|  |  |  |
| --- | --- | --- |
| C:\Users\Administrator\Desktop\AU.png | Voice Controlled AI Assistant for Desktop | **Excel Group Institutions - Wikipedia** |

A MINI PROJECT REPORT

Submitted by

DHANUSH. g (730922202010)

DHINESH KUMAR.n (730922202012)

GOKUL.V (730922202015)

KAVIN KUMAR.K (730922202020)

in partial fulfillment of the requirement

for the award of the degree

of

BAchelor of technology

in

computer science & business systems

EXCEL ENGINEERING COLLEGE (AUTONOMOUS)

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

KOMARAPALAYAM – 637303

novEMBER-2024

**EXCEL ENGINEERING COLLEGE (AUTONOMOUS)**

**KOMARAPALAYAM-637303**

BONAFIDE CERTIFICATE

Certified that this project report titled **“VOICE CONTROLLED AI ASSISTANT FOR DESKTOP”** is the Bonafide work of **DHANUSH .G (730922202010), DHINESH KUMAR.N (730922202012), GOKUL.V (730922202015)** and **KAVIN KUMAR.K (730922202020)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not from part of any other project reported or dissertations on the basis of which degree or award was conferred on an earlier occasion on this or any other candidate.

|  |  |
| --- | --- |
| SIGNATURE  Dr.P.Kumari Ph.D.,  HEAD OF THE DEPARTMENT  Associate Professor and Head  Department of Computer Science  & Business Systems  Excel Engineering College  Komarapalayam – 637303 | SIGNATURE  Mrs.C.Prathipa M.E.,  SUpervisor  Assistant Professor  Department of Computer Science & Business Systems  Excel Engineering College  Komarapalayam – 637303 |

Submitted for the viva-voce examination held on ………………

**Internal Examiner External Examiner**

DECLARATION

We jointly declare that the project report on “**Voice Controlled AI Assistant for Desktop**” is the result of original work done by us and best of our knowledge, similar work has not been submitted to “ANNA UNIVERSITY CHENNAI” for the requirement of Degree of Bachelor of Technology. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of Bachelor of Technology.

**SIGNATURE**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

DHANUSH G

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

DHINESH KUMAR N

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

GOKUL V

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

KAVIN KUMAR K

Place: KOMARAPALAYAM

Date:

**ACKNOWLEDGEMENT**

We wish to express our sincere gratitude to our honourable chairman of Excel Group of Institutions **Dr. Prof. A. K.NATESAN** for providing immense facilities at our institution

We are very proudly rendering our thanks to our Principal **Dr.K.BOMMANNA RAJA** for the facilities and the encouragement given by him to the progress and completion of our project.

We proudly render our immense gratitude to the Head of the Department **Dr.P.KUMARI.,M.E.,Ph.D.,** for her effective leadership, encouragement and guidance in the project.

We are highly indebted to provide our heart full thanks to our supervisor **Mrs.C.PRATHIPA.,M.E.,** Assistant professor for her valuable ideas, encouragement and supportive guidance throughout the project.

We wish to extend our sincere thanks to all faculty members of our Computer Science and Engineering Department for their valuable suggestions, kind cooperation and constant encouragement for successful completion of this project.

We wish to acknowledge the help received from various Departments and various individuals during the preparation and editing stages of the manuscript.

**ABSTRACT**

The **Voice-Controlled AI Assistant for Desktop** is a mini-project aimed at enabling hands-free interaction with desktop computers through voice commands, enhancing user experience and productivity. The system utilizes advanced voice recognition and natural language processing (NLP) technologies to understand and execute tasks such as opening applications, managing files, performing web searches, setting reminders, sending emails, and controlling system settings. By integrating libraries like **Speech Recognition, PyAudio,** and **pyttsx4**, the assistant converts speech into text, processes commands, and provides spoken feedback to users. This allows for a seamless, intuitive user experience, where commands can be issued without the need for manual input devices such as a keyboard or mouse. The project is designed with accessibility in mind, making it particularly useful for individuals with disabilities or those looking for a more efficient way to interact with their computers. Through this AI assistant, users can automate routine tasks, improve multitasking capabilities, and enhance the overall ease of use of desktop systems. Ultimately, the Voice-Controlled AI Assistant for Desktop aims to offer a more interactive, efficient, and user-friendly computing environment by leveraging the power of voice technology for everyday tasks. This project demonstrates the practical implementation of speech-to-text and text-to-speech technologies, alongside modular architecture for task execution, divided into dedicated components for speech recognition, task handling, and application management. The system showcases the capability of AI in simplifying daily tasks and serves as a foundation for future advancements in voice-controlled automation and AI-driven personal assistants. The Voice AI Assistant is a Python-based desktop application designed to enhance human-computer interaction through natural voice commands. Utilizing speech recognition and natural language processing (NLP) techniques.

TABLE OF CONTENTS

CHAPTER NO TITLE PAGE NO.

