



Automated Reminder Letter Generation for Sales Representatives

This VBA code automates the generation of reminder letters to sales representatives, prompting them to take necessary actions regarding outstanding payments from customers. The code extracts relevant data from your Excel workbook, populates predefined Word templates, and creates individual reminder letters for each sales representative. The letters include critical information such as the sales representative's name, state, a table displaying due payment details, and an option to insert a picture for further personalization. This automation streamlines the process of communicating outstanding payment information to your sales team, improving efficiency and financial management.

Video Reference: <https://youtu.be/6oOxlm1zVk8>

How to create a Word Document - Follow the Instruction:-

1. Prepare a Word document.

2. Position the placeholder where you want to display the S.R. (Sales Representative) details.
3. Insert a table with appropriate headings.

You need three Excel Data Sheets:-

Customer Master (Primary Key: Cust_ID)

SR_Master (Primary Key: SR_ID)

Data (Contains Cust_ID and SR_ID to establish connections with the Customer and SR Master sheets)

Create a User Form and add the following elements: a List Box, several Labels, an Image Tool, and two Command Buttons.

Customize the properties such as Name, Caption, Font Format, Color, and Special Effects for these elements from the Properties Window.

Where to store the pictures of Sales Rep?

To display a sales representative's picture in your User Form when clicking on a particular row, you need to store the pictures in a location accessible to your VBA code. Here's a general approach:

Image Storage: Store the sales representatives' pictures in a folder on your computer or a network drive. Make sure the images are named in a way that associates them with the sales representatives, such as using their names or IDs as filenames. For example, you could name the image files after the SR_ID or SR_Name.

Image Path: In your Excel workbook or a separate configuration sheet, you can maintain a mapping of SR_ID or SR_Name to the corresponding image file's path. This mapping will allow your VBA code to dynamically load the correct image when a specific row is clicked.

VBA Code: In your VBA code, you can use the Image control's Picture property to load and display the image. When a row is clicked, your code should read the

corresponding image path from your mapping, and then set the Picture property of the Image control to that path.

' Create the full path to the picture based on the SR name and folder

picturePath = "D:\VBAExcel\" & SRName & ".jpg" ' Adjust the folder path

Folder : D, Dir : VBAExcel, SRName : Gautam Banerjee File type : .jpg

If you have a Sales Representative named "Gautam Banerjee," you can name their picture file "Gautam Banerjee.jpg" or use any other naming convention that makes it easy to associate each picture with the respective Sales Representative.

For example, if you use the SR_Name as the filename, it simplifies the process of retrieving the correct image path when a specific Sales Representative is selected from your User Form. Just ensure that the filenames match the names in your data, so you can easily map between the Sales Representative's name in your data sheet and the corresponding image filename.

Dim wsSR As Worksheet, wsData As Worksheet

Declaring variables outside of procedures (at the top of the code window) has several important advantages and is considered a best practice in VBA programming.

You can declare multiple variables on the same line, separated by commas, as you've done with Dim wsData As Worksheet, wsSR As Worksheet. This is a valid way to declare and initialize multiple variables of the same type in a single line. It can make your code more compact and easier to read when you have multiple variables of the same type to declare.

```
Private Sub FrmExit_Click()
```

```
    Unload Me
```

```
End Sub
```

UserForm's "Exit" button click event. When this button is clicked, it unloads (closes) the UserForm. This is a common practice when you want to close a UserForm or exit a specific part of your VBA program.

Here's what this code does:

Private Sub FrmExit_Click(): This line signifies the beginning of a private subroutine (Sub) named FrmExit_Click. It's associated with a UserForm's "Click" event for an object named "FrmExit." The Private keyword means that this Sub is only accessible within the module where it's defined.

Unload Me: This line is the action performed when the "Exit" button (or object named "FrmExit") is clicked. Unload Me unloads (closes) the current UserForm (Me refers to the current instance of the UserForm), effectively closing it and returning control to the calling code or the Excel interface.

So, when you click the "Exit" button on the UserForm, this code is executed, and the UserForm is closed. This is a common way to provide an exit or close functionality in a UserForm.

```
Private Sub ListBoxSRPopulate()
```

```
    Dim lastSRRow As Long
```

```
    Dim i As Long
```

```
    ' Set a reference to the "SR_Master" worksheet
```

```
    Set wsSR = ThisWorkbook.Sheets("SR_Master")
```

```
    ' Find the last used row in the "SR_Master" sheet
```

```
lastSRRow = wsSR.Cells(wsSR.Rows.Count, "A").End(xlUp).row ' Assuming  
data starts from column B
```

```
' Loop through the "SR_Master" sheet and populate the ProductList ListBox
```

```
For i = 2 To lastSRRow ' Assuming the data starts from row 2
```

```
    SRNameList.AddItem wsSR.Cells(i, 2).value ' SR_Name (second column)
```

```
Next i
```

```
' Initialize variables to store select'ed values
```

```
SRID = ""
```

```
SRName = ""
```

```
End Sub
```

The code is a VBA subroutine that populates a ListBox named SRNameList with values from the "SR_Master" worksheet in Excel. It also initializes two variables, SRID and SRName, to store the selected values from the ListBox. Here's a breakdown of what the code does:

Private Sub ListBoxSRPopulate(): This line defines a private subroutine named ListBoxSRPopulate. It's typically associated with an event, such as the initialization of a UserForm.

Dim lastSRRow As Long and Dim i As Long: These lines declare two variables, lastSRRow and i, both with a Long data type. lastSRRow will be used to store the last used row in the "SR_Master" worksheet, and i will be used as a loop counter.

Set wsSR = ThisWorkbook.Sheets("SR_Master"): This line sets a reference to the "SR_Master" worksheet within the current workbook (ThisWorkbook).

lastSRRow = wsSR.Cells(wsSR.Rows.Count, "A").End(xlUp).Row: This line finds the last used row in column A of the "SR_Master" worksheet by starting from the bottom and moving upwards until it encounters the last non-empty cell. The result is stored in the lastSRRow variable.

For i = 2 To lastSRRow: This line starts a For loop that iterates from 2 (assuming data starts from row 2) to the lastSRRow.

SRNameList.AddItem wsSR.Cells(i, 2).Value: Inside the loop, this line adds items to the SRNameList ListBox. It retrieves the value from the "SR_Master" worksheet in the current row (specified by i) and the second column (column B) and adds it to the ListBox.

SRID = "" and SRName = "": These lines initialize two variables, SRID and SRName, with empty strings. These variables are used to store the selected values from the ListBox.

Overall, this code is designed to populate a ListBox (SRNameList) with values from the "SR_Master" worksheet and prepare two variables (SRID and SRName) to store the selected values. It's often used in the initialization of a UserForm or in response to a specific event that requires loading data into the ListBox

```
Private Sub UserForm_Initialize()
```

```
    ListBoxSRPopulate
```

```
End Sub
```

The code is a part of a VBA UserForm and is associated with the UserForm's "Initialize" event. Here's what it does:

Private Sub UserForm_Initialize(): This line defines a private subroutine named UserForm_Initialize. This subroutine is automatically executed when the UserForm is initialized or loaded.

ListBoxSRPopulate: This line calls another subroutine named ListBoxSRPopulate. This is done to populate a ListBox control with data from the "SR_Master" worksheet when the UserForm is initialized.

In summary, when the UserForm is initialized, the ListBoxSRPopulate subroutine is automatically called, which in turn populates a ListBox control with data from the "SR_Master" worksheet. This is a common practice to ensure that the ListBox is filled with data when the UserForm is displayed to the user.

```

Private Sub SRNameList_Click()
    'On Error Resume Next
    Dim selectedSRRow As Long
    'Set wsSR = ThisWorkbook.Sheets("SR_Master")
    selectedSRRow = SRNameList.ListIndex + 1

    SRID = wsSR.Cells(selectedSRRow + 1, 1).value
    SRName = wsSR.Cells(selectedSRRow + 1, 2).value

    SRIDLbl.Caption = SRID
    SRNameLbl.Caption = SRName

    ' Create the full path to the picture based on the SR name and folder
    picturePath = "D:\VBAExcel\" & SRName & ".jpg" ' Adjust the folder path
    If Dir(picturePath) <> "" Then ' Check if the picture file exists
        SRImage.Picture = LoadPicture(picturePath)
    Else
        SRImage.Picture = LoadPicture("") ' Clear the image if no picture found
    End If

    SRSalesPopulate

    If Err.Number <> 0 Then
        MsgBox "Error: " & Err.Description
    End If

    On Error GoTo 0

```

End Sub

The code is a VBA subroutine associated with the Click event of a ListBox named SRNameList on your UserForm. This code is executed when an item in the ListBox is clicked. Here's a breakdown of what it does:

Private Sub SRNameList_Click(): This line defines a private subroutine named SRNameList_Click, which is associated with the Click event of the SRNameList ListBox.

