



Video Link (This Video): https://youtu.be/xJ0_vbWUnEA

Video Link (Last Video): <https://youtu.be/m7epJngZQZE>

"Hello, friends! Today, let's create a different kind of picture viewer form. While watching this video, make sure to check out our previous tutorials; they contain valuable insights. A programmer should be well-versed in various programming techniques. Whether you're working on data entry or data analysis programs, creating a project like this can prove immensely beneficial.

Let's start by creating the user form. Insert the user form from the 'Insert' menu, and adjust its dimensions by dragging. Then, add an image frame from the toolbox and customize it to your liking. Create a label to display picture descriptions. Next, add labels and number them. Under the 'Format' menu, adjust alignment, size, and spacing. Add two image frames and select the arrow from the 'Picture' option in 'Properties.' Keep in mind that bitmap (bmp) images should have a jpg extension. Optionally, you can create 'Next' and 'Previous' labels.

Now, let's dive into the coding. First, initialize the user form. This function determines which elements are visible when the form opens. Please refer to our previous video for a detailed explanation. In this initialization, I've showcased a picture. I've also created a separate procedure called 'ByDefaultPicture.'

In this procedure, you need to specify the picture folder's reference and use a logical 'if' statement to check if the picture is present in the folder. If found, it will be displayed; otherwise, the loop exits. You should call this procedure during user initialization. Additionally, I've created a variable

with an initial value of 4. The reason for this is that there are labels numbered from 4 to 23 at the bottom of the form. The first label is named 'Label4,' and you'll see why this matters later.

Now, let's tackle the label click event. 'Dim clickedLabel As Object' declares the variable 'clickedLabel' as an object, which means it will store a label from this form's controls. 'currentLabelNumber' is a variable that holds the sixth character of the label (e.g., '4' from 'Label4'). The 'Val' function converts a character to a numeric format. 'i' is another variable whose value will be added to 'currentLabelNumber.' For instance, if the sixth character of 'Label4' is '4,' then 'i' will add 4 to it. I've assigned a value to each label and called the 'Label_Click' procedure accordingly. I've also created a PDF file containing the complete code with detailed explanations; you can download it from the description.

Now, let's write the code for the 'Next' button click function. The critical point to note here is that if 'currentLabelNumber' is greater than 4, it will decrement by 1 on each click. When it reaches 4, it will reset to 23. The image file names range from 'Photo4' to 'Photo23,' so I've used values from 4 to 23. When referencing the picture path, it looks like this: 'E:\Eco\Photo' & currentLabelNumber & '.jpg' in the 'E' drive's 'Eco' folder.

The same logic applies to the 'Previous' button click, with the only difference being an increment instead of a decrement. When it reaches 23, it resets to 4.

Lastly, we need to write the 'UpdateLabelAndImage' code. Clicking 'Next' or 'Previous' will update the picture and label. Create variables for the image and label, assign the picture path to the variable, and use a logical statement to display the picture if found; otherwise, exit the loop.

To close the form, use the 'Unload Me' command. Thanks for watching!"

Bus program complete.

Copy below Command and paste in your application :

Dim currentLabelNumber As Integer

Dim i As Integer

Private Sub CommandButton1_Click()

Unload Me

End Sub

Private Sub Label4_Click()

i = 4

Label_Click

End Sub

Private Sub Label5_Click()

i = 5

Label_Click

End Sub

Private Sub Label6_Click()

i = 6

Label_Click

End Sub

Private Sub Label7_Click()

i = 7

Label_Click

End Sub

Private Sub Label8_Click()

i = 8

Label_Click

End Sub

Private Sub Label9_Click()

i = 9

Label_Click

End Sub

Private Sub Label10_Click()

i = 10

Label_Click

End Sub

Private Sub Label11_Click()

i = 11

Label_Click

End Sub

Private Sub Label12_Click()

i = 12

Label_Click

End Sub

Private Sub Label13_Click()

i = 13

Label_Click

End Sub

Private Sub Label14_Click()

i = 14

Label_Click

End Sub

Private Sub Label15_Click()

i = 15

Label_Click

End Sub

Private Sub Label16_Click()

i = 16

Label_Click

End Sub

Private Sub Label17_Click()

i = 17

Label_Click

End Sub

Private Sub Label18_Click()

i = 18

Label_Click

End Sub

Private Sub Label19_Click()

i = 19

Label_Click

End Sub

Private Sub Label20_Click()

i = 20

Label_Click

End Sub

Private Sub Label21_Click()

i = 21

Label_Click

End Sub

Private Sub Label22_Click()

i = 22

Label_Click

End Sub

Private Sub Label23_Click()

i = 23

Label_Click

End Sub

Private Sub Label_Click()

