LOVELY PROFESSIONAL UNIVERSITY

Project Report

ALARM TOOL USING GUI

For Course

PYTHON PROGRAMMING (INT213)

Submitted By

1.Name – Hemchand Chandravanshi

Registration Number – 12008563

2.Name – Sanjeev Kalyan Mitta

Registration Number - 12013417

Program – B TECH CSE

Semester – Third

Submitted To

Dr. Sagar Dhanraj Panday

Professor

Lovely Professional University

Jalandhar, Punjab, India

Table of Content:

Serial no	Name	Page no
1.	ABSTRACT	2
2.	INTRODUCTION	3
3.	TEAM MEMBERS & CONTRIBUTION	4
4.	METHODOLOGY	5 to 13
5.	WORKING OF ALARM TOOL	14
6.	Result Analysis	15
7.	Conclusion	15
8.	References	16

Abstract:

The main objective of the project is implementation of an Alarm Tool by using python. Now-a-days people are so busy in their daily life and they are forgetting their work at certain time in their day-to-day life by using the alarm they can set the timer for their personal works and will get the alarm when the time comes. The primary function of the alarm tool is to a awaken people from their night's sleep or short naps. Alarm tool are also used in mobile phones, watches, and computers.

Python consists of some very innovative libraries such as datetime and tkinter which help us to build the project using the current date and time. So, these is the purpose of selecting the alarm tool for our project.

1.1 Acknowledgment

I would like to thank my mentor-Prof. Sagar Pande for the advice and inputs on the project. Many thanks to my friends and seniors as well, who spent countless hours to listen and provide feedback.

INTRODUCTION:

2.1 Context

This project has been done a part of my course for the CSE at Lovely Professional University. Given by Sagar Pande. I have two months of time to fulfil the requirements in order to succeed our module.

2.2 Motivations

This project requires good knowledge of Python and GUI(Graphic User Interface).Begin extremely interested in python , the group project was a great occasion to give us the time to learn and confirm our interest In this field. In the group project we can learn, execute and get the experience while doing the project. This is the great opportunity to learn in this course**.**

2.3 Idea

As a first experience, we wanted to make my project as much didactic as possible by approaching every different step of the GUI process and trying to understand them deeply. we chose to take an Alarm Tool. The goal was to get alarm when the time comes which was set by the user.

TEAM MEMBERS:

1. HEMCHAND CHANDRAVANSHI (TEAM LEADER)

Contributions:

1. Coding(joined)
2. GUI(joined)
3. Project Report(joined)

2. SANJEEV KALYAN MITTA

Contributions:

1. Coding(joined)
2. GUI(joined)
3. Project Report(joined)

Methodology:

1. First, we had import all the necessary libraries and modules:

```
from tkinter import *
from tkinter import messagebox
import threading
import time
import datetime as dt
import vlc
import os
from PIL import ImageTk, Image
```

4.1 Libraries and modules.

Tkinter:

Tkinter is the **standard GUI library for Python**. Python when combined with Tkinter provides a fast and easy way to create GUI applications. It helps us to create a dialog box with any information that we want to provide or get from the users. So we have used Tkinter to show the Alarm Toll in effective graphical way.

MessageBox:

MessageBox widget is used to display the message boxes in the python applications. This module is used to display a message.

Threading:

Python threading module allows you to have different parts of your program run concurrently and can simplify your design.so we have used Thread-based parallelism to execute the functions (such as Snooz, Message, alarm_time, and CURRENT_TIME) concurrently.

Time:

This module provides various time-related functions. It has various function which are very useful to use current time in program. So we have used Time module to work with current time in alarm tool.

Datetime:

We needed to use some more functionalities for time which were not there in Time module so we have used Datetime module. Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals.

VLC:

Vlc media player is a free and open-source portable cross-platform media player software and streaming media server developed by the VideoLAN project. So we have used vlc.MediaPlayer() to play Alarm tone. It was not installed already in python. So in order to install vlc module in python we have used the command (“pip install python-vlc”) in command prompt.

