

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By

SAQIB AHAD KHAN
40731086



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

APRIL 2023



**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**



Accredited with Grade “A” by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI– 600119
www.sathyabama.ac.in

DEPARTMENT OF CSE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide Work Of **Saqib Ahad Khan** (Reg. No: 40731086) who carried out the project entitled “Voice To Text tool” under my supervision from January 2023 to April 2023.

Internal Guide

Dr.K.Ishwarya M.E., Ph.D.

HEAD OF THE DEPARTMENT

Dr.L.LAKSHMANAN M.E., Ph.D.

Submitted for Viva voce Examination held on 17-04-2023

Internal Examiner

External Examiner

DECLARATION

I, **Saqib Ahad Khan** hereby declare that the Project Report entitled “**Voice To Text Tool**”_done by me under the guidance of **Dr.K.Ishwarya M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE: 17-04-2023

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.T.Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr.S.Vigneshwari M.E., Ph.D., and Dr.L.Lakshmanan M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.K.Ishwarya M.E., Ph.D.** for his valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

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 is 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. Functionalities of this project include, It can send emails, It can read PDF, It can open command prompt, your favourite IDE, notepad etc. It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube, etc., in a web browser, It can give weather forecast, It can give desktop reminders of your choice. Now the basic question arises in mind that how it is an AI? The virtual assistant that I have created is like if it is not an A.I, but it is the output of a bundle of the statement. But fundamentally, the main purpose of A.I machines is that it can perform human tasks with the same efficiency or even more efficiently than humans. It is a fact that my virtual assistant is not a very good example of A.I., but it is an A.I.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	5
	LIST OF FIGURES	8
	INTRODUCTION	9
1.	1.1 PRESENT SYSTEM	10
	1.2 PROPOSED SYSTEM	11
	1.3 REAL LIFE APPLICATION	12
2.	AIM AND SCOPE OF THE PRESENT INVESTIGATION	13
	2.1 AIM	13
	2.2 SCOPE:	13
	2.3 DATA PREPARATION	14
	2.4. Data Flow	15

3.	EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED	16
	3.1 METHODS USED	16
	3.2 METHODS	16
	3.3 FUNCTIONS	16
	3.4 IMPORTED LIBRARIES	17
	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	22
4.	4.1. FUNCTIONALITY	22
	4.2. USABILITY	22
	4.3 SECURITY	23
	4.4. STABILITY	23
5.	SUMMARY AND CONCLUSIONS	24
	5.1 SUMMARY AND CONCLUSIONS	24
	REFERENCES	25
	APPENDIX	
	A. SCREENSHOTSANDOUTPUTS	27
	B. SOURCE CODE	30

LIST OF FIGURES

FIGURENO.	FIGURENAME	PAGENO.
1	1.1 present system design	12
2	2.3 data preparation	14
3	2.4 data flow for application	15
4	3.4.1- various python libraries	17
5	3.4.2- libraries used	18
6	3.4.3-Speak Function	19
7	3.4.4-SR Function	20
8	3.4.5-Audio to text function	21

Chapter 1 INTRODUCTION

Artificial Intelligence when used with machines, it shows us the capability of thinking like humans. In this, a computer system is designed in such a way that typically requires interaction from human. 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 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 is 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.

As the voice assistant is using Artificial Intelligence hence the result that it is providing are highly accurate and efficient. The assistant can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. The assistant is no less than a human assistant but we can say that this is more effective and efficient to perform any task. The libraries and packages used to make this assistant focuses on the time complexities and reduces time.

The functionalities include, it can send emails, It can read PDF, It can send text on WhatsApp, It can open command prompt, your favourite IDE, notepad etc., It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube, etc., in a web browser, It can give

weather forecast, It can give desktop reminders of your choice. It can have some basic conversation.

Tools and technologies used are VSCODE for making this project, and I created all py files in VSCODE. Along with this I used following modules and libraries in my project. pytsx3, SpeechRecognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, tkinter etc. I have created a live GUI for interacting with the Assistant as it gives a design and interesting look while having the conversation.

1.1 PRESENT SYSTEM

We are familiar with many existing voice assistants like Alexa, Siri, Google Assistant, Cortana which uses concept of language processing, and voice recognition. They listen the command given by the user as per their requirements and performs that specific function in a very efficient and effective manner.

As these voice assistants are using Artificial Intelligence hence the result that they are providing are highly accurate and efficient. These assistants can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. These assistants are no less than a human assistant but we can say that they are more effective and efficient to perform any task. The algorithm used to make these assistant focuses on the time complexities and reduces time.

But for using these assistants one should have an account (like Google account for Google assistant, Microsoft account for Cortana) and can use it with internet connection only because these assistants are going to work with internet connectivity. They are integrated with many devices like, phones, laptops, and speakers etc.

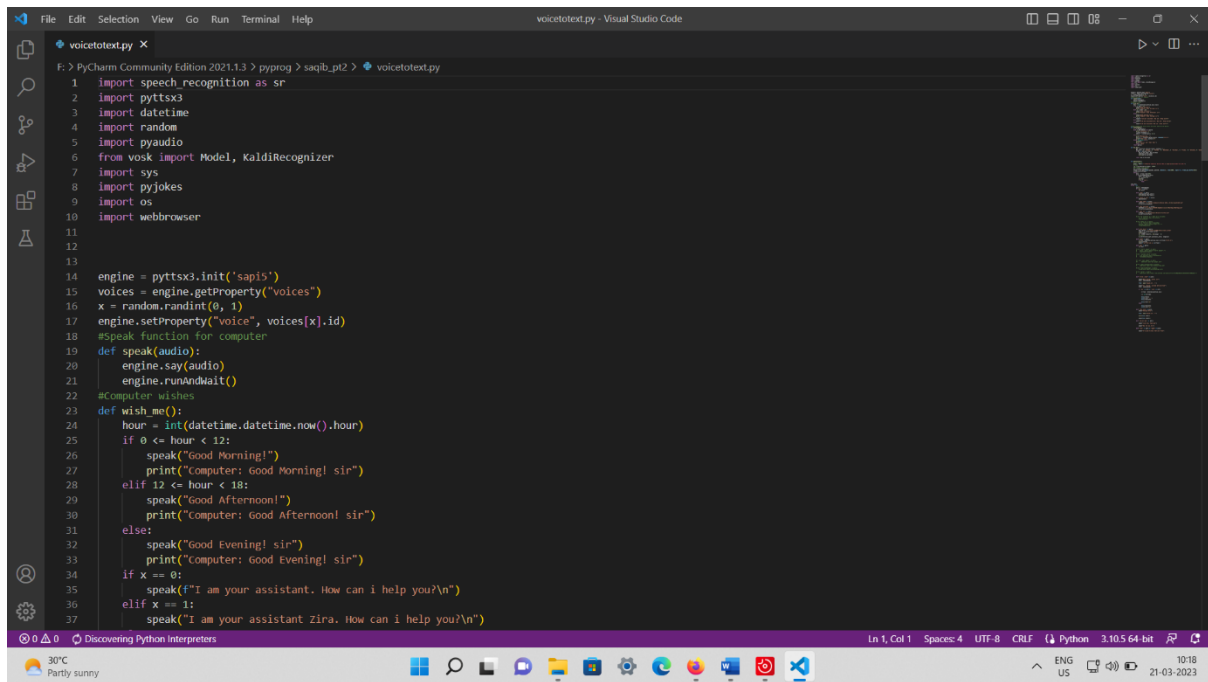


Fig 1.1 present system design

1.2 PROPOSED SYSTEM

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. Jarvis is different from other traditional voice assistants in terms that it is specific to desktop and user does not need to make account to use this, it does not require any internet connection while getting the instructions to perform any specific task.

