AN INDUSTRIAL TRAINING REPORT

Voice Assistant

At

Tathastu by Twowaits

Date: 01/07/2022 - 31/07/2022

Submitted By:

KUMAR SANU

UE208056, 4th Semester

To

Department of Information Technology

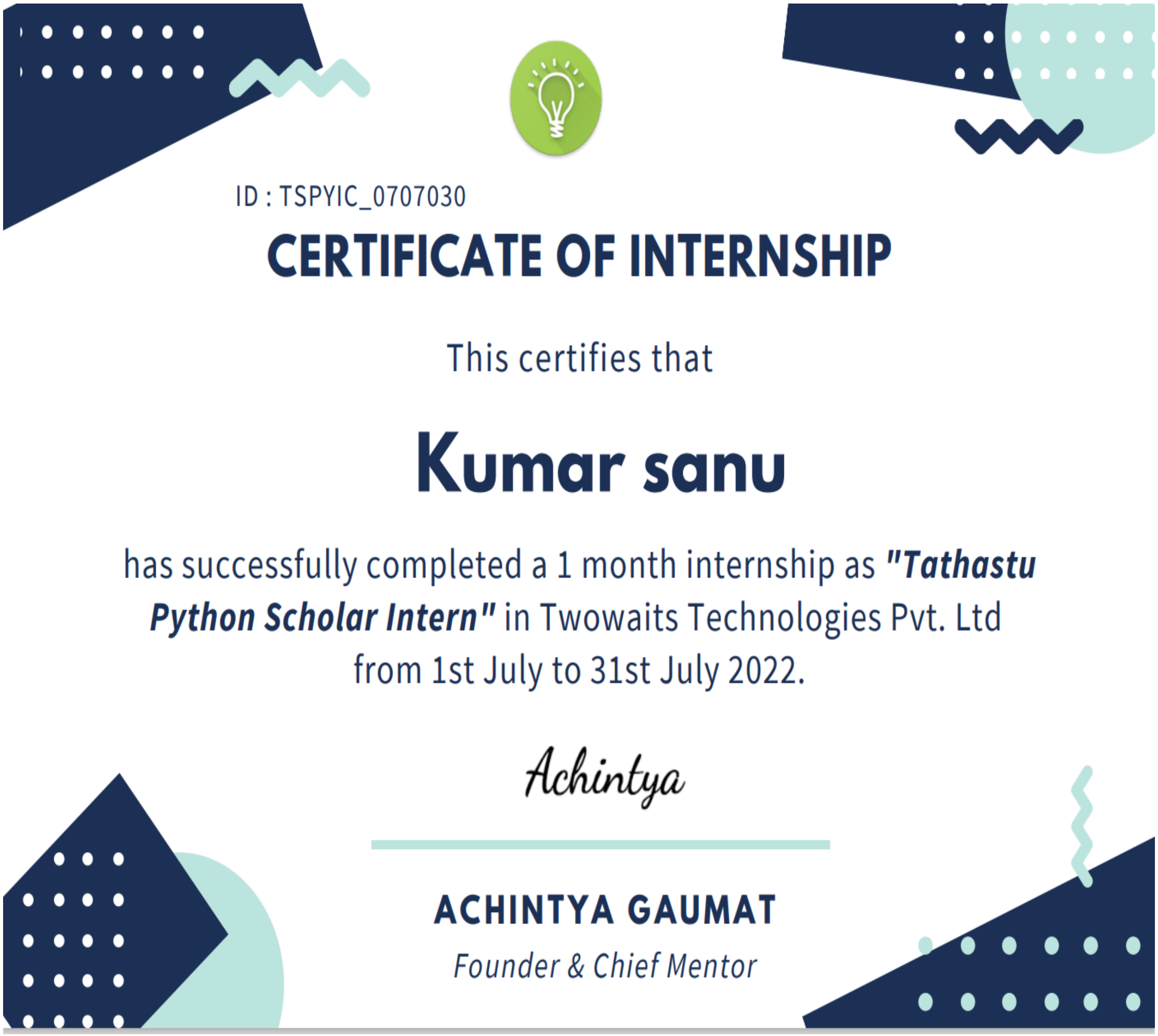
University Institute of Engineering and Technology

Panjab University, Chandigarh

July, 2022

Index

Certificate



ID : TSPYIC_0707030

CERTIFICATE OF INTERNSHIP

This certifies that

Kumar sanu

has successfully completed a 1 month internship as ***"Tathastu Python Scholar Intern"*** in Twowaits Technologies Pvt. Ltd from 1st July to 31st July 2022.

Achintya

ACHINTYA GAUMAT

Founder & Chief Mentor

Acknowledgement

In completing this project report on project titled **VIRTUAL ASSISTANT**, I had to take the help and guideline of a few respected people, who deserve my greatest gratitude.

