Volume: 11 Issue: V Month of publication: May 2023
DOI: https://doi.org/10.22214/ijraset.2023.50132

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

Research Paper on Desktop Assistant

Lilesh Mandhalkar[1], Ishika Potbhare[2], Pratiksha Walande[3], Durgesh Yerme[4], Mr. Chandrapal Chauhan[5]
BE Department of Computer Science Engineering, Government College of Engineering, Chandrapur, Maharashtra, India

Abstract: The main thing of Artificial intelligence (AI) is the consummation of natural dialogue between humans and machines. There are numerous IT companies have used the dialogue systems technology to establish various kinds of Virtual assistants grounded on their operations and areas for adding commerce between mortal and machine, similar as Microsoft's Cortana, Apple's Siri, Amazon Alexa, Google Assistant. Like Microsoft Cortana we've created our own virtual desktop assistant only for Government College Of Engineering Chandrapur (GCOEC) using python which is suitable to run on any windows operating systems. We use python as a programming language because it has a multitude of libraries which can be used to create Artificial Intelligence. Our version of desktop assistant is capable of recognizing voice and voice commands.
Desktop Assistant are programs on digital device like PC that hear and respond to verbal commands. Example, a user can say, "Open GCOEC Website" and the assistant will answer with the sanctioned website of GCOEC.
Desktop Assistants can change the way of life of the people in a different manner. Bias is starting to build up in the community as AI's are getting smarter in their own way to interact with humans in an easy language. The Desktop Assistant is a program that can fete mortal voices and can respond via integrated voice system. This paper will define the working of Desktop assistant, their main problems and limitations. In this paper it's described that the system of creating a Desktop assistant without using pall services, which will allow the expansion of similar bias in the future.
Keywords: Desktop Assistant, Python's Speech Recognition, Python textbook- to- speech library pyttsx3, Python3.9, Python3.10, GCOEC(Government College of Engineering Chandrapur), Artificial Intelligence (AI), Natural Language Processing (NPL).

I. INTRODUCTION

As a personal desktop assistant, it helps the end-user with daily tasks like having casual conversations, searching on Google, looking up videos, word definitions, finding information about medications, making health recommendations based on symptoms, searching for admission- and exam-related information, finding college and scholarship websites, and reminding the user of upcoming events and tasks. Machine learning is used to assess user commands and statements in order to provide the best possible outcome.
Today, practically all tasks are done digitally. With a smartphone in our hands, the entire world is literally at our fingers. We no longer even use our fingers anymore. We only mention the work, and it is completed. There are programmes that allow us to search for terms like "machine learning" and the desktop assistant will provide the results. The job of a virtual assistant is to do that. The foundation of this project is the idea that there is enough freely accessible data and information on the internet to create a virtual desktop assistant capable of making judgements for common user tasks.

II. PROBLEM STATEMENT

We are all familiar with virtual assistants like Cortana, Siri, Google Assistant, and many others that can perform or aid in our daily virtual tasks. To our astonishment, however, our college students (GCOEC) do not have access to such a personal virtual desktop helper. In our college (GCOEC), a lot of students have questions and concerns about the application process, admission costs, exam costs, scholarship websites, exam results, exam schedules, university sports announcements, exam announcements, etc.
In other words, the goal of Desktop Assistant is to lessen workload and assist GCOEC students with their everyday tasks, homework, practicals, assignments and as well as to answer all of their questions.

III. METHODOLOGY

Natural Language Processing is used by virtual assistants to translate user text or voice input into actionable commands. Natural language audio signals are transformed into executable commands or digital data that may be processed by the software when a user requests the Desktop Assistant to carry out a task. To find an acceptable response, this data is then compared to the software's data. Machines can be operated using your own commands by using a virtual assistant. Certain Python installation packages, such as, are used to create virtual assistants.

