erow = Sheets("sheet1").Range("a" & Rows.Count).End(xlUp).Row

Range("a" & erow + 1) = TextBox1.Value

Range("b" & erow + 1) = TextBox2.Value

TextBox1.Value = ""

TextBox2.Value = ""

If IsNumeric(Me.TextBox3.Value) = False Then

MsgBox "Incorrect Mobile Number"

Exit Sub

End If

If Len(Me.TextBox3.Value) < 10 Then

MsgBox "Incomplete Mobile Number"

Exit Sub

End If

Dim lr As Integer

lr = Application.CountA(ActiveSheet.Range("A:A"))

ActiveSheet.Unprotect "1234"

ActiveSheet.Range("A" & lr + 1).Value = Me.TextBox3.Value

ActiveSheet.Protect "1234"

Me.TextBox3.Value = ""

' Check if the employee ID to delete is provided

If employeeIDToDelete = "" Then

MsgBox "No Employee ID provided. Deletion aborted.", vbExclamation

End If

Private Sub CommandButton1\_Click()

Dim ws As Worksheet

Dim empID As String

Dim empName As String

Dim contact As String

' Set reference to the worksheet

Set ws = ThisWorkbook.Sheets("Sheet1") ' Change "Sheet1" to your sheet name

' Read data from text boxes

empID = Trim(TextBox1.Text)

empName = Trim(TextBox2.Text)

contact = Trim(TextBox3.Text)

' Check if the employee ID contains only numerals

If Not IsNumeric(empID) Then

MsgBox "Employee ID should contain only numerals.", vbExclamation, "Invalid Input"

Exit Sub

End If

' Check if the employee name contains only alphabets

If Not IsAlpha(empName) Then

MsgBox "Employee Name should contain only alphabets.", vbExclamation, "Invalid Input"

Exit Sub

End If

' Check if the contact contains only 10 numerical numbers

If Len(contact) <> 10 Or Not IsNumeric(contact) Then

MsgBox "Contact should contain exactly 10 numerical numbers.", vbExclamation, "Invalid Input"

Exit Sub

End If

' Insert the data into the worksheet

With ws

' Find the next available row in column A

Dim nextRow As Long

nextRow = .Cells(.Rows.Count, "A").End(xlUp).Row + 1

' Insert data into the next available row

.Cells(nextRow, 1).Value = empID

.Cells(nextRow, 2).Value = empName

.Cells(nextRow, 3).Value = contact

MsgBox "Data inserted successfully.", vbInformation, "Success"

' Clear the text boxes after successful data entry

TextBox1.Text = ""

TextBox2.Text = ""

TextBox3.Text = ""

End With

End Sub

Private Sub CommandButton1\_Click()

Dim ws As Worksheet

Dim lastRow As Long

Dim empID As String

Dim empName As String

Dim secretCode As String

' Set the worksheet

Set ws = ThisWorkbook.Sheets("Sheet1") ' Change "Sheet1" to your sheet name

' Find the last row with data in column A

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

' Loop through each row to generate secret codes

For i = 2 To lastRow ' Assuming data starts from row 2 and headers are in row 1

' Get employee ID and employee name from the worksheet

empID = ws.Cells(i, 1).Value

empName = ws.Cells(i, 2).Value

' Generate secret code

secretCode = GenerateSecretCode(empID, empName)

' Output the secret code to column C

ws.Cells(i, 3).Value = secretCode

Next i

MsgBox "Secret codes generated successfully!", vbInformation

End Sub

Function GenerateSecretCode(empID As String, empName As String) As String

' Extract first two letters of the employee ID

Dim empIDPrefix As String

If Len(empID) >= 2 Then

empIDPrefix = Left(empID, 2)

Else

empIDPrefix = empID

End If

' Extract last two letters of the employee name

Dim empNameSuffix As String

If Len(empName) >= 2 Then

empNameSuffix = Right(empName, 2)

Else

empNameSuffix = empName

End If

' Concatenate the extracted parts to generate the secret code

GenerateSecretCode = empIDPrefix & empNameSuffix

End Function

Private Sub CommandButton1\_Click()

Dim ws As Worksheet

Dim lastRow As Long

Dim empID As String

Dim empName As String

Dim secretCode As String

' Set the worksheet

Set ws = ThisWorkbook.Sheets("Sheet1") ' Change "Sheet1" to your sheet name

' Find the last row with data in column A

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

' Loop through each row to generate secret codes

For i = 2 To lastRow ' Assuming data starts from row 2 and headers are in row 1

' Get employee ID and employee name from the worksheet

empID = ws.Cells(i, 1).Value

empName = ws.Cells(i, 2).Value

' Generate secret code

secretCode = GenerateSecretCode(empID, empName)

' Output the secret code to column C

ws.Cells(i, 3).Value = secretCode

Next i

MsgBox "Secret codes generated successfully!", vbInformation

End Sub

Function GenerateSecretCode(empID As String, empName As String) As String

' Extract first two letters of the employee ID

Dim empIDPrefix As String

If Len(empID) >= 2 Then

empIDPrefix = Left(empID, 2)

