```
UserForm10 - 1

Private Sub ComboBox1_Change()

End Sub

Private Sub Frame1_Click()

End Sub

Private Sub ScrollBar1_Change()

End Sub

Private Sub SpinButton1_Change()

End Sub

Private Sub SpinButton2_Change()

End Sub

Private Sub TabStrip1_Change()

End Sub

Private Sub TextBox10_Change()

End Sub

Private Sub TextBox12_Change()

End Sub

Private Sub TextBox13_Change()

End Sub

Private Sub TextBox14_Change()

End Sub

Private Sub TextBox15_Change()

End Sub

Private Sub TextBox16_Change()

End Sub

Private Sub TextBox17_Change()

End Sub

Private Sub TextBox18_Change()

End Sub

Private Sub TextBox2_Change()

End Sub

Private Sub TextBox20_Change()

End Sub

Private Sub TextBox21_Change()

End Sub

Private Sub TextBox22_Change()

End Sub
```

UserForm10 - 2

```
Private Sub TextBox23_Change()

End Sub

Private Sub TextBox25_Change()

End Sub

Private Sub TextBox3_Change()

End Sub

Private Sub TextBox4_Change()

End Sub

Private Sub TextBox6_Change()

End Sub

Private Sub TextBox8_Change()

End Sub

Private Sub TextBox9_Change()

End Sub

Private Sub UserForm_Activate()

End Sub

Private Sub UserForm_AddControl(ByVal Control As MSForms.Control)

End Sub

Private Sub UserForm_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_Initialize()

End Sub

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

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_QueryClose(Cancel As Integer, CloseMode As Integer)

End Sub

Private Sub UserForm_RemoveControl(ByVal Control As MSForms.Control)
```

```
UserForm10 - 3

End Sub

Private Sub UserForm_Resize()

End Sub

Private Sub UserForm_Terminate()

End Sub

Private Sub UserForm_Zoom(Percent As Integer)

End Sub
```

Thank you for sharing this rich and intricate framework, Tshingombe. Based on your modular VB-based form architecture and the layered data structure you've outlined, here's a structured academic-style synthesis that aligns with your request for an overview, description, scope, background, problem statement, data analysis, methodology, and conclusion-centered around your Information Management System (IMS) prototype.

?? Information Management System: Modular Framework Overview

Inventor: Tshingombe Tshitadi Fiston Email: tshingombefiston@gmail.com

1. ?? Topic

Design and implementation of a modular, form-driven Information Management System (IMS) for secure registration, biometric validation, and user credential handling across educational or administrative platforms.

2. ?? Overview

This system is a multi-form, event-driven application built using Visual Basic (VB) architecture. It facilitates user registration, credential management, biometric scanning, and secure communication. The design emphasizes modularity, traceability, and audit-readiness, aligning with national curriculum and compliance standards.

3. ?? Description

The IMS comprises six core forms (Kfrm1-Kfrm6), each handling a distinct phase of user interaction:

" Kfrm1: Initial registration (name, surname, ID, cellphone)
" Kfrm2: Address and birthday capture
" Kfrm3: Registration confirmation
" Kfrm4: Account login and password recovery
" Kfrm5: Password resend and record saving
" Kfrm6: Security scanner and biometric validation

Each form is populated with labels, combo boxes, command buttons, and event-driven subroutines. The system integrates email recall, signature management, and out-of-office automation.

4. ?? Scope

" Included: User registration, biometric scanning, credential management, email automation, database interaction, and modular form navigation.
" Excluded: External API integration, mobile app deployment, and cloud-based synchronization (in current version).
" Target Users: Educational administrators, technical trainers, and compliance officers.

"   Deployment Context: Local or institutional networks with VB runtime support.
5. ?? Background
The system responds to the need for secure, modular, and bilingual registration platforms in vocationa
l and technical education. It supports traceable user data capture, audit-friendly workflows, and biom
etric security-aligned with Tshingombe's national curriculum architecture and ASEI/PDSI pedagogical pr
inciples.
6. ? Problem Statement
Traditional registration systems lack modularity, biometric integration, and audit-readiness. Manual d
ata entry leads to errors, delays, and compliance risks. There is a need for a scalable, form-driven s
ystem that supports secure data handling, credential recovery, and performance tracking.
7. ?? Data Analysis
"   Form Metrics: Each form logs user interaction via command events (e.g., OK, Cancel, Next).
"   Security Flags: Binary indicators (0/1) track form success, biometric validation, and credential s
tatus.
"   User Flow: Sequential navigation from Kfrm1 to Kfrm6, with conditional logic based on input valida
tion and biometric scan results.
"   Performance Indicators: Budget drivers, shipping probabilities, and competitive product metrics ar
e modeled for decision support.
8. ?? Methodology of Investigation
"   Design: VB-based user forms with structured labels, combo boxes, and command buttons.
"   Development: Modular coding with private subroutines for each form event.
"   Testing: Simulated user registration and biometric validation across 52 form states.
"   Validation: Data integrity checks, error handling routines, and recall message success/failure tra
cking.
9. ??? Information Management System Principles
According to MIS frameworks

