

E:\clockpy.py

```

1  import tkinter as tk
2  from tkinter import messagebox
3  from datetime import datetime, time, timedelta
4  import winsound # for sound notifications (Windows)
5
6  class AlarmClockApp:
7      def __init__(self, master):
8          self.master = master
9          self.master.title("Python Alarm Clock")
10         self.master.geometry("400x300")
11
12         self.alarm_time = None
13         self.alarm_active = False
14
15         # Time label
16         self.time_label = tk.Label(master, text="", font=("Helvetica", 48))
17         self.time_label.pack(pady=20)
18
19         # Entry for setting alarm time
20         self.entry_label = tk.Label(master, text="Set Alarm Time (HH:MM):")
21         self.entry_label.pack()
22
23         self.entry = tk.Entry(master, font=("Helvetica", 24))
24         self.entry.pack(pady=10)
25
26         # Set alarm button
27         self.set_alarm_button = tk.Button(master, text="Set Alarm", command=self.set_alarm)
28         self.set_alarm_button.pack(pady=10)
29
30         # Stop alarm button
31         self.stop_alarm_button = tk.Button(master, text="Stop Alarm", command=self.stop_alarm,
32 state=tk.DISABLED)
33         self.stop_alarm_button.pack(pady=10)
34
35         # Update clock display every second
36         self.update_clock()
37
38         def update_clock(self):
39             current_time = datetime.now().strftime("%H:%M:%S")
40             self.time_label.config(text=current_time)
41             self.check_alarm()
42
43             self.master.after(1000, self.update_clock) # Schedule the update_clock function to run
44 after 1 second
45
46         def set_alarm(self):
47             alarm_time_str = self.entry.get().strip()
48             try:

```

```

47         self.alarm_time = datetime.strptime(alarm_time_str, "%H:%M")
48         self.entry.delete(0, tk.END) # Clear the entry field
49         self.entry.insert(0, self.alarm_time.strftime("%H:%M")) # Display formatted time in
entry
50         self.entry.config(state=tk.DISABLED) # Disable entry after setting alarm
51         self.set_alarm_button.config(state=tk.DISABLED) # Disable set alarm button
52         self.stop_alarm_button.config(state=tk.NORMAL) # Enable stop alarm button
53         self.alarm_active = True
54         messagebox.showinfo("Alarm Set", f"Alarm set for
{self.alarm_time.strftime('%H:%M')}")
55     except ValueError:
56         messagebox.showerror("Error", "Invalid time format. Please use HH:MM")
57
58     def check_alarm(self):
59         if self.alarm_active and datetime.now().time() >= self.alarm_time.time():
60             self.activate_alarm()
61
62     def activate_alarm(self):
63         self.alarm_active = False # Turn off the alarm
64         self.entry.config(state=tk.NORMAL) # Enable the entry field
65         self.set_alarm_button.config(state=tk.NORMAL) # Enable set alarm button
66         self.stop_alarm_button.config(state=tk.DISABLED) # Disable stop alarm button
67
68         # Display message
69         messagebox.showinfo("Alarm", "Wake up!")
70
71         # Play sound (Windows)
72         winsound.PlaySound("SystemAsterisk", winsound.SND_ALIAS)
73
74     def stop_alarm(self):
75         self.alarm_active = False # Turn off the alarm
76         self.entry.config(state=tk.NORMAL) # Enable the entry field
77         self.set_alarm_button.config(state=tk.NORMAL) # Enable set alarm button
78         self.stop_alarm_button.config(state=tk.DISABLED) # Disable stop alarm button
79
80     def run(self):
81         self.master.mainloop()
82
83 if __name__ == "__main__":
84     root = tk.Tk()
85     app = AlarmClockApp(root)
86     app.run()
87

```