

**UNIVERSITY INSTITUTE OF TECHNOLOGY  
BARKATULLAH UNIVERSITY, BHOPAL**

**Department of Computer Science & Engineering**

**MAJOR PROJECTON  
“AI-BASED VIRTUAL ASSISTANT”**

**Submitted for the Partial Fulfilment of the Requirement for the  
Award of Degree of  
Bachelor of Engineering (B.E.) in Computer Science & Engineering  
Year 2021**

**University Institute of Technology  
Barkatullah University, Bhopal**

**By**

**Abhishek Patel   Swadeep Kherde   Tina Mishra   Tejaswini Singh Bhanwar**

**Under the Guidance  
Of**

**Mr. Bhawani Singh Rathore  
Project Guide,  
CSE Department,  
UIT-BU, Bhopal**

**Dr. Divakar Singh  
Head of Department,  
CSE Department,  
UIT-BU, Bhopal**

**UNIVERSITY INSTITUTE OF TECHNOLOGY,  
BARKATULLAH UNIVERSITY, BHOPAL**

**Department of Computer Science & Engineering**



**CERTIFICATE**

**YEAR 2021**

This is to certify that, **Mr Abhishek Patel, Mr Swadeep Kherde, Ms Tejaswini Singh Bhanwar, Ms Tina Mishra** have completed this project work titled “**AI-Based Virtual Assistant**” in partial fulfilment of the award of the degree of Bachelor of Engineering in Computer Science & Engineering in the year 2021 from Barkatullah University Institute of Technology, Barkatullah University, Bhopal.

**Mr Bhawani Singh  
Rathore**

Project Guide,  
CSE, UIT-BU, Bhopal

**Dr Divakar Singh**

Head of Department  
CSE, UIT-BU, Bhopal

**Prof. Neeraj Gaur**

Director  
UIT-BU, Bhopal

**UNIVERSITY INSTITUTE OF TECHNOLOGY  
BARKATULLAH UNIVERSITY, BHOPAL**

**Department of Computer Science & Engineering**

**DECLARATION  
YEAR 2021**

I hereby declare that the Major Project report entitled “**AI-Based Virtual Assistant**” submitted in the partial fulfilment of the award of Bachelor of Engineering Degree in Computer Science & Engineering from University Institute of Technology, BU, Bhopal during the year 2021 is an authentic record of our project work. This is our original work and has not been submitted earlier for the award of any other degree, diploma or any other certificate.

**Date:**

**Abhishek Patel**

**Swadeep Kherde**

**Tina Mishra**

**Tejaswini Singh Bhanwar**

## **ACKNOWLEDGEMENT**

Success is the manifestation of diligence, perseverance inspired motivation and innovation. The successful completion of our project is solely attributed to constant help guidance and encouragement received by the people around us. Before we get into the thick of things, we would like to add a few heartfelt words for the teachers who are part of this project in numerous ways. Teachers who gave us support right from the project idea were convinced.

First of all, we would like to thanks **Mr Bhawani Singh Rathore (Project Guide)** for allowing us to undergo this project idea. We would also like to pay our sincere regards to our **Dr Divakar Singh (H.O.D., CSE)** for providing an effective platform and support in the development of this project idea and finally, we would like to render our thanks to **Hon. Prof. Neeraj Gaur (Director)** UIT-BU for their guidance in this MAJOR REPORT project titled **“AI-Based Virtual Assistant”**. Last but not least we would like to thank our parents and friends for their support and cooperation. Regardless of the source, we wish to express our gratitude to those who may have contributed to this work, even though anonymously.

**Submitted by: -**

**Abhishek Patel**

**Swadeep Kherde**

**Tina Mishra**

**Tejaswini Singh Bhanwar**

## **ABSTRACT**

AI -Based Virtual assistants are programs on digital devices that listen and respond to verbal commands. A user can say, “What’s the weather?” and the voice assistant will answer with the weather report for that day and location. They could say, “Tell me a story,” and the assistant will jump into a tale. The user could even say, “Order my favourite pizza,” and dinner will be on its way! Voice assistants are so easy to use that many people forget to stop and WONDER how theywork. How do voice assistants understand us? Is it magic? A complex system of codes? An actual person listening on the other end? The answer is less complicated than you might think. The application works like Siri, Google Assistant etc. The U.I of the application is self- explainable and very minimum. It takes voice as input. The system is being designed in such a way that all the services provided by the mobile devices are accessible by the end-user on the user's voice commands.

### **KEYWORDS:**

1. Voice Assistant,
2. Python’s Speech Recognition,
3. Python text-to-speech library
4. Pyttsx3,
5. Python3.8

## **CONTENTS**

<b>Topics</b>	<b>Page no.</b>
<b>LIST OF FIGURES</b>	<b>8-9</b>
<b>LIST OF ABBREVIATIONS</b>	<b>10</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>11-13</b>
1.1: Applications of AI	11
1.2: Benefits of Using Assistants using artificial intelligence a modern approach	12
<b>CHAPTER 2 REQUIREMENT</b>	<b>14-15</b>
2.1: Hardware and Software REquirements	14
2.2: Package Required	14
2.3: Implementation	15
2.3.1: Import the following libraries	15
<b>CHAPTER 3 TECHNICAL DESCRIPTION</b>	<b>16-20</b>
3.1: Python (Programming language)	16
3.2: Statements and control flow	16
3.3: Voice Assistant using python	18
3.3.1: Modules needed	18
3.4: VS Code IDE for python compiler	18
3.4.1: Language support	19
3.4.2: Data collection	20
3.4.3: Version Control	20
<b>CHAPTER 4 PROJECT DESCRIPTION</b>	<b>21-30</b>
4.1: Creating Voice Assistant using python (Steps)	21
4.1.1: Defining Speak Function	21
4.1.2: Creating Our main() function:	21
4.1.3: Defining Greet Function:	22
4.1.4: Defining getcommand Function:	22
4.1.5: Defining Jokes Function:	23
4.1.6: Defining Date Function:	23

4.1.7: Defining Time Function:	24
4.1.8: Defining mail function:	24
4.1.9: Coding logic of AI	24
4.1.10: PyAudio	29
<b>CHAPTER 5 CODING</b>	<b>31-34</b>
5.1: Code for Voice Assistant using python	31
<b>CHAPTER 6 CONCLUSION AND FUTURE WORK</b>	<b>35-40</b>
6.1: Limitation	35
6.2: Benefits	36
6.3: Future Scopes	36
6.3.1: AI in Science and Research	36
6.3.2: AI in Cyber Security	36
6.3.3: AI in Data Analysis	37
6.3.4: AI in Transport	37
6.3.5: AI in Home	38
6.3.6: AI in Healthcare	38
<b>CHAPTER 7 REFERENCES</b>	<b>40</b>

## LIST OF FIGURES

<b>S. No.</b>	<b>Figure No.</b>	<b>Figure Name</b>	<b>Page no.</b>
1.	Fig.1.1	Image	12
2.	Fig.3.1	Visual Studio Code Insiders logo	19
3.	Fig.4.1	Defining Speak Function	21
4.	Fig.4.2	Installation of Pyttsx3	21
5.	Fig.4.3	Writing our speak() Function:	21
6.	Fig.4.4	Creating Our main() function:	21
7.	Fig.4.5	Import datetime	22
8.	Fig.4.6	Defining greet() function	22
9.	Fig.4.7	Installation of SpeechRecognition	22
10.	Fig.4.8	Import speech recognition as sr	22
11.	Fig.4.9	Defining getcommand() function	23
12.	Fig.4.10	add a try and except	23
13.	Fig.4.11	Defining Jokes() Function	23
14.	Fig.4.12	Defining Date() Function	24
15.	Fig.4.13	Defining Time() Function	24
16.	Fig.4.14	Defining mail() function	24
17.	Fig.4.15	Install wikipedia	25
18.	Fig.4.16	Import wikipedia	25
19.	Fig.4.17	To open YouTube site in a web-browser	25
20.	Fig.4.18	To open Google site in a web-browser	26
21.	Fig.4.19	To Play music	26
22.	Fig.4.20	To know the current time	26
23.	Fig.4.21	To open the VS Code Program	27
24.	Fig.4.22	To mail	27



25.	Fig.4.23	Exception	28
26.	Fig.4.24	To Remember	28
27.	Fig.4.25	Tell Us a Joke	28
28.	Fig.4.26	To open facebook	28
29.	Fig.4.27	To open notepad	29
30.	Fig.4.28	To shut down or restart computer	29
31.	Fig.4.29	Download PyAudio from Python unofficial libraries	29
32.	Fig.4.30	Select and download according to your system need	30
33.	Fig.4.31	Save it in same folder where main file is saved	30

## LIST OF ABBREVIATIONS

S. No.	Symbol	Description
1.	AI	Artificial Intelligence
2.	IDE	Integrated Development Environment
3.	NLP	Natural Language Processing
4.	API	Application Programming Interface
5.	MIT	Massachusetts Institute of Technology
6.	HTML	Hyper Text Markup Language
7.	RAM	Random Access Memory
8.	VS	Visual Studio
9.	CSS	Cascading Style Sheets
10.	JSON	Java Script Object Notation
11.	SMTP	Simple Mail Transfer Protocol
12.	TIOBE	The Importance Of Being Earnest

# CHAPTER 1

## INTRODUCTION

Artificial Intelligence is an approach to make a computer, a robot, or a product think how smart humans think. AI is a study of how the human brain thinks, learn, decide and work when it tries to solve problems. And finally, this study outputs intelligent software systems. AI aims to improve computer functions that are related to human knowledge, for example, reasoning, learning, and problem-solving.

Intelligence is intangible. It is composed of

- Reasoning
- Learning
- Problem Solving
- Perception
- Linguistic Intelligence

The objectives of AI research are reasoning, knowledge representation, planning, learning, natural language processing, realization, and the ability to move and manipulate objects. There are long-term goals in the general intelligence sector.

Approaches include statistical methods, computational intelligence, and traditional coding AI. During the AI research related to search and mathematical optimization, artificial neural networks and methods based on statistics, probability, and economics, we use many tools. Computer science attracts AI in the field of science, mathematics, psychology, linguistics, philosophy and so on.

### 1.1: Applications of AI

- **Gaming** – AI plays an important role for a machine to think of a large number of possible positions based on deep knowledge in strategic games. for example, chess, river crossing, N-queens problems, etc.
- **Natural Language Processing** – Interact with the computer that understands natural language spoken by humans.
- **Expert Systems** – Machine or software provide explanation and advice to the users.
- **Vision Systems** – Systems understand, explain, and describe visual input on the computer.
- **Intelligent Robots** – Robots can perform the instructions given by a human.
- **Speech Recognition** – Some AI-based speech recognition systems can hear and express sentences and understand their meanings while a person talks to them. For example, Siri and Google assistant.

