Option Explicit

Private Sub cmdDelete\_Click()

Dim msgValue As VbMsgBoxResult

'msgValue = MsgBox("Do you want to delete the record?", vbYesNo + vbQuestion, "Delete")

msgValue = MsgBox("Delete Not Available", vbOKOnly + vbExclamation, "Delete")

'If msgValue = vbYes Then

' Call DeleteRecord

'End If

End Sub

Private Sub cmdModify\_Click()

Dim msgValue As VbMsgBoxResult

msgValue = MsgBox("Do you want to modify the record?", vbYesNo + vbQuestion, "Modify")

If msgValue = vbYes Then

Call Modify

End If

End Sub

Private Sub cmdReset\_Click()

Dim msgValue As VbMsgBoxResult

msgValue = MsgBox("Do you want to reset the Form?", vbYesNo + vbQuestion, "Reset")

If msgValue = vbYes Then

Call Reset

End If

End Sub

Private Sub cmdSave\_Click()

If Validate = True Then

Dim msgValue As VbMsgBoxResult

msgValue = MsgBox("Do you want to save the data?", vbYesNo + vbQuestion, "Save")

If msgValue = vbYes Then

Call Save

Call Reset

End If

End If

End Sub

Attribute VB\_Name = "Module1"

Option Explicit

Function Validate() As Boolean

' CODE VERIFIES VALUES FOR SOME OF THE DATA

Dim frm As Worksheet

Sheets("FORM").Select

ActiveSheet.Unprotect

Set frm = ThisWorkbook.Sheets("Form")

Validate = True

With frm

.Range("L5").Interior.Color = xlNone

.Range("L6").Interior.Color = xlNone

.Range("L7").Interior.Color = xlNone

.Range("L8").Interior.Color = xlNone

.Range("L9").Interior.Color = xlNone

.Range("L10").Interior.Color = xlNone

.Range("L11").Interior.Color = xlNone

End With

'Validating Tick Number

If Trim(frm.Range("V23").Value) <> 10 Then

MsgBox "Tick Number is incorrect.", vbOKOnly + vbExclamation, "Tick Number"

Validate = False

Exit Function

End If

'Validating Property Number

If Trim(frm.Range("L5").Value) = "" Then

MsgBox "Property Number is blank.", vbOKOnly + vbInformation, "Property Number"

frm.Range("L5").Select

frm.Range("L5").Interior.Color = vbRed

Validate = False

Exit Function

End If

'Validating Property Description

If Trim(frm.Range("L6").Value) = "" Then

MsgBox "Property Description is blank.", vbOKOnly + vbInformation, "Property Description"

frm.Range("L6").Select

frm.Range("L6").Interior.Color = vbRed

Validate = False

Exit Function

End If

'Validating Property Location

If Trim(frm.Range("L7").Value) = "" Then

MsgBox "Property Location is blank", vbOKOnly + vbInformation, "Property Location"

frm.Range("L7").Select

frm.Range("L7").Interior.Color = vbRed

Validate = False

Exit Function

End If

'Validating Unit QTY

If Trim(frm.Range("L8").Value) = "" Then

MsgBox "Unit Qty is Blank", vbOKOnly + vbInformation, "Unit QTY"

frm.Range("L8").Select

frm.Range("L8").Interior.Color = vbRed

Validate = False

Exit Function

End If

'Validating Unit Type

'If Trim(frm.Range("I14").Value) = "" Or Not IsNumeric(Trim(frm.Range("I14").Value)) Then

If Trim(frm.Range("L9").Value) = "" Then

MsgBox "Unit Type is Blank", vbOKOnly + vbInformation, "Unit Type"

frm.Range("L9").Select

frm.Range("L9").Interior.Color = vbRed

Validate = False

Exit Function

End If

'Validating Inspection Type

If Trim(frm.Range("L10").Value) = "" Then

MsgBox "Inspection Type is Blank", vbOKOnly + vbInformation, "Inspection Type"

frm.Range("L10").Select

frm.Range("L10").Interior.Color = vbRed

Validate = False

Exit Function

End If

'Validating Allocated Time (HRs)

If Trim(frm.Range("L11").Value) = "" Then

MsgBox "Allocated Time (HRs) is Blank", vbOKOnly + vbInformation, "Allocated Time (HRs)"

frm.Range("L11").Select

frm.Range("L11").Interior.Color = vbRed

Validate = False

Exit Function

End If

Sheets("FORM").Select

ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True

End Function

Sub Reset()

' CODE CLEARS THE FORM AND REFRESHES FOR NEXT ACTION

Sheets("FORM").Select

ActiveSheet.Unprotect

With Sheets("Form")

.Range("L5").Interior.Color = xlNone

.Range("L5").Value = ""

.Range("L6").Interior.Color = xlNone

.Range("L6").Value = ""

.Range("L7").Interior.Color = xlNone

.Range("L7").Value = ""

.Range("L8").Interior.Color = xlNone

.Range("L8").Value = ""

.Range("L9").Interior.Color = xlNone

.Range("L9").Value = ""

.Range("L10").Interior.Color = xlNone

.Range("L10").Value = ""

.Range("L11").Interior.Color = xlNone

.Range("L11").Value = ""

.Range("L13:N22").Interior.Color = xlNone

.Range("L13:N22").Value = ""

End With

Sheets("FORM").Select

ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True

'Range("A1").Select

Range("L6").Select

End Sub

Sub Save()

