Chapter 1
INTRODUCTION

Voice assistants are software agents that can interpret human speech and respond via synthesized voices. Apple's Siri, Amazon's Alexa, Microsoft's Cortana, and Google's Assistant are the most popular voice assistants and are embedded in smartphones or dedicated home speakers. Users can ask their assistants questions, control home automation devices and media playback via voice, and manage other basic tasks such as email, to-do lists, and calendars with verbal commands. This column will explore the basic workings and common features of today's voice assistants. It will also discuss some of the privacy and security issues inherent to voice assistants and some potential future uses for these devices. As voice assistants become more widely used, librarians will want to be familiar with their operation and perhaps consider them as a means to deliver library services and materials.

1.1 Overview

Speech recognition is one of the most recently developing field of research at both industrial and scientific levels. Until recently, the idea of holding a conversation with a computer seemed pure science fiction. If you asked a computer to "open the pod bay doors"—well, that was only in movies. But things are changing, and quickly. A growing number of people now talk to their desktop, smart phones, asking them to send e-mail and text messages, search for directions, or find information on the Web. Our Project aims at one such application. People have wanted to talk to computers almost from the moment the first computer was invented. The idea of holding meaningful conversation with a computer seemed futuristic, but the technology to make voice interfaces useful and widely available is already here. Several consumer level products developed in the last few years have brought inexpensive voice assistants into everyday use, and more features and platforms are being added all the time. AIVA is one of the virtual assistant.

Today the development of artificial intelligence (AI) systems that are able to organize a natural human-machine interaction (through voice, communication, gestures, facial expressions, etc.) are gaining in popularity. One of the most studied and popular was the direction of interaction, based on the understanding of the machine by the machine of the natural human language. It is no longer a human learns to communicate with a machine, but a machine learns to communicate with a human, exploring his actions, habits, behavior and trying to become his personalized assistant. The work on creating and improving such personalized assistants has been going on for a long time. These systems are constantly improving and improving, go beyond personal computers and have already firmly established themselves in various mobile devices and gadgets. One of the most popular voice assistants are Siri from Apple, Amazon Echo which responds to the name of Alex from Amazon, Cortana from Microsoft, Google Assistant from Google, and the recently appeared intelligent assistant under the name "AIVA".

A virtual assistant is a software agent that can perform tasks or services for an individual. **AIVA** is a Virtual Assistant Virtual powered by Artificial Intelligence. It is able to search the Internet, as if the user gives any query to AIVA it automatically goes to the Internet and get the best fit result for the user's query. AIVA can post comments on the social media

websites such as Facebook, Twitter, Snapchat, Instagram, etc. By just few simple commands. You can also know the weather around you and can get the climate conditions in your region. It can open and launch web applications and the local storage of the user computer. AIVA is the user friendly AI assistant to make the communication easier between the user and the machine.

1.2 Objective

What Are Voice Assistants?

Simply put, voice assistants are the realization of the science fiction dream of interacting with our computers by talking to them. Apple's Siri, Microsoft's Cortana, Amazon's Alexa, and Google's Assistant are all software agents that run on purpose-built speaker devices or smartphones. The software constantly listens for a key word to wake it up. Once it hears that key word, it records the user's voice and sends it to specialized server, which processes and interprets it as a command. Depending on the command, the server will supply the voice assistant with appropriate information to be read back to the user, play the media requested by the user, or complete tasks with various connected services and devices. The number of services that support voice commands is growing rapidly, and Internet-of-Things device manufacturers are also building voice control into their products. Apple's Siri assistant has been around the longest, released as a standalone app in 2010 and bundled into iOS in 2011. Microsoft followed shortly thereafter with Cortana in 2013. Amazon launched Alexa with its Echo-connected home speaker in 2014, and Google's Assistant was announced in 2016 along with its Home speaker and is also embedded in the Google app for Android- based smartphones. Each assistant has its own unique features, but the core functions are the same. Voice assistants differ from earlier voice-activated technologies in that they can respond to a much larger number of commands and questions. This is because they are always connected to the Internet; each interaction is sent back to a central computing system that analyzes the user's voice commands and provides the assistant with the proper response. Earlier voice-activated devices relied on a smaller set of "built-in" commands and responses. Recent advances in natural language processing, also known as computational linguistics, has allowed voice assistants to create meaningful responses quickly. Hirschberg and Manning credit these recent improvements in natural language processing to four things: (i) a vast increase in computing power, (ii) the availability of very large amounts of linguistic data, (iii) the development of highly successful machine learning (ML) methods, and (iv) a much richer understanding of the structure of human language and its deployment in social contexts. As personal computers have grown cheaper and more powerful, and people have created more and more online text to be analyzed, scientists have used that text to train voice assistants to listen and respond to our requests in more natural and meaningful ways. Voice assistants can parse requests phrased in a number of different ways and interpret what the user is most likely to want.

For example, to ask Google's Assistant to remember where one parked his or her car, a user can say any of a number of phrases: "Remember where I parked," "I parked here," "I left the car on 6th street," or "the car is in the south lot" will all get a similar result. Google will remember where the user parked the car and, when asked later, will be able to respond accordingly. The user can ask questions in a similarly natural way; asking "where did I park," "where did I leave the car," or "do you remember where I parked" all trigger the expected response. Natural language processing avoids the user frustration of earlier voice recognition systems, which required specific phrases and patterns in order to work properly.

What Can Voice Assistants Do?

Although each currently available voice assistant has unique features, they share some similarities and are able to perform the following basic tasks:

- Send and read text messages, make phone calls, and send and read email messages.
- Answer basic informational queries ("What time is it? What's the weather forecast? How many ounces are in a cup?").
- Set timers, alarms, and calendar entries.
- Set reminders, make lists, and do basic math calculations.
- Control media playback from connected services such as Amazon, Google Play, iTunes, Pandora, Netflix, and Spotify.
- Control Internet-of-Things-enabled devices such as thermostats, lights, alarms, and locks.
- Tell jokes and stories.
- Comment of social media websites.
- Security related functions.
- Interaction with the user for fun purpose.

In addition to these tasks, voice assistants can add other features, often called "skills," that expand their abilities by interfacing with other programs via voice commands. Amazon's Alexa has skills for playing Jeopardy, ordering your usual drink from your local Starbucks, and summoning an Uber or Lyft using connected account data. Google's Assistant has similar skills but lags behind Amazon in the sheer number of available skills, largely due to being released later. These skills are built by third-party developers, similar to the way apps are developed for smartphones. Google Assistant also integrates with several tools that allow users to create their own skills. Using web services like Tasker and IFTTT (If This Then That), users can craft skills that will allow them to automate social media posts, turn devices on and off, and hundreds of other possibilities. For example, telling Assistant "Good morning" could launch a number actions designed to speed up the user's morning routine: turning on the coffee maker, reading the news and calendar events, opening the garage door and starting the car, and then locking the doors, arming the alarm, and adjusting the thermostat after occupants have left the house for the day. All of the devices necessary to perform these tasks remotely are available today; a voice assistant simply provides the bridge that allows users to issue commands verbally rather than via an app. Voice assistants are available on a wide variety of hardware platforms.

Amazon and Google both market dedicated home speaker devices for their voice assistants. Amazon makes several variations of its Echo product, from the tiny Echo Dot to the Echo Show, which has audio and video capabilities. Google's speaker hardware is called the Home and also comes in mini- and full-size models. Apple is just entering the home speaker market, with the announcement of its Siri-enabled HomePod device. Microsoft has focused on building Cortana into Windows 10 PCs and phones and recently partnered with Harmon Kardon to develop a Cortana enabled home speaker. Assistants are available on most smartphone platforms as well; Google's assistant is integrated into Android phones and can be installed as a separate app on the iPhone, although some features are disabled. Amazon's Alexa has Android and iOS app versions, and Microsoft and Amazon are working together to bring Cortana to Amazon devices and Alexa to PCs. Apple has kept their assistant off of non-iOS devices, but Siri is available on all Apple devices, including iPhones, Macbooks, iPads, and the Apple Watch. As the voice assistant market stabilizes, it seems likely that there will be additional integration and that feature sets across the main voice assistants will become similar.

For the moment, Amazon is the dominant player in the field, due to launching a home product first with a large media library available out of the box. Google is building capacity, and the addition of a home-based speaker and integration into other Google products will drive their market share up. Apple may also become more of a contender with the release of Home Pod late in 2017 and the addition of more Apple-branded connected home products. Microsoft is not likely to gain much traction, as their share of the smartphone market is negligible and they lack a compelling home-based product.

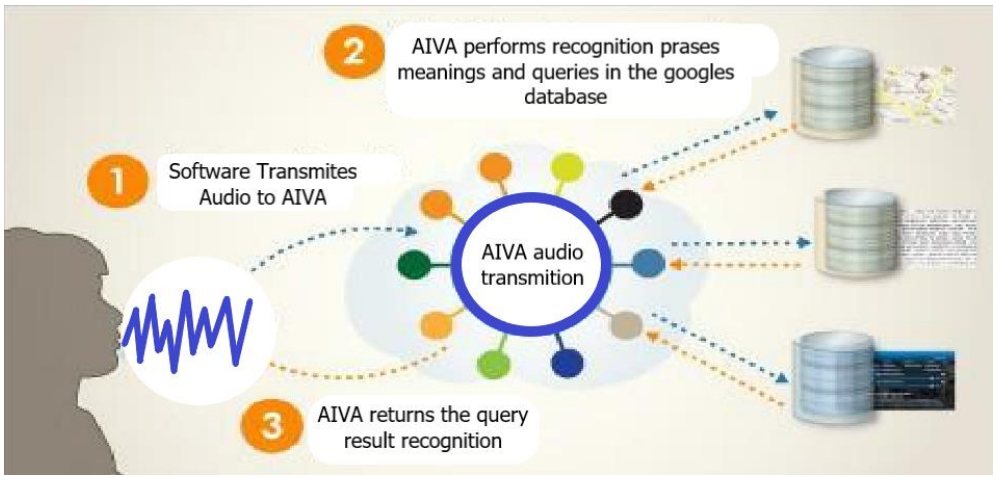


Fig 1.1: Interaction with AIVA

AIVA should be able to work with variety of commands. Opening and launching social media applications can be done easily just by using commands. Posting things on social media by voice command and can login\logout of social media websites by the commands. Help the user with finding results available on the Internet by just asking AIVA all the questions like what, who, why, when. To perform operations like deleting, creating new files on the local storage and also opening folders of the user. AIVA will also be able to play song and videos on both local storage and Internet by saying play command and the name of the song with it. User can any time get the latest climate condition around him\her just by asking AIVA what is the temperature now. It also works with google map. It can search any location or find place search places, AIVA is an AI assistant that automatically update itself with the trending things on the Internet.

1.3 Thesis Structure

The thesis which has been provided is based on the working of AIVA. The complete information on the working model, methodology, testing, experimental setup and results with future scope has been discussed. The basic idea of why the Virtual Assistant is needed and need of implementation is also discussed.

Chapter 2
LITERATURE REVIEW

2.1 Background study

A virtual assistant is a software agent that can perform tasks or services for an individual. Sometimes the term "Chatbot" is used to refer to virtual assistants generally or specifically those accessed by online chat (or in some cases online chat programs that are for entertainment and not useful purposes). As of 2017, the capabilities and usage of virtual assistants are expanding rapidly, with new products entering the market and a strong emphasis on voice user interfaces. An online poll in May 2017 found the most widely used in the US were:

This project mainly concerns the work on windows application development; request calling between different windows applications, the program will reference a lot of APIs from Google, Wikipedia, and windows development skills. Apart from the project itself, there is also some investigation works on the existed products in this area and the tendency of voice product, personal assistant developing. The investigation focus on how those ideas originated what functionalities and services they have how they provide these services to the customers test the product and related functions to get the architect, structure, logical algorithms of those products how they spread and promote the product in marketing; and how they refine and upgrade the products from different versions.

VIRTUAL ASSISTANT	USES
Apple Siri	34%
Google Assistant	19%
Amazon Alexa	6%
Microsoft Cortana	4%

Table 2.1: Uses of Virtual Assistant

Apple and Google have large installed bases of users on smartphones. Microsoft has a large installed base of Windows based personal computers, smartphones and smart speakers. Alexa has a large install base for smart speakers. Virtual assistants use Natural Language Processing to understand the command that human beings give them. A new battleground has emerged in the war of the “smart home”.

A most recent comprehensive textbook, "Fundamentals of Speaker Recognition" by Homayoon Beigi, is an in depth source for up to date details on the theory and practice. A good insight into the techniques used in the best modern systems can be gained by paying attention to government sponsored evaluations such as those organised by DARPA (the largest speech recognition-related project ongoing as of 2007 is the GALE project, which involves both speech recognition and translation components).

"Automatic Speech Recognition: A Deep Learning Approach" (Publisher: Springer) written by D. Yu and L. Deng published near the end of 2014, with highly mathematically-oriented technical detail on how deep learning methods are derived and implemented in modern speech recognition systems based on DNNs and related deep learning methods.This gave us an insight into the conversion algorithm used by Google.

- G. Bohouta, V. Z. Këpuska, "Comparing Speech Recognition Systems (Microsoft API Google API And CMU Sphinx)", Int. Journal of Engineering Research and Application 2017, 2017.

- B. Marr, The Amazing Ways Google Uses Deep Learning AI. Artificial intelligence (AI), sometimes called machine intelligence https://en.wikipedia.org/wiki/Artificial.

- Cortana Intelligence, Google Assistant, Apple Siri.

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

- "CMUSphnix Basic concepts of speech - Speech Recognition process". http://cmusphinx.sourceforge.netlwiki/tutorialconcepts.

- Books are available to read and learn about speech recognition ,these enabled us to see what happens beyond the code.

- Claudio Becchetti, Klucio Prina Ricotti.“Speech Recognition: Theory and C++ Implementation “ 2008 edition, In this book we learned how to write and implement the c++ code.

2.2 Related Work

- In instant messaging apps on both smartphones and via the Web, e.g. Facebook's M (virtual assistant) on both Facebook and Facebook Messenger apps or via the Web.

- Built into a mobile operating system (OS), as are Apple's Siri on iOS devices and BlackBerry Assistant on BlackBerry 10 devices, or into a desktop OS such as Cortana on Microsoft Windows OS.

- Built into a smartphone independent of the OS, as is Bixby on the Samsung Galaxy S8 and Note 8.

- On other mobile apps such as Google Allo.
- Within instant messaging platforms, assistants from specific organizations, such as Aeromexico's Aerobot on Facebook Messenger or Wechat Secretary on WeChat.
- Waibel, Hanazawa, Hinton, Shikano, Lang. (1989) "Phoneme recognition using time-delay neural networks. IEEE Transactions on Acoustics, Speech and Signal Processing."

