A PERSONAL VOICE-ACTIVATED DESKTOP ASSISTANT

A Project Work Report

Submitted in the partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

IN

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Submitted by:

MAYANK GUPTA (20BCS6892)

AAYUSH ADHIKARI (21BCS8881)

MITALI GUPTA (20BCS6890)

KARAN JAIN (21BCS8804)

Under the super vision of:

AMIT GARG

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

APEX INSTITUTE OF TECHNOLOGY

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413, PUNJAB

MONTH & YEAR
FEBRUARY, 2022

BONAFIDE CERTIFICATE

Certified that this project report **"A Personal Voice-Activated Desktop Assistant"** is the bonafide work of **Mitali Gupta, Mayank Gupta, Karan Jain, Aayush Adhikari** who carried out the project work under my supervision.

SIGNATURE SUPERVISOR	HEAD OF DEPARTMENT

Submitted for the project viva-voce examination held on___________________

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

DECLARATION

We declare that this project report titled **“A Personal Voice-Activated Desktop Assistant”** submitted in partial fulfilment of the degree of B. Tech in Mechanical Engineering is a record of original work carried out by us under the supervision of Dr. Gurpreet Singh and has not formed the basis for the award of any other degree or diploma, in this or any other institution or university. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

MAYANK GUPTA (20BCS6892)

AAYUSH ADHIKARI (21BCS8881)

MITALI GUPTA (20BCS6890)

KARAN JAIN (21BCS8804)

ACKNOWLEDGEMENT

The task of completing this project report was quite challenging and was not possible without the help of our friends and college faculty. At the very beginning, we would like to thank Hon. Aman Kaushik (HOD CSE Dept., Chandigarh University) for constantly believing in us and our idea and encouraging us in various instances. Next, we would like to thank **Dr. Amit Grag** (Professor Computer Science Dept., Chandigarh University) our project supervisor, who was one of the first people to believe in our idea and guided us throughout our journey of this project. In multiple instances, he guided us when we lost our path or were misguided by other sources. He helped us with technical issues regarding our project further he guided us in writing the project in the appropriate format.

ABSTRACT

Everything in the twenty-first century specifically is automated, including everyday stuff pretty such as bus doors, air conditioning systems, and turning everything on with a kind of single click, to name a sort of few in a particularly big way. In this fast-paced world, the very current study proposes a newer concept of a voice-controlled gadget that recognizes one's speech, processes the request, and details particularly other related information in a sort of major way. We need to basically build gadgets with generally built-in voice recognition, as well as a facial recognition system, that can really detect a person's speech even in a busy situation, which kind of is quite significant. We basically thought it would generally be amusing to specifically create a personal assistant in Python, demonstrating how everything in the twenty-first century generally is automated, including everyday stuff generally such as bus doors, air conditioning systems, and turning everything on with a fairly single click, to name a definitely few in an actually major way.

Speech recognition may actually be done with the Python Speech Recognition module, which for the most part is quite significant. We essentially utilize the Google Speech API because of it's for all intents and purposes great quality, demonstrating that we need to all intents and purposes build gadgets with definitely built-in voice recognition, as well as a facial recognition system, that can literally detect a person's speech even in a busy situation in a subtle way.

TABLE OF CONTENTS

FEASIBILITY STUDY

Feasibility study can help you determine whether or not you should proceed with your project. It is essential to evaluate cost and benefit. It is essential to evaluate cost and benefit of the proposed system. Five types of feasibility study are taken into consideration.

1. Technical feasibility: It includes finding out technologies for the project, both hardware and software. For virtual assistant, user must have microphone to convey their message and a speaker to listen when system speaks. These are very cheap now a days and everyone generally possess them. Besides, system needs internet connection. While using our virtual assistant, make sure you have a steady internet connection. It is also not an issue in this era where almost every home or office has Wi-Fi.

2. Operational feasibility: It is the ease and simplicity of operation of proposed system. System does not require any special skill set for users to operate it. In fact, it is designed to be used by almost everyone. Kids who still don't know to write can read out problems for system and get answers.

3. Economic feasibility: Here, we find the total cost and benefit of the proposed system over current system. For this project, the main cost is documentation cost. User also would have to pay for microphone and speakers. Again, they are cheap and available. As far as maintenance is concerned, our VA won't cost too much.

4. Organizational feasibility: This shows the management and organizational structure of the project. This project is not built by a team. The management tasks

are all to be carried out by a single person. That won’t create any management issues and will increase the feasibility of the project.

5. Cultural feasibility: It deals with compatibility of the project with cultural environment. Virtual assistant is built in accordance with the general culture. The project is named our VA so as to represent Indian culture without undermining local beliefs. This project is technically feasible with no external hardware requirements. Also, it is simple in operation and does not cost training or repairs. Overall feasibility study of the project reveals that the goals of the proposed system are achievable. Decision is taken to proceed with the project

CHAPTER 1

INTRODUCTION

Virtual assistants are software programs that assist you with day-to-day duties such as weather forecasting, setting reminders, and preparing shopping lists, among other things. They can accept text (online chatbots) or voice commands. To activate the listener, voice-based intelligent assistants require an invoking phrase or wake word, followed by the command. There is a plethora of virtual assistants available, including Apple's Siri, Amazon's Alexa, and Microsoft's Cortana.

Almost all duties are now digitalized in today's world. We have a Smartphone in our hands, and it's like having the entire world at our fingertips. We don't even use our fingers anymore.

We just speak of the task and it is done. There are procedures in place where we can text Dad and say, "I'll be late today." And the text is sent. A Virtual Assistant's job is to do just that. It also helps automate search, discovery, and online purchase processes by supporting specific tasks such as booking a trip or locating the cheapest book online from several eCommerce sites and then giving an interface to place an order.

This system is intended to be used on desktop computers. Personal assistant software helps users be more productive by handling their everyday chores and providing them with information from web sources. And it's done in a matter of seconds. Voice searches have surpassed text searches in popularity. Web searches conducted via mobile devices have just eclipsed those carried out using a computer and the researchers are already estimating that 50 percent of queries will be via voice by 2020. Virtual assistants are proving to be more intelligent than ever. Allow your intelligent assistant to do the heavy lifting for you when it comes to email.

In most cases, a user must manually manage many sets of applications to execute a single operation. For example, a person planning a trip should look for airport codes for neighbouring airports and then look for tickets between those airports on travel websites, arrive at the desired location. A system that can effortlessly manage chores is required. We already have several virtual assistants on staff. However, we hardly ever utilize it. A large number of people struggle with speech recognition. These systems are capable of understanding English phrases, but they are unable to distinguish our dialect. Our pronunciation differs significantly from theirs. They are also more user-friendly on mobile devices than desktop systems. A virtual assistant who can understand English with an Indian accent and work on a desktop system is required. When a virtual assistant is unable to effectively answer inquiries, it is due to a lack of context or an understanding of the question's intent. It can only answer relevant questions through thorough optimization integrating both humans and machine learning.

The main purpose of our intelligent virtual assistant is to answer questions that users may have. Virtual assistants can tremendously save you time. We usually spend hours in online searches and then making the report in our terms of understanding. Our virtual assistant can do that for you. Provide a topic for research and continue with your tasks while it does the research. Another difficult task is to open up applications by switching tabs which takes a lot of our precious time. Just tell the assistant what you want to open and see the magic. One of the main advantages of voice searches is their rapidity. To be a wise voice is reputed to be four times faster than a written search: whereas we can write about 40 words per minute, we are capable of speaking around 150 words during the same period. In this respect, the ability of personal assistants to accurately recognize spoken words is a prerequisite for them to be adopted by consumers.

An application of a natural language processing-based intelligent voice assistant system that may be used to transmit text messages. Messages may be sent and received, and the device's other applications can be used on the platform of was designed and implemented utilizing python android.

In most cases, a user must manually manage many sets of applications in order to execute a single operation. A user planning a trip, for example, should look up airport codes for neighbouring airports and then search travel sites for tickets between airport combinations to get to their destination. A system that can effortlessly manage chores is required. We already have a number of virtual assistants on staff. However, we hardly ever utilize it. A large number of people struggle with speech recognition. These systems are capable of understanding English phrases, but they are unable to distinguish our dialect. Our pronunciation differs significantly from theirs. They are also more user-friendly on mobile devices than desktop systems. A virtual assistant who can understand English with an Indian accent and work on a desktop system.

HARDWARE AND SOFTWARE REQUIREMENTS

The software is designed to be light-weighted so that it doesn’t be a burden on the machine running it. This system is being build keeping in mind the generally available hardware and software compatibility.

Here are the minimum hardware and software requirement for virtual assistant.

Hardware:

- Pentium-pro processor or later.
- RAM 512MB or more. Software:
- Windows 7(32-bit) or above.
- Python 2.7 or later
- Chrome Driver
- Selenium Web Automation
- SQLite

CHAPTER 2

LITERATURE REVIEW

A variety of desktop virtual assistants are already available. This section discusses a few instances of existing virtual assistants on the market, as well as the duties they can perform and their limitations.

- **SIRI (Apple)**

SIRI is a voice-activated personal assistant that interacts with the user, detects instructions, and acts on them. It enhances voice recognition over time by learning to adapt to the user's speech. When it can't figure out what the user wants, it attempts to talk to them.

Siri is designed to offer you a seamless way of interacting with your iPhone, iPad, iPod Touch, Apple Watch, Home Pod or Mac by you speaking to her and her speaking back to you to find or do what you need. You can ask her questions, tell her to show you something or issue her with commands for her to execute on your behalf, hands-free.

Siri has access to every other built-in application on your Apple device - Mail, Contacts, Messages, Maps, Safari and so on - and will call upon those apps to present data or search through their databases whenever she needs to. Ultimately, Siri does all the legwork for you.

It works with the device's calendar, contacts, and music library applications, as well as the device's GPS and camera. It makes use of spatial, temporal, social, and task-based contexts to tailor the agent's behaviour to the user at any given time.

❖ **Tasks that are supported**

- Call someone from my contacts list
- Open an app on my iPhone
- Send a text message to someone
- Schedule a meeting at 9 a.m. tomorrow on my calendar Set your alarm for 5 a.m. tomorrow.
- I'd want to play a certain song from my iTunes library.
- Make a new note.

❖ **Drawback**

SIRI does not have its knowledge store and relies on information recorded in domain and data models to comprehend the world.

➢ **Alexa (Amazon)**

Alexa is a virtual assistant that you may manage with your voice. She can play music, control your smart home, answer questions, and connect you to your favourite services, keeping you organized, informed, protected, and entertained. She's also your shopper because she's an Amazon product.

Alexa is a cloud-based voice assistant that can be used with a growing variety of smart speakers and other Alexa-enabled devices.

❖ Tasks that are supported

- Reminders
- Email
- Calendar, Google Calendar
- Outlook
- Evernote
- Facebook, LinkedIn
- News Feeds

Amazon Alexa, also known simply as **Alexa** is a virtual assistant technology largely based on a Polish speech synthesiser named Ivona, bought by Amazon in 2013. It was first used in the Amazon Echo smart speaker and the Echo Dot, Echo Studio and Amazon Tap speakers developed by Amazon Lab126. It is capable of voice interaction, music playback, making to-do lists, setting alarms, streaming podcasts, playing audiobooks, and providing weather, traffic, sports, and other real-time information, such as news. Alexa can also control several smart devices using itself as a home automation system. Users are able to extend the Alexa capabilities by installing "skills" (additional functionality developed by third-party vendors, in other settings more commonly called apps) such as weather programs and audio features. It uses automatic speech recognition, natural language processing, and other forms of weak AI to perform these tasks.

❖ FUNCTIONS OF ALEXA:

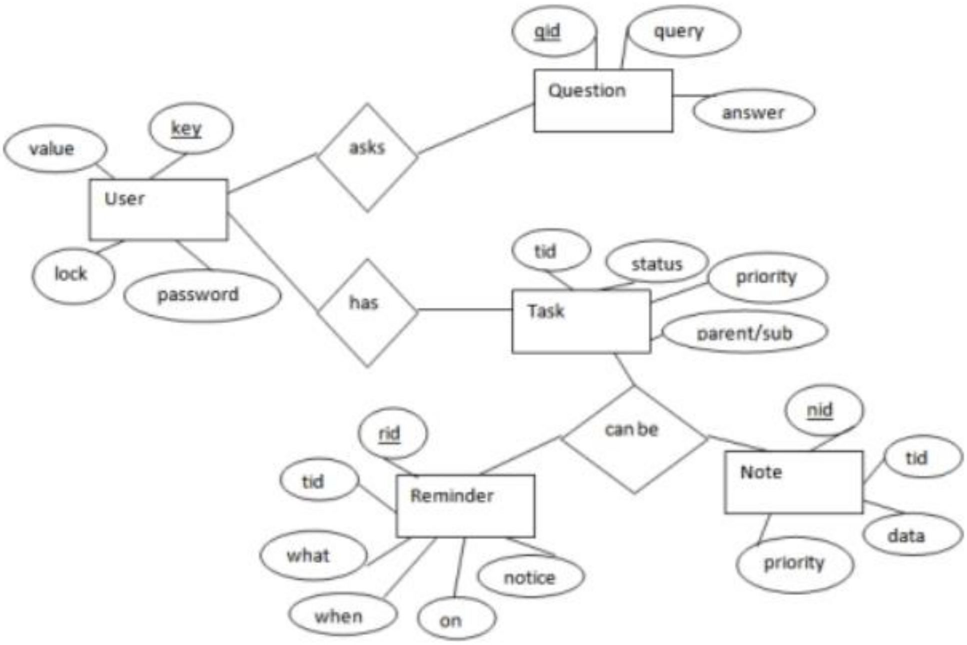
Alexa can perform a number of pre-set functions out-of-the-box such as set timers, share the current weather, create lists, access Wikipedia articles, and many more things. Users say a designated "wake word" (the default is simply "Alexa") to alert an Alexa-enabled device of an ensuing function command. Alexa listens for the command and performs the appropriate function, or skill, to answer a question or command. When questions are asked, Alexa converts sound waves into text which allows it to gather information from various sources. Behind the scenes, the data gathered is then sometimes passed to a variety of sources including Wolfram Alpha, iMDB, AccuWeather, Yelp, Wikipedia, and others to generate suitable and accurate answers. Alexa-supported devices can stream music from the owner's Amazon Music accounts and have built-in support for Pandora and Spotify accounts. Alexa can play music from streaming services such as Apple Music and Google Play Music from a phone or tablet.

In addition to performing pre-set functions, Alexa can also perform additional functions through third-party skills that users can enable Some of the most popular Alexa skills in 2018 included "Question of the Day" and "National Geographic Geo Quiz" for trivia; "TuneIn Live" to listen to live sporting events and news stations; "Big Sky" for hyper local weather updates; "Sleep and Relaxation Sounds" for listening to calming sounds; "Sesame Street" for children's entertainment; and "Fitbit" for Fitbit users who want to check in on their health stats. In 2019, Apple, Google, Amazon, and Zigbee Alliance announced a partnership to make smart home products work together.

