CORE TASK 4

Name :- Shaikh Vasim Farukdin

Employe ID :- TEN/PD/1304

Position :- Python Developer – Associate

Alarm Clock with GUI Python

- It is no doubt that an alarm clock is always handy to alert us whenever we Sleep, take a short nap, or to remind us about the work, we always get oblivious about.

- Our ancestors have been using an alarm clock, going back to its 2,000 years long history but over time, the new advancements in technologies allow us to keep an alarm clock without it containing a dial, gear trains, etc. How? Let's find out further.

Develop an Alarm Clock using Python with GUI

About the Python Project:

- The objective of our project is to implement an alarm clock using Python.
- 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 as well as to provide a user interface to set the alarm according to the requirement in 24-hour format.

Prerequisites:

- This project requires good knowledge of Python and GUI (Graphic User Interface).
- Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.
- All the modules used need not be downloaded beforehand like the other libraries like NumPy, thus this project will be user friendly and accessible in any virtual environment used for python programming.

Project File Structure:

- First, let's check the steps to build an Alarm Clock program in Python:
- Importing all the libraries and modules required
- Putting forward a while loop which takes the argument of the time, the user wants to set the alarm on and automatically breaks when the time is up, with sound
- Create a display window for user input.

Modules Used:

- Tkinter module belongs to a standard library of GUI in Python. It helps us to create a dialog box with any information that we want to provide or get from the users.

- Datetime and time modules in python help us to work with the dates and time of the current day when the user is operating python and to manipulate it too.

Github Code :- https://github.com/vasim-717/Alaram-clock-ten.git

.

This is a code

```
from tkinter import *
from tkinter import ttk
import time
import os
from tkinter import messagebox

root = Tk()
root.title("Vasim Alarm clock for TEN")
root.geometry("350x200+510+200")


def SubmitButton():
    AlarmTime = entry1.get()
    Message1()
    CurrentTime = time.strftime("%H:%M")
    print("the alarm time is: {}".format(AlarmTime))
    while AlarmTime != CurrentTime:
        CurrentTime = time.strftime("%H:%M")
        time.sleep(1)
    if AlarmTime == CurrentTime:
        print("now Alarm Musing Playing")
        os.system("start alarm-music.mp3")
        label2.config(text="Alarm music playing.....")
        messagebox.showinfo(title='Alarm Message',
                            message="{}".format(entry2.get()))


def Message1():
    AlarmTimeLable = entry1.get()
    label2.config(text="the Alarm time is Counting...")
    messagebox.showinfo(title='Alarm clock',
                        message='Alarm will Ring at
{}'.format(AlarmTimeLable))


frame1 = ttk.Frame(root)
```

```
frame1.pack()
frame1.config(height=100, width=100)

label1 = ttk.Label(frame1, text="Enter the Alarm time :")
label1.pack()


entry1 = ttk.Entry(frame1, width=30)
entry1.pack()
entry1.insert(3, "Type here  (Ex-18:15)")

labelAlarmMessage = ttk.Label(frame1, text="Alarm Message:")
labelAlarmMessage.pack()

entry2 = ttk.Entry(frame1, width=30)
entry2.pack()

button1 = ttk.Button(frame1,
                     text="submit", command=SubmitButton)
button1.pack()
label2 = ttk.Label(frame1)
label2.pack()


root.mainloop()
```

Output:-