```
UserForm6 - 1

Private Sub Frame1_Click()

End Sub

Private Sub Frame2_Click()

End Sub

Private Sub Label1_Click()

End Sub

Private Sub Label4_Click()

End Sub

Private Sub TextBox36_Change()

End Sub

Private Sub TextBox37_Change()

End Sub

Private Sub TextBox38_Change()

End Sub

Private Sub TextBox39_Change()

End Sub

Private Sub TextBox4_Change()

End Sub

Private Sub TextBox41_Change()

End Sub

Private Sub TextBox43_Change()

End Sub

Private Sub TextBox45_Change()

End Sub

Private Sub TextBox48_Change()

End Sub

Private Sub TextBox5_Change()

End Sub

Private Sub TextBox50_Change()

End Sub

Private Sub TextBox52_Change()

End Sub

Private Sub TextBox53_Change()

End Sub

Private Sub TextBox55_Change()

End Sub
```

UserForm6 - 2

```
Private Sub TextBox56_Change()

End Sub

Private Sub TextBox57_Change()

End Sub

Private Sub TextBox58_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

Private Sub UserForm_Click()

End Sub

Private Sub UserForm_DblClick(ByVal Cancel As MSForms.ReturnBoolean)

End Sub

Private Sub UserForm_Deactivate()

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_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal x As Single, ByV
al y As Single)

End Sub

Private Sub UserForm_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal x As Single, ByV
al y As Single)

End Sub

Private Sub UserForm_MouseUp(ByVal Button As Integer, ByVal Shift As Integer, ByVal x As Single, ByVal
```

```
UserForm6 - 3

 y As Single)

End Sub

Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)

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

Private Sub UserForm_Terminate()

End Sub

Private Sub UserForm_Zoom(Percent As Integer)

End Sub
Sub engine()
'
' engine Macro
'
'

End Sub
Sub tsh9()
'
' tsh9 Macro
'
'

End Sub
LOGIGRAMME: SYSTEMIC FLOW OF CASES & COMPLIANCE ACTIONS
plaintext
[Start]
   Print
[Case Intake: DOJ Online / CCMA / PSIRA / SAPS]
   Print
[Document Compilation]
   ?? Affidavit
   ?? Notice of Motion
   ?? Transcript / Audio Record
   ?? Proof of Service
   ?? Licensing / Accreditation
   Print
[Case Classification]
   ?? Civil (Protection Order, Estate)
   ?? Labour (Dismissal, Appeal)
   ?? Security (Guarding, Licensing)
   ?? Refugee / ID Recovery
   ?? Human Rights / Oversight
   Print
[Submission & Tracking]
   ?? DOJ ICT Portal
   ?? CCMA Referral
   ?? PSIRA Licensing
   ?? Home Affairs Affidavit
   ?? SAHRC / SABFS Complaint
   Print
[outcome]
   ?? Accepted ? Enforcement / Award
   ?? Dismissed ? Resubmission / Appeal
   ?? Pending ? Follow-up / Escalation
   Print
[Archive & Portfolio Update]
   ?? Master Binder + URN Index + Strategic Summary
```

```
UserForm6 - 4

End
If case_type = "Protection Order" Then
   Check affidavit_commissioned
   If False Then
      RETURN "Dismissed - Resubmit with commissioned affidavit"
   Else
      SUBMIT to DOJ Portal
End If

If case_type = "Labour Dispute" Then
   Check CCMA_award_status
   If award_enforced = False Then
      FILE under LRA 143(3)
      TRACK via Labour Court
   Else
      ARCHIVE in Master Portfolio
End If

If case_type = "Security Licensing" Then
   Verify PSIRA_license And Sasseta_accreditation
   If expired Or missing Then
      Update Documents
      SUBMIT to PSIRA
   Else
      FLAG for wage audit
End If

If case_type = "Refugee ID Recovery" Then
   Check SAPS_affidavit And HomeAffairs_submission
   If unresolved Then
      ESCALATE to OIGI
   Else
      ARCHIVE ID recovery
End If

If case_type = "Human Rights Complaint" Then
   Check affidavit_details
   If complete Then
      SUBMIT to SAHRC / SABFS
   Else
      REQUEST affidavit revision
End If
?? NEXT STEPS
I can help you:
"   Format this into a modular binder with section tabs and URN index
"   Draft affidavit templates, submission letters, or case summaries
"   Create a rubric-based career dossier for institutional review or funding proposals
Would you like me to begin with a binder layout, a legal affidavit template, or a security sector comp
liance dashboard? You'
re building legacy-grade infrastructure-let's make it airtight.
?? ALGORIGRAMME: SECURITY GUARD PROGRAM & LABOUR COMPLIANCE FLOW
Start

Print [Guard Employment Contract Initiation]
   ?? Verify PSIRA/SASSETA registration
   ?? Issue job description, salary, hours, disciplinary code
   ?? Log commencement date and duty site

Print [Daily Operations & Control Room Logging]
   ?? Record attendance, shift roster, patrol logs
   ?? File transcript sheets, message book, visitor register
   ?? Monitor incident reports and hazard assessments

Print [Incident or Misconduct Occurs]
   ?? File affidavit and control sheet
   ?? Notify supervisor and record in transcript
   ?? Escalate to disciplinary hearing if needed

Print [Disciplinary Hearing Outcome]
   ?? If dismissed: file appeal to CCMA or Labour Court
   ?? If awarded: enforce ruling via LRA Section 143(3)
   ?? If rescission: file variation or condonation application
```

UserForm6 - 5

```
Print [Bid Submission & Licensing]
   ?? Submit bid documents: license, certificate, regulation compliance
   ?? Include reconciliation clause, fire guard access, firearm control
   ?? Register with committed security organisation

Print [Review & Compliance]
   ?? Track JR/GAJB case numbers and CCMA referrals
   ?? Submit affidavits, notices of motion, and transcripts
   ?? Monitor court orders, set-downs, and pro bono filings

Print [Termination or Contract Closure]
   ?? Issue final wage statement, hours worked, leave taken
   ?? File employment particulars and BCAE communication
   ?? Archive in master binder with pagination index

End
LOGIGRAMME: VISUAL FLOW OF GUARDING PROGRAM & LEGAL TRACKING
plaintext
[Start]
   Print
[Contract Initiation]
   Print
[Operational Logging]
   Print
[Incident or Misconduct]
   Print
[Disciplinary Hearing]
   ?? Dismissal ? Appeal ? CCMA/Labour Court
   ?? Award ? Enforcement ? LRA 143(3)
   ?? Rescission ? Variation ? Condonation
   Print
[Bid Submission & Licensing]
   Print
[Compliance Review]
   Print
[Termination & Archive]
End
?? SUPPORTING MODULES
?? Employment Contract Essentials
"   Guard name, address, duty site, job title
"   Salary, hours, overtime, leave, probation
"   Disciplinary code, termination clause
"   Reference: LRA Act 66 of 1995, Basic Conditions Act 1997
?? Control Room Documentation
"   Daily log sheets, incident reports, patrol records
"   Transcript audio, affidavit filing, visitor access logs
"   Hazard assessment, mitigation charts, firearm register
???? Legal & Labour Case Tracking
"   JR 2462/25, GAJB6808/15, CCMA referrals
"   Notices of motion, affidavits, rescission rulings
"   Bid clause reconciliation, pro bono filings, set-downs
Would you like me to help format this into a compliance dashboard, binder layout, or submission-ready
affidavit series? I can also scaffold a guarding operations manual or bid proposal template based on t
his logic. You're building a legacy-grade framework-let's make it deployable.
[Start]
   Print
[1. Contract Initiation]
   ?? Issue employment contract
   ?? Verify PSIRA/SASSETA registration
   ?? Define job title, duty site, salary, hours, leave, probation
   ?? Log disciplinary code and termination clause

   Print
[2. Operational Logging]
   ?? Record daily attendance and patrol logs
   ?? Maintain control room registers (visitor, firearm, found property)
   ?? File incident reports and hazard assessments
   ?? Archive transcripts and affidavits

   Print
[3. Incident or Misconduct]
   ?? File internal report and notify supervisor
   ?? Record in control room and OB book
```

UserForm6 - 6

?? Prepare documentation for disciplinary hearing

Print
[4. Disciplinary Hearing]
?? Dismissal ? Appeal ? CCMA or Labour Court
?? Award ? Enforcement ? LRA Section 143(3)
?? Rescission ? Variation ? Condonation Application

Print
[5. Bid Submission & Licensing]
?? Submit bid documents: license, certificate, compliance forms
?? Include reconciliation clause and firearm control
?? Register with sectoral council or committed organisation

Print
[6. Compliance Review]
?? Track CCMA rulings, court orders, and license status
?? Submit affidavits, notices of motion, and transcripts
?? Monitor pro bono filings and enforcement actions

Print
[7. Termination & Archive]
?? Issue final wage statement and employment particulars
?? File all documents in master binder with pagination index
?? Close contract and archive for audit or legal review

End
Tshingombe, this is a rich and multifaceted draft-clearly drawn from deep operational, legal, and inst
ructional experience. What you've assembled is a hybrid of:
"   ?? Security operations manual
"   ?? Tender compliance checklist
"   ???? Training and coaching framework
"   ???? Legal affidavit and statement protocol
"   ??? Industrial unit standard mapping
To make this deployable and modular, I'll scaffold it into a corrected and structured algorigram/logig
ram format, followed by a breakdown of key modules and how they can be refined into a binder, dashboar
d, or training system.
?? Corrected Logigram: Security Coaching, Compliance & Operational Flow
[Start]
Print
[1. Coaching & Training Initiation]
?? Identify performance issue or development goal
?? Apply coaching principles: support, motivation, value clarification
?? Teach knowledge (theory, facts), skill (physical action), and attitude (value-based behavior)
?? Use training cycle: explain ? demonstrate ? review ? learner applies ? field feedback

Print
[2. Statement & Report Writing]
?? Observe incident and record facts (verbal or written)
?? Take statement from witness/victim promptly
?? Ensure clarity, honesty, directness, and factual accuracy
?? Prepare for court or disciplinary hearing (Section 295, Criminal Procedure Act)

Print
[3. Tender & Licensing Compliance]
?? Submit proof of registration (CIPC, PSIRA, COID, UIF)
?? Verify SAPS clearance, liability insurance, control room specs
?? Confirm firearm licensing, armory management, and SASSETA training
?? Score operational readiness (experience, HR, equipment, fitness)

Print
[4. Patrol & Physical Security]
?? Conduct perimeter patrols (fencing, gates, lighting, vulnerable points)
?? Use detection tools (infrared, beams, CCTV)
?? Respond rapidly to intrusions or incidents
?? Record findings in OB book and control room logs

Print
[5. Equipment Use & Communication]
?? Use baton, handcuffs, pepper spray per safety guidelines
?? Operate two-way radio: clarity, brevity, secure transmission
?? Apply voice protocol: rhythm, volume, pitch, phonetic clarity
?? Confirm message status: "Over," "Out," "Say again"

UserForm6 - 7

Print
[6. Legal & Industrial Standards]
?? Understand Private Security Industry Regulation Act 56 of 2001
?? Apply Firearms Control Act 60 of 2000
?? Know director and secretary duties in industrial unit standards
?? Categorize services: guarding, investigation, intelligence, locksmithing