ABSTRACT V

LIST OF ABBREVIATIONS Viii

LIST OF FIGURES ix

1. INTRODUCTION 1

1.1 Overview 2

1.2 DESIGN 2

1.3 OUR ASSISTANT 3

1.4 WHAT IS VOICE ASSISTANT 3

1.5 WHY DO WE NEEd IT 3

1.6 WHERE TO USE IT 3

1. LITERATURE REVIEW 5

2.1 RELATED WORK 5

1. Methodology 8

3.1 EXISTING SYSTEM 8

3.2 PROPOSED SYSTEM 8

3.3 OBJCETIVE OF PROJECT 9

3.4 SOFTWARE REQUIREMENTS 10

3.5 LIBRARIES 10

3.6 Programming language 13

3.7 domain 13

3.8 algorithms used 15

3.9 system design 17

3.10 TYPES OF OPERATIONS 19

1. EXPERIMENTAL SETUP AND PROCEDURE 21

4.1 expermental setup & procedure 21

4.2 EXPERMENTAL RESULT 23

4.3 SUMMARY OF RESULT 24

1. RESULTS AND DISCUSSION 26

5.1 WORKING 26

5.2 LIST OF FEATURES 27

1. SOURCE CODE & SCREEN SHORT 30
2. conclusion 42

7.1 CONCLUSION 42

7.2 future enhancement 42

1. references 45

LIST OF ABBREVIATIONS

**NLP -** Natural Language Processing

**NLTK -** Natural Language Processing Tool Kit

**AI -** Artificial Intelligence

**IOT -** Internet of Things

**API -** Application Programming Interface

**TTS -** Text to Speech

**STT -** Speech to Text

**OOP -** Object Oriented Programme

**IP -** Internet Protocol

**COM -** Communication Port

**RAD -** Rapid Application Development

LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **Figure Name** | **PAGE NO.** |
| 3.1  3.2  3.3  3.4  4.1 | SYSTEM ARCHITECTURE  USE CASE DIAGRAM  COMPONENT DIAGRAM  SEQUENCE DIAGRAM  FLOWCHART | 15  17  17  18  28 |
|  |  |  |
|  |  |  |
|  |  |  |

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

A disease is a condition that affects the individual functioning of body totally. Diseases if neglected will lead to the death of an individual. Diseases can be identified by the symptoms of the body of an individual. Health is the most important in every human’s life. Weekly or monthly check up of one’s health is most important for the prevention and also to stay healthy.

Healthcare is the most crucial parts of the human life. Nowadays, so many are not willing to go to hospital, due to work overload and negligence of their health. The doctors and nurses are putting up maximum efforts to save people’s lives without even considering their own loves. There are also some villages which lack medical facilities.

Accurate and on-time analysis of any health-related problem is important for the prevention and treatment of the illness. The traditional way of diagnosis may not be sufficient in the case of a serious ailment. In this situation, where everything has turned virtual, the doctors and nurses are putting up maximum efforts to save people’s lives even if they have to danger their own.

There are also some remote villages which lack medical facilities. The dataset was processed in ML models Naive Bayes and Decision Tree. While processing the data, symptoms are given as input and the disease was received as an output. This project helps to get the idea about the disease of an individual based on the symptoms he/she have, and get the treatment easily by contacting the concern doctor.

##### 1.2 DESIGN

1. 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.
2. Converting the speech into text will be processed by the assistant.
3. The converted text is now processed to get the required results.
4. 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.
5. Finally, the output to the user's query will be given by converting speech to text.

##### 1.3 OUR ASSISTANT

Our assistant extends to helps us when working on a system in which it is installed.

###### 1.4 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.

###### 1.5 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.

###### 1.6 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.

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.

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.

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.

* **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 IntelligenceBased 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.

#### METHODOLOGY

#### 3.1 EXISTING SYSTEM

From the above literature survey, we have inferred that all the systems existing predict only particular diseases namely lung disease, breast cancer, heart disease, diabetes by implementing various algorithms on the particular datasets.

After implementing various algorithms, the most accurate one is selected and it is used for prediction of disease. Sometimes, we may get confused of what algorithm to use. Also, all the systems find only the particular disease and not the disease based on the symptoms.

#### 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.

**The proposed system will have the following functionality:**

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

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.

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

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: x86 based processor with 2 core , 3.0 Ghz and above.
* RAM: 4GB with 1333Mhz and Above
* OS: Windows
* Microphone
* Plug Point & a Plug
* Internet with minimum 30Mbps

**3.5 Libraries:**

* **Pyttsx4-** 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 2,3 and 4. An application invokes the pyttsx4.init() factory function to get a reference to a pyttsx4. it is a very easy to use tool which converts the entered text into speech. The pyttsx4 module supports two voices first is female and the second is male which is provided by “sapi5” for windows. Command to install: - pip install pyttsx4.
* **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

* **WolfarmAlpha-** 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 WolfarmAlpha 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.
* **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.
* **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.

#### 3.6. 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.

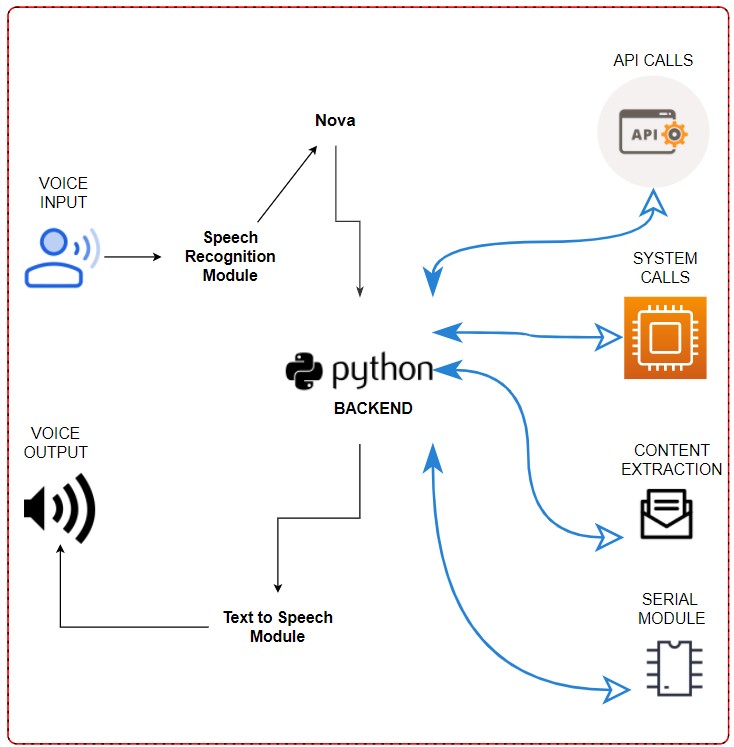
It’s owing to the subsequent strengths that Python has –

* **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 programing 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 programing 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 adaptative.

