MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Virtual Assistant

A project report submitted in partial fulfilment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Neha Chaurasiya

0901CS203D05

Harsh Kumar Mishra

0901CS203D02

Faculty Mentor:

Dr. Anjula Mehto

Assistant Professor, CSE, MITS

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957'

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Virtual Assistant

Submitted By:

Neha Chaurasiya

0901CS203D05

Harsh Kumar Mishra

0901CS203D02

Faculty Mentor:

Dr. Anjula Mehto

Assistant Professor, CSE, MITS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Neha Chaurasiya** (0901CS203D05) has submitted the project report titled **Virtual Assistant** under the mentorship of **Dr. Anjula Mehto** in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

Anjula Mehto 09/05/22

Dr. Anjula Mehto
Faculty Mentor
Assistant Professor
Computer Science and Engineering

09/05/2022

Dr. Manish Dixit
Professor and Head,
Computer Science and Engineering

Dr. Manish Dixit
Professor & HOD
Department of CSE
M.I.T.S. Gwalior

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Harsh Kumar Mishra** (0901CS203D02) has submitted the project report titled **Virtual Assistant** under the mentorship of **Dr. Anjula Mehto** in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

Anjula Mehto
03/05/22

Dr. Anjula Mehto
Faculty Mentor
Assistant Professor
Computer Science and Engineering

09/05/2022

Dr. Manish Dixit
Professor and Head,
Computer Science and Engineering

Dr. Manish Dixit
Professor & HOD
Department of CSE
M.I.T.S. Gwalior

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

We hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Dr. Anjula Mehto, Assistant Professor,** Computer Science & Engineering

We declare that We have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Neha Chaurasiya
0901CS203D05
3rd Year
Computer Science and Engineering

Harsh Kumar Mishra
0901CS203D02
3rd Year
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to our career. We are thankful to our institute, **Madhav Institute of Technology and Science** to allow us to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. We extend our gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

We would sincerely like to thank our department, **Department of Computer Science and Engineering, for allowing** us to explore this project. We humbly thanks **Dr. Manish Dixit,** Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

We are sincerely thankful to our faculty mentors. We are grateful to the guidance of **Dr. Anjula Mehto** Assistant professor, Computer Science & Engineering, for their continued support and guidance throughout the project. We are also very thankful to the faculty and staff of the department.

Neha.

Neha Chaurasiya
0901CS203D05
3rd Year
Computer Science and Engineering

Mishra

Harsh Kumar Mishra
0901CS203D02
3rd Year
Computer Science and Engineering

ABSTRACT

As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the process of converting speech into text. This is commonly used in voice assistants like Alexa, Siri, etc. In Python there is an API called Speech Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to send emails without typing any word, Searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favourite IDE with the help of a single voice command. In the current scenario, advancement in technologies are such that they can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time

सार:

जैसा कि हम जानते है कि पायथन एक उभरती हुई भाषा है इसलिए पायथन में वहयस असिस्टेंट के लिए स्क्रिप्ट लिखना आसान हो जाता है। सहायक के लिए निर्देश उपयोगकर्त्ता की आवश्यकता के अनुसार नियंत्रित किए जा सकते है। वाक् पश्चात भाषण को पाठ मे बदलने की प्रक्रिया है। यह आमतौर पर एलेक्सा, सिरी, आदि जैसे वहयस असिस्टेंट में उपयोग किया जाता है। पायथन में स्पीच रिकग्निशन नामक एक एपीआई है जो हमें भाषण को टेक्स्ट में बदलने की अनुमति देता है। अपना खुद का सहायक बनाना एक दिलचस्प काम था। बिना कोई शब्द लिखे ईमेल भेजना, बिना ब्राउजर खोले गूगल पर सर्च करना, और कई अन्य दैनिक कार्य करना जैसे संगीत बजाना, सिंगल वहयस कमांड की मदद से अपना पसंदीदा आईडीई खोलना आसान हो गया। वर्तमान परिदृश्य में, प्रौद्योगिकियों में प्रगति ऐसी है कि वे किसी भी कार्य को समान प्रभावशीलता के साथ कर सकते हैं या हमसे अधिक प्रभावी ढंग से कह सकते हैं। इस प्रोजेक्ट को बनाने से मुझे एहसास हुआ कि हर क्षेत्र में एआई की अवधारणा मानव प्रयास को कम कर रही है और समय बचा रही है

TABLE OF CONTENTS

List of Figures

CHAPTER 1: PROJECT OVERVIEW

1.1 Introduction

Today the development of artificial intelligence (AI) systems that can 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 who 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. Virtual assistants are software programs that help you ease your day to day tasks, such as showing weather reports, creating remainders, making shopping lists etc. They can take commands via text (online chat bots) or by voice. Voice-based intelligent assistants need an invoking word or wake word to activate the listener, followed by the command. We have so many virtual assistants, such as Apple's Siri, Amazon's Alexa and Google assistant for android.

1.1.1 Objectives :- Perform task of Assistant virtually through software

1.1.2 Problem statement :- In such an era of advancement if people are still struggling to interact with their machine using various input devices then it's not worth it.

1.1.3 Solution :- many voice assistants were developed and are still being improved for better performance and efficiency. The main task of a voice assistant is to minimize the use of input devices like keyboard, mouse, touch pens, etc. This will reduce both the hardware cost and space taken by it.

1.1.4 The project aims :-To develop a personal-assistant for Linux and windows-based systems. Virtual assistant draws its inspiration from virtual assistants like Google assistant for android, and Siri for iOS. It has been designed to provide a user-friendly interface for carrying out a variety of tasks by employing certain well-defined commands. Users can interact with the assistant either through voice commands or using keyboard input.

CHAPTER 2 : LITERATURE REVIEW

2.1 Introduction

2.1.1 Python

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages. Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, Data science etc. Python has a lot of libraries for every need of this project. For JARVIS, libraries used are speech recognition to recognize voice, Pyttsx3 for text to speech, selenium for web automation 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.

