INDEX

PROJECT DESCRIPTION AT A GLANCE

1. INTRODUCTION:

An alarm clock usually requires you to set the specific time you want the alarm to ring. Once you have set your preferred time, the alarm will continuously match the time you provided with the current time. As soon as both the time matches, the alarm rings.

This is a very general idea of how usually a real alarm clock works. The alarm clock I have built follows the same mechanism.

2. WORKING DESCRIPTION:

The programme opens a new window initially where 3 options are provided.

- Alarm
- Countdown
- Stopwatch

The user can choose any of the options by clicking on the button. As soon as it is selected another respective window opens.

a) Alarm:

The user can enter the time at which he/she wants to set alarm in 24-hour format (hour: min: sec). There is also an option for taking notes. The note taken will be displayed as soon as the alarm rings as a notification.

Also, all the notes taken from time to time along with the date and time of alarm will be saved in a csv file, so as to keep a track of your activities.

b) Countdown:

User can set a timer in the countdown. As soon as the timer reaches zero, the user will be notified by the notification centre.

c) Stopwatch:

This window starts with “Welcome”, and contains three buttons, viz. Start, Stop and Reset.

As the user presses ‘Start’ button, the message displayed as “Starting…” indicates that stopwatch has started. ‘Stop’ can be used to pause the stopwatch whereas ‘Reset’ sets the stopwatch back to zero.

3. DEMAND IN REAL WORLD:

Alarm clocks are pretty useful timepieces and have become extremely popular among the common masses. As you may know, in this modern age, it is difficult to find time to do certain jobs due to extreme heavy work pressure and constant tensions on different matters human beings face on daily basis. This sometimes leads to short term memory loses and people tend to forget many things about their daily activities or work-related matters. Sometimes, due to extreme tiredness, a person may fall deep asleep and fail to wake up during a specific time to undergo some sort of unfinished work. Alarm clocks help in these cases. They are a sort of signalling machine which may help the person to get awake from the deep sleep to continue with the unfinished works.

This multipurpose alarm clock not only sets alarm but also has applications like countdown and stopwatch.

4. FUTURE OF THE PROJECT:

Intelligent alarm management continues to attract the interest and application in various industry sectors. The system can be improved by setting up voice featured interactive alarm clocks, stopwatch and countdown which will work on voice commands provided by the user.

