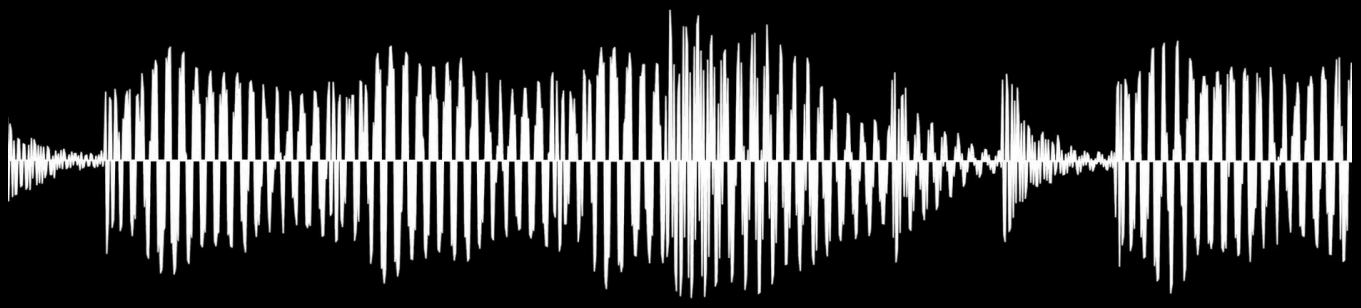


iCompanion

#AssistanceMadeSimple

Kunal Kumar Sahoo

Jash Sheth

Srishti Singh
Class XII-A
Kendriya Vidyalaya No. 1
Sector 30 Gandhinagar

CONTENTS

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank our Principal Sir, Mr. Mohan Chandra Satyawali, for providing us an opportunity to do the project work and giving us all support and guidance which made us complete the project duly. I am extremely thankful to him for his efforts, although he had busy schedule in this Pandemic situation.

We owe our deep gratitude to our Computer Science teacher and Project Guide, Mr. A. K. Meena Sir [PGT CS], who took keen interest on our project work and guided us all along, till the completion of our project by providing us all the necessary information for developing our project, making our work and Computer Science classes just "awesome".

Last but not the least, we would not forget to acknowledge our parents for their kind support, constant motivation and their financial support in our project by providing us all the requirements.

We would also like to mention our seniors and online web-platforms like Stackoverflow, GeeksforGeeks, PyPi and Github for their active and dynamic support of resolving our queries and errors that we encountered during the programming of our project.

Kunal Kumar Sahoo (12124)

Jash Sheth(12119)

Srishti Singh (12149)

Overview

Computer usage has always been typical, although easy, nowadays in this modern digital times, the usage is very competitive as various works are done simultaneously and for productivity works the only things that matter are skills and time. Skills improve by practicing and practicing but there is no alternative for time reduction apart from improving speed which once again develops with skills. We have developed simple but efficient computer programmers for time efficiency in their productivity works for both professionals and amateurs.

Goals

1. To design a tailor-made assistance program for users according to their usage for their systems.
2. To implement all the fundamental programming concepts and skills that we have learned and developed in this current academic year.

Specifications

Operating Systems:

This program has been written exclusively for Windows Operating Systems (Windows 7/8/10). Although this program can also be implemented in Linux Distros (Ubuntu, Debians, Red-Hat etc.) and MacOS (Sierra, Catalina, Big Sur etc.) although not every function can be accessible but the important and main functions would be still usable.

But still for best user experience, Windows 10 is highly recommended.

Development Environment:

This program is fundamentally written in Python Language and hence, any version of Python 3 (Python 3.6/Python 3.8/Python 3.9 etc.) is necessary for execution of this program.

Also, MySQL Community Version software is needed for maintaining a database and hence MySQL software is also required for the same.

