## RD PixelPress – Compress Numerous Images Simultaneously Python code.py

import os # This imports Python's built-in os module. It lets you interact with the operating system. Ex -Get file names and paths , Create or delete folders, Work with directories.  
from tkinter import \*  
from tkinter import filedialog, messagebox  
from PIL import Image # This imports the Image class from the PIL (Pillow) library. Pillow is a popular library for image processing in Python. use Image to open, resize, edit, and save images.  
  
# --- Resize and compress image ---  
def resize\_pic(old\_pic, new\_pic, width, height, log\_callback):  
 try:  
 img = Image.open(old\_pic)  
 img = img.resize((width, height), Image.LANCZOS)  
 img.save(new\_pic, format="JPEG", quality=65, optimize=True)  
 log\_callback(f"✅ {os.path.basename(old\_pic)} - Compressed & saved.")  
 except Exception as e:  
 log\_callback(f"❌ Error: {e}")  
  
# --- Compress all images in selected folder ---  
def entire\_directory(source\_dir, width, height):  
 dest\_dir = os.path.join(os.path.dirname(source\_dir), "Compressed\_Images")  
 os.makedirs(dest\_dir, exist\_ok=True)  
  
 files = os.listdir(source\_dir)  
 image\_files = [f for f in files if f.lower().endswith(('.png', '.jpg', '.jpeg', '.bmp', '.gif', '.tiff', '.webp'))]  
  
 if not image\_files:  
 log("⚠️ No image files found in the selected folder.")  
 return  
  
 log(f"🔄 Found {len(image\_files)} image(s). Starting compression...\n")  
  
 for i, file in enumerate(image\_files, 1):  
 old\_pic = os.path.join(source\_dir, file)  
 new\_filename = os.path.splitext(file)[0] + ".jpg"  
 new\_pic = os.path.join(dest\_dir, new\_filename)  
 resize\_pic(old\_pic, new\_pic, width, height, log)  
  
 log(f"\n✅ All images saved to:\n{dest\_dir}")  
 messagebox.showinfo("Success", f"All images saved to:\n{dest\_dir}")  
  
# --- Logging helper ---  
def log(message):  
 log\_output.insert(END, message + "\n")  
 log\_output.see(END)  
  
# --- Folder browser ---  
def browse\_folder():  
 folder = filedialog.askdirectory()  
 if folder:  
 source\_folder.set(folder)  
  
# --- Start compression ---  
def start\_compression():  
 w = width\_entry.get()  
 h = height\_entry.get()  
  
 try:  
 width = int(w)  
 height = int(h)  
 except ValueError:  
 messagebox.showerror("Invalid Input", "Please enter valid numbers for width and height.")  
 return  
  
 folder = source\_folder.get()  
 if not folder or not os.path.isdir(folder):  
 messagebox.showerror("Invalid Folder", "Please select a valid folder.")  
 return  
  
 log\_output.delete(1.0, END)  
 entire\_directory(folder, width, height)  
  
# --- GUI Setup ---  
root = Tk()  
root.title("🖼️ RD PixelPress")  
root.geometry("750x520")  
root.configure(bg="#f8f9fa")  
  
# Grid layout  
root.columnconfigure(1, weight=1)  
root.rowconfigure(7, weight=1)  
  
# Title  
Label(root, text="📸 RD PixelPress", font=("Segoe UI", 16, "bold"), bg="#f8f9fa", fg="#2c3e50").grid(row=0, column=0, columnspan=3, pady=(10, 0))  
Label(  
 root,  
 text="We Are Here To Compress Numerous Photos Simultaneously",  
 font=("Segoe UI", 10, "italic"),  
 bg="#f8f9fa",  
 fg="#007bff"  
).grid(row=1, column=0, columnspan=3, pady=(0, 10))  
  
# Width Entry  
Label(root, text="Width:", bg="#f8f9fa", font=("Segoe UI", 10)).grid(row=2, column=0, sticky=W, padx=10)  
width\_entry = Entry(root)  
width\_entry.grid(row=2, column=1, sticky=EW, padx=10, pady=5)  
  
# Width Suggestion  
Label(root,  
 text="Suggestion: For ideal compression of passport size photo, take width 600px",  
 bg="#f8f9fa", fg="#cc0000", font=("Segoe UI", 9, "italic")).grid(row=3, column=1, sticky=W, padx=10)  
  
# Height Entry  
Label(root, text="Height:", bg="#f8f9fa", font=("Segoe UI", 10)).grid(row=4, column=0, sticky=W, padx=10)  
height\_entry = Entry(root)  
height\_entry.grid(row=4, column=1, sticky=EW, padx=10, pady=5)  
  
# Height Suggestion  
Label(root,  
 text="Suggestion: For ideal compression of passport size photo, take height 600px",  
 bg="#f8f9fa", fg="#cc0000", font=("Segoe UI", 9, "italic")).grid(row=5, column=1, sticky=W, padx=10)  
  
# Source Folder  
source\_folder = StringVar()  
Label(root, text="Source Folder:", bg="#f8f9fa", font=("Segoe UI", 10)).grid(row=6, column=0, sticky=W, padx=10, pady=5)  
Entry(root, textvariable=source\_folder).grid(row=6, column=1, padx=10, sticky=EW)  
Button(root, text="Browse", command=browse\_folder, bg="#007bff", fg="white").grid(row=6, column=2, padx=10)  
  
# Start button  
Button(root, text="Start Compression", command=start\_compression,  
 bg="#28a745", fg="white", font=("Segoe UI", 10, "bold")).grid(row=7, column=0, columnspan=3, pady=15)  
  
# Log Output  
Label(root, text="Status Log:", bg="#f8f9fa", font=("Segoe UI", 10)).grid(row=8, column=0, sticky=W, padx=10)  
log\_output = Text(root, height=12, wrap=WORD, font=("Consolas", 9))  
log\_output.grid(row=9, column=0, columnspan=3, padx=10, pady=5, sticky="nsew")  
  
# Scrollbar for log  
scrollbar = Scrollbar(root, command=log\_output.yview)  
scrollbar.grid(row=9, column=3, sticky='ns')  
log\_output['yscrollcommand'] = scrollbar.set  
  
# Make log area expandable  
root.rowconfigure(9, weight=1)  
  
# Run GUI  
root.mainloop()