2.1.2 Pyttsx3

Pyttsx3 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 Mac OS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantage is that it works offline.

2.1.3 Speech Recognition

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

CHAPTER 3: PRELIMINARY DESIGN

3.1 Methodology Virtual assistant as the name itself suggests is the assistant of the user which assists the user by performing the tasks which in turn increases the ease at which work is being carried out. The proposed Virtual Assistant can perform many tasks such the user feels very relaxed while performing the tasks as it substantially reduces the effort needed. It can control sound according to the given command. It is capable of forecasting the weather and then it suggests to the user whether one should go out or not. It plays the song according to the mood of the user. It can also open and search in google according to the user requirement and command. It is capable of opening any browser and sites like Wikipedia according to the user's interest. It automatically moves the computer in sleep mode when not in use for 30 sec hence it saves power. It takes a screenshot of the display, when asked for "capture", "capture my screen", "my screen", "screenshot", "take screenshot", the assistant captures the display and stores it in the path specified. It automatically takes the cursor to the online class interface and joins the class according to one's timetable. It continuously monitors the face and eyes of a user and if it finds that a user is drowsy or has moved out of the screen for a long time then it pauses the program currently being used by the user and moves the pc to sleep mode. For better navigation of the user, it can open a map and thus helps in better accessibility.

3.2 Libraries Used

3.2.1 **Python Speech Recognition:** Recognition speech translates the verbal words into written text. Python supports several speech recognition engines like Microsoft Bing Voice Recognition, Google Speech Recognition, etc

3.2.2 **Python Texts To Speech v3(Pyttsx3):** This conservation library in Python translates text to speech. Working offline, this library is compatible with other Python versions of 2 and 3

3.2.3 **Python Audio (PyAudio):** With Pyaudio from Python, you can play and record audio quickly on several devices.

3.2.4 **pywhatkit:** It is python library to send WhatsApp message at a particular time with some additional features.

3.2.5 **Datetime:** This library provides us the actual date and time.

3.2.6 **Wikipedia:** It is a python module for searching anything on Wikipedia.

3.2.7 **Webbrowser:** It provides interface for displaying web-based documents to users.

3.2.8 **os:** It represents Operating System related functionality.