Dim clickedLabel As Object

Set clickedLabel = Me.ActiveControl

'Dim clickedLabel As Integer

' Extract the number from the label's name

currentLabelNumber = Val(Mid(clickedLabel.Caption, 6)) + i

Dim imagePath As String

*imagePath = "E:\Eco\Photo" & currentLabelNumber & ".jpg" ' Adjust the
path as needed*

' Load and display the image in the Image control

If Len(Dir(imagePath)) > 0 Then ' Check if the image file exists

Me.Image1.Picture = LoadPicture(imagePath)

Else

' Clear the Image control if the image file does not exist

Me.Image1.Picture = LoadPicture("") ' This will clear the image control

End If

End Sub

Private Sub CmdNxt_Click()

If currentLabelNumber < 23 Then ' Adjust the upper limit to the total number of labels

currentLabelNumber = currentLabelNumber + 1

Else

' Wrap around to the first label if you reach the end

currentLabelNumber = 4

End If

' Update the label caption and display the image

UpdateLabelAndImage

End Sub

Private Sub CmdPrv_Click()

If currentLabelNumber > 4 Then ' Adjust the lower limit to the first label

currentLabelNumber = currentLabelNumber - 1

Else

' Wrap around to the last label if you reach the beginning

currentLabelNumber = 23 ' Adjust this to the last label number

End If

' Update the label caption and display the image

UpdateLabelAndImage

End Sub

Private Sub UpdateLabelAndImage()

Dim imagePath As String

Dim labelCaption As String

' Construct the image file path based on the currentLabelNumber

*imagePath = "E:\Eco\Photo" & currentLabelNumber & ".jpg" ' Adjust the
path as needed*

' Get the caption for the current label from your array or data source

*labelCaption = "Caption for Label" & currentLabelNumber ' Replace with
your source of captions*

' Load and display the image in the Image control

If Len(Dir(imagePath)) > 0 Then ' Check if the image file exists

Image1.Picture = LoadPicture(imagePath)

Else

' Clear the Image control if the image file does not exist

*Me.Image1.Picture = LoadPicture("E:\Eco\A.jpg") ' This will clear the
image control*

End If

' Set the caption for the label

'Me.Label.Caption = labelCaption

End Sub

Private Sub UserForm_Initialize()

currentLabelNumber = 4 ' Initial label number

byDefaultPicture

End Sub

Private Sub byDefaultPicture()

' Set the default image file path

*defaultImagePath = "E:\Eco\Default.jpg" ' Change to the path of your
default image*

' Load and display the default image in the Image control

```

If Len(Dir(defaultImagePath)) > 0 Then ' Check if the default image file exists

    Image1.Picture = LoadPicture(defaultImagePath)

Else

    ' Clear the Image control if the default image file does not exist

    Image1.Picture = LoadPicture("") ' This will clear the image control

End If

Label3.Caption = "The Eco-Park in Action Area II of New Town is one of the largest open spaces in the city." & vbCrLf & _
"The site is spread across an area of 194 hectares (480 acres) and includes a 42-hectare (100 acres) water body with an island. The Central Business District (CBD) is located to the east of the site and one of the largest convention centres in the country, Biswa Bangla Convention Centre, has been built to its north." & vbCrLf & _
"Photography : Gautam Banerjee"

End Sub

```

The above code appears to be a VBA (Visual Basic for Applications) script designed to work with a Microsoft Office application, such as Excel or Word, within a user form. It seems to be associated with a user interface that allows the user to interact with labels and an image control to display images. Here's an overall breakdown and comments about the programming:



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

Variable Declarations:

The code declares several variables, including `currentLabelNumber`, `i`, `clickedLabel`, and `imagePath`, among others. These variables are used to store and manipulate data throughout the script.

