DESKTOP VERTUAL ASSISTANCE

DESKTOP VERTUAL ASSISTANCE

A mini project-1 report submitted in partial fulfillment of requirements for the award of Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

By

CHAKALI MADHU (199X1A0543)

ARADESI PRAMOD (199X1A0535)

KONDAPALLI VENKATA CHANDAN KUMAR (199X1A0560)

Under the Esteemed guidance of

Sri K.V. Rameswara Reddy
Assistant Professor
Department of C.S.E.

Department of Computer Science and Engineering

G. PULLA REDDY ENGINEERING COLLEGE (Autonomous):KURNOOL

(Affiliated to JNTUA, ANANTAPURAMU)

2021- 2022

Department of Computer Science and Engineering

G. PULLA REDDY ENGINEERING COLLEGE (Autonomous):KURNOOL

(Affiliated to JNTUA, ANANTAPURAMU)

CERTIFICATE

This is to certify that the mini project-1 work entitled ‘DESKTOP VERTUAL ASSISTANCE’ *is a bona fide record of work carried out by*

C. MADHU (199X1A0543)

A. PRAMOD (199XA0535)

K.V. CHANDAN KUMAR (199X1A0560)

Under my guidance and supervision in fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Sri K.V. Rameswara Reddy,
Assistant Professor,
Department of C.S.E.,
G. PullaReddy Engineering College,
Kurnool.

Dr. N. Kasi Viswanath,
Professor & Head of the Department,
Department of C.S.E.
G. PullaReddy Engineering College,
Kurnool.

DECLARATION

We hereby declare that the project titled "DESKTOP VERTUAL ASSISTANCE" is the authentic work carried out by us as students of G. PULLA REDDY ENGINEERING COLLEGE (Autonomous) Kurnool, during June – November 2021 – January 2022 and has not been submitted elsewhere for the award of any degree or diploma in part or in full to any institute.

CHAKALI MADHU
(199X1A0543)

ARADESI PRAMOD
(199X1A0535)

KONDAPALLI VENKATA CHANDAN KUMAR
(199X1A0560)

ACKNOWLEDGEMENT

We wish to express our deep sense of gratitude to our project guide, **Sri K.V. Rameswara Reddy,** Assistant Professor of CSE Department, G. Pulla Reddy Engineering College, for his immaculate guidance, constant encouragement and cooperation which have made possible to bring out this project work.

We are grateful to our project incharge, **Smt. K. Asha Rani**, Asst Professor of CSE Department, G. Pulla Reddy Engineering College, for helping us and giving us the required information needed for our project work.

We are thankful to our Head of the Department **Dr. N. Kasi Viswanath**, for his whole hearted support and encouragement during the project sessions.

We are grateful to our respected Principal **Dr. B. Sreenivasa Reddy**, for providing requisite facilities and helping us in providing such a good environment.

We wish to convey our acknowledgements to all the staff members of the Computer Science Engineering Department for giving the required information needed for our project work.

Finally we wish to thank all our friends and well wishers who have helped us directly or indirectly during the course of this project work.

ABSTRACT

The project aims to develop a personal-assistant. Jarvis draws its inspiration from virtual assistants like Cortana for Windows, Siri for iOS and Alexa for Amazon. It has been designed to provide a user-friendly interface for carrying out a variety of tasks by employing certain well-defined commands. Users can interact with the assistant either through voice commands by Microphone or using keyboard input. It perform the all tasks like AI(Artificial Intelligence).

As a personal assistant, Jarvis assists the end-user with day-to-day activities like general human conversation, searching queries in google, youtube, live weather conditions, news headlines, trending movies, sending mails, sending whatsapp message and reminding the user about the scheduled events and tasks. The user statements/commands are analysed with the help of python predefined modulesto give a solution.

CONTENTS

LIST OF FIGURES

SNO FIGURE NAME PAGE.NO

1. INTRODUCTION

1.1INTRODUCTION

Artificial Intelligence when used with machines, it shows us the capability of thinking like humans. In this, a computer system is designed in such a way that typically requires interaction from human. As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the Alexa, Siri, etc. In Python there is an API called Speech Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to send emails without typing any word, searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favourite IDE with the help of a single voice command. In the current scenario, advancement in technologies are such that they can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time. As the voice assistant is using Artificial Intelligence hence the result that it is providing are highly accurate and efficient. The assistant can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. People employed as virtual assistants often have several years of experience as an administrative assistant or office manager. New opportunities are opening up for virtual assistants who are skilled in social media, content management, blog post writing, graphic design, and Internet marketing. As working from home has become more accepted for both workers and

employers, the demand for skilled virtual assistants is expected to grow. The usage of virtual assistants is expanding rapidly after 2017, more and more products are coming into the market. Due to advancement in the technology many different features are being added in the mobile phone and desktops. To use them with more convenient and fun way we require a means of input which is faster and reliable at the same time. In our project we use voice command to input the data into the system for that the microphone is used which convert acoustic energy into electrical energy. After taking the input there is a requirement to understand the audio signal for this google API is used. Different companies like google, apple use different API’s for this purpose. It is truly a feat that today, one can schedule meetings or send email merely through spoken commands.

The assistant is no less than a human assistant but we can say that this is more effective and efficient to perform any task. The libraries and packages used to make this assistant focuses on the time complexities and reduces time.

As a personal assistant, Jarvis assists the end-user with day-to-day activities like general human conversation, searching queries in various search like, searching for videos, search anything on google or youtube, live weather conditions, news headlines, searching for trending movies, sending emails and whatsapp message and reminding the user about the scheduled events and tasks. The user statements/commands are analysed with the help of machine learning to give an optimal solution.

1.2 OBJECTIVE OF THE PROJECT

A virtual assistant is an independent contractor who provides administrative services to clients while operating outside of the client's office. A virtual assistant typically operates from a home office but can access the necessary planning documents, such as shared calendars, remotely. People employed as virtual assistants often have several years of experience as an administrative assistant or office manager. New opportunities are opening up for virtual assistants who are skilled in social media, content management, blog post writing, graphic design, and Internet marketing. As working from home has become more accepted for both workers and employers, the demand for skilled virtual assistants is expected to grow.

KEY TAKEAWAYS

- A virtual assistant is a self-employed worker who specializes in offering administrative services to clients from a remote location, usually a home office.
- Typical tasks a virtual assistant might perform include scheduling appointments, making phone calls, making travel arrangements, and managing email accounts.
- Some virtual assistants specialize in offering graphic design, blog writing, bookkeeping, social media, and marketing services.

- For an employer, one advantage of hiring a virtual assistant is the flexibility to contract for just the services they need.

Theoretically, a virtual assistant can do anything that another support staff might do. There are limitations, but technology is increasingly offering ways to work around those limitations. For example, they may not be able to physically bring you coffee in the morning, but they can place a coffee or lunch order through a food delivery service. Virtual support duties are not limited to clerical work. They may provide assistance with marketing, web design, bookkeeping, and many other services. Some virtual assistants specialize in a specific skill set, and they only perform duties related to those skills. Other virtual assistants specialize in an industry, such as a virtual assistant whose niche is assisting realtors.

We are all well aware about Cortana, Siri, Google Assistant and many other virtual assistants which are designed to aid the tasks of users in Windows, Android and iOS platforms.

This Software aims at developing a personal assistant for Linux-based systems. The main purpose of the software is to perform the tasks of the user at certain commands, provided in either of the ways, speech or text. It will ease most of the work of the user as a complete task can be done on a single command. Jarvis draws its inspiration from Virtual assistants like Cortana for Windows and Siri for iOS. Users can interact with the assistant either through voice commands or keyboard input.

