Voice Assistant using Artificial Intelligence

Ms. Preethi G
Information Technology
PSG Polytechnic College
Coimbatore, India

Mr. Thiruppugal S
Information Technology
PSG Polytechnic College
Coimbatore, India

Mr. Abishek K
Information Technology
PSG Polytechnic College
Coimbatore, India

Mr. Vishwaa D A
Information Technology
PSG Polytechnic College
Coimbatore, India

Abstract: **Voice assistants are software agents that can interpret human speech and respond via synthesized voices. Apple's Siri, Amazon's Alexa, Microsoft's Crotona, and Google's Assistant are the most popular voice assistants and are embedded in smart phones 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 paper will explore the basic workings and common features of today's voice assistants**

Keywords: Voice Assistants; chat bots; Static voice assistant; Speech Recognition; Text-to-Speech.

I. INTRODUCTION

A voice assistant can be a digital assistant that uses human voice, language process algorithms, and synthesis to pay attention to particular voice commands and come applicable information or perform particular functions as appealed by the user supported commands, commonly known as intents, tell by the user, voice assistants will come applicable information by hearing for particular keywords and filtering out the close noise. While voice assistants may be completely a software system primarily builds on and ready to combine into all devices, some assistants are sketched individually for every unique device applications, like the Amazon Alexa clock. Now a day, voice assistants are combined into some of the devices we intend to use on a daily, like cell phones, computers, and good speakers. The main motive of this project is developed for physically challenged person. In this project we have developed a static voice assistant using python which will perform operations like copy and paste files from one location to another location, to send a message to the users mobile also order pizza and mobile using voice commands. The mass adoption of AI in users' everyday lives is additionally refueling the shift towards voice. One of the most popular voice assistants are Siri, from Apple, Amazon Echo, which responds to the name of Alexa from Amazon, Cortana from Microsoft,Google Assistant from Google, and the recently appeared intelligent assistant under the name AIVA. This paper presents a brief introduction to the architecture and construction of voice assistants.

II. LITERATURE SURVEY

The field of voice-based assistants has observed major advancements and innovations. The main reason behind such rapid growth in this field is its demand in devices like smartwatches or fitness bands, speakers, Bluetooth earphones, mobile phones, laptop or desktop, television, etc. Most of the smart devices that are being brought in the market today have built in voice assistants. The amount of data that is generated nowadays is huge and in order to make our assistant good enough to tackle these enormous amounts of data and give better results we should incorporate our assistants with machine learning and train our devices according to their uses. Along with machine learning other technologies which are equally important are IoT, NLP, Big data access management. The use of voice assistants can ease out a lot of tasks for us. Just give voice command input to the system and all tasks will be completed by the assistant starting from converting your speech command to text command then taking out the keywords from the command and execute queries based on those keywords. In the paper “Speech recognition using flat models” by Patrick Nguyen and all, a novel direct modelling approach for speech recognition is being brought forward which eases out the measure of consistency in the sentences spoken. They have termed this approach as Flat Direct Model (FDM). They did not follow the conventional Markov model and their model is not sequential. Using their approach, a key problem of defining features has been solved. Moreover, the template-based features improved the sentence error rate by 3% absolute over the baseline [2].

Again, in the paper “On the track of Artificial Intelligence: Learning with Intelligent Personal Assistant” by Nil Goksel and all, the potential use of intelligent personal assistants (IPAs) which use advanced computing technologies and Natural Language Processing (NLP) for learning is being examined. Basically, they have reviewed the working system of IPAs within the scope of AI [4].

The application of voice assistants has been taken to some higher level in the paper “Smart Home Using Internet of Things” by Keerthana S and all where they have discussed how the application of smart assistants can lead to developing a smart home system using Wireless Fidelity (Wi-Fi) and Internet of Things. They have used CC3200MCU that has in-built Wi-Fi modules and temperature sensors. The temperature that is sensed by the temperature sensor is sent to the microcontroller unit (MCU) which is then posted to a server and using that data the status of electronic equipment like fan, light etc is monitored and controlled [5].

The application of voice assistants has been beautifully discussed in the paper “An Intelligent Voice Assistant Using Android Platform" by Sutar Shekhar and all where they have stressed on the fact that mobile users can perform their daily task using voice commands instead of typing things or using keys on mobiles. They have also used a prediction technology that will make recommendations based on the user activity [6].

The incorporation of natural language processing (NLP) in voice assistants is really necessary which will also lead to the creation of a trendsetting assistant. These factors have been the key focus of the paper “An Intelligent Chatbot using Natural Language Processing” by Rishabh Shah and all. They have discussed how NLP can help to make assistants smart enough to understand commands in any native language and thus does not prevent any part of the society form enjoying its perks [7].

We also studied the systems developed by Google Text To Speech – Electric Hook Up (GTTS-EHU) for Query-by-example Spoken Term Detection (QbE-STD) and Spoken Term Detection (STD) tasks of the Albayzin 2018 Search on Speech Evaluation. For representing audio documents and spoken queries Stacked bottleneck features (sBNF) are used as frame level acoustic representation. Spoken queries are synthesized, average of sBNF representations is taken and then the average query is used for Qbe-STD [8].

We have seen the integration of technologies like gTTS, AIML (Artificial Intelligence Mark-up Language) in the paper “JARVIS: An interpretation of AIML with integration of gTTS and Python” by TanveeGawand and all where they have adopted the dynamic base Python pyttsx which is a text to speech conversion library in python and unlike alternative libraries, it works offline [9].

The main focus of voice assistants should be to reduce the use of input devices and this fact has been the key point of discussion in the paper

III. PROPOSED SYSTEM

In this proposed concept effective way of implementing a Personal voice assistant, Speech Recognition library has many in-built functions, that will let the assistant understand the command given by user and the response will be sent back to user in voice, with Text to Speech functions. When assistant captures the voice command given by user, the under lying algorithms will convert the voice into text.

Proposed Architecture

The system design consists of
1. Taking the input as speech patterns through microphone.
2. Audio data recognition and conversion into text.
3. Comparing the input with predefined commands.
4. Giving the desired output.

The initial phase includes the data being taken in as speech patterns from the microphone. In the second phase the collected data is worked over and transformed into textual data using NLP. In the next step, this resulting string the data is manipulated through Python Script to finalize the required output process. In the last phase, the produced output is presented either in the form of text or converted from text to speech using TTS.

Features The System shall be developed to offer the following features:

1) It keeps listening continuously in inaction and wakes up into action when called with a particular predetermined functionality.

2) Browsing through the web based on the individual’s spoken parameters and then issuing a desired output through audio and at the same time it will print the output on the screen.

IV. SYSTEM ARCHITECTURE

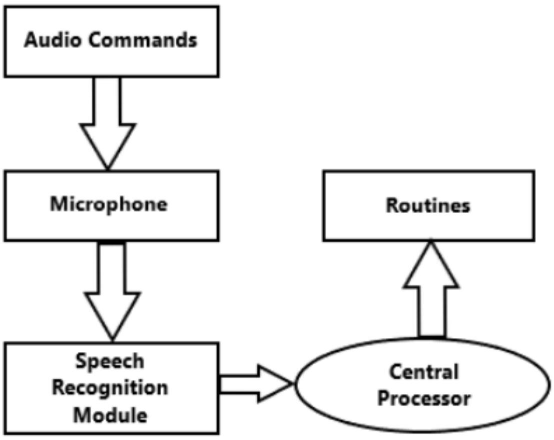


Fig 1 Block diagram of the voice assistant

Basic Workflow

The figure below shows the workflow of the main method of voice assistant. Speech recognition is used to convert speech input to text. This text is then sent to the processor, which determines the character of the command and calls the appropriate script for execution. But that's not the only complexity. No matter how many hours of input, another factor plays a big role in whether a package notices you. Ground noise simply removes the speech recognition device from the target. This may be due to the inability to essentially distinguish between the bark of a dog or the sound near hearing that a helicopter is flying overhead from your voice.

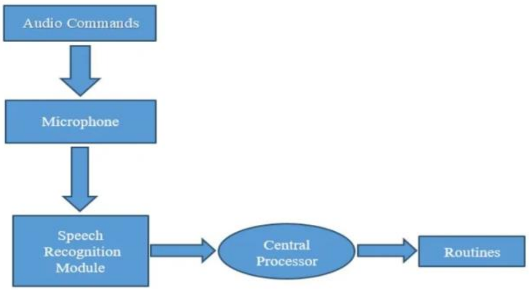


Fig 2 Basic Workflow of the voice assistant

Detailed Workflow

Voice assistants such as Siri, Google Voice, and Bixby are already available on our phones. According to a recent NPR study, around one in every six Americans already has a smart speaker in their home, such as the Amazon Echo or Google Home, and sales are growing at the same rate as smart phone sales a decade ago. At work, though, the voice revolution may still seem a long way off. The move toward open workspaces is one deterrent: nobody wants to be that obnoxious idiot who can't stop ranting at his virtual assistant.

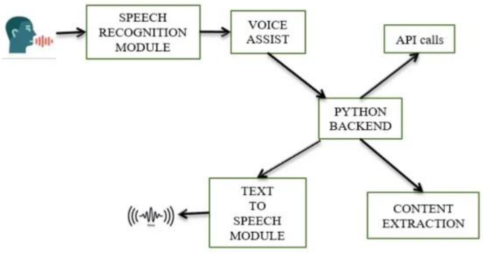


Fig 3 Detailed Workflow of the voice assistant

This Assistant consists of three modules. First is, assistant accepting voice input from user. Secondly, analyzing the input given by the user, and mapping it to the respective intent and function. And the third is, the assistant giving user the result all along with voice. Initially, the assistant will start accepting the user input. After receiving the input, the assistant will convert the analog voice input to the digital text.

Methodology

At the outset we make our program capable of using system voice with the help of sapi5 and pyttsx3. pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3.

The Speech Application Programming Interface or SAPI is an API developed by Microsoft to allow the use of speech recognition and speech synthesis within Windows applications. The main function is then defined where all the capabilities of the program are defined. The proposed system is supposed to have the following functionality:

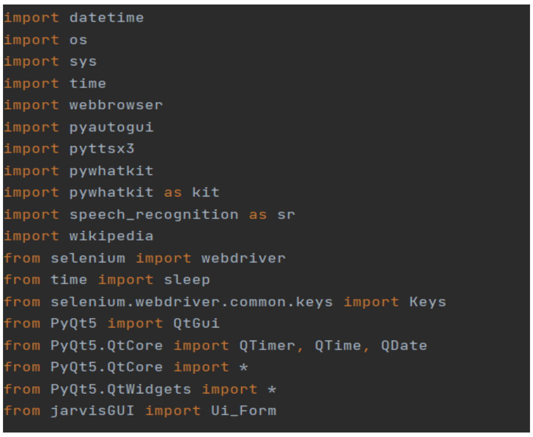
(a) The assistant asks the user for input and keeps listening for commands. The time for listening can be set according to user's requirement.

(b) If the assistant fails to clearly grasp the command it will keep asking the user to repeat the command again and again.

(c) This assistant can be customized to have either male or female voice according to user’s requirement.

(d) The current version of the assistant supports features like Checking weather updates, Sending and checking mails, Search Wikipedia, Open applications, Check time, take note, show note, Open YouTube, Google, Close YouTube, Google, Open and close applications.

Imported modules



```
import datetime
import os
import sys
import time
import webbrowser
import pyautogui
import pyttsx3
import pywhatkit
import pywhatkit as kit
import speech_recognition as sr
import wikipedia
from selenium import webdriver
from time import sleep
from selenium.webdriver.common.keys import Keys
from PyQt5 import QtGui
from PyQt5.QtCore import QTimer, QTime, QDate
from PyQt5.QtCore import *
from PyQt5.QtWidgets import *
from jarvisGUI import Ui_Form
```

Fig 4 Imported modules

Speech Recognitionmodule

Since we're creating a voice assistant app, one of the most critical features is that your assistant knows your voice. In the terminal, execute the following command to install this module.

DateTime module

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

Wikipedia

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

Webbrowser

To perform web search. This module comes built-in with Python.

OS

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

Pyaudio

PyAudio is a set of Python bindings for PortAudio, a cross-platform C++ library interfacing with audio drivers.

PyQt5

PyQt5 is a comprehensive set of Python bindings for Qt v5. It is implemented as more than 35 extension modules and enables Python to be used as an alternative application development language to C++ on all supported platforms including iOS and Android.PyQt5 may also be embedded in C++ based applications to allow users of those applications to configure or enhance the functionality of those applications.

Python Backend

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

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.

Speech to text conversion

Speech Recognition is used to convert speech input to textual output. It decodes the voice and converts it into a textual format which will be understood by pc easily.

Content Extraction

Context extraction (CE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. In most cases, this activity concerns processing human language texts using 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.

Textual output

It decodes the voice command and performs the operation then shows the voice command as textual output in the terminal.

V. EXECUTION

The assistant, on starting, will initially wait for the input to be given from user. If the user gives input command, via voice, the assistant will capture it, and searches for the keyword present in the input command. If the assistant was able to find a keyword related to the given command, then it will perform the task according to the input, and returns the output back to user, in voice also in textual form in the terminal window. If not, the assistant will again start waitingfor the user to give valid input.Each of these functionalities is having their own importance in the whole system working.

User Input - The assistant will wait for the user to click on start or quit button.

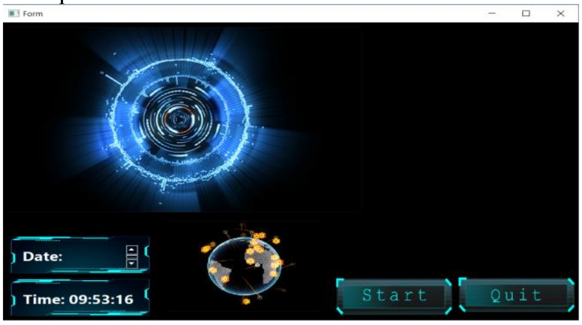
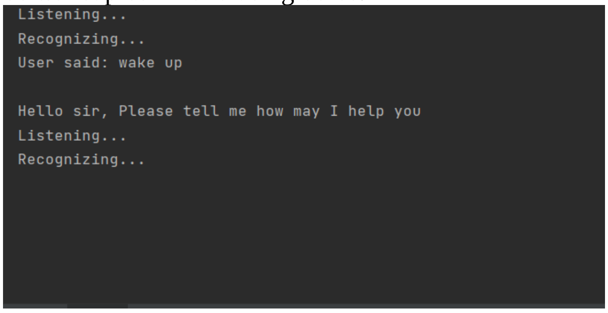


Fig 5Start up GUI for the voice assistant

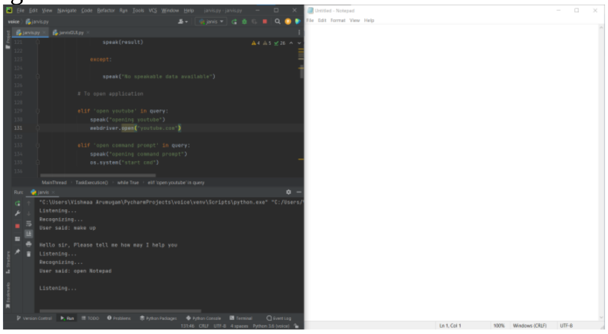
Wakeup command – The assistant will wait for the user to give wakeup command using voice.



```
Listening...
Recognizing...
User said: wake up

Hello sir, Please tell me how may I help you
Listening...
Recognizing...
```

Fig 6 Wake up command by user after click start button in GUI.

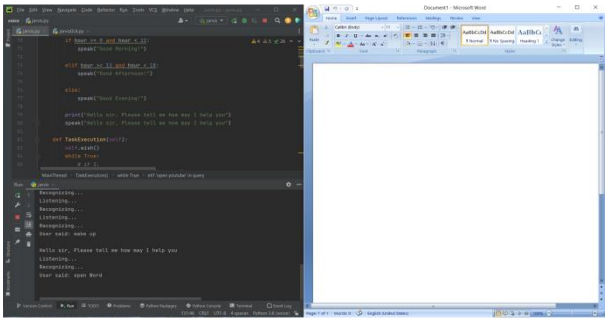
Opening Application – The assistant will wait for the user to give a valid voice command.



```
                speak(result)

            except:

                speak("No speakable data available")

        # To open application

        elif 'open youtube' in query:
            speak("opening youtube")
            webdriver.open("youtube.com")

        elif 'open command prompt' in query:
            speak("opening command prompt")
            os.system("start cmd")
```

```
"C:\Users\Vishmaa Arumugam\PycharmProjects\voice\venv\Scripts\python.exe" "C:/Users/
Listening...
Recognizing...
User said: wake up

Hello sir, Please tell me how may I help you
Listening...
Recognizing...
User said: open Notepad

Listening...
```

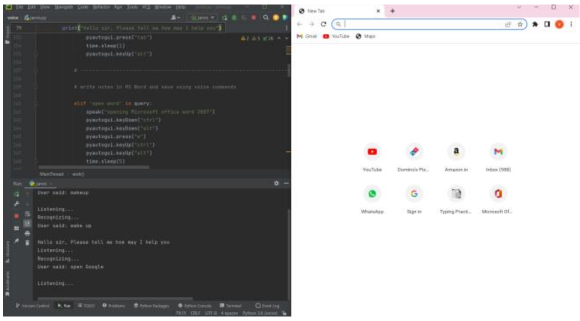
Fig 7 Assistant opening notepad



```
        if hour >= 0 and hour < 12:
            speak("Good Morning!")

        elif hour >= 12 and hour < 18:
            speak("Good Afternoon!")

        else:
            speak("Good Evening!")

        print("Hello sir, Please tell me how may I help you")
        speak("Hello sir, Please tell me how may I help you")

    def TaskExecution(self):
        self.wish()
        while True:
            # if 1:
```

```
Recognizing...
Listening...
Recognizing...
Listening...
Recognizing...
User said: wake up

Hello sir, Please tell me how may I help you
Listening...
Recognizing...
User said: open Word
```

