

PROJECT REPORT
ON
Countdown Clock Timer



REPORT SUBMITTED
TO
VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE
FOR THE PBL OF ITW WORKSHOP - 2
BY:

1	Khushal Anand Malu	231037
2	Anurag Abhay Pande	231044

Class: SY - A

Batch: A2

Batch Teacher:
Prof. Suruchi Dedgaonkar Mam

ABSTRACT:

Our topic for the project-based learning (PBL) for Python Engineering is Countdown Clock Timer and to display the clock.

The objective of this python project is to create a countdown timer and display time. In this python project, the user sets the time and by click on the start button, it starts the count from that time. When the time gets over is when the countdown timer finishes display a pop-up window to the user.

In this project, the user can also use stopwatch where user can start, stop and reset the countdown clock according to user's need.

The project has been kept simple so that it won't be difficult for user to use our program avoiding complex structure.

This project build with basic concept of python and libraries like Tkinter, time, Pillow and datetime.

This project was made using Visual studio and Spyder platform which is an open-source cross platform.

INTRODUCTION AND THEORY:

1]Why did we use Tkinter?

Ans:

1-Tkinter is the standard GUI library for Python. 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.

2- Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, **tkinter** is the most commonly used method.

2]Advantages of Tkinter:

1-Easy to use.

2-Easy to understand.

3-Time efficient.

4-User friendly

3] Why did we use Pillow?

Ans: Pillow is a **Python** Imaging Library (PIL), which adds support for opening, manipulating, saving images and also import images.

4] The functions of **time** module are:

It allows us to handle various operations regarding **time**, its conversions and representations, which find its use in various applications in life

Aim of the project:

1-A user should set a countdown timer.

2-A user can be able to see the current time.

3-If the user wants to access the stopwatch, he could get easy access to it.

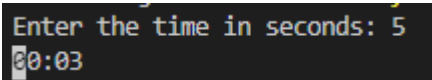
FLOW OF PROGRAM:

STEP 1: By using sleep function from time module, we created a simple countdown clock which return “Time is Up” text after set seconds.

<pre>1 import time as t 2 seconds = int(input("Set a time: ")) 3 for i in range(seconds): 4 print(str(seconds-i) + " seconds remaining \n") 5 t.sleep(1) 6 print("Time is up")</pre>	<pre>Set a time: 3 3 seconds remaining 2 seconds remaining 1 seconds remaining Time is up</pre>
--	--

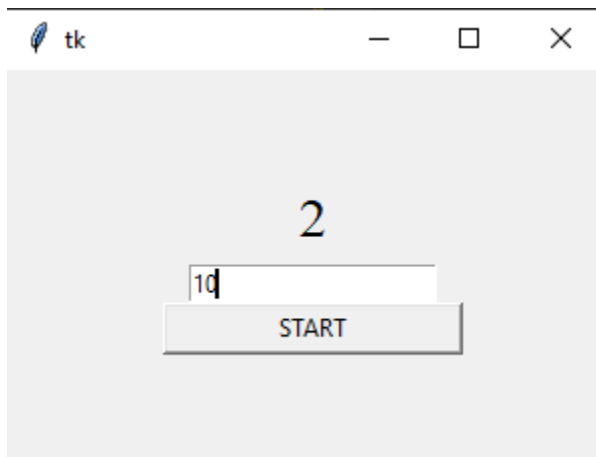
STEP 2: We then updated our clock and define a function called countdown. By using simple while loop we made enhanced our output.

```
1 import time
2
3 def countdown(t):
4
5     while t:
6         mins, secs = divmod(t, 60)
7         timer = '{:02d}:{:02d}'.format(mins, secs)
8         print(timer, end="\r")
9         time.sleep(1)
10        t -= 1
11
12    print('Time is Up!!')
13
14 t = input("Enter the time in seconds: ")
15 countdown(int(t))
```