Presently, Jarvis is being developed as an automation tool and virtual assistant. Among the Various roles played by Jarvis are:

1. To search something on Wikipedia
2. Perform the task like
    - To Open Google
    - To Open Youtube
    - To Open StackOverflow
    - To Open Instagram
    - To Open Facebook
3. To play music

4. To know the current time
5. To open the code in VS code
6. To Send the Emails
7. To Open Notepad
8. To Open Command Prompt
9. IP Address
10. To Open Camera
11. To Send Whatsapp message
12. To Know the Trending Movies
13. To Know the News Headlines
14. To Know the Weather Report
15. To Search something on Google
16. To Search something on Youtube
17. To Get the random Joke
18. To Get the random Advice
19. To Restart the Computer
20. To Shutdown the computer
21. To Know the Current Location

1.3 EXISTING SYSTEM

We are familiar with many existing voice assistants like Alexa, Siri, Google Assistant, Cortana which uses concept of language processing, and voice recognition. Cloud-based programs that require internet-connected devices and/or applications to work. Three such applications are Siri on Apple devices, Cortana on Microsoft Devices and Google Assistant on Android devices. There are also devices dedicated to providing virtual assistance. The most popular ones are available from Amazon, Google and Microsoft. To use the Amazon Echo virtual assistant, called Alexa, users call out the wake word, "Alexa." They listens the command given by the user as per their requirements and performs that specific function in a very efficient and effective manner. As these voice assistants are using Artificial Intelligence hence the result that they are providing are highly accurate and efficient. These assistants can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. These assistants are no less than a human assistant but we can say that they are more effective and efficient to perform any task. The algorithm used to make these assistant focuses on the time complexities and reduces time. But for using these assistants one should have an account (like Google account for Google assistant, Microsoft account for Cortana) and can use it with internet connection only because these assistants are going to work with internet connectivity. They are integrated with many devices like, phones, laptops, and speakers etc.

1.4 PROPOSED SYSTEM

It was an interesting task to make my own assistant. It became easier to send emails without typing any word, searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favourite IDE with the help of a single voice command. Jarvis is different from other traditional voice assistants in terms that it is specific to desktop and user does not need to make account to use this, it does not require any internet connection while getting the instructions to perform any specific task. The IDE used in this project is PyCharm. All the python files were created in PyCharm and all the necessary packages were easily installable in this IDE. For this project following modules and libraries were used i.e. pyttsx3, SpeechRecognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, pyQt etc. I have created a live GUI for interacting with the JARVIS as it gives a design and interesting look while having the conversation. With the advancement JARVIS can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time. Functionalities of this project include, It can send emails, It can read PDF, It can send text on WhatsApp, It can open command prompt, your favorite IDE, notepad etc., It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube, etc., in a web browser, It can give weather forecast, It can give desktop reminders of your choice. It can have some basic conversation.

For Future Prospective

We plan to Integrate Jarvis with mobile using react native, to provide a synchronized experience between the two connected devices.

Further, in the long run, Jarvis is planned to feature auto deployment supporting elastic beanstalk, backup files, and all operations which a general Server Administrator does. The functionality would be seamless enough to replace the Server Administrator with Jarvis.

1.5 ORGANIZATION OF THE REPORT

The first chapter deals with the introduction of the virtual desktop assistant, objective of this project, existing systems and proposed systems of the project. The objective of the project explains about main goal of the project. The existing systems explains about presently working systems and proposed system is about what we are coming up with in the future when compared to the present systems. The second chapter deals with system specifications required for developing the project, it includes hardware and software specifications. The third chapter gives preview about design and implementation with source code, output screens, testing and validation. The fourth chapter end with the conclusion of this project.

2. SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

- Processor: intel i3
- RAM: 4GB
- Hard disk: 40 GB

2.2 SOFTWARE SPECIFICATION

- Operating System: Windows
- Programming Language: Python
- Platform: VS Code

3. LITERATURE SURVEY

1.The seeds of modern AI were planted by classical philosophers who attempted to describe the process of human thinking as the mechanical manipulation of symbols.

2.In the 1940s and 50s, a handful of scientists from a variety of fields (mathematics, psychology, engineering, economics and political science) began to discuss the possibility of creating an artificial brain. The field of artificial intelligence research was founded as an academic discipline in 1956.

3.Artificial intelligence is the ability of machines to perform certain tasks, which need the intelligence showcased by humans and animals. This definition is often ascribed to Marvin Minsky and John McCarthy from the 1950s, who were also known as the fathers of the field.

4.Artificial intelligence allows machines to understand and achieve specific goals. AI includes machine learning via deep learning.

5.The former refers to machines automatically learning from existing data without being assisted by human beings.

6.Deep learning allows the machine to absorb huge amounts of unstructured data such as text, images, and audio.

7.In the Modern Era of fast moving technology we can do things which we never thought we could do before but, to achieve and accomplish these thoughts there is a need for a platform which can automate all our tasks with ease and comfort.

8. Thus we need to develop a Personal Assistant having brilliant powers of deduction and the ability to interact with the surroundings just by one of the materialistic form of human interaction i.e. Human Voice.

9. In 1956, American computer scientist John McCarthy organised the Dartmouth Conference, at which the term ‘Artificial Intelligence’ was first adopted. Research centres popped up across the United States to explore the potential of AI.

10. Researchers Allen Newell and Herbert Simon were instrumental in promoting AI as a field of computer science that could transform the world.

4. DESIGN ANALYSIS AND IMPLIMENTATION

4.1 INTRODUCTION

Virtual assistant devices and technology

Virtual assistants are typically cloud-based programs that require internet-connected devices and/or applications to work. Three such applications are Siri on Apple devices, Cortana on Microsoft Devices and Google Assistant on Android devices.

There are also devices dedicated to providing virtual assistance. The most popular ones are available from Amazon, Google and Microsoft. To use the Amazon Echo virtual assistant, called Alexa, users call out the wake word, "Alexa." A light on the device signals to the user it is ready to receive a command, which typically involves simple language requests, such as "what is the weather today," or "play pop music." Those requests are processed and stored in Amazon's cloud.

The technologies that power virtual assistants require massive amounts of data, which feeds artificial intelligence (AI) platforms, including machine learning, natural language processing and speech recognition platforms. As the end user interacts with a virtual assistant, the AI programming uses sophisticated algorithms to learn from data input and become better at predicting the end user's needs.

Virtual assistant capabilities

Virtual assistants typically perform simple jobs for end users, such as adding tasks to a calendar; providing information that would normally be searched in a web browser; or controlling and checking the status of smart home devices, including lights, cameras and thermostats.

Users also task virtual assistants to make and receive phone calls, create text messages, get directions, hear news and weather reports, find hotels or restaurants, check flight reservations, hear music, or play games.

4.2 DESCRIPTION OF KEY PARAMETERS

STARTING VS CODE

Visual Studio Code is a source-code editor made by Microsoft for windows, Linux and macOSby Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax, highlighting, code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional that add additional functionality.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++. It is based on the Electron framework, which is used to develop Node.js Web application that run on the Blink layout engine.

DEFINING SPEAK FUNCTION

The first and foremost thing for A.I assistance is that it should be able to speak. To make our Desktop Assistance talk, we will make a function called speak (). This function will take audio as an argument, and then it will pronounce it.

Now, the next thing we need is audio. We must supply audio so that we can pronounce it using the speak () function we made. We are going to install a module called Pyttsx3.

Pyttsx3, A python library that will help us to convert text to speech. In short, it is a text-to-speech library. It works offline, and it is compatible with Python 2 as well as Python 3.

Installation

```
Pip install pyttsx3
```

SAPI5

Microsoft Speech API (SAPI5) is the technology for voice recognition and synthesis provided by Microsoft. Starting with Windows XP, it ships as part of the Windows OS. If you are using a different OS.

VOICEID

Voice ID helps us to select the different voices, for which voice the system going to speak.

voice[0].id = male voice

voice[1].id = female voice

MAIN FUNCTION

Here we will create our main function, and inside the main function we will add the speak () and all speech tasks done here.

DEFINING WISH ME FUNCTION

Now, we will make a wishme () function that will make our J.A.R.V.I.S. wish or greet the user according to the time of computer or pc. To provide current or live time to A.I., we need to import a module called datetime. Import this module to your program by

Import datetime.

Here, have stored the current hour or time integer into a variable named hour. Now, we will use this hour value inside an if-else loop.