CHAPTER 3

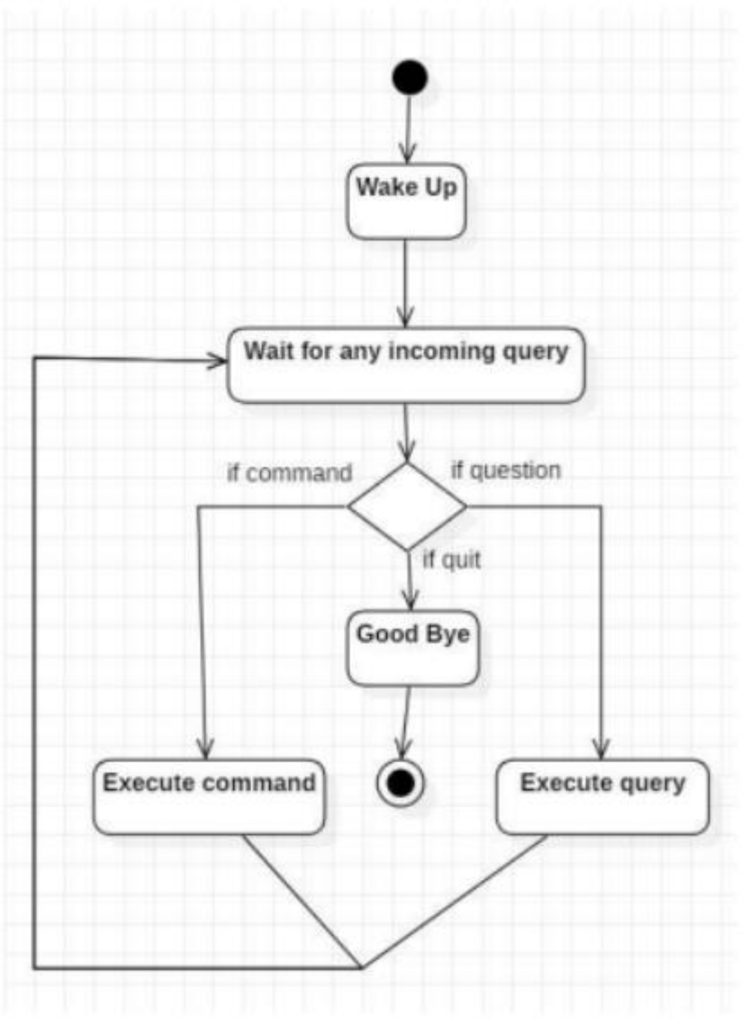
DESIGN FLOW/PROCESS

❖ ER DIAGRAM



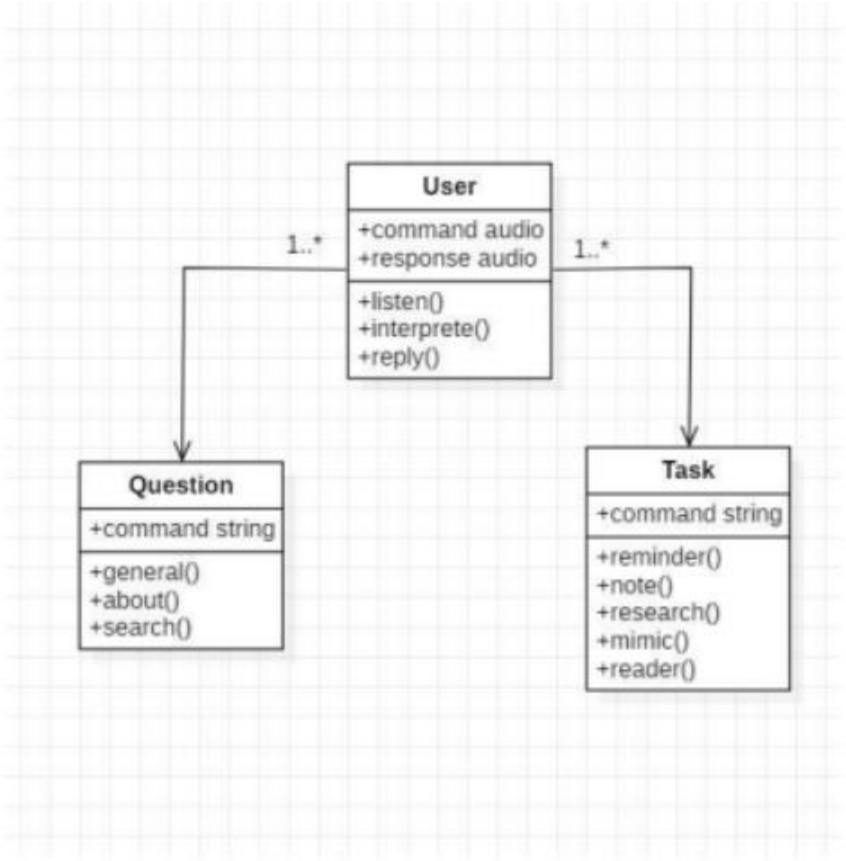
The above diagram shows entities and their relationship for a virtual assistant system. We have a user of a system who can have their keys and values. It can be used to store any information about the user. Say, for key "name" value can be "Jim". For some keys user might like to keep secure. There he can enable lock and set a password (voice clip). Single user can ask multiple questions. Each question will be given ID to get recognized along with the query and its corresponding answer. User can also be having n number of tasks. These should have their own unique id and status i.e. their current state. A task should also have a priority value and its category whether it is a parent task or child task of an older task.

❖ ACTIVITY DIAGRAM



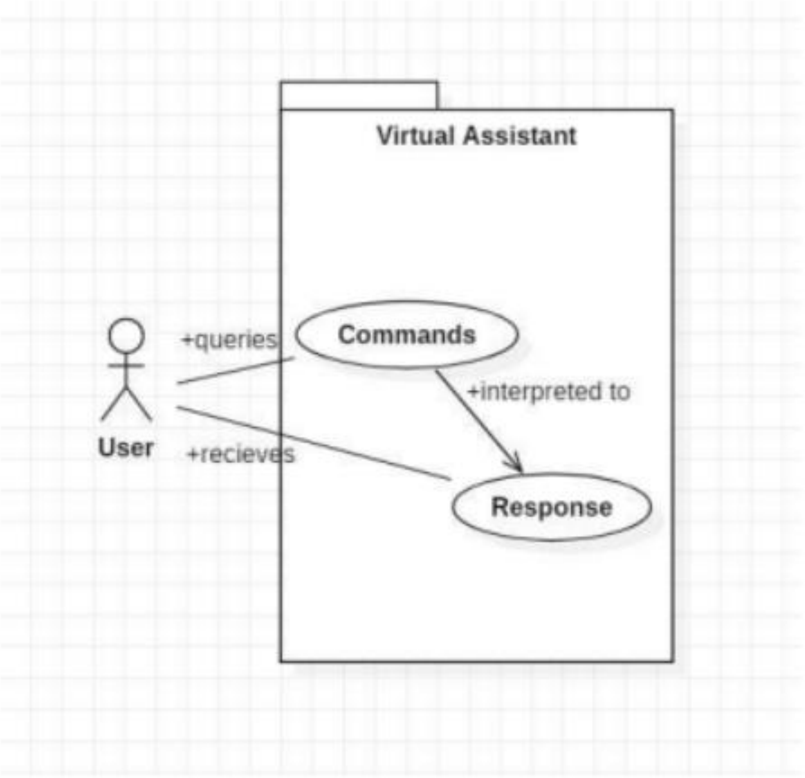
Initially, the system is in idle mode. As it receives any wake up cal it begins execution. The received command is identified whether it is a questionnaire or a task to be performed. Specific action is taken accordingly. After the Question is being answered or the task is being performed, the system waits for another command. This loop continues unless it receives quit command. At that moment, it goes back to sleep.

❖ CLASS DIAGRAM



The class user has 2 attributes command that it sends in audio and the response it receives which is also audio. It performs function to listen the user command. Interpret it and then reply or sends back response accordingly. Question class has the command in string form as it is interpreted by interpret class. It sends it to general or about or search function based on its identification. The task class also has interpreted command in string format. It has various functions like reminder, note, mimic, research and reader.

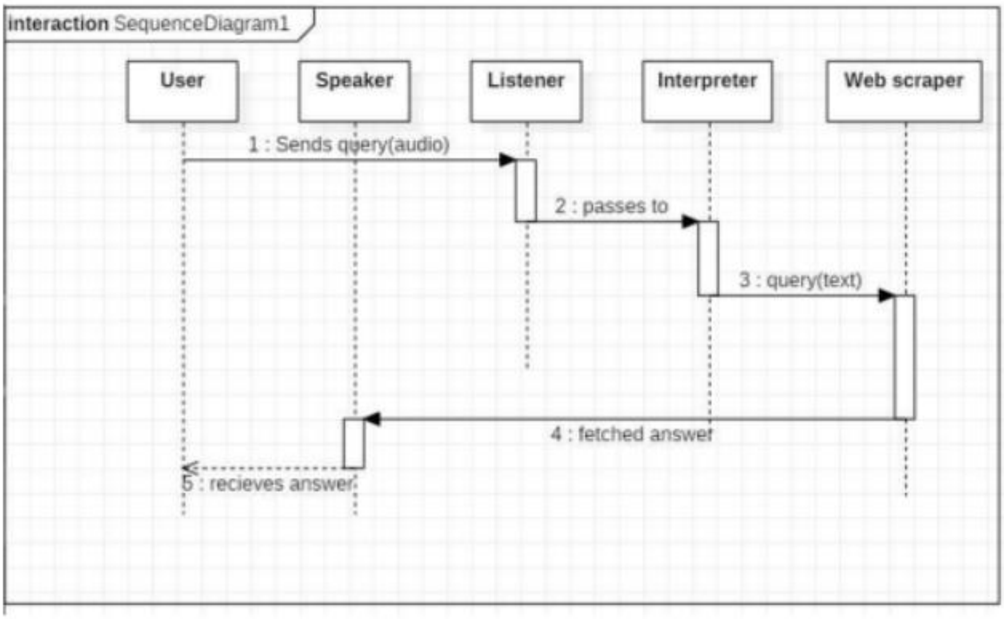
❖ USE CASE DIAGRAM



In this project there is only one user. The user queries command to the system. System then interprets it and fetches answer. The response is sent back to the user.

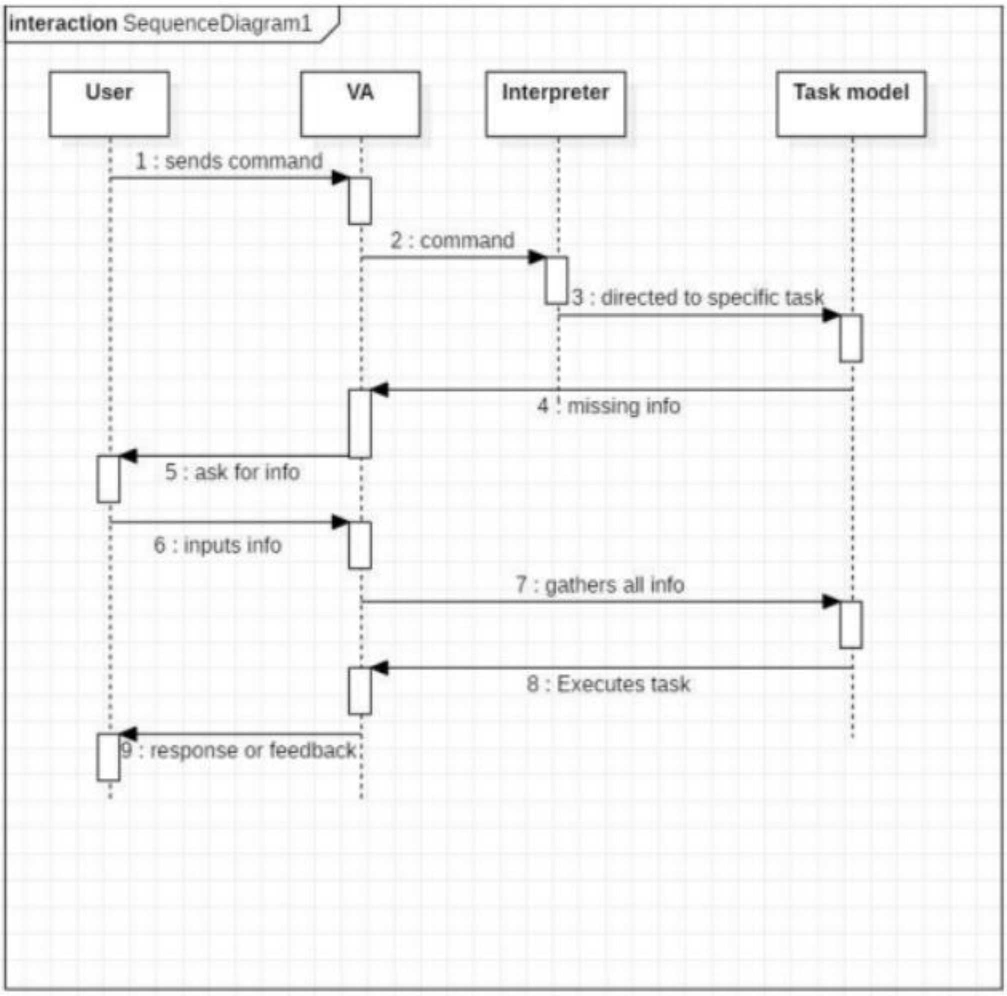
❖ **SEQUENCE DIAGRAM**

❖ Sequence diagram for Query-Response



The above sequence diagram shows how an answer asked by the user is being fetched from internet. The audio query is interpreted and sent to Web scraper. The web scraper searches and finds the answer. It is then sent back to speaker, where it speaks the answer to user.

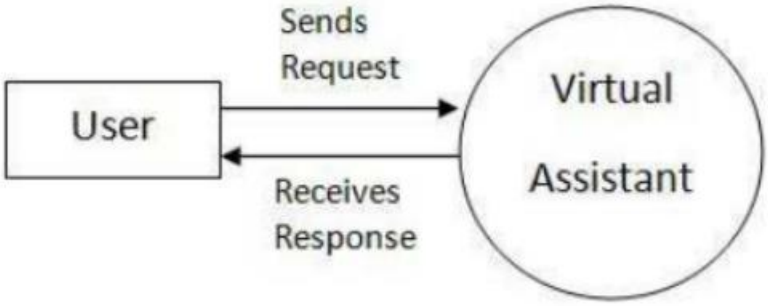
❖ Sequence diagram for Task Execution



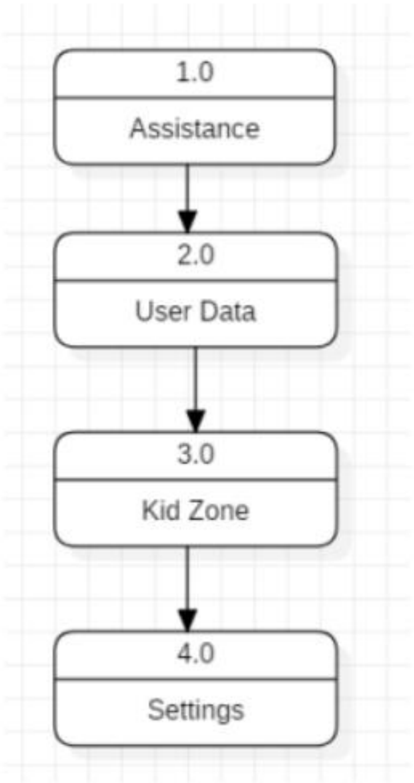
The user sends command to virtual assistant in audio form. The command is passed to the interpreter. It identifies what the user has asked and directs it to task executer. If the task is missing some info, the virtual assistant asks user back about it. The received information is sent back to task and it is accomplished. After execution feedback is sent back to user.