Source Code

```
'''iAssistance Source Code
Made by Kunal Kumar Sahoo
        Jash Sheth
        Srishti Singh
'''

from __future__  import print_function

import ctypes
import datetime
import json
import os
import psutil
import os.path
import pickle
import smtplib
import time
import webbrowser
import platform
from termcolor import colored
from urllib.request import urlopen
import pyjokes
import pyttsx3
import pytz
import requests
import speech_recognition as sr
import wikipedia
import winshell
from google.auth.transport.requests import Request
from google_auth_oauthlib.flow import InstalledAppFlow
from googleapiclient.discovery import build
from mysql.connector import MySQLConnection
import csv
import matplotlib.pyplot as plt
from ecapture import ecapture as ec
```

```
def virtualAssistant():

    SCOPES = ['https://www.googleapis.com/auth/calendar.readonly']
    MONTHS = ["january", "february", "march", "april", "may", "june", "july",
"august", "september", "october", "november","december"]
    DAYS = ["monday", "tuesday", "wednesday", "thursday", "friday",
"saturday", "sunday"]
    DAY_EXTENTIONS = ["rd", "th", "st", "nd"]

    def credAccess():
        with open("userCred.txt", 'rt') as fin:
            creds = fin.readlines()
            username = creds[1]
            USERNAME = username[0:len(username) - 1]
            email_id = creds[3]
            USER_EMAIL_ID = email_id[0:len(email_id) - 1]
            password = creds[5]
            USER_EMAIL_PASS = password[0:len(password) - 1]
            sql = creds[7]
            USER_SQL_PASS = sql[0:len(sql) - 1]
            return USERNAME, USER_EMAIL_ID,
USER_EMAIL_PASS, USER_SQL_PASS
```

```
        def speak(text):
                engine = pyttsx3.init("sapi5")  # object creation
                """ RATE"""
                rate = engine.getProperty('rate')  # getting details of current speaking
rate
                engine.setProperty('rate', 125)  # setting up new voice rate

                """VOLUME"""
                volume = engine.getProperty('volume')  # getting to know current
volume level (min=0 and max=1)
                engine.setProperty('volume', 1.0)  # setting up volume level  between 0
and 1

                """VOICE"""
                voices = engine.getProperty('voices')  # getting details of current
voice
                # engine.setProperty('voice', voices[0].id)  #changing index, changes
voices. o for male
                engine.setProperty('voice', voices[0].id)  # changing index, changes
voices. 1 for female

                engine.say(text)
                engine.runAndWait()

        recognitionMode = int(input('''By which mode would you like to give
commands :
                1 - Voice Mode
                2 - Script Mode\n'''))
```

```
def take_command():
        global query, endTime
        if recognitionMode == 1:
                r = sr.Recognizer()
                with sr.Microphone() as source:
                        print("Listening...")
                        r.pause_threshold = 1
                        audio = r.listen(source)
                try:
                        print("Recognizing...")
                        query = r.recognize_google(audio, language='en-in')
                        print(f"User said: {query}\n")
                except Exception as e:
                        print("Say that again please...")
                        return "None"

        elif recognitionMode == 2:
                try:
                        query = input('Enter Command : ')
                except Exception as e:
                        print(e)
        return query.lower()
```

```python
def authenticate_google():
    """Shows basic usage of the Google Calendar API.
    Prints the start and name of the next 10 events on the user's calendar.
    """
    creds = None
    if os.path.exists('token.pickle'):
        with open('token.pickle', 'rb') as token:
            creds = pickle.load(token)

    if not creds or not creds.valid:
        if creds and creds.expired and creds.refresh_token:
            creds.refresh(Request())
        else:
            flow = InstalledAppFlow.from_client_secrets_file(
                'credentials.json', SCOPES)
            creds = flow.run_local_server(port=0)

        with open('token.pickle', 'wb') as token:
            pickle.dump(creds, token)

    service = build('calendar', 'v3', credentials=creds)

    return service
```

```
        def activity(today, initTime, endTime):

                deltaTime = endTime - initTime
                duration = deltaTime.total_seconds() /60

                importedData = []

                with open('screenActivity.csv', 'r') as file:
                        reader = csv.reader(file)
                        for row in reader:
                                importedData.append(row)
                dict = {}
                for i in range(len(importedData)):
                        if len(importedData) != 0:
                                dict[importedData[i][0]] =
round(float(importedData[i][1]), 2)
                for i in dict:
                        if i == today:
                                print(dict[i])
                                dict[i] = duration + dict[i]
                                break
                else:
                        dict[today] = duration
                print(dict)
                with open('screenActivity.csv', 'w') as file:
                        for key in dict.keys():
                                file.write("%s,%s\n" % (key, dict[key]))
                x = []
                y = []
                with open('screenActivity.csv', 'r') as csvfile:
                        plots = csv.reader(csvfile, delimiter=',')
                        for row in plots:
                                x.append((row[0]))
                                y.append(float(row[1]))
                plt.plot(x, y)
                plt.xticks(rotation=30)
                plt.xlabel('Date')
                plt.ylabel('Duration(sec)')
                plt.title('Screen Activity')
                plt.show()
```

```
        def get_events(day, service):
                # Call the Calendar API
                date = datetime.datetime.combine(day, datetime.datetime.min.time())
                end_date = datetime.datetime.combine(day,
datetime.datetime.max.time())
                utc = pytz.UTC
                date = date.astimezone(utc)
                end_date = end_date.astimezone(utc)
                events_result = service.events().list(calendarId='primary',
timeMin=date.isoformat(), timeMax=end_date.isoformat(), singleEvents=True,
orderBy='startTime').execute()
                events = events_result.get('items', [])
                if not events:
                        speak('No upcoming events found.')
                else:
                        speak(f"You have {len(events)} events on this day.")
                        for event in events:
                                start = event['start'].get('dateTime',
event['start'].get('date'))
                                print(start, event['summary'])
                                start_time = str(start.split("T")[1].split("-")[0])
                                if int(start_time.split(":")[0]) < 12:
                                        start_time = start_time + "am"
                                else:
                                        start_time = str(int(start_time.split(":")[0]) - 12) +
start_time.split(":")[1]
                                        start_time = start_time + "pm"

                                speak(event["summary"] + " at " + start_time)

        def get_date(text):
                text = text.lower()
                today = datetime.date.today()
                if text.count("today") > 0:
                        return today
                day = -1
                day_of_week = -1
                month = -1
                year = today.year
```

```
        for word in text.split():
            if word in MONTHS:
                month = MONTHS.index(word) + 1
            elif word in DAYS:
                day_of_week = DAYS.index(word)
            elif word.isdigit():
                day = int(word)
            else:
                for ext in DAY_EXTENTIONS:
                    found = word.find(ext)
                    if found > 0:
                        try:
                            day = int(word[:found])
                        except:
                            pass

        if month < today.month and month != -1:  # if the month mentioned is before the current month set the year to the next
            year = year + 1
        if month == -1 and day != -1:  # if we didn't find a month, but we have a day
            if day < today.day:
                month = today.month + 1
            else:
                month = today.month

        # if we only found a dta of the week
        if month == -1 and day == -1 and day_of_week != -1:
            current_day_of_week = today.weekday()
            dif = day_of_week - current_day_of_week

            if dif < 0:
                dif += 7
                if text.count("next") >= 1:
                    dif += 7

            return today + datetime.timedelta(dif)

        if day != -1:
            return datetime.date(month=month, day=day, year=year)
```

```
        def note(text):
                date = datetime.datetime.now()
                file_name = str(date).replace(":", "-") + "-note.txt"
                with open(file_name, "w") as f:
                        f.write(text)
                speak("I've made a note of that.")
                osCommandString = f"notepad.exe {file_name}"
                os.system(osCommandString)

        def sendEmail(USER_EMAIL_ID, USER_EMAIL_PASS, to, content):
                server = smtplib.SMTP('smtp.gmail.com', 587)
                server.ehlo()
                server.starttls()

                server.login (USER_EMAIL_ID, USER_EMAIL_PASS ')
                server.sendmail(USER_EMAIL_ID,  to, content)
                server.close()

        def history(text):
                date_time = datetime.datetime.now().strftime("%d-%m-%Y,
%H:%M:%S")
                f = open("Command history.txt", "a")
                f.write(f"{date_time} : {text}\n")

        def wishMe(name):
                hour = int(datetime.datetime.now().hour)
                if hour >= 0 and hour < 12:
                        speak("Good Morning!" + str(name))
                elif hour >= 12 and hour < 16:
                        speak("Good Afternoon!" + str(name))
                else:
                        speak("Good Evening!" + str(name))
```

```
        def contactBook(USER_SQL_PASS):
                global emailID
                mydb = MySQLConnection(host="localhost", user="root",
password=USER_SQL_PASS, database="contact_book")
                mycursor = mydb.cursor()
                name1 = str(input("Enter the name : "))
                mycursor.execute("SELECT * FROM contact_table")
                myresult = mycursor.fetchall()
                for x in myresult:
                        if name1 == x[0]:
                                emailID = x[1]
                                break
                else:
                        print("Record not found")
                        insertChoice = input("Would you like to add contact ? ")
                        if insertChoice.lower().startswith("y"):
                                mydb1 = MySQLConnection(host="localhost",
user="root", password="computer", database="contact_book")
                                mycursor1 = mydb1.cursor()
                                mail = str(input("Enter the e-mail ID of the new contact :
"))
                                sql = "INSERT INTO contact_table(name, email_id)
VALUES(%s,%s)"
                                val = (name1, mail)
                                mycursor1.execute(sql, val)
                                mydb1.commit()
                                print(mycursor1.rowcount, "record inserted.")
                        mydb = MySQLConnection(host="localhost", user="root",
password=USER_SQL_PASS, database="contact_book")
                        mycursor = mydb.cursor()
                        mycursor.execute("SELECT * FROM contact_table")
                        myresult = mycursor.fetchall()
                        for x in myresult:
                                if name1 == x[0]:
                                        emailID = x[1]
                                        break
                return emailID
```

```
        if __ name__  == "__ main__ ":
                USERNAME, USER_EMAIL_ID, USER_EMAIL_PASS,
USER_SQL_PASS = credAccess()
                wishMe(USERNAME)
                initTime = datetime.datetime.now()
                today = str(datetime.date.today())
                WAKE = assname = "computer"
                SERVICE = authenticate_google()
                bat = psutil.sensors_battery()
                if 60 <= bat[0] <= 100:
                        print(colored(f"Outstanding performance: {bat[0]} % battery
remaining", "green"))
                elif 30 <= bat[0] < 60:
                        print(colored(f"Good performance: {bat[0]} % battery
remaining", "yellow"))
                else:
                        print(colored(f"Plug-in required: {bat[0]} % battery
remaining"), "red")

                spec = platform.uname()
                print("System = ", spec[0])
                print("Host Name = ", spec[1])
                print("Release(Windows) = ", spec[2])
                print("PC's Version = ", spec[3])
                print("Machine = ", spec[4])
                print("PC's Processor= ", spec[5])
                print("Starting...")
                speak("Initialized")
                while True:
                        text = take_command()

                        if text.count(WAKE) > 0:
                                text = text.replace(WAKE, "")
                                CALENDAR_STRS = ["what do i have", "do i any have
plans"]
                                for phrase in CALENDAR_STRS:
                                        if phrase in text:
                                                date = get_date(text)
                                                if date:
                                                        get_events(date, SERVICE)
                                                else:
                                                        speak("I don't understand")
```

```
                    NOTE_STRS = ["make a note", "write this down",
"remember this"]
                    for phrase in NOTE_STRS:
                        if phrase in text:
                            speak("What would you like me to write
down?")
                            note_text = take_command()
                            note(note_text)

                    if "wikipedia" in text:
                        try:
                            speak('Searching Wikipedia...')
                            text = text.replace("wikipedia", "")
                            results = wikipedia.summary(text,
sentences=2)
                            speak("According to Wikipedia")
                            print(results)
                            speak(results)
                        except Exception as e:
                            print(e)

                    elif 'open youtube' in text:
                        speak("Here you go to Youtube")
                        webbrowser.open("https://www.youtube.com")

                    elif 'search' in text:
                        speak("Searching")
                        search = text.replace("search", "")
                        link =
f"https://www.google.com.tr/search?q={search}"
                        webbrowser.open(link)
                    elif 'open stackoverflow' in text:
                        speak("Here you go to Stack Over flow. Happy
coding!")
                        url = "https://www.stackoverflow.com"
                        webbrowser.open(url)

                    elif 'play' in text:
                        music = text.replace("play", "")
                        link =
f"https://music.youtube.com/search?q={music}"
                        webbrowser.open(link)
```

```
                elif 'time' in text:
                        strTime =
datetime.datetime.now().strftime("%H:%M:%S")
                        print(strTime)
                        speak(f"the time is {strTime}")

                elif 'narrate' in text:
                        speak('What shall I narrate?')
                        string = input("What shall I narrate : ")
                        speak(string)

                elif 'email' in text or 'send email' in text:
                        try:
                                speak("What should I say?")
                                content = take_command()
                                speak("whom should i send")
                                to = contactBook(USER_SQL_PASS)
                                sendEmail(USER_EMAIL_ID,
                                USER_EMAIL_PASS,to, content)
                                speak("Email has been sent !")
                        except Exception as e:
                                print(e)
                                speak("Sorry, I am not able to send this
email")


                elif 'how are you' in text: #3
                        speak("I am fine. Thanks for asking")
                        speak(f"How are you?")
                        text = take_command()
                        if 'fine' in text or "good" in text:
                                speak("It's good to know that your fine")
                elif "change name" in text:
                        speak("What would you like to call me")
                        WAKE = take_command()
                        speak("Thanks for giving me special name")
                elif 'exit' in text or 'good bye' in text: endTime =
                        datetime.datetime.now() print(f"Time
                        duration of Usage : {endTime -
initTime}")
                        speak("Thanks for giving me your time")
                        activity(today, initTime, endTime)
                        exit()
```

```
                elif 'joke' in text:
                        joke = pyjokes.get_joke()
                        print(joke)
                        speak(joke)

                elif "why you came to world" in text:#4
                        speak("To help you out, further thanks to team
JKS")


                elif "who are you" in text:#1
                        speak("I am iCompanion your personal assistant")

                elif 'change background' in text:
                        adr =  "C:\\Windows\\Web\\Wallpaper\\Theme1"
                          #Mind the identation mistake here
        ctypes.windll.user32.SystemParametersInfoW(20,0,adr,0)
                        speak("Background changed successfully")

                elif 'news' in text:
                        try:
                                jsonObj =
urlopen('''http://newsapi.org/v2/top-

headlines?country=in&apiKey=b34c76c69a4048dfa815774ae73ce139''')
                                data = json.load(jsonObj)
                                speak('So, here I have few latest news for
you across the world')
                                i = 1
                                for item in data['articles']:
                                        if i <= 5:
                                                print(str(i) + '. ' + item['title'] +
'\n')

                                                speak(item['title'] + '\n')
                                                i += 1
                                except Exception as e:
                                print(str(e))
                                speak('Here are some headlines from the
Times of India,Happy reading')
                                url =
"https://timesofindia.indiatimes.com/home/headlines"
                                webbrowser.open(url)
                                time.sleep(6)
```

```
                elif 'lock window' in text: speak("locking the
                        device")
                        ctypes.windll.user32.LockWorkStation()
                elif 'shutdown' in text:
                        endTime = datetime.datetime.now()
                        print(f"Time duration of Usage : {endTime -
initTime}")
                        speak("Hold On a Sec ! Your system is on its way
to shut down")
                        os.system("shutdown -s")
                        activity(today, initTime, endTime)

                elif 'empty recycle bin' in text:
                        winshell.recycle_bin().empty(confirm=False,
show_progress=False, sound=True)
                        speak("Recycle Bin Recycled")
                elif "don't listen" in text or "stop listening" in text:
                        speak("for how much time you want me to stop
listening commands")
                        a = int(take_command())
                        time.sleep(a)

                elif 'make a timer' in text or 'timer' in text:
                        def countdown(t):
                                while t > 0:
                                        print(t)
                                        t -= 1
                                        time.sleep(1)

                        speak("For how much time should I set the
timer?")
                        seconds = int(input("For how much time should I
set the timer: "))
                        countdown(seconds)
                        print(colored("Time's Up"),"red")
                        speak("Time's Up")
```

```
                elif "where is" in text:
                    text = text.replace("where is", "")
                    location = text
                    speak("User asked to Locate")
                    speak(location)
                    url =
"https://www.google.nl/maps/place/{location}
                    webbrowser.open(url)
                elif "restart" in text:
                    endTime = datetime.datetime.now()
                    print(f"Time duration of Usage : {endTime -
initTime}")
                    os.system("shutdown -r")
                    activity(today, initTime, endTime)

                elif "hibernate" in text or "sleep" in text:
                    speak("Hibernating")
                    os.system("shutdown -h")

                elif "log off" in text or "sign out" in text:
                    speak("Make sure all the application are closed
before sign-out")
                    time.sleep(5)
                    os.system("shutdown -l")
                    activity(today, initTime, endTime)
```

```
                elif "weather" in text:
                        api_key = "6c7e7e30ff6df9bc6b22fb28c227ff24"
                        base_url =
"https://api.openweathermap.org/data/2.5/weather?"
                        speak("what is the city name")
                        city_name = take_command()
                        complete_url = base_url + "appid=" + api_key +
"&q=" + city_name
                        response = requests.get(complete_url)
                        x = response.json()
                        if x["cod"] != "200":
                                data = response.json()
                                main = data['main']
                                temp = main['temp']
                                temperature = round((temp - 273.15), 2)
                                humidity = main['humidity']
                                pressure = main['pressure']
                                report = data['weather']
                                print(f"{city_name:-^30}")
                                print(f"{temperature}°C in {city_name}")
                                speak(f"{temperature}°C in {city_name}")
                                print(f"{humidity}%humidity")
                                speak(f"{humidity}% humidity")
                                print(f"Pressure is {pressure} hPa")
                                speak(f"Pressure is {pressure} hPa")
                                print(f"Report says that there is
{report[0]['description']}")
                                speak(f"Report says that there is
{report[0]['description']}")
                        else:
                                speak("City Not Found")
                elif "reboot" in text:
                        endTime = datetime.datetime.now()
                        print(f"Time duration of Usage : {endTime -
initTime}")
                        i = 3
                        while i >= 1:
                                speak("Rebooting in")
                                speak(i)
                                i -= 1
                        speak("Rebooting now")
                        activity(today, initTime, endTime)
                        virtualAssistant()
```

```
                elif "history" in text:
                        osCommandString = f"notepad.exe Command
history.txt"
                        os.system(osCommandString)
                        speak("Would you like me to clear your command
history ?")
                        confirmation = take_command()
                        if confirmation.lower().startswith("y"):
                                open("Command history.txt", 'w').close()

                elif 'exit' in text or 'good bye' in text: endTime =
                        datetime.datetime.now() print(f"Time
                        duration of Usage : {endTime -
initTime}")
                        speak("Thanks for giving me your time")
                        activity(today, initTime, endTime)
                        exit()

                elif 'clear screen' in text:
                        os.system('cls')

                elif "camera" in text or "take a photo" in text:
                        ec.capture(0, f"{assname} Camera ", "img.jpg")

                history(text)


#CALLING THE WHOLE PROGRAM RIGHT FROM HERE
if __ name__  == "__ main__ ":
        try:
                virtualAssistant()
        except Exception:
                print(Exception)
```

What Can Our Program Do ???