OS:

The OS module in Python provides functions for interacting with the operating system. This module provides a portable way of using Operating system depending functionality. We have used os module to check the path of music file that weather the music file is there in system or not.

ImageTk:

We have used ImageTk for background image in Alarm Tool.

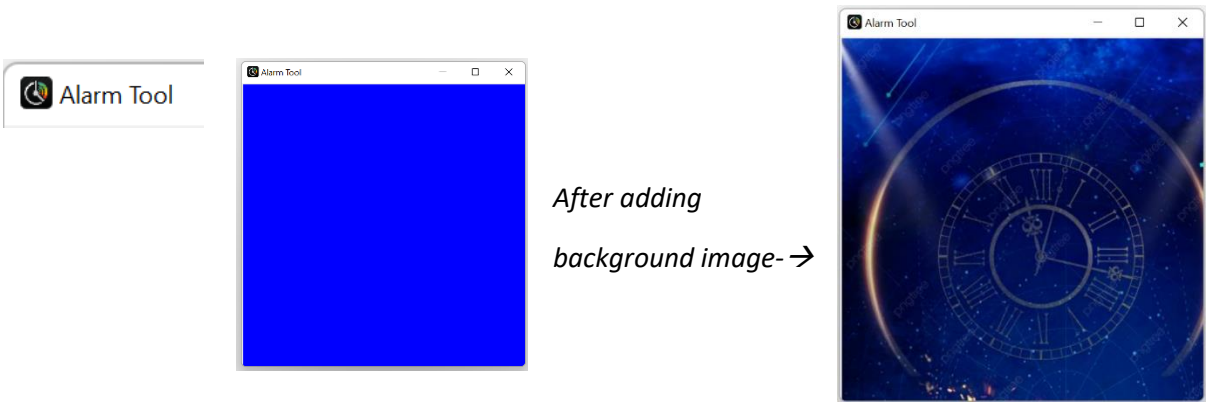
Iconbitmap:

iconbitmap(bitmap) sets the icon of the window/frame widget to bitmap . The bitmap must be an ico type. So we have used it for Alarm tool icon.

2. In second step we have created the GUI window, set the geometry, title and icon for Alarm Tool window. And we have created the canvas to add background image and various text into Alarm Tool window.

```
Alarm_Tool = Tk()
Alarm_Tool.geometry('400x400')
Alarm_Tool.title('Alarm Tool')
Alarm_Tool.iconbitmap('Alarm1.ico')
canv = Canvas(Alarm_Tool, width=400, height=400, bg='blue')
canv.grid(row=0,column=0)
img = ImageTk.PhotoImage(Image.open(r"image_724407.jpeg"))
canv.create_image(0, 0, anchor=NW, image=img)
```

4.2.1 Code for Tab length and width, icon and project title and for canvas.



4.2.2 Icon ,canvas with blue colour background and GUI window, canvas with background window.

In the above pic symbol is the icon and “Alarm Tool” is the project name.

3. Third step was to create alarm_time function to match alarm time with current time and then play the default or user entered Alarm Tone and show the alert message. It will call the Snooz function also. And to provide the Alarm_Time, snooz_time, and Tone this alarm_time function we have created the entries inside the tkinter Tk() function. That would be displayed after calling this alarm_time function.

```
def alarm_time():
    global STOP_ALARM
    STOP_ALARM = False
    global snooz_time
    global st
    global p
    try:
        st=snooz_time.get()
    except TclError:
        st=0
    Alarm_Time = f'{hour.get()}:{min.get()}:{sec.get()}'
    Tone = song.get()
    if os.path.exists(Tone):
        p = vlc.MediaPlayer(r"%s"%Tone)
    else:
        p = vlc.MediaPlayer(r'Awesomemorning Alarm.mp3')
    print(Alarm_Time)
    while True:
        if Can_alarm:
            break
        current_time=dt.datetime.now()
        Time=time.strftime("%H:%M:%S")
        Time1=f'{current_time.hour}:{current_time.minute}:{current_time.second}'
        if((Time == Alarm_Time) or (Time1 == Alarm_Time)):
            p.play()
            time.sleep(0.1)
            threading.Thread(target=Message).start()
            threading.Thread(target=Snooz).start()
            break
```

4.3 Code : to match alarm time with current time and play alarm

3.1 In the above code the snooz_time, Tone and Alarm_Time will be taken by following entries which are created inside Tk() function.

