Required: 4 Excel Sheets, Insert 4 List boxes, Labels, Text Boxes, Command Buttons etc.

Presented by Gautam Banerjee

E-mail : gincom1@yahoo.com

Create a List Box : Name it CustNameList

```
Private Sub CustNameListDisplay()
Dim wsCust As Worksheet
Set wsCust = ThisWorkbook.Worksheets("Customer_Master")
```

Copy the Code Below :

```
Dim LblCustId As String
Dim LblProdId As String
Dim LblSRId As String


Private Sub DataListDisplay()

    Dim ws As Worksheet
    Set ws = ThisWorkbook.Worksheets("Data")

    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).row

    Dim rng As Range
    Set rng = ws.Range("A1:I" & lastRow) ' Assuming data starts from A2

    DataListBox.Clear
    DataListBox.List = rng.value
```

```
    TotRecord.Caption = "Total Records: " & lastRow - 1

    Dim TotalY, TotalN, TotalYN
    Dim i As Integer
    Dim YesNo As String
    YesNo = "Y"

    For i = 2 To lastRow
        If ws.Cells(i, 8).value = YesNo Then
         TotalY = TotalY + ws.Cells(i, "D").value
        Else
        TotalN = TotalN + ws.Cells(i, "D").value
         End If
        TotalYN = TotalYN + ws.Cells(i, "D").value
        Next i

    AmtRecd.Caption = "Rs. " & Format(TotalY, "###,##0.00")
    AmtDues.Caption = "Rs. " & Format(TotalN, "###,##0.00")
    TotAmt.Caption = "Rs. " & Format(TotalYN, "###,##0.00")

End Sub



Private Sub CustNameListDisplay()
```

```
    Dim wsCust As Worksheet
    Set wsCust = ThisWorkbook.Worksheets("Customer_Master")

    Dim lastRow As Long
    lastRow = wsCust.Cells(wsCust.Rows.Count, "A").End(xlUp).row

    Dim rng As Range
    Set rng = wsCust.Range("A2:B" & lastRow) ' Assuming data starts
from A2

    CustNameList.Clear
    CustNameList.List = rng.value

End Sub

Private Sub ProductNameListDisplay()

    Dim ws As Worksheet
    Set ws = ThisWorkbook.Worksheets("Product_Master")

    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).row

    Dim rng As Range
```

```
    Set rng = ws.Range("A2:B" & lastRow) ' Assuming data starts from
A2


    ProductNameList.Clear
    ProductNameList.List = rng.value


End Sub


Private Sub SRNameListDisplay()


    Dim ws As Worksheet
    Set ws = ThisWorkbook.Worksheets("SR_Master")


    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).row


    Dim rng As Range
    Set rng = ws.Range("A2:B" & lastRow) ' Assuming data starts from
A2


    SRNameList.Clear
    SRNameList.List = rng.value


End Sub
```

```
Private Sub CustNameList_Click()

    Dim wsCust As Worksheet
    Set wsCust = ThisWorkbook.Worksheets("Data")

    Dim lastRow As Long
    lastRow = wsCust.Cells(wsCust.Rows.Count, "A").End(xlUp).row

    Dim rng As Range
    Set rng = wsCust.Range("A2:A" & lastRow)

    Dim DataRange As Long
    DataRange = lastRow

    LblCustId = CustNameList.List(CustNameList.ListIndex, 0)

    Dim TotalCustY, TotalCustN, TotalCustYN
    Dim i As Integer
    Dim YesNo As String
    YesNo = "Y"

    For i = 1 To lastRow
       If wsCust.Cells(i, 1).value = LblCustId Then
          If wsCust.Cells(i, 8).value = "Y" Then
             TotalCustY = TotalCustY + wsCust.Cells(i, "D").value
```

```
            Else
                TotalCustN = TotalCustN + wsCust.Cells(i, "D").value
            End If
         End If
        TotalCustYN = TotalCustY + TotalCustN
        Next i

    LblCName.Caption = CustNameList.List(CustNameList.ListIndex, 1)

