```
Sub ets()

End Sub
Private Sub CommandButton1_Click()

End Sub

Private Sub CommandButton2_Click()

End Sub

Private Sub Frame1_Click()

End Sub

Private Sub Label1_Click()

End Sub

Private Sub Label2_Click()

End Sub

Private Sub Label3_Click()

End Sub

Private Sub Label4_Click()

End Sub

Private Sub TextBox2_Change()

End Sub

Private Sub TextBox3_Change()

End Sub

Private Sub UserForm_Click()

End Sub

Private Sub UserForm_DblClick(ByVal Cancel As MSForms.ReturnBoolean)

End Sub

Private Sub UserForm_Error(ByVal Number As Integer, ByVal Description As MSForms.ReturnString, ByVal S
Code As Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext As Long, ByVal Cance
lDisplay As MSForms.ReturnBoolean)

End Sub

Private Sub UserForm_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

End Sub

Private Sub UserForm_MouseUp(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal
 Y As Single)

End Sub

Private Sub UserForm_RemoveControl(ByVal Control As MSForms.Control)

End Sub

Private Sub UserForm_Resize()

End Sub

Private Sub UserForm_Scroll(ByVal ActionX As MSForms.fmScrollAction, ByVal ActionY As MSForms.fmScroll
Action, ByVal RequestDx As Single, ByVal RequestDy As Single, ByVal ActualDx As MSForms.ReturnSingle,
ByVal ActualDy As MSForms.ReturnSingle)
```

```

End Sub

End Sub
Private Sub Label5_Click()
End Sub
Private Sub Label6_Click()
End Sub
Private Sub Label7_Click()
End Sub
Private Sub Label8_Click()
End Sub
Private Sub Label9_Click()
End Sub
Private Sub TextBox1_Change()
End Sub
Private Sub TextBox10_Change()
End Sub
Private Sub TextBox11_Change()
End Sub
Private Sub TextBox12_Change()
End Sub
Private Sub TextBox13_Change()
End Sub
Private Sub TextBox14_Change()
End Sub
Private Sub TextBox15_Change()
End Sub
Private Sub TextBox16_Change()
End Sub
Private Sub TextBox17_Change()
End Sub
Private Sub TextBox19_Change()
End Sub
Private Sub TextBox20_Change()
End Sub
Private Sub TextBox21_Change()
End Sub
```

```
Private Sub TextBox23_Change()

End Sub


End Sub

Private Sub TextBox5_Change()

End Sub

Private Sub TextBox6_Change()

End Sub

Private Sub TextBox8_Change()

End Sub

Private Sub TextBox9_Change()

End Sub

Private Sub UserForm_Activate()

End Sub

Private Sub UserForm_AddControl(ByVal Control As MSForms.Control)

End Sub

Private Sub UserForm_BeforeDragOver(ByVal Cancel As MSForms.ReturnBoolean, ByVal Control As MSForms.Co
ntrol, ByVal Data As MSForms.DataObject, ByVal X As Single, ByVal Y As Single, ByVal State As MSForms.
fmDragState, ByVal Effect As MSForms.ReturnEffect, ByVal Shift As Integer)

End Sub

Private Sub UserForm_BeforeDropOrPaste(ByVal Cancel As MSForms.ReturnBoolean, ByVal Control As MSForms
.Control, ByVal Action As MSForms.fmAction, ByVal Data As MSForms.DataObject, ByVal X As Single, ByVal
 Y As Single, ByVal Effect As MSForms.ReturnEffect, ByVal Shift As Integer)

End Sub



End Sub


End Sub

Private Sub UserForm_Deactivate()

End Sub


End Sub



End Sub




Private Sub UserForm_KeyUp(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
End Sub
End Sub


Private Sub ComboBox1_Change()

End Sub
```

```
Private Sub ComboBox2_Change()
End Sub
Private Sub ComboBox3_Change()
End Sub
Private Sub ComboBox5_Change()
End Sub

End Sub

End Sub
Private Sub CommandButton3_Click()
End Sub
Private Sub CommandButton4_Click()
End Sub

End Sub

End Sub
Private Sub OptionButton1_Click()
End Sub

End Sub

End Sub

End Sub

End Sub

End Sub

End Sub
Private Sub UserForm_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
End Sub

End Sub
End Sub

Private Sub UserForm_Terminate()

End Sub

Private Sub UserForm_Zoom(Percent As Integer)

End Sub
Option Explicit

Private Const SHEET_CASES As String = "Cases"