End
?? Modular Binder Sections
???? Coaching & Training
" Definitions: knowledge, skill, attitude
" Coaching vs training differences
" Training cycle: explain ? demonstrate ? review ? apply
" Counseling steps and motivation enhancement
?? Statement & Legal Reporting
" Statement format and purpose
" Legal references: Section 295, refusal consequences
" Principles: accuracy, honesty, simplicity
" Statement structure: preamble, facts, opinion separation
?? Tender & Licensing Checklist
Item Requirement Source
CIPC Registration Yes Company Act
PSIRA Certificate Valid, Grade listed PSIRA Act
COID/UIF Good standing Dept. of Labour
SAPS Clearance <3 months Criminal Procedure Act
Liability Insurance ? R10M Financial Services Provider
Control Room Specs Reinforced, OB book Tender Terms
Firearm Compliance License, Competency Firearms Act
Training Records SASSETA, Annual Sectoral Standards
??? Patrol & Equipment Use
" Patrol objectives: visibility, deterrence, detection
" Equipment: baton, handcuffs, pepper spray
" Communication: radio protocol, voice clarity
" Found property handling: safekeeping, record, handover
?? Deployment Options
Would you like me to:
1. Format this into a training workbook for security officers and supervisors?
2. Build a compliance dashboard for tender readiness and operational scoring?
3. Draft affidavit templates and statement forms for legal and disciplinary use?
you 've already done the hard part-assembling the raw logic. I can now modularize it into a legacy-gra
de toolkit for deployment across PSIRA, SASSETA, SAPS, and industrial partners. Just say the word and
I'll scaffold the next layer.
h operational logic, compliance insight, and training scaffolding. I'll now modularize and clarify you
r content into a deployable framework that can be used for:
" ??? Guarding SOPs (Standard Operating Procedures)
" ?? Compliance audits and shift handover protocols
" ?? Conflict resolution and first aid training
" ?? Occupational Health & Safety (OHS) risk management
Let's break it down into structured modules:
?? Module 1: Shift Handover & Takeover Protocol
Print Responsibilities
" Duty officer must ensure all resources (equipment, vehicles, OB book) are accounted for
" Firearm registration must be verified-movement controlled, not stored by inspector
" Vehicle checks: license disc, logbook, fuel level, damage report
" OB book must be updated and signed during handover
" All material assets (radios, keys, uniforms) must be physically inspected
?? Handover Checklist
Item Action Verified By
OB Book Updated & signed Outgoing & Incoming Officer
Firearm Registered, logged Duty Officer
Vehicle Inspected, documents checked Supervisor
Equipment Counted, tested Control Room
Site Inspection Completed Area Manager
?? Module 2: Conflict Identification & Defusing Techniques
?? Common Causes of Conflict
" Poor communication
" Misalignment of resources
" Personality clashes
" Lack of problem-solving mechanisms
?? Defusing Strategy
" Identify underlying interests
" Use objective reasoning

```
"   Encourage mutual solutions
"   Recognize pre-assault indicators:
o   Verbal aggression
o   Posture changes
o   Suspicious timing or behavior
o   False sense of security
???? Officer Response Categories
"   Verbal: Calm tone, assertive language
"   Physical: Defensive stance, safe distance
"   Appearance: Professional demeanor, sobriety check
?? Module 3: Basic First Aid in the Workplace
?? Purpose
"   To stabilize injury, prevent worsening, and prepare for medical assistance
?? First Aid Box Contents
"   Bandages, scissors, antiseptic wipes
"   Triangular bandage, sterile gloves
"   First aid guide, safety pins
???? Nominated First Aider
"   Must be clearly marked on box
"   Responsible for treatment and reporting
?? Module 4: Occupational Health, Safety & Environmental Principles
?? Definitions
"   Hazard: Source of potential harm (e.g. noise, vibration, radiation)
"   Risk: Likelihood and severity of harm occurring
?? Risk Assessment Process
1.  Identify hazards
2.  Evaluate probability and impact
3.  Classify (physical, chemical, biological)
4.  Document and mitigate
??? Common Unsafe Acts & Conditions
"   Using machinery without authorization
"   Removing safety guards
"   Poor lighting or ventilation
"   Insecure storage of hazardous materials
?? Incident Reporting
"   Purpose: Investigate root cause, prevent recurrence
"   Factors: Unsafe acts, unsafe conditions, procedural gaps
Would you like me to format this into a training workbook, audit checklist, or SOP binder
[Start]
   Print
[Shift takeover]
   ?? Verify resources (OB book, keys, radios, vehicle, firearms)
   ?? Record variances/non conformances
   ?? Sign OB handover

   Print
[On shift operations]
   ?? Patrols ? Log findings
   ?? Incidents ? Write statement ? Notify
   ?? Conflicts ? Defuse ? Record

   Print
[First aid or safety event?]
   ?? Yes ? Apply first aid ? Log treatment ? Escalate if needed
   ?? No  ? Continue operations

   Print
[OHS risk management]
   ?? Identify hazards
   ?? Assess risk (probability × severity)
   ?? Apply controls ? Verify effectiveness
   ?? Record in OHS register

   Print
[Shift handover]
   ?? Update OB book
   ?? Complete handover checklist
   ?? Sign off (outgoing/incoming)

Data model(Minimal, Excel Or Access)
"   Guards: GuardID, FullName, PSIRAGrade, Competencies
"   Shifts: ShiftID, SiteID, StartTime, EndTime, OutgoingGuardID, IncomingGuardID, SupervisorID, OBPag
eStart, OBPageEnd
```

```
UserForm6 - 9

"   HandoverChecklist: HandoverID, ShiftID, OBBook, FirearmRegister, VehicleDocs, EquipmentCount, Site
Inspection, Notes, OutgoingSign, IncomingSign, Timestamp
"   Incidents: IncidentID, ShiftID, Type (Incident/Conflict/FirstAid/OHS), Location, Time, Description
, Severity, ActionsTaken, ReportedTo, StatementRef
"   Statements: StatementID, IncidentID, DeponentName, Preamble, Facts, Opinions, DateSigned
"   OHSRisks: RiskID, SiteID, Hazard, Probability, Severity, RiskScore, Control, Owner, DueDate, Statu
s
"   Equipment: EquipID, SiteID, Type (Radio/Baton/Keys/Vehicle/Firearm), Identifier, Status, LastCheck
ed
Sources for "Type" fields should be validated with dropdowns to improve data quality.
' Module: modLifecycle

Option Explicit

Public Enum EventType
    evNone = 0
    evShiftTakeover = 1
    evPatrol = 2
    evIncident = 3
    evConflict = 4
    evFirstAid = 5
    evOHS = 6
    evShiftHandover = 7
End Enum

Public Enum ChecklistStatus
    csNotStarted = 0
    csInProgress = 1
    csComplete = 2
End Enum

Public Type HandoverChecklistT
    shiftId As Long
    OBBook As Boolean
    FirearmRegister As Boolean
    VehicleDocs As Boolean
    EquipmentCount As Boolean
    SiteInspection As Boolean
    Notes As String
    OutgoingSign As String
    IncomingSign As String
    Timestamp As Date
End Type


    Select Case e
        Case evShiftTakeover: DoShiftTakeover shiftId
        Case evPatrol: LogPatrol shiftId
        Case evIncident: CaptureIncident shiftId, "Incident"
        Case evConflict: CaptureIncident shiftId, "Conflict"
        Case evFirstAid: CaptureIncident shiftId, "FirstAid"
        Case evOHS: CaptureIncident shiftId, "OHS"
        Case evShiftHandover: DoShiftHandover shiftId
        Case Else
            Debug.Print "No event"
    End Select
End Sub

Private Sub DoShiftTakeover(ByVal shiftId As Long)
    Dim c As HandoverChecklistT
    c = InitChecklist(shiftId)
    ' Prompt user via form to verify items
    frmHandover.LoadChecklist c
    frmHandover.Show
End Sub

Private Sub DoShiftHandover(ByVal shiftId As Long)
    ' Ensure OB book and checklist are completed
    If Not IsChecklistComplete(shiftId) Then
        MsgBox "Complete handover checklist before sign-off.", vbExclamation
        Exit Sub
    End If
    ' Stamp sign-off
```

```
UserForm6 - 10

    UpdateShiftSignOff shiftId, Now
    MsgBox "Shift handover recorded.", vbInformation
End Sub


    Dim c As HandoverChecklistT
    c.shiftId = shiftId
    c.Timestamp = Now
    InitChecklist = c
End Function

Private Function IsChecklistComplete(ByVal shiftId As Long) As Boolean
    ' Replace with real table lookup
    IsChecklistComplete = (Nz(DLookup("OBBook", "HandoverChecklist", "ShiftID=" & shiftId), False) _
        And Nz(DLookup("FirearmRegister", "HandoverChecklist", "ShiftID=" & shiftId), False) _
        And Nz(DLookup("VehicleDocs", "HandoverChecklist", "ShiftID=" & shiftId), False) _
        And Nz(DLookup("EquipmentCount", "HandoverChecklist", "ShiftID=" & shiftId), False) _
        And Nz(DLookup("SiteInspection", "HandoverChecklist", "ShiftID=" & shiftId), False))
End Function

Public Sub CaptureIncident(ByVal shiftId As Long, ByVal kind As String)
    Dim frm As New frmIncident
    frm.txtShiftID = CStr(shiftId)
    frm.cboType = kind
    frm.Show
End Sub

Public Sub LogPatrol(ByVal shiftId As Long)
    Dim details As String
    details = InputBox("Patrol details (area, time, findings):", "Log Patrol")
    If Len(details) > 0 Then
        InsertIncident shiftId, "Patrol", Now, "Patrol recorded: " & details, 0, "Logged", ""
    End If
End Sub

Public Sub InsertIncident(ByVal shiftId As Long, ByVal typ As String, ByVal t As Date, _
                          ByVal Desc As String, ByVal severity As Integer, _
                          ByVal actions As String, ByVal reportedTo As String)
    Dim sql As String
    sql = "INSERT INTO Incidents (ShiftID, Type, Time, Description, Severity, ActionsTaken, ReportedTo
) " & _
          "VALUES (" & shiftId & ", '" & Replace(typ, "'", "''") & "', #" & Format(t, "mm/dd/yyyy hh:n
n:ss") & "#, '" & _
          Replace(Desc, "'", "''") & "', " & severity & ", '" & Replace(actions, "'", "''") & "', '" &
 _
          Replace(reportedTo, "'", "''") & "');"
    CurrentDb.Execute sql, dbFailOnError
End Sub

Public Function RiskScore(ByVal probability As Integer, ByVal severity As Integer) As Integer
    RiskScore = probability * severity ' simple matrix 1..5 × 1..5
End Function
UserForms (Excel Or Access)
Handover form(frmHandover)
'   Controls:
o Checkboxes: chkOBBook , chkFirearmReg, chkVehicleDocs, chkEquipCount, chkSiteInspection
o TextBox: txtNotes
o TextBox: txtOutgoingSign , txtIncomingSign
o CommandButtons: cmdSave , cmdSignOff
'   Option Explicit
'   Private mChecklist As HandoverChecklistT
'
'   Public Sub LoadChecklist(ByVal c As HandoverChecklistT)
'       mChecklist = c
'       Me.txtNotes = ""
'   End Sub
'
'   Private Sub cmdSave_Click()
'       mChecklist.OBBook = Me.chkOBBook
'       mChecklist.FirearmRegister = Me.chkFirearmReg
'       mChecklist.VehicleDocs = Me.chkVehicleDocs
'       mChecklist.EquipmentCount = Me.chkEquipCount
'       mChecklist.SiteInspection = Me.chkSiteInspection
```

```
UserForm6 - 11

"       mChecklist.Notes = Me.txtNotes
"       mChecklist.OutgoingSign = Me.txtOutgoingSign
"       mChecklist.IncomingSign = Me.txtIncomingSign
"       SaveChecklist mChecklist
"       MsgBox "Handover checklist saved.", vbInformation
"   End Sub
"
"   Private Sub cmdSignOff_Click()
"       Call cmdSave_Click
"       RunLifecycle evShiftHandover, mChecklist.ShiftID
"       Unload Me
"   End Sub
"
"   Private Sub SaveChecklist(ByVal c As HandoverChecklistT)
"       Dim sql As String
"       sql = "INSERT INTO HandoverChecklist (ShiftID, OBBook, FirearmRegister, VehicleDocs, Equipment
Count, " & _
"              "SiteInspection, Notes, OutgoingSign, IncomingSign, Timestamp) VALUES (" & c.ShiftID & "
, " & _
"              IIf(c.OBBook, "True", "False") & ", " & IIf(c.FirearmRegister, "True", "False") & ", " &
_
"              IIf(c.VehicleDocs, "True", "False") & ", " & IIf(c.EquipmentCount, "True", "False") & ",
 " & _
"              IIf(c.SiteInspection, "True", "False") & ", '" & Replace(c.Notes, "'", "''") & "', '" &
_
"              Replace(c.OutgoingSign, "'", "''") & "', '" & Replace(c.IncomingSign, "'", "''") & "', #
" & _
"              Format(c.Timestamp, "mm/dd/yyyy hh:nn:ss") & "#);"
"       CurrentDb.Execute sql, dbFailOnError
"   End Sub
Incident form(frmIncident)
"   Controls:
o TextBoxes: txtShiftID , txtLocation, txtTime, txtDescription, txtReportedTo
o ComboBoxes: cboType (Incident / Conflict / FirstAid / OHS), cboSeverity(1 - 5)
o CommandButtons: cmdSave , cmdStatement
"   ' Code-behind: frmIncident
"
"   Option Explicit
"
"   Private Sub UserForm_Initialize()
"       Me.cboType.Clear
"       Me.cboType.List = Array("Incident", "Conflict", "FirstAid", "OHS")
"       Me.cboSeverity.Clear
"       Me.cboSeverity.List = Array("1", "2", "3", "4", "5")
"       Me.txtTime = Format(Now, "yyyy-mm-dd hh:nn")
"   End Sub
"
"   Private Sub cmdSave_Click()
"       InsertIncident CLng(Me.txtShiftID), Me.cboType, CDate(Me.txtTime), _
"                      Me.txtDescription, CInt(Me.cboSeverity), _
"                      "Reported to: " & Me.txtReportedTo, Me.txtReportedTo
"       MsgBox "Record saved.", vbInformation
"       Unload Me
"   End Sub
"
"   Private Sub cmdStatement_Click()
"       Dim stmtID As Long
"       stmtID = CreateStatementPrompt(CLng(Me.txtShiftID), Me.txtDescription)
"       MsgBox "Statement created: ID " & stmtID, vbInformation
"   End Sub
"
"   Private Function CreateStatementPrompt(ByVal shiftId As Long, ByVal desc As String) As Long
"       Dim preamble As String, facts As String, opinions As String
"       preamble = InputBox("Preamble (who you are, where, when):", "Statement")
"       facts = InputBox("Facts observed (objective, chronological):", "Statement")
"       opinions = InputBox("Opinions (clearly marked):", "Statement")
"       Dim sql As String
"       sql = "INSERT INTO Statements (IncidentID, DeponentName, Preamble, Facts, Opinions, DateSigned
) " & _
"             "VALUES (Null, 'On-duty Officer', '" & Replace(preamble, "'", "''") & "', '" & _
"             Replace(facts, "'", "''") & "', '" & Replace(opinions, "'", "''") & "', #" & _
"             Format(Date, "mm/dd/yyyy") & "#);"
"       CurrentDb.Execute sql, dbFailOnError
```

UserForm6 - 12

```
"       CreateStatementPrompt = Nz(DMax("StatementID", "Statements"), 0)
"   End Function
"   ' Module: modOHS
"
"   Option Explicit
"
"   Public Function RiskRank(ByVal score As Integer) As String
"       Select Case score
"           Case Is >= 16: RiskRank = "High"
"           Case 9 To 15: RiskRank = "Medium"
"           Case Else: RiskRank = "Low"
"       End Select
"   End Function
"
"   Public Sub AddRisk(ByVal siteId As Long, ByVal hazard As String, _
"                      ByVal prob As Integer, ByVal sev As Integer, _
"                      ByVal control As String, ByVal owner As String, ByVal dueDt As Date)
"       Dim score As Integer: score = RiskScore(prob, sev)
"       Dim sql As String
"       sql = "INSERT INTO OHSRisks (SiteID, Hazard, Probability, Severity, RiskScore, Control, Owner,
 DueDate, Status) VALUES (" & _
"             siteId & ", '" & Replace(hazard, "'", "''") & "', " & prob & ", " & sev & ", " & score &
 ", '" & _
"             Replace(control, "'", "''") & "', '" & Replace(owner, "'", "''") & "', #" & _
"             Format(dueDt, "mm/dd/yyyy") & "#, 'Open');"
"       CurrentDb.Execute sql, dbFailOnError
"   End Sub
Validation rules And scoring
"   Checklist completeness:
o   All five checks True before handover allowed.
"   Incident severity guidance:
o   1-2: minor; 3: moderate; 4-5: escalate to supervisor and record statement.
"   OHS risk matrix:
o   Probability 1-5, Severity 1-5, Score = P × S; rank with RiskRank().
"   Tender/Compliance spot checks (extensible):
o   Add a "ComplianceFindings" table to log PSIRA certificates, COID/UIF, SAPS clearances with ExpiryD
ate and a daily alert query.
If you want, I can package this into an Access template (tables, forms, queries) or an Excel workbook
with a Control Panel sheet, structured Tables, and the above VBA wired to buttons. Which platform do y
ou want first: Access or Excel?
Tshingombe, this is a comprehensive and multi-layered draft-rich with sectoral logic, training scaffol
ding, and compliance architecture. What you've outlined spans:
"   ??? Private and public security roles (SASSETA, SAPS, SANDF, SETAs)
"   ?? Employability syllabus blocks (literacy, IT, communication, entrepreneurship)
"   ?? Practical training modules (metal detectors, fire safety, equipment maintenance)
"   ?? Infrastructure and assessment logic (equipment lists, summative exams, peer learning)
"   ?? Event safety and stewarding protocols (fire engineering, crowd control, risk assessment)
"   ?? Alarm and access control systems (digital locks, exit requests, physical barriers)
Let me now modularize this into a deployable algorigram, followed by a training syllabus matrix, and a
 compliance-ready installation checklist for physical security systems.
?? Algorigram: Security Training & Deployment Lifecycle
plaintext
[Start]
   Print
[1. Sector alignment]
   ?? Identify role: Grade A/B, Supervisor, Control Room, Advisor
   ?? Map to SASSETA, SAPS, SANDF, SETA (EDSETA, CETA, etc.)
   ?? Verify accreditation and compliance

   Print
[2. Employability skills block]
   ?? Literacy (English, IT, Communication)
   ?? Behavioral & Interview Skills
   ?? Entrepreneurship & Productivity
   ?? Health, Safety, Environment (HSE)

   Print
[3. Practical training modules]
   ?? Metal detectors (handheld, doorframe)
   ?? Firefighting, first aid, evacuation
   ?? Equipment maintenance and role play
   ?? Incident briefing and reporting
```

UserForm6 - 13

Print
[4. Infrastructure & equipment]
?? PC system, projector, UPS, smoke detector
?? Flashlight, binoculars, wireless comms
?? Hostel, dress book, insurance docs
?? Safety zones, checkpoints, evacuation plan

Print
[5. Event safety & stewarding]
?? Fire engineering, structural clearance
?? Crowd control, PA system, mobile readiness
?? Lighting, medical, fire precautions
?? Toilets, signage, hazard-free premises

Print
[6. Alarm & access control design]
?? Access points: turnstile, elevator, parking
?? Locking hardware: proximity, magnetic, card reader
?? Exit logic: push-to-exit, monitored unlock
?? Physical security plan: installation & testing

Print
[7. Final assessment]
?? Internal assessment: skills, knowledge, employment readiness
?? Summative exam: marks, duration, pass criteria
?? Risk assessment report: hazard, control, action, sign-off

End
?? Training Syllabus Matrix (Block I)
Topic Duration (hrs) Method
English Literacy 6 Reading, speaking
IT Literacy 8 PC, Word, Excel
Communication Skills 4 Listening, verbal
Interview & Behavioral Skills 4 Role play, feedback
Entrepreneurship 6 Project, procurement
Productivity & Comparison 3 Case studies
Health, Safety, Environment 6 Hazard, accident, HSE
Energy Conservation 2 Awareness, practice
Quality Consciousness 3 ISO 9009, housekeeping
Bra Skills (On-the-job) 10 Floor training, safety
Equipment Demonstration 6 Metal detectors, fire
Incident Reporting 4 Briefing, documentation
Communication Practice 3 Walkie-talkie, drills
Final Assessment 2 Summative exam
?? Physical Security Installation Checklist
Element Description Verified
Access Control Points Turnstile, elevator, parking barriers ?
Locking Hardware Digital proximity, magnetic switch, card reader ?
Alarm Signal Logic Siren, monitored condition, push-to-exit ?
Exit Mechanism Request-to-exit button, monitored unlock ?
Physical Barriers Reinforced doors, fencing, structural clearance ?
Control Room Integration PA system, comms, CCTV, mobile readiness ?
Fire Safety Equipment, rubbish control, combustible storage ?
Medical Facilities First aiders, marked room, clearance ?
Toilets & Signage Clearly marked, accessible, accountable ?
Risk Assessment Report Hazard ID, person at risk, control measures, sign-off ?
Would you like me to scaffold t
[Start]
Print
[1. Sector alignment]
?? Identify role: Grade A/B, Supervisor, Control Room, Advisor
?? Map to SASSETA, SAPS, SANDF, SETA (EDSETA, CETA, etc.)
?? Verify accreditation and compliance

Print
[2. Employability skills block]
?? Literacy (English, IT, Communication)
?? Behavioral & Interview skills
?? Entrepreneurship & Productivity
?? Health, Safety, Environment (HSE)

Print
[3. Practical training modules]