As stated earlier, this program can be used while surfing, relaxing, doing some productivity and almost every possible task which includes the computer. So it has a wide variety of functionality too. Few of them are:

1. Instantly make notes
2. Fetch wikipedia summary
3. Open Youtube, StackOverFlow and various other websites directly
4. Play any song directly from internet
5. Search anything directly in Internet
6. Tell daily News Headlines
7. Weather Forecasting
8. Crack Jokes
9. Display Events of a Particular Date
10. Display current time from computer clock
11. Narrate any sentence/ Pronounce any difficult word
12. Send e-mails to your contacts which can be directly accessed from their names
13. Interaction with User
14. Change it's Name and Voice (on a gender basis)
15. Record your screen activity time and displays a graph of it for better clarity
16. Change computer background
17. Shutdown/Restart/Hibernate/Sleep your computer. It can even Empty your Recycle Bin!
18. Create a timer for the user
19. Show Geographic location of any place
20. Access your webcam and take your photograph
21. And many more features can be easily added……..

OUTPUT

A separate CD would be enclosed herewith the report file where the whole recording of the output would be there. Still here are screenshots of the execution of the program executed from Command Prompt.

```
By which mode would you like to give commands :
        1 - Voice Mode
        2 - Script Mode
1
←[33mGood performance: 52 % battery remaining←[0m
System =  Windows
Host Name =  SACHIN-SHETH
Release(Windows) =  10
PC's Version =  10.0.19041
Machine =  AMD64
PC's Processor=  Intel64 Family 6 Model 142 Stepping 12, GenuineIntel
```

```
Listening...
Recognizing...
User said: computer tell me about weather

Listening...
Recognizing...
User said: Gandhinagar

---------gandhinagar----------
27.93°C in gandhinagar
34% humidity
Pressure is 1011 hPa
Report says that there is broken clouds
```

```
Listening...
Recognizing...
User said: computer time

18:22:44
```

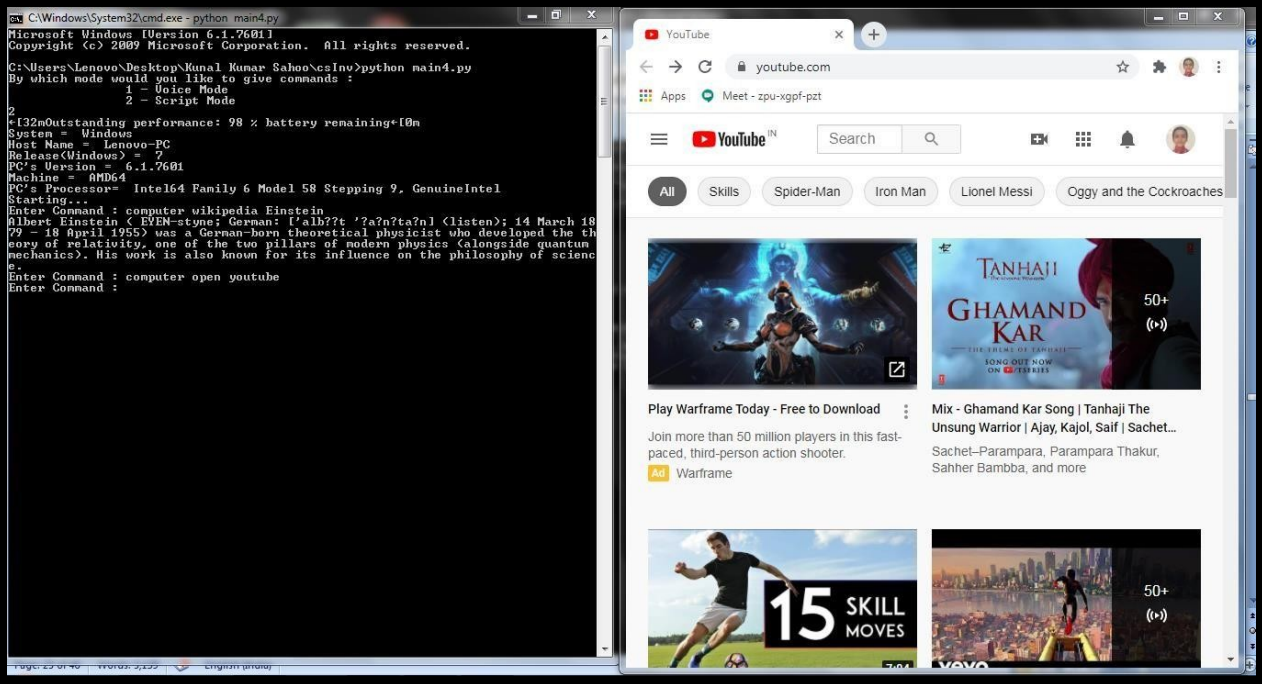
```
Listening...
Recognizing...
User said: computer open stackoverflow
```

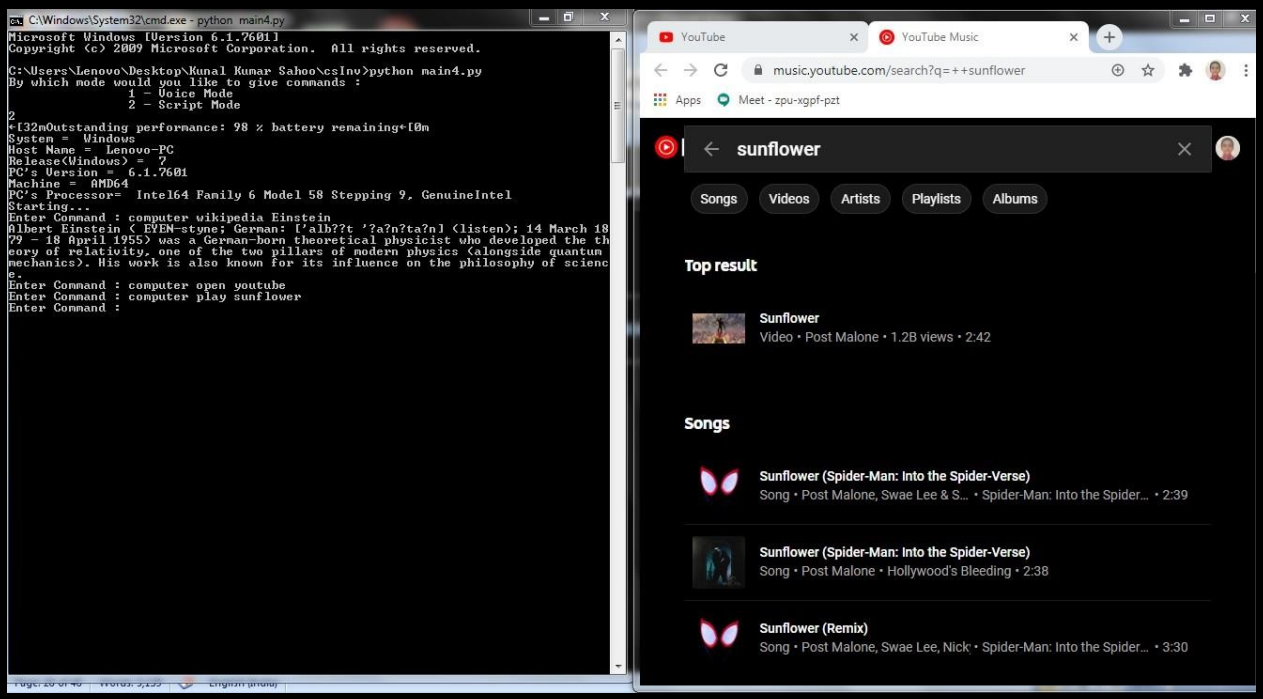
```
Listening...
Recognizing...
User said: computer how are you

Listening...
Recognizing...
User said: I am fine
```

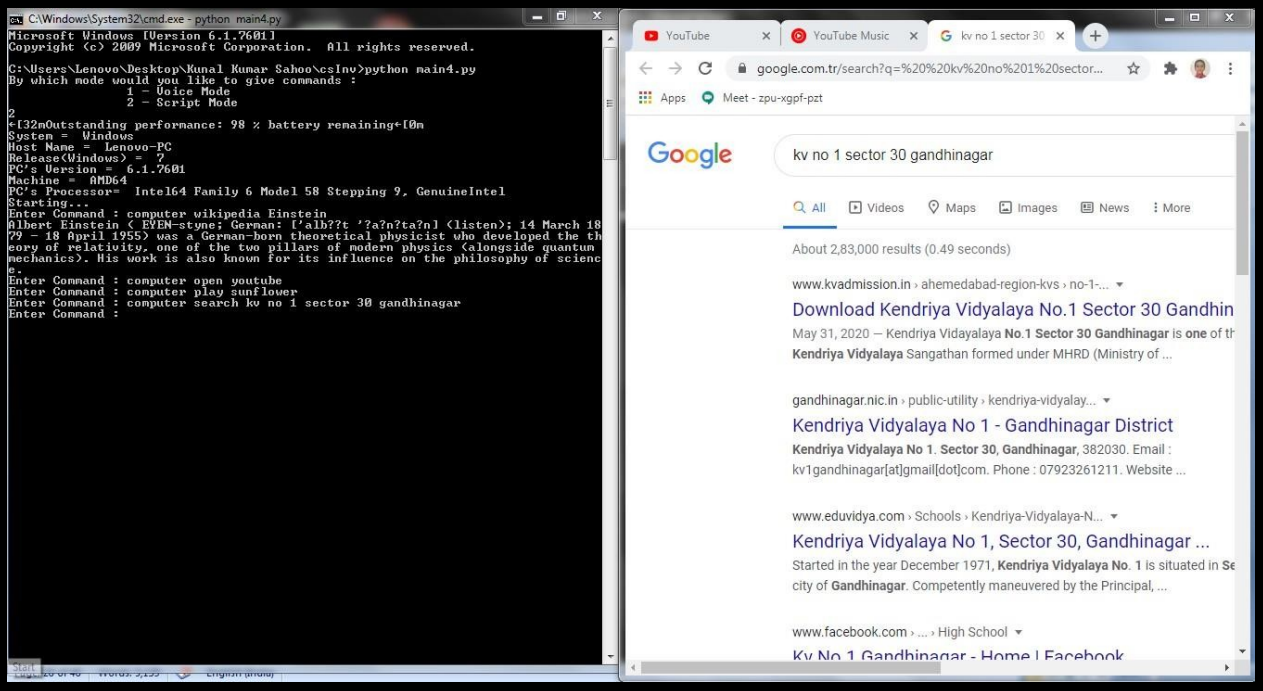
```
Listening...
Recognizing...
User said: computer tell me news

1. Germany makes preparations for coronavirus vaccinations - Mint

2. Vivo V20 Moonlight Sonata Colour Option Launched in India: Price, Offers - Gadgets 360

3. Apple iPhone 12 mini handled on video - GSMArena.com news - GSMArena.com

4. Some recovered Covid patients can still carry virus - The Tribune India

5. "Groomed Many Karyakartas Including Me": PM's Tribute To Keshubhai Patel - NDTV
```

```
Listening...
Recognizing...
User said: computer recita joke

How come there is no obfuscated Perl contest? Because everyone would win.
```

```
Enter Command : computer wikipedia Einstein
Albert Einstein ( EYEN-styne; German: ['alb??t '?a?n?ta?n] (listen); 14 March 18
79 - 18 April 1955) was a German-born theoretical physicist who developed the th
eory of relativity, one of the two pillars of modern physics (alongside quantum
mechanics). His work is also known for its influence on the philosophy of scienc
e.
Enter Command :
```








```
Enter Command : computer what do I have on Saturday ?
2020-10-31T08:00:00+05:30 XII- Chemistry 8 am-9 am
Enter Command : computer what do I have on 18th ?
2020-11-18T08:00:00+05:30 XII- Chemistry 8 am-9 am
```

