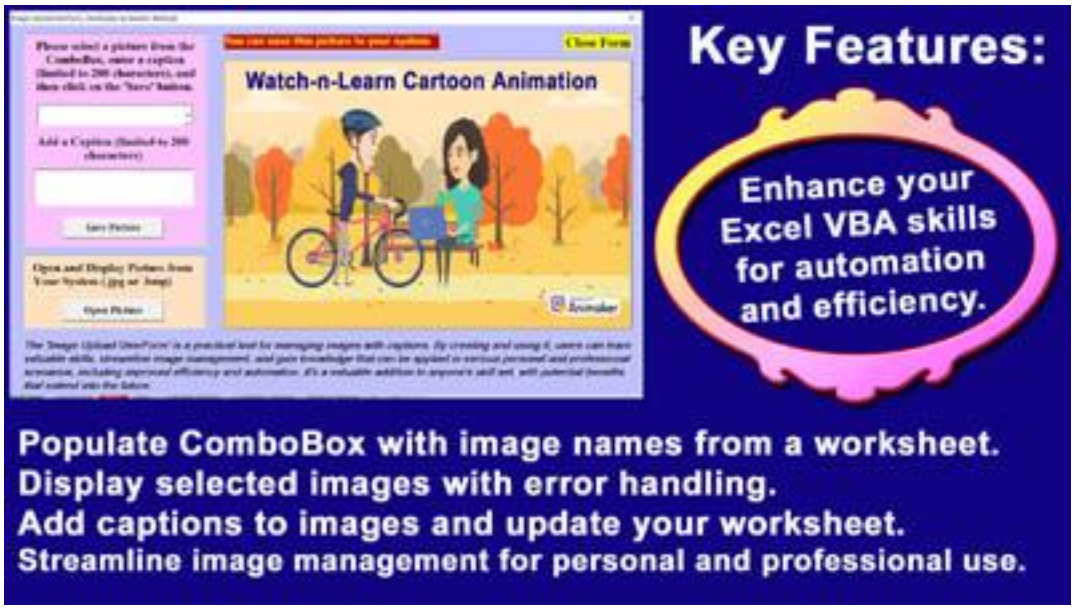

Video Link (4 This Video) : https://youtu.be/EVxw2KRQY-k

Video Link (3 Last Video) : https://youtu.be/j4voq5KbPSs

Video Link (2 Last Video) : https://youtu.be/xJ0_vbWUnEA

Video Link (1 Last Video) : https://youtu.be/YDWwF6FAIGg

Title: "**Excel VBA Image Management Project - Functionality and Purpose**"

Welcome to our Excel VBA Image Management Tool tutorial and demonstration! In this video, we'll walk you through the creation and functionality of an advanced Image Management UserForm using Excel VBA.

Description: "In this video, we delve into the functionalities and purpose of our Excel VBA Image Management Project. This project is designed to simplify image management within Excel, making it an invaluable tool for various personal and professional scenarios. Let's explore the key components and objectives of this project:

ComboBox Selection: The ComboBox1_Change() subroutine enables users to select different tourist spots from a ComboBox. When a selection is made, the project constructs the image file path based on the selected tourist spot's name and folder path.

Default Image: The byDefaultPicture() subroutine sets a default image, which is displayed when the form loads. This feature ensures a pleasant user experience and serves as a fallback when specific images are not available.

FileDialog Integration: The CommandButtonOpen_Click() subroutine allows users to open a file dialog and select an image. It checks for the existence and validity of the selected image file, displaying it if found. If the selected image is not valid, it provides a custom error message.

Adding Images: The CommandButtonAdd_Click() subroutine facilitates the addition of new image names and captions to an Excel worksheet. This functionality enables users to maintain an organized database of images and their associated descriptions.

Initialization: The UserForm_Initialize() subroutine initializes the FileDialog1 object, populates the ComboBox with image names from an Excel worksheet, and sets up the file dialog for selecting images. It also provides an introductory message about the project's purpose and benefits.

User-Friendly Interface: The project's user-friendly interface simplifies the process of selecting, viewing, and managing images, making it accessible to users with varying levels of Excel expertise.

Educational Value: By creating and using this Image Management UserForm, users can acquire valuable Excel VBA skills. This project demonstrates techniques for working with UserForms, file dialogs, dynamic image loading, and data entry into Excel worksheets.

Efficiency and Automation: The Image Management Project streamlines image-related tasks, offering enhanced efficiency and automation. This project can be adapted for various use cases, such as cataloging images, creating presentations, or managing visual content.

Join us in this video to explore the intricacies of this Excel VBA Image Management Project. Whether you're an Excel enthusiast looking to expand your skill set or a professional seeking to improve image management, this project holds the key to enhanced productivity and organization.

Don't forget to like, share, and subscribe for more tutorials and Excel VBA insights. Download the complete code and resources to get started on your own image management journey."

हमारे एक्सेल वीबीए इमेज मैनेजमेंट टूल ट्यूटोरियल और प्रदर्शन में आपका स्वागत है! इस वीडियो में, हम आपको एक्सेल वीबीए का उपयोग करके एक उन्नत छवि प्रबंधन यूजरफॉर्म के निर्माण और कार्यक्षमता के बारे में बताएंगे।

विवरण: "इस वीडियो में, हम अपने एक्सेल वीबीए इमेज मैनेजमेंट प्रोजेक्ट की कार्यक्षमता और उद्देश्य पर प्रकाश डालते हैं। यह प्रोजेक्ट एक्सेल के भीतर इमेज मैनेजमेंट को सरल बनाने के लिए डिज़ाइन किया गया है, जो इसे विभिन्न व्यक्तिगत और व्यावसायिक परिदृश्यों के लिए एक अमूल्य टूल बनाता है। आइए प्रमुख घटकों का पता लगाएं और इस परियोजना के उद्देश्य:

कॉम्बोबॉक्स चयन: कॉम्बोबॉक्स1_चेंज() सबरूटीन उपयोगकर्ताओं को कॉम्बोबॉक्स से विभिन्न पर्यटन स्थलों का चयन करने में सक्षम बनाता है। जब कोई चयन किया जाता है, तो प्रोजेक्ट चयनित पर्यटक स्थल के नाम और फ़ोल्डर पथ के आधार पर छवि फ़ाइल पथ का निर्माण करता है।

डिफ़ॉल्ट छवि: बायडिफॉल्टपिक्चर() सबरूटीन एक डिफ़ॉल्ट छवि सेट करता है, जो फॉर्म लोड होने पर प्रदर्शित होती है। यह सुविधा एक सुखद उपयोगकर्ता अनुभव सुनिश्चित करती है और विशिष्ट छवियां उपलब्ध नहीं होने पर फ़ॉलबैक के रूप में कार्य करती है।

FileDialog एकीकरण: CommandButtonOpen_Click() सबरूटीन उपयोगकर्ताओं को एक फ़ाइल संवाद खोलने और एक छवि का चयन करने की अनुमति देता है। यह चयनित छवि फ़ाइल के अस्तित्व और वैधता की जाँच करता है, यदि पाया जाता है तो उसे प्रदर्शित करता है। यदि चयनित छवि मान्य नहीं है, तो यह एक कस्टम त्रुटि संदेश प्रदान करती है।