Personal Assistant	Developer	Open Source	Always On	Smartphone App
Siri	Apple Inc.	No	Yes	Yes
Google Assistant	Google	No	Yes	Yes
Alexa	Amazon.com	No	Yes	Yes
Cortana	Microsoft	No	Yes	Yes
Bixby	Samsung	No	No	Yes
M	Facebook	No	-	-

Table 2.2: Other Voice Assistant

In April 2017, a software development kit (SDK) was released, allowing third-party developers to build their own hardware that can run Google Assistant. It has been integrated into Raspberry Pi, cars from Audi and Volvo, and smart home appliances, including fridges, washers, and ovens, from companies including iRobot, LG, General Electric, and D-Link. Google updated the SDK in December 2017 to add several features that only the Google Home smart speakers and Google Assistant smartphone apps had previously supported.

The features include:

- Letting third-party device makers incorporate their own "Actions on Google" commands for their respective products
- Incorporating text-based interactions and more languages

- Allowing users to set a precise geographic location for the device to enable improved location-specific queries.

On May 2, 2018, Google announced a new program on their blog that focuses on investing in the future of Google Assistant through early-stage startups. Their focus was to build an environment where developers could build richer experiences for their users. This includes startups that broaden Assistant's features, are building new hardware devices, or simply differentiating in different industries.

Fig 2.1: Other Voice Assistant

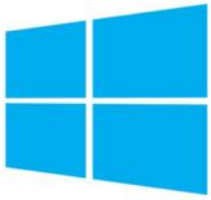
Chapter 3
OVERVIEW OF METHODOLOGY

3.1 Technologies Used

The technologies used for the development of the application and all the required computer languages to develop an application are listed below.

3.1.1 Windows OS:

Windows 10 Operating System is selected for development for a number of reasons. Windows operating system is familiar with each and every user. There is a huge community of Windows developers providing useful libraries for the development of the application and most important of all, Windows is backed by Microsoft.



Windows 10

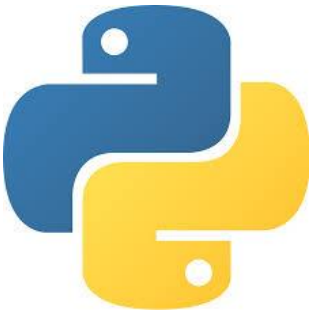
Windows 10 is a series of personal computer operating systems produced by Microsoft as part of its Windows NT family of operating systems. It is the successor to Windows 8.1, and was released to manufacturing on July 15, 2015, and broadly released for retail sale on July 29, 2015. Windows 10 receives new builds on an ongoing basis, which are available at no additional cost to users, in addition to additional test builds of Windows 10 which are available to Windows Insiders. Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as security patches, over their ten-year lifespan of extended support.

3.1.2 Python:

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Van Rossum led the language community until stepping down as leader in July 2018.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object oriented, imperative, functional and procedural. It also has a comprehensive standard library.



Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL), capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum's long influence on Python is reflected in the title given to him by the Python community: *Benevolent Dictator For Life* (BDFL) – a post from which he gave himself permanent vacation on July 12, 2018.

Python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. In January 2017, Google announced work on a Python 2.7 to Go transcompiler to improve performance under concurrent workloads.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation.

3.1.3 PyCharm:

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django.

The beta version was released in July 2010, with the 1.0 arriving 3 months later. Version 2.0 was released on 13 December 2011, version 3.0 on 24 September 2013.

PyCharm Community Edition, the open source version of PyCharm, became available on 22 October 2013.

PyCharm is cross-platform, with Windows, MacOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features, released under a proprietary license.

It competes mainly with a number of other Python-oriented IDEs, including Eclipse's PyDev, and the more broadly focused Komodo IDE.

3.1.4 AWS:

Amazon Web Services is a subsidiary of Amazon.com that provides on demanding cloud computing platforms to individuals, companies and governments, on a paid subscription basis. The technology allows subscribers to have at their disposal a virtual cluster of computers, available all the time, through the Internet. **Amazon Polly** a cloud service by Amazon Web Services, a subsidiary of Amazon.com that converts text into lifelike speech. It allows to create applications that talk, and build entirely new categories of speech-enabled products. It was launched in November 2016.

The AWS technology is implemented at server farms throughout the world, and maintained by the Amazon subsidiary. Fees are based on a combination of usage, the hardware/OS/software/networking features chosen by the subscriber, required availability, redundancy, security, and service options. Subscribers can pay for a single virtual AWS computer, a dedicated physical computer, or clusters of either. As part of the subscription agreement, Amazon provides security for subscribers' system. AWS operates from many global geographical regions including 6 in North America.

In 2017, AWS comprised more than 90 services spanning a wide range including computing, storage, networking, database, analytics, application services, deployment, management, mobile, developer tools, and tools for the Internet of Things. The most popular include Amazon Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (S3). Most services are not exposed directly to end users, but instead offer functionality through APIs for developers to use in their applications. Amazon Web Services' offerings are accessed over HTTP, using the REST architectural style and SOAP protocol.

3.2 Functionality Overview

This project is focusing on the windows development over the voice control (recognition, generate and analyze corresponding commands, intelligent responses automatically), Google products and relevant APIs (Google map, Google weather, Google search and etc), Wikipedia API and mobile device references ranging from Speech-To-Text, Text-To-Speech technology, Bluetooth headset support As all those functionalities and services for the project have been explained, the main structure and construction of the project has been basically illustrated with its goals.

This is an empirical qualitative study, based on reading above mentioned literature and testing their examples. Tests are made by programming according to books and online resources, with the explicit goal to find best practices and a more advanced understanding of Voice Assistant.

The system works on various keywords which has been provided in the code and will reply if those keywords are spoken by the user. We have used PyCharm instead of any other python editor version as it makes programming simple with its predefined libraries and ease of access to commands and syntaxes.

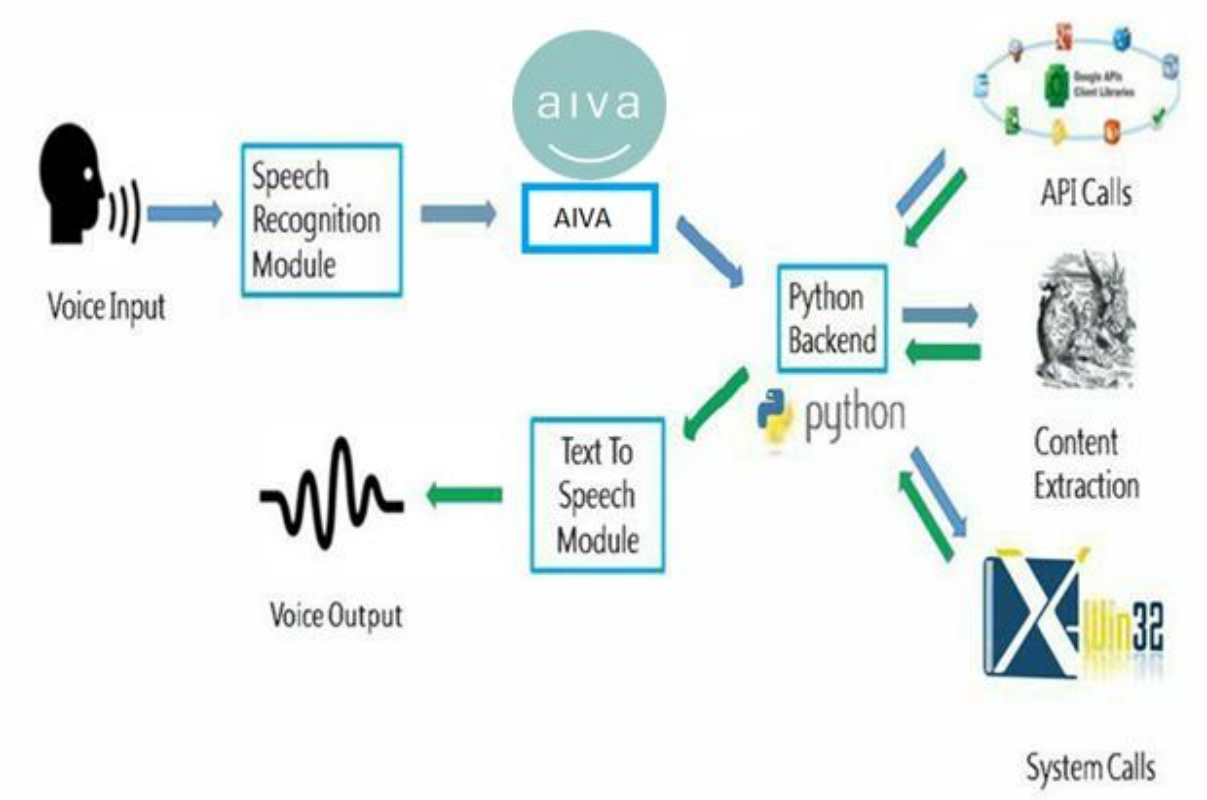


Fig 3.1: Workflow

We have used speech recognition here so that the spoken words are converted into text whenever the user says something. For this we have used Google Speech API, instead of reading out the reply from the text form we have used various mp3s which contain the message that has to be delivered to the user which have been implemented with the help of the play sound module and these text to speech conversion has been done with the help of amazon polly services.

We have also made use of boto3 module in order to make available amazon services. We have used dictionaries and lists in order to define mp3s and keywords so that the same command has two options to work on. Rive Script is a simple scripting language for giving intelligence to chat bots and other conversational entities. It's a plain text, line-based scripting language with goals of being simple to learn, quickly to type, and easy to read and maintain.

This is in contrast to other chat bot languages that require you to read and write ugly XML code (like AIML), or memorize lots of random symbols and "line noise" to write and read your code. In this we have implemented a simple chat bot code for which we have defined

well defined corpus. It fetches questions and answers based on the '+' and '-'. Whenever a user enters a query it uses + for matching the question and the – consists of answers.

The program should firstly be started on the PyCharm/Python Application; the initial mode of the program is Voice mode since this program aims at making a voice assistant program. After the program has been started, the user should have correct voice input "command/request" to make those functions work properly. And this program includes the functions and services of: mailing, location services, checking weather, Google searching engine, Wikipedia searching engine, Bluetooth headset support, help menu. The details below explain how those functions work and different possibilities while facing different commands.

Chapter 4
DESIGN AND IMPLEMENTATION

4.1 DESIGN

4.1.1 ARCHITECTURAL OVERVIEW

According to the overall description in the context, the purpose of the project is to develop an windows application that provides an intelligent voice assistant with the functionalities as mail reading, location services, checking weather, searching engine (Google, Wikipedia), Bluetooth headset support, help menu and local file handling.

The work started with analyzing the audio commands given by the user through microphone. The can be anything like reading of text/email, making calls, opening of apps, etc.

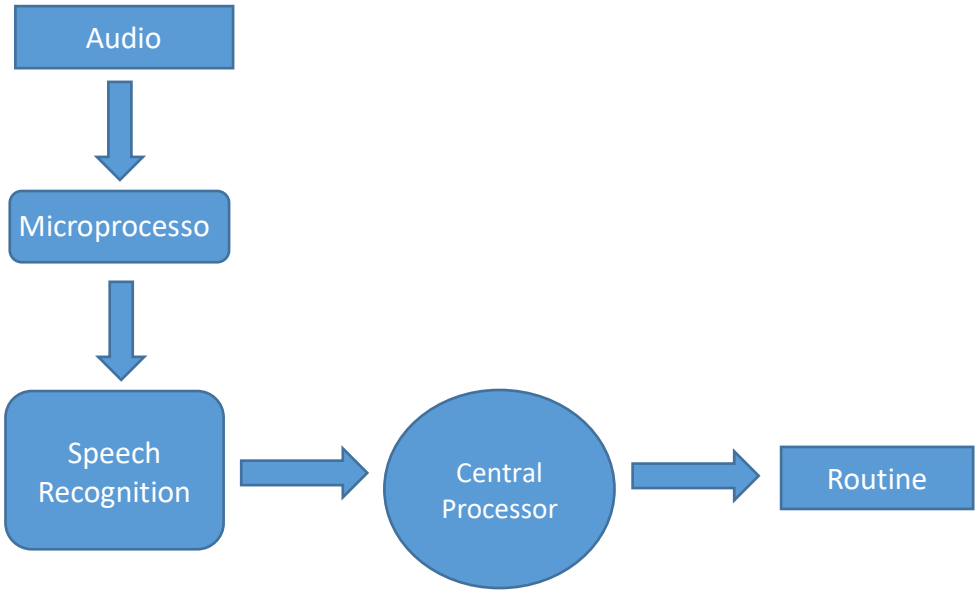


Fig.4.1: Basic Workflow

Fig.2 shows the workflow of the basic process of the voice assistant. The further process is carried out by Speech Recognition Module which is processed by the central processor. Speech recognition software can analyze the sounds you make by filtering what you say, digitizing it to a format it can "read", and then analyzing it for meaning. Then, based on algorithms and previous input, it can make a highly accurate educated guess as to what you are saying. It gets to know the speaker's use of language. Unsurprisingly, if the speech recognition software is only used by one person, it will be trained specifically for how that person talks.

4.1.2 MODULES OF THE PROJECT

Fig.3 shows the detailed process where central processor works according to the speech recognition module and give appropriate text. It becomes increasingly more complex when a device or software is geared towards multiple different markets around the world. This is

because engineers have to program the ability to understand infinite more variations; language, dialects, accents, phrasing.

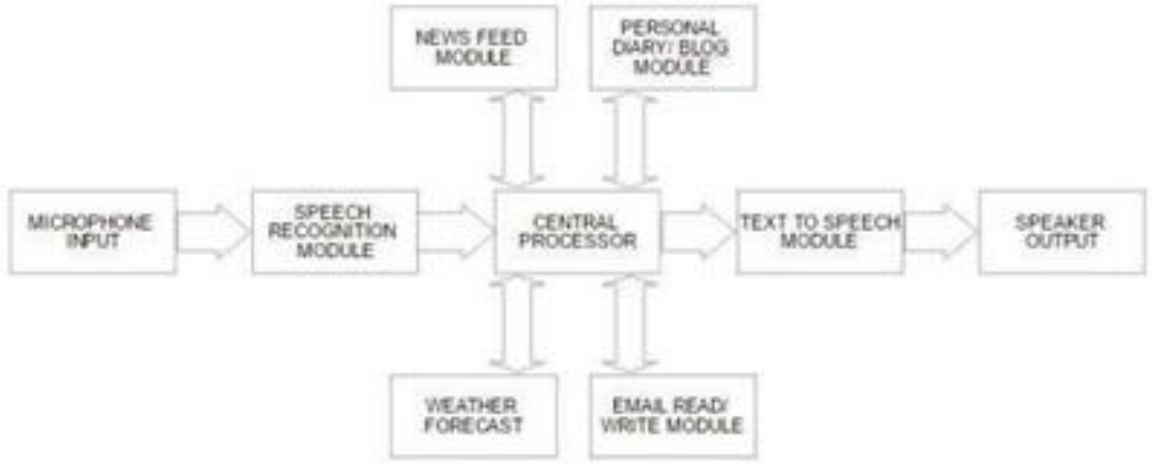


Fig.4.2: Block Diagram of module

But, the complexities don't stop there. Even with hundreds of hours of input, other factors can play a huge role in whether or not the software can understand you. Background noise can easily throw a speech recognition device off track. This is because it does not inherently have the ability to distinguish the ambient sounds it "hears" of a dog barking or a helicopter flying overhead, from your voice. Engineers have to program that ability into the device; they conduct data collection of these ambient sounds and "tell" the device to filter them out. Another factor is the way humans naturally shift the pitch of their voice to accommodate for noisy environments; speech recognition systems can be sensitive to these pitch changes.