UserForm6 - 14

```
   ?? Metal detectors (handheld, doorframe)
   ?? Firefighting, first aid, evacuation
   ?? Equipment maintenance and role play
   ?? Incident briefing and reporting

   Print
[4. Infrastructure & equipment]
   ?? PC system, projector, UPS, smoke detector
   ?? Flashlight, binoculars, wireless comms
   ?? Hostel, dress book, insurance docs
   ?? Safety zones, checkpoints, evacuation plan

   Print
[5. Event safety & stewarding]
   ?? Fire engineering, structural clearance
   ?? Crowd control, PA system, mobile readiness
   ?? Lighting, medical, fire precautions
   ?? Toilets, signage, hazard-free premises

   Print
[6. Alarm & access control design]
   ?? Access points: turnstile, elevator, parking
   ?? Locking hardware: proximity, magnetic, card reader
   ?? Exit logic: push-to-exit, monitored unlock
   ?? Physical security plan: installation & testing

   Print
[7. Final assessment]
   ?? Internal assessment: skills, knowledge, employment readiness
   ?? Summative exam: marks, duration, pass criteria
   ?? Risk assessment report: hazard, control, action, sign-off

End
Minimal data model (Excel/Access)
"   Learners: LearnerID, FullName, IDNo, Role, PSIRAGrade, UnitStandards, AccreditationStatus
Minimal data model (Excel/Access)
"   Learners: LearnerID, FullName, IDNo, Role, PSIRAGrade, UnitStandards, AccreditationStatus
"   Modules: ModuleID, Name, Category (Employability/Practical/Event/AccessControl), Hours, Method
"   Enrolments: EnrolID, LearnerID, ModuleID, StartDate, EndDate, Status (Planned/In Progress/Done)
"   Assessments: AssessID, LearnerID, ModuleID, Type (Formative/Summative), Score, MaxScore, PassMark,
 Result, Assessor, Date
"   Risks: RiskID, SiteID, Hazard, Probability(1-5), Severity(1-5), RiskScore, Control, Owner, DueDate
, Status
"   Events: EventID, SiteID, Name, Date, Venue, RiskReportRef, StewardPlanRef, SignOff
"   Installations: InstallID, SiteID, AccessPoint, LockType, ReaderType, DoorSwitch, REXType, TestedBy
, TestDate, Result
Tip: In Excel, make each table an official ListObject (Ctrl+T); in Access, mirror names for 1:many rel
ationships.
Visual Basic (VBA) core: lifecycle state machine
vb
' Module: modLifecycle
Option Explicit

Public Enum Stage
    stNone = 0
    stSector = 1
    stEmployability = 2
    stPractical = 3
    stInfrastructure = 4
    stEventSafety = 5
    stAccessDesign = 6
    stFinalAssessment = 7
End Enum


    Select Case s
        Case stSector: SectorAlignment learnerId
        Case stEmployability: LaunchEmployability learnerId
        Case stPractical: LaunchPractical learnerId
        Case stInfrastructure: CheckInfrastructure learnerId
        Case stEventSafety: EventSafetyPlan learnerId
        Case stAccessDesign: AccessControlDesign learnerId
        Case stFinalAssessment: FinaliseAssessment learnerId
```

```
        Case Else: MsgBox "No stage selected.", vbInformation
    End Select
End Sub

Private Sub SectorAlignment(ByVal learnerId As Long)
    ' Map role ? standards/accreditation checklist
    frmSector.Tag = CStr(learnerId)
    frmSector.Show
End Sub

Private Sub LaunchEmployability(ByVal learnerId As Long)
    OpenModules learnerId, "Employability"
End Sub

Private Sub LaunchPractical(ByVal learnerId As Long)
    OpenModules learnerId, "Practical"
End Sub

Private Sub CheckInfrastructure(ByVal learnerId As Long)
    frmInfra.Tag = CStr(learnerId)
    frmInfra.Show
End Sub

Private Sub EventSafetyPlan(ByVal learnerId As Long)
    frmEventSafety.Tag = CStr(learnerId)
    frmEventSafety.Show
End Sub

Private Sub AccessControlDesign(ByVal learnerId As Long)
    frmAccessDesign.Tag = CStr(learnerId)
    frmAccessDesign.Show
End Sub

Private Sub FinaliseAssessment(ByVal learnerId As Long)
    frmSummative.Tag = CStr(learnerId)
    frmSummative.Show
End Sub

Public Sub OpenModules(ByVal learnerId As Long, ByVal category As String)
    frmModules.cboCategory.Value = category
    frmModules.txtLearnerID.Value = CStr(learnerId)
    frmModules.Show
End Sub
' Module: modAssess
Option Explicit

Public Function PassFail(ByVal score As Double, ByVal maxScore As Double, ByVal passMark As Double) As
 String
    If maxScore <= 0 Then
        PassFail = "Invalid"
    ElseIf (score / maxScore) * 100 >= passMark Then
        PassFail = "Pass"
    Else
        PassFail = "Fail"
    End If
End Function

Public Sub SaveAssessment(ByVal learnerId As Long, ByVal moduleId As Long, _
                          ByVal aType As String, ByVal score As Double, ByVal maxScore As Double, _
                          ByVal passMark As Double, ByVal assessor As String)
    Dim result As String: result = PassFail(score, maxScore, passMark)
    Dim sql As String
    sql = "INSERT INTO Assessments (LearnerID, ModuleID, Type, Score, MaxScore, PassMark, Result, Asse
ssor, [Date]) VALUES (" & _
          learnerId & "," & moduleId & ",'" & aType & "'," & score & "," & maxScore & "," & passMark &
 ",'" & _
          result & "','" & Replace(assessor, "'", "''") & "', #" & Format(Date, "mm/dd/yyyy") & "#)"
    CurrentDb.Execute sql, dbFailOnError
End Sub
' Module: modRisk
Option Explicit
```

```
UserForm6 - 16

    RiskScore = p * s ' 1..5 × 1..5
End Function

Public Function RiskBand(ByVal score As Integer) As String
    Select Case score
        Case Is >= 16: RiskBand = "High"
        Case 9 To 15: RiskBand = "Medium"
        Case Else: RiskBand = "Low"
    End Select
End Function

Public Sub AddRisk(ByVal siteId As Long, ByVal hazard As String, ByVal p As Integer, ByVal s As Intege
r, _
                   ByVal control As String, ByVal owner As String, ByVal dueDt As Date)
    Dim sc As Integer: sc = RiskScore(p, s)
    Dim sql As String
    sql = "INSERT INTO Risks (SiteID, Hazard, Probability, Severity, RiskScore, Control, Owner, DueDat
e, Status) VALUES (" & _
          siteId & ",'" & Replace(hazard, "'", "''") & "'," & p & "," & s & "," & sc & ",'" & _
          Replace(control, "'", "''") & "','" & Replace(owner, "'", "''") & "',#" & Format(dueDt, "mm/
dd/yyyy") & "#,'Open')"
    CurrentDb.Execute sql, dbFailOnError
End Sub
' Module: modInstall
Option Explicit

Public Sub LogAccessPoint(ByVal siteId As Long, ByVal accessPoint As String, _
                          ByVal lockType As String, ByVal readerType As String, ByVal doorSwitch As St
ring, _
                          ByVal rexType As String, ByVal testedBy As String, ByVal testDate As Date, B
yVal result As String)
    Dim sql As String
    sql = "INSERT INTO Installations (SiteID, AccessPoint, LockType, ReaderType, DoorSwitch, REXType,
TestedBy, TestDate, Result) VALUES (" & _
          siteId & ",'" & Replace(accessPoint, "'", "''") & "','" & Replace(lockType, "'", "''") & "',
'" & _
          Replace(readerType, "'", "''") & "','" & Replace(doorSwitch, "'", "''") & "','" & Replace(re
xType, "'", "''") & _
          "','" & Replace(testedBy, "'", "''") & "',#" & Format(testDate, "mm/dd/yyyy") & "#,'" & Repl
ace(result, "'", "''") & "')"
    CurrentDb.Execute sql, dbFailOnError
End Sub
Userforms to wire quickly (Excel or Access)
"   frmSector: Role (Grade A/B, Supervisor, Control Room, Advisor), SASSETA US mapped, Accreditation c
hecklist (PSIRA, SETA), cmdSave ? write to Learners.
"   frmModules: txtLearnerID, cboCategory, list of Modules by category, cmdEnroll ? append Enrolments;
 cmdComplete ? set Status=Done.
"   frmInfra: equipment checklist (PC, projector, UPS, smoke detector, radios, binoculars), cmdSave ?
infra log.
"   frmEventSafety: fields for Event, Venue, PA/mobile checks, lighting, medical, fire, toilets, signa
ge; "Generate Risk Report" ? adds Risks rows.
"   frmAccessDesign: access point grid (turnstile, elevator, parking), lock/reader/door switch/REX; cm
dTest ? LogAccessPoint.
"   frmSummative: select Learner & Modules, enter scores, pass mark; SaveAssessment.
If you prefer Excel, map forms to tables on sheets with ListObjects and replace CurrentDb.Execute with
 worksheet writes.
Security system algorigram and logigram [Start]
   Print
[1. Sector alignment]
   ?? Role mapping: Grade A/B, Supervisor, Control Room, Advisor
   ?? Standards: SASSETA US, PSIRA, SAPS/SANDF interfaces, SETAs (EDTP, CETA)
   ?? Accreditation and license verification

   Print
[2. Design brief intake]
   ?? Site survey ? zones, entry/exit, critical assets
   ?? Threat/risk profile ? likelihood × impact
   ?? Compliance constraints ? tender specs, OHS, Fire, POPIA

   Print
[3. System architecture]
   ?? Access control (readers, locks, REX, door contacts)
   ?? CCTV (cameras, lenses, NVR, storage, networks)
```

?? Alarms (PIRs, panic buttons, duress, smoke/heat)
?? Perimeter (beams, electric/razor/mesh fencing)
?? Control rooms (workstations, VMS, UPS, comms)

Print
[4. Equipment schedule and vector symbol map]
?? Device taxonomy ? symbol set ? drawing legend
?? Bill of materials (BOM) with quantities and locations
?? Cable routes and power budget

Print
[5. Installation & commissioning]
?? Method statements, SABS/SANS references
?? Test plans: door logic, alarm signalling, camera views
?? Acceptance criteria and as-built drawings

Print
[6. Event safety & emergency planning]
?? Evac routes, assembly points, fire zones
?? Stewarding, PA/voice, medical, lighting checks
?? Fire equipment, drills, and maintenance plan

Print
[7. Training & operations]
?? Employability skills block (IT, literacy, communication)
?? Practical modules (detectors, first aid, radio)
?? SOPs: reporting, conflict de-escalation, shift handover

Print
[8. Assessment & audit]
?? Summative assessments, logbooks
?? OHS risk register and mitigations
?? Tender compliance checklist and evidence binder

End
Device taxonomy and symbol legend (vector-ready)
" Access control
o Card reader, keypad, biometric reader, door contact, maglock/strike, request-to-exit, turnstile, b
arrier gate.
" CCTV
o Fixed dome, bullet, PTZ, thermal, ANPR, encoder, NVR, VMS workstation.
" Alarms
o Panic button, PIR, glass-break, duress pedal, siren/strobe, smoke/heat detector, control panel, ke
ypad.
" Perimeter
o Active IR beam, microwave barrier, electric fence energizer, fence sensor, gate loop detector.
" Communications/infra
o PoE switch, UPS, patch panel, fiber tray, wireless bridge, network cabinet.
" Guarding/safety
o Two-way radio, body-worn camera, first-aid box, fire extinguisher, hydrant, dry riser test point.
" Keys/firearms controls (where lawfully applicable)
o Key cabinet, armory register, safe, revolver record entry (no depiction of misuse).
" Traffic and crowd
o Traffic cone, bollard, signage, queue barrier, handheld wand.
Tip: Create a drawing legend mapping DeviceType ? VectorSymbolName so your CAD/diagram tool auto-place
s the correct icon.
Minimal data model (Excel/Access)
" Sites: SiteID, Name, Address, RiskClass, PSIRARef
" Zones: ZoneID, SiteID, Name, Purpose, RiskScore
" Devices: DeviceID, SiteID, ZoneID, DeviceType, MakeModel, Identifier, X, Y, Floor, PowerW, PoE(Boo
lean), Status
" Links: LinkID, FromDeviceID, ToDeviceID, Medium (UTP/Fiber/Power/Signal), LengthM
" Tests: TestID, DeviceID, TestType, Date, Result, Technician, Notes
" BOM: BomID, DeviceType, MakeModel, Qty, UnitCost, Extended
" Risks: RiskID, SiteID, Hazard, Probability(1-5), Severity(1-5), Score, Control, Owner, DueDate, St
atus
" Training: ModuleID, Name, Category, Hours, Method
" Assessments: AssessID, LearnerID, ModuleID, Score, MaxScore, PassMark, Result, Date
CCTV plan checklist (quick)
" Coverage: entrances, cash/asset points, perimeters, parking, control room
" Camera choice: FoV, lux, WDR, IR, resolution, lens (mm), mount
" Storage: retention (days), bitrate calc, RAID, UPS autonomy
" Network: PoE budget, VLANs, uplink capacity, fiber where >90 m

```
"   Legal: signage, privacy zones, footage handling (chain of custody)
Emergency plan diagram layers
"   Evac routes and stair cores per floor
"   Assembly points and muster counts
"   Fire zones, extinguishers, hydrants, risers
"   Emergency lighting and PA/voice nodes
"   Disabled refuge points, lift restrictions
"   Steward positions and radio channels
Fire safety engineering tasks
"   Training: fire awareness, warden training, extinguisher practicals
"   Maintenance: dry riser/hydrant testing, extinguisher service schedule
"   Risk assessment: ignition sources, fuel loads, vulnerable persons
"   Plans: pre-incident plans, escape diagrams on each floor, drill log
VBA (Excel / Access): core modules
1) Device registry and BOM builder
' Module: modDevices
Option Explicit

Public Sub AddDevice(ByVal siteId As Long, ByVal zoneId As Long, ByVal devType As String, _
                     ByVal makeModel As String, ByVal ident As String, _
                     ByVal x As Double, ByVal y As Double, ByVal floor As String, _
                     ByVal powerW As Double, ByVal isPoE As Boolean)
    Dim sql As String
    sql = "INSERT INTO Devices (SiteID, ZoneID, DeviceType, MakeModel, Identifier, X, Y, Floor, PowerW
, PoE, Status) VALUES (" & _
          siteId & "," & zoneId & ",'" & Clean(devType) & "','" & Clean(makeModel) & "','" & Clean(id
ent) & "'," & _
          x & "," & y & ",'" & Clean(floor) & "'," & powerW & "," & IIf(isPoE, "True", "False") & ",'
Planned')"
    CurrentDb.Execute sql, dbFailOnError
End Sub

Public Sub BuildBOM(ByVal siteId As Long)
    Dim rs As DAO.Recordset, sql As String
    CurrentDb.Execute "DELETE FROM BOM WHERE 1=1"
    sql = "SELECT DeviceType, MakeModel, Count(*) AS Qty FROM Devices WHERE SiteID=" & siteId & " GROU
P BY DeviceType, MakeModel"
    Set rs = CurrentDb.OpenRecordset(sql, dbOpenSnapshot)
    Do While Not rs.EOF
        CurrentDb.Execute "INSERT INTO BOM (DeviceType, MakeModel, Qty, UnitCost, Extended) VALUES ('"
 & _
                          Clean(rs!DeviceType) & "','" & Clean(rs!makeModel) & "'," & rs!Qty & ",0,0)"
        rs.MoveNext
    Loop
    rs.Close
    ' Compute Extended when UnitCost later captured
    CurrentDb.Execute "UPDATE BOM SET Extended = Nz(UnitCost,0)*Nz(Qty,0)"
    MsgBox "BOM built. Update UnitCost to price the project.", vbInformation
End Sub

Private Function Clean(ByVal s As String) As String
    Clean = Replace(Nz(s, ""), "'", "''")
End Function
' Module: modRisk
Option Explicit


    RiskScore = prob * sev
End Function


    Select Case score
        Case Is >= 16: RiskBand = "High"
        Case 9 To 15: RiskBand = "Medium"
        Case Else: RiskBand = "Low"
    End Select
End Function

    Dim sc As Integer: sc = RiskScore(prob, sev)
    Dim sql As String
    sql = "INSERT INTO Risks (SiteID, Hazard, Probability, Severity, Score, Control, Owner, DueDate, S
tatus) VALUES (" & _
          siteId & ",'" & Replace(hazard, "'", "''") & "'," & prob & "," & sev & "," & sc & ",'" & _
```

```
          Replace(control, "'", "''") & "','" & Replace(owner, "'", "''") & "',#" & Format(dueDt, "mm
/dd/yyyy") & "#,'Open')"
    CurrentDb.Execute sql, dbFailOnError
' Module: modCommission
Option Explicit

Public Sub QueueDoorTests(ByVal siteId As Long)
    Dim rs As DAO.Recordset, sql As String, devId As Long
    sql = "SELECT DeviceID FROM Devices WHERE SiteID=" & siteId & " AND DeviceType IN ('DoorContact','
Maglock','REX','Reader')"
    Set rs = CurrentDb.OpenRecordset(sql, dbOpenSnapshot)
    Do While Not rs.EOF
        devId = rs!deviceId
        AddTest devId, "PowerFailRelease"
        AddTest devId, "REXUnlock"
        AddTest devId, "DoorForcedAlarm"
        rs.MoveNext
    Loop
    rs.Close
    MsgBox "Door tests queued for commissioning.", vbInformation
End Sub

Public Sub AddTest(ByVal deviceId As Long, ByVal testType As String)
    Dim sql As String
    sql = "INSERT INTO Tests (DeviceID, TestType, [Date], Result, Technician, Notes) VALUES (" & _
          deviceId & ",'" & Replace(testType, "'", "''") & "',#" & Format(Now, "mm/dd/yyyy hh:nn:ss")
 & "#,'Pending','','')"
    CurrentDb.Execute sql, dbFailOnError
End Sub
' Module: modDiagram
Option Explicit

Public Sub ExportNodesEdges(ByVal siteId As Long, ByVal folder As String)
    ExportNodes siteId, folder & "\nodes.csv"
    ExportEdges siteId, folder & "\edges.csv"
    MsgBox "Exported nodes and edges CSV. Import to your diagram tool.", vbInformation
End Sub

Private Sub ExportNodes(ByVal siteId As Long, ByVal path As String)
    Dim rs As DAO.Recordset, f As Integer
    Set rs = CurrentDb.OpenRecordset("SELECT DeviceID, DeviceType, Identifier, X, Y, Floor FROM Device
s WHERE SiteID=" & siteId, dbOpenSnapshot)
    f = FreeFile
    Open path For Output As #f
    Print #f, "id,label,type,x,y,floor"
    Do While Not rs.EOF
        Print #f, rs!deviceId & ",""" & rs!Identifier & """,""" & rs!DeviceType & """," & rs!x & "," &
 rs!y & ",""" & rs!floor & """"
        rs.MoveNext
    Loop
    Close #f: rs.Close
End Sub

Private Sub ExportEdges(ByVal siteId As Long, ByVal path As String)
    Dim rs As DAO.Recordset, f As Integer
    Set rs = CurrentDb.OpenRecordset( _
        "SELECT L.LinkID, L.FromDeviceID, L.ToDeviceID, L.Medium, L.LengthM " & _
        "FROM Links AS L INNER JOIN Devices AS D ON L.FromDeviceID=D.DeviceID " & _
        "WHERE D.SiteID=" & siteId, dbOpenSnapshot)
    f = FreeFile
    Open path For Output As #f
    Print #f, "id,source,target,medium,length_m"
    Do While Not rs.EOF
        Print #f, rs!LinkID & "," & rs!FromDeviceID & "," & rs!ToDeviceID & ",""" & rs!Medium & ""","
& Nz(rs!LengthM, 0)
        rs.MoveNext
    Loop
    Close #f: rs.Close
End Sub
' Module: modTraining
Option Explicit

Public Sub SeedTraining()
```

```
UserForm6 - 20

    AddModule "English literacy", "Employability", 6, "Reading, writing, speaking"
    AddModule "IT literacy", "Employability", 8, "PC, Word, Excel"
    AddModule "Radio communications", "Practical", 3, "Protocol, phonetic, drills"
    AddModule "Metal detectors", "Practical", 3, "Handheld & doorframe"
    AddModule "First aid basics", "Practical", 3, "Stabilize, report, escalate"
    AddModule "Fire awareness", "Practical", 3, "Extinguishers, evacuation"
    AddModule "Alarm & access control", "Practical", 4, "Readers, locks, REX"
    AddModule "Incident reporting", "Practical", 4, "Statements, OB, chain of custody"
End Sub

Private Sub AddModule(ByVal Name As String, ByVal cat As String, ByVal hrs As Integer, ByVal method As
 String)
    CurrentDb.Execute "INSERT INTO Training (Name, Category, Hours, Method) VALUES ('" & _
        Replace(Name, "'", "''") & "','" & Replace(cat, "'", "''") & "'," & hrs & ",'" & Replace(metho
d, "'", "''") & "')"
End Sub

Public Sub RecordAssessment(ByVal learnerId As Long, ByVal moduleName As String, _
                            ByVal score As Double, ByVal maxScore As Double, ByVal passMark As Double)
    Dim moduleId As Long
    moduleId = Nz(DLookup("ModuleID", "Training", "Name='" & Replace(moduleName, "'", "''") & "'"), 0)
    If moduleId = 0 Then
        MsgBox "Module not found: " & moduleName, vbExclamation
        Exit Sub
    End If
    Dim result As String
    If maxScore > 0 And (score / maxScore) * 100 >= passMark Then
        result = "Pass"
    Else
        result = "Fail"
    End If
    CurrentDb.Execute "INSERT INTO Assessments (LearnerID, ModuleID, Score, MaxScore, PassMark, Result
, [Date]) VALUES (" & _
                      learnerId & "," & moduleId & "," & score & "," & maxScore & "," & passMark & ",'
" & result & "',#" & Format(Date, "mm/dd/yyyy") & "#)"
End Sub
Access control logic checks (commissioning)
"   Power fail releases maglocks as required (life-safety).
"   REX triggers timed unlock; forced open alarms if contact state invalid.
"   Dual-reader doors enforce anti-passback where configured.
"   Panic inputs latch to alarm with audible/visual indication in control room.
"   Doors with card-only entry have free-egress or monitored REX push-to-exit.
Quick-start: 80 "vector shapes" mapping concept
Create a lookup table VectorLegend(DeviceType, VectorSymbolName, Layer) and prefill with the most comm
on 40-80 items across:
"   Access: CardReader, Keypad, Biometric, Maglock, Strike, DoorContact, REXButton, Turnstile, Barrier
Gate.
"   CCTV: DomeCam, BulletCam, PTZ, ThermalCam, ANPR, NVR, VMS, Monitor.
"   Alarm: PanicButton, PIR, GlassBreak, Duress, Siren, Strobe, Smoke, Heat, Panel, Keypad.
"   Perimeter: IRBeamTx, IRBeamRx, MicrowaveBarrier, ElectricFence, FenceSensor.
"   Infra: PoESwitch, UPS, Rack, PatchPanel, FiberTray, WiFiBridge.
"   Safety: FireExtinguisher, Hydrant, DryRiser, FirstAid, ExitSign.
"   Ops: Radio, BWC, KeyCabinet, Safe.
"   Traffic: Cone, Bollard, QueueBarrier, Sign.
Use this to drive your diagram legend and auto-place symbols when importing nodes into a drawing tool.
What do you want next?
"   An Access template with these tables, forms, and macros wired, or
"   An Excel workbook with buttons for "Build BOM", "Export Diagram CSV", "Queue Door Tests", and "See
d Training".
Tell me the platform, and I'll tailor the files, plus pre-seed PSIRA grades, SASSETA unit standard ref
erences, and a Johannesburg sample site to accelerate deployment.
Algorigram and logigram: Compliance-to-pricing pipeline
plaintext
[Start]
   Print
[1. Standards intake]
   ?? Register compulsory specs (VC) and SANS/IEC clauses
   ?? Load Government Gazette amendments (notice ? impact)
   ?? Define project policy: security, safety, electrical scope

   Print
[2. Design capture]
   ?? Supply type (1?/3?), fault level, earthing system
```