छवियाँ जोड़ना: CommandButtonAdd_Click() सबरूटीन एक्सेल वर्कशीट में नई छवि नाम और कैप्शन जोड़ने की सुविधा प्रदान करता है। यह कार्यक्षमता उपयोगकर्ताओं को छवियों और उनके संबंधित विवरणों का एक व्यवस्थित डेटाबेस बनाए रखने में सक्षम बनाती है।

आरंभीकरण: UserForm_Initialize() सबरूटीन FileDialog1 ऑब्जेक्ट को आरंभ करता है, एक्सेल वर्कशीट से छवि नामों के साथ कॉम्बोबॉक्स को पॉप्युलेट करता है, और छवियों के चयन के लिए फ़ाइल संवाद सेट करता है। यह परियोजना के उद्देश्य और लाभों के बारे में एक परिचयात्मक संदेश भी प्रदान करता है।

उपयोगकर्ता के अनुकूल इंटरफ़ेस: प्रोजेक्ट का उपयोगकर्ता के अनुकूल इंटरफ़ेस छवियों को चुनने, देखने और प्रबंधित करने की प्रक्रिया को सरल बनाता है, जिससे यह एक्सेल विशेषज्ञता के विभिन्न स्तरों वाले उपयोगकर्ताओं के लिए सुलभ हो जाता है।

शैक्षिक मूल्य: इस छवि प्रबंधन यूजरफॉर्म को बनाकर और उपयोग करके, उपयोगकर्ता मूल्यवान एक्सेल वीबीए कौशल प्राप्त कर सकते हैं। यह प्रोजेक्ट यूजरफॉर्म, फ़ाइल संवाद, गतिशील छवि लोडिंग और एक्सेल वर्कशीट में डेटा प्रविष्टि के साथ काम करने की तकनीकों को प्रदर्शित करता है।

दक्षता और स्वचालन: छवि प्रबंधन परियोजना छवि से संबंधित कार्यों को सुव्यवस्थित करती है, बढ़ी हुई दक्षता और स्वचालन की पेशकश करती है। इस प्रोजेक्ट को विभिन्न उपयोग के मामलों के लिए अनुकूलित किया जा सकता है, जैसे छवियों को सूचीबद्ध करना, प्रस्तुतियाँ बनाना, या दृश्य सामग्री का प्रबंधन करना।

इस एक्सेल वीबीए इमेज मैनेजमेंट प्रोजेक्ट की जटिलताओं का पता लगाने के लिए इस वीडियो में हमसे जुड़ें। चाहे आप अपने कौशल सेट का विस्तार करने के इच्छुक एक्सेल उत्साही हों या छवि प्रबंधन में सुधार करने वाले पेशेवर हों, यह परियोजना बढ़ी हुई उत्पादकता और संगठन की कुंजी रखती है।

अधिक ट्यूटोरियल और एक्सेल वीबीए अंतर्दृष्टि के लिए लाइक, शेयर और सब्सक्राइब करना न भूलें। अपनी स्वयं की छवि प्रबंधन यात्रा शुरू करने के लिए संपूर्ण कोड और संसाधन डाउनलोड करें।"

Here's a step-by-step guide and a sample code to help you achieve this:

Step 1: Prepare Your Data

Before creating the UserForm, make sure you have an Excel sheet with the list of image names and their corresponding captions.

Step 2: Create the Image Upload UserForm

In Excel, press Alt + F11 to open the VBA editor.

Go to "Insert" > "UserForm" to create a new UserForm.

Add the following controls to your UserForm:

Image control to display the selected image.

ComboBox control to select existing images.

CommandButton control to open a file dialog for image selection.

TextBox control to enter a caption for the newly added image.

Another CommandButton control to add the selected image.

Save your workbook as a macro-enabled workbook (.xlsm) and close it. When you reopen it, your UserForm should appear automatically, allowing you to select, view, and add images to your collection.

This enhanced Picture Viewer now includes the ability to upload new images with captions. Users can open a file dialog, select an image, enter a caption, and add it to the collection.

Ensure Proper Reference: First, ensure that you have the proper reference to the "Microsoft Office xx.x Object Library" in your VBA project, where "xx.x" represents your Office version (e.g., 16.0 for Office 2016). To check this, open the VBA editor, go to "Tools" > "References," and make sure that the library is selected.

In my case, I have Microsoft Office 15.0 office (Excel, Access and Word) Library.

चरण 1: अपना डेटा तैयार करें

यूजरफॉर्म बनाने से पहले, सुनिश्चित करें कि आपके पास छवि नामों और उनके संबंधित कैप्शन की सूची वाली एक एक्सेल शीट है।

चरण 2: इमेज अपलोड यूजरफॉर्म बनाएं

1. Excel में, VBA Editor खोलने के लिए Alt + F11 दबाएँ।

2. नया यूजरफॉर्म बनाने के लिए "इन्सर्ट" > "यूजरफॉर्म" पर जाएं।

3. अपने यूजरफॉर्म में निम्नलिखित नियंत्रण जोड़ें:

• चयनित छवि प्रदर्शित करने के लिए छवि नियंत्रण।

• मौजूदा छवियों का चयन करने के लिए कॉम्बोबॉक्स नियंत्रण ।

• छवि चयन के लिए फ़ाइल संवाद खोलने के लिए कमांडबटन नियंत्रण।

• नई जोड़ी गई छवि के लिए कैप्शन दर्ज करने के लिए टेक्स्टबॉक्स नियंत्रण।

• चयनित छवि को जोड़ने के लिए एक अन्य कमांडबटन नियंत्रण।

अपनी कार्यपुस्तिका को मैक्रो-सक्षम कार्यपुस्तिका (.xlsm) के रूप में Save करें और इसे बंद करें। जब आप इसे दोबारा खोलते हैं, तो आपका यूजरफॉर्म स्वचालित रूप से दिखाई देना चाहिए, जिससे आप अपने संग्रह में छवियों को चुन सकते हैं, देख सकते हैं और जोड़ सकते हैं।

इस उन्नत पिक्चर व्यूअर में अब कैप्शन के साथ नई छवियां अपलोड करने की क्षमता शामिल है। उपयोगकर्ता एक फ़ाइल संवाद खोल सकते हैं, एक छवि का चयन कर सकते हैं, एक कैप्शन दर्ज कर सकते हैं और इसे संग्रह में जोड़ सकते हैं।

उचित संदर्भ सुनिश्चित करें: सबसे पहले, सुनिश्चित करें कि आपके पास अपने VBA प्रोजेक्ट में "Microsoft Office xx.x ऑब्जेक्ट लाइब्रेरी" का उचित संदर्भ है, जहां "xx.x" आपके Office संस्करण का प्रतिनिधित्व करता है (उदाहरण के लिए, Office 2016 के लिए 16.0)। इसे जाँचने के लिए, VBA संपादक खोलें, "टूल्स" > "संदर्भ" पर जाएँ और सुनिश्चित करें कि लाइब्रेरी चयनित है।

मेरे मामले में, मेरे पास माइक्रोसॉफ्ट ऑफिस 15.0 ऑफिस (एक्सेल, एक्सेस और वर्ड) लाइब्रेरी है ।