The completion of this project report gives me much Pleasure. I would like to show my gratitude to Sir **Achintya Gaumat** for giving me a good guideline for project throughout numerous consultations. I would also like to expand my deepest gratitude to all those who have directly and indirectly guided me in writing this project report.

Introduction

In today’s era almost all tasks are digitalized. We have Smartphone in hands and it is nothing less than having world at your fingertips. These days we aren’t even using fingers. We just speak of the task and it is done. There exist systems where we can say and task is done. That is the task of a Virtual Assistant.

Virtual Assistants are software programs that help you ease your day-to-day tasks, such as showing weather report, creating reminders, making shopping lists etc. They can take commands via text (online chat bots) or by voice.

This system is designed to be used efficiently on desktops. Personal assistant software improves user productivity by managing routine tasks of the user and by providing information from online sources to the user.

Voice searches have dominated over text search. Virtual assistants are turning out to be smarter than ever. Allow your intelligent assistant to make email work for you. Detect intent, pick out important information, automate processes, and deliver personalized responses. This project was started on the premise that there is sufficient amount of openly available data and information on the web that can be utilized to build a virtual assistant that has access to making intelligent decisions for routine user activities.

Purpose of virtual assistant is to being capable of voice interaction, music playback, making to-do lists, setting alarms, streaming podcasts, playing audiobooks, and providing weather, traffic, sports, and other real-time information, such as news. Virtual assistants enable users to speak natural language voice commands in order to operate the device and its Apps. There is an increased overall awareness and a higher level of comfort demonstrated specifically by millennial consumers. In this ever-evolving digital world where speed, efficiency, and convenience are constantly being optimized, it’s clear that we are moving towards less screen interaction.

The Technology Used

Python

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages.

Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, data science etc. Python has a lot of libraries for every need of this project. For Virtual Assistant, libraries used are speech recognition to recognize voice, Pyttsx for text to speech etc.

Python is reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most the time, and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code (because you will have more time to optimize) than writing everything in a low-level language.

Pyttsx

Pyttsx stands for Python Text to Speech. It is a cross-platform Python wrapper for text-to-speech synthesis. It is a Python package supporting common text-to-speech engines on MacOS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantage is that it works offline.

Speech Recognition

This is a library for performing speech recognition, with support for several engine and APIs, online and offline. It supports APIs like Google Cloud Speech API, IBM Speech to Text, Microsoft Bing Voice Recognition etc.

Wikipedia

As we all know Wikipedia is a great source of knowledge just like GeeksforGeeks we have used the Wikipedia module to get information from Wikipedia or to perform a Wikipedia search. To install this module type the below command in the terminal.

Web browser

To perform Web Search. This module comes built-in with Python.

Pyjokes

Pyjokes is used for the collection of Python Jokes over the Internet.

Datetime

Date and Time are used to showing Date and Time. This module comes built-in with Python.

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 a messenger that delivers your request to the provider that you’re requesting it from and then delivers the response back to you.