By using utils.py file it wishes when the task perform.

```
opening_text = [
    "Cool, I'm on it sir.",
    "Okay sir, I'm working on it.",
    "Just a second sir.",
]
```

DEFINING TAKE COMMAND FUNCTION

The next most important thing for our A.I. assistant is that it should take command with the help of the microphone of the user's system. So, now we will make a takeCommand() function. With the help of the takeCommand() function, our A.I. assistant will return a string output by taking microphone input from the user.

Before defining the takeCommand() function, we need to install a module called **Speech Recognition** Install this module by:

Import SpeechRecogition as sr

Next, using the recognize_google() method from the *Recognizer* class, we try to recognize the audio. The recognize_google() method performs speech recognition on the audio passed to it, using the **Google Speech Recognition API**.

Speech Recognition incorporates computer science and linguistics to identify spoken words and converts them into text. It allows computers to understand human language.

Speech recognition is a machine's ability to listen to spoken words and identify them. You can make a query or give a reply. You can even program some devices to respond to these spoken words. You can do speech recognition in python with the help of computer programs that take in input from the microphone, process it, and convert it into a suitable form.

The algorithms used in this form of technology include PLP features, Viterbi search, deep neural networks, discrimination training, WFST framework, etc. If you are interested in Google’s new inventions, keep checking their recent publications on speech.

We have successfully created our takeCommand() function. Now we are going to add a try and except block to our program to handle errors effectively.

4.3 THE OVERALL BLOCK DIAGRAM OF VOICE ASSISTANCE

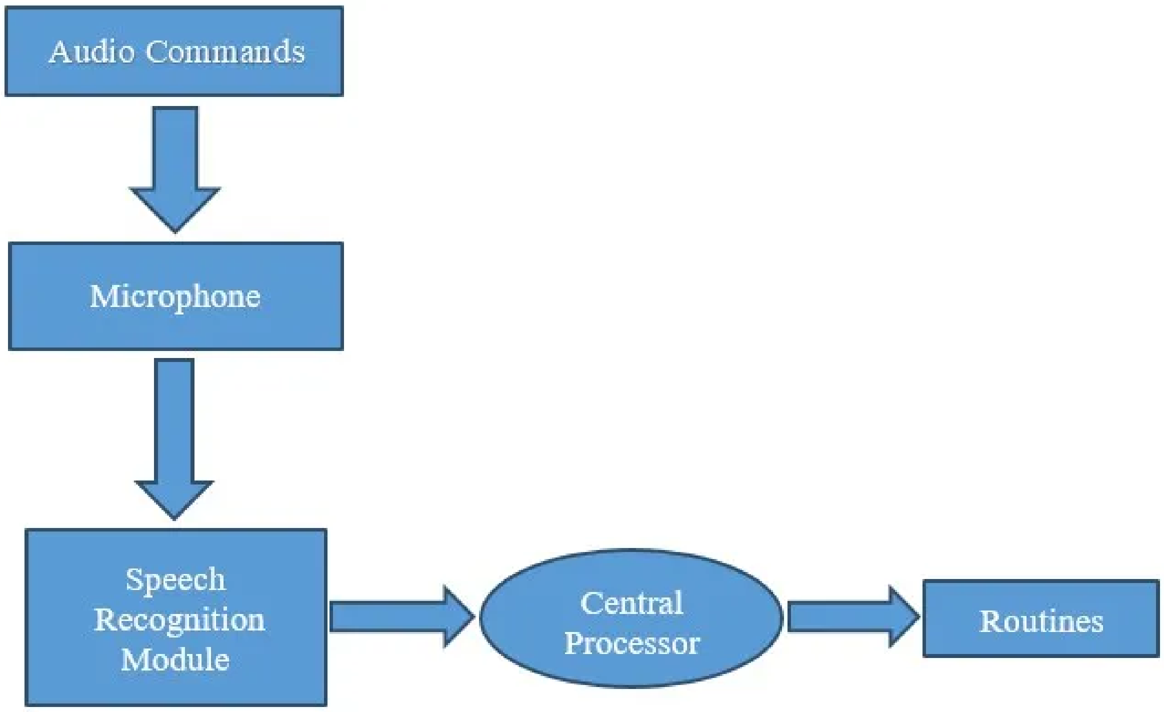


Fig 4.3.1 Block Diagram of voice assistance

4.4 TASKS OF THE DESKTOP VIRTUAL ASSISTANT

Defining Task 1: To search something on Wikipedia

To do Wikipedia searches, we need to install and import the Wikipedia module into our program. Type the below command to install the Wikipedia module.

Pip install Wikipedia

After successfully installing the Wikipedia module, import it into the program by writing an import statement.

we have used an if statement to check whether Wikipedia is in the user's search query or not. If Wikipedia is found in the user's search query, then two sentences from the summary of the Wikipedia page will be converted to speech with the speak function's help.

Defining Task 2: To open the tasks using webbrowser

Import webbrowser

In Python, webbrowser module is **a convenient web browser controller**. It provides a high-level interface that allows displaying Web-based documents to users. webbrowser can also be used as a CLI tool.

To open any website, we need to import a module called **webbrowser**. It is an in-built module, and we do not need to install it with a pip statement; we can directly import it into our program by writing an import statement.

Here, we are using an elif loop to check whether Youtube is in the user's query. Let' suppose the user gives a command as assistance, open youtube. So, open youtube will be in the user's query, and the elif condition will be true.

By applying the same logic we can perform tasks like

A. Open Youtube
B. Open Google
C. Open StackOverflow
D. Open Facebook
E. Open Instagram

Defining Task 3: To play music

To play music, we need to import a module called OS. Import this module directly with an import statement.

It is possible to automatically perform many operating system tasks. The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.

Import OS

we first opened our music directory and then listed all the songs present in the directory with the OS module's help. With the help of **os.startfile**, you can play any song of your choice. I am playing the first song in the directory. However, you can also play a random song with the help of a random module. Every time you command to play music, assistance will play any random song from the song directory.

Defining Task 4: To know the current time

We are using datetime() function and storing the current or live system time into a variable called strTime. After storing the time in strTime, we are passing this variable as an argument function. Now, the time string will be converted into speech.

Import datetime

Python Datetime module comes built into Python, so there is no need to install it externally. Python Datetime module supplies classes to work with date and time. ... datetime. It’s a combination of date and time along with the attributes year, month, day, hour, minute, second, microsecond, and tzinfo.

Defining Task 5: To Open the vs code program

It is also work by using the OS module

To open the VS Code or any other application, we need the code path of the application.

Steps to get the code path of the application:

Step 1: Open the file location.

Step 2: Right-click on the application and click on properties.

Step 3: Copy the target from the target section.

After copying the target of the application, save the target into a variable. Here, I am saving the target into a variable called codePath, and then we are using the OS module to open the application.

Defining Task 6: To send Email

To send an email, we need to import a module called smtplib.

Import 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**. For details of SMTP and ESMTP operation, consult RFC 821 (Simple Mail Transfer Protocol) and RFC 1869 (SMTP Service Extensions).

- Simple Mail Transfer Protocol(SMTP) is a protocol that allows us to send emails route emails between mail servers. An instance method called **sendmail** is present in the SMTP module. This method allows. It takes 3 parameters:
- **The sender:** Email address of the sender.
- **The receiver:** T Email of the receiver.
- **The message:** A string message which needs to be sent to one or more than one recipient.

Defining a send Email function

We will create a **sendEmail()** function, which will help us send emails to one or more than one recipient.

```
def sendEmail(to, content):

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

    server.ehlo()

    server.starttls()

    server.login('youremail@gmail.com', 'your-password')
    server.sendmail('youremail@gmail.com', to, content)

    server.close()
```

Note: Do not forget *to “enable the less secure apps” feature* in your Gmail account. Otherwise, the sendEmail function will not work properly.

Calling the sendEMail() function is from inside the main function

We are using the try and except block to handle any possible error while sending emails.

For implementing further task we need to create some another python files, methods and API’s.

4.4.1 HOW TO SET UP A OFFLINE FUNCTIONS

Inside the functions folder, create a Python file called os_ops.py. In this file, we'll create various functions to interact with the OS.