    CustPayRecd.Caption = "Rs. " & Format(TotalCustY, "###,##0.00")
    CustPayDue.Caption = "Rs. " & Format(TotalCustN, "###,##0.00")
    Label21.Caption = "Rs. " & Format(TotalCustYN, "###,##0.00")

    GB.BackColor = vbRed
    GB.ForeColor = vbYellow
End Sub

Private Sub ProductNameList_Click()

    Dim wsCust As Worksheet
    Set wsCust = ThisWorkbook.Worksheets("Data")

    Dim lastCustRow As Long
    lastCustRow = wsCust.Cells(wsCust.Rows.Count,
"A").End(xlUp).row
```

```
    Dim rngCust As Range
    Set rngCust = wsCust.Range("A2:A" & lastCustRow)

    Dim wsProd As Worksheet
    Set wsProd = ThisWorkbook.Worksheets("Product_Master")

    Dim lastProdRow As Long
    lastProdRow = wsProd.Cells(wsProd.Rows.Count,
"A").End(xlUp).row

    Dim rngProd As Range
    Set rngProd = wsProd.Range("A2:A" & lastProdRow)

    Dim DataRange As Long
    DataRange = lastProdRow

    LblProdId = ProductNameList.List(ProductNameList.ListIndex, 0)

    Dim TotalProdY, TotalProdN, TotalProdYN

    For i = 1 To lastCustRow
       If wsCust.Cells(i, 6).value = LblProdId Then
          'If wsCust.Cells(i, 8).value = "Y" Then
             'TotalProdY = TotalProdY + wsCust.Cells(i, "D").value
          'Else
             'TotalProdN = TotalProdN + wsCust.Cells(i, "D").value
```

```
            'End If
            TotalProdY = TotalProdY + wsCust.Cells(i, "D").value
         End If
        'TotalProdYN = TotalProdYN + TotalProdY
        Next i

    LblPName.Caption =
ProductNameList.List(ProductNameList.ListIndex, 1)

    'Label28.Caption = "Rs. " & Format(TotalY, "###,##0.00")
    'Label20.Caption = "Rs. " & Format(TotalN, "###,##0.00")
    Label28.Caption = "Rs. " & Format(TotalProdY, "###,##0.00")

    GB.BackColor = vbBlue
    GB.ForeColor = vbWhite

End Sub

Private Sub SRNameList_Click()
    Dim ws3Cust As Worksheet
    Set ws3Cust = ThisWorkbook.Worksheets("Data")

    Dim lastCustRow As Long
    lastCustRow = ws3Cust.Cells(ws3Cust.Rows.Count,
"A").End(xlUp).row
```

```
Dim rngCust As Range
Set rngCust = ws3Cust.Range("A2:A" & lastCustRow)

Dim wsSR As Worksheet
Set wsSR = ThisWorkbook.Worksheets("Product_Master")

Dim lastSRRow As Long
lastSRRow = wsSR.Cells(wsSR.Rows.Count, "A").End(xlUp).row

Dim rngSR As Range
Set rngSR = wsSR.Range("A2:A" & lastSRRow)

Dim DataRange As Long
DataRange = lastSRRow

LblSRId = SRNameList.List(SRNameList.ListIndex, 0)

Dim TotalsrY, TotalsrN, TotalsrYN

For i = 1 To lastCustRow

   If ws3Cust.Cells(i, 9).value = LblSRId Then
      'If wsCust.Cells(i, 8).value = "Y" Then
         'TotalProdY = TotalProdY + wsCust.Cells(i, "D").value
      'Else
```

```
            'TotalProdN = TotalProdN + wsCust.Cells(i, "D").value
          'End If
          TotalsrY = TotalsrY + ws3Cust.Cells(i, "D").value
        End If

      Next i

    LblSRName.Caption = SRNameList.List(SRNameList.ListIndex, 1)

    'Label28.Caption = "Rs. " & Format(TotalY, "###,##0.00")
    'Label20.Caption = "Rs. " & Format(TotalN, "###,##0.00")
    Label32.Caption = "Rs. " & Format(TotalsrY, "###,##0.00")

    GB.BackColor = vbGreen
    GB.ForeColor = vbBlue

End Sub


Private Sub UserForm_Initialize()

    DataListDisplay
    CustNameListDisplay
    ProductNameListDisplay
    SRNameListDisplay
```

```
    lblDate.Caption = "Today is " & Format(Date, "dddd") & ", Date: " & Format(Date, "dd/mm/yyyy")

End Sub
```

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!