##### 3.7 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.



**Fig 3.1 System Architecture**

#### 3.8 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 Audio Source instance) to account for ambient noise.

##### 3.7.2 SPEECH TO TEXT & TEXT TO SPEECH CONVERSION

* We have Included sapi5 and speak TTS Engines which can process the same.
* 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.

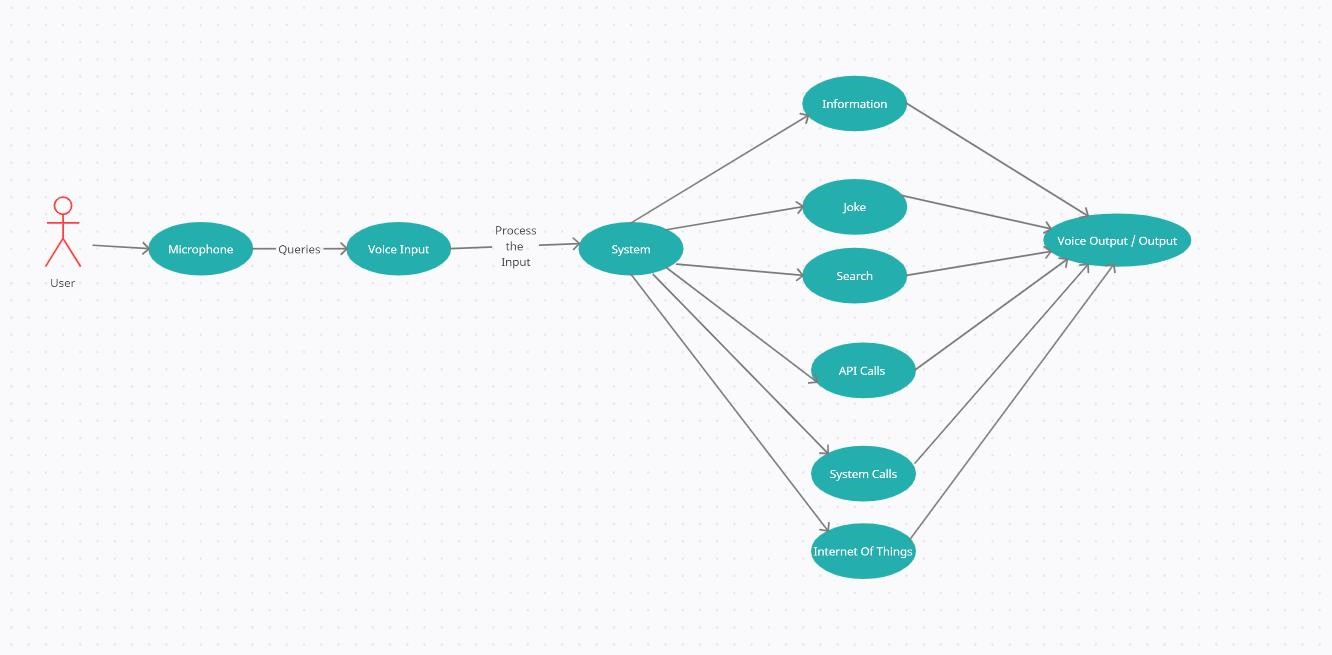
##### 3.7.3 PROCESS & EXECUTES THE REQUIRED COMMAND

* 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.

##### 3.7.4 NLP (Natural Language Processing)

* **Tokenization**: Breaking down text into smaller parts, such as words or sentences. For example, the phrase "Open my email" can be tokenized into the words ["Open", "my", "email"].
* **Stopword Removal**: Removing common words (like "the", "is", "and") that don’t add significant meaning to a query. This helps to focus on key terms in user input.
* **Stemming and Lemmatization**: Reducing words to their base or root form. For example, "running" and "ran" would both be reduced to "run." This helps with understanding different variations of a word.
* **Part-of-Speech Tagging**: Identifying parts of speech (like nouns, verbs, adjectives) for each word in a sentence, which helps in understanding the grammatical structure of a sentence.
* **Text Classification**: Categorizing text into predefined categories. This could help in a voice assistant project by identifying the type of command or query (e.g., "weather" vs. "open app").­**3.9 SYSTEM DESIGN:**

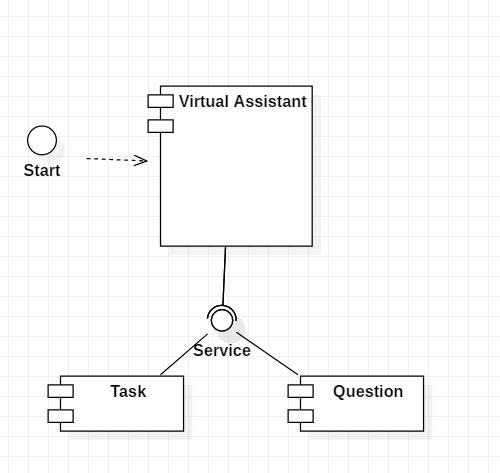
**3.8.1 USE CASE DIAGRAM:**



**Figure 3.2**

 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.