Import subprocess

The subprocess module **allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes**. This module intends to replace several older modules and functions: OS. system OS.

```
import os
import subprocess as sp

paths = {
    'notepad': "C:\\WINDOWS\\system32\\notepad.exe",
    'calculator': "C:\\Windows\\System32\\calc.exe",
}
```

In the above script, we have created a dictionary called paths which has a software name as the key and its path as the value. You can change the paths according to your system and add more software paths if you need to do so.

Defining Task 7: To Open the Notepad

We'll use these functions to open Notepad++ and by defining the function.

We can call the function from main function by using query format in if – else ladder.

```
elif 'open notepad' in query:
            open_notepad()
```

Defining Task 8: To Open the command prompt

We'll use this function to open the command prompt in our system.

We can call the function from main function by using query format in if – else ladder.

```
 elif 'open command prompt' in query:
            open_cmd()
```

Defining Task 8: To Open the camera

We'll use this function to open the camera in our system. We'll be using the subprocess module to run the command.

We can call the function from main function by using query format in if – else ladder.

```
 elif 'open camera' in query:
            open_camera()
```

4.4.2 HOW TO SET UP ONLINE FUNCTIONS

Let's create a file called online_ops.py within the functions directory, and start creating these functions one after another. For now, add the following code in the file:

Import requests

The requests module **allows you to send HTTP requests using Python**. The HTTP request returns a Response Object with all the response data (content, encoding, status, etc).

The requests module provides the POST method that can directly send the data to the server with the help of the URL and values of the parameters. It is a more likely request method used for submitting any web form or uploading any file.

Import pywhatkit

pywhatkit is **a** Python library for sending WhatsApp messages at a certain time, it has several other features too. Following are some features of pywhatkit module: Send WhatsApp messages. Play a YouTube video. Perform a Google Search.

Using pywhatkit module:

1. Open the downloaded file and search for an application named chrome drive, copy its path, for windows, it should look like this – C:/Users/…/chromedriver.exe.
2. Then call the function pywhatkit. ...
3. Next call the function pywhatkit.

Import decouple

Python Decouple is a Python library aimed at making it easier for developers to separate their configuration settings from code. Originally designed for Django, it is now a generic Python tool for storing parameters and defining constant values separate from your code.

Before going to the remaining tasks we need to know about the API's

API

Application programming interfaces, or APIs, simplify software development and innovation by enabling applications to exchange data and functionality easily and securely.

An application programming interface, or API, enables companies to open up their applications' data and functionality to external third-party developers, business partners, and internal departments within their companies. This allows services and products to communicate with each other and leverage each other's data and functionality through a documented interface. Developers don't need to know how an API is implemented; they simply use the interface to communicate with other products and services. API use has surged over the past decade, to the degree that many of the most popular web applications today would not be possible without APIs.

An API is a set of defined rules that explain how computers or applications communicate with one another. APIs sit between an application and the web server, acting as an intermediary layer that processes data transfer between systems.

1. **A client application initiates an API call** to retrieve information—also known as a *request*. This request is processed from an application to the web server via the API's Uniform Resource Identifier (URI) and includes a request verb, headers, and sometimes, a request body.
2. **After receiving a valid request**, the API makes a call to the external program or web server.
3. **The server sends a *response*** to the API with the requested information.
4. **The API transfers the data** to the initial requesting application.

Defining Task 9: What is my IP address

ipify provides a simple public IP address API. We just need to make a GET request on this URL: https://api64.ipify.org/?format=json. It returns JSON

{"ip":"122.173.170.82"}

JSON

JSON stands **for JavaScript Object Notation**. ... In Python, the text of JSON is read as quoted-string which contains the value in key-value mapping within { }. Once parsed, it is available as a dictionary object in Python. Python comes with a built-in package called json for encoding and decoding JSON data.

JSON can **store Lists, bools, numbers, tuples and dictionaries**. ... Use the import function to import the JSON module. import json. The JSON module is mainly used to convert the python dictionary above into a JSON string that can be written into a file.

We can then simply return the IP from the JSON data. So, let's create this method:

```
def find_my_ip():
    ip_address = requests.get('https://api64.ipify.org?format=json').json()
    return ip_address["ip"]
```

Defining Task 10: To search something on google

Here we'll be using *PyWhatKit* for searching on Google.

```
def search_on_google(query):
    kit.search(query)
```

It has a method search() that helps us search on Google instantly.

Defining Task 11: To search something on Youtue

For playing videos on YouTube, we are using *PyWhatKit*. We have already imported it as kit.

```
def play_on_youtube(video):
    kit.playonyt(video)
```

PyWhatKit has a playonyt() method that accepts a topic as an argument. It then searches the topic on YouTube and plays the most appropriate video. It uses PyAutoGUI under the hood.

PyAutoGUI

PyAutoGUI is a Python automation library used to click, drag, scroll, move, etc. It can be used to click at an exact position.

Defining Task 12: Send Whatsup message

We'll be using *PyWhatKit* once again for sending WhatsApp messages.

```
def send_whatsapp_message(number, message):
    kit.sendwhatmsg_instantly(f"+91{number}", message)
```

Our method accepts two arguments – the phone number number and the message. It then calls the sendwhatmsg_instantly() method to send a WhatsApp message. Make sure you've already logged in into your WhatsApp account on WhatsApp for Web.

Defining Task 13: To get latest news headlines

To fetch the latest news headlines, we'll be using NewsAPI. Signup for a free account on NewsAPI and get the API Key. Add the **NEWS_API_KEY** in the .env file.

In the above method, we're first creating an empty list called news_headlines. We are then making a GET request on the API URL specified in the NewsAPI Documentation. A sample JSON response from the request looks like this:

```
{
  "status": "ok",
  "totalResults": 38,
  "articles": [
    {
      "source": {
        "id": null,
        "name": "Sportskeeda"
    },
    "author": "Aniket Thakkar",
    "title": "Latest Free Fire redeem code to get Weapon loot crate today (14 October 2021) - Sportskeeda",
    "description": "Gun crates are one of the ways that players in Free Fire can obtain impressive and appealing gun skins.",
    "url": "https://www.sportskeeda.com/free-fire/latest-free-fire-redeem-code-get-weapon-loot-crate-today-14-october-2021",
    "urlToImage": "https://staticg.sportskeeda.com/editor/2021/10/d0b83-16341799119781-1920.jpg",
    "publishedAt": "2021-10-14T03:51:50Z",
    "content": null
   },
   {
    "source": {
     "id": null,
     "name": "NDTV News"
    },
    "author": null,
    "title": "BSF Gets Increased Powers In 3 Border States: What It Means - NDTV",
    "description": "Border Security Force (BSF) officers will now have the power toarrest, search, and of seizure to the extent of 50 km inside three newstates sharing international boundaries with Pakistan and Bangladesh.",
    "url": "https://www.ndtv.com/india-news/bsf-gets-increased-powers-in-3-border-states-what-it-means-2574644",
    "urlToImage": "https://c.ndtvimg.com/2021-08/eglno7qk_-bsf-recruitment-2021_625x300_10_August_21.jpg",
    "publishedAt": "2021-10-14T03:44:00Z",
    "content": "This move is quickly snowballing into a debate on state autonomy. New Delhi: Border Security Force (BSF) officers will now have the power to arrest, search, and of seizure to the extent of 50 km ins… [+4143 chars]"
   },
```

Since the news is contained in a list called articles, we are creating a variable articles with the value res['articles']. Now we are iterating over this articles list and appending the article["title"] to the news_headlines list. We are then returning the first five news headlines from this list.

Defining Task 14: To get the Weather report

To get the weather report, we're using the OpenWeatherMap API. Signup for a free account and get the APP ID. Make sure you add the **OPENWEATHER_APP_ID** in the .env file.