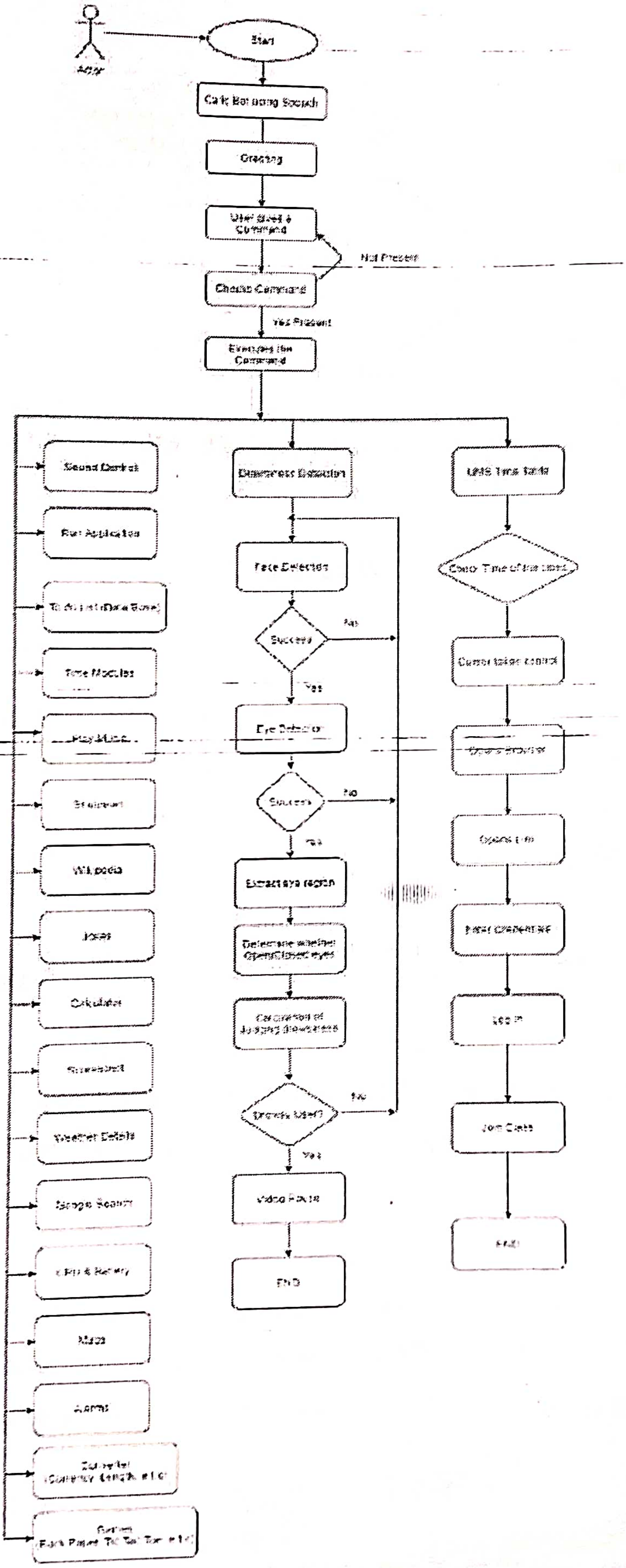


Fig. 3.2 Flow Chart

CHAPTER 4 : Methodology of Virtual Assistant Using Python

4.0. Introduction

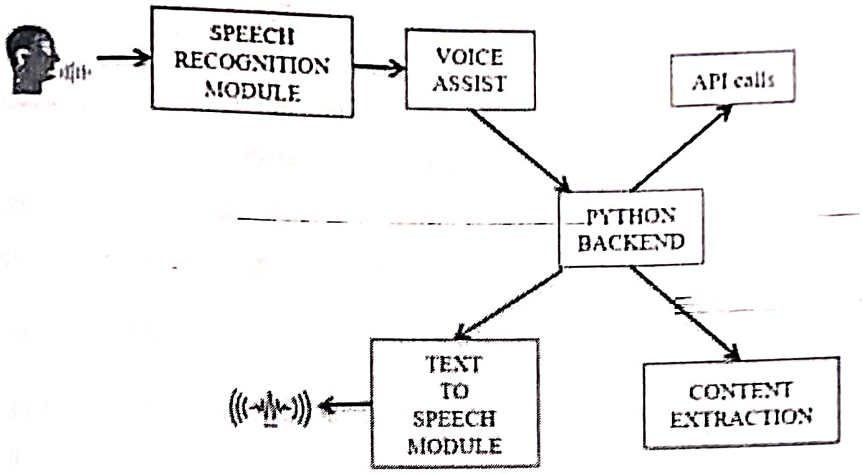


Fig. 4.0 Working

4.1 Speech Recognition module

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 centre 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.

4.2 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.

4.3 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.

4.4 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.

4.5 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.

4.6 Proposed Plan of Work

The work started with analyzing the audio commands given by the user through the microphone. This can be anything like getting any information, operating a computer's internal files, etc. 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.

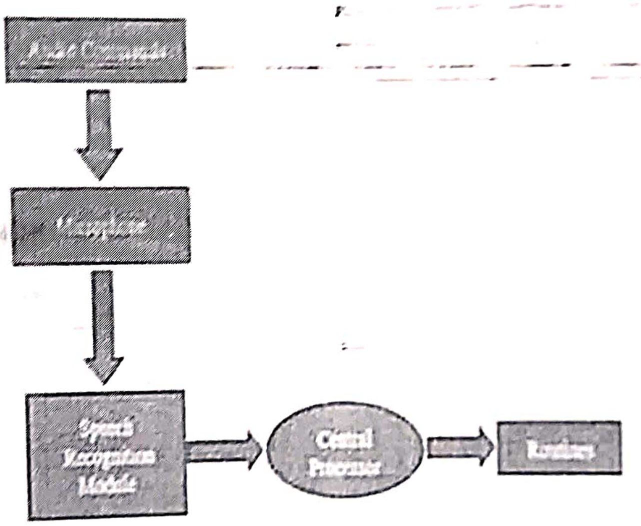


Fig. 4.6 Proposed Work Model

CHAPTER 5: PROJECT IMPLEMENTATION

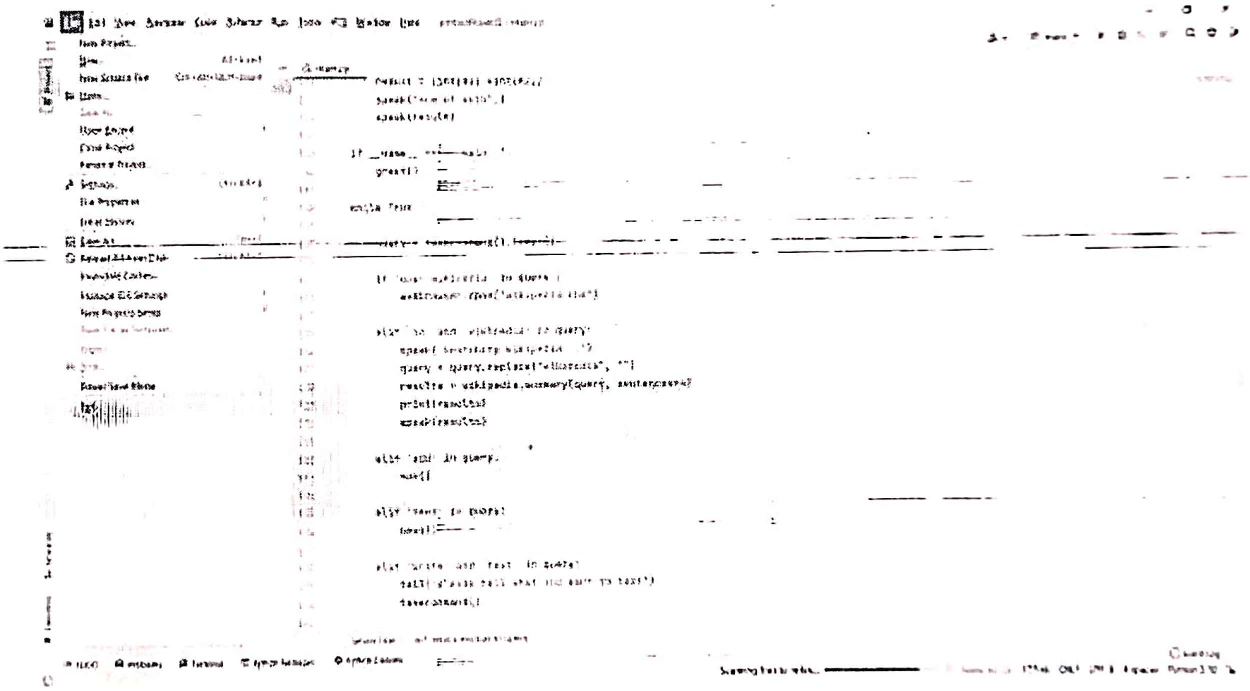
5.1 What can this A.I. assistant do for you ?

- It can send emails on your behalf.
- It can play music for you.
- It can do Wikipedia searches for you.
- It is capable of opening websites like Google, Youtube, etc., in a web browser.
- It is capable of opening your code editor or IDE with a single voice command.