The IDE used in this project is VSCODE. All the python files were created in VSCODE and all the necessary packages were easily installable in this IDE. For this project following modules and libraries were used i.e., pyttsx3, SpeechRecognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, tkinter etc. I have created a live GUI for interacting with the Assistant as it gives a design and interesting look while having the conversation.

With the advancement Assistant can perform any task with same effectiveness or can say more effectively than us. By this project, I realized that the concept of AI in every field is decreasing human effort and saving time. Functionalities of this project include, it can send emails, it can read PDF, it can send text on WhatsApp, it can open command prompt, your favourite IDE, notepad etc., It can play music, it can do Wikipedia searches for you.

1.3 REAL LIFE APPLICATION

- **Saves time:** This is a desktop assistant which works on the voice command offered to it, it can do voice searching, voice-activated device control and can let us complete a set of tasks.
- **Conversational interaction:** It makes it easier to complete any task as it automatically does it by using the essential module or libraries of Python, in a conversational interaction way. Hence any user when instruct any task to it, they feel like giving task to a human assistant because of the conversational interaction for giving input and getting the desired output in the form of task done.
- **Reactive nature:** The desktop assistant is reactive which means it know human language very well and understand the context that is provided by the user and gives response in the same way, i.e., human understandable language, English. So, user finds its reaction in an informed and smart way.
- **Multitasking:** The main application of it can be its multitasking ability. It can ask for continuous instruction one after other until the user “QUIT” it.

CHAPTER-2 AIM AND SCOPE OF PRESENT INVESTIGATION

2.1 AIM

To create a voice-based application using python which is used to control the computer over the voice.

2.2 SCOPE

In our daily life, we use our computer to do our personal works and do our office works using keyboard and mouse, and many of the people cannot access the computer using the mouse and keyboard, because they are disabled or do not know how to use the computer, so we have created the application which can access the system using the voice commands.

The project aims to develop a personal-assistant for window-based systems, which can also be used in Linux-based systems to control the system over the voice. It has been designed to provide a user-friendly interface for carrying out a variety of tasks by employing certain well-defined commands.

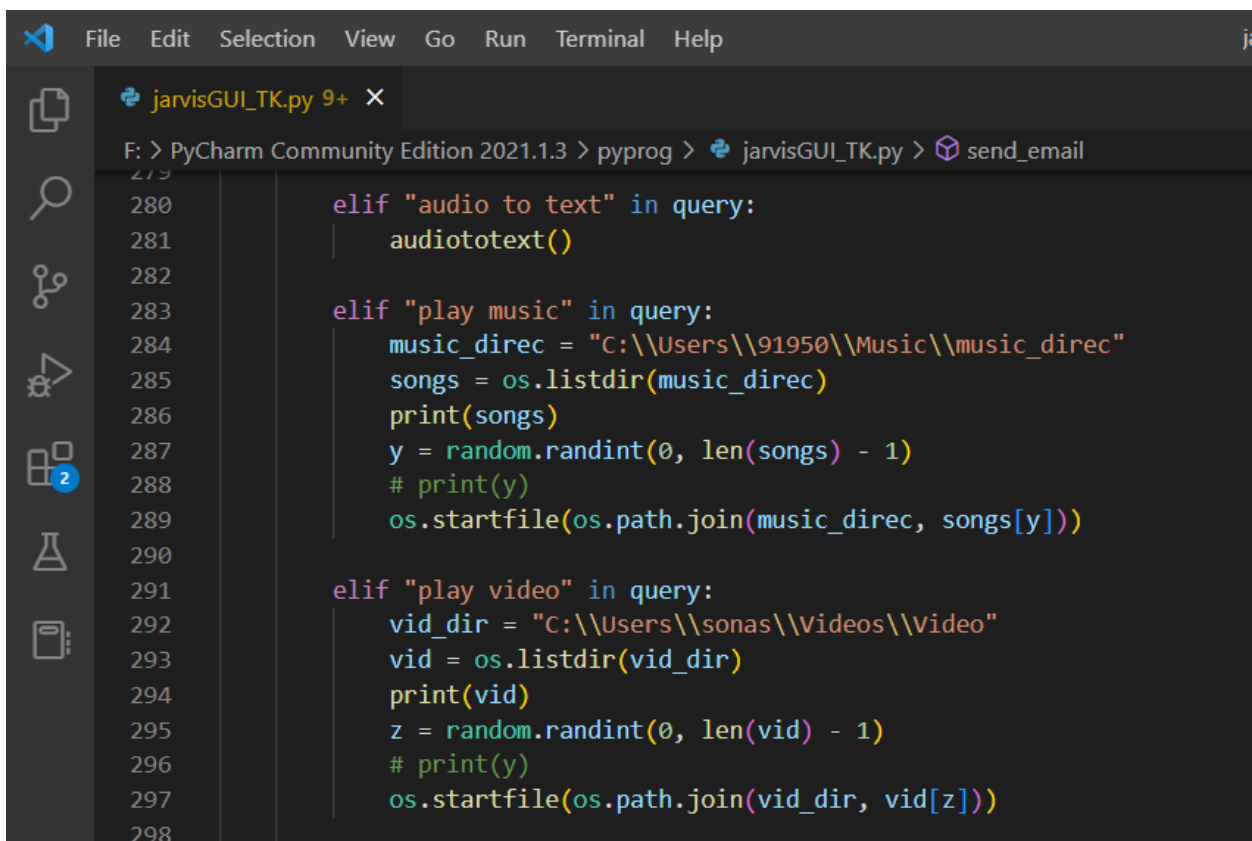
Automating repeated tasks to a voice-activated applications frees up the human time and resources. And saves time of humans in do the work as compared without the application.

- Make the Program to learn more on its own and develop a new skill in it.
- The Application can also be developed for the android user to control their phones.
- Voice commands can be encrypted for the maintenance of security of the system.
- Search Engine with voice interactions and Reminder and To-Do application.

2.3 DATA PREPARATION

In this project as a backbone tool python is used to carry out the applications program. With the help of a software called Visual Studio Code, It is an IDE i.e., Integrated Development Environment which has many features like it supports scientific tools (like matplotlib, NumPy, SciPy etc) web frameworks (example Django, web2py and Flask) refactoring in Python, integrated python debugger, code completion, code and project navigation etc. It also provides Data Science when used with Anaconda. And, it has an open-source web application that allows to create and share documents and has live code and visualization.

After importing the required libraries, the data needed for the project is been taken from the files in the computer, here we have used just the hard disk as the source of data but in future we are trying to improve it to an where it runs on the database.



```
File Edit Selection View Go Run Terminal Help
jarvisGUI_TK.py 9+ X
F: > PyCharm Community Edition 2021.13 > pyprog > jarvisGUI_TK.py > send_email
279
280     elif "audio to text" in query:
281         audiototext()
282
283     elif "play music" in query:
284         music_direc = "C:\\Users\\91950\\Music\\music_direc"
285         songs = os.listdir(music_direc)
286         print(songs)
287         y = random.randint(0, len(songs) - 1)
288         # print(y)
289         os.startfile(os.path.join(music_direc, songs[y]))
290
291     elif "play video" in query:
292         vid_dir = "C:\\Users\\sonas\\Videos\\Video"
293         vid = os.listdir(vid_dir)
294         print(vid)
295         z = random.randint(0, len(vid) - 1)
296         # print(y)
297         os.startfile(os.path.join(vid_dir, vid[z]))
298
```

. Fig 2.3 data preparation

2.4. Data Flow

The data flow for application is as follow:

The system is designed using the concept of Artificial Intelligence and with the help of necessary packages of Python. Python provides many libraries and packages to perform the tasks, for example pyPDF2 can be used to read PDF. The details of these packages are mentioned in Chapter 3 of this report.