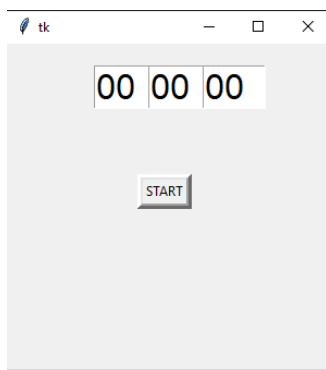


Enter the time in seconds: 5
00:03

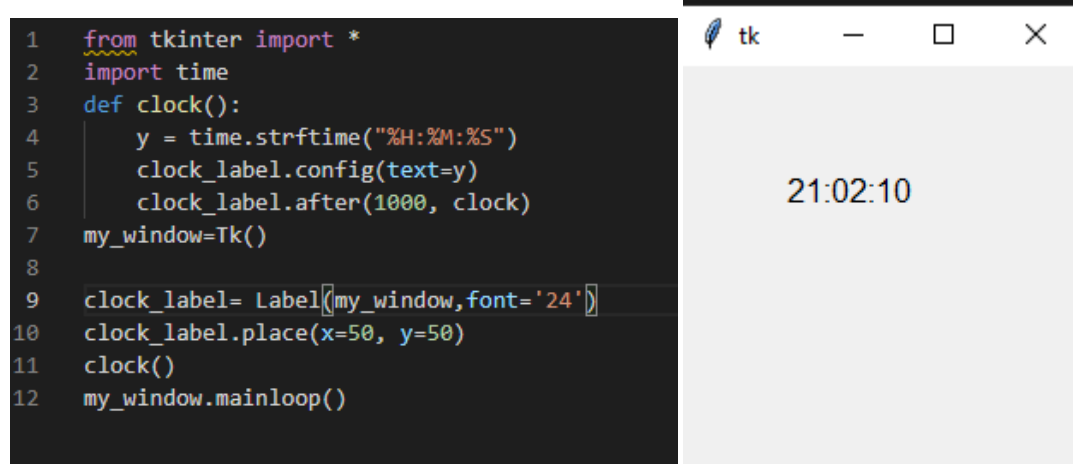
STEP 3: Then imported Tkinter which is GUI library used for graphics and then learned Tkinter and then applied graphics to the countdown clock.



STEP 4: Then by using different entry function from Tkinter we make user able to input time in hour, min and second format.

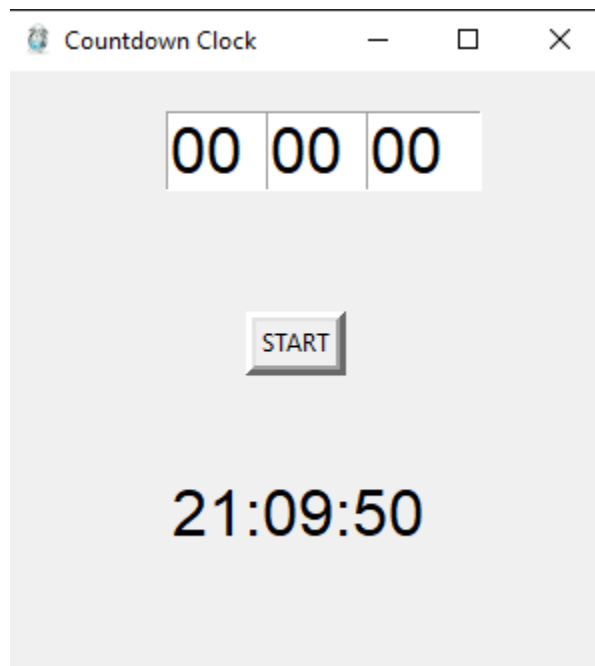


STEP 5: After this we made a clock which displays time and applied graphics to it as well.