Dim selectedSRRow As Long: This line declares a variable selectedSRRow as a Long. This variable will be used to store the selected row index in the SRNameList ListBox.

selectedSRRow = SRNameList.ListIndex + 1: This line calculates the selected row index by adding 1 to the ListIndex property of the SRNameList ListBox. The ListIndex property indicates which item is selected in the ListBox, and it starts from 0.

SRID = wsSR.Cells(selectedSRRow + 1, 1).Value and SRName = wsSR.Cells(selectedSRRow + 1, 2).Value: These lines retrieve the values of the Sales Representative ID (SRID) and Sales Representative Name (SRName) from the "SR_Master" worksheet based on the selected row in the SRNameList ListBox.

SRIDLbl.Caption = SRID and SRNameLbl.Caption = SRName: These lines update the captions of two labels (SRIDLbl and SRNameLbl) on your UserForm with the selected Sales Representative ID and Name.

picturePath = "D:\VBAExcel\" & SRName & ".jpg": This line constructs the file path to the Sales Representative's picture based on their name and a specified folder path. It assumes that pictures are stored as JPG files with filenames matching the Sales Representative names.

If Dir(picturePath) <> "" Then: This line checks if the picture file exists in the specified folder.

SRImage.Picture = LoadPicture(picturePath): If the picture file exists, it loads and displays the picture in an Image control named SRImage on your UserForm.

Else: If no picture file is found, it clears the image control by setting it to a blank picture.

SRSalesPopulate: This line calls a subroutine named SRSalesPopulate, which presumably populates some data related to the selected Sales Representative.

If Err.Number <> 0 Then: This line checks if any errors occurred during the execution of the code.

MsgBox "Error: " & Err.Description: If an error occurred, it displays an error message with a description of the error.

On Error GoTo 0: This line resets the error handling to the default state, which means that any subsequent errors will not be suppressed.

In summary, this code handles the event when an item in the SRNameList ListBox is clicked. It updates labels, loads a picture based on the selected Sales Representative's name, and calls a subroutine to populate additional data related to the selected Sales Representative. It also includes error handling to handle any potential errors that may occur during execution.

Download the Below Code

Watch the video, open an Excel workbook, create a form, insert the required tools, name it, change the captions, format, etc. Double-click anywhere on the form to open the code window and simply paste the code. That's it. Run the program.



Gautam Banerjee

“Helping beginners learn something new is a great way to share your knowledge and make a positive impact”.

Email: gincom1@yahoo.com

If you have any queries, please visit our "Contact Us" page.

Thank You. See you again!



Gautam Banerjee

Age: 63

Pay by UPI

9748327614

To save an Excel workbook with VBA code enabled, you need to follow these steps:

Open your Excel workbook that contains the VBA code.

Press Alt + F11 to open the Visual Basic for Applications (VBA) editor.

In the VBA editor, make sure your code is saved and ready to be included in the workbook. Ensure that all your macros and code are working correctly.

Close the VBA editor by clicking the "X" button in the top-right corner or pressing Ctrl + S to save your code and then closing the VBA editor window.

Go back to your Excel workbook.

Click on "File" in the Excel ribbon to open the File menu.

Choose "Save As" from the menu.

In the Save As dialog box, select the location where you want to save your workbook.

In the "Save as type" dropdown menu, choose "**Excel Macro-Enabled Workbook (*.xlsm)**".

Provide a name for your workbook, and then click the "Save" button.

Your Excel workbook will be saved as a macro-enabled workbook (.xlsm) with the VBA code included. You can reopen it at any time, and the VBA code will be available for execution.

Please make sure to keep backups of your workbook, especially if you plan to share it with others, to prevent accidental loss of your VBA code.

Name of Excel Sheets : Customer_Master, SR_Master, Data

List Box Name : SRNameList

Image Frame Name : SRImage

Command Buttons Name : FrmExit (For Close Form)

: SrReminder (For Letter Printing)

Labels Name : SRIDLbl (For SR_ID),

: SRNameLbl (For Name)

: SalesValueLbl (For Total Sales Value)

: Label1 (For Paid Amount)

: Label2 (For Due Amount)