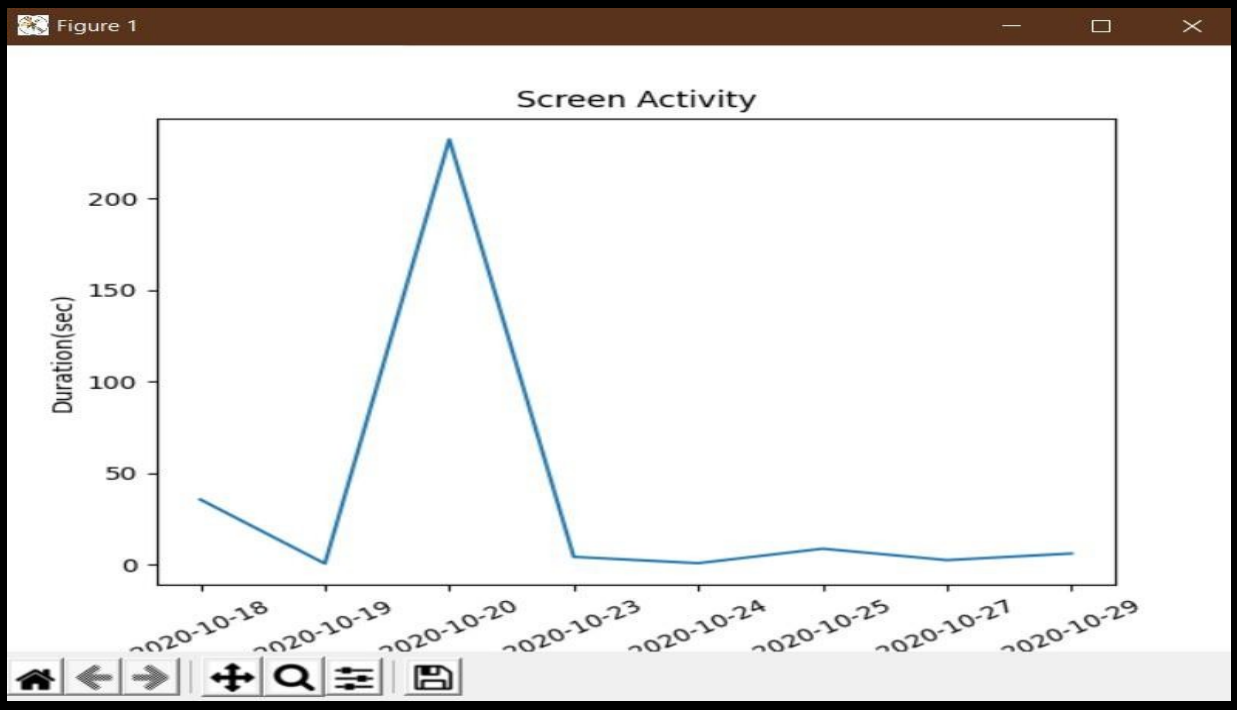
```
Enter Command : computer send an email
Enter Command : this is a test email
Enter the name : jash sheth
Enter Command : computer send an email
Enter Command : hello lil bro! happy bday champ!!! may u have a bright career
Enter the name : harsheel
Record not found
Would you like to add contact ? yes
Enter the e-mail ID of the new contact : harsheel.sahoo2008@gmail.com
1 record inserted.
```

```
C:\Users\Lenovo\Desktop\Kunal Kumar Sahoo\csInv>python main4.py
By which mode would you like to give commands :
                1 - Voice Mode
                2 - Script Mode
2
←[32mOutstanding performance: 98 % battery remaining←[0m
System =  Windows
Host Name =  Lenovo-PC
Release(Windows) =  7
PC's Version =  6.1.7601
Machine =  AMD64
PC's Processor=  Intel64 Family 6 Model 58 Stepping 9, GenuineIntel
Starting...
Enter Command : computer send an email
Enter Command : this is a test email
Enter the name : jash sheth
Enter Command : computer send an email
Enter Command : hello lil bro! happy bday champ!!! may u have a bright career
Enter the name : harsheel
Record not found
Would you like to add contact ? yes
Enter the e-mail ID of the new contact : harsheel.sahoo2008@gmail.com
1 record inserted.
Enter Command : computer history
```

File Edit Format View Help

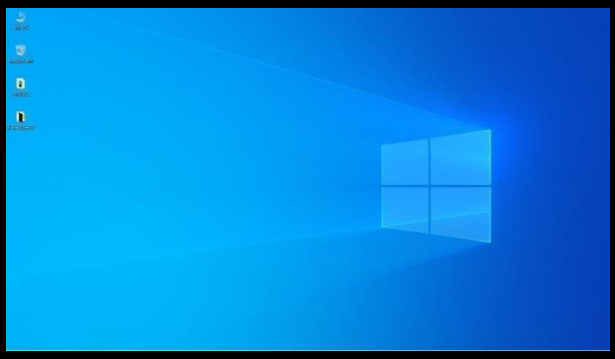
```
20-10-2020, 15:13:09 :  email
20-10-2020, 15:13:56 :  email
20-10-2020, 15:14:33 :  email
20-10-2020, 15:24:41 :  send email
20-10-2020, 15:26:51 :  send email
20-10-2020, 15:26:51 :          server.login('assist.jks@gmail.com', 'yourpasswordi
20-10-2020, 15:26:51 :          server.sendmail('assist.jks@gmail.com', to, content
20-10-2020, 15:26:51 :                          mydb1 = mysqlconnection(host="locall
20-10-2020, 15:26:51 :          wake = assname = ""
20-10-2020, 16:17:48 :  email
20-10-2020, 16:20:04 :  send email
21-10-2020, 14:28:32 :  send an email
21-10-2020, 14:29:39 :   einstein
21-10-2020, 14:29:53 :  recite me a joke
21-10-2020, 14:31:11 :  news
21-10-2020, 14:32:41 :  weather
21-10-2020, 14:38:01 :  news
21-10-2020, 14:38:17 :  dont listen
21-10-2020, 14:38:26 :  do not listen
21-10-2020, 14:40:46 :  don't listen
21-10-2020, 14:46:11 :  news
21-10-2020, 14:50:34 :  joke
21-10-2020, 14:50:50 :   kunal
21-10-2020, 14:51:37 :  news
21-10-2020, 16:15:28 :  clear screen
21-10-2020, 16:15:45 :  what do i have on friday ?
22-10-2020, 13:20:09 :  joke
23-10-2020, 13:31:16 :  narrate
24-10-2020, 16:41:44 :  send an email
25-10-2020, 14:02:56 : send email
30-10-2020, 16:57:58 :   einstein
30-10-2020, 16:59:26 :  open youtube
30-10-2020, 17:01:07 :  play sunflower
30-10-2020, 17:02:28 :  search kv no 1 sector 30 gandhinagar
30-10-2020, 17:07:09 :  what do i have on monday ?
30-10-2020, 17:07:19 :  what  do i have on friday
30-10-2020, 17:07:38 :  what do i have on friday
30-10-2020, 17:07:56 :  what do i have on tuesday
30-10-2020, 17:08:06 :  what do i have on wednesday
30-10-2020, 17:08:17 :  what do i have on 18th
30-10-2020, 17:09:49 :  what do i have 21st july ?
30-10-2020, 17:10:16 :  what do i have on saturday ?
30-10-2020, 17:11:10 :  what do i have on 18th november ?
30-10-2020, 17:12:38 :  what do i have on saturday ?
30-10-2020, 17:13:05 :  what do i have on 18th ?
30-10-2020, 17:13:17 :  am i free on 21st july 2021 ?
30-10-2020, 17:16:44 :  send an email
30-10-2020, 17:17:57 :  send an email
```

```
Listening...
Recognizing...
User said: computer exit

Time duration of Usage : 0:06:06.022130
{'2020-10-18': 35.56, '2020-10-19': 0.59, '2020-10-20': 232.21, '2020-10-23': 4.28, '2020-10-24': 0.82, '2020-10-25': 8.73, '2020-10-27': 2.5, '2020-10-29': 6.1003688333333335}
```



```
Enter Command : comp change background
```

Before

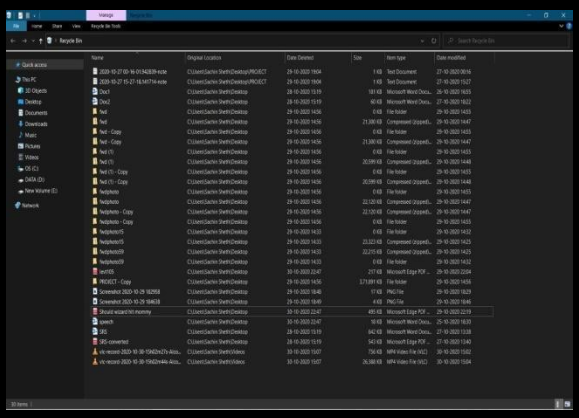


After

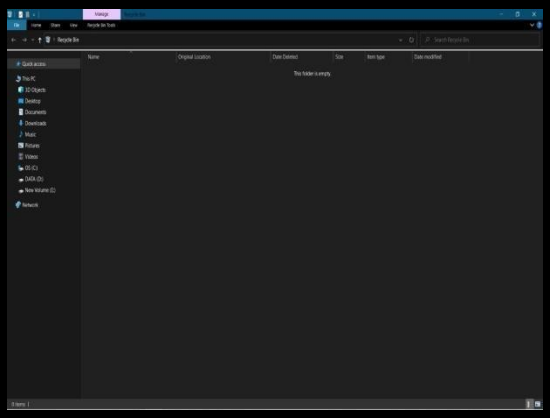
```
Enter Command : computer change name
Enter Command : comp
Enter Command : comp time
20:13:38
```

```
Enter Command : computer empty recycle bin
```