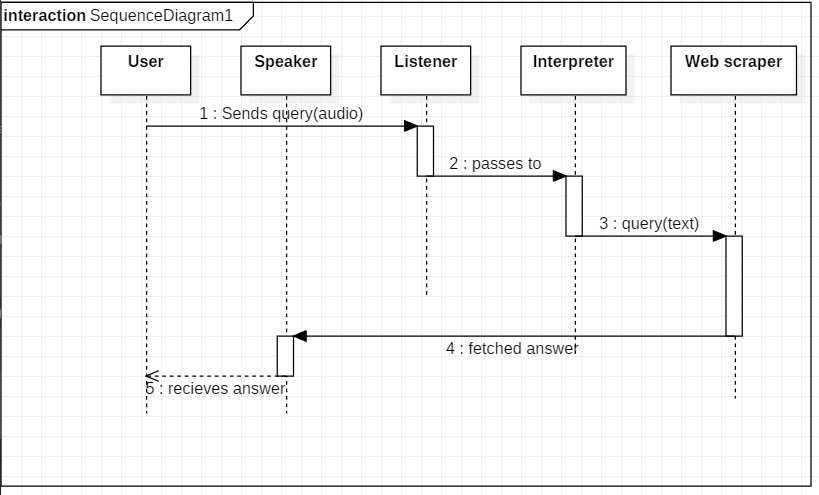
**3.8.2 COMPONENT DIAGRAM:**



**Figure 3.3**

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

**3.8.2 SEQUENCE DIAGRAM:**



###### Figure 3.4

 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.

**3.10 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.

**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.

**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.

**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.

**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.

**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**

**TEMPERATURE & WEATHER :**

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

**Keyword**: Temperature.

**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:**

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

**CALCULATE:**

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

**Keyword :** calculate (along with the equation)

**OPEN:**

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

**Keyword:** Open

**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

**EXIT:**

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

**Keyword:** Exit or Quit

**JOKE:**

If the user asks for a joke, it tells a one liner joke to the user.

**Keyword:** Funny or Joke.

**DATE AND TIME:**

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

**Keyword:** date or time or date and time

**FACT:**

If the user asks for some logical fact, it tells

**Keyword:** Fact

**CHAPTER 4**

**EXPERIMENTAL SETUP AND PROCEDURE**

## 4.1 Experimental Setup and Procedure

#### Hardware Setup

* **Computer System**: The voice assistant is developed and tested on a desktop computer with specifications including an Intel i3 processor, 14GB RAM, GT 710 GPU, a 120GB SSD for the operating system, and additional storage of 300GB HDD. This setup provides adequate computational power for running the voice assistant and handling its various functionalities.
* **Microphone**: An external microphone is used to capture audio input from the user, which is processed through Python’s speech recognition library for converting voice to text.

#### Software Setup

1. **Programming Language and Libraries**: The project is developed using Python, with libraries such as:
2. **Speech Recognition**: For capturing and processing user voice input.

**NLTK**: The Natural Language Toolkit (NLTK) is a powerful Python library used for natural language processing (NLP), which helps computers understand, analyze, and manipulate human language. NLTK provides tools and resources for a wide range of text processing tasks, such as tokenization, stemming, lemmatization, part-of-speech tagging, and text classification. It’s widely used in projects that require language processing, like chatbots, sentiment analysis, and text analysis tools.

**Tokenization**: Breaking down text into smaller parts, such as words or sentences. For example, the phrase "Open my email" can be tokenized into the words ["Open", "my", "email"].

**Stopword Removal**: Removing common words (like "the", "is", "and") that don’t add significant meaning to a query. This helps to focus on key terms in user input.

**Stemming and Lemmatization**: Reducing words to their base or root form. For example, "running" and "ran" would both be reduced to "run." This helps with understanding different variations of a word.

**Part-of-Speech Tagging**: Identifying parts of speech (like nouns, verbs, adjectives) for each word in a sentence, which helps in understanding the grammatical structure of a sentence.

**Text Classification**: Categorizing text into predefined categories. This could help in a voice assistant project by identifying the type of command or query (e.g., "weather" vs. "open app").

**Other Libraries:** Additional libraries for features like opening applications, fetching data from APIs (for weather and search queries), and basic system commands.

1. **IDE**: Python code is written and debugged using a suitable Integrated Development Environment (IDE) like PyCharm or VS Code.

#### Procedure

1. **System Setup**:
   * Ensure that Python and required libraries are installed on the system.
   * Connect the microphone and test the audio input/output settings to confirm they are working correctly.
2. **Project Initialization**:
   * Run the main script (main.py) to initialize the voice assistant.
   * The GUI window appears, built with PyQt5, providing an interface for user interaction.
3. **User Query Processing**:
   * When a user gives a voice command, the Speech Recognition module captures the audio and converts it into text.
   * The NLTK library preprocesses this text, cleaning and organizing it into a format that the assistant can interpret accurately.
4. **Executing Commands**:
   * Based on the processed command, the assistant determines the task to be executed, such as retrieving information from Wikipedia, opening an application, or fetching weather updates.
   * The assistant performs the task and provides a response either through text display or audio feedback.
5. **Testing and Evaluation**:
   * Each feature, such as Wikipedia search, weather fetch, and application control, is individually tested to ensure functionality and user-friendliness.
   * Modifications are made based on the test results to improve accuracy and reliability.

#### 4.2 Experimental Results

In testing, the voice assistant effectively recognized commands, interacted with the user interface, and executed tasks with minimal delay. Adjustments were made to improve the natural language processing to handle a broader range of queries.

**VoiceRecognitionAccuracy**  
The SpeechRecognition library provided approximately 85-90% accuracy in recognizing user commands under normal conditions. Factors such as background noise, microphone quality, and user pronunciation affected the recognition rate. Through repeated tests, it was noted that:

* In a quiet environment, the assistant recognized commands with high accuracy.
* Background noise and unclear pronunciations occasionally led to misinterpretation of words.

**NaturalLanguageProcessing(NLP)Performance**  
Using the NLTK library for tokenization, stopword removal, and lemmatization improved the assistant’sunderstanding of user input. Test results showed that:

* Simple queries such as “What is the weather?” or “Open calculator” were processed accurately.
* Complex or ambiguous queries required additional refinement, as some sentences were misinterpreted if they included multiple keywords.