❖ **DATA FLOW DIAGRAM**

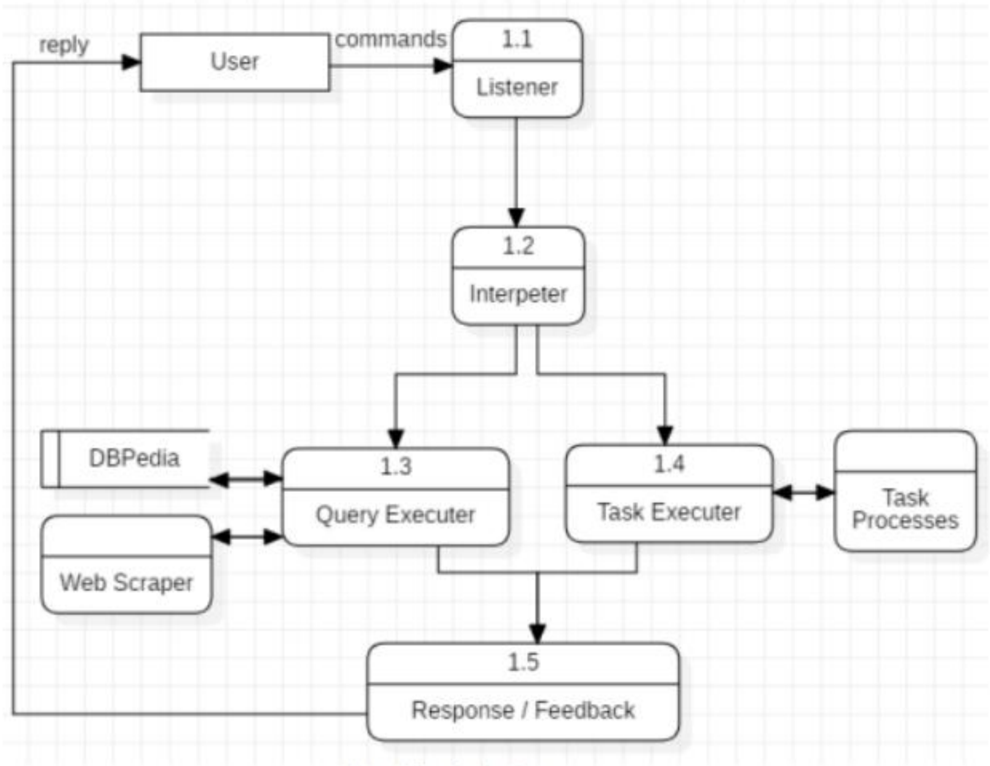
❖ DFD Level 0 (Context Level Diagram)



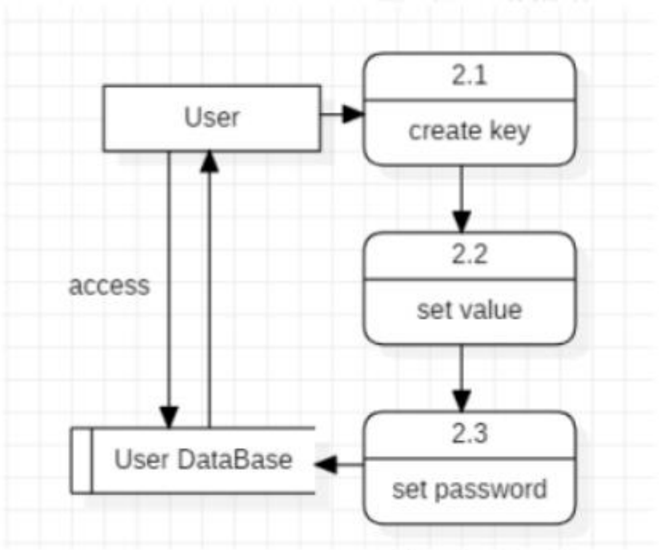
❖ DFD Level 1



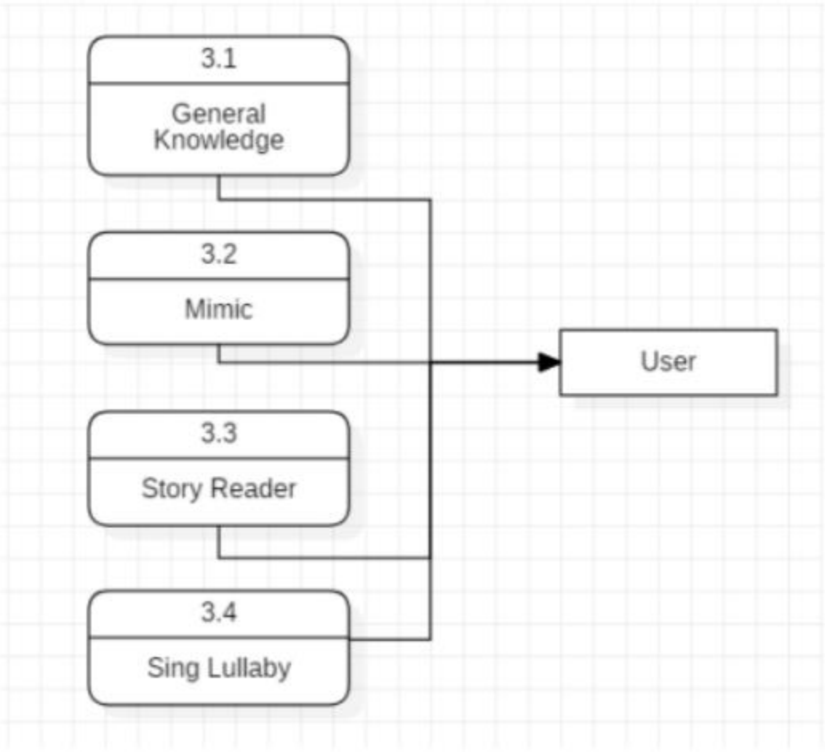
❖ DFD Level 2



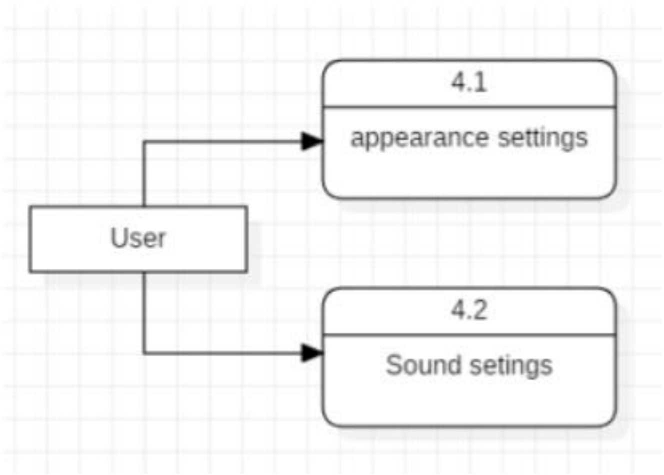
Data Flow in Assistance



Managing User Data

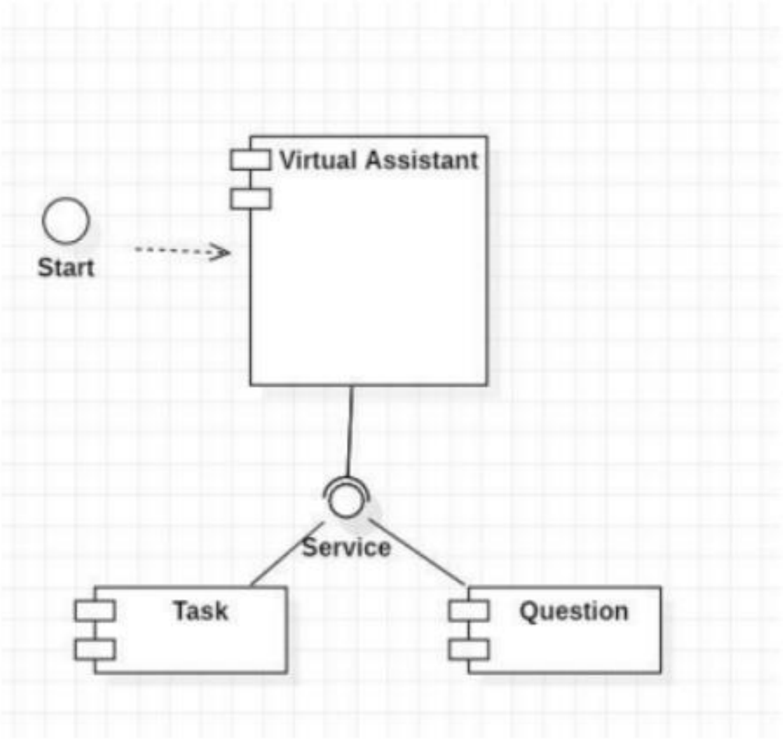


Data Flow in Kid Zone



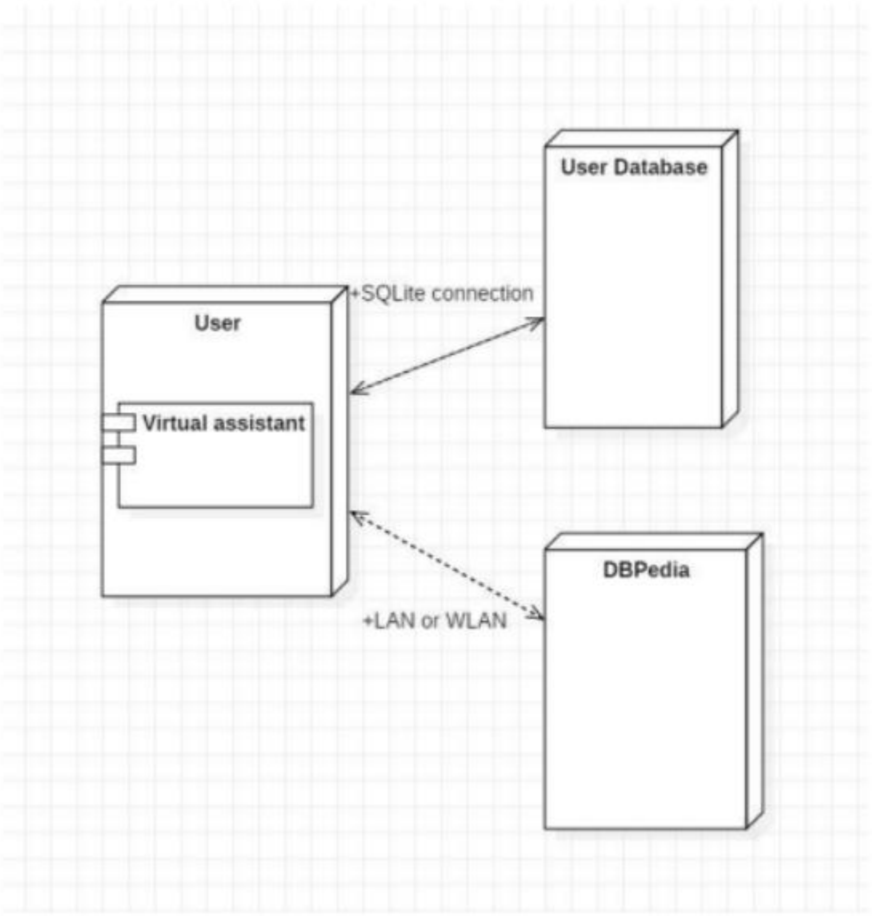
Settings of virtual Assistant

❖ COMPONENT DIAGRAM



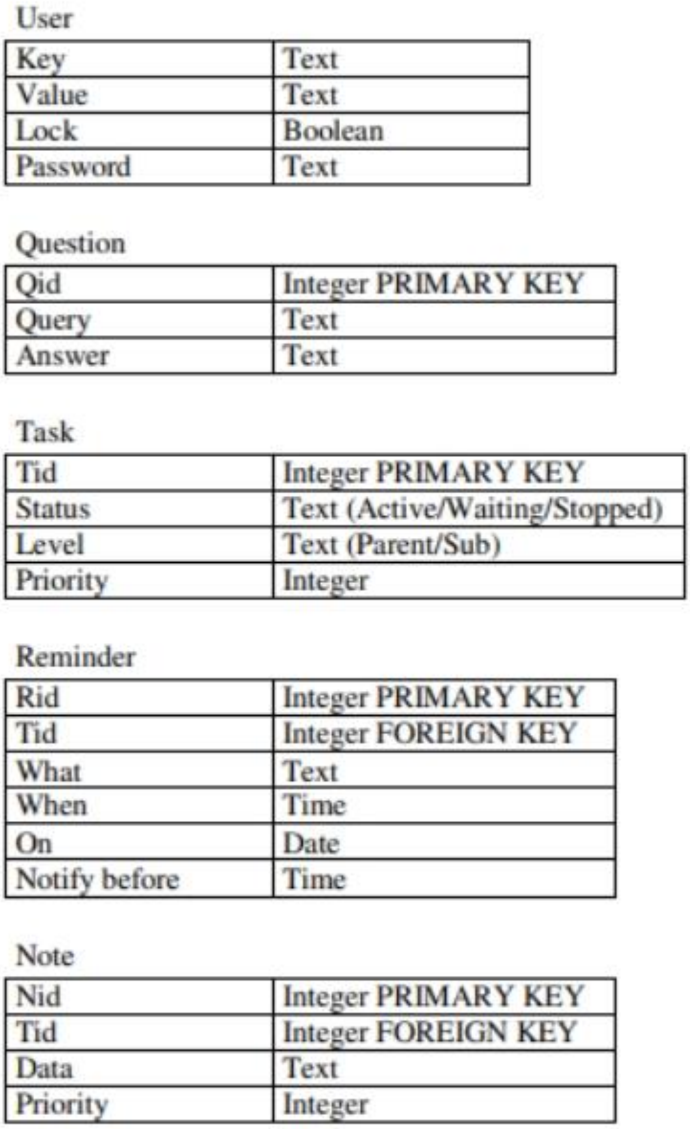
The main component here is the Virtual Assistant. It provides two specific service, executing Task or Answering your question.

❖ DEPLOYMENT DIAGRAM



The user interacts with SQLite database using SQLite connection in Python code. The knowledge database DBPedia must be accessed via internet connection. This requires LAN or WLAN / Ethernet network.

❖ DATA DICTIONARY



User

Key	Text
Value	Text
Lock	Boolean
Password	Text

Question

Qid	Integer PRIMARY KEY
Query	Text
Answer	Text

Task

Tid	Integer PRIMARY KEY
Status	Text (Active/Waiting/Stopped)
Level	Text (Parent/Sub)
Priority	Integer

Reminder

Rid	Integer PRIMARY KEY
Tid	Integer FOREIGN KEY
What	Text
When	Time
On	Date
Notify before	Time

Note

Nid	Integer PRIMARY KEY
Tid	Integer FOREIGN KEY
Data	Text
Priority	Integer

CHAPTER 4

REQUIREMENT AND ANALYSIS

System Analysis is about complete understanding of existing systems and finding where the existing system fails. The solution is determined to resolve issues in the proposed system. It defines the system. The system is divided into smaller parts. Their functions and inter relation of these modules are studied in system analysis.

Problem definition Usually, user needs to manually manage multiple sets of applications to complete one task. For example, a user trying to make a travel plan needs to check for airport codes for nearby airports and then check travel sites for tickets between combinations of airports to reach the destination. There is need of a system that can manage tasks effortlessly. We already have multiple virtual assistants. But we hardly use it. There are number of people who have issues in voice recognition.

These systems can understand English phrases but they fail to recognize in our accent. Our way of pronunciation is way distinct from theirs. Also, they are easy to use on mobile devices than desktop systems. There is need of a virtual assistant that can understand English in Indian accent and work on desktop system. When a virtual assistant is not able to answer questions accurately, it’s because it lacks the proper context or doesn’t understand the intent of the question. Its ability to answer questions relevantly only happens with rigorous optimization, involving both humans and machine learning.

Continuously ensuring solid quality control strategies will also help manage the risk of the virtual assistant learning undesired bad behaviours. They require large amount of information to be fed in order for it to work efficiently.

Virtual assistant should be able to model complex task dependencies and use these models to recommend optimized plans for the user. It needs to be tested for finding optimum paths when a task has multiple sub-tasks and each sub-task can have its own sub-tasks. In such a case there can be multiple solutions to paths, and the it should be able to consider user preferences, other active tasks, priorities in order to recommend a particular plan.

Personal assistant software is required to act as an interface into the digital world by understanding user requests or commands and then translating into actions or recommendations based on agent's understanding of the world. Our VA focuses on relieving the user of entering text input and using voice as primary means of user input. Agent then applies voice recognition algorithms to this input and records the input. It then uses this input to call one of the personal information management applications such as task list or calendar to record a new entry or to search about it on search engines like Google, Bing or Yahoo etc. Focus is on capturing the user input through voice, recognizing the input and then executing the tasks if the agent understands the task. Software takes this input in natural language, and so makes it easier for the user to input what he or she desires to be done. Voice recognition software enables hands free use of the applications, lets users to query or command the agent through voice interface. This helps users to have access to the agent while performing other tasks and thus enhances value of the system itself.

Our VA also have ubiquitous connectivity through Wi-Fi or LAN connection, enabling distributed applications that can leverage other APIs exposed on the web without a need to store them locally. Virtual assistants must provide a wide variety of services.

These include:

- Providing information such as weather, facts from e.g. Wikipedia etc.
- Set an alarm or make to-do lists and shopping lists.
- Remind you of birthdays and meetings.
- Play music from streaming services such as Saavn and Gaana.
- Play videos, TV shows or movies on televisions, streaming from e.g. Netflix or Hotstar.
- Book tickets for shows, travel and movies.

❖ PRESENT SYSTEM

We are familiar with many existing voice assistants like Alexa, Siri, Google Assistant, Cortana which uses concept of language processing, and voice recognition. They listen 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.

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. Our AI 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, Speech Recognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyautogui, pyQt etc. By making this project, we 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 make a reminder for you in a 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 have some basic conversation.

❖ Modules Imported:

Python:

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages. Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, data science etc. Python has a lot of libraries for every need of this project. For Our VA, libraries used are speech recognition to recognize voice, Pyttsx for text to speech, selenium for web automation etc. Python is reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most the time, and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code (because you will have more time to optimize) than writing everything in a low-level language.

- **WolframAlpha** –

It is used to compute expert-level answers of any command using Wolfram's algorithms, knowledgebase and AI technology.

- **JSON** –

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON is built on two structures:

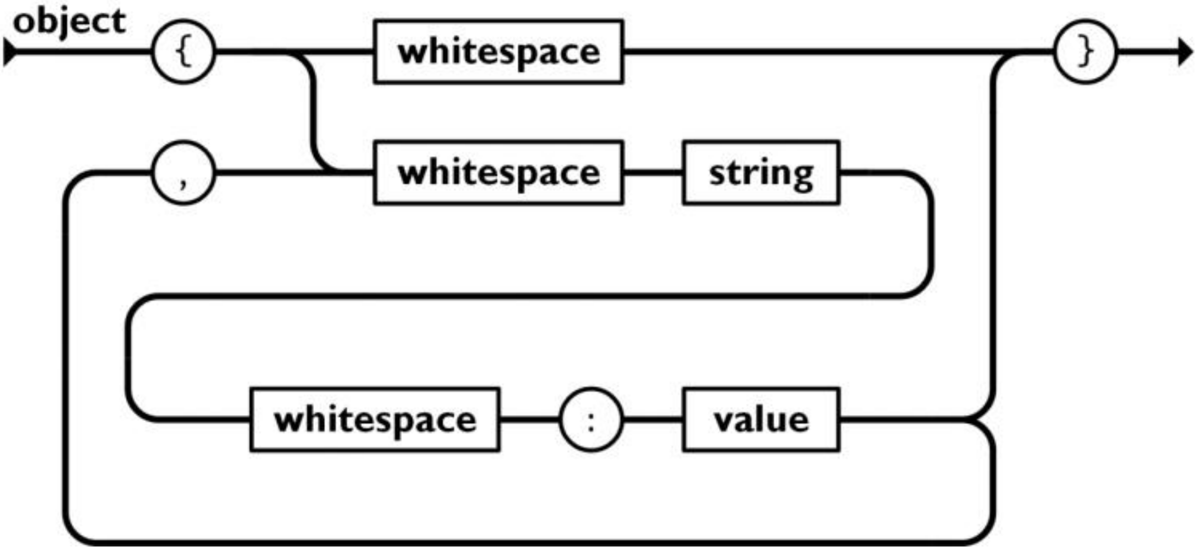
A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.

An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence.

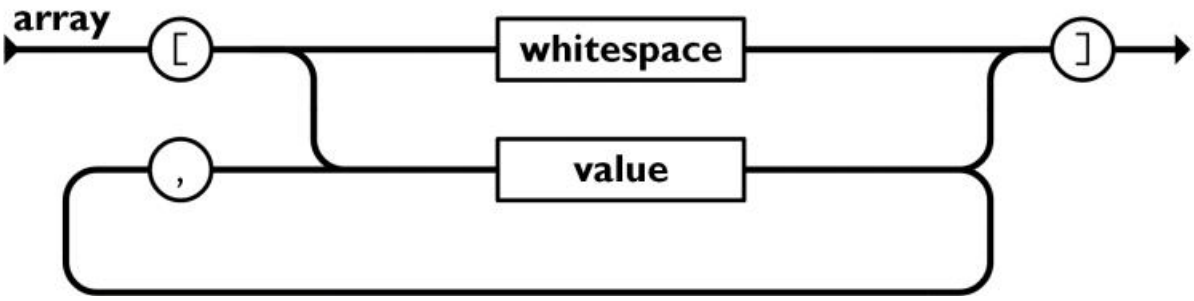
These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

In JSON, they take on these forms:

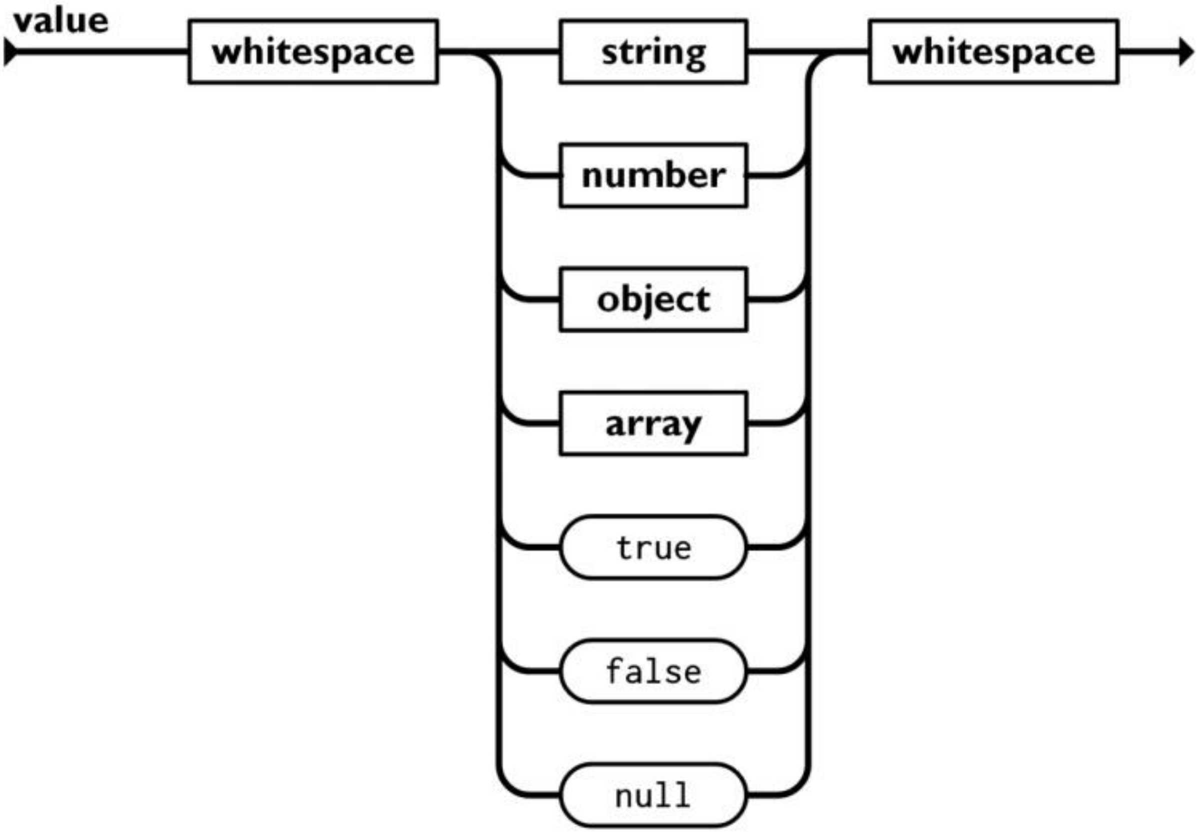
An *object* is an unordered set of name/value pairs. An object begins with **{***left brace* and ends with **}** *right brace*. Each name is followed by **:***colon* and the name/value pairs are separated by **,***comma*.



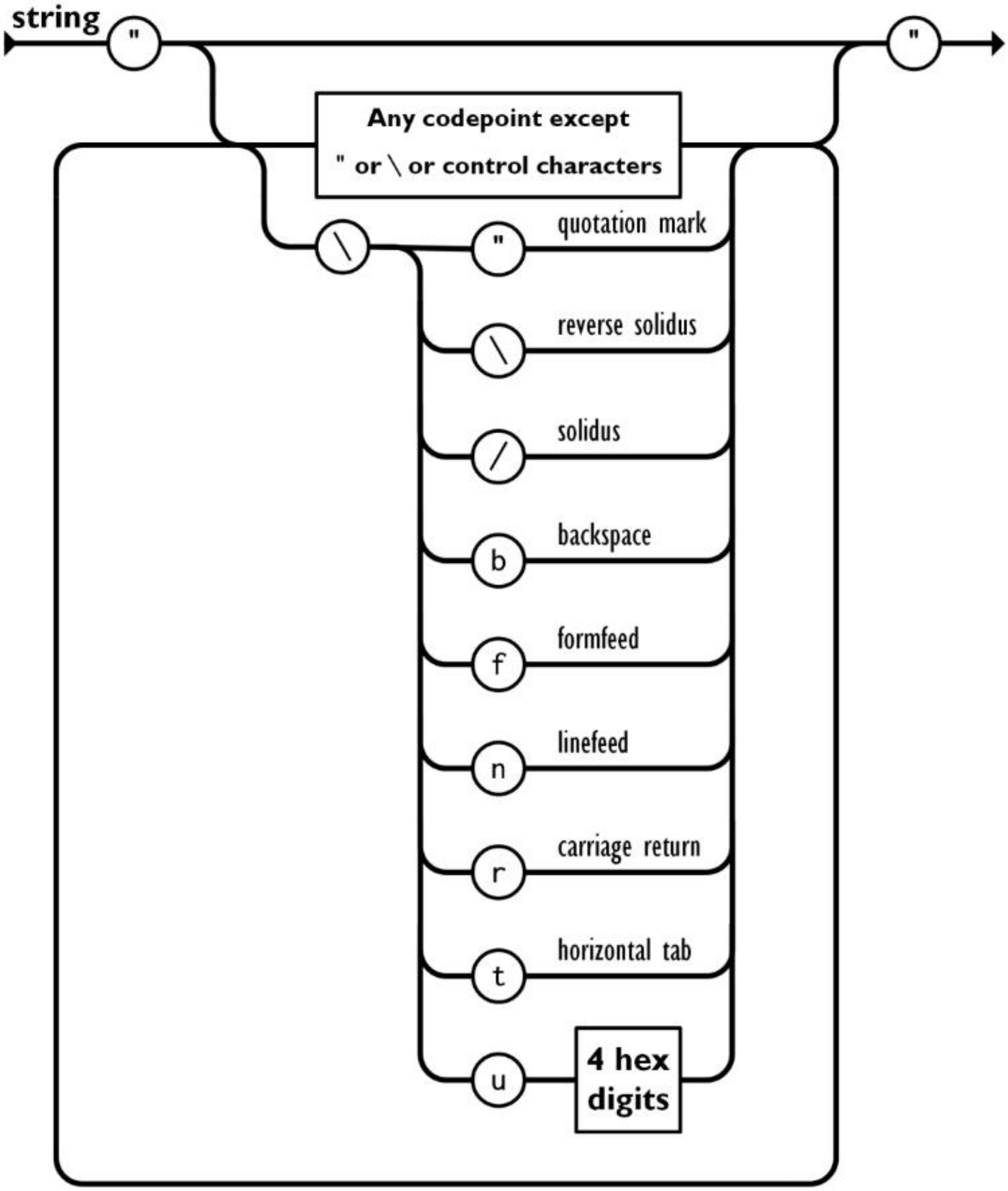
An array is an ordered collection of values. An array begins with [left bracket and ends with (]) right bracket. Values are separated by (,) comma.



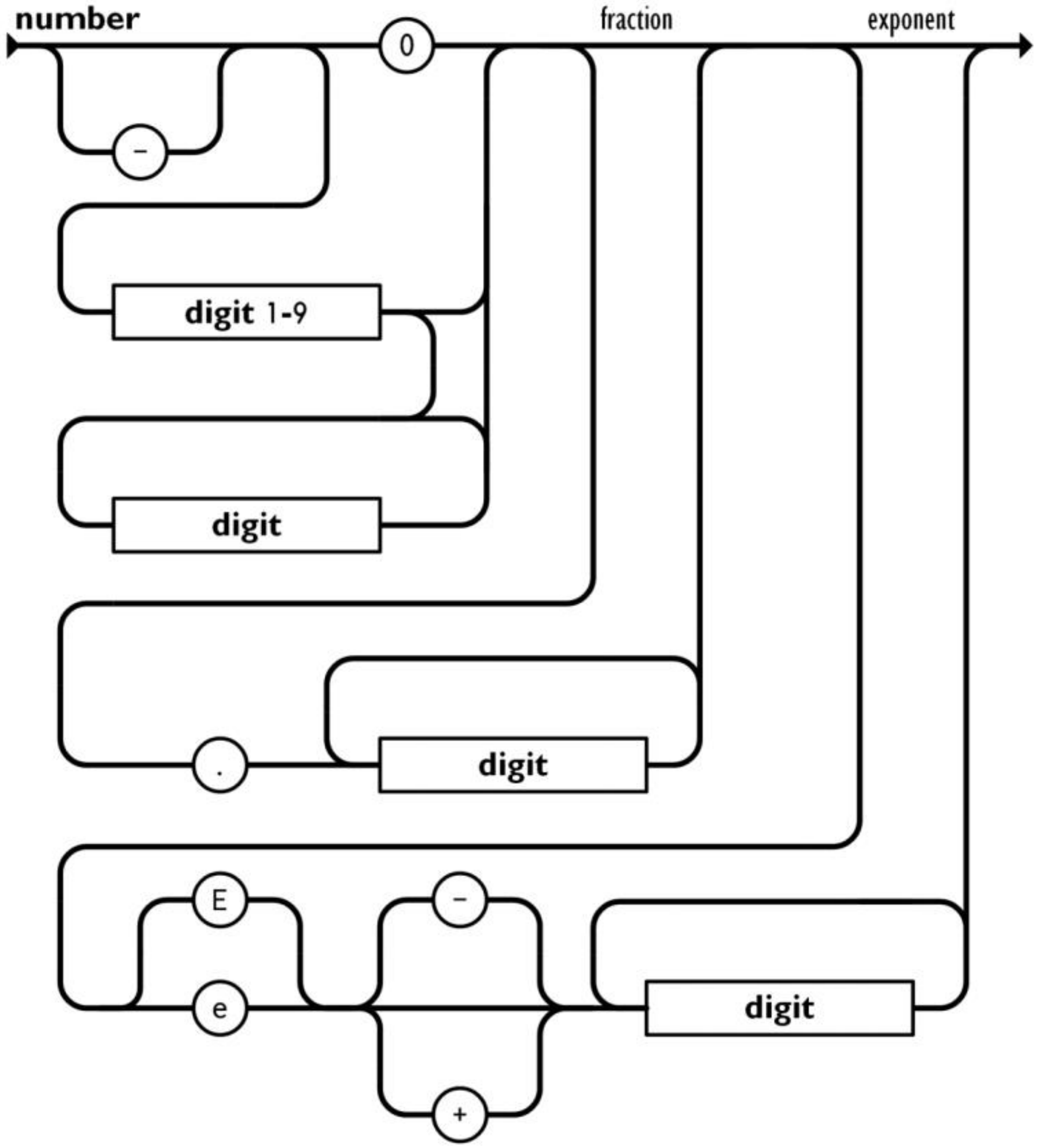
A value can be a string in double quotes, or a number, or true or false or null, or an object or an array. These structures can be nested.



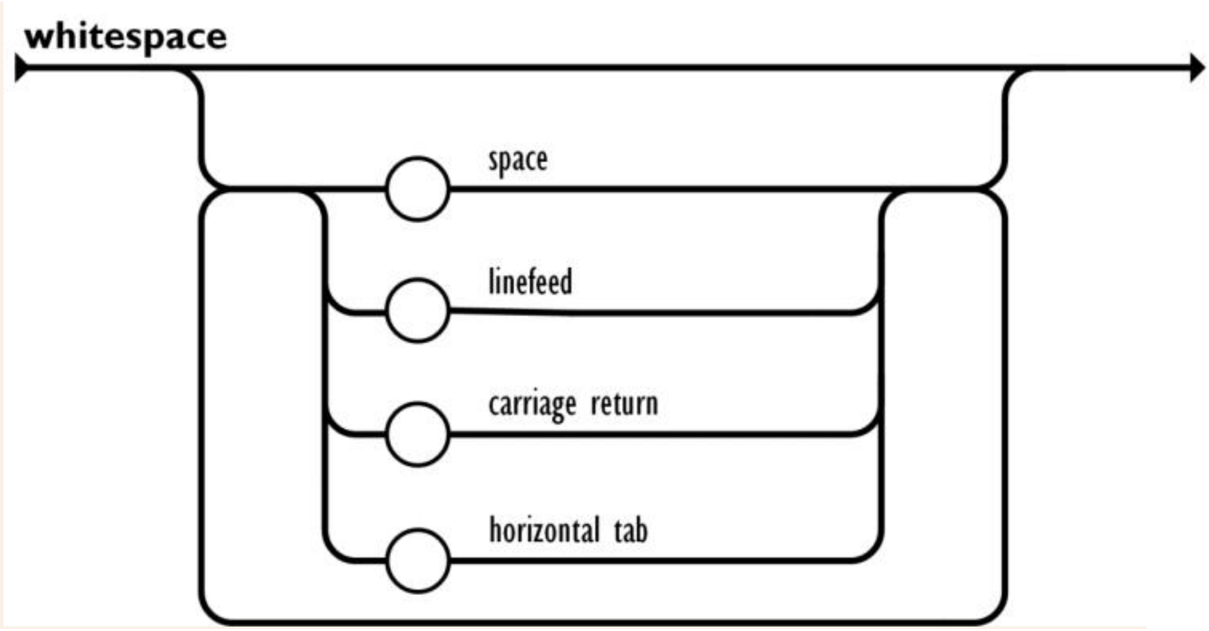
A string is a sequence of zero or more Unicode characters, wrapped in double quotes, using backslash escapes. A character is represented as a single character string. A string is very much like a C or Java string.