5.2 Start pycham codes

I use pycharm IDE for my project start new project and make a file name virtual assistant

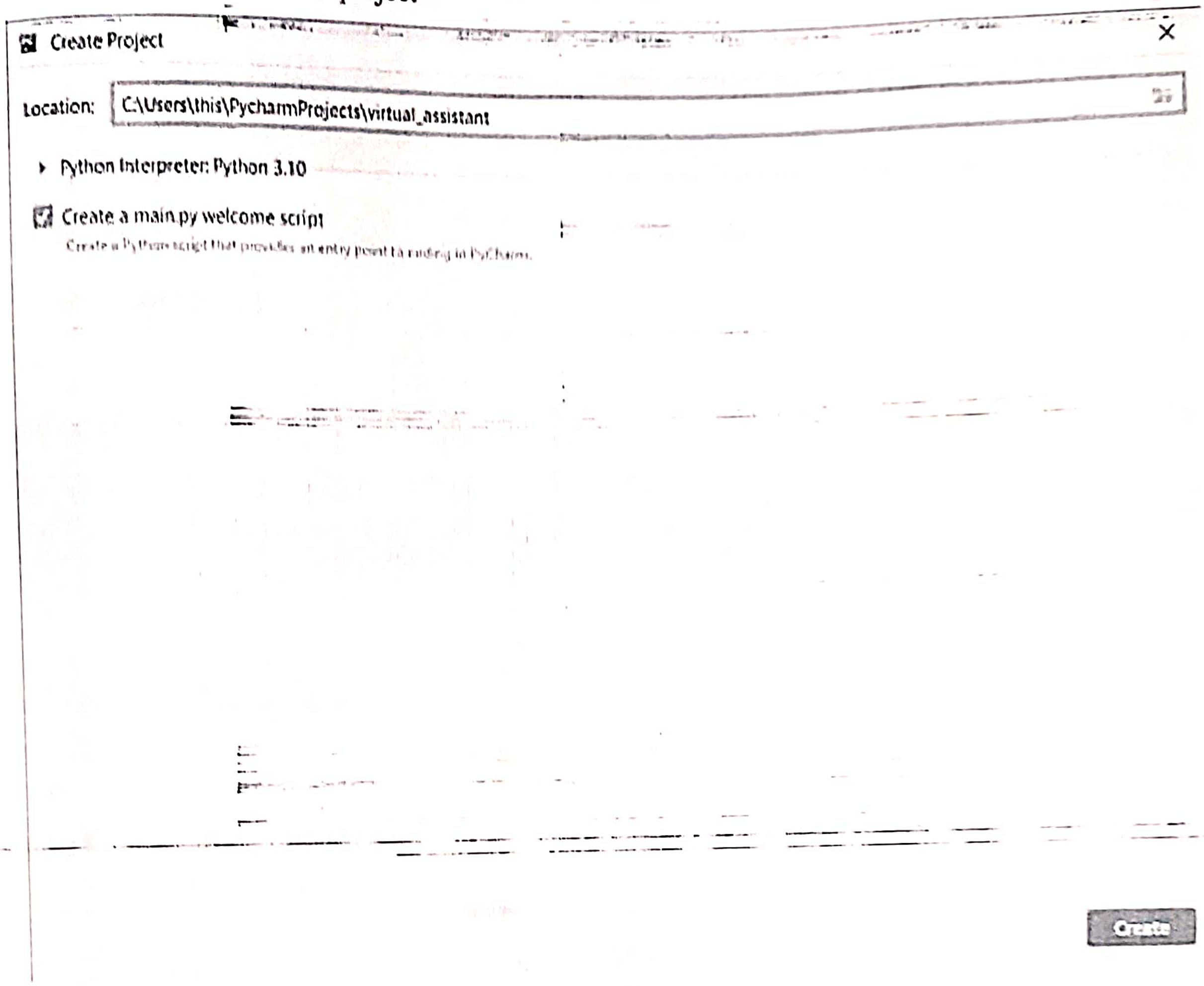
Step 1 :-Go to files and select new project



```
        [illegible]
        speak([illegible])
        speak(result)

if __name__ == "__main__":
    greet()

    while True:
        query = [illegible]

        if 'open wikipedia' in query:
            [illegible]

        elif [illegible] in query:
            speak('Searching Wikipedia...')
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=2)
            print(results)
            speak(results)

        elif [illegible] in query:
            [illegible]

        elif [illegible] in query:
            [illegible]

        elif [illegible] in query:
            [illegible]
            [illegible]
```

Fig. 5.2.1 Project Step 1

Step 2:- Then name the new project



Create Project

Location: C:\Users\this\PycharmProjects\virtual_assistant

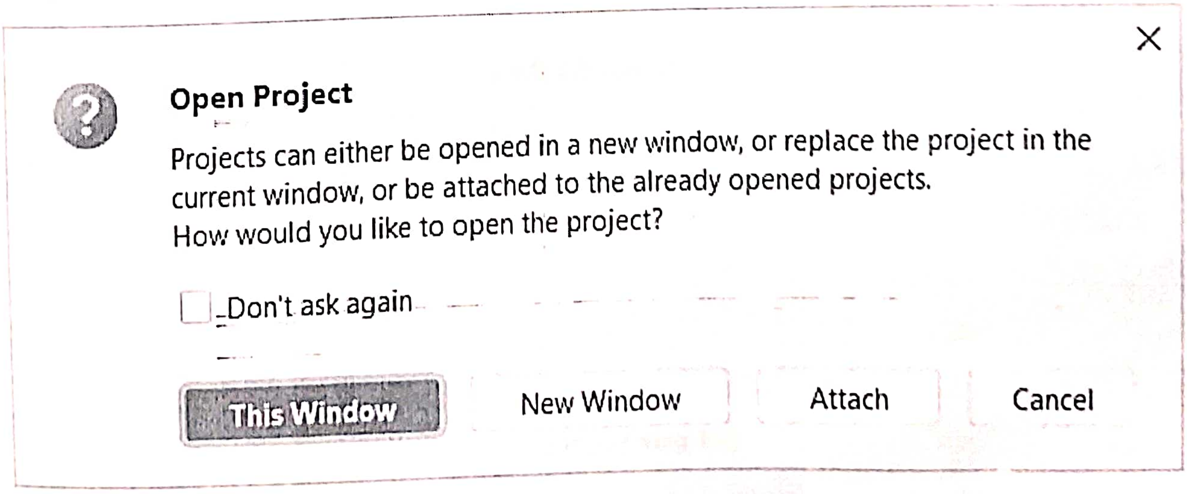
Python Interpreter: Python 3.10

☑ Create a main.py welcome script

Create a Python script that provides an entry point to coding in PyCharm.