**ResponseTime**  
On average, the voice assistant responded within 1–2 seconds for basic commands, including:

* **Wikipedia Search**: Approximately 2-3 seconds to retrieve information, depending on the length of the article.
* **Weather Information**: Less than 2 seconds, based on the speed of the internet connection.
* **System Commands (e.g., opening applications)**: Instantaneous response, limited only by the application’s loading time.

**Error Handling and Improvements**  
During testing, a few areas for improvement were identified:

* **Voice Command Ambiguity**: Commands like “Open my email” and “Check email” were sometimes interpreted differently. Implementing additional NLP techniques could reduce misinterpretation.
* **Connection Errors for Online Data**: Occasional internet connection issues affected weather and Wikipedia searches. Implementing error messages in the GUI for network issues helped notify the user when data couldn’t be retrieved.

**OverallUserSatisfaction**  
Based on informal user feedback, the voice assistant was well-received in terms of ease of use, response time, and feature variety. However, feedback suggested further improvements in handling natural language variations and enhancing the microphone’s accuracy in noisy settings.

#### 4.3 Summary of Results

The voice assistant effectively handled core functionalities with high accuracy and quick response times. Improvements in handling complex language and additional error handling mechanisms were noted as areas for future enhancements.

* Achieved 85-90% accuracy in a quiet environment.
* Performance affected by background noise and pronunciation, suggesting the need for further noise filtering and speech enhancement techniques
* NLTK-based preprocessing (tokenization, stopword removal, lemmatization) improved understanding of straightforward commands.
* Complex or multi-part queries were sometimes misunderstood, indicating that additional NLP processing or more advanced parsing techniques could improve accuracy.
* Basic commands executed almost instantly, with application control showing immediate response.
* API-dependent commands (Wikipedia, weather) responded within 1-3 seconds, with response time largely influenced by network connectivity.
* Error handling mechanisms (e.g., handling network failures for API calls) were effective in improving user experience by providing relevant notifications.
* Further improvements, like more detailed error messages, were identified as beneficial for complex command handling.
* The assistant correctly interpreted various ways of phrasing simple commands (e.g., "Open browser" and "Start Chrome").
* Variability was less successful with more complex phrases, suggesting a need for synonyms and contextual recognition to be expanded.

# CHAPTER 5

result 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 form 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.

#### 5.1. WORKING

They might say, “Hey Siri!” or simply, “Alexa!” Whatever the signal word is, it wakes up the device. It starts with a signal word. Users say the names of their voice assistants for the same reason. 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.

**5.2 List of features**

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 webdrive 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.

**FLOWCHART DIAGRAM:**

##### 

##### Figure 4.1

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

1. **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.

1. **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.

1. **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.

1. **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.

1. **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.

