

ABSTRACT

A Voice Assistant is one of the hot topics in the current world that are programs that listens to human's verbal command and respond to them which makes it a human computer/device interaction. In the current days, a voice assistant is everywhere which is a lot useful in these busy days. Nowadays, almost everyone in the current world is using voice assistant because it's everywhere starting from Google smartphone assistant which even 5 years old kids will know how to use because of the current world pandemic which makes them use smartphones till Amazon's Alexa which will be very useful to do works starting from entertaining the users. A virtual assistant is a technology based on artificial intelligence. This software uses a device's microphone to receive voice request while the voice output takes place at the speaker . but most exciting thing happens between this two actions .It is a combination of several different technologies : voice recognition , voice analysis and language processing It is completely developed using one of the most powerful language python. Voice assistants are ingenious programs that effortlessly respond to your verbal commands on digital devices. They revolutionize the way we interact with technology, making tasks as simple as asking for the weather forecast or opening a YouTube. With just spoken words, these digital companions spring into action catering to our needs in a remarkably intuitive manner. Voice assistant, such as Siri and google assistant, operate on a straightforward principle that blends advanced technology with user-friendly interfaces. These ingenious applications harness the power of cutting-edge tools, such as Python's Speech Recognition and Python Text-to -speech library pyttsx3, to comprehend and process vocal input effortlessly.

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ABSTRACT	i
	LIST OF FIGURES	iv
1.	INTRODUCTION	1
1.1.	OVERVIEW	2
1.2.	DESIGN	3
1.3.	VOICE ASSISTANT	3
1.3.1	WHAT IS VOICE ASSISTANT	3
1.3.2	WHY DO WE NEED IT	3
1.3.3	WHERE TO USE	4
2.	LITERATURE SURVEY	
2.1.	RELATED WORK	5-8
3.	METHODOLOGY	9
3.1.	EXISTING SYSTEM	10
3.2.	PROPOSED SYSTEM	10
3.3.	OBJECTIVE OF THE PROJECT	10
3.4.	SOFTWARE AND HARDWARE REQUIREMENTS	11
3.4.1.	SOFTWARE REQUIREMENTS	11
3.4.2.	HARDWARE REQUIREMENTS	11
3.4.3.	LIBRARIES	12-14
3.5.	PROGRAMMING LANGUAGES	15
3.5.1.	PYTHON	15
3.5.2.	DOMAIN	16
3.6.	SYSTEM ARCHITECTURE	17
3.7.	ALGORITHMS USED	

3.7.1. SPEECH RECOGNITION MODULE	17
3.7.2. SPEECH TO TEXT & TEXT TO SPEECH CONVERSION	17
3.7.3. PROCESS & EXECUTES THE REQUIRED COMMAND	18
3.8. SYSTEM DESIGN	19
3.8.1 USE CASE DIAGRAM	20
3.8.2 COMPONENT DIAGRAM	20
3.8.3 SEQUENCE DIAGRAM	21
3.9. FEASIBILITY STUDY	22
3.10. TYPES OF OPERATION	23-25
4. RESULTS AND DISCUSSION	26
WORKING	26
APPENDICES	31
SOURCE CODE	31-38
SCREENSHOTS	39-41
CONCLUSION	42
4.1 CONCLUSION	42
4.1 FUTURE WORK	43
REFERENCES	44

LIST OF FIGURES

Figure No.	Figure Name	Page No.
3.1	SYSTEM ARCHITECTURE	17
3.2	USE CASE DIAGRAM	19
3.3	COMPONENT DIAGRAM	20
3.4	SEQUENCE DIAGRAM	21
3.5	SEQUENCE DIAGRAM (Answering the user)	22
4.1	FLOWCHART	28

CHAPTER 1

INTRODUCTION

The very first voice activated product was released in 1922 as Radio Rex. This toy was very simple, wherein a toy dog would stay inside a dog house until the user exclaimed its name,

“Rex” at which point it would jump out of the house. This was all done by an electromagnet tuned to the frequency similar to the vowel found in the word Rex, and predated modern computers by over 20 years.

In the 21st century, human interaction is being replaced by automation very quickly. One of the main reasons for this change is performance. There’s a drastic change in technology rather than advancement. In today’s world, we train our machines to do their tasks by themselves or to think like humans using technologies like Machine Learning, Neural Networks, etc. Now in the current era, we can talk to our machines with the help of virtual assistants.

Virtual assistants are software programs that help you ease your day to day tasks, such as showing weather reports, giving daily news, searching the internet etc. They can take commands by voice. Voice-based intelligent assistants need an invoking word or wake word to activate the listener, followed by the command. We have so many virtual assistants, such as

Apple’s Siri, Amazon’s Alexa and Microsoft’s Cortana and Amazon's Alexa and this has been an inspiration for us to do this as a project. This system is designed to be used efficiently on desktops. Voice assistants are programs on digital devices that listen and respond to verbal commands. A user can say, “What's the weather?” and the voice assistant will answer with the weather report for that day and location.

1.1 OVERVIEW

voice assistants are software systems designed to integrate seamlessly into various devices. They possess the capability to understand and execute commands given through voice. By listening for specific keywords and filtering out background noise, they retrieve relevant information or carry out requested actions. While voice assistants can be found in a wide range of devices, such as cell phones, computers, and smart speakers, there are also assistants tailored specifically for individual device applications. To overcome this, digital signal processing techniques like Feature Extraction and Feature Matching have been introduced to represent the voice signal effectively. In this project, we utilize a speech engine that leverages the technique of Feature Extraction. By extracting key features from the voice signal, we can simplify the complex information contained within it. These features serve as vital building blocks for creating various functionalities that assist humans in their daily lives. By reducing the efforts required in different tasks, these functionalities aim to enhance our overall productivity and convenience. Our primary goal is to develop a range of functionalities that directly benefit individuals in their day-to-day routines. Whether it's simplifying mundane tasks or providing valuable assistance, we strive to make human lives easier. By harnessing the power of speech technology and feature extraction techniques, we can create innovative solutions that address the diverse needs of users. These functionalities have the potential to revolutionize various industries and significantly impact the way we interact with technology.

1.2 DESIGN

- a) The voice assistant takes an input word which is called as "signal word" to be activated. so, it takes in the signal word and starts operating for the user commands.
- b) Converting the speech into text will be processed by the assistant.

- c) The converted text is now processed to get the required results.
- d) The text given by the user should contain one or two keywords that determine what query is to be executed. If the keyword doesn't match any of the queries in the code then the assistant asks the user to speak again.
- e) Finally, the output to the user's query will be given by converting speech to text.

1.3 VOICE ASSISTANT

Our assistant "NOVA" extends to helps us when working on a system in which it is installed. We can access by calling the wake word "Hello NOVA".

1.3.1 WHAT IS VOICE ASSISTANT

A voice assistant, also known as an intelligent personal assistant or a connected speaker, is a new type of device that is based on natural language speech recognition and is offered by popular companies like Apple, Amazon, and Google. We got inspired by that and created one our self.

1.3.2 WHY DO WE NEED IT

Usually, typing out and searching or doing day-to-day tasks becomes hectic. But our life doesn't need to be like that. One can ask for help to voice assistants. They let the users to perform a task using a speech command, as well as retrieve information via voice synthesis.

Following are the reasons to have a voice assistant.

- Minimal Effort
 - It's easier to say a few words than type them on a small smartphone screen.
- Eyes Free
 - One can be as blind as a bat, but a voice assistant will always help you. Our ears are enough. One can also ask the bot about something while cooking at the same time.

- Fast response
- Imagine how much time you have to spend to find some information on a website? Or how many clicks do you need to make before you find the thing you need in a mobile application? Voice assistants don't generate such difficulties. One can ask a question and you have the answer.

1.3.3 WHERE TO USE IT

Voice search has been a hot topic of discussion. Voice visibility will undoubtedly be a challenge. This is due to the lack of a visual interface for voice assistants. Users cannot see or interact with a voice interface unless it is linked to the Alexa or Google Assistant app. Search behavior patterns will change dramatically as a result.

Brands are currently undergoing a transformation in which touchpoints are transforming into listening points, and organic search will be the primary means of brand visibility. Advertising agencies are becoming more popular as voice search grows in popularity. Voice assistants will also continue to offer more individualized experiences as they get better at differentiating between voices. The number of people using voice assistants is expected to grow. According to the Voice bot Smart Speaker Consumer Adoption Report 2018, almost ten percent of people who do not own a smart speaker plan to purchase one. If this holds true, the user base of smart speaker users will grow 50 percent, meaning a quarter of adults in the United States will own a smart speaker.

CHAPTER 2

LITERATURE SURVEY

2.1 RELATED WORK

This field of virtual assistants having speech recognition has seen some major advancements or innovations. This is mainly because of its demand in devices like smartwatches or fitness bands, speakers, Bluetooth earphones, mobile phones, laptop or desktop, television, etc. Almost all the digital devices which are coming nowadays are coming with voice assistants which help to control the device with speech recognition only. A new set of techniques is being developed constantly to improve the performance of voice automated search.