?? Circuits: load, length, PF, conductor, installation method
?? Security/ICT systems: access, CCTV, alarms, automation
?? Fire/OHS: egress, zones, equipment, drills

Print
[3. Compliance verification]
?? Voltage drop ? 5% (SANS 10142 baseline)
?? Protection: breaker/RCBO/RCD selection (VC references)
?? Positioning & access, DB assembly/busbar limits
?? CoC prerequisites checklist

Print
[4. Cost-price-reward]
?? Build BOM (materials, labour, subcontract, compliance)
?? Compute price (overheads, margin, risk, VAT)
?? Rewards/penalties (KPI/LD) model

Print
[5. Gazette alignment]
?? Map notices ? affected standards
?? Create site impact checks and actions
?? Version the design/CoC with change log

Print
[6. Commissioning & handover]
?? Electrical tests (insulation/earth/loop/RCD)
?? Security system tests (door logic, alarms, retention)
?? Issue CoC, as-builts, O&M pack

Print
[7. Monitoring & enforcement]
?? OHS/Fire inspections, NCR (defence/offence) register
?? Close-out rewards; track warranty actions

End
Key technical anchors
" Supply and voltages:
o Single-phase 230-240 V, 50 Hz; typical 60-100 A service.
o Three-phase 400/230 V (line-line/line-neutral).
o Extra-low voltage ? 50 V AC or DC (controls, comms).
" Voltage drop limit (design to ? 5%):
o Single-phase: target ? 11.5 V drop on 230 V.
o Three-phase: target ? 20 V drop on 400 V.
" Protection and assemblies:
o Circuit breakers (VC ref, conformance).
o Earth-leakage/RCD where applicable (not a substitute for basic protection).
o DB accessibility; busbar current density per SANS/IEC assembly spec.
" CoC gate:
o Design conformance + verified tests + documentation before energizing.
Voltage drop formulae:
?V1?=I?(Rcos??+Xsin??)?2L\Delta V_{1\phi} = I \cdot (R\cos\varphi + X\sin\varphi) \cdot 2L
?V3?=3?I?(Rcos??+Xsin??)?L\Delta V_{3\phi} = \sqrt{3} \cdot I \cdot (R\cos\varphi + X\sin\varphi) \cdo
t L
%?V=?VVrated×100\% \Delta V = \frac{\Delta V}{V_\text{rated}} \times 100
Minimal data model (Excel or Access)
" Standards(StdID, Code, Title, Clause, Threshold, Unit, Notes)
" Gazettes(GazID, NoticeNo, GazetteNo, Date, Subject, AffectedStandards, Impact, Status)
" Sites(SiteID, Name, SupplyType, DeclaredV, FaultLevel, Earthing, Env)
" Circuits(CircuitID, SiteID, Name, Phase(1/3), LengthM, CurrentA, PF, Conductor(Cu/Al), CSAmm2, R_o
hm_km, X_ohm_km, Vrated, dV, dVpct, Pass)
" Devices(DeviceID, SiteID, System(Access/CCTV/Alarm/Automation), Type, MakeModel, Qty, UnitCost, La
bourHrs, TestHrs, ComplianceRefs)
" BOM(BomID, SiteID, Item, Qty, UnitCost, LabourHrs, SubcontractRate, Extended)
" ComplianceChecks(CheckID, SiteID, Area, StdRef, Result, Evidence, Responsible, Due)
" Costs(CostID, SiteID, Type(Materials/Labour/Subcontract/Compliance/Risk), Amount, Notes)
" PriceModel(PriceID, SiteID, BaseCost, OverheadsPct, MarginPct, RiskPct, PriceExcl, VATPct, PriceIn
cl)
" Rewards(RewardID, SiteID, Trigger, Amount, Status)
" NCR(EnfID, SiteID, Category(Defence/Offence), Description, Date, Outcome, FinancialImpact)
" ' Module: modVoltDrop
" Option Explicit
"
" Public Function VoltDropSingle(ByVal I As Double, ByVal R As Double, _

```
"                                   ByVal X As Double, ByVal PF As Double, _
"                                   ByVal L_m As Double) As Double
"       Dim L_km As Double, sinphi As Double
"       L_km = L_m / 1000#
"       sinphi = Sqr(1 - PF ^ 2)
"       VoltDropSingle = I * (R * PF + X * sinphi) * (2# * L_km)
"   End Function
"
"   Public Function VoltDropThree(ByVal I As Double, ByVal R As Double, _
"                                 ByVal X As Double, ByVal PF As Double, _
"                                 ByVal L_m As Double) As Double
"       Dim L_km As Double, sinphi As Double
"       L_km = L_m / 1000#
"       sinphi = Sqr(1 - PF ^ 2)
"       VoltDropThree = Sqr(3#) * I * (R * PF + X * sinphi) * L_km
"   End Function
"
"   Public Function VoltDropPct(ByVal dV As Double, ByVal Vrated As Double) As Double
"       VoltDropPct = (dV / Vrated) * 100#
"   End Function
"
"   Public Function PassVoltDrop(ByVal isThreePhase As Boolean, ByVal I As Double, _
"                                ByVal R As Double, ByVal X As Double, ByVal PF As Double, _
"                                ByVal L_m As Double, ByVal Vrated As Double, _
"                                Optional ByVal limitPct As Double = 5#) As Boolean
"       Dim dV As Double, pct As Double
"       If isThreePhase Then
"           dV = VoltDropThree(I, R, X, PF, L_m)
"       Else
"           dV = VoltDropSingle(I, R, X, PF, L_m)
"       End If
"       pct = VoltDropPct(dV, Vrated)
"       PassVoltDrop = (pct <= limitPct)
"   End Function
"   ' Module: modStandards
"   Option Explicit
"
"   Public Sub SeedStandards()
"       AddStd "SANS 10142", "Wiring of premises", "Volt drop ? 5%", "%"
"       AddStd "VC 8036", "Circuit breakers", "Conformant device selection", "n/a"
"       AddStd "VC 8003", "Earth-leakage (RCD/ELU)", "Coverage per zone/use", "mA"
"       AddStd "Assemblies", "DB assemblies", "Busbar current density per spec", "A/mm²"
"       AddStd "Plugs/Sockets", "Outlets/adaptors", "Pattern and safety", "n/a"
"   End Sub
"
"   Private Sub AddStd(ByVal code As String, ByVal title As String, _
"                      ByVal clause As String, ByVal unit As String)
"       CurrentDb.Execute "INSERT INTO Standards (Code, Title, Clause, Unit, Notes) VALUES ('" & _
"           Clean(code) & "','" & Clean(title) & "','" & Clean(clause) & "','" & Clean(unit) & "','')"
"   End Sub
"
"   Public Sub AddGazette(ByVal noticeNo As String, ByVal gazetteNo As String, _
"                         ByVal gazDate As Date, ByVal subject As String, _
"                         ByVal affected As String, ByVal impact As String)
"       CurrentDb.Execute "INSERT INTO Gazettes (NoticeNo, GazetteNo, [Date], Subject, AffectedStandar
ds, Impact, Status) VALUES ('" & _
"           Clean(noticeNo) & "','" & Clean(gazetteNo) & "',#" & Format(gazDate, "mm/dd/yyyy") & "#,'"
 & _
"           Clean(subject) & "','" & Clean(affected) & "','" & Clean(impact) & "','Open')"
"   End Sub
"
"   Public Sub BuildImpactChecklist(ByVal gazId As Long, ByVal siteId As Long)
"       Dim list As String, arr() As String, i As Long
"       list = Nz(DLookup("AffectedStandards", "Gazettes", "GazID=" & gazId), "")
"       If Len(list) = 0 Then Exit Sub
"       arr = Split(list, ",")
"       For i = LBound(arr) To UBound(arr)
"           AddCheck siteId, "Global", Trim$(arr(i)), "Pending", "Gazette#" & gazId
"       Next i
"   End Sub
"
"   Private Sub AddCheck(ByVal siteId As Long, ByVal area As String, ByVal stdRef As String, _
"                        ByVal result As String, ByVal notes As String)
```

```
UserForm6 - 23

"       CurrentDb.Execute "INSERT INTO ComplianceChecks (SiteID, Area, StdRef, Result, Evidence, Respo
nsible, Due) VALUES (" & _
"           siteId & ",'" & Clean(area) & "','" & Clean(stdRef) & "','" & Clean(result) & "','" & Clea
n(notes) & "','', Null)"
"   End Sub
"
"   Private Function Clean(ByVal s As String) As String
"       Clean = Replace(Nz(s, ""), "'", "''")
"   End Function
"   ' Module: modPricing
"   Option Explicit
"
"   Public Sub AddDeviceCost(ByVal siteId As Long, ByVal item As String, _
"                            ByVal qty As Double, ByVal unitCost As Double, _
"                            ByVal labourHrs As Double, ByVal labourRate As Double, _
"                            ByVal subcontract As Double, ByVal compliance As Double)
"       Dim mat As Double, lab As Double, total As Double
"       mat = qty * unitCost
"       lab = labourHrs * labourRate
"       total = mat + lab + subcontract + compliance
"       AddCost siteId, "Materials", mat, item
"       AddCost siteId, "Labour", lab, item
"       If subcontract > 0 Then AddCost siteId, "Subcontract", subcontract, item
"       If compliance > 0 Then AddCost siteId, "Compliance", compliance, item
"   End Sub
"
"   Private Sub AddCost(ByVal siteId As Long, ByVal typ As String, ByVal amount As Double, ByVal notes
 As String)
"       CurrentDb.Execute "INSERT INTO Costs (SiteID, Type, Amount, Notes) VALUES (" & _
"                         siteId & ",'" & Replace(typ, "'", "''") & "'," & amount & ",'" & Replace(not
es, "'", "''") & "')"
"   End Sub
"
"   Public Function BuildBaseCost(ByVal siteId As Long) As Double
"       Dim rs As DAO.Recordset, sum As Double
"       Set rs = CurrentDb.OpenRecordset("SELECT Amount FROM Costs WHERE SiteID=" & siteId, dbOpenSnap
shot)
"       Do While Not rs.EOF
"           sum = sum + Nz(rs!Amount, 0#)
"           rs.MoveNext
"       Loop
"       rs.Close
"       BuildBaseCost = sum
"   End Function
"
"   Public Sub SavePrice(ByVal siteId As Long, ByVal overheadsPct As Double, _
"                        ByVal marginPct As Double, ByVal riskPct As Double, ByVal vatPct As Double)
"       Dim baseCost As Double, priceExcl As Double, priceIncl As Double
"       baseCost = BuildBaseCost(siteId)
"       priceExcl = baseCost * (1# + riskPct / 100#) * (1# + overheadsPct / 100#) * (1# + marginPct /
100#)
"       priceIncl = priceExcl * (1# + vatPct / 100#)
"       CurrentDb.Execute "INSERT INTO PriceModel (SiteID, BaseCost, OverheadsPct, MarginPct, RiskPct,
 PriceExcl, VATPct, PriceIncl) VALUES (" & _
"           siteId & "," & baseCost & "," & overheadsPct & "," & marginPct & "," & riskPct & "," & pri
ceExcl & "," & vatPct & "," & priceIncl & ")", dbFailOnError
"   End Sub
"
"   Public Sub RegisterReward(ByVal siteId As Long, ByVal trigger As String, ByVal amount As Double)
"       CurrentDb.Execute "INSERT INTO Rewards (SiteID, Trigger, Amount, Status) VALUES (" & _
"           siteId & ",'" & Replace(trigger, "'", "''") & "'," & amount & ",'Pending')"
"   End Sub
"
"   Public Sub SetRewardStatus(ByVal rewardId As Long, ByVal status As String)
"       CurrentDb.Execute "UPDATE Rewards SET Status='" & Replace(status, "'", "''") & "' WHERE Reward
ID=" & rewardId, dbFailOnError
"   End Sub
"   ' Module: modCoC
"   Option Explicit
"
"   Public Function PassCoC(ByVal siteId As Long) As Boolean
"       ' Gate: all compliance checks Pass and all circuits Pass
"       Dim badChecks As Long, badCircuits As Long
```

```
UserForm6 - 24

"       badChecks = Nz(DCount("*", "ComplianceChecks", "SiteID=" & siteId & " AND Result<>'Pass'"), 0)
"       badCircuits = Nz(DCount("*", "Circuits", "SiteID=" & siteId & " AND Pass=False"), 0)
"       PassCoC = (badChecks = 0 And badCircuits = 0)
"   End Function
"
"   Public Sub ExportCoCReport(ByVal siteId As Long, ByVal path As String)
"       Dim f As Integer: f = FreeFile
"       Open path For Output As #f
"       Print #f, "CoC Pre check Report for SiteID " & siteId
"       Print #f, "ComplianceChecks not Pass: " & DCount("*", "ComplianceChecks", "SiteID=" & siteId &
 " AND Result<>'Pass'")
"       Print #f, "Circuits not Pass: " & DCount("*", "Circuits", "SiteID=" & siteId & " AND Pass=Fals
e")
"       Close #f
"       MsgBox "CoC pre check exported.", vbInformation
"   End Sub
Example workflow
1.  Seed standards and register Gazette notices (amendments affecting breakers, RCDs, plugs/sockets).
2.  Capture circuits; compute ?V\Delta V and flag Pass ? 5%.
3.  Add devices (security/ICT/fire), build costs (materials, labour, subcontract, compliance).
4.  Save price (overheads, margin, risk, VAT) and register rewards (e.g., "Zero NCR at handover").
5.  Build impact checklist from Gazette, close items with evidence.
6.  Run CoC pre-check; if Pass, generate report for handover pack.
If you want, I can tailor this into an Access template or Excel workbook with a Control Panel (buttons
: "Seed Standards", "Add Gazette", "Validate Circuits", "Build Costs", "Compute Price", "CoC Pre check
", "Export Reports"). Which platform should I prepare first, and do you want a preloaded Johannesburg
site profile (230/400 V, typical Cu resistances, default PF)?
[Start]
  Print
[1. Inputs]
  ?? Labour catalog (category ? ZAR/hr)
  ?? Materials catalog (item ? ZAR/unit)
  ?? Subcontract catalog (BOQ lines ? ZAR/unit)
  ?? Productivity norms (hrs/unit by skill)
  ?? Compliance tasks (tests, CoC, inspections)
  ?? Site factors (location, access, shift, risk)

  Print
[2. Scope capture]
  ?? Electrical: DBs, circuits, points, cable runs
  ?? Lighting: rooms, lux targets, control
  ?? Security/ICT: CCTV, access, intercom
  ?? Plumbing: geyser SANS 10254 tasks
  ?? Provisional items and contingencies

  Print
[3. Quantification]
  ?? Take-off ? quantities (units)
  ?? Labour hours = qty × hrs/unit
  ?? Materials cost = qty × unit cost
  ?? Subcontract = qty × sub rate
  ?? Compliance/test hours

  Print
[4. Cost build-up]
  ?? Direct costs: labour + materials + subcontract
  ?? Compliance/testing (fixed/percent)
  ?? Overheads (%)
  ?? Risk/contingency (%)
  ?? Margin (%)

  Print
[5. Price and outputs]
  ?? Price excl. VAT
  ?? VAT (SA default 15%)
  ?? Price incl. VAT
  ?? BOQ with unit rates
  ?? Logs: assumptions, versions, approvals

  Print
[6. Review and sign-off]
  ?? Sensitivity (rates, productivity, risk)
  ?? Freeze baseline; export schedules
```