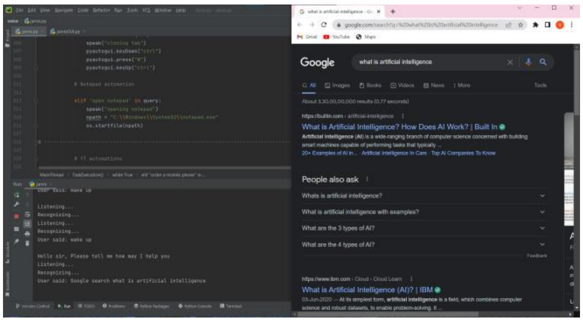
Fig 8 Assistant opening Microsoft office word



```
        print("Hello sir, Please tell me how may I help you")
                pyautogui.press("tab")
                time.sleep(1)
                pyautogui.keyUp("alt")

            # --------------------------------------------------------

            # write notes in MS Word and save using voice commands

            elif 'open word' in query:
                speak("opening Microsoft office word 2007")
                pyautogui.keyDown("ctrl")
                pyautogui.keyDown("alt")
                pyautogui.press("w")
                pyautogui.keyUp("ctrl")
                pyautogui.keyUp("alt")
                time.sleep(5)
```

```
User said: makeup

Listening...
Recognizing...
User said: wake up

Hello sir, Please tell me how may I help you
Listening...
Recognizing...
User said: open Google

Listening...
```

Fig 9 Assistant opening Google chrome



```
speak("closing tab")
pyautogui.keyDown("ctrl")
pyautogui.press("W")
pyautogui.keyUp("ctrl")

# Notepad automation

elif 'open notepad' in query:
    speak("opening notepad")
    npath = "C:\\Windows\\System32\\notepad.exe"
    os.startfile(npath)

# YT automations
```

```
User said: wake up

Listening...
Recognizing...
Listening...
Recognizing...
User said: wake up

Hello sir, Please tell me how may I help you
Listening...
Recognizing...
User said: Google search what is artificial intelligence
```

what is artificial intelligence

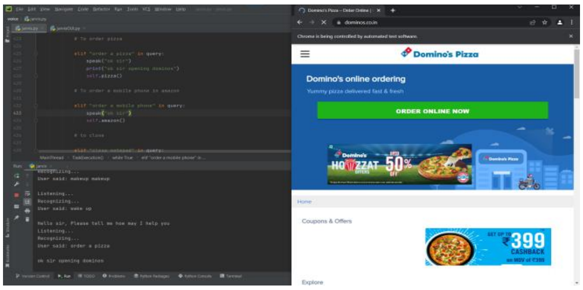
About 3,30,00,00,000 results (0.77 seconds)

https://builtin.com › artificial-intelligence
What is Artificial Intelligence? How Does AI Work? | Built In
Artificial intelligence (AI) is a wide-ranging branch of computer science concerned with building smart machines capable of performing tasks that typically ...
20+ Examples of AI in... · Artificial Intelligence in Cars · Top AI Companies To Know

People also ask
Whats is artificial intelligence?
What is artificial intelligence with examples?
What are the 3 types of AI?
What are the 4 types of AI?

https://www.ibm.com › Cloud › Cloud Learn
What is Artificial Intelligence (AI)? | IBM
03-Jun-2020 — At its simplest form, artificial intelligence is a field, which combines computer science and robust datasets, to enable problem-solving. It ...