Else

empIDPrefix = empID

End If

' Extract last two letters of the employee name

Dim empNameSuffix As String

If Len(empName) >= 2 Then

empNameSuffix = Right(empName, 2)

Else

empNameSuffix = empName

End If

' Concatenate the extracted parts to generate the secret code

GenerateSecretCode = empIDPrefix & empNameSuffix

End Function

E:\BHUVANA\sample.xlsx

Private Sub CommandButton1\_Click()

Dim wb As Workbook

Dim ws As Worksheet

Dim rng As Range

Dim filePath As String

Dim sheetName As String

Dim cell As Range

Dim secretCode As String

Dim nextEmptyCol As Long

Dim empID As String

Dim empName As String

Dim empNumber As String

Dim userInputForm As Object

Dim empRow As Long

Dim deleteRow As Boolean

' Prompt the user to enter the file path

filePath = InputBox("Enter the path to the Excel file:", "File Path")

If filePath = "" Then Exit Sub ' User canceled

' Open the workbook

On Error Resume Next ' Enable error handling

Set wb = Workbooks.Open(filePath)

On Error GoTo 0 ' Disable error handling

If wb Is Nothing Then

MsgBox "Unable to open the file. Please check the file path and try again.", vbExclamation

Exit Sub

End If

' Prompt the user to enter the sheet name

sheetName = InputBox("Enter the name of the worksheet:", "Sheet Name")

If sheetName = "" Then

wb.Close False ' Close the workbook without saving changes

Exit Sub ' User canceled

End If

' Set the worksheet

On Error Resume Next

Set ws = wb.Sheets(sheetName)

On Error GoTo 0

If ws Is Nothing Then

MsgBox "Worksheet '" & sheetName & "' not found.", vbExclamation

wb.Close False ' Close the workbook without saving changes

Exit Sub

End If

' Determine the used range of the worksheet

Set rng = ws.UsedRange

' Find the next empty column

nextEmptyCol = ws.Cells(1, ws.Columns.Count).End(xlToLeft).Column + 1

' Generate the secret code and write it to the next empty column for each row

For Each cell In rng.Rows

secretCode = Left(cell.Cells(1, 1).Value, 2) & Right(cell.Cells(1, 2).Value, 2)

ws.Cells(cell.Row, nextEmptyCol).Value = secretCode

Next cell

' Display the user form to manage data

Set userInputForm = VBA.UserForms.Add("UserInputForm")

' Show existing data if applicable

userInputForm.Controls("txtExistingEmpID").Text = ws.Cells(1, 1).Value ' Assuming existing data starts from row 2

userInputForm.Controls("txtExistingEmpName").Text = ws.Cells(1, 2).Value

userInputForm.Controls("txtExistingEmpNumber").Text = ws.Cells(1, 3).Value

' Show existing data if applicable

userInputForm.txtExistingEmpID.Value = ws.Cells(2, 1).Value ' Assuming existing data starts from row 2

userInputForm.txtExistingEmpName.Value = ws.Cells(2, 2).Value

userInputForm.txtExistingEmpNumber.Value = ws.Cells(2, 3).Value

userInputForm.Show

' Retrieve new employee input from the user form

empID = userInputForm.txtNewEmpID.Value

empName = userInputForm.txtNewEmpName.Value

empNumber = userInputForm.txtNewEmpNumber.Value

' Validate the user input

If Not IsNumeric(empID) Or Not IsNumeric(empNumber) Or Not IsNumeric(empID) Then

MsgBox "Invalid input. Employee ID and Employee Number should contain only numbers. Employee Name should contain only alphabets.", vbExclamation

wb.Close False ' Close the workbook without saving changes

Exit Sub

End If

' Find the first empty row in the worksheet

empRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row + 1

' Write the new employee input to the appropriate columns

ws.Cells(empRow, 1).Value = empID

ws.Cells(empRow, 2).Value = empName

ws.Cells(empRow, 3).Value = empNumber

' Prompt the user if they want to delete a row

deleteRow = MsgBox("Do you want to delete a row? Click Yes to proceed.", vbYesNo) = vbYes

If deleteRow Then

' Prompt the user to enter the row number to delete

empRow = InputBox("Enter the row number you want to delete:", "Delete Row")

If IsNumeric(empRow) And empRow > 0 And empRow <= ws.Cells(ws.Rows.Count, 1).End(xlUp).Row Then

ws.Rows(empRow).Delete

MsgBox "Row " & empRow & " deleted successfully.", vbInformation

Else

MsgBox "Invalid row number or row cannot be deleted.", vbExclamation

End If

End If

' Close the workbook without saving changes

'wb.Close True

' Clean up

Set rng = Nothing

Set ws = Nothing

Set wb = Nothing

Set userInputForm = Nothing

End Sub