Before

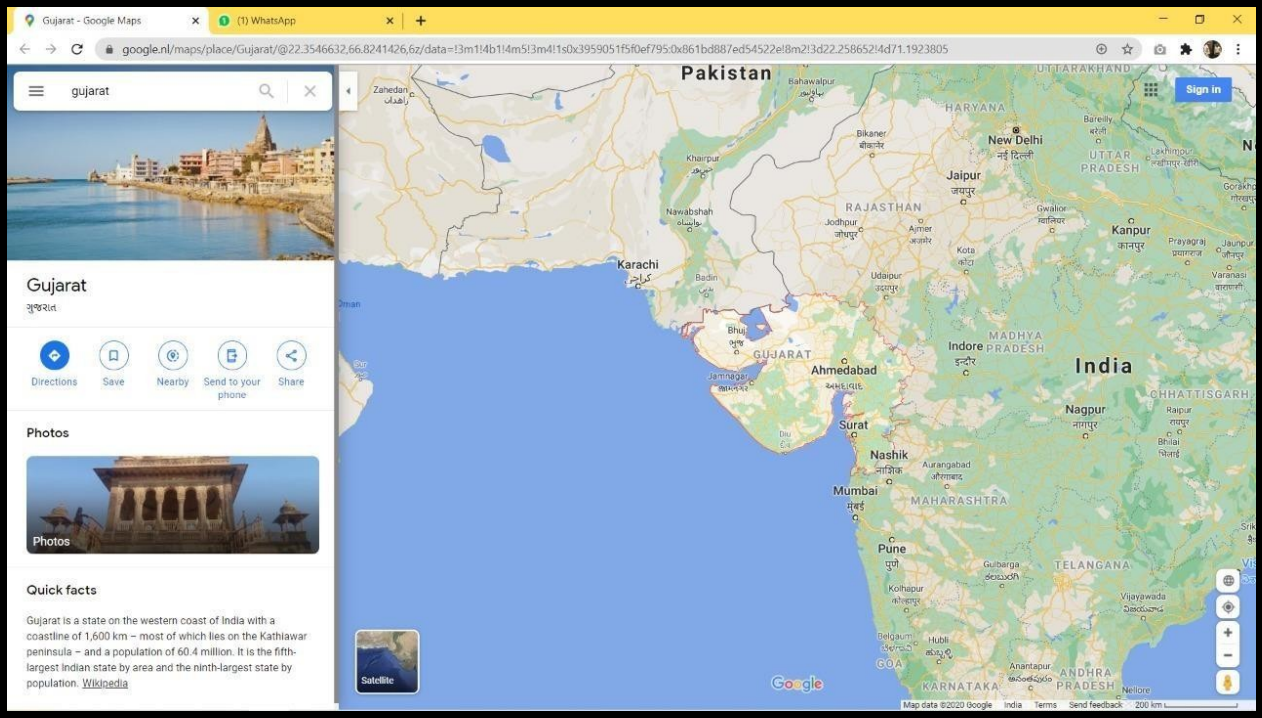


After



```
Enter Command : computer make a stopwatch
For how much time should I set the timer: 24
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
Time's Up
```

```
Starting...
Enter Command : computer where is gujarat
```

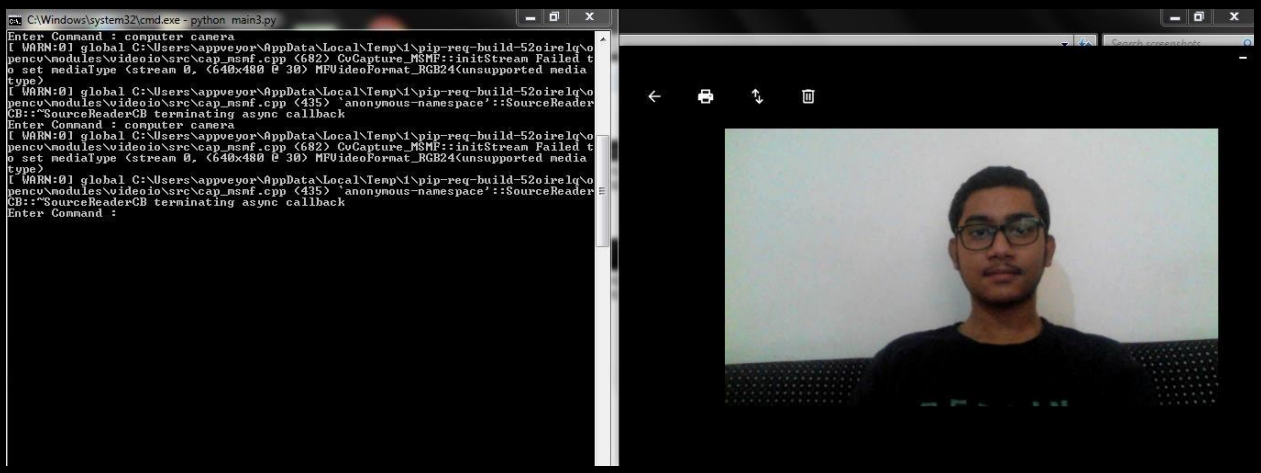


Before

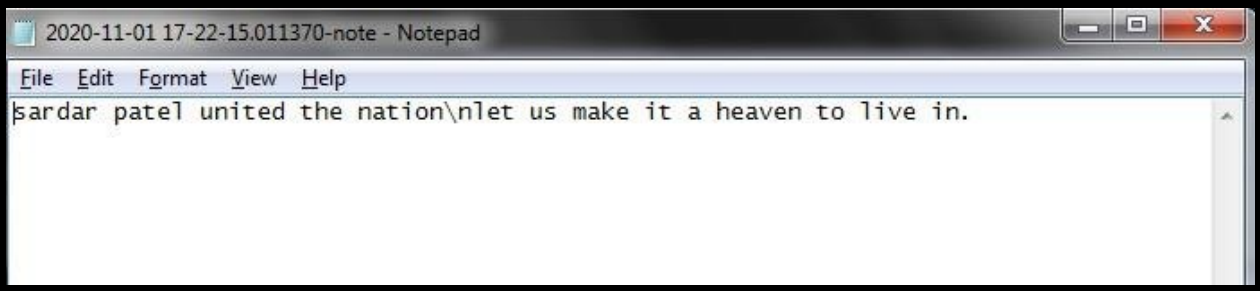
```
C:\Windows\system32\cmd.exe - python  main4.py

C:\Users\Lenovo\Desktop\Kunal Kumar Sahoo\csInv>python main4.py
By which mode would you like to give commands :
                1 - Voice Mode
                2 - Script Mode
2
←[32mOutstanding performance: 99 % battery remaining←[0m
System =  Windows
Host Name =  Lenovo-PC
Release(Windows) =  7
PC's Version =  6.1.7601
Machine =  AMD64
PC's Processor=  Intel64 Family 6 Model 58 Stepping 9, GenuineIntel
Starting...
Enter Command : computer news
1. IPL: Dhoni confirms he will play for Chennai Super Kings in 2021 - Times of I
ndia

2. Nitish Kumar By His Side, PM Points To Huge Crowd At Rallies Amid Covid - NDT
V

3. I visited Dennistoun, and I have no idea why the Scottish neighborhood was vo
ted one of the coolest in the wor - Business Insider India

4. Ravi Shastri reveals mystery behind Rohit Sharma's absence from Indian teams
for Australia tour - India Today

5. Six IPL teams in race for three play-off spots: Who has to do what to qualify
 - The Indian Express

Enter Command : computer weather
Enter Command : Gandhinagar
---------gandhinagar----------
31.93°C in gandhinagar
21% humidity
Pressure is 1011 hPa
Report says that there is few clouds
Enter Command : computer clear screen
```

After

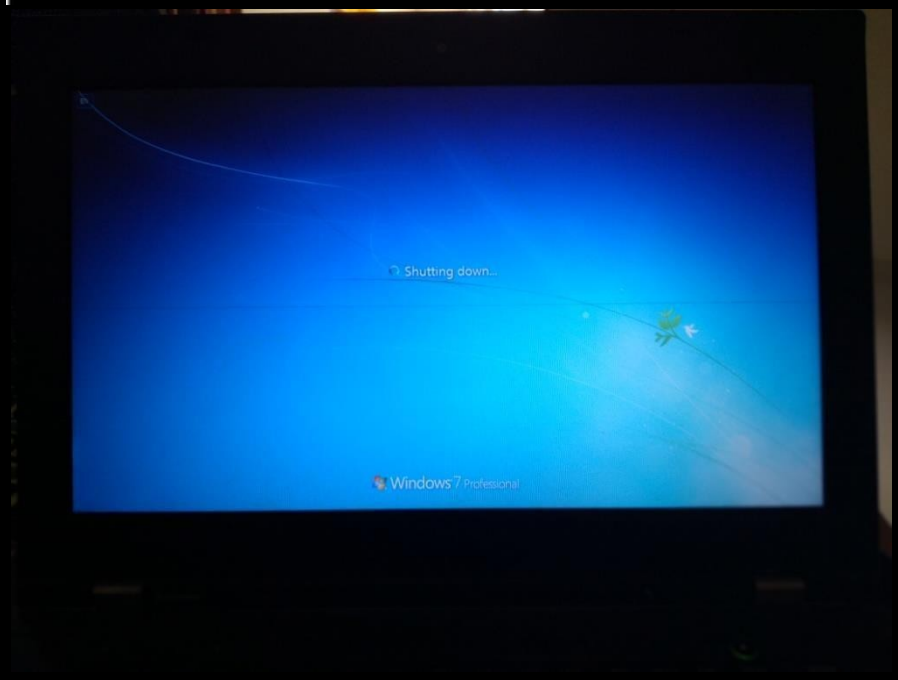
```
C:\Windows\system32\cmd.exe - python  main4.py
Enter Command :
```




```
Enter Command : computer make a note
Enter Command : Sardar Patel united the nation\nLet us make it a Heaven to live
in.
```




```
Enter Command : computer shutdown
```



How Does Our Program Speak ?

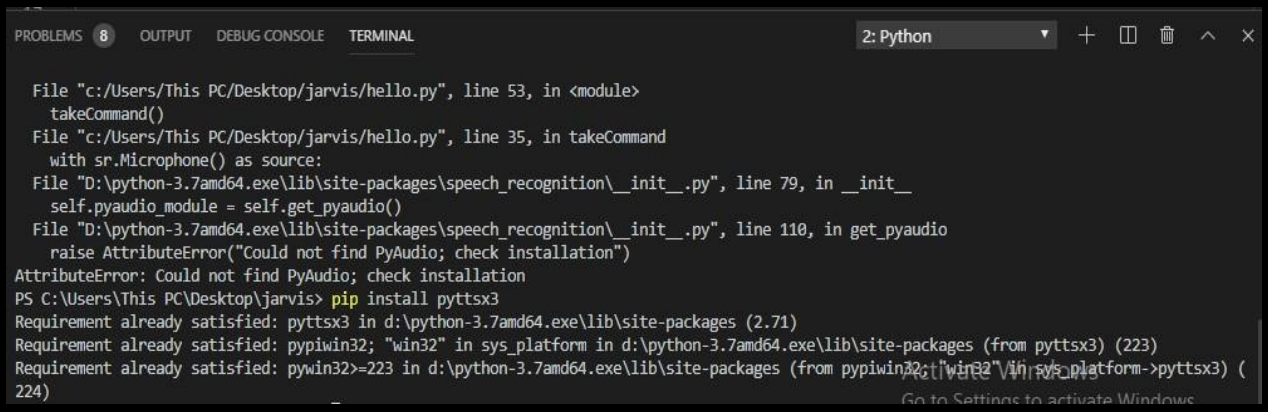
Pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. it is a very easy to use tool which converts the entered text into speech.
The pyttsx3 module supports two voices first is female and the second is male which is provided by “sapi5” for windows.

It supports three TTS (Text To Speech) engines :

- **sapi5** – SAPI5 on Windows
- **nsss** – NSSpeechSynthesizer on Mac OS X
- **espeak** – eSpeak on every other platform

Installation
To install the pyttsx3 module, first of all, you have to open the terminal and write

```
pip install pyttsx3
```



Code : Python program to convert text to speech

```
def speak(text):
        engine = pyttsx3.init("sapi5") # object creation

        """ SETTING UP RATE OF SPEAKING"""
        rate = engine.getProperty('rate')   # getting details of current speaking rate
        engine.setProperty('rate', 125)     # setting up new voice rate

        """SETTING UP VOLUME OF VOICE"""
        volume = engine.getProperty('volume')   #getting to know current volume level (min=0 and max=1)
        engine.setProperty('volume',1.0)    # setting up volume level  between 0 and 1

          """SETTING UP GENDER OF VOICE"""
        voices = engine.getProperty('voices')       #getting details of current voice
        #engine.setProperty('voice', voices[0].id)  #changing index, changes voices. 0 for male
        engine.setProperty('voice', voices[0].id)   #changing index, changes voices. 1 for female

        '''MAKING THE FUNCTION SPEAK A GIVEN STRING'''
        engine.say(text)
        engine.runAndWait()
```

How Does Our Program Listen and Understand ?

Speech Recognition is an important feature in several applications used such as home automation, artificial intelligence, etc.

In python, we have a specific library known as "speechRecognition" which takes voice input from a microphone and returns an equivalent string as an output.

Installation

To install the speechRecognition module, first of all, you have to open the terminal and write

```
pip install Speechrecognition
```

```
(har) C:\Users\ABC>pip install Speechrecognition
Collecting Speechrecognition
  Using cached SpeechRecognition-3.8.1-py2.py3-none-any.whl (32.8 MB)
Installing collected packages: Speechrecognition
Successfully installed Speechrecognition-3.8.1
```

Code:

```
def take_command(recognitionMode):
#Take microphone input from the user and returns string output
if recognitionMode == 1:
        r = sr.Recognizer()
        with sr.Microphone() as source:
                print("Listening...")
                r.pause_threshold = 1
                audio = r.listen(source)
        try:
                print("Recognizing...")
                query = r.recognize_google(audio, language='en-in')
                print(f"User said: {query}\n")
        except Exception as e:
                # print(e)
                print("Say that again please...")
                return "None"
```

```
    #Take direct string input from user
    elif recognitionMode == 2:
        try:
            query = input('Enter Command : ')
        except Exception:
            print(e)
    return query.lower()
```

How Does Program Authenticate Our Credentials for Calendar Events ?

For this purpose we use a free service by Google Developer Console which provides developers free APIs (Application Program Interface) which basically defines interactions between multiple software intermediaries.

Specifically we use a web-based API which verifies the user's GoogleAccount and then fetches information of user's Google Calendar in "READ MODE" in HTML format and using simple HTML parsing using beautifulSoup4 (a basic Python module for formatting HTML data) displays the events for a particular date that is instructed by the user.

First of all we simply download a desktop API in the name of "credentials.json" which is an OAuth Client ID for initiation and verification of the access. Then we download a few Python libraries using pip for the program to work.

Installation:

```
pip install --upgrade google-api-python-client google-auth-httplib2 google-auth-oauthlib
```

```
(har) C:\Users\ABC>pip install --upgrade google-api-python-client google-auth-httplib2 google-auth-oauthlib
Collecting google-api-python-client
  Downloading google_api_python_client-1.12.5-py2.py3-none-any.whl (61 kB)
     |████████████████████████████████| 61 kB 652 kB/s
Collecting google-auth-httplib2
  Downloading google_auth_httplib2-0.0.4-py2.py3-none-any.whl (9.1 kB)
Collecting google-auth-oauthlib
  Downloading google_auth_oauthlib-0.4.1-py2.py3-none-any.whl (18 kB)
Collecting httplib2<1dev,>=0.15.0
  Downloading httplib2-0.18.1-py3-none-any.whl (95 kB)
     |████████████████████████████████| 95 kB 600 kB/s
Collecting uritemplate<4dev,>=3.0.0
  Downloading uritemplate-3.0.1-py2.py3-none-any.whl (15 kB)
Requirement already satisfied, skipping upgrade: six<2dev,>=1.13.0 in c:\users\abc\har\lib\site-packages (from google-api-python-client) (1.15.0)
Collecting google-auth>=1.16.0
  Downloading google_auth-1.22.1-py2.py3-none-any.whl (114 kB)
     |████████████████████████████████| 114 kB 1.6 MB/s
Collecting google-api-core<2dev,>=1.21.0
  Downloading google_api_core-1.23.0-py2.py3-none-any.whl (91 kB)
     |████████████████████████████████| 91 kB 970 kB/s
Collecting requests-oauthlib>=0.7.0
  Downloading requests_oauthlib-1.3.0-py2.py3-none-any.whl (23 kB)
Collecting cachetools<5.0,>=2.0.0
  Downloading cachetools-4.1.1-py3-none-any.whl (10 kB)
Requirement already satisfied, skipping upgrade: setuptools>=40.3.0 in c:\users\abc\har\lib\site-packages (from google-auth>=1.16.0->google-api-python-client) (49.6.0)
Collecting pyasn1-modules>=0.2.1
  Downloading pyasn1_modules-0.2.8-py2.py3-none-any.whl (155 kB)
     |████████████████████████████████| 155 kB 3.3 MB/s
Collecting rsa<5,>=3.1.4; python_version >= "3.5"
  Downloading rsa-4.6-py3-none-any.whl (47 kB)
     |████████████████████████████████| 47 kB 368 kB/s
Requirement already satisfied, skipping upgrade: requests<3.0.0dev,>=2.18.0 in c:\users\abc\har\lib\site-packages (from google-api-core<2dev,>=1.21.0->google-api-python
-client) (2.24.0)
Collecting googleapis-common-protos<2.0dev,>=1.6.0
  Downloading googleapis_common_protos-1.52.0-py2.py3-none-any.whl (100 kB)
     |████████████████████████████████| 100 kB 1.1 MB/s
Requirement already satisfied, skipping upgrade: pytz in c:\users\abc\har\lib\site-packages (from google-api-core<2dev,>=1.21.0->google-api-python-client) (2020.1)
Requirement already satisfied, skipping upgrade: protobuf>=3.12.0 in c:\users\abc\har\lib\site-packages (from google-api-core<2dev,>=1.21.0->google-api-python-client) (
3.13.0)
Collecting oauthlib>=3.0.0
  Downloading oauthlib-3.1.0-py2.py3-none-any.whl (147 kB)
     |████████████████████████████████| 147 kB 6.8 MB/s
Collecting pyasn1<0.5.0,>=0.4.6
```