Now Start coding:

```
Dim wsSR As Worksheet
```

```
Dim wsData As Worksheet
```

```
' This subroutine populates the ListBox with SR names
```

```
Private Sub ListBoxSRPopulate()
```

```
    Dim lastSRRow As Long
```

```
    Dim i As Long
```

```
    ' Set a reference to the "SR_Master" worksheet
```

```
    Set wsSR = ThisWorkbook.Sheets("SR_Master")
```

```
    ' Find the last used row in the "SR_Master" sheet
```

```
    lastSRRow = wsSR.Cells(wsSR.Rows.Count, "A").End(xlUp).Row ' Assuming  
data starts from column B
```

```
    ' Loop through the "SR_Master" sheet and populate the ProductList ListBox
```

```
    For i = 2 To lastSRRow ' Assuming the data starts from row 2
```

```
        SRNameList.AddItem wsSR.Cells(i, 2).Value ' SR_Name (second column)
```

```
    Next i
```

```
    ' Initialize variables to store selected values
```

```
    SRID = ""
```

```
    SRName = ""
```

```
End Sub
```

```
' This subroutine handles the Click event of the Exit button
```

```
Private Sub FrmExit_Click()
```

```
    Unload Me
```

```
End Sub
```

```
' This subroutine handles the Click event of the SRNameList ListBox
```

```
Private Sub SRNameList_Click()
```

```
    Dim selectedSRRow As Long
```

```
    selectedSRRow = SRNameList.ListIndex + 1
```

```
' Get SRID and SRName based on the selected row
```

```
SRID = wsSR.Cells(selectedSRRow + 1, 1).Value
```

```
SRName = wsSR.Cells(selectedSRRow + 1, 2).Value
```

```
' Update labels with SR information
```

```
SRIDLbl.Caption = SRID
```

```
SRNameLbl.Caption = SRName
```

```
' Create the full path to the picture based on the SR name and folder
```

```
picturePath = "D:\VBAExcel\" & SRName & ".jpg" ' Adjust the folder path
```

```
If Dir(picturePath) <> "" Then ' Check if the picture file exists
```

```
    SRImage.Picture = LoadPicture(picturePath)
```

```
Else
```

```
    SRImage.Picture = LoadPicture("") ' Clear the image if no picture found
```

```
End If
```

```

' Populate SR sales information
    SRSalesPopulate
End Sub

' This subroutine calculates SR sales information
Private Sub SRSalesPopulate()
    Dim i As Long

    Set wsData = ThisWorkbook.Sheets("Data")

    TotalsrSales = 0
    totalpaid = 0
    totaldues = 0

    ' Loop through the "Data" sheet and calculate total sales for the selected SRID
    For i = 2 To wsData.Cells(wsData.Rows.Count, "A").End(xlUp).Row
        If wsData.Cells(i, "J").Value = SRIDLbl.Caption Then
            If wsData.Cells(i, "I").Value = "Y" Then
                totalpaid = totalpaid + wsData.Cells(i, "E").Value
            Else
                totaldues = totaldues + wsData.Cells(i, "E").Value
            End If
        End If
    Next i

```

```

TotalsrSales = totalpaid + totaldues
SalesValueLbl.Caption = "Rs." & Format(TotalsrSales, "#0.00")
Label1.Caption = "Rs." & Format(totalpaid, "#0.00")
Label2.Caption = "Rs." & Format(totaldues, "#0.00")
End Sub

' This subroutine handles the Click event of the SrReminder button
Private Sub SrReminder_Click()
    ' Call the GenerateReminderLetters subroutine to generate reminder letters
    GenerateReminderLetters
End Sub

' This subroutine generates reminder letters for SRs with outstanding dues
Sub GenerateReminderLetters()
    Dim WordApp As Object
    Dim WordDoc As Object
    Dim wsData As Worksheet
    Dim wsSR As Worksheet
    Dim wsCust As Worksheet
    Dim SRRow As Long
    Dim SR_ID As String
    Dim SR_Name As String
    Dim SR_State As String
    Dim ReminderPath As String
    Dim ReminderFileName As String

```

```
Dim WordTable As Object
Dim WordRange As Object
Dim DocText As String
Dim TotalOutstanding As Double
```

```
' Set references to worksheets
```

```
Set wsData = ThisWorkbook.Sheets("Data")
```

```
Set wsSR = ThisWorkbook.Sheets("SR_Master")
```

```
Set wsCust = ThisWorkbook.Sheets("Customer_Master")
```

```
' Define the path for the reminder letters
```

```
ReminderPath = ThisWorkbook.Path & "\\Reminder_Letters\"
```

```
' Create the folder if it doesn't exist
```

```
If Len(Dir(ReminderPath, vbDirectory)) = 0 Then
```

```
    Mkdir ReminderPath
```

```
End If
```

```
' Set up Word
```

```
On Error Resume Next
```

```
Set WordApp = CreateObject("Word.Application")
```

```
On Error GoTo 0
```

```
If WordApp Is Nothing Then
```

```
    MsgBox "Microsoft Word is not installed on this computer. Please install Word  
to generate reminder letters."
```