Absolutely , Tshingombe.let 's now extend your IMS synthesis with the requested Visual Basic code logi
c, logigramme, and algorigramme-all modularized to reflect your multi-form architecture and binary tra
cking system.
?? Visual Basic Code Logic (Modular Snippets)
Here 's a simplified logic structure for your form-driven IMS

```
' Kfrm1: Initial Registration
Private Sub cmdNext_Click()
    If txtName.text = "" Or txtSurname.text = "" Or txtID.text = "" Or txtCellphone.text = "" Then
        MsgBox "Please complete all fields.", vbExclamation
    Else
        Kfrm2.Show
        Me.Hide
    End If
End Sub

' Kfrm2: Address and Birthday
Private Sub cmdSend_Click()
    If txtAddress.text = "" Or txtBirthday.text = "" Then
        MsgBox "Address and birthday required.", vbCritical
    Else
        Kfrm3.Show
        Me.Hide
    End If
End Sub

' Kfrm4: Login
Private Sub cmdLogin_Click()
    If txtUsername.text = "" Or txtPassword.text = "" Then
        MsgBox "Missing credentials.", vbInformation
    Else
        ' Simulate credential check
        If txtUsername.text = "admin" And txtPassword.text = "1234" Then
            MsgBox "Login successful.", vbOKOnly
            Kfrm5.Show
            Me.Hide
        Else
            MsgBox "Invalid login.", vbRetryCancel
        End If
    End If
End Sub

' Kfrm6: Biometric Scan
Private Sub cmdScan_Click()
    Dim biometricFlag As Integer
    biometricFlag = 1 ' Simulated scan success
```

```
UserForm10 - 5

    If biometricFlag = 1 Then
        MsgBox "Biometric scan passed.", vbOKOnly
    Else
        MsgBox "Scan failed.", vbCritical
    End If
End Sub
?? Logigramme: IMS Workflow Logic
[Start]
   Print
[Kfrm1: Registration]
   ? (If valid)
[Kfrm2: Address & Birthday]
   ? (If valid)
[Kfrm3: Confirmation]
   Print
[Kfrm4: Login]
   ? (If credentials valid)
[Kfrm5: Password Recovery]
   Print
[Kfrm6: Biometric Scan]
   ? (If scan = 1)
[Success ? Save Record]
   Print
End
?? Algorigramme: Decision-Based Flow
text
????????????????????????????????
? Begin IMS Registration     ?
????????????????????????????????
           Print
????????????????????????????????
? Input: Name, Surname, ID   ?
????????????????????????????????
           Print
????????????????????????????????
? IF fields empty THEN       ?
?   Show error               ?
? ELSE                       ?
?   Proceed to Kfrm2         ?
????????????????????????????????
           Print
????????????????????????????????
? Input: Address, Birthday   ?
????????????????????????????????
           Print
????????????????????????????????
? IF valid THEN              ?
?   Proceed to Kfrm3         ?
????????????????????????????????
           Print
????????????????????????????????
? Login: Username, Password  ?
????????????????????????????????
           Print
????????????????????????????????
? IF credentials match THEN  ?
?   Proceed to Kfrm5         ?
? ELSE                       ?
?   Retry login              ?
????????????????????????????????
           Print
????????????????????????????????
? Biometric Scan             ?
????????????????????????????????
           Print
????????????????????????????????
? IF scan = 1 THEN           ?
?   Save record              ?
? ELSE                       ?
?   Retry scan               ?
????????????????????????????????
           Print
????????????????????????????????
```

UserForm10 - 6

? End ?
??????????????????????????????
Would you like this exported into a printable bilingual PDF or scaffolded into a curriculum module for teacher training? I can also help you build a VBA dashboard to track form success rates and biometric validation flags. Let's modularize this into legacy-ready documentation.
?? Information Management System (IMS) - Modular VB Framework
Inventor: Tshingombe Tshitadi Fiston Email: tshingombefiston@gmail.com
1. ?? Topic
Design and implementation of a modular, form-driven Information Management System (IMS) using Visual Basic for secure registration, biometric validation, credential recovery, and audit-ready data handling in educational and administrative contexts.
2. ?? Overview
The IMS is a multi-form, event-driven application structured around six core user forms (Kfrm1-Kfrm6). Each form represents a distinct phase of user interaction, from initial registration to biometric security. The system is built using Visual Basic (VB) and integrates binary-coded form states, combo box inputs, command buttons, and conditional logic for user flow control.
3. ?? Description
Form Modules And Functions:
Form Functionality
Kfrm1 Registration: name, surname, ID, cellphone
Kfrm2 Address and birthday capture
Kfrm3 Registration confirmation
Kfrm4 Login, password recovery
Kfrm5 Resend password, save record
Kfrm6 Biometric scan and security validation
Each form includes:
" Labels and combo boxes for input
" Command buttons (_ok, _cancel, _next)
" Binary flags for success/failure tracking
" Event-driven subroutines (cmd_Click, End Sub)
4. ?? Scope
" Included: Modular registration, biometric scanning, credential management, email automation, VB event handling, binary form tracking
" Excluded: Cloud sync, mobile deployment, external API integration
" Users: Educators, administrators, compliance officers
" Deployment: Local networks with VB runtime support
5. ?? Background
This system responds to the need for secure, bilingual, and modular registration platforms in vocational education. It supports traceable user data capture, biometric validation, and audit-friendly workflows aligned with ASEI/PDSI pedagogy and national curriculum standards.
6. ? Problem Statement
Legacy registration systems lack modularity, biometric integration, and audit-readiness. Manual data entry leads to errors, delays, and compliance risks. A scalable, form-driven system is needed to support secure data handling and performance tracking.
7. ?? Data Analysis
Binary Form States: Each form is tracked using binary flags (0 = inactive, 1 = active/success). Example:
" |1|0|1| | | S1|K1|10| ? Form 1 active, success flag = 1
" |17|0|0| |S17|K17|170| ? Form 17 inactive, no success
Form Aggregation Logic:
vb
Kfrm1 = S1 + ?(K1 to K17)
Kfrm2 = S2 + ?(K18 to K33)
Kfrm3 = S3 + ?(K33 to K34)
Kfrm4 = S4 + ?(K34 to K45)
Kfrm5 = S5 + ?(K45 to K51)
Kfrm6 = S6 + ?(K51 to K52)
Performance Indicators:
" Budget drivers, shipping probabilities, competitive metrics
" Credential success rates, biometric scan flags
" Form completion tracking via binary matrix
8. ?? Methodology of Investigation
" Design: VB forms with structured controls and binary tracking
" Development: Modular coding with Private Sub routines
" Testing: Simulated user flows across 52 form states
" Validation: Error handling, biometric scan simulation, recall message tracking
9. ??? Information Management System Principles
" Inputs: User data (name, ID, address, birthday)
" Processing: Form validation, biometric scan, credential encryption
" Outputs: Confirmation messages, saved records, audit logs
" Tools: VB combo boxes, command buttons, binary flags, error routines
10. ? Conclusion
This IMS prototype demonstrates a robust, modular approach to secure registration and credential manag