5. CONCLUSION:

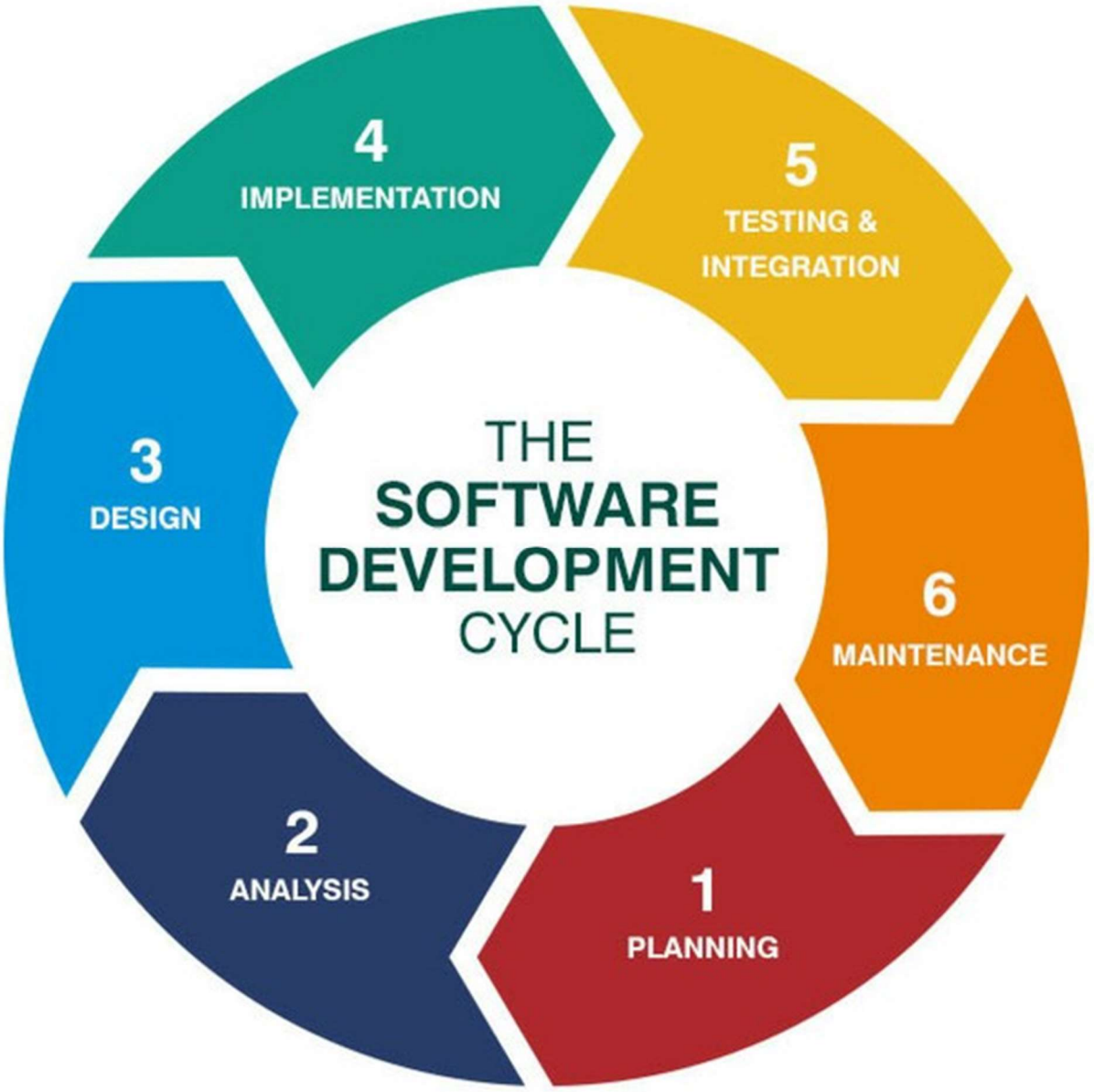
Clock combines all of the functionality you need into one simple package. Set alarms; while taking notes and keeping a track of all your notes from every new alarm setting, add timers, and run a stopwatch all at a single place.

The notes managing system adds all the notes taken to a csv file for future reference of your records.

HARDWARE AND SOFTWARE REQUIREMENTS

- Operating system: Microsoft Windows 10
- Processor: Intel(R) Core(TM) i3-6100U CPU@2.30GHz, 2301 Mhz, 2 Core(s), 4 Logical Processor(s)
- RAM: 4.00 GB
- Disk free space: 20.4 GB
- System type: 64-bit Operating System, x64-based processor