Private isInitializing As Boolean

'==========================
' Lifecycle
'==========================

    On Error Resume Next
    isInitializing = True

    EnsureCasesSheet

    ' Populate top-level lists
    With Me.ComboBox1 ' Case Type
        .Clear
        .AddItem "Refund"
        .AddItem "Compensation"
        .AddItem "Recognition"
        .AddItem "Insurance claim"
    End With

    With Me.ComboBox3 ' Issuing Body
        .Clear
        .AddItem "Institution"
        .AddItem "SETA"
        .AddItem "QCTO"
        .AddItem "CCMA"
        .AddItem "Department of Employment and Labour"
        .AddItem "Other"
    End With

    With Me.ComboBox5 ' Desired Outcome
        .Clear
        .AddItem "Refund"
        .AddItem "Credit"
        .AddItem "Provisional certificate"
        .AddItem "Appeal"
        .AddItem "Escalation"
        .AddItem "Correction/Letter of completion"
    End With

    ' Priority toggle
    Me.OptionButton1.Caption = "Visa/Job critical"
    Me.OptionButton1.Value = False

    ' Sensible defaults
    Me.ComboBox1.ListIndex = -1
    Me.ComboBox2.Clear
    Me.ComboBox3.ListIndex = -1
    Me.ComboBox5.ListIndex = -1

    isInitializing = False
    On Error GoTo 0
End Sub

    On Error Resume Next
    If Me.ComboBox1.ListCount > 0 Then Me.ComboBox1.SetFocus
    On Error GoTo 0
End Sub

    ' No special teardown
End Sub

'==========================
' Dynamic lists & helpers
'==========================
    ' Case Type changed -> repopulate Scenario list
    If isInitializing Then Exit Sub
    FillScenarioList Me.ComboBox1.Value
    SuggestOutcome
End Sub


    If isInitializing Then Exit Sub
    SuggestOutcome
End Sub

    ' Issuing body selected; no-op or future routing logic
End Sub


    ' User prefers a specific outcome; respect selection
End Sub

    ' Toggle priority; could visually cue user
End Sub

    ' Container click; no action
End Sub

    ' Could display help or open a guidance sheet
    MsgBox "Select Case Type ? Scenario ? Issuing Body ? Desired Outcome. Then Submit or Save Draft.",
 vbInformation, "Help"
End Sub

'==========================
' Commands
'==========================

    ' Submit (final)
    If Not ValidateForm(True) Then Exit Sub

    Dim caseId As String
    caseId = GenerateCaseId

    Dim ws As Worksheet
    Set ws = ThisWorkbook.Worksheets(SHEET_CASES)

    Dim r As Long
    r = NextFreeRow(ws)

    ws.Cells(r, 1).Value = Now
    ws.Cells(r, 2).Value = caseId
    ws.Cells(r, 3).Value = Nz(Me.ComboBox1.Value)
    ws.Cells(r, 4).Value = Nz(Me.ComboBox2.Value)
    ws.Cells(r, 5).Value = Nz(Me.ComboBox3.Value)
    ws.Cells(r, 6).Value = Nz(Me.ComboBox5.Value)
    ws.Cells(r, 7).Value = IIf(Me.OptionButton1.Value, "High", "Normal")
    ws.Cells(r, 8).Value = "Submitted"
    ws.Cells(r, 9).Value = "" ' Notes (optional)

    MsgBox "Case submitted: " & caseId, vbInformation, "Success"

    ResetForm
End Sub

    ' Save draft (partial allowed)
    Dim caseId As String
    caseId = GenerateCaseId
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Worksheets(SHEET_CASES)

    Dim r As Long
    r = NextFreeRow(ws)

    ws.Cells(r, 1).Value = Now
    ws.Cells(r, 2).Value = caseId
    ws.Cells(r, 3).Value = Nz(Me.ComboBox1.Value)
    ws.Cells(r, 4).Value = Nz(Me.ComboBox2.Value)
    ws.Cells(r, 5).Value = Nz(Me.ComboBox3.Value)
    ws.Cells(r, 6).Value = Nz(Me.ComboBox5.Value)
    ws.Cells(r, 7).Value = IIf(Me.OptionButton1.Value, "High", "Normal")
    ws.Cells(r, 8).Value = "Draft"
    ws.Cells(r, 9).Value = "" ' Notes

    MsgBox "Draft saved: " & caseId, vbInformation, "Saved"
End Sub

    ' Reset
    ResetForm
End Sub

    ' Close
    Unload Me
End Sub

'==========================
' User experience events