The data in this project is nothing but user input, whatever the user says, the application performs the task accordingly. The user input is nothing specific but the list of tasks which a user wants to get performed in human language i.e., English.

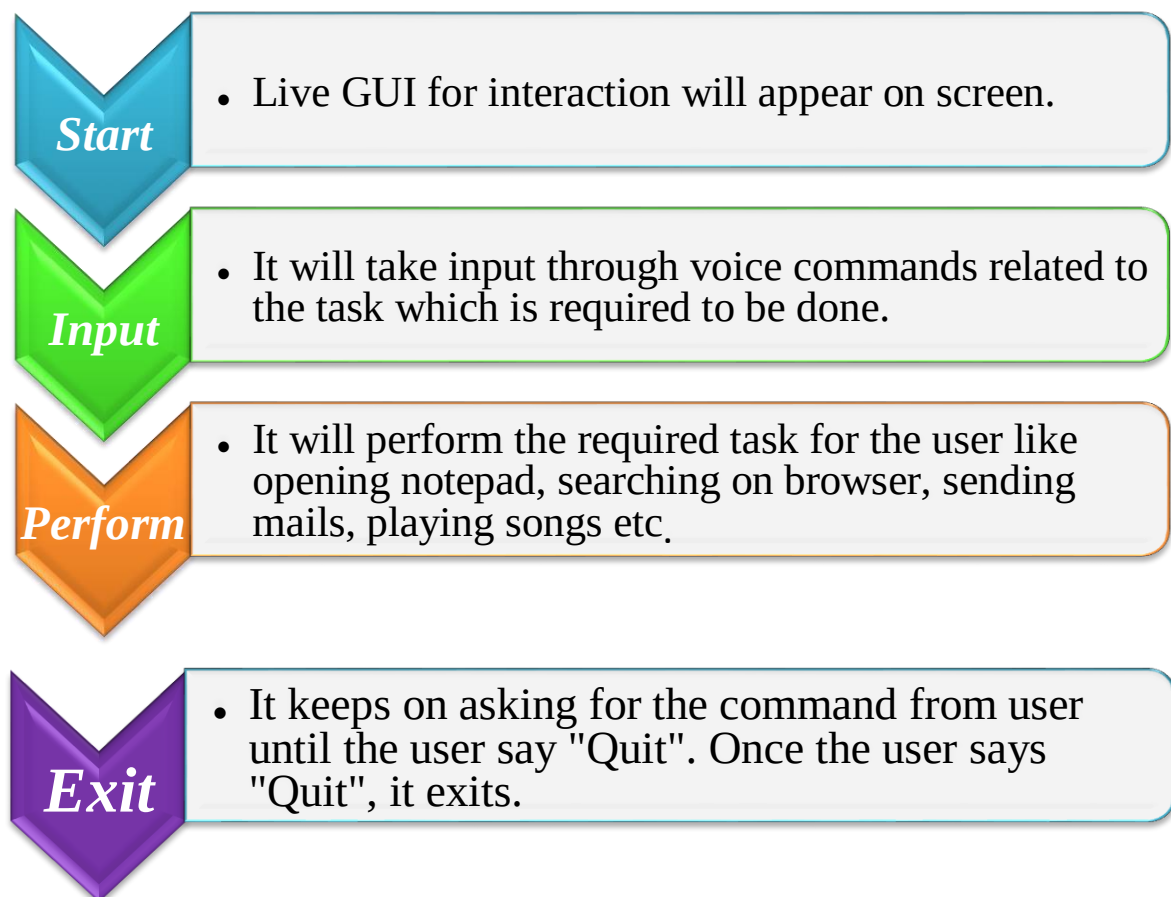


Figure 2.4 data flow for application

CHAPTER-3 EXPERIMENTAL OR MATERIAL AND METHODS, ALGORITHMS USED

3.1 METHODS USED

While loops, for loops, if-else statements and def functions in python

3.2 METHODS

While loop: The while loop in Python is “used to iterate over a block of code as long as the test expression (condition) is true.” We generally use this loop when we do not know the number of times to iterate beforehand.

For loop: A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string). This is less like the for keyword in other programming languages, and works more like an iterator method as found in other object-orientated programming languages.

If-Else statements: The if-else statement is used to execute both the true part and the false part of a given condition. If the condition is true, the if block code is executed and if the condition is false, the else block code is executed.

Def Functions: Python def keyword is used to define a function; it is placed before a function name that is provided by the user to create a user-defined function. In python, a function is a logical unit of code containing a sequence of statements indented under a name given using the “def” keyword.

3.3 FUNCTIONS

TakeCommand(): The function is used to take the command as input through microphone of user and returns the output as string.

WishMe (): This function greets the user according to the time like Good Morning, Good Afternoon and Good Evening.

TaskExecution (): This is the function which contains all the necessary task execution definition like sendEmail(), PDF_Reader(), news() and many conditions in if condition like “open google”, “open notepad”, “search on Wikipedia” ,”play music” and “open command prompt” etc.

Speak(): This is the Function which contains all the necessary commands for the computer to make it speak and convert the text to speech.

3.4 IMPORTED LIBRARIES

Libraries are collections of prewritten code that users can use to optimize tasks. In project as python is used for implementation tool, it has the most libraries as compared to other programming languages. More than of 60% machine learning developers use and goes for python as it is easy to learn. As python has comparatively large collection of libraries let us look at the libraries that came in handy for mammographic dataset.