Pyaudio

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

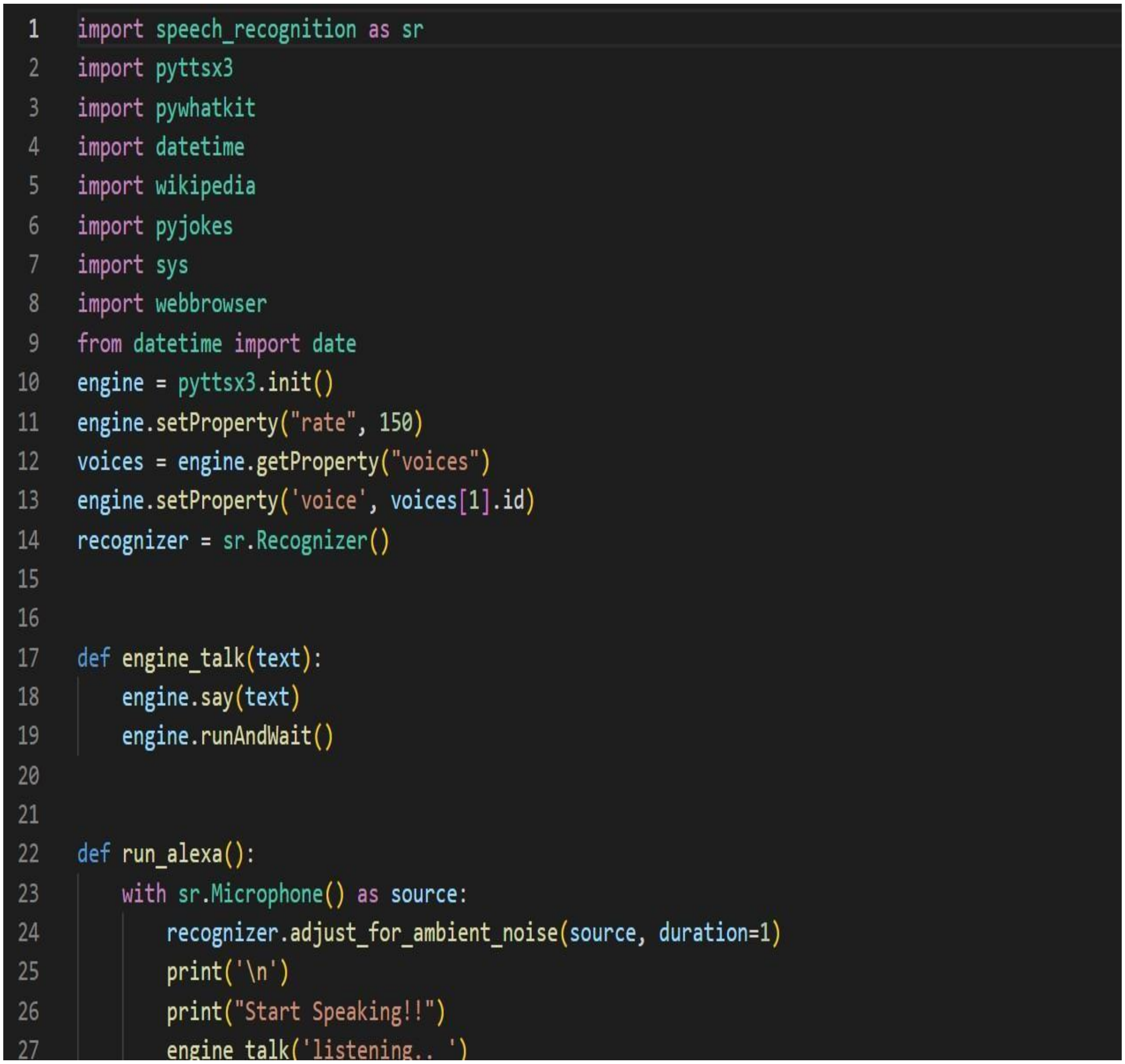
Project details

```
import speech_recognition as sr

import pyttsx3 import pywhatkit

import      datetime      import

wikipedia import pyjokes import

sys

import webbrowser
from   datetime   import   date   engine   =

pyttsx3.init() engine.setProperty("rate",

150)                  voices                   =

engine.getProperty("voices")

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

recognizer = sr.Recognizer()




def engine_talk(text):

    engine.say(text)

    engine.runAndWait()




def run_a():
with sr.Microphone() as source:

        recognizer.adjust_for_ambient_noise(source,

        duration=1) print('\n') print("Start Speaking!!")

        engine_talk('listening.. ')
recordedaudio = recognizer.listen(source) try:

        command = recognizer.recognize_google(recordedaudio, language='en-in')

        command = command.lower()
        if 'alexa' in command:
command = command.replace('alexa', '') print('you
            said '+command)

        else:
print('you said : '+command)



if 'hello' in command:
print('Hello how can i help you ??')
            engine_talk('Hello, how can i help you ??')

        elif 'who are you' in command:
print('I am mini alexa your virtual assistant master')
            engine_talk(

                'I am mini alexa your virtual assistant master. how can i help
you ??')


        elif 'can you do' in command:
print('''i can play songs on youtube , tell you a joke, search on
wikipedia, tell date and time,find your location, locate area on map,

   open different websites like instagram, youtube,gmail, git hub, stack overflow
and searches on google.How may i help you ??''')

            engine_talk('''i can play songs on youtube , tell you a joke, search
on wikipedia, tell date and time,find your location, locate area on map,

   open different websites like insta gram, youtube,gmail, git hub, stack
overflow and searches on google. How may i help you ??''')
        elif 'play' in command:

            song = command.replace('play', '')
            print('Playing' + song)

        engine_talk('Playing' + song)

        pywhatkit.playonyt(song) elif

        'date and time' in command:

            today = date.today()

            time = datetime.datetime.now().strftime('%I:%M %p')
```