**CHAPTER 6**

**SOURCE CODE**

**SOURCE CODE:**

**A) Main.py:**

**from speech\_module import wishMe, takeCommand, speak**

**from task\_module import perform\_task**

**if \_\_name\_\_ == "\_\_main\_\_":**

**wishMe()**

**last\_query = ""**

**while True:**

**query = takeCommand().lower()**

**if query != "none":**

**last\_query = query**

**perform\_task(query)**

**# Extra conversation units**

**if "how are you" in query or "how r u" in query:**

**print("I'm just a program, but I'm functioning well. Thank you for asking!")**

**speak("I'm just a program, but I'm functioning well. Thank you for asking!")**

**elif "help" in query:**

**print("You can ask me to open websites, tell you the time, create or delete files, and more.")**

**speak("You can ask me to open websites, tell you the time, create or delete files, and more.")**

**elif "was that helpful" in query:**

**speak("I hope it was! Let me know if you need more assistance.")**

**elif "love you" in query or "love u" in query:**

**print("I am just a Program :( ")**

**speak("I am just a program, and please don't think about that...")**

**elif "what's your name" in query or "who are you" in query:**

**print("I am your voice assistant, here to help you!")**

**speak("I am your voice assistant, here to help you!")**

**B) Speech\_Module.py:**

**import pyttsx4**

**import speech\_recognition as sr**

**import datetime**

**import colorama**

**from colorama import Fore, Style**

**# Initialize TTS engine**

**engine = pyttsx4.init('sapi5')**

**voices = engine.getProperty('voices')**

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

**# Function to convert text to speech**

**def speak(audio):**

**engine.say(audio)**

**engine.runAndWait()**

**# Function to wish the user based on the time**

**def wishMe():**

**hour = int(datetime.datetime.now().hour)**

**if hour >= 0 and hour < 12:**

**speak("Good Morning!")**

**elif hour >= 12 and hour < 18:**

**speak("Good Afternoon!")**

**else:**

**speak("Good Evening!")**

**speak("I am your assistant. How can I help you today?")**

**# Function to take voice command from user**

**def takeCommand():**

**recognizer = sr.Recognizer()**

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

**print(Fore.GREEN + "Listening..." + Style.RESET\_ALL)**

**recognizer.pause\_threshold = 1 # responce time**

**recognizer.energy\_threshold = 300 # noice reducing**

**audio = recognizer.listen(source)**

**try:**

**print(Fore.YELLOW + "Recognizing..." + Style.RESET\_ALL)**

**query = recognizer.recognize\_google(audio, language='en-in')**

**print(Fore.CYAN + f"User said: {query}\n" + Style.RESET\_ALL)**

**except sr.UnknownValueError:**

**print(Fore.RED + "Sorry, I did not catch that. Please say that again..." + Style.RESET\_ALL)**

**return "None"**

**except sr.RequestError:**

**print(Fore.RED + "Could not request results from Google Speech Recognition service." + Style.RESET\_ALL)**

**return "None"**

**return query**

**C) Task\_Module.py:**

**import webbrowser**

**import wikipedia**

**import os**

**import datetime**

**import os**

**import nltk**

**from nltk.corpus import stopwords**

**from nltk.tokenize import word\_tokenize**

**from nltk.stem import WordNetLemmatizer**

**from speech\_module import speak**

**# Initialize NLP components**

**lemmatizer = WordNetLemmatizer()**

**stop\_words = set(stopwords.words('english'))**

**# Download necessary resources for nltk**

**nltk.download('punkt', quiet=True)**

**nltk.download('stopwords', quiet=True)**

**nltk.download('wordnet', quiet=True)**

**# Preprocess query using NLP**

**def preprocess\_query(query):**

**tokens = word\_tokenize(query.lower())**

**filtered\_tokens = [word for word in tokens if word not in stop\_words]**

**lemmatized\_tokens = [lemmatizer.lemmatize(token) for token in filtered\_tokens]**

**processed\_query = ' '.join(lemmatized\_tokens)**

**return processed\_query**

**# Perform tasks based on the query**

**def perform\_task(query):**

**query = preprocess\_query(query)**

**if 'open youtube' in query:**

**speak('Opening YouTube')**

**webbrowser.open("https://www.youtube.com")**

**elif 'open google' in query:**

**speak('Opening Google')**

**webbrowser.open("https://www.google.com")**

**elif 'instagram' in query:**

**speak('Opening Instagram')**

**webbrowser.open("https://instagram.com")**

**elif 'facebook' in query:**

**speak("Opening Facebook")**

**webbrowser.open("https://facebook.com")**

**elif 'search for' in query or 'google' in query:**

**search\_term = query.replace("search for", "").replace("google", "").strip()**

**if search\_term:**

**speak(f"Searching Google for {search\_term}")**

**webbrowser.open(f"https://www.google.com/search?q={search\_term}")**

**else:**

**speak("What do you want to search for?")