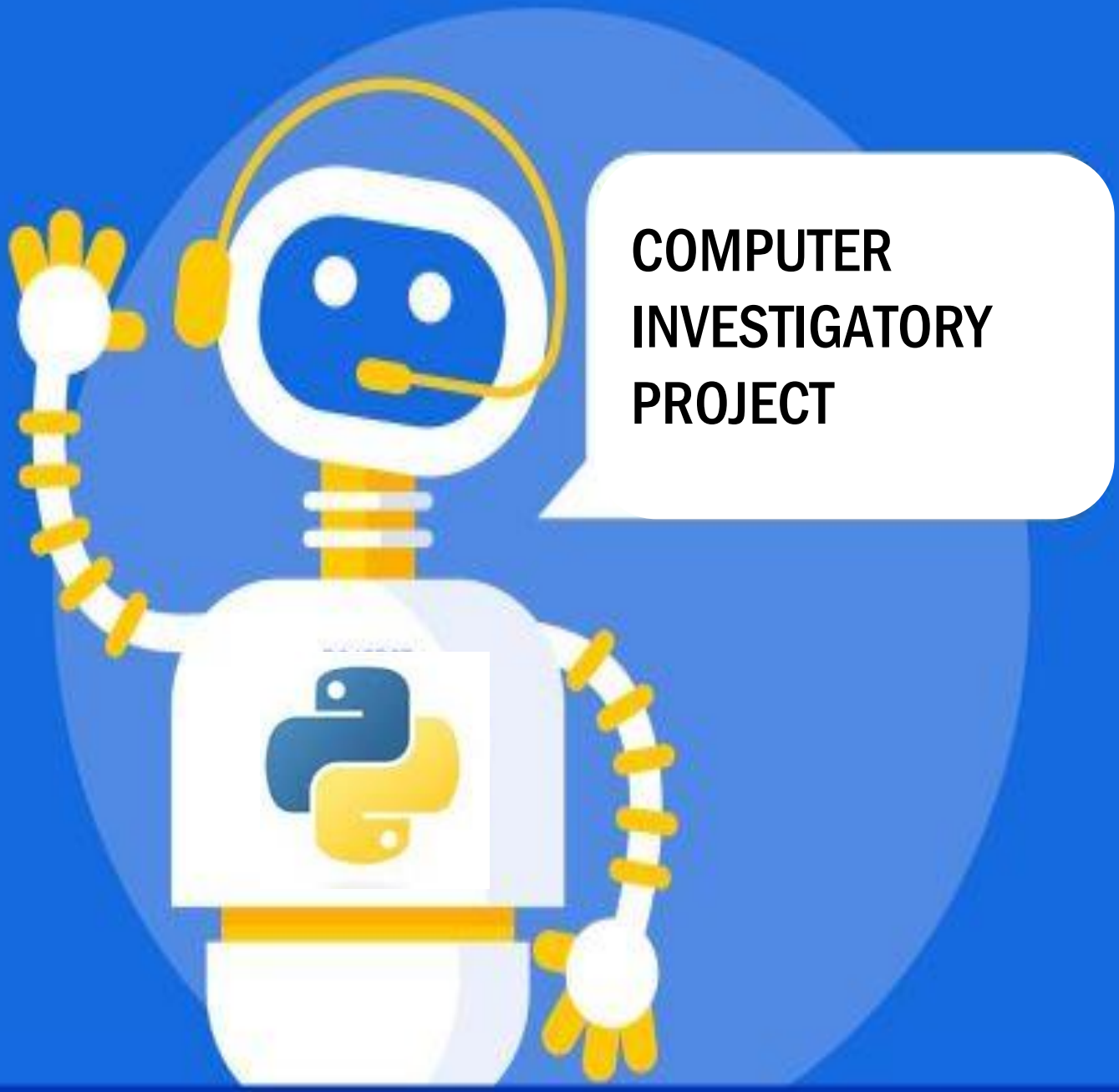
COMPUTER INVESTIGATORY PROJECT



KEANE COUTINHO - 12 STD B SECTION

CERTIFICATE

This is hereby to certify that, the original and genuine investigation work has been carried out to investigate about the subject matter and the related data collection and investigation has been completed solely, sincerely, and satisfactorily by **Keane Coutinho of Class XII B, 'Our Own High School'** regarding his project titled '**VIRTUAL ASSISSTANT**'.

Roll No: _____ Academic year: 2020-2021

Date__/__/____ _____

Teacher In-Charge

_____ _____

Internal Examiner External Examiner

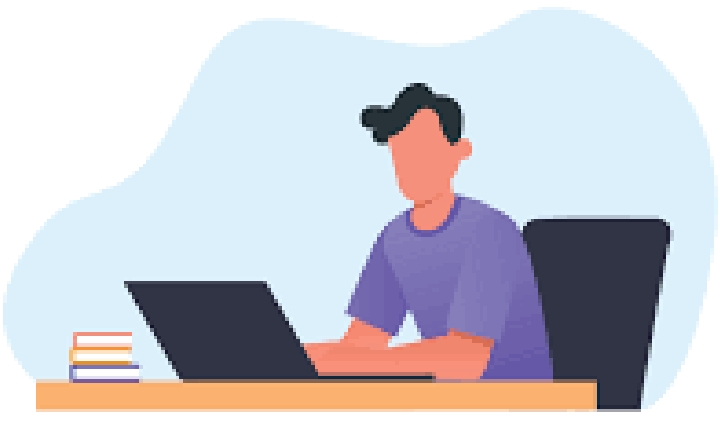
ACKNOWLEDGEMENT

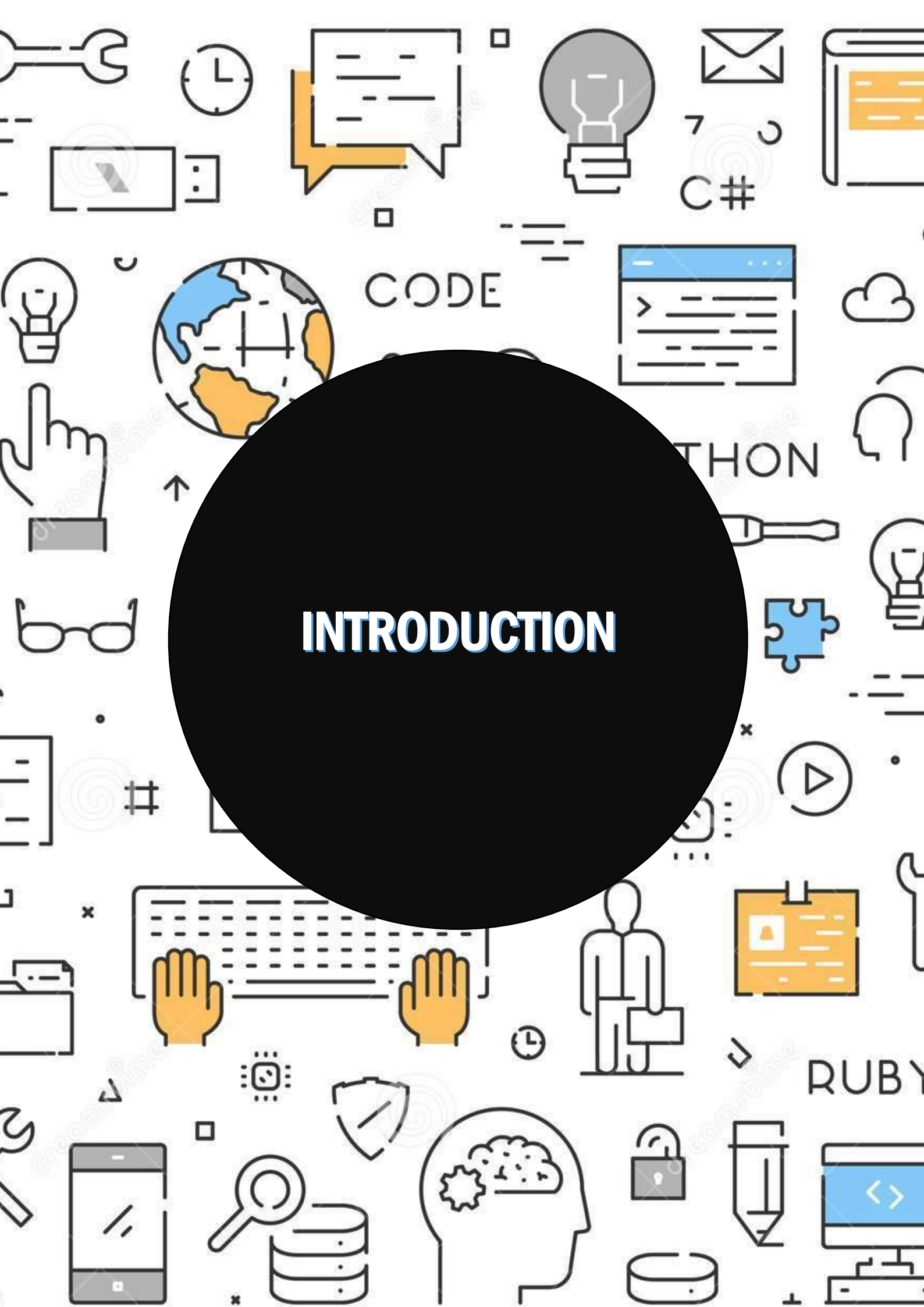
It would be my utmost pleasure to express my sincere gratitude to my Computer Science Teacher **Mr. YADAV SINGH SIR** in providing a helping hand in this project. Their valuable guidance, support and supervision all through this project titled "**VIRTUAL ASSISSTANT**", are responsible for attaining its present form.

I would like to thank him for teaching us computer science from the very basics thus strengthening our root and making us understand complex chapters easily.

This project has been made not only for fetching marks but also for knowledge.

Table of Contents




INTRODUCTION

Python is an interpreted, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high-level dynamic data types, and classes.

Python combines remarkable power with very clear syntax. Python permits its users to write programs and files in fewer lines of code as compared to other programming languages like C++ and Java.

The program can be modified at any instant in accordance with the user's requirements.

Python provides the user to write clear programs both on a large and small scale.

History of Python

Python was perceived in the late 1980s and its implementation was started in December 1989 by Guido van Rossum at CWI in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling and interfacing with the Amoeba operating system.

Guido Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community, benevolent dictator for life (BDFL).

Many of Python's features originated from an interpreted language called ABC. Rossum wanted to correct some of ABC's problems and keep some of its features. Guido Van Rossum published the first version of Python code (version 0.9.0) at in February 1991.

This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.

Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced.

This release included list comprehensions, a full garbage collector and it was supporting Unicode. With this release the development process was changed and became more transparent and community backed.

Significant Features of Python

❖ Easy to use:

Python is a very developer-friendly language which means that anyone and everyone can learn to code it in a couple of hours or days. As compared to other object-oriented programming languages like Java, C, C++, and C#, Python is one of the easiest to learn.

❖ Open and free source:

Python is an open-source programming language which means that anyone can create and contribute to its development. Python has an online forum where thousands of coders gather daily to improve this language further. Along with this Python is free to download and use in any operating system, be it Windows, Mac or Linux.

❖ Support for GUI:

GUI or Graphical User Interface is one of the key aspects of any programming language because it can add flair to code and make the results more visual. Python has support for a wide array of GUIs which can easily be imported to the interpreter, thus making this one of the most favorite languages for developers.

❖ High level language:

Python has been designed to be a high-level programming language, which means that when you code in Python you don't need to be aware of the coding structure, architecture as well as memory management.

Uses of python

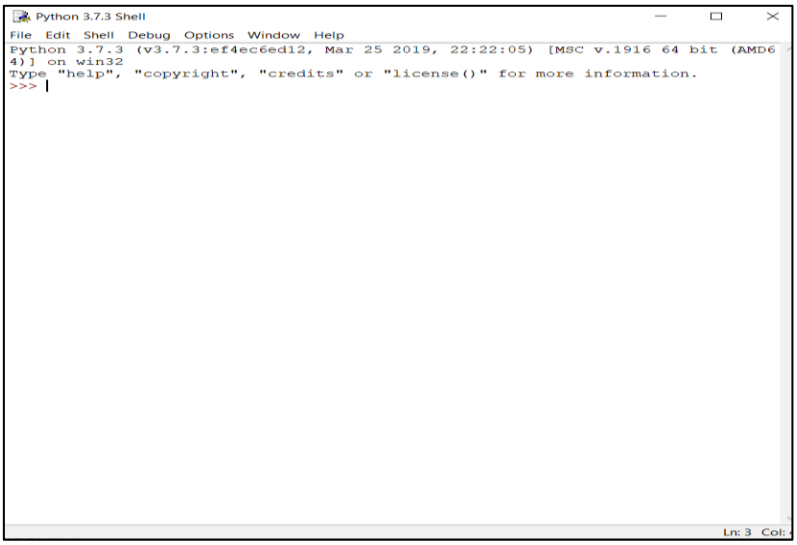
Python is in the following places:

- ❖ In operations pf google search engine, YouTube, etc.
- ❖ Intel, cisco, HP, IBM, etc. use python for hardware testing.
- ❖ The popular social media application “INSTAGRAM” was made by just using python.
- ❖ I-robot uses python to develop commercial Robot.
- ❖ NASA and others use python for their scientific programming task.

Types of python shells

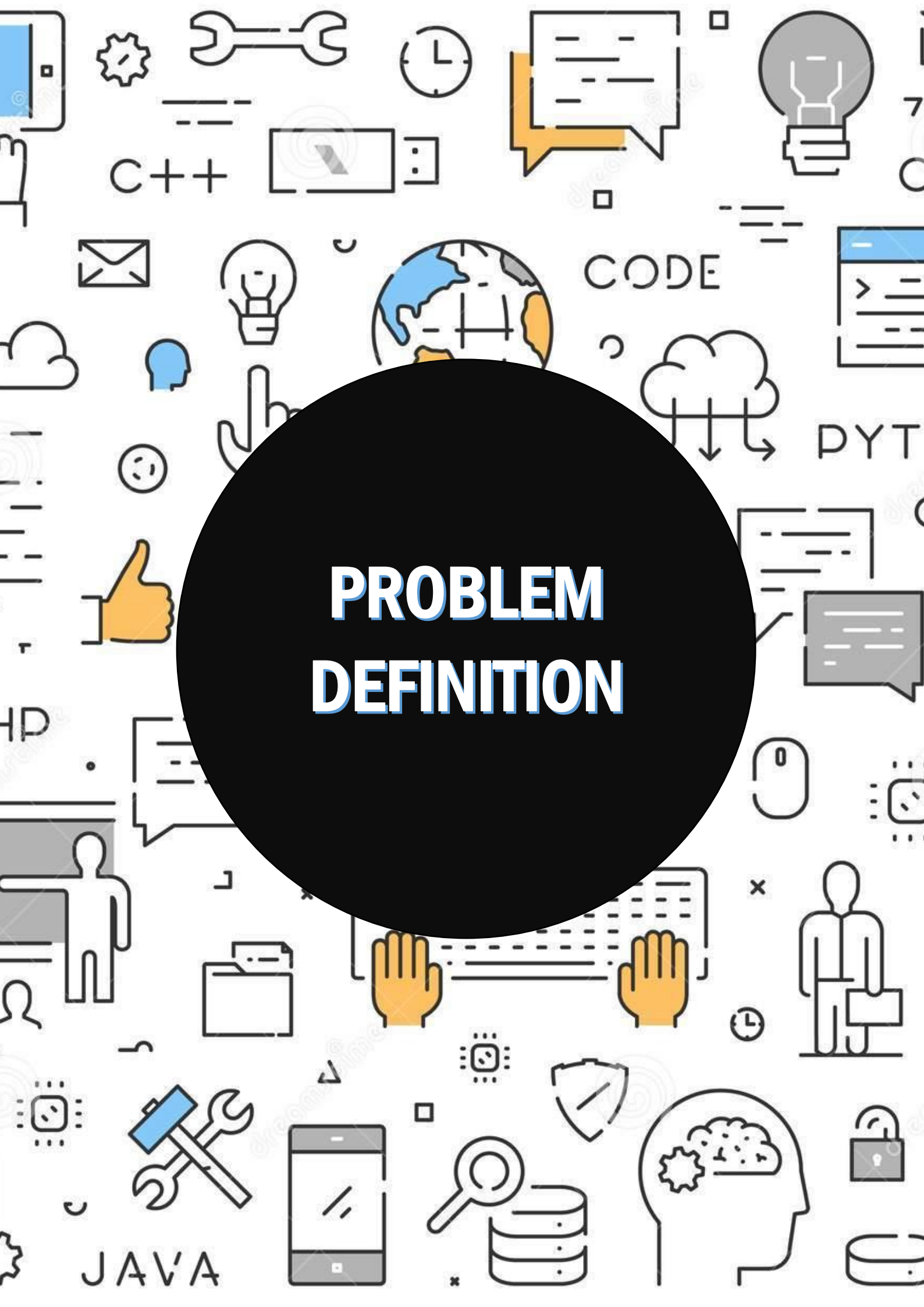
1. Interactive Mode

Working in the interactive mode we will start python on our computer. When we start up the IDLE what we see is a welcome message of python interpreter with revision details and python prompt, i.e, '>>>'. This primary prompt indicating that the interpreter is expecting a python command. Interpreter uses prompt to indicate that it is ready for instructions.



2. Script Mode

In script mode, we type python program in a file and then use the interpreter to execute the content from the file. Working in interactive mode is convenient for beginners and for the testing small pieces of code, as we can test them immediately. But for coding more than a few lines, we should always save our code so that we may modify and reuse the code.



PROBLEM DEFINITION

Aim

To make a “Virtual Assistant”, which takes the task assigned by the user then completes it for the user.

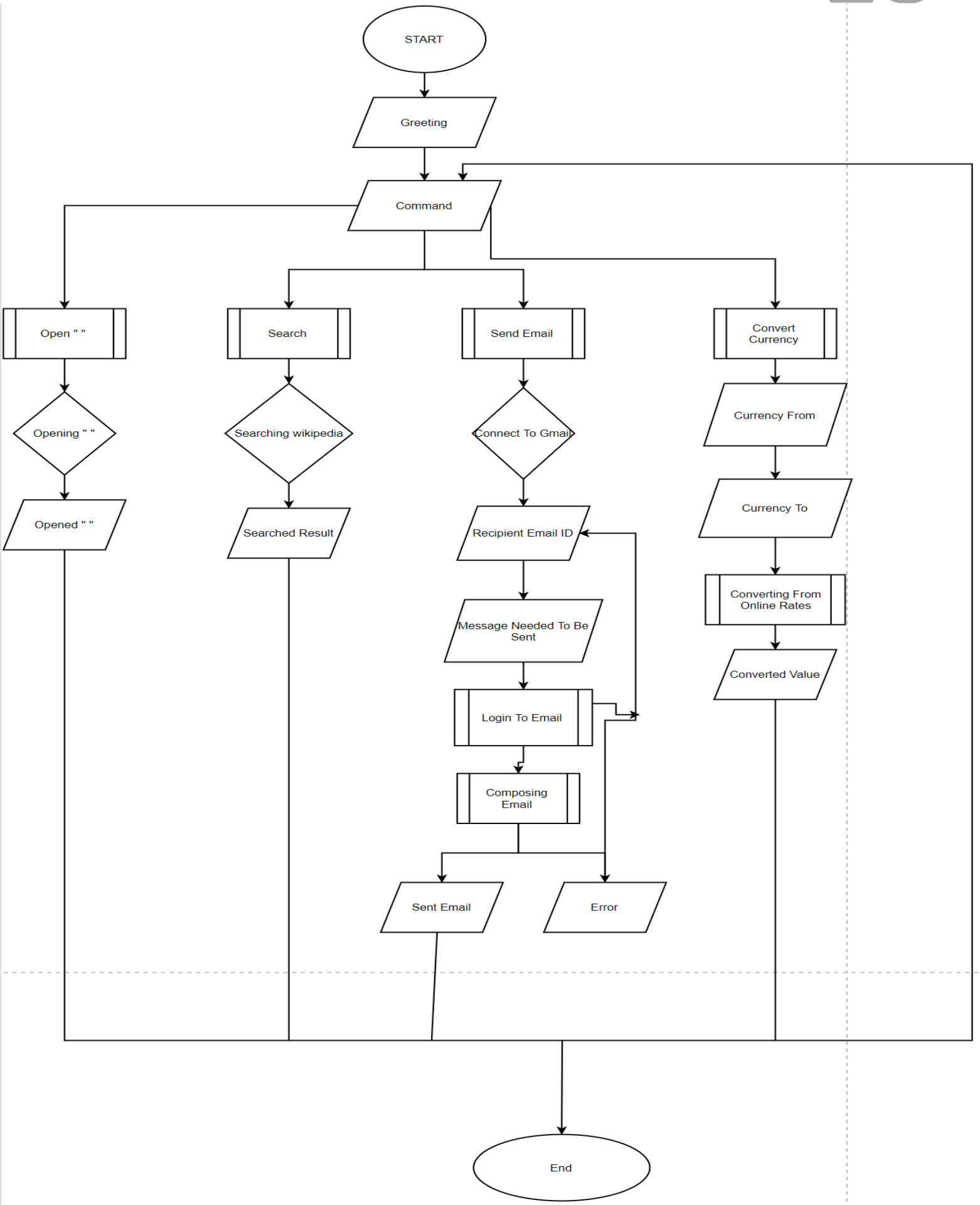
Objective

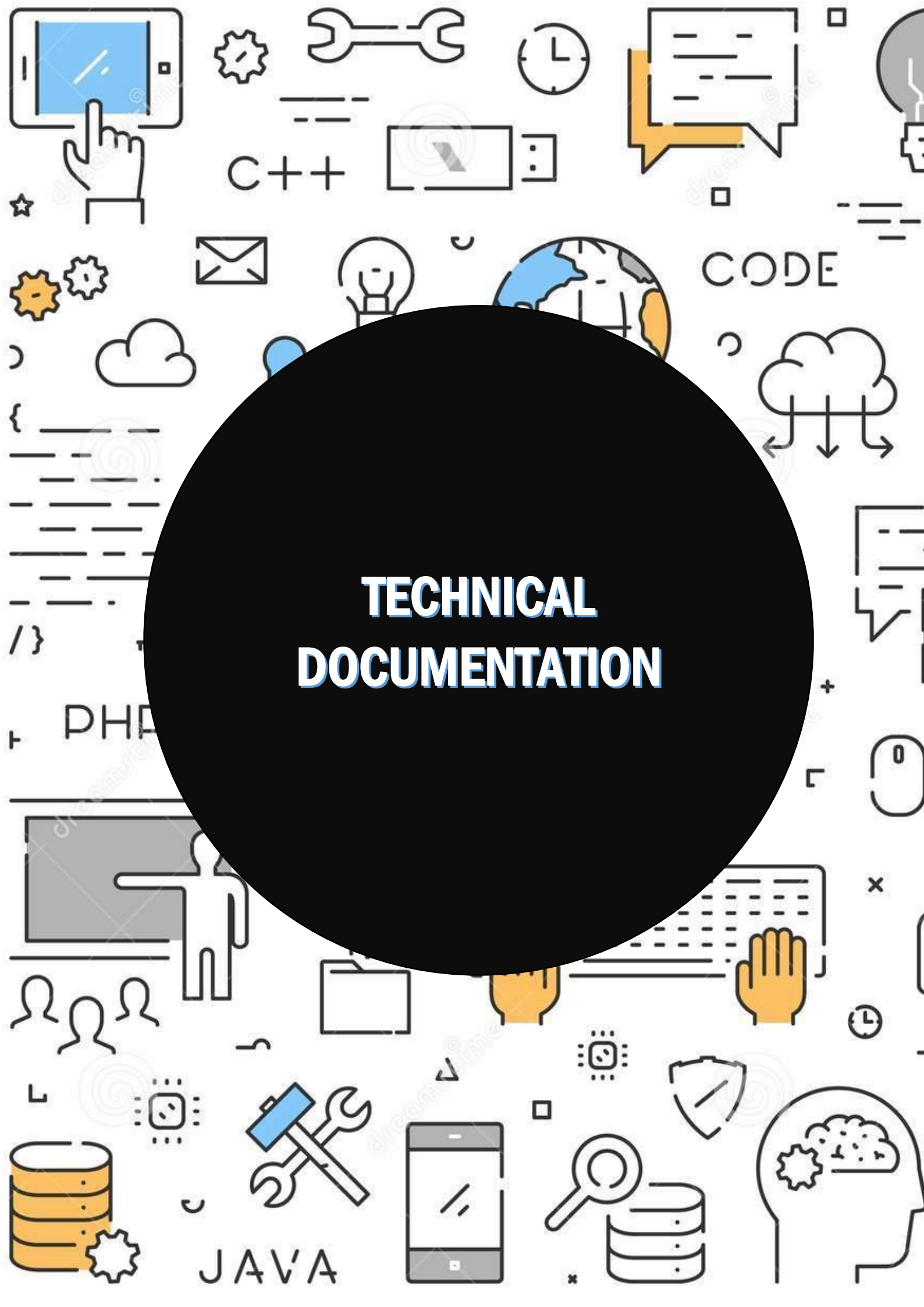
The program is a “Virtual Assistant” , which is made to make the life of the user easier by doing the simple task given by the user to “Virtual Assistant”.

The user can ask the “Virtual Assistant” Multiple task for it like play music convert currency and open apps and websites and may other things.

In the end, the user can get the “Virtual Assistant” To multiple task until his satisfaction.

FLOWCHART


START
Greeting
Command
Open " "
Search
Send Email
Convert Currency
Opening " "
Searching wikipedia
Connect To Gmail
Currency From
Opened " "
Searched Result
Recipient Email ID
Currency To
Message Needed To Be Sent
Converting From Online Rates
Converted Value
Login To Email
Composing Email
Sent Email
Error
End

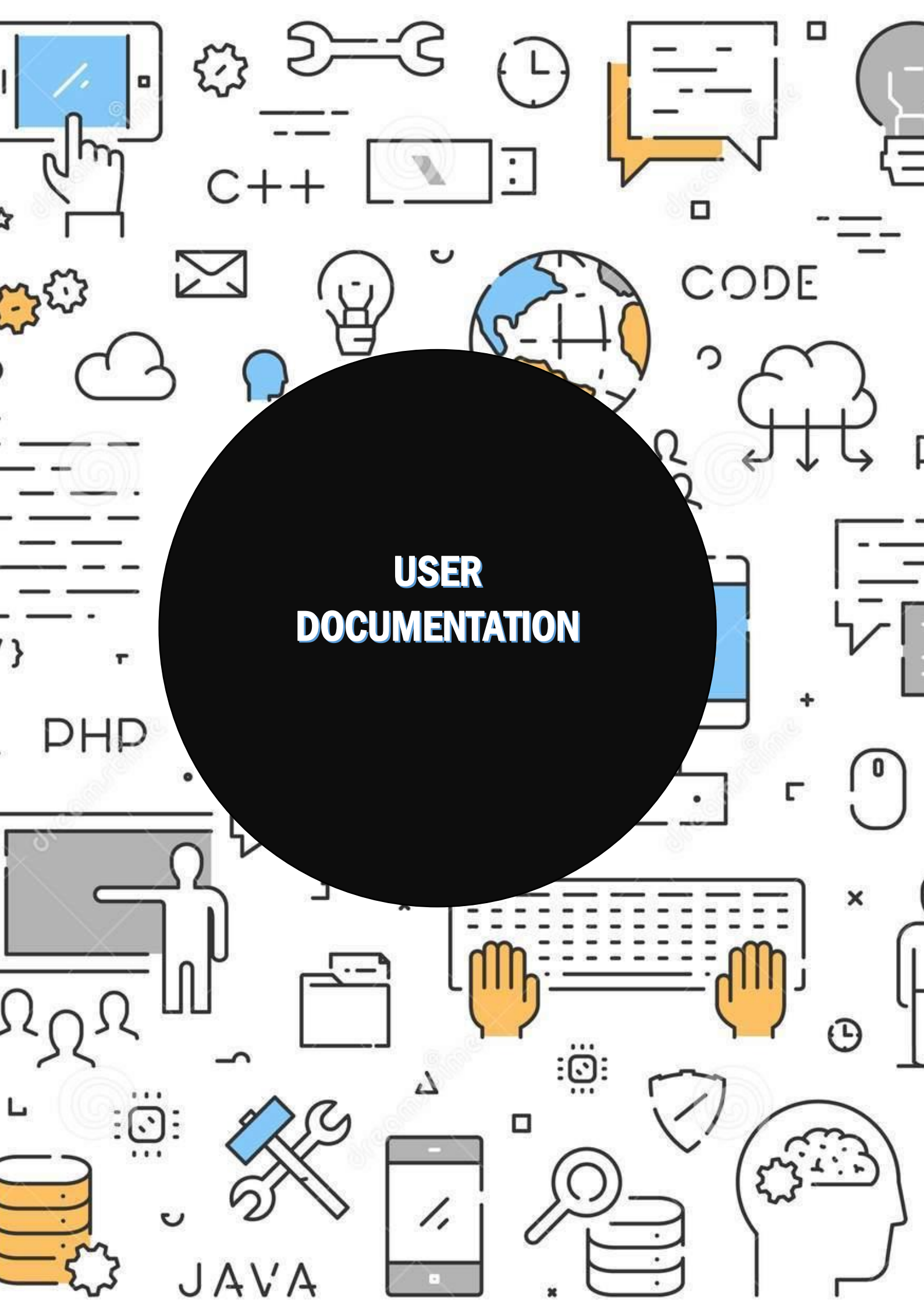


TECHNICAL DOCUMENTATION

Our program of "Virtual Assistant" was coded individually by our members and was then grouped and accessed through the main function. For coding this program, we used the following modules/features of python:

1. **Website Module:** We used the website module to open the desired URL through python.
2. **'def' function:** It is used to call all the functions specified above and print various menus. Also, opens and uses files, and creates objects of the classes used.
3. **Dictionary:** We used dictionary to make a dictionary of the countries and the states in the selected country.
4. **'For, if, elif and while' loop:** We have used these loops thorough out the program which helps beginners to easily understand the coding.
5. **Pyttsx3 Module:** is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3.
6. **OS Module:** The OS module in python provides functions for interacting with the operating system. OS, comes under Python's standard utility modules. This module provides a portable way of using operating system dependent functionality.
7. **Datetime Module:** 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.
8. **GTTS Module:** GTTS (Google Text-to-speech), a python library and CLI tool to interface with Google Translate text-to-speech API.
9. **Engineio Module:** This project implements Python based Engine.IO client and server that can run standalone or integrated with a variety of Python web frameworks and applications.

10. **Warnings Module:** The warnings filter controls whether warnings are ignored, displayed, or turned into errors (raising an exception).
11. **Calendar Module:** This module allows you to output calendars like the Unix cal program, and provides additional useful functions related to the calendar.
12. **Speech_Recognition Module:** Library for performing speech recognition, with support for several engines and API's, online and offline.
13. **Def function:** It is used to call all the functions specified above and print various menus. Also, opens and uses files, and creates objects of the classes used.
14. **Wikipedia Module:** Python library that makes it easy to access and parse data from Wikipedia.
15. **Wolframalpha Module:** It is an API which can compute expert-level answers using Wolfram's algorithms, knowledgebase and AI technology. It is made possible by the Wolfram Language.
16. **Random Module:** imports the random module, which contains a variety of things to do with random number generation
17. **Mysql.connector Module:** MySQL Connectors a driver for connecting to a MySQL database server through the Open Database Connectivity (ODBC) application program interface (API), which is the standard means of connecting to any database.
18. **Time Module:** The time function returns the number of seconds passed since epoch.
19. **Smtplib Module:** The smtplib module defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon
20. **Requests Module:** The requests module allows you to send HTTP requests using Python.

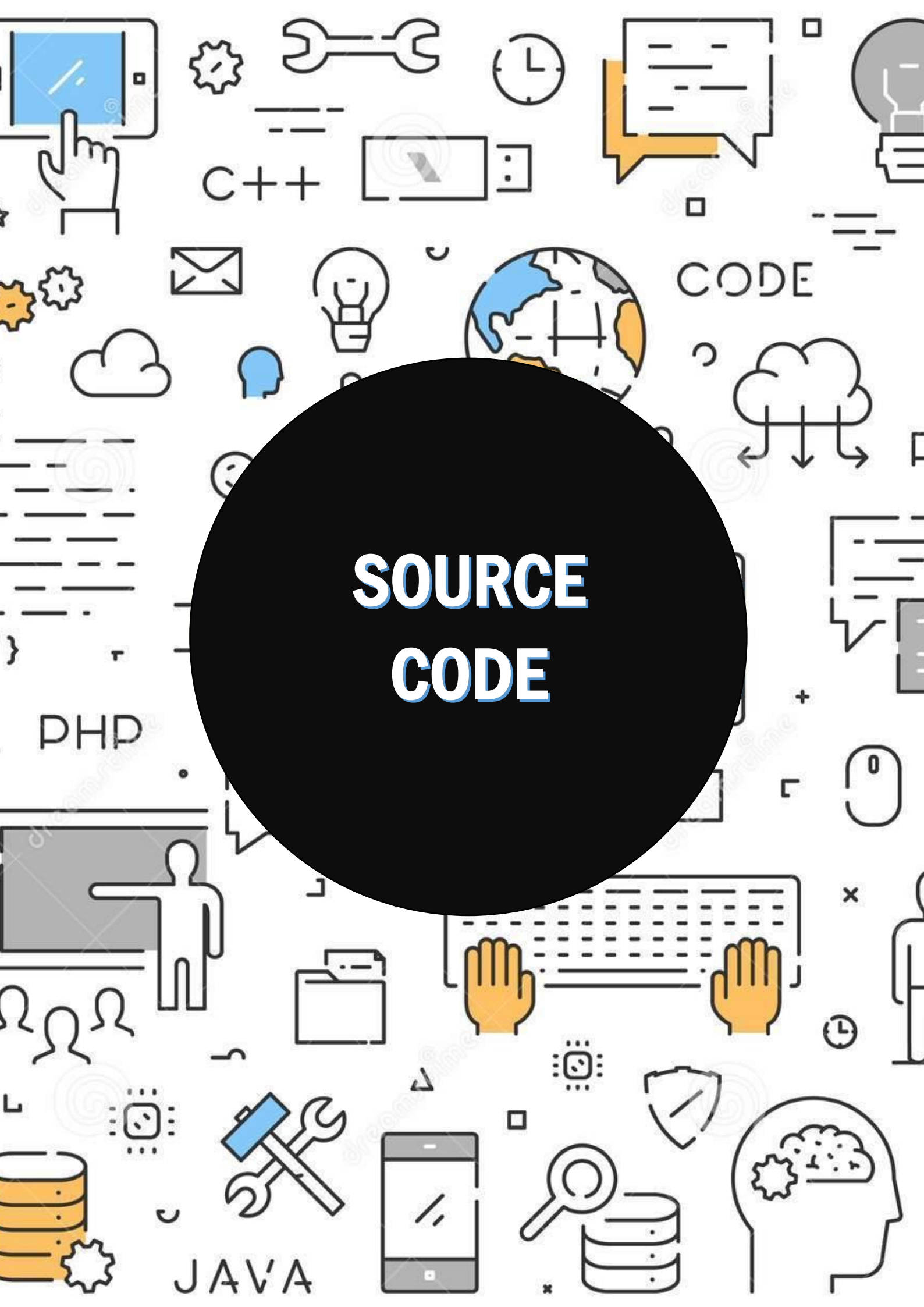


USER DOCUMENTATION

- ❖ As we know researches on AI system vastly increase since the last decade. Everyday new AI model/prototype has been introduced by the people around the globe.
- ❖ By following the same flow this project comes into play. This project is only one of its kinds since various features of different categories will be installed.
- ❖ We have seen that these types of system are dedicated to a specific task, but this system also function other tasks too. Below are some tasks on which this system shall function:

    - Chat-bot
    - Providing answers to user about his/her queries from search engine over the internet
    - Personal assistant

- ❖ Above are the main features on which this system will function. Additional features could also be installed.
- ❖ This system will take speech commands as input. Input will then send serially to the application. Application will then process the command and generate a code.
- ❖ Since this project function by getting commands from the user, so it's necessary that user should be able to pronounce English commands to make system function. Except this, no special skills are required.


SOURCE
CODE
C++
CODE
PHP
JAVA

```python
import pyttsx3
import os
import datetime
from gtts import gTTS
import engineio
import warnings
import calendar
import webbrowser
import speech_recognition as sr
import wikipedia
import wolframalpha
import random
import mysql.connector
import time
import smtplib
import requests

engineio = pyttsx3.init('sapi5')
client = wolframalpha.Client('P29A7R-A25WR2RW6J')
voices = engineio.getProperty('voices')
engineio.setProperty('voice', voices[len(voices)-1].id)
rate = engineio.getProperty('rate')
engineio.setProperty('rate', 200)
engineio.setProperty('voice', voices[0].id)
vol = engineio.getProperty('volume')
engineio.setProperty('volume', 1)
command = ['hey', 'hello', 'yo']
reply = ['welcome', 'hello', "what 's app"]
db = mysql.connector.connect(host='localhost', user='root', passwd='Ahmed123@22')
con = db.cursor()


def speak(text):
    engineio.say(text)
    engineio.runAndWait()


def wishme():
    currentH = int(datetime.datetime.now().hour)
    if currentH >= 0 and currentH < 12:
        speak('Good Morning Sir!')
        speak('My Name is FRIDAY, Your virtual assistant!')
        speak('How May I help you')

    if currentH >= 12 and currentH < 18:
        speak('Good Afternoon Sir!')
        speak('My Name is FRIDAY, Your virtual assistant!')
        speak('How May I help you')

    if currentH >= 18 and currentH != 0:
        speak('Good Evening Sir!')
        speak('My Name is FRIDAY, Your virtual assistant!')
        speak('How May I help you')


def greeting(text):
    GREETING_INPUTS = ['hi', 'hey', 'hola', 'greetings', 'wassup', 'hello']

    GREETING_RESPONSES = ['howdy', 'whats good', 'hello', 'hey there']

    for word in text.split():
```

```
            if word.lower() in GREETING_INPUTS:
                return random.choice(GREETING_RESPONSES) + '.'

    return ''


def takecommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 1
        audio = r.listen(source)
    try:
        print("Recogizing...")
        query = r.recognize_google(audio, language="en-in")
        print(f"user Said {query}\n")

    except Exception as e:
        speak("Sorry sir! I didn\'t get that! Try typing the command!")
        query = str(input('Command: '))
    return query


passwd='Ahmed123@22'

if __name__ == "__main__":
    wishme()
    while True:
        query = takecommand().lower()
        if "wikipedia" in query.lower():
            speak("Seaching please wait")
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=2)
            speak("According to Wikipedia")
            speak(results)

        elif 'open youtube' in query:
            speak('ok sir, opening youtube')
            webbrowser.open('youtube.com')
            time.sleep(10)

        elif 'open gmail' in query:
            speak('ok sir, opening youtube')
            webbrowser.open('www.gmail.com')
            time.sleep(10)

        elif 'open google' in query.lower():
            speak('ok sir, opening google')
            webbrowser.open('google.com')
            time.sleep(10)

        elif 'open instagram' in query:
            speak('ok sir, opening instagram')
            webbrowser.open('www.instagram.com')
            time.sleep(10)

        elif 'open discord' in query:
            speak('ok sir, opening discord')
            webbrowser.open('www.discord.com')
            time.sleep(10)

```

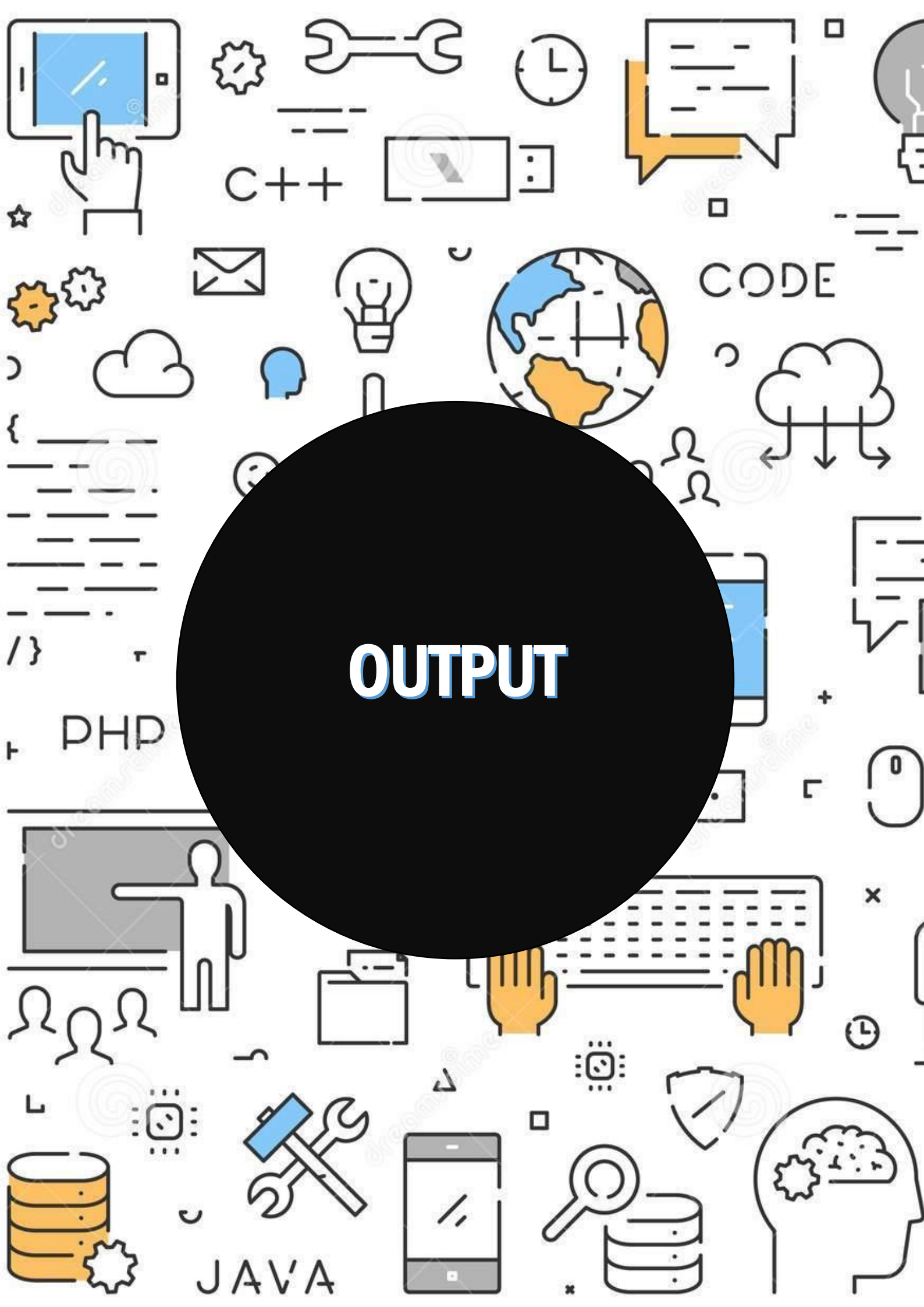
```
    elif 'open reddit' in query:
        speak('ok sir, opening reddit')
        webbrowser.open('www.reddit.com')
        time.sleep(10)

    elif 'open pinterest' in query:
        speak('ok sir, opening pinterest')
        webbrowser.open("www.pinterest.com")
        time.sleep(10)

    elif 'play music' in query:
        music_folder = 'C:\\Users\\Dhia\\Music\\Music\\'
        music = ['Luicid_Dreams','Osas.mp3']
        random_music = music_folder + random.choice(music) + '.mp3'
        os.system(random_music)

        speak('Okay, here is your music! Enjoy!')
        time.sleep(20)

    elif 'open crunchyroll' in query:
        speak('ok sir, opening crunchyroll')
        webbrowser.open('www.crunchyroll.com')
        time.sleep(10)

    elif "convert currency" in query:

        api_key = 'S80I2IDHP41FI8V7'
        speak("enter currency code to covert from ")
        from_c = takecommand().lower()
        speak("enter currency code to covert to ")
        to_c = takecommand().lower()
        f = from_c.upper()
        t = to_c.upper()
        speak("enter the amount:")
        amt = float(takecommand())
        base = 'https://www.alphavantage.co/query?function=CURRENCY_EXCHANGE_RATE'
        main = base + '&from_currency=' + f + '&to_currency=' + t + '&apikey=' + api_key
        response = requests.get(main)
        result = response.json()
        key=result['Realtime Currency Exchange Rate']
        rate=key['5. Exchange Rate']
        rate=float(rate)
        print(rate)
        print(type(rate))
        print()
        print('Realtime Exchange Rate')
        print(f'1{f}:{rate} {t}')
        print()
        print('Converted Amount ')
        print(f'{amt} {f}:{float(rate) * amt} {t}')


    elif 'open whatsapp' in query:
        speak('ok sir, opening whatsapp')
        webbrowser.open('web.whatsapp.com')
        time.sleep(10)

    elif 'open vox cinemas' in query:
        speak("ok sir, opening vox cinemas")
        webbrowser.open("https://uae.voxcinemas.com/")
        time.sleep(10)
```

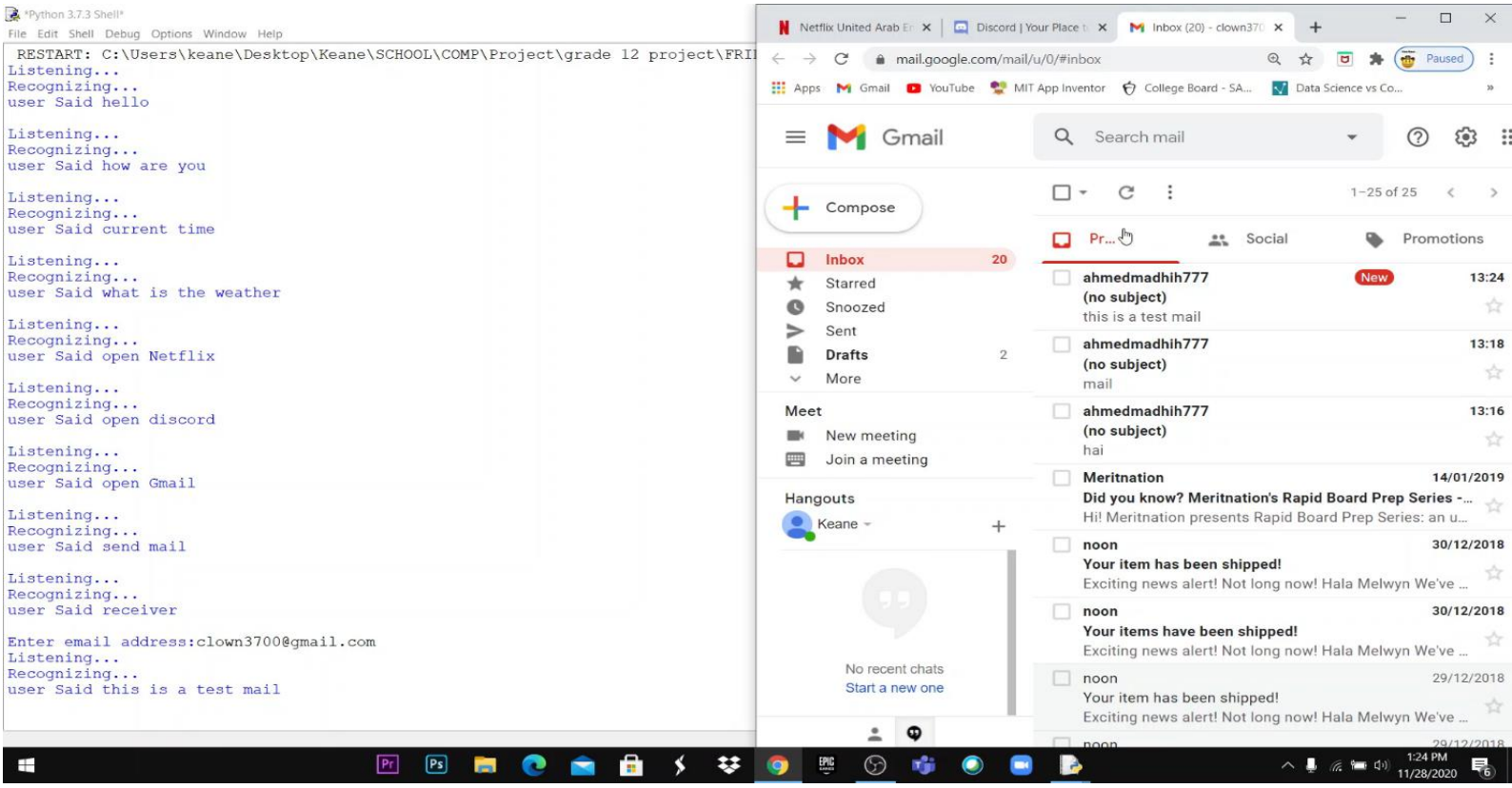
```
    elif 'open zoom' in query:
        speak("ok sir, opening zoom")
        webbrowser.open("www.zoom.us")
        time.sleep(10)

    elif 'order food' in query:
        speak("ok sir, opening several websites for you ")
        webbrowser.open("www.mcdonalds.com")
        webbrowser.open("uae.kfc.me")
        webbrowser.open("www.zomato.com")
        webbrowser.open("www.talabat.com")
        webbrowser.open("www.swiggy.com")
        webbrowser.open("www.deliveroo")
        time.sleep(10)

    elif 'open staz play' in query:
        speak("ok sir, opening staz play")
        webbrowser.open("https://arabia.starzplay.com/")
        time.sleep(10)

    elif 'open prime video' in query:
        speak('ok sir, primevideo')
        webbrowser.open('www.primevideo.com')
        time.sleep(10)

    elif 'open netflix' in query:
        speak('ok sir, opening netflix')
        webbrowser.open('www.netflix.com')
        time.sleep(10)

    elif 'open microsoft teams' in query:
        speak('ok sir, opening microsoft teams')
        webbrowser.open('www.teams.microsoft.com')
        time.sleep(10)

    elif 'open google meet' in query:
        speak('ok sir, open Google meet')
        webbrowser.open('www.meet.google.com')
        time.sleep(10)

    elif 'open krunker' in query:
        speak('ok sir, opening krunker')
        webbrowser.open('www.krunker.com')
        time.sleep(10)

    elif 'open code' in query:
        speak('ok sir, opening the code')
        codepath = "C:\\Users\\Dhia\\AppData\\Local\\Programs\\Python\\Python36\\FRIDAY project.py"
        os.startfile(codepath)

    elif "open tic tac toe" in query:
        speak("ok sir, opening tic tac toe")
        Game = "C:\\Users\\Dhia\\AppData\\Local\\Programs\\Python\\Python36\\tic tac toe.py"
        os.startfile(Game)
        time.sleep(10)

    elif "open calculator" in query:
        speak("ok sir, opening calculator")
        calc = "C:\\Users\\Dhia\\AppData\\Local\\Programs\\Python\\Python36\\Calculator.py"
        os.startfile(calc)
        time.sleep(10)
```

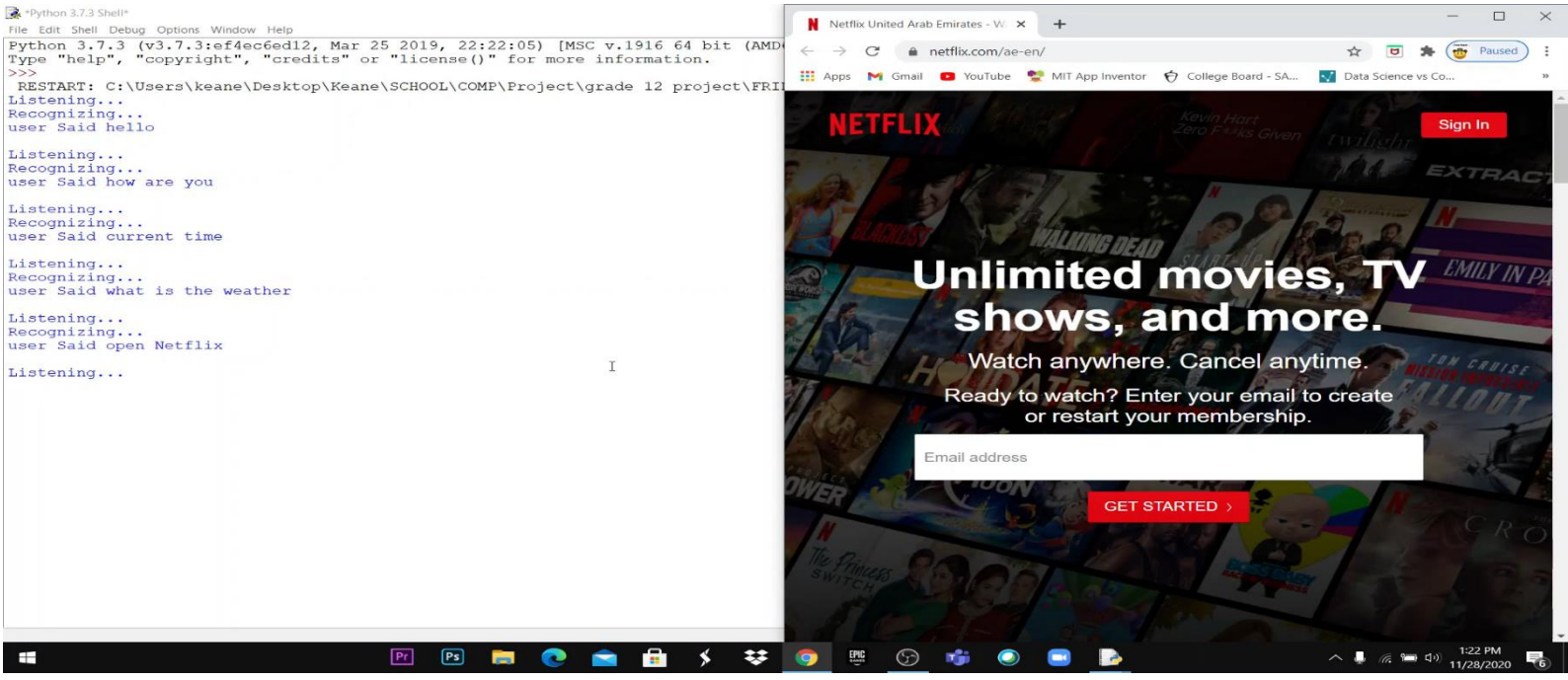
```
    elif "play songs" in query:
        speak('okay')
        print()

    elif "current time" in query:
        speak(datetime.datetime.now())

    elif "what\'s up" in query or 'how are you' in query:
        stMsgs = ['Just doing my thing!', 'I am fine!', 'Nice!', 'I am nice and full of energy']
        speak(random.choice(stMsgs))

    elif 'hello' in query:
        speak('Hello Sir')

    elif "open gmail" in query:
        speak('okay')
        webbrowser.open_new("gmail.com")
        time.sleep(10)

    elif 'send mail' in query:
        speak('Who is the recipient? ')
        recipient = takecommand()
        speak('okay, Enter the email address')
        receiver=input(str("Enter email address:"))

        if 'receiver' in recipient:
            try:
                speak('What should I say? ')
                content = takecommand()

                server = smtplib.SMTP('smtp.gmail.com', 587)
                server.ehlo()
                server.starttls()
                server.login("ahmedmadhih777@gmail.com", passwd)
                server.sendmail('ahmedmadhih777@gmail.com', receiver, content)
                server.close()
                speak('Email sent!')

            except:
                speak('Sorry Sir! I am unable to send your message at this moment!')


    elif 'bye' in query:
        speak('Bye Sir, have a good day.')
        quit()

    elif 'nothing' in query or 'abort' in query or 'stop' in query:
        speak('okay')
        speak('Bye Sir, have a good day.')
        quit()

    elif 'search from database' in query:
        speak('Which database sir')
        database = takecommand().lower()
        con.execute(f'use {database}')
        speak('say table name')
        table = takecommand().lower()
        try:
            con.execute(f'select * from {table}')
            re = con.fetchall()
            for c in re:
```

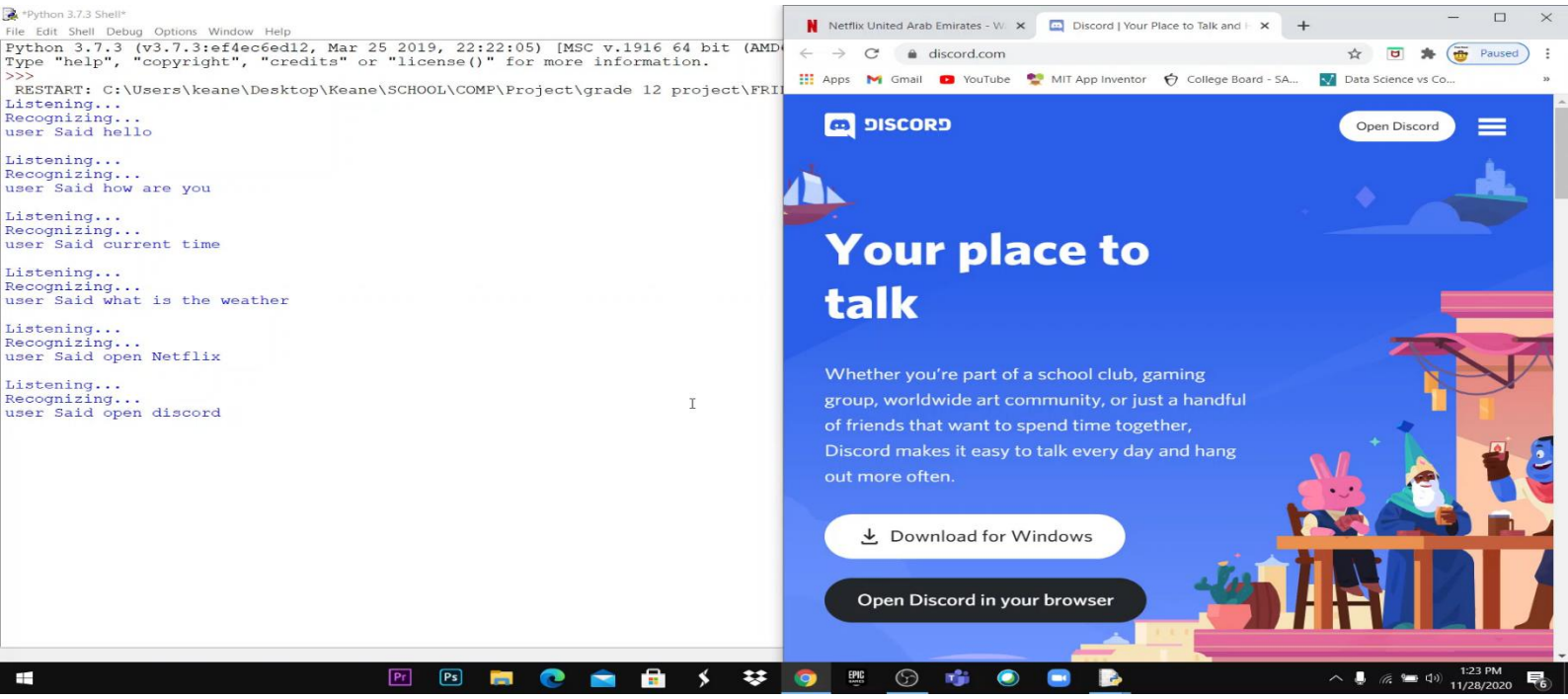
```
                for c in re:
                    print(c)

            except:
                speak('table not found')

        elif query in command:
            w = random.randint(0, 2)
            speak(reply[w])

        elif "wait for some time" in query:
            time.sleep(10)

        else:
            query = query
            speak('Searching...')
            try:
                try:
                    res = client.query(query)
                    results = next(res.results).text
                    speak('Got it. ')
                    speak('WOLFRAM-ALPHA says -')
                    speak(results)

                except:
                    results = wikipedia.summary(query, sentences=2)
                    speak('Got it.')
                    speak('WIKIPEDIA says - ')
                    speak(results)

            except:
                webbrowser.open('www.google.com')

        speak('Next Command! Sir!')


def assistantResponse(text):
    print(text)

    myobj = gTTS(text=text, lang='en', slow=False)

    myobj.save('assistant_response.mp3')

    os.system('start assistant_response.mp3')


def wakeWord(text):
    WAKE_WORDS = ['hey computer', 'okay computer']
```

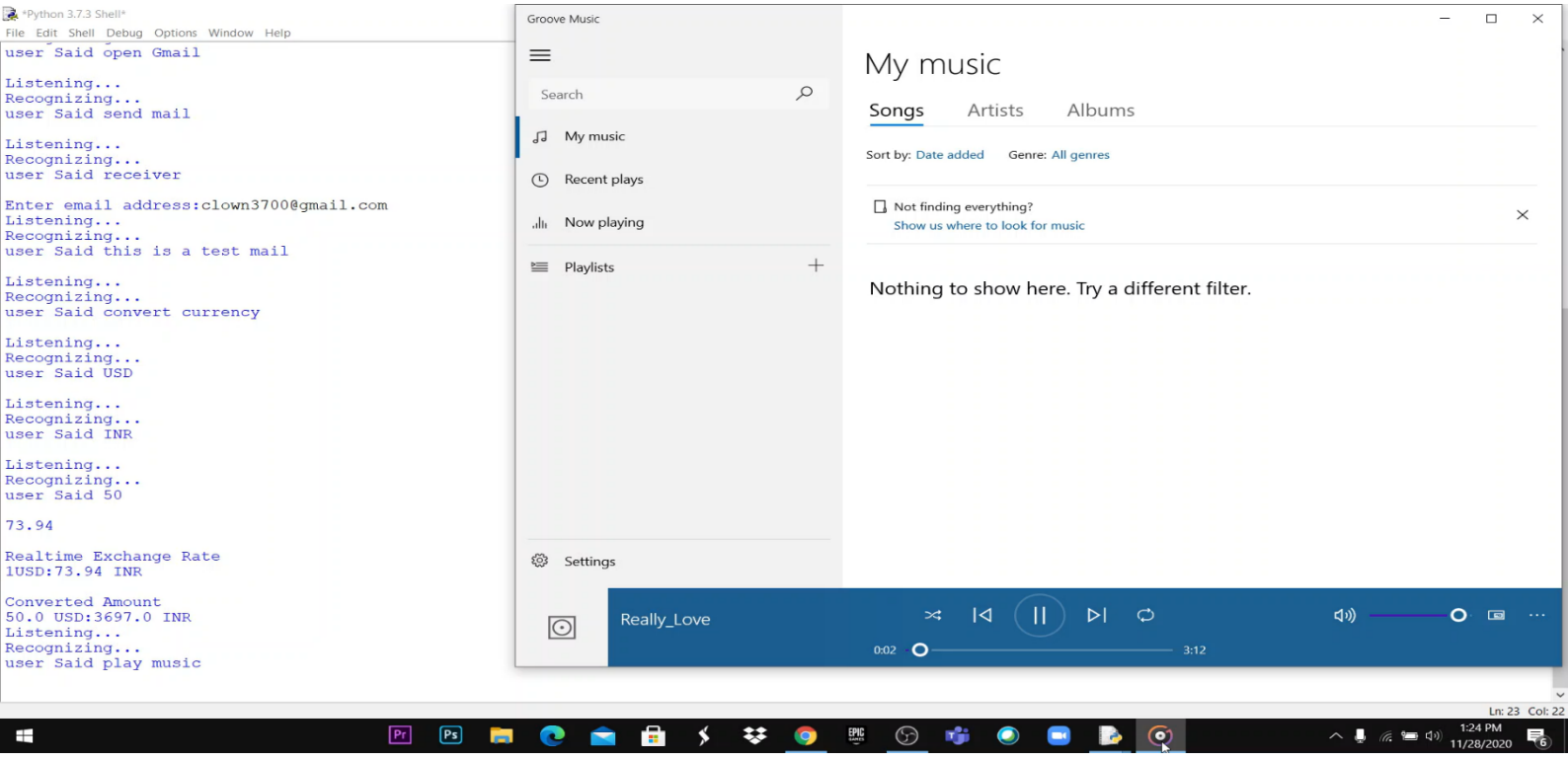
```
355
356         text = text.lower()
357         for phrase in WAKE_WORDS:
358             if phrase in text:
359                 return True
360
361         return False
362
363
364     def getPerson(text):
365         wordList = text.split()
366
367         for i in range(0, len(wordList)):
368             if i + 3 <= len(wordList) - 1 and wordList[i].lower() == 'who' and wordList[i + 1].lower() == 'is':
369                 return wordList[i + 2] + ' ' + wordList[i + 3]
370
371
372     while True:
373
374         text = recordAudio()
375         response = ''
376
377         if (wakeWord(text) == True):
378
379             response = response + greeting(text)
380
381             if ('date' in text):
382                 get_date = getDate()
383                 response = response + ' ' + get_date
384
385             if ('time' in text):
386                 now = datetime.datetime.now()
387                 meridiem = ''
388                 if now.hour >= 12:
389                     meridiem = 'p.m'
390                     hour = now.hour - 12
391                 else:
392                     meridiem = 'a.m'
393                     hour = now.hour
394
395                 if now.minute < 10:
396                     minute = '0' + str(now.minute)
397                 else:
398                     minute = str(now.minute)
399
400                 response = response + ' ' + 'It is ' + str(hour) + ':' + minute + ' ' + meridiem + ' .'
401
402             if ('who is' in text):
403                 person = getPerson(text)
404                 wiki = wikipedia.summary(person, sentences=2)
405                 response = response + ' ' + wiki
406
407             assistantResponse(response)
408
```


C++
CODE
PHP
JAVA
OUTPUT


Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
RESTART: C:\Users\keane\Desktop\Keane\SCHOOL\COMP\Project\grade 12 project\FRI
Listening...
Recognizing...
user Said hello
Listening...
Recognizing...
user Said how are you
Listening...
Recognizing...
user Said current time
Listening...
Recognizing...
user Said what is the weather
Listening...
Recognizing...
user Said open Netflix
Listening...
Recognizing...
user Said open discord
Listening...
Recognizing...
user Said open Gmail
Listening...
Recognizing...
user Said send mail
Listening...
Recognizing...
user Said receiver
Enter email address:clown3700@gmail.com
Listening...
Recognizing...
user Said this is a test mail
Gmail
Search mail
Compose
Inbox 20
Starred
Snoozed
Sent
Drafts 2
More
Meet
New meeting
Join a meeting
Hangouts
Keane
No recent chats
Start a new one
Social
Promotions
ahmedmadhih777 (no subject) this is a test mail 13:24
ahmedmadhih777 (no subject) mail 13:18
ahmedmadhih777 (no subject) hai 13:16
Meritnation 14/01/2019
Did you know? Meritnation's Rapid Board Prep Series -...
noon 30/12/2018
Your item has been shipped!
noon 30/12/2018
Your items have been shipped!
noon 29/12/2018
Your item has been shipped!
1:24 PM
11/28/2020


Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\keane\Desktop\Keane\SCHOOL\COMP\Project\grade 12 project\FRI
Listening...
Recognizing...
user Said hello
Listening...
Recognizing...
user Said how are you
Listening...
Recognizing...
user Said current time
Listening...
Recognizing...
user Said what is the weather
Listening...
Recognizing...
user Said open Netflix
Listening...
NETFLIX
Sign In
Unlimited movies, TV shows, and more.
Watch anywhere. Cancel anytime.
Ready to watch? Enter your email to create or restart your membership.
Email address
GET STARTED
1:22 PM
11/28/2020


Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\keane\Desktop\Keane\SCHOOL\COMP\Project\grade 12 project\FRI
Listening...
Recognizing...
user Said hello
Listening...
Recognizing...
user Said how are you
Listening...
Recognizing...
user Said current time
Listening...
Recognizing...
user Said what is the weather
Listening...
Recognizing...
user Said open Netflix
Listening...
Recognizing...
user Said open discord
Netflix United Arab Emirates - W
Discord | Your Place to Talk and
discord.com
Paused
Apps
Gmail
YouTube
MIT App Inventor
College Board - SA...
Data Science vs Co...
DISCORD
Open Discord
Your place to talk
Whether you're part of a school club, gaming group, worldwide art community, or just a handful of friends that want to spend time together, Discord makes it easy to talk every day and hang out more often.
Download for Windows
Open Discord in your browser
1:23 PM
11/28/2020


Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
user Said open Gmail
Listening...
Recognizing...
user Said send mail
Listening...
Recognizing...
user Said receiver
Enter email address:clown3700@gmail.com
Listening...
Recognizing...
user Said this is a test mail
Listening...
Recognizing...
user Said convert currency
Listening...
Recognizing...
user Said USD
Listening...
Recognizing...
user Said INR
Listening...
Recognizing...
user Said 50
73.94
Realtime Exchange Rate
1USD:73.94 INR
Converted Amount
50.0 USD:3697.0 INR
Listening...
Recognizing...
user Said play music
Groove Music
Search
My music
Recent plays
Now playing
Playlists
Settings
My music
Songs
Artists
Albums
Sort by: Date added
Genre: All genres
Not finding everything?
Show us where to look for music
Nothing to show here. Try a different filter.
Really_Love
0:02
3:12
Ln: 23 Col: 22
1:24 PM
11/28/2020

LIMITATIONS

1. Lack of GUI (graphic user interface)
2. The user can not be heard every time.
3. Limited number of queries can be asked.

SUGGESTIONS FOR IMPROVEMENT

1. Make use of graphic user interface to make it more appealing.
2. Train the assistant to produce better answers.
3. Make a bar to type the queries.

BIBLIOGRAPHY

1. Computer science textbook By: Sumita Arora
2. www.wikipedia.com
3. www.draw.io
4. www.stackoverflow.com
5. www.pypi.org