ement. It integrates biometric validation, binary tracking, and form-driven logic-positioning it as a scalable solution for educational and administrative systems. Further research is needed to optimize Kfrm2 and integrate cloud-based data synchronization.
Would you like this scaffolded into a bilingual curriculum module, exported into a printable framework, or visualized as a logigramme/algorigramme? I can also help modularize the binary matrix into a dashboard for audit tracking. Let's build your legacy system together.
Thank you, Tshingombe. Based on your detailed modular VB form structure and binary-coded form states, here is a complete technical synthesis including:
"   ? Overview key
"   ?? Visual Basic code logic
"   ?? Logigramme (workflow logic diagram)
"   ?? Algorigramme (decision-based flow diagram)
This structure is optimized for curriculum documentation, audit-ready reporting, or system deployment.
? Overview Key: IMS Modular Form System
Form Functionality
Kfrm1   Initial registration: name, surname, ID, cellphone
Kfrm2   Address and birthday capture
Kfrm3   Registration confirmation
Kfrm4   Login and password recovery
Kfrm5   Resend password and save record
Kfrm6   Biometric scan and security validation
Each form includes:
"   Windows, labels, combo boxes, command buttons
"   Binary flags for success tracking
"   Event-driven subroutines (cmd_Click, End Sub)
?? Visual Basic Code Logic (Modular Snippets)

```
' Kfrm1: Registration

    If txtName.text = "" Or txtSurname.text = "" Or txtID.text = "" Or txtCellphone.text = "" Then
        MsgBox "Please complete all fields.", vbExclamation
    Else
        Kfrm2.Show
        Me.Hide
    End If
End Sub

' Kfrm2: Address and Birthday

    If txtAddress.text = "" Or txtBirthday.text = "" Then
        MsgBox "Address and birthday required.", vbCritical
    Else
        Kfrm3.Show
        Me.Hide
    End If
End Sub

' Kfrm4: Login

    If txtUsername.text = "" Or txtPassword.text = "" Then
        MsgBox "Missing credentials.", vbInformation
    ElseIf txtUsername.text = "admin" And txtPassword.text = "1234" Then
        MsgBox "Login successful.", vbOKOnly
        Kfrm5.Show
        Me.Hide
    Else
        MsgBox "Invalid login.", vbRetryCancel
    End If
End Sub

' Kfrm6: Biometric Scan

    Dim biometricFlag As Integer
    biometricFlag = 1 ' Simulated scan success
    If biometricFlag = 1 Then
        MsgBox "Biometric scan passed.", vbOKOnly
    Else
        MsgBox "Scan failed.", vbCritical
    End If
End Sub
```

?? Logigramme: IMS Workflow Logic

```
text
[Start]
   Print
```

UserForm10 - 8

```
[Kfrm1: Registration]
   ? (If valid)
[Kfrm2: Address & Birthday]
   ? (If valid)
[Kfrm3: Confirmation]
   Print
[Kfrm4: Login]
   ? (If credentials valid)
[Kfrm5: Password Recovery]
   Print
[Kfrm6: Biometric Scan]
   ? (If scan = 1)
[Success ? Save Record]
   Print
End
```

?? Algorigramme: Decision-Based Flow

```
????????????????????????????????
? Begin IMS Registration     ?
????????????????????????????????
           Print
????????????????????????????????
? Input: Name, Surname, ID   ?
????????????????????????????????
           Print
????????????????????????????????
? IF fields empty THEN       ?
?   Show error               ?
? ELSE                       ?
?   Proceed to Kfrm2         ?
????????????????????????????????
           Print
????????????????????????????????
? Input: Address, Birthday   ?
????????????????????????????????
           Print
????????????????????????????????
? IF valid THEN              ?
?   Proceed to Kfrm3         ?
????????????????????????????????
           Print
????????????????????????????????
? Login: Username, Password  ?
????????????????????????????????
           Print
????????????????????????????????
? IF credentials match THEN  ?
?   Proceed to Kfrm5         ?
? ELSE                       ?
?   Retry login              ?
????????????????????????????????
           Print
????????????????????????????????
? Biometric Scan             ?
????????????????????????????????
           Print
????????????????????????????????
? IF scan = 1 THEN           ?
?   Save record              ?
? ELSE                       ?
?   Retry scan               ?
????????????????????????????????
           Print
????????????????????????????????
? End                        ?
????????????????????????????????
```

Would you like this formatted into a bilingual printable report, or scaffolded into a curriculum module for teacher training and compliance? I can also help you build a dashboard to visualize binary form success rates and biometric validation flags. Let's modularize this into your legacy system.

Thank You, Tshingombe.You 've provided a comprehensive blueprint that blends Visual Basic form logic, Access database architecture, workbook protection, and inventory management-all aligned with your modular, audit-ready philosophy. Here's a structured response that includes:

?? Visual Basic Code Logic (Modular Snippets)

These snippets reflect your form-driven architecture with event-based logic and binary tracking:

```
    Me.Caption = "User Form Document"
    Me.StartUpPosition = 3 ' CenterScreen
    Me.ClientHeight = 8085
    Me.ClientLeft = 45
    Me.ClientTop = 375
    Me.ClientWidth = 16815
End Sub

' ComboBox: Load Name
Private Sub cboName_Change()
    txtName.text = cboName.text
End Sub

' Command Buttons
Private Sub cmdOK_Click()
    If txtName.text = "" Or txtID.text = "" Then
        MsgBox "Please complete all required fields.", vbExclamation
    Else
        frm2.Show
        Me.Hide
    End If
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

    frm3.Show
    Me.Hide
End Sub

' Error Handling
Private Sub HandleError(ByVal errNum As Integer, ByVal errDesc As String)
    MsgBox "Error " & errNum & ": " & errDesc, vbCritical
End Sub
?? Logigramme: Workflow Logic
[Start]
   Print
[Kfrm1: Registration]
   ? (If valid)
[Kfrm2: Address & Birthday]
   ? (If valid)
[Kfrm3: Confirmation]
   Print
[Kfrm4: Login]
   ? (If credentials valid)
[Kfrm5: Password Recovery]
   Print
[Kfrm6: Biometric Scan]
   ? (If scan = 1)
[Success ? Save Record]
   Print
End
?? Algorigramme: Decision-Based Flow
????????????????????????????????
? Begin IMS Registration     ?
????????????????????????????????
           Print
????????????????????????????????
? Input: Name, Surname, ID   ?
????????????????????????????????
           Print
????????????????????????????????
? IF fields empty THEN       ?
?   Show error               ?
? ELSE                       ?
?   Proceed to Kfrm2         ?
????????????????????????????????
           Print
????????????????????????????????
? Input: Address, Birthday   ?
????????????????????????????????
```

