VOICE BASED VIRTUAL ASSISTANT FOR WINDOWS USING AI

Minor project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

V.YASHWANTH REDDY (20UECS1003) **(VTU12688)**
S.SAI GANESH REDDY (20UECS0906) **(VTU16997)**
FAIZUL AZIZ (19UECS0266) **(VTU16249)**

Under the guidance of
Mr. Sharad Shandhi Ravi,M.Tech.,
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2023

VOICE BASED VIRTUAL ASSISTANT FOR WINDOWS USING AI

Minor project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

V.YASHWANTH REDDY (20UECS1003) **(VTU12688)**
S.SAI GANESH REDDY (20UECS0906) **(VTU16997)**
FAIZUL AZIZ (19UECS0266) **(VTU16249)**

Under the guidance of
MR. Sharad Shandhi Ravi,M.Tech.,
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2023

CERTIFICATE

It is certified that the work contained in the project report titled "VOICE BASED VIRTUAL ASSISTANT FOR WINDOWS USING AI" by V.YASHWANTH REDDY (20UECS1003), S.SAI GANESH REDDY (20UECS1003), FAIZUL AZIZ (19UECS0266) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Mr. Sharad Shandhi Ravi
Assistant Professor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2023

Signature of Head of the Department
Dr. Murali Dhar M. S
Associate Professor & HoD
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2023

Signature of the Dean
Dr. V. Srinivasa Rao
Professor & Dean
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2023

DECLARATION

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

V.YASHWANTH REDDY

Date: / /

S.SAI GANESH REDDY

Date: / /

FAIZUL AZIZ

Date: / /

APPROVAL SHEET

This project report entitled VOICE BASED VIRTUAL ASSISTANT FOR WINDOWS USING AI by V.YASHWANTH REDDY (20UECS1003), S.SAI GANESH REDDY (20UECS1003), FAIZUL AZIZ (19UECS0266) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Mr.Sharad Shandhi Ravi, M.Tech.,

Date: / /

Place:Chennai

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering,Dr.M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our Internal Supervisor **Mr.SHARAD SHANDHI RAVI,M.TECH.,** for his cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

V.YASHWANTH REDDY (20UECS1003)
S.SAI GANESH REDDY (20UECS0906)
FAIZUL AZIZ (19UECS0266)

ABSTRACT

Voice-based Virtual Assistants has become increasingly popular in recent years due to advancements in artificial intelligence (AI) technology. In this project, a voice-based virtual assistant is developed for the Windows operating system using AI. The virtual assistant is designed to understand and respond to natural language voice commands from users, and perform a variety of tasks such as opening and clos- ing applications, searching the web, setting reminders, and sending messages. The system uses deep learning algorithms to improve the accuracy of its speech recog- nition (SR) and natural language processing (NLP) capabilities. This project aims to provide an efficient and convenient way for users to interact with their Windows computer, reducing the need for manual input and increasing productivity.The development of a voice-based virtual assistant for Windows using AI represents a significant advancement in the field of human-computer interaction (HCI). The virtual assistant is designed to understand and respond to natural language voice commands from users, performing tasks such as opening and closing applications, searching the web, setting reminders, and sending messages. The system uses deep learning algorithms to improve the accuracy of its speech recognition (SR) and natural language processing (NLP) capabilities, enabling it to understand a wide variety of commands.

Keywords: Natural Language Processing (NLP), Artificial Intelligence (AI), Speech Recognition (SR), Human-Computer Interaction (HCI)

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

LIST OF ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
ML	MachineLearning
NLP	Natural Language Processing
HCI	Human-Computer Interaction
SR	Speech Recognition
TTS	Text-To-Speech

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

In today's era almost all tasks are digitalized. We have Smartphone in hands and it is nothing less than having world at your finger tips. These days we aren't even using fingers. We just speak of the task and it is done. There exist systems where we can say Text Dad, "I'll be late today." And the text is sent.

Virtual Assistants are software programs that help you ease your day to day tasks, such as showing weather report, creating reminders, making shopping lists etc. They can take commands via text (online chat bots) or by voice. Voice based intelligent assistants need an invoking word or wake word to activate the listener, followed by the command.

For my project the wake word is crypterium .We have so many virtual assistants, such as Apple's Siri, Amazon's Alexa and Microsoft's Cortana. For this project, wake word was chosen crypterium.

A voice-based virtual assistant for Windows using AI is a promising application of AI and NLP technologies that has the potential to greatly enhance the way users interact with their Windows-based devices. By leveraging the power of voice commands and natural language understanding, it can provide a convenient, efficient, and accessible way for users to perform tasks, improve productivity, and enhance overall user experience. However, it also requires careful consideration of various technical and design aspects to create a reliable, secure, and user-friendly virtual assistant.

Using artificial intelligence (AI), a virtual assistant can understand natural language commands, recognize user preferences, and provide personalized responses to users' queries. This technology has advanced significantly in recent years, enabling virtual assistants to perform a wide range of tasks, from setting reminders and scheduling

appointments to controlling smart home devices and providing weather updates.

Voice-based virtual assistants have become increasingly popular over the years due to their convenience and ease of use. Windows is one of the most widely used operating systems in the world, and integrating a voice-based virtual assistant into it can enhance the user experience by providing a more natural way of interacting with the system.

1.2 Aim of the Project

The primary aim of project is to provide an efficient and convenient way for users to interact with their Windows computer, reducing the need for manual input and increasing productivity. By using voice commands instead of typing or clicking, users can perform tasks more quickly and easily, especially those who may have mobility or vision impairments.

Additionally, the project aims to continuously improve the system's accuracy and functionality by using AI technology. As users interact with the virtual assistant and provide feedback, the system can learn and adapt to their preferences, making it even more efficient over time. The use of data analytics can also help identify patterns in user behavior and provide personalized recommendations and assistance.

1.3 Project Domain

Virtual assistants use natural language processing (NLP) to match user text or voice input to executable commands. Many continually learn using artificial intelligence techniques including machine learning. Some of these assistants like Google Assistant(which contains Google Lens) and Samsung Bixby also have the added ability to do image processing to recognize objects in the image to help the users .

A Voice-based Virtual Assistant for Windows using AI could be in the field of artificial intelligence, natural language processing, voice recognition, and user interface

design. The project could involve developing a conversational agent that can understand and respond to user requests using voice commands. The virtual assistant could perform tasks such as opening applications, setting reminders, searching the web, providing weather updates, and answering general knowledge questions. The project could also involve integrating machine learning algorithms to improve the accuracy and efficiency of the virtual assistant's performance over time.

1.4 Scope of the Project

Voice assistants will continue to offer more individualized experiences as they get better at differentiating between voices. However, it's not just developers that need to address the complexity of developing for voice as brands also need to understand the capabilities of each device and integration and if it makes sense for their specific brand.

They will also need to focus on maintaining a user experience that is consistent within the coming years as complexity becomes more of a concern. This is because the visual interface with voice assistants is missing. Users simply cannot see or touch a voice interface.

Developing the capability for the virtual assistant to execute tasks based on user commands. This may include integrating with various Windows applications, services, and APIs to perform actions such as opening applications, sending emails, creating calendar events, playing media, and more.

Chapter 2

LITERATURE REVIEW

[1] Douglas O'Shaughnessy et al,(2019) described about statement and speech which is most significant. In the communication between human and machine arrangement was done through analog signal which is converted by speech signal to digital wave. This technology is massively utilized, it has limitless uses and permit machines to reply appropriately and consistently to user voices, also offers useful and appreciated facilities. Speech Recognition System (SRS) is rising gradually and has indefinite applications. The research has revealed the summary of the procedure; it is a simple mode.