SOFTWARE DEVELOPMENT LIFE CYCLE



SOURCE CODE

```
#importing necessary modules
from tkinter import *
from tkinter import messagebox
import tkinter as Tkinter
from datetime import datetime
import datetime
import time
import winsound
from plyer import notification


root=Tk()       #to create a windows
root.configure(background='salmon1')     #to set background
color
root.geometry("250x250")        #to fix size
root.title("Multipurpose Alarm Clock")       #title of window

#StopWatch
def stop():

   counter=66600
   running=False
   def counter_label(label):
      def count():
         if running:
            global counter

            # To manage the intial delay.
            if counter==66600:
```

```
            display="Starting..."
        else:
            tt=datetime.fromtimestamp(counter)
            string=tt.strftime("%H:%M:%S")
            display=string

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

    # Triggering the start of the counter.
    count()

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


# Stop function of the stopwatch
def Stop():
    global running
    start['state']='normal'
    stop['state']='disabled'
    reset['state']='normal'
    running=False
```

```
    # Reset function of the stopwatch
    def Reset(label):
        global counter
        counter=66600

        # If rest is pressed after pressing stop.
        if running==False:
            reset['state']='disabled'
            label['text']='Welcome!'

            # If reset is pressed while the stopwatch is running.
        else:
            label['text']='Starting...'

    r1=Toplevel(root)
    r1.title("Stopwatch")
    r1.configure(background='SlateBlue1')

    # Fixing the window size.
    r1.minsize(width=250, height=70)
    r1.iconbitmap(r"C:\Users\Anushka
Dwivedi\OneDrive\Desktop\Project cs 2021\stopwatch.ico")
    label=Tkinter.Label(r1, text="Welcome!", fg="black",
font="Verdana 30 bold", bg="SlateBlue1")
    label.pack()
    f=Tkinter.Frame(r1)
    start=Tkinter.Button(f, text='Start', width=6,
command=lambda:Start(label))
```

```
    stop=Tkinter.Button(f, text='Stop',width=6,
state='disabled', command=Stop)
    reset=Tkinter.Button(f, text='Reset',width=6,
state='disabled', command=lambda:Reset(label))
    f.pack(anchor='center',pady=5)
    start.pack(side="left")
    stop.pack(side="left")
    reset.pack(side="left")
    r1.mainloop()

#Alarm Clock
def alarmCall():
    r2=Toplevel(root)

    # The Variables we require to set the alarm(initialization):
    hour = StringVar()
    min = StringVar()
    sec = StringVar()
    text=StringVar()
    def alarm(set_alarm_timer):
        while True:
            time.sleep(1)
            current_time=datetime.datetime.now()
            now=current_time.strftime("%H:%M:%S")
            date=current_time.strftime("%d/%m/%Y")

            if now==set_alarm_timer:

winsound.PlaySound("sound.wav",winsound.SND_ASYNC)
                break
```

```
    def actual_time():
        set_alarm_timer=f"{hour.get()}:{min.get()}:{sec.get()}"
        alarm(set_alarm_timer)
        note=str(text.get())
        notification.notify(
        title="Alarm Clock",
        message=("Time to Wake up!\n"+note),
        app_icon=r'C:\Users\Anushka
Dwivedi\OneDrive\Desktop\Project cs 2021\clock.ico',
        timeout=10
        )

        #Saving Notes
        def Note():
            import datetime
            Date=datetime.date.today()
            Time=datetime.datetime.now().time()
            Message=str(text)

            import csv
            with open('NotesTaken.csv', mode='a', newline='') as
file:
                writer=csv.writer(file, delimiter=',', quotechar='"' )
                writer.writerow([Date, Time, Message])
            file.close()
        Note()
```

```
    r2.title("Alarm")
    r2.iconbitmap(r"C:\Users\Anushka
Dwivedi\OneDrive\Desktop\Project cs 2021\clock.ico")
    r2.geometry("520x250")
    r2.configure(background='lemon chiffon')
    time_format=Label(r2, text= "Enter time in 24 hour
format!",
fg="red",bg="black",font="Arial").place(x=165,y=140)
    addTime1 = Label(r2,text = "Hour",font=60,bg='lemon
chiffon')
    addTime2 = Label(r2,text = "Min",font=60,bg='lemon
chiffon')
    addTime3 = Label(r2,text = "Sec",font=60,bg='lemon
chiffon')
    setYourAlarm = Label(r2,text = "When to wake you
up",fg="blue",font=("Helevetica",10,"bold"))
    note=Label(r2,text="  Take Note
",fg="blue",font=("Helevetica",10,"bold"))


    #Time required to set the alarm clock:
    hourTime= Entry(r2,textvariable = hour,bg =
"pink",width=3)
    minTime= Entry(r2,textvariable = min,bg =
"pink",width=3)
    secTime = Entry(r2,textvariable = sec,bg =
"pink",width=3)
    messageinput=Entry(r2,textvariable = text,bg =
"pink",width=30)
```

```
    #To take the time input by user:
    submit = Button(r2,text = "Set Alarm",fg="red",width =
10,command = actual_time).place(x=230,y=180)

    #grid
    hourTime.grid(row=5, column=2, padx=10, pady=10)
    minTime.grid(row=5, column=3, padx=10, pady=10)
    secTime.grid(row=5, column=4, padx=10, pady=10)
    messageinput.place(x=174,y=97)
    note.grid(row=6, column=1, padx=10, pady=10)
    addTime1.grid(row=3, column=2, padx=10, pady=10)
    addTime2.grid(row=3, column=3, padx=10, pady=10)
    addTime3.grid(row=3, column=4, padx=10, pady=10)
    setYourAlarm.grid(row=5, column=1, padx=10, pady=10)



    r2.mainloop()
    #Execution of the window.



#Countdown
def CountDown():
    r3=Toplevel(root)
    r3.geometry("300x250")              #setting geometry of tk
window
    r3.title("Countdown")            #Title of the window
    r3.configure(background='OliveDrab1')

    #Declaration of variables
```