- **Handwriting Recognition** – The handwriting recognition software reads the text written on paper and recognize the shapes of the letters and convert it into editable text.

Have you ever wondered how cool it would be to have your own A.I. assistant? Imagine how easier it would be to send emails without typing a single word, doing Wikipedia searches without opening web browsers, and performing many other daily tasks like playing music with the help of a single voice command. It's named Desktop Voice Assistant JARVIS with Voice Recognition Intelligence, which takes the user input in form of the user's voice and processes it and returns the output in various ways like the search result speaks out to the end-user.

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

- It can send emails for you.
- 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.

## 1.2: Benefits of Using Assistants using artificial intelligence a modern approach

- **Less human efforts**

To own an assistant who is completely virtual and has no days off or excuses is a great deal, right? One of the most beneficial points about having an AI assistant is that there is no need to spend money on its salary or labour charges is completely free. Thus, it serves as a great help in various applications of artificial intelligence.



Fig.1.1

- **More productivity**

These AI assistants or bots have human intelligence and can work twice the speed of a physical human being hence giving more and efficient output. The Artificial Intelligence development company invests in making bots as assistants for large scale production.

- **Flexible work**

From making calls to arranging meetings everything you need can be done with the help of a smart assistant. It's adaptable to human commands hence flexible to your orders. Voice modulation works as a human voice with the conversational AI platform added in a robot to behave like a virtual but real assistant.

- **Accuracy is everything**

The human brain has more power than computers but the human body errs and makes mistakes. The smart ai Assistant will be accurate every time. The big amounts of calculations can be done way faster and just at your voice command.

- **Faster approach**

All you need is a good internet connection and your smart Assistant will do everything for you at an instance. It has been proved by the best artificial intelligence companies that an artificial intelligence personal assistant responds to the user within a second.

- **Optimize workflow**

A smart assistant is not only used for personal use or a single device. It can be used in industries and business companies to optimize work. The smart Assistant can take care of the tasks which have less priority and let the workers handle big deals and issues. Companies using artificial intelligence have discerned a major difference in their work environment as having a smart assistant makes it easy.

- **Mobility of the device**

One of the most beneficial outcomes of having a smart assistant is that it can be carried on your mobile phone whenever you want wherever you wish.

## CHAPTER 2

### REQUIREMENT

#### 2.1: HARDWARE AND SOFTWARE REQUIREMENTS

The software is designed to be light-weighted so that it doesn't be a burden on the machine running it. This system is being built keeping in mind the generally available hardware and software compatibility. Here are the minimum hardware and software requirements for a virtual assistant.

##### **Hardware:**

- Pentium-pro processor or later.
- RAM 512MB or more.

##### **Software:**

- Windows 10(32-bit) or above.
- Python 2.7 or later
- Chrome Driver
- Selenium Web Automation
- SQLite

#### 2.2: PACKAGE REQUIRED:

To build a personal voice assistant it's necessary to install the following packages in your system using the pip command.

- 1) **Speech recognition** — Speech recognition is an important feature used in house automation and artificial intelligence devices. The main function of this library is it tries to understand whatever the humans speak and converts the speech to text.
- 2) **pyttsx3** — pyttsx3 is a text to speech conversion library in python. This package supports text to speech engines on Mac os x, Windows and Linux.
- 3) **Wikipedia** — Wikipedia is a multilingual online encyclopedia used by many people from the academic community ranging from freshmen to students to professors who wants to gain information over a particular topic. This package in python extracts data required from Wikipedia.
- 4) **datetime** — This is an inbuilt module in python and it works on date and time.
- 6) **time** — The time module helps us to display time
- 7) **OS** — This module is a standard library in python and it provides the function to interact with an

operating system.

- 8) Web browser** — This is an in-built package in python. It extracts data from the web
- 9) Subprocess** — This is a standard library use to process various system commands like to log off or to restart your PC.
- 11) request** - The request module is used to send all types of the HTTP request. Its accepts URL as parameters and gives access to the given URL'S.
- 12) Wolfram alpha** — Wolfram Alpha is an API that can compute expert-level answers using Wolfram's algorithms, knowledge base and AI technology. It is made possible by the Wolfram Language.

## **2.3: Implementation:**

### **2.3.1: Import the following libraries –**

```
import Speech recognition as sr
```

```
import pyttsx3
```

```
import datetime
```

```
import Wikipedia
```

```
import Web browser as wb
```

```
import OS
```

```
import datetime
```

```
import sys
```

```
import random
```

```
import smtplib
```

## CHAPTER 3

### TECHNICAL DESCRIPTION

#### 3.1: Python (Programming language)

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was created in the late 1980s, and first released in 1991, by Guido van Rossum as a successor to the ABC programming language. Python 2.0, released in 2000, introduced new features, such as list comprehensions, and a garbage collection system with reference counting, and was discontinued version 2.7 in 2020. Python 3.0, released in 2008, was a major revision of the language that is not completely backwards-compatible and much Python 2 code does not run unmodified on Python 3. With Python 2's end-of-life, only Python 3.6.x and later are supported, with older versions still supporting e.g., Windows 7 (and old installers not restricted to 64-bit Windows).

Python interpreters are supported for mainstream operating systems and available for a few more (and in the past supported many more). A global community of programmers develops and maintains Python, a free and open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and Python development.

As of December 2020, Python ranked third in TIOBE's index of most popular programming languages, behind C and Java.

#### 3.2: Statements and control flow

Python's statements include (among others):

- The assignment statement (token '=', the equals sign).
- The if statement, which conditionally executes a block of code, along with else and elif (a contraction of else-if).
- The for statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block.
- The while statement, which executes a block of code as long as its condition is true.



- The `try` statement, which allows exceptions raised in its attached code block to be caught and handled by `except` clauses; it also ensures that clean-up code in a `finally` block will always be run regardless of how the block exits.
- The `raise` statement, used to raise a specified exception or re-raise a caught exception.
- The `class` statement, which executes a block of code and attaches its local namespace to a class, for use in object-oriented programming.
- The `def` statement, which defines a function or method.
- The `break` statement, exits from the loop.
- The `continue` statement, skips this iteration and continues with the next item.
- The `return` statement, used to return a value from a function.
- The `import` statement, which is used to import modules whose functions or variables can be used in the current program. There are three ways of using `import`: `import <module name> [as <alias>]` or `from <module name> import *` or `from <module name> import <definition 1> [as <alias 1>], <definition 2> [as <alias 2>], ...`.
- The `print` statement was changed to the `print()` function in Python 3.

The assignment statement ('=') operates by binding a name as a reference to a separate, dynamically allocated object. Since the name's storage location doesn't contain the indicated value, it is improper to call it a variable. Names may be subsequently rebound at any time to any object. Since a name is a generic reference holder it is unreasonable to associate a fixed data type with it. However, at a given time, a name will be bound to *some* object, which will have a type. This association is referred to as dynamic typing.

Python does not support tail call optimization or first-class continuations, and, according to Guido van Rossum, it never will. However, better support for coroutine-like functionality is provided in 2.5, by extending Python's generators. Before 2.5, generators were lazy iterators; information was passed unidirectionally out of the generator. From Python 2.5, it is possible to pass information back into a generator function, and from Python 3.3, the information can be passed through multiple stack levels.

### 3.3: Voice Assistant using python

As we know Python is a suitable language for scriptwriters and developers. Let's write a script for Voice Assistant using Python. The query for the assistant can be manipulated as per the user's need.

Speech recognition is the process of converting audio into text. This is commonly used in voice assistants like Alexa, Siri, etc. Python provides an API called **Speech Recognition** to allow us to convert audio into text for further processing. In this article, we will look at converting large or long audio files into text using the Speech Recognition API in python.

#### 3.3.1: Modules needed

- ❖ Pytsx3
- ❖ PyAudio
- ❖ Speech Recognition
- ❖ Wikipedia
- ❖ OS

### 3.4: VS Code IDE for python compiler

Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Microsoft has released Visual Studio Code's source code on the VSCode repository of GitHub, under the permissive MIT License, while the compiled releases are freeware.

In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 50.7% of 87,317 respondents reporting that they use it.



Fig. 3.1

Visual Studio Code logo

Visual Studio Code is a source code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js and C++. It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blinklayout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language. It supports several programming languages and a set of features that

differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette.

Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, and debuggers, perform static code analysis and add code lines using the Language Server Protocol.

Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software.

Visual Studio Code allows users to set the code page in which the active document is saved, the newline character, and the programming language of the active document. This allows it to be used on any platform, in any locale, and for any given programming language.

### **3.4.1: Language support**

Out-of-the-box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript,

TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

#### **3.4.2: Data collection**

Visual Studio Code collects usage data and sends it to Microsoft, although this can be disabled. In addition, because of the open-source nature of the application, the telemetry code is accessible to the public, who can see exactly what is collected. According to Microsoft, the data is shared with Microsoft-controlled affiliates and subsidiaries, although law enforcement may request it as part of a legal process.

#### **3.4.3: Version Control**

Source Control is a built-in feature of Visual Studio Code. It has a dedicated tab inside of the menu bar where you can access version control settings and view changes made to the current project. To use the feature, you must link Visual Studio Code to any supported version control system (Git, SVN, perforce etc.). This allows you to create repositories as well as make a push and pull requests directly from the Visual Studio Code program

## CHAPTER 4

### PROJECT DESCRIPTION

#### 4.1: Creating Voice Assistant using python (Steps)

##### 4.1.1: Defining Speak Function

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

```
def speak(audio):  
    pass    #For now, we will write the conditions later.
```

Fig.4.1

Now, the next thing we need is audio. We must supply audio so that we can pronounce it using the `speak()` function we made. We are going to install a module called `pyttsx3`.

##### What is pyttsx3?

- A python library which 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 the Python 3.

##### Installation:

```
pip install pyttsx3
```

Fig. 4.2

##### Writing our `speak()` Function:

We made a function called `speak()` at the starting of this tutorial. Now, we will write our `speak()` function so that it can convert our text to speech.

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

Fig 4.3

##### 4.1.2: Creating Our `main()` function:

Now, we will create a `main()` function, and inside this `main()` Function, we will call our `speak` function.

```
if __name__ == "__main__":  
    speak("good morning")
```

Fig.4.4

#### 4.1.3: Defining Greet Function:

Now, we are going to make a **Greet()** function, that will make our AI ASSISTANT wish or greet the user according to the time of computer or pc. To provide current or live time to A.I., we need to import a module called datetime. Import this module to your program, by:

```
import datetime
```

Fig.4.5

Now, let's start defining the **greet()** function:

```
def greet():  
    hrs = int(datetime.datetime.now().hour)  
    if hrs < 12:  
        speak("good morning ")  
    elif 12 < hrs < 18:  
        speak("good afternoon")  
    elif hrs > 18:  
        speak("good evening")
```

Fig.4.6

Here, we have stored the integer value of the current hour or time into a variable named hour. Now, we will use this hour value inside an if-else loop.

#### 4.1.4: Defining getcommand Function:

The next most important thing for our A.I. assistant is that it should be able to take command with the help of the microphone of the user's system. So, now we will make a **getcommand()** function. With the help of the takeCommand() function, our A.I. assistant will be able to return a string output by taking microphone input from the user.

Before defining the takeCommand() function, we need to install a module called **Speech Recognition**. Install this module by:

```
pip install speechRecognition
```

Fig.4.7

After successfully installing this module, import this module into the program by writing an import statement.

```
import speechRecognition as sr
```

Fig.4.8

Let's start coding the getcommand() function.

```
def getcommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("listening...")
        r.pause_threshold = 1
        audio = r.listen(source)
```

Fig.4.9

We have successfully created our getcommand() function. Now we are going to add a try and except block to our program to handle errors effectively.

```
try:
    print("Recognizing...")
    query = r.recognize_google(audio, language='en-us')

    print(f"User said: {query}\n")

except:

    print("Say that again please...")
    return "None"
return query
```

Fig.4.10

#### 4.1.5: Defining Jokes Function:

Here our AI ASSISTANT will tell us a joke. If the user wants to Entertain. It gives us an exciting joke you will laugh definitely.

```
def jokes():
    joke = pyjokes.get_joke()
    print(joke)
    speak(joke)
```

Fig.4.11

We have successfully created our jokes function. Now we are going to add a try and except block to our program to handle errors effectively.

#### 4.1.6: Defining Date Function:

Here our AI assistant will be able to give information about the date in the format of month year and day.

```
def date_():
    year = int(datetime.datetime.now().year)
    month = int(datetime.datetime.now().month)
    day = int(datetime.datetime.now().day)
    speak("the current date is")
    speak(day)
    speak(month)
    speak(year)
```

Fig.4.12

#### 4.1.7: Defining Time Function:

```
def time():
    speak("current time is")
    speak(datetime.datetime.now().strftime("%H:%M:%S"))
```

Fig.4.13

In the above, code with are using datetime() function and storing the current or live of the system into a variable called strTime. After storing the time in strTime, we are passing this variable as an argument in speak function. Now, the time string will be converted into the speech.

#### 4.1.8: Defining mail function :

Now, we will create a **mail()** function, which will help us to send emails to one or more recipients.

```
def mail(to, content):
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('useremail@gmail.com', 'your-password')
    server.mail('useremail@gmail.com', to, content)
    server.close()
```

Fig.4.14

In the above code, we are using the SMTP module, which we have already discussed above.

**Note:** Do not forget to 'enable the less secure apps' feature in your Gmail account. Otherwise, the mail function will not work properly.

#### 4.1.9: Coding logic of AI

Now, we will develop logic for different commands such as Wikipedia searches, playing music, etc.



### Defining Task 1: To search something on Wikipedia

To do Wikipedia searches, we need to install and import the Wikipedia module into our program. Type the below command to install the Wikipedia module:

```
pip install wikipedia
```

Fig.4.15

After successfully installing the Wikipedia module, import it into the program by writing an import statement.

```
if __name__ == "__main__":
    print("Hi!")
    greet()
    speak("Hello User this is AI Assistant. how may I help u")

    # date_()
    while True:
        query = getcommand().lower()
        if "open notepad" in query:
            path = "C:\\Windows\\System32\\notepad.exe"
            os.startfile(path)
        elif "open facebook" in query:
            wb.open("www.facebook.com")
        elif 'wikipedia' in query:
            speak("searching wikipedia...")
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=1)
            speak("according to wikipedia")
            print(results)
            speak(results)
```

Fig 4.16

In the above code, we have used an if statement to check whether Wikipedia is in the search query of the user or not. If Wikipedia is found in the user's search query, then two sentences from the summary of the Wikipedia page will be converted to speech with the help of speak function.

### Defining Task 2: To open YouTube site in a web-browser

To open any website, we need to import a module called web-browser. It is an in-built module, and we do not need to install it with pip statement, we can directly import it into our program by writing an import statement.

```
elif "open youtube" in query:
    wb.open("www.youtube.com")
```

Fig.4.17

Here, we are using the elif loop to check whether the YouTube is in the query of the user or not. Let's suppose, the user gives command as "AI ASSISTANT open youtube." So, open YouTube will be in the user's query, and the elif condition will be true.

### Defining Task 3: To open Google site in a web-browser

```
elif "google" in query:
    speak("what you want me to google")
    cm = getcommand().lower()
    wb.open(f"{cm}")
```

Fig.4.18

We are opening Google in a web browser by applying the same logic that we used to open YouTube.

### Defining Task 4: To play music

To play music, we need to import a module called OS. Import this module directly with an import statement.

```
elif "play music" in query:
    location = 'D:\\mobile backup\\Gym song'
    songs = os.listdir(location)
    print(songs, end="\n")
    x= random.randint(0, (len(songs)-1))
    print(x)
    os.startfile(os.path.join(location, songs[x]))
```

Fig.4.19

In the above code, we first opened our music directory and then listed all the songs present in the directory with the help of the OS module. With the help of os.startfile, you can play any song of your choice. I am playing the first song in the directory. However, you can also play a random song with the help of a random module. Every time you command to play music, AI assistant will play any random song from the song directory.

### Defining Task 5: To know the current time

```
elif "time" in query:
    time()
```

Fig.4.20

In the above, code with are using datetime() function and storing the current or live of the system into a variable called strTime. After storing the time in strTime, we are passing this variable as an argument in speak function. Now, the time string will be converted into the speech.

### Defining Task 6: To open the VS Code Program

```
elif "visual code" in query:
    path =
"C:\\Users\\Teja\\AppData\\Local\\Programs\\Microsoft VS
Code\\Code.exe"
    os.startfile(path)
```

Fig .4.21

To open the VS Code or any other application, we need the code path of the application.

#### Steps to get the code path of the application:

**Step 1:** Open the file location.

**Step 2:** Right-click on the application and click on properties.

**Step 3:** Copy the target from the target section.

After copying the target of the application, save the target into a variable. Here, I am saving the target into a variable called codePath, and then we are using the OS module to open the application.

### Defining Task 7: To mail

To send an email, we need to import a module called smtplib.

#### What is smtplib?

- Simple Mail Transfer Protocol (SMTP) is a protocol that allows us to send emails and route emails between mail servers. An instance method called **mail** is present in the SMTP module. This instance method allows us to send an email. It takes 3 parameters:
- **The sender:** Email address of the sender.
- **The receiver:** Email of the receiver.
- **The message:** A string message which needs to be sent to one or more than one recipient.

Now, we will create a **mail()** function, which will help us to send emails to one or more recipients.

```
elif 'email to user' in query:
    try:
        speak("What should I say?")
        content = getcommand()
        to = "useryourEmail@gmail.com"
        sendEmail(to, content)
        speak("Email has been sent!")
```

Fig.4.22

```
except Exception as e:
    print(e)
    speak("Oops!! I am getting trouble in sending email")
```

Fig.4.23

We are using the try and except block to handle any possible error that can occur while sending mail.

### Defining Task 8: To Remember

This is used to remember things. our AI ASSISTANT can remember anything if you said that remember.

```
elif "remember that" in query:
    speak("what should I remember")
    notes = getcommand()
    speak(f"you said that {notes}")
    remember = open("text.txt", "w")
    remember.write(notes)
    remember.close()

elif "remember anything" in query:
    remember = open("text.txt", "r")
    speak(f"you once said that remember AI Assistant")
    {remember.read() }")
    print(remember.read())
    remember.close()
```

Fig 4.24

### Defining Task 9: Tell Us a Joke

This task is used for entertainment purpose.

```
def jokes():
    joke = pyjokes.get_joke()
    print(joke)
    speak(joke)
```

Fig 4.25

### Defining Task 10: To open Facebook

This task is used for the opening of Facebook page

```
elif "open facebook" in query:
    wb.open("www.facebook.com")
```

Fig.4.26

### Defining Task 11: To open notepad

This task is used to open a notepad to note any required information.

```
while True:
    query = getcommand().lower()
    if "open notepad" in query:
```

```
path = "C:\\Windows\\System32\\notepad.exe"
os.startfile(path)
```

Fig.4.27

## Defining Task 12: To shut down or restart the computer

This task is used to automatically shut down, restart, or log out the window.

```
elif 'logout' in query:
    os.system("shutdown - 1")

elif 'shutdown' in query:
    os.system("shutdown /s /t 1")

elif 'restart' in query:
    os.system("shutdown /r /t 1")
```

Fig.4.28

### 4.1.10: PyAudio:

PyAudio provides Python binding for portAudio the cross-platform audio input library with pyaudio you can easily use Python to play and record audio on a variety of platform Here To recognize the voice of the user and to understand it we have to download and install the play audio file.