UserForm6 - 25

End
Catalogs, rates, and BOQ templates
Labour rate guideline
Category Typical scope Guideline ZAR/hr Productivity hrs/unit (editable)
Skilled worker DB install, terminations, testing 250-450 1.5 (DB)
Semi-skilled worker Chasing, pulling, mounting 150-250 0.6 (point)
General worker Carry, clean, assist 100-170 0.2 (point)
Supervisor QA, permits, sign-off 350-600 0.3 (DB)
Sources: set your own rates per current wage tables or agreement; the above are placeholders. Concrete grade, if applicable to sleeves/ducting: e.g., 25 MPa or 30 MPa - price via materials catalog.
Daily cost roll-up
" Formula (team day cost):
Daily cost=?(Hours?Rate)+Consumables+Plant+Travel\text{Daily cost} = \sum (\text{Hours} \cdot \text{Rate}) + \text{Consumables} + \text{Plant} + \text{Travel}
" Formula (unit labour cost):
Unit labour cost=hrs/unit?rate\text{Unit labour cost} = \text{hrs/unit} \cdot \text{rate}
Subcontractor rate sheet (electrical)
Description Unit Rate ZAR/unit Notes
Basic distribution board installation (single-phase) each Cover, mount, gland, label
Basic distribution board installation (three-phase) each Include torque test
Three-phase protection set (main + RCD + SPD) set Device spec per design
Fit ripple relay + geyser contactor each Includes control wiring
Fit sub-board to outbuilding each Cable, trenching extra
Supply and lay 3-phase cable from boundary m Specify size and trench class
Supply and lay 2-phase cable from boundary m Clarify phases/neutral
Cable to external light m UV-rated
Supply/install boundary box each Metering per utility
Fit stove/oven/hob point excl. light each 32 A or per plate rating
Fit internal light point each Box, wire, test
Recessed light point (slab) each Allow coring/boxing
Double plug point each Dedicated or ring per design
DStv decoder conduit point each RG6 by others?
Waterproof plug point each IP65
TV point conduit only each Draw cord
Telecom sleeve conduit m 50 mm sleeve
Dimmer switch each Rated to load
Two-way switch each
Motor gate point (excl. motor) each 230 V feed
Photo-cell (day/night) each With contactor if needed
Shaver point each Isolated
Heated towel rail point each
Air-conditioner point each Per BTU/amp
Underfloor heating point each RCD required
Garage door operator point each Ceiling drop
Bathroom heater point each IP zone check
Audio speaker point each Conduit to hub
Pool pump point (water-right) each G_land, IP
Doorbell point with 12 V transformer each SELV
Intercom supply + 12 V transformer point With isolator
Supply-fix small power trunking m Include accessories
Add a "Scope" column if you need to model inclusions/exclusions precisely.
VBA cost engine and catalogs (Excel/Access)
1) Core types and helpers
vb
```
' Module: modTypes
Option Explicit

Public Type RateItem
    Name As String
    rate As Double   ' ZAR per hour or per unit
End Type

Public Type BOQItem
    Code As String
    Desc As String
    Unit As String
    Qty As Double
    MatUnit As Double
    SubUnit As Double
    HrsPerUnit As Double
    CrewMixSkilled As Double   ' fraction of hours
    CrewMixSemi As Double
```

```
UserForm6 - 26

    CrewMixGeneral As Double
    OverheadsPct As Double
    RiskPct As Double
    MarginPct As Double
End Type

Public Function NzD(ByVal v As Variant, ByVal d As Double) As Double
    If IsNull(v) Or IsEmpty(v) Then
        NzD = d
    Else
        NzD = v
    End If
End Function

Public Function Round2(ByVal v As Double) As Double
    Round2 = WorksheetFunction.Round(v, 2)
End Function
' Module: modCatalog
Option Explicit

Public Sub SeedLabourRates(ByVal rSkilled As Double, ByVal rSemi As Double, ByVal rGen As Double, ByVa
l rSup As Double)
    PutRate "Skilled", rSkilled
    PutRate "Semi", rSemi
    PutRate "General", rGen
    PutRate "Supervisor", rSup
End Sub

Private Sub PutRate(ByVal Name As String, ByVal rate As Double)
#If Win64 Then
    ' Excel Table: LabourRates(Name, Rate)
#End If
    With Sheet1.ListObjects("LabourRates").ListRows.Add
        .Range(1, 1).Value = Name
        .Range(1, 2).Value = rate
    End With
End Sub

Public Function GetRate(ByVal Name As String) As Double
    Dim lo As ListObject, r As ListRow
    Set lo = Sheet1.ListObjects("LabourRates")
    For Each r In lo.ListRows
        If StrComp(CStr(r.Range(1, 1).Value), Name, vbTextCompare) = 0 Then
            GetRate = NzD(r.Range(1, 2).Value, 0#)
            Exit Function
        End If
    Next r
    GetRate = 0#
End Function
' Module: modCost
Option Explicit


    Dim skilled As Double, semi As Double, gen As Double
    Dim rateS As Double, rateM As Double, rateG As Double
    Dim mat As Double, subc As Double, lab As Double
    Dim base As Double, withOH As Double, withRisk As Double, withMargin As Double

    ' Labour hours
    skilled = item.Qty * item.HrsPerUnit * NzD(item.CrewMixSkilled, 0#)
    semi = item.Qty * item.HrsPerUnit * NzD(item.CrewMixSemi, 0#)
    gen = item.Qty * item.HrsPerUnit * NzD(item.CrewMixGeneral, 0#)

    ' Rates
    rateS = GetRate("Skilled")
    rateM = GetRate("Semi")
    rateG = GetRate("General")

    ' Costs
    lab = skilled * rateS + semi * rateM + gen * rateG
    mat = item.Qty * NzD(item.MatUnit, 0#)
    subc = item.Qty * NzD(item.SubUnit, 0#)
    base = lab + mat + subc
```

```
    ' Uplifts
    withOH = base * (1# + NzD(item.OverheadsPct, 0#) / 100#)
    withRisk = withOH * (1# + NzD(item.RiskPct, 0#) / 100#)
    withMargin = withRisk * (1# + NzD(item.MarginPct, 0#) / 100#)

    CalcLinePrice = Round2(withMargin)
End Function

Public Function PriceInclVAT(ByVal priceExcl As Double, Optional ByVal vatPct As Double = 15#) As Doub
le
    PriceInclVAT = Round2(priceExcl * (1# + vatPct / 100#))
End Function
' Module: modBOQ
Option Explicit

Public Sub PriceBOQ()
    Dim lo As ListObject, r As ListRow, itm As BOQItem
    Dim priceEx As Double, priceIn As Double

    Set lo = Sheet1.ListObjects("BOQ") ' Columns: Code,Desc,Unit,Qty,MatUnit,SubUnit,HrsPerUnit,CrewSk
,CrewSe,CrewGe,OH,Risk,Margin,PriceEx,PriceIn
    For Each r In lo.ListRows
        itm.Code = r.Range(1, 1).Value
        itm.Desc = r.Range(1, 2).Value
        itm.Unit = r.Range(1, 3).Value
        itm.Qty = NzD(r.Range(1, 4).Value, 0#)
        itm.MatUnit = NzD(r.Range(1, 5).Value, 0#)
        itm.SubUnit = NzD(r.Range(1, 6).Value, 0#)
        itm.HrsPerUnit = NzD(r.Range(1, 7).Value, 0#)
        itm.CrewMixSkilled = NzD(r.Range(1, 8).Value, 0#)
        itm.CrewMixSemi = NzD(r.Range(1, 9).Value, 0#)
        itm.CrewMixGeneral = NzD(r.Range(1, 10).Value, 0#)
        itm.OverheadsPct = NzD(r.Range(1, 11).Value, 0#)
        itm.RiskPct = NzD(r.Range(1, 12).Value, 0#)
        itm.MarginPct = NzD(r.Range(1, 13).Value, 0#)

        priceEx = CalcLinePrice(itm)
        priceIn = PriceInclVAT(priceEx, Sheet1.Range("VATPct").Value)

        r.Range(1, 14).Value = priceEx
        r.Range(1, 15).Value = priceIn
    Next r
End Sub
' Module: modLighting
Option Explicit

Public Function MaintainedLux(ByVal lumensPerFitting As Double, ByVal CU As Double, _
                              ByVal MF As Double, ByVal fittings As Long, ByVal area_m2 As Double) As
Double
    Dim totalLum As Double
    totalLum = lumensPerFitting * fittings * CU * MF
    MaintainedLux = Round2(totalLum / area_m2)
End Function

Public Function RequiredFittings(ByVal targetLux As Double, ByVal lumensPerFitting As Double, _
                                 ByVal CU As Double, ByVal MF As Double, ByVal area_m2 As Double) As L
ong
    Dim n As Double
    n = (targetLux * area_m2) / (lumensPerFitting * CU * MF)
    RequiredFittings = Application.WorksheetFunction.Ceiling_Precise(n, 1)
End Function
"   Lumen method equations:
E=N?F?CU?MFAE = \frac{N \cdot F \cdot CU \cdot MF}{A}
N=?E?AF?CU?MF?N = \left\lceil \frac{E \cdot A}{F \cdot CU \cdot MF} \right\rceil
"   Typical starting targets (edit to spec):
o   Entrance hall: 200-300 lux
o   Stairs: 100-150 lux
o   Kitchen task areas: 300-500 lux
o   Bedroom ambient: 100-200 lux
6) Voltage drop checker (reuse-ready)
' Module: modVDrop
Option Explicit
```

```
Public Function dV_1ph(ByVal i As Double, ByVal r As Double, ByVal x As Double, _
                       ByVal PF As Double, ByVal L_m As Double) As Double
    Dim L_km As Double, sphi As Double
    L_km = L_m / 1000#
    sphi = Sqr(1 - PF ^ 2)
    dV_1ph = i * (r * PF + x * sphi) * (2# * L_km)
End Function

Public Function dV_3ph(ByVal i As Double, ByVal r As Double, ByVal x As Double, _
                       ByVal PF As Double, ByVal L_m As Double) As Double
    Dim L_km As Double, sphi As Double
    L_km = L_m / 1000#
    sphi = Sqr(1 - PF ^ 2)
    dV_3ph = Sqr(3#) * i * (r * PF + x * sphi) * L_km
End Function

Public Function dV_pct(ByVal dV As Double, ByVal Vrated As Double) As Double
    dV_pct = (dV / Vrated) * 100#
```

Lighting and controls overview

" Definitions:

o Lamp: Light source (LED, fluorescent, halogen).

o Luminaire: Fitting housing and optics.

o Colour: CCT (K) and CRI; select per space use.

o Control: Switches, dimmers, occupancy, daylight sensors.

" Regulatory anchors:

o Safety: Low-voltage installation rules; luminaire connection systems to appropriate safety standards.

o Energy efficiency: Target maintained illuminance with appropriate MF and CU; apply controls to reduce burn hours.

" Ballasts/drivers:

o Electronic driver: For LED; ensure dimming protocol match (0-10 V, DALI).

" Planning steps:

o Define target lux per space.

o Select luminaire photometrics and CU.

o Choose MF based on environment and maintenance cycle.

o Calculate fittings via the lumen method.

o Assign circuits and controls (two-way, dimmer, sensor).

" Sample calculation:

A=20 m2, E=300 lux, F=2000 lm, CU=0.6, MF=0.8A = 20\ \text{m}^2,\ E=300\ \text{lux},\ F=2000\ \text{lm},\ CU=0.6,\ MF=0.8

N=?300?202000?0.6?0.8?=?6.25?=7N = \left\lceil \frac{300 \cdot 20}{2000 \cdot 0.6 \cdot 0.8} \right\rceil = \lceil 6.25 \rceil = 7

Compliance checklists and safety hooks

" Electrical CoC preconditions:

o Disconnection and isolation: Clear, lockable, accessible.

o Protection selection: Breakers, RCD/RCBO matched to circuits.

o Voltage drop: Design to ? 5% branch and feeder limits.

o DB assembly: Labeling, torque logs, busbar ratings.

o Positioning: Height, ingress, accessibility, zone ratings in wet areas.