As per the OpenWeatherMap API, we need to make a GET request on the above-mentioned URL with the city name. We'll get a JSON response as:

```
{
    "coord": {
        "lon": 85,
        "lat": 24.7833
    },
    "weather": [
        {
            "id": 721,
            "main": "Haze",
            "description": "haze",
            "icon": "50d"
        }
    ],
    "base": "stations",
    "main": {
        "temp": 26.95,
        "feels_like": 26.64,
        "temp_min": 26.95,
        "temp_max": 26.95,
        "pressure": 1011,
        "humidity": 36
    },
    "visibility": 3000,
    "wind": {
        "speed": 2.57,
        "deg": 310
    },
    "clouds": {
        "all": 57
    },
    "dt": 1637227634,
    "sys": {
        "type": 1,
        "id": 9115,
        "country": "IN",
        "sunrise": 1637195904,
        "sunset": 1637235130
    },
    "timezone": 19800,
    "id": 1271439,
    "name": "Gaya",
    "cod": 200
}
```

We'll just need the weather, temperature, and feels_like from the above response.

Defining Task 15: To get the trending movies

To get the trending movies, we'll be using The Movie Database (TMDB) API. Signup for a free account and get the API Key. Add the **TMDB_API_KEY** in the .env file.

Just as we did for the latest news headlines, we are creating trending_movies list. Then, as per the TMDB API, we're making a GET request. A sample JSON response looks like this:

```
{
    "poster_path": "/5pVJ9SuuO72IgN6i9kMwQwnhGHG.jpg",
    "video": false,
    "vote_average": 0,
    "overview": "Peter Parker is unmasked and no longer able to separate his normal life from the high-stakes of being a Super Hero. When he asks for help from Doctor Strange the stakes become even more dangerous, forcing him to discover what it truly means to be Spider-Man.",
    "release_date": "2021-12-15",
    "id": 634649,
    "adult": false,
    "backdrop_path": "/vK18znei8Uha2z7ZhZtBa40HIrm.jpg",
    "vote_count": 0,
    "genre_ids": [28, 12, 878],
    "title": "Spider-Man: No Way Home",
    "original_language": "en",
    "original_title": "Spider-Man: No Way Home",
    "popularity": 1084.815,
    "media_type": "movie"
  },
```

From the above response, we just need the title of the movie. We get the results which is a list and then iterate over it to get the movie title and append it to the trending_movies list. In the end, we return the first five elements of the list.

Defining Task 16: To get random joke

To get a random joke, we just need to make a GET request on this URL: https://icanhazdadjoke.com/.

```
def get_random_joke():
    headers = {
        'Accept': 'application/json'
    }
    res = requests.get("https://icanhazdadjoke.com/", headers=headers).json()
    return res["joke"]
```

Defining Task 17: To get random advice

To get a piece of random advice, we're using the Advice Slip API.

```
def get_random_advice():
    res = requests.get("https://api.adviceslip.com/advice").json()
    return res['slip']['advice']
```

Defining Task 18: To restart computer

Shutdown timer on Windows 10 is very simple with the command shutdown -s -t. 15 minutes, 30 minutes, 1 hour. There are several ways to shutdown but it is simple way.

```
 elif 'restart' in query:
            os.system("shutdown /s /t 5")
```

Defining Task 19: To shutdown computer

To shutdown the computer by calling the query.

```
elif 'shutdown' in query:
            os.system("rundll32.exe powrprof.dll,SetSuspendState 0,1,0")
```

Defining Task 20: To take screenshot

Pyautogui

PyAutoGUI is a Python module which can automate your GUI and programmatically control your keyboard and mouse. This article illustrates the GUI functions to create display boxes. alert() : Displays a simple message box with text and a single OK button. Returns the text of the button clicked on.

```
 elif 'take screenshot' in query:
            speak("Sir, please tell me name for this screenshot file")
            name = takeCommand().lower()
            speak("Please sir hold the screen for few seconds, I am taking screenshot")
            time.sleep(5)
            img=pyautogui.screenshot()
            img.save(f"{name}.png")
            speak("I am done sir, the screenshot is in our main floder.")
```

Defining Task 21: To know the current location

By using the api.ipify.org to know the current location.

```
elif 'where i am' in query or 'where we are' in query:
            speak("wait sir, let me check")
            try:
                ipAdd = requests.get('https://api.ipify.org').text
                print(ipAdd)
                url = 'https://get.geojs.io/v1/ip/geo/'+ipAdd+'.json'
                geo_requests = requests.get(url)
                geo_data = geo_requests.json()
                city = geo_data['city']
                country = geo_data['country']
                speak(f" we are in {city} city of {country} country")
            except Exception as e:
                speak("sorry sir,Due to network issues i am not able to find where we are")
```

4.5 SOURCE CODE

for offline functions os_ops.py

```
import os
import subprocess as sp

paths = {
    'notepad': "C:\\WINDOWS\\system32\\notepad.exe",
    'calculator': "C:\\Windows\\System32\\calc.exe",
}


def open_notepad():
    os.startfile(paths['notepad'])


def open_cmd():
    os.system('start cmd')

def open_camera():
    sp.run('start microsoft.windows.camera:', shell=True)
```

for online functions online_ops.py

```
import smtplib
from wsgiref import headers
import requests
import wikipedia
import pywhatkit as kit
from decouple import config
from email.message import EmailMessage
import requests

NEWS_API_KEY = "a6bb48eda85e406ba83b95a640439419"
TMDB_API_KEY = "5bfd5cd2882afc20040eb65bc225d1fb"
OPENWEATHER_APP_ID = "876516cdfb6f567e72955961e1abb8ad"


def find_my_ip():
    ip_address = requests.get('https://api64.ipify.org?format=json').json()
    return ip_address["ip"]



def play_on_youtube(video):
    kit.playonyt(video)


def search_on_google(query):
    kit.search(query)


def send_whatsapp_message(number, message):
    kit.sendwhatmsg_instantly(f"+91{number}", message)



def get_random_joke():
    headers = {
        'Accept': 'application/json'
    }
    res = requests.get("https://icanhazdadjoke.com/", headers=headers).json()
    return res["joke"]
```

```
def get_random_advice():
    res = requests.get("https://api.adviceslip.com/advice").json()
    return res['slip']['advice']


def get_latest_news():
    news_headlines = []
    res = requests.get(
        f"https://newsapi.org/v2/everything?q=apple&from=2022-01-16&to=2022-01-
16&sortBy=popularity&apiKey={NEWS_API_KEY}").json()
    articles = res["articles"]
    for article in articles:
        news_headlines.append(article["title"])
    return news_headlines[:5]

#https://newsapi.org/v2/everything?q=apple&from=2022-01-16&to=2022-01-
16&sortBy=popularity&apiKey={NEWS_API_KEY}

#f"https://newsapi.org/v2/top-
headlines?country=in&apiKey={NEWS_API_KEY}&category=general").json()


def get_trending_movies():
    trending_movies = []
    res = requests.get(

f"https://api.themoviedb.org/3/trending/movie/day?api_key={TMDB_API_KEY}").json()
    results = res["results"]
    for r in results:
        trending_movies.append(r["original_title"])
    return trending_movies[:5]

#https://api.themoviedb.org/3/movie/550?api_key=5bfd5cd2882afc20040eb65bc225d1fb
```

```
def get_weather_report(city):
    res = requests.get(

f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={OPENWEATHER_AP
P_ID}&units=metric").json()
    weather = res["weather"][0]["main"]
    temperature = res["main"]["temp"]
    feels_like = res["main"]["feels_like"]
    return weather, f"{temperature}℃", f"{feels_like}℃"
```

for main executive function jp.py

```
from urllib.parse import quote_from_bytes
import pyttsx3
import speech_recognition as sr
import pyaudio
import datetime
import wikipedia
import webbrowser
import os
import smtplib
from decouple import config
from random import choice
from pprint import pprint
from utils import opening_text
import requests
import pyautogui
import time

from fun.os_ops import open_camera, open_cmd, open_notepad
from fun.online_ops import find_my_ip,
play_on_youtube,search_on_google,get_random_advice,get_random_joke,send_whatsapp_message,get_latest_news,get_trending_movies,get_weather_report

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

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

def wishme():
    hour = int(datetime.datetime.now().hour)
    if hour>=0 and hour<12:
        speak("Good Morning!")
    elif hour>=12 and hour<18:
        speak("Good Afternoon!")
    else:
        speak("Good Evening!")
    speak("I am your desktop assistant. Please tell me how may I help you")


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

    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")
        if not 'exit' in query or 'stop' in query:
            speak(choice(opening_text))
        else:
            speak("Have a good day sir!")
            exit()

       # print(f"User said: {query}\n")

    except Exception as e:
        # print(e)
        print("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('chakalimadhu2222@gmail.com','9440253530')
    server.sendmail('chakalimadhu2222@gmail.com',to,content)
    server.close();



if __name__ == '__main__':
    #speak("madhu how r u")
    wishme()
    if 1:
        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 'open youtube' in query:
    webbrowser.open("www.youtube.com")

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

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

elif  'play music' in query:
    music_dir = 'C:\\Users\\DELL\\Music\\mysongs'
    songs = os.listdir(music_dir)
    print(songs)
    os.startfile(os.path.join(music_dir,songs[0]))

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

elif 'open code' in query:
    codePath = "C:\\Users\\DELL\\AppData\\Local\\Programs\\Microsoft VS
Code\\Code.exe"
```