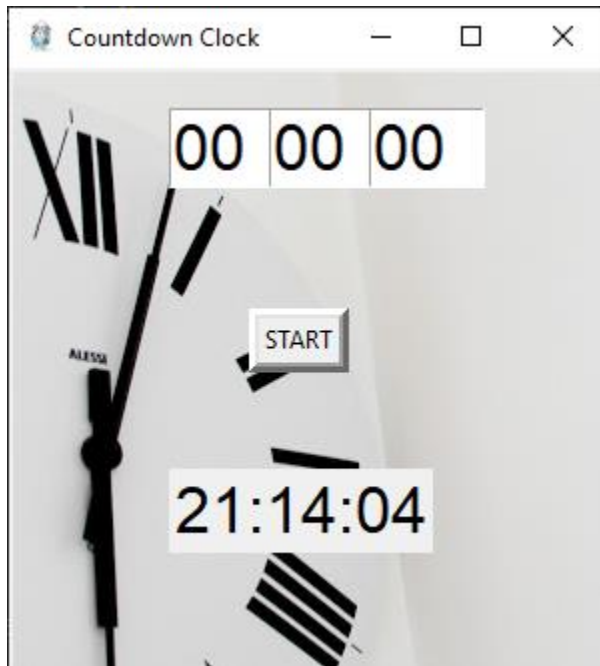


STEP 6: This step we add title and made an icon for a project by using (iconbitmap).

STEP 7: This step was the combination of step 2 and step 3 means that we mixed the countdown clock and the clock to display it together.



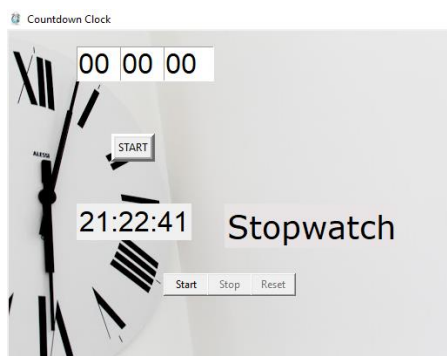
STEP 8: For background image, we first imported PIL (pillow library) and put an image in the background to make it attractive.



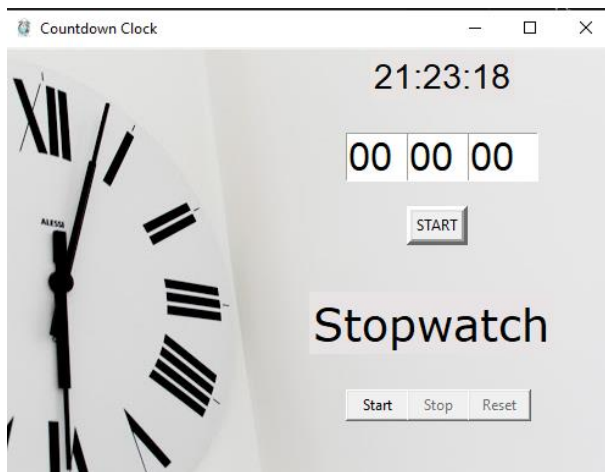
STEP 9: Now we prepared a stop watch.



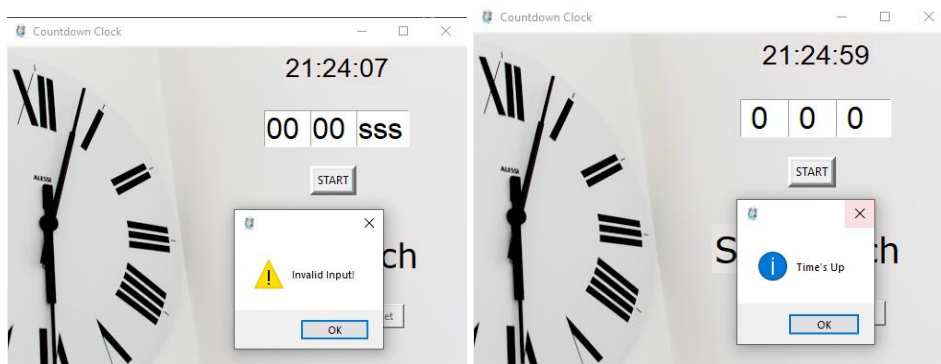
STEP 10: Then we added out stopwatch in main program.