" Geyser (water heater) essentials:

o Pressure control, expansion relief, and safety valve.

o Drip tray, overflow routing, isolator within sight.

o Bonding/earthing, correctly rated circuit and disconnector.

" Security integration:

o SELV/ELV segregation from mains.

o Dedicated power with RCD where required.

o Functional tests: CCTV recording, door access fail-safe/fail-secure, intercom power, and retention.

SASSETA policing modules matrix (template)

Module	Learning outcome focus	Credit	Assessment artifacts
Regulatory framework	Evidence law; CPA in policing; common law/statutory offences; CJS overview		Case briefs; scenario analyses
Criminal investigation	Preliminary investigation; statements; reports		Simulated dockets; chain-of-custody logs
Community policing	Principles; ethics and professional conduct		Community engagement plan
Client service centre	Manage CSC; detention management		SOPs; custody register audit
Tactical policing	Street survival; proportional force; complaint reception		Practical drills; use-of-force reflections
Operations (SAPS)	Collision scene; domestic incident handling		Incident reports; checklists
Detective/forensic	Evidence handling; basic forensics		Evidence collection worksheets
MIS and admin	Information management; frameworks; comms		Dashboards; comms templates
Traffic operations	Road management; vehicle relations		Traffic ops plan

UserForm6 - 29

Paralegal basics Police science intro; public policy; admin of justice Notes; policy memos; quizzes
" Link to trade modules: Where electrical/security installations occur in policing facilities, align safety testing (PAT, RCD tests, DB audits) with operational security tests (CCTV, access control), and document both in a unified commissioning report.
What would help me tailor this for you
" Provide your current ZAR/hr rates for skilled, semi-skilled, general, supervisor.
" Pick your platform: Excel workbook with tables and buttons, or an Access database with forms.
" Confirm VAT and overhead defaults: I'll set 15% VAT and your preferred overhead/risk/margin.
" Share your preferred lux targets per space, and I'll preload a lighting sheet for a Johannesburg residential/commercial baseline.
Tshingombe , you 've compiled a powerful and multidimensional archive-spanning licensing administration, electrical compliance, cyber security, career assessments, and FOI procedures. To support your licensing administrator role and infrastructure oversight, I'll scaffold a Visual Basic-driven framework that integrates:
" ?? Licensing and duty tracking
" ?? Career assessment and training records
" ?? Engineering and security compliance
" ?? Cost analysis and amendment bill tracking
" ?? FOI and legal request validation
Let's break this into a deployable algorigram, logigram, and modular VBA codebase for your licensing administrator dashboar
[Start]
  Print
[1. Role and Facility Setup]
  ?? Assign roles: Lead DevOps, Intelligence Officer, Infrastructure Engineer
  ?? Register training facility and duty posts
  ?? Configure licensing blocks (PC, Office 365, Defender, Clipchamp)

  Print
[2. License Capture and Automation]
  ?? Online license form (IO, cyber, certificate)
  ?? Duty book automation (MetBook, posted agreements)
  ?? Block mark validation (field name vs current value)

  Print
[3. Career and Assessment Tracking]
  ?? Record training modules (Eaton, Schneider, Microsoft)
  ?? Score assessments (VFD, C445, HVAC, biometrics)
  ?? Generate transcript and badge summary

  Print
[4. Cost and Amendment Bill Analysis]
  ?? Labour rate guidelines (skill level, wage/hr)
  ?? Subcontractor rates (electrical BOQ)
  ?? Amendment bill tracking (SABS, OSHA, IEC)
  ?? Licensing cost vs reward matrix

  Print
[5. FOI and Legal Request Validation]
  ?? Validate FOI request (Section 8 compliance)
  ?? Capture ID/address proof
  ?? Generate legal annex and response log

  Print
[6. Reporting and Export]
  ?? Export licensing dashboard
  ?? Generate career summary
  ?? Submit FOI/legal reports

End
?? Logigram: Licensing Form Logic
plaintext
[Start]
  Print
[License Form Submission]
  ?? Field: Certificate ? Value inspected vs current
  ?? Field: Cyber ? Value match check
  ?? Field: Duty Posted ? Agreement validation

  Print
[Automation Trigger]
  ?? If values match ? auto-approve

```
UserForm6 - 30

   ?? If mismatch ? flag for manual review

   Print
[Duty Book Update]
   ?? Log MetBook entry
   ?? Timestamp and role assignment

End
1.  License Form Capture
"   ' Module: modLicenseForm
"   Option Explicit
"
"   Public Sub CaptureLicenseForm()
"       Dim certVal As String, cyberVal As String, dutyVal As String
"       certVal = InputBox("Enter Certificate Value:")
"       cyberVal = InputBox("Enter Cyber Value:")
"       dutyVal = InputBox("Enter Duty Posted Value:")
"
"       If certVal = GetCurrentValue("Certificate") And _
"          cyberVal = GetCurrentValue("Cyber") And _
"          dutyVal = GetCurrentValue("DutyPosted") Then
"           MsgBox "License values verified. Auto-approved.", vbInformation
"           LogMetBook certVal, cyberVal, dutyVal
"       Else
"           MsgBox "Mismatch detected. Manual review required.", vbExclamation
"       End If
"   End Sub
"
"   Private Function GetCurrentValue(ByVal fieldName As String) As String
"       ' Simulate lookup from database or sheet
"       Select Case fieldName
"           Case "Certificate": GetCurrentValue = "Valid"
"           Case "Cyber": GetCurrentValue = "Secure"
"           Case "DutyPosted": GetCurrentValue = "DefenderB"
"           Case Else: GetCurrentValue = ""
"       End Select
"   End Function
"
"   Private Sub LogMetBook(ByVal cert As String, ByVal cyber As String, ByVal duty As String)
"       ' Simulate logging duty post
"       Debug.Print "MetBook Entry: " & cert & ", " & cyber & ", " & duty & " @ " & Now
"   End Sub
"   ' Module: modCareerAssessment
"   Option Explicit
"
"   Public Sub RecordAssessment(ByVal title As String, ByVal typ As String, _
"                               ByVal completed As Boolean, ByVal score As Double, ByVal status As Str
ing)
"       Dim sql As String
"       sql = "INSERT INTO Assessments (Title, Type, Completed, Score, Status) VALUES ('" & _
"             Replace(title, "'", "''") & "','" & Replace(typ, "'", "''") & "'," & _
"             IIf(completed, "True", "False") & "," & score & ",'" & Replace(status, "'", "''") & "')"
"       CurrentDb.Execute sql, dbFailOnError
"   End Sub
"   ' Module: modFOI
"   Option Explicit
"
"   Public Function ValidateFOIRequest(ByVal description As String, ByVal hasIDProof As Boolean, _
"                                      ByVal hasAddressProof As Boolean) As Boolean
"       If Len(description) < 10 Then
"           MsgBox "FOI request too vague. Must describe information clearly.", vbExclamation
"           ValidateFOIRequest = False
"       ElseIf Not hasIDProof Or Not hasAddressProof Then
"           MsgBox "Missing ID or address proof. FOI request invalid.", vbExclamation
"           ValidateFOIRequest = False
"       Else
"           MsgBox "FOI request valid under Section 8.", vbInformation
"           ValidateFOIRequest = True
"       End If
"   End Function
"   ' Module: modCostReward
"   Option Explicit
"
```

UserForm6 - 31

```
"   Public Function ComputeReward(ByVal baseCost As Double, ByVal complianceBonus As Double, _
"                                 ByVal riskPenalty As Double) As Double
"       Dim reward As Double
"       reward = baseCost + complianceBonus - riskPenalty
"       ComputeReward = Round(reward, 2)
"   End Function
?? Licensing Dashboard Fields
Field Name  Value When Inspected    Current Value   Same Value
Certificate Valid   Valid   Yes
Cyber   Secure  Secure  Yes
Duty Posted DefenderB   DefenderB   Yes
?? Career Roles and Training Modules
"   Roles: Licensing Administrator, Lead DevOps Engineer, Intelligence Officer, Infrastructure Enginee
r
"   Training Modules:
o   VFD commissioning pre-test
o   Power Expert C445 demo
o   HVAC and airborne transmission
o   Biometrics integration
o   Arc flash and short circuit analysis
o   FOI request handling
o   Microsoft 365 admin and Defender security
o   Eaton switchgear and warranty diagnostics
Would you like me to format this into a licensing dashboard workbook, a career transcript generator, o
r a FOI response template next? You've built a legacy-grade framework-let's make it certifiable, autom
atable, and ready for deployment across Microsoft, Eaton, Schneider, and public sector platforms.
Visual Basic command dashboard for troop leading and policing operations
you 're building a living doctrine. Below is a deployable package: a clear logigram, commander action
keys, and a modular VBA codebase that turns your planning doctrine into trackable orders, readiness, a
ssessment, traffic studies, evidence logs, and cyber incidents.
[Commander Selected]
      Print
[1. Receive Mission] ? capture mission + constraints
      Print
[2. Issue WARNORD] ? initial tasks + timelines
      Print
[3. Tentative Plan] ? COA develop + intel + resources
      Print
[4. Initiate Movement] ? staging + relief factor + patrol density
      Print
[5. Reconnaissance] ? threat, terrain, population, traffic nodes
      Print
[6. Complete Plan] ? OPORD + annexes (Intel, Ops, Sustain, Signal)
      Print
[7. Disseminate Orders] ? units, MPs, traffic, forensics, cyber
      Print
[8. Supervise & Assess] ? compliance, response, speed studies, crime trends
      ? (feedback loops to steps 3-6 upon variation/indicators)
Action keys, modified steps, and key outputs
"   Commander action keys: Mission intake, order production, dissemination, resource allocation, super
vision.
"   Modified steps (TLP-aligned): Receive ? Warn ? Plan ? Move ? Recon ? Complete ? Issue ? Supervise.
"   Key outputs: OPORD + annexes; readiness matrix; resource allocation sheet; patrol plan; traffic en
forcement study; forensic/evidence log; cyber incident report; compliance dashboard; assessment narrat
ive.
VBA program modules
Paste into an Access/Excel VBA project. Adjust sheet/table names to your environment.
1) Core types, commander selection, step control
' Module: modCommand
Option Explicit

Public Enum TLPStep
    ReceiveMission = 1
    IssueWARNORD = 2
    MakeTentativePlan = 3
    InitiateMovement = 4
    ConductRecon = 5
    CompletePlan = 6
    IssueOPORD = 7
    SuperviseAssess = 8
End Enum

Public Type Commander
```

```
UserForm6 - 32

    CommanderID As String
    Name As String
    Rank As String
    Unit As String
End Type

Public CurrentCommander As Commander
Public CurrentStep As TLPStep

Public Sub SelectCommander(ByVal id As String, ByVal nm As String, ByVal rk As String, ByVal un As Str
ing)
    CurrentCommander.CommanderID = id
    CurrentCommander.Name = nm
    CurrentCommander.Rank = rk
    CurrentCommander.Unit = un
    CurrentStep = ReceiveMission
    Debug.Print "Commander selected: " & nm & " (" & rk & "), Unit: " & un
End Sub

 0)
    If targetStep <> 0 Then
        CurrentStep = targetStep
    Else
        If CurrentStep < SuperviseAssess Then
            CurrentStep = CurrentStep + 1
        End If
    End If
    Debug.Print "Advanced to step: " & CurrentStep
End Sub
' Module: modOrders
Option Explicit

Public Sub ProduceOrder(ByVal mission As String, ByVal constraints As String)
    ' Persist to table/sheet as needed
    Debug.Print "Mission: " & mission
    Debug.Print "Constraints: " & constraints
    AdvanceStep MakeTentativePlan
End Sub

Public Sub DisseminateOrder(ByVal recipients As String, ByVal channels As String)
    ' recipients: e.g., "MP;Traffic;Forensics;Cyber;Patrols"
    ' channels: e.g., "Radio;Email;Dashboard"
    Debug.Print "OPORD disseminated to: " & recipients & " via " & channels
    AdvanceStep SuperviseAssess
End Sub

Public Function ResourceAllocationOk(ByVal personnel As Long, ByVal required As Long, _
                                     ByVal vehicles As Long, ByVal vehReq As Long, _
                                     ByVal commsOk As Boolean) As Boolean
    ResourceAllocationOk = (personnel >= required) And (vehicles >= vehReq) And commsOk
End Function

Public Sub ValidateAndIssueOPORD(ByVal annexIntel As String, ByVal annexOps As String, _
                                 ByVal annexSustain As String, ByVal annexSignal As String, _
                                 ByVal recipients As String, ByVal channels As String)
    If ResourceAllocationOk(Cells(2, 2).Value, Cells(2, 3).Value, Cells(3, 2).Value, Cells(3, 3).Value
, Cells(4, 2).Value) Then
        Debug.Print "Annex I: " & annexIntel
        Debug.Print "Annex O: " & annexOps
        Debug.Print "Annex S: " & annexSustain
        Debug.Print "Annex C: " & annexSignal
        AdvanceStep IssueOPORD
        DisseminateOrder recipients, channels
    Else
        Debug.Print "Resource allocation insufficient. Adjust plan."
    End If
End Sub
' Module: modPrep
Option Explicit

Public Sub TroopPreparation(ByVal drillsDone As Boolean, ByVal SOPReviewed As Boolean, _
                            ByVal equipmentCheck As Boolean, ByVal commsCheck As Boolean)
    Dim ready As Boolean
```

```
UserForm6 - 33

    ready = drillsDone And SOPReviewed And equipmentCheck And commsCheck
    Debug.Print "Troop readiness: " & IIf(ready, "READY", "NOT READY")
    If ready Then AdvanceStep InitiateMovement
End Sub
' Module: modAssessment
Option Explicit

Public Type OpsMetrics
    patrols As Long
    incidents As Long
    AvgResponseMin As Double
    CompliancePct As Double
    CertificatesIssued As Long
End Type

Public Function ComputeCompliance(ByVal completed As Long, ByVal planned As Long) As Double
    If planned = 0 Then
        ComputeCompliance = 0
    Else
        ComputeCompliance = Round((completed / planned) * 100, 1)
    End If
End Function

Public Function SeverityIndex(ByVal incidents As Long, ByVal patrols As Long, _
                              ByVal avgResp As Double) As Double
    ' Lower is better; weight response time
    Dim idx As Double
    If patrols = 0 Then
        idx = incidents * (1 + avgResp / 60)
    Else
        idx = (incidents / patrols) * (1 + avgResp / 60)
    End If
    SeverityIndex = Round(idx, 3)
End Function

Public Sub RecordDailyStats(ByVal dayStamp As Date, ByVal complaints As Long, _
                            ByVal responses As Long, ByVal avgRespMin As Double)
    ' Persist to sheet/table as needed
    Debug.Print Format(dayStamp, "yyyy-mm-dd") & " | Complaints: " & complaints & _
                " | Responses: " & responses & " | AvgResp: " & avgRespMin & " min"
End Sub
' Module: modTraffic
Option Explicit

Public Function Percentile85(ByRef speeds() As Double) As Double
    Dim n As Long, k As Double, idx As Long
    n = UBound(speeds) - LBound(speeds) + 1
    QuickSort speeds, LBound(speeds), UBound(speeds)
    k = 0.85 * (n + 1)
    idx = WorksheetFunction.Min(WorksheetFunction.Max(Application.WorksheetFunction.RoundUp(k, 0), 1),
 n)
    Percentile85 = speeds(LBound(speeds) + idx - 1)
End Function

Private Sub QuickSort(ByRef a() As Double, ByVal first As Long, ByVal last As Long)
    Dim low As Long, high As Long, pivot As Double, temp As Double
    low = first: high = last: pivot = a((first + last) \ 2)
    Do While low <= high
        Do While a(low) < pivot: low = low + 1: Loop
        Do While a(high) > pivot: high = high - 1: Loop
        If low <= high Then
            temp = a(low): a(low) = a(high): a(high) = temp
            low = low + 1: high = high - 1
        End If
    Loop
    If first < high Then QuickSort a, first, high
    If low < last Then QuickSort a, low, last
End Sub

Public Function EnforcementThreshold(ByVal p85 As Double, ByVal margin As Double) As Double
    EnforcementThreshold = p85 + margin
End Function
Tip: The 85th percentile speed is the value below which 85% of observed vehicles travel: v85=percentil
```

```
UserForm6 - 34

e0.85(V)v_{85} = \text{percentile}_{0.85}(V).
' Module: modForensics
Option Explicit

Public Sub LogEvidence(ByVal caseID As String, ByVal item As String, ByVal collector As String, _
                       ByVal packageType As String, ByVal hazards As String)
    ' Example packageType: "Padded box", "Paper bag", "Clean vial"
    Debug.Print "Case " & caseID & " | Item: " & item & " | By: " & collector & _
                " | Package: " & packageType & " | Hazards: " & hazards & " | " & Now
End Sub

Public Sub TransferCustody(ByVal caseID As String, ByVal item As String, ByVal fromOfficer As String,
_
                           ByVal toOfficer As String)
    Debug.Print "CoC: " & caseID & " | " & item & " | " & fromOfficer & " -> " & toOfficer & " @ " & N
ow
End Sub
' Module: modCyber
Option Explicit

Public Sub LogCyberIncident(ByVal caseID As String, ByVal category As String, _
                            ByVal indicators As String, ByVal victimAcct As String)
    ' category: "Phishing", "Carding", "Malware", "Extortion", "MoneyLaundering"
    Debug.Print "Cyber " & caseID & " | " & category & " | IOC: " & indicators & " | Victim: " & victi
mAcct
End Sub
Output artifacts
Output  Description Trigger
OPORD + annexes Intel, Operations, Sustainment, Signal  ValidateAndIssueOPORD
Readiness matrix    Drills/SOP/equipment/comms status   TroopPreparation
Resource sheet  Personnel, vehicles, comms check    ResourceAllocationOk
Patrol plan Area, density, relief factor    ProduceOrder ? CompletePlan
Traffic study   85th percentile speed, threshold    Percentile85, EnforcementThreshold
Forensic CoC    Evidence log and transfers  LogEvidence, TransferCustody
Cyber report    Incident classification and IOC LogCyberIncident
Compliance dashboard    Completed vs planned, severity  ComputeCompliance, SeverityIndex
Quick Start
"   Select commander: SelectCommander "CMD-102", "T. Tshingombe", "CPT", "Ops-GP"
"   Produce order: ProduceOrder "Stabilize sector; reduce collisions and burglary by 20% in 30 days",
"No live-fire; night ops allowed"
"   Prep troops: TroopPreparation True, True, True, True
"   Validate OPORD: ValidateAndIssueOPORD "Threat map + silver crime", "Patrol grid + traffic nodes",
"Medical + logistics", "Radio plan", "MP;Traffic;Forensics;Cyber", "Dashboard;Radio"
"   Assess daily: RecordDailyStats Date, 42, 39, 11.5
If you want, I can package this into an Excel workbook with ready-made sheets (Orders, Resources, Patr
ols, Traffic, Forensics, Cyber, Assessment) and wire all macros to buttons for field use.
Start]
  Print
[Define Requirements]
  o Technical perf. factors
  o Environmental(shock, temp, EMC, radiation)
  o Manufacturing & QC
  o Schedule & Organization
  o Personnel availability
  o Cost(acquisition + lcc)
  o Past performance & priority
  o Additional factors
  Print
[Set Evaluation Criteria & Weights]
  Print
[Offer Intake & Compliance Check]
  ?? If non-compliant ? Disqualify/Clarify
  ?? If compliant ? Score
  Print
[Technical & Management Scoring]
  Print
[Lifecycle Cost Analysis]
  Print
[Risk & Past Performance Adjustment]
  Print
[Total Weighted Score + Best Value Tradeoff]
  Print
[Select Contractor & Document Rationale]
```