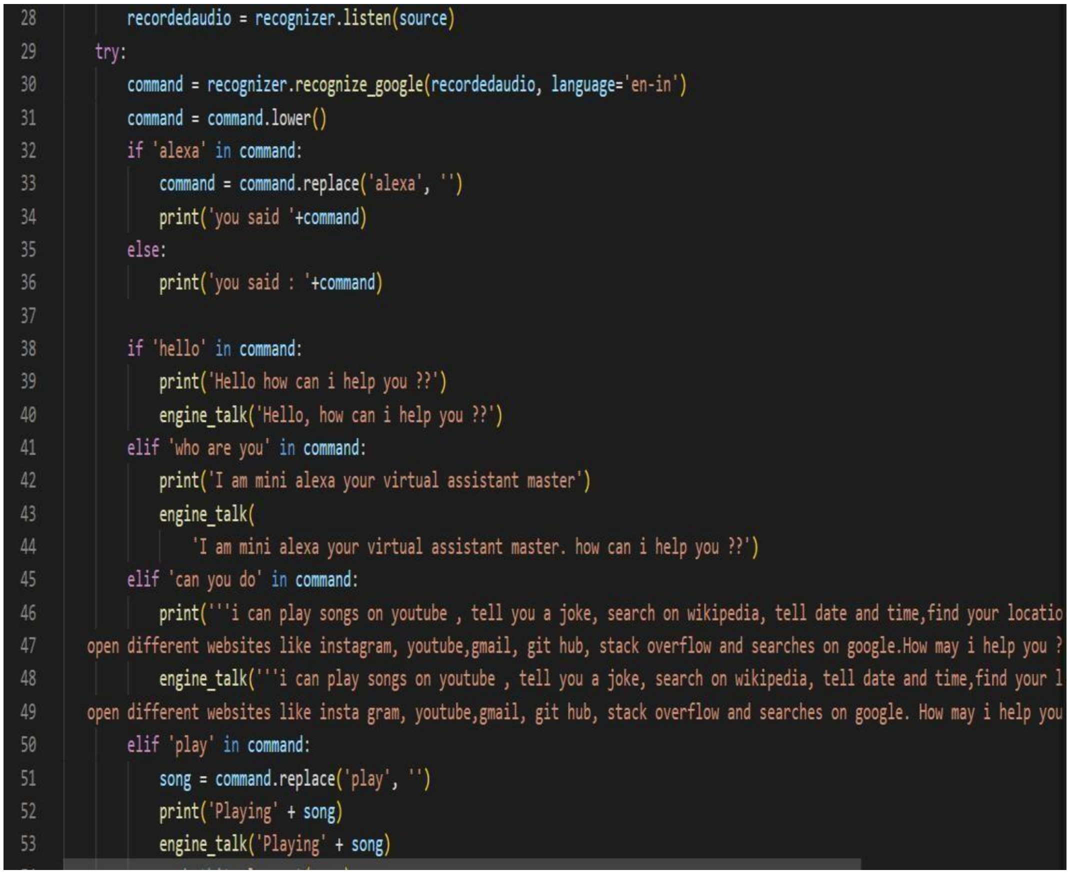
```
d2 = today.strftime("%B %d, %Y")
print("Today's Date is ", d2, 'Current time is', time)
        engine_talk('Today is : ' + d2) engine_talk('and current
        time is ' + time) elif 'time and date' in command: today =
        date.today()
time = datetime.datetime.now().strftime('%I:%M %p')



d2 = today.strftime("%B %d, %Y")
print("Today's  Date  is  ",  d2,  'Current  time  is',  time)
            engine_talk('Current time is ' + time) engine_talk('and
            Today is : ' + d2)



elif 'time' in command:
time = datetime.datetime.now().strftime('%I:%M %p') print('The
        current time is ' + time) engine_talk('The current time
        is') engine_talk(time) elif 'date' in command: today =
        date.today()
            print("Today's date:", today)



d2 = today.strftime("%B %d, %Y")
            print("Today's Date is ", d2)
            engine_talk('The todays date is')
            engine_talk(d2)

        elif 'tell me about' in command:
name = command.replace('tell me about', '') info =
            wikipedia.summary(name, 1) print(info)

            engine_talk(info)
elif 'wikipedia' in command:
name = command.replace('wikipedia', '') info =
        wikipedia.summary(name, 1) print(info)
        engine_talk(info) elif 'what is' in
        command:
            name = command.replace('what is ', '')
        info = wikipedia.summary(name, 1)
        print(info) engine_talk(info) elif 'who is
        ' in command:
            name = command.replace('who is', '')
            info = wikipedia.summary(name, 1)
            print(info)
            engine_talk(info)
elif 'what is ' in command:

            search = 'https://www.google.com/search?q='+command
            print(' Here is what i found on the internet..')
            engine_talk('searching... Here is what i found on the internet..')
            webbrowser.open(search)
        elif 'joke' in command:

            _joke = pyjokes.get_joke()
            print(_joke) engine_talk(_joke)
        elif 'search' in command:
search = 'https://www.google.com/search?q='+command

            engine_talk('searching... ')
            webbrowser.open(search)
        elif "my location" in command:
url = "https://www.google.com/maps/search/Where+am+I+?/"

            webbrowser.get().open(url)
engine_talk("You must be somewhere near here, as per Google maps") elif 'locate
        ' in command:
            engine_talk('locating ...')
loc = command.replace('locate', '') if 'on map'
            in loc:
                loc = loc.replace('on map', ' ')
url = 'https://google.nl/maps/place/'+loc+'/&amp;'
```

```
      webbrowser.get().open(url) print('Here is the location
    of '+loc) engine_talk('Here is the location of '+loc) elif
    'on map' in command: engine_talk('locating ...') loc =
    command.split(" ") print(loc[1]) url =
    'https://google.nl/maps/place/'+loc[1] + '/&amp;'
    webbrowser.get().open(url)
print('Here is the location of '+loc[1]) engine_talk('Here
      is the location of '+loc[1])


elif 'location of' in command:
loc = command.replace('find location of', '')
      engine_talk('locating ...') url =
      'https://google.nl/maps/place/'+loc+'/&amp;'


      webbrowser.get().open(url) print('Here is
    the location of '+loc) engine_talk('Here is
    the location of '+loc) elif 'where is ' in
    command: engine_talk('locating ...')
loc = command.replace('where is', '')
url = 'https://google.nl/maps/place/'+loc+'/&amp;'

      webbrowser.get().open(url)  print('Here  is  the
    location of '+loc) engine_talk('Here is the location
    of   '+loc)   elif   'open   google'   in   command:
    print('opening   google   ...')   engine_talk('opening
    google..')
    webbrowser.open_new('https://www.google.co.in/')
    elif 'gmail' in command:
      print('opening gmail ...')
    engine_talk('opening gmail..')
    webbrowser.open_new('https://mail.google.com/')
```

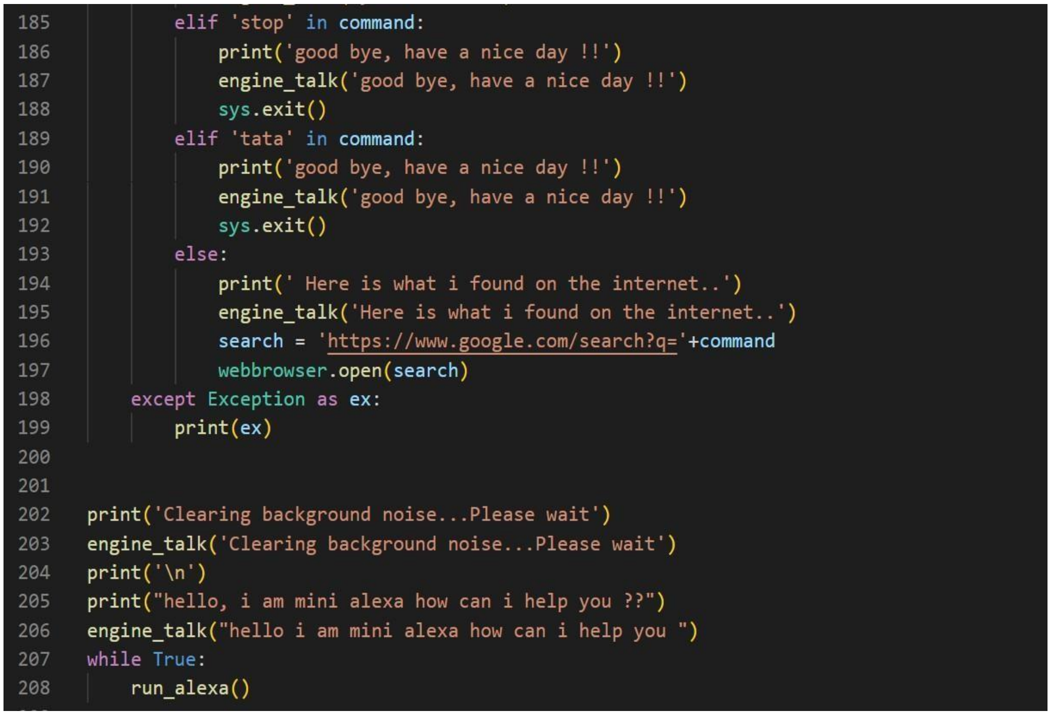
```
        elif 'open youtube' in command: print('opening you
        tube ...')
            engine_talk('opening you tube..')
        webbrowser.open_new('https://www.youtube.com/') elif
        'open instagram' in command: print('opening
        instagram ...') engine_talk('opening insta gram...')
        webbrowser.open_new('https://www.instagram.com/')
        elif 'open stack overflow' in command:
        print('opening stackoverflow ...')
        engine_talk('opening stack overflow...')
        webbrowser.open_new('https://stackoverflow.com/')
        elif 'open github' in command: print('opening git
        hub ...') engine_talk('opening git hub...')
        webbrowser.open_new('https://github.com/') elif
        'bye' in command:
            print('good bye, have a nice day !!')
        engine_talk('good bye, have a nice day !!')
        sys.exit() elif 'thank you' in command:
        print("your welcome") engine_talk('your
        welcome')
        elif 'stop' in command:
print('good bye, have a nice day !!') engine_talk('good
            bye, have a nice day !!') sys.exit()
        elif 'tata' in command:
print('good bye, have a nice day !!') engine_talk('good
            bye, have a nice day !!') sys.exit()
        else:
print(' Here is what i found on the internet..') engine_talk('Here
    is what i found on the internet..') search =
    'https://www.google.com/search?q='+command
    webbrowser.open(search) except Exception as ex:
        print(ex)


print('Clearing background noise...Please wait')
engine_talk('Clearing background noise...Please wait')
print('\n') print("hello, i am mini alexa how can i help
you ??") engine_talk("hello i am mini alexa how can i
help you ") while True:
    run_alexa()
```



