```
import pyttsx3  # pip install pyttsx3
import speech_recognition as sr  # pip install
speechRecognition
import datetime
import wikipedia  # pip install wikipedia
import webbrowser
import os
import smtplib
import pyjokes
import pywhatkit
import json
import spotipy
import requests


engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
# print(voices[1].id)
engine.setProperty('voice', voices[0].id)


def speak(audio):
    engine.say(audio)
    engine.runAndWait()


def time():
    Time = datetime.datetime.now().strftime("%I:%M:%S %p")
    speak (Time)


def wishMe():

    speak('welcome back sir')
    speak('the current time is')
    time()
    hour = int(datetime.datetime.now().hour)
    if hour >= 0 and hour < 12:
        speak("Good Morning!")

    elif hour >= 12 and hour < 18:
        speak("Good Afternoon!")

    elif hour >= 18 and hour < 24:
        speak("Good Evening!")
    else:
        speak("Good night")

    speak("I am Jarvis Sir. Please tell me how may I help you")




def takeCommand():
    # It takes microphone input from the user and returns
string output

    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 0.7
        audio = r.listen(source,phrase_time_limit= 5)

    try:
        print("Recognizing...")
        query = r.recognize_google(audio, language='en-in')
        print(query)

    except Exception as e:
        # print(e)
        print("Say that again please...")
        speak("Say that again please")

        return "None"
    return query





def sendEmail(to, content):

    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('www.yukeshwaran2003@gmail.com',
'yukeshwaran3')
    server.sendmail('www.yukeshwaran2003@gmail.com', to, 
content)






if __name__ == "__main__":
    wishMe()
    while True:
    
        query = takeCommand().lower()

        # Logic for executing tasks based on query
        if 'wikipedia' in query:
            speak('Searching Wikipedia...')
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=2)
            speak("According to Wikipedia")
            print(results)
            speak(results)


        elif 'search on google' in query:
            speak('What do you want to search on Google, sir?')
            query = takeCommand().lower()
            search_on_google(query)



        elif 'open youtube' in query:
            webbrowser.open("https://www.youtube.com/")


        elif 'open google' in query:
            webbrowser.open("https://www.google.com/")

        elif 'open insta' in query:
            webbrowser.open("https://www.instagram.com/")

        elif 'open whatsapp' in query:
            webbrowser.open("https://web.whatsapp.com/")
        elif 'open spotify' in query:
            webbrowser.open("https://open.spotify.com/track/
7fZBQnc0zXwVybgCIrQQil?si=14af68d3a287467b")

        elif 'open showflix' in query:
            webbrowser.open('https://showflix.in/')

        elif 'open facebook' in query:
            webbrowser.open('https://www.facebook.com/')

        elif 'open stackoverflow' in query:
            webbrowser.open("https://stackoverflow.com/")








        elif 'calculator' in query:

            from chatterbot import ChatBot

# naming the ChatBot calculator
# using mathematical evaluation logic
# the calculator AI will not learn with the user input
            Bot = ChatBot(name = 'Calculator',
                            read_only = True,
                            logic_adapters =
["chatterbot.logic.MathematicalEvaluation"],
                            storage_adapter =
"chatterbot.storage.SQLStorageAdapter")


# clear the screen and start the calculator
            print('\033c')
            print("Hello, I am a calculator. How may I help
you?")
            while (True):
    # take the input from the user
                user_input = input("me: ")

    # check if the user has typed quit to exit the prgram
                if user_input.lower() == 'quit':
                    print("Exiting")
                    break

    # otherwise, evaluate the user input
    # print invalid input if the AI is unable to comprehend the
input
                try:
                    response = Bot.get_response(user_input)
                    print("Calculator:", response)
                except:
                    print("Calculator: Please enter valid
input.")
```

```
        elif 'music' in query:
            music_dir = 'G:\My songs'
            songs = os.listdir(music_dir)
            print(songs)
            os.startfile(os.path.join(music_dir, songs[1]))

        elif ' favourite ' in query:
            music_dir = 'G:\English songs'
            songs = os.listdir(music_dir)
            print(songs)
            os.startfile(os.path.join(music_dir, songs[1]))

        elif 'time' in query:

            # importing whole module
            from tkinter import *
            from tkinter.ttk import *

            # importing strftime function to
            # retrieve system's time
            from time import strftime

            # creating tkinter window
            root = Tk()
            root.title('Clock')


            # This function is used to
            # display time on the label
            def time():
                string = strftime('%I:%M:%S %p')
                lbl.config(text=string)
                lbl.after(1000, time)


            # Styling the label widget so that clock
            # will look more attractive
            lbl = Label(root, font=('calibri', 100 , ),
                        background='purple',
                        foreground='white')
            speak('ok sir')

            # Placing clock at the centre
            # of the tkinter window
            lbl.pack(anchor='center')
            time()

            mainloop()


        elif 'date' in query:
            def date():
                year = int(datetime.datetime.now() .year)
                month= int(datetime.datetime.now().month)
                date= int(datetime.datetime.now().day)
                speak (date)
                speak (month)
                speak (year)

            date()
        elif 'open code' in query:
            codePath = "C:\\Users\\kishancjx\\AppData\\Local\
\Programs\\Microsoft VS Code\\Code.exe"
            os.startfile(codePath)
            # you should enter your path here

        elif 'joke' in query:
            print(pyjokes.get_joke())
            speak(pyjokes.get_joke())


        elif 'are you single' in query:
            print(' I am in a realtionship with wifi')
            speak(' I am in a realtionship with wifi')


        elif 'with me' in query:
            print('sorry I am busy')
            speak('sorry I am busy')



        elif 'who developed you' in query:
            print('my master yukesh developed me')
            speak('my master yukesh developed me')


        elif 'tell my name' in query:
            print('yukesh')
            speak('yukesh')
        if 'play' in query:
            song = query.replace('play', '')
            speak('playing' + song)
            pywhatkit.playonyt(song)
```

```
        elif 'email to Lokesh' in query:
            try:
                speak("What should I say?")
                content = takeCommand()
                to = "wwww.jacklokesh2001@gmail.com"
                sendEmail(to, content)
                speak("Email has been sent!")
            except Exception as e:
                print(e)
                speak("Sorry my friend yukesh. I am not able to
send this email")
```

```
from spotipy.oauth2 import SpotifyClientCredentials
Reels_uri = 'spotify:artist: 0QrHnTrFHQfM5vzLbVc7HC'
spotify =
spotipy.Spotify(client_credentials_manager=SpotifyClientCredent
ials(client_id='37c3f19f8d9a48fbabc28375bb4285bf',client_secret
='c49050029b234f5ead9d5997e8cfbd07',))
results = spotify.artist_albums (Reels_uri, album_type='album')
albums = results['items']

while results['next']:
    results = spotify.next(results)
    albums.extend(results['items'])

for album in albums:
    print(album['name'])
```

```python
EMAIL = config("EMAIL")
PASSWORD = config("PASSWORD")


def send_email(receiver_address, subject, message):
    try:
        email = EmailMessage()
        email['To'] = receiver_address
        email["Subject"] = subject
        email['From'] = EMAIL
        email.set_content(message)
        s = smtplib.SMTP("smtp.gmail.com", 587)
        s.starttls()
        s.login(EMAIL, PASSWORD)
        s.send_message(email)
        s.close()
        return True
    except Exception as e:
        print(e)
        return False
```