- Download PyAudio from Python unofficial libraries

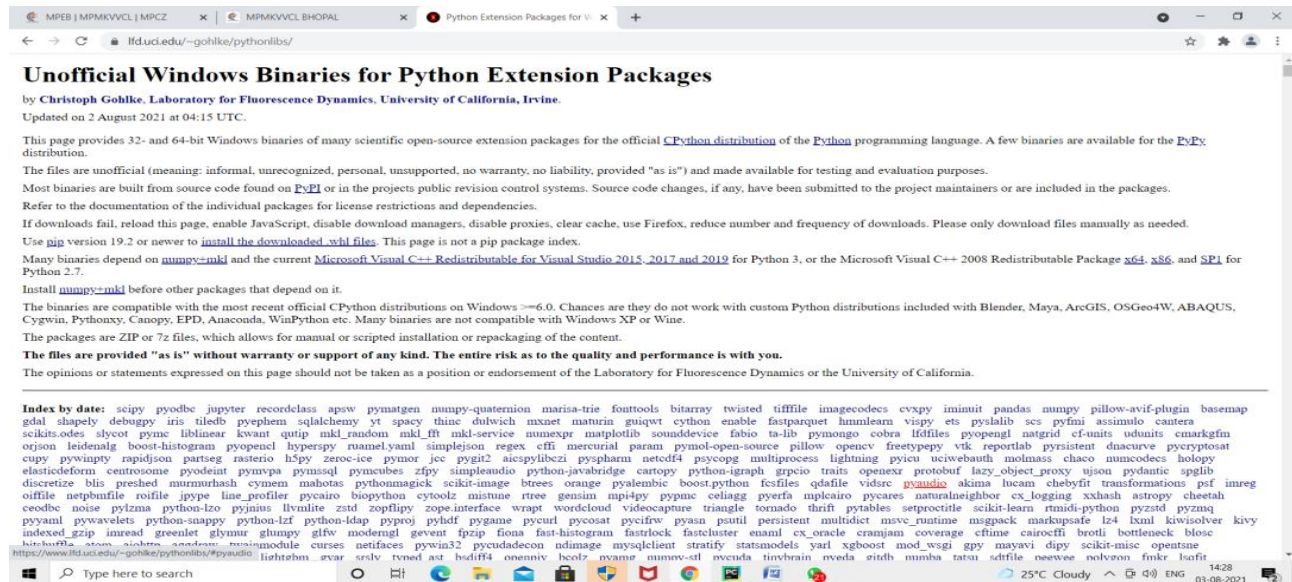


Fig 4.29

- Select and download according to your system need

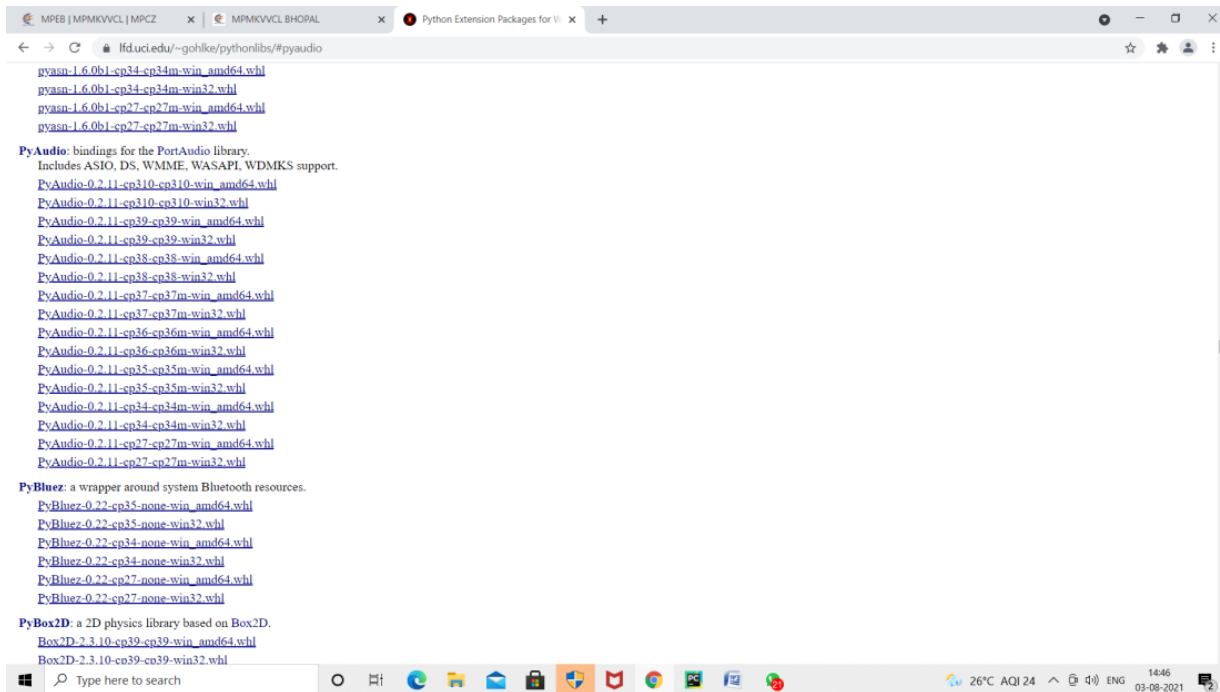


Fig 4.30

- Save it in same folder where main file is saved

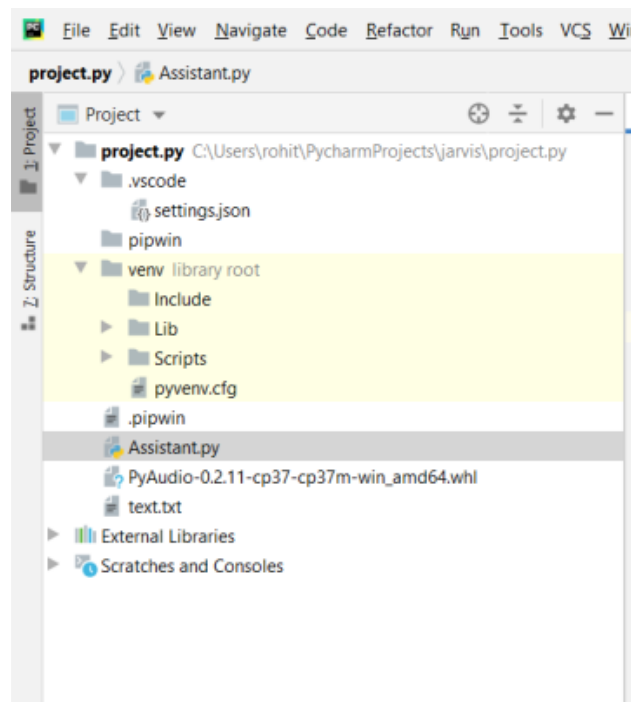


Fig 4.31

## CHAPTER 5

### CODING

#### 5.1: Code for AI-Based Assistant using python

```
import sys

import pyjokes
import pyttsx3
import datetime

import speech_recognition as sr
import wikipedia
import webbrowser as wb
import os
import random
import smtplib

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)
voicerate = 200
engine.setProperty('rate', voicerate)

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

def time():
    speak("current time is")
    speak(datetime.datetime.now().strftime("%H:%M:%S"))

def date_():
    year = int(datetime.datetime.now().year)
    month = int(datetime.datetime.now().month)
    day = int(datetime.datetime.now().day)
    speak("the current date is")
    speak(day)
    speak(month)
    speak(year)

def wishme():
```

```

hrs = int(datetime.datetime.now().hour)
if hrs < 12:
    speak("good morning ")
elif 12 < hrs < 18:
    speak("good afternoon")
elif hrs > 18:
    speak("good evening")

def getcommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("listening...")
        r.pause_threshold = 1
        audio = r.listen(source)
    try:
        print("Recognizing...")
        query = r.recognize_google(audio, language='en-us')

        print(f"User said: {query}\n")

    except:

        print("Say that again please...")
        return "None"
    return query

def jokes():
    joke = pyjokes.get_joke()
    print(joke)
    speak(joke)

def sendEmail(to, content):
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('useremail@gmail.com', 'your-password')
    server.sendmail('useremail@gmail.com', to, content)
    server.close()

if __name__ == "__main__":
    print("Hii!")
    wishme()
    speak("Hello User this is AI Assistant. how may i help u")

```



```

# date_()
while True:
    query = getcommand().lower()
    if "open notepad" in query:
        path = "C:\\Windows\\System32\\notepad.exe"
        os.startfile(path)
    elif "open facebook" in query:
        wb.open("www.facebook.com")
    elif 'wikipedia' in query:
        speak("searching wikipedia...")
        query = query.replace("wikipedia", "")
        results = wikipedia.summary(query, sentences=1)
        speak("according to wikipedia")
        print(results)
        speak(results)

    elif "open youtube" in query:
        wb.open("www.youtube.com")

    elif "google" in query:
        speak("what you want me to google")
        cm = getcommand().lower()
        wb.open(f"{cm}")

    elif "play music" in query:
        location = 'D:\\mobile backup\\Gym song'
        songs = os.listdir(location)
        print(songs, end="\n")
        x = random.randint(0, (len(songs)-1))
        print(x)
        os.startfile(os.path.join(location, songs[x]))

    elif "time" in query:
        time()

    elif 'chrome' in query:
        path = "C:\\Program Files (x86)\\Google\\Chrome\\Application\\chrome.exe"
        os.startfile(path)

    elif "visual code" in query:
        path = "C:\\Users\\Teja\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"
        os.startfile(path)

    elif 'email to user' in query:
        try:
            speak("What should I say?")
            content = getcommand()

```

```

        to = "useryourEmail@gmail.com"
        sendEmail(to, content)
        speak("Email has been sent!")

except Exception as e:
    print(e)
    speak("Opps!! I am getting trouble to send email")

elif 'logout' in query:
    os.system("shutdown - 1")

elif 'shutdown' in query:
    os.system("shutdown /s /t 1")

elif 'restart' in query:
    os.system("shutdown /r /t 1")

elif "remember that" in query:
    speak("what should i remember")
    notes = getcommand()
    speak(f"you said that {notes}")
    remember = open("text.txt", "w")
    remember.write(notes)
    remember.close()

elif "remember anything" in query:
    remember = open("text.txt", "r")
    speak(f"you once said that remember AI Assistant {remember.read()}")
    print(remember.read())
    remember.close()

elif "joke" in query:
    jokes()
elif 'stop' in query:
    speak("thank you")
    sys.exit()
elif 'nothing' in query:
    speak("thank you")
    sys.exit()
speak("what can i do for you")

```

## CHAPTER 6

### CONCLUSION AND FUTURE WORK

The main aim of the project was to develop a Desktop Assistant that will be used to identify answers related to user-submitted questions. To provide sufficient information that is required by the user. Background research took place, which included an overview of the conversation procedure and any relevant desktop Assistant available. A desktop Assistant already in user was excellent service that is provided. The developed system is made on the python programming language to be more specific Python 3.8. Different libraries were used such as Speech Recognition, Text to Speech Converter, Short Mail Transferring Protocols (SMTP). It provides information regarding the weather, News, it can play music, it can search for topics on Wikipedia, can set up an alarm, display the current date and time. Users can collect information through this application. It reduces both manpower and time. Due to support of NLP users can ask queries in a very formal way. No need to ask queries in a very strict and specific way. The user should aware of the general rules of the English Language. The goal is to provide people with a quick and easy way to have their questions answered.

#### 6.1: Limitation

❖ **It will not work without a net.**

It will work only by connecting to the Internet. If you run it without the internet or if you ask something then this will not work. Friends will work only when your mobile is connected to the internet and its mobile application is on your mobile, then it will only work.

❖ **Hindi is not fully supported.**

It does not fully support Hindi. If you ask for some words in Hindi that are not always used, the AI assistant will not answer it

❖ **Hang on to your mobile.**

If you use it, your mobile's storage and RAM are full or the mobile hangs up. For this reason, you cannot use it well. There are many problems with using your mobile.

❖ **Maximum battery use.**

Your mobile battery gets down very quickly due to which you have to charge the mobile again and again because the AI Assistant app is 24 hours a day because the Battery life of your mobile decreases.

❖ **Heat your mobile.**

Your mobile phone gets hot very quickly, due to which the problem starts in the mobile and the battery becomes very quickly down.