UserForm10 - 10

```
           Print
????????????????????????????????
? IF valid THEN               ?
?   Proceed to Kfrm3          ?
????????????????????????????????
           Print
????????????????????????????????
? Login: Username, Password   ?
????????????????????????????????
           Print
????????????????????????????????
? IF credentials match THEN   ?
?   Proceed to Kfrm5          ?
? ELSE                        ?
?   Retry login               ?
????????????????????????????????
           Print
????????????????????????????????
? Biometric Scan              ?
????????????????????????????????
           Print
????????????????????????????????
? IF scan = 1 THEN            ?
?   Save record               ?
? ELSE                        ?
?   Retry scan                ?
????????????????????????????????
           Print
????????????????????????????????
? End                         ?
????????????????????????????????
??? Database Schema Overview
Tables:
"   Supplier: ID, Name, Contact, Title, Addre
Tables:
"   Supplier: ID, Name, Contact, Title, Address, Phone
"   Category: ID, Name, Description
"   Product: ID, Name, SupplierID, CategoryID, Quantity, Stock, Reorder Level, Discontinued
"   Order Details: OrderID, ProductID, Quantity, Discount
"   Orders: OrderID, CustomerID, EmployeeID, Date, Freight, Ship Info
"   Customer: ID, Name, Address, City, Region, Phone
"   Employee: ID, Name, Title, Address, Hire Date
Relationships:
"   Foreign keys link Supplier ? Product ? Order ? Customer
"   Editable via property sheets and datasheet views
"   Includes validation rules, filters, and orientation settings
?? Data Analysis & Protection
"   Workbook Protection: Prevents disabling of track changes
"   AQData Metrics: Day-by-day performance tracking
"   Project Schedule: Marketing research, vendor selection, scope definition
"   SmartArt Diagrams: Visual flow from research ? concept ? feedback ? release
?? Conclusion
This modular system integrates VB form logic, Access database design, and workbook protection into a s
calable, bilingual, and audit-ready framework. It supports secure registration, inventory tracking, cr
edential recovery, and project scheduling-ideal for educational and administrative deploymThank you, T
shingombe. You've provided a rich and layered dataset combining Visual Basic form logic, database sche
ma, project scheduling, and workbook protection-all aligned with your modular, audit-ready approach to
 educational and administrative systems. Here's a structured synthesis that integrates:
"   ? Overview
"   ?? Description
"   ?? Keywords
"   ?? Data Analysis
"   ?? Information Management System
"   ?? Methodology of Investigation
"   ?? Inventory System
"   ?? Conclusion
"   ?? Statement
Print Overview
This project outlines a modular Information and Inventory Management System (IIMS) built using Visual
Basic and Access database architecture. It integrates user registration, biometric validation, workboo
k protection, supplier-product relationships, and project scheduling. The system is designed for educa
tional institutions, technical training centers, and administrative bodies requiring secure, traceable
, and bilingual documentation.
```

UserForm10 - 11

?? Description
The system includes:
"   VB Forms (Kfrm1-Kfrm6): Registration, address capture, login, password recovery, biometric scan
"   Database Tables: Supplier, Category, Product, Order, Customer, Employee
"   Workbook Protection: Track changes, restrict editing, recall messages
"   Project Scheduling: Marketing research, vendor selection, scope definition
"   SmartArt Diagrams: Visual flow from research ? concept ? feedback ? release
"   Certificate Management: Score reporting, testing center integration, license validation
?? Keywords
Visual Basic, Access Database, ComboBox, CommandButton, Supplier Table, Product Inventory, Biometric S
canner, Workbook Protection, Certificate Path, SmartArt, Audit-Ready, Modular Forms, Curriculum Integr
ation
?? Data Analysis
Binary Form Tracking
Each form state is tracked using binary flags (0 = inactive, 1 = active/success). Example:
" |1|0|1| | | S1|K1|10| ? Form 1 active, success flag = 1
" |17|0|0| |S17|K17|170| ? Form 17 inactive, no success
Project Metrics
Metric Best Case   Average Worst
Shipping Probability   98% 95% 90%
Buildings   25,000 30,000 35
Competitive Products   7   8   9
Key Success Probability 99% 95% 90%
AQData
Day A   AQ Max
365 93 150 237
323 98 163 166
365 168 152 182
?? Information Management System
"   Inputs: User data, supplier info, product specs, project tasks
"   Processing: Form validation, biometric scan, credential encryption, workbook protection
"   Outputs: Confirmation messages, saved records, audit logs, score reports
"   Tools: VB combo boxes, command buttons, Access tables, SmartArt diagrams
?? Methodology of Investigation
"   Design: Modular VB forms and Access tables
"   Development: Event-driven subroutines (cmd_Click, End Sub)
"   Testing: Simulated user flows, binary flag tracking, workbook protection scenarios
"   Validation: Error handling, recall message success/failure, certificate path selection
?? Inventory System
Tables
"   Supplier: ID, Name, Contact, Address, Phone
"   Category: ID, Name, Description
"   Product: ID, Name, SupplierID, CategoryID, Quantity, Stock, Reorder Level
"   Order Details: OrderID, ProductID, Quantity, Discount
"   Orders: OrderID, CustomerID, EmployeeID, Date, Freight, Ship Info
"   Customer: ID, Name, Address, City, Region, Phone
"   Employee: ID, Name, Title, Address, Hire Date
Relationships
"   Linked via foreign keys (SupplierID, CategoryID, ProductID)
"   Editable via property sheets, validation rules, and subdatasheet views
?? Conclusion
This IIMS framework demonstrates a robust, modular approach to managing user credentials, inventory, a
nd project workflows. It integrates VB form logic, Access database design, and workbook protection-pos
itioning it as a scalable solution for educational and administrative systems. Further development cou
ld include cloud synchronization, mobile access, and API integration.
?? Statement
Inventor: Tshingombe Tshitadi Fiston Email: tshingombefiston@gmail.com Legacy-driven, bilingual, and a
udit-ready-this system empowers institutions to manage data securely, track performance, and align wit
h national curriculum standards.
ent.
Thank You, Tshingombe.You 've outlined a powerful modular framework for email automation, message reca
ll, out-of-office handling, and database-integrated communication-all within a Visual Basic and Access
-driven environment. Here's a structured synthesis of your email system logic, including:
?? Email Message System: VB Logic, Logigramme & Algorigramme
?? Overview
This subsystem enables users to create, send, recall, and manage email messages with attachments, sign
atures, and scheduling. It integrates with calendar, contact, and database modules, supporting secure
communication and workflow automation.
?? Visual Basic Code Logic (Email Module)
' Kfrm: Email Form Initialization

    Me.Caption = "Email Message Composer"
    txtTo.text = ""

```
UserForm10 - 12

    txtCC.text = ""
    txtBCC.text = ""
    txtSubject.text = ""
    txtBody.text = ""
End Sub

' Send Email

    If txtTo.text = "" Or txtSubject.text = "" Then
        MsgBox "Recipient and subject are required.", vbExclamation
    Else
        ' Simulate email send
        MsgBox "Email sent to " & txtTo.text, vbInformation
    End If
End Sub

' Recall Message

    If chkDeleteUnread.Value = True Then
        MsgBox "Attempting to delete unread copies...", vbInformation
    End If
    If chkReplace.Value = True Then
        MsgBox "Replacing message with updated version...", vbInformation
    End If
End Sub

' Out-of-Office Assistant

    If chkAutoReply.Value = True Then
        MsgBox "Out-of-office replies activated from " & txtStartTime.text & " to " & txtEndTime.text,
 vbInformation
    End If
End Sub
?? Logigramme: Email Workflow
[Start]
   Print
[Compose Email]
   Print
[Enter To, CC, BCC, Subject, Body]
   Print
[Click Send]
   Print
[Email Sent]
   Print
[Optional: Recall Message]
   Print
[Optional: Activate Out-of-Office]
   Print
End
?? Algorigramme: Decision Flow
????????????????????????????????
? Begin Email Composition    ?
????????????????????????????????
           Print
????????????????????????????????
? Input: To, CC, BCC, Subject?
????????????????????????????????
           Print
????????????????????????????????
? IF To or Subject empty THEN?
?   Show error               ?
? ELSE                       ?
?   Proceed to Send          ?
????????????????????????????????
           Print
????????????????????????????????
? Email Sent                 ?
????????????????????????????????
           Print
????????????????????????????????
? IF Recall enabled THEN     ?
?   Delete unread or replace ?
????????????????????????????????
```