PLEASE DONATE

Gautam Banerjee

Age: 63

Pay by UPI

9748327614

READ THE BELOW DESCRIPTION TO UNDERSTAND THE ABOVE CODE LINE BY LINE:

This code segment declares a variable wsCust to hold a reference to a specific worksheet named "Customer_Master" within the workbook containing the VBA code.

Dim lastRow As Long lastRow = wsCust.Cells(wsCust.Rows.Count, "A").End(xlUp).Row

Here, a variable lastRow is declared to store the last row number with data in column A of the "Customer_Master" worksheet. It uses the xlUp method to find the last non-empty cell in column A.

Defining a Range:

Dim rng As Range Set rng = wsCust.Range("A2:B" & lastRow)

The code segment creates a Range object named rng that spans from cell A2 to the last row in column B, capturing the data in columns A and B (assuming a 2-column dataset).

Populating a List Box:

CustNameList.Clear

CustNameList.List = rng.Value

End Sub

The code clears the contents of a ListBox control named CustNameList and then assigns the values from the rng range to the List property of the ListBox. This is effectively populating the ListBox with the data found in the specified range.

In summary, the code retrieves data from the "Customer_Master" worksheet (columns A and B) and populates a ListBox control named CustNameList with this data. The ListBox is cleared first to ensure that the data is not appended to any existing entries.

Create a List Box : Name it ProductNameList

```
Private Sub ProductNameListDisplay()

    Dim ws As Worksheet
    Set ws = ThisWorkbook.Worksheets("Product_Master")

    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).row

    Dim rng As Range
    Set rng = ws.Range("A2:B" & lastRow)
    ProductNameList.Clear
    ProductNameList.List = rng.value
```

End Sub

Subroutine Declaration:

Private Sub ProductNameListDisplay()

This line declares the start of a private subroutine named "ProductNameListDisplay." A private subroutine means it can only be accessed within the module where it is defined.

Worksheet Declaration:

Dim ws As Worksheet Set ws = ThisWorkbook.Worksheets("Product_Master")

This segment declares a variable ws to hold a reference to the "Product_Master" worksheet within the workbook containing the VBA code.

Finding the Last Row:

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

The code calculates the last row with data in column A of the "Product_Master" worksheet using the xlUp method. It then stores this value in the variable lastRow.

Defining a Range:

Dim rng As Range Set rng = ws.Range("A2:B" & lastRow)

This part creates a Range object named rng that covers data in columns A and B of the "Product_Master" worksheet, from cell A2 to the last row determined earlier.

Populating a ListBox:

ProductNameList.Clear ProductNameList.List = rng.Value

This section clears the contents of the ListBox named "ProductNameList" and then assigns the values from the rng range to the List property of the ListBox. This effectively populates the ListBox with the data found in the specified range.

Subroutine End:

vbaCopy code

End Sub

This line marks the end of the subroutine.

In summary, this VBA subroutine reads data from columns A and B of the "Product_Master" worksheet, starting from A2 and going down to the last non-empty row in column A. It then clears the contents of the ListBox named "ProductNameList" and fills it with the data from the specified range.

Create a List Box : Name it SRNameList

```
Private Sub SRNameListDisplay()

    Dim ws As Worksheet
    Set ws = ThisWorkbook.Worksheets("SR_Master")

    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).row

    Dim rng As Range
```

```
    Set rng = ws.Range("A2:B" & lastRow) ' Assuming data starts from
A2


    SRNameList.Clear
    SRNameList.List = rng.value


End Sub
```

Create a List Box : Name it CustNameList

The below VBA code seems to be part of a Click event handler for a ListBox control named "CustNameList." This code appears to be designed to perform calculations and update various labels and components based on the selected item in the list. Let's break down the code step by step:

Worksheet Declaration:

Dim wsCust As Worksheet Set wsCust = ThisWorkbook.Worksheets("Data")

This code segment declares a worksheet variable wsCust that refers to the "Data" worksheet within the workbook containing the VBA code.