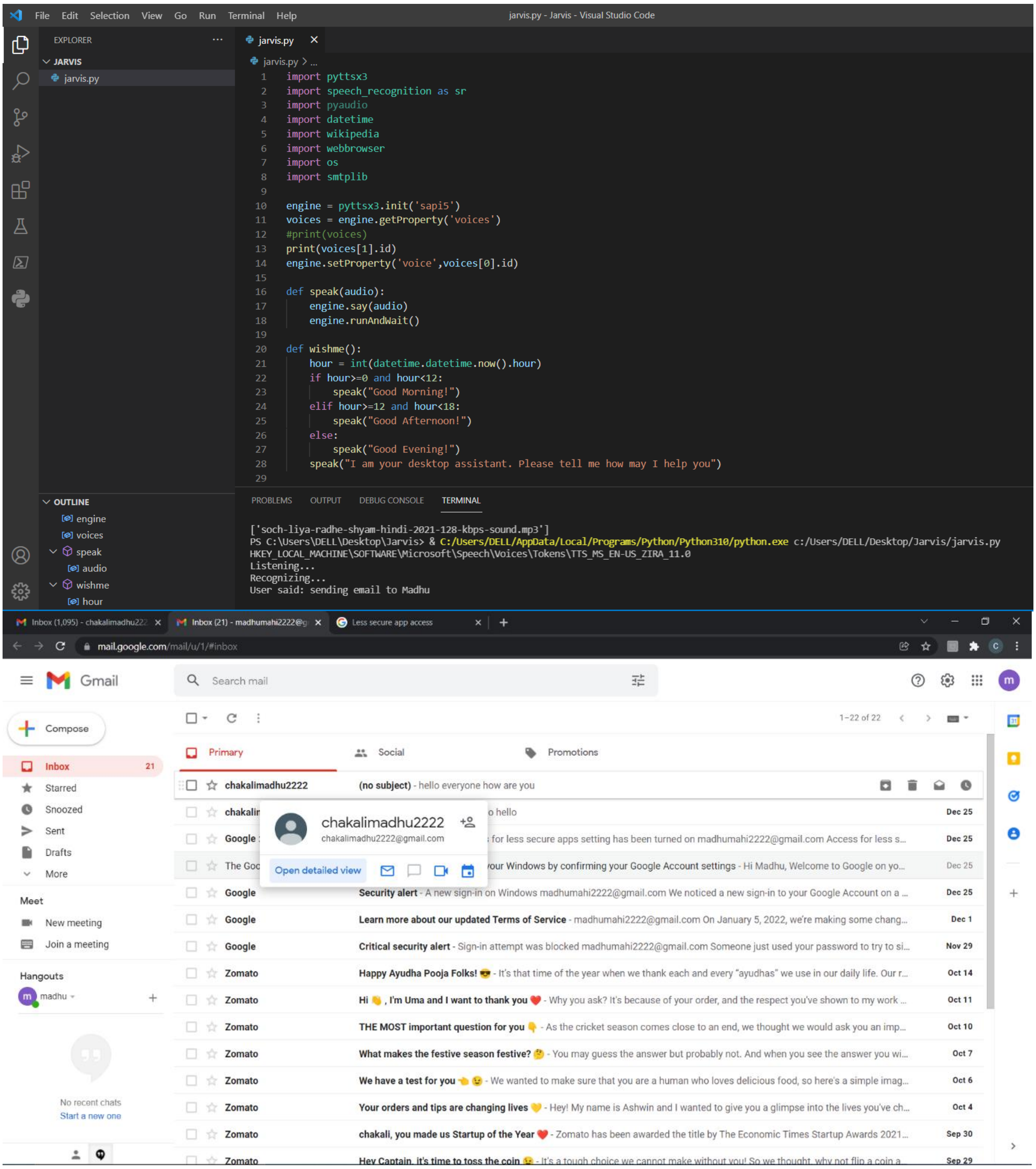
```
            os.startfile(codePath)



        elif 'email to madhu' in query:
            try:
                speak("What should I say?")
                content = takeCommand()
                to = "madhumahi2222@gmail.com"
                sendEmail(to,content)
                speak("Email has been sent!")
            except Exception as e:
                print(e)
                speak("Sorry my friend harry bhai. I am not able to send this email")

        elif 'open notepad' in query:
            open_notepad()

        elif 'open command prompt' in query:
            open_cmd()

        elif 'open camera' in query:
            open_camera()

        elif 'ip address' in query:
                ip_address = find_my_ip()
                speak(f'Your IP Address is {ip_address}.\n For your convenience, I am printing it
on the screen sir.')
                print(f'Your IP Address is {ip_address}')

        elif 'search on youtube' in query:
            speak('What do you want to play on Youtube, sir?')
            video = takeCommand().lower()
            play_on_youtube(video)
```

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

elif "send a whatsapp message" in query:
    speak(
        'On what number should I send the message sir? Please enter in the console: ')
    number = input("Enter a number:")
    speak("What is the message sir?")
    message = takeCommand().lower()
    send_whatsapp_message(number, message)
    speak("I've sent the message sir.")

elif 'tell me joke' in query:
    speak(f"Hope you like this one sir")
    joke = get_random_joke()
    speak(joke)
    speak("For your convenience, I am printing it on the screen sir.")
    pprint(joke)

elif "give an advice" in query:
    speak(f"Here's an advice for you, sir")
    advice = get_random_advice()
    speak(advice)
    speak("For your convenience, I am printing it on the screen sir.")
    pprint(advice)

elif "news update" in query:
    speak(f"I'm reading out the latest news headlines, sir")
    speak(get_latest_news())
    speak("For your convenience, I am printing it on the screen sir.")
```

```
            print(*get_latest_news(), sep='\n')

        elif "trending movies" in query:
            speak(f"Some of the trending movies are: {get_trending_movies()}")
            speak("For your convenience, I am printing it on the screen sir.")
            print(*get_trending_movies(), sep='\n')

        elif 'climate' in query:
            ip_address = find_my_ip()
            city = requests.get(f"https://ipapi.co/{ip_address}/city/").text
            speak(f"Getting weather report for your city {city}")
            weather, temperature, feels_like = get_weather_report(city)
            speak(f"The current temperature is {temperature}, but it feels like {feels_like}")
            speak(f"Also, the weather report talks about {weather}")
            speak("For your convenience, I am printing it on the screen sir.")
            print(f"Description: {weather}\nTemperature: {temperature}\nFeels like:
{feels_like}")

        elif 'restart' in query:
            os.system("shutdown /s /t 5")

        elif 'shutdown' in query:
            os.system("rundll32.exe powrprof.dll,SetSuspendState 0,1,0")

        elif 'take screenshot' in query:
            speak("Sir, please tell me name for this screenshot file")
            name = takeCommand().lower()
            speak("Please sir hold the screen for few seconds, I am taking screenshot")
            time.sleep(5)
            img=pyautogui.screenshot()
            img.save(f"{name}.png")
            speak("I am done sir, the screenshot is in our main floder.")
```

```
elif 'open facebook' in query:
    webbrowser.open("www.facebook.com")