**

**elif 'wikipedia' in query:**

**speak('Searching Wikipedia...')**

**query = query.replace("wikipedia", "").strip()**

**try:**

**results = wikipedia.summary(query, sentences=2)**

**speak("According to Wikipedia")**

**print(results)**

**speak(results)**

**except wikipedia.exceptions.DisambiguationError:**

**speak("There are multiple results for this. Please be more specific.")**

**except wikipedia.exceptions.PageError:**

**speak("Sorry, I couldn't find anything on Wikipedia about that.")**

**# System Applications and Folders**

**elif 'open notepad' in query:**

**speak("Opening Notepad")**

**os.startfile('C:\\Windows\\system32\\notepad.exe')**

**elif 'open browser' in query or 'open web browser' in query:**

**speak("Opening Web browser")**

**os.startfile('C:\\ProgramFiles(x86)\\Microsoft\\Edge\\Application\\msedge.exe')**

**elif 'open excel' in query or 'open microsoft excel' in query:**

**speak("Opening M S excel")**

**os.startfile('C:\\Program Files\\MicrosoftOffice\\Office16\\EXCEL.EXE')**

**elif 'open word' in query or 'open microsoft word' in query:**

**speak("Opening M S word")**

**os.startfile('C:\\ProgramFiles\\MicrosoftOffice\\Office16\\WINWORD.EXE')**

**elif 'open PPT' in query or 'open power point' in query or'open powerpoint' in query :**

**speak("Opening M S Power Point")**

**os.startfile('C:\\ProgramFiles\\MicrosoftOffice\\Office16\\POWERPNT.EXE')**

**elif 'play music' in query:**

**music\_dir = 'C:\\Users\\Admin\\Desktop\\Voice Assistant 2.0\\Songs'**

**songs = os.listdir(music\_dir)**

**if songs:**

**speak('Playing music')**

**os.startfile(os.path.join(music\_dir, songs[0]))**

**else:**

**speak('No music found in the directory.')**

**elif 'time' in query:**

**current\_time = datetime.datetime.now().strftime("%H:%M:%S")**

**print(current\_time)**

**speak(f"The time is {current\_time}")**

**elif 'date' in query:**

**current\_date = datetime.datetime.now().strftime("%d-%m-%Y")**

**print(current\_date)**

**speak(f"The date is {current\_date}")**

**elif 'shutdown the system' in query or 'shut down the system' in query:**

**speak("Shutting Down the System")**

**os.system("shutdown /s /t 1")**

**elif 'quit' in query or 'exit' in query:**

**speak('Goodbye!')**

**exit()**

**D) AUTOMATIC MODULES INSTALLER (Batch Script) :**

**@echo off**

**color 4**

**echo (" First you should Install PYTHON on your PC before you run this Module")**

**pause**

**echo EXECUTE AUTO MODULE INSTALLER**

**color 3**

**echo ( Processes Completed 0%)**

**pip install pyautogui**

**pip install pyttsx4**

**echo ( Processes Completed 5%)**

**pip install pyaudio**

**echo (Processes Completed 10%)**

**pip install SpeechRecognition**

**echo ( Processes Completed 30%)**

**pip install wikipedia**

**echo (Processes Completed 40%)**

**color F**

**pip install pyjokes**

**pip install nltk**

**echo ( Processes Completed 70%)**

**pip install setuptools**

**pip install wikipedia-api**

**echo ( Processes Completed 80%)**

**pip install colorama**

**color 3**

**echo ( Processes Complete 90%)**

**:: Download necessary NLTK resources**

**python -m nltk.downloader punkt**

**python -m nltk.downloader stopwords**

**python -m nltk.downloader wordnet**

**color 2**

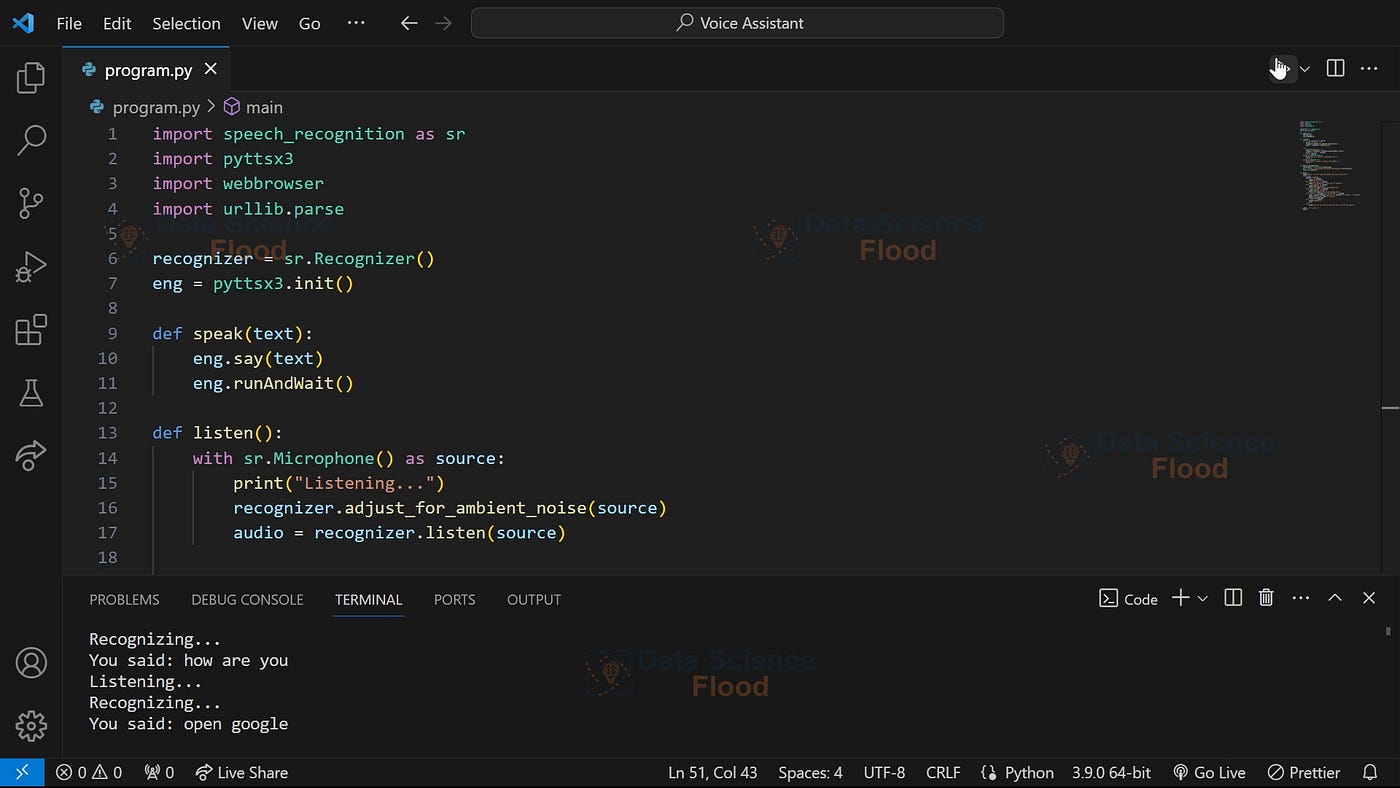
**echo ( Processes Completed 100%)**

**echo All modules installed successfully!**

**pause**

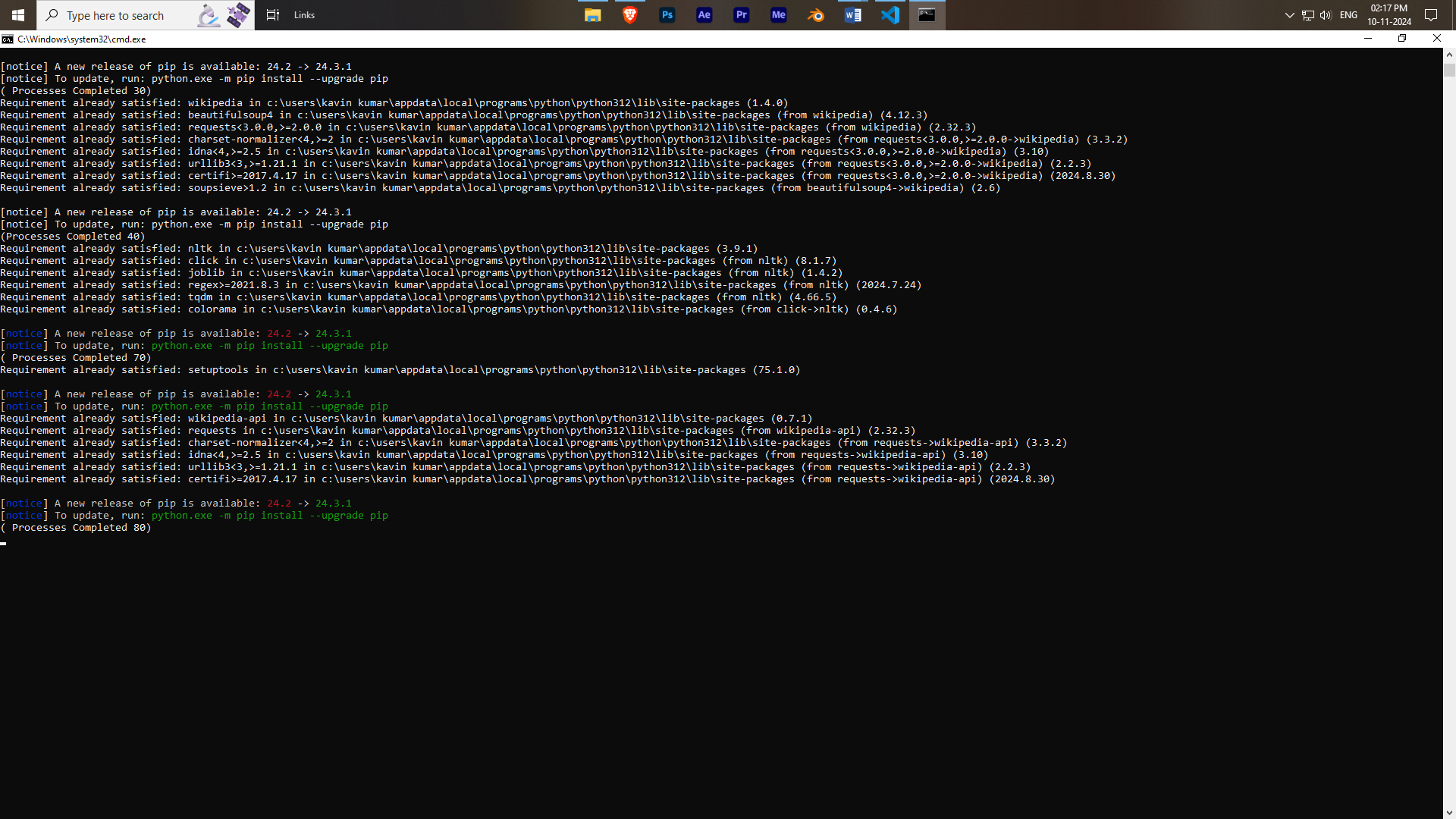
* This is a Windows Batch script designed to automate the installation of Python libraries and resources necessary for this voice assistant project.
* Batch scripts are text files that contain a series of commands to be executed by the command prompt (CMD) in Windows. Here, it installs Python packages automatically so you don’t have to run each command individually.
* This script can be very helpful for quickly setting up your development environment!