As the amount of data is increasing exponentially now known as Big Data the best way to improve the results of virtual assistants is to incorporate our assistants with machine learning and train our devices according to their uses. Other major techniques that are equally important are Artificial Intelligence, Internet of Things, Big Data access and management, etc. With the use of voice assistants, we can automate the task easily, just give the input to the machine in the speech form and all the tasks will be done by it from converting your speech into text form to taking out keywords from that text and execute the query to give results to the user.

Machine Learning is just a subset of Artificial Intelligence. This has been one of the most helpful advancements in technology. Before AI we were the ones who were upgrading technology to do a task but now the machine is itself able to counter new tasks and solve it without need to involve the humans to evolve it.

This has been helpful in day-to-day lifestyle. From mobile phones to personal desktops to mechanical industries these assistants are in very much demand for automating tasks and increasing efficiency.

- **Nivedita Singh** (2021) et al. proposed a voice assistant using python speech to text (STT) module and had performed some api calls and system calls which has led to developing a voice assistant using python which allows the user to run any type of command through voice without interaction of keyboard. This can also run on hybrid platforms. Therefore, this paper lacks in some parts like the system calls that aren't much supported.
- **Abeed Sayyed** (2021) et al. presented a paper on Desktop Assistant AI using python with IOT features and also used Artificial Intelligence (AI) features along with a SQLite DB with the use of Python. This Project has a Database connection and a query framework but lacks API call and System calls features.
- **P.Krishnaraj** (2021) et al. presented a project on Portable Voice Recognition with GUI Automation, This system uses Google's online speech recognition system for converting speech input to text along with Python. Therefore, this project has a GUI and is also has a portable framework. Accuracy of this text to speech (TTS) engine is comparatively less and also lacks IoT.
- **Rajdip Paul** (2021) et al. presented a project named A Novel Python-based Voice Assistance System for reducing the Hardware Dependency of Modern Age Physical Servers. This Author has proposed assistant project with python as a backend supporting system calls, api calls and various features. This Project is quite well responsive with api calls, also needs improvement in understanding and reliability.
- **V. Geetha** (2021) et al. presented a project named The Voice Enabled Personal Assistant for Pc using Python. This Author has proposed assistant project with python

as a backend and features like turning our PC off, or restarting it, or reciting some latest news, are just one voice command away. Also, this project has well supported library not every API will have the capability to convert the raw JSON data into text. And there is a delay in processing request calls.

- **Dilawar Shah Zwakman** (2021) et al. proposed the Usability Evaluation of Artificial Intelligence-Based Voice Assistants which can give proper response to the user's request. It also has a feature where it can make an appointment with the person mention by the user through voice but it lacks API calls.

- **Dimitrios Buhalis** (2021) et al. proposed a paper on In-room Voice-Based AI Digital Assistants

Transforming On-Site Hotel Services and Guests' Experiences. Where voice assistant is used for hotel services. It'll be very useful in this current COVID-19 era. Human Touch is considered as a danger in this COVID time and with a voice assistant, loss of human touch is not considered as an advantage. It can also be used to control the temperature controls and room light controls but it needs Complex Integration and Staff Training.

- **Philipp Sprengholz** (2021) et al. has proposed Ok Google: Using virtual assistants for data collection in psychological and behavioural research which is a survey mate that they have developed which is an extension of the Google Assistant that was used to check the reliability and validity of data collected by this test. Possible answers and synonyms are defined for every different type of questions so, it can be used to analyse the behaviour of an individual. As it is a psychological and behavioural research assistant.

- **Rahul Kumar** (2020) et al. has proposed Power Efficient Smart Home with voice assistant by which we can say that a Voice Assistant is one of the important part of the Smart home which is becoming one of the major things in the current world as it can operate the Home Appliances just with voice which also increase the home security because of the smart locks but it requires a reliable internet connection which is crucial and sometimes, the user might lock themselves out of their own house.

- **Benedict D. C** (2020) et al. proposed Consumer decisions with artificially intelligent voice assistants that will have stronger psychological reactions to the system's look on human like behaviours. The assistant has an IoT (Internet of Things) features. It can also order stuffs which the user want but there are some cons in this paper. Voice assistant relies on the speaker's ability to represent the decision alternatives to catch up in voice dialogues and another main disadvantage is that, it lacks system calls.
- **Tae-Kook Kim** (2020) el at. has proposed a Short Research on Voice Control System Based on Artificial Intelligence Assistant which states AI assistant system using open API artificial intelligence, and the conditional auto-run system, IFTTT (IF This, Then That). It can control the system using the Raspberry PI board but it lacks system calls.

CHAPTER 3

METHODOLOGY

3.1 EXISTING SYSTEM

voice assistant have become an integral part of our lives. These intelligent digital companions utilize human-like voice technology, language processing algorithm, and synthesis to cater to our needs. Whether it's finding information or performing specific functions, voice assistant are there to heed our commands. Our voice assistant is equipped with a plethora of functionalities that aim to simplify daily tasks. From copying and pasting files between locations to sending messages to a user's mobile device, our voice assistant can do it all. Furthermore, it can even help with ordering pizza or purchasing a new mobile phone, all through simple voice commands..

3.2 PROPOSED SYSTEM

We are proposing a system in an efficient way of implementing a Personal voice assistant, Speech Recognition library has many in-built functions, that will let the assistant understand the command given by user and the response will be sent back to user in voice, with Text to Speech functions. When assistant captures the voice command given by user, the under lying algorithms will convert the voice into text. And according to the keywords present in the text (command given by user), respective action will be performed by the assistant.

This is made possible with the functions present in different libraries. Also, the assistant was able to achieve all the functionalities with help of some API's. We had used these APIs for functionalities like performing calculations, extracting news from web sources, and for telling the weather. We will be sending a request, and through the API, we're getting the respective output. API's like WOLFRAMALPHA, are very helpful in performing things like calculations, making small web searches. And for getting the data from web. In this way, we are able to extract news from the web sources, and send them as input to a function for further purposes. Also, we have libraries like Random and many other libraries, each corresponding to a different technology. We used the library OS to implement Operating System related functionalities like Shutting down a system, or restarting a system.

At the outset we make our program capable of using system voice with the help of sapi5 and pyttsx3. pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3. The Speech Application Programming Interface or SAPI is an API developed by Microsoft to allow the use of speech recognition and speech synthesis within Windows applications. Then we define the speak function to enable the program to speak the outputs.

After that we will define a function to take voice commands using the system microphone. The main function is then defined where all the capabilities of the program are defined.

□ The proposed system will have the following functionality:

(a) The system will keep listening for commands and the time for listening is variable which can be changed according to user requirements.

(b) If the system is not able to gather information from the user input it will keep asking again to repeat till the desired number of times.

(c) The system can have both male and female voices according to user requirements.

(d) Features supported in the current version include playing music, texts, search on Wikipedia, or opening system installed applications, opening anything on the web browser, etc.

3.3 OBJECTIVE OF PROJECT

Main objective of building personal assistant software (a virtual assistant) is using semantic data sources available on the web, user generated content and providing knowledge from knowledge databases. The main purpose of an intelligent virtual assistant is to answer questions that users may have. This may be done in a business environment, for example, on the business website, with a chat interface. On the mobile platform, the intelligent virtual assistant is available as a call-button operated service where a voice asks the user “What can I do for you?” and then

responds to verbal input. Virtual assistants can tremendously save you time. We spend hours in online research and then making the report in our terms of understanding.

Provide a topic for research and continue with your tasks while the assistant does the research. Another difficult task is to remember test dates, birthdates or anniversaries. It comes with a surprise when you enter the class and realize it is class test today. Just tell assistant in advance about your tests and she reminds you well in advance so you can prepare for the test. One of the main advantages of voice searches is their rapidity. In fact, 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 during the same period of time. In this respect, the ability of personal assistants to accurately recognize spoken words is a prerequisite for them to be adopted by consumers.

3.4 SOFTWARE AND HARDWARE REQUIREMENTS

3.4.1 Software Requirements:

- ✓ Python 3.5 & Above
- ✓ Windows 7 And Above

3.4.2 Hardware Requirements:

- ✓ Processor: Intel Core i5
- ✓ RAM: 4GB
- ✓ OS: Windows / Mac
- ✓ Relay & Microphone

- ✓ **Pytttsx3**- It is a text to speech conversion library in python which is used to convert the text given in the parenthesis to speech. It is compatible with python 2 and 3. An application invokes the `pytttsx3.init()` factory function to get a reference to a `pytttsx3`. it is a very easy to use tool which converts the entered text into speech. The `pytttsx3` module supports two voices first is female and the second is male which is provided by “sapi5” for windows. Command to install: - `pip install pytttsx3` It supports three TTS engines: - sapi5- To run on windows nsss - NSSpeechSynthesizer on Mac OS X espeak – eSpeak on every other platform

- ✓ **Speech_recognition**- It allows computers to understand human language. Speech recognition is a machine's ability to listen to spoken words and identify them. We can then use speech recognition in Python to convert the spoken words into text, make a query or give a reply. Python supports many speech recognition engines and APIs, including Google Speech Engine, Google Cloud Speech API.