'==========================

    ' ESC closes; Ctrl+S saves draft
    If KeyCode = vbKeyEscape Then
        Unload Me
    ElseIf KeyCode = vbKeyS And (Shift And fmCtrlMask) = fmCtrlMask Then
        CommandButton2_Click
    End If
End Sub

    ' No-op
End Sub


    ' No-op
End Sub

    ' Hook for responsive layout if needed
End Sub

    ' No-op
End Sub


End Sub

End Sub


    ' Keep default behavior
End Sub

    ' Optionally reposition/resize controls here
End Sub

'==========================
' Helpers
'==========================
Private Sub FillScenarioList(ByVal caseType As String)
    Me.ComboBox2.Clear

    Select Case LCase$(Trim$(caseType))
        Case "refund"
```

```
            Me.ComboBox2.AddItem "Training not delivered"
            Me.ComboBox2.AddItem "Material defects / not as described"
            Me.ComboBox2.AddItem "Admin error in registration"
            Me.ComboBox2.AddItem "Overbilling"
        Case "compensation"
            Me.ComboBox2.AddItem "Diploma printing delay (loss of opportunity)"
            Me.ComboBox2.AddItem "Application rejected without due cause"
            Me.ComboBox2.AddItem "Published without registration confirmation"
        Case "recognition"
            Me.ComboBox2.AddItem "Request provisional certificate"
            Me.ComboBox2.AddItem "Request letter of completion"
            Me.ComboBox2.AddItem "Appeal assessment outcome"
        Case "insurance claim"
            Me.ComboBox2.AddItem "Policy claim for learning costs"
            Me.ComboBox2.AddItem "Denied claim appeal"
        Case Else
            ' Generic fallback
            Me.ComboBox2.AddItem "Other"
    End Select
End Sub

Private Sub SuggestOutcome()
    ' Suggest an outcome based on scenario keywords (non-binding)
    Dim s As String
    s = LCase$(Nz(Me.ComboBox2.Value))

    If s Like "*not delivered*" Or s Like "*overbilling*" Then
        SelectOutcomeIfExists "Refund"
    ElseIf s Like "*printing*" Or s Like "*provisional*" Or s Like "*completion*" Then
        SelectOutcomeIfExists "Provisional certificate"
    ElseIf s Like "*rejected*" Or s Like "*appeal*" Then
        SelectOutcomeIfExists "Appeal"
    ElseIf s Like "*published*" Or s Like "*admin*" Then
        SelectOutcomeIfExists "Correction/Letter of completion"
    End If
End Sub

Private Sub SelectOutcomeIfExists(ByVal text As String)
    Dim i As Long
    For i = 0 To Me.ComboBox5.ListCount - 1
        If StrComp(Me.ComboBox5.List(i), text, vbTextCompare) = 0 Then
            Me.ComboBox5.ListIndex = i
            Exit For
        End If
    Next i
End Sub

Private Function ValidateForm(ByVal isFinal As Boolean) As Boolean
    ValidateForm = False

    Dim missing As String
    missing = ""

    If Len(Trim$(Nz(Me.ComboBox1.Value))) = 0 Then missing = missing & "- Case Type" & vbCrLf
    If Len(Trim$(Nz(Me.ComboBox2.Value))) = 0 Then missing = missing & "- Scenario" & vbCrLf
    If Len(Trim$(Nz(Me.ComboBox3.Value))) = 0 Then missing = missing & "- Issuing Body" & vbCrLf

    If isFinal And Len(missing) > 0 Then
        MsgBox "Please complete the following before submitting:" & vbCrLf & vbCrLf & missing, vbExcla
mation, "Incomplete"
        Exit Function
    End If

    ValidateForm = True
End Function

Private Sub ResetForm()
    isInitializing = True

    Me.ComboBox1.ListIndex = -1
    Me.ComboBox2.Clear
    Me.ComboBox3.ListIndex = -1
    Me.ComboBox5.ListIndex = -1
    Me.OptionButton1.Value = False

    isInitializing = False
End Sub

Private Function GenerateCaseId() As String
    GenerateCaseId = "CASE-" & Format(Now, "yymmdd-hhnnss")
End Function

Private Function NextFreeRow(ws As Worksheet) As Long
    Dim r As Long
    r = ws.Cells(ws.Rows.Count, 1).End(xlUp).row
