

## Task 1:- To do list application

### Code:-

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

def add_task():
    task = entry_task.get()
    if task != "":
        priority = priority_var.get()
        task_with_priority = f"{task} - Priority: {priority}"
        listbox_tasks.insert(tk.END, task_with_priority)
        entry_task.delete(0, tk.END)
    else:
        messagebox.showwarning("Warning", "Please enter a task.")

def delete_task():
    try:
        task_index = listbox_tasks.curselection()[0]
        listbox_tasks.delete(task_index)
    except IndexError:
        messagebox.showwarning("Warning", "Please select a task to delete.")

def edit_task():
    try:
        task_index = listbox_tasks.curselection()[0]
        selected_task = listbox_tasks.get(task_index)
        edited_task = simpledialog.askstring("Edit Task", "Edit task:", initialvalue=selected_task)
        if edited_task:
            listbox_tasks.delete(task_index)
            listbox_tasks.insert(task_index, edited_task)
    except IndexError:
        messagebox.showwarning("Warning", "Please select a task to edit.")

def save_tasks():
    tasks = listbox_tasks.get(0, tk.END)
    file_path = filedialog.asksaveasfilename(defaultextension=".txt", filetypes=[("Text files",
    "*.txt")])
    if file_path:
        with open(file_path, "w") as f:
            for task in tasks:
                f.write(task + "\n")

def load_tasks():
    file_path = filedialog.askopenfilename(filetypes=[("Text files", "*.txt")])
```

```
if file_path:
    try:
        with open(file_path, "r") as f:
            tasks = f.readlines()
            listbox_tasks.delete(0, tk.END)
            for task in tasks:
                listbox_tasks.insert(tk.END, task.strip())
    except FileNotFoundError:
        messagebox.showwarning("Warning", "No saved tasks found.")
```

```
# Create the main window
root = tk.Tk()
root.title("To-Do List")
root.geometry("400x400")
root.configure(bg="#f0f0f0")
```

```
# Create the task entry
entry_task = tk.Entry(root, width=40)
entry_task.pack(pady=10)
```

```
# Create priority label and dropdown
priority_label = tk.Label(root, text="Priority:", bg="#f0f0f0")
priority_label.pack()
```

```
priority_var = tk.StringVar(root)
priority_var.set("Low")
priority_dropdown = tk.OptionMenu(root, priority_var, "Low", "Medium", "High")
priority_dropdown.config(bg="ffffff")
priority_dropdown.pack()
```

```
# Create the task list
listbox_tasks = tk.Listbox(root, width=50)
listbox_tasks.pack()
```

```
# Create buttons
button_add_task = tk.Button(root, text="Add Task", width=15, command=add_task,
bg="#4CAF50", fg="white")
button_add_task.pack(pady=5)
```

```
button_delete_task = tk.Button(root, text="Delete Task", width=15, command=delete_task,
bg="#f44336", fg="white")
button_delete_task.pack(pady=5)
```

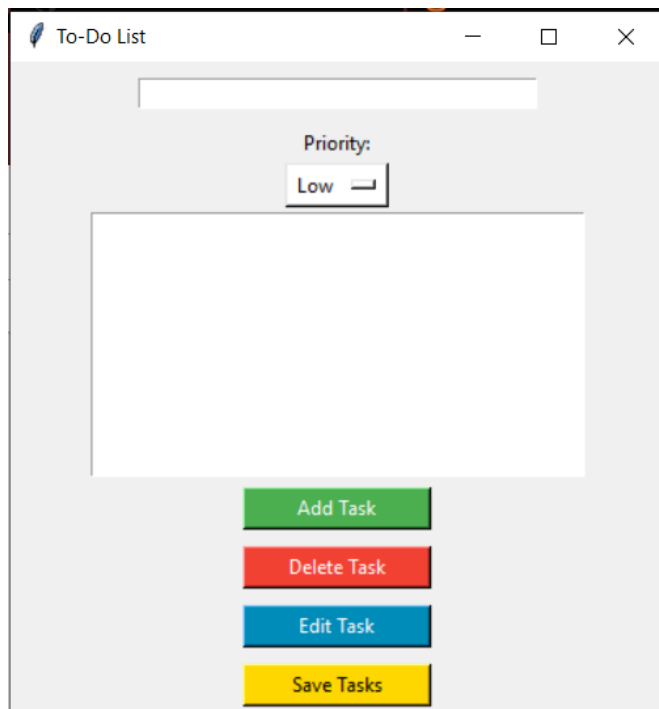
```
button_edit_task = tk.Button(root, text="Edit Task", width=15, command=edit_task,  
bg="#008CBA", fg="white")  
button_edit_task.pack(pady=5)
```

```
button_save_tasks = tk.Button(root, text="Save Tasks", width=15, command=save_tasks,  
bg="#FFD700", fg="black")  
button_save_tasks.pack(pady=5)
```

```
button_load_tasks = tk.Button(root, text="Load Tasks", width=15, command=load_tasks,  
bg="#20B2AA", fg="white")  
button_load_tasks.pack(pady=5)
```

```
# Run the main loop  
root.mainloop()
```

### Output:-



To-Do List

— □ ×

wake up at 6

Priority:

High

Low  
Medium  
High

Add Task

Delete Task

Edit Task

Save Tasks

To-Do List

— □ ×

Priority:

Medium

wake up at 6 - Priority: High  
break fast - Priority: Medium

Add Task

Delete Task

Edit Task

Save Tasks