STEP 11: Now, our code is ready but the table was overlapping the background image, so by adjusting the co-ordinates we placed the table in a correct position where it did not overlap.



STEP 12: Then we imported messagebox for printing the message for “invalid input” and “time is up”.



SOURCE CODE OF THE PROGRAM:

```
import time
from tkinter import *
from tkinter import messagebox
from PIL import ImageTk, Image
from datetime import datetime

counter = 66600
running = False

def clock():
    y = time.strftime("%H:%M:%S")
    clock_label.config(text=y)
    clock_label.after(1000, clock)

def startCountdown():
    try:
        userinput = int(hour.get())*3600 + int(minute.get())*60 + int(second.get())
    except:
        messagebox.showwarning('', 'Invalid Input!')
    while userinput > -1:
        mins, secs = divmod(userinput, 60)

        hours=0
        if mins >60:

            hours, mins = divmod(mins, 60)

        hour.set("{0:2d}".format(hours))
        minute.set("{0:2d}".format(mins))
        second.set("{0:2d}".format(secs))

        root.update()
        time.sleep(1)

        if (userinput == 0):
            messagebox.showinfo("", "Time's Up")
            userinput -= 1

def counter_label(label):
    def count():
        if running:
```

```

        global counter

        if counter==66600:
            display="Starting..."
        else:
            stamp = datetime.fromtimestamp(counter)
            string = stamp.strftime("%H:%M:%S")
            display=string

        label['text']=display
        label.after(1000, count)
        counter += 1

    count()

def Start(label):
    global running
    running=True
    counter_label(label)
    start['state']='disabled'
    stop['state']='normal'
    reset['state']='normal'

def Stop():
    global running
    start['state']='normal'
    stop['state']='disabled'
    reset['state']='normal'
    running = False

def Reset(label):
    global counter
    counter=66600

    if running==False:
        reset['state']='disabled'
        stopwatch_label['text']='Stopwatch'

    else:
        stopwatch_label['text']='Starting...'

root = Tk()
root.geometry("500x350")

```

```

root.title("Countdown Clock")

root.iconbitmap("Clock.ico")
myimg = ImageTk.PhotoImage(Image.open('bg.jpg'))

bgimage = Label(root, image = myimg)
bgimage.place(x=0 , y =0)

clock_label= Label(root,font='Ariel 21', bg='#e8e4e4')
clock_label.place(x=300, y=5)
clock()

hour=StringVar()
minute=StringVar()
second=StringVar()

hour.set("00")
minute.set("00")
second.set("00")

fv = ("Arial",24)

hour_tf= Entry(root, width=3, font=fv, textvariable=hour)
hour_tf.place(x=280,y=70)

mins_tf= Entry(root, width=3, font=fv, textvariable=minute)
mins_tf.place(x=330,y=70)

sec_tf = Entry(root, width=3, font=fv, textvariable=second)
sec_tf.place(x=380,y=70)

start_btn = Button(root, text='START', bd='5', command= startCountdown)
start_btn.place(x = 330,y = 130)

stopwatch_label = Label(root, text="Stopwatch", fg="black", font="Verdana
28", bg='#e8e4e5')
stopwatch_label.place(x = 250,y = 200)

start = Button(root, text='Start', width=6, command=lambda:Start(stopwatch
_label))
start.place(x=280,y=280)

stop = Button(root, text='Stop',width=6,state='disabled', command=Stop)

```

```
stop.place(x=330,y=280)

reset = Button(root, text='Reset',width=6, state='disabled', command=lambda  
a:Reset(stopwatch_label))
reset.place(x=380,y=280)

root.mainloop()
```