    If r < 2 Then
        NextFreeRow = 2
    Else
        NextFreeRow = r + 1
    End If
End Function

Private Sub EnsureCasesSheet()
    Dim ws As Worksheet
    On Error Resume Next
    Set ws = ThisWorkbook.Worksheets(SHEET_CASES)
    On Error GoTo 0

    If ws Is Nothing Then
        Set ws = ThisWorkbook.Worksheets.Add(After:=ThisWorkbook.Worksheets(ThisWorkbook.Worksheets.Co
unt))
        ws.Name = SHEET_CASES
    End If

    ' Headers if empty
    If ws.Cells(1, 1).Value = "" Then
        ws.Cells(1, 1).Value = "DateTime"
        ws.Cells(1, 2).Value = "CaseID"
        ws.Cells(1, 3).Value = "CaseType"
        ws.Cells(1, 4).Value = "Scenario"
        ws.Cells(1, 5).Value = "IssuingBody"
        ws.Cells(1, 6).Value = "DesiredOutcome"
        ws.Cells(1, 7).Value = "Priority"
        ws.Cells(1, 8).Value = "Status"
        ws.Cells(1, 9).Value = "Notes"
    End If
End Sub

Private Function Nz(ByVal v) As String
    If IsNull(v) Then
        Nz = ""
    Else
        Nz = CStr(v)
    End If
End Function




Private Sub CommandButton8_Click()

End Sub

Private Sub CommandButton9_Click()

End Sub



End Sub


End Sub


End Sub



End Sub

Private Sub ListBox1_Click()

End Sub

Private Sub ScrollBar1_Change()

End Sub


End Sub

P

End Sub

End Sub


End Sub

Private Sub UserForm_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByV
al Y As Single)

End Sub



End Sub



End Sub



End Sub



End Sub



End Sub


End Sub



End Sub



End Sub



End Sub

Private Sub MultiPage1_Change()

End Sub



End Sub
```

```
Private Sub SpinButton1_Change()
End Sub
Private Sub SpinButton2_Change()
End Sub
Private Sub TabStrip1_Change()
End Sub

End Sub

End Sub

End Sub

End Sub

End Sub
Private Sub TextBox18_Change()
End Sub

End Sub

End Sub
Private Sub TextBox22_Change()
End Sub

End Sub

End Sub

End Sub


Application.ScreenUpdating = False
Dim sDate As String
On Error Resume Next
sDate = MyCalendar.DatePicker(Me.txtDOB)
Me.txtDOB.Value = Format(sDate, "dd-mmm-yyyy")
On Error GoTo 0
```

```
Application.ScreenUpdating = True
End Sub


Private Sub imgCalendar_Click()

Application.ScreenUpdating = False

Dim sDate As String

On Error Resume Next

sDate = MyCalendar.DatePicker(Me.txtDOB)

Me.txtDOB.Value = Format(sDate, "dd-mmm-yyyy")
Sub Reset_Form()
Dim iRow As Long

With frmDataEntry

    .txtStudentName.text = ""
    .txtStudentName.BackColor = vbWhite

    .txtFatherName.text = ""
    .txtFatherName.BackColor = vbWhite

    .txtDOB.text = ""
    .txtDOB.BackColor = vbWhite

    .optFemale.Value = False
    .optMale.Value = False

    .txtMobile.Value = ""
    .txtMobile.BackColor = vbWhite

    .txtEmail.Value = ""
    .txtEmail.BackColor = vbWhite

    .txtAddress.Value = ""
    .txtAddress.BackColor = vbWhite

    .txtRowNumber.Value = ""
    .txtImagePath.Value = ""

    .imgStudent.Picture = LoadPicture(vbNullString)

    .cmdSubmit.Caption = "Submit"

    '.cmbCourse.Clear
    .cmbCourse.BackColor = vbWhite

    'Dynamic range based on Support Sheet
    shSupport.Range("A2", shSupport.Range("A" & Rows.Count).End(xlUp)).Name = "Dynamic"

    .cmbCourse.RowSource = "Dynamic"

    .cmbCourse.Value = ""

    .cmbCourse.Value = ""

    'Assigning RowSource to lstDatabase

    .lstDatabase.ColumnCount = 12
    .lstDatabase.ColumnHeads = True

    .lstDatabase.ColumnWidths = "30,70,70,40,45,70,60,60,70,0,0,0"

    iRow = shDatabase.Range("A" & Rows.Count).End(xlUp).row + 1 ' Identify last blank row

    If iRow > 1 Then

        .lstDatabase.RowSource = "Database!A2:L" & iRow
    Else

        .lstDatabase.RowSource = "Database!A2:L2"

    End If


End With
End Sub

On Error GoTo 0

Application.ScreenUpdating = True

 Set oRegEx = CreateObject("VBScript.RegExp")