[2] F. M. Sultan et al,(2020) explained that most general mode of communication among human beings is speech. As this is the utmost technique, human beings would identical to utilize speech to interrelate with machines too. Because of this, autonomous speech identification has got a lot of reputation. Most techniques for speech recognition be like Dynamic Time Warping (DTW), HMM. For the feature mining of speech Mel Frequency Cepstrum Coefficients (MFCC) has been utilized which offers a group of characteristic vectors of speech waveform.

[3] H. M. Niazi et al, (2019) described a comprehensive review of virtual assistants that are available for Windows operating systems. The authors review different virtual assistants, including Cortana, Alexa, and Google Assistant. The paper highlights the features and limitations of each assistant and provides recommendations for selecting a virtual assistant for a specific task.

[4] K. S. Kulkarni et al, (2018) proposed a virtual assistant system that is designed specifically for Windows OS. The system utilizes speech recognition and natural language processing techniques to understand user commands and perform tasks. The system was tested on a small group of users, and the results showed that it was able to accurately understand user commands and perform tasks.

[5] M. Shahzad et al,(2019) described about the Language that is the utmost significant means of communication and speech is its major interface. The interface for human to machine, speech signal was converted into analog and digital wave shape as a machine understood.

[6] Q.J. Awan et al. (2020) explained a virtual assistant for Windows operating systems that utilizes artificial intelligence techniques to perform tasks. The authors use natural language processing algorithms and deep learning models to improve the accuracy of the assistant. The system was tested on a small group of users, and the results showed that it was able to accurately perform tasks.

[7] R. A. Jimenez et al, (2018) described about the development of a voice-activated personal assistant system for Windows operating system. The system utilizes speech recognition and natural language processing techniques to understand user commands and perform tasks. The authors tested the system on a small group of users, and the results showed that it was able to accurately understand user commands and perform tasks.

[8] S. Kumar et al, (2021) described about virtual assistants for Windows operating systems. The authors review different virtual assistants, including Cortana, Alexa, and Siri. The paper highlights the features and limitations of each assistant and provides recommendations for selecting a virtual assistant for a specific task. The authors also discuss the future trends of virtual assistants and their potential impact on human-computer interaction.

[9] S. Chen et al,(2018) described a voice-controlled personal assistant for Windows using Python and the Microsoft Speech API to perform various tasks such as opening applications, sending emails, and playing music.as well as a proposal for building a smart voice assistant for Windows using AI and machine learning techniques.

[10] S. R. Bhadane et al, (2019) explained about Windows voice-based virtual assistant that uses artificial intelligence techniques such as machine learning and natural language processing to perform tasks. The authors developed a prototype system and tested it on a small group of users. The results showed that the system

was able to accurately perform tasks and improve its accuracy with increased usage.

[11] S.K. Das et al,(2019) presented a Windows-based voice assistant that uses NLP techniques to recognize voice commands and execute various tasks, such as opening applications, searching the web, and sending emails.

[12] S. J. Atal et al,(2020) explained regarding speech analysis, and result is regularly completed in combination with pitch analysis. The research described a pattern recognition technique for determining whether a given slice of a speech signal should be categorized as voiced speech, unvoiced speech, or silence, depending on dimensions finished on signal. The main restriction of the technique is the requirement for exercise the algorithm on exact set of dimensions picked, and for the specific recording circumstances.

[13] T. H. Le et al ,(2020) explained about Intelligent Personal Assistant for Windows using AI. This provides a review of various NLP techniques used for voice-based virtual assistants, such as speech recognition, sentiment analysis, intent detection, and dialogue management.

[14] V. J. Pillai et al ,(2018) described a voice-controlled personal assistant for Windows using NLP techniques, such as speech recognition, intent recognition, and dialogue management,a virtual assistant for Windows using the Microsoft Bot Framework and Language Understanding Intelligent Service (LUIS) to recognize and respond to voice commands.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The term virtual assistant, or virtual personal assistant, is also commonly used to describe contract workers who work from home doing administrative tasks typically performed by executive assistants or secretaries. Virtual assistants are typically cloud-based programs that require internet-connected devices and/or applications to work. Three such applications are Siri on Apple devices, Cortana on Microsoft Devices and Google Assistant on Android devices. There are also devices dedicated to providing virtual assistance. The most popular ones are available from Amazon, Google and Microsoft. To use the Amazon Echo virtual assistant, called Alexa, users call out the wake word, “Alexa.”

3.2 Proposed System

To design a device that acts as a digital organizer to provide variety of services to its master. It will look at examples of intelligent programs with natural language processing that are currently available, with different categories of support, and examine the potential usefulness of one specific piece of software as a VPA.

It continues to expand its digital abilities in organizing events, ordering food, playing music, guiding services for travelling, game prediction etc. It is suggested that new technologies 5 may soon make the idea of virtual personal assistants a reality.

3.3 Feasibility Study

3.3.1 Economic Feasibility

Voice assistants have become increasingly popular over the past few years, and their economic feasibility can be evaluated from various angles. From the consumer side, voice assistants are generally affordable and accessible, with many devices such as smartphones, smart speakers, and home assistants offering built-in voice assistant capabilities. This has made them a popular choice for consumers who want a hands-free and convenient way to interact with technology. From the business side, voice assistants can offer significant cost savings and operational efficiencies. For example, businesses can use voice assistants to automate customer service and support, reducing the need for human customer service representatives. Additionally, voice assistants can help businesses collect and analyze customer data to improve their products and services. However, there are also some challenges to the economic feasibility of voice assistants. For example, developing and maintaining a high-quality voice assistant can be expensive, especially for small and medium-sized businesses. Additionally, concerns around data privacy and security may impact consumer trust in voice assistants, which could limit their widespread adoption and economic viability. Overall, the economic feasibility of voice assistants depends on factors such as the cost of development and implementation, consumer adoption and trust, and the potential cost savings and operational efficiencies they can provide. While there are challenges, the growing popularity of voice assistants suggests that they have significant economic potential for both consumers and businesses.

3.3.2 Technical Feasibility

Voice assistants have already been developed and are currently being used by millions of people around the world. The technology behind voice assistants is continuously evolving and improving, making it increasingly feasible. The technical feasibility of a voice assistant depends on several factors, including the hardware and software requirements, the availability of NLP algorithms, and the quality of speech recognition technology. To create a voice assistant,a microphone is needed to capture the user's voice. Speech recognition software is used to analyze and interpret the user's spoken words. NLP software helps to understand the user's intent and context by analyzing the language used in the speech. Text-to-speech software

is used to convert the response generated by the system into an audio output that can be heard by the user. AI and ML algorithms are used to improve the accuracy of the voice assistant's speech recognition and NLP capabilities over time. All of these components have been developed and are readily available. Therefore, it is technically feasible to create a voice assistant. However, the quality and performance of the voice assistant will depend on the quality of 10 these components and how they are integrated together.

3.3.3 Social Feasibility

The proposed system produces best results and gives high performance. It can be implemented easily. It deals with the compatibility of the project with the cultural environment. A virtual assistant is built in accordance with the general culture. This project is technically feasible with no external hardware requirements. also, it is simple in operate and does not cost training or repairs. Overall feasibility study of the project reveals that the goals of the proposed system is achievable. A decision is taken to proceed with the project.

3.4 System Specification

3.4.1 Hardware Specification

- Intel core i5
- Ram (minimum 4 GB)
- 10 GB of free hard disk OR SSD(recommended)

3.4.2 Software Specification

- Visual studio or pycharms
- Windows 7(32-bit) or above
- Python 3.2 or later
- Chrome Driver
- Selenium Web Automation

3.4.3 Standards and Policies

Sample attached

Anaconda Prompt

Anaconda prompt is a type of command line interface which explicitly deals with the ML(MachineLearning) modules.And navigator is available in all the Windows,Linux and MacOS.The anaconda prompt has many number of IDE's which make the coding easier. The UI can also be implemented in python.

Standard Used: ISO/IEC 27001

Jupyter

It's like an open source web application that allows us to share and create the documents which contains the live code, equations, visualizations and narrative text. It can be used for data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning.