Event Handlers:

The code defines event handler procedures for various controls within the user form, such as `CommandButton1_Click`, `Label4_Click`, `CmdNxt_Click`, and `CmdPrv_Click`. These event handlers respond to user interactions with buttons and labels.

Label Click Event:

The Label_Click event handler is used to display images corresponding to labels when they are clicked. It extracts a number from the label's caption, combines it with the value of i, constructs a file path, and loads and displays the image in an image control. If the image file does not exist, it loads a default image.

```
Private Sub Label_Click()  
    ' Declare a variable to store the clicked label control  
    Dim clickedLabel As Object  
  
    ' Set the clickedLabel variable to the label control that was clicked  
    Set clickedLabel = Me.ActiveControl  
  
    ' Extract the number from the label's name  
    ' This code assumes that the label names have a specific format like  
    "LabelX" where X is a number.  
    ' It extracts the number part and adds it to the variable i.  
    currentLabelNumber = Val(Mid(clickedLabel.Caption, 6)) + i  
  
    ' Construct the image file path based on the currentLabelNumber  
    Dim imagePath As String  
    imagePath = "E:\Eco\Photo" & currentLabelNumber & ".jpg" ' Adjust the  
    path as needed  
  
    ' Load and display the image in the Image control  
    If Len(Dir(imagePath)) > 0 Then ' Check if the image file exists  
        Me.Image1.Picture = LoadPicture(imagePath)  
    Else  
        ' Clear the Image control if the image file does not exist  
        Me.Image1.Picture = LoadPicture("") ' This will clear the image  
        control  
    End If  
End Sub
```

Here's a breakdown of the code:

Dim clickedLabel As Object: This line declares a variable named clickedLabel to store the clicked label control.

Set clickedLabel = Me.ActiveControl: This line sets the clickedLabel variable to the label control that was clicked. It uses the ActiveControl property of the user form to determine which label was clicked.

currentLabelNumber = Val(Mid(clickedLabel.Caption, 6)) + i: This line extracts a number from the caption of the clicked label. It assumes that your label names have a specific format like

"LabelX" where X is a number. It extracts the number by taking a substring of the caption starting from the 6th character (assuming "Label" is the first 5 characters) and converts it to a numeric value. It then adds this number to the i variable.

imagePath = "E:\Eco\Photo" & currentLabelNumber & ".jpg": This line constructs the file path for the image based on the currentLabelNumber. It assumes that your image files are named in a way that corresponds to the label numbers.

If Len(Dir(imagePath)) > 0 Then: This line checks if the image file specified by imagePath exists.

Me.Image1.Picture = LoadPicture(imagePath): If the image file exists, this line loads and displays the image in the Image1 control.

Me.Image1.Picture = LoadPicture(""): If the image file does not exist, this line clears the Image1 control by loading a blank picture.

In summary, this subroutine is designed to dynamically load and display images in an Image control based on which label was clicked, and it extracts the label number from the label's name to determine which image to display.

UpdateLabelAndImage Subroutine:

This subroutine constructs the file path for the image based on the currentLabelNumber, retrieves the label caption (which should come from your data source), and updates the image control and label caption accordingly. If the image file doesn't exist, it displays a default image.

```
Private Sub UpdateLabelAndImage()  
    Dim imagePath As String  
    Dim labelCaption As String  
  
    ' Construct the image file path based on the currentLabelNumber  
    imagePath = "E:\Eco\Photo" & currentLabelNumber & ".jpg" ' Adjust the  
    path as needed  
  
    ' Get the caption for the current label from your array or data source  
    labelCaption = "Caption for Label" & currentLabelNumber ' Replace with  
    your source of captions  
  
    ' Load and display the image in the Image control  
    If Len(Dir(imagePath)) > 0 Then ' Check if the image file exists  
        Image1.Picture = LoadPicture(imagePath)
```

```

Else
    ' Clear the Image control if the image file does not exist
    Me.Image1.Picture = LoadPicture("E:\Eco\A.jpg") ' This will clear
the image control
End If

' Set the caption for the label
'Me.Label.Caption = labelCaption
End Sub

```

Here's an explanation of each part:

Dim imagePath As String: This line declares a variable named `imagePath` to store the file path of the image.