### ❖ **High data use.**

If you use AI Assistant in a lot of amounts, then your mobile internet ends very quickly. So that you have to recharge again and again. Because it lasts 24 hours and it keeps waiting for your command.

## **6.2: Benefits**

- It is quite easy to use as compared to other **AI assistance**.
- It can easily work with multiple commands.
- **AI Assistant** can be programmed with a custom command option.
- It is a secure system to involve in your daily life.
- It is an extremely helpful and useful system for disabled people.

## **6.3: Future Scopes**

Technology may be the engine but people will always be the drivers of growth. The **AI-Based Virtual Assistant** will continue to play an important role in the development strategy of business because it provides a double-edged solution to improve profitability by effectively lowering costs and increasing revenue through efficiency. The user experience will become much better, making interactions richer and more natural. Their ability to process information on-device will open a whole new range of opportunities. All in all, **AI-Based Virtual Assistants** will become more complex ecosystems that can support you in multiple areas of your everyday lives.

### **6.3.1: AI in Science and Research**

AI is making lots of progress in the scientific sector. Artificial Intelligence can handle large quantities of data and processes it quicker than human minds. This makes it perfect for research where the sources contain high data volumes.

AI is already making breakthroughs in this field. A great example is ‘Eve,’ which is an AI-based robot. It discovered an ingredient of toothpaste that can cure a dangerous disease like Malaria. Imagine a common substance present in an everyday item that is capable of treating Malaria; it’s a significant breakthrough, no doubt.

Drug discovery is a fast-growing sector, and AI is aiding researchers considerably in this regard. Biotechnology is another field where researchers are using AI to design microorganisms for industrial applications. Science is witnessing significant changes thanks to AI and ML.

### **6.3.2: AI in Cyber Security**

Cybersecurity is another field that’s benefitting from AI. As organizations are transferring their data to IT networks and the cloud, the threat of hackers is becoming more significant.

One triumphant attack can wreak havoc on an organization. To keep their data and resources secure, organizations are making massive investments in cybersecurity. The future scope of AI in cybersecurity is bright.

Cognitive AI is an excellent example of this field. It detects and analyses threats, while also providing insights to the analysts for making better-informed decisions. By using Machine Learning algorithms and Deep Learning networks, the AI gets better and more durable over time. This makes it capable of fighting more advanced threats that might develop with them.

Many institutions are using AI-based solutions to automate the repetitive processes present in cybersecurity. For example, IBM has IBM Resilient, which is an agnostic and open platform that gives infrastructure and hub for managing security responses.

Another field is fraud detection. AI can help in detecting frauds and help organizations and people in avoiding scams. For example, Recurrent Neural Networks are capable of detecting fraud in their early stages. They can scan extensive quantities of transactions and classify them according to their trustworthiness. By identifying fraudulent transactions and tendencies, organizations can save a lot of time and resources. It surely lessens the risk of losing money.

### **6.3.3: AI in Data Analysis**

Data analysis can benefit largely from AI and ML. AI algorithms are capable of improving with iterations, and this way, their accuracy, and precision increase accordingly. AI can help data analysts with handling and processing large datasets.

AI can identify patterns and insights that human eyes can't notice without putting in a lot of effort. Moreover, it is faster and more scalable at doing so. For example, Google Analytics has Analytics Intelligence, which uses machine learning to help webmasters get insights on their websites faster. You can ask Analytics Intelligence a question in simple English, and it would give you a prompt reply. It also provides webmasters with Smart Lists, Smart Goals, Conversion Probability, and other features that help the webmaster in improving the results of their site.

The scope of AI in data analytics is rising rapidly. Another example of AI applications in this sector is predicting outcomes from data. Such systems use the analytics data to predict results and the appropriate course of action to achieve those results. Learn more about AI applications.

As mentioned earlier, AI systems can handle tons of data and process it much faster than humans. So, they can take customer data and make more accurate predictions of customer behaviour, preferences, and other required factors. Helixa.ai is a great example of such an AI application. They use AI to provide insights into customer (or audience) behaviour for higher accuracy and better results. Agencies and marketers can use their services to build precise buyer personas and create better-targeted ad campaigns.

### **6.3.4: AI in Transport**

The transport sector has been using AI for decades. Aeroplanes have been using autopilot to steer them in the air since 1912. An autopilot system controls the trajectory of a plane, but it isn't restricted to aircraft alone. Ships and spacecraft also use autopilot to help them maintain the correct

course.

Autopilot helps the human operator and assists them in heading in the right direction. A pilot of a modern aircraft usually works for 7 minutes; the autopilot handles most of the steering of the plane. This allows the pilots to focus on other more important areas of the flight, such as the weather and the trajectory of the plane.

Another area where the future scope of AI is quite broad is driverless cars. Many companies are developing autonomous vehicles, which will rely heavily on AI and ML to operate optimally. Experts believe self-driving cars will bring many long-term and short-term benefits, including lower emissions and enhanced road safety. For example, self-driving cars will be free from human errors, which account for 90% of traffic accidents. Many companies, including Tesla and Uber, are developing these vehicles.

### **6.3.5: AI in Home**

AI has found a special place in people's homes in the form of Smart Home Assistants. Amazon Echo and Google Home are popular smart home devices that let you perform various tasks with just voice commands.

You can order groceries, play music, or even switch on/off the lights in your living room with just a few voice commands. Both of them rely on Voice Recognition technologies, which are a result of Artificial Intelligence and Machine Learning. They constantly learn from the commands of their users to understand them better and become more efficient.

Smart assistants are also present in mobile phones. Apple's Siri and Google Assistant are great examples of this sort. They also learn to recognize their users' voices to interpret them better all the time. And they can perform a plethora of tasks. Microsoft also has a smart assistant, which is called Cortana.

You can use these smart assistants for various tasks such as:

- Playing a song
- Asking a question
- Buying something online
- Opening an app
- There's a lot of room left for improvement, but surely, the scope of AI in the smart home sector is booming.

### **6.3.6: AI in Healthcare**

The medical sector is also using this technology to its advantage. AI is helping medical researchers and professionals in numerous ways.

For example, the Knight Career Institute and Intel have made a collaborative cancer cloud. This

cloud takes data from the medical history of cancer (and similar) patients to help doctors in making a better diagnosis. Preventing cancer from moving to higher stages is its most effective treatment at this time.

We've already mentioned how AI is helping researchers in their field too. Apart from finding a cure for cancer, some organizations are using AI to help patients get telemedicine. The UK's National Health Service uses Google's DeepMind platform to detect health risks in people through apps.

Wrong diagnoses are a significant problem in the medical sector. AI can help doctors in avoiding these errors by providing them with relevant databases and recommendations. It can analyse the database of patients with similar symptoms and suggest the treatment that was the most successful in those cases.

Many major organizations, including IBM and Microsoft, are collaborating with medical institutions to solve the various problems present in the healthcare sector.

AI can also help in reducing medical costs by preventing diseases beforehand and helping doctors in making better diagnoses. BCIs (Brain-computer Interfaces) is another area where the medical sector is utilizing AI. These interfaces help in predicting problems related to speaking or moving that might develop due to problems in the brain. They use AI to help these patients overcome these issues, too, by decoding neural activate.

## **CHAPTER 7**

### **REFERENCES**

1. GitHub repository and Code: <https://github.com/pali34/xuxuAssisant>
2. GitHub Pypi repository to contribute: <https://github.com/yadav/AI>
3. AI Library: <https://pypi.org/project/>
4. YouTube Channel: <https://www.youtube.com/pali34>
5. Demo and Code
6. [www.google.com](http://www.google.com)
7. [www.wikipedia.com](http://www.wikipedia.com)
8. <https://www.lfd.uci.edu/~gohlke/pythonlibs/>



Don't Worry! This report is 100% safe & secure. It's not available publically and it's not accessible by search engines (Google, Yahoo, Bing, etc)

## Sentence

```
import sys import pyjokes import pyttsx3 import datetime import speech_recognition as sr import
wikipedia import webbrowser as wb import os import random import smtplib engine =
pyttsx3.init('sapi5') voices = engine.getProperty('voices') engine.setProperty('voice', voices[0].id)
voicerate = 200 engine.setProperty('rate', voicerate) def speak(audio): engine.say(audio)
engine.runAndWait() def time(): speak("current time is")
speak(datetime.datetime.now().strftime("%H:%M:%S")) def date_(): year =
int(datetime.datetime.now().year) month = int(datetime.datetime.now().month) day =
int(datetime.datetime.now().day) speak("the current date is") speak(day) speak(month) speak(year) def
wishme(): hrs = int(datetime.datetime.now().hour) if hrs < 12> 18: speak("good evening") def
getcommand(): r = sr.Recognizer() with sr.Microphone() as source: print("listening...")
r.pause_threshold = 1 audio = r.listen(source) try: print("Recognizing...") query =
r.recognize_google(audio, language='en-us') print(f"User said: {query}\n") except: print("Say that
again please...") return "None" return query def jokes(): joke = pyjokes.get_joke() print(joke)
speak(joke) def sendEmail(to, content): server = smtplib.SMTP('smtp.gmail.com', 587) server.ehlo()
server.starttls() server.login('useremail@gmail.com', 'your-password')
server.sendmail('useremail@gmail.com', to, content) server.close() if __name__ == "__main__": print("Hii!")
wishme() speak("Hello User this is AI Assistant. how may i help u") # date_() while True: query =
getcommand().lower() if "open notepad" in query: path = "C:\\Windows\\System32\\notepad.exe"
os.startfile(path); elif "open facebook" in query: wb.open("www.facebook.com") elif
'wikipedia' in query: speak("searching wikipedia...") query = query.replace("wikipedia", "") results =
wikipedia.summary(query, sentences=1) speak("according to wikipedia") print(results) speak(results)
elif "open youtube" in query: wb.open("www.youtube.com") elif "google" in query: speak("what you
want me to google") cm = getcommand().lower() wb.open(f"{cm}") elif "play music" in query: location
= 'D:\\mobile backup\\Gym song' songs = os.listdir(location) print(songs,end="\n") x=
random.randint(0, (len(songs)-1)) print(x) os.startfile(os.path.join(location, songs[x])); elif
"time" in query: time() elif 'chrome' in query: path = "C:\\Program Files
(x86)\\Google\\Chrome\\Application\\chrome.exe" os.startfile(path); elif "visual code" in
query: path = "C:\\Users\\Teja\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"
os.startfile(path); elif 'email to user' in query: try: speak("What should I say?") content =
getcommand() to = "useryourEmail@gmail.com" sendEmail(to, content) speak("Email has been sent!")
except Exception as e: print(e) speak("Oops!! I am getting trouble to send email") elif 'logout' in query:
os.system("shutdown - 1"); elif 'shutdown' in query: os.system("shutdown /s /t 1");
elif 'restart' in query: os.system("shutdown /r /t 1"); elif "remember that" in query:
speak("what should i remember") notes = getcommand() speak(f"you said that {notes}") remember =
open("text.txt", "w") remember.write(notes) remember.close() elif "remember anything" in query:
remember = open("text.txt", "r") speak(f"you[41] once said that remember AI Assistant
{remember.read()}") print(remember.read()) remember.close() elif "joke" in query: jokes() elif 'stop' in
query: speak("thank you") sys.exit() elif 'nothing' in query: speak("thank you") sys.exit() speak("what
can i do for you")
```

<b>Report Title:</b>	pr
<b>Report Link:</b> (Use this link to send report to anyone)	<a href="https://www.check-plagiarism.com/plag-report/37384f6fdbdd522a2fae851a5c1b494e38741628197982">https://www.check-plagiarism.com/plag-report/37384f6fdbdd522a2fae851a5c1b494e38741628197982</a>
<b>Report Generated Date:</b>	05 August, 2021
<b>Total Words:</b>	1221
<b>Total Characters:</b>	4560
<b>Keywords/Total Words Ratio:</b>	0%
<b>Excluded URL:</b>	No
<b>Unique:</b>	86%
<b>Matched:</b>	14%

## Sentence wise detail:

```
import sys import pyjokes import pyttsx3 import datetime import speech_recognition as sr import wikipedia
```

```
import webbrowser as wb import os import random import smtplib engine = pyttsx3. init(sapi5;) voices = engine. (0)
```

```
import webbrowser as wb import os import random import smtplib engine = pyttsx3. getProperty(voices;) engine. (1)
```

```
import webbrowser as wb import os import random import smtplib engine = pyttsx3. setProperty(voice;, voices[0]. (2)
```

```
id) voicerate = 200 engine.
```

```
setProperty(rate;, voicerate) def speak(audio): engine. say(audio) engine. runAndWait()
```

```
def time(): speak(current time is;) speak(datetime. datetime. now().
```

```
strftime(quot;%H:%M:%Squot;)) def date_(): year = int(datetime. datetime. now(). year) month =
```

```
int(datetime. datetime. now().
```

```
month) day = int(datetime. datetime. now().
```

```
day) speak(quot;the current date isquot;) speak(day) speak(month) speak(year) def wishme(): hrs = int(datetime. datetime. now().
```

```
hour) if hrs < 12: speak(quot;good morning quot;) elif 12 < hrs < 18: speak(quot;good afternoonquot;) elif hrs > 18: speak(quot;good eveningquot;) def getcommand(): r = sr. Recognizer() with sr. (3)
```

```
Microphone() as source: print(quot;listening...quot;) r. pause_threshold = 1 audio = r. (4)
```

```
listen(source) try: print(quot;Recognizing...quot;) query = r.
```

```
recognize_google(audio, language=en-us;) print(fquot;User said: {query}\nquot;)
```

```
except: print(quot;Say that again please...
```

[42]

```
quot;) return quot;Nonequot; return query def jokes(): joke = pyjokes.
```

```
get_joke() print(joke) speak(joke) def sendEmail(to, content): server = smtplib. SMTP(smtp. gmail. com;,
```

```
587) server. ehlo() server. starttls() server. login(useremail@gmail. (5) com;,
```

```

&#039;your-password&#039;) server. sendmail(&#039;useremail@gmail. (5) com&#039;,
&#039;your-password&#039;) server. com&#039;, to, content) server. (7) close() if name == &quot;
main &quot;; print(&quot;Hii!
&quot;) wishme() speak(&quot;Hello User this is AI Assistant.
how may i help u&quot;) # date_() while True: query = getcommand().
lower() if &quot;open notepad&quot; in query: path = &quot;C:\\Windows\\System32\\notepad.
exe&quot; os. startfile(path) elif &quot;open facebook&quot; in query: wb. open(&quot;www.
facebook.
com&quot;) elif &#039;wikipedia&#039; in query: speak(&quot;searching wikipedia...&quot;) query
= query. replace(&quot;wikipedia&quot;, &quot;&quot;) results = wikipedia.
summary(query, sentences=1) speak(&quot;according to wikipedia&quot;) print(results) speak(results)
elif &quot;open youtube&quot; in query: wb. open(&quot;www. youtube.
com&quot;) elif &quot;google&quot; in query: speak(&quot;what you want me to google&quot;) cm
= getcommand(). lower() wb. (7)
open(f{cm}&quot;) elif &quot;play music&quot; in query: location = D:\\mobile backup\\Gym
song&#039; songs = os. listdir(location) print(songs,end=&quot;\n&quot;) x= random.
randint(0, (len(songs)-1)) print(x) os. startfile(os. path.
join(location, songs[x])) elif &quot;time&quot; in query: time() elif &#039;chrome&#039; in query:
path = &quot;C:\\Program Files (x86)\\Google\\Chrome\\Application\\chrome. exe&quot; os.
startfile(path) elif &quot;visual code&quot; in query: path =
&quot;C:\\Users\\Teja\\AppData\\Local\\Programs\\Microsoft VS Code\\Code. exe&quot; os.
startfile(path) elif &#039;email to user&#039; in query: try: speak(&quot;What should I say? &quot;)
content = getcommand() to = &quot;useryourEmail@gmail.
com&quot; sendEmail(to, content) speak(&quot;Email has been sent!

```

```

&quot;) except Exception as e: print(e) speak(&quot;Opps!!
I am getting trouble to send email&quot;) elif &#039;logout&#039; in query: os.
system(&quot;shutdown - 1&quot;) elif &#039;shutdown&#039; in query: os.
system(&quot;shutdown /s /t 1&quot;) elif &#039;restart&#039; in query: os.
system(&quot;shutdown /r /t 1&quot;) elif &quot;remember that&quot; in query: speak(&quot;what
should i remember&quot;) notes = getcommand() speak(f&quot;you said that {notes}&quot;)
remember = open(&quot;text.
txt&quot;, &quot;w&quot;) remember. write(notes) remember.
close() elif &quot;remember anything&quot; in query: remember = open(&quot;text. txt&quot;,
&quot;r&quot;) speak(f&quot;you once said that remember AI Assistant {remember. read()}&quot;)
print(remember. read()) remember.
close() elif &quot;joke&quot; in query: jokes() elif &#039;stop&#039; in query: speak(&quot;thank
you&quot;) sys. exit() elif &#039;nothing&#039; in query: speak(&quot;thank you&quot;) sys.

```

**Match Urls:**

- 0: <https://www.geeksforgeeks.org/voice-assistant-using-python/>
- 1: <https://akpythonblog.blogspot.com/2021/03/desktop-voice-assistant-using-40-lines.html>
- 2: [https://medium.com/analytics-vidhya/speech-synthesizer-using-python-b3f1c83a1fc8?source=post\\_internal\\_links](https://medium.com/analytics-vidhya/speech-synthesizer-using-python-b3f1c83a1fc8?source=post_internal_links) ----- 2
- 3: [https://www.programcreek.com/python/example/107721/speech\\_recognition.Recognizer](https://www.programcreek.com/python/example/107721/speech_recognition.Recognizer)
- 4: <https://www.codespeedy.com/design-jarvis-algorithm-using-python/>
- 5: <https://pastebin.com/g9MD1dZr>
- 6: <https://www.themidom.in/2021/07/how-to-send-email-using-python.html>
- 7: <https://pubmed.ncbi.nlm.nih.gov/33945893/>

### Keywords Density

One Word	2 Words	3 Words
query 6.31%	query speak 1.52%	int datetime datetime 1.01%
speak 5.81%	datetime datetime 1.26%	startfile path elif 0.76%
elif 4.29%	os startfile 1.01%	os system shutdown 0.76%
time 4.04%	int datetime 1.01%	exe os startfile 0.76%
date 3.54%	os system 0.76%	query os system 0.76%

## Plagiarism Report

By check-plagiarism.com

## Sentence

CHAPTER 4 PROJECT DESCRIPTION 4.1: Creating Voice Assistant using python (Steps) 4.1.3: Defining Greet Function: Now, we are going to make a Greet() function, that will make our AI ASSISTANT wish or greet the user according to the time of computer or pc. To provide current or live time to A.I., we need to import a module called datetime. Import this module to your program, by: `import datetime` Fig.4.5 Now, let's start defining the greet() function: `def greet(): hrs = int(datetime.datetime.now().hour) if hrs < 12 > 18: speak("good evening")` Fig.4.6 Here, we have stored the integer value of the current hour or time into a variable named hour. Now, we will use this hour value inside an if-else loop.

4.1.4: Defining getcommand Function: The next most important thing for our A.I. assistant is that it should be able to take command with the help of the microphone of the user's system. So, now we will make a getcommand() function. With the help of the takeCommand() function, our A.I. assistant will be able to return a string output by taking microphone input from the user. Before defining the takeCommand() function, we need to install a module called Speech Recognition. Install this module by: `pip install speechRecognition` Fig.4.7 After successfully installing this module, import this module into the program by writing an import statement. `import speechRecognition as sr` Fig.4.8 Let's start coding the getcommand() function. `def getcommand(): r = sr.Recognizer() with sr.Microphone() as source: print("listening...") r.pause_threshold = 1 audio = r.listen(source)` Fig.4.9 We have successfully created our getcommand() function. Now we are going to add a try and except block to our program to handle errors effectively. `try: print("Recognizing...") query = r.recognize_google(audio, language='en-us') print(f"User said: {query}\n") except: print("Say that again please...") return "None" return query` Fig.4.10

4.1.5: Defining Jokes Function: Here our AI ASSISTANT will tell us a joke. If the user wants to Entertain. It gives us an exciting joke you will laugh definitely. `def jokes(): joke = pyjokes.get_joke() print(joke) speak(joke)` Fig.4.11 We have successfully created our jokes function. Now we are going to add a try and except block to our program to handle errors effectively.

4.1.6: Defining Date Function: Here our AI assistant will be able to give information about the date in the format of month year and day. `def date_(): year = int(datetime.datetime.now().year) month = int(datetime.datetime.now().month) day = int(datetime.datetime.now().day) speak("the current date is") speak(day) speak(month) speak(year)` Fig.4.12

4.1.7: Defining Time Function: `def time(): speak("current time is") speak(datetime.datetime.now().strftime("%H:%M:%S"))` Fig.4.13

In the above, code with are using datetime() function and storing the current or live of the system into a variable called strTime. After storing the time in strTime, we are passing this variable as an argument in speak function. Now, the time string will be converted into the speech. 4.1.8: Defining mail function : Now, we will create a mail() function, which will help us to send emails to one or more recipients. `def mail(to, content): server = smtplib.SMTP('smtp.gmail.com', 587) server.ehlo() server.starttls() server.login('useremail@gmail.com', 'your-password') server.mail('useremail@gmail.com', to, content) server.close()` Fig.4.14

In the above code, we are <sup>[45]</sup> using the SMTP module, which we have already discussed above. Note: Do not forget to 'enable the less secure apps' feature in your Gmail account. Otherwise, the mail function will not work properly. 4.1.9: Coding logic of AI Now, we will develop logic for different commands such as Wikipedia searches, playing music, etc.

