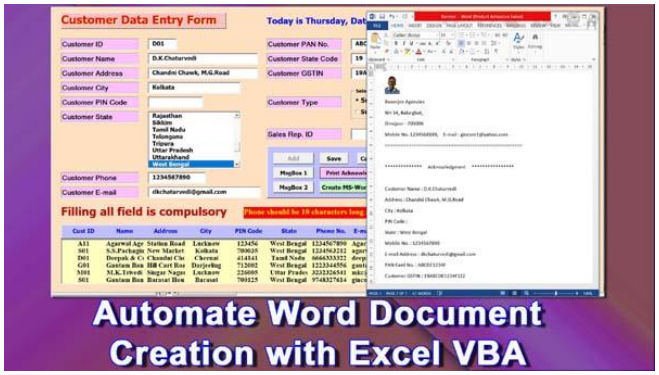
Add Two Command Buttons:
Name : 1. CmdAckPrint and 2. CmdPrint

1. CmdAckPrint procedure

```
Private Sub CmdAckPrint_Click()
   SaveTextBoxValueToWord      'Call below Procedure here
End Sub

Sub SaveTextBoxValueToWord()
   Dim wdApp As Object ' Word Application
   Dim wdDoc As Object ' Word Document
   Dim txtName, txtAdd, txtCity, txtPIN, txtState, txtMobile, txtEmail,
txtPan, txtGST As String ' Value from UserForm TextBox

   ' Create a new instance of Word
   Set wdApp = CreateObject("Word.Application")

   ' Add a new document
   Set wdDoc = wdApp.Documents.Add

   ' Show the Word application
   wdApp.Visible = True

   ' Get the value from the UserForm TextBox
   txtName = custdataentryform.TxtCustName.Value
   txtAdd = custdataentryform.TxtCustAdd.Value
   txtCity = custdataentryform.TxtCustCity.Value
   txtPIN = custdataentryform.TxtCustPin.Value
   txtState = custdataentryform.StatesList.Value
   txtMobile = custdataentryform.TxtCustPhone.Value
   txtEmail = custdataentryform.TxtCustEmail.Value
```

Gautam Banerjee
E-mail: gincom1@yahoo.com



When the user clicks the "Print Acknowledgment" button, the `CmdAckPrint_Click` event is triggered. This event, in turn, calls the `SaveTextBoxValueToWord` procedure, which automates the process of collecting user input from the UserForm and generating a formatted Word document. This seamless integration empowers you to swiftly create professional acknowledgment slips with a single button click, saving you time and effort.

txtPan = custdataentryform.TxtCustPan.Value txtGST = custdataentryform.TxtCustGstin.Value ' Insert logo or picture wdDoc.InlineShapes.AddPicture "D:\ExcelVBA\GB.png" ' Replace with the path to your image wdDoc.Content.InsertAfter "" & vbNewLine wdDoc.Content.InsertAfter "Banerjee Agencies" & vbNewLine wdDoc.Content.InsertAfter "NH 34, Balurghat," & vbNewLine wdDoc.Content.InsertAfter "Dinajpur - 705006" & vbNewLine wdDoc.Content.InsertAfter "Mobile No. 123456XXXX, E-mail : gincom1@yahoo.com" & vbNewLine wdDoc.Content.InsertAfter "==========" & vbNewLine & vbNewLine wdDoc.Content.InsertAfter "********** Acknowledgment **********" & vbNewLine & vbNewLine ' Insert the value into the Word document wdDoc.Content.InsertAfter "Customer Name : " & txtName & vbNewLine wdDoc.Content.InsertAfter "Address : " & txtAdd & vbNewLine wdDoc.Content.InsertAfter "City : " & txtCity & vbNewLine wdDoc.Content.InsertAfter "PIN Code : " & txtPIN & vbNewLine wdDoc.Content.InsertAfter "State : " & txtState & vbNewLine wdDoc.Content.InsertAfter "Mobile No. : " & txtMobile & vbNewLine wdDoc.Content.InsertAfter "E-mail Address : " & txtEmail & vbNewLine wdDoc.Content.InsertAfter "PAN Card No. : " & txtPan & vbNewLine wdDoc.Content.InsertAfter "Customer GSTIN : " & txtGST & vbNewLine	This code segment establishes a bridge between your Excel VBA application and Microsoft Word. It initializes the necessary variables, **`wdApp`** for the Word application instance and **`wdDoc`** for the Word document instance. Upon creating a new Word application instance, a fresh document is added to it. The Word application window becomes visible, allowing you to interact with and manipulate the document through your Excel VBA script. This integration forms the foundation for automating the generation and formatting of Word documents directly from Excel, enhancing your workflow and productivity. In this section, the values entered by the user in the Excel UserForm are captured and stored in individual variables, such as **`txtName`**, **`txtAdd`**, **`txtCity`**, and so on. These variables hold the user's information, including their name, address, city, PIN code, state, mobile number, email, PAN card number, and GSTIN. Furthermore, an image, indicated by the path "D:\ExcelVBA\GB.png", is inserted into the Word document. This image could be a logo or any other picture you wish to include in the generated document. Replace the image path with the actual location of your image file to ensure accurate insertion.

```
   wdDoc.Content.InsertAfter
"================================================" &
vbNewLine
   wdDoc.Content.InsertAfter "Note : If you found any error, Please
inform us     " & vbNewLine
   wdDoc.Content.InsertAfter "Thanking you     "

   ' Save the Word document
   wdDoc.SaveAs "D:\ExcelVBA\Receipt.docx" ' Replace with the desired
file path

   ' Clean up
   Set wdDoc = Nothing
   Set wdApp = Nothing
End Sub
```

2. CmdPrint Procedure

```
Private Sub CmdPrint_Click()
   SaveTextBoxValueToWordAsTableWithHeading
End Sub


Sub SaveTextBoxValueToWordAsTableWithHeading()
   Dim wdApp As Object ' Word Application
   Dim wdDoc As Object ' Word Document
   Dim wdTable As Object ' Word Table
   Dim txtName, txtAdd, txtCity, txtPIN, txtState, txtMobile, txtEmail,
txtPan, txtGST As String ' Value from UserForm TextBox

   ' Create a new instance of Word
   Set wdApp = CreateObject("Word.Application")
```

Within this portion of the code, an introduction and acknowledgment header are thoughtfully inserted into the Word document. A blank line is initially added to provide spacing, followed by lines of text that collectively set the tone for the acknowledgment slip.

The inserted lines include contact details such as the entity name "**Banerjee Agencies," the physical address "NH 34, Balurghat," the location "Dinajpur - 705006," and contact information "Mobile No. 123456XXXX, E-mail: gincom1@yahoo.com**." These details enhance the credibility of the acknowledgment and provide recipients with a means of communication.

Within this code segment, the user input data collected from the Excel UserForm is seamlessly integrated into the Word document. Each line corresponds to a specific data field, such as customer name, address, city, PIN code, state, mobile number, email address, PAN card number, and customer GSTIN.

Using the variables `txtName`, `txtAdd`, `txtCity`, and so on, the code dynamically combines the labels with the corresponding user input data, ensuring

```
    ' Add a new document
    Set wdDoc = wdApp.Documents.Add

    ' Show the Word application
    wdApp.Visible = True

    ' Get the value from the UserForm TextBox
    txtName = custdataentryform.TxtCustName.Value
    txtAdd = custdataentryform.TxtCustAdd.Value
    txtCity = custdataentryform.TxtCustCity.Value
    txtPIN = custdataentryform.TxtCustPin.Value
    txtState = custdataentryform.StatesList.Value
    txtMobile = custdataentryform.TxtCustPhone.Value
    txtEmail = custdataentryform.TxtCustEmail.Value
    txtPan = custdataentryform.TxtCustPan.Value
    txtGST = custdataentryform.TxtCustGstin.Value



    ' Create a new table with 2 columns
    Set wdTable = wdDoc.Tables.Add(Range:=wdDoc.Content,
NumRows:=11, NumColumns:=2)
    wdTable.Borders.Enable = True ' Add borders to the table


    ' Insert data into the table cells

    wdTable.Cell(1, 1).Range.Text = ""
    wdTable.Cell(1, 2).Range.Text = ""
    wdTable.Cell(2, 1).Range.Text = "Customer Name"
    wdTable.Cell(2, 2).Range.Text = txtName
    wdTable.Cell(3, 1).Range.Text = "Address"
    wdTable.Cell(3, 2).Range.Text = txtAdd
```

that each piece of information is accurately and neatly presented within the Word document.