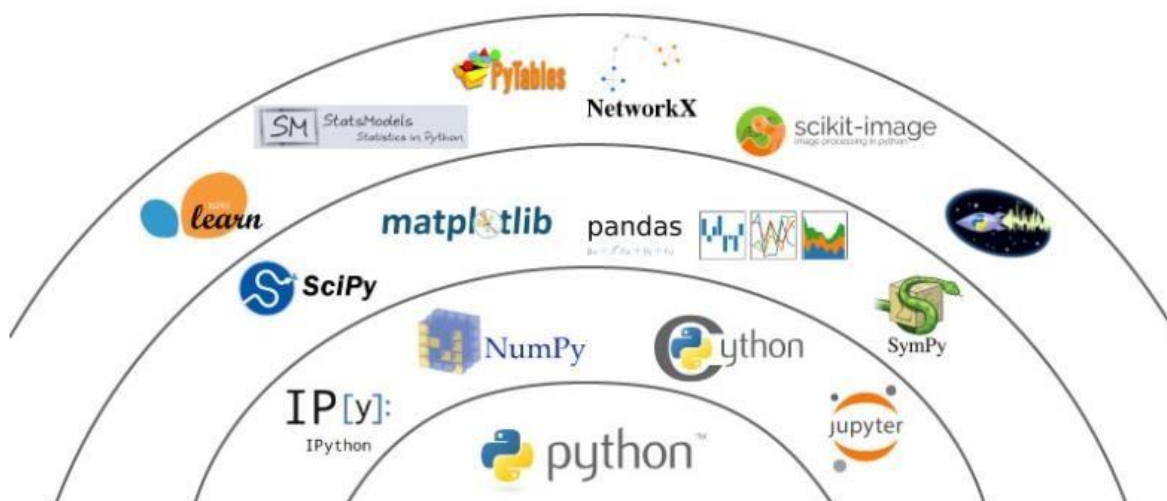
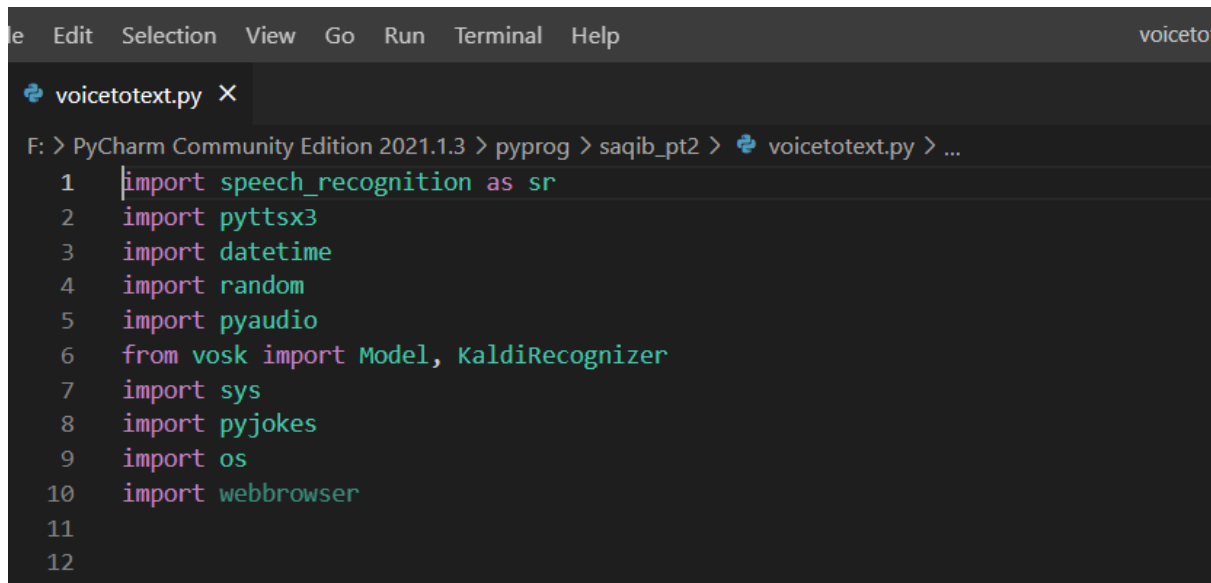


Fig 3.4.1- various python libraries

PYTHON LIBRARIES

A screenshot of the PyCharm Community Edition 2021.1.3 IDE. The top menu bar shows 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', 'Terminal', and 'Help'. The breadcrumb path at the top reads 'F: > PyCharm Community Edition 2021.1.3 > pyprog > saqib_pt2 > voicetotext.py > ...'. The editor window shows the file 'voicetotext.py' with the following Python code:

```
1 import speech_recognition as sr
2 import pyttsx3
3 import datetime
4 import random
5 import pyaudio
6 from vosk import Model, KaldiRecognizer
7 import sys
8 import pyjokes
9 import os
10 import webbrowser
11
12
```

Fig 3.4.2- libraries used in the project

In Application following python libraries were used:

pyttsx3: It is a python library which converts text to speech. 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. An application invokes the `pyttsx3.init()` factory function to get a reference to a `pyttsx3`. Engine instance. it is a very easy to use tool which converts the entered text into speech. The `pyttsx3` module supports two voices first is female and the second is male which is provided by “sapi5” for windows.

It supports three TTS engines:

- *sapi5* – SAPI5 on Windows
- *nsss* – NSSpeechSynthesizer on Mac OS X
- *espeak* – eSpeak on every other platform

```

48
49 engine = pyttsx3.init('sapi5')
50 voices = engine.getProperty("voices")
51 x = random.randint(0, 1)
52 # print(voices[x].id)
53 # print(x)
54 engine.setProperty("voice", voices[x].id)
55
56 def speak(audio):
57     engine.say(audio)
58     engine.runAndWait()
59

```

Fig 3.4.3-Speak Function

SpeechRecognition: It is a python module which converts speech to text. Speech recognition is a machine's ability to listen to spoken words and identify them. You can then use speech recognition in Python to convert the spoken words into text, make a query or give a reply. You can even program some devices to respond to these spoken words. You can do speech recognition in python with the help of computer programs that take in input from the microphone, process it, and convert it into a suitable form.

Speech recognition seems highly futuristic, but it is present all around you. Automated phone calls allow you to speak out your query or the query you wish to be assisted on; your virtual assistants like Siri or Alexa also use speech recognition to talk to you seamlessly.

Speech recognition is the process of converting spoken words to text. Python supports many speech recognition engines and APIs, including Google Speech Engine, Google Cloud Speech API, Microsoft Bing Voice Recognition, and IBM Speech to Text.

```

76
77 def takeCommand():
78     r = sr.Recognizer()
79     with sr.Microphone() as source:
80         print("Listening...\n")
81         r.pause_threshold = 1
82         audio = r.listen(source, 0, 5)
83
84     try:
85         print("Recognizing...")
86         query1 = r.recognize_google(audio, language="en-in")
87         print(f"User said: {query1}\n")
88
89     except Exception as er:
90         print(er)
91         print("Sorry I can't hear that.")
92         return "None"
93     return query1
94

```

Fig 3.4.4-Speech Recognition Function

VOSK and KALDI: **VOSK** is a speech recognition toolkit that supports over 20 languages (e.g., English, German, Hindi, etc.) and dialects. It works offline and even on lightweight devices like Raspberry Pi. Its portable models are only 50Mb each. However, there are much bigger models available. **KALDI** is a research speech recognition toolkit which implements many states of the art algorithms. **VOSK** is a practical speech recognition library which comes with a set of accurate models, scripts, practices and provides ready to use speech recognition for different platforms like mobile applications or Raspberry Pi.

The Kaldi model used in VOSK is compiled from 3 data sources: dictionary. acoustic model. language model. This is speech recognition module is trained using the datasets of different voices, languages and made it recognize the sounds/voices while speaking in the mic.

```

168
169 def audiototext():
170     model = Model(r"F:\PyCharm Community Edition 2021.1.3\pyprog\vosk-model-en-in-0.5")
171     rec = KaldiRecognizer(model, 16000)
172
173     mic = pyaudio.PyAudio()
174     stream = mic.open(format=pyaudio.paInt16, channels=1, rate=16000, input=True, frames_per_buffer=8192)
175     stream.start_stream()
176
177     while 1:
178         data = stream.read(4096)
179
180         if rec.AcceptWaveform(data):
181             text = rec.Result()
182             p = text[14:-3]
183             print(p)
184
185             if p == 'exit':
186                 break
187

```

Fig 3.4.5-Audio to text function

Datetime: This library provides us the actual date and time. It is a combination of date and time along with the attribute's year, month, day, hour, minute, second, microsecond, and tzinfo. time delta – A duration expressing the difference between two date, time, or datetime instances to microsecond resolution. tzinfo – It provides time zone information objects.

Pyjokes: Python supports creation of random jokes using one of its libraries. Let us explore it a little more, “Pyjokes” is a python library that is used to create one-line jokes for programmers. Informally, it can also be referred as a fun python library which is simple to use.