Modules of the Project:-

A. Speech Recognition

The system uses Google's online speech recognition system for converting speech input to text. The speech input Users can obtain texts from the special corpora organized on the computer network server at the information center from the microphone is temporarily stored in the system which is then sent to Google cloud for speech recognition. The equivalent text is then received and fed to the central processor.

From the technology perspective, speech recognition has a long history with several waves of major innovations. Most recently, the field has benefited from advances in deep learning and big data. The advances are evidenced not only by the surge of academic papers published in the field, but more importantly by the worldwide industry adoption of a variety of deep learning methods in designing and deploying speech recognition systems.

The term *voice recognition* or *speaker identification* refers to identifying the speaker, rather than what they are saying. Recognizing the speaker can simplify the task of translating speech in systems that have been trained on a specific person's voice or it can be used to authenticate or verify the identity of a speaker as part of a security process.

B. Python Backend

The python backend get the output from the speech recognition module and then identifies whether the command or the speech output is an API Call, Context Extraction, and System Call. The output is then send back to the python backend to give the required output to the user.

It is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

C. API Calls

API stands for Application Programming Interface. An API is a software intermediary that allows two applications to talk to each other. In other words, an API is the messenger that delivers your request to the provider that you're requesting it from and then delivers the response back to you.

In computer programming, an **application programming interface** (**API**) is a set of subroutine definitions, communication protocols, and tools for building software. In general terms, it is a set of clearly defined methods of communication among various components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer.

An API may be for a web-based system, operating system, database system, computer hardware, or software library.

An API specification can take many forms, but often includes specifications for routines, data structures, object classes, variables, or remote calls. POSIX, Windows API and ASPI are examples of different forms of APIs. Documentation for the API usually is provided to facilitate usage and implementation.

D. Context Extraction

Context extraction (CE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. In most of the cases this activity concerns processing human language texts by means of natural language processing (NLP). Recent activities in multimedia document processing like automatic annotation and content extraction out of images/audio/video could be seen as context extraction TEST RESULTS.

In general objective, the ACE program is motivated by and addresses the same issues as the MUC program that preceded it. The ACE program, however, defines the research objectives in terms of the target objects (i.e., the entities, the relations, and the events) rather than in terms of the words in the text. For example, the so-called "named entity" task, as defined in MUC, is to identify those words (on the page) that are names of entities. In ACE, on the other hand, the corresponding task is to identify the entity so named. This is a different task, one that is more abstract and that involves inference more explicitly in producing an answer. In a real sense, the task is to detect things that "aren't there".

While the ACE program is directed toward extraction of information from audio and image sources in addition to pure text, the research effort is restricted to information extraction from text. The actual transduction of audio and image data into text is not part of the ACE research effort, although the processing of ASR and OCR output from such transducers is.

E. System Calls

In computing, a system call is the programmatic way in which a computer program requests a service from the kernel of the operating system it is executed on. This may include hardware-related services (for example, accessing a hard disk drive), creation and execution of new processes, and communication with integral kernel services such as process scheduling. System calls provide an essential interface between a process and the operating system.

In most systems, system calls can only be made from user space processes, while in some systems, OS/360 and successors for example, privileged system code also issues system calls.

System calls in most Unix-like systems are processed in kernel mode, which is accomplished by changing the processor execution mode to a more privileged one, but no *process* context switch is necessary – although a *privilege* context switch does occur. The hardware sees the world in terms of the execution mode according to the processor status register, and processes are an abstraction provided by the operating system. A system call does not generally require a context switch to another process; instead, it is processed in the context of whichever process invoked it.

F. Text-to-Speech Module

Text-to-Speech (TTS) refers to the ability of computers to read text aloud. A TTS Engine converts written text to a phonemic representation, then converts the phonemic representation to waveforms that can be output as sound. TTS engines with different languages, dialects and specialized vocabularies are available through third-party publishers.

Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood clearly. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written words on a home computer. Many computer operating systems have included speech synthesizers since the early 1990s.

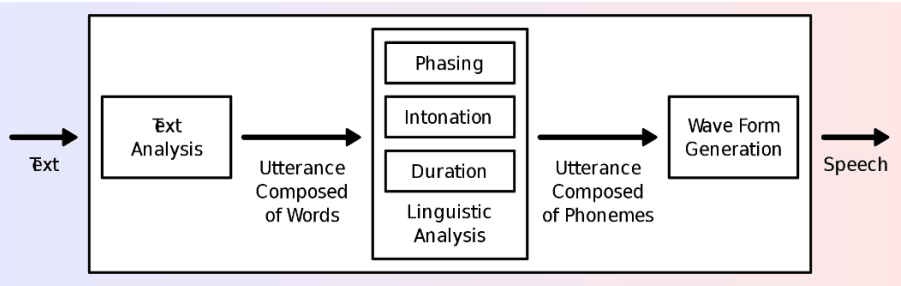


Fig.4.3: Block Diagram of Text to Speech

A text-to-speech system (or "engine") is composed of two parts:

A front-end and a back-end. The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called text normalization, pre-processing, or tokenization. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences.

The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end—often referred to as the synthesizer—then converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the target prosody (pitch contour, phoneme durations), which is then imposed on the output speech.

4.2 IMPLEMENTATION

Where am I" / "Show my current location", the program will present the current location of the user on the map. Navigation "How can I go to Pune" / "Navigation to Pune", the program will present the routes to "Pune" on the map with the highlighted route from the current location to Lund.

"What's the weather for today", the current weather condition for local place will be show. "What's the weather in Malmo", the current weather condition for Malmo will be show. Weather check other days: "What's the weather next few days", the forecast in next 4 days will be show

"Google China", the keyword 'Google' is detected and the result will be presented on the web browser by searching 'China' on Google. Project development and implementation as it has been previous stated, the program is mainly concerns with the techniques of windows development, different APIs for Google products. During the development, the developers did the same cycle in each phase of analyze requirements, construct design, implement the

solutions in pair programming mode and test the result. The development is carried out as its primary planning which guide the work process of how to work with the program, how much time should the each of the developers spent in every week, the rescores needed for developing and how to handle the problems while it came up. The project was efficiently completed under the development model and the resources we found in early time were really useful when implementing the program.

- Project usage & prospect, potential

The project is very useful and owns a large potential use in different industries. Although the program primary concerns more about how to do the personal assistant on windows application using the voice, the concept of voice recognition can be applied in different industries as in many situations it will be more convenient, save a lot of time and helpful especially for those who have difficulty in working with manual operations. Thus, the concept is only for programming the windows application.

For the program itself, it is a collection of 15 functions that are frequently used on a mobile phone. The user can enjoy different services within this platform. Therefore, it is easy to use with simple operation compared with the traditional working strategies which the user should well know how to work with the mobile phone. In addition, the program which works using the voice is helpful for those who prefer voice operation and those who have difficulty /disability with the manual operations. The primary objective of the program is to provide services using the voice, and it enables more people who can enjoy this program. The prospect of the program can be more applications or products developed using the voice control, and it could in some sense change the working forms that is totally different from the traditional form. As people can easily operate and have a lot of fun from it, it owns an enlightened prospect as CORTANA succeed in attracting people in the market.

4.3 Operation and Maintenance

- **Operation**

  - Mail Handling:

The user can send an email to the person in his contacts and with person's email address. He or she must have a command with the email keyword like "Mail", "Post" and a valid name; the email will be send if the person is found in the contacts. They are different forms to send the message; the list below shows the correct command can do the email sending. "Mail Bellis it will rain today", send an email to Bellis the content "it will rain today", the program will capture the keyword "Mail" and the content "it will rain today", and then the program will check the mobile contacts and get the email address corresponding to "Bellis" and send the message to "Bellis". "Post Mimy a boy is waiting for you" send an email to "Mimy" with the content "a boy is waiting for you".

  - Location services:

The user can use this service to locate the user's position or get the routes to the destination by giving the city name. There are different ways to locate the position or navigate to a specific city. The use must use the keywords "Search" and "my location" to let the application to know he or she wants to locate the current position. And the keywords "go to" and the name of the place to get the route to the destination.

"Search my current location", the program will present the current location of the user on the map.

Navigation “How can I go to KDK” / “Navigation to KDK”, the program will present the routes to “KDK” on the map with the highlighted route from the current location to KDK.

- Checking weather:

The user can use the application to check the weather for recent days in local place or specific location. He or she should say the keyword “weather”, then the user should notify the date that should be presented as “today/tomorrow/the day after tomorrow” if he or she wants to get the information about the other days otherwise the application will default set the date as today, and the user can also can choose to tell about the place name “in Malmo”, the application will check the weather belong to that place, otherwise the place will be set as locally.

Weather check today: “Search What's the weather for today”, the current weather condition for local place will be show. “What's the weather in Malmo”, the current weather condition for Malmo will be show.

Weather check other days: “Google What's the weather next few days”, the forecast in next 4 days will be show.

“Google What's the weather next few days in Malmo”, the forecast for Malmo in next 4 days will be show.

- Google searching engine:

The Google search engine is activated by the user commands which contain ‘Google’ or ‘Search’. By detecting the search keyword and search request, the Google search engine will returns the search result displayed on the browser on the mobile phone.

“Google China”, the keyword ‘Google’ is detected and the result will be presented on the web browser by searching ‘China’ on Google.

“Try to Google Java API”, the user can have the keyword Google in the middle of a request and the result of searching ‘Java API’ on Google will be displayed on the web browser.

“Search for apple”, the user can also use the keyword ‘search’ to do the Google search, this command will have the result of searching ‘apple’ on Google.

- Wikipedia searching engine:

Whenever the user wants to search any content in Wikipedia, it is possible to do in this program by having a command contain the keyword ‘Wikipedia’. If ‘Wikipedia’ is detected by the program, the program will automatically give the result by search the content after ‘Wikipedia’ in Wikipedia.

“Wikipedia Android”, the keyword ‘Wikipedia’ is detected, and the program will return the result by searching ‘Android’ on Wikipedia.

“Wikipedia true love”, the keyword ‘Wikipedia’ is detected, and the program will return the result by search the content after ‘Wikipedia’, which is ‘true love’ on Wikipedia.

- Translator:

The user should have the keyword ‘translate’ as the keywords to define this is a translate request, and ‘in’ as keyword to indicate the objective language. As the user have the command contains these keywords, the translator will return the result with the text in the objective language.

“Translate I love you in Chinese”, as ‘translate’ and ‘in’ are detected by the program, the program will call the translator with ‘I love you’ as the original text and Chinese as the objective language, the result will be the Chinese words of ‘I love you’.

"Translate How to say hello in Swedish", as 'Translate' and 'in' are detected by the program, the program will activate the translator with 'hello' as the original text and Swedish as the objective language, the result will be the Swedish text of 'hello'.

- **Maintenance**

After the program is completed, the program still needs long term maintenance to make it available and stable to execute. The program will be test after a certain period of time and debug each of the function and possible bugs, whenever a potential bug is detected; the program needs to be refined to a better design. Meanwhile, there will update and add more data to the database to increase the database capacity. Depending on the new keywords, responses, relevant data found that could be applied in this program; the database will always be improved and can handle more and more cases.

Chapter 5
EXPERIMENTAL SETUP AND RESULT

5.1 Screenshot of Result

- **Face of AIVA (Front-end):**

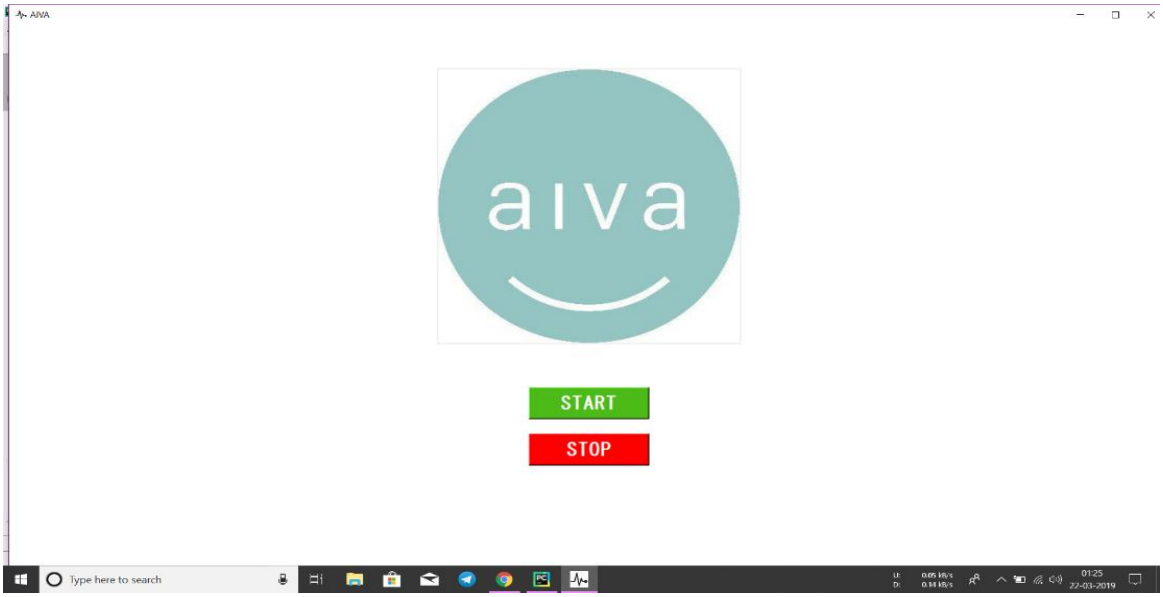


Fig. 5.1.1

- Front End:

The user can run the application and press the start the interaction with the AIVA. By pressing the start button user calls the main program of AIVA in which all the command are been defined. And the second stop button does the work of stopping the main program of AIVA and closing the main window of the software shown in fig 5.1.1.