Fig. 5.2.2 Project Step 2

Step 3:- Then open in new window



Open Project

Projects can either be opened in a new window, or replace the project in the current window, or be attached to the already opened projects.
How would you like to open the project?

□ Don't ask again

Fig. 5.2.3 Project Step 3

Step 4 :- Then again go to new and go to new then select python file

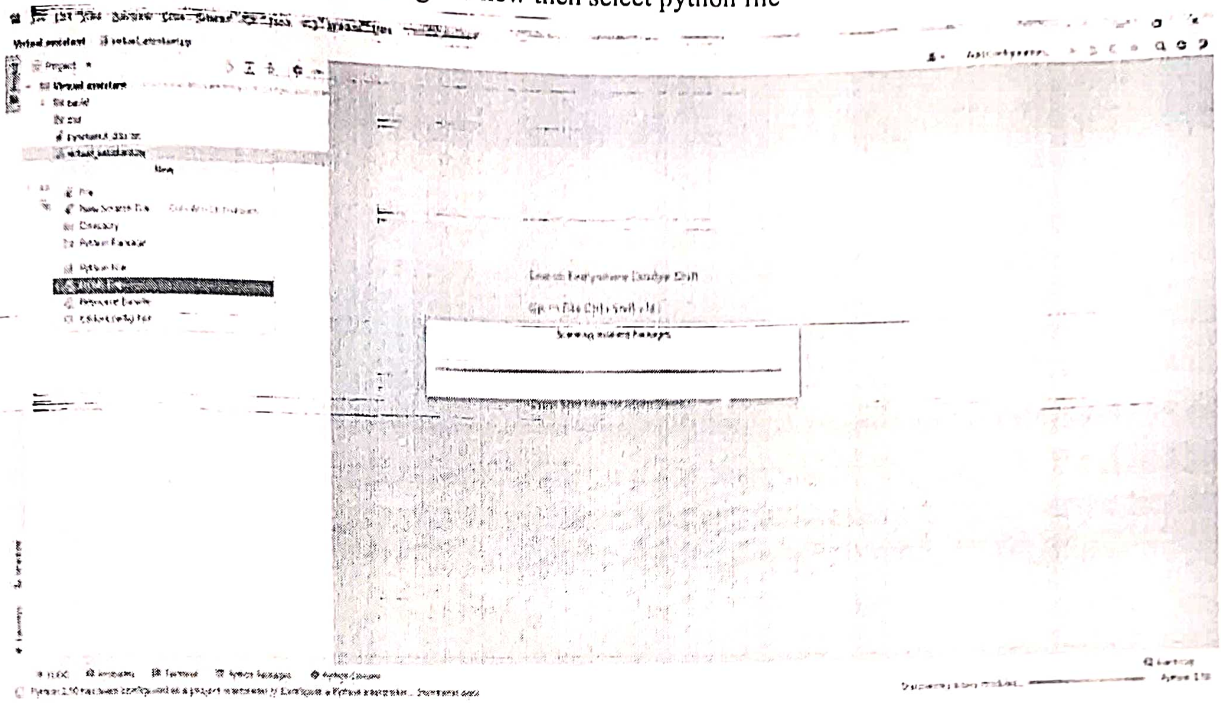


Fig. 5.2.4 Project Step 4

Step 5 :- then name python file.py

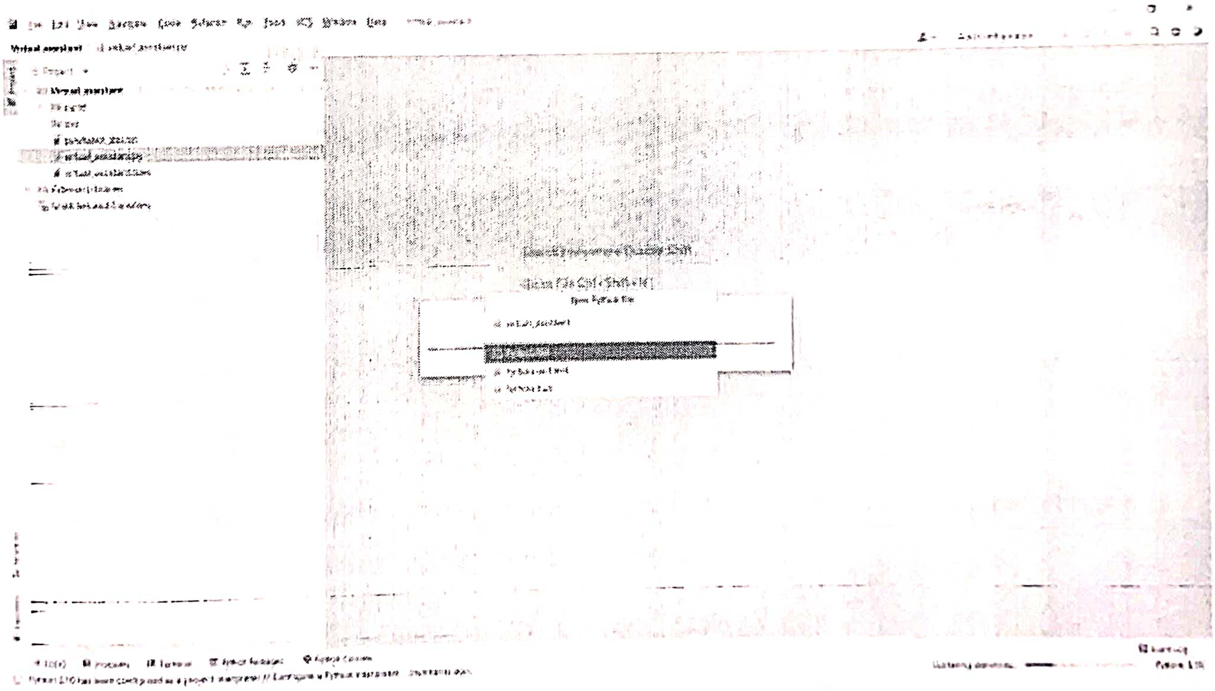
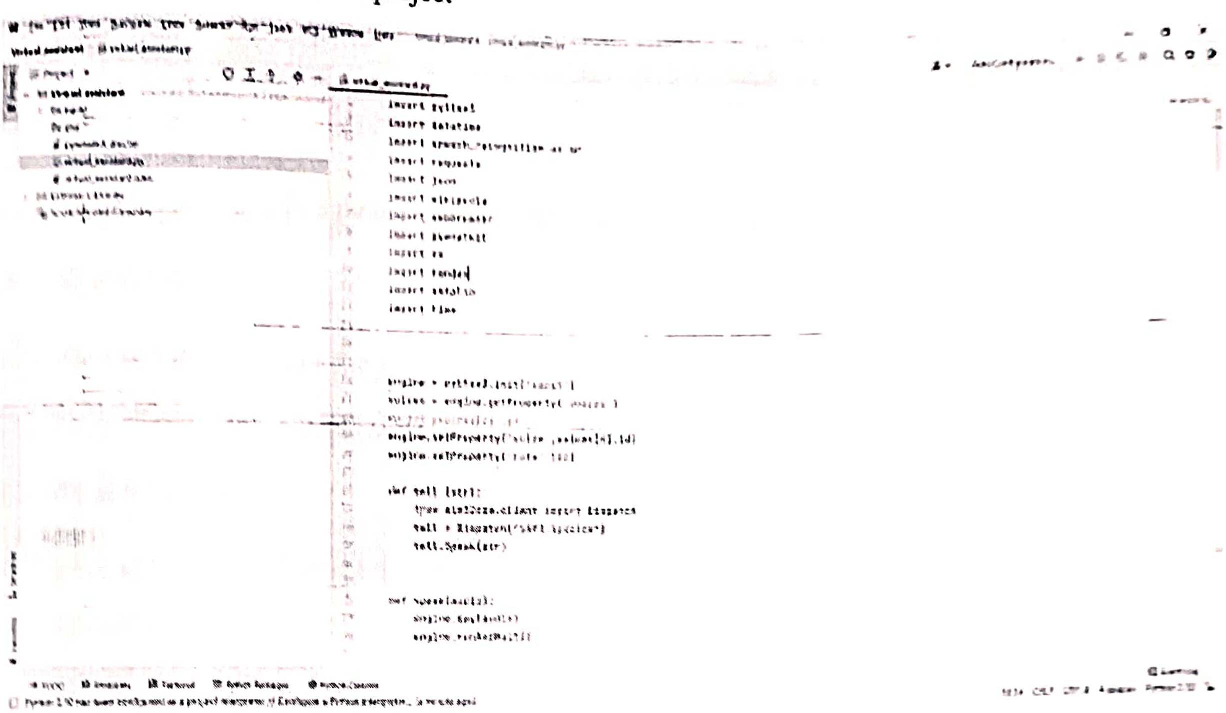


Fig. 5.2.5 Project Step 5

Step 6:- Then write the code of project



```
import pyttsx3
import datetime
import speech_recognition as sr
import requests
import json
import wikipedia
import webbrowser
import pywhatkit
import os
import [illegible]
import smtplib
import time


engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
[illegible]
engine.setProperty('voice',voices[0].id)
engine.setProperty('rate', [illegible])

def talk(str):
    from win32com.client import Dispatch
    speak = Dispatch("SAPI.SpVoice")
    speak.Speak(str)


def speak(audio):
    engine.say(audio)
    engine.runAndWait()
```

Fig. 5.2.6 Project Step 6

5.3 Def speak function

The first and foremost thing for an A.I. assistant is that it should be able to speak. To make our virtual assistant talk, we will make a function called **speak().** This function will take audio as an argument, and then it will pronounce it

```
def speak(audio):

engine.say(audio)

engine.runAndWait()
```

5.4 What is pyttsx3?

A python library that will help us to convert text to speech. In short, it is a text-to-speech library. It works offline, and it is compatible with Python 2 as well as Python 3.

5.5 Installation:

```
pip install pyttsx3
```

In case you receive such errors:

No module named win32com.client

No module named win32

No module named win32ap

Then, install pypiwin32 by typing the below command in the terminal :

pip install pypiwin32.

After successfully installing pyttsx3, import this module into your program.

5.6 What is sapi 5 ?

- Microsoft development speech API.
- Helps in synthesis and recognition of voice.