Defining Task 1: To search something on Wikipedia To do Wikipedia searches, we need to install and import the Wikipedia module into our program. Type the below command to install the Wikipedia module: `pip install wikipedia` Fig.4.15 After successfully installing the Wikipedia module, import it into the program by writing an import statement. if

```
name__ == "__main__": print("Hii!") greet() speak("Hello User this is AI Assistant. how may I help u") #
date_() while True: query = getcommand().lower() if "open notepad" in query: path =
"C:\\Windows\\System32\\notepad.exe" os.startfile(path) elif "open facebook" in query:
wb.open("www.facebook.com") elif 'wikipedia' in query: speak("searching wikipedia...") query =
query.replace("wikipedia", "") results = wikipedia.summary(query, sentences=1) speak("according to
wikipedia") print(results) speak(results) Fig 4.16
```

In the above code, we have used an if statement to check whether Wikipedia is in the search query of the user or not. If Wikipedia is found in the user's search query, then two sentences from the summary of the Wikipedia page will be converted to speech with the help of speak function.

Defining Task 2: To open YouTube site in a web-browser To open any website, we need to import a module called web- browser. It is an in-built module, and we do not need to install it with pip statement, we can directly import it into our program by writing an import statement. elif "open youtube" in query: `wb.open("www.youtube.com")` Fig.4.17 Here, we are using the elif loop to check whether the YouTube is in the query of the user or not. Let' suppose, the user gives command as "AI ASSISTANT open youtube." So, open YouTube will be in the user's query, and the elif condition will be true. Defining

Task 3: To open Google site in a web-browser elif "google" in query: `speak("what you want me to google") cm = getcommand().lower() wb.open(f"{cm}")` Fig.4.18 We are opening Google in a web browser by applying the same logic that we used to open YouTube. Defining Task 4: To play music To play music, we need to import a module called OS. Import this module directly with an import statement. elif "play music" in query: `location = 'D:\\mobile backup\\Gym`

```
song' songs = os.listdir(location) print(songs,end="\n") x= random.randint(0, (len(songs)-1)) print(x)
os.startfile(os.path.join(location, songs[x])) Fig.4.19
```

In the above code, we first opened our music directory and then listed all the songs present in the directory with the help of the OS module. With the help of `os.starfile`, you can play any song of your choice. I am playing the first song in the directory. However, you can also play a random song with the help of a random module. Every time you command to play music, J.A.R.V.I.S. will play any random song from the song directory. Defining Task 5: To know the current time elif "time" in query: `time()` Fig.4.20 In the above, code with are using `datetime()` function and storing the current or live of the system into a variable called `strTime`. After storing the time in `strTime`, we are passing this variable as an argument in speak function. Now, the time string will be converted into the speech. Defining Task 6: To open the VS Code Program elif "visual code" in query: `path = "C:\\Users\\Teja\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe" os.startfile(path)` Fig .4.21 To open the VS Code or any other application, we need the code path of the application. Steps to get the code path of the application: Step 1: Open the file location. Step 2: Right-click on the application and click on properties. Step 3: Copy the target from the target section. After copying the target of the application, save the target into a variable. Here, I am saving the target into a variable called `codePath`, and then we are using the os module to open the application. Defining Task 7: To mail To send an email, we need to import a module called `smtplib`. What is `smtplib`? • Simple Mail Transfer Protocol (SMTP) is a protocol that allows us to send emails and route emails between mail servers. An instance method called `mail` is present in the SMTP module. This instance method allows us to send an email. It takes 3 parameters: • The sender: Email address of the sender. • The receiver: Email of the receiver. • The message: A string message which needs to be sent to one or more than one recipient. Now, we will create a `mail()` function, which will help us to send emails to one or more recipients. elif 'email to user' in query: `try: speak("What should I say?") content`

= getcommand() to = "useryourEmail@gmail.com" sendEmail(to, content) speak("Email has been sent!") Fig.4.22 except Exception as e: print(e) speak("Oops!! I am getting trouble in sending email") Fig.4.23 We are using the try and except block to handle any possible error that can occur while sending mail. Defining Task 8: To Remember This is used to remember things. our AI ASSISTANT can remember anything if you said that remember. elif "remember that" in query: speak("what should I remember") notes = getcommand() speak(f"you said that {notes}") remember = open("text.txt", "w") remember.write(notes) remember.close() elif "remember anything" in query: remember = open("text.txt", "r") speak(f"you once said that remember AI Assistant {remember.read()}") print(remember.read()) remember.close() Fig 4.24 Defining Task 9: Tell Us a Joke This task is used for entertainment purpose. Fig 4.25 Defining Task 10: To open Facebook This task is used for the opening of Facebook page elif "open facebook" in query: wb.open("www.facebook.com") Fig.4.26 Defining Task 11: To open notepad This task is used to open a notepad to note any required information. while True: query = getcommand().lower() if "open notepad" in query: path = "C:\\Windows\\System32\\notepad.exe" os.startfile(path); Fig.4.27 Defining Task 12: To shut down or restart the computer This task is used to automatically shut down, restart, or log out the window. elif 'logout' in query: os.system("shutdown - l"); elif 'shutdown' in query: os.system("shutdown /s /t 1"); elif 'restart' in query: os.system("shutdown /r /t 1"); Fig.4.28 4.1.10: PyAudio: PyAudio provides Python binding for portAudio the cross-platform audio input library with pyaudio you can easily use Python to play and record audio on a variety of platform Here To recognize the voice of the user and to understand it we have to download and install the play audio file. • Download PyAudio from Python unofficial libraries Fig 4.29 • Select and download according to your system need Fig 4.30 • Save it in same folder where main file is saved Fig 4.31

<b>Report Title:</b>	ed
<b>Report Link:</b> (Use this link to send report to anyone)	<a href="https://www.check-plagiarism.com/plag-report/3738472baee8b553480000996a6bd8a92ffd31628198134">https://www.check-plagiarism.com/plag-report/3738472baee8b553480000996a6bd8a92ffd31628198134</a>
<b>Report Generated Date:</b>	05 August, 2021
<b>Total Words:</b>	2819
<b>Total Characters:</b>	11931
<b>Keywords/Total Words Ratio:</b>	0%
<b>Excluded URL:</b>	No
<b>Unique:</b>	71%
<b>Matched:</b>	29%
<b>Sentence wise detail:</b>  CHAPTER 4 PROJECT DESCRIPTION 4. (0)	



1: Creating Voice Assistant using python (Steps) 4.1. (1)

3: Defining Greet Function: Now, we are going to make a Greet() function, that will make our AI ASSISTANT wish or greet the user according to the time of computer or pc.

To provide current or live time to A. I. , we need to import a module called datetime. (2) Import this module to your program, by: import datetime Fig.4.

5 Now, lets start defining the greet() function: def greet(): hrs = int(datetime. datetime. now().

hour) if hrs < 12: speak(good morning ") elif 12 <= hrs < 18: speak("good afternoon") elif hrs >= 18: speak("good evening") Fig.4.

6 Here, we have stored the integer value of the current hour or time into a variable named hour. Now, we will use this hour value inside an if-else loop. 4.1.

4: Defining getcommand Function: The next most important thing for our A. I. assistant is that it should be able to take command with the help of the microphone of the user's system. (3)

So, now we will make a getcommand() function. With the help of the takeCommand() function, our A. I. (2)

So, now we will make a getcommand() function. assistant will be able to return a string output by taking microphone input from the user. (5)

Before defining the takeCommand() function, we need to install a module called Speech Recognition. Install this module by: pip install speechRecognition Fig.4.

7 After successfully installing this module, import this module into the program by writing an import statement. import speechRecognition as sr Fig.4.

8 Let's start coding the getcommand() function. def getcommand(): r = sr. Recognizer() with sr. (6)

Microphone() as source: print("listening...") r. pause\_threshold = 1 audio = r. (7)  
listen(source) Fig.4.

9 We have successfully created our getcommand() function. Now we are going to add a try and except block to our program to handle errors effectively. (8)

9 We have successfully created our getcommand() function. try: print("Recognizing...")  
query = r. (9)

9 We have successfully created our getcommand() function. recognize\_google(audio,  
language='en-us') print(f"User said: {query}\n") except: print("Say  
that again please... (10)

") return "None" return query Fig.4.10 4.1. 5: Defining Jokes Function: Here our  
AI ASSISTANT will tell us a joke. (11)

If the user wants to Entertain.

It gives us an exciting joke you will laugh definitely. def jokes(): joke = pyjokes.

get\_joke() print(joke) speak(joke) Fig.4.

11 We have successfully created our jokes function.

Now we are going to add a try and except block to our program to handle errors effectively. 4.1.

6: Defining Date Function: Here our AI assistant will be able to give information about the date in



the format of month year and day.

```
def date_(): year = int(datetime. datetime. now(). year) month = int(datetime. datetime. now(). month) day = int(datetime. datetime. now().
```

```
day) speak(&quot;the current date is&quot;); speak(day) speak(month) speak(year) Fig.4.12 4.1.
```

```
7: Defining Time Function: def time(): speak(&quot;current time is&quot;); speak(datetime. datetime. now(). strftime(&quot;%H:%M:%S&quot;)) Fig.4.
```

13 In the above, code with are using datetime() function and storing the current or live of the system into a variable called strTime. After storing the time in strTime, we are passing this variable as an argument in speak function. (12)

Now, the time string will be converted into the speech. 4.1.

8: Defining mail function : Now, we will create a mail() function, which will help us to send emails to one or more recipients.

```
def mail(to, content): server = smtplib. SMTP(smtp. gmail. com&#039;;, 587) server. ehlo() server. starttls() server.
```

```
login(&#039;useremail@gmail. (13)
```

```
com&#039;;, &#039;your-password&#039;) server. mail(&#039;useremail@gmail.
```

```
com&#039;;, to, content) server. close() Fig.4.
```

14 In the above code, we are using the SMTP module, which we have already discussed above. Note: Do not forget to &#039;enable the less secure apps&#039; feature in your Gmail account. (14)

Otherwise, the mail function will not work properly. 4.1.