- **Response on Hi/Hello :**

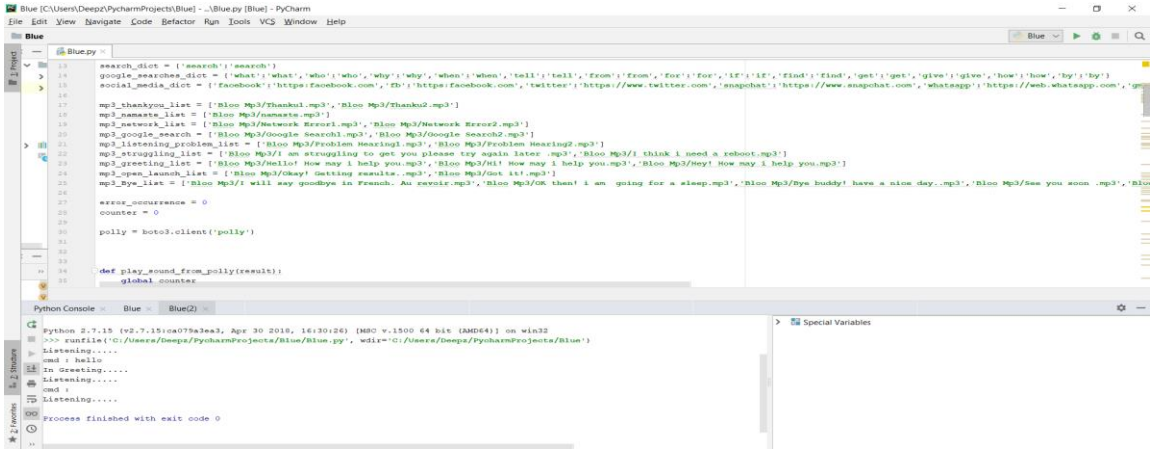


Fig. 5.1.2

o Greeting User:

After the main program of aiva is called by pressing the start button the first voice output given by the user to the aiva is “hi/hello/hey! This is your artificial intelligence AIVA”.

- **Response on File Handling :**

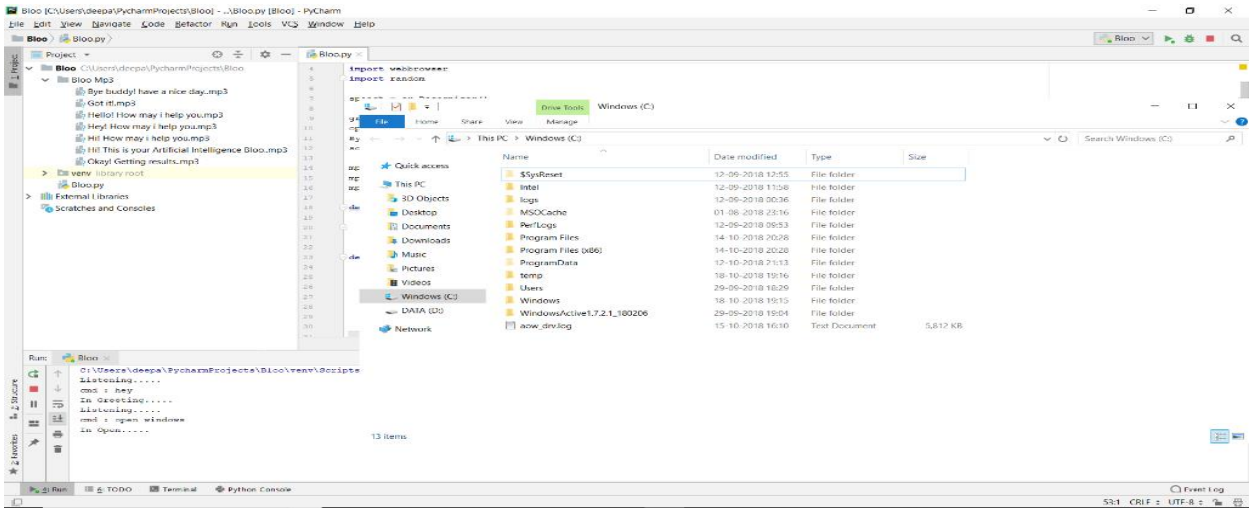


Fig. 5.1.3

o Local Storage:

To handle the local storage on the user’s pc or system the user need to give the command of “open, launch, start” in front of the file name, folder name, local disk storage name such as shown in the above fig 5.1.3 “open windows” in which ‘open’ is the command for opening the file or folder and ‘windows’ is the local disk name.

- **Response on Social Media :**

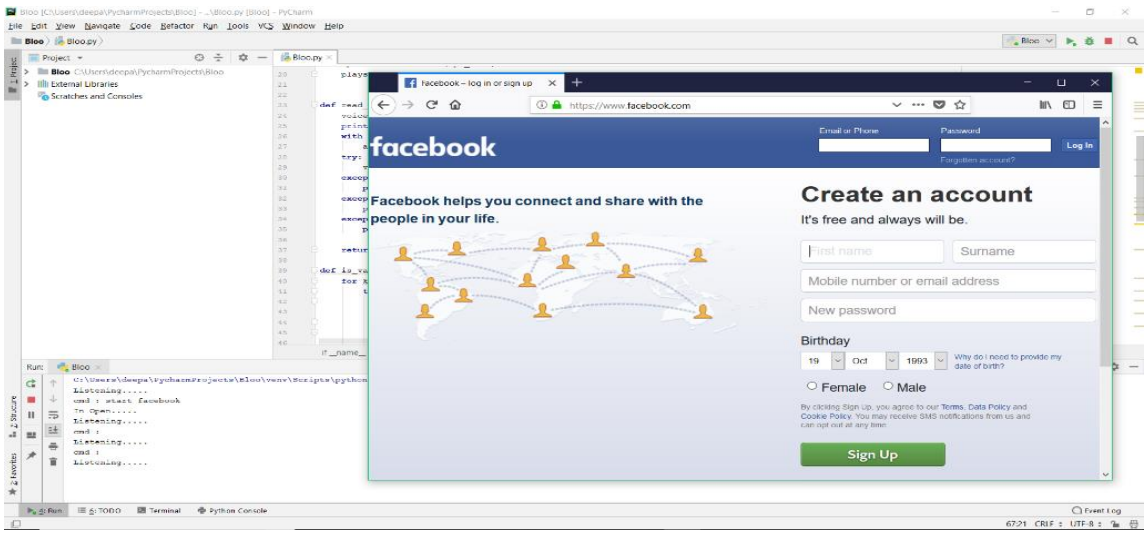


Fig. 5.1.4

o Social Media:

To Deal with social media like Facebook, Instagram, Twitter, and many more user just need to use the command as “Open”, “Start”, “Launch” and followed by the social media name that is required by the user shown in fig 5.1.4.

The user can also post the status on the Facebook by just giving the voice input to AIVA as “Post” command and the “Social Media” name with the post that need to be posted on the user Facebook profile. Such as “Post on my Facebook Good Moring” that is it.

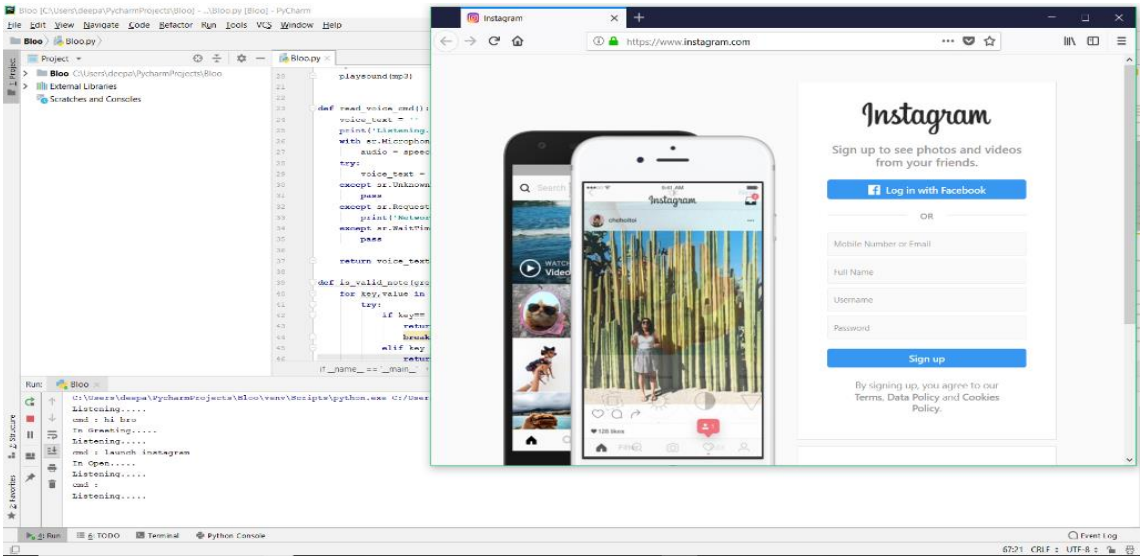


Fig. 5.1.5

- Social Media:

To Deal with social media like Facebook, Instagram, Twitter, and many more user just need to use the command as “Open”, “Start”, “Launch” and followed by the social media name that is required by the user shown in fig 5.1.5.

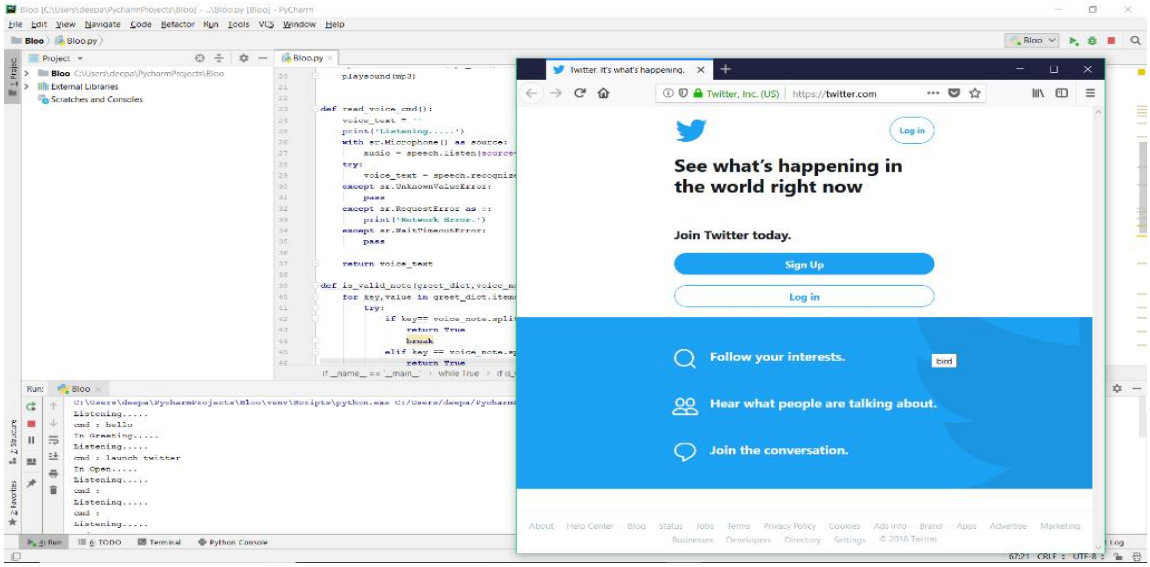


Fig. 5.1.6

- Social Media:

To Deal with social media like Facebook, Instagram, Twitter, and many more user just need to use the command as “Open”, “Start”, “Launch” and followed by the social media name that is required by the user shown in fig 5.1.6.

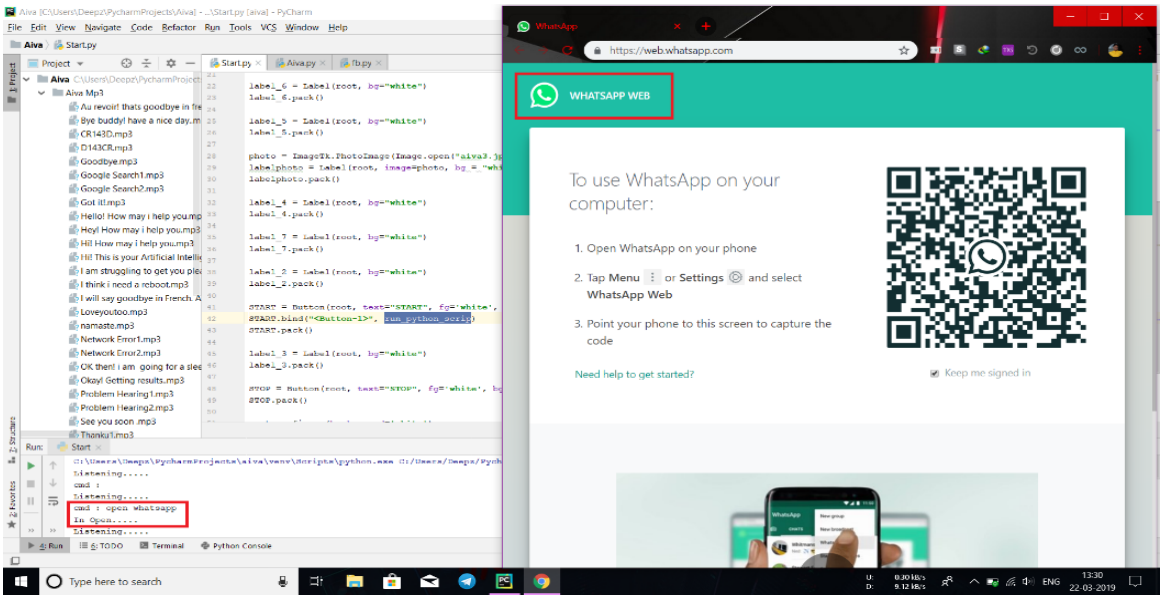


Fig. 5.1.7

○ Social Media:

To Deal with social media like Facebook, Instagram, Twitter, and many more user just need to use the command as “Open”, “Start”, “Launch” and followed by the social media name that is required by the user shown in fig 5.1.7.