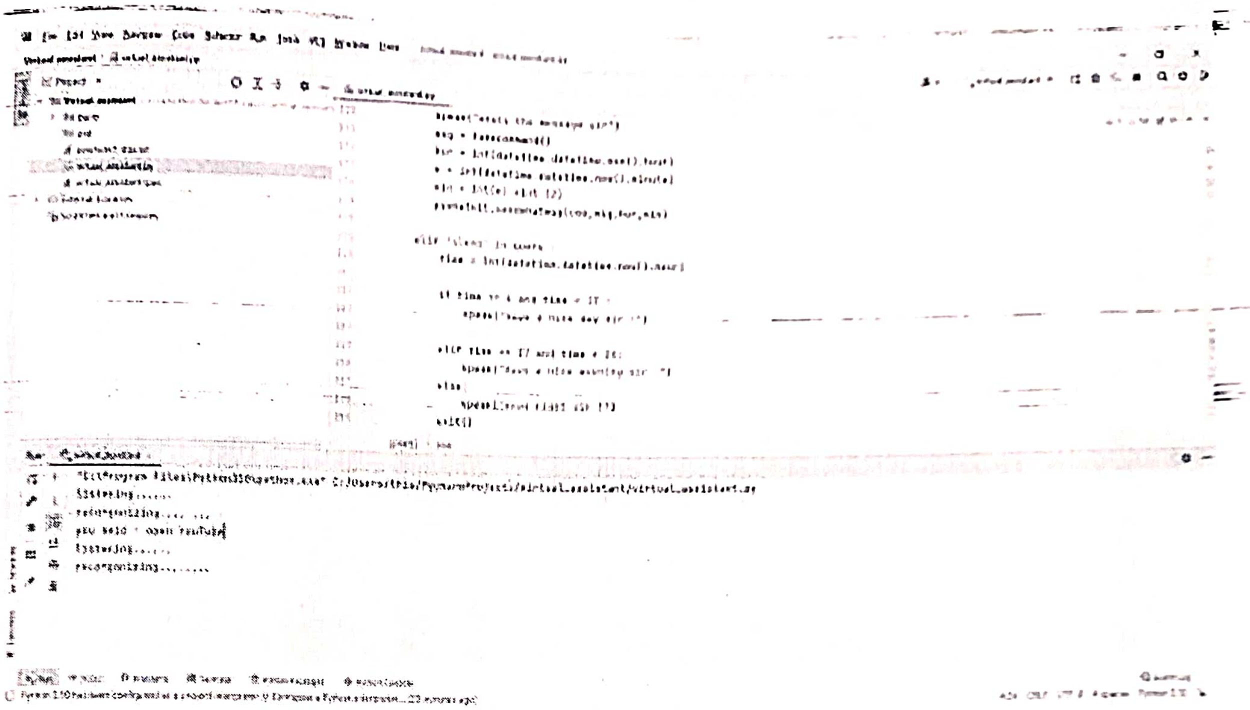
5.7 What is voice id ?

- voice id helps us to select different voices.
- voice[0].id = Male voice
- voice[1].id = Female voice

5.8 To take command :- To take command from user speech reorganization is used

```
def takecommand () :
t = sr.Recognizer()
with sr.Microphone() as source :
print("listening......")
t.energy_threshold=800
t.pause_threshold=1
audio =t.listen(source)
try:
print("recorgnnizing........")
query = t.recognize_google(audio,language='en-in')
print("you said :",query)
```

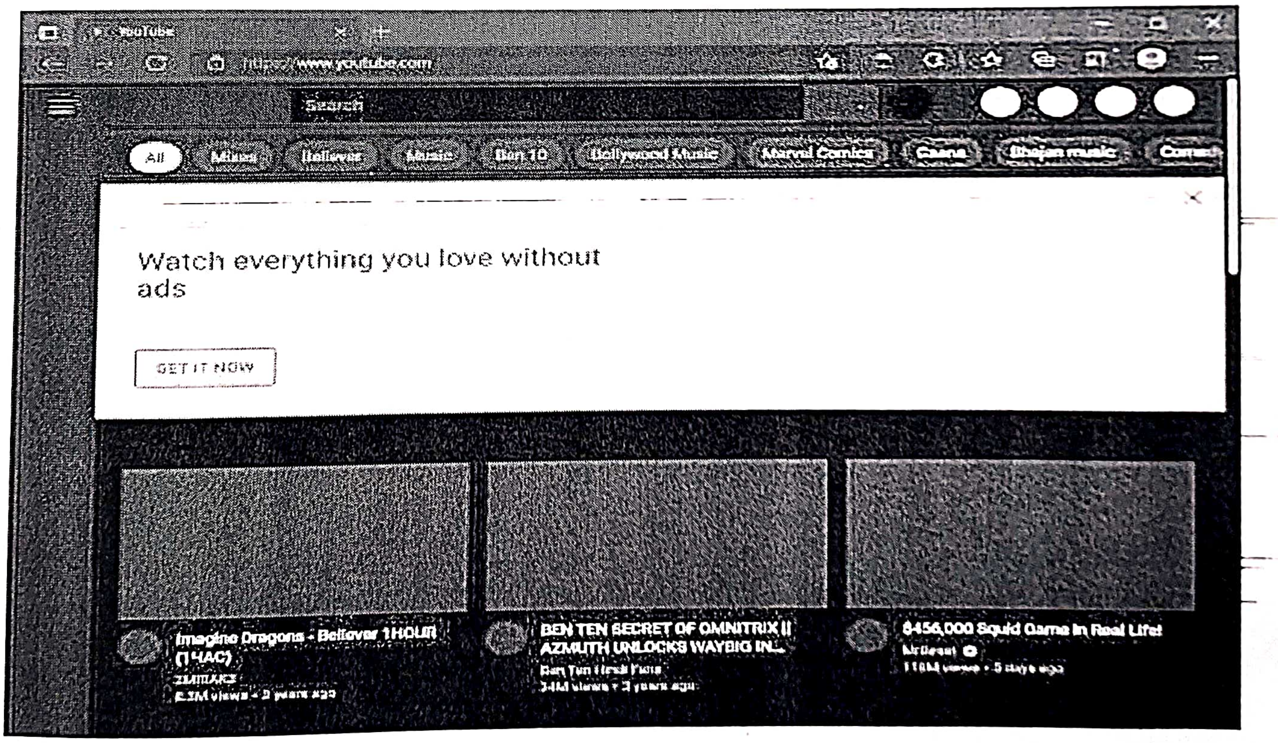
5.9 OUTPUT



```
            speak("state the message sir")
            msg = takecommand()
            hor = int(datetime.datetime.now().hour)
            m = int(datetime.datetime.now().minute)
            min = int(m) +int (2)
            pywhatkit.sendwhatmsg(cno,msg,hor,min)

        elif 'sleep' in query :
            time = int(datetime.datetime.now().hour)

            if time >= 6 and time < 17 :
                speak("have a nice day sir !")

            elif time >= 17 and time < 24:
                speak("have a nice evening sir !")
            else:
                speak("good night sir !")
            exit()
```

```
"C:\Program Files\Python310\python.exe" C:/Users/[illegible]/PycharmProjects/virtual_assistant/virtual_assistant.py
listening......
recognizing.... ...
you said : open youtube
listening.....
recognizing........
```