Copy the Below Full Code :

```
Dim FileDialog1 As FileDialog ' Declare the FileDialog1 object at the module level



Private Sub ComboBox1_Change()
```

```
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("PictureData") ' Replace with your actual worksheet name

    ' Check if an item is selected in the ListBox

    If ComboBox1.ListIndex >= 0 Then
        ' Get the selected item (tourist spot) from the ListBox
        Dim SelectedPicture As String
        SelectedPicture = ComboBox1.List(ComboBox1.ListIndex)
        ' Construct the image file path based on the selected tourist spot name and folder path

        imagePath = "E:\MyPicture\" & SelectedPicture & ".jpg" ' Assuming image file extension is .jpg

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

    End If
End Sub
```

```
Private Sub Label5_Click()
    Unload Me
    MainMenuForm.Show
End Sub

Private Sub UserForm_Initialize()

    ' Initialize the FileDialog1 object
    Set FileDialog1 = Application.FileDialog(msoFileDialogFilePicker)

   ' Populate the ComboBox with image names from your worksheet
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("ImageData") ' Replace with your sheet name

    Dim cell As Range
    For Each cell In ws.Range("A2:A" & ws.Cells(ws.Rows.Count, "A").End(xlUp).row)
        ComboBox1.AddItem cell.value
    Next cell

    ' Initialize the file dialog
    With FileDialog1
        .title = "Select an Image"
        .Filters.Clear
        .Filters.Add "Image Files", "*.jpg; *.jpeg; *.png; *.gif; *.bmp"
    End With
```

```
    byDefaultPicture
    Label6.caption = "The 'Image Upload UserForm' is a practical tool for managing images with captions. By creating and using it, users can learn valuable skills, streamline image management, and gain knowledge that can be applied in various personal and professional scenarios, including improved efficiency and automation. It's a valuable addition to anyone's skill set, with potential benefits that extend into the future."

End Sub

Private Sub byDefaultPicture()

 ' Set the default image file path
    defaultImagePath = "C:\Users\user\Desktop\Cartoon1.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

End Sub

Private Sub CommandButtonOpen_Click()
    ' Show the file dialog and allow the user to select an image
```

```
    If FileDialog1.Show = -1 Then
        ' Get the selected file path
        Dim selectedImagePath As String
        selectedImagePath = FileDialog1.SelectedItems(1)

        On Error Resume Next

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

            ' Display a custom error message
            MsgBox "Image not found at the specified location or invalid file format." & _
            vbCrLf & "Please select a valid image file.", vbExclamation, "Error - Gautam Banerjee"

        End If

        On Error GoTo 0
    End If
End Sub

Private Sub CommandButtonAdd_Click()
```

```
' Get the selected image name and caption
Dim selectedImageName As String
selectedImageName = ComboBox1.value
Dim caption As String
caption = TextBoxCaption.Text

' Validate input
If selectedImageName = "" Or caption = "" Then
    MsgBox "Please select an image and enter a caption.", vbExclamation
    Exit Sub
End If

' Update your worksheet (replace "YourSheetName" with your sheet name)
Dim ws As Worksheet
Set ws = ThisWorkbook.Sheets("NewSheet")
Dim newRow As Long
newRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).row + 1

' Add the new image name and caption to the worksheet
ws.Cells(newRow, "A").value = selectedImageName
ws.Cells(newRow, "B").value = caption

' Inform the user that the image has been added
MsgBox "Image added successfully!", vbInformation

' Clear the TextBox and refresh the ComboBox
```

```
    TextBoxCaption.Text = ""
    ComboBox1.AddItem selectedImageName
End Sub
```

Let’s Describe the Line by Line Coding. Don’t copy the below mentioned Code. It is using only for describing the code function / meaning purposes. UserForm in VBA and is automatically executed when the UserForm is initialized or loaded. Its primary purpose is to prepare the UserForm and its controls for user interaction. Here's a breakdown of what this code does:

Dim FileDialog1 As FileDialog ' Declare the FileDialog1 object at the module level

(Outside the procedure, Top of the Code Window)

Declaring FileDialog1 at the module level means it can be accessed by any procedure or function within the module, making it a global object in that module. By declaring FileDialog1 at the module level, it's not limited to just the UserForm_Initialize procedure. Other procedures within the same module can also use FileDialog1 without needing to declare it again. This can be useful if you plan to use FileDialog1 in multiple procedures within the same module.

आइए कोडिंग का वर्णन करें। नीचे उल्लिखित कोड की नकल न करें। इसका उपयोग केवल कोड फ़ंक्शन/अर्थ उद्देश्यों का वर्णन करने के लिए किया जा रहा है। VBA में UserForm और UserForm आरंभ या लोड होने पर स्वचालित रूप से निष्पादित होता है। इसका प्राथमिक उद्देश्य यूजर इंटरैक्शन के लिए यूजरफॉर्म और उसके नियंत्रण तैयार करना है। यह कोड क्या करता है इसका विवरण यहां दिया गया है:

Dim FileDialog1 को FileDialog के रूप में 'मॉड्यूल स्तर पर FileDialog1 ऑब्जेक्ट को घोषित करें

(प्रक्रिया के बाहर, कोड विंडो के शीर्ष पर)

मॉड्यूल स्तर पर FileDialog1 घोषित करने का मतलब है कि इसे मॉड्यूल के भीतर किसी भी प्रक्रिया या फ़ंक्शन द्वारा एक्सेस किया जा सकता है, जिससे यह उस मॉड्यूल में एक वैश्विक ऑब्जेक्ट बन जाता है। मॉड्यूल स्तर पर FileDialog1 घोषित करके, यह केवल UserForm_Initialize प्रक्रिया तक सीमित नहीं है। समान मॉड्यूल के भीतर अन्य प्रक्रियाएं भी FileDialog1 को दोबारा घोषित करने की आवश्यकता के बिना उपयोग कर सकती हैं। यदि आप एक ही मॉड्यूल के भीतर कई प्रक्रियाओं में FileDialog1 का उपयोग करने की योजना बना रहे हैं तो यह उपयोगी हो सकता है।

```
Private Sub UserForm_Initialize()

    ' Initialize the FileDialog1 object
    Set FileDialog1 = Application.FileDialog(msoFileDialogFilePicker)

   ' Populate the ComboBox with image names from your worksheet
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("ImageData") ' Replace with your sheet name

    Dim cell As Range
    For Each cell In ws.Range("A2:A" & ws.Cells(ws.Rows.Count, "A").End(xlUp).row)
        ComboBox1.AddItem cell.value
    Next cell

    ' Initialize the file dialog
    With FileDialog1
        .title = "Select an Image"
        .Filters.Clear
        .Filters.Add "Image Files", "*.jpg; *.jpeg; *.png; *.gif; *.bmp"
    End With

    byDefaultPicture
    Label6.caption = "The 'Image Upload UserForm' is a …"

End Sub
```

Set FileDialog1 = Application.FileDialog(msoFileDialogFilePicker)

This code initializes the **FileDialog1** object, which is used to create a file picker dialog for selecting images. It uses the **Application.FileDialog** method and sets it to the **msoFileDialogFilePicker**

type. This line initializes the **FileDialog1** object. It's used to create a file picker dialog, allowing users to select image files.

Dim ws As Worksheet: This line declares a variable ws to represent a worksheet within the Excel workbook.

Set ws = ThisWorkbook.Sheets("ImageData"): Here, the ws variable is assigned to reference a specific worksheet named "ImageData." You should replace "ImageData" with the name of the worksheet where your image data is stored.