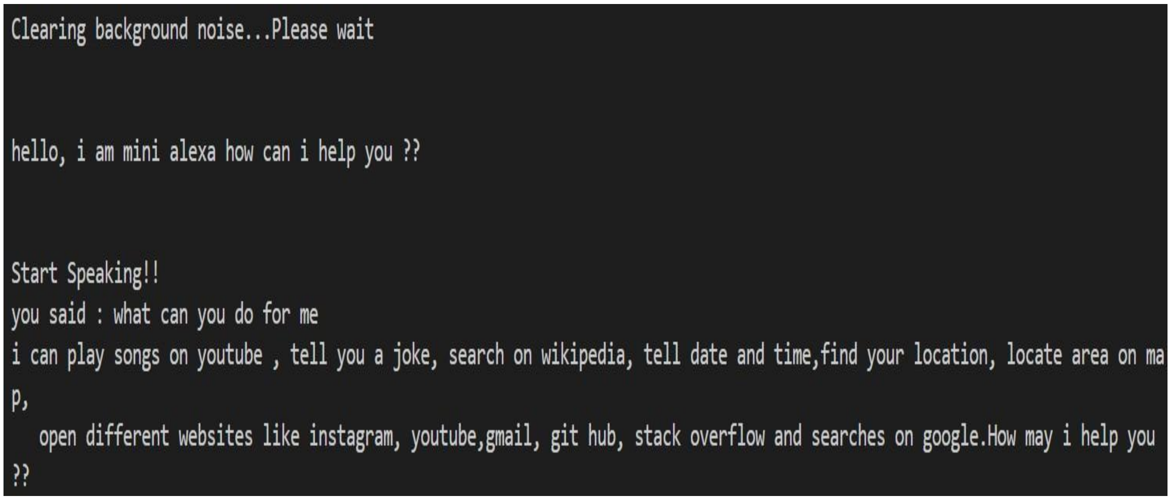
```
import speech_recognition as sr
import pyttsx3
import pywhatkit
import datetime
import wikipedia
import pyjokes
import sys
import webbrowser
from datetime import date
engine = pyttsx3.init()
engine.setProperty("rate", 150)
voices = engine.getProperty("voices")
engine.setProperty('voice', voices[1].id)
recognizer = sr.Recognizer()


def engine_talk(text):
    engine.say(text)
    engine.runAndWait()


def run_alexa():
    with sr.Microphone() as source:
        recognizer.adjust_for_ambient_noise(source, duration=1)
        print('\n')
        print("Start Speaking!!")
        engine_talk('listening...')
```



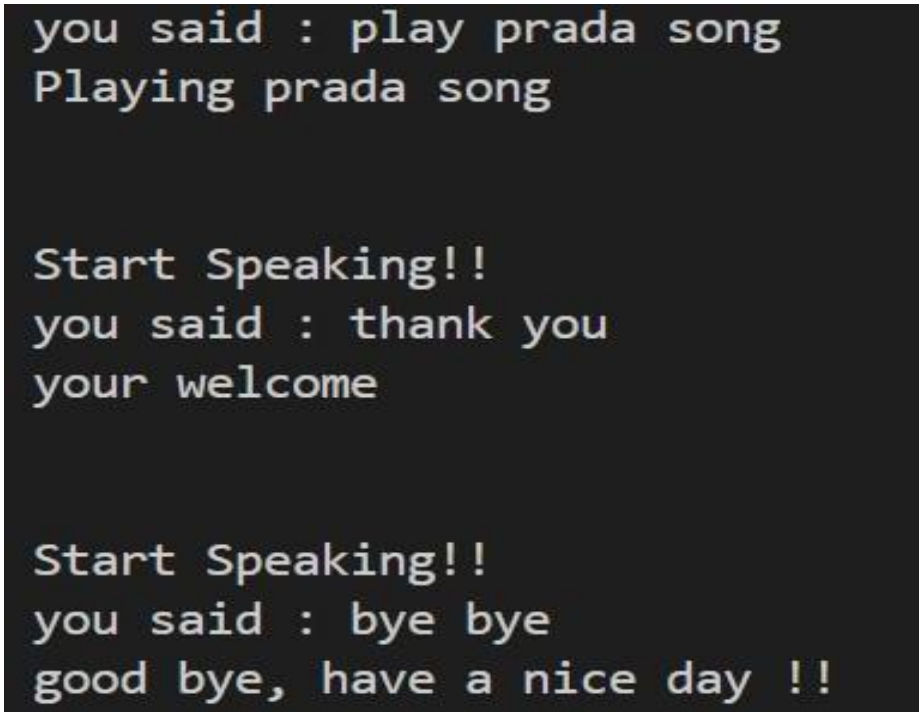
```python
        recordedaudio = recognizer.listen(source)
    try:
        command = recognizer.recognize_google(recordedaudio, language='en-in')
        command = command.lower()
        if 'alexa' in command:
            command = command.replace('alexa', '')
            print('you said '+command)
        else:
            print('you said : '+command)

        if 'hello' in command:
            print('Hello how can i help you ??')
            engine_talk('Hello, how can i help you ??')
        elif 'who are you' in command:
            print('I am mini alexa your virtual assistant master')
            engine_talk(
                'I am mini alexa your virtual assistant master. how can i help you ??')
        elif 'can you do' in command:
            print('''i can play songs on youtube , tell you a joke, search on wikipedia, tell date and time,find your locatio
    open different websites like instagram, youtube,gmail, git hub, stack overflow and searches on google.How may i help you ?
            engine_talk('''i can play songs on youtube , tell you a joke, search on wikipedia, tell date and time,find your l
    open different websites like insta gram, youtube,gmail, git hub, stack overflow and searches on google. How may i help you
        elif 'play' in command:
            song = command.replace('play', '')
            print('Playing' + song)
            engine_talk('Playing' + song)
```



```python
        elif 'stop' in command:
            print('good bye, have a nice day !!')
            engine_talk('good bye, have a nice day !!')
            sys.exit()
        elif 'tata' in command:
            print('good bye, have a nice day !!')
            engine_talk('good bye, have a nice day !!')
            sys.exit()
        else:
            print(' Here is what i found on the internet..')
            engine_talk('Here is what i found on the internet..')
            search = 'https://www.google.com/search?q='+command
            webbrowser.open(search)
    except Exception as ex:
        print(ex)


print('Clearing background noise...Please wait')
engine_talk('Clearing background noise...Please wait')
print('\n')
print("hello, i am mini alexa how can i help you ??")
engine_talk("hello i am mini alexa how can i help you ")
while True:
    run_alexa()
```

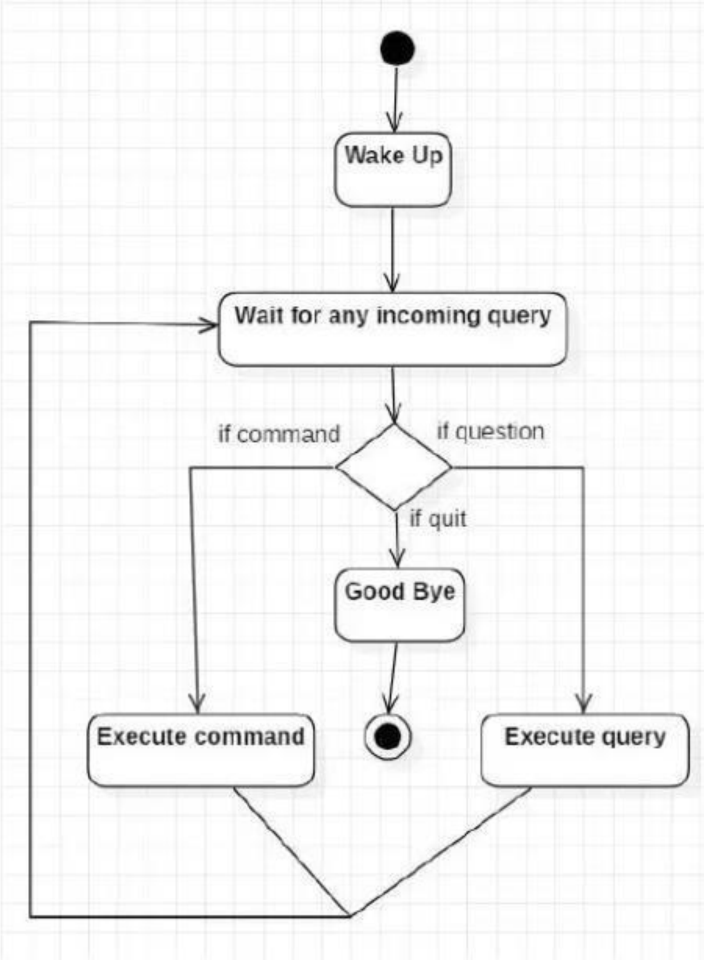


```
Clearing background noise...Please wait


hello, i am mini alexa how can i help you ??


Start Speaking!!
you said : what can you do for me
i can play songs on youtube , tell you a joke, search on wikipedia, tell date and time,find your location, locate area on ma
p,
   open different websites like instagram, youtube,gmail, git hub, stack overflow and searches on google.How may i help you
??
```