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

elif 'where i am' in query or 'where we are' in query:
    speak("wait sir, let me check")
    try:
        ipAdd = requests.get('https://api.ipify.org').text
        print(ipAdd)
        url = 'https://get.geojs.io/v1/ip/geo/'+ipAdd+'.json'
        geo_requests = requests.get(url)
        geo_data = geo_requests.json()
        city = geo_data['city']
        country = geo_data['country']
        speak(f" we are in {city} city of {country} country")
    except Exception as e:
        speak("sorry sir,Due to network issues i am not able to find where we are")
```

4.5.1 OUTPUT SCREENS



```
import pyttsx3
import speech_recognition as sr
import pyaudio
import datetime
import wikipedia
import webbrowser
import os
import smtplib

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

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

def wishme():
    hour = int(datetime.datetime.now().hour)
    if hour>=0 and hour<12:
        speak("Good Morning!")
    elif hour>=12 and hour<18:
        speak("Good Afternoon!")
    else:
        speak("Good Evening!")
    speak("I am your desktop assistant. Please tell me how may I help you")
```

```
['soch-liya-radhe-shyam-hindi-2021-128-kbps-sound.mp3']
PS C:\Users\DELL\Desktop\Jarvis> & C:/Users/DELL/AppData/Local/Programs/Python/Python310/python.exe c:/Users/DELL/Desktop/Jarvis/jarvis.py
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_EN-US_ZIRA_11.0
Listening...
Recognizing...
User said: sending email to Madhu
```

chakalimadhu2222 (no subject) - hello everyone how are you

chakalimadhu2222
chakalimadhu2222@gmail.com

chakalin … o hello Dec 25

Google … for less secure apps setting has been turned on madhumahi2222@gmail.com Access for less s... Dec 25

The Goo … your Windows by confirming your Google Account settings - Hi Madhu, Welcome to Google on yo... Dec 25

Google Security alert - A new sign-in on Windows madhumahi2222@gmail.com We noticed a new sign-in to your Google Account on a ... Dec 25

Google Learn more about our updated Terms of Service - madhumahi2222@gmail.com On January 5, 2022, we're making some chang... Dec 1

Google Critical security alert - Sign-in attempt was blocked madhumahi2222@gmail.com Someone just used your password to try to si... Nov 29

Zomato Happy Ayudha Pooja Folks! 😎 - It's that time of the year when we thank each and every "ayudhas" we use in our daily life. Our r... Oct 14

Zomato Hi 👋 , I'm Uma and I want to thank you ❤️ - Why you ask? It's because of your order, and the respect you've shown to my work ... Oct 11

Zomato THE MOST important question for you 👇 - As the cricket season comes close to an end, we thought we would ask you an imp... Oct 10

Zomato What makes the festive season festive? 🤔 - You may guess the answer but probably not. And when you see the answer you wi... Oct 7

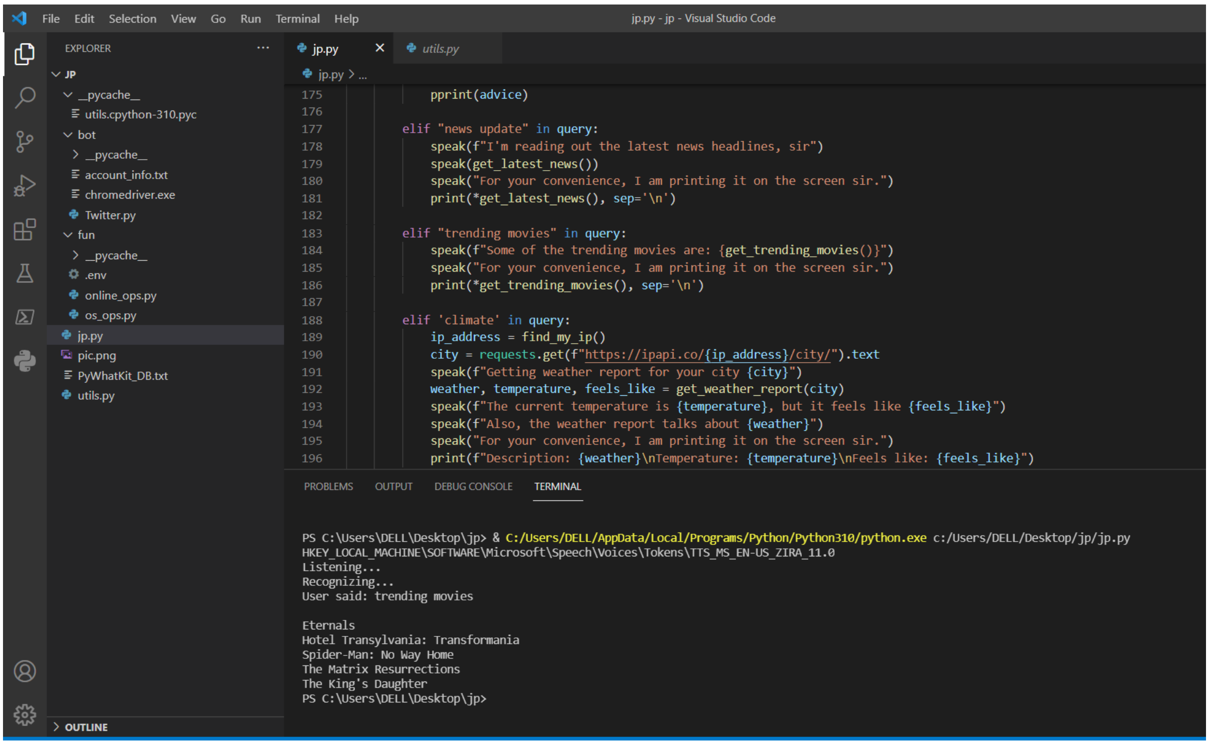
Zomato We have a test for you 👈 😉 - We wanted to make sure that you are a human who loves delicious food, so here's a simple imag... Oct 6

Zomato Your orders and tips are changing lives 💛 - Hey! My name is Ashwin and I wanted to give you a glimpse into the lives you've ch... Oct 4

Zomato chakali, you made us Startup of the Year ❤️ - Zomato has been awarded the title by The Economic Times Startup Awards 2021... Sep 30

Zomato Hey Captain, it's time to toss the coin 😉 - It's a tough choice we cannot make without you! So we thought, why not flip a coin a... Sep 29

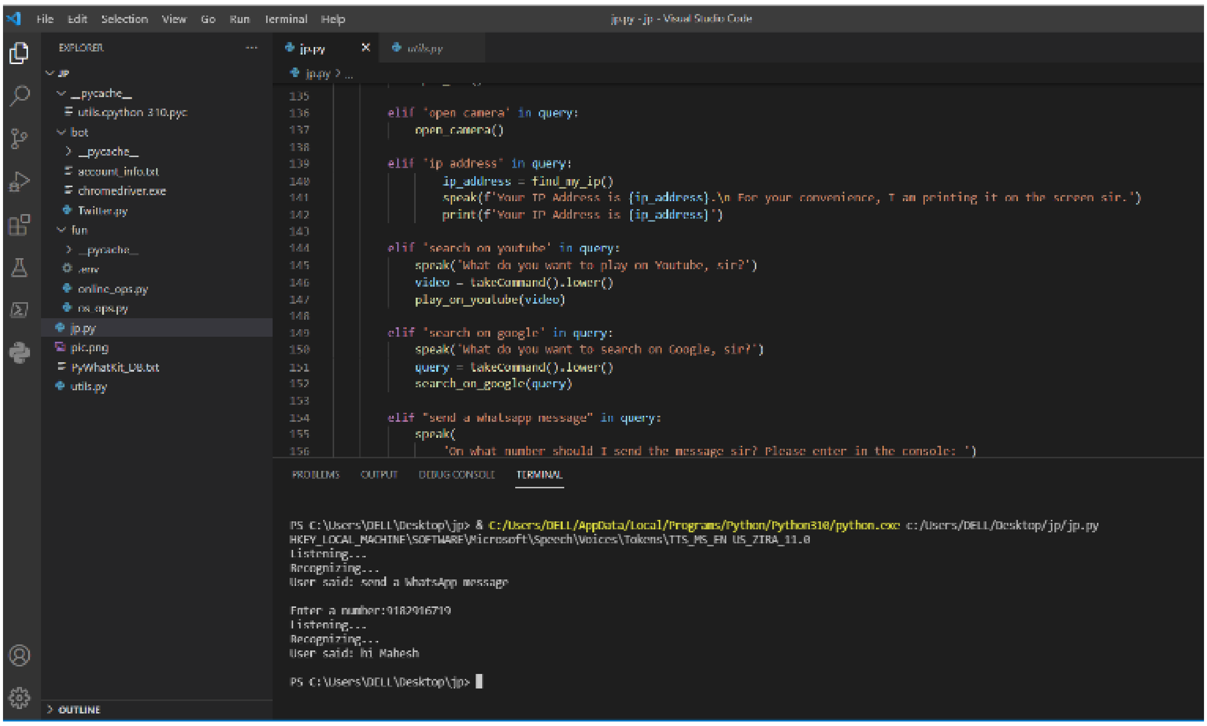
Fig (i) 4.5.1 Output screen for Task .5



