```
import tkinter as tk
from tkinter import messagebox, simpledialog
import json

class TodoApp:
    def __init__(self, root):
        self.root = root
        self.root.title("To-Do List")
        self.tasks = []
        self.load_tasks()

        self.frame = tk.Frame(self.root)
        self.frame.pack(pady=10)

        self.listbox = tk.Listbox(self.frame, width=50, height=10, selectmode=tk.SINGLE)
        self.listbox.pack(side=tk.LEFT, padx=5)

        self.scrollbar = tk.Scrollbar(self.frame, orient=tk.VERTICAL)
        self.scrollbar.config(command=self.listbox.yview)
        self.scrollbar.pack(side=tk.RIGHT, fill=tk.Y)
        self.listbox.config(yscrollcommand=self.scrollbar.set)

        self.entry = tk.Entry(self.root, width=50)
        self.entry.pack(pady=5)

        self.add_button = tk.Button(self.root, text="Add Task", command=self.add_task)
        self.add_button.pack(pady=2)

        self.complete_button = tk.Button(self.root, text="Mark Completed",
command=self.mark_completed)
        self.complete_button.pack(pady=2)
```

```
        self.edit_button = tk.Button(self.root, text="Edit Task", command=self.edit_task)
        self.edit_button.pack(pady=2)

        self.delete_button = tk.Button(self.root, text="Delete Task", command=self.delete_task)
        self.delete_button.pack(pady=2)

        self.save_button = tk.Button(self.root, text="Save Tasks", command=self.save_tasks)
        self.save_button.pack(pady=2)

        self.quit_button = tk.Button(self.root, text="Quit", command=self.root.quit)
        self.quit_button.pack(pady=2)

        self.refresh_list()

    def add_task(self):
        task = self.entry.get()
        if task:
            self.tasks.append({"task": task, "completed": False})
            self.entry.delete(0, tk.END)
            self.refresh_list()
        else:
            messagebox.showwarning("Warning", "Task cannot be empty!")

    def mark_completed(self):
        try:
            index = self.listbox.curselection()[0]
            self.tasks[index]["completed"] = True
            self.refresh_list()
        except IndexError:
            messagebox.showwarning("Warning", "Please select a task to mark completed.")
```

```
    def edit_task(self):
        try:
            index = self.listbox.curselection()[0]
            new_task = simpledialog.askstring("Edit Task", "Edit the task:",
initialvalue=self.tasks[index]["task"])
            if new_task:
                self.tasks[index]["task"] = new_task
                self.refresh_list()
        except IndexError:
            messagebox.showwarning("Warning", "Please select a task to edit.")

    def delete_task(self):
        try:
            index = self.listbox.curselection()[0]
            del self.tasks[index]
            self.refresh_list()
        except IndexError:
            messagebox.showwarning("Warning", "Please select a task to delete.")

    def refresh_list(self):
        self.listbox.delete(0, tk.END)
        for task in self.tasks:
            display_text = f"{'[✓]' if task['completed'] else '[ ]'} {task['task']}"
            self.listbox.insert(tk.END, display_text)

    def save_tasks(self):
        with open("tasks.json", "w") as file:
            json.dump(self.tasks, file)
        messagebox.showinfo("Info", "Tasks saved successfully!")
```

```
    def load_tasks(self):
        try:
            with open("tasks.json", "r") as file:
                self.tasks = json.load(file)
        except FileNotFoundError:
            self.tasks = []

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