Web browser: It provides interface for displaying web-based documents to users. In Python, web browser module is a convenient web browser controller. It provides a high-level interface that allows displaying Web-based documents to users. web browser can also be used as a CLI tool.

CHAPTER-4 RESULTS AND PERFORMANCE ANALYSIS

The system testing is done on fully integrated system to check whether the requirements are matching or not. The system testing for desktop assistant focuses on the following four parameters:

4.1. FUNCTIONALITY

In this we check the functionality of the system whether the system performs the task which it was intended to do. To check the functionality each function was checked and run, if it can execute the required task correctly then the system passes in that functionality test. For example, to check whether application can search on Google or not, as we can see in the figure 4.1, user said “Open Google”, then application asked, “What should I search on Google?” then user said, “What is Python”, Application open Google and searched for the required input.

4.2. USABILITY

Usability of a system is checked by measuring the easiness of the software and how user friendly it is for the user to use, how its responses to each query that is being asked by the user.

It makes it easier to complete any task as it automatically does it by using the essential module or libraries of Python, in a conversational interaction way. Hence any user when instruct any task to it, they feel like giving task to a human assistant because of the conversational interaction for giving input and getting the desired output in the form of task done.

The desktop assistant is reactive which means it know human language very well and understand the context that is provided by the user and gives response in the same way, i.e., human understandable language, English. So, user finds its reaction in an informed and smart way.

The main application of it can be its multitasking ability. It can ask for continuous instruction one after other until the user “QUIT” it. It asks for the instruction and listen

the response that is given by user without needing any trigger phase and then only executes the task.

4.3 SECURITY

The security testing mainly focuses on vulnerabilities and risks. As Application is a local desktop application, hence there is no risk of data breaching through remote access. The software is dedicated to a specific system so when the user logs in, it will be activated

4.4. STABILITY

Stability of a system depends upon the output of the system, if the output is bounded and specific to the bounded input then the system is said to be stable. If the system works on all the poles of functionality, then it is stable.

CHAPTER-5

5.1 SUMMARY AND CONCLUSIONS

- We have modelled the Application for the computer using python and methods of while, for loops, functions, and if-else statement.
- In this project the computer system is being controlled over voice and can convert the speech into the text to take input and to make the computer speak the text is converted into speech/voice.
- In the future we can add more categories, add Artificial Intelligence in the application and make it more user friendly and improves its quality. I and my team members are interested in studying these methods and implement in another AI and Machine learning.

This application 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, realized some scope of enhancement in the future which are mentioned above at page no. and there were some limitations encountered which is mentioned below:

5.2 LIMITATIONS

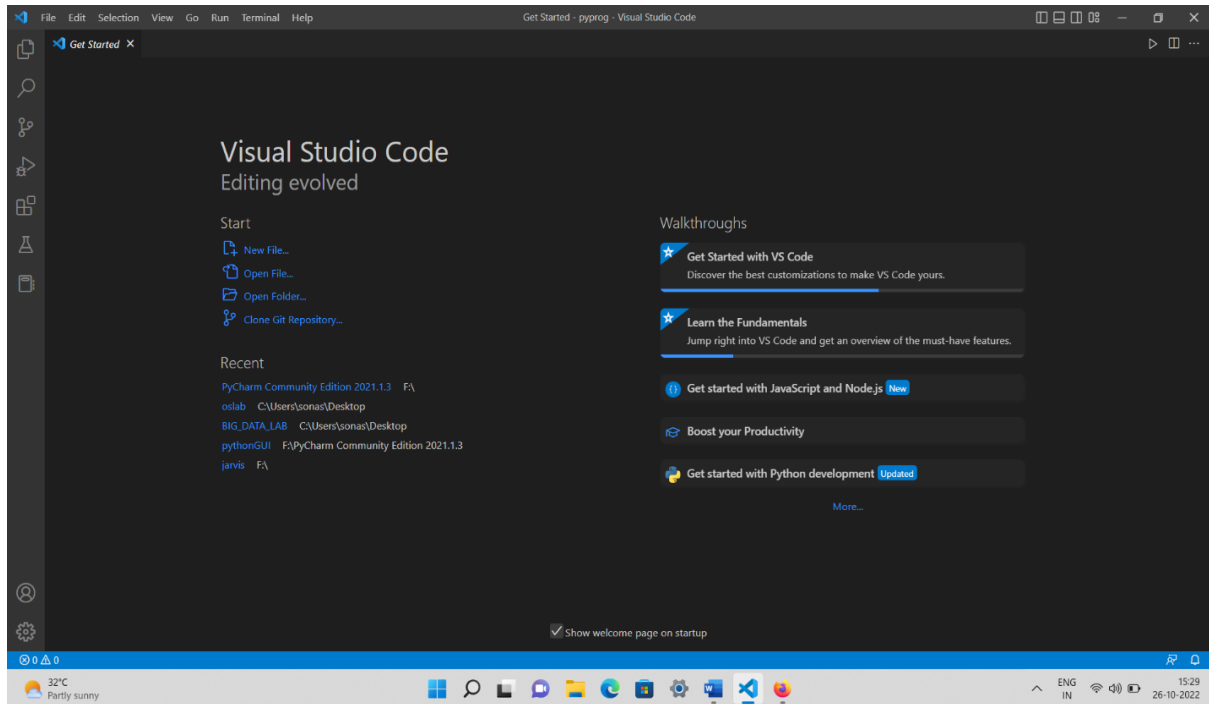
- Security is somewhere an issue, there is no voice command encryption in this project.
- Background voice can interfere
- Misinterpretation because of accents and may cause inaccurate results.

REFERNCES

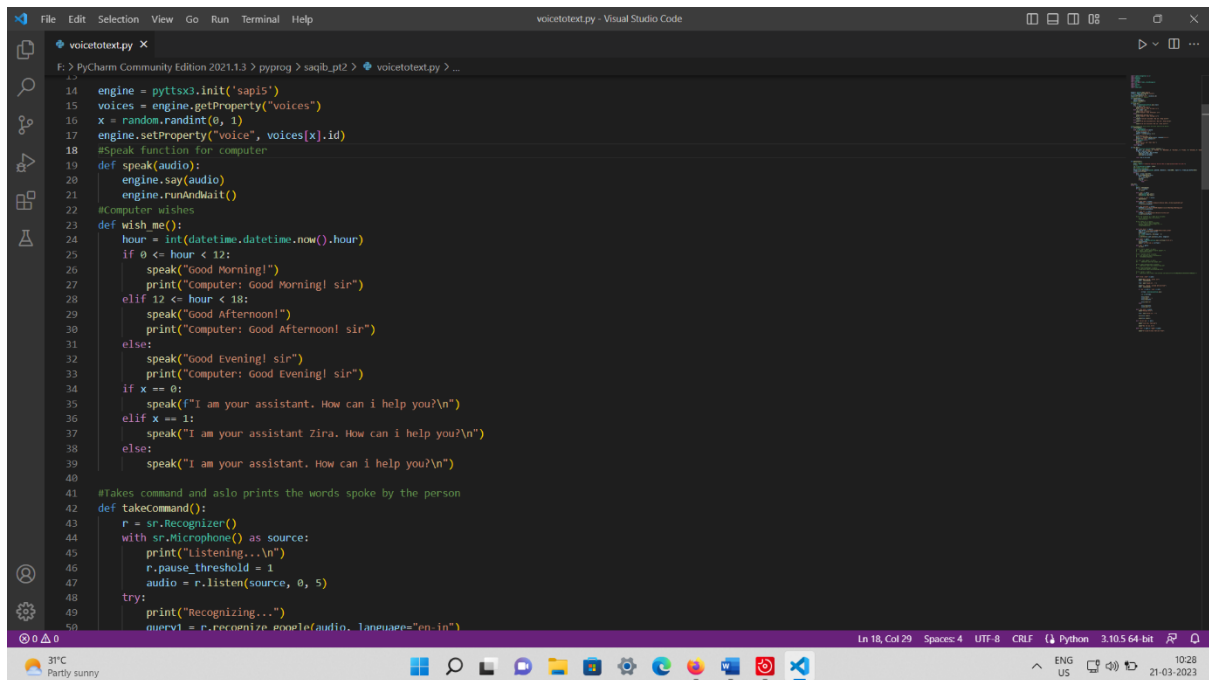
- [1] For Learning Python Language:
<https://wiki.python.org/moin/ReferenceBooks>
- [2] For Learning of Tkinter:
<https://docs.python.org/3/library/tk.html>
- [3] For VOSK API:
<https://alphacephei.com/vosk/>