1) *Speech Recognition:* To transform speech input to text, the system makes advantage of Google's online speech recognition system. By using this, customers can talk into a microphone, which is momentarily stored in the system before being sent to the Google Cloud for speech recognition, and receive text in exchange from a special corpus arranged on a computer network server at the information centre. The voice assistant application then receives the identical text and sends it to it.
2) *Python Back-end:* Python is used throughout the program's back end. The speech recognition module's voice input is exchanged into output by the Python back-end, which then determines whether the given instruction is a Context Extraction, API Call, or System Call. The response is then forwarded again to produce the desired result.
3) *API Calls:* Application Programming Interface (API) is a type of software intermediary that facilitates communication between two applications. In other words, API acts as a communication relay, sending users' requests to providers and returning users' responses.
4) *Content Extraction:* Machine-readable documents that are unstructured or semi-structured can automatically extract structured information via context extraction. Natural language processing (NLP) is used in this activity to process texts written in human languages. Content extraction could be defined as processes like automatic annotation and content extraction from various photos, videos, and audio files.

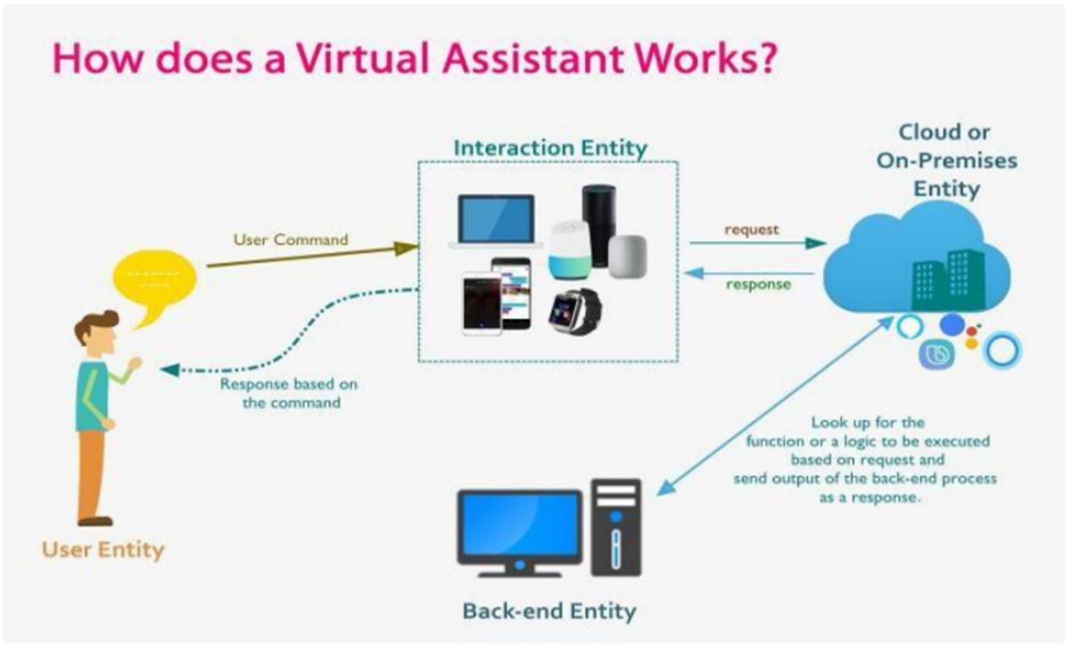

How does a Virtual Assistant Works?
Interaction Entity
Cloud or On-Premises Entity
User Command
request
response
Response based on the command
Look up for the function or a logic to be executed based on request and send output of the back-end process as a response.
User Entity
Back-end Entity

