Python code snippet that creates a simple GUI application using the Tkinter library. This application allows the user to upload an image, make changes to it based on a description (e.g., change brightness, apply filters, or alter colors), and includes a switch option to adjust color combinations. The code ensures no API usage to avoid errors. The logic for image processing uses the Pillow (PIL) library.

In [3]:
```
import tkinter as tk
from tkinter import filedialog
from tkinter import messagebox
from tkinter import ttk
from PIL import Image, ImageTk, ImageEnhance, ImageOps

# Function to open an image
def upload_image():
    global img, img_display, uploaded_image_path

    file_path = filedialog.askopenfilename(
        filetypes=[("Image files", "*.jpg;*.jpeg;*.png;*.bmp;*.gif")]
    )
    if not file_path:
        return

    try:
        img = Image.open(file_path)
        uploaded_image_path = file_path
        img_display = ImageTk.PhotoImage(img.resize((400, 400)))
        canvas.itemconfig(image_container, image=img_display)
        messagebox.showinfo("Success", "Image uploaded successfully!")
    except Exception as e:
        messagebox.showerror("Error", f"Unable to open image: {str(e)}")

# Function to apply color combination based on switch
def apply_color_combination():
    global img, img_display

    if not img:
        messagebox.showwarning("Warning", "Please upload an image first!")
        return

    try:
        if color_combination_switch.get():
            img_modified = ImageOps.colorize(img.convert("L"), "blue", "yellow")
        else:
            img_modified = img.convert("L")  # Grayscale
        img_display = ImageTk.PhotoImage(img_modified.resize((400, 400)))
```

```
        canvas.itemconfig(image_container, image=img_display)
    except Exception as e:
        messagebox.showerror("Error", f"Error applying color combination: {str(e)}")

# Function to apply changes based on user description
def apply_changes():
    global img, img_display

    if not img:
        messagebox.showwarning("Warning", "Please upload an image first!")
        return

    description = description_entry.get()

    try:
        # Brightness enhancement example
        if "brighten" in description.lower():
            enhancer = ImageEnhance.Brightness(img)
            img = enhancer.enhance(1.5)
        elif "darken" in description.lower():
            enhancer = ImageEnhance.Brightness(img)
            img = enhancer.enhance(0.7)

        # Contrast enhancement example
        elif "contrast" in description.lower():
            enhancer = ImageEnhance.Contrast(img)
            img = enhancer.enhance(1.5)

        img_display = ImageTk.PhotoImage(img.resize((400, 400)))
        canvas.itemconfig(image_container, image=img_display)
        messagebox.showinfo("Success", "Changes applied based on description!")
    except Exception as e:
        messagebox.showerror("Error", f"Error applying changes: {str(e)}")

# Initialize the main window
root = tk.Tk()
root.title("Image Modifier Tool")
root.geometry("600x600")
root.resizable(False, False)

# Canvas to display image
canvas = tk.Canvas(root, width=400, height=400, bg="white")
canvas.pack(pady=10)
image_container = canvas.create_image(200, 200, anchor=tk.CENTER)
```

```
# Description entry
description_label = tk.Label(root, text="Enter modification description:")
description_label.pack(pady=5)
description_entry = tk.Entry(root, width=50)
description_entry.pack(pady=5)

# Switch for color combination
color_combination_switch = tk.BooleanVar()
color_combination_checkbutton = ttk.Checkbutton(
    root, text="Enable Color Combination", variable=color_combination_switch, command=apply_color_combination
)
color_combination_checkbutton.pack(pady=5)

# Buttons
button_frame = tk.Frame(root)
button_frame.pack(pady=10)
upload_button = ttk.Button(button_frame, text="Upload Image", command=upload_image)
upload_button.grid(row=0, column=0, padx=10)
apply_button = ttk.Button(button_frame, text="Apply Changes", command=apply_changes)
apply_button.grid(row=0, column=1, padx=10)

# Run the application
root.mainloop()
```

Key Features: Upload Image: Users can upload an image of supported formats (e.g., .jpg, .png). Modify Description: Users can input a description like "brighten," "darken," or "increase contrast" to make changes to the image. Color Combination Switch: A toggle to apply predefined color combinations, like grayscale or a blue-yellow colorize effect. User-Friendly GUI: Built using Tkinter for simplicity and ease of use. Required Libraries: Pillow (install using pip install pillow) This code is self-contained and does not rely on external APIs. Let me know if you'd like further adjustments!

In [ ]: