

PERSONAL ASSISTANT USING PYTHON
BACHELOR OF ENGINEERING
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted By

MOHD MAAZ WAHEED	1604-20-735-052
MOHD AFTAB AHMED	1604-20-735-055
MOHD GHOUSE TAZEEM	1604-20-735-058

Under the Guidance of

Mrs. SHUBHANGI SAXENA

Assistant Professor (ECED)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY

(Affiliated to OSMANIA UNIVERSITY, Hyderabad)

Banjara Hills, Hyderabad, 500034.

2022-2023

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 is an interesting task to make our 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 favorite 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, we realized that the concept of AI in every field is decreasing human effort and saving time.

Functionalities of this project include:

1. It can send emails.
2. It can read PDF
3. It can send text on WhatsApp
4. It can open command prompt, your favorite IDE, notepad etc.
5. It can play music
6. It can do Wikipedia searches for you
7. It can open websites like Google, YouTube, etc., in a web browser
8. It can give weather forecast
9. It can give desktop reminders of your choice
10. It can have some basic conversation

Now the basic question arises in mind that how it is an AI? The virtual assistant that we are going to create 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.

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 favorite 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, we 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 favorite 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 PyCharm IDE for making this project, and I created all py files in PyCharm. Along with this I used following modules and libraries in my project. pyttsx3, SpeechRecognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, PyQt 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.

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 favorite IDE with the help of a single voice command. It 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 PyCharm. All the python files were created in PyCharm 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, PyQt 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, it 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 send text on WhatsApp, It can open command prompt, your favorite 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.

CHAPTER 2

SYSTEM DESIGN

2.1. DATA FLOW

The data flow for the assistant is as follow:

