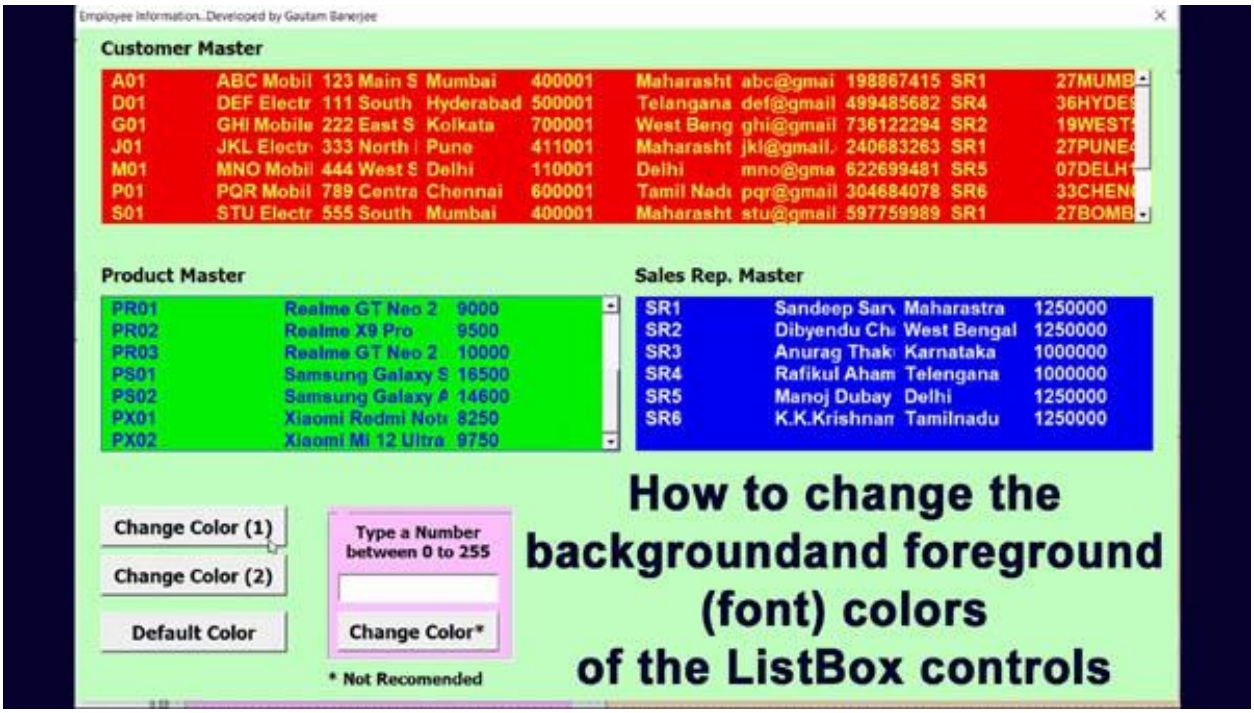

Insert List boxes, Command Buttons, Labels and Text Boxes. Connect Sheets with List boxes. Name Listboxes, Command Buttons in the property box. List box Name : CustNameList, ProductNameList and SRNameList. Command Buttons Name : CmdChangeColor1, CmdChangeColor2 and CmdChangeColor3. Check the name at the line of Private Sub command.

Private Sub UserForm_Initialize()

CustNameListDisplay

ProductNameListDisplay

SRNameListDisplay

```
    ApplyListBoxFormatting CustNameList, RGB(240, 0, 0), RGB(250, 250, 50), "Arial", 14, True

    ApplyListBoxFormatting ProductNameList, RGB(0, 240, 0), RGB(0, 0, 254), "Arial", 18, True

    ApplyListBoxFormatting SRNameList, RGB(0, 0, 240), RGB(250, 255, 254), "Arial", 22, True

End Sub
```

The given code is part of a VBA UserForm's Initialize event procedure. It is executed automatically when the UserForm is loaded or initialized. This procedure is responsible for setting up the initial state and appearance of the UserForm and its associated ListBox controls (CustNameList, ProductNameList, and SRNameList).

Here's a breakdown of what the code is doing:

CustNameListDisplay, ProductNameListDisplay, and SRNameListDisplay are likely subroutines or functions that populate the corresponding ListBoxes (CustNameList, ProductNameList, and SRNameList) with data. These subroutines are not shown in the provided code, but they are presumably responsible for filling the ListBoxes with relevant items.