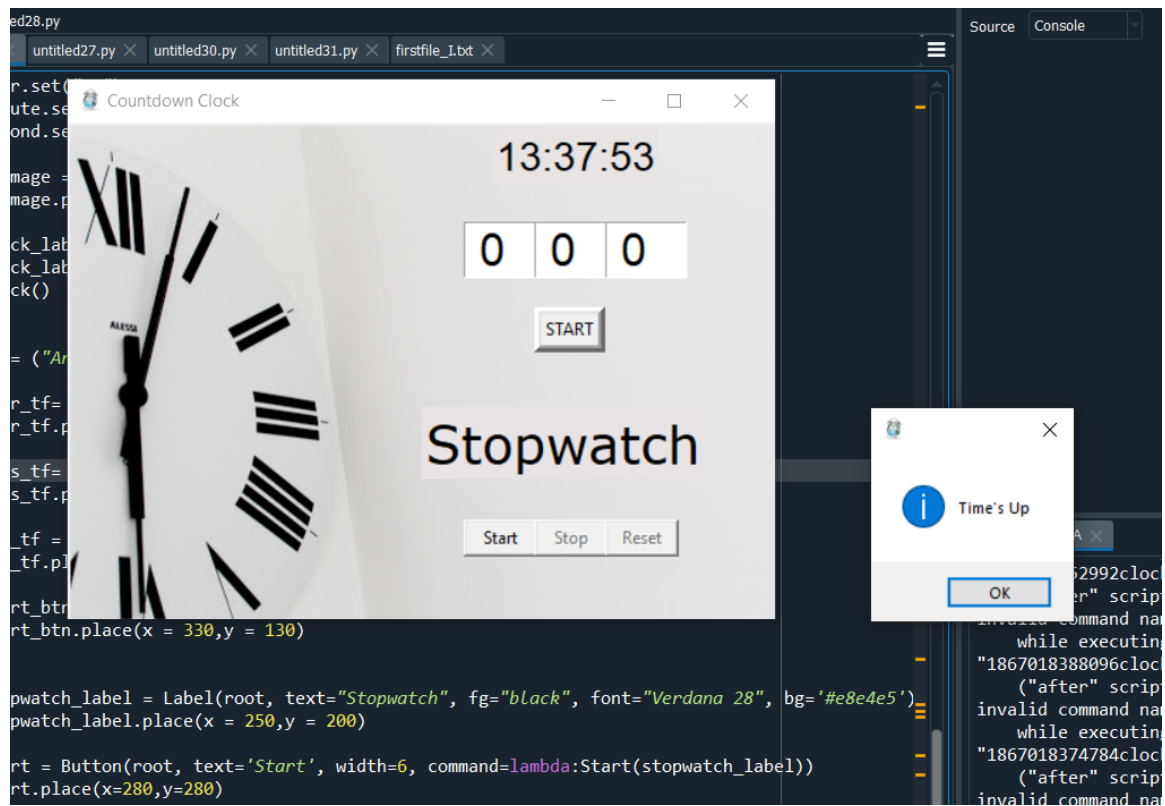


Clock.ico



bg.jpg

OUTPUT:



LIST OF THE TOPICS/ CONCEPT WHICH ARE COVERED :

- 1. Functions.**
- 2. Pillow (PIL library)**
- 3. Using of Conditional statements (if , else, elif)**
- 4. Declaring a string variable.**
- 5. Divmod operator**
- 6. Importing different modules which are needed in project.**
- 7. Using of Tkinter for graphics.**
- 8. Using messagebox module for printing messages like Invalid input etc.**
- 9. Importing icon for our project by using iconbitmap.**
- 10.Using the entry widget to make entries like set an alarm.**
- 11.Using button widget to create a button in the table that gives command.**
- 12.How to use the co-ordinates to change the alignment of the table.**

CONCLUSION:

Hence Successfully completed and executed the project “Countdown Clock” which shows the count down .

REFERENCES:

<https://docs.python.org/3/library/time.html>

<https://docs.python.org/3/library/tkinter.html>

<https://docs.python.org/3/library/datetime.html>

<https://tkdocs.com/tutorial/index.html>

<https://wiki.python.org/moin/TkInter>

https://www.tutorialspoint.com/python/python_gui_programming.htm

<https://www.youtube.com/watch?v=YXPyB4XeYLA>