These entries are created by following code inside tkinter Tk() function using different widgets :

(i). For Alarm_Time :

```
canv.create_text(64,22, text="Set Alarm :", fill ="white" , font=('Arial',16,'bold'))

canv.create_text(115,50, text="Hour   Min   Sec", fill ="white" , font=('Arial',15,'bold'))
canv.create_text(149,101, text="Recommendation : Enter in 24 hour format", fill ="white" , font=('Arial',10))

hour = StringVar()
min=StringVar()
sec=StringVar()

En1=Entry(Alarm_Tool,textvariable=hour,font=10,width=5)
En1.place(x=27,y=65)

En2=Entry(Alarm_Tool,textvariable=min,font=10,width=5)
En2.place(x=87,y=65)

En3=Entry(Alarm_Tool,textvariable=sec,font=10,width=5)
En3.place(x=147,y=65)
```

Output:

4.3.1 Entry for Alarm_Time.

Recommendation for Set Alarm entry: The user need to enter the time in 24 hour format, either in zero padded decimal number (like – 07:33:03) or without zero padded (like – 7:33:3) , do not mix both (like – 07:33:3).

(ii). For Tone:

```
song=StringVar()

canv.create_text(103,140, text="Enter Alarm Tone :",fill="white",font=('Arial',16,'bold'))
En4=Entry(Alarm_Tool,textvariable=song,font=10)
En4.place(x=27,y=154)
canv.create_text(211,190, text="Recommendation : Enter full path of file without quotation marks", fill = "white",font=('Arial',10))
```

Output:

4.3.2 Entry for Tone.

Recommendation: for this entry the user need to enter the complete path of music file without quotation marks, if user will not remove quotation marks then the default tone will play.

(iii). For snooz_time:

```
snooz_time = IntVar()

canv.create_text(105,230, text="Enter Snooz Time :",fill="white",font=('Arial',16,'bold'))
En5=Entry(Alarm_Tool,textvariable=snooz_time,font=10)
En5.place(x=27,y=246)
canv.create_text(131,283, text="Recommendetion : Enter in second ",fill="white",font=('Arial',10))
```

Output:

4.3.3 Entry for snooz_time.

Recommendation: in this entry the user need to enter the time in second

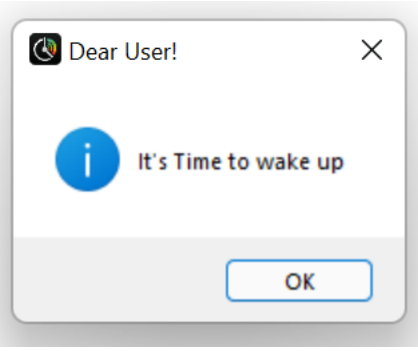
4. Our 4th step was to create the function Message() and Snooz() which will be call from inside the alarm_time() function. And after it they will run independently(means without termination of these functions another function will also get control to start).

```
def Message():
    messagebox.showinfo("Dear User!","It's Time to wake up")

def Snooz():
    global st
    if st!=0:
        if STOP_ALARM:
            return
        time.sleep(15)
        p.stop()
        time.sleep(st)
        if STOP_ALARM:
            return
        p.play()
        Snooz()
    if st==0:
        for i in range(7):
            if STOP_ALARM:
                return
            time.sleep(20)
            p.stop()
            if STOP_ALARM==True or i==5:
                return
            p.play()
```

4.4.1 Code : to show message box and start snooze

In the above code message box will show at the time of alarm and if the user will enter the snooze time then the alarm will play for 15 sec(we have used less time so that while presentation we can show it quickly) and after snooze time it will again play for 15 sec. and it will continuously done till the user will not stop it. And if user will not enter the snooz time then the alarm will play for 2 minute and it will get turn off after that.