A number is very much like a C or Java number, except that the octal and hexadecimal formats are not used.



Whitespace can be inserted between any pair of tokens. Excepting a few encoding details, that completely describes the language.



Speech recognition –

Speech recognition, or speech-to-text, is the ability of a machine or program to identify words spoken aloud and convert them into readable text. Rudimentary speech recognition software has a limited vocabulary and may only identify words and phrases when spoken clearly. More sophisticated software can handle natural speech, different accents and various languages.

Speech recognition uses a broad array of research in computer science, linguistics and computer engineering. Many modern devices and text-focused programs have speech recognition functions in them to allow for easier or hands-free use of a device.

How does speech recognition work?

Speech recognition systems use computer algorithms to process and interpret spoken words and convert them into text. A software program turns the sound a microphone records into written language that computers and humans can understand, following these four steps:

- analyse the audio;
- break it into parts;
- digitize it into a computer-readable format; and
- use an algorithm to match it to the most suitable text representation.

Speech recognition software must adapt to the highly variable and context-specific nature of human speech. The software algorithms that process and organize audio into text are trained on different speech patterns, speaking styles, languages, dialects, accents and phrasings. The software also separates spoken audio from background noise that often accompanies the signal.

To meet these requirements, speech recognition systems use two types of models:

- **Acoustic models.** These represent the relationship between linguistic units of speech and audio signals.

- **Language models.** Here, sounds are matched with word sequences to distinguish between words that sound similar.

PyAudio –

PyAudio is a set of Python bindings for PortAudio, a cross- platform C++ library interfacing with audio drivers. PyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms. PyAudio is inspired by:

PortAudio/fast audio:

Python bindings for PortAudio v18 API.
tkSnack: cross-platform sound toolkit for Tcl/Tk and Python.
To use PyAudio, first instantiate PyAudio using PyAudio. PyAudio(), which sets up the PortAudio system.

To record or play audio, open a stream on the desired device with the desired audio parameters using pyaudio. PyAudio.open(). This sets up a pyaudio.Stream to play or record audio.

Play audio by writing audio data to the stream using pyaudio.Stream.write(), or read audio data from the stream using pyaudio.Stream.read().

Note that in "blocking mode", each pyaudio.Stream.write() or pyaudio.Stream.read() blocks until all the given/requested frames have been played/recorded. Alternatively, to generate audio data on the fly or immediately process recorded audio data, use the "callback mode" outlined below.

Use pyaudio.Stream.stop_stream() to pause playing/recording, and pyaudio.Stream.close() to terminate the stream.

Finally, terminate the portaudio session using pyaudio.PyAudio.terminate()

gTTS-

Google's text-to-speech packages converts your audio questions command to text. The response from the look-up function that you write for fetching answer to the question or command is converted in an audio form by gTTS. This package interface with Google Translate's API.

Datetime –

The datetime module supplies classes for manipulating dates and times.
While date and time arithmetic are supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation.

Datetime package is used to showing Date and Time. This datetime module comes with built in Python.

Wikipedia-

It is a multilingual free online encyclopaedia written and maintained by a community of volunteers through open collaboration and a wiki-based editing system. Individual contributors, also called editors, are known as Wikipedia's. Wikipedia is the largest and most-read reference work in history. It is consistently one of the 15 most popular websites ranked by Alexa; as of 2022, Wikipedia was ranked the 10th most popular site. It is hosted by the Wikimedia Foundation, an American non-profit organization funded mainly through donations.

On January 15, 2001, Jimmy Wales and Larry Sanger launched Wikipedia; Sanger coined its name as a portmanteau of "wiki" and "encyclopaedia." Wales was influenced by the "spontaneous order" ideas associated with Friedrich Hayek and the Austrian School of economics, after being exposed to these ideas by Austrian economist and Mises Institute Senior Fellow Mark Thornton. Initially available only in English, versions in other languages were quickly developed. Its combined editions comprise more than 58 million articles, attracting around 2 billion unique device visits per month and more than 17 million edits per month (1.9 edits per second) as of November 2020. In 2006, *Time* magazine stated that the policy of allowing anyone to edit had made Wikipedia the "biggest (and perhaps best) encyclopaedia in the world.

Wikipedia has received praise for its enablement of the democratization of knowledge, extent of coverage, unique structure, culture, and reduced amount of commercial bias, but criticism for exhibiting systemic bias, particularly gender bias against women and alleged ideological bias. Its reliability was frequently criticized in the 2000s but has improved over time; it has been generally praised in the late 2010s and early 2020s. Its coverage of controversial topics such as American politics and major events such as the COVID-19 pandemic has received substantial media attention. It has been censored by world governments, ranging from specific pages to the entire site. Nevertheless, it has become an element of popular culture, with references in books, films, and academic studies. In April 2018, Facebook and YouTube announced that they would help users detect fake news by suggesting fact-checking links to related Wikipedia articles

We all know Wikipedia is a great and huge source of knowledge just like GeeksforGeeks or any other sources we have used the Wikipedia module in our project to get more information from Wikipedia or to perform a Wikipedia search. To install this Wikipedia module, use pip install Wikipedia.

- **WebBrowser –**

The WebBrowser module provides a high-level interface to allow displaying web-based documents to users. Under most circumstances, simply calling the open() function from this module will do the right thing.

Under Unix, graphical browsers are preferred under X11, but text-mode browsers will be used if graphical browsers are not available or an X11 display isn't available. If text-mode browsers are used, the calling process will block until the user exits the browser.

If the environment variable BROWSER exists, it is interpreted as the os.pathsep-separated list of browsers to try ahead of the platform defaults. When the value of a list part contains the string %s, then it is interpreted as a literal browser command line to be used with the argument URL substituted for %s; if the part does not contain %s, it is simply interpreted as the name of the browser to launch.

For non-Unix platforms, or when a remote browser is available on Unix, the controlling process will not wait for the user to finish with the browser, but allow the remote browser to maintain its own windows on the display. If remote browsers are not available on Unix, the controlling process will launch a new browser and wait.

The script **WebBrowser** can be used as a command-line interface for the module. It accepts a URL as the argument. It accepts the following optional parameters: -n opens the URL in a new browser window, if possible; -t opens the URL in a new browser page ("tab"). The options are, naturally, mutually exclusive. Usage example:

```
python -m webbrowser -t "https://www.python.org"
```

OS –

The OS module in Python provides functions for interacting with the os. OS comes under Python’s standard utility modules. This module provides a way of using operating system dependent functionality. 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.

You first need to import the os module to interact with the underlying operating system. So, import it using the import os statement before using its functions.

❖ Getting Current Working Directory

The getcwd() function confirms returns the current working directory.

Example: Get Current Working Directory

```
>>> import os

>>> os.getcwd()

'C:\\Python37'
```

Pyjokes –

Pyjokes is used for collection Jokes over the Internet. Pyjokes is add in our project because it adds jokes in our project. It is very interesting Pyjokes is the one-line joke which makes our project interesting.

Installation

You can simply install it using pip with the following command:

pip install Pyjokes.

Usage

Now to use it, we need to import the installed library in our python Script using the following command:

import pyjokes

Smtplib-

The simple mail transfer protocol library is a Python library for sending emails using the Simple Mail Transfer Protocol (SMTP). The smtplib is a built-in module in python; do not need to install it. It abstracts all the complexities of SMTP away. It provides a Simple Mail Transfer Protocol (SMTP) client implementation.

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).

class smtplib.**SMTP**(*host=''*, *port=0*, *local_hostname=None*, [*timeout*,]*source_address=None*)

An SMTP instance encapsulates an SMTP connection. It has methods that support a full repertoire of SMTP and ESMTP operations. If the optional host and port parameters are given, the SMTP connect() method is called with those parameters during initialization. If specified, *local_hostname* is used as the FQDN of the local host in the HELO/EHLO command. Otherwise, the local hostname is found using socket.getfqdn(). If the connect() call returns anything other than a success code, an SMTPConnectError is raised. The optional *timeout* parameter specifies a timeout in seconds for blocking operations like the connection attempt (if not specified, the global default timeout setting will be used). If the timeout expires, TimeoutError is raised. The optional source_address parameter allows binding to some specific source address in a machine with multiple network interfaces, and/or to some specific source TCP port. It takes a 2-tuple (host, port), for the socket to bind to as its source address before connecting. If omitted (or if host or port are '' and/or 0 respectively) the OS default behavior will be used.

For normal use, you should only require the initialization/connect, sendmail(), and SMTP.quit() methods. An example is included below.

The SMTP class supports the with statement. When used like this, the SMTP QUIT command is issued automatically when the with statement exits. E.g.:

```
>>>
>>> from smtplib import SMTP
>>> with SMTP("domain.org") as smtp:
...     smtp.noop()
...
(250, b'Ok')
>>>
```

Requests- Requests module allows you to send http requests using python. It is used for making GET and POST requests. It abstracts the complexities of making requests behind a beautiful, simple API.

Requests allows you to send HTTP/1.1 requests extremely easily. There’s no need to manually add query strings to your URLs, or to form-encode your POST data. Keep-alive and HTTP connection pooling are 100% automatic, thanks to urllib3.

FEATURES of Requests:

- Requests is ready for today’s web.
- Keep-Alive & Connection Pooling

- International Domains and URLs
- Sessions with Cookie Persistence
- Browser-style SSL Verification
- Automatic Content Decoding
- Basic/Digest Authentication
- Elegant Key/Value Cookies
- Automatic Decompression
- Unicode Response Bodies
- HTTP(S) Proxy Support
- Multipart File Uploads
- Streaming Downloads
- Connection Timeouts
- Chunked Requests
- .netrc Support
- Requests officially supports Python 3.7+, and runs great on PyPy.

Implementation Work Details

a desktop assistant is a voice assistant that can perform many daily tasks of desktop like playing music, opening your favourite IDE with the help of a single voice command. Our AI 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.

CHAPTER-5

METHODOLOGY

An application of a natural language processing-based intelligent voice assistant system that may be used to transmit text messages. Messages may be sent and received, and the device's other applications can be used on the platform of was designed and implemented utilizing python android.

- **System Architecture**

The System consists of the following stages:

- Collection of data as voice or speech.
- Analysis and conversion of the data into text.
- Storing and processing of the data.
- Generation of speech from the processed data.
- Task Automation on the voiced demand.

The speech and voice samples are gathered as data and saved as input to the following level in the first stage. The data is processed and then turned to text in the second stage. The text is then inspected once it has been converted, and a Python script is used to analyse and identify the command's answer. The output is created in the form of voice from the basic text in the next stage, based on the identified answer.

The aforesaid procedure is then automated utilizing voice commands in the last stage. TTS may be used for everything from basic content to conversational changes.

The following contains the list of features the proposed system will provide:

- It continuously listens for its name and will activate as a response after calling.

- It continues learning the arrangement of inquiries posed to it identified with its setting which it recollects for what's to come. So, when a similar setting is referenced, it begins a discussion with you posing similar inquiries.

- It can perform arithmetic calculations upon specific voice commands and can give back the response through voice.

- It can search the internet based on the user’s commands and reply with either interactive questions or definitive answers through voice.

- The cloud server to update is a firebase cloud server.

- It can automatically connect to a cloud server through auto synchronization and can keep itself updated.

- The architecture based on the Internet of Things can help the user to connect the smart devices.

- Other highlights, for example, playing music, setting a caution, checking climate states of gadget's area. Setting updates, spell-right, and so forth can be performed by a contribution from the client's voice.

➢ **Methodology for voice email capability**

The voice mail capability is enabled and is the key feature to the assistant. Initially if mail command is enabled to assistant by the user the assistant process the requisites of the mail like receiver mail-id, subject and message body of the mail all these requisites are taken as input through voice and it is taken accurately.

Firstly, step is initialized with auto credentials which are fed with assistant automatically so its auto composed. The modules involved in this design are predefined and open-source modules and they are HTTPs requests module, smtplib module which is accompanied with email MIME Handlers which is very useful to holding attachments and also it makes us to take the files from working directory of PC to email through python operating program workspace.

The smtplib module characterizes a SMTP customer session object that can be utilized to send mail to any Internet machine with a SMTP or ESMTP audience daemon. SMTP represents Simple Mail Transfer Protocol. The smtplib modules is valuable for speaking with mail servers to send letters. Sending letters is finished with Python's smtplib utilizing a SMTP server.

Genuine use differs relying upon unpredictability of the email and settings of the email server, the directions here depend on sending email through Gmail. The other two noteworthy fragments of the pack are the "parser" and the "generator". The parser takes the serialized adjustment of an email message (a surge of bytes) and changes over it into a tree of Email Message objects.

The generator takes an Email Message and changes it over into a serialized byte stream. (The parser and generator also handle surges of substance characters, anyway this utilization is incapacitated as it is too easy to even consider evening consider winding up with messages that are not authentic in some way or another.)

The email pack advances a valiant exertion to cover the nuances of the distinctive administering RFCs from the application. Sensibly the application should have the alternative to treat the email message as a composed tree of Unicode substance and twofold associations, without obsessing about how these are addressed when serialized. Before long, nevertheless, it is much of the time essential to think about likely a segment of the standards overseeing MIME messages and their structure, expressly the names and nature of the MIME "content sorts" and how they recognize multipart reports. For the most part this data should simply be required for progressively complex applications, and still, by the day's end it should simply be the huge level structure being alluded to, and not the nuances of how those structures are addressed.

Since MIME substance types are used comprehensively in present day web programming (not just email), this will be a characteristic plan to various programming engineers.

❖ REAL LIFE APPLICATION

➢ Saves time: Our AI is a desktop voice assistant which works on the voice command offered to it, it can do voice searching, voice-activated device control and can let us complete a set of tasks.

➢ Conversational interaction: It makes it easier to complete any task as it automatically does it by using the essential module or libraries of Python, in a 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.

➢ Reactive nature: 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.

- Multitasking: The main application of it can be its multitasking ability. It can ask for continuous instruction one after other until the user “QUIT” it.

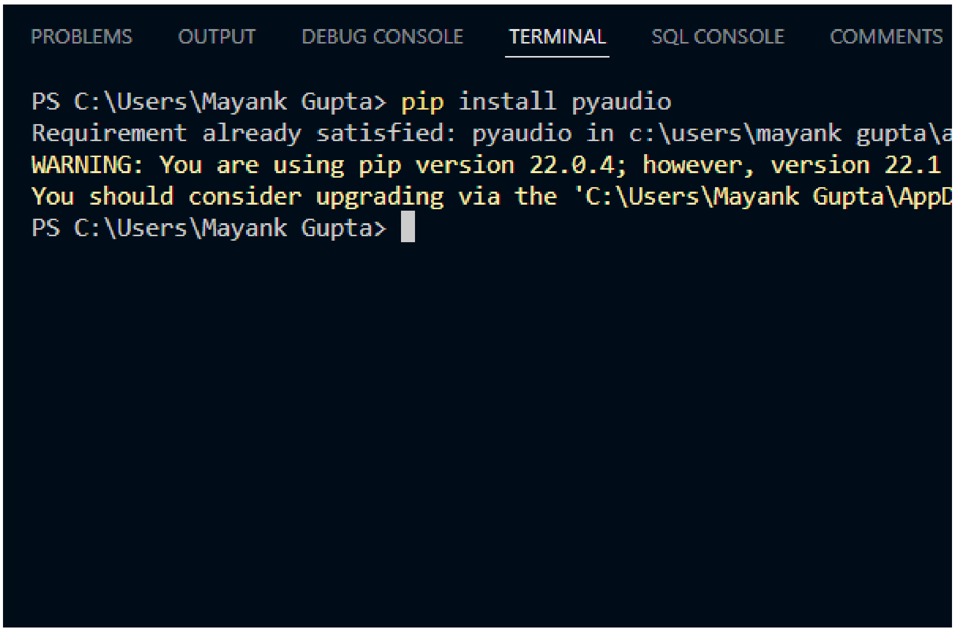
- No Trigger phase: It asks for the instruction and listen the response that is given by user without needing any trigger phase and then only executes the task a desktop assistant is a voice assistant that can perform many daily tasks of desktop like playing music, opening your favourite IDE with the help of a single voice command. Our AI 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.

CHAPTER 6:

Result Analysis and Validations

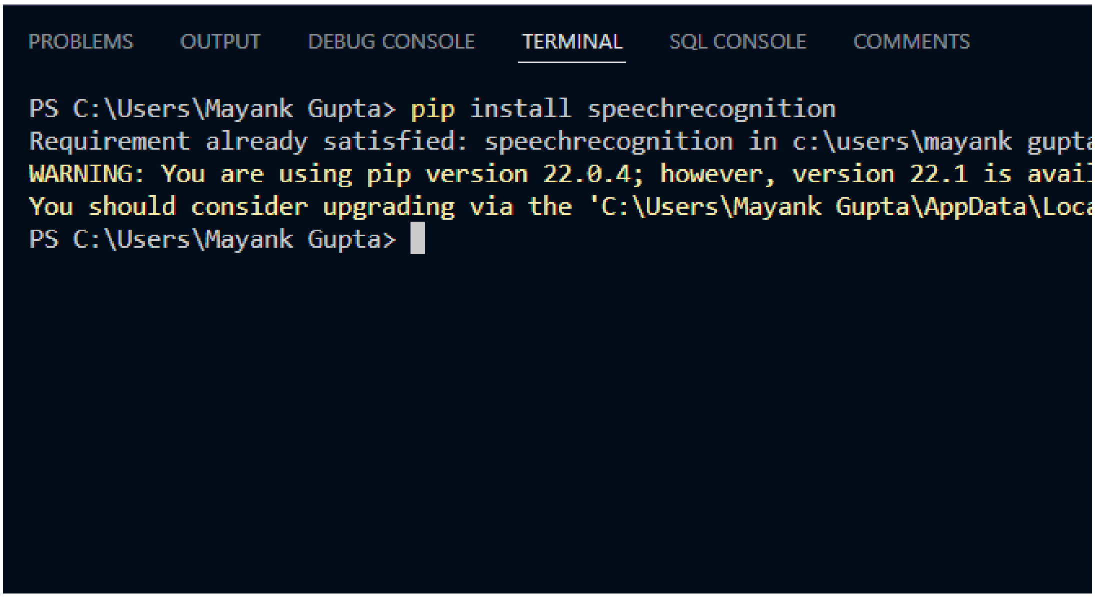
1. Installations of Modules:
   (All modules have been already installed)

PyAudio:



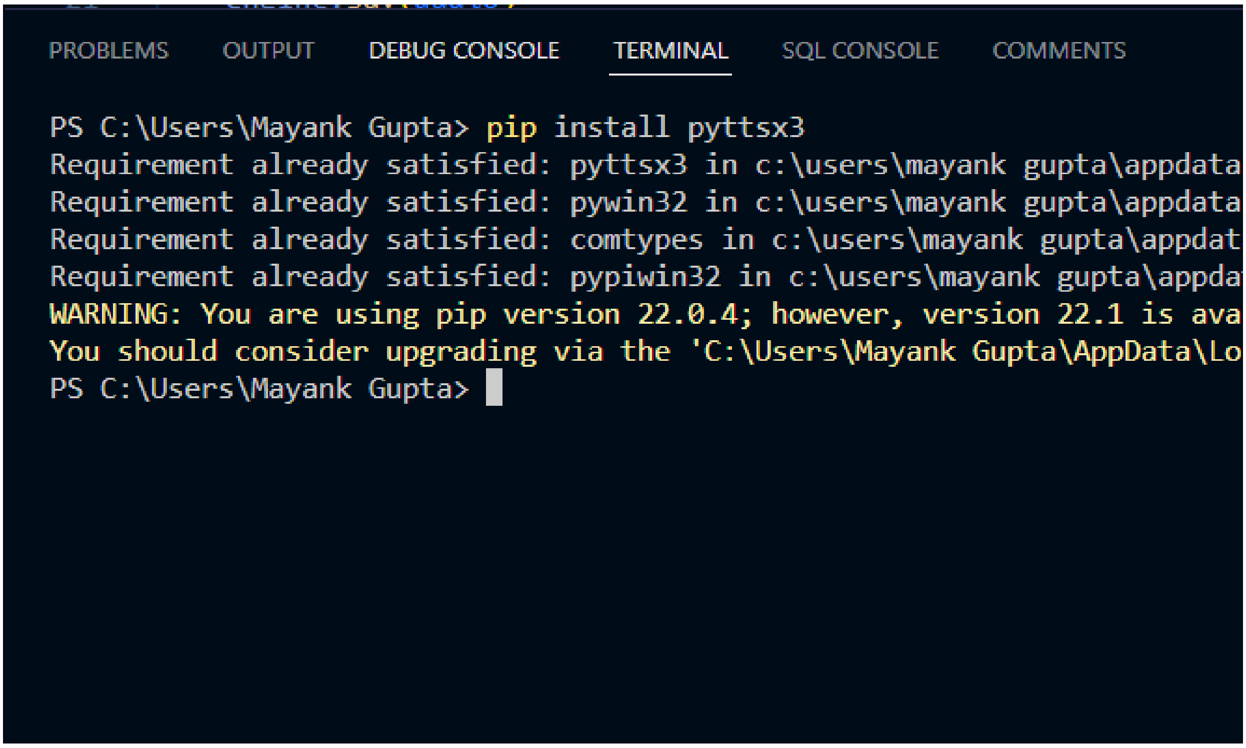
```
PS C:\Users\Mayank Gupta> pip install pyaudio
Requirement already satisfied: pyaudio in c:\users\mayank gupta\a
WARNING: You are using pip version 22.0.4; however, version 22.1
You should consider upgrading via the 'C:\Users\Mayank Gupta\AppD
PS C:\Users\Mayank Gupta>
```

Speech Recognition:



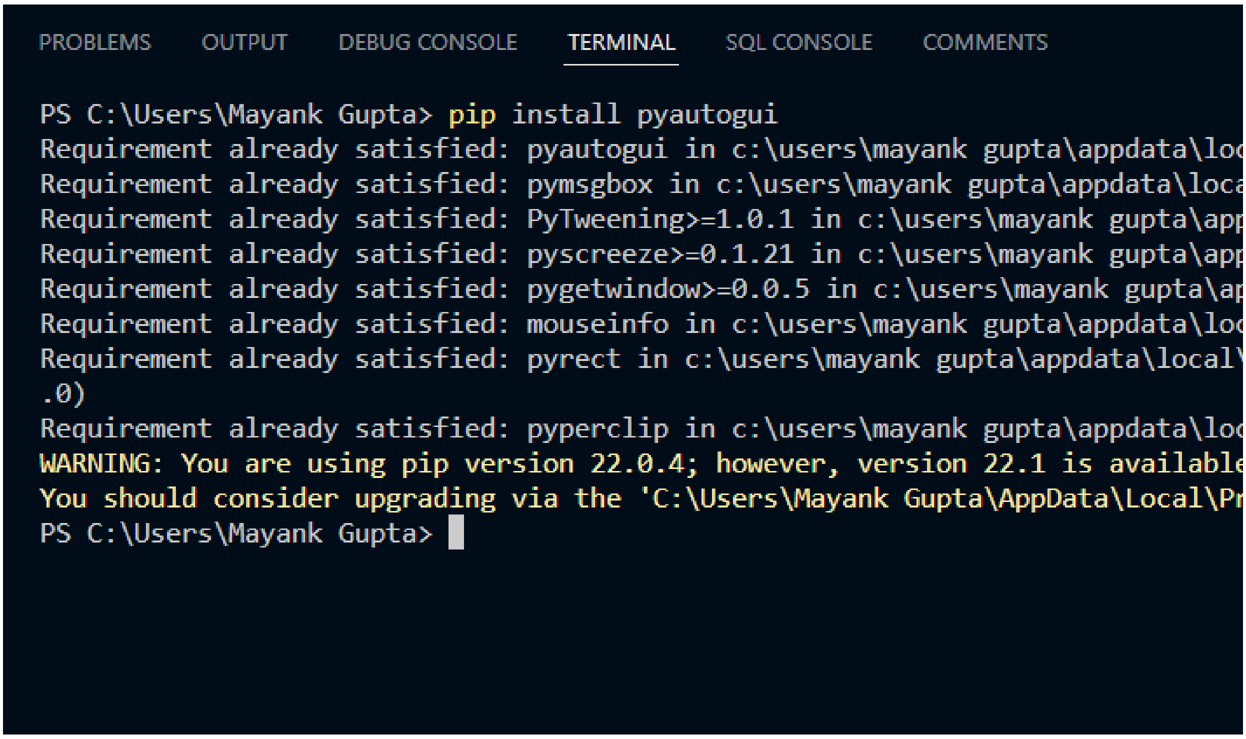
```
PS C:\Users\Mayank Gupta> pip install speechrecognition
Requirement already satisfied: speechrecognition in c:\users\mayank gupta
WARNING: You are using pip version 22.0.4; however, version 22.1 is avail
You should consider upgrading via the 'C:\Users\Mayank Gupta\AppData\Loca
PS C:\Users\Mayank Gupta>
```

Pyttsx3:



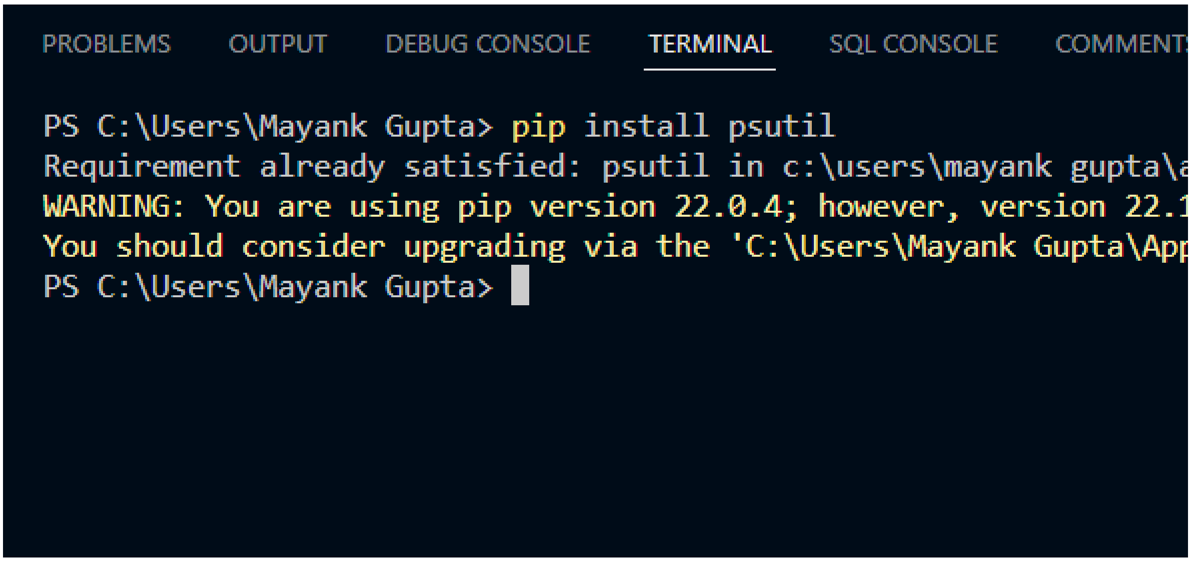
```
PS C:\Users\Mayank Gupta> pip install pyttsx3
Requirement already satisfied: pyttsx3 in c:\users\mayank gupta\appdata
Requirement already satisfied: pywin32 in c:\users\mayank gupta\appdata
Requirement already satisfied: comtypes in c:\users\mayank gupta\appdat
Requirement already satisfied: pypiwin32 in c:\users\mayank gupta\appda
WARNING: You are using pip version 22.0.4; however, version 22.1 is ava
You should consider upgrading via the 'C:\Users\Mayank Gupta\AppData\Lo
PS C:\Users\Mayank Gupta> 
```

PyautoGui:



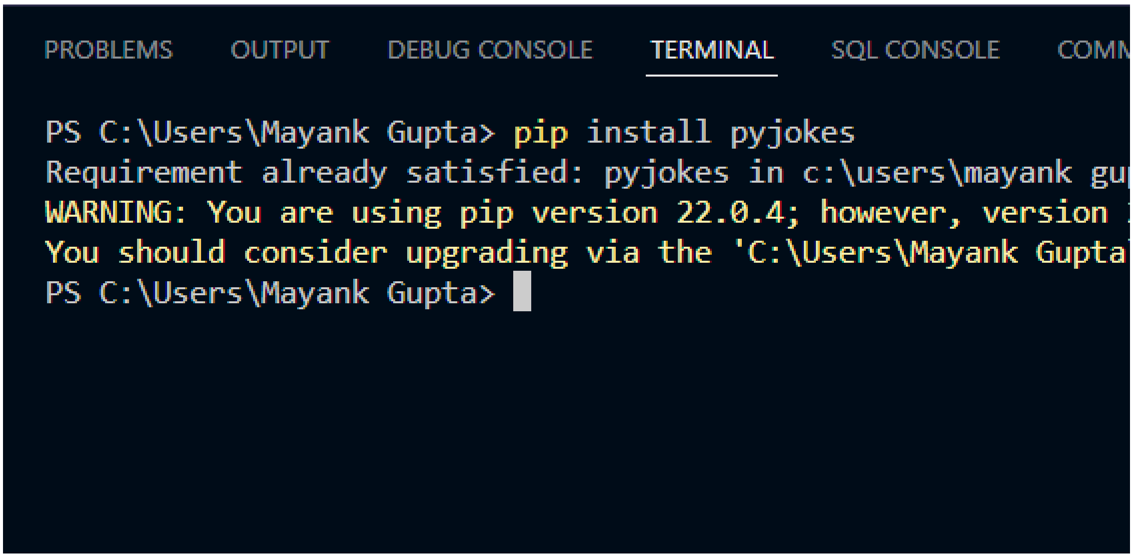
```
PS C:\Users\Mayank Gupta> pip install pyautogui
Requirement already satisfied: pyautogui in c:\users\mayank gupta\appdata\loc
Requirement already satisfied: pymsgbox in c:\users\mayank gupta\appdata\loca
Requirement already satisfied: PyTweening>=1.0.1 in c:\users\mayank gupta\app
Requirement already satisfied: pyscreeze>=0.1.21 in c:\users\mayank gupta\app
Requirement already satisfied: pygetwindow>=0.0.5 in c:\users\mayank gupta\ap
Requirement already satisfied: mouseinfo in c:\users\mayank gupta\appdata\loc
Requirement already satisfied: pyrect in c:\users\mayank gupta\appdata\local\
.0)
Requirement already satisfied: pyperclip in c:\users\mayank gupta\appdata\loc
WARNING: You are using pip version 22.0.4; however, version 22.1 is available
You should consider upgrading via the 'C:\Users\Mayank Gupta\AppData\Local\Pr
PS C:\Users\Mayank Gupta> 
```

- psutil:



```
PS C:\Users\Mayank Gupta> pip install psutil
Requirement already satisfied: psutil in c:\users\mayank gupta\a
WARNING: You are using pip version 22.0.4; however, version 22.1
You should consider upgrading via the 'C:\Users\Mayank Gupta\App
PS C:\Users\Mayank Gupta> █
```

- Pyjokes:



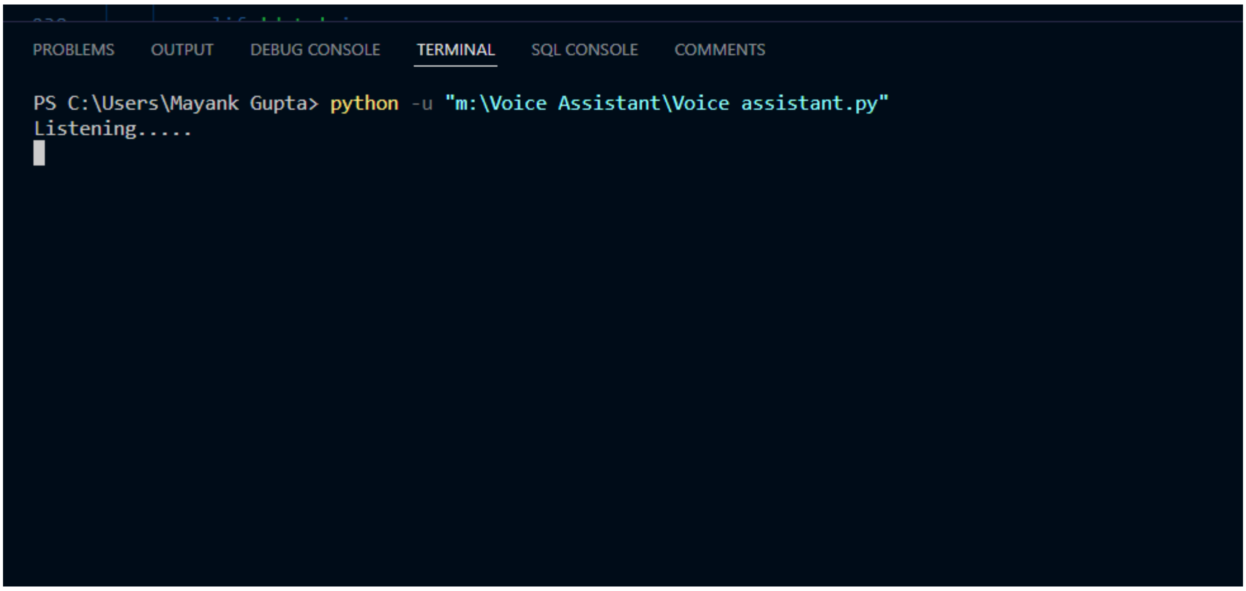
```
PS C:\Users\Mayank Gupta> pip install pyjokes
Requirement already satisfied: pyjokes in c:\users\mayank gu
WARNING: You are using pip version 22.0.4; however, version
You should consider upgrading via the 'C:\Users\Mayank Gupta
PS C:\Users\Mayank Gupta> █
```

And many other libraries or modules from your choice.

Function: pip install {Library name}

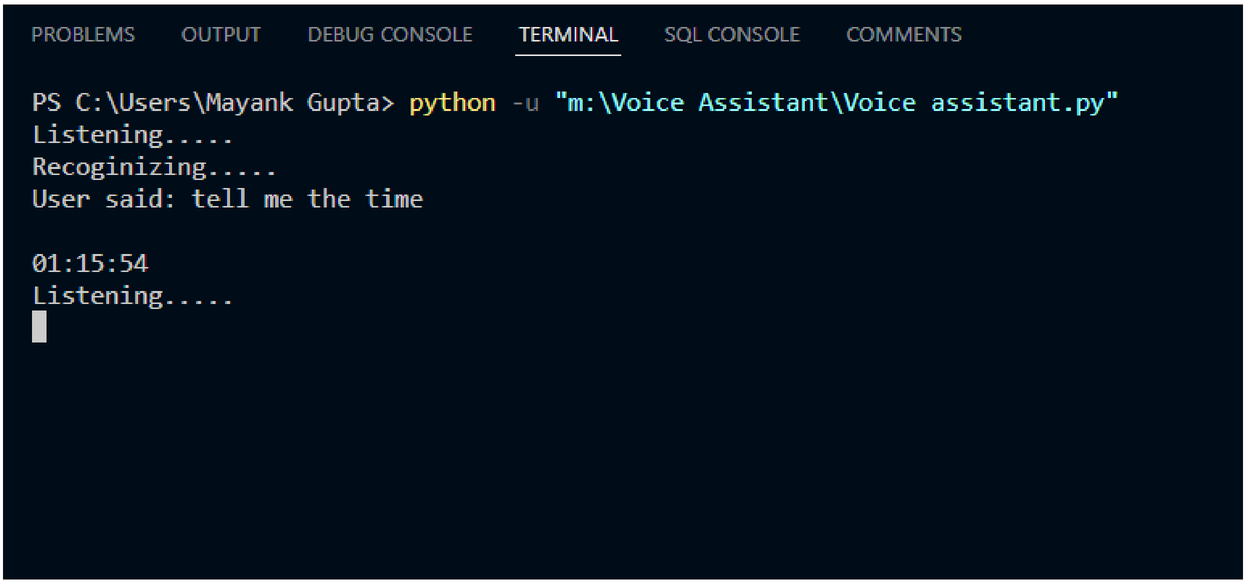
2. Input/Output of the Codes:

- When we run the code, it will greet us according to the time and start listening us, then recognize what we said.



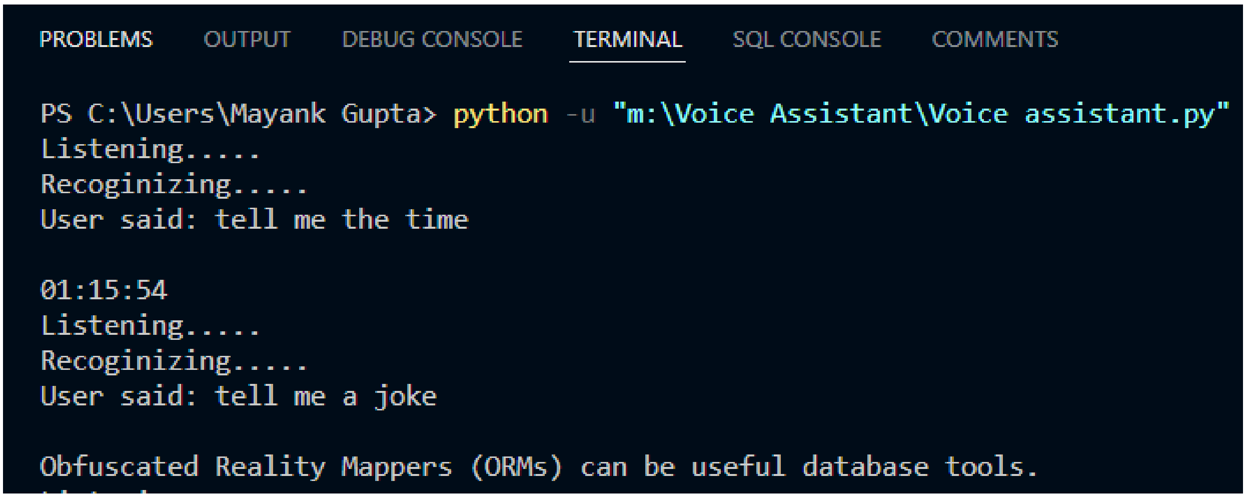
```
PS C:\Users\Mayank Gupta> python -u "m:\Voice Assistant\Voice assistant.py"
Listening.....
```

- Asking time:



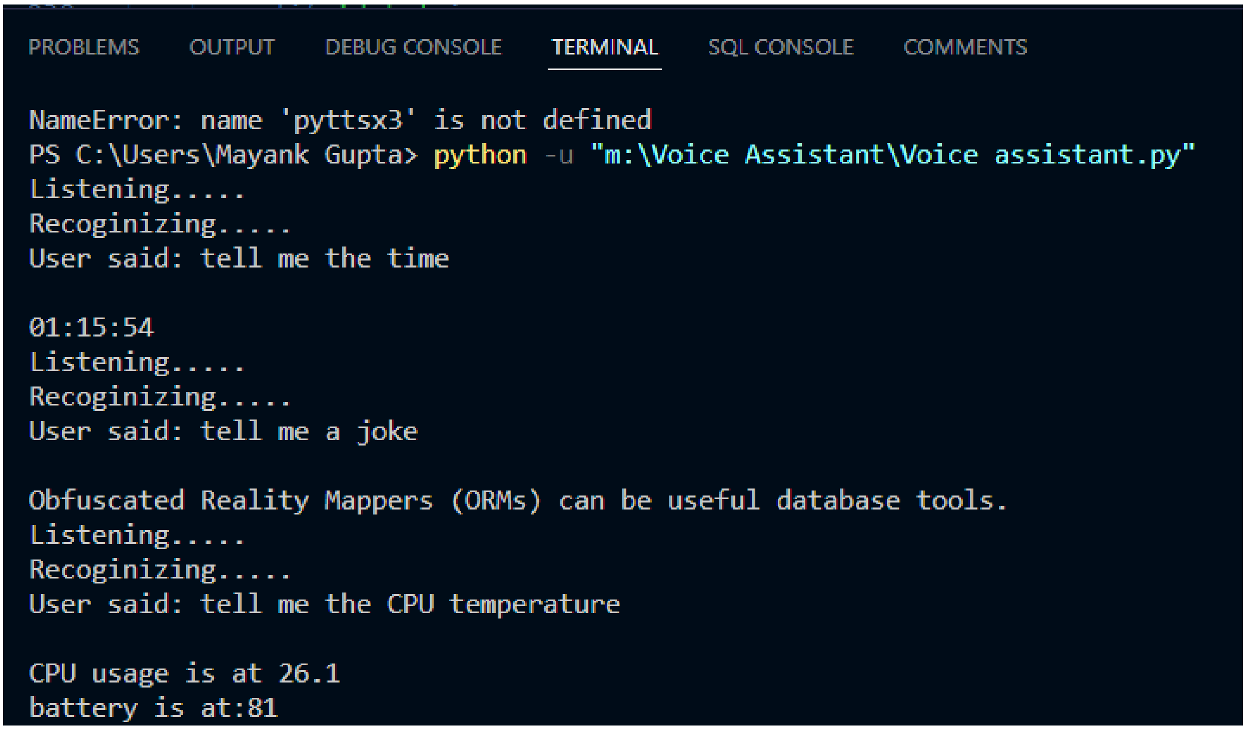
```
PS C:\Users\Mayank Gupta> python -u "m:\Voice Assistant\Voice assistant.py"
Listening.....
Recoginizing.....
User said: tell me the time

01:15:54
Listening.....
```

- Asking Jokes:



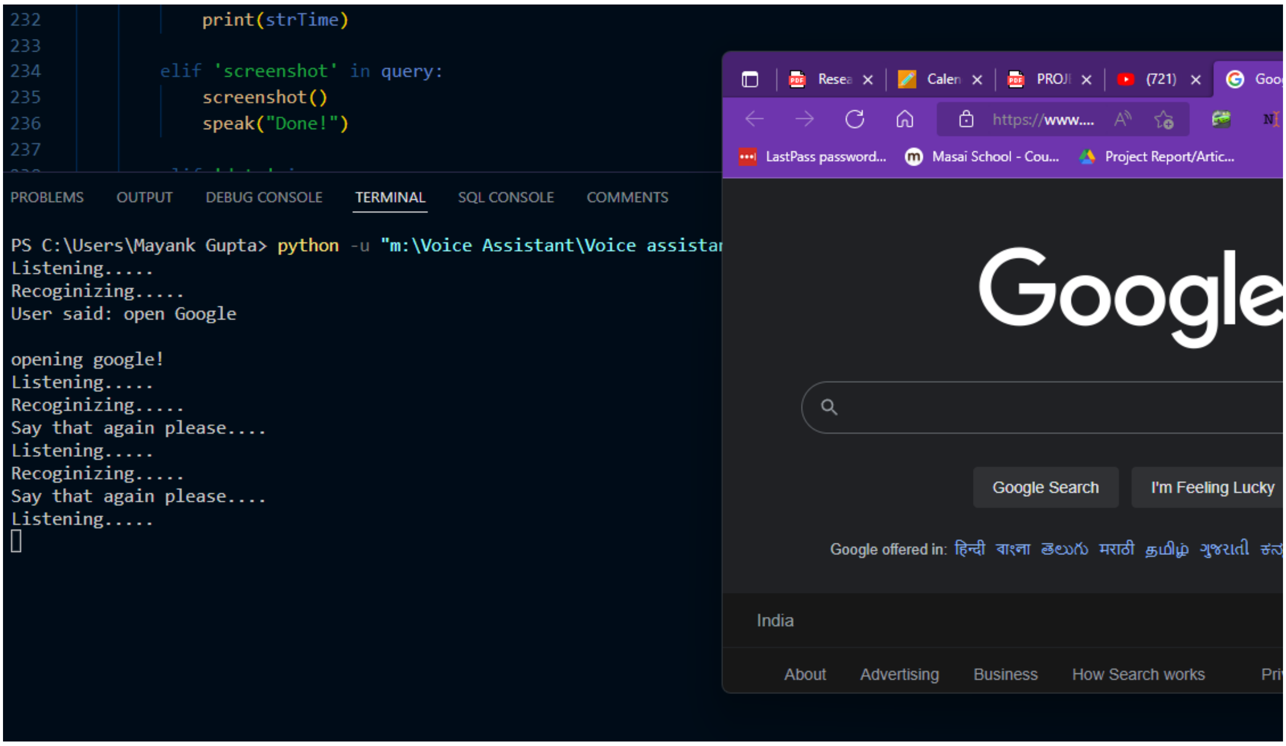
```
PS C:\Users\Mayank Gupta> python -u "m:\Voice Assistant\Voice assistant.py"
Listening.....
Recoginizing.....
User said: tell me the time

01:15:54
Listening.....
Recoginizing.....
User said: tell me a joke

Obfuscated Reality Mappers (ORMs) can be useful database tools.
```

- Asking about CPU temperatures and battery percentage:



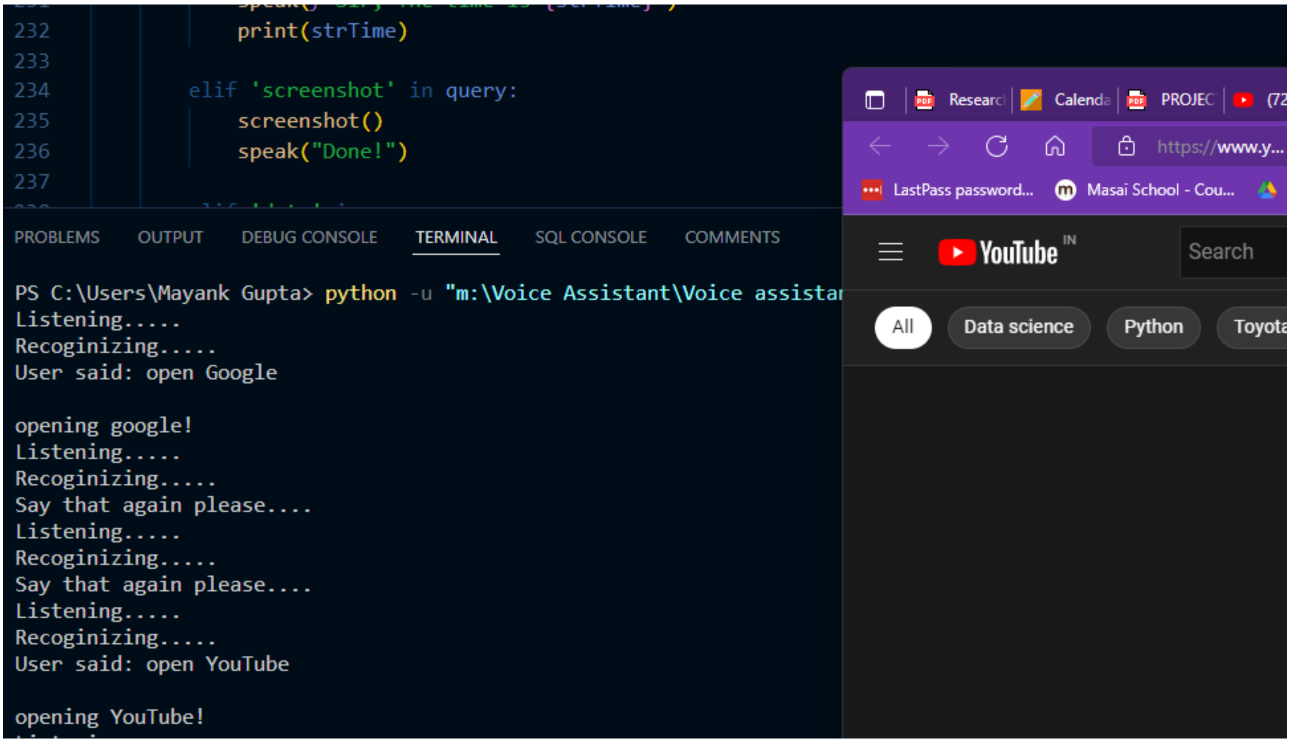
```
NameError: name 'pyttsx3' is not defined
PS C:\Users\Mayank Gupta> python -u "m:\Voice Assistant\Voice assistant.py"
Listening.....
Recoginizing.....
User said: tell me the time

01:15:54
Listening.....
Recoginizing.....
User said: tell me a joke

Obfuscated Reality Mappers (ORMs) can be useful database tools.
Listening.....
Recoginizing.....
User said: tell me the CPU temperature

CPU usage is at 26.1
battery is at:81
```

Asking to open Chrome, YouTube, etc:



```
            print(strTime)

        elif 'screenshot' in query:
            screenshot()
            speak("Done!")
```

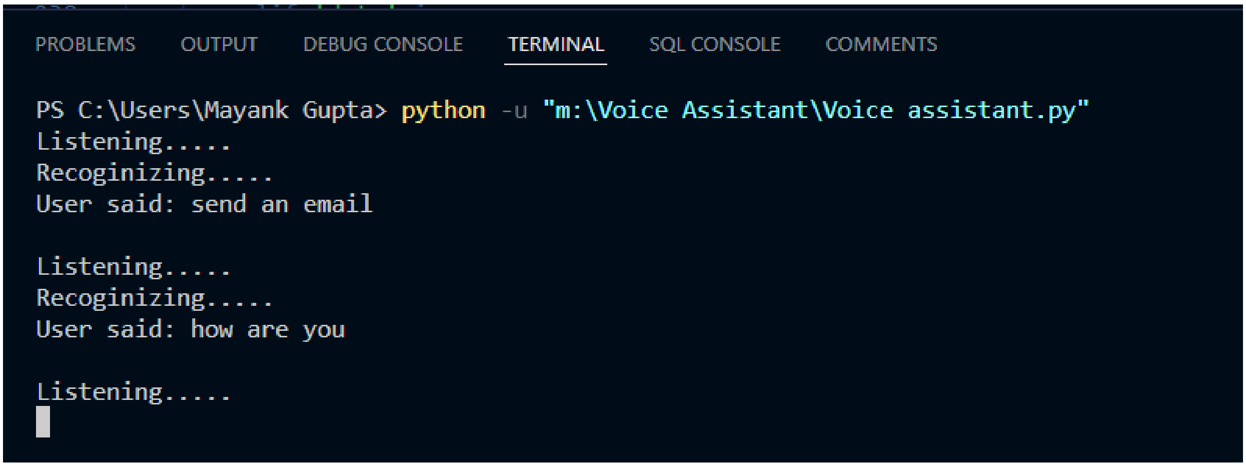
```
PS C:\Users\Mayank Gupta> python -u "m:\Voice Assistant\Voice assistar
Listening.....
Recoginizing.....
User said: open Google

opening google!
Listening.....
Recoginizing.....
Say that again please....
Listening.....
Recoginizing.....
Say that again please....
Listening.....
```



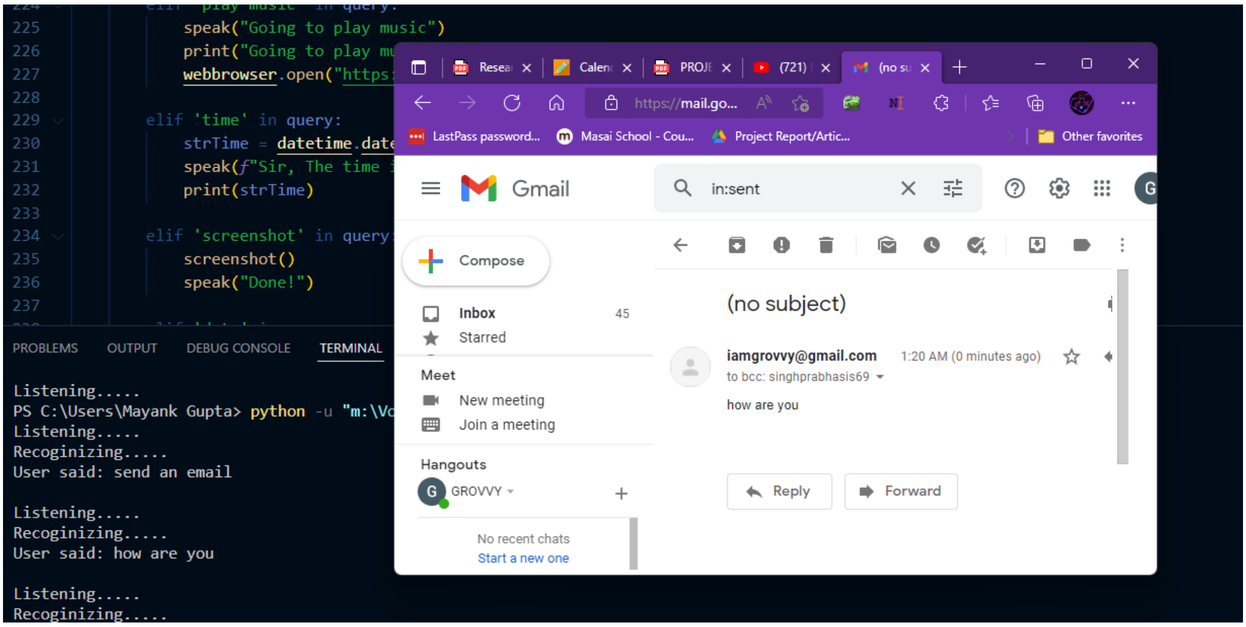
```
            print(strTime)

        elif 'screenshot' in query:
            screenshot()
            speak("Done!")
```

```
PS C:\Users\Mayank Gupta> python -u "m:\Voice Assistant\Voice assistar
Listening.....
Recoginizing.....
User said: open Google

opening google!
Listening.....
Recoginizing.....
Say that again please....
Listening.....
Recoginizing.....
Say that again please....
Listening.....
Recoginizing.....
User said: open YouTube

opening YouTube!
```

Sending Emails:



```
PS C:\Users\Mayank Gupta> python -u "m:\Voice Assistant\Voice assistant.py"
Listening.....
Recoginizing.....
User said: send an email

Listening.....
Recoginizing.....
User said: how are you

Listening.....
```



```
            speak("Going to play music")
            print("Going to play mu
            webbrowser.open("https:

        elif 'time' in query:
            strTime = datetime.date
            speak(f"Sir, The time i
            print(strTime)

        elif 'screenshot' in query:
            screenshot()
            speak("Done!")
```

```
Listening.....
PS C:\Users\Mayank Gupta> python -u "m:\Vo
Listening.....
Recoginizing.....
User said: send an email

Listening.....
Recoginizing.....
User said: how are you

Listening.....
Recoginizing.....
```

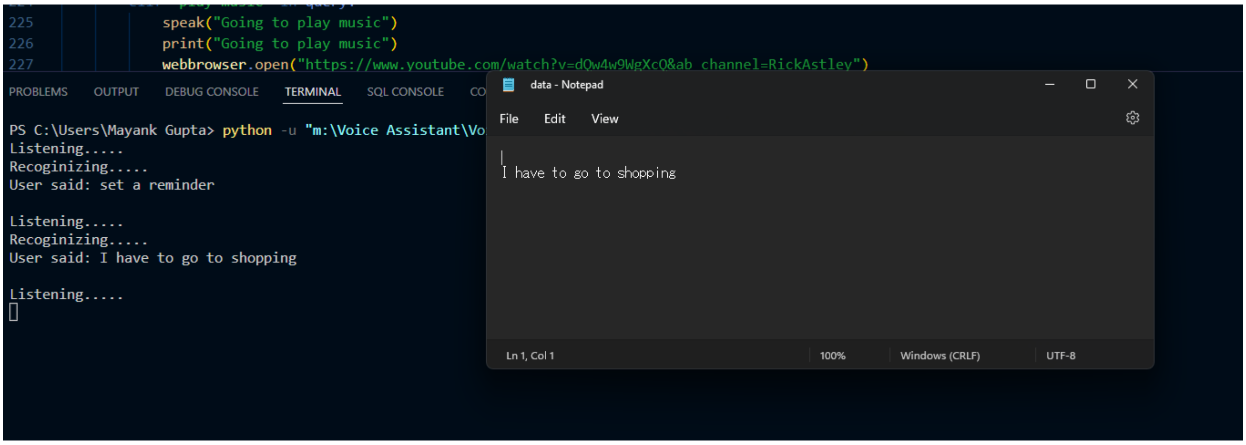
(no subject)

iamgrovvy@gmail.com 1:20 AM (0 minutes ago)

to bcc: singhprabhasis69

how are you

Adding something in reminder list:

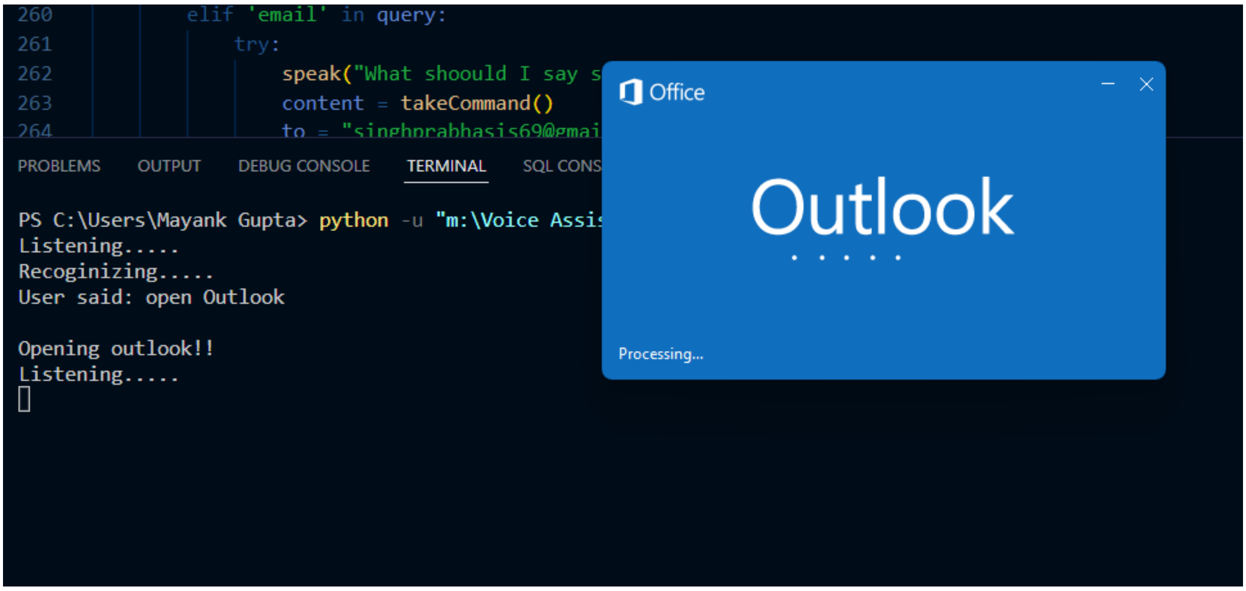


```
            speak("Going to play music")
            print("Going to play music")
            webbrowser.open("https://www.youtube.com/watch?v=dQw4w9WgXcQ&ab_channel=RickAstley")
```

```
PS C:\Users\Mayank Gupta> python -u "m:\Voice Assistant\Vo
Listening.....
Recoginizing.....
User said: set a reminder

Listening.....
Recoginizing.....
User said: I have to go to shopping

Listening.....
```

I have to go to shopping

- Open certain Applications:



```
        elif 'email' in query:
            try:
                speak("What shoould I say s
                content = takeCommand()
                to = "singhprabhasis69@gmai
```

```
PS C:\Users\Mayank Gupta> python -u "m:\Voice Assis
Listening.....
Recoginizing.....
User said: open Outlook

Opening outlook!!
Listening.....
```

It has many other functions too, like playing music or weather forecasting etc.

CHAPTER 7

System testing

The system testing is done on fully integrated system to check whether the requirements are matching or not. The system testing for our AI desktop assistant focuses on the following four parameters:

- 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 our AI can search on Google or not user said "Open Google", then our asked," What should I search on Google?" then user said, "What is Python", Our AI open Google and searched for the required input

- 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.

- SECURITY

The security testing mainly focuses on vulnerabilities and risks. As Our AI 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.

- 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.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

The desktop assistant's efficiency has been greatly increased in terms of functionality, and it also monitors the user's security using voice recognition technologies.

It's called advanced desktop assistant since it includes a lot of sophisticated capabilities that Microsoft Cortana doesn't have.

With significant features, this assistant is also focused on the user interface, which is necessary because Cortana lacks a user-friendly interface. A new desktop assistant is required to provide next-generation services, allowing us to make desktops as user-friendly as mobile devices.

The Contact's Application is used to store the client's contact details and also encourages the customer to make a call or send an SMS to other people individual who is using the chemical stored in this app.

- **<u>LIMITATIONS</u>:**
    - Security is somewhere an issue, there is no voice command encryption in this project.
    - Background voice can interfere
    - Misinterpretation because of accents and may cause inaccurate results.
    - Our AI cannot be called externally anytime like other traditional assistants like Google Assistant can be called just by saying, "Ok Google!"

- **FUTURE SCOPE:**

Virtual assistants are now available and are quick and responsive, but there is still a long way to go. The present systems' understanding and dependability must be improved. It has vastly improved.

Today's aides are still insufficient. In critical situations, it's dependable. These helpers' future will be bright. Artificial Intelligence (AI) should be implemented into virtual assistants. Machine Learning and Neural Networks are examples of intelligence. IoT and networks, for example.

With the addition of these, we shall be able to reach new heights thanks to technological advancements. What really is the situation?

What virtual assistants can accomplish is far beyond what we now have, achieved up until now Our AI is a virtual reality that most of us have seen.

REFERENCES

1. Website referred:

- www.stackoverflow.com
- www.pythonprogramming.net
- www.codecademy.com
- www.tutorialspoint.com

2. Books Referred:

- Python Programming -Kiran Gurbani
- Learning Python – Mark Lutz
- Hands-on Machine Learning with Scikit-Learn (O'Reilly)

3. Youtube Channels Referred:

- Code with Harry
- Edureka!
- Programming with Mosh

4. Documents Referred:

C.K. Gomathy. (2010),"Cloud Computing: Business Management for Effective Service Oriented Architecture" International Journal of Power Control Signal and Computation (IJPCSC), Volume 1, Issue IV,

Mohit Bansal, Dr. T. K. Thivakaran, "Analysis of Speech Recognition using Convolutional Neural Network", Journal of Engineering Sciences, Vol 11, Issue 1, 2020, Page 285-291.

C K Gomathy and V Geetha. Article: A Real Time Analysis of Service Based Using Mobile Phone Controlled Vehicle Using DTMF For Accident Prevention. International Journal of Computer Applications 138(2):11-13, March 2016. Published By Foundation of Computer Science (FCS), NY, USA, ISSN No: 0975-8887

Ossama Abdel-Hamid, Abdelrahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu, “Convolutional Neural Networks for Speech Recognition”, IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol.22,2010.

Dr. Jaydeep Patil, Atharva Shewale, Ekta Bhushan, Alister Fernandes, Rucha Khartadkar, "A Voice Based Assistant Using Google Dialogflow and Machine Learning", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011,

Nil Goksel-Canbek2 Mehmet EminMutlu, “On the track of Artificial Intelligence: Learning with Intelligent Personal Assistant” International Journal of Human Sciences, 13(1), 592-601. doi:10.14687/ijhs.v13i1.3549.