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

    #setting the default value as 0
    hour.set("00")
    minute.set("00")
    second.set("00")

    #Use of Entry class to take input from the user
    hourEntry= Entry(r3, width=3, font=("Arial",18,""),
                     textvariable=hour)
    hourEntry.place(x=80,y=20)

    minuteEntry= Entry(r3, width=3, font=("Arial",18,""),
                       textvariable=minute)
    minuteEntry.place(x=130,y=20)

    secondEntry= Entry(r3, width=3, font=("Arial",18,""),
                       textvariable=second)
    secondEntry.place(x=180,y=20)


    def submit():
        try:
                # the input provided by the user
                # stored in here :temp
            temp = int(hour.get())*3600 + int(minute.get())*60 +
int(second.get())
        except:
```

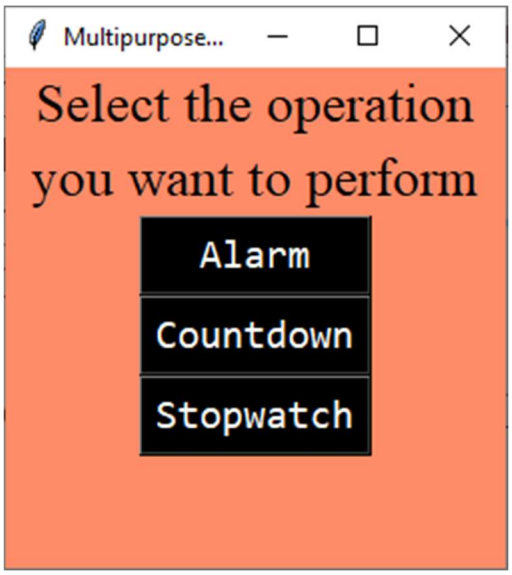
```
            print("Please input the right value")
        while temp >-1:

                # divmod(firstvalue = temp//60, secondvalue =
temp%60)
            mins,secs = divmod(temp,60)

                # Converting the input entered in mins or secs to
hours,
                # mins ,secs(input = 110 min --> 120*60 = 6600
=> 1hr :
                # 50min: 0sec)
            hours=0
            if mins >60:

                    # divmod(firstvalue = temp//60, secondvalue
                    # = temp%60)
                hours, mins = divmod(mins, 60)

                # using format () method to store the value up to
                # two decimal places
            hour.set("{0:2d}".format(hours))
            minute.set("{0:2d}".format(mins))
            second.set("{0:2d}".format(secs))

                # updating the GUI window after decrementing
the
                # temp value every time
            r3.update()
            time.sleep(1)
```

```
            # when temp value = 0; then a messagebox pop's
up
            # with a message:"Time's up"
            if (temp==0):
                messagebox.showinfo("Time Countdown",
"Time's up ")
                #countdown notification
                notification.notify(
                title="Countdown",
                message="Time's up!!",
                app_icon=r'C:\Users\Anushka
Dwivedi\OneDrive\Desktop\Project cs 2021\timer.ico',
                timeout=10
                )

            # after every one sec the value of temp will be
decremented
            # by one
            temp -= 1

    # button widget
    btn = Button(r3, text='Set Time Countdown', bd='5',
                 command= submit)
    btn.place(x = 70,y = 120)