 With oRegEx
    .Pattern = "^[\w-\.]{1,}\@([\da-zA-Z-]{1,}\.){1,}[\da-zA-Z-]{2,3}$"
    ValidEmail = .Test(Email)
 End With
 Set oRegEx = Nothing

GetImagePath = ""

With Application.FileDialog(msoFileDialogFilePicker) ' File Picker Dialog box

    .AllowMultiSelect = False
    .Filters.Clear      ' Clear the exisiting filters
    .Filters.Add "Images", "*.gif; *.jpg; *.jpeg" 'Add a filter that includes GIF and JPEG images

    ' show the file picker dialog box
    If .Show <> 0 Then

        GetImagePath = .SelectedItems(1) ' Getting the path of selected file name

    End If

End With
End Function
Sub CreateFolder()
Dim strFolder As String ' To hold the folter path where we need to replicate the image

strFolder = ThisWorkbook.Path & Application.PathSeparator & "Images"
'Check Directory exist or not. If not exist then it will return blank
     If Dir(strFolder, vbDirectory) = "" Then
         MkDir strFolder ' Make a folder with the name of 'Images'
     End If
End Sub



Sub LoadImange()
Dim imgSourcePath As String ' To store the path of image selected by user
Dim imgDestination As String ' To store the path of image selected by user

imgSourcePath = Trim(GetImagePath()) ' Call the Function

If imgSourcePath = "" Then Exit Sub

Call CreateFolder   'Create Image folder if not exist

imgDestination = ThisWorkbook.Path & Application.PathSeparator & _
frmDataEntry.txtStudentName & "." & Split(imgSourcePath, ".")(UBound(Split(imgSourcePath, ".")))

FileCopy imgSourcePath, imgDestination ' Code to copy image

frmDataEntry.imgStudent.PictureSizeMode = fmPictureSizeModeStretch 'Stretch mode
frmDataEntry.imgStudent.Picture = LoadPicture(imgDestination) ' Loading picture to imgStudent
frmDataEntry.txtImagePath.Value = imgDestination ' Assigning the path to text boxFunction ValidEntry()
 As Boolean

ValidEntry = True

With frmDataEntry
```

```
    'Default Color

    .txtStudentName.BackColor = vbWhite
    .txtFatherName.BackColor = vbWhite
    .txtDOB.BackColor = vbWhite
    .txtMobile.BackColor = vbWhite
    .txtEmail.BackColor = vbWhite
    .txtAddress.BackColor = vbWhite
    .cmbCourse.BackColor = vbWhite

    'Validating Student Name

    If Trim(.txtStudentName.Value) = "" Then
        MsgBox "Please enter Student's name.", vbOKOnly + vbInformation, "Student Name"
        .txtStudentName.BackColor = vbRed
        .txtStudentName.SetFocus
        ValidEntry = False
        Exit Function
    End If


    'Validating Father's name

    If Trim(.txtFatherName.Value) = "" Then
        MsgBox "Please enter Father's name.", vbOKOnly + vbInformation, "Father Name"
        .txtFatherName.BackColor = vbRed
        .txtFatherName.SetFocus
        ValidEntry = False
        Exit Function
    End If

    'Validating DOB

    If Trim(.txtDOB.Value) = "" Then
        MsgBox "DOB is blank. Please enter DOB.", vbOKOnly + vbInformation, "Invalid Entry"
        .txtDOB.BackColor = vbRed
        ValidEntry = False
        Exit Function
    End If


    'Validating Gender

    If .optFemale.Value = False And .optMale.Value = False Then
        MsgBox "Please select gender.", vbOKOnly + vbInformation, "Invalid Entry"
        ValidEntry = False
        Exit Function
    End If

    'Validating Course

    If Trim(.cmbCourse.Value) = "" Then
        MsgBox "Please select the Course from drop-down.", vbOKOnly + vbInformation, "Course Applied"
        .cmbCourse.BackColor = vbRed
        ValidEntry = False
        Exit Function
    End If

    'Validating Mobile Number

    If Trim(.txtMobile.Value) = "" Or Len(.txtMobile.Value) < 10 Or Not IsNumeric(.txtMobile.Value) Th
en
        MsgBox "Please enter a valid mobile number.", vbOKOnly + vbInformation, "Invalid Entry"