Exit Sub

End If

WordApp.Visible = True ' Optional: Make Word application visible

' Loop through SR IDs

For SRRow = 2 To wsSR.Cells(wsSR.Rows.Count, "A").End(xlUp).Row

SR_ID = wsSR.Cells(SRRow, "A").Value

SR_Name = wsSR.Cells(SRRow, "B").Value

SR_State = wsSR.Cells(SRRow, "C").Value

SR_Image = "D:\VBAExcel\" & SR_Name & ".jpg"

TotalOutstanding = 0

' Create a new Word document based on your template

Set WordDoc = WordApp.Documents.Add(Template:=ThisWorkbook.Path &
"Reminder_Letters\SRDueLetterwithTable.docx")

' Get the Word table and range

Set WordTable = WordDoc.Tables(1)

Set WordRange = WordDoc.Range

' Replace placeholders in the Word document

With WordRange.Find

.Text = "#SRID#"

.Replacement.Text = SR_ID

.Execute Replace:=wdReplaceAll

End With

With WordRange.Find

.Text = "#SRName#"

.Replacement.Text = SR_Name

.Execute Replace:=wdReplaceAll

End With

With WordRange.Find

.Text = "#SRState#"

.Replacement.Text = SR_State

.Execute Replace:=wdReplaceAll

End With

' Loop through data and populate the Word table with dues details

On Error Resume Next

For Each Cell In wsData.Range("A2:I" & wsData.Cells(wsData.Rows.Count, "A").End(xlUp).Row)

If Cell.Offset(0, 8).Value = SR_ID And Cell.Offset(0, 7).Value = "N" Then

WordTable.Rows.Add

WordTable.Cell(WordTable.Rows.Count, 1).Range.Text =
wsData.Cells(Cell.Row, "B").Value ' Customer Name

WordTable.Cell(WordTable.Rows.Count, 2).Range.Text =
wsData.Cells(Cell.Row, "C").Value ' Invoice No

WordTable.Cell(WordTable.Rows.Count, 3).Range.Text =
wsData.Cells(Cell.Row, "D").Value ' Invoice Date

```
WordTable.Cell(WordTable.Rows.Count, 4).Range.Text =  
Format(wsData.Cells(Cell.Row, "E").Value, "#0.00") ' Amount Dues
```

```
TotalOutstanding = TotalOutstanding + wsData.Cells(Cell.Row,  
"D").Value
```

```
End If
```

```
Next Cell
```

```
On Error GoTo 0
```

```
' Insert the total outstanding amount after the table
```

```
WordTable.Rows.Add ' Add an empty row after the table
```

```
WordTable.Cell(WordTable.Rows.Count, 3).Range.Text = "Total Dues: Rs. "
```

```
WordTable.Cell(WordTable.Rows.Count, 4).Range.Text =  
Format(TotalOutstanding, "#0.00")
```

```
' Save the document with a unique name
```

```
ReminderFileName = ReminderPath & "SRReminder_" & SR_ID & ".docx"
```

```
WordDoc.SaveAs2 Filename:=ReminderFileName
```

```
WordDoc.Close
```

```
Next SRRow
```

```
' Release Word objects
```

```
WordApp.Quit
```

```
Set WordDoc = Nothing
```

```
Set WordApp = Nothing
```

```
MsgBox "Reminder letters have been generated and saved in the  
'Reminder_Letters' folder."
```

```
End Sub
```

```
' This subroutine is executed when the UserForm initializes
```

```
Private Sub UserForm_Initialize()
```

```
    ' Populate the SRNameList ListBox when the form initializes
```

```
    ListBoxSRPopulate
```

```
End Sub
```

Change your Excel Sheet Name, Column name, List Box Name, Command buttons name, Variables name, File path, Word File Name etc.



Gautam Banerjee

“Helping beginners learn something new is a great way to share your knowledge and make a positive impact”.

Email: gincom1@yahoo.com

If you have any queries,
please visit our "Contact Us" page.

Thank You. See you again!



Gautam Banerjee

Age: 63

Pay by UPI

9748327614

Don't waste time; start today. I am here to help you. If you have any questions, feel free to ask me. To do so, click on 'Contact Me' at the bottom of my web page and submit your question.

Please note that this website is free and provided by Wix.com. If you want to learn how to develop your own website without writing a single line of code, visit my channel and watch videos titled 'How to Create a Free Website'