    # infinite loop which is required to
    # run tkinter program infinitely
    # until an interrupt occurs
```

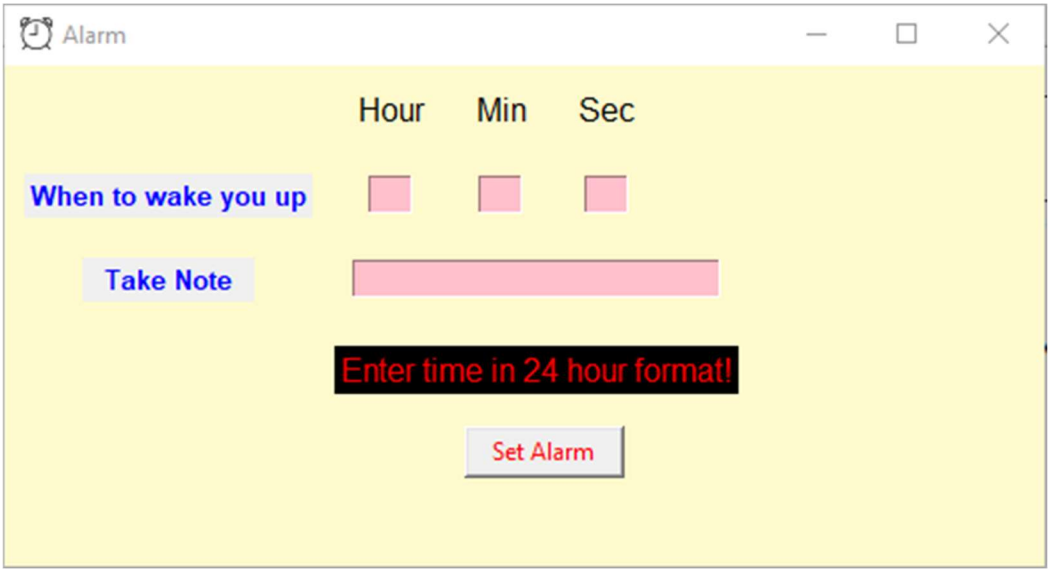
```
    r3.iconbitmap(r"C:\Users\Anushka
Dwivedi\OneDrive\Desktop\Project cs 2021\timer.ico")
    r3.mainloop()



#Labels
label1=Label(root, text="Select the operation",font=("Times
New Roman",20),fg='black',bg='salmon1')
label2=Label(root, text="you want to
perform",font=("Times New
Roman",20),fg='black',bg='salmon1')
#place labels
label1.pack()
label2.pack()

#create the button
button1=Button(root, text='  Alarm  ',font=("Consolas",15),
bg='black',fg='white', command=alarmCall)
button2=Button(root,
text='Countdown',font=("Consolas",15),
bg='black',fg='white', command=CountDown)
button3=Button(root, text='Stopwatch',
font=("Consolas",15),bg='black', fg='white', command=stop)