WORKING ENVIRONMENT

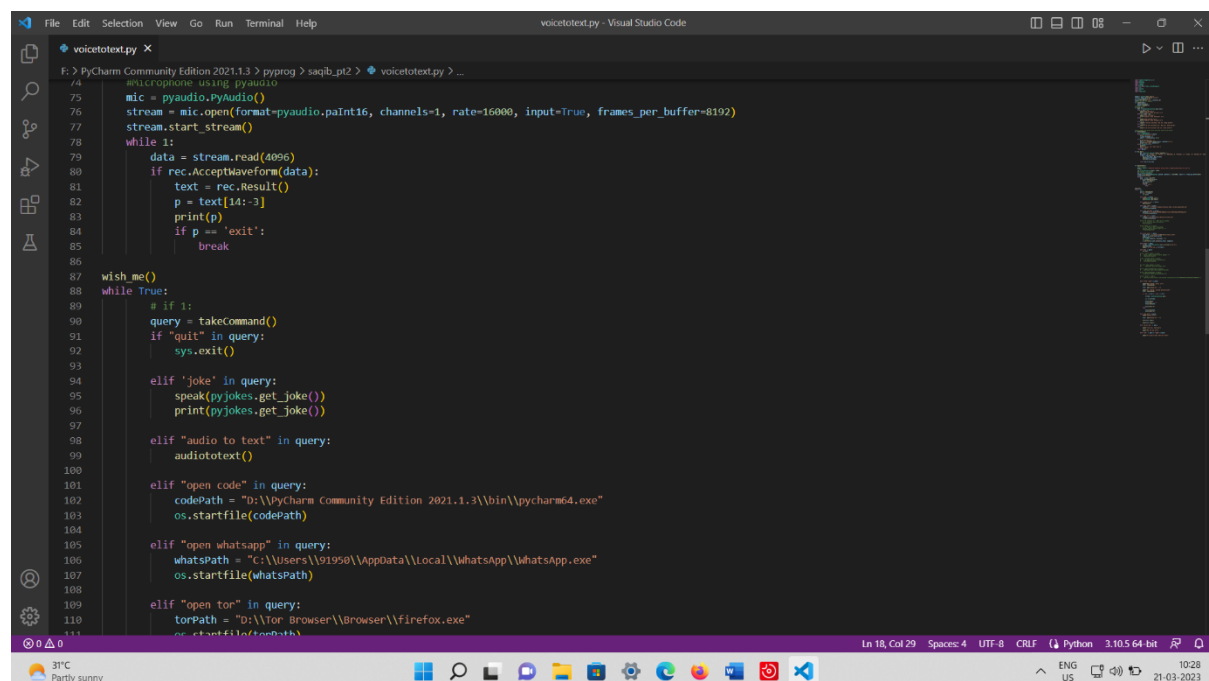
Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.



