This Python script creates a simple desktop-based dashboard where users can upload an image and generate an English story about it, avoiding web-based platforms or API errors. It uses tkinter for the GUI and Pillow for image handling. The steps involve: Uploading an Image: A user selects an image from their computer using a file dialog. Displaying the Image: The uploaded image is resized and displayed on the dashboard. Story Generation: A predefined story is created based on the uploaded image's format when the user presses a button.

In [7]:

```python
import tkinter as tk
from tkinter import filedialog
from PIL import Image, ImageTk
import os

def upload_image():
    file_path = filedialog.askopenfilename(
        filetypes=[("Image Files", "*.png;*.jpg;*.jpeg;*.bmp;*.gif")]
    )
    if file_path:
        img = Image.open(file_path)
        img.thumbnail((400, 400))  # Resize for the GUI
        img_tk = ImageTk.PhotoImage(img)
        img_label.config(image=img_tk)
        img_label.image = img_tk
        app_data["image_path"] = file_path

def generate_story():
    if "image_path" not in app_data or not os.path.exists(app_data["image_path"]):
        story_label.config(text="Please upload an image first!")
        return

    file_extension = os.path.splitext(app_data["image_path"])[1].lower()
    if file_extension in ['.jpg', '.jpeg']:
        story = "This image captures vivid memories in brilliant colors."
    elif file_extension == '.png':
        story = "A crisp and detailed image with endless possibilities."
    elif file_extension == '.bmp':
        story = "A nostalgic journey back to classic image formats."
    elif file_extension == '.gif':
        story = "A lively image bringing motion to moments."
    else:
        story = "An intriguing image full of mysteries yet to unfold."

    story_label.config(text=story)

root = tk.Tk()
root.title("Image Story Generator")
```

```
app_data = {}

upload_btn = tk.Button(root, text="Upload Image", command=upload_image)
upload_btn.pack()

img_label = tk.Label(root)
img_label.pack()

story_btn = tk.Button(root, text="Generate Story", command=generate_story)
story_btn.pack()

story_label = tk.Label(root, text="", wraplength=400, justify="center")
story_label.pack()

root.mainloop()
```

In [ ]: