CHAPTER -1
INTRODUCTION

1.1 General Introduction

Notification support people in remembering things. It is a little piece of text that appears on the desktop or mobile screen to inform the user about the updates or any other significant chunks of information. This information enables the user to focus on vital events and ignore the less significant ones. The notifications are stored in the notification bar, which we can refer to after completing the work.

Desktop Notifier is a straightforward application that generates a notification message in the form of a pop-up message on the desktop. The main objective of the desktop notification application that we will learn to develop today is to constantly remind us of the different things that we require to accomplish throughout the day.This task is like a to-do list, where we have a set of goats to accomplish. And the desktop notifier will constantly notify us of the different to-do and actions to take throughout the day.

Desktop notifications allow agents to easily screen inbound calls and make an informed decision about whether to accept a call. Desktop notifications also boost call center productivity, as agents can perform other tasks on their computers and still be alerted to an inbound call. Desktop alerts are attention grabbers that cannot be missed. Used to send urgent messages, routine communications or emergency notifications, desktop alerts feature a pop-up that appears on the user’s monitor, over other applications and commands attention. This highly-visible vehicle allows administrators to deliver custom notifications to selected groups or teams or to the entire network. This ensures that the right people get the message each and every time. A notification is a message that Android displays outside your app's UI to provide the user with reminders, communication from other people, or other timely information from your app. Users can tap the notification to open your app or take an action directly from the notification

Desktop alerts are attention grabbers that cannot be missed. Used to send urgent messages, routine communications or emergency notifications, desktop alerts feature a pop-up that appears on the user’s monitor, over other applications and commands attention. This highly-visible vehicle allows administrators to deliver custom notifications to selected groups or teams or to the entire network. This ensures that the right people get the message each and every time . A notification is a message that Android displays outside your app's UI to provide the user with reminders, communication from other people, or other timely information from your app. Users can tap the notification to open your app or take an action directly from the notification.

Notification support people in remembering things. It is a little piece of text that appears on the desktop or mobile screen to inform the user about the updates or any other significant chunks of information. This information enables the user to focus on vital events and ignore . the less significant ones. The notifications are stored in the notification bar, which we can refer to after completing the work. A Desktop Notifier is a straightforward application that generates a notification message in the form of a pop-up message on the desktop. The main objective of the desktop notification application that we will learn to develop today is to constantly remind us of the different things that we require to accomplish throughout the day.

Desktop Notifier is a straightforward application that generates a notification message in the form of a pop-up message on the desktop. The main objective of the desktop notification application that we will learn to develop today is to constantly remind us of the different things that we require to accomplish throughout the day

Notification support people in remembering things. It is a little piece of text that appears on the desktop or mobile screen to inform the user about the updates or any other significant chunks of information. This information enables the user to focus on vital events and ignore the less significant ones. The notifications are stored in the notification bar, which we can refer to after completing the work

Fig 1 working of desktop notification

Organization of Documentation:

Chapter-2: describes the Introduction to python programming.

Chapter-3: shows the methodology of project.

Chapter-4: discus about results.

CHAPTER2

INTRODUCTION TO PYTHON

2.1 WHAT IS PYTHON

Python is developed by **Guido van Rossum**. Guido van Rossum started implementing Python in 1989. Python is a very simple programming language so even if you are new to programming, you can learn python without facing any issues.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting language to connect existing components together. Python is a simple, easy-to-learn syntax that emphasizes readability and therefore reduces the cost of programming maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.

Often programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-debug-test cycle is incredibly fast. Debugging Python programs is easy because a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn’t catch the exception, the interpreter prints a stack trace. A source-level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, and stepping through the code a line at a time.

2.2 WHY PYTHON IS USED?

Python language is one of the most accessible programming languages available because it has simplified syntax and is not complicated, which gives more emphasis on natural language. Due to its ease of learning and usage, python codes can be easily written and executed much faster than other programming languages.

Python has a lot of modules, they are essentially python script files that can contain variables, functions, and classes. Python modules help us in organizing our code and then referencing them in other classes or python scripts. Python has a robust and large standard library that makes it stand out from other programming languages. Its standard library contains a wide range of modules, operations and web service tools that you can select and use for your applications without writing code

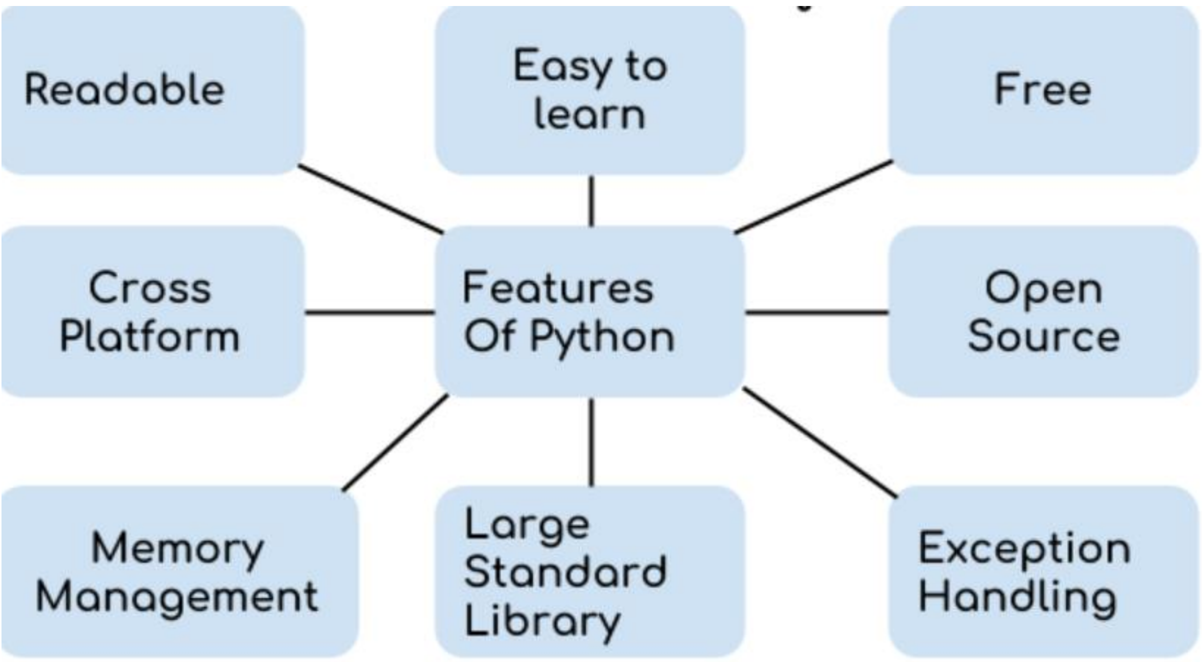


Fig 2 features of python

CHAPTER -3
METHODOLOGY

3.1 Block diagram:

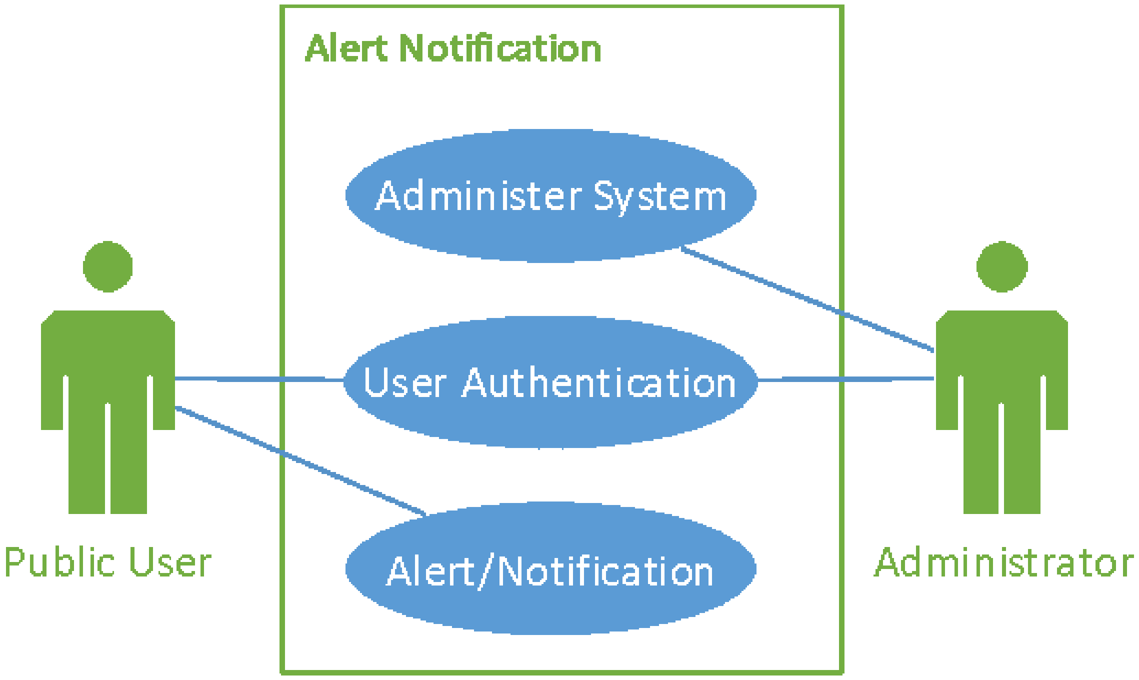
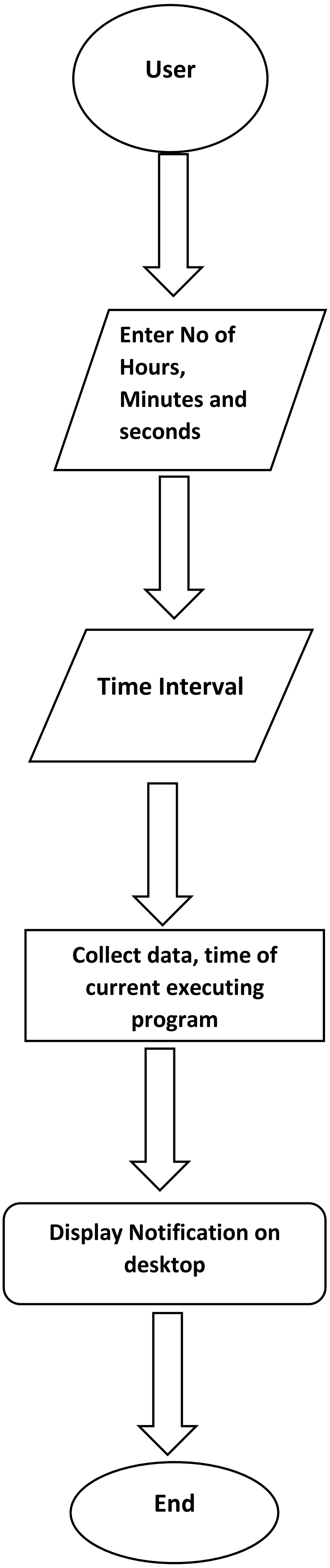


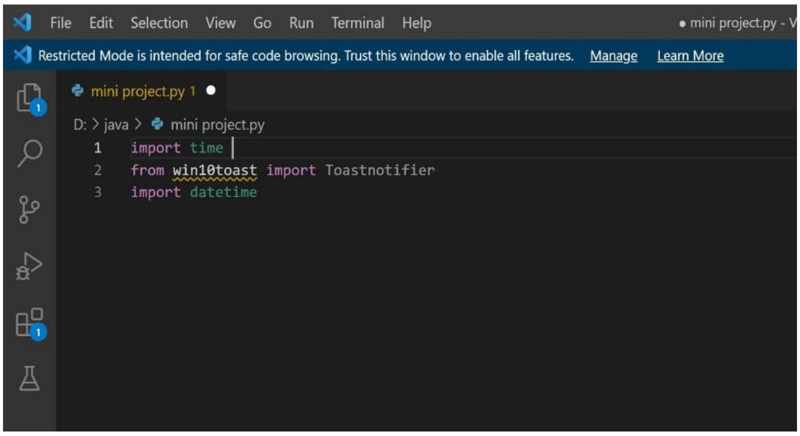
Fig 3 Architecture Of Notification Working

Import modules win10Toast , datetime . win10Toast required us to call ToastNotifier() to display pop message on desktop . datetime module helps us to store present date and time of a program . create a function_name with getTimeinput() in that function initialize inputs as no of hours , minutes , seconds and convert hours , minutes , seconds into seconds respectively assigned to time_interval and return time_interval . create a log file and store present date and time of current executing program with the help of strftime module . create a function named as notify() . In that function call ToastNotifier() which is imported from win10toast module .To display notification as “Time to take a break” set timer from current executing program time to user defined time .

3.2 Flow Chart :


User
Enter No of Hours, Minutes and seconds
Time Interval
Collect data, time of current executing program
Display Notification on desktop
End

3.3 IMPORTED MODULES


File Edit Selection View Go Run Terminal Help
● mini project.py - V
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
mini project.py 1 ●
D: > java > mini project.py

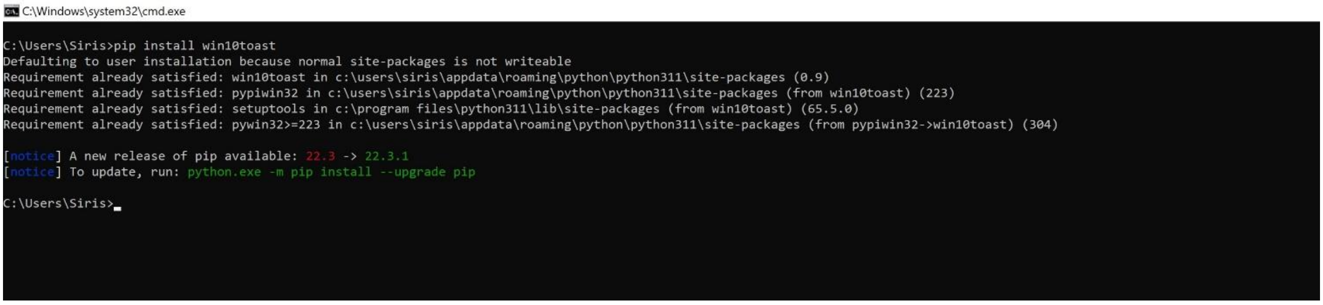
```python
import time
from win10toast import Toastnotifier
import datetime
```

Fig.4 Modules Imported

A. **Time :** Python time module allows to work with time in Python. It allows functionality like getting the current time, pausing the Program from executing, etc. So before starting with this module we need to import it .

B. **Toast notifier :** To create notifications we have to import the win10toast module. Then create an object to toast notifier class and by using the method Show toast we create a notification. It contains *header* or title of that notification, actual message, duration of that notification and icon for that notification .

C. **datetime :** 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. Its a combination of date and time along with the attributes year, month, day, hour, minute, second, microsecond .

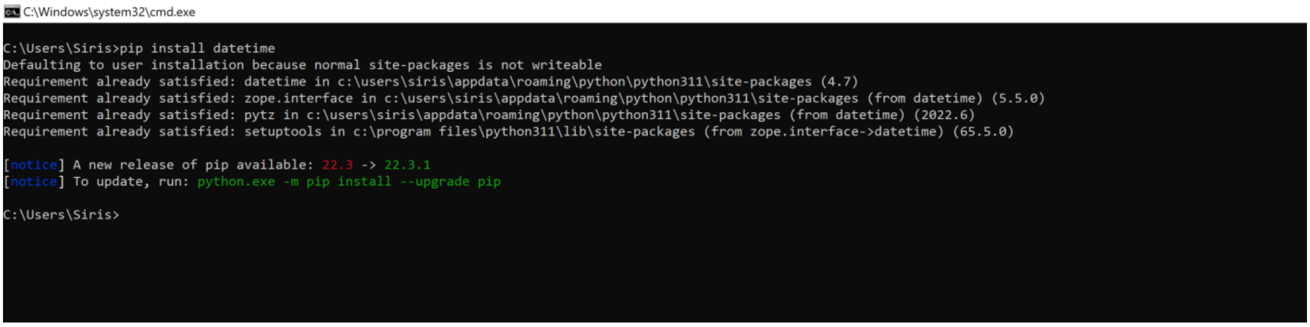


```
C:\Users\Siris>pip install win10toast
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: win10toast in c:\users\siris\appdata\roaming\python\python311\site-packages (0.9)
Requirement already satisfied: pypiwin32 in c:\users\siris\appdata\roaming\python\python311\site-packages (from win10toast) (223)
Requirement already satisfied: setuptools in c:\program files\python311\lib\site-packages (from win10toast) (65.5.0)
Requirement already satisfied: pywin32>=223 in c:\users\siris\appdata\roaming\python\python311\site-packages (from pypiwin32->win10toast) (304)

[notice] A new release of pip available: 22.3 -> 22.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\Siris>_
```

Fig.5 Install pip win packages



```
C:\Users\Siris>pip install datetime
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: datetime in c:\users\siris\appdata\roaming\python\python311\site-packages (4.7)
Requirement already satisfied: zope.interface in c:\users\siris\appdata\roaming\python\python311\site-packages (from datetime) (5.5.0)
Requirement already satisfied: pytz in c:\users\siris\appdata\roaming\python\python311\site-packages (from datetime) (2022.6)
Requirement already satisfied: setuptools in c:\program files\python311\lib\site-packages (from zope.interface->datetime) (65.5.0)

[notice] A new release of pip available: 22.3 -> 22.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\Siris>
```

Fig.6 Install pip collected packages

3.4 DATA PROCESSING :

- Import the required modules which are required to access different applications as shown in the system architecture.

❖ PROCEDURE :

1. We know the notification works on date and time , so the first part is to import the time module , date time module to detect the date and time of execution and install win10toast module to get notification symbol .

```
#part 1(date and time recognition)

import time
from win10toast import Toastnotifier
import datetime
```

2. At the end of this first part completion , successfully modules are imported .

3. create a function name called getTimeinput() , initialize inputs for hours , minutes Seconds and convert all inputs into seconds and assigned to time_interval.

```
#part 2(Intializing inputs)

def getTimeInput():

    hour = int(input("hours of interval :"))

    minutes = int(input("Mins of interval :"))

    seconds = int(input("Secs of interval :"))

time_interval=seconds+(minutes*60)+(hour*3600)

    return time_interval
```

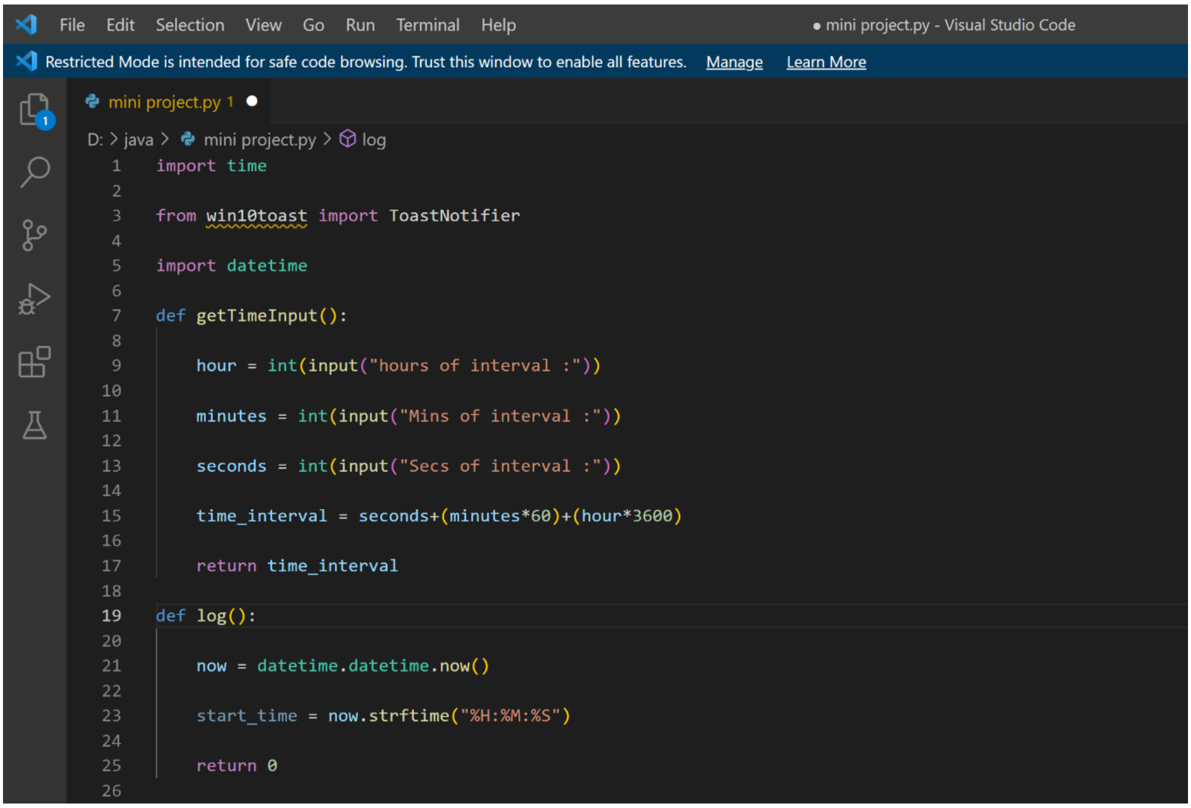
4. now, create a log file in that take datetime.now() , start_time as “%H:%M:%S”.

```
#part 3(taking date time and start time in log file)

def log():

        now=datetime.datetime.now()

        start_time=now.strftime(“%H:%M:%S”)
```



File Edit Selection View Go Run Terminal Help
● mini project.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
mini project.py 1
D: > java > mini project.py > log

```python
import time

from win10toast import ToastNotifier

import datetime

def getTimeInput():

    hour = int(input("hours of interval :"))

    minutes = int(input("Mins of interval :"))

    seconds = int(input("Secs of interval :"))

    time_interval = seconds+(minutes*60)+(hour*3600)

    return time_interval

def log():

    now = datetime.datetime.now()

    start_time = now.strftime("%H:%M:%S")

    return 0

```

Fig .7 modules and Intializing inputs

5. Now , create function name with notify and take ToastNotifier from winıotoast module

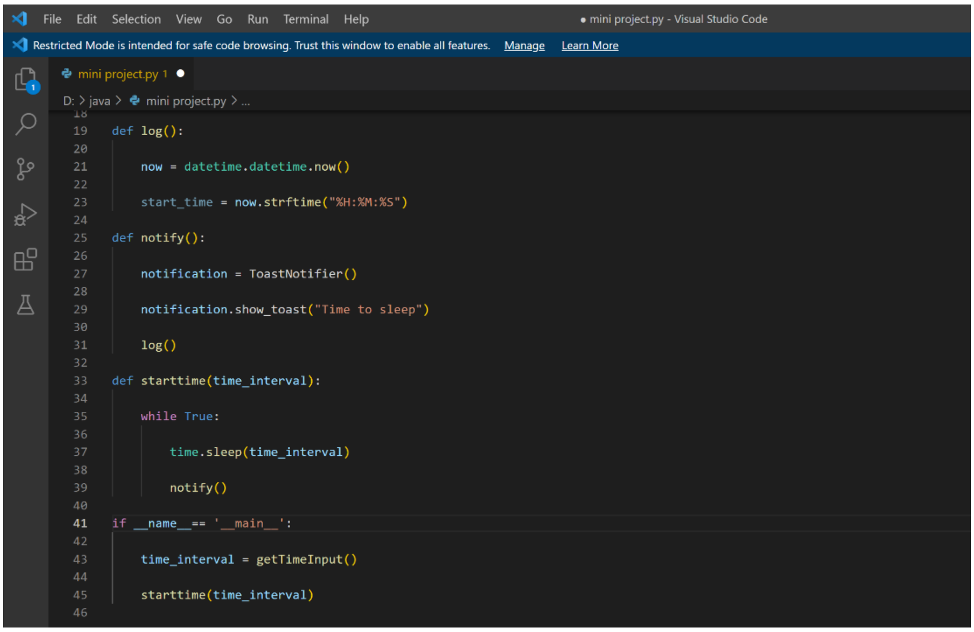
Alert notification as “Time to take a break”.

```
#part 4(taking ToastNotifier from win10toast)

def notify():
        notification=ToastNotifier()
        notification.show_toast(“Time to take a break”)
        log()
def starttime(time_interval):
        while True:
                time.sleep(time_interval)
                notify()
```

6. Create an object for this class and implement code to get notfication

```
If __name_==”__main__”:
        Time_interval=getTimeinput()
        startTime(time_interval)
```


File
Edit
Selection
View
Go
Run
Terminal
Help
● mini project.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features.
Manage
Learn More
mini project.py 1 ●
D: > java > mini project.py > ...

```
def log():

    now = datetime.datetime.now()

    start_time = now.strftime("%H:%M:%S")

def notify():

    notification = ToastNotifier()

    notification.show_toast("Time to sleep")

    log()

def starttime(time_interval):

    while True:

        time.sleep(time_interval)

        notify()

if __name__== '__main__':

    time_interval = getTimeInput()

    starttime(time_interval)
```

Fig. 8 Implementation of Notification Working

3.5 CODE THAT EXECUTED

```
import time
from win10toast import Toastnotifier
import datetime

def getTimeInput():

    hour = int(input("hours of interval :"))

    minutes = int(input("Mins of interval :"))

    seconds = int(input("Secs of interval :"))

    time_interval = seconds+(minutes*60)+(hour*3600)

return time_interval

def log():

    now=datetime.datetime.now()

    start_time=now.strftime(“%H:%M:%S”)


def notify():

    notification=ToastNotifier()

    notification.show_toast(“Time to take a break”)

    log()

def starttime(time_interval):

    while True:

        time.sleep(time_interval)

        notify()
If __name_==”__main__”:

    Time_interval=getTimeinput()

    startTime(time_interval)
```

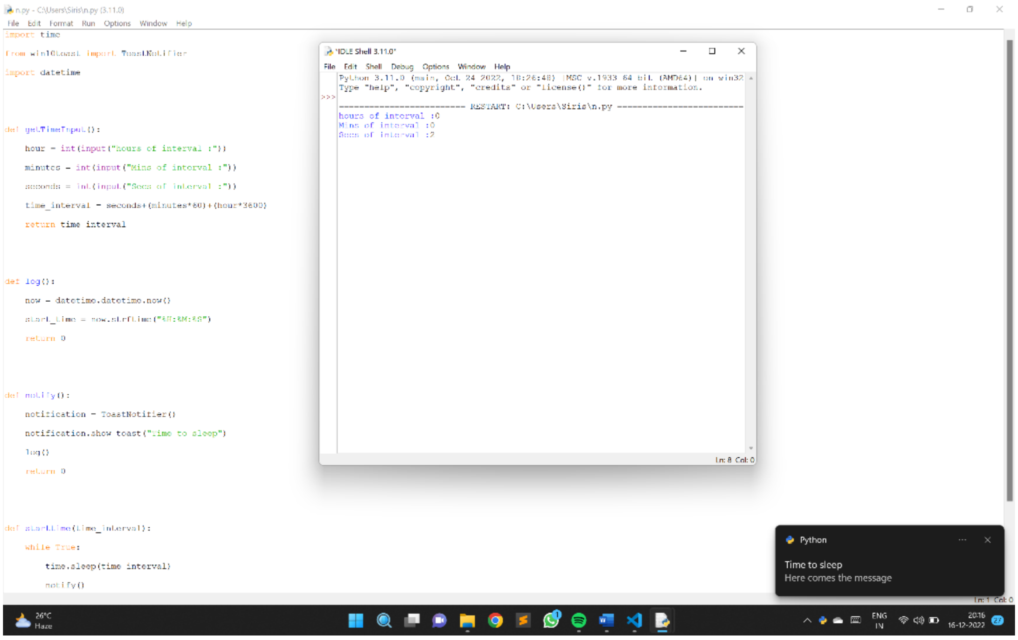
CHAPTER-4
RESULTS

IMPLEMENTATION :

Enter No of Hours: 0

Enter No of Minutes: 0

Enter No of Seconds: 2


n.py - C:\Users\Siris\n.py (3.11.0)
File Edit Format Run Options Window Help
IDLE Shell 3.11.0
File Edit Shell Debug Options Window Help
Ln: 8 Col: 0
Ln: 1 Col: 0
26°C
Haze
ENG
IN
20:16
16-12-2022

```
import time
from win10toast import ToastNotifier
import datetime


def getTimeInput():
    hour = int(input("hours of interval :"))
    minutes = int(input("Mins of interval :"))
    seconds = int(input("Secs of interval :"))
    time_interval = seconds+(minutes*60)+(hour*3600)
    return time_interval


def log():
    now = datetime.datetime.now()
    start_time = now.strftime("%H:%M:%S")
    return 0


def notify():
    notification = ToastNotifier()
    notification.show_toast("Time to sleep")
    log()
    return 0


def startTime(time_interval):
    while True:
        time.sleep(time_interval)
        notify()
```

```
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======================= RESTART: C:\Users\Siris\n.py =======================
hours of interval :0
Mins of interval :0
Secs of interval :2
```

Python

Time to sleep

Here comes the message

Fig.9 Practical output

Finally, this notifications support people in remembering things. It is a little piece of text that appears on the desktop or mobile screen to inform the user about the updates or any other significant chunks of information. This information enables the user to focus on vital events and ignore the less significant ones . It will send remainders such as to take break, rest, sleep etc.

CONCLUSION

In this **project** we have discussed about Desktop Notification System Using Python . Finally, we concluded that we can know how much time we are spending on desktop therefore , we know it will reduce human eye sight , when we take breaks frequently our health conditions will also improve . Generally , most of the people too much reliable on desktop by watching movies , playing video games or any other purpose. This project sets a remainder how much time you will spend time on desktop . It is like alarm .This desktop notification system makes a human daily life progress in a good manner.

APPLICATIONS:

- Desktop Alerts are Most Effective for Sending High-Priority Messages to Staff, Such as Breaking News, Emergency Notifications or Essential Employee Communications.
- It Will Send Remainders Such as to Take Breaks, Rest, Sleep, Uses in Improving Health Conditions

FUTURE DEVELOPMENTS:

In further , we can implement alarm clock , stopwatch , etc
You can use the following key features in Python modules.

- Timer
- Win sound
- Win toast
- Plyer

REFERENCES

- https://www.geeksforgeeks.org/desktop-notifier-python/
- https://www.youtube.com/watch?v=cfjPfYuZBZs
- https://github.com/topics/desktop-notifier
- https://www.javatpoint.com/desktop-notifier-in-python