        .txtMobile.BackColor = vbRed
        .txtMobile.SetFocus
        ValidEntry = False
        Exit Function
    End If

    'Validating Email

    If ValidEmail(Trim(.txtEmail.Value)) = False Then
```

```
        MsgBox "Please enter a valid email address.", vbOKOnly + vbInformation, "Invalid Entry"
        .txtEmail.BackColor = vbRed
        .txtEmail.SetFocus
        ValidEntry = False
        Exit Function
    End If

    'Validating Address

    If Trim(.txtAddress.Value) = "" Then
        MsgBox "Address is blank. Please enter a valid address.", vbOKOnly + vbInformation, "Invalid E
ntry"
        .txtAddress.BackColor = vbRed
        ValidEntry = False
        Exit Function
    End If

    'Validating Image

    If .imgStudent.Picture Is Nothing Then
       MsgBox "Please upload the PP Size Photo.", vbOKOnly + vbInformation, "Picture"
        ValidEntry = False
        Exit Function
    End If

End With
End Function



Sub Submit_Data()
Dim iRow As Long

If frmDataEntry.txtRowNumber.Value = "" Then

 iRow = shDatabase.Range("A" & Rows.Count).End(xlUp).row + 1 ' Identify last blank row

Else
    iRow = frmDataEntry.txtRowNumber.Value

End If

With shDatabase.Range("A" & iRow)

.Offset(0, 0).Value = "=Row()-1" 'S. No.

.Offset(0, 1).Value = frmDataEntry.txtStudentName.Value 'Student's Name

.Offset(0, 2).Value = frmDataEntry.txtFatherName.Value    'Father's Name

.Offset(0, 3).Value = frmDataEntry.txtDOB.Value   'DOB

.Offset(0, 4).Value = IIf(frmDataEntry.optFemale.Value = True, "Female", "Male")  'Gender

.Offset(0, 5).Value = frmDataEntry.cmbCourse.Value    'Qualification

.Offset(0, 6).Value = frmDataEntry.txtMobile.Value    'Mobile Number

.Offset(0, 7).Value = frmDataEntry.txtEmail.Value     'Email

.Offset(0, 8).Value = frmDataEntry.txtAddress.Value   'Address

.Offset(0, 9).Value = frmDataEntry.txtImagePath.Value   'Photo

.Offset(0, 10).Value = Application.UserName    'Submitted By

.Offset(0, 11).Value = Format([Now()], "DD-MMM-YYYY HH:MM:SS")   'Submitted On

'Reset the form

Call Reset_Form

Application.ScreenUpdating = True
```

```
MsgBox "Data submitted successfully!"
End Sub



Function Selected_List() As Long
Dim i As Long
Selected_List = 0
If frmDataEntry.lstDatabase.ListCount = 1 Then Exit Function ' If no items exist in List Box
For i = 0 To frmDataEntry.lstDatabase.ListCount - 1
If frmDataEntry.lstDatabase.Selected(i) = True Then
   Selected_List = i + 1
   Exit For
End If
Next i
End Function
End Function

Sub Show_Form()
frmDataEntry.Show
End Sub

Private Sub cmdLoadImage_Click()
If Me.txtStudentName.Value = "" Then
 MsgBox "Please enter Student's first.", vbOKOnly + vbCritical, "Error"
 Exit Sub
End If

Call LoadImange
End Sub



Private Sub UserForm6_Initialize()
Call Reset_Form
End Sub



Private Sub cmdSubmit_Click()
Dim i As VbMsgBoxResult

i = MsgBox("Do you want to submit the data?", vbYesNo + vbQuestion, "Submit Data")

If i = vbNo Then Exit Sub

If ValidEntry Then

    Call Submit_Data

End If
End Sub


Private Sub cmdReset_Click()
Dim i As VbMsgBoxResult

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

If i = vbNo Then Exit Sub

Call Reset_Form
End Sub


Private Sub lstDatabase_DblClick(ByVal Cancel As MSForms.ReturnBoolean)

If Selected_List = 0 Then
     MsgBox "No row is selected.", vbOKOnly + vbInformation, "Edit"
     Exit Sub
End If
```

```
Dim sGender As String

'Me.txtRowNumber = Selected_List + 1 ' Assigning Selected Row Number of Database Sheet

Me.txtRowNumber = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 0) + 1

'Assigning the Selected Reocords to Form controls

frmDataEntry.txtStudentName.Value = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 1)

frmDataEntry.txtFatherName.Value = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 2)