Dim cell As Range: This line declares a cell variable as a Range object. It's used to loop through cells in a specific column of the worksheet.

For Each cell In ws.Range("A2:A" & ws.Cells(ws.Rows.Count, "A").End(xlUp).Row): This loop goes through each cell in column A of the "ImageData" worksheet, starting from cell A2 and ending at the last used cell in column A. It's used to populate a ComboBox (likely named ComboBox1) with image names stored in this column.

ComboBox1.AddItem cell.Value: Inside the loop, each cell's value (image name) is added to the ComboBox. This populates the ComboBox with image names.

With FileDialog1: This block of code initializes the properties of the FileDialog1 object. It sets the title of the file dialog to "Select an Image" and defines a filter to display only image files with specific extensions (e.g., .jpg, .jpeg, .png, .gif, .bmp). This restricts users to selecting image files only.

byDefaultPicture: This line calls another subroutine named byDefaultPicture. It's likely used to set a default image in an Image control on the UserForm.

Label6.Caption = "The 'Image Upload UserForm'…": Here, the caption (text) of Label6 is set to describe the UserForm's purpose. This label is typically used to provide information or instructions to users.

In summary, this code prepares the UserForm by initializing a file picker dialog, populating a ComboBox with image names from a specific worksheet, setting up filters for image files, and providing some default content and descriptions to guide the user.

FileDialog1 = एप्लिकेशन.FileDialog(msoFileDialogFilePicker) सेट करें

यह कोड FileDialog1 ऑब्जेक्ट को प्रारंभ करता है, जिसका उपयोग छवियों को चुनने के लिए फ़ाइल पिकर संवाद बनाने के लिए किया जाता है। यह एप्लिकेशन.फाइलडायलॉग विधि का उपयोग करता है और इसे msoFileDialogFilePicker प्रकार पर सेट करता है। यह पंक्ति FileDialog1 ऑब्जेक्ट को प्रारंभ करती है। इसका उपयोग फ़ाइल पिकर संवाद बनाने के लिए किया जाता है, जो उपयोगकर्ताओं को छवि फ़ाइलों का चयन करने की अनुमति देता है।

वर्कशीट के रूप में डिम डब्ल्यूएस: यह लाइन एक्सेल वर्कबुक के भीतर वर्कशीट का प्रतिनिधित्व करने के लिए एक वेरिएबल डब्ल्यूएस घोषित करती है।

सेट ws = ThisWorkbook.Sheets("ImageData"): यहां, ws वेरिएबल को "ImageData" नामक एक विशिष्ट वर्कशीट को संदर्भित करने के लिए असाइन किया गया है। आपको "इमेजडेटा" को उस वर्कशीट के नाम से बदलना चाहिए जहां आपका छवि डेटा संग्रहीत है।

रेंज के रूप में डिम सेल: यह लाइन एक सेल वेरिएबल को रेंज ऑब्जेक्ट के रूप में घोषित करती है। इसका उपयोग वर्कशीट के एक विशिष्ट कॉलम में कोशिकाओं के माध्यम से लूप करने के लिए किया जाता है।

प्रत्येक सेल के लिए ws.Range("A2:A" & ws.Cells(ws.Rows.Count, "A").End(xlUp).Row): यह लूप "ImageData" के कॉलम A में प्रत्येक सेल से होकर गुजरता है "वर्कशीट, सेल A2 से शुरू होकर कॉलम A में अंतिम उपयोग किए गए सेल पर समाप्त होती है। इसका उपयोग इस कॉलम में संग्रहीत छवि नामों के साथ एक कॉम्बोबॉक्स (संभवतः कॉम्बोबॉक्स1 नाम दिया गया) को पॉप्युलेट करने के लिए किया जाता है।

ComboBox1.AddItem cell.Value: लूप के अंदर, प्रत्येक सेल का मान (छवि नाम) कॉम्बोबॉक्स में जोड़ा जाता है। यह कॉम्बोबॉक्स को छवि नामों से भर देता है।

FileDialog1 के साथ: कोड का यह ब्लॉक FileDialog1 ऑब्जेक्ट के गुणों को प्रारंभ करता है। यह फ़ाइल संवाद का शीर्षक "एक छवि चुनें" पर सेट करता है और केवल विशिष्ट एक्सटेंशन (जैसे, .jpg, .jpeg, .png, .gif, .bmp) वाली छवि फ़ाइलों को प्रदर्शित करने के लिए एक फ़िल्टर को परिभाषित करता है। यह उपयोगकर्ताओं को केवल छवि फ़ाइलों का चयन करने तक ही सीमित रखता है।

बायडिफॉल्टपिक्चर: यह लाइन बायडिफॉल्टपिक्चर नामक एक अन्य सबरूटीन को कॉल करती है। इसका उपयोग संभवतः UserForm पर छवि नियंत्रण में एक डिफ़ॉल्ट छवि सेट करने के लिए किया जाता है।

लेबल6.कैप्शन = "'इमेज अपलोड यूजरफॉर्म'...": यहां, लेबल6 का कैप्शन (टेक्स्ट) यूजरफॉर्म के उद्देश्य का वर्णन करने के लिए सेट किया गया है। इस लेबल का उपयोग आमतौर पर उपयोगकर्ताओं को जानकारी या निर्देश प्रदान करने के लिए किया जाता है।

संक्षेप में, यह कोड एक फ़ाइल पिकर डायलॉग को आरंभ करके, एक विशिष्ट वर्कशीट से छवि नामों के साथ एक कॉम्बोबॉक्स को पॉप्युलेट करके, छवि फ़ाइलों के लिए फ़िल्टर सेट करके, और उपयोगकर्ता का मार्गदर्शन करने के लिए कुछ डिफ़ॉल्ट सामग्री और विवरण प्रदान करके यूजरफॉर्म तैयार करता है।

```
Private Sub byDefaultPicture()


 ' Set the default image file path

   defaultImagePath = "C:\Users\user\Desktop\Cartoon1.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

End Sub
```

This byDefaultPicture subroutine sets a default image to be displayed in an Image control when the userform is initialized. Here's a line-by-line explanation of what this code does:

This line declares a subroutine named byDefaultPicture. Subroutines are blocks of code that can be executed when called.

This section sets the path to the default image file. In this case, the default image file is located at "C:\Users\user\Desktop\Cartoon1.jpg". You can change this path to point to your own default image.

This part of the code performs the following actions:

It checks if the file specified in defaultImagePath exists using Len(Dir(defaultImagePath)) > 0. If the file exists, it proceeds to load and display it in an Image control named Image1 using Image1.Picture = LoadPicture(defaultImagePath).

If the file doesn't exist, it clears the Image1 control by setting it to a blank image using Image1.Picture = LoadPicture(""). This ensures that the Image control is empty when there is no default image available.

In summary, the byDefaultPicture subroutine sets a default image to be displayed in the Image1 control when the userform is initialized. If the default image file exists, it loads and displays it; otherwise, it leaves the Image control empty. This is a helpful feature to provide users with an initial image when they open the userform.

