

```
1 from tkinter import *
2 from tkinter import messagebox
3 import tkinter as tk
4 from PIL import ImageTk, Image
5
6
7
8 # Create the main window
9 window = Tk()
10 window.title("To-Do App")
11 # Set background color
12 window.configure(bg="#f5f5f5")
13 # Set the icon for the application window
14 window.iconbitmap(r"C:\Users\Abdur Razzaq\Downloads\
    target.ico")
15
16 # Set the background image
17 image = Image.open(r"C:\Users\Abdur Razzaq\
    PycharmProjects\pythonProject\bg.jpeg")
18 image = image.resize((750, 400), Image.BILINEAR) #
    Resize the image the way you want
19 background_image = ImageTk.PhotoImage(image)
20
21 # Create a label to hold the background image
22 background_label = Label(window, image=
    background_image)
23 background_label.place(x=0, y=0, relwidth=1,
    relheight=1)
24
25
26
27 # Creating a list to store tasks
28 tasks = []
29
30 # Function --> To Add The Task
31 def add_task():
32     def save_task():
33         new_task = entry_task.get()
34         if new_task:
35             tasks.append(new_task)
36             messagebox.showinfo("Success", "Task
```

```

36 added successfully!")
37         update_list()
38         root.destroy()
39     else:
40         messagebox.showwarning("Warning", "Please
        enter a task!")
41
42     root = Toplevel(window)
43     root.title("Add Task")
44     # Set bg-color
45     root.configure(bg="#f5f5f5")
46
47     entry_task = Entry(root, width=30)
48     entry_task.pack(pady=10)
49
50     button_save = Button(root, text="Save", command=
    save_task)
51     button_save.pack(pady=5)
52
53 # Function --> To delete The Task
54 def delete_task():
55     selected_task = listbox_tasks.curselection()
56     if selected_task:
57         confirmed = messagebox.askyesno("Confirmation
        ", "Are you sure you want to delete this task?")
58         if confirmed:
59             tasks.pop(selected_task[0])
60             update_list()
61     else:
62         messagebox.showwarning("Warning", "Please
        select a task to delete!")
63
64 # Function --> To edit The Task
65 def edit_task():
66     selected_task = listbox_tasks.curselection()
67     if selected_task:
68         current_task = tasks[selected_task[0]]
69
70         def save_changes():
71             new_task = entry_edit.get()
72             if new_task:

```

```

73         tasks[selected_task[0]] = new_task
74         messagebox.showinfo("Success", "Task
    edited successfully!")
75         update_list()
76         root.destroy()
77     else:
78         messagebox.showwarning("Warning", "
Please enter a task!")
79
80     root = Toplevel(window)
81     root.title("Edit Task")
82     # Set background color
83     root.configure(bg="#f5f5f5")
84
85     entry_edit = Entry(root, width=30)
86     entry_edit.insert(0, current_task)
87     entry_edit.pack(pady=10)
88
89     button_save = Button(root, text="Save
Changes", command=save_changes)
90     button_save.pack(pady=5)
91     else:
92         messagebox.showwarning("Warning", "Please
select a task to edit!")
93
94 # Function --> To Mark a Task As Completed
95 def complete_task():
96     selected_task = listbox_tasks.curselection()
97     if selected_task:
98         task = tasks[selected_task[0]]
99         if task.endswith(" \u2013"):
100             messagebox.showinfo("Info", "Task
already marked as complete!")
101         else:
102             task = task + " \u2013"
103             tasks[selected_task[0]] = task
104             update_list()
105             messagebox.showinfo("Success", "Task
marked as complete!")
106     else:
107         messagebox.showwarning("Warning", "Please

```

```
107 select a task to mark as complete!")
108
109 # Function --> To Update The Task --> example Rename
110 def update_list():
111     listbox_tasks.delete(0, END)
112     for task in tasks:
113         listbox_tasks.insert(END, task)
114
115 # Frame To hold the task-list and Scrollbar
116 frame_task = Frame(window)
117 frame_task.pack()
118
119 #listbox To Display the Tasks
120 listbox_tasks = Listbox(frame_task, bg="white", fg="
    black", height=15, width=50, font="Helvetica")
121 listbox_tasks.pack(side=LEFT)
122
123 # Create a scrollbar for the listbox
124 scrollbar_task = Scrollbar(frame_task)
125 scrollbar_task.pack(side=RIGHT, fill=Y)
126 listbox_tasks.config(yscrollcommand=scrollbar_task.
    set)
127 scrollbar_task.config(command=listbox_tasks.yview)
128
129 # Creating Buttons for Adding, Deleting, Editing,
    and Completing tasks
130 button_add = Button(window, text="Add Task", width=
    50, command=add_task)
131 button_add.pack(pady=5)
132
133 button_delete = Button(window, text="Delete Task",
    width=50, command=delete_task)
134 button_delete.pack(pady=5)
135
136 button_edit = Button(window, text="Edit Task", width
    =50, command=edit_task)
137 button_edit.pack(pady=5)
138
139 button_complete = Button(window, text="Complete Task
    ", width=50, command=complete_task)
140 button_complete.pack(pady=5)
```

```
141  
142 window.mainloop()
```