Dim labelCaption As String: This line declares a variable named `labelCaption` to store the caption for the label.

imagePath = "E:\Eco\Photo" & currentLabelNumber & ".jpg": This line constructs the file path of the image based on the `currentLabelNumber` variable. It assumes that the images are named "Photo1.jpg," "Photo2.jpg," and so on, and they are located in the "E:\Eco" directory. You can adjust the path and naming convention as needed.

labelCaption = "Caption for Label" & currentLabelNumber: This line sets the `labelCaption` variable to a caption based on the `currentLabelNumber`. You should replace this with your actual source of captions.

If Len(Dir(imagePath)) > 0 Then: This line checks if the image file specified by `imagePath` exists.

Image1.Picture = LoadPicture(imagePath): If the image file exists, this line loads and displays the image in the `Image1` control.

Me.Image1.Picture = LoadPicture("E:\Eco\A.jpg"): If the image file does not exist, this line clears the `Image1` control by loading a default image located at "E:\Eco\A.jpg." You can replace this with the path to your own default image.

Me.Label.Caption = labelCaption: This line is currently commented out (indicated by the single quote `'`). If you want to set the caption for a label on your form, you can uncomment this line and replace `Me.Label` with the specific label you want to update.

In summary, this subroutine constructs the image file path and caption based on `currentLabelNumber`, checks if the image file exists, and updates the `Image1` control with the image and optionally updates a label with the caption.

CmdNxt Click and CmdPrv Click Events:

These event handlers allow the user to navigate through a series of labels and images by incrementing or decrementing the `currentLabelNumber`. It also calls the `UpdateLabelAndImage` subroutine to update the label caption and display the corresponding image.

```
Private Sub CmdNxt_Click()  
    ' Check if the currentLabelNumber is less than 23  
    If currentLabelNumber < 23 Then ' Adjust the upper limit to the total  
number of labels  
        ' If it is, increment the currentLabelNumber by 1  
        currentLabelNumber = currentLabelNumber + 1  
    Else  
        ' If the currentLabelNumber is 23 or greater, wrap around to the  
first label (which has the number 4)  
        currentLabelNumber = 4  
    End If  
  
    ' Update the label caption and display the image  
    UpdateLabelAndImage  
End Sub
```

Here's an explanation of each part:

Private Sub CmdNxt_Click(): This line declares a private subroutine (a named block of code) that will run when the "CmdNxt" button is clicked.

If currentLabelNumber < 23 Then: This line starts an "if" statement. It checks if the value of the variable `currentLabelNumber` is less than 23.

' Adjust the upper limit to the total number of labels: This comment is a note to the developer indicating that they should change the upper limit to match the total number of labels they have. In this case, it's set to 23.

currentLabelNumber = currentLabelNumber + 1: If the condition in the "if" statement is true (i.e., `currentLabelNumber` is less than 23), this line increases the value of `currentLabelNumber` by 1.

Else: If the condition in the "if" statement is false (i.e., `currentLabelNumber` is not less than 23), the code inside the "else" block is executed.

currentLabelNumber = 4: In the "else" block, this line resets `currentLabelNumber` to 4, effectively wrapping around to the first label.

UpdateLabelAndImage: Finally, after either incrementing or resetting `currentLabelNumber`, this line calls another subroutine named `UpdateLabelAndImage`. This subroutine is responsible for updating the label's caption and displaying an image based on the new `currentLabelNumber`.

In summary, when the "CmdNxt" button is clicked, this code checks if `currentLabelNumber` is less than 23. If it is, it increments the number; otherwise, it resets it to 4 and then updates the label's caption and displays the corresponding image. This creates a simple image navigation feature where you can cycle through a series of images using the "Next" button.