Standard Used: ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 General Architecture

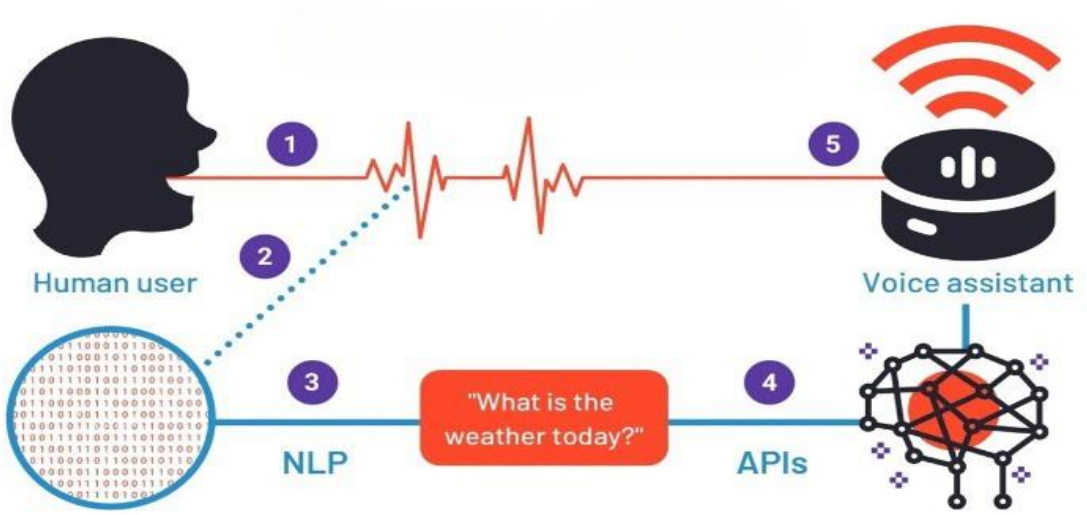


Figure 4.1: **General Architecture**

Figure 4.1 describes architecture follows a traditional partitioning of conversational systems, with separate components for speech recognition, natural language understanding, dialog management, natural language generation, and audio output, (audio files or text to speech). This architecture does not rule out combining some of these components in specific systems. This architecture aims at serving, among others, the following most popular highlevel use cases for IPAs

4.2 Design Phase

4.2.1 Data Flow Diagram of Voice based Virtual Assistant

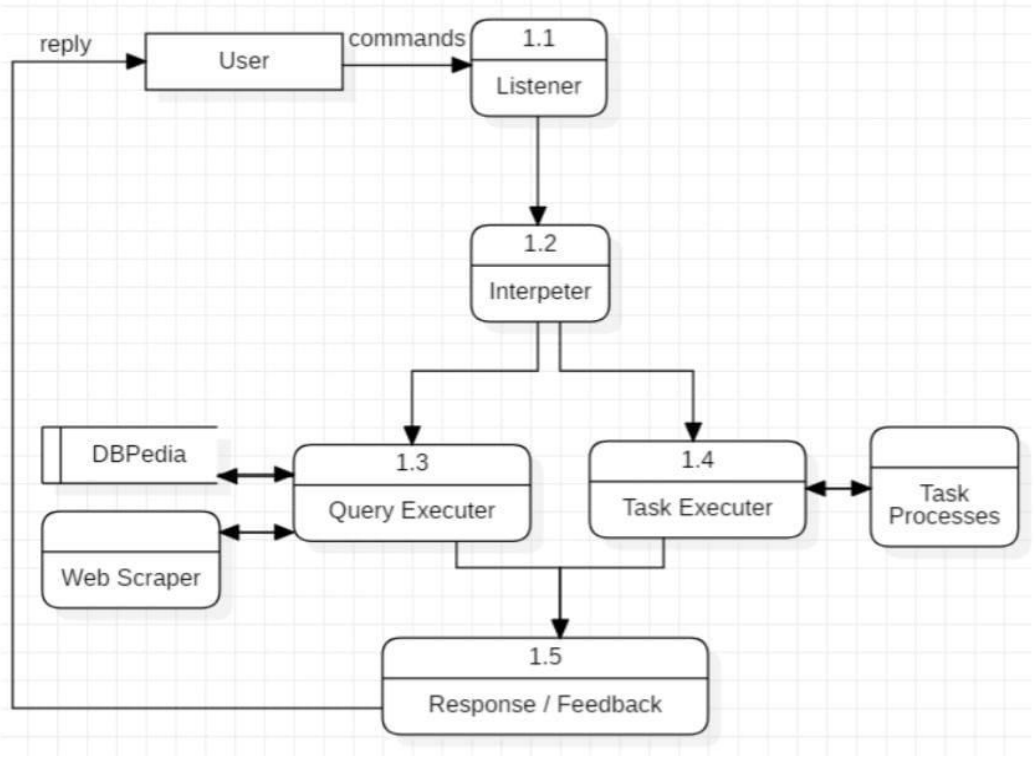


Figure 4.2: **Data Flow Diagram**

Figure 4.2, explains User's Voice Input: This is the initial input provided by the user through their voice.Audio Stream: This is the audio data captured from the user's voice input and processed by the speech-to-text engine. Speech-to-Text Conversion Engine: This engine converts the audio stream into text data that can be processed by the natural language understanding system.Text Data: This is the text data generated by the speech-to-text engine and used as input for the natural language understanding system. Natural Language Understanding System: This system analyzes the text data to extract relevant entities and determine the user's intent. Entities and Intent: These are the outputs of the natural language understanding system, which are used by the dialogue management system to determine the appropriate response. Dialogue Management System: This system uses the entities and intent provided by the natural language understanding system to determine the appropriate response to the user's input. Text-to-Speech Synthesis Engine: This engine converts the response generated by the dialogue management system into an audio stream that can be played back to the user.

4.2.2 Use Case Diagram of Voice based Virtual Assistant

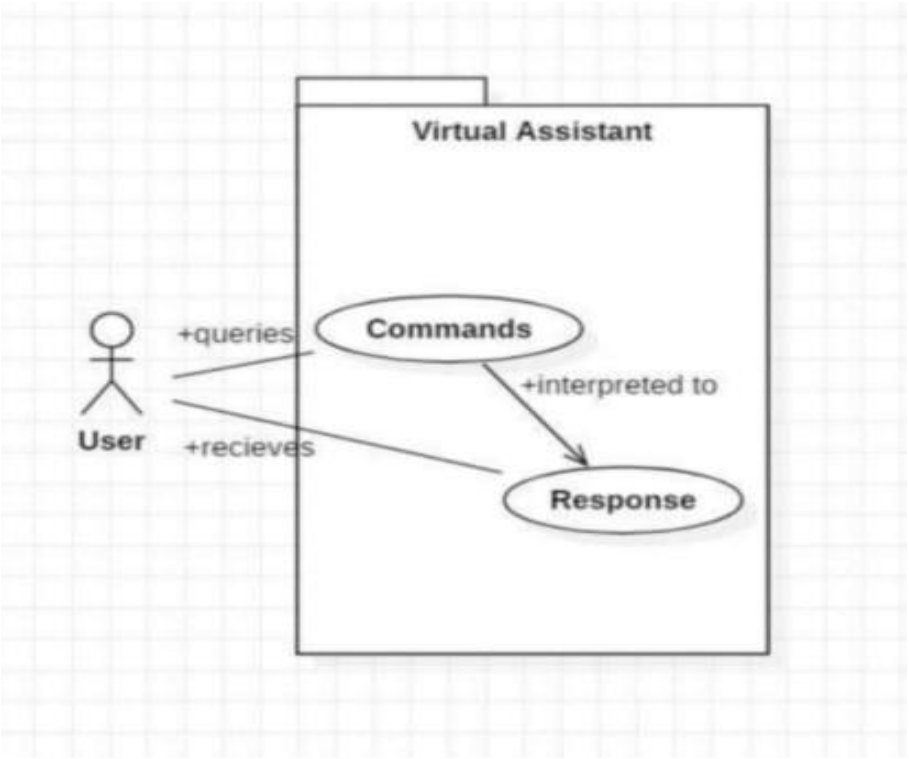


Figure 4.3: **Use Case Diagram**

Figure 4.3 describes about the use case diagram for a voice-based virtual assistant that illustrate different ways in which the user can interact with the system, the different tasks that the virtual assistant is capable of performing, and the various external services or devices that the virtual assistant can interact with to provide a seamless and person- alized experience for the user.