- **Response on Mail Handing :**

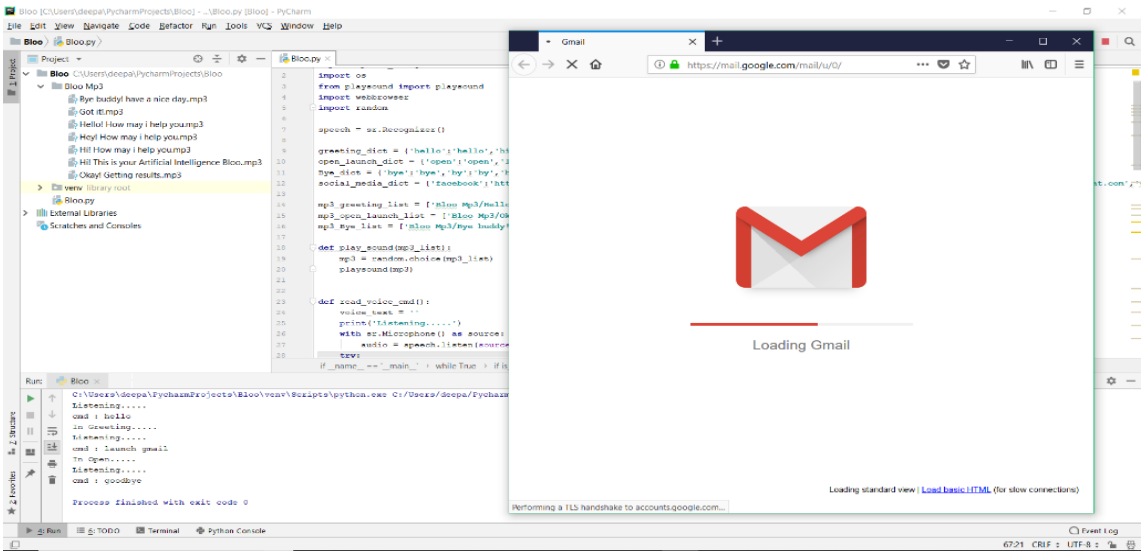


Fig. 5.1.8

○ Mail Handling:

The user can open e-mail, g-mail. He or she must have a command with the email keyword like “Launch G-mail/E-Mail”, “Open G-mail/E-mail”. They are different forms to read the message the above shows the correct command that can do the email reading. The user can use the command “open”, “Start”, “Launch” to launch the g-mail or email application by voice input and then the user can proceed the mail exchanging and broadcasting.

- **Response on Google Searches:**

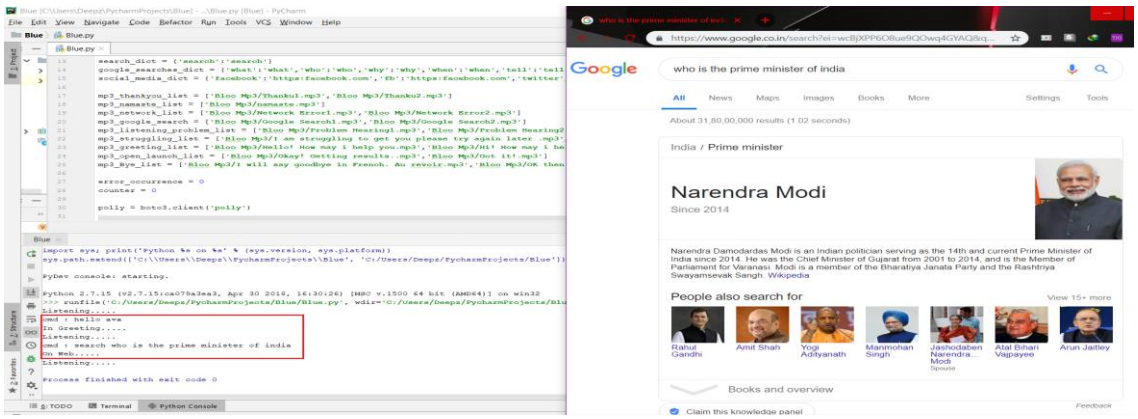


Fig. 5.1.9

○ Google searching engine:

The Google search engine is activated by the user commands which contain 'Google' or 'Search'. By detecting the search keyword and search request, the Google search engine will returns the search result displayed on the browser. "Google China", the keyword 'Google' is detected and the result will be presented on the web browser by searching 'China' on Google. "Try to Google Java API", the user can have the keyword Google in the middle of a request and the result of searching 'Java API' on Google will be displayed on the web browser. "Search for apple", the user can also use the keyword 'search' to do the Google search, this command will have the result of searching 'apple' on Google.

- **Response on Voice Google Searches:**

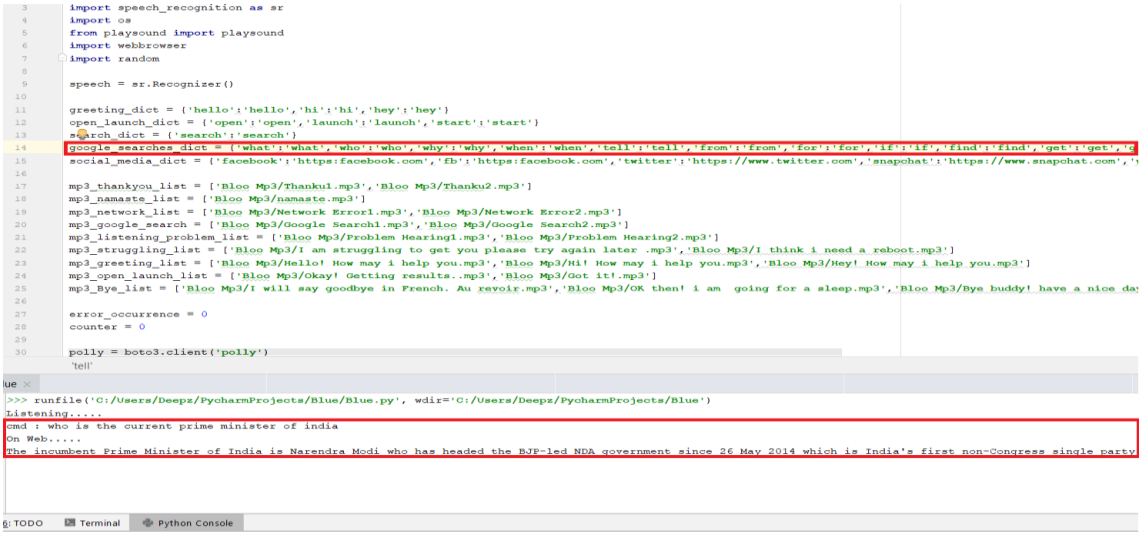


Fig. 5.1.10

○ Google searching engine:

The Google search engine is activated by the user commands which contain only the query. By detecting the search keyword and search request, the Google search engine will returns the search result by voice output.

- **Response on Wikipedia Searches:**

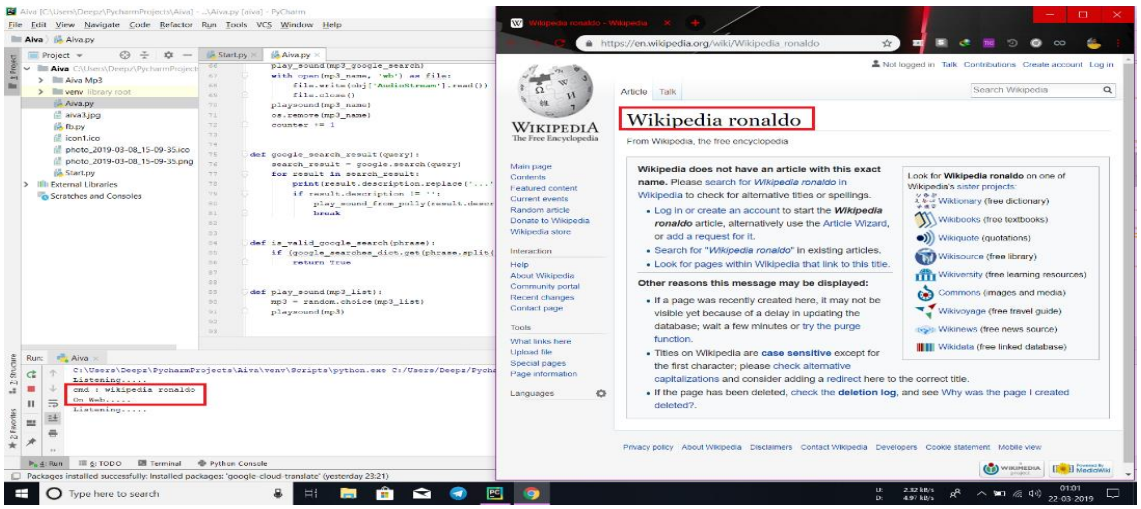


Fig. 5.1.11

○ Wikipedia search engine:

Whenever the user wants to search any content in Wikipedia, it is possible to do in this program by having a command contain the keyword 'Wikipedia'. If 'Wikipedia' is detected by the program, the program will automatically give the result by search the content in Wikipedia. "Wikipedia Ronaldo", the keyword 'Wikipedia' is detected, and the program will return the result by searching 'Ronaldo' on Wikipedia as in fig 5.1.11. "Wikipedia true love", the keyword 'Wikipedia' is detected, and the program will return the result by search the content after 'Wikipedia', which is 'true love' on Wikipedia shown in fig 5.1.11.

Wikipedia searching engine, the search engine enable the use to search anything on Wikipedia. By detecting the search keyword and search request, the Wikipedia search engine will returns the Wikipedia result displayed on the browser on the mobile phone.

- **Response on Opening Microsoft Applications:**

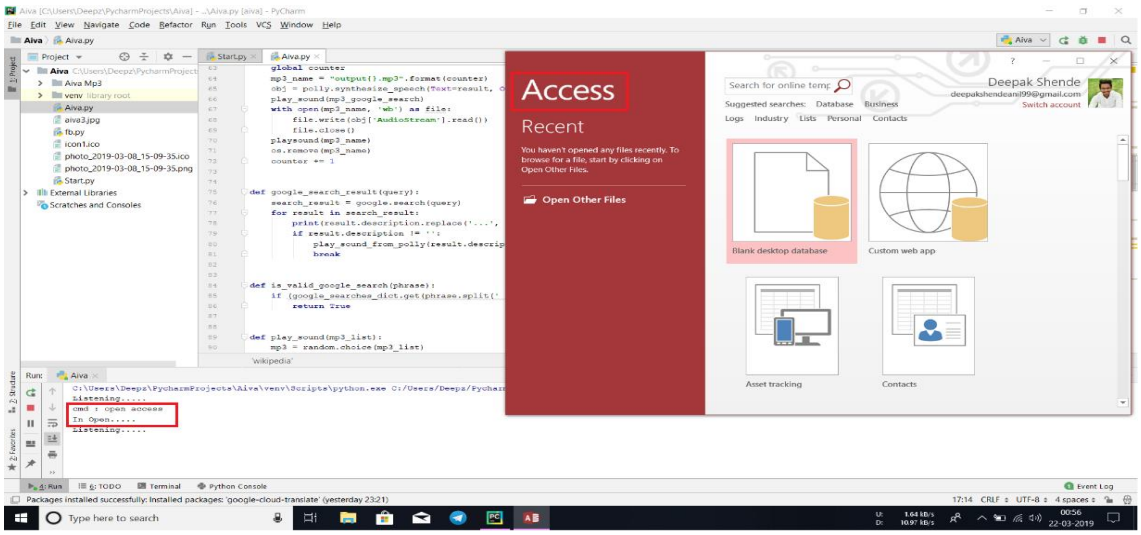


Fig. 5.1.12

- Microsoft Applications (Access):

To Deal with MS Access user just need to use the command as “Open”, “Start”, “Launch” and followed by the word “Access” in the voice input that is it the MS Access will open in a second as shown in fig 5.1.12.

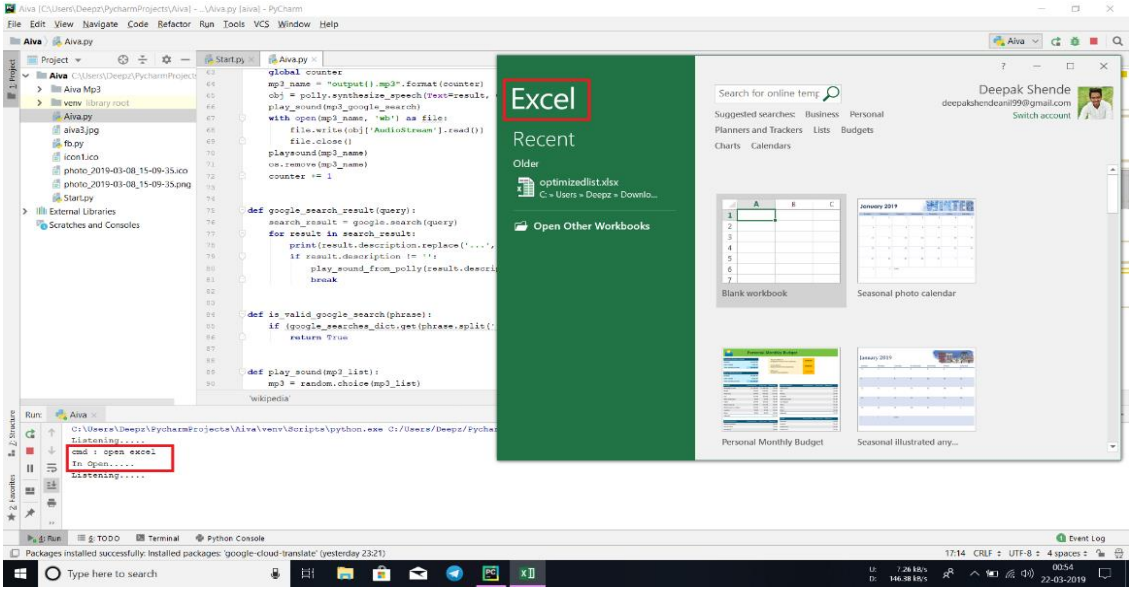


Fig. 5.1.13

- Microsoft Applications (Excel):

To Deal with MS Excel user just need to use the command as “Open”, “Start”, “Launch” and followed by the word “Excel” in the voice input that is it the MS Excel will open in a second as shown in fig 5.1.13.

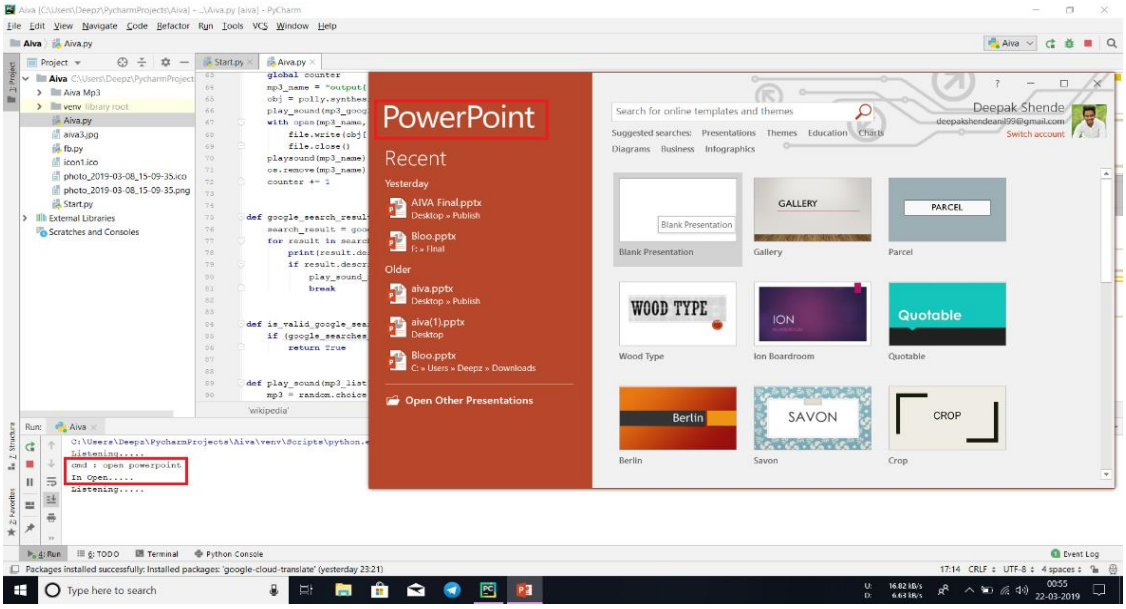


Fig. 5.1.14

- Microsoft Applications (PowerPoint):

To Deal with MS PowerPoint user just need to use the command as “Open”, “Start”, “Launch” and followed by the word “PowerPoint” in the voice input that is it the MS PowerPoint will open in a second as shown in fig 5.1.14.