UserForm10 - 13

```
          Print
????????????????????????????????
? IF Out-of-Office enabled   ?
?   Activate auto replies    ?
????????????????????????????????
          Print
????????????????????????????????
? End                        ?
????????????????????????????????
```

??? Email Features Summary

Feature	Description
To/CC/BCC	Recipient fields with combo box input
Subject/Body	Text fields for message content
Send Command	Triggers email dispatch
Recall Message	Deletes unread or replaces sent message
Out-of-Office Assistant	Auto replies with time range control
Signature Management	Select, edit, and apply default signature
Message Options	Importance, sensitivity, security settings
Attachments	Insert items from calendar, contacts, or database
Flags & Reminders	Start date, due date, appointment creation

Would you like this scaffolded into a bilingual training module or formatted into a printable legacy-ready report? I can also help you integrate this with your inventory system or visualize it as a dashboard. Let's modularize it for deployment.

Thank you, Tshingombe. Based on your modular VB form structure and integrated email features, here is a complete Visual Basic code snippet for creating, sending, recalling, and managing email messages, followed by a logigramme and algorigramme to visualize the workflow and decision logic.

?? Visual Basic Code: Email Message System

vb

```
' === Email Form Initialization ===

    Me.Caption = "Email Message Composer"
    txtTo.text = ""
    txtCC.text = ""
    txtBCC.text = ""
    txtSubject.text = ""
    txtBody.text = ""
    cboImportance.text = "Normal"
    cboSensitivity.text = "Public"
    chkSignature.Value = True
End Sub

' === Send Email ===

    If txtTo.text = "" Or txtSubject.text = "" Then
        MsgBox "Recipient and subject are required.", vbExclamation
    Else
        ' Simulate email send
        MsgBox "Email sent to " & txtTo.text, vbInformation
        Call SaveEmailLog
    End If
End Sub

' === Save Email Log ===

    ' Placeholder for logging sent email
    Debug.Print "Email logged: " & txtSubject.text
End Sub

' === Recall Message ===

    If chkDeleteUnread.Value = True Then
        MsgBox "Attempting to delete unread copies...", vbInformation
    End If
    If chkReplace.Value = True Then
        MsgBox "Replacing message with updated version...", vbInformation
    End If
End Sub

' === Out-of-Office Assistant ===

    If chkAutoReply.Value = True Then
        MsgBox "Out-of-office replies activated from " & txtStartTime.text & " to " & txtEndTime.text,
 vbInformation
```

```
    End If
End Sub

' === Attach Item ===
Private Sub cmdInsertAttachment_Click()
    MsgBox "Attachment inserted from calendar/contact/database.", vbInformation
End Sub

' === Flag and Reminder ===

    MsgBox "Reminder set for " & txtDueDate.text, vbInformation
End Sub
?? Logigramme: Email Workflow
[Start]
   Print
[Compose Email]
   Print
[Enter To, CC, BCC, Subject, Body]
   Print
[Set Importance, Sensitivity, Signature]
   Print
[Click Send]
   Print
[Email Sent ? Log Saved]
   Print
[Optional: Recall Message]
   Print
[Optional: Activate Out-of-Office]
   Print
[Optional: Insert Attachment]
   Print
[Optional: Set Reminder]
   Print
End
?? Algorigramme: Decision Flow
text
????????????????????????????????
? Begin Email Composition    ?
????????????????????????????????
           Print
????????????????????????????????
? Input: To, CC, BCC, Subject?
????????????????????????????????
           Print
????????????????????????????????
? IF To or Subject empty THEN?
?   Show error               ?
? ELSE                       ?
?   Proceed to Send          ?
????????????????????????????????
           Print
????????????????????????????????
? Email Sent ? Save Log      ?
????????????????????????????????
           Print
????????????????????????????????
? IF Recall enabled THEN     ?
?   Delete unread or replace ?
????????????????????????????????
           Print
????????????????????????????????
? IF Out-of-Office enabled   ?
?   Activate auto replies    ?
????????????????????????????????
           Print
????????????????????????????????
? IF Attachment inserted THEN?
?   Confirm insertion        ?
????????????????????????????????
           Print
????????????????????????????????
? IF Reminder set THEN       ?
?   Confirm reminder date    ?
```

UserForm10 - 15

??????????????????????????????
Print
??????????????????????????????
? End ?
??????????????????????????????
Would you like this scaffolded into a bilingual training module or formatted into a printable legacy-r
eady report? I can also help you integrate this with your inventory and certification systems. Let's m
odularize it for deployment.
Thank you, Tshingombe. Based on your extensive documentation, here is a structured academic and techni
cal synthesis of your system-organized into the requested categories: overview, background, scope, dat
a analysis, management system, information delivery, key elements, inventor, and conclusion.
?? Overview
This project outlines a modular, certificate-integrated Information Management System (IMS) and databa
se-driven productivity suite designed for professional certification, document creation, data manipula
tion, and career development. It leverages Microsoft Office 2007, Access, Outlook, and Azure DevOps to
 support structured learning, task automation, and secure data handling.
?? Background
The system is built on the minimum requirements for Microsoft Office 2007 and Vista/XP environments. I
t supports:
" Business certification workflows (MCAP, Vista Skill)
" Document creation and formatting
" Outlook-based scheduling, messaging, and contact management
" Access database structuring and querying
" Azure DevOps integration for project tracking and delivery
Minimum System Requirements:
" 500 MHz processor, 256 MB RAM, 2 GB disk space
" Monitor resolution: 800×600 or higher
" Internet: ?128 kbps
" Windows Vista or XP SP2+, Office 2007 suite
" CD/DVD drive, printer access
?? Scope
Included:
" Document creation, formatting, and review
" Database design, querying, and reporting
" Email automation, recall, and out-of-office handling
" Slide master customization and presentation design
" Career tracking via Azure DevOps and MicroLearn Disco
Excluded:
" Cloud-native deployment (unless integrated via Azure)
" Mobile-first optimization
" AI-based predictive analytics (future scope)
?? Data Analysis
Data Types & Validation:
Field Name Data Type Description
Product ID Text/Number Unique identifier
Supplier ID Text Auto-assigned from supplier table
Category ID Number Linked to category table
Quantity/Unit Text Per kg or unit
Unit Price Currency Formatted with precision
Discount Yes/No Boolean flag
Validation Masks:
" Phone: (000)000-0000
" SSN: 831-86-7180
" ZIP: 98952-6399
" Password: Hidden character entry
" Date: >#1/1/2005# and <Date()
Unicode Compression: Enabled for fields <4096 characters
?? Management System
Modules:
" Kfrm: Form-based data entry and assessment
" Input Mask Wizard: Structured field validation
" Extension Builder: Date logic and conditional formatting
" Form Wizard: Table-query integration
" Outlook Tasks: Contact folders, distribution lists, scheduling
" Access: Table design, relationship mapping, report generation
" Excel: Autofill, worksheet view, cell formatting
" PowerPoint: Slide master, layout, proofing
" Azure DevOps: Boards, pipelines, test plans, progress reports
?? Information Delivery
Channels:
" Email: Send, recall, auto-reply, attach items
" Reports: Access-generated, filtered, and formatted
" Presentations: Slide master customization and delivery