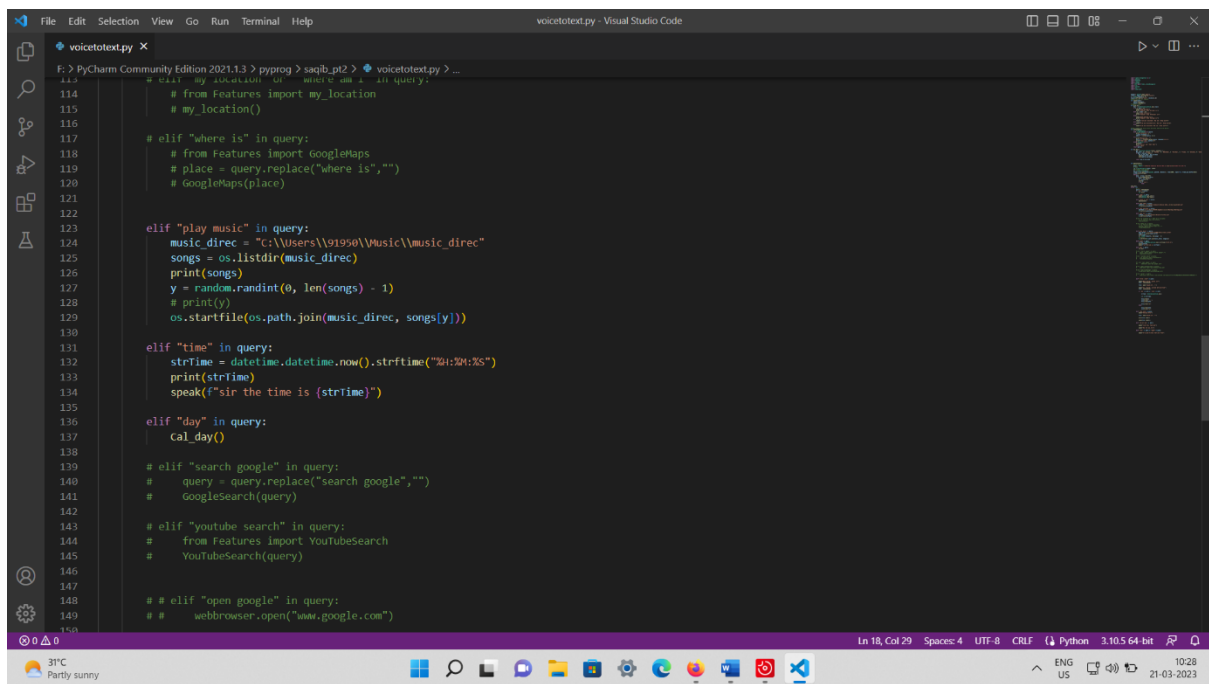
Chapter-6 Screenshots and Outputs



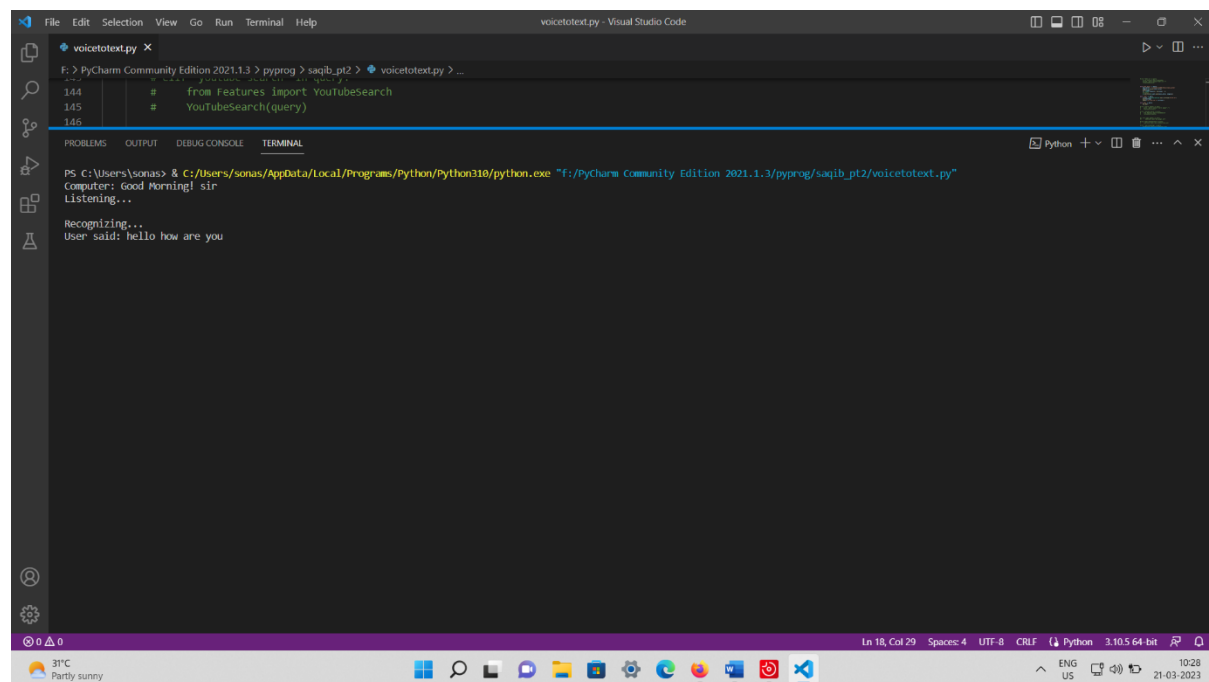
```
14 engine = pyttsx3.init('sapi5')
15 voices = engine.getProperty("voices")
16 x = random.randint(0, 1)
17 engine.setProperty("voice", voices[x].id)
18 #Speak function for computer
19 def speak(audio):
20     engine.say(audio)
21     engine.runAndWait()
22 #Computer wishes
23 def wish_me():
24     hour = int(datetime.datetime.now().hour)
25     if 0 <= hour < 12:
26         speak("Good Morning!")
27         print("Computer: Good Morning! sir")
28     elif 12 <= hour < 18:
29         speak("Good Afternoon!")
30         print("Computer: Good Afternoon! sir")
31     else:
32         speak("Good Evening! sir")
33         print("Computer: Good Evening! sir")
34     if x == 0:
35         speak("I am your assistant. How can i help you?\n")
36     elif x == 1:
37         speak("I am your assistant Zira. How can i help you?\n")
38     else:
39         speak("I am your assistant. How can i help you?\n")
40
41 #Takes command and aslo prints the words spoke by the person
42 def takeCommand():
43     r = sr.Recognizer()
44     with sr.Microphone() as source:
45         print("Listening...\n")
46         r.pause_threshold = 1
47         audio = r.listen(source, 0, 5)
48     try:
49         print("Recognizing...")
50         query = r.recognize_google(audio, language="en-in")
```



```
74 #Microphone using pyaudio
75 mic = pyaudio.PyAudio()
76 stream = mic.open(format=pyaudio.paInt16, channels=1, rate=16000, input=True, frames_per_buffer=8192)
77 stream.start_stream()
78 while 1:
79     data = stream.read(4096)
80     if rec.AcceptWaveform(data):
81         text = rec.Result()
82         p = text[14:-3]
83         print(p)
84         if p == 'exit':
85             break
86
87 wish_me()
88 while True:
89     a = 1
90     query = takeCommand()
91     if "quit" in query:
92         sys.exit()
93
94     elif "joke" in query:
95         speak(pyjokes.get_joke())
96         print(pyjokes.get_joke())
97
98     elif "audio to text" in query:
99         audiototext()
100
101     elif "open code" in query:
102         codePath = "D:\\PyCharm Community Edition 2021.1.3\\bin\\pycharm64.exe"
103         os.startfile(codePath)
104
105     elif "open whatsapp" in query:
106         whatsPath = "C:\\Users\\91950\\AppData\\Local\\WhatsApp\\WhatsApp.exe"
107         os.startfile(whatsPath)
108
109     elif "open tor" in query:
110         torPath = "D:\\Tor Browser\\Browser\\Firefox.exe"
111         os.startfile(torPath)
```



```
113 # call my_location() if my_location is in query:
114 # from Features import my_location
115 # my_location()
116
117 # elif "where is" in query:
118 # from Features import GoogleMaps
119 # place = query.replace("where is", "")
120 # GoogleMaps(place)
121
122
123 elif "play music" in query:
124     music_dir = "c:\\Users\\91950\\Music\\music_dir"
125     songs = os.listdir(music_dir)
126     print(songs)
127     y = random.randint(0, len(songs) - 1)
128     # print(y)
129     os.startfile(os.path.join(music_dir, songs[y]))
130
131
132 elif "time" in query:
133     strTime = datetime.datetime.now().strftime("%H:%M:%S")
134     print(strTime)
135     speak(f"Sir the time is {strTime}")
136
137 elif "day" in query:
138     cal_day()
139
140 # elif "search google" in query:
141 #     query = query.replace("search google", "")
142 #     GoogleSearch(query)
143
144 # elif "youtube search" in query:
145 #     from Features import YouTubeSearch
146 #     YouTubeSearch(query)
147
148 # # elif "open google" in query:
149 #     # webbrowser.open("www.google.com")
150
```



```
PS C:\Users\sonas> & C:\Users\sonas\AppData\Local\Programs\Python\Python310\python.exe "f:/PyCharm Community Edition 2021.1.3/pyprog/saqib_pt2/voicetotext.py"
Computer: Good Morning! sir
Listening...

Recognizing...
User said: hello how are you
```

```

F:\> PyCharm Community Edition 2021.1.3 > pyprog > saqib_pt2 > voicetotext.py > ...
144 # from Features import YouTubeSearch
145 # YouTubeSearch(query)
146

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\sonas> & C:\Users\sonas\AppData\Local\Programs\Python\Python310\python.exe "F:\PyCharm Community Edition 2021.1.3\pyprog\saqib_pt2\voicetotext.py"
Computer: Good Morning! sir
Listening...

Recognizing...
User said: hello how are you

Listening...

Recognizing...

Sorry I can't hear that.
Listening...

Recognizing...

Sorry I can't hear that.
Listening...

Recognizing...
User said: only English I love

Listening...