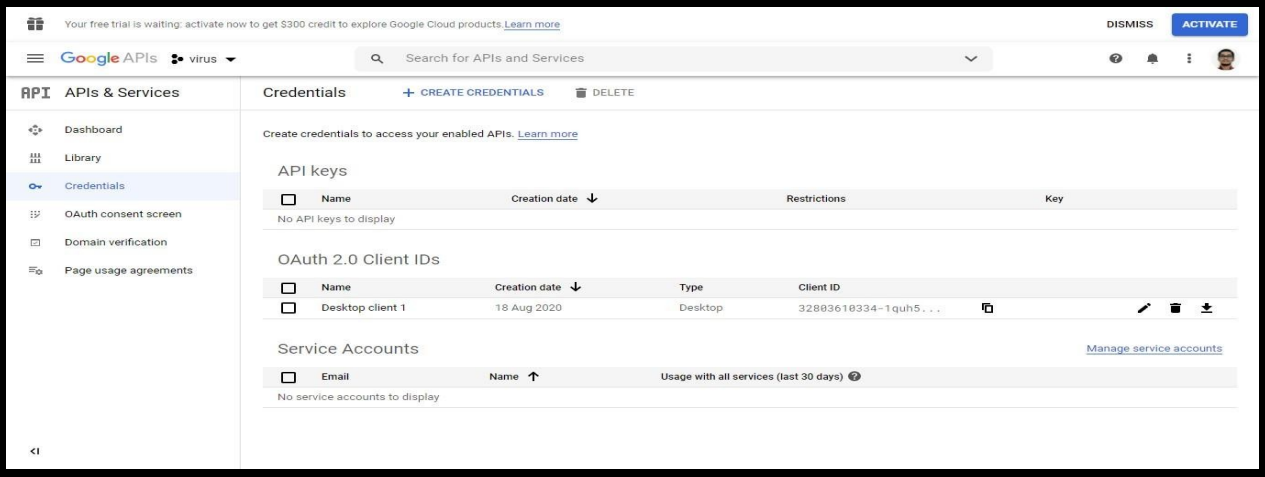


fig:- Getting OAuth Client ID in a JSON format

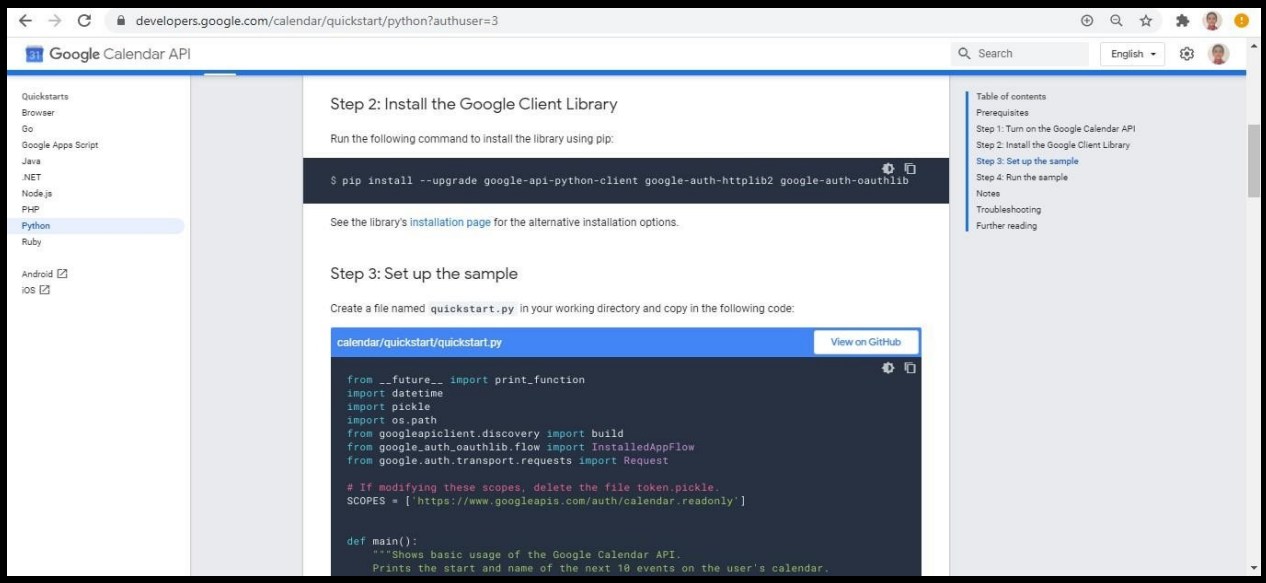


fig:- Getting Source Code for setting up the Calendar feature

Code :

```
        #DEFINING FUNCTION TO AUTHENTICATE GOOGLE ACCOUNT FOR
CALENDAR ACCESSIBILITY STORE THE DATA IN A PICKLE FILE
        def authenticate_google():
                """Shows basic usage of the Google Calendar API.
                Prints the start and name of the next 5 events on the user's calendar.
                """
                creds = None
                if os.path.exists('token.pickle'):
                        with open('token.pickle', 'rb') as token:
                                creds = pickle.load(token)

                if not creds or not creds.valid:
                        if creds and creds.expired and creds.refresh_token:
                                creds.refresh(Request())
                        else:
                                flow = InstalledAppFlow.from_client_secrets_file(
                                        'credentials.json', SCOPES)
                                creds = flow.run_local_server(port=0)

                        with open('token.pickle', 'wb') as token:
                                pickle.dump(creds, token)

                service = build('calendar', 'v3', credentials=creds)

                return service

        #DEFINING FUNCTION TO  GET CALENDAR EVENTS OF USER ON A
GIVEN DAY
        def get_events(day, service):
                # Call the Calendar API
                date = datetime.datetime.combine(day, datetime.datetime.min.time())
                end_date = datetime.datetime.combine(day,
datetime.datetime.max.time())
                utc = pytz.UTC
                date = date.astimezone(utc)
                end_date = end_date.astimezone(utc)

                events_result = service.events().list(calendarId='primary',
timeMin=date.isoformat(), timeMax=end_date.isoformat(),

        singleEvents=True,
```

```
        orderBy='startTime').execute()
            events = events_result.get('items', [])

            if not events:
                speak('No upcoming events found.')
            else:
                speak(f"You have {len(events)} events on this day.")


                for event in events:
                    start = event['start'].get('dateTime',
event['start'].get('date'))
                    print(start, event['summary'])
                    #Accessing time of event as an event and parse it
                    try:
                        start_time = str(start.split("T")[1].split("-")[0])
                        if int(start_time.split(":")[0]) < 12:
                            start_time = start_time + "am"
                        else:
                            start_time = str(int(start_time.split(":")[0]) -
12) + start_time.split(":")[1]
                            start_time = start_time + "pm"

                        speak(event["summary"] + " at " + start_time)

                    except Exception as e:
                        speak(event["summary"])
```

How Does Our Program Get Date?

For this task we have developed a specific function which takes input of the user's command with the help of take_command() function (which accepts a voice command and returns a string). Then using fundamental concepts of string manipulation, list slicing, if-elif-else nesting we extract a date or day and relatively compare it with the computer's local calendar and clock system and return a specific date for the calendar to fetch events.

Code:

```
def get_date(text):
    text = text.lower()
    today = datetime.date.today()
    if text.count("today") > 0:
        return today
    day = -1
    day_of_week = -1
    month = -1
    year = today.year
    for word in text.split():
        if word in MONTHS:
            month = MONTHS.index(word) + 1
        elif word in DAYS:
            day_of_week = DAYS.index(word)
        elif word.isdigit():
            day = int(word)
        else:
            for ext in DAY_EXTENTIONS:
                found = word.find(ext)
                if found > 0:
                    try:
                        day = int(word[:found])
                    except:
                        pass
    if month < today.month and month != -1:  # if the month mentionedis before the current month set the year to the next
        year = year+1
    if month == -1 and day != -1:  # if we didn't find a month, but wehave a day
        if day < today.day:
            month = today.month + 1
        else:
            month = today.month
```

```
        # if we only found a data of the week
        if month == -1 and day == -1 and day_of_week != -1:
                current_day_of_week = today.weekday()
                dif = day_of_week - current_day_of_week
                if dif < 0:
                        dif += 7
                        if text.count("next") >= 1:
                                dif += 7
                return today + datetime.timedelta(dif)
        if day != -1:
                return datetime.date(month=month, day=day, year=year)
```

How Does Our Program Send an EMail ?

Our program uses e-mail with the help of a trivial networking concept which is known as “Simple Mail Transfer Protocol” i.e. SMTP. For this we have used a standard Python library known as “smtplib”.

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.

Code:

```
def sendEmail(USER_EMAIL_ID, USER_EMAIL_PASS, to, content):

        server = smtplib.SMTP('smtp.gmail.com', 587)

        server.ehlo()

        server.starttls()

        # Enable low security in gmail

        server.login(USER_EMAIL_ID, USER_EMAIL_PASS)

        server.sendmail(USER_EMAIL_ID, to, content)

        server.close()
```

For this the parameters such as USER_EMAIL_ID and USER_EMAIL_PASS are accessed via a text file “userCred.txt”. Also, the parameter “to” is accessed with the help of MySQL database in which a table of contact names and email-ids associated with them is saved and “content” parameter is taken by user’s input with the help of take_command() function.

How Does Our Program Fetch the EMail ID of the Contact ?

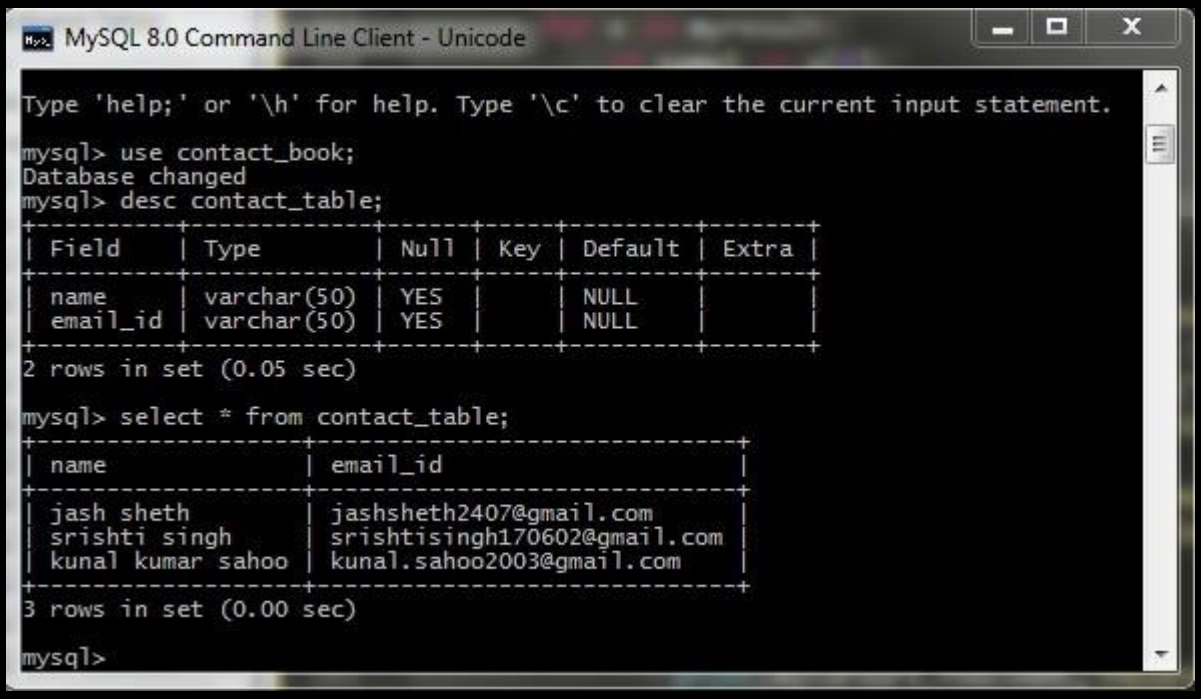
This task is done with the help of MySQL-Python connectivity. Under root privileges we have created a database “contact_book” in which a relation is created under the name of “contact_table” which has a 2nd Degree relation. The two attributes are : “name” and “email_id” which are having data type VARCHAR(50) for convenience.

SQL Queries:

mysql> CREATE DATABASE contact_book; USE contact book;

mysql> CREATE TABLE contact_table(name VARCHAR(50), email_id VARCHAR(50));

After creation of the table, contacts can be simply added manually in the SQL prompt itself. Another method is to call the sendEmail() function by which contact name can be searched in the database and if the contact is not found then itcould be added right there itself.



Installation:

```
pip install mysql-connector-python
```

Code:

```
        def contactBook(USER_SQL_PASS):
                #Trying to access e-mail id of a given contact
                mydb = MySQLConnection(host="localhost", user="root",
password=USER_SQL_PASS, database="contact_book")
                mycursor = mydb.cursor()
                name1 = str(input("Enter the name : "))
                mycursor.execute("SELECT * FROM contact_table")
                myresult = mycursor.fetchall()
                for x in myresult:
                        if name1 == x[0]:
                                emailID = x[1]
                                break
                #If the contact is not present in the database
                else:
                        print("Record not found")
                        insertChoice = input("Would you like to add contact ? ")
                        #Add contact name and e-mail id in the database
                        if insertChoice.lower().startswith("y"):
                                mydb1 = MySQLConnection(host="localhost",
user="root", password="computer", database="contact_book")
                                mycursor1 = mydb1.cursor()
```

```
                        mail = str(input("Enter the e-mail ID of the new contact :
"))
                        sql = "INSERT INTO contact_table(name, email_id)
VALUES(%s,%s)"
                        val = (name1, mail)
                        mycursor1.execute(sql, val)
                        mydb1.commit()
                        print(mycursor1.rowcount,"record inserted.")

                #After the database is updated with the new contact, send e-
mail to the recipient
                mydb = MySQLConnection(host="localhost", user="root",
password=USER_SQL_PASS, database="contact_book")
                mycursor = mydb.cursor()
                mycursor.execute("SELECT * FROM contact_table")
                myresult = mycursor.fetchall()
                for x in myresult:
                        if name1 == x[0]:
                                emailID = x[1]
                                break
        return emailID
```

How Does Our Program Record Our Screen Activity Time?

