AUTOMATIC CLOCK AND TIMER

A Course Project Report submitted to the

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY ,HYDERABAD

in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING-CYBER SECURITY

Submitted By
A V B PAAVAN-21071A6201

Under the Guidance of

LALITHA MAM

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with ‘A++’ Grade (CGPA: 3.73/4.0)
NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech. Programmes
Approved by AICTE, New Delhi, Affiliated to JNTU-H, Recognised as “College with Potential for Excellence” by UGC
VignanaJyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad TS 500 090 India

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with ‘A++’ Grade (CGPA: 3.73/4.0)
NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech. Programmes
Approved by AICTE, New Delhi, Affiliated to JNTU-H, Recognised as “College with Potential for Excellence” by UGC
VignanaJyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad TS 500 090 India

TAMASOMA JYOTIRGAMAYA
Estd. 1995

CERTIFICATE

This is to certify that **A.V.B PAAVAN(21071A6201),** have successfully completed their Course Based Project work at **Computer Science & Engineering Department** of **Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology,** Hyderabad entitled **“AUTOMATIC CLOCK AND TIMER”** in partial fulfillment of the requirements for the award of **B.Tech** during the academic year 2022-2023**.**

This work is carried out under my supervision and has not been submitted to any other University/ Institute for award of any degree/ diploma.

DECLARATION

This is to certify that the project work entitled **"AUTOMATIC CLOCK AND TIMER"** submitted in VNR Vignana Jyothi Institute of Engineering & Technology in partial fulfillment of requirement for the award of Bachelor of Technology in Computer Science and Engineering is a bonafide report of the work carried out by us under the guidance and supervision of LALITHA MAM, Assistant Professor, Department of CYS,DS,AND AI&DS VNRVJIET. To the best of our knowledge, this report has not been submitted in any form to any university or institution for the award of any degree or diploma.

ACKNOWLEDGEMENT

An endeavor over a long period can be successful only with the advice and support of many well-wishers. We take this opportunity to express our gratitude and appreciation to all of them.

First of all we thank the lord almighty who has been with us from the beginning to the end of our project. We are indebted to our venerable principal **Dr. C. D. Naidu** for this unflinching devotion, which led us to complete this project. The support, encouragement given by him and his motivation lead us to complete this project.

We wish to express our profound gratitude to **Dr M.RAJASEKHAR HOD OF CSE-CYS DS AND AI &DS** Department**, VNR Vignana Jyothi Institute of Engineering and Technology** for their constant and dedicated service to brighten our career.

With great pleasure we express our gratitude to the internal guide **LALITHA MAM** for his timely help, constant guidance, cooperation, support and encouragement throughout this project.

Finally we wish to express our deep sense of gratitude and sincere thanks to our parents, friends and all our well-wishers who have technically and non-technically contributed for the successful completion of our course based project.

ABSTRACT

- Time tracking is critical to managing your projects. If you’re programming in Python, you’re in luck: The language gives you tools to build your own timers. A timer program can be useful for not only monitoring your own time, but measuring how long your Python program takes to run.
- A timer in Python is a time-tracking program. Python developers can create timers with the help of Python’s time modules.

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Stopwatches

Timers that count up from zero are frequently called stopwatches. We can use these to record the amount of time it takes to complete a task.

Programmers often use stopwatch timers to compare the performance of various Python solutions. By seeing how long each solution takes to execute, you can choose the program that runs the fastest. Over time, this will allow you to better understand computational complexity and build intuition for choosing efficient solutions. These skills go a long way towards becoming a proficient programmer.

Countdown Timers

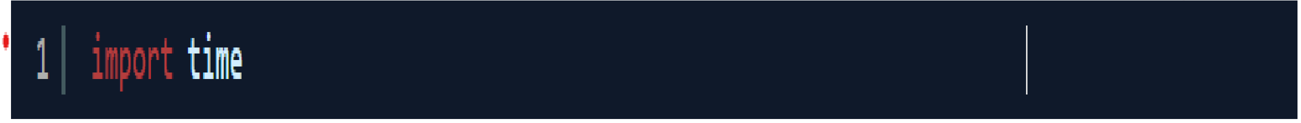
There are also countdown timers, set with a specific amount of time that depletes until the timer reaches zero. You can use Python to build countdown timers that serve different purposes.

For instance, you can build a cooking countdown timer to ensure you don’t leave food in the oven for too long. Countdown timers can also work to display the time remaining in a sporting event or during an exam. They are also used to count down the hours and minutes to a movie release or big event. The possibilities are endless.

The Python Time Module:

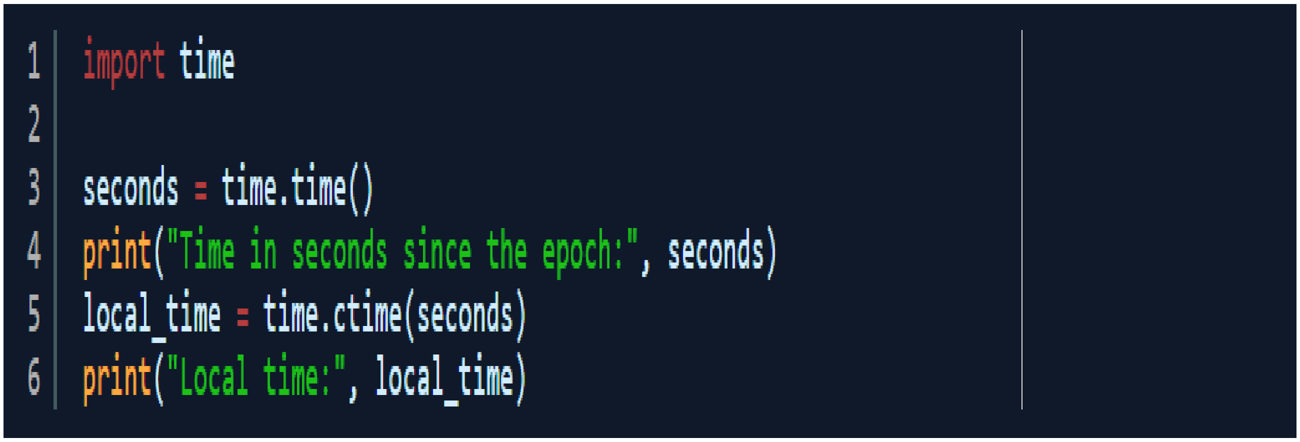
One of the 200 modules in Python’s standard library is the **time module**. This module contains the functions we’ll need to build a simple timer in Python.

To use the time module in Python, we first import it into our program:



```python
import time
```

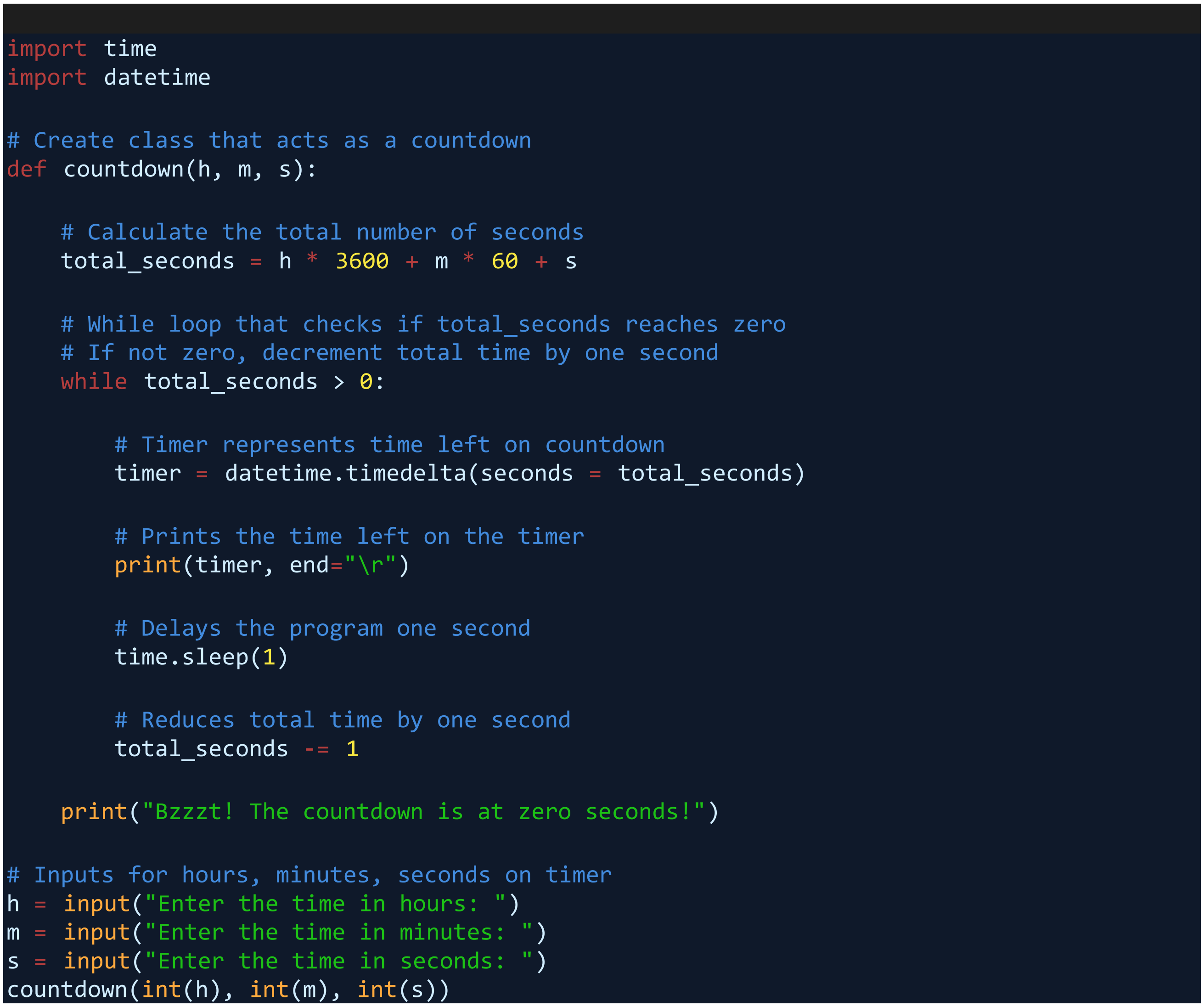
- **The time.time() function:**
  We call the time() function, located in the time module, as time.time(). The first time references the module, whereas the second is the function itself. The time() function returns the number of seconds that have passed since the epoch.



```python
import time

seconds = time.time()
print("Time in seconds since the epoch:", seconds)
local_time = time.ctime(seconds)
print("Local time:", local_time)
```

CHAPTER 3
IMPLEMENTATION

A Simple Countdown Timer

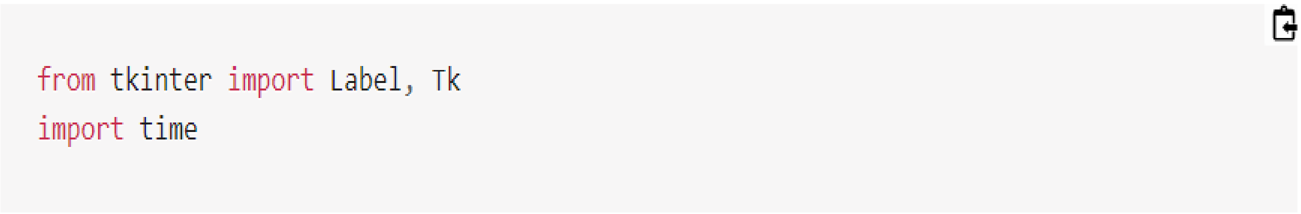


```python
import time
import datetime

# Create class that acts as a countdown
def countdown(h, m, s):

    # Calculate the total number of seconds
    total_seconds = h * 3600 + m * 60 + s

    # While loop that checks if total_seconds reaches zero
    # If not zero, decrement total time by one second
    while total_seconds > 0:

        # Timer represents time left on countdown
        timer = datetime.timedelta(seconds = total_seconds)

        # Prints the time left on the timer
        print(timer, end="\r")

        # Delays the program one second
        time.sleep(1)

        # Reduces total time by one second
        total_seconds -= 1

    print("Bzzzt! The countdown is at zero seconds!")

# Inputs for hours, minutes, seconds on timer
h = input("Enter the time in hours: ")
m = input("Enter the time in minutes: ")
s = input("Enter the time in seconds: ")
countdown(int(h), int(m), int(s))
```

DIGITAL CLOCK USING PYTHON:

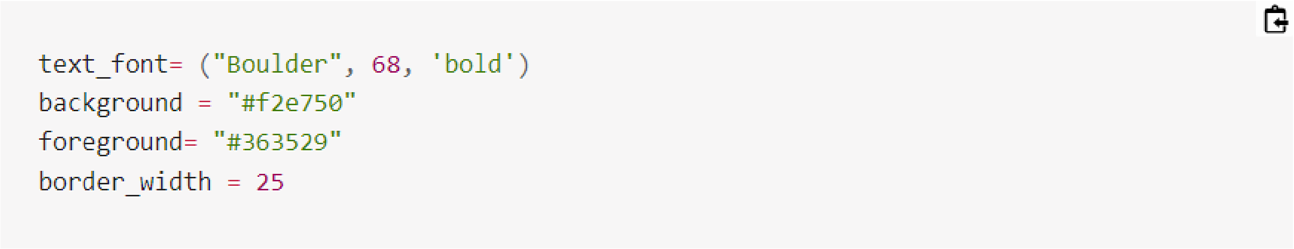
In this section, I will show you how to create a digital clock. This is a simple task to get started with the Tkinter library in Python, which is a built-in package that comes with Python. Tkinter has some cool features that can be used to build simple apps

Now let’s see how to create a digital clock GUI application with Python. I will first start with importing the libraries:



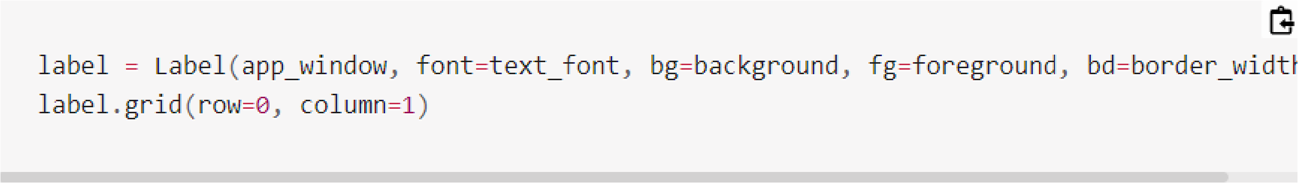
```python
from tkinter import Label, Tk
import time
```

Now here I will define the font of the time and its color, its border width, and the background color of the digital clock:



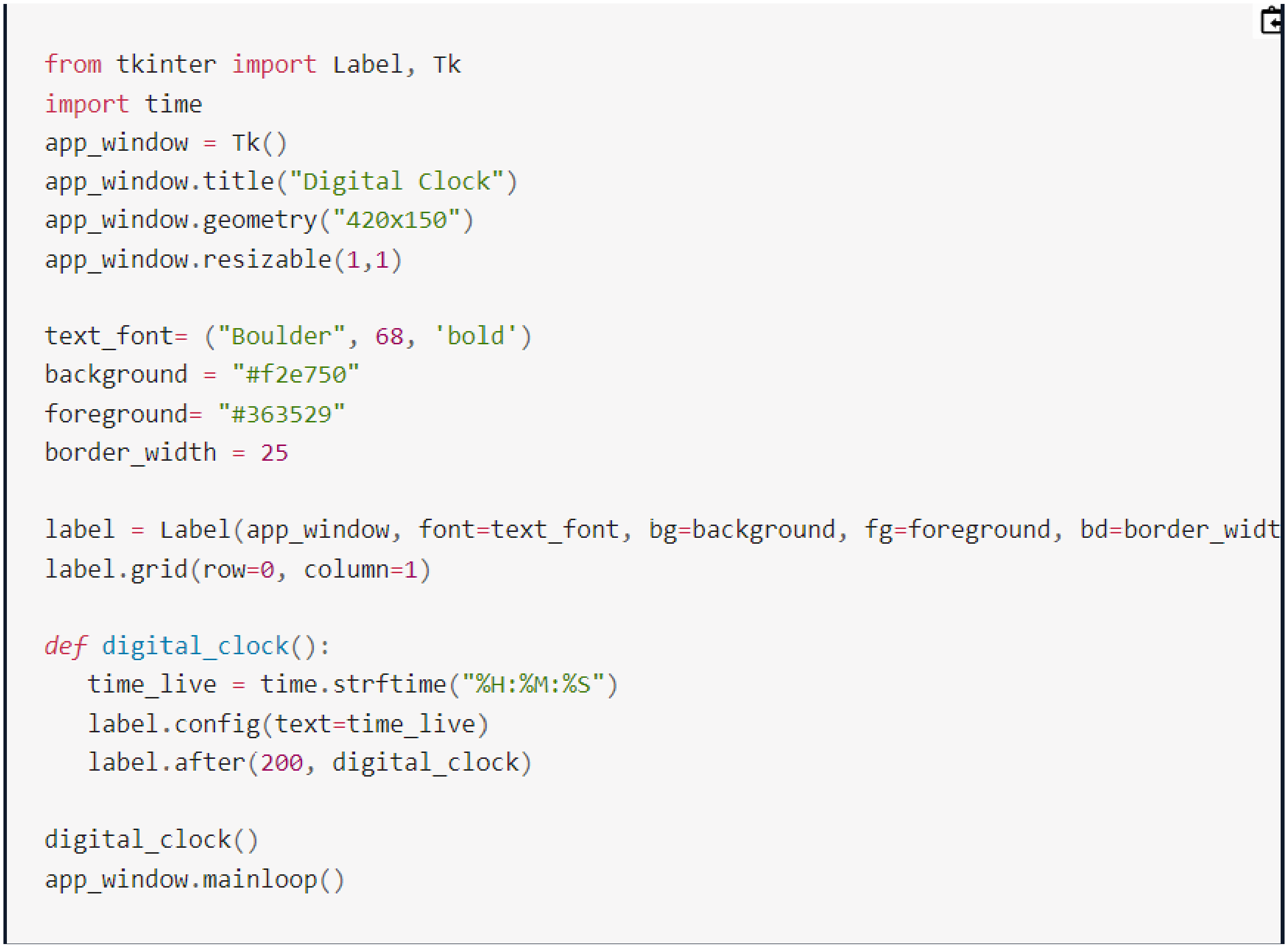
```python
text_font= ("Boulder", 68, 'bold')
background = "#f2e750"
foreground= "#363529"
border_width = 25
```

Now here I will combine all the elements to define the label of the clock application:



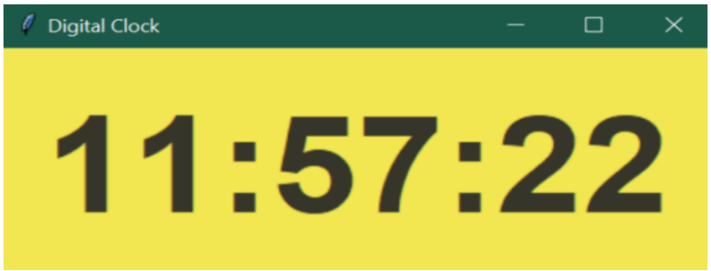
```python
label = Label(app_window, font=text_font, bg=background, fg=foreground, bd=border_widt
label.grid(row=0, column=1)
```

COMPLETE CODE:



```
from tkinter import Label, Tk
import time
app_window = Tk()
app_window.title("Digital Clock")
app_window.geometry("420x150")
app_window.resizable(1,1)

text_font= ("Boulder", 68, 'bold')
background = "#f2e750"
foreground= "#363529"
border_width = 25

label = Label(app_window, font=text_font, bg=background, fg=foreground, bd=border_widt
label.grid(row=0, column=1)

def digital_clock():
   time_live = time.strftime("%H:%M:%S")
   label.config(text=time_live)
   label.after(200, digital_clock)

digital_clock()
app_window.mainloop()
```