यह बायडिफॉल्टपिक्चर सबरूटीन उपयोगकर्ता प्रपत्र प्रारंभ होने पर छवि नियंत्रण में प्रदर्शित होने के लिए एक डिफ़ॉल्ट छवि सेट करता है। यह कोड क्या करता है इसकी पंक्ति-दर-पंक्ति व्याख्या यहां दी गई है:

यह पंक्ति byDefaultPicture नामक एक सबरूटीन घोषित करती है। सबरूटीन्स कोड के ब्लॉक होते हैं जिन्हें कॉल करने पर निष्पादित किया जा सकता है।

यह अनुभाग डिफ़ॉल्ट छवि फ़ाइल के लिए पथ सेट करता है। इस मामले में, डिफ़ॉल्ट छवि फ़ाइल "C:\Users\user\Desktop\Cartoon1.jpg" पर स्थित है। आप अपनी डिफ़ॉल्ट छवि को इंगित करने के लिए इस पथ को बदल सकते हैं।

कोड का यह भाग निम्नलिखित क्रियाएं करता है:

यह लेन(डीर(डिफॉल्टइमेजपाथ)) > 0 का उपयोग करके जांच करता है कि डिफ़ॉल्टइमेजपाथ में निर्दिष्ट फ़ाइल मौजूद है या नहीं। यदि फ़ाइल मौजूद है, तो यह Image1.Picture = LoadPicture(defaultImagePath) का उपयोग करके Image1 नामक छवि नियंत्रण में इसे लोड करने और प्रदर्शित करने के लिए आगे बढ़ता है।

यदि फ़ाइल मौजूद नहीं है, तो यह Image1.Picture = LoadPicture("") का उपयोग करके इसे एक रिक्त छवि पर सेट करके Image1 नियंत्रण को साफ़ करता है। यह सुनिश्चित करता है कि जब कोई डिफ़ॉल्ट छवि उपलब्ध न हो तो छवि नियंत्रण खाली है।

संक्षेप में, बायडिफॉल्टपिक्चर सबरूटीन यूजरफॉर्म आरंभ होने पर इमेज1 नियंत्रण में प्रदर्शित होने के लिए एक डिफ़ॉल्ट छवि सेट करता है। यदि डिफ़ॉल्ट छवि फ़ाइल मौजूद है, तो यह उसे लोड और प्रदर्शित करती है; अन्यथा, यह छवि नियंत्रण को खाली छोड़ देता है। जब उपयोगकर्ता उपयोगकर्ता प्रपत्र खोलते हैं तो उन्हें प्रारंभिक छवि प्रदान करने के लिए यह एक उपयोगी सुविधा है।

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 Google Pay

9748327614

```
Private Sub ComboBox1_Change()
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("PictureData") ' Replace with your actual worksheet name

    ' Check if an item is selected in the ListBox

    If ComboBox1.ListIndex >= 0 Then
```

```
    ' Get the selected item (tourist spot) from the ListBox
    Dim SelectedPicture As String
    SelectedPicture = ComboBox1.List(ComboBox1.ListIndex)
    ' Construct the image file path based on the selected tourist spot name and folder path

    imagePath = "E:\MyPicture\" & SelectedPicture & ".jpg" ' Assuming image file extension is .jpg

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

  End If
End Sub
```

Here's a description of the ComboBox1_Change subroutine:

Private Sub ComboBox1_Change()

This line declares a subroutine named ComboBox1_Change. Subroutines are blocks of code that are executed when an event occurs, in this case, when the value of ComboBox1 changes.

Dim ws As Worksheet Set ws = ThisWorkbook.Sheets("PictureData") ' Replace with your actual worksheet name

Here, a reference to a worksheet named "PictureData" is created and assigned to the variable ws. This worksheet is the source of data for the ComboBox and images.

' Check if an item is selected in the ListBox If ComboBox1.ListIndex >= 0 Then

This section of the code checks whether an item is selected in the ComboBox. It does this by verifying that the ListIndex property of the ComboBox is greater than or equal to 0, which indicates that an item is selected.

' Get the selected item (tourist spot) from the ListBox

Dim SelectedPicture As String

SelectedPicture = ComboBox1.List(ComboBox1.ListIndex)

Here, the selected item from the ComboBox is retrieved and stored in the variable SelectedPicture. This item represents the name of a tourist spot.

' Construct the image file path based on the selected tourist spot name and folder path

imagePath = "E:\MyPicture\" & SelectedPicture & ".jpg" ' Assuming image file extension is .jpg

This part of the code constructs the file path for the image associated with the selected tourist spot. It combines a folder path ("E:\MyPicture") with the selected tourist spot's name and adds the ".jpg" file extension.

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

The final part of the code loads and displays the image in an Image control named Image1. It checks if the image file specified by imagePath exists using **Len(Dir(imagePath)) > 0**. If the file exists, it loads and displays the image; otherwise, it clears the Image control to show no image.

In summary, the ComboBox1_Change subroutine responds to changes in the ComboBox's selected item. When a new item is selected, it constructs a file path for the corresponding image and displays it in the Image control. If no item is selected or the image file doesn't exist, it clears the Image control.

यहां ComboBox1_Change सबरूटीन का विवरण दिया गया है:

निजी उप कॉम्बोबॉक्स1_चेंज()

यह पंक्ति ComboBox1_Change नामक एक सबरूटीन घोषित करती है। सबरूटीन्स कोड के ब्लॉक होते हैं जो किसी घटना के घटित होने पर निष्पादित होते हैं, इस मामले में, जब कॉम्बोबॉक्स1 का मान बदलता है।

वर्कशीट सेट के रूप में मंद ws = ThisWorkbook.Sheets("PictureData") ' अपने वास्तविक वर्कशीट नाम से बदलें

यहां, "पिक्चरडेटा" नामक वर्कशीट का एक संदर्भ बनाया गया है और वेरिएबल ws को सौंपा गया है। यह वर्कशीट कॉम्बोबॉक्स और छवियों के लिए डेटा का स्रोत है।

' जांचें कि क्या कोई आइटम सूची बॉक्स में चुना गया है यदि ComboBox1.ListIndex >= 0 फिर

कोड का यह अनुभाग जाँचता है कि कॉम्बोबॉक्स में कोई आइटम चुना गया है या नहीं। यह यह सत्यापित करके करता है कि कॉम्बोबॉक्स की लिस्टइंडेक्स प्रॉपर्टी 0 से अधिक या उसके बराबर है, जो इंगित करती है कि एक आइटम चुना गया है।

'लिस्टबॉक्स से चयनित वस्तु (पर्यटन स्थल) प्राप्त करें

स्ट्रिंग के रूप में मंद चयनित चित्र

चयनित चित्र = कॉम्बोबॉक्स1.सूची(कॉम्बोबॉक्स1.लिस्टइंडेक्स)

यहां, कॉम्बोबॉक्स से चयनित आइटम को पुनर्प्राप्त किया जाता है और वेरिएबल चयनित चित्र में संग्रहीत किया जाता है। यह आइटम एक पर्यटक स्थल के नाम का प्रतिनिधित्व करता है।

'चयनित पर्यटन स्थल के नाम और फ़ोल्डर पथ के आधार पर छवि फ़ाइल पथ का निर्माण करें

imagePath = "E:\MyPicture\" और चयनितPicture और ".jpg" 'मान लें कि छवि फ़ाइल एक्सटेंशन .jpg है