UserForm6 - 35

  Print
[Contract Phase & WBS/IMS Baseline]
End
Evaluation criteria set
"   Technical performance: Requirements coverage, design approach, test/evaluation approach, product l
ayout, environmental robustness (shock, temp, EMC/rad), reliability/maintainability.
"   Production/manufacturing: Manufacturing process maturity, quality control/assurance.
"   Programmatics: Planning and schedule realism, organizational structure, available personnel.
"   Cost: Acquisition price, lifecycle cost (O&M, maintenance, training, disposal).
"   Experience: Priority experience, past performance.
"   Additional factors: Cybersecurity posture, data rights/deliverables completeness, integration risk
.
Each criterion receives a weight (0-1). Ratings can be qualitative mapped to numeric (e.g., Outstandin
g=1.0, Good=0.8, Acceptable=0.6, Marginal=0.4, Unacceptable=0.0).
VBA modules (Excel VBA)
Paste into a standard Excel VBA project. Create a sheet "Offers" with one row per offer and columns na
med as referenced below, or adapt field names in code comments.
1) Criteria and ratings
vb
```
' Module: modCriteria
Option Explicit

Public Type Criterion
    Name As String
    weight As Double ' 0..1, sum ? 1
End Type

Public Criteria() As Criterion
Public RatingsMapNames() As String
Public RatingsMapValues() As Double

Public Sub InitCriteria()
    Dim i As Long
    ReDim Criteria(1 To 14)
    i = 0
    i = i + 1: Criteria(i).Name = "TechnicalPerformance": Criteria(i).weight = 0.12
    i = i + 1: Criteria(i).Name = "DesignApproach": Criteria(i).weight = 0.08
    i = i + 1: Criteria(i).Name = "TestEvalApproach": Criteria(i).weight = 0.08
    i = i + 1: Criteria(i).Name = "ProductSupplyReq": Criteria(i).weight = 0.05
    i = i + 1: Criteria(i).Name = "ProductLayout": Criteria(i).weight = 0.04
    i = i + 1: Criteria(i).Name = "ManufacturingProcess": Criteria(i).weight = 0.07
    i = i + 1: Criteria(i).Name = "QualityAssurance": Criteria(i).weight = 0.06
    i = i + 1: Criteria(i).Name = "PlanningSchedule": Criteria(i).weight = 0.08
    i = i + 1: Criteria(i).Name = "OrgStructure": Criteria(i).weight = 0.04
    i = i + 1: Criteria(i).Name = "AvailablePersonnel": Criteria(i).weight = 0.05
    i = i + 1: Criteria(i).Name = "PastPerformance": Criteria(i).weight = 0.08
    i = i + 1: Criteria(i).Name = "PriorityExperience": Criteria(i).weight = 0.05
    i = i + 1: Criteria(i).Name = "AdditionalFactors": Criteria(i).weight = 0.05
    i = i + 1: Criteria(i).Name = "CyberDataRights": Criteria(i).weight = 0.05
End Sub

Public Sub InitRatingsMap()
    ' Qualitative ? numeric mapping
    ReDim RatingsMapNames(1 To 5)
    ReDim RatingsMapValues(1 To 5)
    RatingsMapNames(1) = "Outstanding": RatingsMapValues(1) = 1#
    RatingsMapNames(2) = "Good":        RatingsMapValues(2) = 0.8
    RatingsMapNames(3) = "Acceptable":  RatingsMapValues(3) = 0.6
    RatingsMapNames(4) = "Marginal":    RatingsMapValues(4) = 0.4
    RatingsMapNames(5) = "Unacceptable": RatingsMapValues(5) = 0#
End Sub

Public Function MapRating(ByVal label As String) As Double
    Dim i As Long
    For i = LBound(RatingsMapNames) To UBound(RatingsMapNames)
        If StrComp(label, RatingsMapNames(i), vbTextCompare) = 0 Then
            MapRating = RatingsMapValues(i)
            Exit Function
        End If
    Next i
    ' If numeric text or unknown label, try Val
    MapRating = Val(label)
End Function
```

```
UserForm6 - 36

' Module: modOffers
Option Explicit

Public Type Offer
    Name As String
    AcquisitionPrice As Double
    LCC_OandM As Double
    LCC_Maint As Double
    LCC_Training As Double
    LCC_Disposal As Double
    DiscountRate As Double
    Ratings() As Double ' aligned to Criteria
    Compliant As Boolean
    Phase As String ' e.g., "RFP", "EMD", "LRIP", "FRP"
End Type


    Dim o As Offer, i As Long
    o.Name = Cells(rowIdx, "A").Value             ' Offer name
    o.AcquisitionPrice = Cells(rowIdx, "B").Value ' Price
    o.LCC_OandM = Cells(rowIdx, "C").Value
    o.LCC_Maint = Cells(rowIdx, "D").Value
    o.LCC_Training = Cells(rowIdx, "E").Value
    o.LCC_Disposal = Cells(rowIdx, "F").Value
    o.DiscountRate = Cells(rowIdx, "G").Value
    o.Compliant = (Cells(rowIdx, "H").Value = True)
    o.Phase = Cells(rowIdx, "I").Value

    ReDim o.Ratings(1 To UBound(Criteria))
    ' Columns J.. map to each criterion label or numeric
    For i = 1 To UBound(Criteria)
        o.Ratings(i) = MapRating(Cells(rowIdx, "J").Offset(0, i - 1).Value)
    Next i

    LoadOfferFromSheet = o
End Function
' Module: modScoring
Option Explicit


    Dim r As Double, t As Long, lcc As Double
    r = o.DiscountRate ' e.g., 0.08
    ' Simple stream: O&M + Maint + Training spread evenly over years; Disposal at end
    For t = 1 To years
        lcc = lcc + (o.LCC_OandM + o.LCC_Maint + o.LCC_Training) / ((1 + r) ^ t)
    Next t
    lcc = lcc + o.LCC_Disposal / ((1 + r) ^ years)
    NetPresentValueLCC = o.AcquisitionPrice + lcc
End Function


    Dim i As Long, s As Double, wsum As Double
    For i = 1 To UBound(Criteria)
        s = s + o.Ratings(i) * Criteria(i).weight
        wsum = wsum + Criteria(i).weight
    Next i
    If wsum > 0 Then s = s / wsum
    WeightedScore = Round(s, 4)
End Function


    ' Combine technical/management score with cost realism (lower LCC ? higher normalized score)
    Dim tech As Double, lcc As Double, costScore As Double, denom As Double
    tech = WeightedScore(o)
    lcc = NetPresentValueLCC(o)
    ' Normalize cost score against a scale; use inverse scaling with guard
    denom = Application.WorksheetFunction.Max(lcc, 1#)
    costScore = 1# / denom
    ' Combine (rescale costScore linearly by multiplying by a factor for readability)
    BestValueScore = Round((1 - costWeight) * tech + costWeight * NormalizeCost(costScore), 4)
End Function

Private Function NormalizeCost(ByVal x As Double) As Double
```

```
UserForm6 - 37

    ' Simple min-max substitute using observed bounds stored on sheet (cells M2=MinLCC, N2=MaxLCC ? pr
efilled)
    Dim minC As Double, maxC As Double
    minC = Range("M2").Value: maxC = Range("N2").Value
    If maxC <= minC Then
        NormalizeCost = 0.5 ' fallback
    Else
        NormalizeCost = (x - (1 / maxC)) / ((1 / minC) - (1 / maxC))
        If NormalizeCost < 0 Then NormalizeCost = 0
        If NormalizeCost > 1 Then NormalizeCost = 1
    End If
End Function
' Module: modReport
Option Explicit


    ' riskFactor: 0 (no risk) to 0.2 (reduce score up to 20%)
    Dim base As Double
    base = WeightedScore(o)
    RiskAdjustedScore = Round(base * (1 - riskFactor), 4)
End Function

Public Sub EvaluateAllOffers()
    Dim r As Long, lastRow As Long, o As Offer
    Dim bv As Double, lcc As Double, tech As Double
    Dim risk As Double, costW As Double

    InitCriteria
    InitRatingsMap

    lastRow = Cells(Rows.Count, "A").End(xlUp).row
    costW = Range("P2").Value ' cost weight 0..1

    ' Pre-compute min/max LCC for normalization
    Dim minL As Double, maxL As Double
    minL = 1E+99: maxL = 0

    For r = 2 To lastRow
        o = LoadOfferFromSheet(r)
        If o.Compliant Then
            lcc = NetPresentValueLCC(o)
            If lcc < minL Then minL = lcc
            If lcc > maxL Then maxL = lcc
        End If
    Next r

    Range("M2").Value = minL: Range("N2").Value = maxL

    For r = 2 To lastRow
        o = LoadOfferFromSheet(r)
        If o.Compliant Then
            tech = WeightedScore(o)
            lcc = NetPresentValueLCC(o)
            risk = Cells(r, "Q").Value ' risk factor per offer (0..0.2)
            bv = BestValueScore(o, costW)

            Cells(r, "U").Value = tech
            Cells(r, "V").Value = lcc
            Cells(r, "W").Value = RiskAdjustedScore(o, risk)
            Cells(r, "X").Value = bv
            Cells(r, "Y").Value = IIf(Cells(r, "R").Value >= 0.6 And Cells(r, "S").Value = "Acceptable
", "Select", "Consider")
        Else
            Cells(r, "Y").Value = "Non-compliant"
        End If
    Next r
End Sub
' Module: modEnv
Option Explicit

Public Function EnvComplianceScore(ByVal shockG As Double, ByVal tempC As Double, _
                                   ByVal emcOk As Boolean, ByVal radKradTID As Double, _
                                   ByVal requiredShockG As Double, ByVal requiredTempC As Double, _
```

```
                                    ByVal requiredRadKrad As Double) As Double
    Dim s As Double, c As Long
    ' Shock
    If shockG >= requiredShockG Then s = s + 1
    c = c + 1
    ' Temperature (binary meet)
    If tempC >= requiredTempC Then s = s + 1
    c = c + 1
    ' EMC
    If emcOk Then s = s + 1
    c = c + 1
    ' Radiation tolerance (TID)
    If radKradTID >= requiredRadKrad Then s = s + 1
    c = c + 1

    EnvComplianceScore = s / c ' 0..1
End Function
' Module: modPhysics
Option Explicit

Public Function Deceleration(ByVal vi As Double, ByVal vf As Double, ByVal distance As Double) As Doub
le
    ' Returns constant deceleration a (m/s^2) using v^2 = u^2 + 2 a s
    ' vi: initial speed (m/s); vf: final speed (m/s); distance s (m)
    If distance = 0 Then
        Deceleration = 0
    Else
        Deceleration = (vf ^ 2 - vi ^ 2) / (2# * distance)
    End If
End Function

Public Function StoppingDistance(ByVal vi As Double, ByVal a As Double) As Double
    ' Distance to stop from speed vi under constant deceleration a (<0)
    If a = 0 Then
        StoppingDistance = 0
    Else
        StoppingDistance = -(vi ^ 2) / (2# * a)
    End If
End Function
Equations:
"   Kinematics: vf2=vi2+2?a?sv_f^2 = v_i^2 + 2 \cdot a \cdot s
"   Solve for aa: a=vf2?vi22?sa = \dfrac{v_f^2 - v_i^2}{2 \cdot s}
"   Stopping distance for vf=0v_f=0: s=?vi22?as = -\dfrac{v_i^2}{2 \cdot a}
Example usage
"   Initialize and evaluate:
o   Step: Init criteria and map, then compute for all offers.
o   Action: Run EvaluateAllOffers from the Offers sheet with columns:
"   A: OfferName, B: AcquisitionPrice, C-F: LCC components, G: DiscountRate, H: Compliant (TRUE/FALSE)
, I: Phase, J-W: Ratings per criterion label or numeric, Q: Risk factor, P2: Cost weight (0-1).
"   Compute environmental fit:
o   Step: Add required thresholds in cells; call EnvComplianceScore in a helper column.
o   Action: Use its result to cap TechnicalPerformance rating if environment fit < 1.0.
"   Physics check for tests:
o   Step: Use Deceleration(vi, vf, s) to validate test profiles for braking/shock.
Tailoring to policing and military police operations
"   Map technical performance:
o   Requirements coverage: Patrol management, traffic speed study capability (85th percentile), forens
ic chain-of-custody, cyber incident logging.
"   Manufacturing/QC:
o   Label: Data integrity, audit logs, evidence packaging compliance, uptime targets.
"   Programmatics:
o   Label: Shift scheduling, relief factors, population/patrol density modeling, readiness matrices.
"   Experience/past performance:
o   Label: Prior deployments in civil-security contexts, accuracy of collision/traffic analytics, case
 throughput.
"   Additional factors:
o   Label: Data rights, privacy, cybersecurity hardening, integration with existing IMS/WBS.
If you want, I can provide a ready-to-use Excel template with named headers, color-coded status (Selec
t/Consider/Non-compliant), and buttons wired to these macros.
Logigram and Visual Basic for OHS Act electrical machinery compliance
you 're bringing law, engineering, and operations into one spine. Below is a clear logigram and a modu
lar VBA package to operationalize the Occupational Health and Safety Act, 1993 and Electrical Machiner
y Regulations (incl. SANS calibration/good practice), with controls for access, switching, PPE, cleara
```

