import tkinter as tk

from tkinter import filedialog

from PIL import Image, ImageTk

from google import genai

import threading

import time

def analyze\_image(image\_path):

# Show processing indicator

processing\_label.pack()

result\_text.set("Processing... Please wait.")

image\_label.config(image='') # Clear previous image preview

root.update()

client = genai.Client(api\_key='') # Replace with your API key

image = Image.open(image\_path)

img\_display = image.resize((300, 300))

img\_display = ImageTk.PhotoImage(img\_display)

image\_label.config(image=img\_display)

image\_label.image = img\_display

# Simulate slow network/API for demo purposes

time.sleep(1) # You can remove this later

response = client.models.generate\_content(

model="gemini-2.5-flash",

contents=[image, "Tell me about this image"]

)

description = response.text

print("Gemini Description:", description)

damage\_keywords = ["broken", "cracked", "damaged", "shattered", "defective", "faulty"]

if any(keyword in description.lower() for keyword in damage\_keywords):

result\_text.set("Product is Damaged – Refund Applicable✅")

else:

result\_text.set("Product is OK – Refund Not Applicable❌")

# Hide processing indicator

processing\_label.pack\_forget()

def open\_file\_dialog():

file\_path = filedialog.askopenfilename(

title="Select Image",

filetypes=[("Image Files", "\*.jpg \*.jpeg \*.png")]

)

if file\_path:

# Run analyze\_image in a separate thread to keep GUI responsive

threading.Thread(target=analyze\_image, args=(file\_path,), daemon=True).start()

# --- Build Full-Screen UI ---

root = tk.Tk()

root.title("Product Image Damage Analyzer")

root.attributes('-fullscreen', True)

exit\_btn = tk.Button(root, text="Exit", command=root.destroy, font=("Arial", 14), bg="red", fg="white")

exit\_btn.pack(side=tk.BOTTOM, pady=10)

label = tk.Label(root, text="Upload product image for damage analysis", font=("Arial", 20))

label.pack(pady=30)

upload\_btn = tk.Button(root, text="Select Image", command=open\_file\_dialog, width=25, height=3, font=("Arial", 16))

upload\_btn.pack(pady=20)

# Image preview label

image\_label = tk.Label(root)

image\_label.pack(pady=20)

# Processing indicator

processing\_label = tk.Label(root, text="⏳ Processing... Please wait", font=("Arial", 20), fg="blue")

# Result label

result\_text = tk.StringVar()

result\_label = tk.Label(root, textvariable=result\_text, font=("Arial", 24), fg="green")

result\_label.pack(pady=30)

root.mainloop()