कोड का यह भाग चयनित पर्यटन स्थल से जुड़ी छवि के लिए फ़ाइल पथ बनाता है। यह चयनित पर्यटक स्थल के नाम के साथ एक फ़ोल्डर पथ ("E:\MyPicture") को जोड़ता है और ".jpg" फ़ाइल एक्सटेंशन जोड़ता है।

' छवि को छवि नियंत्रण में लोड करें और प्रदर्शित करें यदि लेन (Dir (imagePath))> 0 फिर 'जांचें कि क्या छवि फ़ाइल मौजूद है Image1.Picture = LoadPicture (imagePath) अन्यथा' यदि छवि फ़ाइल मौजूद नहीं है तो छवि नियंत्रण साफ़ करें Image1 .Picture = LoadPicture('') ' इससे छवि नियंत्रण समाप्त हो जाएगा

कोड का अंतिम भाग छवि को Image1 नामक छवि नियंत्रण में लोड और प्रदर्शित करता है। यह जांच करता है कि इमेजपाथ द्वारा निर्दिष्ट छवि फ़ाइल लेन (डीर (इमेजपाथ))> 0 का उपयोग करके मौजूद है या नहीं। यदि फ़ाइल मौजूद है, तो यह छवि को लोड और प्रदर्शित करती है; अन्यथा, यह कोई छवि न दिखाने के लिए छवि नियंत्रण को साफ़ कर देता है।

संक्षेप में, ComboBox1_Change सबरूटीन, ComboBox के चयनित आइटम में परिवर्तन पर प्रतिक्रिया करता है। जब कोई नया आइटम चुना जाता है, तो यह संबंधित छवि के लिए एक फ़ाइल पथ बनाता है और इसे छवि नियंत्रण में प्रदर्शित करता है। यदि कोई आइटम चयनित नहीं है या छवि फ़ाइल मौजूद नहीं है, तो यह छवि नियंत्रण को साफ़ कर देता है।

```
Private Sub CommandButtonOpen_Click()

    ' Show the file dialog and allow the user to select an image
```

```
    If FileDialog1.Show = -1 Then
        ' Get the selected file path
        Dim selectedImagePath As String
        selectedImagePath = FileDialog1.SelectedItems(1)

        On Error Resume Next

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

            ' Display a custom error message
            MsgBox "Image not found at the specified location or invalid file format." & _
            vbCrLf & "Please select a valid image file.", vbExclamation, "Error - Gautam Banerjee"

        End If

        On Error GoTo 0
    End If
End Sub
```

Here's a description of the CommandButtonOpen_Click subroutine:

Private Sub CommandButtonOpen_Click()

This line declares a subroutine named CommandButtonOpen_Click. Subroutines are blocks of code that are executed when a specific event occurs, in this case, when the user clicks the "Open" command button on the user form.

' Show the file dialog and allow the user to select an image If FileDialog1.Show = -1 Then

This section of the code opens a file dialog box that allows the user to select an image file. It does this by calling the Show method of the FileDialog1 object. If the user selects a file and clicks "Open," the method returns -1, indicating that a file was selected.

' Get the selected file path **Dim selectedImagePath As String**

selectedImagePath = FileDialog1.SelectedItems(1)

Here, the code retrieves the file path of the selected image and stores it in the variable **selectedImagePath**. This path is essential for loading and displaying the image.

On Error Resume Next

This line of code temporarily instructs VBA to continue executing code even if an error occurs, effectively disabling error notifications. It's often used when dealing with situations where you expect errors, such as file operations.

' Load and display the selected image **If Len(Dir(selectedImagePath)) > 0 Then**

This part of the code checks whether the selected image file exists. It uses the Dir function to determine if the file specified by selectedImagePath exists. If the length of the result (returned by Len) is greater than 0, it means the file exists.

Image1.Picture = LoadPicture(selectedImagePath)

If the selected image file exists, this line loads and displays the image in an Image control named Image1. The LoadPicture function is used to load the image.

Else

If the selected image file does not exist (the condition in the previous If statement is not met), the code proceeds to this section.

' Clear the Image control if the image file does not exist **Image1.Picture = LoadPicture("")** ' This will clear the image control

Here, the code clears the Image1 control by loading an empty picture, effectively removing any previously displayed image.

' Display a custom error message **MsgBox "Image not found at the specified location or invalid file format." & _ vbCrLf & "Please select a valid image file.", vbExclamation, "Error - Gautam Banerjee"**

In this final part, the code displays a custom error message using the MsgBox function. This message informs the user that the selected image file was not found or is in an invalid format. It provides guidance to select a valid image file.

On Error GoTo 0

Finally, this line resets the error handling to its default state, allowing VBA to resume normal error notifications.

In summary, the CommandButtonOpen_Click subroutine is responsible for allowing the user to select an image file through a file dialog, checking if the file exists, and displaying it in the user form. If the file does not exist, it clears the image control and shows an error message to the user.

यहां CommandButtonOpen_Click सबरूटीन का विवरण दिया गया है:

प्राइवेट सब कमांडबटनओपन_क्लिक()

यह पंक्ति CommandButtonOpen_Click नामक एक सबरूटीन घोषित करती है। सबरूटीन्स कोड के ब्लॉक होते हैं जो किसी विशिष्ट घटना के घटित होने पर निष्पादित होते हैं, इस मामले में, जब उपयोगकर्ता उपयोगकर्ता फॉर्म पर "ओपन" कमांड बटन पर क्लिक करता है।

'फ़ाइल संवाद दिखाएं और उपयोगकर्ता को एक छवि चुनने की अनुमति दें यदि FileDialog1.Show = -1 तो

कोड का यह अनुभाग एक फ़ाइल संवाद बॉक्स खोलता है जो उपयोगकर्ता को एक छवि फ़ाइल का चयन करने की अनुमति देता है। यह FileDialog1 ऑब्जेक्ट की शो विधि को कॉल करके ऐसा करता है। यदि उपयोगकर्ता किसी फ़ाइल का चयन करता है और "खोलें" पर क्लिक करता है, तो विधि -1 लौटाती है, जो दर्शाती है कि फ़ाइल का चयन किया गया था।

'चयनित फ़ाइल पथ प्राप्त करें स्ट्रिंग के रूप में मंद चयनितइमेजपाथ

चयनितइमेजपाथ = FileDialog1.SelectedItems(1)

यहां, कोड चयनित छवि के फ़ाइल पथ को पुनर्प्राप्त करता है और इसे वेरिएबल चयनितइमेजपाथ में संग्रहीत करता है। छवि को लोड करने और प्रदर्शित करने के लिए यह पथ आवश्यक है।

त्रुटि पर अगला पुनरारंभ करें

कोड की यह पंक्ति अस्थायी रूप से VBA को त्रुटि होने पर भी कोड निष्पादित करना जारी रखने का निर्देश देती है, जिससे त्रुटि सूचनाएं प्रभावी रूप से अक्षम हो जाती हैं। इसका उपयोग अक्सर उन स्थितियों से निपटने के लिए किया जाता है जहां आप त्रुटियों की उम्मीद करते हैं, जैसे फ़ाइल संचालन।