For this purpose, we first of all have created a CSV File to store date of usage and total time the program has been activated and then the data is fetched in a dictionary and then differentiates into two lists where the "i" index element of the first list is the date and that of the second list is the time duration. These two lists are taken into account and data of last 10 days is displayed with the help of a Line Graph whenever the program is terminated.

The date is taken from the clock of the computer and the time interval is calculated by simply subtracting the computer time when the program is instructed to terminate by the time when the program was executed.

Code:

```
def activity(today, initTime, endTime):

    deltaTime = endTime - initTime
    duration = deltaTime.total_seconds() /60
    importedData = []

    with open('screenActivity.csv', 'r') as file:
        reader = csv.reader(file)
        for row in reader:
            importedData.append(row)
    dict = {}
    for i in range(len(importedData)):
        if len(importedData) != 0:
            dict[importedData[i][0]] =
round(float(importedData[i][1]), 2)
    for i in dict:
        if i == today:
            print(dict[i])
            dict[i] = duration + dict[i]
            break
    else:
        dict[today] = duration
    print(dict)
    with open('screenActivity.csv', 'w') as file:
        for key in dict.keys():
            file.write("%s,%s\n" % (key, dict[key]))
    x = []
    y = []
    with open('screenActivity.csv', 'r') as csvfile:
        plots = csv.reader(csvfile, delimiter=',')
        for row in plots:
            x.append((row[0]))
            y.append(float(row[1]))
    plt.plot(x, y)
    plt.xticks(rotation=30)
    plt.xlabel('Date')
    plt.ylabel('Duration(sec)')
    plt.title('Screen Activity')
    plt.show()
```

How Does Our Program Record All The Instructions Given By The User ?

This is also done with the help of simple concept of file handling i.e. file appending. After every command is executed, the whole string is appended into a text file which can be later accessed via a Text Editor or right through the program itself and the records can also be deleted for user's privacy or memory management by simply opening the file in write mode by which the file gets overwritten bya new file.

Code:

```
def history(text):
        date_time = datetime.datetime.now().strftime(" %d-%m-%Y, %H:%M:%S")
        f = open("Command history.txt", "a")
        f.write(f"{date_time} : {text}\n")
```

How Does Our Program Greet The User ?

Greeting is simply done with the help of system clock. With the help of "datetime" module the current time of clock when the program is executed is recieved. Then with the help of if-elif-else conditional statements, the greet statement is returned and then concatenating the string with the user name which is accessed through userCred.txt, the program speaks the whole new string which is a simple greet statement.

Code:

```
        def wishMe(name):
                hour = int(datetime.datetime.now().hour)
                if hour >= 0 and hour < 12:
                        speak("Good Morning!" + str(name))
                elif hour >= 12 and hour < 16:
                        speak("Good Afternoon!" + str(name))
                else:
                        speak("Good Evening!" + str(name))
```

Fetching Summary from Wikipedia

Wikipedia is a free-open encyclopaedia which consists of information of almost everything we know today. Mostly it contains all the information to the public but mostly in general only the summary part of it matters the most because it briefly describes the whole topic in a very simple manner.

Wikipedia is also a Python library which can be installed via pip and is used to access data of wikipedia with the help of APIs in HTML format and then parse this data and format it into a presentable form using BeautifulSoup4.

Installation:

```
pip install wikipedia
```

```
(har) C:\Users\ABC>pip install wikipedia
Processing c:\users\abc\appdata\local\pip\cache\wheels\07\93\05\72c05349177dca2e0ba31a33ba4f7907606f7ddef303517c6a\wikipedia-1.4.0-py3-none-any.whl
Requirement already satisfied: requests<3.0.0,>=2.0.0 in c:\users\abc\har\lib\site-packages (from wikipedia) (2.24.0)
Requirement already satisfied: beautifulsoup4 in c:\users\abc\har\lib\site-packages (from wikipedia) (4.9.1)
Requirement already satisfied: idna<3,>=2.5 in c:\users\abc\har\lib\site-packages (from requests<3.0.0,>=2.0.0->wikipedia) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\abc\har\lib\site-packages (from requests<3.0.0,>=2.0.0->wikipedia) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\users\abc\har\lib\site-packages (from requests<3.0.0,>=2.0.0->wikipedia) (1.25.10)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\abc\har\lib\site-packages (from requests<3.0.0,>=2.0.0->wikipedia) (2020.6.20)
Requirement already satisfied: soupsieve>1.2 in c:\users\abc\har\lib\site-packages (from beautifulsoup4->wikipedia) (2.0.1)
Installing collected packages: wikipedia
Successfully installed wikipedia-1.4.0
```

Code:

```
if "wikipedia" in text:
        try:
                speak('Searching Wikipedia...')
                text = text.replace("wikipedia", "")
                results = wikipedia.summary(text, sentences=2)
                speak("According to Wikipedia")
                print(results)
                speak(results)

        except Exception as e:
                print(e)
```

Web-Browsing using Python

The webbrowser module in python provides a high level interface to allow displaying Web-based document to users. In our program we have used its most simple open() function to just directly search any string directly in system's default web-browser which is mostly Microsoft Edge/Internet Explorer in caseof Windows, Safari in case of MacOS and Mozilla Firefox in case of Linux distros.

Code:

```
elif 'search' in text:
        speak("Searching")
        search = text.replace("search", "")
        link = f"https://www.google.com.tr/search?q={search}"
        webbrowser.open(link)
```

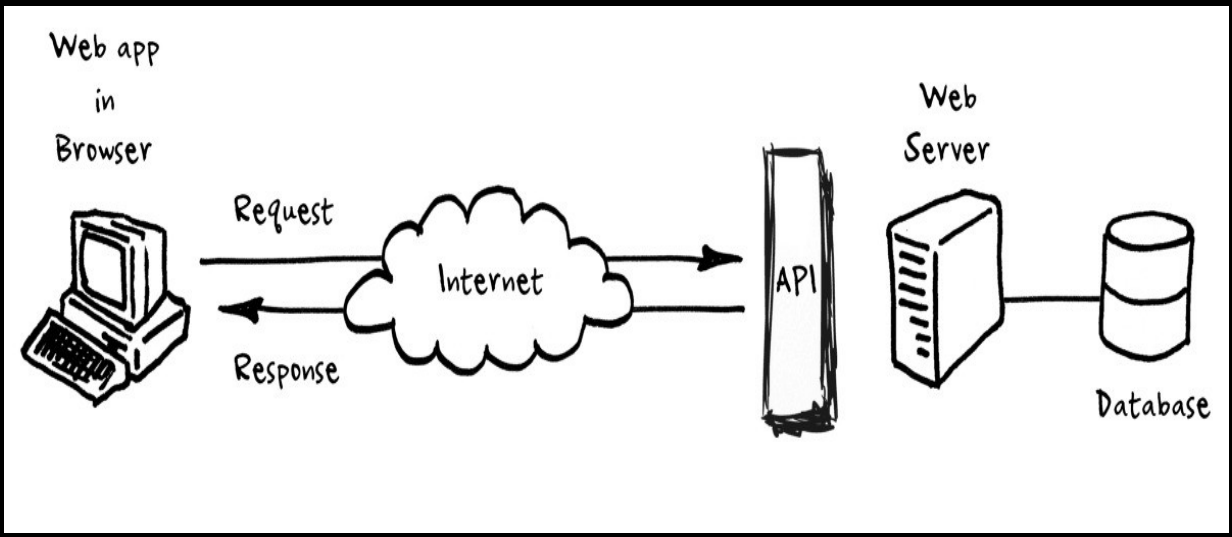
Getting News Headlines using Web Scraping and Python

For fetching news headlines, we once again used another free web service i.e. **NewsApi**. NewsAPI is a simple HTTP REST API for searching and retrieving live articles from all over web using a simple API key.

Code:

```
elif 'news' in text:
        try:
                jsonObj = urlopen('''http://newsapi.org/v2/top-
headlines?country=in&apiKey=b34c76c69a4048dfa815774ae73ce139''')
                data = json.load(jsonObj)
                speak('So, here I have few latest news for you across the world')
                i = 1
                for item in data['articles']:
                        if i <= 5:
                                print(str(i) + '. ' + item['title'] + '\n')
                                speak(item['title'] + '\n')
                                i += 1
        except Exception as e:
                print(str(e))
                speak('Here are some headlines from the Times of India,Happy
reading')
                url =  "https://timesofindia.indiatimes.com/home/headlines"
                webbrowser.open(url)
                time.sleep(6)
```

How An API Works?



Getting Weather Forecast using Web Scraping and Python

Again for getting weather forecast of any city across the globe we took in use a free web-based service called Openweather Map which provide web-based API keys to get weather forecast of any city right from their database which they collect with the help of National Meteorological Reports and satellite image processing.

Code:

```
elif "weather" in text:
        api_key = "6c7e7e30ff6df9bc6b22fb28c227ff24"
        base_url = "https://api.openweathermap.org/data/2.5/weather?"
        speak("what is the city name")
        city_name = take_command()
        complete_url = base_url + "appid=" + api_key + "&q=" + city_name
        response = requests.get(complete_url)
        x = response.json()
        if x["cod"] != "200":
                data = response.json()
                main = data['main']
                temp = main['temp']
                temperature = round((temp - 273.15), 2)
                humidity = main['humidity']
                pressure = main['pressure']
                report = data['weather']
                print(f"{city_name:-^30}")
                print(f"{temperature}°C in {city_name}")
                speak(f"{temperature}°C in {city_name}")
                print(f"{humidity}% humidity")
                speak(f"{humidity}% humidity")
                print(f"Pressure is {pressure} hPa")
                speak(f"Pressure is {pressure} hPa")
                print(f"Report says that there is {report[0]['description']}")
                speak(f"Report says that there is {report[0]['description']}")
        else:
                speak("City Not Found")
```

BIBLIOGRAPHY

1. https://www.python.org/downloads/windows/
2. https://stackoverflow.com/questions/60307452/pyttsx3-initialization-error-cant-use-pyttsx3
3. https://www.geeksforgeeks.org/speech-recognition-in-python-using-google-speech-api/
4. https://www.w3schools.com/python/python_mysql_getstarted.asp
5. www.pypi.org
6. https://openweathermap.org/
7. https://newsapi.org/docs
8. https://developers.google.com/calendar/quickstart/python?authuser=3
9. https://dev.mysql.com/downloads/installer/
10. https://stackoverflow.com/questions/50243506/caching-sha2-password-is-not-supported-mysql?rq=1
11. Automate The Boring Stuff With Python by AI Sweigart
12. Computer Science with Python by Preeti Arora
13. www.mypythonkvs.wordpress