The code uses the ApplyListBoxFormatting subroutine to customize the appearance of the ListBox controls. This subroutine seems to be a custom function that applies formatting to a ListBox. The provided code is calling this subroutine for each of the three ListBoxes.

ApplyListBoxFormatting is being called with the following arguments for CustNameList:

CustNameList: The ListBox control to which formatting will be applied.

RGB(240, 0, 0): The background color (red) of the ListBox items.

RGB(250, 250, 50): The font color (yellow) of the ListBox items.

"Arial": The font name for the ListBox items.

14: The font size for the ListBox items.

True: Specifies whether the font is bold for the ListBox items.

Similar arguments are provided for ProductNameList and SRNameList with different color, font size, and other formatting settings.

The purpose of this code is to ensure that when the UserForm is initialized, the ListBox controls (CustNameList, ProductNameList, and SRNameList) are populated with data and formatted with specified colors, font properties, and sizes. This customization enhances the visual appearance of the ListBox controls and provides a consistent and visually appealing user interface.

```
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:J" & lastRow) ' Assuming data starts from A2
    CustNameList.Clear
    CustNameList.List = rng.value
End Sub
```

The provided code is a subroutine named CustNameListDisplay that populates the CustNameList ListBox control with data from the "Customer_Master" worksheet. Here's a breakdown of what the code is doing:

Worksheet Setup:

The code starts by declaring a worksheet object named wsCust and setting it to refer to the worksheet named "Customer_Master" in the workbook where the code is running (ThisWorkbook).

Finding the Last Row:

The code determines the last used row in column A of the "Customer_Master" worksheet using the .End(xlUp) method. This is done to find the last row of data in the worksheet.

Defining the Data Range:

A Range object named rng is defined to cover the range from cell A2 to column J of the last used row in the "Customer_Master" worksheet. This is the range from which data will be extracted for the ListBox.

Clearing and Populating the ListBox:

The CustNameList ListBox is cleared to ensure that it's empty before populating it with new data.

The List property of the CustNameList ListBox is assigned the values from the rng Range object. This effectively populates the ListBox with the values from the specified range.

In summary, when this CustNameListDisplay subroutine is called, it fetches data from the "Customer_Master" worksheet's range starting from A2 to J(lastRow) and populates the CustNameList ListBox with the retrieved data. This subroutine is typically called as part of the UserForm's initialization process or when data needs to be refreshed in the ListBox.

Produxt_Master and SR master List box description as above. Only need to change the Variables name and Sheet Name.

```
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:C" & 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:D" & lastRow) ' Assuming data starts from A2

    SRNameList.Clear
```

```
    SRNameList.List = rng.value

End Sub
```

(Description as above Customaster_Master List box code)

```
Private Sub ApplyListBoxFormatting(lstBox As MSForms.ListBox, bgColor As Long, fontColor As Long, fontName As String, fontSize As Integer, isBold As Boolean)
    ' Change background color
    lstBox.BackColor = bgColor

    ' Change font style
    With lstBox.Font
        .Name = "Arial"
        .Size = 14
        .Bold = True
    End With

   ' Set font color
    lstBox.ForeColor = fontColor

    ' Apply design principles
    lstBox.BorderStyle = fmBorderStyleSingle
    lstBox.BorderColor = fontColor
```

End Sub

The provided code defines a subroutine named ApplyListBoxFormatting that applies specific formatting to a given ListBox control. Here's a breakdown of what the code does:

Parameter Inputs:

The subroutine takes several input parameters:

lstBox (MSForms.ListBox): The ListBox control to which formatting will be applied.

bgColor (Long): The background color to be applied.

fontColor (Long): The font color to be applied.

fontName (String): The font name to be applied.

fontSize (Integer): The font size to be applied.

isBold (Boolean): Whether the font should be bold.

Background Color:

The background color of the lstBox ListBox control is set to the specified bgColor.

Font Style:

The code uses the With statement to change the font properties of the lstBox ListBox control.

The font name is set to "Arial".

The font size is set to 14 (although this value is hard-coded and doesn't use the fontSize parameter).

The font boldness is set to True.

Font Color:

The font color of the lstBox ListBox control is set to the specified fontColor.

Design Principles:

The border style of the lstBox ListBox control is set to fmBorderStyleSingle, creating a single-line border around the ListBox items.

The border color is set to the specified fontColor.

In summary, the ApplyListBoxFormatting subroutine is designed to apply consistent formatting to a ListBox control. However, there are a few things to note:

The fontName, fontSize, and isBold parameters are not utilized within the subroutine.