**SCREENSHOTS**

****

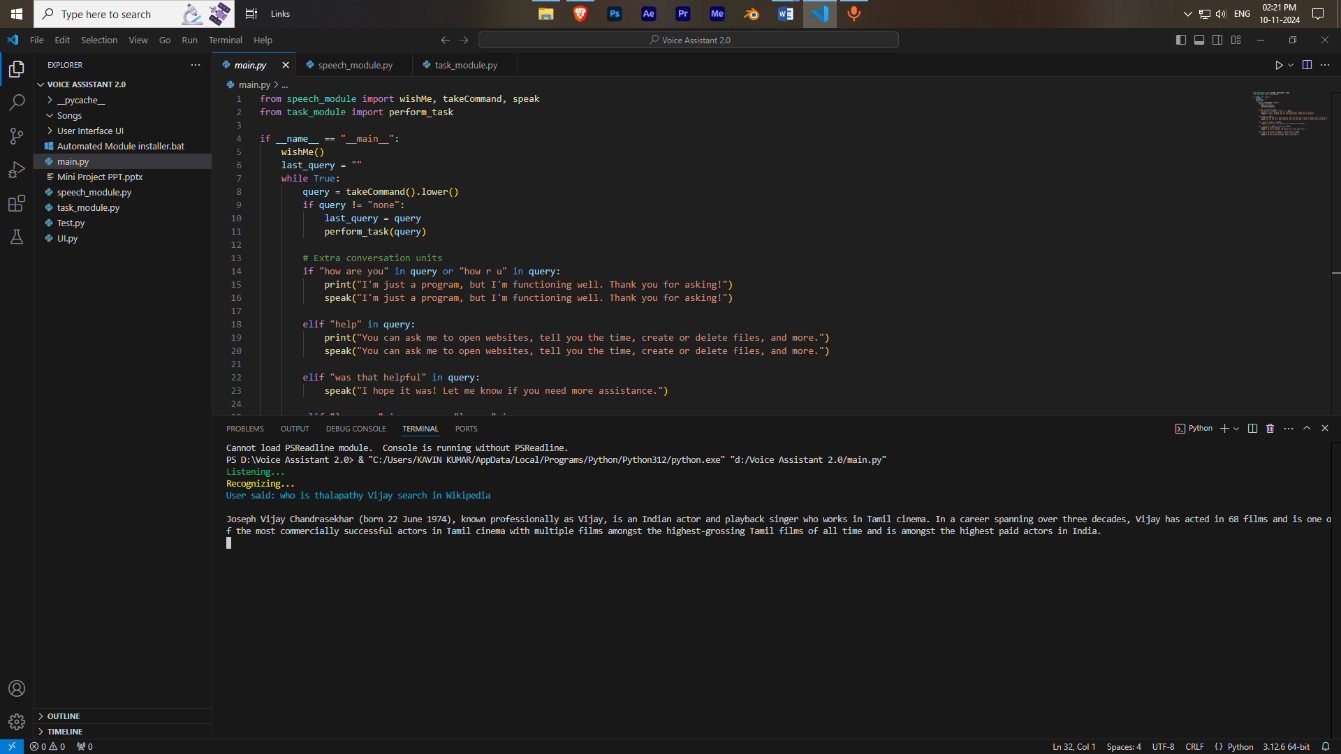
**Figure 6.1**

* In this Picture the main code is executed successfully and the Voice Assistant Programme Reconized Properly an Give the Output Mannerly.

****

**Figure 6.2**

* In this above Figure 6.2 , we created an auto bat file for Automatic Installer for all the modules we used in this program.
* If we run this programme in new PC and another device this program shows the error of (modules not found) so we programmed this bat file for automatic installation of all modules from the Internet.



**Figure 6.3**

* In this above Figure 6.3 all the moules were working perfectly fine with integrated with all the other source of internet to improve the user experience.

**CHAPTER 7**

**CONCLUSION**

**7.1 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 do .

**7.2 FUTURE ENHANCEMENT**

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. s

1. 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.

1. 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. Customers are unlikely to need to be instructed on how to utilize the technology by VUI makers. People can instead ask their voice assistant for assistance.

**CHAPTER 8**

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.](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.