'चयनित छवि को लोड करें और प्रदर्शित करें यदि लेन(Dir(selectedImagePath)) > 0 फिर

कोड का यह भाग जाँचता है कि चयनित छवि फ़ाइल मौजूद है या नहीं। यह निर्धारित करने के लिए Dir फ़ंक्शन का उपयोग करता है कि चयनितImagePath द्वारा निर्दिष्ट फ़ाइल मौजूद है या नहीं। यदि परिणाम की लंबाई (लेन द्वारा लौटाया गया) 0 से अधिक है, तो इसका मतलब है कि फ़ाइल मौजूद है।

छवि1.चित्र = लोडचित्र(चयनितइमेजपाथ)

यदि चयनित छवि फ़ाइल मौजूद है, तो यह पंक्ति छवि को Image1 नामक छवि नियंत्रण में लोड और प्रदर्शित करती है। लोडपिक्चर फ़ंक्शन का उपयोग छवि को लोड करने के लिए किया जाता है।

अन्य

यदि चयनित छवि फ़ाइल मौजूद नहीं है (पिछले यदि कथन की स्थिति पूरी नहीं हुई है), तो कोड इस अनुभाग पर आगे बढ़ता है।

' यदि छवि फ़ाइल मौजूद नहीं है तो छवि नियंत्रण साफ़ करें Image1.Picture = LoadPicture('') ' इससे छवि नियंत्रण साफ़ हो जाएगा

यहां, कोड एक खाली चित्र लोड करके, पहले प्रदर्शित किसी भी छवि को प्रभावी ढंग से हटाकर Image1 नियंत्रण को साफ़ करता है।

' एक कस्टम त्रुटि संदेश MsgBox प्रदर्शित करें "छवि निर्दिष्ट स्थान या अमान्य फ़ाइल प्रारूप में नहीं मिली।" & _ vbCrLf & "कृपया एक वैध छवि फ़ाइल का चयन करें।", vb विस्मयादिबोधक, "त्रुटि - गौतम बनर्जी"

इस अंतिम भाग में, कोड MsgBox फ़ंक्शन का उपयोग करके एक कस्टम त्रुटि संदेश प्रदर्शित करता है। यह संदेश उपयोगकर्ता को सूचित करता है कि चयनित छवि फ़ाइल नहीं मिली या अमान्य प्रारूप में है। यह एक वैध छवि फ़ाइल का चयन करने के लिए मार्गदर्शन प्रदान करता है।

त्रुटि GoTo 0 पर

अंत में, यह लाइन त्रुटि प्रबंधन को उसकी डिफ़ॉल्ट स्थिति में रीसेट कर देती है, जिससे VBA को सामान्य त्रुटि सूचनाएं फिर से शुरू करने की अनुमति मिलती है।

```
Private Sub CommandButtonAdd_Click()

    ' Get the selected image name and caption

    Dim selectedImageName As String

    selectedImageName = ComboBox1.value

    Dim caption As String

    caption = TextBoxCaption.Text
```

```
    ' Validate input
    If selectedImageName = "" Or caption = "" Then
        MsgBox "Please select an image and enter a caption.", vbExclamation
        Exit Sub
    End If

    ' Update your worksheet (replace "YourSheetName" with your sheet name)
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("NewSheet")
    Dim newRow As Long
    newRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).row + 1

    ' Add the new image name and caption to the worksheet
    ws.Cells(newRow, "A").value = selectedImageName
    ws.Cells(newRow, "B").value = caption

    ' Inform the user that the image has been added
    MsgBox "Image added successfully!", vbInformation

    ' Clear the TextBox and refresh the ComboBox
    TextBoxCaption.Text = ""
    ComboBox1.AddItem selectedImageName
End Sub
```

Here's a description of the CommandButtonAdd_Click subroutine:

Private Sub CommandButtonAdd_Click()

This line declares a subroutine named CommandButtonAdd_Click. Subroutines are blocks of code that are executed when a specific event occurs, in this case, when the user clicks the "Add" command button on the user form.

' Get the selected image name and caption

Dim selectedImageName As String

selectedImageName = ComboBox1.value

Dim caption As String

caption = TextBoxCaption.Text

In this section of the code, the subroutine retrieves two pieces of information from the user interface:

selectedImageName: This variable stores the value selected in a ComboBox named ComboBox1. The ComboBox typically contains a list of image names, and this line of code captures the selected image name.

caption: This variable stores the text entered into a TextBox named TextBoxCaption. The TextBox is used for entering captions for the selected images.

' Validate input **If selectedImageName = "" Or caption = "" Then**

MsgBox "Please select an image and enter a caption.", vbExclamation

Exit Sub

End If

Here, the code performs input validation to ensure that both the image name and the caption are provided by the user. If either of them is empty, a message box is displayed with an exclamation icon (vbExclamation) and a message asking the user to select an image and enter a caption. The Exit Sub statement is used to exit the subroutine if validation fails, preventing further execution of the code.

' Update your worksheet (replace "YourSheetName" with your sheet name) **Dim ws As Worksheet**

Set ws = ThisWorkbook.Sheets("NewSheet")

This part of the code defines a variable ws that represents a specific worksheet in the Excel workbook. The worksheet is specified by name within the Sheets collection. You should replace "NewSheet" with the name of the actual worksheet where you want to store the image names and captions.

Dim newRow As Long

newRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).row + 1

Here, the code calculates the row where the new image name and caption will be added in the worksheet. It finds the last used row in column "A" of the worksheet (ws) and adds 1 to it to determine the next available row.

' Add the new image name and caption to the worksheet **ws.Cells(newRow, "A").value = selectedImageName ws.Cells(newRow, "B").value = caption**

This section writes the selected image name and the caption to the calculated row in the worksheet.

' Inform the user that the image has been added MsgBox "Image added successfully!", vbInformation

After successfully adding the image name and caption to the worksheet, the code displays a message box with an information icon (vbInformation) to notify the user that the image has been added.

' Clear the TextBox and refresh the **ComboBox TextBoxCaption.Text = "" ComboBox1.AddItem selectedImageName**

Finally, this part of the code clears the caption TextBox and refreshes the ComboBox. Clearing the TextBox ensures that it's ready for entering the next caption, and refreshing the ComboBox adds the newly added image name so that the user can easily select it for further operations.

In summary, the CommandButtonAdd_Click subroutine is responsible for capturing user input (selected image name and caption), validating the input, adding this information to a specified worksheet, and providing feedback to the user about the success of the operation.