Finding the Last Row:

Dim lastRow As Long lastRow = wsCust.Cells(wsCust.Rows.Count, "A").End(xlUp).Row

The code calculates the last row with data in column A of the "Data" worksheet using the xlUp method. It stores this value in the variable lastRow.

Defining Ranges:

Dim rng As Range Set rng = wsCust.Range("A2:A" & lastRow)

This part creates a Range object named rng that covers data in column A of the "Data" worksheet, from cell A2 to the last row determined earlier.

Looping through Data:

For i = 1 To lastRow

This loop iterates through each row of data in the specified range (rng) and performs calculations based on the conditions specified inside the loop.

Conditions and Calculations:

```
If wsCust.Cells(i, 1).Value = LblCustId Then
    If wsCust.Cells(i, 8).Value = "Y" Then
        TotalCustY = TotalCustY + wsCust.Cells(i, "D").Value
Else TotalCustN = TotalCustN + wsCust.Cells(i, "D").Value
    End If
End If
```

Within the loop, this section checks if the value in column A (presumably a customer ID) matches the value stored in LblCustId. Depending on the value in column H (presumably a "Y" or "N" for payment received), it updates the variables TotalCustY or TotalCustN with the corresponding payment amounts.

Updating Labels and Components:

LblCName.Caption = CustNameList.List(CustNameList.ListIndex, 1)
CustPayRecd.Caption = "Rs. " & Format(TotalCustY, "###,##0.00")
CustPayDue.Caption = "Rs. " & Format(TotalCustN, "###,##0.00")
Label21.Caption = "Rs. " & Format(TotalCustYN, "###,##0.00")

These lines update various label captions on the user interface with calculated values.

Changing Component Colors:

GB.BackColor = vbRed

GB.ForeColor = vbYellow

This code changes the background color and text color of a component (possibly a GroupBox or other container) to red and yellow, respectively.

Please make sure that the variables, labels, and components mentioned in the code correspond correctly to those present in your user interface. Also, ensure that the logic of calculating payment amounts and updating labels aligns with your requirements and data structure.

ListBox control named "ProductNameList

```
Private Sub ProductNameList_Click()

    Dim wsCust As Worksheet
    Set wsCust = ThisWorkbook.Worksheets("Data")

    Dim lastCustRow As Long
    lastCustRow = wsCust.Cells(wsCust.Rows.Count,
"A").End(xlUp).row

    Dim rngCust As Range
    Set rngCust = wsCust.Range("A2:A" & lastCustRow)

    Dim wsProd As Worksheet
    Set wsProd = ThisWorkbook.Worksheets("Product_Master")

    Dim lastProdRow As Long
    lastProdRow = wsProd.Cells(wsProd.Rows.Count,
"A").End(xlUp).row

    Dim rngProd As Range
    Set rngProd = wsProd.Range("A2:A" & lastProdRow)

    Dim DataRange As Long
    DataRange = lastProdRow
```

```
    LblProdId = ProductNameList.List(ProductNameList.ListIndex, 0)

    Dim TotalProdY, TotalProdN, TotalProdYN

    For i = 1 To lastCustRow
        If wsCust.Cells(i, 6).value = LblProdId Then
            'If wsCust.Cells(i, 8).value = "Y" Then
                'TotalProdY = TotalProdY + wsCust.Cells(i, "D").value
            'Else
                'TotalProdN = TotalProdN + wsCust.Cells(i, "D").value
            'End If
            TotalProdY = TotalProdY + wsCust.Cells(i, "D").value
         End If
        'TotalProdYN = TotalProdYN + TotalProdY
        Next i

    LblPName.Caption =
ProductNameList.List(ProductNameList.ListIndex, 1)

    'Label28.Caption = "Rs. " & Format(TotalY, "###,##0.00")
    'Label20.Caption = "Rs. " & Format(TotalN, "###,##0.00")
    Label28.Caption = "Rs. " & Format(TotalProdY, "###,##0.00")

    GB.BackColor = vbBlue
    GB.ForeColor = vbWhite
```