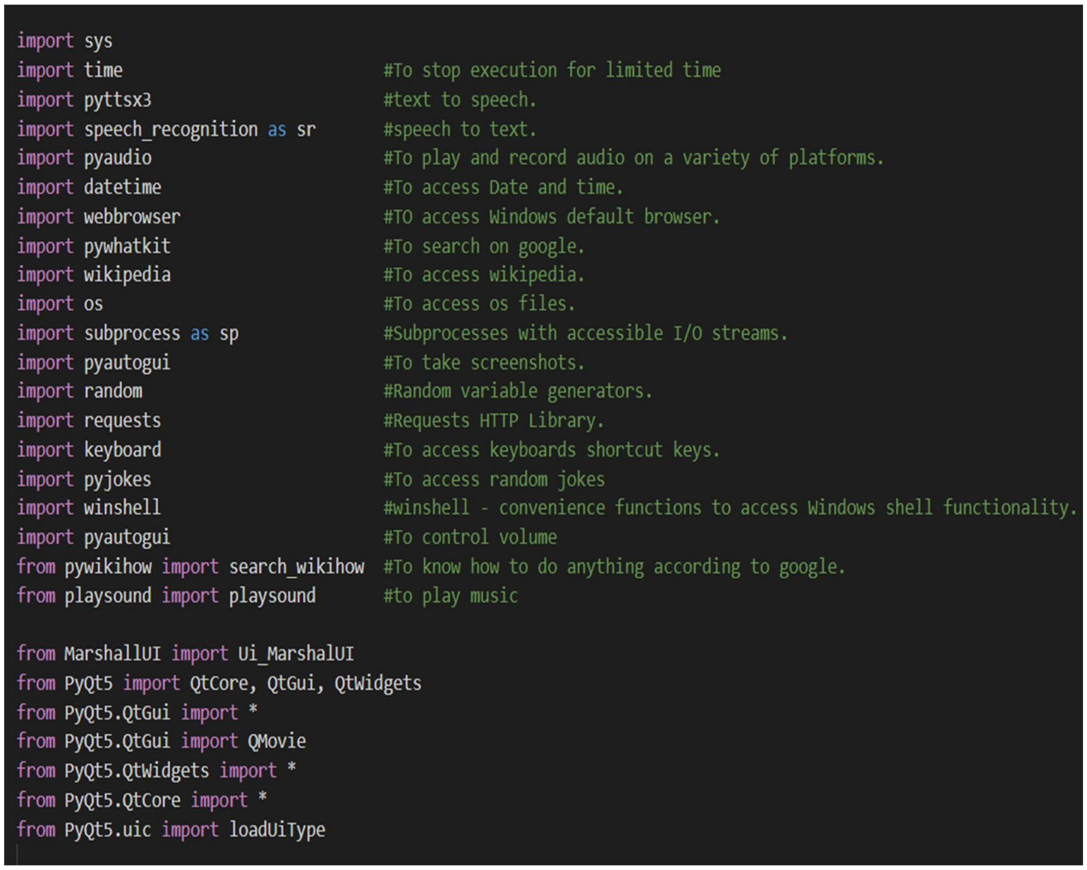
Fig. 1

5) *System Calls:* In system calls, a programmatic technique is used where a computer program asks the operating system's kernel for a service. These services could be connected to hardware services, such as accessing a hard drive, starting and running new processes, or interacting with process scheduling. It offers a crucial interface between the OS and the process.
6) *Google-Text-to-Speech:* Basically, Text-To-Speech is used to turn user-provided Text into Speech. In other words, a Text To Speech Engine transforms written text into phonetic representation, which is subsequently transformed into waveforms, producing sound. Text To Speech has advanced significantly, and third-party publishers now offer it in a variety of languages.

IV. SYSTEM ARCHITECTURE

1) *Wolfram Alpha-* Using Wolfram's algorithms, knowledge base, and AI technology, it computes expert-level responses to any command.
2) *JSON-* JavaScript Object Notation. When data is transferred from a server to a web page, JSON is used. JSON "self-describes" and is simple to comprehend.
3) *Speech Recognition-* Voice recognition is the ability of a machine to comprehend human speech. To create software that can be used to run machines on command, we are leveraging the Python Google Voice API in our project. To recognise voice instructions, the Pyaudio Python module must be installed. With the pip install Pyaudio command, Pyaudio may be installed.

4) *gTTS*- Google’s text-to-speech. Your voice query command is converted to text using Google's text-to-speech software. gTTS transforms the response from the look-up function you create for retrieving the answer to the query or command into an audio format. The Google Translate API is interfaced with via this package.
5) *Datetime*- Date and time are displayed using the Datetime package. Python is already included with this datetime module.