```
            pprint(advice)

        elif "news update" in query:
            speak(f"I'm reading out the latest news headlines, sir")
            speak(get_latest_news())
            speak("For your convenience, I am printing it on the screen sir.")
            print(*get_latest_news(), sep='\n')

        elif "trending movies" in query:
            speak(f"Some of the trending movies are: {get_trending_movies()}")
            speak("For your convenience, I am printing it on the screen sir.")
            print(*get_trending_movies(), sep='\n')

        elif 'climate' in query:
            ip_address = find_my_ip()
            city = requests.get(f"https://ipapi.co/{ip_address}/city/").text
            speak(f"Getting weather report for your city {city}")
            weather, temperature, feels_like = get_weather_report(city)
            speak(f"The current temperature is {temperature}, but it feels like {feels_like}")
            speak(f"Also, the weather report talks about {weather}")
            speak("For your convenience, I am printing it on the screen sir.")
            print(f"Description: {weather}\nTemperature: {temperature}\nFeels like: {feels_like}")
```

```
PS C:\Users\DELL\Desktop\jp> & C:/Users/DELL/AppData/Local/Programs/Python/Python310/python.exe c:/Users/DELL/Desktop/jp/jp.py
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_EN-US_ZIRA_11.0
Listening...
Recognizing...
User said: trending movies

Eternals
Hotel Transylvania: Transformania
Spider-Man: No Way Home
The Matrix Resurrections
The King's Daughter
PS C:\Users\DELL\Desktop\jp>
```

Fig (ii) 4.5.1 Output screen for Task .14



```

        elif 'open camera' in query:
            open_camera()

        elif 'ip address' in query:
                ip_address = find_my_ip()
                speak(f'Your IP Address is {ip_address}.\n For your convenience, I am printing it on the screen sir.')
                print(f'Your IP Address is {ip_address}')

        elif 'search on youtube' in query:
            speak('What do you want to play on Youtube, sir?')
            video = takeCommand().lower()
            play_on_youtube(video)

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

        elif "send a whatsapp message" in query:
            speak(
                'On what number should I send the message sir? Please enter in the console: ')
```

```
PS C:\Users\DELL\Desktop\jp> & C:/Users/DELL/AppData/Local/Programs/Python/Python310/python.exe c:/Users/DELL/Desktop/jp/jp.py
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_EN US_ZIRA_11.0
Listening...
Recognizing...
User said: send a WhatsApp message

Enter a number:9182916719
Listening...
Recognizing...
User said: hi Mahesh

PS C:\Users\DELL\Desktop\jp> █
```

Fig (iii) 4.5.1 Output screen for Task .11

5. TESTING AND VALIDATION

For text data, VOICE ASSISTANCE:

- The algorithms used in this form of technology include PLP features, Viterbi search, deep neural networks, discrimination training, WFST framework, etc. If you are interested in Google's new inventions, keep checking their recent publications on speech. The algorithms used by Google are available in an open-source format.
- The system testing is done on fully integrated system to check whether the requirements are matching or not. The system testing for desktop assistant focuses on the following four parameters:
- 5.1. FUNCTIONALITY: In this we check the functionality of the system whether the system performs the task which it was intended to do. To check the functionality each function was checked and run, if it is able to execute the required task correctly then the system passes in that particular functionality test. For example to check whether J can search on Google or not, as we can see in the figure 7.1, user said "Open Google", then Jarvis asked, "What should I search on Google?" then user said, "What is Python", Jarvis open Google and searched for the required input. Figure 7.1 Input through voice commands.
- Removes any word under a certain user-defined frequency threshold.
- NLP refers to the branch of computer science and more specificially, the branch of artificial intelligence, concerned with giving computers the ability to understand text
- And spoken words in much the same way human can
- 5.2. USABILITY: Usability of a system is checked by measuring the easiness of the software and how user friendly it is for the user to use, how it responses to each query that is being asked by the user. It makes it easier to complete any task as it automatically do it by using the essential module or libraries of Python, in a conversational interaction way. Hence any user when instruct any task to it, they feel like giving task to a human assistant because of the conversational interaction for giving input and getting the desired output in the form of task done. The desktop assistant is reactive which means it know human language very well and understand the context that is provided by the user and gives response in the same way, i.e. human understandable language, English. So user finds its reaction in an informed and smart way. The main application of it can be its multitasking ability. It can ask for

continuous instruction one after other until the user “QUIT” it. It asks for the instruction and listen the response that is given by user without needing any trigger phase and then only executes the task.

- 5.3. SECURITY: The security testing mainly focuses on vulnerabilities and risks. As is a local desktop application, hence there is no risk of data breaching through remote access. The software is dedicated to a specific system so when the user logs in, it will be activated.
- 5.4. STABILITY: Stability of a system depends upon the output of the system, if the output is bounded and specific to the bounded input then the system is said to be stable. If the system works on all the poles of functionality then it is stable.
- 5.5 ACCURACY: This virtual assistant has greater accuracy. It can take the input and gives the output in both voice and text quickly. But we don’t compare our virtual assistant accuracy with big gaints like google assistant, apple siri, Microsoft cortona, etc
- It can perform the every basic operation a virtual assistant can perform.

6.CONCLUSION

Through this voice assistant, we have automated various services using a single line command. It eases most of the tasks of the user like searching the web, retrieving weather forecast details, vocabulary help and medical related queries. We aim to make this project a complete server assistant and make it smart enough to act as a replacement for a general server administration. The future plans include integrating Jarvis with mobile using React Native to provide a synchronised experience between the two connected devices. Further, in the long run, Voice Assistance is planned to feature auto deployment supporting elastic beanstalk, backup files, and all operations which a general Server Administrator does. The functionality would be seamless enough to replace the Server Administrator with Jarvis.

Currently, the project aims to provide the Users with a Virtual Assistant that would not only aid in their daily routine tasks like searching the web, extracting weather data, vocabulary help and many others but also help in automation of various activities.

In the long run, we aim to develop a complete server assistant, by automating the entire server management process - deployment, backups, auto-scaling, logging, monitoring and make it smart enough to act as a replacement for a 6 general server administrator

7. REFERENCES:

- Artificial Intelligence | Third Edition | By Pearson
- Natural Language Processing by Steven Bird
- https://www.freecodecamp.org/news/python-project-how-to-build-your-own-jarvis-using-python/
- https://indiantts.com/blog/how-speech-recognition-synthesis-work-which-algorithm-used-voice-recognition/#:~:text=Which%20Algorithm%20is%20Used%20in%20Speech%20Recognition%3F%20The,inventions%2C%20keep%20checking%20their%20recent%20publications%20on%20speech
- https://www.codewithharry.com/videos/python-tutorials-for-absolute-beginners-120
- https://www.computerworld.com/article/3252218/cortana-explained-why-microsofts-virtual-assistant-is-wired-for-business.html
- https://en.wikipedia.org/wiki/Amazon_Alexa
- https://www.pocket-lint.com/apps/news/google/137722-what-is-google-assistant-how-does-it-work-and-which-devices-offer-it
- https://www.pocket-lint.com/apps/news/apple/112346-what-is-siri-apple-s-personal-voice-assistant-explained