UpdateLabelAndImage: Finally, after either incrementing or resetting `currentLabelNumber`, this line calls another subroutine named `UpdateLabelAndImage`. This subroutine is responsible for updating the label's caption and displaying an image based on the new `currentLabelNumber`.

```
Private Sub CmdPrv_Click()  
    ' Check if the currentLabelNumber is greater than 4  
    If currentLabelNumber > 4 Then  
        ' Decrease the currentLabelNumber by 1 to show the previous  
        label/image  
        currentLabelNumber = currentLabelNumber - 1  
    Else  
        ' If the currentLabelNumber is already at the lower limit (4 in  
        this case),  
        ' wrap around to the last label (assuming the last label number is  
        23).  
        currentLabelNumber = 23 ' Adjust this to the last label number in  
        your case.  
    End If  
  
    ' Update the label caption and display the image  
    UpdateLabelAndImage  
End Sub
```

Here's a breakdown of the code:

If `currentLabelNumber` > 4 Then: This line checks if the `currentLabelNumber` is greater than 4. This condition ensures that you don't go below the lower limit (4 in this case).

`currentLabelNumber` = `currentLabelNumber` - 1: If the condition is met, it decreases the `currentLabelNumber` by 1. This effectively moves to the previous label/image.

Else: If the condition is not met (meaning `currentLabelNumber` is already at the lower limit), this block of code is executed.

`currentLabelNumber` = 23: In this case, it wraps around to the last label number (23 in this example). You should adjust this number to match the actual last label number in your setup.

UpdateLabelAndImage: Finally, after adjusting the `currentLabelNumber`, the code calls the `UpdateLabelAndImage` subroutine to update the label caption and display the image corresponding to the new `currentLabelNumber`.

In summary, this subroutine allows you to navigate to the previous label/image by decrementing the currentLabelNumber, and it handles wrapping around to the last label if you reach the lower limit.

UserForm Initialize Event:

This event handler sets the initial value of currentLabelNumber and calls the byDefaultPicture subroutine to load and display a default image.

```
Private Sub UserForm_Initialize()  
    ' Set the initial label number to 4  
    currentLabelNumber = 4  
  
    ' Call the byDefaultPicture subroutine to set up default content  
    byDefaultPicture  
End Sub
```

Private Sub UserForm_Initialize(): This line defines a private subroutine called UserForm_Initialize. This subroutine is automatically executed when the user form is initialized.

currentLabelNumber = 4: This line sets the currentLabelNumber variable to an initial value of 4. It appears that currentLabelNumber is a variable used to keep track of the currently displayed label/image. By setting it to 4 initially, you are starting with the label/image labeled as "Label4" or the fourth label/image in your form.

byDefaultPicture: This line calls another subroutine named byDefaultPicture. This subroutine is responsible for setting up default content, which might include loading a default image and setting default captions or labels. The specifics of what this subroutine does would depend on the actual implementation in your code.

In summary, this code ensures that when the user form is initialized, the currentLabelNumber is set to 4 (indicating the initial label/image to display), and the byDefaultPicture subroutine is called to set up any default content for your form.

byDefaultPicture Subroutine:

This subroutine loads and displays a default image along with a multi-line caption in Label3.Caption.

```

Private Sub byDefaultPicture()
    ' Set the default image file path
    defaultImagePath = "E:\Eco\Default.jpg" ' Change to the path of your
    default image

    ' Load and display the default image in the Image control
    If Len(Dir(defaultImagePath)) > 0 Then ' Check if the default image
    file exists
        Image1.Picture = LoadPicture(defaultImagePath)
    Else
        ' Clear the Image control if the default image file does not exist
        Image1.Picture = LoadPicture("") ' This will clear the image
        control
    End If

    ' Set the caption for Label3
    Label3.Caption = "The Eco-Park in Action Area II of New Town is one of
    the largest open spaces in the city." & vbCrLf &
        "The site is spread across an area of 194 hectares
    (480 acres) and includes a 42-hectare (100 acres) water body with an
    island. The Central Business District (CBD) is located to the east of the
    site, and one of the largest convention centers in the country, Biswa
    Bangla Convention Centre, has been built to its north." & vbCrLf &
        "Photography: Gautam Banerjee"
End Sub

```

Here's what each part of the subroutine does:

defaultImagePath: This variable stores the file path to your default image. You've set it to "E:\Eco\Default.jpg". You can change this path to point to the location of your default image file.