frmDataEntry.txtDOB.Value = Format(Me.lstDatabase.List(Me.lstDatabase.ListIndex, 3), "dd-mmm-yyyy")

sGender = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 4)

If sGender = "Female" Then
    frmDataEntry.optFemale.Value = True
Else
    frmDataEntry.optMale.Value = True
End If

frmDataEntry.cmbCourse.Value = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 5)
frmDataEntry.txtMobile.Value = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 6)
frmDataEntry.txtEmail.Value = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 7)
frmDataEntry.txtAddress.Value = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 8)
frmDataEntry.imgStudent.Picture = LoadPicture(Me.lstDatabase.List(Me.lstDatabase.ListIndex, 9))
frmDataEntry.txtImagePath = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 9)
Me.cmdSubmit.Caption = "Update"
MsgBox "Please make the required changes and Click on Update."
End Sub



If Selected_List = 0 Then

     MsgBox "No row is selected.", vbOKOnly + vbInformation, "Delete"
     Exit Sub

End If

Dim i As VbMsgBoxResult

Dim row As Long

row = Me.lstDatabase.List(.lstDatabase.ListIndex, 0) + 1

i = MsgBox("Do you want ot delete the selected record?", vbYesNo + vbQuestion, "Delete")

If i = vbNo Then Exit Sub

ThisWorkbook.Sheets("Database").Rows(row).Delete

Call Reset ' Refresh the controls with latest information

MsgBox "Selected record has been successfully deleted.", vbOKOnly + vbInformation, "Delete"
End Sub


If Selected_List = 0 Then

     MsgBox "No row is selected.", vbOKOnly + vbInformation, "Edit"
     Exit Sub

End If

Dim sGender As String

Me.txtRowNumber = Me.lstDatabase.List(.lstDatabase.ListIndex, 0) + 1

'Assigning the Selected Reocords to Form controls

frmDataEntry.txtStudentName.Value = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 1)
frmDataEntry.txtFatherName.Value = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 2)

frmDataEntry.txtDOB.Value = Format(Me.lstDatabase.List(Me.lstDatabase.ListIndex, 3), "dd-mmm-yyyy")

sGender = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 4)

If sGender = "Female" Then
    frmDataEntry.optFemale.Value = True
Else
    frmDataEntry.optMale.Value = True
End If
frmDataEntry.cmbCourse.Value = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 5)
frmDataEntry.txtMobile.Value = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 6)
frmDataEntry.txtEmail.Value = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 7)
frmDataEntry.txtAddress.Value = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 8)
frmDataEntry.imgStudent.Picture = LoadPicture(Me.lstDatabase.List(Me.lstDatabase.ListIndex, 9))
frmDataEntry.txtImagePath = Me.lstDatabase.List(Me.lstDatabase.ListIndex, 9)
Me.cmdSubmit.Caption = "Update"
MsgBox "Please make the required changes and Click on Update."

    Select Case ComboBox1.Value
        Case "Electrical Simulation"
            Label1.Caption = "Domain: Electrical"
        Case "Portfolio Builder"
            Label1.Caption = "Domain: Portfolio"
        Case "Rubric Mapping"
            Label1.Caption = "Domain: Rubric"
    End Select
End Sub

    Label2.Caption = "Rubric Level: Intermediate"
End Sub


    If ComboBox3.Value = "" Or ComboBox4.Value = "" Then
        MsgBox "Please select all required rubric parameters.", vbExclamation
        Exit Sub
    End If
    Label3.Caption = "Simulation executed successfully."
End Sub

    MsgBox "Credential artifact generated and submitted.", vbInformation
End Sub
()
    Label5.Caption = "Rubric template loaded: " & ListBox1.Value
End Sub
()
    MsgBox "Ensure rubric alignment with SAQA/NQF thresholds.", vbInformation
End Sub




End Subtshingombe fiston
```