Fig. 5.9.1 Running Project in IDE



Watch everything you love without ads

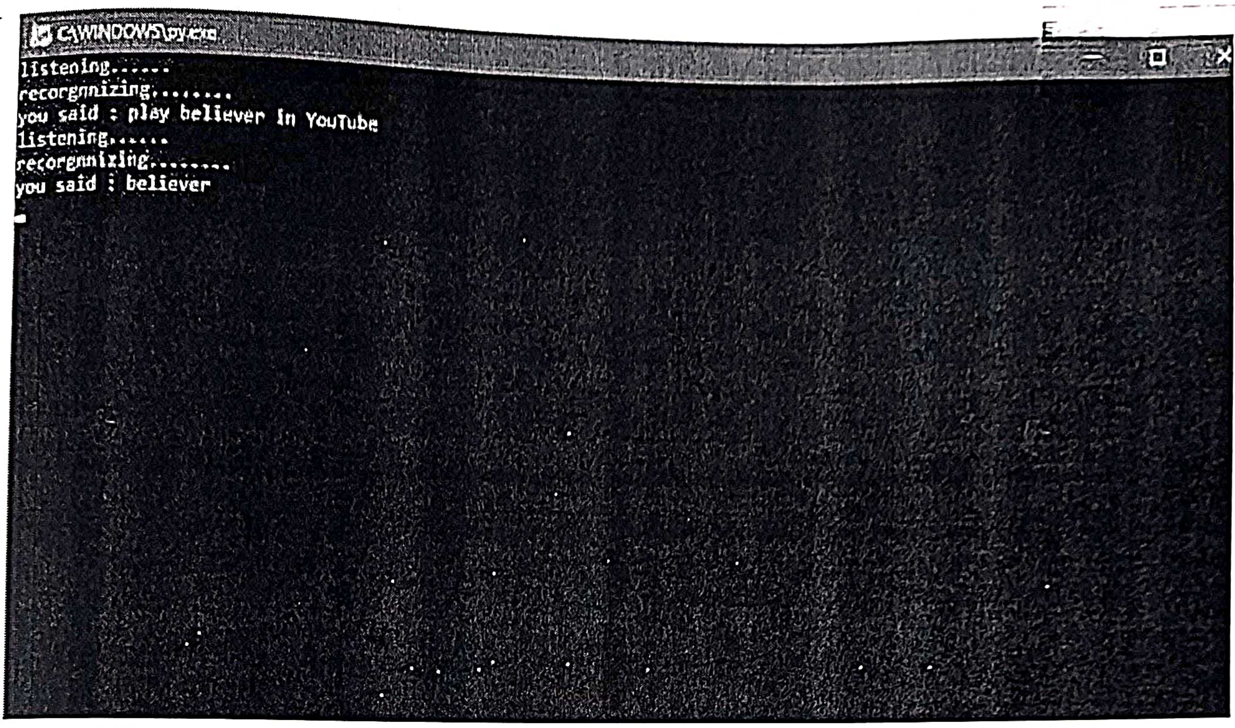
GET IT NOW

Imagine Dragons - Believer 1HOUR (1 ЧАС)
ZLMAKE
8.3M views • 2 years ago

BEN TEN SECRET OF OMNITRIX || AZMUTH UNLOCKS WAYBIG IN...
Ben Ten Hindi Fans
34M views • 3 years ago

$456,000 Squid Game In Real Life!
MrBeast
110M views • 5 days ago

Fig. 5.9.2 Running Project in IDE



```
listening......
recorgnnizing........
you said : play believer in YouTube
listening......
recorgnnizing........
you said : believer
```

Fig. 5.9.3 Project Run in Terminal

CHAPTER 6 : Project Requirement

The system requirements to build virtual assistant are given below.

6.1 Windows-Based Requirements

Computers running Microsoft Windows must meet the following minimum Hardware and Software requirements.

- Microsoft Windows 8.1 or newer(32- or 64- bit) in window 10 it can run without corrupting any program
- 4 GB RAM minimum, 8 GB RAM recommended;
- 2 GB of available disk space minimum, 4 GB Recommended
- 1280 * 800 minimum screen resolution
- it requires processor i3 7th gen and more for running

6.2 Software requirements

- python 3.7 or newer

6.3 Hardware Requirements

- Laptop / Computer
- it requires net seed more than 1.5Mbps (for searching news or fetch information from internet)

CHAPTER 7 : CONCLUSION

Virtual assistant is a very helpful voice assistant without any doubt as it saves time of the user by conversational interactions, its effectiveness and efficiency. But while working on this project, there were some limitations encountered and also realized some scope of enhancement in the future which are mentioned below .

7.1 Limitations

7.1.1 Security is somewhere an issue, there is no voice command encryption in this project.

7.1.2 Background voice can interfere.

7.1.3 Misinterpretation because of accents and may cause inaccurate results.

7.1.4 Voice Assistant cannot be called externally anytime like other traditional assistants like Google Assistant can be called just by saying, “Ok Google!”

7.2 Scope for Future Work

7.2.1 Make Virtual assistant to learn more on its own and develop a new skill in it.

7.2.2 Virtual assistant android app can also be developed.

7.2.3 Voice commands can be encrypted to maintain security

REFERENCES

- R. Belvin, R. Burns, and C. Hein, "Development of the HRL route navigation dialogue system," in Proceedings of ACL-HLT, 2001
- V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T.J.Hazen,and L.Hetherington, "JUPITER: A Telephone Based Conversational Interface for Weather Information," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 85–96, 2000.
- M. Kolss, D. Bernreuther, M. Paulik, S. St"ucker, S. Vogel, and A. Waibel, "Open Domain Speech Recognition & Translation: Lectures and Speeches," in Proceedings of ICASSP, 2006.