Command to install :- `pip install SpeechRecognition`

- ✓ **WolframAlpha**- Wolfram Alpha is an API which can compute expert-level answers using Wolfram's algorithms, knowledgebase and AI technology. It is made possible by the Wolfram Language. The WolframAlpha API provide a web-based API allowing the computational and presentation capabilities of WolframAlpha to be integrated into web, mobile and desktop applications. Command to install :- `pip install wolframalpha`

- ✓ **Randfacts**- Randfacts is a python library that generates random facts. We can use `randfacts.get_fact()` to return a random fun fact. Command to install :- `pip install randfacts`

- ✓ **Pyjokes**- Pyjokes is a python library that is used to create one-line jokes for the users. Informally, it can also be referred as a fun python library which is pretty simple to use.

Command to install :- `pip install pyjokes`

- ✓ **Datetime**- This module is used to get the date and time for the user. This is a built-in module so there is no need to install this module externally. Python Datetime module supplies classes to work with date and time. Date and datetime are an object in Python, so when we manipulate them, we are actually manipulating objects and not string or timestamps.
- ✓ **Random2**- Python version 2 has a module named "random". This module provides a Python 3 ported version of Python 2.7's random module. It has also been back-ported to work in Python 2.6. In Python 3, the implementation of randrange() was changed, so that even with the same seed you get different sequences in Python 2 and 3.
- ✓ **Math**- This is a built-in module which is used to perform mathematical tasks. For example, math.cos() which returns the cosine of a number or math.log() returns the natural logarithm of a number, or the logarithm of number to base.
- ✓ **Warnings**- The warning module is actually a subclass of Exception which is a built-in class in Python. A warning in a program is distinct from an error. Conversely, a warning is not critical. It shows some message, but the program runs.
- ✓ **OS**- The os module is a built-in module which provides functions with which the user can interact with the os when they are running the program. This module provides a portable way of using operating system-dependent functionality. This module has functions with which the user can open the file which is mentioned in the program.
- ✓ **Serial**- This module encapsulates the access for the serial port. It provides backends for Python running on Windows, OSX, Linux, BSD and Iron Python. The module named "serial" automatically selects the appropriate backend.
 Command to install :- pip install pyserial
- ✓ **Time**- This module provides many ways of representing time in code, such as objects, numbers, and strings. It also provides functionality other than representing time, like waiting during code execution and measuring the efficiency of our code.

This is a built-in module so the installation is not necessary.

- ✓ **Wikipedia** :-This is a Python library that makes it easy to access and parse data from Wikipedia. Search Wikipedia, get article summaries, get data like links and images from a page, and more. Wikipedia is a multilingual online encyclopedia. Command to install :- pip install wikipedia

- ✓ **Selenium Webdrive**- The selenium module is used to automate web browser interaction from Python. Several browsers/drivers are supported (Firefox, Chrome, Internet Explorer), as well as the Remote protocol. The supported python versions are python 3.5 and above.

Command to install :- pip install selenium

- ✓ **Requests**- The requests module allows you to send HTTP requests using Python. The HTTP request returns a Response Object with all the response data. With it, we can add content like headers, form data, multipart files, and parameters via simple Python libraries. It also allows you to access the response data of Python in the same way.

Command to install :- pip install requests

- ✓ **Webbrowser**- 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. It accepts a URL as the argument with the following optional parameters: -n opens the URL in a new browser window, if possible, and -t opens the URL in a new browser tab. This is a built-in module so installation is not required.

3.5. PROGRAMMING LANGUAGES

3.5.1 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 this project, libraries used are speech recognition to recognize voice, Pyttsx for text to speech, selenium for web automation etc.

It's owing to the subsequent strengths that Python has –

- ✓ **Easy to be told and perceive-** The syntax of Python is simpler; thence it's comparatively straightforward, even for beginners conjointly, to be told and perceive the language.
- ✓ **Multi-purpose language** – Python could be a multi-purpose programming language as a result of it supports structured programming, object-oriented programming yet as practical programming.
- ✓ **Support of open supply community** – As being open supply programming language, Python is supported by awfully giant developer community. Because of this, the bugs square measure simply mounted by the Python community. This characteristic makes Python terribly strong and adaptive.

3.5.2 DOMAIN

The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

A *thing* in the internet of things can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has builtin sensors to alert the driver when tire pressure is low or any other natural or man-made object that can be assigned an Internet Protocol (IP) address and is able to transfer data over a network.

Increasingly, organizations in a variety of industries are using IoT to operate more efficiently, better understand customers to deliver enhanced customer service, improve decision-making and increase the value of the business. An IoT ecosystem consists of web-enabled smart devices that use embedded systems, such as processors, sensors and communication hardware, to collect, send and act on data they acquire from their environments.

IoT devices share the sensor data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analysed or analysed locally. Sometimes, these devices communicate with other related devices and act on the information they get from one another. The devices do most of the work without human intervention, although people can interact with the devices -- for instance, to set them up, give them instructions or access the data.

3.6. SYSTEM ARCHITECTURE

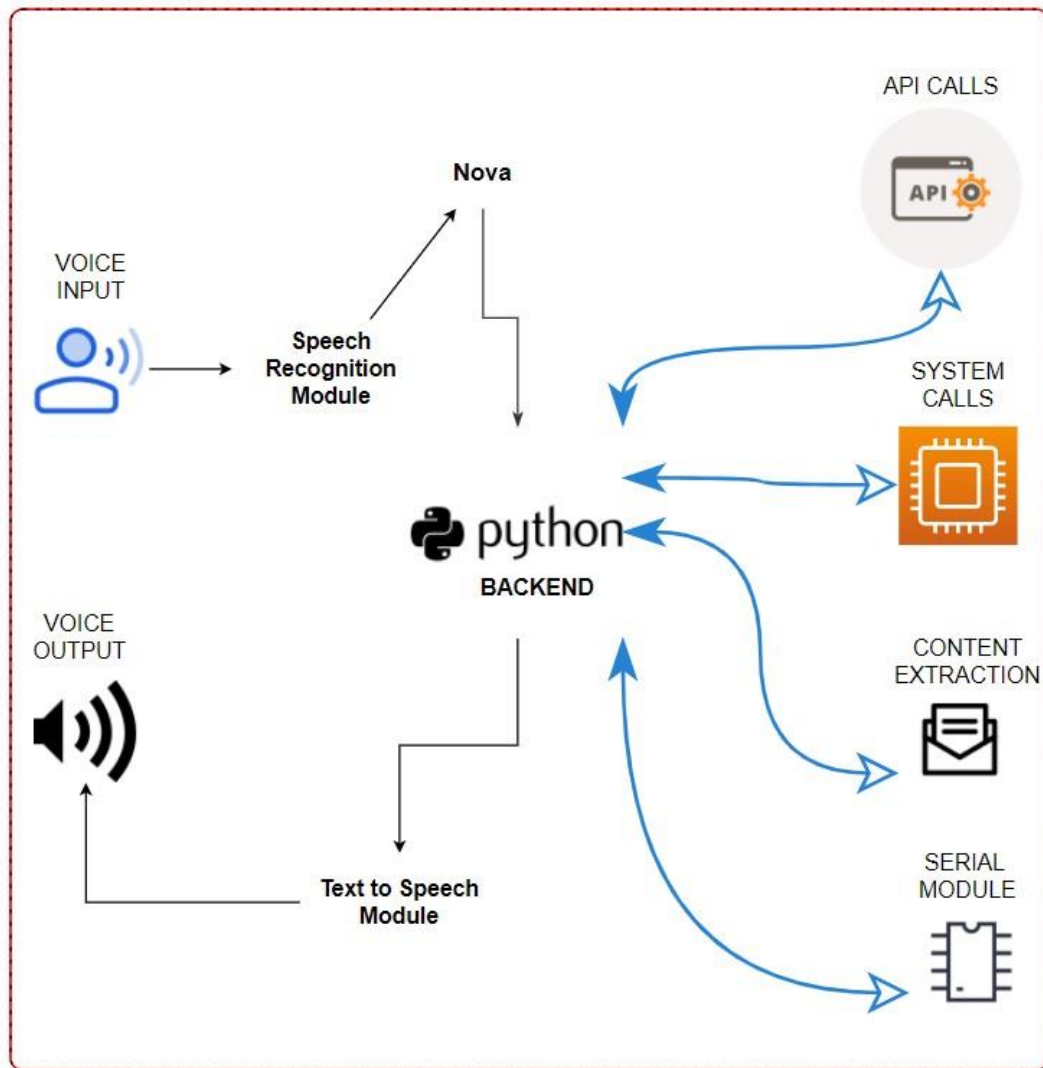


Fig 3.1 System Architecture

3.7 ALGORITHMS USED

3.7.1 SPEECH RECOGNITION MODULE

- ✓ The class which we are using is called Recognizer.
- ✓ It converts the audio files into text and module is used to give the output in speech.
- ✓ **Energy threshold** function represents the energy level threshold for sounds. Values below this threshold are considered silence, and values above this threshold are considered speech.

- ✓ Recognizer instance.adjust_for_ambient_noise(source, duration = 1), adjusts the energy threshold dynamically using audio from source (an AudioSource instance) to account for ambient noise.

3.7.2 SPEECH TO TEXT & TEXT TO SPEECH CONVERSION

- ✓ Pyttsx3 is a text-to-speech conversion library in Python. And can change the Voice, Rate and Volume by specific commands.
- ✓ Python provides an API called Speech Recognition to allow us to convert audio into text for further processing converting large or long audio files into text using the Speech Recognition API in python.
- ✓ We have Included sapi5 and espeak TTS Engines which can process the same.

3.7.3 PROCESS & EXECUTES THE REQUIRED COMMAND

- ✓ The said command is converted into text via speech recognition module and further stored in a temp.
- ✓ Then, Analyze the user's text via temp and decide what the user needs based on input provided and runs the while loop.
- ✓ Then, Commands are executed.

3.8 SYSTEM DESIGN:

3.8.1 USE CASE DIAGRAM:

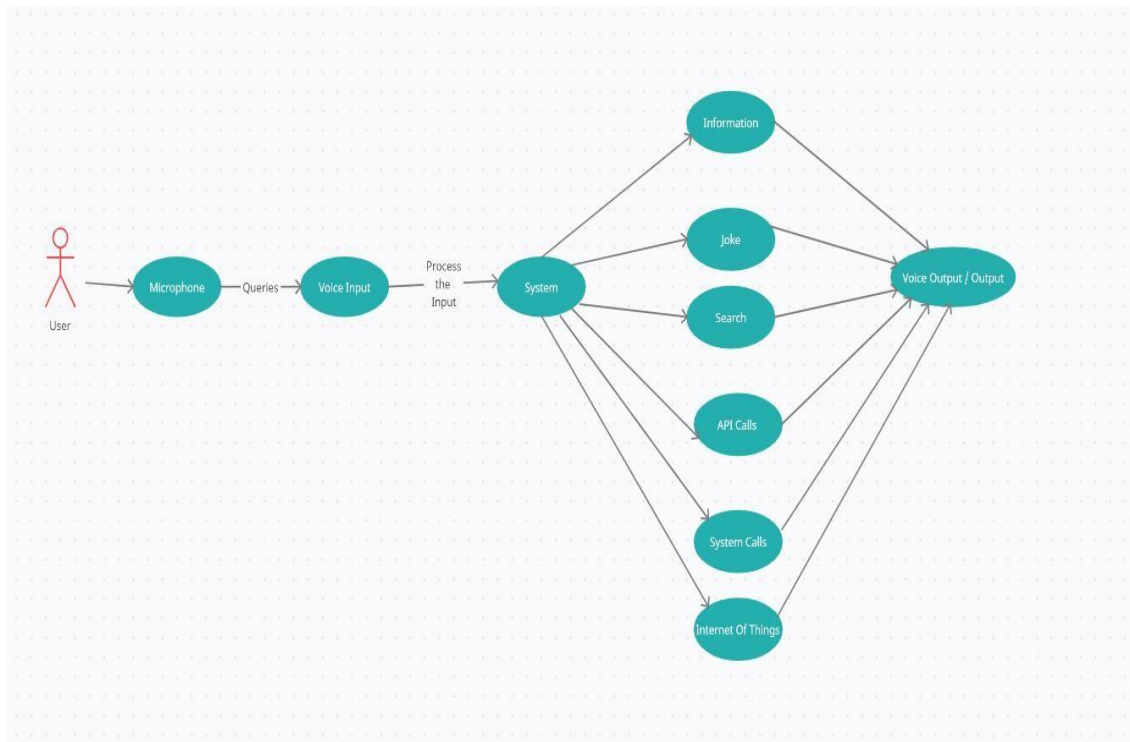


Fig 3.2. Use Case Diagram

- ✓ 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.

3.8.2 COMPONENT DIAGRAM:

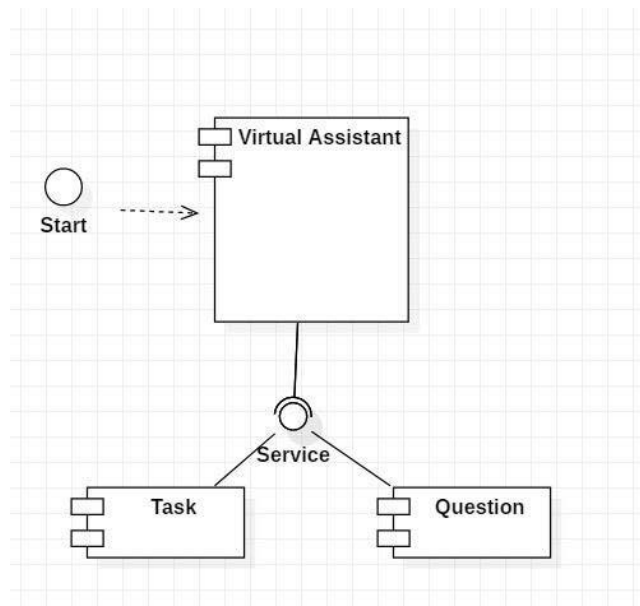


Fig 3.3. Component Diagram

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

SEQUENCE DIAGRAM:

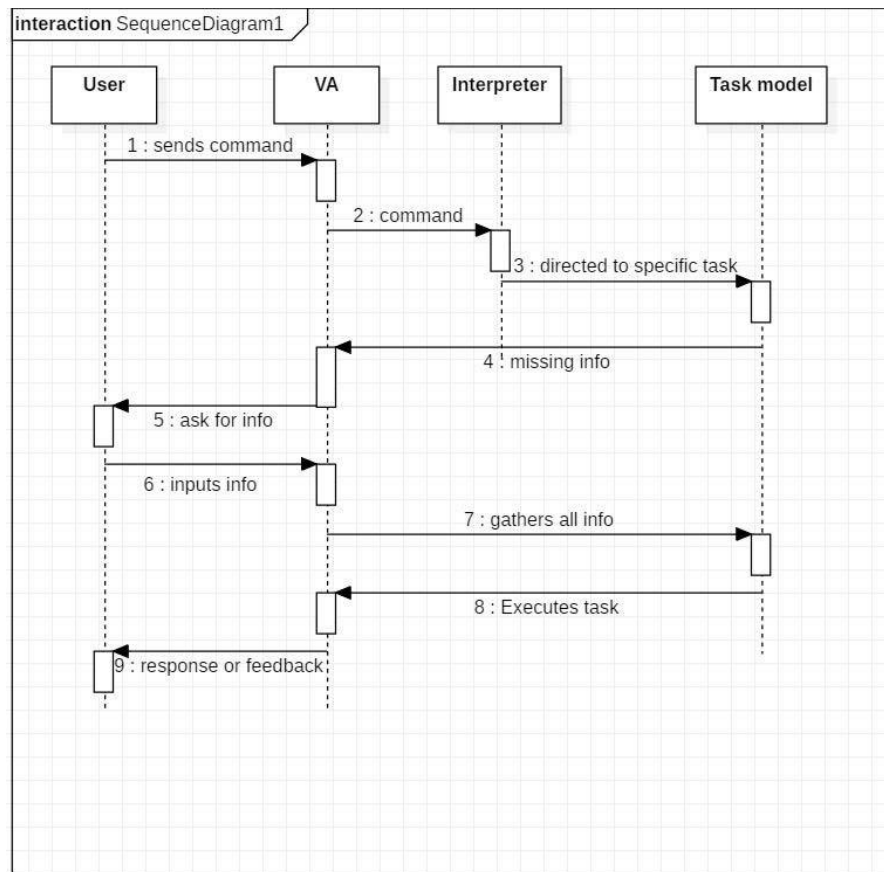


Fig 3.4. Sequence Diagram

- ✓ 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.