The use cases for a voice-based virtual assistant could include tasks such as setting reminders, scheduling appointments, making phone calls, sending messages, playing music, controlling smart home devices, and providing information on various topics. Each of these use cases would be represented by a separate box in the use case diagram, with arrows showing the flow of information and actions between the user and the virtual assistant.

4.2.3 Class Diagram of Voice based Virtual Assistant

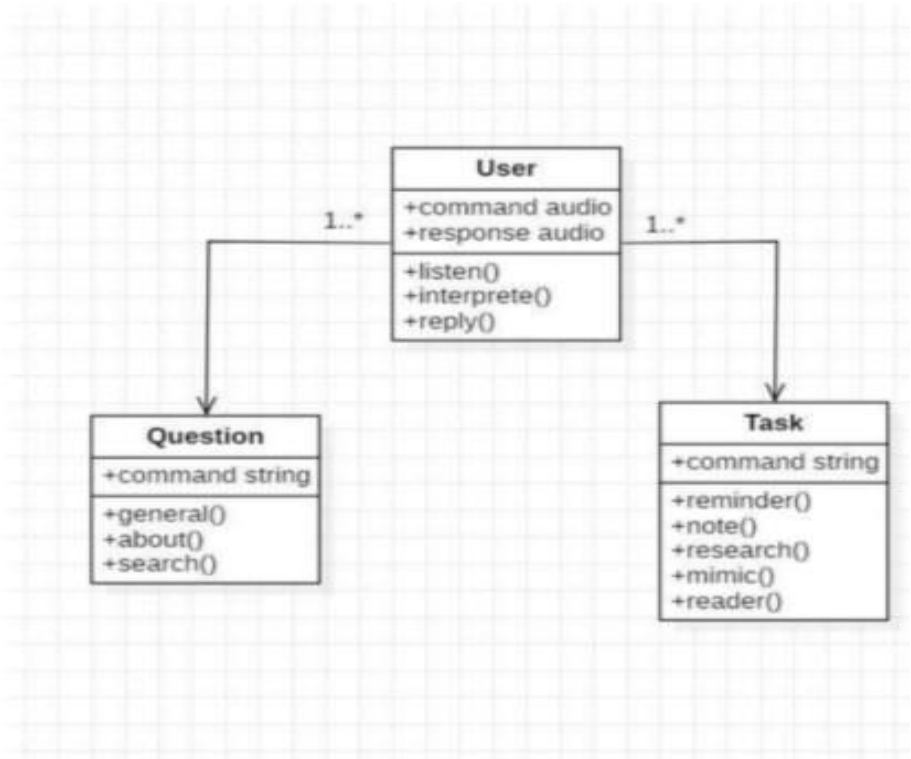


Figure 4.4: **Class Diagram**

Figure 4.4 explains about class user has 2 attributes command that it sends in audio and the response it receives which is also audio. It performs function to listen the user command. Interpret it and then reply or sends back response accordingly. Question class has the command in string form as it is interpreted by interpret class. It sends it to general or about or search function based on its identification. The task class also has interpreted command in string format. It has various functions like reminder, note, mimic, research and reader.

4.2.4 Sequence Diagram of Voice based Virtual Assistant

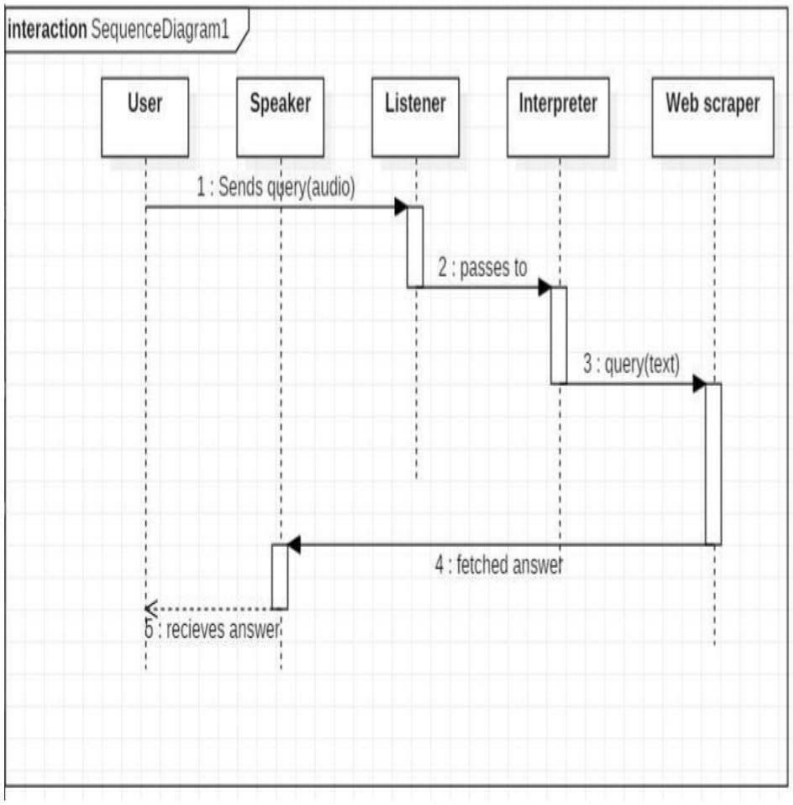


Figure 4.5: **Sequence Diagram**

Figure 4.5 describes the sequence diagram that shows how an answer asked by the user is being fetched from internet. The audio query is interpreted and sent to Web scraper. The web scraper searches and finds the answer. It is then sent back to speaker, where it speaks the answer to user.

A sequence diagram for a voice-based virtual assistant system would typically illustrate the flow of events that occur when a user interacts with the system. It would show the various components of the system and how they interact with each other to process user requests and generate responses.

4.2.5 Collaboration diagram of Voice based Virtual Assistant

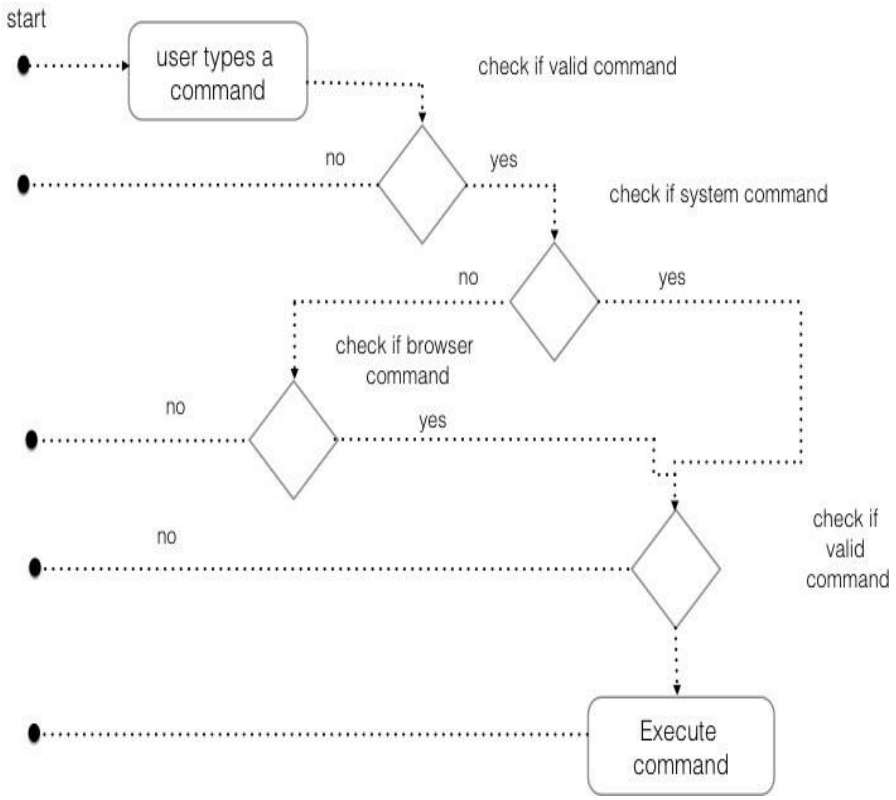


Figure 4.6: **Collaboration diagram**

Figure 4.6 is about collaboration diagram for a voice-based virtual assistant would typically show the interactions between the different components or modules involved in the system. These components may include the following:

User Interface: The diagram would depict the interaction between the user interface, which could be a mobile app, a web interface, or a voice-enabled device, and the virtual assistant. This could include the user providing voice commands or input through the interface, such as asking questions, giving commands, or making requests.

Natural Language Processing (NLP) Module: The diagram would illustrate the interaction between the speech recognition module and the NLP module. The NLP module would process the text generated by the speech recognition module to analyze and understand the meaning of the user's input. It may involve tasks such as

identifying intent, extracting entities, and understanding context.

4.2.6 Activity Diagram of Voice based Virtual Assistant

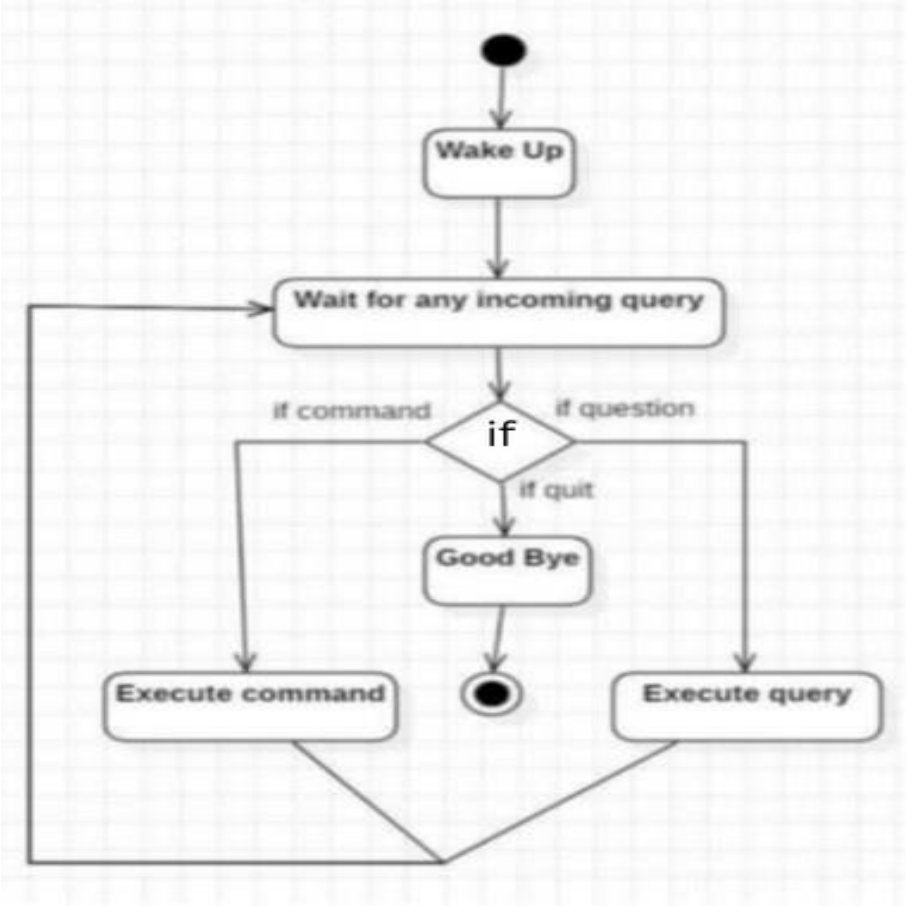


Figure 4.7: **Activity Diagram**

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm

1. Installing Libraries

i) SpeechRecognition pip install SpeechRecognition SpeechRecognition is a Python speech recognition library that is used to convert our human speech into text.

ii) PyAudio SpeechRecognition library doesn't work alone, it uses PyAudio Library. PyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library. Using PyAudio, you can easily use Python to play and record audio on a variety of platforms

2. Speech Recognition We start by importing speech recognition, we do not import the Pyaudio library but speech recognition uses it for conversion purposes. The speech recognition module is used to create a Recognizer() object which takes audio data as input captured by another Microphone() object. This is then passed to the recognize google() function for actual speech recognition to text. We print this to verify if our setup is working properly as expected.

3. Activating Personal Assistant We have to name our personal assistant, so here we add a logic to recognize if the speech contains the word or not by using a simple if condition. If the speech matches the name we given then we just print it. We will put more logic in the upcoming sections.

4. Adding Speech for Assistant In the above, we converted speech to text for our assistant to understand what we say but we also want it to reply back to us. So for this, we have to add text to speech capability to the assistant. We use Pyttsx3 library text to speech conversion, using it as an engine that answers us back or reads the output of our question. Here we initialize pysttsx3 after the listener and we test it by making it read some of the sample text.

5. Making AI Assistant Take “Play Song” Command We have come this far by establishing to and fro communication with Max but how about making him do something for example asking him to play a song.

6. Adding DateTime and Wikipedia access to AI Assistant We will now add more capabilities into our Max such as telling time or date and having the information about anything from Wikipedia. For this, we use Python built-in library datetime and external library wikipedia.

7. Adding personal Information about Assistant Our assistant is ready to do some task which we assign We extract the song name from the voice command and print it to test if it is working properly. In the next section, we will add code to actually make Max play the song from the internet.

4.3.2 Pseudo Code

```
class AIAssistant:
    def process_command(self, voice_input):
        # Use speech-to-text to convert the voice input into text
        text_input = speech_to_text(voice_input)

        # Use natural language processing to determine the user's intent
        intent = nlp(text_input)

        # Use the intent to determine the appropriate action to take
        if intent == "open application":
            application_name = extract_application_name(text_input)
            open_application(application_name)
            return f"Opening {application_name} now."
        elif intent == "search":
            query = extract_search_query(text_input)
            search_results = perform_search(query)
            return f"Here are the search results for {query}: {search_results}."
        else:
            return "I'm sorry, I don't understand. Can you please try again?"
```

4.4 Module Description

4.4.1 User Commands

Considering the USER to speak the commands to virtual assistant to do the necessary needed task's.

4.4.2 Developing Python Code

Developing Python code for the virtual assistant.

4.5 Steps to execute/run/implement the project

4.5.1 Installing Python Modules

Install the relevant modules and libraries.

4.5.2 Acccepting Commands

The Audio input/accepting verbal commands function.