End Sub

The above VBA code seems to be part of a Click event handler for a ListBox control named "ProductNameList." This code appears to be designed to perform calculations and update various labels and components based on the selected item in the list. However, there are a few lines that are commented out (surrounded by single quotes), which might have been intended for more advanced functionality. Here's a breakdown of the code:

Worksheet Declarations: Two worksheets are declared: wsCust for the "Data" worksheet and wsProd for the "Product_Master" worksheet.

Finding Last Rows:

lastCustRow is calculated as the last non-empty row in column A of the "Data" worksheet.

lastProdRow is calculated as the last non-empty row in column A of the "Product_Master" worksheet.

Defining Ranges:

rngCust covers data in column A of the "Data" worksheet, from cell A2 to the last row determined earlier.

rngProd covers data in column A of the "Product_Master" worksheet, from cell A2 to the last row determined earlier.

Looping through Data: The loop iterates through each row of data in rngCust.

Conditions and Calculations:

This section checks if the value in column F (presumably a product ID) matches the value stored in LblProdId.

The payment values (column D) are added to TotalProdY for each matching product.

Updating Labels and Components:

LblPName is updated with the product name based on the selected item.

Label28 is updated with the formatted TotalProdY value.

Changing Component Colors: This code changes the background color and text color of a component (possibly a GroupBox or other container) to blue and white, respectively.

The code is calculating the total payment for a specific product based on its ID. The commented-out lines suggest that you might have had plans to calculate totals for different payment statuses ("Y" and "N") as well. If you decide to incorporate those functionalities, you can uncomment and modify those lines accordingly.

Please ensure that the variables, labels, and components mentioned in the code correspond correctly to those present in your user interface. Additionally, ensure that the logic of calculating payment amounts and updating labels aligns with your requirements and data structure.

```
Private Sub CustNameList_Click()

    Dim wsCust As Worksheet
    Set wsCust = ThisWorkbook.Worksheets("Data")

    Dim lastRow As Long
    lastRow = wsCust.Cells(wsCust.Rows.Count, "A").End(xlUp).row

    Dim rng As Range
    Set rng = wsCust.Range("A2:A" & lastRow)

    Dim DataRange As Long
```

```
DataRange = lastRow

LblCustId = CustNameList.List(CustNameList.ListIndex, 0)

Dim TotalCustY, TotalCustN, TotalCustYN
Dim i As Integer
Dim YesNo As String
YesNo = "Y"

For i = 1 To lastRow
   If wsCust.Cells(i, 1).value = LblCustId Then
      If wsCust.Cells(i, 8).value = "Y" Then
         TotalCustY = TotalCustY + wsCust.Cells(i, "D").value
      Else
         TotalCustN = TotalCustN + wsCust.Cells(i, "D").value
      End If
    End If
   TotalCustYN = TotalCustY + TotalCustN
   Next i

LblCName.Caption = CustNameList.List(CustNameList.ListIndex, 1)

CustPayRecd.Caption = "Rs. " & Format(TotalCustY, "###,##0.00")
CustPayDue.Caption = "Rs. " & Format(TotalCustN, "###,##0.00")
Label21.Caption = "Rs. " & Format(TotalCustYN, "###,##0.00")
```

```
    GB.BackColor = vbRed
    GB.ForeColor = vbYellow
End Sub
```

The above VBA code seems to be part of a Click event handler for a ListBox control named "CustNameList." This code appears to be designed to perform calculations and update various labels and components based on the selected item in the list. Here's a breakdown of the code:

Worksheet Declaration:

Dim wsCust As Worksheet Set wsCust = ThisWorkbook.Worksheets("Data")

This code segment declares a worksheet variable wsCust that refers to the "Data" worksheet within the workbook containing the VBA code.