In this concluding part of the script, the Word document's content is seamlessly concluded with a visual separator line made up of equal signs. Following this, a note is added that invites recipients to inform you of any errors they may have noticed, fostering a sense of transparency and cooperation.

The acknowledgment concludes with a heartfelt "Thanking you," signifying your gratitude to the recipient for their attention and cooperation.

After the content has been successfully inserted and formatted, the Word document is saved to a designated file path using the `wdDoc.SaveAs` method. This ensures that the meticulously created acknowledgment is stored for future reference.

Lastly, the script performs essential cleanup operations by releasing the memory associated with the `wdDoc` and `wdApp` objects, thus optimizing resources and ensuring a tidy conclusion to the automation process.

Incorporate this description to provide a comprehensive understanding of this particular

```
    wdTable.Cell(4, 1).Range.Text = "City"
    wdTable.Cell(4, 2).Range.Text = txtCity
    wdTable.Cell(5, 1).Range.Text = "PIN Code"
    wdTable.Cell(5, 2).Range.Text = txtPIN
    wdTable.Cell(6, 1).Range.Text = "State"
    wdTable.Cell(6, 2).Range.Text = txtState
    wdTable.Cell(7, 1).Range.Text = "Mobile No."
    wdTable.Cell(7, 2).Range.Text = txtMobile
    wdTable.Cell(8, 1).Range.Text = "E-mail Address"
    wdTable.Cell(8, 2).Range.Text = txtEmail
    wdTable.Cell(9, 1).Range.Text = "PAN Card No."
    wdTable.Cell(9, 2).Range.Text = txtPan
    wdTable.Cell(10, 1).Range.Text = "Customer GSTIN"
    wdTable.Cell(10, 2).Range.Text = txtGST

    ' Insert heading lines before the tabl

    wdTable.Range.InsertBefore "***** Acknowledgment *****"
    wdTable.Range.InsertBefore "==============================="
& vbNewLine
    wdTable.Range.InsertBefore "Mobile No. 123456XXXX,    E-mail :
gincom1@yahoo.com" & vbNewLine
    wdTable.Range.InsertBefore "Dinajpur - 705006" & vbNewLine
    wdTable.Range.InsertBefore "NH 34, Balurghat," & vbNewLine
    wdTable.Range.InsertBefore "Banerjee Agencies" & vbNewLine
    wdTable.Range.InsertBefore "Banerjee Agencies" & vbNewLine
    wdTable.Range.InsertBefore "  " & vbNewLine
    wdDoc.InlineShapes.AddPicture "D:\ExcelVBA\GB.png" ' Replace with
the path to your image
    wdTable.Range.InsertAfter "Please check the data, if you found any
error," & vbNewLine
    wdTable.Range.InsertAfter "Contact us or our Sales Representative
immediately"
```

code segment within your documentation or explanations.

If you found this Excel VBA automation solution valuable and it has saved you time and effort, please consider making a small donation to support our efforts in creating more useful content and tools. Your contribution helps us continue to provide free resources and tutorials that empower individuals like you to streamline their workflows and enhance productivity.

Your support is greatly appreciated and motivates us to continue creating and sharing high-quality content to help you excel in your projects.

[Donate Now by UPI : 97483 27614]

Thank you for being a part of our community and for considering making a donation. Your generosity makes a significant impact!

' Save the Word document wdDoc.SaveAs "D:\ExcelVBA\ReceiptWithTableAndHeading.docx" ' Replace with the desired file path ' Clean up Set wdTable = Nothing Set wdDoc = Nothing Set wdApp = Nothing End Sub	NAME : GAUTAM BANERJEE AGE : 63 YEARS SERVICE : RETIRED FROM A PRIVATE COMPANY