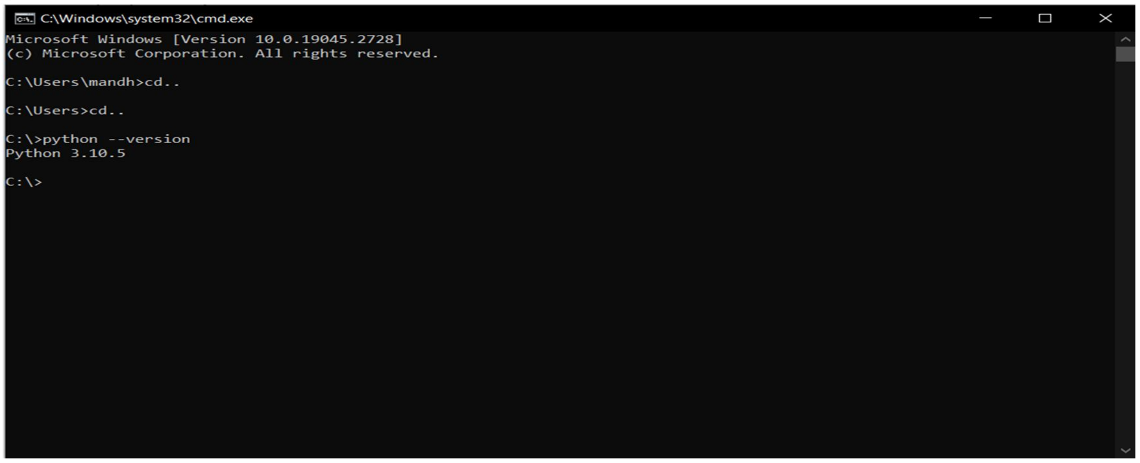


```
import sys
import time                         #To stop execution for limited time
import pyttsx3                      #text to speech.
import speech_recognition as sr     #speech to text.
import pyaudio                      #To play and record audio on a variety of platforms.
import datetime                     #To access Date and time.
import webbrowser                   #TO access Windows default browser.
import pywhatkit                    #To search on google.
import wikipedia                    #To access wikipedia.
import os                           #To access os files.
import subprocess as sp             #Subprocesses with accessible I/O streams.
import pyautogui                    #To take screenshots.
import random                       #Random variable generators.
import requests                     #Requests HTTP Library.
import keyboard                     #To access keyboards shortcut keys.
import pyjokes                      #To access random jokes
import winshell                     #winshell - convenience functions to access Windows shell functionality.
import pyautogui                    #To control volume
from pywikihow import search_wikihow #To know how to do anything according to google.
from playsound import playsound     #to play music

from MarshallUI import Ui_MarshalUI
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtGui import *
from PyQt5.QtGui import QMovie
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.uic import loadUiType
```

Fig. 2: System Architecture

6) *Wikipedia*- We have utilized the Wikipedia module in our project to get more information from Wikipedia or to do a Wikipedia search because, as we all know, Wikipedia is a great and enormous source of knowledge, just like GeeksforGeeks or any other sources.
7) *Webbrowser*- To conduct a web search. Python already includes this module.
8) *OS*- Python's OS module offers tools for communicating with the operating system. The fundamental utility modules for Python include OS. The functions OS module provides allows us to operate on underlying Operating System tasks, irrespective of it being a Windows Platform, Macintosh or Linux
9) *Pyjokes*- Pyjokes is a tool for collecting jokes online. Pyjokes are included in our project because it includes jokes. It's quite fascinating. You can create jokes with this module.
10) *Pyaudio*- PortAudio is a cross-platform C++ library that interfaces with audio drivers. PortAudio's Python bindings are called PyAudio.
11) *Smtplib*- The Python library for sending emails using the Simple Mail Transfer Protocol is called the simple mail transfer protocol library (SMTP). Python includes the smtplib module by default; you don't need to install it. It removes all of SMTP's complexity through abstraction. It offers client implementation for Simple Mail Transmission Protocol (SMTP).
12) *Requests*- Python's Requests module enables you to send http requests. You can use it to send GET and POST queries. It hides the difficulties of submitting requests behind a lovely, straightforward API.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mandh>cd..

C:\Users>cd..