Finding the Last Row:

Dim lastRow As Long lastRow = wsCust.Cells(wsCust.Rows.Count, "A").End(xlUp).Row

The code calculates the last row with data in column A of the "Data" worksheet using the xlUp method. It stores this value in the variable lastRow.

Defining a Range:

Dim rng As Range Set rng = wsCust.Range("A2:A" & lastRow)

This part creates a Range object named rng that covers data in column A of the "Data" worksheet, from cell A2 to the last row determined earlier.

Looping through Data:

For i = 1 To lastRow

Next i

This loop iterates through each row of data in the specified range (rng) and performs calculations based on the conditions specified inside the loop.

Conditions and Calculations:

```
If wsCust.Cells(i, 1).Value = LblCustId Then
    If wsCust.Cells(i, 8).Value = "Y" Then
        TotalCustY = TotalCustY + wsCust.Cells(i, "D").Value
    Else
        TotalCustN = TotalCustN + wsCust.Cells(i, "D").Value
    End If
End If
```

Within the loop, this section checks if the value in column A (presumably a customer ID) matches the value stored in LblCustId. Depending on the value in column H (presumably a "Y" or "N" for payment received), it updates the variables TotalCustY or TotalCustN with the corresponding payment amounts.

Updating Labels and Components:

```
LblCName.Caption = CustNameList.List(CustNameList.ListIndex, 1)
CustPayRecd.Caption = "Rs. " & Format(TotalCustY, "###,##0.00")
CustPayDue.Caption = "Rs. " & Format(TotalCustN, "###,##0.00")
Label21.Caption = "Rs. " & Format(TotalCustYN, "###,##0.00")
```

These lines update various label captions on the user interface with calculated values.

Changing Component Colors:

GB.BackColor = vbRed

GB.ForeColor = vbYellow

This code changes the background color and text color of a component (possibly a GroupBox or other container) to red and yellow, respectively.

Make sure that the variables, labels, and components mentioned in the code correspond correctly to those present in your user interface. Additionally, ensure that the logic of calculating payment amounts and updating labels aligns with your requirements and data structure.

Product Name List Click event, See earlier description:-

Private Sub ProductNameList_Click()

```
    Dim wsCust As Worksheet
    Set wsCust = ThisWorkbook.Worksheets("Data")

    Dim lastCustRow As Long
    lastCustRow = wsCust.Cells(wsCust.Rows.Count, "A").End(xlUp).row

    Dim rngCust As Range
    Set rngCust = wsCust.Range("A2:A" & lastCustRow)

    Dim wsProd As Worksheet
    Set wsProd = ThisWorkbook.Worksheets("Product_Master")
```

```
    Dim lastProdRow As Long
    lastProdRow = wsProd.Cells(wsProd.Rows.Count, "A").End(xlUp).row

    Dim rngProd As Range
    Set rngProd = wsProd.Range("A2:A" & lastProdRow)

    Dim DataRange As Long
    DataRange = lastProdRow

    LblProdId = ProductNameList.List(ProductNameList.ListIndex, 0)

    Dim TotalProdY, TotalProdN, TotalProdYN

    For i = 1 To lastCustRow
       If wsCust.Cells(i, 6).value = LblProdId Then
          'If wsCust.Cells(i, 8).value = "Y" Then
             'TotalProdY = TotalProdY + wsCust.Cells(i, "D").value
          'Else
             'TotalProdN = TotalProdN + wsCust.Cells(i, "D").value
          'End If
          TotalProdY = TotalProdY + wsCust.Cells(i, "D").value
        End If
       'TotalProdYN = TotalProdYN + TotalProdY
       Next i
```

```
    LblPName.Caption                                                    =
ProductNameList.List(ProductNameList.ListIndex, 1)

    'Label28.Caption = "Rs. " & Format(TotalY, "###,##0.00")
    'Label20.Caption = "Rs. " & Format(TotalN, "###,##0.00")
    Label28.Caption = "Rs. " & Format(TotalProdY, "###,##0.00")

    GB.BackColor = vbBlue
    GB.ForeColor = vbWhite

End Sub
```