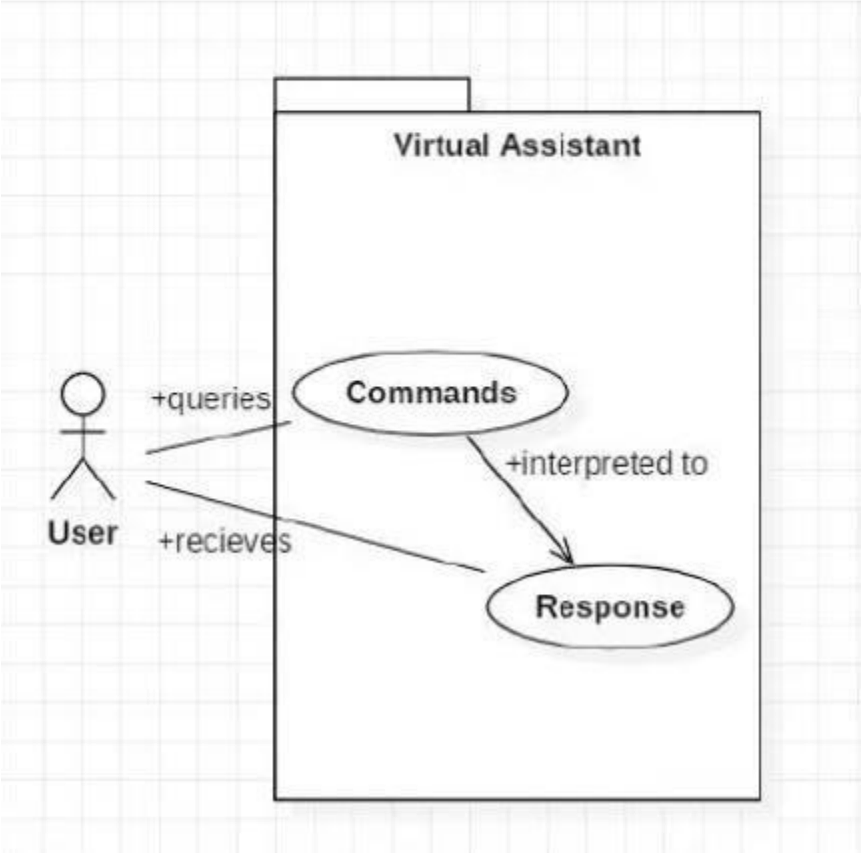
```
you said : play prada song
Playing prada song


Start Speaking!!
you said : thank you
your welcome


Start Speaking!!
you said : bye bye
good bye, have a nice day !!
```

ACTIVITY DIAGRAM

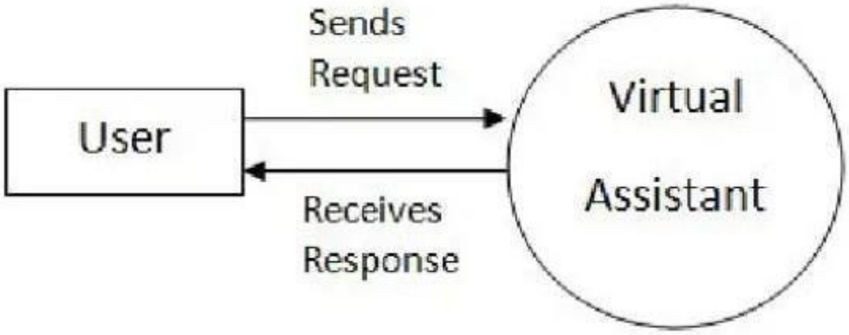


Initially, the system is in idle mode. As it receives any wake-up call it begins execution.The received command is identified whether it is a questionnaire or a task to be performed.Specific action is taken accordingly. After the Question is being answered or the task is being performed, the system waits for another command. This loop continues unless it receives quitcommand. At that moment, it goes back to sleep.

USE CASE DIAGRAM



DATA FLOW DIAGRAM



Conclusion

The project is built using open-source software modules with VS Code 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.

It not only works on human commands but also give responses to the user based on the query being asked or the words spoken by the user such as opening tasks and operations. It is greeting the user the way the 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 every task. The entire system works on the verbal input.

Scope

The virtual assistants which are currently available are fast and responsive but we still have to go a long way. The understanding and reliability of the current systems need to be improved a lot. The assistants available nowadays are still not reliable in critical scenarios. The future of these assistants will have the virtual assistants incorporated with Artificial Intelligence which includes Machine Learning, Neural Networks, etc. and IoT. With the incorporation of these technologies, we will be able to achieve new heights. What the virtual assistants can achieve is much beyond what we have achieved till now. Most of us have seen Jarvis, that is a virtual assistant developed by iron man which is although fictional but this has set new standards of what we can achieve using voice-activated virtual assistants.

Bibliography

For the development of this project, I have taken reference from internet and shared those experiences during the designing, feasibility and actual coding of the project, here are the few: -

➢ Learning Python Basics.

➢ Tutorials from Youtube.

➢ Google for reference.