The font properties within the With lstBox.Font block are hardcoded, which means the formatting applied may not match the intended parameter values.

You might want to revise the code to utilize the input parameters and apply dynamic formatting based on the provided values.

```
Private Sub CmdChangeColor1_Click()


    Dim newBgColor As Long, NewForeColor As Long

    ' Show color picker or input field to get new background color

    newBgColor = RGB(247, 151, 247)

    NewForeColor = RGB(0, 0, 255)


    ' Apply new background color to ListBox

    ApplyListBoxFormatting CustNameList, newBgColor, NewForeColor, "Arial", 16, True

    ApplyListBoxFormatting ProductNameList, newBgColor, ProductNameList.ForeColor, "Arial", 18, True

    ApplyListBoxFormatting SRNameList, newBgColor, SRNameList.ForeColor, "Arial", 14, True
```

End Sub

The provided code is a subroutine named CmdChangeColor1_Click that is likely associated with a command button click event. This subroutine is responsible for changing the background color and font color of three ListBox controls (CustNameList, ProductNameList, and SRNameList) based on predefined color values. Here's what the code does:

New Color Values:

Two variables, newBgColor and NewForeColor, are declared to store new background and font colors.

In this example, newBgColor is set to RGB(247, 151, 247) and NewForeColor is set to RGB(0, 0, 255).

Apply New Formatting:

The code then calls the ApplyListBoxFormatting subroutine three times to apply the new background color and font color to each of the three ListBoxes.

For CustNameList, the new background color is applied, and the new font color is set to the same font color as before.

For ProductNameList, the new background color is applied, and the font color remains unchanged.

For SRNameList, the new background color is applied, and the font color remains unchanged.

Parameters for Formatting:

The ApplyListBoxFormatting subroutine is called with different combinations of parameters to set the background color, font color, font name, font size, and boldness for each ListBox.

In summary, this code represents an event handler for a command button click. When the button is clicked, the background color of the ListBoxes (CustNameList, ProductNameList, and SRNameList) is changed to a specific color (newBgColor), and the font color remains unchanged for two of the ListBoxes (ProductNameList and SRNameList). This functionality allows for

quick and consistent customization of the ListBox appearance using predefined color values.

```
Private Sub CmdChangeColor2_Click()

    Dim newBgColor As Long, NewForeColor As Long
   ' Show color picker or input field to get new background color
   newBgColor = RGB(250, 210, 12)
   NewForeColor = RGB(250, 0, 0)

   ' Apply new background color to ListBox
   ApplyListBoxFormatting CustNameList, newBgColor, NewForeColor, "Arial", 16, True
   ApplyListBoxFormatting ProductNameList, newBgColor, ProductNameList.ForeColor, "Arial", 18, True
   ApplyListBoxFormatting SRNameList, newBgColor, SRNameList.ForeColor, "Arial", 14, True

End Sub

Private Sub CmdChangeColor3_Click()

    Dim newBgColor As Long, NewForeColor As Long
   ' Show color picker or input field to get new background color
   newBgColor = RGB(255, 255, 255)
   NewForeColor = RGB(0, 0, 0)
```

```
    ' Apply new background color to ListBox
    ApplyListBoxFormatting CustNameList, newBgColor, NewForeColor, "Arial", 16, True
    ApplyListBoxFormatting ProductNameList, newBgColor, NewForeColor, "Arial", 18, True
    ApplyListBoxFormatting SRNameList, newBgColor, NewForeColor, "Arial", 14, True
End Sub

Private Sub CommandButtonChangeBgColor_Click()
    Dim newBgColor As Long

    ' Get the user's input from the textbox
    If IsNumeric(TextBoxBgColor.Text) Then
        newBgColor = CLng(TextBoxBgColor.Text)
        ' Apply the new background color to ListBox
        ApplyListBoxFormatting CustNameList, newBgColor, CustNameList.ForeColor, "Arial", 10, True
    Else
        MsgBox "Please enter a valid numeric color value."
    End If
End Sub
```

The provided code is a subroutine named CommandButtonChangeBgColor_Click that is likely associated with a command button click event. This subroutine is responsible for changing the background color of a ListBox control (CustNameList) based on a numeric

color value entered by the user into a TextBox (TextBoxBgColor). Here's what the code does:

New Background Color:

A variable named newBgColor of type Long is declared to store the new background color value.

Get User's Input:

The code checks if the value entered in the TextBoxBgColor is numeric using the IsNumeric function.