C:\>python --version
Python 3.10.5

C:\>
```

Fig. 3- Python Version

V. RESULT

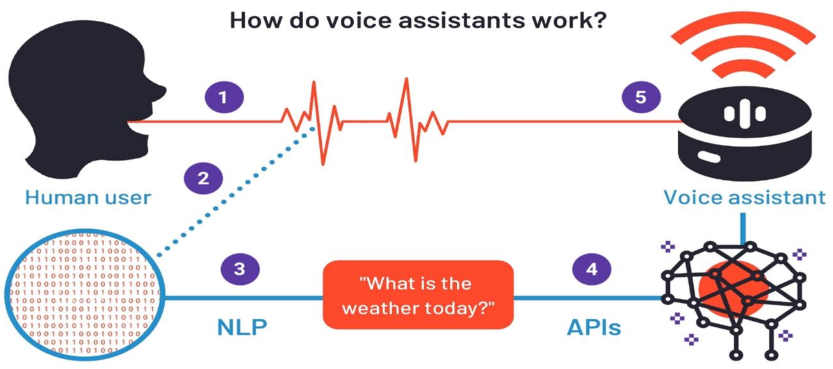
Desktop assistant is less time-consuming. Desktop assistant is a software that understands commands and completes tasks assigned by clients. Virtual assistant uses NLP to match user voice or text input with executable commands. With the help of a Desktop assistant, you can run your machine-like laptop or PCs on your own command. It is a fast process; therefore, it saves time. A desktop assistant is working for you at set times, so always available to you and able to adapt to changing needs quickly. A desktop assistant will be available to students of GCOEC and, should their workload enable, help others too, such as family and colleagues.

VI. CONCLUSION

In this paper, we examined using Python to create a personal desktop assistant. Humans' lives are made easier with desktop assistants. The freedom to only hire a virtual assistant for the services they require. For the Government College of Engineering Chandrapur, we developed a virtual assistant in Python similar to Alexa, Cortana, Siri, and Google Assistant. For this project, we make use of artificial intelligence technology. Using a virtual personal assistant can help you manage or organise your calendar. Because they are more movable, dependable, and always accessible, virtual personal assistants are also more trustworthy than human personal assistants. Our virtual assistant will learn more about you and inform you of suggestions while also accepting directions. We can anticipate this software.

VII. OUTCOME

In this study, we have developed a voice assistant which can perform any kind of task in exchange of commands given by the users without any error. We have added more features like it will listen to the users' voice only and will not be activated from environmental noise. The modular nature of this project makes it easy to understand and more flexible. We can add more features to the program without disturbing the functionalities. All the packages required in python programming language have been installed and the code was implemented using VS Code Integrated Development Environment (IDE). The python version used for this project was 3.x and the data of different noises was also taken from the environment.



VIII. SUMMARY

In this study, we created a desktop assistant that, in exchange for user commands, can flawlessly carry out any kind of task. Further features have been added, such as the fact that it will only respond to user speech and won't be activated by background noise. This project is more adaptable and simple to grasp because of its modular design. Without changing the functionality of the program, we can add more features. The Python programming language's necessary packages have all been installed, and the VS Code Integrated Development Environment was used to write the code (IDE). Python versions 3.9 and 3.10 were used for this project, and the data for various noises was also obtained using an environmental methodology.

REFERENCES

[1] Research Paper on Desktop Voice Assistant: Vishal Kumar Dhanraj (076)
[2] Research paper on Desktop Voice Assistant: Prof. Ranak Jain (05)
[3] Artificial Intelligence Project Idea blogs [data-flair.training/blogs]
[4] How to build virtual assistant in python blog (Author: Dante Sblendorio)
[5] Making own AI assistant blog on CodeX (Author: Ramil Jivanni)
[6] Python programming [www.pythonprogramming.net]
[7] Python documentation [www.python.org]
[8] Desktop Assistant from Wikipedia
[9] Designing Personal Assistant Software for Task Management using Semantic Web Technologies and Knowledge Databases.
[10] Richard Krisztian Csaky, "Desktop Assistant and related Research paper Notes with Images,
[11] Chatbot Learning: Everything you need to know about machine learning chatbots (2020).
[12] How to use an API with Python (Beginner's Guide).
[13] Introduction to Machine Learning using Python (January 2019).