#place the button
button1.pack()
button2.pack()
button3.pack()
root.mainloop()
```

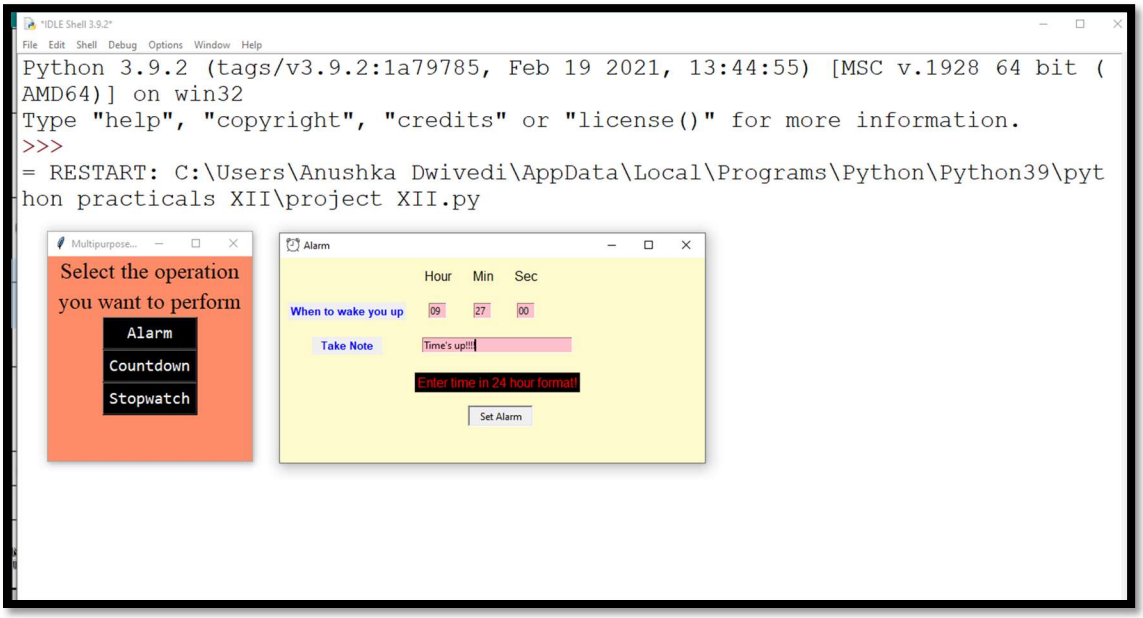
OUTPUT SCREEN

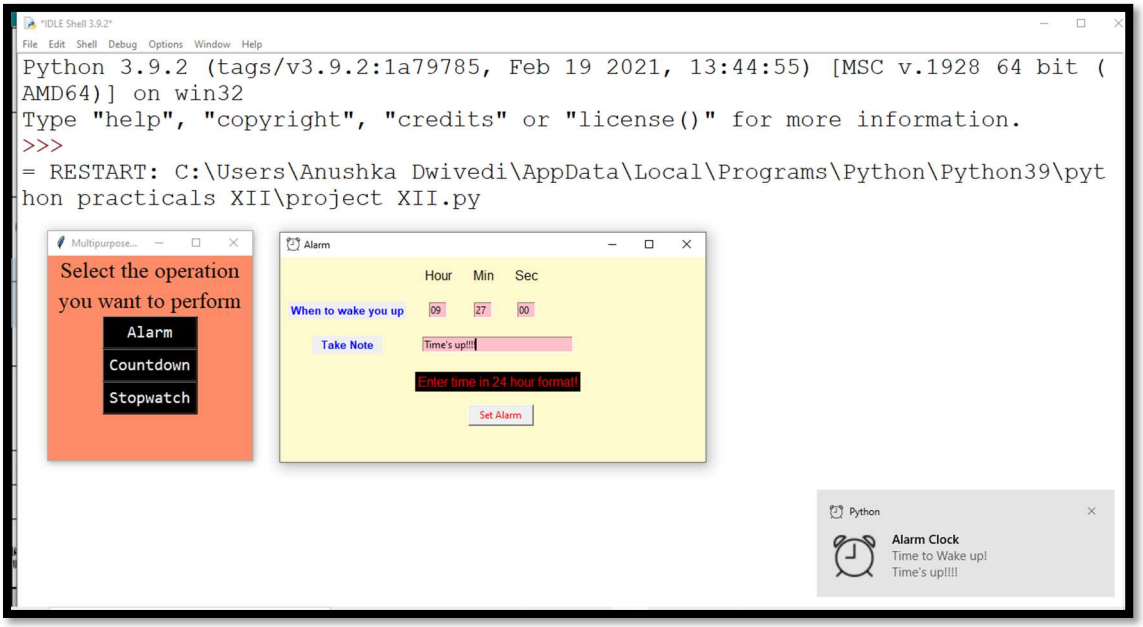
Main window

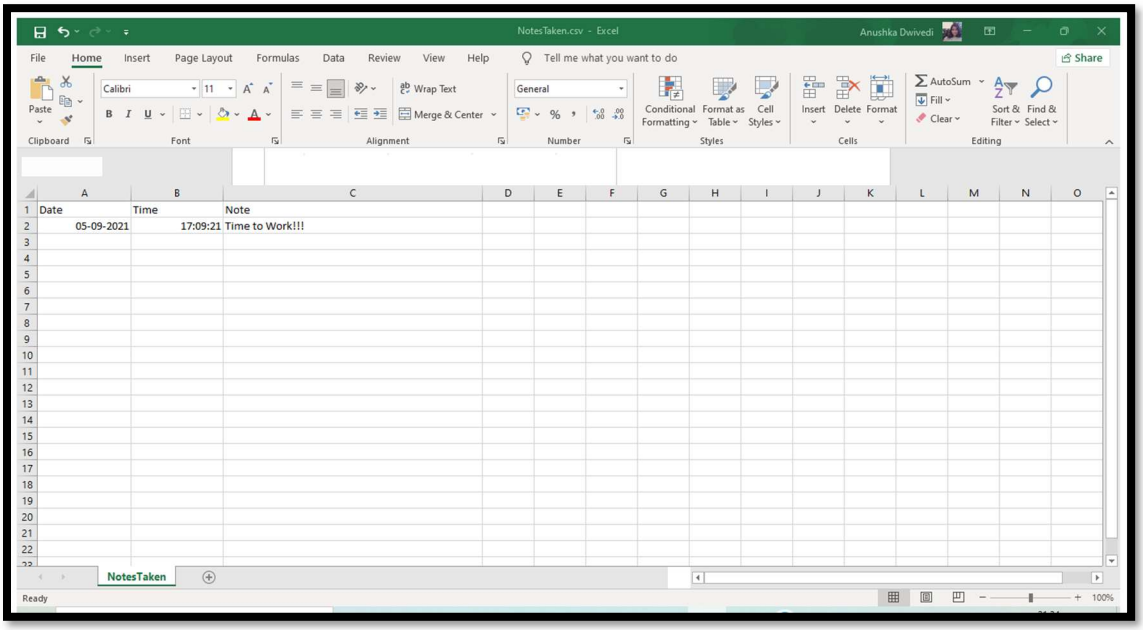


Alarm

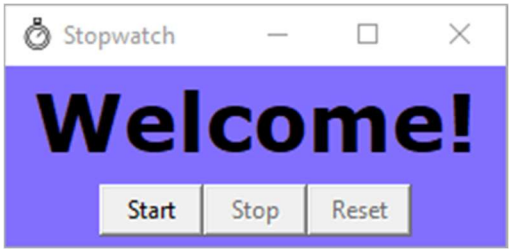


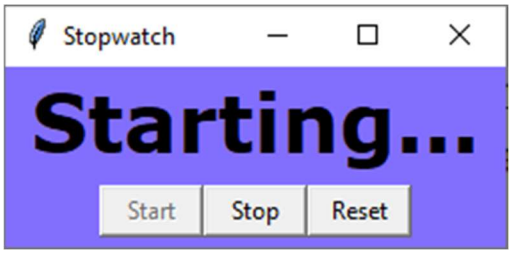

IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Anushka Dwivedi\AppData\Local\Programs\Python\Python39\python practicals XII\project XII.py
Multipurpose...
Select the operation you want to perform
Alarm
Countdown
Stopwatch
Alarm
Hour Min Sec
When to wake you up
09 27 00
Take Note
Time's up!!!!
Enter time in 24 hour format!
Set Alarm

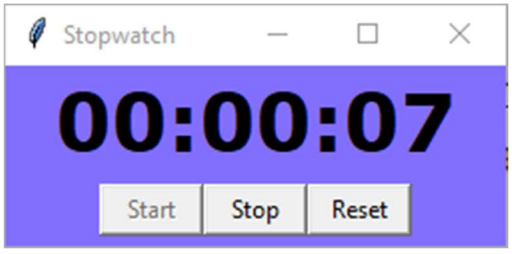

IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Anushka Dwivedi\AppData\Local\Programs\Python\Python39\python practicals XII\project XII.py