' CODE WRITES THE UPDATE VALUES TO THE DATABASE TAB IN THE WORKBOOK

Dim frm As Worksheet

Dim Database As Worksheet

Dim iRow As Long

Dim iSerial As Variant

Set frm = ThisWorkbook.Sheets("Form")

Set Database = ThisWorkbook.Sheets("Database")

If Trim(frm.Range("M1").Value) = "" Then

iRow = Database.Range("A" & Application.Rows.Count).End(xlUp).Row + 1

If iRow = 2 Then

iSerial = 1

Else

iSerial = Database.Cells(iRow - 1, 1).Value + 1

End If

Else

iRow = frm.Range("L1").Value

iSerial = frm.Range("M1").Value

End If

With Database

.Cells(iRow, 3).Value = frm.Range("L7").Value

.Cells(iRow, 8).Value = frm.Range("O13").Value

.Cells(iRow, 9).Value = frm.Range("O14").Value

.Cells(iRow, 10).Value = frm.Range("O15").Value

.Cells(iRow, 11).Value = frm.Range("O16").Value

.Cells(iRow, 12).Value = frm.Range("O17").Value

.Cells(iRow, 13).Value = frm.Range("O18").Value

.Cells(iRow, 14).Value = frm.Range("O19").Value

.Cells(iRow, 15).Value = frm.Range("O20").Value

.Cells(iRow, 16).Value = frm.Range("O21").Value

.Cells(iRow, 17).Value = frm.Range("O22").Value

End With

'frm.Range("L1").Value = ""

'frm.Range("M1").Value = ""

End Sub

Sub Modify()

' CODE PULLS DATA FROM DATABASE AND LOADS INTO FORM TAB FOR MANIPULATION

Dim iRow As Long

Dim iSerial As Variant

Sheets("FORM").Select

ActiveSheet.Unprotect

iSerial = Application.InputBox("Please enter Serial Number to make modification.", "Modify", , , , , , 2)

On Error Resume Next

iRow = Application.WorksheetFunction.IfError \_

(Application.WorksheetFunction.Match(iSerial, Sheets("Database").Range("A:A"), 0), 0)

On Error GoTo 0

If iRow = 0 Then

MsgBox "No record found.", vbOKOnly + vbCritical, "No Record"

Exit Sub

End If

Sheets("Form").Range("L1").Value = iRow

Sheets("Form").Range("M1").Value = iSerial

Sheets("Form").Range("L5").Value = Sheets("Database").Cells(iRow, 1).Value

Sheets("Form").Range("L6").Value = Sheets("Database").Cells(iRow, 2).Value

Sheets("Form").Range("L7").Value = Sheets("Database").Cells(iRow, 3).Value

Sheets("Form").Range("L8").Value = Sheets("Database").Cells(iRow, 4).Value

Sheets("Form").Range("L9").Value = Sheets("Database").Cells(iRow, 5).Value

Sheets("Form").Range("L10").Value = Sheets("Database").Cells(iRow, 6).Value

Sheets("Form").Range("L11").Value = Sheets("Database").Cells(iRow, 7).Value

Sheets("FORM").Select

ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True

End Sub

Sub DeleteRecord()

' CODE DELETES RECORD FROM DATABASE, THERE IS NO RECOVERY FROM THIS!!!!!!!!!!!!!!!!!

Dim iRow As Long

Dim iSerial As Long

iSerial = Application.InputBox("Please enter S.No. to delete the recor.", "Delete", , , , , , 1)

On Error Resume Next

iRow = Application.WorksheetFunction.IfError \_

(Application.WorksheetFunction.Match(iSerial, Sheets("Database").Range("A:A"), 0), 0)

On Error GoTo 0

If iRow = 0 Then

MsgBox "No record found.", vbOKOnly + vbCritical, "No Record"

Exit Sub

End If

Sheets("Database").Cells(iRow, 1).EntireRow.Delete shift:=xlUp

End Sub

Attribute VB\_Name = "Module3"

Sub Sheet\_Protection()

Attribute Sheet\_Protection.VB\_ProcData.VB\_Invoke\_Func = " \n14"

' Sheet\_Protection Macro

Sheets("FORM").Select

ActiveSheet.Unprotect

Sheets("FORM").Select

ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True

End Sub

Sub Print\_ICA\_Reports()

Attribute Print\_ICA\_Reports.VB\_ProcData.VB\_Invoke\_Func = " \n14"

'

' Print\_Reports Macro

'

'

Call PT\_Refresh

Sheets("ICA").Select

ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True, IgnorePrintAreas:=False

End Sub

Sub Print\_IFI\_Reports()

'

' Print\_Reports Macro

'

'

Call PT\_Refresh

Sheets("IFI").Select

ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True, IgnorePrintAreas:=False

End Sub

Sub Print\_ISR\_Reports()

'

' Print\_Reports Macro

'

'

Call PT\_Refresh

Sheets("ISR").Select

ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True, IgnorePrintAreas:=False

End Sub

Sub Print\_VD\_Reports()

'

' Print\_Reports Macro

'

'

Call PT\_Refresh

Sheets("VD").Select

ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True, IgnorePrintAreas:=False

End Sub

Sub Print\_VML\_Reports()

'

' Print\_Reports Macro

'

'

Call PT\_Refresh

Sheets("VML").Select

ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True, IgnorePrintAreas:=False

End Sub

Sub PT\_Refresh()

Attribute PT\_Refresh.VB\_ProcData.VB\_Invoke\_Func = " \n14"

'

' PT\_Refresh Macro

'

'

Range("A7").Select

ActiveWorkbook.RefreshAll

End Sub