4.5.3 Responding to Input Commands

The Audio input/accepting and responding to verbal commands function.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

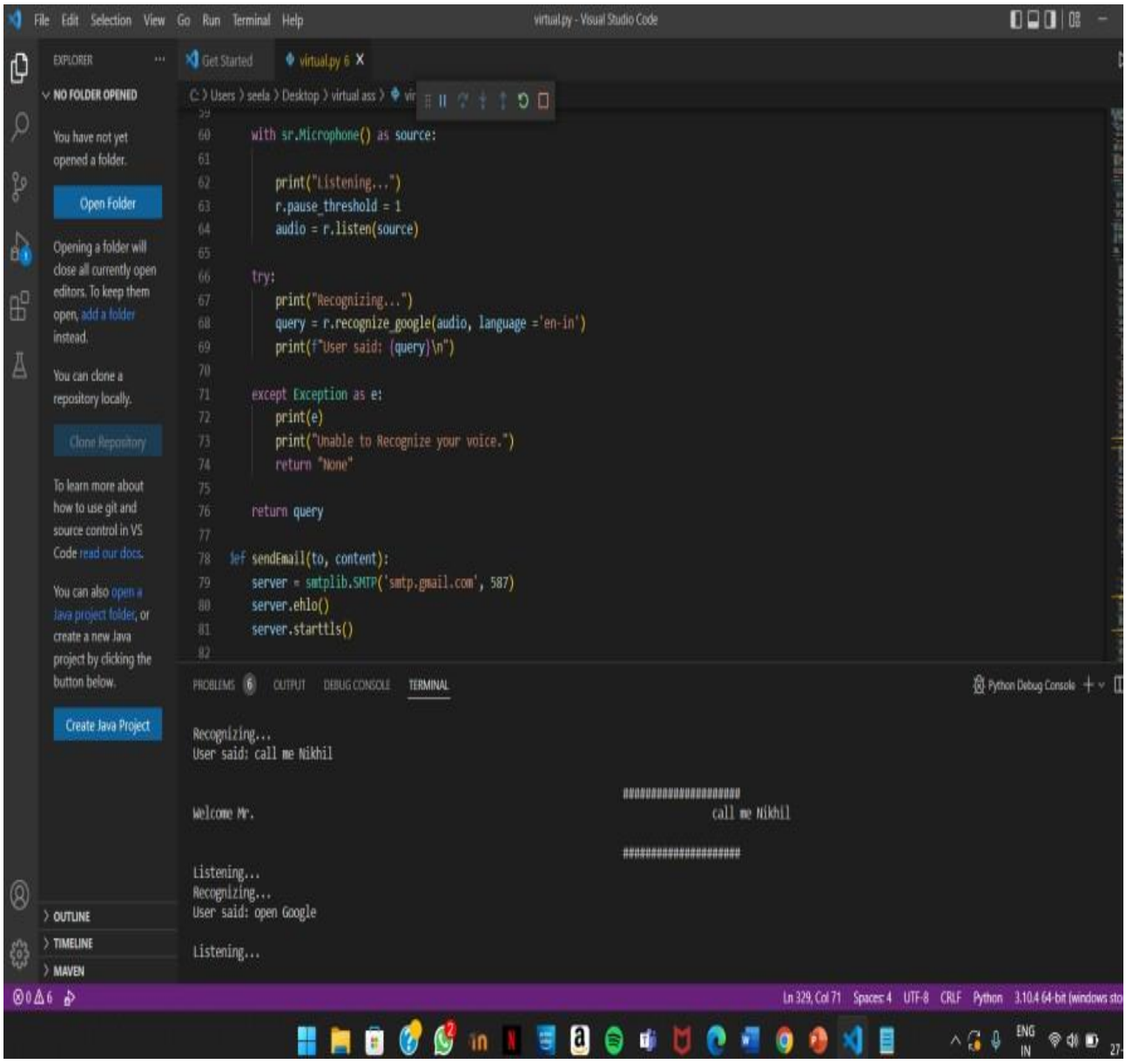
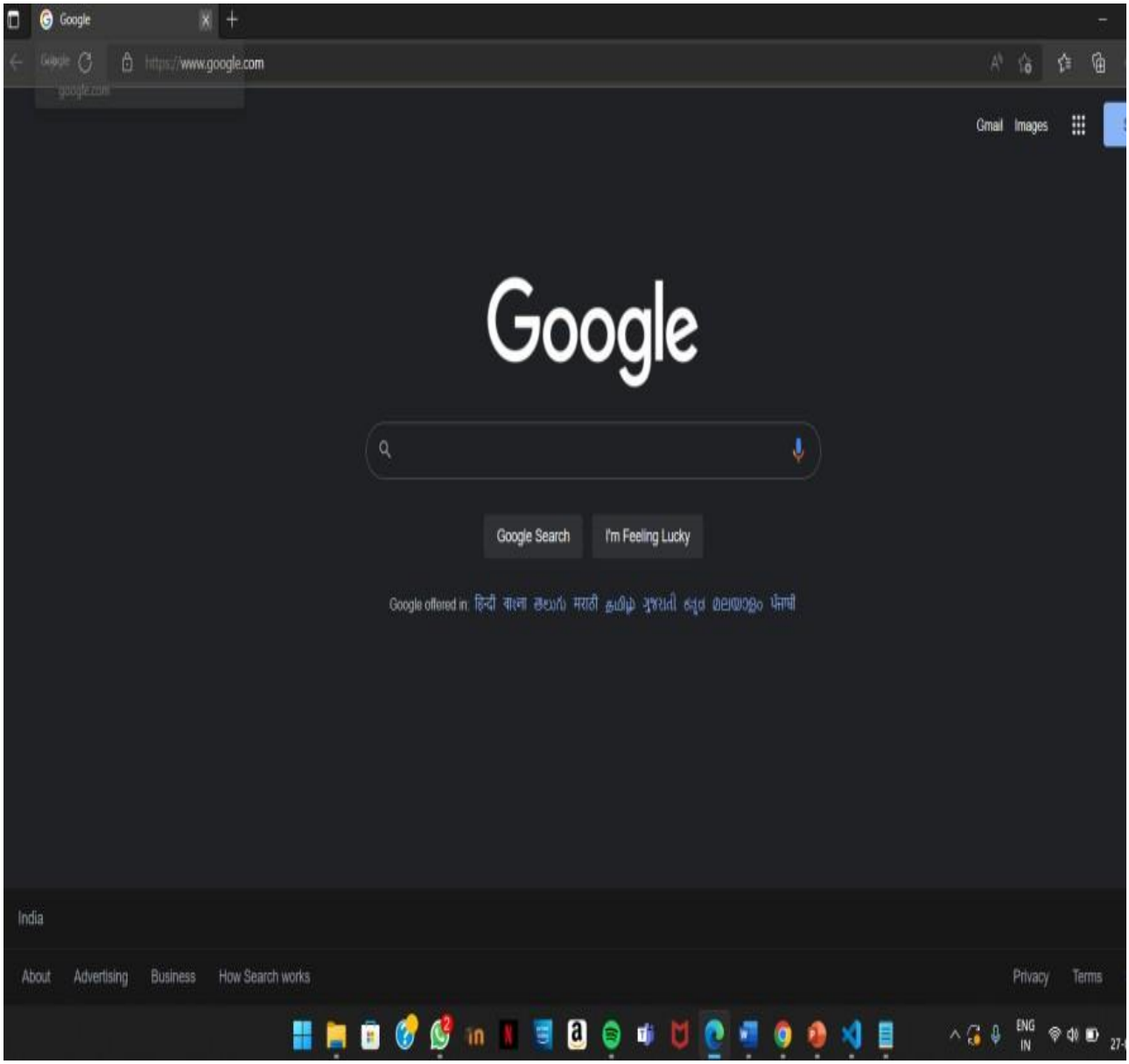


Figure 5.1: **Program Code Image**

5.1.2 Output Design



In the above fig,. it explains about ”open Google” command which is a specific voice command that can be given to a voice-based virtual assistant that uses AI. When the user says ”open Google,” the virtual assistant understands the command and triggers an action to open the Google search engine on a device, such as a computer or a smartphone.

To accomplish this, the virtual assistant uses natural language processing (NLP) algorithms to interpret the user’s voice command and match it to the appropriate action. The NLP algorithms analyze the user’s voice input, recognize the specific words and phrases used, and determine the user’s intent.

Once the virtual assistant has determined the user's intent to open Google, it can then execute the appropriate action to open the Google search engine. This action can be triggered using application programming interfaces (APIs) that allow the virtual assistant to communicate with the device and launch the Google search engine.