nces, electric fences, and compliance reporting.
Logigram of compliance workflow
[Start]
  Print
[Define scope and assets]
  o Generation/Transmission/Distribution to point of supply
  o Overhead/Underground conductors, substations, switchgear
  o Electric fence systems (SANS 60335-2-76)
  o Lamps ? 50 V, HF sources, machinery
  o Confined/enclosed spaces
  Print
[Hazard identification]
  o Electrical (shock/arc), HF/RF, radiation, oxygen-deficient atmospheres
  o Unauthorized access/handling risks
  o Crossing spans (power/communication), waterway clearances, explosives proximity
  Print
[Controls planning]
  o PPE, LOTO/PTW, signage and notices at entrances
  o Access control (authorized persons only)
  o Switching/isolating arrangements (neutral/phase isolation rules)
  o Clearance distances and crossing rules
  o Calibration and SANS good practice schedule
  Print
[Implementation]
  o Execute switching plans and lockouts
  o Barriers, fencing, labels, temperature ratings
  o Electric fence compliance and registration
  o Supplier/Employer duties and remedial actions
  Print
[Inspection & testing]
  o Design/manufacture/installation checks
  o Routine/Type tests, calibration confirmation
  o Record deviations and corrective actions (with deadlines)
  Print
[Assessment & reporting]
  o Compliance score (fact % rating)
  o Non-conformances and risk level
  o Notices, permits, audit trail
  Print
[Closeout & monitoring]
  o Verify remediation, re-test, sign-off
  o Schedule next inspections
End
Data structure for Excel/Access
Create sheets/tables. Use these names to match the code.
"   ComplianceRules
o ruleID: text
o Clause: text
o Description: text
o Criticality: text (high / Med / low)
o weight: Number (0 - 1)
o   Target: Text/Number (e.g., "Yes", 50, "SANS-60335-2-76")
o   Category: Text (Access, Switching, Clearance, Fence, PPE, Calibration)
"   Assets
o   AssetID, Type, Location, Voltage, Phase, HFSource, ConfinedSpace, FenceType, Substation, Overhead,
 WaterCrossing, NearExplosives
"   Inspections
o InspectionID, Date, Inspector, assetID, ruleID, ObservedValue, PassFail, Notes, RemedialDueDate
"   Authorizations
o personID, Name, Role, AuthorizedFor, ValidTo
"   Permits
o   PermitID, AssetID, Type (PTW/LOTO/Confined), IssuedTo, Start, End, Status
"   Reports
o ReportID, periodStart, periodEnd, CompliancePct, HighFindings, OpenActions, GeneratedOn
VBA modules
Paste into Excel VBA. Adjust sheet names/columns as per your workbook.
1) Configuration and helpers
vb
```
' Module: modConfig
Option Explicit

Public Const DAYS_REMEDIAL_DEFAULT As Long = 30 ' configurable SLA
Public Const LAMP_SAFE_MAX_V As Double = 50
```

```
UserForm6 - 40

Public Const NEUTRAL_ISOLATION_PROHIBITED As Boolean = True ' unless full phase isolation is arranged
Public Const SANS_ELECTRIC_FENCE As String = "SANS 60335-2-76"

Public Function IsYes(ByVal v As Variant) As Boolean
    IsYes = (UCase$(Trim$(CStr(v))) Like "Y*") Or (v = True) Or (UCase$(Trim$(CStr(v))) = "YES")
End Function


    If IsError(v) Or IsEmpty(v) Or v = "" Then NzD = d Else NzD = CDbl(v)
End Function

Public Function NzS(ByVal v As Variant, Optional ByVal d As String = "") As String
    If IsError(v) Or IsEmpty(v) Then NzS = d Else NzS = CStr(v)
End Function
2) Rule engine and scorin
' Module: modCompliance
Option Explicit

Public Type RuleEval
    ruleID As String
    category As String
    weight As Double
    Pass As Boolean
    score As Double ' Pass ? Weight, Fail ? 0 (or partial if numeric tolerance)
End Type


    Dim r As RuleEval, passRule As Boolean, score As Double
    r.ruleID = ruleID: r.weight = weight

    Select Case True
        Case IsNumeric(target)
            passRule = (NzD(observed) >= NzD(target))
        Case UCase$(CStr(target)) = "YES"
            passRule = IsYes(observed)
        Case Else
            passRule = (Trim$(CStr(observed)) = Trim$(CStr(target)))
    End Select

    score = IIf(passRule, weight, 0#)
    r.Pass = passRule
    r.score = score
    EvaluateRule = r
End Function

Public Sub ScoreInspectionRow(ByVal rowIdx As Long)
    ' Sheet: Inspections (A:InspectionID, B:Date, C:Inspector, D:AssetID, E:RuleID, F:ObservedValue, G
:PassFail, H:Notes, I:RemedialDueDate, J:Score)
    Dim shI As Worksheet, shR As Worksheet, f As Range, rEval As RuleEval
    Dim ruleID As String, observed As Variant, weight As Double, target As Variant, category As String

    Set shI = ThisWorkbook.Sheets("Inspections")
    Set shR = ThisWorkbook.Sheets("ComplianceRules")

    ruleID = shI.Cells(rowIdx, "E").Value
    observed = shI.Cells(rowIdx, "F").Value

    Set f = shR.Range("A:A").Find(What:=ruleID, LookIn:=xlValues, LookAt:=xlWhole)
    If f Is Nothing Then
        shI.Cells(rowIdx, "G").Value = "N/A"
        shI.Cells(rowIdx, "J").Value = 0
        Exit Sub
    End If

    weight = NzD(f.Offset(0, 4).Value) ' Weight col E
    target = f.Offset(0, 5).Value      ' Target col F
    category = f.Offset(0, 6).Value    ' Category col G

    rEval = EvaluateRule(ruleID, observed, target, weight)
    shI.Cells(rowIdx, "G").Value = IIf(rEval.Pass, "Pass", "Fail")
    shI.Cells(rowIdx, "J").Value = rEval.score
    shI.Cells(rowIdx, "K").Value = category
```

```
UserForm6 - 41

    ' Auto-assign remedial due date for fails if empty
    If Not rEval.Pass And shI.Cells(rowIdx, "I").Value = "" Then
        shI.Cells(rowIdx, "I").Value = DateAdd("d", DAYS_REMEDIAL_DEFAULT, Date)
    End If
End Sub

Public Sub ScoreAllInspections()
    Dim shI As Worksheet, lastRow As Long, r As Long, totalW As Double, sumScore As Double
    Set shI = ThisWorkbook.Sheets("Inspections")
    lastRow = shI.Cells(shI.Rows.Count, "A").End(xlUp).row

    totalW = 0: sumScore = 0
    For r = 2 To lastRow
        ScoreInspectionRow r
        sumScore = sumScore + NzD(shI.Cells(r, "J").Value)
    Next r

    ' Total theoretical weight from rule table
    Dim shR As Worksheet, lastRule As Long, rr As Long
    Set shR = ThisWorkbook.Sheets("ComplianceRules")
    lastRule = shR.Cells(shR.Rows.Count, "A").End(xlUp).row
    For rr = 2 To lastRule
        totalW = totalW + NzD(shR.Cells(rr, "E").Value)
    Next rr

    Dim pct As Double
    If totalW > 0 Then pct = Round((sumScore / totalW) * 100, 1)
    ThisWorkbook.Sheets("Reports").Range("D2").Value = pct ' CompliancePct
    ThisWorkbook.Sheets("Reports").Range("G2").Value = Now ' GeneratedOn
End Sub
' Module: modDomain
Option Explicit

' Access control and signage
Public Function IsAuthorized(ByVal personID As String, ByVal assetID As String) As Boolean
    Dim sh As Worksheet, f As Range
    Set sh = ThisWorkbook.Sheets("Authorizations")
    Set f = sh.Range("A:A").Find(What:=personID, LookAt:=xlWhole)
    If f Is Nothing Then
        IsAuthorized = False
    Else
        IsAuthorized = (InStr(1, ";" & f.Offset(0, 3).Value & ";", ";" & assetID & ";", vbTextCompare)
 > 0) _
                       And (f.Offset(0, 4).Value >= Date)
    End If
End Function

' Neutral isolation rule (3-phase AC or 3-wire DC)
Public Function SwitchingArrangementValid(ByVal isPolyphase As Boolean, ByVal isolatesNeutralOnly As B
oolean, _
                                          ByVal isolatesAllPhases As Boolean) As Boolean
    If isPolyphase Then
        If NEUTRAL_ISOLATION_PROHIBITED And isolatesNeutralOnly Then
            SwitchingArrangementValid = False
        Else
            SwitchingArrangementValid = isolatesAllPhases
        End If
    Else
        SwitchingArrangementValid = True
    End If
End Function

' Clearance checks for crossings and waterways
Public Function CrossingClearanceOk(ByVal designKV As Double, ByVal spanM As Double, _
                                    ByVal clearanceM As Double, ByVal overWater As Boolean) As Boolean
    ' Simple conservative rule of thumb (configure to your standard in rules table):
    ' Higher voltage or over-water ? higher clearance required
    Dim required As Double
    required = IIf(overWater, 8#, 6#)
    If designKV > 1.1 Then required = required + 1.5
    If spanM > 150 Then required = required + 0.5
    CrossingClearanceOk = (clearanceM >= required)
End Function
```

UserForm6 - 42

```
' Electric fence compliance
Public Function ElectricFenceCompliant(ByVal stdRef As String, ByVal isBatteryFence As Boolean, _
                                       ByVal certificatePresent As Boolean, ByVal registrationPresent
As Boolean) As Boolean
    Dim stdOk As Boolean
    stdOk = (InStr(1, UCase$(stdRef), UCase$(SANS_ELECTRIC_FENCE), vbTextCompare) > 0)
    ElectricFenceCompliant = stdOk And certificatePresent And registrationPresent
End Function

' Lamp ? 50 V rule
Public Function LampVoltageSafe(ByVal lampV As Double) As Boolean
    LampVoltageSafe = (lampV <= LAMP_SAFE_MAX_V)
End Function

' Calibration confirmation (SANS/good practice)
Public Function CalibrationValid(ByVal lastCalDate As Date, ByVal calIntervalDays As Long) As Boolean
    CalibrationValid = (DateDiff("d", lastCalDate, Date) <= calIntervalDays)
End Function
' Module: modPermits
Option Explicit

Public Function IssuePermit(ByVal assetID As String, ByVal typ As String, _
                            ByVal issuedTo As String, ByVal startDt As Date, ByVal endDt As Date) As S
tring
    Dim sh As Worksheet, nextRow As Long, pid As String
    Set sh = ThisWorkbook.Sheets("Permits")
    nextRow = sh.Cells(sh.Rows.Count, "A").End(xlUp).row + 1
    pid = "PTW-" & assetID & "-" & Format(Now, "yymmddhhmm")
    sh.Cells(nextRow, "A").Value = pid
    sh.Cells(nextRow, "B").Value = assetID
    sh.Cells(nextRow, "C").Value = typ
    sh.Cells(nextRow, "D").Value = issuedTo
    sh.Cells(nextRow, "E").Value = startDt
    sh.Cells(nextRow, "F").Value = endDt
    sh.Cells(nextRow, "G").Value = "Open"
    IssuePermit = pid
End Function

Public Sub ClosePermit(ByVal permitID As String)
    Dim sh As Worksheet, f As Range
    Set sh = ThisWorkbook.Sheets("Permits")
    Set f = sh.Range("A:A").Find(What:=permitID, LookAt:=xlWhole)
    If Not f Is Nothing Then f.Offset(0, 6).Value = "Closed"
End Sub
' Module: modReports
Option Explicit

Public Sub GenerateMonthlyReport(ByVal periodStart As Date, ByVal periodEnd As Date)
    Dim shI As Worksheet, shR As Worksheet, reportRow As Long, passCount As Long, failCount As Long
    Set shI = ThisWorkbook.Sheets("Inspections")
    Set shR = ThisWorkbook.Sheets("Reports")

    Dim lastRow As Long, r As Long, d As Date
    lastRow = shI.Cells(shI.Rows.Count, "A").End(xlUp).row
    passCount = 0: failCount = 0

    For r = 2 To lastRow
        d = shI.Cells(r, "B").Value
        If d >= periodStart And d <= periodEnd Then
            If shI.Cells(r, "G").Value = "Pass" Then passCount = passCount + 1 Else failCount = failCo
unt + 1
        End If
    Next r

    reportRow = shR.Cells(shR.Rows.Count, "A").End(xlUp).row + 1
    shR.Cells(reportRow, "A").Value = "RPT-" & Format(Now, "yymmddhhmm")
    shR.Cells(reportRow, "B").Value = periodStart
    shR.Cells(reportRow, "C").Value = periodEnd
    shR.Cells(reportRow, "D").Value = Round(100 * passCount / Application.Max(1, passCount + failCount
), 1)
    shR.Cells(reportRow, "E").Value = failCount
    shR.Cells(reportRow, "F").Value = "Generated"
```

UserForm6 - 43

```
    shR.Cells(reportRow, "G").Value = Now
End Sub
Seed rule examples (add to ComplianceRules)
"   Access control
o   RuleID: ACC-ENTRY-NOTICE | Clause: Display notice at entrances | Target: Yes | Weight: 0.05 | Cate
gory: Access
o   RuleID: ACC-UNAUTH-PROHIBIT | Clause: Prohibit unauthorized entry/handling | Target: Yes | Weight:
 0.08 | Category: Access
"   Switching/Isolation
o   RuleID: SW-NEUTRAL-ISO | Clause: Neutral not isolated unless phases isolated | Target: Yes | Weigh
t: 0.10 | Category: Switching
o   RuleID: SW-SWITCHGEAR-L^K | Clause: Distribution boxes lockable; only authorized to open/work | Ta
rget: Yes | Weight: 0.07 | Category: Switching
"   Lamp and HF
o   RuleID: LMP-50V-MAX | Clause: Operating lamp ? 50 V | Target: 50 | Weight: 0.06 | Category: Equipm
ent
"   Electric fence
o   RuleID: FEN-SANS-60335 | Clause: Electric fence complies with SANS 60335-2-76 | Target: SANS 60335
-2-76 | Weight: 0.10 | Category: Fence
o   RuleID: FEN-CERT-REG | Clause: Certificate and registration present | Target: Yes | Weight: 0.08 |
 Category: Fence
"   Clearances & crossings
o   RuleID: CLR-WATER-LVL | Clause: Clearance over normal high water level adequate | Target: Yes | We
ight: 0.08 | Category: Clearance
o   RuleID: CLR-COMMS-SPAN | Clause: Power/communication crossing span protective measures | Target: Y
es | Weight: 0.05 | Category: Clearance
"   Calibration and SANS
o   RuleID: CAL-SANS-GP | Clause: Calibration confirmed per SANS/good practice | Target: Yes | Weight:
 0.08 | Category: Calibration
"   PPE and confined/enclosed
o   RuleID: PPE-AVAILABLE | Clause: Proper PPE for task | Target: Yes | Weight: 0.07 | Category: PPE
o   RuleID: CNS-SPACE-CONTROL | Clause: Confined/enclosed space controls active | Target: Yes | Weight
: 0.08 | Category: Confined
"   Explosives proximity
o   RuleID: EXP-DISTANCE | Clause: Required distance from power lines to explosives | Target: Yes | We
ight: 0.08 | Category: Clearance
Populate Inspections with ObservedValue (Yes/No or numeric), and the engine will compute Pass/Score an
d compliance percentage.
Quick usage
"   Score inspections: Run ScoreAllInspections to compute Pass/Fail, Score, and update Reports!D2 with
 compliance %.
"   Check switching: Use SwitchingArrangementValid(True, True, False) to validate neutral/phase isolat
ion.
"   Fence check: ElectricFenceCompliant "SANS 60335-2-76", True, True, True
"   Clearance: CrossingClearanceOk 11#, 180#, 9#, True
"   Calibration: CalibrationValid DateSerial(2025,6,1), 365
Optional policing and legal admin hooks
"   Case file link: Add Inspections.CaseID to tie to policing incident records.
"   Regulatory pack: Export Reports rows per month as your OHSA evidence set, attach PTW, Authorizatio
ns, and Inspections for audits.
"   Paralegal records: Log entrance notices, prohibitions, and LOTO evidence as exhibits in legal prac
tice management.
Do you want this packaged as an Excel template with named headers, colored status, and buttons to run
the macros? If yes, tell me your preferred sheet names and any additional rules you want preloaded (e.
g., specific clearance distances or fence battery system variants).


Sub engtsh()

End Sub
Sub entsh()

End Sub
Sub ents4()

End Sub
Sub ents()

End Sub
Sub tf()

End Sub
Sub tft()
```

UserForm6 - 44

End Sub

registrar record information manangement system // policy // security safety sector training authority verification , competency . defensive offensive action correct //legal traffi

Frame1

safety and security sector framework regulatory / coommunication admnis /inf managent policing and traffic and parallegal , metropolitain , systm

sdf login system user name

application experience career on job //

profile investigat private /complain

Label15

registrat/ ob book , pocket book , acess book , incidence book . counter book, judgen book

Label16

report research book marksheet performence

Label17

unity , course subject instructor , coactchin ,legal provide

Label18

private security , safety duty charge , barganing

Label19

portal career expeience

Label20

essment course subject

Label21

Label13

Label22

submitted

persap/polfin system user

application experience career

detective / investigation center , community police , safety security

Label33

docket system case number /registrat/ ob book , pocket book ,

Label34

report research book marksheet performence

Label35

unity standan

Label36

union police barganing , private police

Label37

portal career

Label38

Label30

Label39

Label31

Label40

submitted

ok

cancel

Page1 Page2

Page1 Page2