TO DO LIST APPLICATION:

import tkinter as tk

from tkinter import messagebox

import json

import os

# File to store tasks

TASKS\_FILE = 'tasks.json'

# Load tasks from the JSON file

def load\_tasks():

if os.path.exists(TASKS\_FILE):

with open(TASKS\_FILE, 'r') as file:

return json.load(file)

return []

# Save tasks to the JSON file

def save\_tasks(tasks):

with open(TASKS\_FILE, 'w') as file:

json.dump(tasks, file, indent=4)

# Add a task

def add\_task():

title = title\_entry.get()

description = description\_entry.get()

category = category\_entry.get()

if title and description and category:

task = {

'title': title,

'description': description,

'completed': False,

'category': category

}

tasks.append(task)

save\_tasks(tasks)

update\_task\_list()

clear\_entries()

else:

messagebox.showwarning("Input Error", "Please fill in all fields.")

# Mark a task as completed

def mark\_completed():

selected\_task\_index = task\_listbox.curselection()

if selected\_task\_index:

task\_index = selected\_task\_index[0]

tasks[task\_index]['completed'] = True

save\_tasks(tasks)

update\_task\_list()

else:

messagebox.showwarning("Selection Error", "Please select a task.")

# Delete a task

def delete\_task():

selected\_task\_index = task\_listbox.curselection()

if selected\_task\_index:

task\_index = selected\_task\_index[0]

tasks.pop(task\_index)

save\_tasks(tasks)

update\_task\_list()

else:

messagebox.showwarning("Selection Error", "Please select a task.")

# Update the task list display

def update\_task\_list():

task\_listbox.delete(0, tk.END)

for index, task in enumerate(tasks):

status = "✔️" if task['completed'] else "❌"

task\_listbox.insert(tk.END, f"{status} {task['title']} - {task['description']} (Category: {task['category']})")

# Clear input fields

def clear\_entries():

title\_entry.delete(0, tk.END)

description\_entry.delete(0, tk.END)

category\_entry.delete(0, tk.END)

# Initialize the main window

app = tk.Tk()

app.title("To-Do List Application")

# Create UI components

frame = tk.Frame(app)

frame.pack(pady=10)

title\_label = tk.Label(frame, text="Title:")

title\_label.grid(row=0, column=0)

title\_entry = tk.Entry(frame)

title\_entry.grid(row=0, column=1)

description\_label = tk.Label(frame, text="Description:")

description\_label.grid(row=1, column=0)

description\_entry = tk.Entry(frame)

description\_entry.grid(row=1, column=1)

category\_label = tk.Label(frame, text="Category:")

category\_label.grid(row=2, column=0)

category\_entry = tk.Entry(frame)

category\_entry.grid(row=2, column=1)

add\_button = tk.Button(frame, text="Add Task", command=add\_task)

add\_button.grid(row=3, columnspan=2)

mark\_completed\_button = tk.Button(app, text="Mark Completed", command=mark\_completed)

mark\_completed\_button.pack(pady=5)

delete\_button = tk.Button(app, text="Delete Task", command=delete\_task)

delete\_button.pack(pady=5)

task\_listbox = tk.Listbox(app, width=50)

task\_listbox.pack(pady=10)

# Load existing tasks and update the list display

tasks = load\_tasks()

update\_task\_list()

# Run the application

app.mainloop()