4.4.2 Output of message box .

4.4.3 entries for snooz time

5. Our 5th step was to create function for current time : Current_Time() and show it on the Alarm tool window, this function will run independently from all functions. We have used threading to call this function so that it should run concurrently.

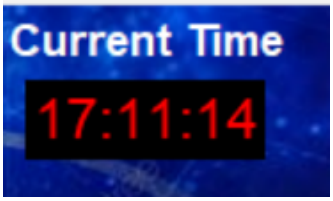
```
def CURRENT_TIME():
    TIME = time.strftime("%H:%M:%S")
    lbl1.config(text=TIME)
    lbl1.after(1000,CURRENT_TIME)
```

This function will config the label which is created inside Tk().

```
canv.create_text(300,15,text="Current Time",fill ="white",font=('Arial',13,'bold'))
lbl1=Label(Alarm_Tool, fg='red',bg='black',font=10)
canv.create_window(300,45,window=lbl1)
threading.Thread(target=CURRENT_TIME).start()
```

This above code will create the text in canvas as “Current time” and it will create label for time . And this code will call the CURRENT_TIME() function also, we must call the CURRENT_TIME function after creating the label and text otherwise it will cause error.

Output of above two code :



6. Our 6th step was to create the Stop_alarm() and Cancel_alarm() functions and create Buttons for it in Alarm tool window . So that the we can stop or cancel the alarm.

Firstly we have created functions:

```
def Cancel_alarm():
    global Can_alarm
    Can_alarm = True

def Stop_alarm():
    global STOP_ALARM
    STOP_ALARM= True
    p.stop()

STOP_ALARM = False
```

To call these functions we have created buttons inside Tk() and we will give function call as command to these buttons

```
SetBtn1=Button(Alarm_Tool, text='Stop',fg='white',bg='red',font=('Arial',15,'bold'),width=5,height=1,command=Stop_alarm)
SetBtn1.place(x=141,y=315)
CnclBtn=Button(Alarm_Tool, text='Cancel Alarm',fg='white',bg='red',font=('Arial',15,'bold'),height=1,command=Cancel_alarm)
CnclBtn.place(x=250,y=315)
```

Output:

Now if the user want to stop the alarm after Alarming then they need to click on Stop button and if they have just set the alarm and want to cancel it before Alarm Time then they need to click on Cancel Alarm button.

7. Our 7th step was to create the Save button and provide command to this button as function call for Al_Tool(), inside this function alarm_time() funciton (using thread module) will call to save alarm and ring at time.

```
Can_alarm = False

def Al_Tool():
    global Can_alarm
    threading.Thread(target=alarm_time).start()
    Can_alarm = False
    return
```