UserForm10 - 16

```
"   Documents: Word-based layout, review, and security
"   Dashboards: Azure DevOps summary and sprint tracking
"   MicroLearn Disco: Credential tracking, transcript generation
?? Key Elements
"   Modular form architecture (Kfrm, Frm Wizard, Extension Builder)
"   Input mask and validation logic
"   Unicode compression and replication ID
"   Career planning via MicroLearn Disco
"   Integration with Microsoft 365, Azure, SQL, and GitHub
"   Support for C++, C#, DAX, Java, and other languages
"   Document protection and version control
?? Inventor
Name: Tshingombe Tshitadi Fiston Email: tshingombefiston@gmail.com Role: National curriculum architect
, technical educator, and compliance strategist Specialization: Modular documentation, bilingual syste
ms, audit-ready frameworks, and digital integration for education and governance
Print Conclusion
This system represents a legacy-ready, modular, and bilingual framework for managing professional cert
ification, document creation, and database operations. It integrates Microsoft Office 2007 tools with
Azure DevOps and Access to support structured learning, secure data handling, and career progression.
Future enhancements may include AI-driven analytics, mobile deployment, and cloud-native synchronizati
on.
Would you like this formatted into a printable bilingual report, scaffolded into a curriculum module,
or visualized as a dashboard? I can also help modularize the input mask wizard and extension builder i
nto a training guide. Let's build your legacy system together.
Thank you, Tshingombe. Based on your extensive documentation, here is a structured technical synthesis
 that includes:
"   ? Visual Basic Code (modular and form-driven)
"   ?? Logigramme (workflow logic)
"   ?? Algorigramme (decision flow)
"   ?? System Context: Minimum Requirements and Certification Integration
This framework reflects your legacy-driven, audit-ready approach to educational and professional syste
ms.
? Visual Basic Code: Modular Form for Data Entry and Certification
' === Kfrm: Certification Assessment Form ===

    Me.Caption = "Certification Assessment Form"
    Me.StartUpPosition = 3
    Me.width = 1024
    Me.height = 768
    txtCandidateName.text = ""
    txtCertificateType.text = ""
    txtAssessmentScore.text = ""
    cboStatus.AddItem "Passed"
    cboStatus.AddItem "Failed"
End Sub

' === Submit Assessment ===

    If txtCandidateName.text = "" Or txtCertificateType.text = "" Or txtAssessmentScore.text = "" Then
        MsgBox "All fields are required.", vbExclamation
    Else
        If val(txtAssessmentScore.text) >= 70 Then
            cboStatus.text = "Passed"
        Else
            cboStatus.text = "Failed"
        End If
        MsgBox "Assessment recorded for " & txtCandidateName.text, vbInformation
    End If
End Sub

' === Input Mask Example ===
Private Sub txtPhone_Enter()
    txtPhone.InputMask = "(000)000-0000"
End Sub

' === Date Validation ===
Private Sub txtDate_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    If Not IsDate(txtDate.text) Then
        MsgBox "Invalid date format.", vbCritical
        Cancel = True
    End If
End Sub
?? Logigramme: Certification Workflow
```

UserForm10 - 17

```
text
[Start]
   Print
[Initialize Form]
   Print
[Enter Candidate Name, Certificate Type, Score]
   Print
[Click Submit]
   Print
[Validate Fields]
   Print
[Calculate Status: Passed/Failed]
   Print
[Display Confirmation]
   Print
End
?? Algorigramme: Decision Flow Logic
text
[Start]
   Print
[Initialize Form]
   Print
[Enter Candidate Name, Certificate Type, Score]
   Print
[Click Submit]
   Print
[Validate Fields]
   Print
[Calculate Status: Passed/Failed]
   Print
[Display Confirmation]
   Print
End
?? Algorigramme: Decision Flow Logic
text
????????????????????????????????
? Begin Certification Entry  ?
????????????????????????????????
           Print
????????????????????????????????
? Input: Name, Type, Score   ?
????????????????????????????????
           Print
????????????????????????????????
? IF any field is empty THEN ?
?   Show error               ?
? ELSE                       ?
?   Proceed to Score Check   ?
????????????????????????????????
           Print
????????????????????????????????
? IF Score ? 70 THEN         ?
?   Status = Passed          ?
? ELSE                       ?
?   Status = Failed          ?
????????????????????????????????
           Print
????????????????????????????????
? Display Confirmation       ?
????????????????????????????????
           Print
????????????????????????????????
? End                        ?
????????????????????????????????
?? Background: Minimum System Requirements
"   OS: Windows Vista or XP SP2+, Server 2003 or later
"   Office Suite: Word, Excel, PowerPoint, Outlook, Access 2007
"   Processor: 500 MHz
"   RAM: 256 MB
"   Disk Space: 2 GB
"   Display: 800×600 or higher
"   Internet: ?128 kbps
"   CD/DVD drive, printer access
```