If the entered value is numeric, the code converts it to a Long data type using the CLng function and assigns it to the newBgColor variable.

Apply New Background Color:

If a valid numeric color value is entered, the code calls the ApplyListBoxFormatting subroutine to apply the new background color to the CustNameList ListBox control.

The font color, font name, font size, and boldness remain unchanged.

Invalid Input Handling:

If the entered value is not numeric, a message box displays an error message asking the user to enter a valid numeric color value.

In summary, this code provides a user interface for changing the background color of the CustNameList ListBox control by allowing the user to enter a numeric color value into a TextBox. If the entered value is valid, the background color is changed accordingly using the ApplyListBoxFormatting subroutine. If the entered value is not numeric, an error message is shown to the user.

Here is the Full Code :

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

```
Private Sub CmdChangeColor1_Click()

    Dim newBgColor As Long, NewForeColor As Long
    ' Show color picker or input field to get new background color
    newBgColor = RGB(247, 151, 247)
```

```
    NewForeColor = RGB(0, 0, 255)

    ' Apply new background color to ListBox
    ApplyListBoxFormatting CustNameList, newBgColor, NewForeColor, "Arial", 16, True
    ApplyListBoxFormatting ProductNameList, newBgColor, ProductNameList.ForeColor, "Arial", 18, True
    ApplyListBoxFormatting SRNameList, newBgColor, SRNameList.ForeColor, "Arial", 14, True

End Sub

Private Sub CmdChangeColor2_Click()

     Dim newBgColor As Long, NewForeColor As Long
    ' Show color picker or input field to get new background color
    newBgColor = RGB(250, 210, 12)
    NewForeColor = RGB(250, 0, 0)

    ' Apply new background color to ListBox
    ApplyListBoxFormatting CustNameList, newBgColor, NewForeColor, "Arial", 16, True
    ApplyListBoxFormatting ProductNameList, newBgColor, ProductNameList.ForeColor, "Arial", 18, True
    ApplyListBoxFormatting SRNameList, newBgColor, SRNameList.ForeColor, "Arial", 14, True
```

```
End Sub

Private Sub CmdChangeColor3_Click()

    Dim newBgColor As Long, NewForeColor As Long
   ' Show color picker or input field to get new background color
   newBgColor = RGB(255, 255, 255)
   NewForeColor = RGB(0, 0, 0)

   ' Apply new background color to ListBox
   ApplyListBoxFormatting CustNameList, newBgColor, NewForeColor, "Arial", 16, True
   ApplyListBoxFormatting ProductNameList, newBgColor, NewForeColor, "Arial", 18, True
   ApplyListBoxFormatting SRNameList, newBgColor, NewForeColor, "Arial", 14, True
End Sub
Private Sub CommandButtonChangeBgColor_Click()
   Dim newBgColor As Long

   ' Get the user's input from the textbox
   If IsNumeric(TextBoxBgColor.Text) Then
      newBgColor = CLng(TextBoxBgColor.Text)
      ' Apply the new background color to ListBox
      ApplyListBoxFormatting CustNameList, newBgColor, CustNameList.ForeColor, "Arial", 10, True
   Else
```

```
        MsgBox "Please enter a valid numeric color value."
    End If
End Sub

Private Sub UserForm_Initialize()

    CustNameListDisplay
    ProductNameListDisplay
    SRNameListDisplay

    ApplyListBoxFormatting CustNameList, RGB(240, 0, 0), RGB(250, 250, 50), "Arial", 14, True
    ApplyListBoxFormatting ProductNameList, RGB(0, 240, 0), RGB(0, 0, 254), "Arial", 18, True
    ApplyListBoxFormatting SRNameList, RGB(0, 0, 240), RGB(250, 255, 254), "Arial", 22, True
End Sub

Private Sub ApplyListBoxFormatting(lstBox As MSForms.ListBox, bgColor As Long, fontColor As Long, fontName As String, fontSize As Integer, isBold As Boolean)
    ' Change background color
    lstBox.BackColor = bgColor

    ' Change font style
    With lstBox.Font
        .Name = "Arial"
```

```
        .Size = 14
        .Bold = True
    End With

   ' Set font color
    lstBox.ForeColor = fontColor

    ' Apply design principles
    lstBox.BorderStyle = fmBorderStyleSingle
    lstBox.BorderColor = fontColor

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:J" & 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:C" & 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:D" & lastRow) ' Assuming data starts from A2

    SRNameList.Clear
    SRNameList.List = rng.value

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