7.1 Code for Al_tool() function and to call alarm_time().

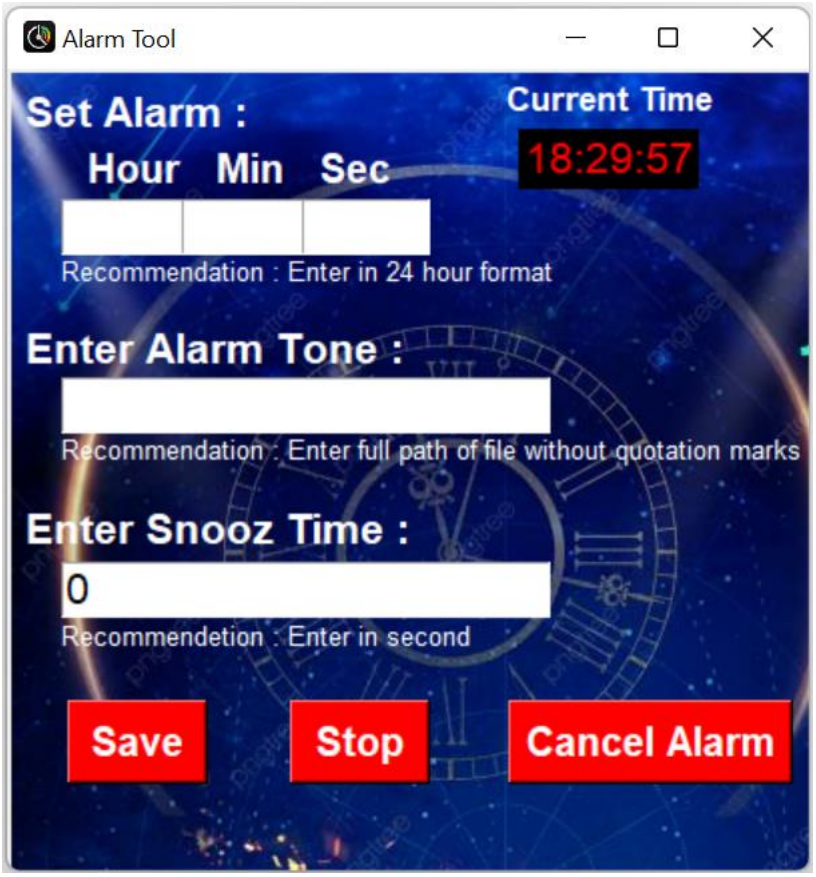
Button to call above function

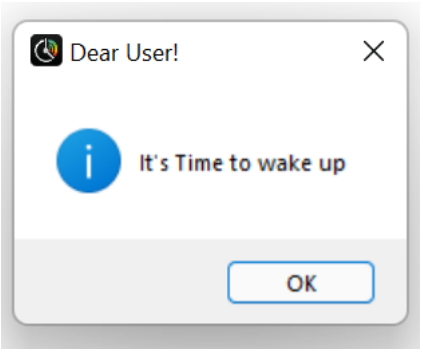
```
SetBtn=Button(Alarm_Tool, text='Save',fg='white',bg='red',font=('Arial',15,'bold'),width=5,height=1,command=Al_Tool)
SetBtn.place(x=30,y=315)
```

Output :

After entering the alarm time user can save the alarm using this save button in Alarm Tool window.

Final window of Alarm Tool :





- When alarm ring

Recommendation : Reside all the functions (alarm_time, Snooz, Message, CURRENT_TIME, Stop_alarm, Cancel_alarm, and Al_Tool) inside the Tk() function for better performance.

WORKING OF ALARM TOOL

How this application works?

1. After running the code the alarm tool will appear and the user need to enter the alarm time in 24 hour format.

2. Now If the user want to change the default Alarm Tone then they need to enter the complete path of desired Tone without quotation marks.

3. If user want to skip the changing Alarm Tone then they can skip it , the default Alarm Tone will play at the time of Alarm.

4. Now If the user want Snooze Time then they can enter the Snooze Time in second (we have given the second access to the code so that during presentation we can show the working of application quickly.)

5. If user want to skip this snooze they can skip the alarm will play for 2 minute continuously and it will stop after it.

6. Now user need to press the Save button and alarm time will save and alarm will ring at the time of alarm.

7. During playing the alarm tone by app if user want to stop it then they need to Click on Stop button.

8. After saving the alarm now if user want to cancel alarm before the alarm time then they need to click on Cancel Alarm button and the alarm will cancel.

Result Analysis:

Result Analysis means help us to understand how the user are learnings, and how we have performed in each area of our project.

1.In this process we have learnt how we need to take the steps from the starting. It is a new experience for us and we had learnt our mistakes and rectified it by discussing each other.

2. We have implemented our project in IDLE and few of the commands were written to install the items which are required for our project.

Conclusion:

With this project in Python, we have successfully made the Alarm Clock. We used the popular GUI library for rendering graphics on a display window. We learned how to extract the current time from the computer and to use it for manipulation using the Date Time library. This way we can set an alarm in the computer interface using python programming which rings with the default machine sound for Windows.

By this Alarm Tool the user can implemented in the laptops or computers and we are so exited when we run the project and when we see the output.

Thanks for our course and our lecture Sagar Pande for conducting the projects for getting the experience and knowledge.

References:

1. ***docs.python.org***
2. ***www.stackoverflow.com.python***
3. ***www.geeksforgeeks.org.python***