If Len(Dir(defaultImagePath)) > 0 Then: This line checks if the default image file exists at the specified path. It uses the Dir function to check if a file with the given path exists.

Image1.Picture = LoadPicture(defaultImagePath): If the default image file exists, this line loads and displays the default image in the Image1 control on your user form using the LoadPicture function.

If the default image file does not exist, it clears the Image1 control by setting its Picture property to an empty string. This ensures that if the default image is missing, the image control will be empty.

Finally, it sets the caption for Label3. The Label3.Caption property is set to a multi-line text that provides information about the Eco-Park, including its size, location, and photography credits.

This subroutine essentially sets up the default appearance of your user form when it's initialized. It loads a default image, if available, and sets a default caption for one of your labels (Label3). You can modify the default image path and caption text to fit your specific requirements.

Overall Comments:

The code seems to be designed for a user-friendly interface to display images and associated captions based on user interactions with labels.

It handles navigation between different images and provides fallback mechanisms to display default images when necessary.

There are some hardcoded paths to image files, and the captions for the labels appear to be constructed programmatically or should be fetched from a data source not shown in this code.

You may need to adjust the file paths, label captions, and other details to match your specific use case.

Please note that the functionality and usability of this code depend on the specific requirements and the context in which it is used. You should ensure that the file paths, image files, and captions are correctly set up for your application.

हेलो फ्रेंड्स, आज पिक्चर व्यूअर का दूसरा अलग टाइप का फॉर्म बनाते हैं। इस वीडियो देखने के साथ पिछले वीडियो जरूर देखिये नहीं तो आप काफी कुछ मिस करेंगे। एक प्रोग्रामर के लिए बहुत सारा प्रोग्रामिंग तरीका आना चाहिए। अगर आप डाटा एंट्री या डाटा एनालिसिस प्रोग्राम बनाना चाहते हैं फिर भी इस Type of प्रोग्राम बनाना आप के लिए बेहद फायदेमंद साबित हो सकता है।

तो चलिए सबसे पहले शुरू करते हैं यूजर फॉर्म बनाना। आप इन्सर्ट से यूजर फॉर्म इन्सर्ट करें। ड्रैग करके फॉर्म को लम्बाई और चौड़ाई सेट करें। अब टूल बॉक्स से इमेज फ्रेम इन्सर्ट करें और ड्रैग करके बना लीजिये। उसके बाद बनाइये लेबल जो पिक्चर के टाइटल डिस्क्रिप्शन शो करेगा। अब कुछ लेबल बनाकर नंबर लिख दीजिये। मेनू तब में फॉर्मेट तब पर क्लिक करें और अलाइन, साइज और स्पेसिंग सेट कर दें। दो इमेज फ्रेम बनाइये, प्रॉपर्टीज में पिक्चर ऑप्शन से एरो सेलेक्ट करें। ध्यान रहे bmp ये jpg एक्सटेंशन होना चाहिए। आप चाहे तो लेबल बनाकर नेक्स्ट प्रीवियस भी लिख सकते हैं। चलिए अब कोडिंग लिखते हैं। सबसे पहले यूजर इनिशियलाइज़ लिखते हैं। इस का काम है जब फॉर्म ओपन होगा उस समय फॉर्म पर कोनसी आइटम दिखेगा। आप कृपा करके इससे पहला बाला वीडियो देख लें। बार बार एक ही बात करने से समय बर्बाद होगा।

यूजर इनिशियलाइज़ में मैंने एक पिक्चर को शो किया | मैंने अलग से एक procedure बनाये है "ByDefaultPicture" इस procedure में आपको पिक्चर फोल्डर का रिफरेन्स देना है और एक if लॉजिकल कमांड दे कर बताना है अगर पिक्चर नाम फोल्डर में है तो पिक्चर को शो करना है और अगर पिक्चर नहीं मिल पता है तो लूप से बहार निकल जाना है | यूजर इनिशियलाइज़ में इस procedure को कॉल करना है | इसके अलावा एक वेरिएबल बनाकर वैल्यू ४ बना दिया है | ४ वैल्यू लिखने कारन है फॉर्म के निचे जो १, २, ३, लेबल है उसमे पहला लेबल का नाम Label4 है इसलिए | बाद में आपको पता चलेगा इसका कारन |