5.2 Testing

5.3 Types of Testing

5.3.1 Unit Testing

Input

```
engine = pyttsx3.init('sapi 5')
voices = engine.get Property ('voices')
engine.set Property ('voice ,voices' [ 1 ].id)
```

Test result

5.3.2 Integration Testing

Input

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

Test result

5.3.3 System Testing

Input

```
if 'wikipedia' in query :
speak ('Searching Wikipedia ...')
query = query.replace('wikipedia')
results = wikipedia.summary ( query , sentences=3)
speak ('According to Wikipedia')
```

```
print (results)
```

5.3.4 Test Result

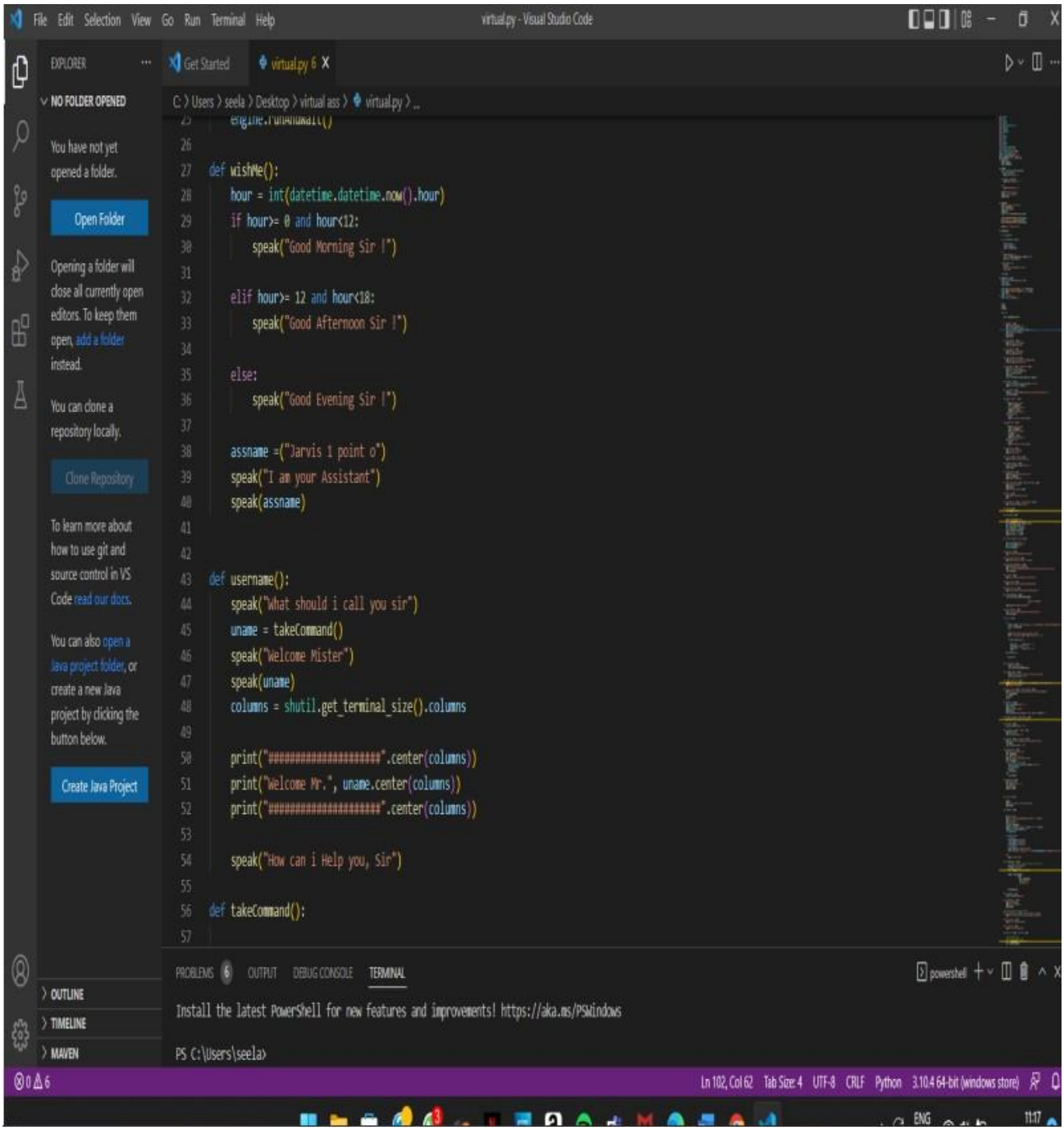


Figure 5.2: **Code Execution Image**

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

Voice-Controlled Devices uses Natural Language Processing to process the language spoken by the human and understand the query and process the query and respond to the human with the result. It is designed to minimize the human efforts and control the device with just human Voice. The device can also be designed to interact with other intelligent voice-controlled devices like IoT applications and devices, weather reports of a city from the Internet, send an email to a client, add events on the calendar, etc.

6.2 Comparison of Existing and Proposed System

Existing system:(Decision tree)

The term virtual assistant, or virtual personal assistant, is also commonly used to describe contract workers who work from home doing administrative tasks typically performed by executive assistants or secretaries. Virtual assistants are typically cloud-based programs that require internet-connected devices and/or applications to work. Three such applications are Siri on Apple devices, Cortana on Microsoft Devices and Google Assistant on Android devices. There are also devices dedicated to providing virtual assistance. The most popular ones are available from Amazon, Google and Microsoft. To use the Amazon Echo virtual assistant, called Alexa, users call out the wake word, ”Alexa.” A light on the device signals to the user it is ready to receive a command, which typically involves simple language requests, such as ”what is the weather today,” or ”play pop music.” Those requests are processed and stored in Amazon’s cloud. The technologies that power virtual assistants require massive amounts of data, which feeds artificial intelligence (AI) platforms, including machine learning, natural language processing and speech recognition platforms.

6.3 Sample Code

```
import speech_recognition as sr
import pyttsx3
import datetime
import os

def listen():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        audio = r.listen(source)
        try:
            query = r.recognize_google(audio)
            print(f"You said: {query}")
        except:
            print("Sorry, could not understand your command.")
            query = ""
        return query.lower()

engine = pyttsx3.init()

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

def assistant():
    speak("Hi, how can I help you?")
    while True:
        command = listen()
        if "hello" in command:
            speak("Hi there!")
        elif "open notepad" in command:
            os.system("notepad")
        elif "what time is it" in command:
            current_time = datetime.datetime.now().strftime("%H:%M")
            speak(f"The current time is {current_time}.")
        elif "exit" in command:
            speak("Goodbye!")
            break
        else:
            speak("Sorry, I don't understand.")

assistant()
```