Jul 23, 2025, 3:10 PM (2 days ago)

to me
Qeios
Peer-approved Preprints Archive

- About
- Ethics
- Plans
- Sign Up Free
- Log in

Views

4,047

Downloads

314
Peer Reviewers

29
Citations

0
Article has an altmetric score of 2
Make Action
PDF
Field

Computer Science
Subfield

Information Systems
Open Peer Review
Preprint

2.79 | 29 peer reviewers
Research Article Dec 11, 2023
https://doi.org/10.32388/JGU5FH

Web-Based Crime Management System for Samara City Main Police Station

Demelash Lemmi Ettisa1, Minota Milkias2

Abstract

Crime is a human experience, and it must be controlled. The Samara town police station plays a significant role in controlling crime. However, the management of crime activities is done manually, which is due to the lack of an automated system that supports the station workers in communicating with citizens to share information and store, retrieve, and manage crime activities. To control crime efficiently, we need to develop online crime management systems.

This project, entitled "Web-Based Crime Management System," is designed to develop an online application in which any citizen can report crimes; if anybody wants to file a complaint against crimes, they must enjoy online communication with the police. This project provides records of crimes that have led to disciplinary cases in addition to being used to simply retrieve information from the database. The system implemented is a typical web-based crime record management system based on client-server architecture, allowing data storage and crime record interchange with police stations.

Corresponding author: Demelash Lemmi Ettisa, nicemanyes@su.edu.et

Chapter One
1. Introduction to the Study

The "Crime Management System" is a web-based website for online complaining and computerized management of crime records (Khan et al., 2008).

A criminal is a popular term used for a person who has committed a crime or has been legally convicted of a crime. "Criminal" also means being connected with a crime. When certain acts or people are involved in or related to a crime, they are termed as criminal (Wex, 2023).

Samara City 's main police station is located in Samara City, within the Afar Regional State. It was established in 1984 E.C. with the purpose of protecting local communities from criminal activities. The Samara City police station is situated near the diesel suppliers in Samara City. In the first phase, there was a small number of police members, including commanders, inspectors, and constables. But recently, more than 170 police members have been employed. It is a well-organized police station that serves in crime prevention; the detection and conviction of criminals depend on a highly responsive manner. The effectiveness of this station is based on how efficient, reliable, and fast it is. As a consequence, the station maintains a large volume of information. To manage their information requirements, the station is currently using an information system. This system is manual and paper-based, where information is passed hand-to-hand, and information is kept in hard-copy paper files stored ordinarily in fili

```
End Sub

    MsgBox
    ' Trigger Python backend for signal acquisition
    Shell "python capture_signal.py", vbNormalFocus
End Sub


    MsgBox "Running Full Diagnostic..."
    ' Trigger full algorigramme pipeline
    Shell "python run_diagnostic.py", vbNormalFocus
End Sub


    ' Update SNR threshold
    Dim snrThreshold As Double
    snrThreshold = Val(TextBox2.text)
    ' Store or pass to backend
End Sub


    MsgBox "Fourier Transform Module"
End Sub


    MsgBox "SNR Evaluation Module"
End Sub


    MsgBox "Bandwidth Classification"
End Sub


    MsgBox "Linearity Check"
End Sub




Sub tsh()

End Sub
Sub she1()

End Sub
Sub wsh1()

End Sub
Sub wb2()

End Sub
Sub wb1()

End Sub
Sub wb4()

End Sub
```