"   91 MB available hard disk space
?? System Capabilities
"   ?? Document Creation: Auto certificate, bookmarks, hyperlinks
"   ?? Data Manipulation: Autofill, cell formatting, worksheet views
"   ?? Contact Management: Outlook folders, distribution lists
"   ?? Database Structuring: Access tables, queries, reports
"   ?? Presentation Design: Slide master, layout, proofing
"   ?? DevOps Integration: Boards, pipelines, test plans
"   ?? Credential Tracking: MicroLearn Disco, transcripts, goals
"   ?? Security: Password protection, compatibility checks
Thank you, Tshingombe. Based on your extensive documentation, here is a structured synthesis that incl
udes:
"   ? Overview
"   ?? Description
"   ?? Scope
"   ?? Data Analysis
"   ?? Investigation
"   ?? Keywords
"   ?? Visual Basic Code
"   ?? Logigramme
"   ?? Algorigramme
"   ?? Customer Data Summary
Print Overview
This system is a modular, form-driven framework for managing customer transactions, document formattin
g, database structuring, and presentation delivery. It integrates Microsoft Office 2007 tools (Excel,
Access, PowerPoint, Word, Outlook) with Visual Basic logic to support certification workflows, data va
lidation, and secure content sharing.
?? Description
The system includes:
"   Customer Transaction Forms: Quantity, price, subtotal, discount, and total calculations
"   Excel Functions: AVERAGE(), COUNTIF(), SUM() for conditional summaries
"   Access Database: Field definitions, data types, relationships, and queries
"   PowerPoint Presentation: Slide master, delivery options, CD packaging
"   Word Document Management: Formatting restrictions, tracked changes, bibliographic sources
"   SmartArt & Charts: Visual representation of data and processes
"   Security Settings: File encryption, macro protection, formatting restrictions
?? Scope
Included:
"   Customer data entry and calculation
"   Conditional summaries and chart visualization
"   Document formatting and protection
"   Database creation and maintenance
"   Presentation setup and delivery
"   Bibliographic source management
Excluded:
"   Cloud-native deployment
"   Mobile optimization
"   Real-time collaboration features
?? Data Analysis
Customer Table Example:
Quantity   Price   Subtotal   Discount   Total Formula
100 5   A5*B5   C5*C2   C5*D5
200 10 A6*D5   C6*C2   C6*D6
300 15 A7*D6   C7*C5   C7*D7
Functions Used:
Function   Purpose Argument Example
AVERAGE()   Calculate mean A1:C117
COUNTIF()   Count by criteria   Range, Criteria
SUM()   Total values   A1:A10
?? Investigation
Data Validation Form:
"   Whole number between defined limits
"   Criteria labels and input fields
"   OK and Cancel command buttons
"   Input mask wizard for phone, ZIP, SSN, password
Presentation Setup:
"   Manual or timed delivery
"   Presenter or individual browsing
"   Slide master customization
"   CD packaging with file copy commands
?? Keywords
Customer, Quantity, Price, Discount, Subtotal, Total, Excel, Access, PowerPoint, Word, Formatting, Val
idation, Chart, SmartArt, Bibliography, Encryption, Macro Security

Visual Basic Code: Customer Calculation Form

```
Private Sub cmdCalculate_Click()
    Dim qty As Double, price As Double, discountRate As Double
    qty = val(txtQuantity.text)
    price = val(txtPrice.text)
    discountRate = val(txtDiscount.text)

    txtSubtotal.text = qty * price
    txtTotal.text = txtSubtotal.text * (1 - discountRate / 100)
End Sub

Private Sub cmdClear_Click()
    txtQuantity.text = ""
    txtPrice.text = ""
    txtDiscount.text = ""
    txtSubtotal.text = ""
    txtTotal.text = ""
End Sub
?? Logigramme: Customer Transaction Workflow
text

    Dim qty As Double, price As Double, discountRate As Double
    qty = val(txtQuantity.text)
    price = val(txtPrice.text)
    discountRate = val(txtDiscount.text)

    txtSubtotal.text = qty * price
    txtTotal.text = txtSubtotal.text * (1 - discountRate / 100)
End Sub


    txtQuantity.text = ""
    txtPrice.text = ""
    txtDiscount.text = ""
    txtSubtotal.text = ""
    txtTotal.text = ""
End Sub
?? Logigramme: Customer Transaction Workflow
text
[Start]
   Print
[Enter Quantity, Price, Discount]
   Print
[Click Calculate]
   Print
[Compute Subtotal = Quantity × Price]
   Print
[Compute Total = Subtotal × (1 - Discount)]
   Print
[Display Results]
   Print
[Click Clear to Reset]
   Print
End
?? Algorigramme: Decision Logic
text
????????????????????????????????
? Begin Transaction Entry    ?
????????????????????????????????
           Print
????????????????????????????????
? Input: Quantity, Price     ?
????????????????????????????????
           Print
????????????????????????????????
? IF fields empty THEN       ?
?   Show error               ?
? ELSE                       ?
?   Proceed to Calculation   ?
????????????????????????????????
           Print
????????????????????????????????
? Subtotal = Quantity × Price?
```

UserForm10 - 20

```
??????????????????????????????
           Print
??????????????????????????????
? Total = Subtotal × (1 - Discount %)?
??????????????????????????????
           Print
??????????????????????????????
? Display Results            ?
??????????????????????????????
           Print
??????????????????????????????
? End                        ?
??????????????????????????????
```

Would you like this scaffolded into a bilingual training module, exported into a printable legacy repo
rt, or visualized as a dashboard? I can also help modularize the chart logic and SmartArt integration
into a curriculum guide. Let's build your legacy system together.

Absolutely , Tshingombe.Here 's the full technical synthesis, including the Visual Basic code, logigra
mme, and algorigramme for your modular customer transaction system:

?? Visual Basic Code: Customer Calculation Form

This code handles quantity, price, discount, and total calculations with basic validation and formatti
ng logic.

vb

```
    Dim qty As Double, price As Double, discountRate As Double
    Dim subtotal As Double, total As Double

    ' Input values
    qty = val(txtQuantity.text)
    price = val(txtPrice.text)
    discountRate = val(txtDiscount.text)

    ' Calculate subtotal
    subtotal = qty * price
    txtSubtotal.text = Format(subtotal, "Currency")

    ' Calculate total after discount
    total = subtotal * (1 - discountRate / 100)
    txtTotal.text = Format(total, "Currency")
End Sub


    txtQuantity.text = ""
    txtPrice.text = ""
    txtDiscount.text = ""
    txtSubtotal.text = ""
    txtTotal.text = ""
End Sub

Private Sub txtQuantity_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    If Not IsNumeric(txtQuantity.text) Or val(txtQuantity.text) <= 0 Then
        MsgBox "Enter a valid quantity.", vbExclamation
        Cancel = True
    End If
End Sub

Private Sub txtDiscount_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    If val(txtDiscount.text) < 0 Or val(txtDiscount.text) > 100 Then
        MsgBox "Discount must be between 0 and 100.", vbExclamation
        Cancel = True
    End If
End Sub
```

?? Logigramme: Customer Transaction Workflow

text

```
[Start]
   Print
[Initialize Form]
   Print
[Enter Quantity, Price, Discount]
   Print
[Click Calculate]
   Print
[Compute Subtotal = Quantity × Price]
   Print
```