अब लेबल क्लिक इवेंट लिखते है | Dim clickedLabel As Object में डिम का मतलब है declare क्लिकेडलेबल वेरिएबल नाम है और इस वेरिएबल में एक ऑब्जेक्ट स्टोर किया गया है | ऑब्जेक्ट का मतलब इस फॉर्म का लेबल controls | currentLabelNumber एक वेरिएबल है और उस वेरिएबल में label4 का sixth करैक्टर मतलब 4 को स्टोर किया गया है | Val function स्ट्रिंग (मतलब करैक्टर) को नंबर फॉर्मेट convert करेगा | i एक वेरिएबल है जिसका वैल्यू currentLabelNumber वैल्यू से ऐड होगा | जैसे label4 का sixth करैक्टर मतलब 4 के साथ i के वैल्यू ऐड होगा | ऊपर मैंने हर लेबल का एक वैल्यू स्टोर कर दिया और लेबल_क्लिक procedure को कॉल कर दिया है | मैंने एक पीडीऍफ़ फाइल बना दिया है आप उस पीडीऍफ़ फाइल को डाउनलोड कर लीजिये | उसमे पूरा कोडिंग है साथ ही बिस्तार से कोड के बारे में बताया गया है | लिंक डिस्क्रिप्शन में मिलेगा |

अब नेक्स्ट बटन क्लिक फंक्शन के कोड लिखता हु | इस में सिर्फ इतना बताना है ही currentLabelNumber के वैल्यू ४ से ज्यादा है तो हर क्लिक पर १ माइनस होता रहेगा | जैसे ही वैल्यू ४ हो जायेगा फिर से currentLabelNumber का वैल्यू २३ हो जायेगा | मेरे पास जो पिक्चर है उस पिक्चर का नाम Photo4 से लेकर Photo23 तक है | इसलिए मैंने वैल्यू ४ से २३ का इस्तेमाल किया है | जॉब भी मैं पिक्चर रिफरेन्स देता हूँ तो E:\Eco\Photo" & currentLabelNumber & ".jpg" मतलब E Drive Eco Folder में Photo + वेरिएबल का वैल्यू मेशन करता हूँ |

Same कमांड prv बटन के लिए भी लिखना है | सिर्फ minus 1 के जगह Plus 1 होगा | जैसे ही वैल्यू 23 हो जायेगा फिर से currentLabelNumber का वैल्यू 4 हो जायेगा |

अब UpdateLabelAndImage कोड लिखना है | next या Prev बटन पर क्लिक करने पर पिक्चर और लेबल update हो जायेगा | पहले एक इमेज और एक लेबल के लिए वेरिएबल बनाना है | उस वेरिएबल पर Picture path का रिफरेन्स देना है | उसके बाद एक लॉजिकल कमांड देने होंगे ये बताने के लिए पिक्चर मिलने पर पिक्चर मिलने पर पिक्चर को शो करना है और अगर न मिले तो लूप से बहार जाना है |

अब एक एग्जिट कमांड लिखना है फॉर्म को क्लोज करने के लिए जो है Unload me |

बस प्रोग्राम कम्पलीट |

कृपया ध्यान दें कि इस कोड की कार्यक्षमता और उपयोगिता विशिष्ट आवश्यकताओं और उस संदर्भ पर निर्भर करती है जिसमें इसका उपयोग किया जाता है। आपको यह सुनिश्चित करना चाहिए कि फ़ाइल पथ, छवि फ़ाइलें और कैप्शन आपके एप्लिकेशन के लिए सही ढंग से सेट किए गए हैं।

(Please note that the functionality and usability of this code depend on the specific requirements and the context in which it is used. You should ensure that the file paths, image files, and captions are correctly set up for your application.)

धन्यवाद |



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 Google Pay
9748327614**