यहां CommandButtonAdd_Click सबरूटीन का विवरण दिया गया है:

निजी उप कमांडबटनAdd_Click()

यह पंक्ति CommandButtonAdd_Click नामक एक सबरूटीन घोषित करती है। सबरूटीन्स कोड के ब्लॉक होते हैं जो किसी विशिष्ट घटना के घटित होने पर निष्पादित होते हैं, इस मामले में, जब उपयोगकर्ता उपयोगकर्ता फॉर्म पर "जोड़ें" कमांड बटन पर क्लिक करता है।

'चयनित छवि का नाम और कैप्शन प्राप्त करें

चयनित छविनाम को स्ट्रिंग के रूप में मंद करें

चयनितImageName = ComboBox1.value

स्ट्रिंग के रूप में मंद कैप्शन

कैप्शन = टेक्स्टबॉक्सकैप्शन.टेक्स्ट

कोड के इस अनुभाग में, सबरूटीन उपयोगकर्ता इंटरफ़ेस से जानकारी के दो टुकड़े पुनर्प्राप्त करता है:

चयनितइमेजनाम: यह वेरिएबल कॉम्बोबॉक्स1 नामक कॉम्बोबॉक्स में चयनित मान को संग्रहीत करता है। कॉम्बोबॉक्स में आमतौर पर छवि नामों की एक सूची होती है, और कोड की यह पंक्ति चयनित छवि नाम को कैप्चर करती है।

कैप्शन: यह वेरिएबल टेक्स्टबॉक्सकैप्शन नामक टेक्स्टबॉक्स में दर्ज किए गए टेक्स्ट को संग्रहीत करता है। टेक्स्टबॉक्स का उपयोग चयनित छवियों के लिए कैप्शन दर्ज करने के लिए किया जाता है।

' इनपुट मान्य करें यदि चयनित है तो छवि नाम = "" या कैप्शन = "" फिर

संदेश बॉक्स "कृपया एक छवि चुनें और एक कैप्शन दर्ज करें।", vb विस्मयादिबोधक

उप से बाहर निकलें

अगर अंत

यहां, कोड यह सुनिश्चित करने के लिए इनपुट सत्यापन करता है कि छवि नाम और कैप्शन दोनों उपयोगकर्ता द्वारा प्रदान किए गए हैं। यदि उनमें से कोई भी खाली है, तो विस्मयादिबोधक चिह्न (vbExclamation) के साथ एक संदेश बॉक्स प्रदर्शित होता है और एक संदेश उपयोगकर्ता से एक छवि का चयन करने और एक कैप्शन दर्ज करने के लिए कहता है। यदि सत्यापन विफल हो जाता है, तो कोड के आगे निष्पादन को रोकने के लिए एग्जिट सब स्टेटमेंट का उपयोग सबरूटीन से बाहर निकलने के लिए किया जाता है।

' अपनी वर्कशीट को अपडेट करें ("योरशीटनेम" को अपनी शीट के नाम से बदलें) वर्कशीट के रूप में डिम डब्लूएस

सेट ws = ThisWorkbook.Sheets("NewSheet")

कोड का यह भाग एक वेरिएबल ws को परिभाषित करता है जो एक्सेल वर्कबुक में एक विशिष्ट वर्कशीट का प्रतिनिधित्व करता है। वर्कशीट को शीट्स संग्रह के भीतर नाम से निर्दिष्ट किया गया है। आपको "न्यूशीट" को वास्तविक वर्कशीट के नाम से बदलना चाहिए जहां आप छवि नाम और कैप्शन संग्रहीत करना चाहते हैं।

नई पंक्ति को उतनी ही लंबी मंद करें

newRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).row + 1

यहां, कोड उस पंक्ति की गणना करता है जहां वर्कशीट में नया छवि नाम और कैप्शन जोड़ा जाएगा। यह वर्कशीट (डब्ल्यूएस) के कॉलम "ए" में अंतिम प्रयुक्त पंक्ति ढूंढता है और अगली उपलब्ध पंक्ति निर्धारित करने के लिए इसमें 1 जोड़ता है।

' वर्कशीट में नया छवि नाम और कैप्शन जोड़ें ws.Cells(newRow, "A").value = चयनितImageName ws.Cells(newRow, "B").value = कैप्शन

यह अनुभाग चयनित छवि नाम और कैप्शन को वर्कशीट में गणना की गई पंक्ति में लिखता है।

' उपयोगकर्ता को सूचित करें कि छवि जोड़ दी गई है MsgBox "छवि सफलतापूर्वक जोड़ी गई!", vbInformation

वर्कशीट में छवि का नाम और कैप्शन सफलतापूर्वक जोड़ने के बाद, कोड उपयोगकर्ता को सूचित करने के लिए एक सूचना आइकन (vbInformation) के साथ एक संदेश बॉक्स प्रदर्शित करता है कि छवि जोड़ दी गई है।

' टेक्स्टबॉक्स को साफ़ करें और कॉम्बोबॉक्स को रीफ्रेश करें TextBoxCaption.Text = "" ComboBox1.AddItem चयनितImageName

अंत में, कोड का यह भाग कैप्शन टेक्स्टबॉक्स को साफ़ करता है और कॉम्बोबॉक्स को रीफ्रेश करता है। टेक्स्टबॉक्स को साफ़ करने से यह सुनिश्चित होता है कि यह अगले कैप्शन में प्रवेश करने के लिए तैयार है, और कॉम्बोबॉक्स को रीफ्रेश करने से नया जोड़ा गया छवि नाम जुड़ जाता है ताकि उपयोगकर्ता आगे के संचालन के लिए इसे आसानी से चुन सके।

संक्षेप में, CommandButtonAdd_Click सबरूटीन उपयोगकर्ता इनपुट (चयनित छवि नाम और कैप्शन) को कैप्चर करने, इनपुट को मान्य करने, इस जानकारी को एक निर्दिष्ट वर्कशीट में जोड़ने और ऑपरेशन की सफलता के बारे में उपयोगकर्ता को फीडबैक प्रदान करने के लिए जिम्मेदार है।

```
Private Sub Label5_Click()

    Unload Me

    MainMenuForm.Show

End Sub
```

The Unload Me statement is used to close and unload the current user form. In this context, it means that when the user clicks on "Label5," it will close the current form. This can be useful for providing a way to exit or navigate back to a previous form or menu.

After unloading the current form, this line of code instructs Excel to show another user form named "MainMenuForm." It opens or displays the main menu form, allowing the user to navigate to different parts of the application or perform various tasks.

In summary, the Label5_Click subroutine is typically used for creating a user interface with navigation capabilities. When the user clicks on "Label5," it closes the current form and opens the main menu form, facilitating user interaction and navigation within the application.

अनलोड मी स्टेटमेंट का उपयोग वर्तमान उपयोगकर्ता फॉर्म को बंद करने और अनलोड करने के लिए किया जाता है। इस संदर्भ में, इसका मतलब है कि जब उपयोगकर्ता "लेबल5" पर क्लिक करता है, तो यह वर्तमान फॉर्म को बंद कर देगा। यह पिछले फॉर्म या मेनू से बाहर निकलने या वापस नेविगेट करने का तरीका प्रदान करने के लिए उपयोगी हो सकता है।

वर्तमान फॉर्म को अनलोड करने के बाद, कोड की यह पंक्ति एक्सेल को "मेनमेनूफॉर्म" नामक एक अन्य उपयोगकर्ता फॉर्म दिखाने का निर्देश देती है। यह मुख्य मेनू फॉर्म को खोलता या प्रदर्शित करता है, जिससे उपयोगकर्ता को एप्लिकेशन के विभिन्न हिस्सों में नेविगेट करने या विभिन्न कार्य करने की अनुमति मिलती है।

संक्षेप में, Label5_Click सबरूटीन का उपयोग आम तौर पर नेविगेशन क्षमताओं के साथ उपयोगकर्ता इंटरफ़ेस बनाने के लिए किया जाता है। जब उपयोगकर्ता "लेबल5" पर क्लिक करता है, तो यह वर्तमान फॉर्म को बंद कर देता है और मुख्य मेनू फॉर्म को खोलता है, जिससे एप्लिकेशन के भीतर उपयोगकर्ता इंटरैक्शन और नेविगेशन की सुविधा मिलती है।

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 Google Pay

9748327614