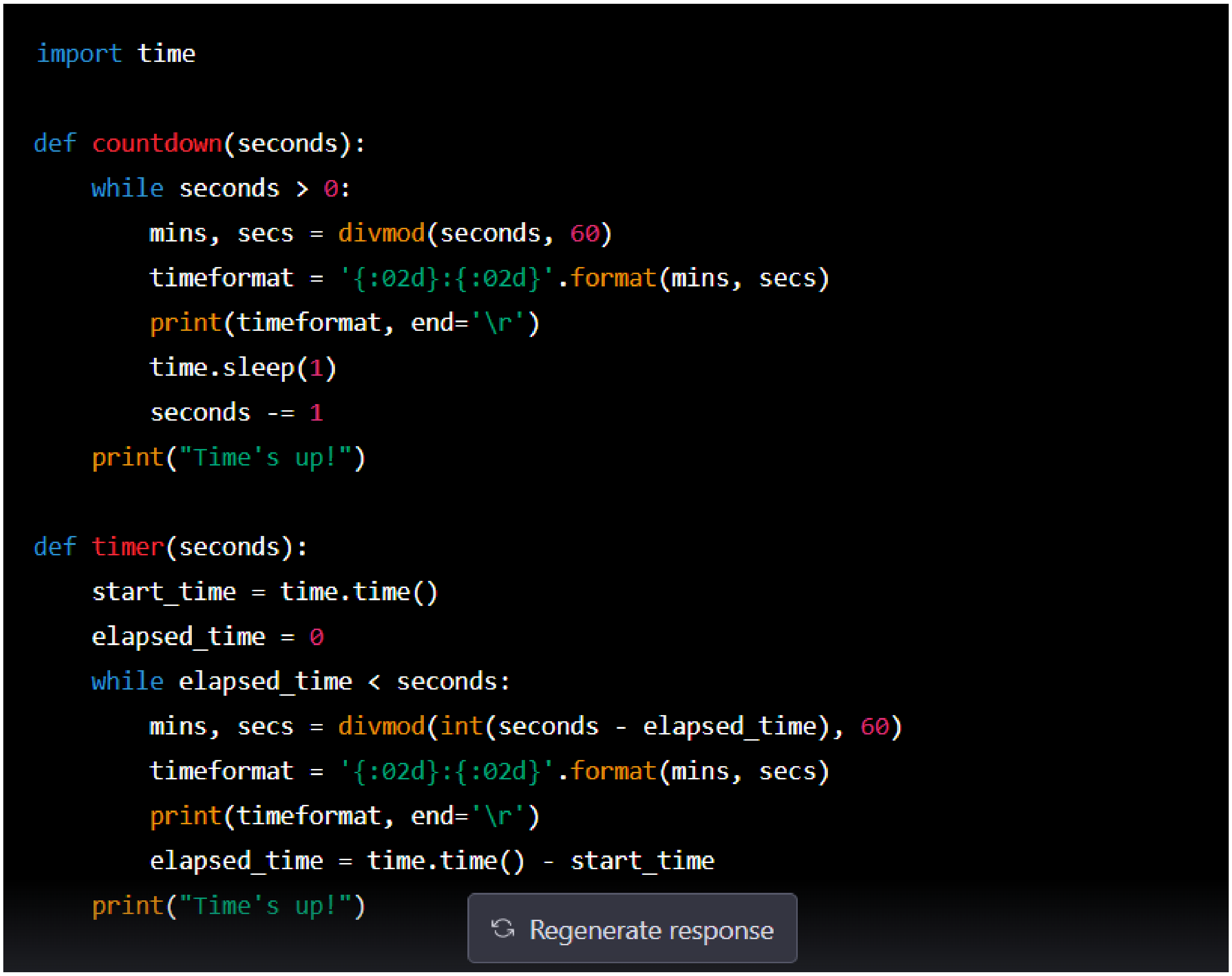
OUTPUT:



Digital Clock

```
11:57:22
```

PYTHON CODE TO DISPLAY COUNTDOWN CLOCK AND TIMER:



```
import time

def countdown(seconds):
    while seconds > 0:
        mins, secs = divmod(seconds, 60)
        timeformat = '{:02d}:{:02d}'.format(mins, secs)
        print(timeformat, end='\r')
        time.sleep(1)
        seconds -= 1
    print("Time's up!")

def timer(seconds):
    start_time = time.time()
    elapsed_time = 0
    while elapsed_time < seconds:
        mins, secs = divmod(int(seconds - elapsed_time), 60)
        timeformat = '{:02d}:{:02d}'.format(mins, secs)
        print(timeformat, end='\r')
        elapsed_time = time.time() - start_time
    print("Time's up!")
```

This script uses the time module to create a countdown clock and timer. The **countdown**function takes an argument **seconds** and counts down from that number to 0, printing the remaining time on each iteration of the loop. Once the countdown is complete, the function prints "Time's up!".

The **timer** function works similarly, but instead of counting down from a specified time, it counts up from 0 to the specified number of seconds. Once the time has elapsed, the function prints "Time's up!".

To use this script, simply call the **countdown** or **timer** function with the desired number of seconds as an argument. For example, **countdown(60)** will count down from 60 seconds, while **timer(60)** will count up to 60 seconds.

There are different ways to write a Python program to show the current date and time. One of the most popular ways is to use the *datetime* module. Another way is to use the time* module. Here are some examples of code using both modules:

Using datetime module:

```
import datetime
currentDT = datetime.datetime.now()
print(currentDT)
```

Using time module:

```
import time
currentTM = time.gmtime()
print(currentTM)
```

CONCLUSION:

Python Countdown Timer can be utilized to wait for a certain duration of time in the idle state before reiterating the same piece of code in the loop again as required. Python’s time library contains a predefined sleep() function.

A Python Alarm Clock Script includes essential libraries such as DateTime and Tkinter, which assist us in constructing projects utilizing the current date and time. They also give a user interface to set the alarm according to the demand in a 24-hour format.