Fig 10 Assistant gathering information from Google

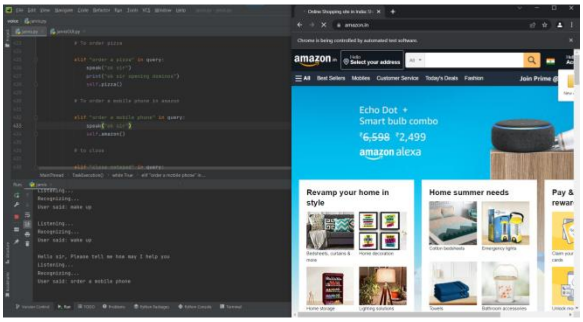


```
# To order pizza

elif "order a pizza" in query:
    speak("ok sir")
    print("ok sir opening dominos")
    self.pizza()

# To order a mobile phone in amazon

elif "order a mobile phone" in query:
    speak("ok sir")
    self.amazon()

# to close

elif "close notepad" in query:
```

```
Recognizing...
User said: makeup makeup

Listening...
Recognizing...
User said: wake up

Hello sir, Please tell me how may I help you
Listening...
Recognizing...
User said: order a pizza

ok sir opening dominos
```

Chrome is being controlled by automated test software.

Fig 11 Assistant ordering pizza in dominos



```
# To order pizza

elif "order a pizza" in query:
    speak("ok sir")
    print("ok sir opening dominos")
    self.pizza()

# To order a mobile phone in amazon

elif "order a mobile phone" in query:
    speak("ok sir")
    self.amazon()

# to close

elif "close notepad" in query:
```

```
Listening...
Recognizing...
User said: make up

Listening...
Recognizing...
User said: wake up

Hello sir, Please tell me how may I help you
Listening...
Recognizing...
User said: order a mobile phone
```

Chrome is being controlled by automated test software.

Fig 12 Assistant ordering a mobile in Amazon

VI. CONCLUSION

This paper presents a comprehensive overview of the design and development of a Static Voice enabled personal assistant for pc using Python programming language. This Voice enabled personal assistant, in today's life style will be more effective in case of saving time and helpful to differently abled people, compared to that of previous days. This Assistant works properly to perform some tasks given by user. Furthermore, there are many things that this assistant is capable of doing, like sending message to user mobile, YouTube automation, gathering information from Wikipedia and Google, with just one voice command.

Through this voice assistant, we have automated various services using a single line command. It eases most of the tasks of the user like searching the web etc, We aim to make this project a complete server assistant and make it smart enough to act as a replacement for a general server administration.The project is built using open source software modules with PyCharm community backing which can accommodate any updates shortly. The modular nature of this project makes it more flexible and easy to add additional features without disturbing current system functionalities.

VII. REFRENCES

[1] Shaughnessy, IEEE, “Interacting with Computers by Voice: Automatic Speech Recognition and Synthesis” proceedings of the IEEE, vol. 91, no. 9, september 2003.

[2] Patrick Nguyen, Georg Heigold, Geoffrey Zweig, “Speech Recognition with Flat Direct Models”, IEEE Journal of Selected Topics in Signal Processing, 2010.

[3] Mackworth (2019-2020), Python code for voice assistant: Foundations of Computational Agents- David L. Poole and Alan K. Mackworth.

[4] Nil Goksel, CanbekMehmet ,EminMutlu, “On the track of Artificial Intelligence: Learning with Intelligent Personal Assistant”, proceedings of International Journal of Human Sciences, 2016.

[5] Keerthana S, Meghana H, Priyanka K, Sahana V. Rao, Ashwini B “Smart Home Using Internet of Things ”, proceedings of Perspectives in Communication , Embedded -systems and signal processing, 2017.

[6] Sutar Shekhar, P. Sameer, Kamad Neha, Prof. Devkate Laxman, “An Intelligent Voice Assistant Using Android Platform”, IJARCSMS, ISSN: 232-7782, 2017.

[7] Rishabh Shah, Siddhant Lahoti, Prof. Lavanya. K, “An Intelligent Chatbot using Natural Language Processing”, International Journal of Engineering Research , Vol.6 , pp.281-286, 2017.

[8] Luis Javier Rodríguez-Fuentes, Mikel Peñagarikano, AparoVarona, Germán Bordel, “GTTS-EHU Systems for the Albayzin 2018 Search on Speech Evaluation”, proceedings of IberSPEECH, Barcelona, Spain, 2018.

[9] Ravivanshikumar ,Sangpal,Tanvee ,Gawand,SahilVaykar, “JARVIS: An interpretation of AIML with integration of gTTS and Python”, proceedings of the 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kanpur, 2019.

[10] Luis Javier Rodríguez-Fuentes, Mikel Peñagarikano, AparoVarona, Germán Bordel, “GTTS-EHU Systems for the Albayzin 2018 Search on Speech Evaluation”, proceedings of IberSPEECH, Barcelona, Spain, 2018.