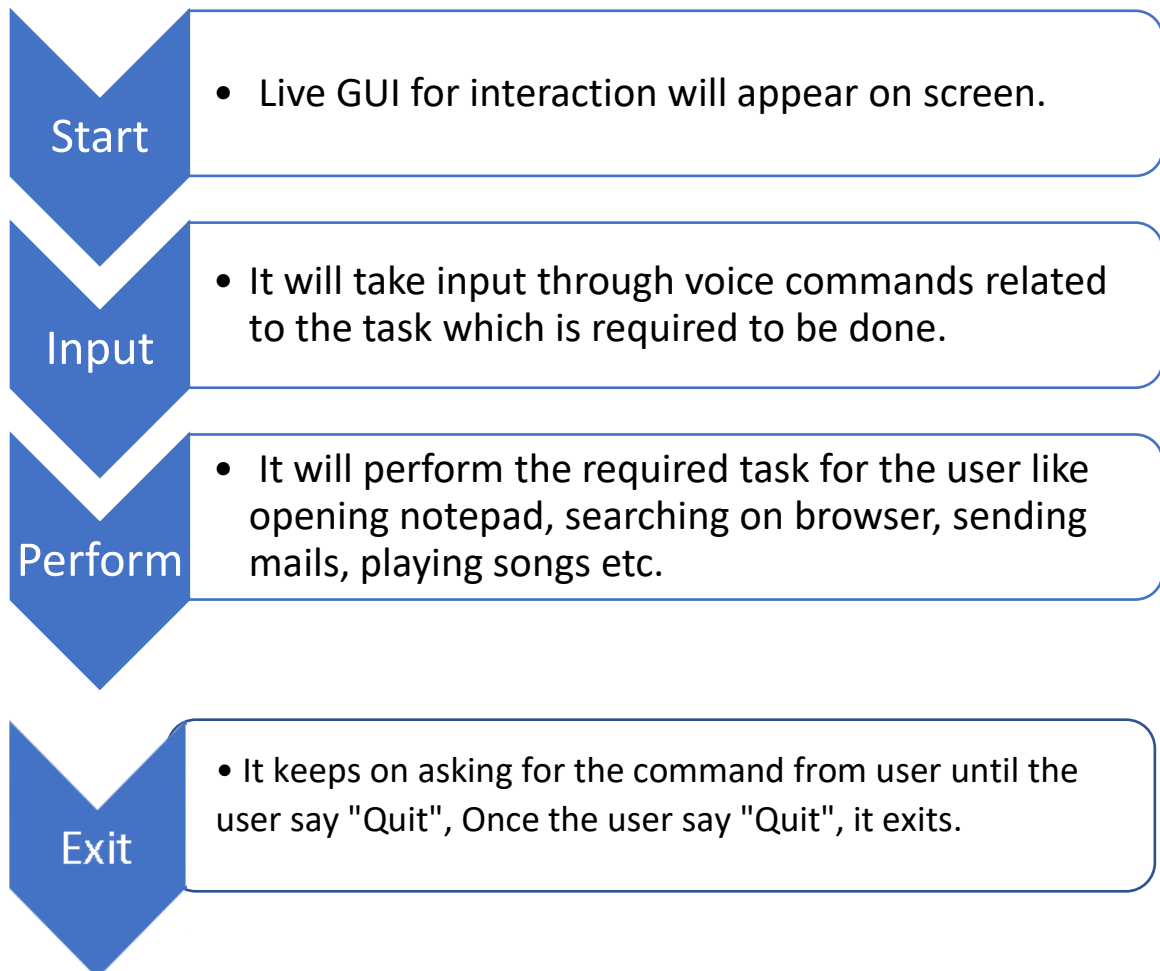


Figure 2.1 Data flow for Personal Assistant

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 assistant 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.

CHAPTER 3

SOFTWARE DETAILS

The IDE used in this project is PyCharm. All the python files were created in PyCharm 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, PyQt etc. I have created a live GUI for interacting with the assistant as it gives a design and interesting look while having the conversation.

3.1. VSCODE

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.

3.2. PYQT5 FOR LIVE GUI

PyQt5 is the most important python binding. It contains set of GUI widgets. PyQt5 has some important python modules like QtWidgets, QtCore, QtGui, and QtDesigner etc.

3.3. PYTHON LIBRARIES

In this assistant, following python libraries were used:

3.3.1. pyttsx3: It is a python library which converts text to speech.

3.3.2. SpeechRecognition: It is a python module which converts speech to text.

3.3.3. pywhatkit: It is python library to send WhatsApp message at a particular time with some additional features. **3.3.4. Datetime:** This library provides us the actual date and time.

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

3.3.6. Smtplib: Simple mail transfer protocol that allows us to send mails and to route mails between mail servers.

3.3.7. pyPDF2: It is a python module which can read, split, merge any PDF.

3.3.8. Pyjokes: It is a python libraries which contains lots of interesting jokes in it.

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

3.3.10. Pyautogui: It is a python libraries for graphical user interface.

3.3.11. os: It represents Operating System related functionality.

3.3.12. sys: It allows operating on the interpreter as it provides access to the variables and functions that usually interact strongly with the interpreter.

CHAPTER 4

IMPLEMENTATION WORK DETAILS

A desktop assistant is a voice assistant that can perform many daily tasks of desktop like playing music, opening your favorite IDE with the help of a single voice command. It 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.

4.1. REAL LIFE APPLICATION

4.1.1. Saves time: It is a desktop voice 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.

4.1.2. 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.

4.1.3. Reactive nature: The desktop assistant is reactive which means it knows human language very well and understands 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.

4.1.4. Multitasking: The main application of it can be its multitasking ability. It can ask for continuous instruction one after another until the user "QUIT" it.

4.1.5. No Trigger phase: It asks for the instruction and listens the response that is given by user without needing any trigger phase and then only executes the task.

4.2. DATA IMPLEMENTATION AND PROGRAM EXECUTION

As the first step, install all the necessary packages and libraries. The command used to install the libraries is "pip install" and then import it.

FUNCTIONS

takeCommand()

wishMe()

taskExecution()

CHAPTER 5

SOURCE CODE

CHAPTER 6

INPUT/OUTPUT SCREENSHOTS

CHAPTER 7

SYSTEM TESTING

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

7.1. 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 is able to execute the required task correctly then the system passes in that particular functionality test. For example to check whether the assistant can search on Google or not, as we can see in the figure 7.1, user said “Open Google”, then assistant asked, “What should I search on Google?” then user said, “What is Python”, assistant open Google and searched for the required input.

7.2. 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 it responses to each query that is being asked by the user.

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.

7.3. SECURITY

The security testing mainly focuses on vulnerabilities and risks. As this 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.

7.4. 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 8

CONCLUSION

This 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:

8.1. LIMITATIONS

- 8.1.1. Security is somewhere an issue, there is no voice command encryption in this project.
- 8.1.2. Background voice can interfere.
- 8.1.3. Misinterpretation because of accents and may cause inaccurate results.
- 8.1.4. This assistant cannot be called externally anytime like other traditional assistants like Google Assistant can be called just by saying, "Ok Google!"

8.2. SCOPE FOR FUTURE WORK

- 8.2.1. Make the assistant learn more on its own and develop a new skill in it(using Machine Learning).
- 8.2.2. Assistant android app can also be developed.
- 8.2.3. Make more assistant voice terminals.
- 8.2.4. Voice commands can be encrypted to maintain security.