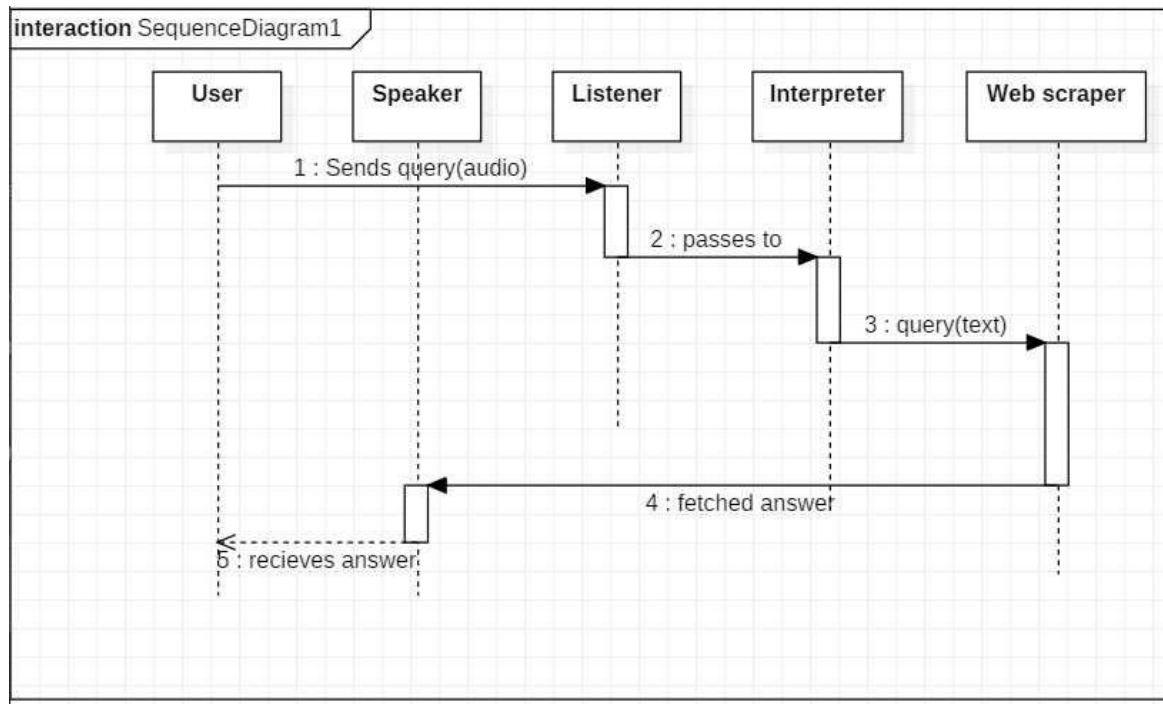


Fig 3.5. Sequence Diagram (Answering the user)

✓ 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.

3.9 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, 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, it 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. 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.

3.10. TYPES OF OPERATION

- Information:

If we ask for some information, it opens up wikipedia and asks us the topic on which we want the information, then it clicks on the wikipedia search box using its xpath, searches the topic in the search box and clicks the search button using the xpath of the button and reads a paragraph about that topic.

Keyword: information

- Plays the video which we ask:

If we ask it to play a video, it opens up YouTube and asks us the name of the video which it wants to play. After that, it clicks on the search YouTube search box using its xpath, then it clicks on the search button using its xpath and clicks the first result of the search using the xpath of the first video.

Keyword: Play and video or music

- News of the day:

If we ask for the news, it reads out the Indian news of the day on which it is asked.

Keyword: news

- Temperature and Weather:

If the user asks the temperature, it gives the current temperature. Keyword: temperature

- Joke:

If the user asks for a joke, it tells a one liner joke to the user. Keyword: funny or joke

- Fact:

If the user asks for some logical fact, it tells a fact to the user. Keyword: fact

- Game:

The assistant can play the number guessing game with the user. First, it asks for the lower and the upper limit between which the number should be. Then it initializes a random number between that upper and lower limit. After that, it uses a formula to calculate the number of turns within which the user should guess the number.

Keyword: game

- Restart the system:

The assistant restarts the system if the user asks the assistant to restart the system.

Keyword: Restart the system or Reboot the system

- Open:

The assistant will open some of the folders and applications which the user asks the assistant to open.

Keyword: Open

- Date and Time:

If the user asks for the date or time, the assistant tells it.

Keyword: date or time or date and time

- Calculate:

The assistant will calculate the equations which the user tells it to calculate using wolframalpha API key.

Keyword : calculate (along with the equation)

- . Tells its name:

The assistant tells its name if the user asks it. The name of the assistant is Next
Gen Optimal Assistant NOVA.

Keyword: Name

- Exit:

The assistant will stop assisting the user if the user asks it to exit.

Keyword: exit or end or stop.

CHAPTER 4

RESULTS AND DISCUSSION

The project work of the voice assistant has been clearly explained in this report, how useful it is and how we can rely on a voice assistant for performing any/every task which the user needs to complete and how the assistant is developing everyday which we can hope that it'll be one of the biggest technology in the current technological world. Development of the software is almost completed from our side and it's working fine as expected which was discussed for some extra development. So, maybe some advancement might come in the near future where the assistant which we developed will be even more useful than it is now.

4.1. WORKING

It starts with a signal word. Users say the names of their voice assistants for the same reason. They might say, "Hey Siri!" or simply, "Alexa!" Whatever the signal word is, it wakes up the device. It signals to the voice assistant that it should begin paying attention. After the voice assistant hears its signal word, it starts to listen. The device waits for a pause to know you've finished your request. The voice assistant then sends our request over to its source code. Once in the source code, our request is compared to other requests. It's split into separate commands that our voice assistant can understand. The source code then sends these commands back to the voice assistant. Once it receives the commands, the voice assistant knows what to do next. If it understands, the voice assistant will carry out the task we asked for. For example, "Hey NOVA! What's the weather?" NOVA reports back to us in seconds. The more directions the devices receive, the better and faster they get at fulfilling our requests. The user gives the voice input through microphone and the assistant is triggered by the wake up word and performs the STT (Speech to Text) and converts it into a text and understands the Voice input and further performs the task said by the user repeatedly and delivers it via TTS (Text to Speech) module via AI Voice.

These are the important features of the voice assistant but other than this, we can do an plenty of things with the assistant.

List of features that can be done with the assistant:

- Playing some video which, the user wants to see.
- Telling some random fact at the start of the day with which the user can do their work in an informative way and the user will also learn something new.
- One of the features which will be there in every assistant is playing some game so that the user can spend their free time in a fun way.
- Users might forget to turn off the system which might contain some useful data but with a voice assistant, we can do that even after leaving the place where the system is just by commanding the assistant to turn the system off.

As discussed about the mandatory features to be listed in voice assistant are implemented in this work, brief explanation is given below.

API CALLS

We have used API keys for getting news information from newsapi and weather forecast from openweathermap which can accurately fetch information and give results to the user.

SYSTEM CALLS

In this feature, we have used OS & Web Browser Module to access the desktop, calculator, task manager, command prompt & user folder. This can also restart the pc and open the chrome application.

CONTENT EXTRATION

This can Perform content extraction from YouTube, Wikipedia and Chrome using the web driver module from selenium which provides all the implementations for the webdriver like searching for a specific video to play, to get a specific information in google or from Wikipedia.

SERIAL MODULES

Finally, we used the serial module for implementing the Internet of Things (IOT) feature for this project. It is a module which acquires the access for the serial port of the Arduino board and used port number 11 and COM3.

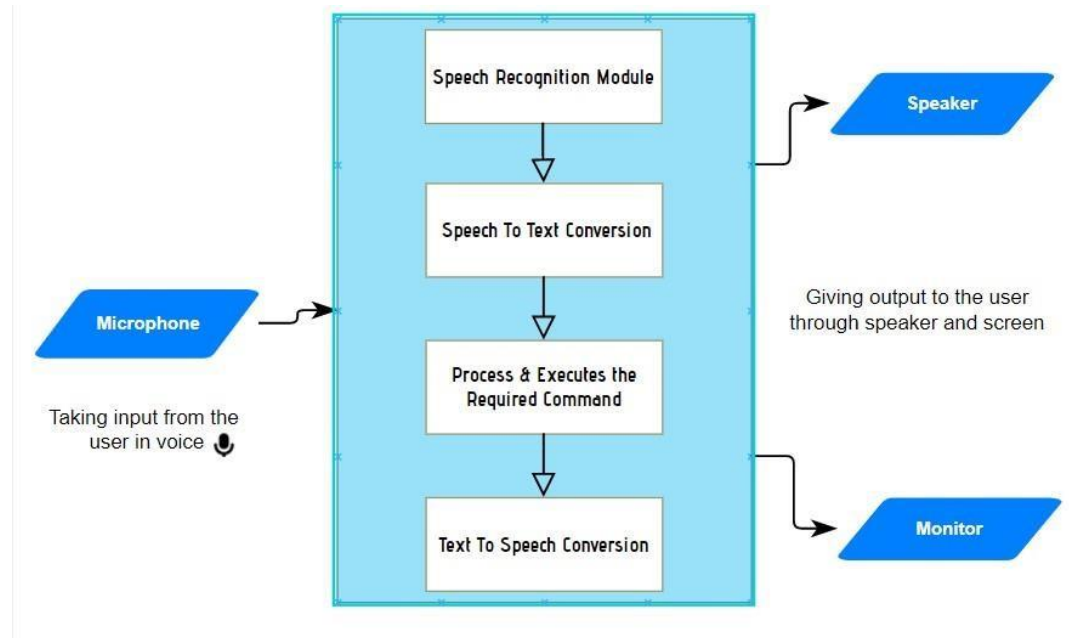


Fig 4.1. Flowchart

1) Must provide the user any information which they ask for: -

The user might need any information which will be available on the internet but searching for that information and reading that takes a lot of time but with the help of a voice assistant, we can complete that task of getting the information sooner than searching and reading it. So, this is a small proof that a voice assistant helps the user to save time

2) Telling the day's hot news in the user's location: -

In Common, watching a news channel just to know the important news in one's location takes a lot of time and the user might even want to listen to some news which is unnecessary to them or a news of some different location before getting to know the news which they want needs a lot of patience to the user but having a voice assistant makes all that nothing, it'll give the news of the location which the user wants to now or the news which they want to know.

3) Telling some joke to chill up the moment: -

Now let's be honest, everyone would have had at least one moment in their life where they were so tensed up or had an argument with their close people. So, these moments can be chilled up at least ten percentage with some random joke which might cool us that moment or stop that fight. We even have a quote stating "Laughter is the best medicine" which is relatable to the words mentioned here in this paragraph.

4) Opening the file/folder which the user wants: -

In the busy world, everything should do quick else, our schedule will get changed and sometimes we need assistance of someone to complete that task quickly but, if we have a voice assistant, we can complete that task in right away in a hustle freeway. For example, let's say the user is doing some documentation but after a while, he needs some file for reference and he goes searching for that file which wastes a lot of time and he ends up missing the deadline but, with a voice assistant we can do the searching part in a quick way by commanding the assistant to open the folder. So, by this we can say that it is one of the important features of a voice assistant.

5) Telling the temperature/weather at the user's location: -

Let's start this with a question, why is it important for us to know the weather of the day? or why is it important for us to monitor the weather every day? The answer is pretty simple it forewarns the users asking about the weather telling that "it might rain today so carry an umbrella if you go out" or "It will be a sunny day so wear a sun glass". So, by this we can say that this is also a must have feature.

6) Searching for what the user asks:

Today in the 20th century, we people often get doubts and we need to clear that doubt as soon as possible else that one doubt will be multiplied and at the end, we'd have n doubts and to clear the doubts searching the question in the internet will give us an answer and clear our doubts and asking that to the assistant will save a lot of time. Other than clearing the doubts, we need to search a lot of questions or topics in the internet to keep up with the trend and we can do this searching just by giving command to our assistant, asking it to search a specific topic/question.

7) Internet of Things:

The final important feature which is the most important feature and that is Internet of Things which is a lot useful because, it'll save a lot of time. Let's take an example, let's say that there is a person with a walking disability and he has to turn on the fan but the switch is a bit far and he can't walk but what he can do is that, he can tell the assistant to turn on the fan and that will turn it on. This is just one example but with the help of IoT, we can do a lot of helpful stuffs like this. These are the important features of the voice assistant but other than this, we can do an ample of stuffs with the assistant.

APPENDICES

A) SOURCE CODE:

```
import speech_recognition as sr

import wolframalpha

from YT_auto import music

from selenium_web_driver import inforr

from News import *

import randfacts

from pyjokes import *

from weather import *

import datetime

from search import sear

import random2

import math

import warnings

import open

import os

import serial

import time

arduino = serial.Serial(port='COM3', baudrate=115200, timeout=.1)


warnings.filterwarnings("ignore")

engine = p.init()

rate = engine.getProperty('rate')

engine.setProperty('rate', 150)

voices = engine.getProperty('voices')

engine.setProperty('voice', voices[0].id)

def speak(text):
```

```

engine.say(text)
engine.runAndWait()
def wishme():
    hour = int(datetime.datetime.now().hour)
    if hour > 0 and hour < 12:
        return ("Morning")
    elif hour >= 12 and hour < 16:
        return ("Afternoon")
    elif hour >= 16 and hour < 19:
        return ("evening")
    else:
        return ("night")

def quitApp():
    hour = int(datetime.datetime.now().hour)
    if hour >= 3 and hour < 18:
        print("have a good day sir")
        speak("have a good day sir")
    else:
        print("Goodnight sir")
        speak("Goodnight sir")
    print("Offline")
    exit(0)

def write_read(x):
    arduino.write(bytes(x, 'utf-8'))
    time.sleep(0.05)
    data = arduino.readline()
    return data

#flags

```

```
Light_status_flag = False
```

```
today_date = datetime.datetime.now()
```

```
r = sr.Recognizer()
```

```
speak("Tell the wake up word")
```

```
wake = "hello Nova"
```

```
with sr.Microphone() as source:
```

```
    r.energy_threshold = 10000
```

```
    r.adjust_for_ambient_noise(source, 1.2)
```

```
    print("Listening")
```

```
    audio = r.listen(source)
```

```
    wakeword = r.recognize_google(audio)
```

```
    print(wakeword)
```

```
if wake == wakeword:
```

```
    while True:
```

```
        speak("hello sir, good " + wishme() + ", i'm here to assist you.")
```

```
        speak("How are you")
```

```
with sr.Microphone() as source:
```

```
    r.energy_threshold = 10000
```

```
    r.adjust_for_ambient_noise(source, 1.2)
```

```
    print("Listening")
```

```
    audio = r.listen(source)
```

```
    text = r.recognize_google(audio)
```

```
    print(text)
```

```
if "what" and "about" and "you" in text:
```

```
    speak("I am also having a good day")
```

```
if __name__ == "__main__":
```

```

while True:

    speak("What can i do for you??")

    with sr.Microphone() as source:

        r.energy_threshold = 10000

        r.adjust_for_ambient_noise(source, 1.2)

        print('Listening.....')

        audio = r.listen(source)

        text2 = r.recognize_google(audio)

        if "information" in text2:

            speak("You need information related to which topic")

            with sr.Microphone() as source:

                r.energy_threshold = 10000

                r.adjust_for_ambient_noise(source, 1.2)

                print('Listening.....')

                audio = r.listen(source)

                infor = r.recognize_google(audio)

            speak("Searching {} in wikipedia".format(infor))

            print("Searching {} in wikipedia".format(infor))

            assist = inforr()

            assist.get_info(infor)

        elif "play" and "video" in text2:

            speak("Which video you want me to play??")

```

```

with sr.Microphone() as source:

    r.energy_threshold = 10000

    r.adjust_for_ambient_noise(source, 1.2)

    print('Listening.....')

    audio = r.listen(source)

```

```

        vid = r.recognize_google(audio)

        speak("Playing {} on youtube".format(vid))

        print("Playing {} on youtube".format(vid))

        assist = music()

        assist.play(vid)
elif "news" in text2:

    speak("Sure sir, Now I will read news for you")

    arr = news()

    for i in range(len(arr)):

        print(arr[i])

        speak((arr[i]))


elif "temperature" in text2:

    speak("Temperature in Chennai is" + str(temp()) + " degree celcius" + " and with " + str(des()))

    print("Temperature in Chennai is" + str(temp()) + " degree celcius" + " and with " + str(des()))

elif "funny" in text2:

    speak("Get ready for some chuckles")

    joke = pyjokes.get_joke()

    speak(joke)

    print(joke)

elif "your name" in text2:

    speak("My name is Next genn Optimal Voice Assistant Nova")


elif "fact" in text2:

    speak("Sure sir , ")

x = randfacts.getFact()

    speak("Did you know that," + x)

    print(x)

elif "search" in text2:

```



```

speak("What should i search for sir")
with sr.Microphone() as source:
    r.energy_threshold = 10000
    r.adjust_for_ambient_noise(source, 1.2)
    print('Listening.....')
    audio = r.listen(source)
    searc = r.recognize_google(audio)
speak("Searching {} in Google".format(searc))
print("Searching {} in Google".format(searc))
asist = sear()
asist.get_infoo(searc)
elif "game" in text2:
    speak("enter your lower limit sir")
    with sr.Microphone() as source:
        r.energy_threshold = 10000
        r.adjust_for_ambient_noise(source, 1.2)
        print('Listening.....')
        audio = r.listen(source)
        lower = int(r.recognize_google(audio))
    speak("now, Enter your upper limit")
    with sr.Microphone() as source:
        r.energy_threshold = 10000
        r.adjust_for_ambient_noise(source, 1.2)
        print('Listening.....')
        audio = r.listen(source)
        upper = int(r.recognize_google(audio))
    x = random2.randint(lower, upper)
    speak("\n\tYou've only " + str(round(math.log(upper - lower + 1, 2))) + " chances to guess the
integer!\n") print("\n\tYou've only " + str(round(math.log(upper - lower + 1, 2))) + " chances to guess
the integer!\n" + str(upper), str(lower))
count = 0

```

```

while count < math.log(upper - lower + 1, 2):
    count += 1
    speak("start guessing")
    speak("Guess a number")
with sr.Microphone() as source:
    r.energy_threshold = 10000
    r.adjust_for_ambient_noise(source, 1.2)
    print('Listening.....')
    audio = r.listen(source)
    guess = int(r.recognize_google(audio))
if x == guess:
    print("Congratulations you did it in " + str(count) + " try")
    speak("Congratulations you did it in " + str(count) + " try")
    break
elif x > guess:
    print("You guessed too small!")
    speak("You guessed too small!")
elif x < guess:
    print("You Guessed too high!")
    speak("You Guessed too high!")
if count >= math.log(upper - lower + 1, 2):
    print("\nThe number is %d" % x)
    speak("\nThe number is %d" % x)
    print("\tBetter Luck Next time!")
    speak("\tBetter Luck Next time!")
elif "reboot the system" in text2:
    speak("Do you wish to restart your computer ?")
with sr.Microphone() as source:
    r.energy_threshold = 10000
    r.adjust_for_ambient_noise(source, 1.2)

```

```

    print('Listening.....')

    audio = r.listen(source)

    restart = r.recognize_google(audio)

elif "light off" in text2:

    #if Light_status_flag == True:

    cmd = "OFF"

    Status = write_read(cmd)

    speak("Lights are turned off")

    #elif Light_status_flag == False:

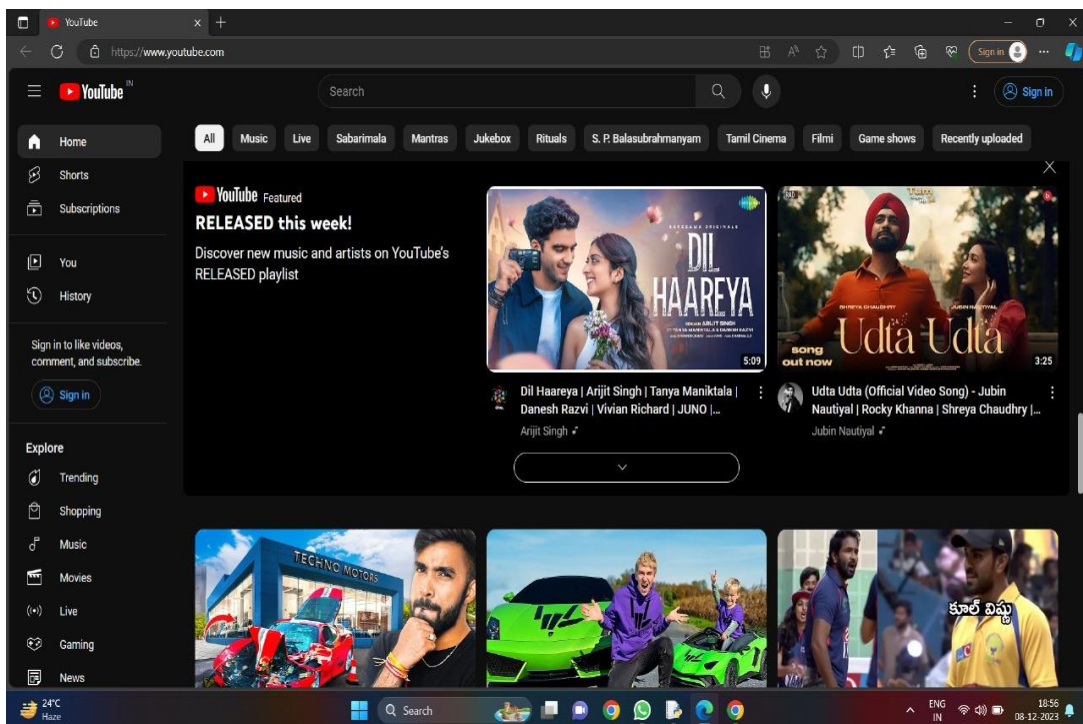
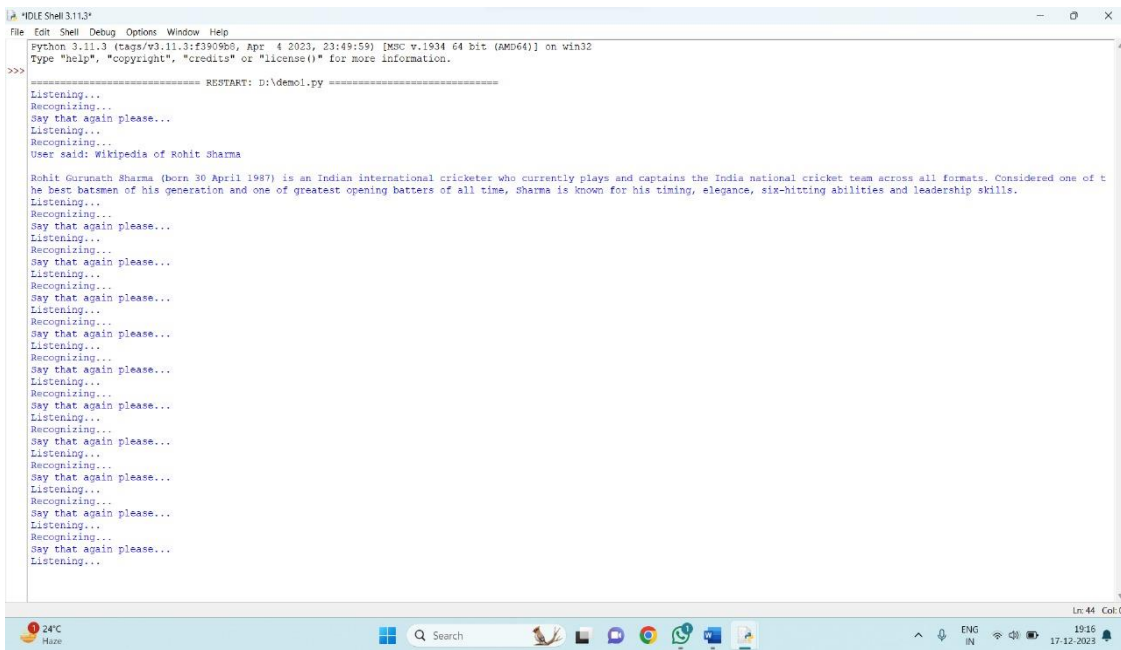
elif "stop" or "exit" or "end" in text2:

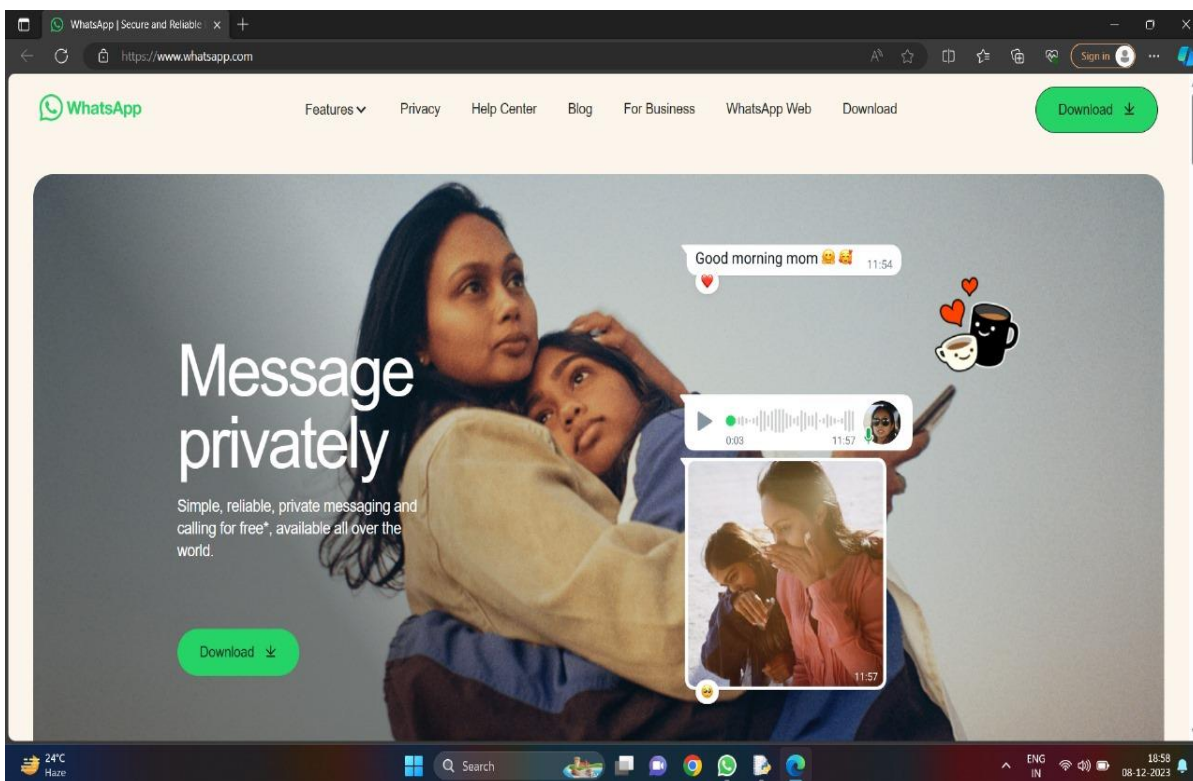
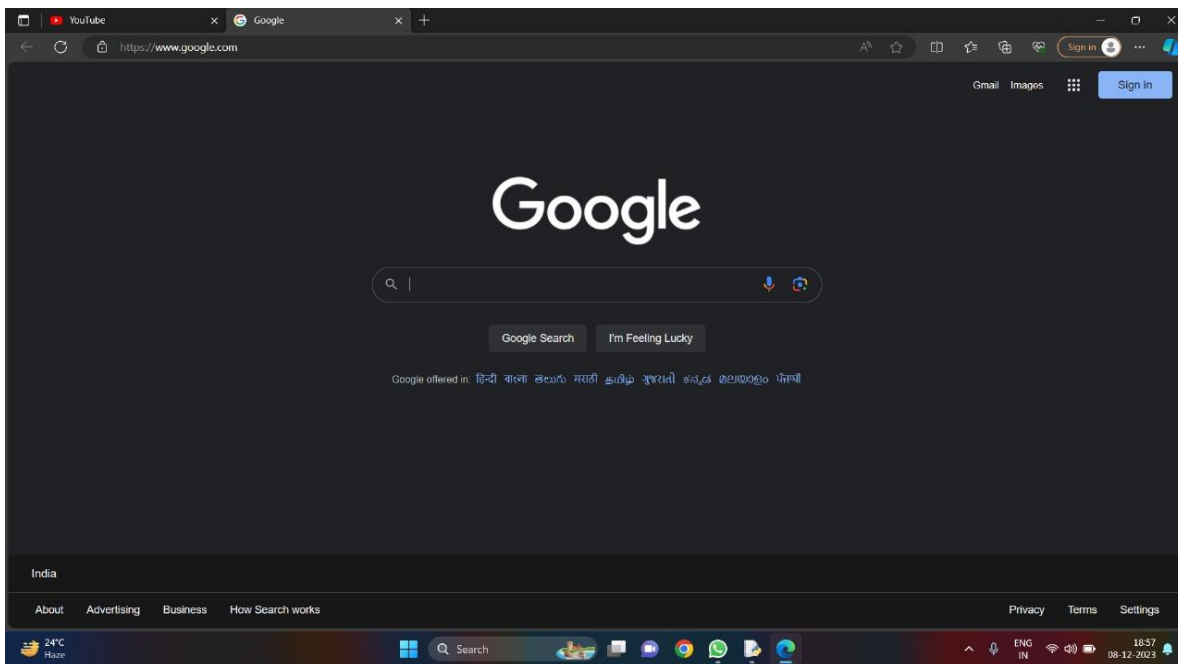
    speak("It's a pleasure helping you and I am always here to help you out!")

    quitApp()