9: Coding logic of AI Now, we will develop logic for different commands such as Wikipedia searches, playing music, etc. Defining Task 1: To search something on Wikipedia To do Wikipedia searches, we need to install and import the Wikipedia module into our program. (12)

Type the below command to install the Wikipedia module: pip install wikipedia Fig.4.

15 After successfully installing the Wikipedia module, import it into the program by writing an import statement. if  
name\_\_==\_\_main\_\_&quot;;: print(&quot;Hii! (16) &quot;); greet() speak(&quot;Hello User this is AI Assistant.

```
how may I help u&quot;); # date_() while True: query = getcommand().
```

```
lower() if &quot;open notepad&quot; in query: path = &quot;C:\\Windows\\System32\\notepad. exe&quot;; os. startfile(path) elif &quot;open facebook&quot; in query: wb. open(&quot;www. facebook.
```

```
com&quot;); elif &#039;wikipedia&#039; in query: speak(&quot;searching wikipedia...&quot;);
```

```
query = query. replace(&quot;wikipedia&quot;;, &quot;&quot;); results = wikipedia.
```

```
summary(query, sentences=1) speak(&quot;according to wikipedia&quot;); print(results) speak(results) Fig 4.
```

16 In the above code, we have used an if statement to check whether Wikipedia is in the search query of the user or not. If Wikipedia is found in the user&#039;s search query, then two sentences from the summary of the Wikipedia page will be converted to speech with the help of speak function. (8)

Defining Task 2: To open YouTube site in a web-browser To open any website, we need to import a module called web- browser. It is an in-built module, and we do not need to install it with pip

statement, we can directly import it into our program by writing an import statement. (8)

elif &quot;open youtube&quot; in query: wb. open(&quot;www. youtube. com&quot;;) Fig.4.

17 Here, we are using the elif loop to check whether the YouTube is in the query of the user or not. Let&#039; suppose, the user gives command as &quot;AI ASSISTANT open youtube. (8)

17 Here, we are using the elif loop to check whether the YouTube is in the query of the user or not. &quot; So, open YouTube will be in the user&#039;s query, and the elif condition will be true. (19)

Defining Task 3: To open Google site in a web-browser elif google&quot; in query: speak(&quot;what you want me to google&quot;;) cm = getcommand(). lower() wb. open(f&quot;{cm}&quot;;) Fig.4.

18 We are opening Google in a web browser by applying the same logic that we used to open YouTube.

Defining Task 4: To play music To play music, we need to import a module called OS. Import this module directly with an import statement. (19)

elif &quot;play music&quot; in query: location = D:\\mobile backup\\Gym song&#039; songs = os. listdir(location) print(songs,end=&quot;\n&quot;;) x= random. randint(0, (len(songs)-1)) print(x) os. startfile(os. path. join(location, songs[x])) Fig.4.

19 In the above code, we first opened our music directory and then listed all the songs present in the directory with the help of the OS module. With the help of os. (12)

19 In the above code, we first opened our music directory and then listed all the songs present in the directory with the help of the OS module. starfile, you can play any song of your choice. (19)

19 In the above code, we first opened our music directory and then listed all the songs present in the directory with the help of the OS module. I am playing the first song in the directory. (3)

19 In the above code, we first opened our music directory and then listed all the songs present in the directory with the help of the OS module. However, you can also play a random song with the help of a random module. (3)

Every time you command to play music, J. A. R. V. I. S. will play any random song from the song directory. (12) Defining Task 5: To know the current time elif &quot;time&quot; in query: time() Fig.4.

20 In the above, code with are using datetime() function and storing the current or live of the system into a variable called strTime. After storing the time in strTime, we are passing this variable as an argument in speak function. (19)

20 In the above, code with are using datetime() function and storing the current or live of the system into a variable called strTime. Now, the time string will be converted into the speech. (19)

Defining Task 6: To open the VS Code Program elif &quot;visual code&quot; in query: path = &quot;C:\\Users\\Teja\\AppData\\Local\\Programs\\Microsoft VS Code\\Code. exe&quot; os. startfile(path) Fig .4.

21 To open the VS Code or any other application, we need the code path of the application. Steps to get the code path of the application: Step 1: Open the file location. (12)

21 To open the VS Code or any other application, we need the code path of the application. Step 2: Right-click on the application and click on properties. (12)

21 To open the VS Code or any other application, we need the code path of the application. Step 3: Copy the target from the target section. (8)

21 To open the VS Code or any other application, we need the code path of the application. After copying the target of the application, save the target into a variable. (8)

21 To open the VS Code or any other application, we need the code path of the application. Here, I am saving the target into a variable called codePath, and then we are using the os module to open the application. (19)

Defining Task 7: To mail To send an email, we need to import a module called smtplib. What is smtplib? • Simple Mail Transfer Protocol (SMTP) is a protocol that allows us to send emails and route emails between mail servers. (12)

An instance method called mail is present in the SMTP module. This instance method allows us to send an email. (12)

An instance method called mail is present in the SMTP module. It takes 3 parameters: • The sender: Email address of the sender. (12)

- The receiver: Email of the receiver. • The message: A string message which needs to be sent to one or more than one recipient. (12)

Now, we will create a mail() function, which will help us to send emails to one or more recipients.

```
elif email to user'#039; in query: try: speak(What should I say?
```

```
&quot;;) content = getcommand() to = &quot;useryourEmail@gmail.
```

```
com&quot; sendEmail(to, content) speak(&quot;Email has been sent!&quot;);) Fig.4. 22 except
```

```
Exception as e: print(e) speak(&quot;Opps!!
```

```
I am getting trouble in sending email&quot;);) Fig.4.
```

23 We are using the try and except block to handle any possible error that can occur while sending mail. Defining Task 8: To Remember This is used to remember things.

our AI ASSISTANT can remember anything if you said that remember.

```
elif &quot;remember that&quot;; in query: speak(&quot;what should I remember&quot;);) notes =  
getcommand() speak(f&quot;you said that {notes}&quot;);) remember = open(&quot;text.
```

```
txt&quot;;, &quot;w&quot;);) remember. write(notes) remember.
```

```
close() elif &quot;remember anything&quot;; in query: remember = open(&quot;text. txt&quot;;,
```

```
&quot;r&quot;);) speak(f&quot;you once said that remember AI Assistant {remember.
```

```
read()}&quot;);) print(remember. read()) remember. close() Fig 4.
```

24 Defining Task 9: Tell Us a Joke This task is used for entertainment purpose. Fig 4.

25 Defining Task 10: To open Facebook This task is used for the opening of Facebook page elif &quot;open facebook&quot;; in query: wb. open(&quot;www. facebook. com&quot;);) Fig.4.

26 Defining Task 11: To open notepad This task is used to open a notepad to note any required information. while True: query = getcommand().

```
lower() if &quot;open notepad&quot;; in query: path = &quot;C:\\Windows\\System32\\notepad.  
exe&quot;; os. startfile(path) Fig.4.
```

27 Defining Task 12: To shut down or restart the computer This task is used to automatically shut down, restart, or log out the window.

```
elif &quot;logout&quot;; in query: os.
```

```
system(&quot;shutdown - 1&quot;); elif &#039;shutdown&#039;; in query: os.  
system(&quot;shutdown /s /t 1&quot;); elif &#039;restart&#039;; in query: os.  
system(&quot;shutdown /r /t 1&quot;);
```

 Fig.4.28 4.1.

10: PyAudio: PyAudio provides Python binding for portAudio the cross-platform audio input library with pyaudio you can easily use Python to play and record audio on a variety of platform Here To recognize the

voice of the user and to understand it we have to download and install the play audio file. (19)

- Download PyAudio from Python unofficial libraries Fig 4.

29 • Select and download according to your system need Fig 4. 30 • Save it in same folder where main file is saved Fig 4.31

### Match Urls:

0: <https://www.escondido.org/Data/Sites/1/media/pdfs/Planning/oakcreek/CD/VolII-Chapter4-ProjectDescription.pdf>

1: [https://www.reddit.com/r/linux/comments/gcdpcn/linux\\_voice\\_assistant\\_with\\_pythondetails\\_and/](https://www.reddit.com/r/linux/comments/gcdpcn/linux_voice_assistant_with_pythondetails_and/)

2: <https://coderspacket.com/jarvis-ai-desktop-voice-assistant-using-python>

3: <https://pulkitsinghdev.hashnode.dev/iron-man-jarvis-ai-desktop-voice-assistant>

4: [https://github.com/shivamkumar0611/AI\\_Jarvis](https://github.com/shivamkumar0611/AI_Jarvis)

5: [https://www.programcreek.com/python/example/107721/speech\\_recognition.Recognizer](https://www.programcreek.com/python/example/107721/speech_recognition.Recognizer)

6: <https://stackoverflow.com/questions/65769263/assertionerror-audio-source-must-be-entered-before-listening>

7: <https://spyboy.blog/2021/01/12/iron-man-jarvis-ai-desktop-voice-assistant-in-python/comment-page-1/>

8: <https://www.codespeedy.com/print-mic-name-device-id-in-python/>

9: <https://github.com/scorpion004/luna-voice-assistant/blob/master/luna.py>

10: <https://comidoc.net/udemy/learn-to-create-ai-voice-assistant-jarvis-with-python>

11: <https://www.codewithharry.com/videos/python-tutorials-for-absolute-beginners-120/>

12: <https://pastebin.com/g9MD1dZr>

13: <https://spyboy.blog/2021/01/12/iron-man-jarvis-ai-desktop-voice-assistant-in-python/>

14: <https://codefather.tech/blog/if-name-main-python/>

15: <https://medium.com/analytics-vidhya/a-guide-to-your-own-a-i-voice-assistant-using-python-17f79c94704>

16: <https://nurserysource.com/journal/harry-jarvis-partner-0218a6>

17: [https://kaseloguxov.weebly.com/uploads/1/3/1/4/131406527/vumikemin\\_tirogiza\\_nuziwav\\_neraru.pdf](https://kaseloguxov.weebly.com/uploads/1/3/1/4/131406527/vumikemin_tirogiza_nuziwav_neraru.pdf)

18: <https://medium.com/lets-talk-ml/guide-to-your-very-own-a-i-virtual-assistant-in-python-d8120ff228a8>

19: <https://www.merriam-webster.com/dictionary/file>

### Keywords Density

One Word	2 Words	3 Words
time 3.23%	defining task 1.17%	int datetime datetime 0.39%
mail 3.03%	import module 0.68%	import module called 0.39%
query 2.74%	ai assistant 0.68%	system shutdown 1 0.29%
open 2.64%	module called 0.49%	exe os startfile 0.29%
speak 2.15%	datetime datetime 0.49%	query wb open 0.29%

## Plagiarism Report

By check-plagiarism.com