```

```

jarvisGUI.py - pyprog - Visual Studio Code

jarvisGUI.py > pdf_rer

343 elif "day" in query:
344     Cal_day()
345 elif "where is" in query:
346     from Features import GoogleMaps
347     place = query.replace("where is","")
348     GoogleMaps(place)
349 elif "how are you" in query:
350     speak("I am fine, Thank you")
351     speak("How are you, Sir")
352 elif "fine" in query or "good" in query:
353     speak("It's good to know that your fine")
354 elif "read the book" in query:
355     pdf_rer()
356 elif "joke" in query:
357     speak(pyjokes.get_joke())
358     print(pyjokes.get_joke())
359 elif "show cpu" in query:
360     condition()
361

LOG (VoskAPI:ReadDataFiles():model.cc:213) Decoding params beam=13 max-active=7000 lattice-beam=6
LOG (VoskAPI:ReadDataFiles():model.cc:216) Silence phones 1:2:3:4:5:6:7:8:9:10
LOG (VoskAPI:RemoveOrphanNodes():nnet-nnet.cc:948) Removed 1 orphan nodes.
LOG (VoskAPI:RemoveOrphanComponents():nnet-nnet.cc:847) Removing 2 orphan components.
LOG (VoskAPI:collapse():nnet-utils.cc:1488) Added 1 components, removed 2
LOG (VoskAPI:ReadDataFiles():model.cc:240) Loading i-vector extractor from F:\PyCharm Community Edition 2021.1.3\pyprog\vosk-model-en-in-0.5\ivector\final.ie
LOG (VoskAPI:computeDerivedVars():ivector-extractor.cc:183) Computing derived variables for iVector extractor
LOG (VoskAPI:computeDerivedVars():ivector-extractor.cc:284) Done.
LOG (VoskAPI:ReadDataFiles():model.cc:279) Loading HCLG from F:\PyCharm Community Edition 2021.1.3\pyprog\vosk-model-en-in-0.5\graph\HCLG.fst
LOG (VoskAPI:ReadDataFiles():model.cc:284) Loading words from F:\PyCharm Community Edition 2021.1.3\pyprog\vosk-model-en-in-0.5\graph\words.txt
LOG (VoskAPI:ReadDataFiles():model.cc:303) Loading wininfo F:\PyCharm Community Edition 2021.1.3\pyprog\vosk-model-en-in-0.5\graph\phones\word_boundary.int
LOG (VoskAPI:ReadDataFiles():model.cc:310) Loading subtract G.fst model from F:\PyCharm Community Edition 2021.1.3\pyprog\vosk-model-en-in-0.5\rescore\G.fst
LOG (VoskAPI:ReadDataFiles():model.cc:312) Loading CARPA model from F:\PyCharm Community Edition 2021.1.3\pyprog\vosk-model-en-in-0.5\rescore\G.carpa

```

SOURCE CODE

```
import speech_recognition as sr
import pyttsx3
import datetime
import random
import pyaudio
from vosk import Model, KaldiRecognizer
import sys
import pyjokes
import os
import webbrowser

engine = pyttsx3.init('sapi5')
voices = engine.getProperty("voices")
x = random.randint(0, 1)
engine.setProperty("voice", voices[x].id)
#Speak function for computer
def speak(audio):
    engine.say(audio)
    engine.runAndWait()
#Computer wishes
def wish_me():
    hour = int(datetime.datetime.now().hour)
    if 0 <= hour < 12:
        speak("Good Morning!")
        print("Computer: Good Morning! sir")
    elif 12 <= hour < 18:
        speak("Good Afternoon!")
        print("Computer: Good Afternoon! sir")
    else:
        speak("Good Evening! sir")
        print("Computer: Good Evening! sir")
    if x == 0:
        speak(f"I am your assistant. How can i help you?\n")
    elif x == 1:
        speak("I am your assistant Zira. How can i help you?\n")
    else:
        speak("I am your assistant. How can i help you?\n")
#Takes command and aslo prints the words spoke by the person
```

```

def takeCommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...\n")
        r.pause_threshold = 1
        audio = r.listen(source, 0, 5)
    try:
        print("Recognizing...")
        query1 = r.recognize_google(audio, language="en-in")
        print(f"User said: {query1}\n")
    except Exception as er:
        print(er)
        print("Sorry I can't hear that.")
        return "None"
    return query1

def Cal_day():
    day = datetime.datetime.today().weekday() + 1
    Day_dict = {1: 'Monday', 2: 'Tuesday', 3: 'Wednesday', 4: 'Thursday',
5: 'Friday', 6: 'Saturday', 7: 'Sunday'}
    if day in Day_dict.keys():
        day_of_the_week = Day_dict[day]
        speak(day_of_the_week)
        print(day_of_the_week)

    return day_of_the_week

def audiototext():
    #File location
    model = Model(r"F:\PyCharm Community Edition 2021.1.3\pyprog\vosk-model-
en-in-0.5")
    # Model
    rec = KaldiRecognizer(model, 16000)
    #Microphone using pyaudio
    mic = pyaudio.PyAudio()
    stream = mic.open(format=pyaudio.paInt16, channels=1, rate=16000,
input=True, frames_per_buffer=8192)
    stream.start_stream()
    while 1:
        data = stream.read(4096)
        if rec.AcceptWaveform(data):
            text = rec.Result()
            p = text[14:-3]
            print(p)
            if p == 'exit':
                break

wish_me()

```

```

while True:
    # if 1:
    query = takeCommand()
    if "quit" in query:
        sys.exit()

    elif 'joke' in query:
        speak(pyjokes.get_joke())
        print(pyjokes.get_joke())

    elif "audio to text" in query:
        audiototext()

    elif "open code" in query:
        codePath = "D:\\PyCharm Community Edition
2021.1.3\\bin\\pycharm64.exe"
        os.startfile(codePath)

    elif "open whatsapp" in query:
        whatsPath =
"C:\\Users\\91950\\AppData\\Local\\WhatsApp\\WhatsApp.exe"
        os.startfile(whatsPath)

    elif "open tor" in query:
        torPath = "D:\\Tor Browser\\Browser\\firefox.exe"
        os.startfile(torPath)

    # elif "my location" or " where am i" in query:
    #     from Features import my_location
    #     my_location()

    # elif "where is" in query:
    #     from Features import GoogleMaps
    #     place = query.replace("where is","")
    #     GoogleMaps(place)

    elif "play music" in query:
        music_direct = "C:\\Users\\91950\\Music\\music_direct"
        songs = os.listdir(music_direct)
        print(songs)
        y = random.randint(0, len(songs) - 1)
        # print(y)
        os.startfile(os.path.join(music_direct, songs[y]))

    elif "time" in query:
        strTime = datetime.datetime.now().strftime("%H:%M:%S")
        print(strTime)
        speak(f"sir the time is {strTime}")

```

```

elif "day" in query:
    Cal_day()

# elif "search google" in query:
#     query = query.replace("search google","")
#     GoogleSearch(query)

# elif "youtube search" in query:
#     from Features import YouTubeSearch
#     YouTubeSearch(query)

# # elif "open google" in query:
# #     webbrowser.open("www.google.com")

# elif "open stackoverflow" in query:
#     webbrowser.open("www.stackoverflow.com")

# elif "open duckduckgo" in query:
#     webbrowser.open("www.duckduckgo.com")

# elif "quran" in query:
#     webbrowser.open("https://www.youtube.com/playlist?list=PLuM8gIEB
e3i6iGPwZyFDu7ElEHwMyGm-v")

elif "write a note" in query:

    speak("What should i write, sir")
    note = takeCommand()

    file = open('Saqib.txt', 'a')

    speak("Sir, Should i include date and time")
    snfm = takeCommand()

    if 'yes' in snfm or 'sure' in snfm:

        strTime = datetime.datetime.now()

        a = str(strTime)

        file.write(a)
        file.write(" :- ")
        file.write(note)

        file.write("\n")

```

```
        else:

            file.write(note)
            file.write("\n")

    elif "show note" in query:
        speak("Showing Notes")

        file = open("Saqib.txt", "r")

        print(file.read())

        speak(file.read(6))

    elif 'how are you' in query:

        speak("I am fine, Thank you")

        speak("How are you, Sir")

    elif 'fine' in query or "good" in query:

        speak("It's good to know that your fine")
```