```

SCREENSHOTS





```
demopy - D:\demo.py (3.11.3)
File Edit Format Run Options Window Help
import pyttsx3 #pip install pyttsx3
import speech_recognition as sr #pip install speechRecognition
import datetime
import wikipedia #pip install wikipedia
import webbrowser
import os
import smtplib
import subprocess

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('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<5:
        speak("Good Afternoon!")
    else:
        speak("Good Evening!")
    speak("I am Jarvis")

def takeCommand():
    #It takes microphone input from the user and returns it as a string
    r = sr.Recognizer()
    with sr.Microphone() as mic:
        print("Listening...")
        r.pause_threshold = 1
        audio = r.listen(mic)

    try:
        print("Recognizing...")
        query = r.recognize_google(audio)
        print(f"User said: {query}")
    except Exception as e:
        print(e)
        speak("Sorry, I did not hear you. Please try again.")
        audio = r.listen(mic)

===== RESTART: D:\demo.py =====
Listening...
Recognizing...
User said: what's the time
Listening...
Recognizing...
User said: what's the date
Listening...

Python 3.11.3 (tags/v3.11.3:f3009b8, Apr 4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

```
IDLE Shell 3.11.3
File Edit Shell Debug Options Window Help
Python 3.11.3 (tags/v3.11.3:f3009b8, Apr 4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

CHAPTER 5 CONCLUSION 5.1. CONCLUSION

As stated before, "voice assistant is one of the biggest problem solver" and you can see that in the proposals with the examples that it is in fact one of the biggest problem solver of the current world. We can see that voice assistant is one of the major evolving artificial intelligence in the current world once again on seeing the proposal examples because at the past, the best feature which a voice assistant had was telling the date and searching the web and giving the results but now look at the functions that it can do so with this, we can say that it is a evolving software in the current world. The main idea is to develop the assistant even more advanced than it is now and make it the best ai in the world which will save an ample of time for its users. I would like to conclude with the statement that we will try our best and give one of the best voice assistants which we are able to.

5.2. FUTURE SCOPE

We are entering the era of implementing voice-activated technologies to remain relevant and competitive. Voice-activation technology is vital not only for businesses to stay relevant with their target customers, but also for internal operations. Technology may be utilized to automate human operations, saving time for everyone. Routine operations, such as sending basic emails or scheduling appointments, can be completed more quickly, with less effort, and without the use of a computer, just by employing a simple voice command. People can multitask as a result, enhancing their productivity. Furthermore, relieving employees from hours of tedious administrative tasks allows them to devote more time to strategy meetings, brainstorming sessions, and other jobs that need creativity and human interaction.

1) Sending Emails with a voice assistant:

Emails, as we all know, are very crucial for communication because they can be used for any professional contact, and the finest service for sending and receiving emails is, as we all know, GMAIL. Gmail is a Google-created free email service. Gmail can be accessed over the web or using third-party apps that use the POP or IMAP protocols to synchronize email content.

To integrate Gmail with Voice Assistant we have to utilize Gmail API. The Gmail API allows you to access and control threads, messages, and labels in your Gmail mailbox.

2)scheduling appointments using a voice assistant:

The demands on our time increase as our company grows. A growing number of people want to meet with us. We have a growing number of people who rely on us. We must check in on certain projects or set aside time to chat with possible business leads. There won't be enough hours in the day if we keep doing things the old way.

We need to get a better handle on our full-time schedule and devise a strategy for arranging appointments that doesn't interfere with our most critical job. By working with a virtual scheduler or, in other words, a virtual assistant, we let someone else worry about the organization and prioritize our schedule while we focus on the work.

3)Improved Interface of a voice assistant (VUI):

Voice user interfaces (VUIs) allow users to interact with a system by speaking commands. VUIs include virtual assistants like Amazon's Alexa and Apple's Siri. The real advantage of a VUI is that it allows users to interact with a product without using their hands or their eyes while focusing on anything else.

-Other benefits of a Voice user interface (VUI):

Speed and Efficiency:

Hands-free interactions are possible with VUIs. This method of interaction eliminates the need to click buttons or tap on the screen. The major means of human communication is speech. People have been using speech to form relationships for ages. As a result, solutions that allow customers to do the same are extremely valuable. Furthermore, even for experienced texters, dictating text messages has been demonstrated to be faster than typing. Hands-free interactions, at least in some circumstances, save time and boost efficiency.

Intuitiveness and convenience:

Intuitive user flow is required of high-quality VUIs, and technical advancements are expected to continue to improve the intuitiveness of voice interfaces. Compared to graphical UIs, VUIs require less cognitive effort from the user. Furthermore, everyone – from a small child to your grandmother – can communicate. As a result, VUI designers are in a better position than GUI designers, who run the danger of producing incomprehensible menus and exposing users to the agony of poor interface design.

REFERENCES

- [1] K. Noda, H. Arie, Y. Suga, T. Ogata, Multimodal integration learning of robot behavior using deep neural networks, Elsevier: Robotics and Autonomous Systems, 2014.

- [2] Artificial intelligence (AI), sometimes called machine intelligence. https://en.wikipedia.org/wiki/Artificial_intelligence.

- [3] Deepak Shende, RiaUmahiya, Monika Raghorte, AishwaryaBhisikar, AnupBhange, "AI Based Voice Assistant Using Python", Journal of Emerging Technologies and Innovative Research (JETIR), February 2019, Volume 6, Issue 2.

- [4] J. B. Allen, "From lord rayleigh to shannon: How do humans decode speech," in International Conference on Acoustics, Speech and Signal Processing, 2002.

- [5] John Levis and Ruslan Suvorov, "Automatic Speech Recognition".

- [6] B.H. Juang and Lawrence R. Rabiner, "Automatic Speech Recognition - A Brief History of the Technology Development".

- [7] AbhayDekate, ChaitanyaKulkarni, RohanKilledar, "Study of Voice Controlled Personal Assistant Device", International Journal of Computer Trends and Technology (IJCTT) – Volume 42 Number 1 – December 2016.