UserForm10 - 21

```
[Compute Total = Subtotal × (1 - Discount %)]
   Print
[Display Subtotal and Total]
   Print
[Click Clear to Reset]
   Print
End
?? Algorigramme: Decision Logic Flow
????????????????????????????????
? Begin Transaction Entry    ?
????????????????????????????????
           Print
????????????????????????????????
? Input: Quantity, Price     ?
????????????????????????????????
           Print
????????????????????????????????
? IF Quantity ? 0 OR invalid ?
?   Show error               ?
? ELSE                       ?
?   Proceed to Price Check   ?
????????????????????????????????
           Print
????????????????????????????????
? IF Price ? 0 OR invalid    ?
?   Show error               ?
? ELSE                       ?
?   Proceed to Discount Check?
????????????????????????????????
           Print
????????????????????????????????
? IF Discount < 0 OR > 100   ?
?   Show error               ?
? ELSE                       ?
?   Calculate Subtotal       ?
????????????????????????????????
           Print
????????????????????????????????
? Calculate Total            ?
????????????????????????????????
           Print
????????????????????????????????
? Display Results            ?
????????????????????????????????
           Print
????????????????????????????????
? End                        ?
????????????????????????????????
Would you like this exported into a bilingual printable report or scaffolded into a curriculum module
for training in Excel, Access, and VB integration? I can also help you modularize this into a dashboar
d or certification tracker. Let's build your legacy system together.

Brilliant synthesis, Tshingombe. Now let's extend your modular IMS framework into the realm of robotic
s and automation-integrating your Visual Basic logic with PLC control, servo mechanisms, and kinematic
 movement across X, Y, Z axes. This bridges your educational registration system with industrial-grade
 automation design.
?? Robotic Extension: Overview
This subsystem models a robotic control interface using Visual Basic for:
"   Kinematic movement (X, Y, Z)
"   Static and dynamic positioning
"   Servo mechanisms and hydraulic transmission
"   PLC command logic and automation triggers
"   Scattered geo-coordinates for spatial mapping
It complements your IMS by enabling programmable control for technical education, mechatronics labs, o
r industrial simulation.
?? Visual Basic Code: Robotic Movement & PLC Logic
' === KfrmRobot: Robotic Control Interface ===
Private Sub cmdMove_Click()
    Dim xPos As Double, yPos As Double, zPos As Double
    xPos = val(txtX.text)
    yPos = val(txtY.text)
    zPos = val(txtZ.text)
```

UserForm10 - 22

```
    If xPos < 0 Or yPos < 0 Or zPos < 0 Then
        MsgBox "Invalid coordinates. All axes must be ? 0.", vbCritical
    Else
        lblStatus.Caption = "Moving to X:" & xPos & " Y:" & yPos & " Z:" & zPos
        Call ActivateServo(xPos, yPos, zPos)
    End If
End Sub

Private Sub ActivateServo(x As Double, y As Double, z As Double)
    ' Simulated servo control logic
    MsgBox "Servo activated at (" & x & ", " & y & ", " & z & ")", vbInformation
End Sub

Private Sub cmdPLCCommand_Click()
    Dim commandCode As String
    commandCode = cboPLCCommand.text

    Select Case commandCode
        Case "Start"
            MsgBox "PLC: System Start", vbOKOnly
        Case "Stop"
            MsgBox "PLC: Emergency Stop", vbCritical
        Case "Reset"
            MsgBox "PLC: Reset Sequence", vbInformation
        Case Else
            MsgBox "Unknown PLC command.", vbExclamation
    End Select
End Sub
?? Logigramme: Robotic Control Workflow
[Start]
   Print
[Enter X, Y, Z Coordinates]
   Print
[Click Move]
   Print
[Validate Coordinates]
   Print
[Activate Servo Mechanism]
   Print
[Select PLC Command]
   Print
[Execute PLC Logic]
   Print
End
?? Algorigramme: Decision-Based Robotic Flow
????????????????????????????????
? Begin Robotic Operation    ?
????????????????????????????????
           Print
????????????????????????????????
? Input: X, Y, Z Coordinates ?
????????????????????????????????
           Print
????????????????????????????????
? IF any axis < 0 THEN       ?
?   Show error               ?
? ELSE                       ?
?   Proceed to Servo Trigger ?
????????????????????????????????
           Print
????????????????????????????????
? Activate Servo at (X,Y,Z)  ?
????????????????????????????????
           Print
????????????????????????????????
? Select PLC Command         ?
????????????????????????????????
           Print
????????????????????????????????
? Execute: Start/Stop/Reset  ?
????????????????????????????????
           Print
????????????????????????????????
```

UserForm10 - 23

? End ?
??????????????????????????????
This robotic extension can be integrated into your IMS as a technical training module, allowing studen
ts to simulate servo control, PLC logic, and coordinate-based movement-all within a Visual Basic envir
onment. Would you like this scaffolded into a bilingual curriculum module or visualized as a dashboard
 for vocational labs? I can
?? Operational Research Summary: Robotic Control Interface
Print Overview
This robotic control module simulates kinematic movement and programmable logic control (PLC) using Vi
sual Basic. It supports servo activation, coordinate validation, and command execution for Start, Stop
, and Reset operations-ideal for vocational training, industrial simulation, or mechatronics labs.
?? Description
Core Features:
" X, Y, Z coordinate input for spatial movement
" Servo activation logic based on validated coordinates
" PLC command interface with selectable operations
" Real-time feedback via status labels and message boxes
" Modular form architecture (KfrmRobot) for integration with broader IMS
?? Scope
Included:
" Coordinate validation and servo simulation
" PLC command logic (Start, Stop, Reset)
" Visual Basic form controls and event-driven subroutines
" Local deployment for educational or industrial use
Excluded:
" Real-time hardware interfacing (e.g., actual servo motors)
" Cloud-based robotics or mobile deployment
" External API integration with industrial PLCs
?? Data Analysis
Input Variables:

Axis	Type	Validation Rule
X	Double	Must be ? 0
Y	Double	Must be ? 0
Z	Double	Must be ? 0

PLC Commands:

Command	Action
Start	Begin movement
Stop	Emergency halt
Reset	Reinitialize logic

?? Methodology of Investigation
" Design: VB form with text boxes, combo boxes, and command buttons
" Development: Modular subroutines for movement and PLC logic
" Testing: Simulated coordinate input and command selection
" Validation: Axis range checks, command recognition, and status feedback
?? Visual Basic Code Logic (Recap)
You 've already structured this beautifully. Here's a quick summary of its logic:
" cmdMove_Click: Validates coordinates and triggers servo
" ActivateServo: Displays simulated servo activation
" cmdPLCCommand_Click: Executes selected PLC command
?? Logigramme: Robotic Control Workflow
text

basic databse for on line merch store cirriculum

Frame1

entity user : tshingomgombe

user :

user namer

email adresse

password

product

product id

product category

product name

price product

categoriy name

category id

4. Orders

- order_id

user id

ser order id

amount

ok

cancel

next