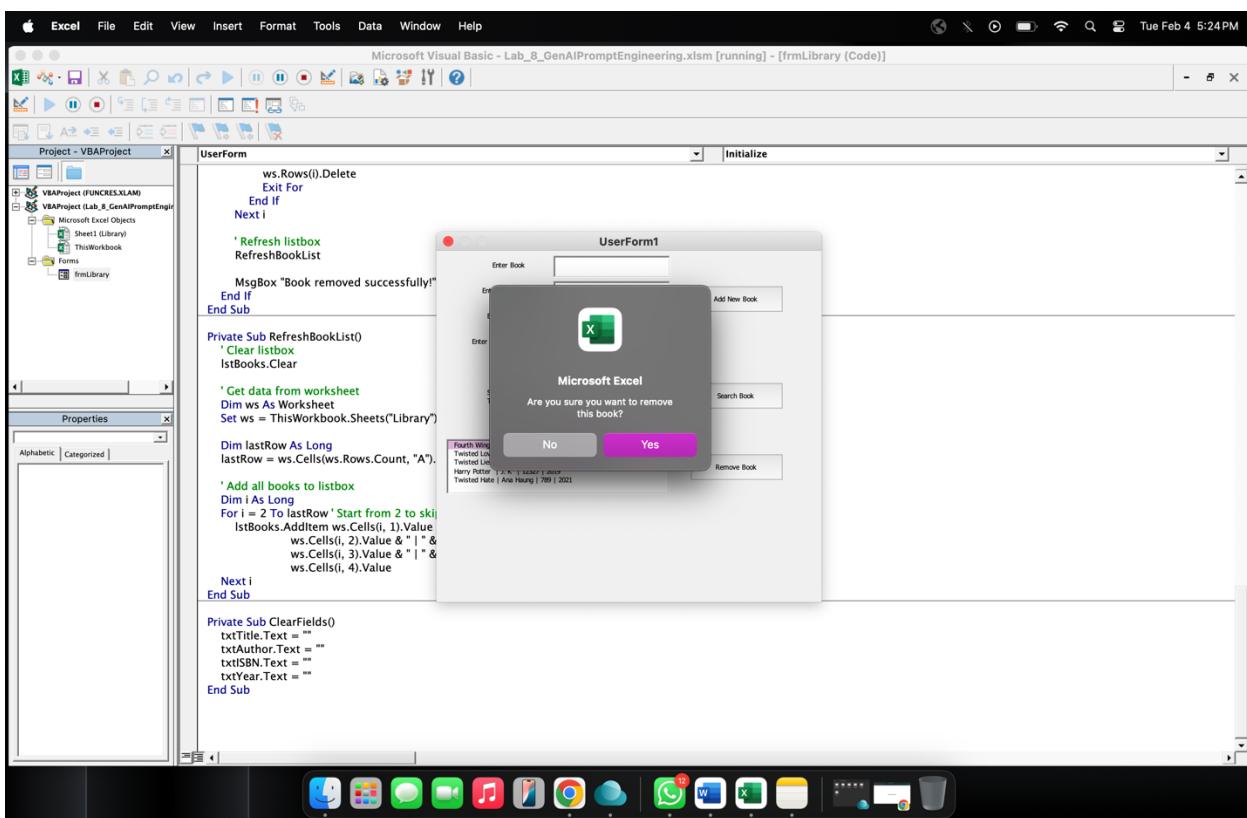
    'Get data from worksheet
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Library")

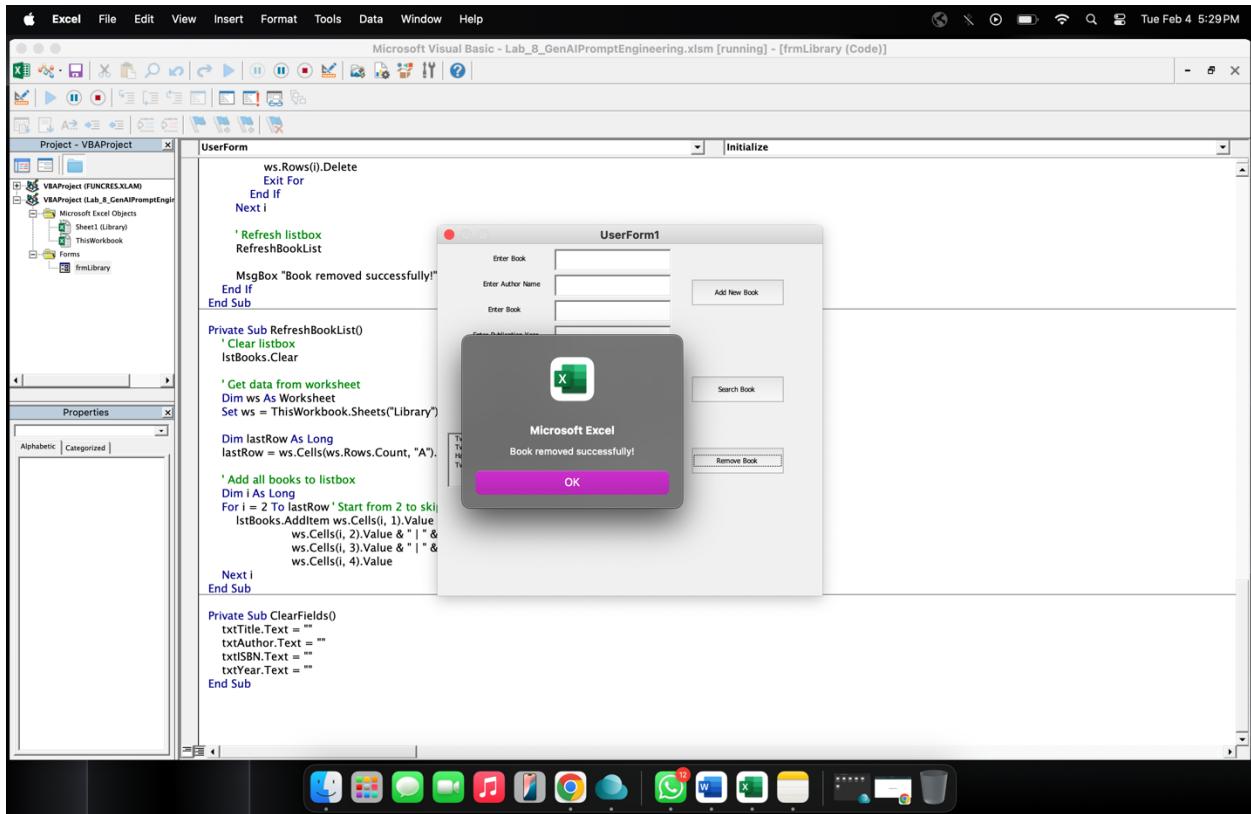
    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

    'Add all books to listbox
    Dim i As Long
    For i = 2 To lastRow 'Start from 2 to skip header
        lstBooks.AddItem ws.Cells(i, 1).Value & " | " & _
            ws.Cells(i, 2).Value & " | " & _
            ws.Cells(i, 3).Value & " | " & _
            ws.Cells(i, 4).Value
    Next i
End Sub

Private Sub ClearFields()
    txtTitle.Text = ""
    txtAuthor.Text = ""
    txtISBN.Text = ""
    txtYear.Text = ""
End Sub

```

**Output:****Add Book****Delete Book**



Search Book