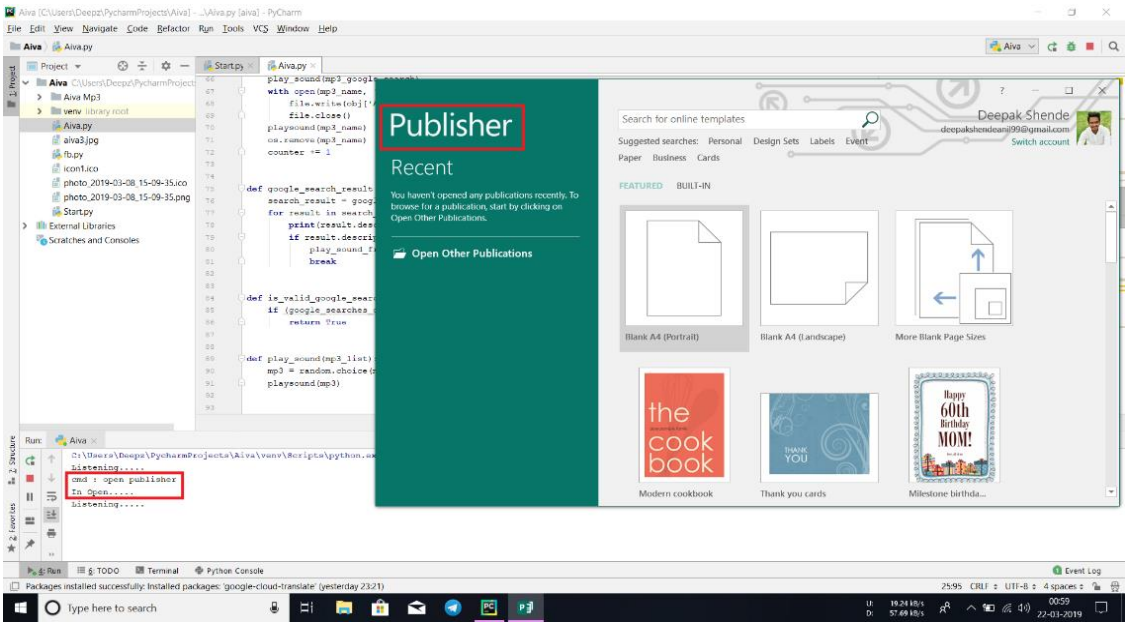


Fig. 5.1.15

- Microsoft Applications (Publisher):

To Deal with MS Publisher user just need to use the command as "Open", "Start", "Launch" and followed by the word "Publisher" in the voice input that is it the MS Publisher will open in a second as shown in fig 5.1.15.

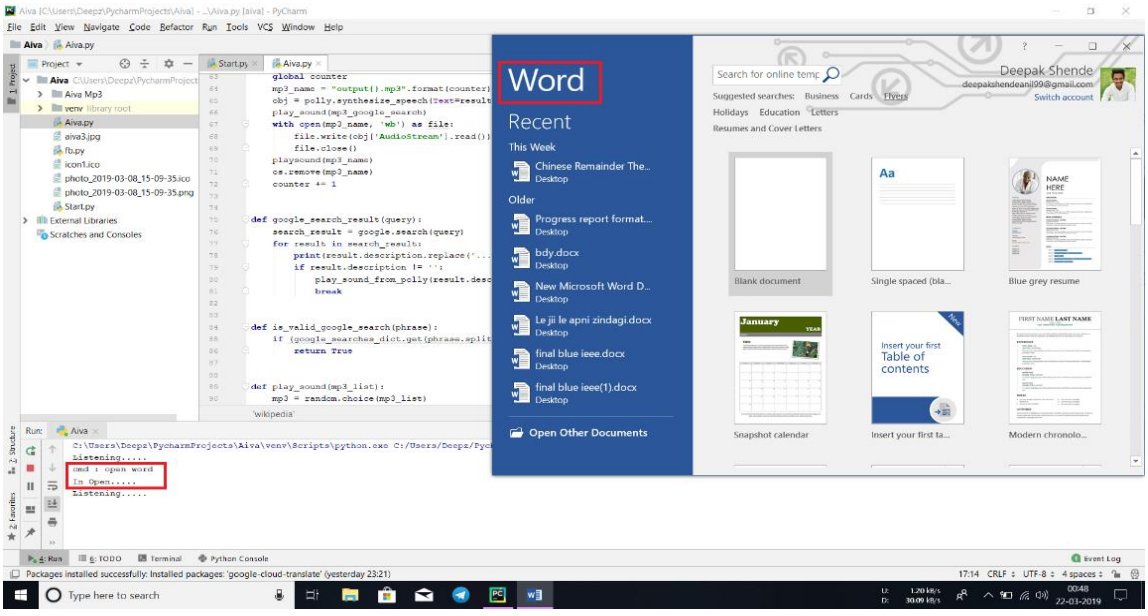


Fig. 5.1.16

- Microsoft Applications (Word):

To Deal with MS Word user just need to use the command as "Open", "Start", "Launch" and followed by the word "Word" in the voice input that is it the MS Word will open in a second as shown in fig 5.1.16.

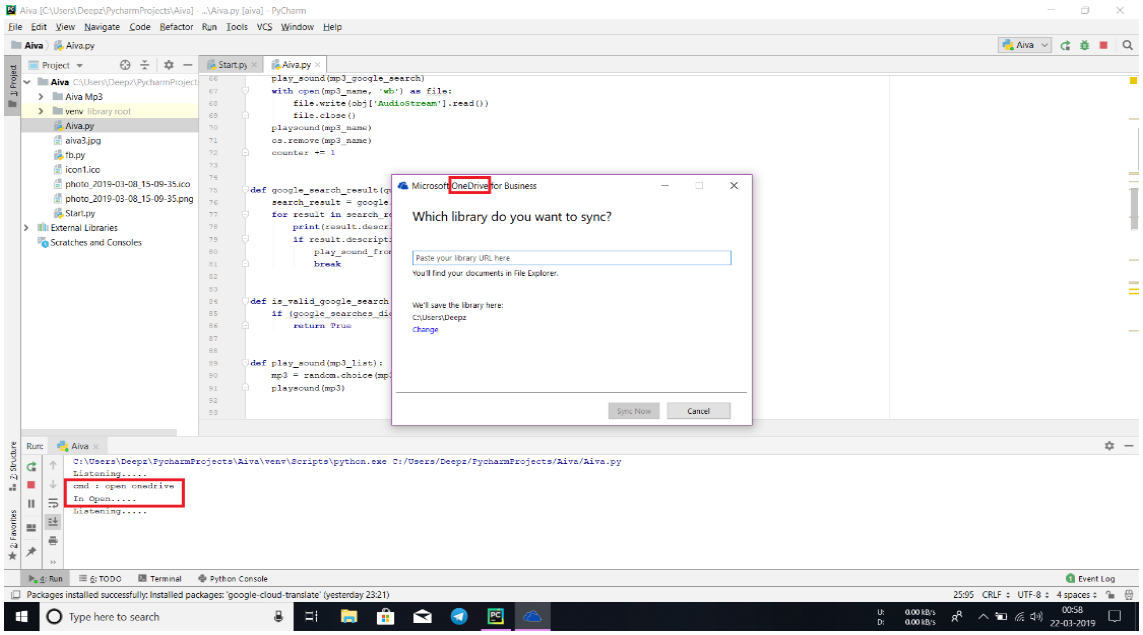


Fig. 5.1.17

- Microsoft Applications (OneDrive):

To Deal with MS OneDrive user just need to use the command as “Open”, “Start”, “Launch” and followed by the word “OneDrive” in the voice input that is it the MS OneDrive will open in a second as shown in fig 5.1.17.

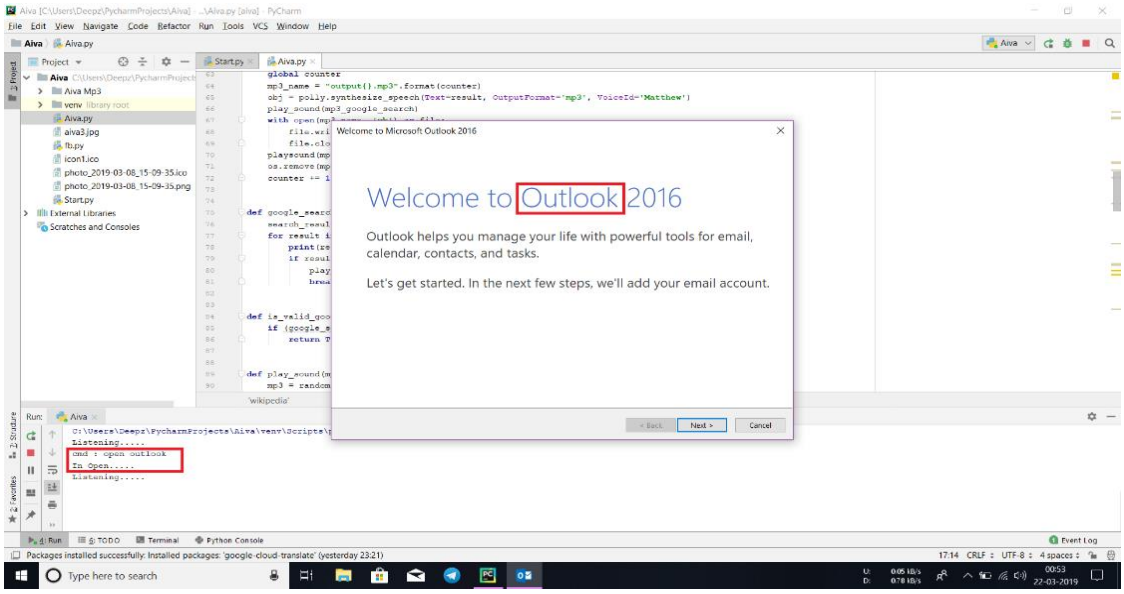


Fig. 5.1.18

- Microsoft Applications (Outlook):

To Deal with MS Outlook user just need to use the command as “Open”, “Start”, “Launch” and followed by the word “Outlook” in the voice input that is it the MS Outlook will open in a second as shown in fig 5.1.18.

- **Response on Location Works:**

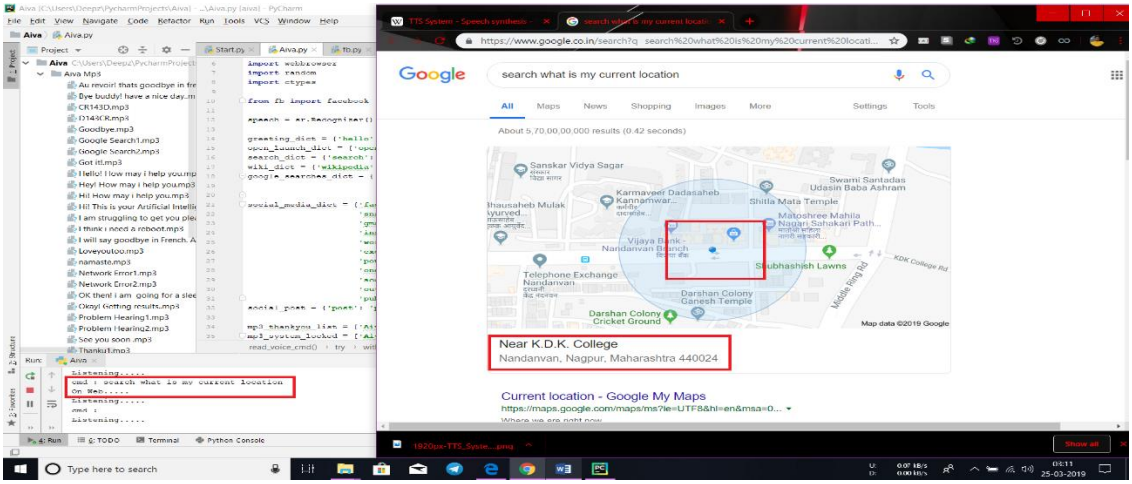


Fig. 5.1.19

○ Location services:

The user can use this service to locate the user's position or get the routes to the destination by giving the city name. There are different ways to locate the position or navigate to a specific city. The use must use the keywords "search" and "my location" to let the application to know he or she wants to locate the current position. And the keywords "go to" and the name of the place to get the route to the destination. Locate position "Search Where am I" / "Search my current location", the program will present the current location of the user on the map. Navigation "Search How can I go to Pune", the program will present the routes to "Pune" on the map with the highlighted route from the current location to Nagpur.

- **Response on Weather Conditions:**

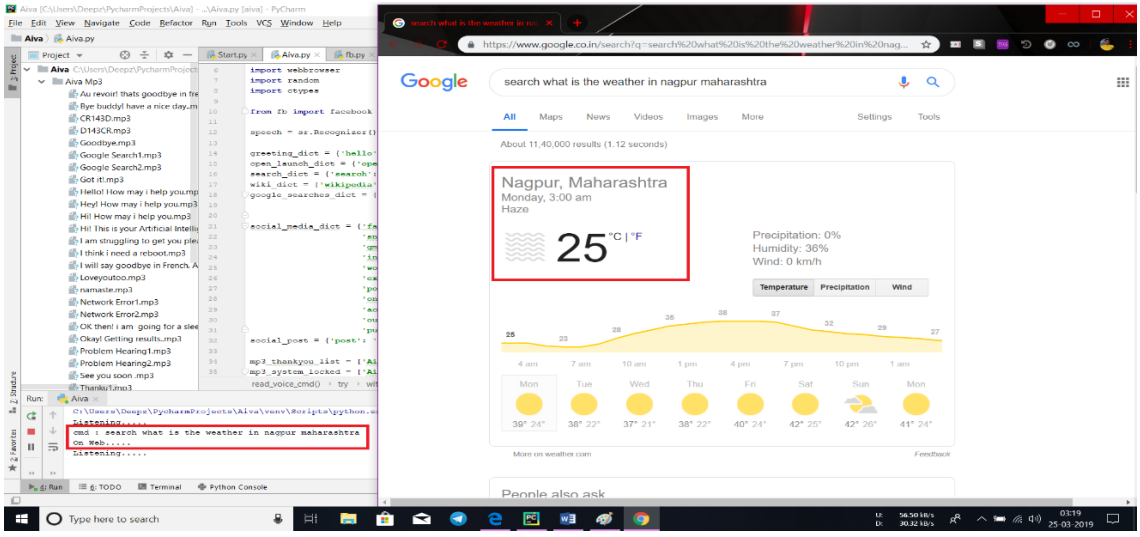


Fig. 5.1.20

○ Weather Services:

Weather Services works in two categories depending on the request. If it has been required to present the current location of the user, the location services check the GEO info by using the Google Map Service and give back the result as a map with the current location. If it has been required to provide the route trace from the current position to a specific city, the location service check the GEO info of both the origination and the destination, and then it

connect to the most possible and best fit weather forecast system then the output is shown to the user on the screen shown in fig 5.1.20.