SR Name List Click event, See earlier description:-

```
Private Sub SRNameList_Click()
    Dim ws3Cust As Worksheet
    Set ws3Cust = ThisWorkbook.Worksheets("Data")

    Dim lastCustRow As Long
    lastCustRow          =          ws3Cust.Cells(ws3Cust.Rows.Count,
"A").End(xlUp).row

    Dim rngCust As Range
    Set rngCust = ws3Cust.Range("A2:A" & lastCustRow)
```

```
Dim wsSR As Worksheet
Set wsSR = ThisWorkbook.Worksheets("Product_Master")

Dim lastSRRow As Long
lastSRRow = wsSR.Cells(wsSR.Rows.Count, "A").End(xlUp).row

Dim rngSR As Range
Set rngSR = wsSR.Range("A2:A" & lastSRRow)

Dim DataRange As Long
DataRange = lastSRRow

LblSRId = SRNameList.List(SRNameList.ListIndex, 0)

Dim TotalsrY, TotalsrN, TotalsrYN

For i = 1 To lastCustRow

    If ws3Cust.Cells(i, 9).value = LblSRId Then
        'If wsCust.Cells(i, 8).value = "Y" Then
            'TotalProdY = TotalProdY + wsCust.Cells(i, "D").value
        'Else
            'TotalProdN = TotalProdN + wsCust.Cells(i, "D").value
        'End If
```

```
            TotalsrY = TotalsrY + ws3Cust.Cells(i, "D").value
        End If

    Next i

    LblSRName.Caption = SRNameList.List(SRNameList.ListIndex, 1)

    'Label28.Caption = "Rs. " & Format(TotalY, "###,##0.00")
    'Label20.Caption = "Rs. " & Format(TotalN, "###,##0.00")
    Label32.Caption = "Rs. " & Format(TotalsrY, "###,##0.00")

    GB.BackColor = vbGreen
    GB.ForeColor = vbBlue

End Sub
```

UserForm_Initialize

```
Private Sub UserForm_Initialize()

    DataListDisplay
    CustNameListDisplay
    ProductNameListDisplay
    SRNameListDisplay
```

lblDate.Caption = "Today is " & Format(Date, "dddd") & ", Date: " & Format(Date, "dd/mm/yyyy")

End Sub

The provided VBA code appears to be part of the UserForm_Initialize event handler. This event occurs when the user form is initialized, often when it is first displayed. The code you provided is likely setting up and populating various controls on your user form. Here's a breakdown of the code:

Initializing Data Lists:

DataListDisplay

CustNameListDisplay

ProductNameListDisplay

SRNameListDisplay

These lines are calling four subroutines (presumably named DataListDisplay, CustNameListDisplay, ProductNameListDisplay, and SRNameListDisplay) that presumably populate different ListBox controls with data. Each of these subroutines is likely designed to populate its respective ListBox control with specific data.

Setting the Date Label:

lblDate.Caption = "Today is " & Format(Date, "dddd") & ", Date: " & Format(Date, "dd/mm/yyyy")

This line sets the caption of a label (presumably named lblDate) to display the current date. It uses the Format function to present the date in a specific format, including the day of the week and the date itself.

The UserForm_Initialize event handler is responsible for setting up the initial state of the user form when it's loaded. It appears to be loading data into ListBox controls and displaying the current date.

Make sure that the names of the ListBox controls (CustNameList, ProductNameList, SRNameList) and label (lblDate) match the actual names you have used in your user form. Also, ensure that the subroutines being called (DataListDisplay, CustNameListDisplay, ProductNameListDisplay, SRNameListDisplay) are defined in your code and properly populate the ListBox controls with the intended data.

By appropriately setting up the user form and its controls during initialization, you provide a smooth and informative experience for users interacting with your application.

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!

PLEASE DONATE

Gautam Banerjee

Age: 63

Pay by UPI

9748327614