

```

# -*- coding: utf-8 -*-
"""
Created on Fri Jul 18 20:42:46 2025

@author: Lenovo
"""
import tkinter as tk
from tkinter import messagebox
import json
import os

FILE_NAME = "todo_data.json"

def load_tasks():
    if os.path.exists(FILE_NAME):
        with open(FILE_NAME, "r") as f:
            return json.load(f)
    return []

def save_tasks(tasks):
    with open(FILE_NAME, "w") as f:
        json.dump(tasks, f)

class ToDoApp:
    def __init__(self, root):
        self.root = root
        self.root.title("To-Do List")
        self.root.geometry("400x500")
        self.tasks = load_tasks()

        self.task_var = tk.StringVar()

        self.title = tk.Label(root, text="My To-Do List", font=("Arial", 16, "bold"))
        self.title.pack(pady=10)

        self.task_entry = tk.Entry(root, textvariable=self.task_var, width=30, font=("Arial", 12))
        self.task_entry.pack(pady=5)

        self.add_btn = tk.Button(root, text="Add Task", command=self.add_task, width=20)
        self.add_btn.pack(pady=5)

        self.task_listbox = tk.Listbox(root, selectmode=tk.SINGLE, font=("Arial", 12), width=35, height=15)
        self.task_listbox.pack(pady=10)

        self.delete_btn = tk.Button(root, text="Delete Selected Task", command=self.delete_task, width=20)
        self.delete_btn.pack(pady=5)

        self.mark_btn = tk.Button(root, text="Mark as Done", command=self.mark_done, width=20)
        self.mark_btn.pack(pady=5)

        self.refresh_tasks()

    def add_task(self):
        task = self.task_var.get().strip()
        if task:
            self.tasks.append({"task": task, "done": False})
            self.task_var.set("")
            self.refresh_tasks()
            save_tasks(self.tasks)
        else:
            messagebox.showwarning("Input Error", "Task cannot be empty!")

    def delete_task(self):
        selected = self.task_listbox.curselection()
        if selected:
            del self.tasks[selected[0]]
            self.refresh_tasks()
            save_tasks(self.tasks)
        else:
            messagebox.showwarning("Select Task", "Please select a task to delete.")

    def mark_done(self):
        selected = self.task_listbox.curselection()
        if selected:
            self.tasks[selected[0]]["done"] = True
            self.refresh_tasks()
            save_tasks(self.tasks)
        else:
            messagebox.showwarning("Select Task", "Please select a task to mark as done.")

    def refresh_tasks(self):
        self.task_listbox.delete(0, tk.END)
        for task in self.tasks:
            status = " " if task["done"] else " -"
            self.task_listbox.insert(tk.END, f"[{status}] {task['task']}")

if __name__ == "__main__":
    root = tk.Tk()
    app = ToDoApp(root)
    root.mainloop()

```