- **Response on Translate:**

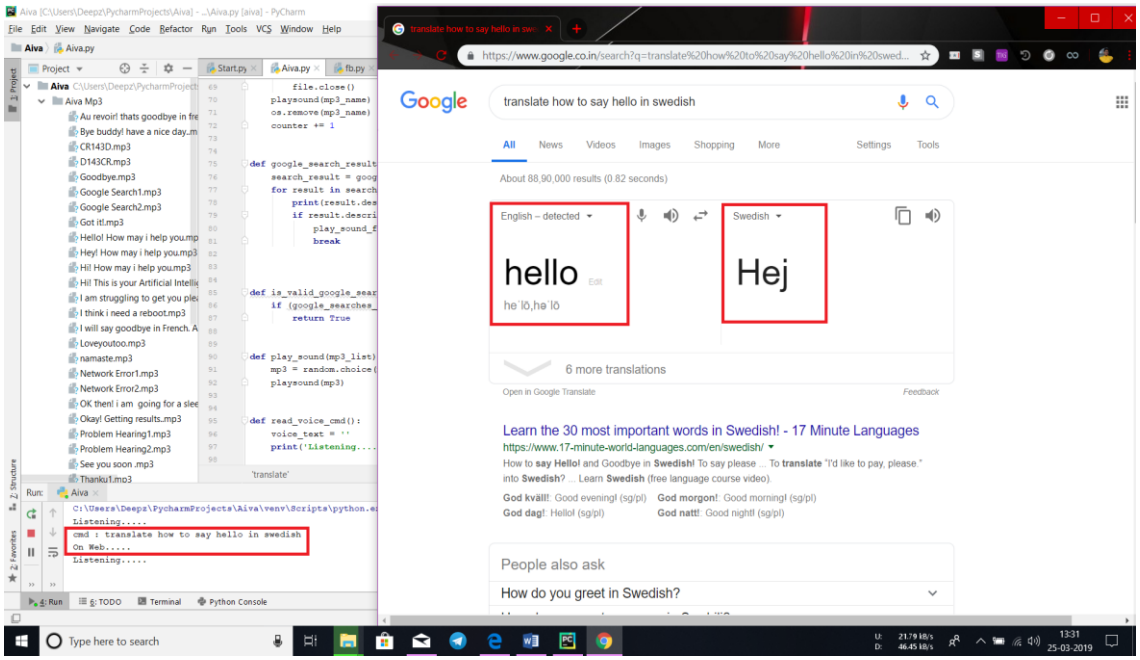


Fig. 5.1.21

- Translator:

The user should have the keyword ‘translate’ as the keywords to define this is a translate request, and ‘in’ as keyword to indicate the objective language. As the user have the command contains these keywords, the translator will return the result with the text in the objective language shown in fig 5.1.21.

“Translate I love you in Chinese”, as ‘translate’ and ‘in’ are detected by the program, the program will call the translator with ‘I love you’ as the original text and Chinese as the objective language, the result will be the Chinese words of ‘I love you’.

“Translate How to say hello in Swedish”, as ‘Translate’ and ‘in’ are detected by the program, the program will activate the translator with ‘hello’ as the original text and Swedish as the objective language, the result will be the Swedish text of ‘hello’.

5.2 Result

The Model and Flow Chart (see Figure-19) describes the develop process that include all the phases in the software development life cycle. This chart is well illustrating how the project is carried out and how the development was managed. The project started with the motivation and brain storm, repeatedly implement in the developing life cycle until the system has been fully constructed.

- Brain storm, the project start with the ideas from the brain storm. Here the basic ideas and design the primary concepts, prototype of the program have been obtained.

- While the ideas has been obtained, it has been analyzed which of them can be accomplished and make sure the structure of the project.
- According to the requirements that had been identified, collected all the resources and useful references from any channel, together with the programming skills and experiences, the design items were pointed out.
- Implement each individual design item based on the planning, structure and references.
- Test each single module that has been implemented and fix the possible bugs appear in the code implementation and make sure the functions are well constructed.
- Integrate all the individual sections to contribute to a complete system.
- Applied the best fit result strategies to get the best searched result.
- Debug the system and optimize the project from the possible aspects.
- Build the product and pack all the stuffs as a whole.

Chapter 6
CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

The more humanized the program is, more easier the user can use it. People should accept that even if developers constantly try to add more predefined commands, more responses to it, analyze and respond to the command more intelligently, the program will never be completely comprehensive and contain all the possible circumstances that the users meets. Nevertheless, the program will certainly be improved and be more user-friendly if there can be more readable commands, more humanized structure and more intelligent response. One such is Aiva it not only works on human commands but also give responses to the user on the basis of query being asked or the words spoken by the user such as opening tasks and operations .It is greeting the user the way user feels more comfortable and feels free to interact with the voice assistant. The application should also eliminate any kind of unnecessary manual work required in the user life of performing each and every task. The entire system works on the verbal input rather than the text one.

The complexity and accuracy of voice recognition technology and voice assistant software have grown exponentially in the last few years. Currently available voice assistant products from Apple, Amazon, Google, and Microsoft allow users to ask questions and issue commands to computers in natural language. There are many possible future uses of this technology, from home automation to translation to companionship and support for the elderly. However, there are also several problems with the currently available voice assistant products. Privacy and security controls will need to be improved before voice assistants should be used for anything that requires confidentiality. Librarians should monitor these products and be ready to provide assistance to their patrons with these devices. They should also explore the possibilities for providing library materials via voice assistants as the technology matures.

- Project development and implementation:

As it has been previous stated, the program is mainly concerns with the techniques of Android development, Java programming, Database management, Cloud computing, different APIs for Google products, Bing translate and etc. The program is developed by two developers and follows the extreme programming model. During the eight weeks development, the developers did the same cycle in each phase of analyze requirements, construct design, implement the solutions in pair programming mode and test the result. The development is carried out as its primary planning which guide the work process of how to work with the program, how much time should the each of the developers spent in every week, the rescores needed for developing and how to handle the problems while it came up. The project was efficiently completed under the development model and the resources we found in early time were really useful when implementing the program.

- Project usage & prospect, potential:

The project is very useful and owns a large potential use in different industries. Although the program primary concerns more about how to do the personal assistant on Android phone using the voice, the concept of voice recognition can be applied in different industries as in many situations it will be more convenient, save a lot of time and helpful especially for those who have difficulty in working with manual operations. Thus, the concept is only for programming the Android application. For the program itself, it is a collection of 15 functions that are frequently used on a mobile phone. The user can enjoy different services within this platform. Therefore, it is easy to use with simple operation compared with the traditional working strategies which the user should well know how to work with the mobile phone. In addition, the program which works using the voice is helpful for those who prefer voice operation and those who have difficulty /disability with the manual operations. The primary objective of the program is to provide services using the voice, and it enables more people who can enjoy this program. The prospect of the program can be more applications or products developed using the voice control, and it could in some sense change the working forms that is totally different from the traditional form. As people can easily operate and have a lot of fun from it, it owns an enlightened prospect as SIRI succeed in attracting people in the market.

- Project experience & teamwork:

Apart from the program, we as the developers have improved a lot from the degree project. It is quite different from what we previously experienced in the working model, volume of tasks, and the problems we have encountered. In conclusion, we have been improved a lot from the project development, and gained development experience as well as programming skills; the most important is work as a team for a long term, challenge development.

6.2 Future Scope

No program has a perfect design without any flaws; it is the same here in this program. Even though the program is completed with all the primary functions implemented and work properly, there are still many things that can be done with this program. As the future improvement, the potential work that can be implemented ranging from adding more functions to offering the user a more comprehensive, convenient program, refining the logic to make the program more humanized and easy to use, add more possible keywords, responses and data in this program, interface optimization and etc.

The possibility of added functionality required in making the assistant more accurate and fast while the interaction with the user .This project can be further improved by implementing the voice command in google search queries. Better speech recognition so that the user can get prompt output and applications such as locking pc or opening pc on the commands of the use Form Filling Functionality: Sometimes user are facing trouble while filling the form by their own each and every time so there might chances of adding the feature like saving user’s data and when in need the forms get automatically filled by simple commands.

Interface optimization, the interface can be further improved to make it nice to the users. Currently the interface design meets the basic requirement to present everything for this program, and thus users are able to interact with the program through this interface, but the

interface can always be optimized and more suitable constructed .Voice assistants have the potential to radically change how users interact with computers.

For many users, the ability to read and type is a barrier to accessing information. Voice assistants can bridge the information gap for those users. Recent research has shown that voice assistants can benefit dementia sufferers by providing an ever-present voice that can answer the same questions again and again without losing patience and offer encouragement when needed. For others, reading the instructions their physician provides can be difficult. Building these abilities into currently available consumer technologies would be much more cost effective than a purpose-built device, and many users would already be comfortable operating these devices. Voice assistants could also read books and other long-form documents to users. Although they still sound somewhat robotic, the vocal qualities of voice assistants are rapidly improving. Once they improve enough to not be off- putting, every book could be an audiobook.

Voice assistants also have the potential to revolutionize translation. Google recently announced a new set of earbuds that pair with its voice assistant for real-time voice translation. Users launch the application by asking the assistant to help them speak a language. The user's phone captures audio spoken by the other party, relays it to Google's translation servers, and plays the translated version in the user's earpiece. When the user wants their speech translated, he or she presses a button on the earpiece, which sends the audio to the server, where it is translated and relayed to the listener via the speaker on the user's handset. While Google's translation is notoriously error-prone, particularly for medical or colloquial language, the near-instant results should make it useful for simple conversations.

Voice assistants may be useful for library promotion and management as well. There are already tools available that allow libraries to create skills for voice assistants that list events at the library in local community calendars. Coding additional features that would allow patrons to hear hours and announcements, renew their items, hear what new items are available, or schedule a consultation with a librarian would be relatively simple. More complex tasks, such as searching databases and requesting interlibrary loans, are probably out of the scope of a tool that can only provide feedback via voice. Voice assistants could also easily be programmed to act as virtual tour guides in smaller gallery or exhibit spaces. Patrons could ask the assistant to tell them about an exhibit, and the assistant can read back prepared remarks. Libraries with a technology focus may want to consider lending these devices and providing basic training so that patrons can experiment with these devices in their homes as well.

REFERENCES

[1] Deepak Shende, Aishwarya Bhisikar, Monika Raghorte, Ria Umahiya “AI Based Voice Assistant Using Python” Journal of Emerging Technologies and Innovative Research (JETIR) February 2019, Volume 6, Issue 2.

[2] G. Bohouta, V. Z. Këpuska, "Comparing Speech Recognition Systems (Microsoft API Google API And CMU Sphinx)", Int. Journal of Engineering Research and Application 2017, 2017.

[3] B. Marr, The Amazing Ways Google Uses Deep Learning AI.

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

[5] Cortana Intelligence, Google Assistant, Apple Siri.

[6] Hill, J., Ford, W.R. and Farreras, I.G., 2015. Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations. Computers in Human Behavior, 49, pp.245-250.

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

[8] "CMUSphnix Basic concepts of speech - Speech Recognition process". http://cmusphinx.sourceforge.netlwiki/tutorialconcepts

[9] Huang, J., Zhou, M. and Yang, D., 2007, January. Extracting Chatbot Knowledge from Online Discussion Forums. In IJCAI(Vol. 7, pp. 423-428).

[10] Thakur, N., Hiwrale, A., Selote, S., Shinde, A. and Mahakalkar, N., Artificially Intelligent Chatbot.

[10] Mohasi, L. and Mashao, D., 2006. Text-to-Speech Technology in Human-Computer Interaction. In 5th Conference on Human Computer Interaction in Southern Africa, South Africa (CHISA 2006, ACM SIGHI) (pp. 79-84).

[11] Fryer, L.K. and Carpenter, R., 2006. Bots as language learning tools. Language Learning & Technology.

PUBLICATIONS

I, Deepak Shende with my project partners Ria Umahiya, Aishwarya Bhisikar & Monika Raghorte have presented the first paper entitled “AI Based Voice Assistant Using Python” in Journal of Emerging Technologies and Innovative Research (JETIR) February 2019, Volume 6, Issue 2.

I, Deepak Shende with my project partners RiaUmahiya, AishwaryaBhisikar& Monika Raghorte have presented the second paper entitled “AI Based Voice Assistant Using Speech Recognition” in SPARK February 2019.

APPENDIX A: HARD COPY OF PUBLICATION

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal
www.jetir.org | editor@jetir.org

Certificate of Publication

The Board of
Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)
Is hereby awarding this certificate to
Deepak Shende
In recognition of the publication of the paper entitled
AI Based Voice Assistant Using Python
Published In JETIR (www.JETIR.org) ISSN UGC Approved & 5.87 Impact Factor
Published in Volume 6 Issue 2 , February-2019

EDITOR
EDITOR IN CHIEF

JETIR1902381 Research Paper Weblink http://www.jetir.org/view?paper=JETIR1902381 Registration ID : 196993

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal
www.jetir.org | editor@jetir.org

Certificate of Publication

The Board of
Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)
Is hereby awarding this certificate to
Ria Umahiya
In recognition of the publication of the paper entitled
AI Based Voice Assistant Using Python
Published In JETIR (www.JETIR.org) ISSN UGC Approved & 5.87 Impact Factor
Published in Volume 6 Issue 2 , February-2019

EDITOR
EDITOR IN CHIEF

JETIR1902381 Research Paper Weblink http://www.jetir.org/view?paper=JETIR1902381 Registration ID : 196993

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal

www.jetir.org | editor@jetir.org

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

Monika Raghorte

In recognition of the publication of the paper entitled

AI Based Voice Assistant Using Python

Published In JETIR (www.JETIR.org) ISSN UGC Approved & 5.87 Impact Factor

Published in Volume 6 Issue 2 , February-2019

EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR1902381 Research Paper Weblink http://www.jetir.org/view?paper=JETIR1902381 Registration ID : 196993

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal

www.jetir.org | editor@jetir.org

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

Aishwarya Bhisikar

In recognition of the publication of the paper entitled

AI Based Voice Assistant Using Python

Published In JETIR (www.JETIR.org) ISSN UGC Approved & 5.87 Impact Factor

Published in Volume 6 Issue 2 , February-2019

EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR1902381 Research Paper Weblink http://www.jetir.org/view?paper=JETIR1902381 Registration ID : 196993

"Service to the Society through Quality Technical Education"

K.D.K. COLLEGE OF ENGINEERING, NAGPUR

KDKCE'S

SPARK'19

XV National Conference on Emerging Technology Trends in Engineering & Project Competition

Coordinated by Department of Electronics Engineering

Certificate

This is to Certify that Mr./Ms. Deepak Shende

from KDK College of Engineering has Coordinated / participated / Presented

a Paper Titled AI-based voice assitant using speech recognition in National Conference, Spark-2019 held on 23rd March, 2019

Computer Technology in group.