Multipurpose...
Select the operation you want to perform
Alarm
Countdown
Stopwatch
Alarm
Hour Min Sec
When to wake you up
09 27 00
Take Note
Time's up!!!!
Enter time in 24 hour format!
Set Alarm
Python
Alarm Clock
Time to Wake up!
Time's up!!!!


NotesTaken.csv - Excel
Anushka Dwivedi
File Home Insert Page Layout Formulas Data Review View Help
Tell me what you want to do
Share
Paste
Calibri
11
Wrap Text
Merge & Center
General
Conditional Formatting
Format as Table
Cell Styles
Insert
Delete
Format
AutoSum
Fill
Clear
Sort & Filter
Find & Select
Clipboard
Font
Alignment
Number
Styles
Cells
Editing
Date
Time
Note
05-09-2021
17:09:21
Time to Work!!!
NotesTaken
Ready
100%

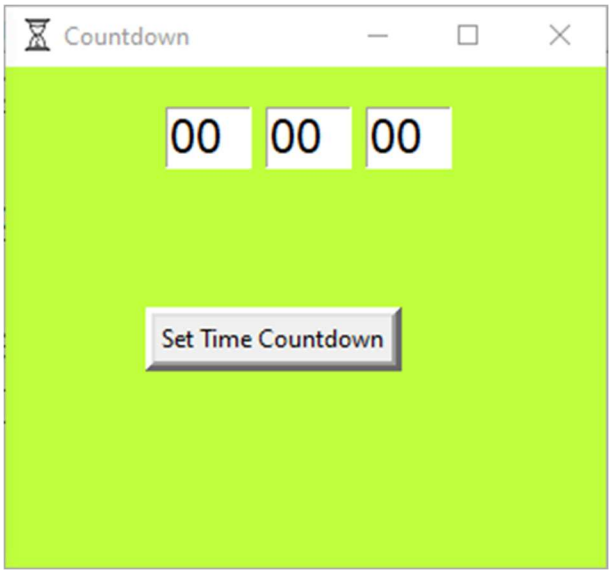
Stopwatch

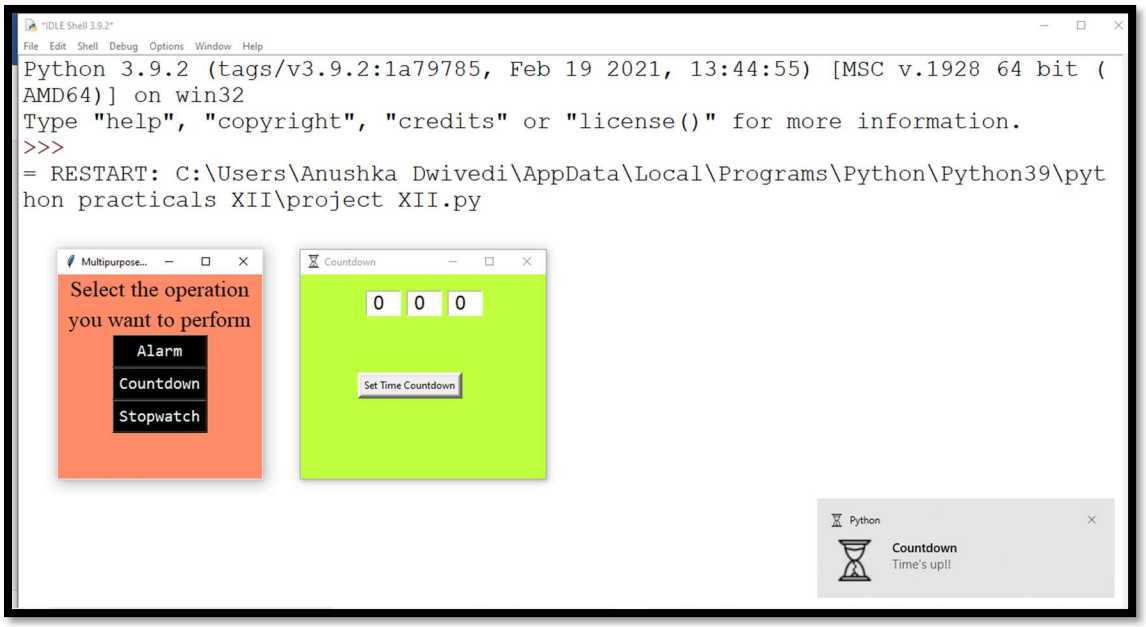

Stopwatch
Welcome!
Start
Stop
Reset


Stopwatch
Starting...
Start
Stop
Reset


Stopwatch
00:00:07
Start
Stop
Reset

Countdown


Countdown
00
00
00
Set Time Countdown


IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Anushka Dwivedi\AppData\Local\Programs\Python\Python39\pyt
hon practicals XII\project XII.py
Multipurpose...
Select the operation
you want to perform
Alarm
Countdown
Stopwatch
Countdown
0
0
0
Set Time Countdown
Python
Countdown
Time's up!!

BIBLIOGRAPHY

1. https://www.youtube.com/watch?v=HHddgUtliBg

2. https://stackoverflow.com/questions/51121144/access-stringvar-as-a-normal-string-in-python

3. https://www.geeksforgeeks.org/how-to-change-the-tkinter-label-font-size/

4. https://www.youtube.com/watch?v=237dNNQhD3Q

5. https://stackoverflow.com/questions/37400974/unicode-error-unicodeescape-codec-cant-decode-bytes-in-position-2-3-trunca