He /She has secured — position in Paper Presentation/Project Competition.

Prof. (Mrs.) J.S. Gawai, Co-Convener | Dr. P.D. Khandait, Convener & HOD Etrx. Dept. | Dr. A.M. Badar, (Vice Principal, KDKCE) | Dr. D.P. Singh, (Principal, KDKCE)

"Service to the Society through Quality Technical Education"

K.D.K. COLLEGE OF ENGINEERING, NAGPUR

KDKCE'S

SPARK'19

XV National Conference on Emerging Technology Trends in Engineering & Project Competition

Coordinated by Department of Electronics Engineering

Certificate

This is to Certify that Mr./Ms. Aishwarya Bhisikar

from KDK College of Engineering has Coordinated / participated / Presented

a Paper Titled AI-based voice assitant using speech recognition in National Conference, Spark-2019 held on 23rd March, 2019

Computer Technology in group.

He /She has secured — position in Paper Presentation/Project Competition.

Prof. (Mrs.) J.S. Gawai, Co-Convener | Dr. P.D. Khandait, Convener & HOD Etrx. Dept. | Dr. A.M. Badar, (Vice Principal, KDKCE) | Dr. D.P. Singh, (Principal, KDKCE)

"Service to the Society through Quality Technical Education"

K.D.K. COLLEGE OF ENGINEERING, NAGPUR

KDKCE'S

SPARK'19

XV National Conference on Emerging Technology Trends in Engineering & Project Competition

Coordinated by Department of Electronics Engineering

Certificate

This is to Certify that Mr./Ms. Monika P. Raghorte

from KDK College of Engineering has Coordinated / participated / Presented

a Paper Titled AI-based voice assitant using speech recognition in National Conference, Spark-2019 held on 23rd March, 2019

Computer Technology in group.

He /She has secured - position in Paper Presentation/Project Competition.

Prof. (Mrs.) J.S. Gawai, Co-Convener | Dr. P.D. Khandait, Convener & HOD Etrx. Dept. | Dr. A.M. Badar, (Vice Principal, KDKCE) | Dr. D.P. Singh, (Principal, KDKCE)

"Service to the Society through Quality Technical Education"

K.D.K. COLLEGE OF ENGINEERING, NAGPUR

KDKCE'S

SPARK'19

XV National Conference on Emerging Technology Trends in Engineering & Project Competition

Coordinated by Department of Electronics Engineering

Certificate

This is to Certify that Mr./Ms. Ria Umahiya

from KDK college of Engineering has Coordinated / participated / Presented

a Paper Titled AI-based voice assistant using python speech recognition in National Conference, Spark-2019 held on 23rd March, 2019

Computer Technology in group.

He /She has secured - position in Paper Presentation/Project Competition.

Prof. (Mrs.) J.S. Gawai, Co-Convener | Dr. P.D. Khandait, Convener & HOD Etrx. Dept. | Dr. A.M. Badar, (Vice Principal, KDKCE) | Dr. D.P. Singh, (Principal, KDKCE)

APPENDIX B: CODING

- **Start.py**

```
import os
from tkinter import *
from PIL import Image, ImageTk

root = Tk()
python_scripts = {'here': 'Aiva.py'}


def run_python_scrip(voice_note):
    for key, value in python_scripts.items():
        os.system('python {}'.format(value))


root.geometry('1600x900')
root.title("AIVA")
root.iconbitmap(r'icon1.ico')
label_1 = Label(root, bg="white")
label_1.pack()
label_6 = Label(root, bg="white")
label_6.pack()
label_5 = Label(root, bg="white")
label_5.pack()
photo = ImageTk.PhotoImage(Image.open("aiva3.jpg"))
labelphoto = Label(root, image=photo, bg = "white")
labelphoto.pack()
label_4 = Label(root, bg="white")
label_4.pack()
label_7 = Label(root, bg="white")
label_7.pack()
label_2 = Label(root, bg="white")
label_2.pack()

START = Button(root, text="START", fg='white', bg='#228B22', font=("gothic", 20, 'bold'),
width=10)
START.bind("<Button-1>", run_python_scrip)
START.pack()

label_3 = Label(root, bg="white")
label_3.pack()

STOP = Button(root, text="STOP", fg='white', bg='red', font=("gothic", 20, 'bold'), width=10,
command=root.destroy)
STOP.pack()

root.configure(background='white')
root.mainloop()
```

- **Aiva.py**

```
from google import google
import boto3
import speech_recognition as sr
import os
from playsound import playsound
import webbrowser
import random
import ctypes
from fb import facebook

speech = sr.Recognizer()

greeting_dict = {'hello': 'hello', 'hi': 'hi', 'hey': 'hey'}
open_launch_dict = {'open': 'open', 'launch': 'launch', 'start': 'start'}
search_dict = {'search': 'search','google':'google','navigate':'navigate','translate':'translate'}
wiki_dict = {'wikipedia': 'wikipedia'}
google_searches_dict = {'what': 'what', 'who': 'who', 'why': 'why', 'when': 'when', 'tell': 'tell',
'from': 'from',
                'for': 'for', 'if': 'if', 'find': 'find', 'get': 'get', 'give': 'give', 'how': 'how',
                'by': 'by'}
social_media_dict = {'facebook': 'https:facebook.com', 'fb': 'https:facebook.com', 'twitter':
'https://www.twitter.com',
                'snapchat': 'https://www.snapchat.com', 'whatsapp': 'https://web.whatsapp.com',
                'gmail': 'https://mail.google.com', 'linkedin': 'https://in.linkedin.com',
                'instagram': 'https://www.instagram.com', 'insta': 'https://www.instagram.com',
                'word': 'C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Word 2016',
                'excel': 'C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Excel 2016',
                'powerpoint': 'C:\ProgramData\Microsoft\Windows\Start
Menu\Programs\PowerPoint 2016',
                'onedrive': 'C:\ProgramData\Microsoft\Windows\Start
Menu\Programs\OneDrive for Business',
                'access': 'C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Access
2016',
                'outlook': 'C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Outlook
2016',
                'publisher': 'C:\ProgramData\Microsoft\Windows\Start
Menu\Programs\Publisher 2016'}
social_post = {'post': 'post'}

mp3_thankyou_list = ['Aiva Mp3/Thanku1.mp3', 'Aiva Mp3/Thanku2.mp3']
mp3_system_locked = ['Aiva Mp3/Your system is locked. You need to login to
continue.mp3',
                'Aiva Mp3/Your windows is locked. You need to login to countinue.mp3']
mp3_namaste_list = ['Aiva Mp3/namaste.mp3']
mp3_network_list = ['Aiva Mp3/Network Error1.mp3', 'Aiva Mp3/Network Error2.mp3']
mp3_google_search = ['Aiva Mp3/Google Search1.mp3', 'Aiva Mp3/Google Search2.mp3']
mp3_listening_problem_list = ['Aiva Mp3/Problem Hearing1.mp3', 'Aiva Mp3/Problem
Hearing2.mp3']
```

```
mp3_struggling_list = ['Aiva Mp3/I am struggling to get you please try again later .mp3',
                       'Aiva Mp3/I think i need a reboot.mp3']
mp3_greeting_list = ['Aiva Mp3/Hello! How may i help you.mp3', 'Aiva Mp3/Hi! How may i
help you.mp3',
                     'Aiva Mp3/Hey! How may i help you.mp3']
mp3_open_launch_list = ['Aiva Mp3/Okay! Getting results..mp3', 'Aiva Mp3/Got it!.mp3']
mp3_Bye_list = ['Aiva Mp3/I will say goodbye in French. Au revoir.mp3', 'Aiva Mp3/OK
then! i am  going for a sleep.mp3',
                'Aiva Mp3/Bye buddy! have a nice day..mp3', 'Aiva Mp3/See you soon .mp3',
                'Aiva Mp3/Au revoir! thats goodbye in french.mp3', 'Aiva Mp3/Goodbye.mp3']

error_occurrence = 0
counter = 0

polly = boto3.client('polly')


def to_be_posted(voice_note):
    for key in social_media_dict.keys():
        if key in voice_note:
            return key


def play_sound_from_polly(result, is_google=False):
    global counter
    mp3_name = "output{}.mp3".format(counter)
    obj = polly.synthesize_speech(Text=result, OutputFormat='mp3', VoiceId='Matthew')
    play_sound(mp3_google_search)
    with open(mp3_name, 'wb') as file:
        file.write(obj['AudioStream'].read())
        file.close()
    playsound(mp3_name)
    os.remove(mp3_name)
    counter += 1


def google_search_result(query):
    search_result = google.search(query)
    for result in search_result:
        print(result.description.replace('...', '').rsplit('.', 3)[0])
        if result.description != '':
            play_sound_from_polly(result.description.replace('...', '').rsplit('.', 3)[0])
            break



def is_valid_google_search(phrase):
    if (google_searches_dict.get(phrase.split(' ')[0]) == phrase.split(' ')[0]):
        return True
```

```
def play_sound(mp3_list):
    mp3 = random.choice(mp3_list)
    playsound(mp3)


def read_voice_cmd():
    voice_text = ''
    print('Listening.....')

    global error_occurrence

    try:
        with sr.Microphone() as source:
            audio = speech.listen(source=source, timeout=10, phrase_time_limit=5)
        voice_text = speech.recognize_google(audio)
    except sr.UnknownValueError:

        if error_occurrence == 0:
            play_sound(mp3_listening_problem_list)
            error_occurrence += 1
        elif error_occurrence == 1:
            play_sound(mp3_struggling_list)
            error_occurrence += 1

    except sr.RequestError as e:
        print('Network Error.')
        play_sound(mp3_network_list)
    except sr.WaitTimeoutError:
        if error_occurrence == 0:
            play_sound(mp3_listening_problem_list)
            error_occurrence += 1
        elif error_occurrence == 1:
            play_sound(mp3_struggling_list)
            error_occurrence += 1

    return voice_text


def is_valid_note(greet_dict, voice_note):
    for key, value in greet_dict.items():
        try:
            if value == voice_note.split(' ')[0]:
                return True
                break
            elif key == voice_note.split(' ')[1]:
                return True
                break
        except IndexError:
            pass
```

```
    return False


if __name__ == '__main__':

    playsound('Aiva Mp3/Hi! This is your Artificial Intelligence AVA.mp3')

    while True:

        voice_note = read_voice_cmd().lower()
        print('cmd : {}'.format(voice_note))

        if is_valid_note(greeting_dict, voice_note):
            print('In Greeting.....')
            play_sound(mp3_greeting_list)
            continue

        elif is_valid_note(open_launch_dict, voice_note):
            print('In Open.....')
            play_sound(mp3_open_launch_list)
            if (is_valid_note(social_media_dict, voice_note)):
                key = voice_note.split(' ')[1]
                webbrowser.open(social_media_dict.get(key))
            else:
                os.system('explorer c:\\"{}"'.format(voice_note.replace('open ', '').replace('launch ',
'')))
            continue

        elif is_valid_google_search(voice_note):
            print('On Web.....')
            playsound('Aiva Mp3/Got it!.mp3')
            # webbrowser.open('https://www.google.co.in/search?q={}'.format(voice_note))
            google_search_result(voice_note)
            continue

        elif is_valid_note(search_dict, voice_note):
            print('On Web.....')
            playsound('Aiva Mp3/Got it!.mp3')
            webbrowser.open('https://www.google.co.in/search?q={}'.format(voice_note))
            # google_search_result(voice_note)
            continue

        elif is_valid_note(wiki_dict, voice_note):
            print('On Web.....')
            playsound('Aiva Mp3/Got it!.mp3')
            webbrowser.open('https://en.wikipedia.org/wiki/{}'.format(voice_note))
            # google_wikipedia_result(voice_note)
            continue
```

```
    elif 'namaste' in voice_note:
        play_sound(mp3_namaste_list)
        continue

    elif 'post' in voice_note:
        media = to_be_posted(voice_note)
        if media == 'facebook':
            facebook().post_on_wall(voice_note.split(media + ' ')[1].capitalize())
            play_sound_from_polly('The Post is liive now.')
            continue

    elif 'lock' in voice_note:
        for value in ['pc', 'system', 'window', 'windows']:
            ctypes.windll.user32.LockWorkStation()
            play_sound(mp3_system_locked)
            exit()

    elif 'thank you' in voice_note:
        play_sound(mp3_thankyou_list)
        continue

    elif 'goodbye' in voice_note:
        play_sound(mp3_Bye_list)
        exit()
```

- **Fb.py**

```
import facebook

access_token = "User’s FB Access code"

fb = facebook.GraphAPI(access_token)

class facebook:
    def post_on_wall(self, message):
        fb.put_object('me', 'feed', message=message)
```

PROGRAM OUTCOMES

1. **Engineering knowledge:** Apply knowledge of mathematics, science and Computer Technology fundamentals in the design and development of software to Work in IT industry and other related areas.

2. **Problem analysis**: Take up responsibility in identify, Formulate, review literature and analyze complex engineering problems related to CT and reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

3. **Design/Development of solutions:** Design solutions for complex engineering problems related to CT and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety and the cultural societal and environmental considerations.

4. **Conduct Investigations of Complex problems:** Use research–based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to investigate the complex problem and suggest probable solution.

5. **Modern Tool Usage:** Create, Select and apply appropriate techniques, resources and modern engineering and IT tools to computer technology related complex engineering activities with an understanding of the limitations.

6. **The Engineer and Society:** Apply Reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the CT professional engineering practice.

7. **Environment and Sustainability:** Understand the impact of the CT professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply Ethical Principles and commit to professional ethics , responsibilities and norms of the engineering practice.

9. **Individual and Team Work:** Execute the project development individually and also in a teamwork.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large such as able to comprehend and with write effective reports and design documentation, make effective presentations and give and receive clear instructions. Participate in academic activities such as group discussions, seminars, conferences and co curricular and extracurricular events.

11. **Project Management and Finance:** Demonstrate knowledge and understanding of the engineering management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multi disciplinary environments.

12. **Life-Long Learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning the broadest context of technological change.

PROGRAM OUTCOMES MAPPING:

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
Mapping	3	3	3	3	3	2	2	3	3	3	2	3