Output

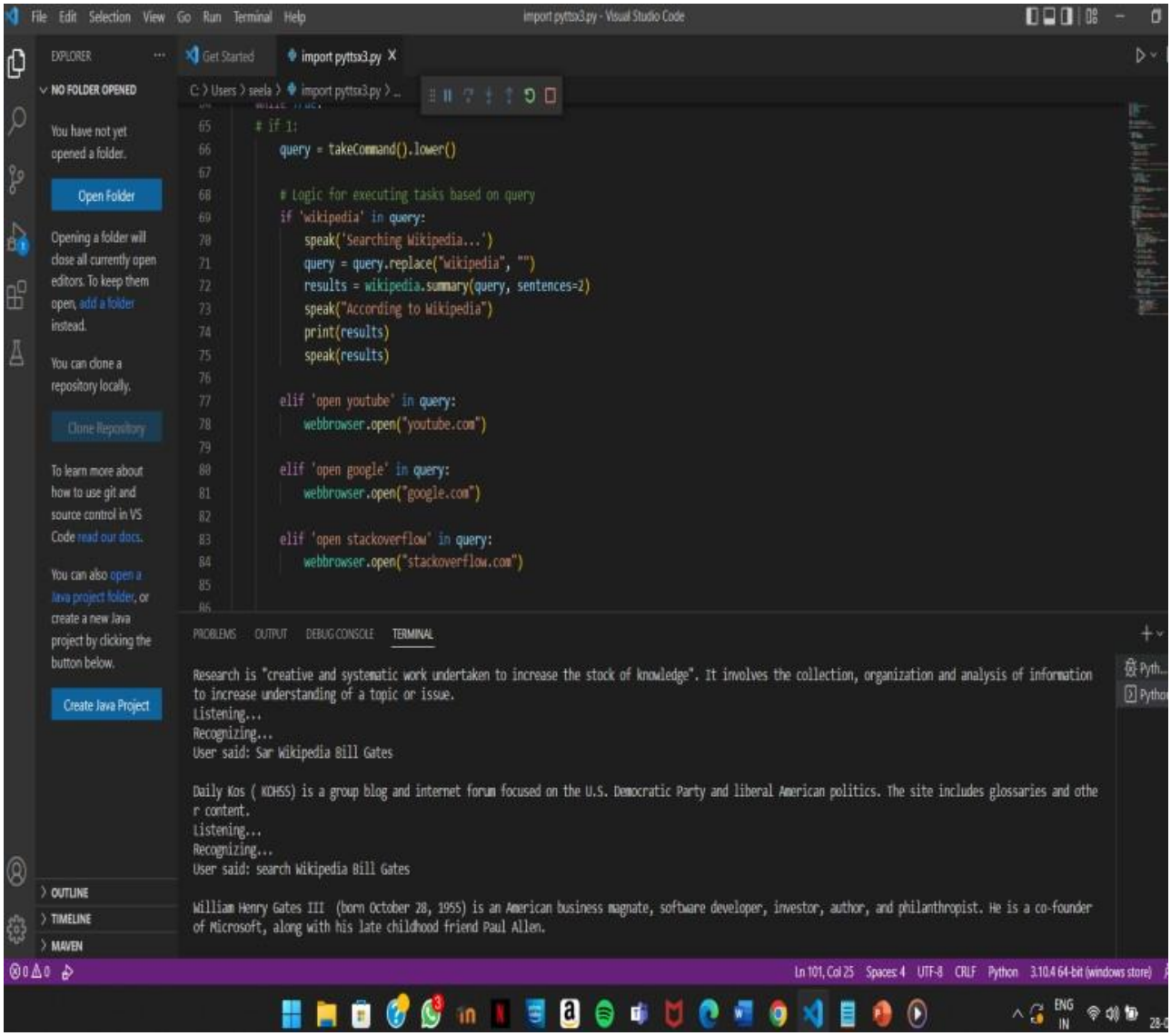


Figure 6.1: **Command for wikipedia output**

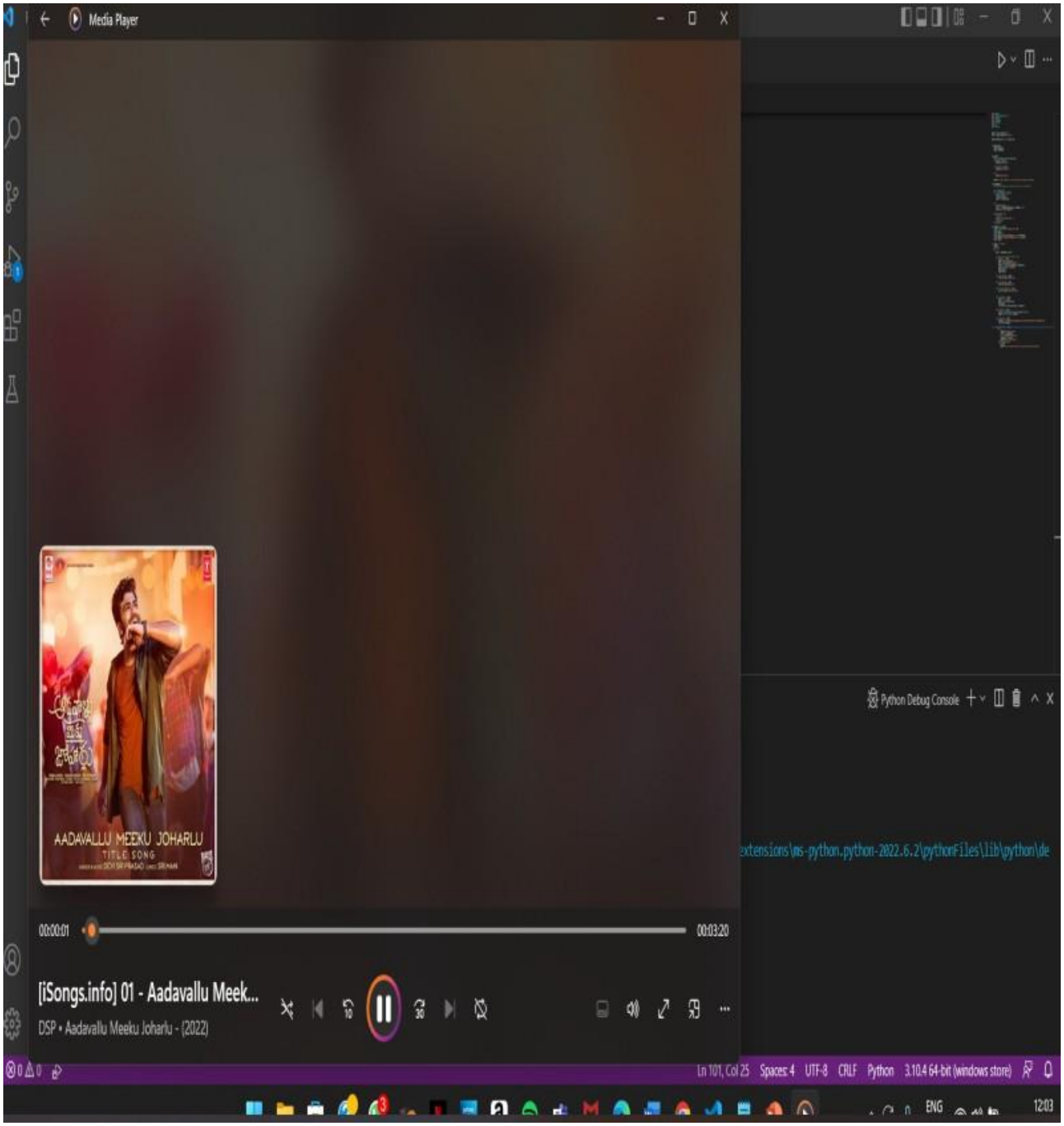


Figure 6.2: **Command used to -PLAY MUSIC**

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

A Voice-based Virtual Assistant for Windows using AI has the potential to be a highly effective tool for enhancing productivity and convenience. AI-powered virtual assistants can understand natural language and respond to voice commands, allowing users to perform tasks without the need for manual input. With the advancements in AI technology, virtual assistants can now also learn from user behavior and preferences to offer more personalized recommendations and suggestions. However, as with any technology, there are also potential privacy and security concerns that must be addressed. Overall, a voice-based virtual assistant for Windows using AI has the potential to greatly enhance the user experience and improve productivity, but it is important to implement proper security measures to ensure the protection of user data.

7.2 Future Enhancements

It will further develop the work process, consumer loyalty, and deals and develop your suggestions to your customers.Artificial intelligence-fueled Virtual Assistants can likewise make customized Emails - email automation for every one of your customers and take notes of significant central issues during a gathering. Meanwhile, voice assistants are special programs that can recognize humans via speech recognition and natural language processing technologies, “translate” words into machine commands, then perform the desired tasks. They are also getting smarter with time - the more you interact with a voice assistant, the better it can understand your preferences and match its actions to your expectations.

Chapter 8

PLAGIARISM REPORT

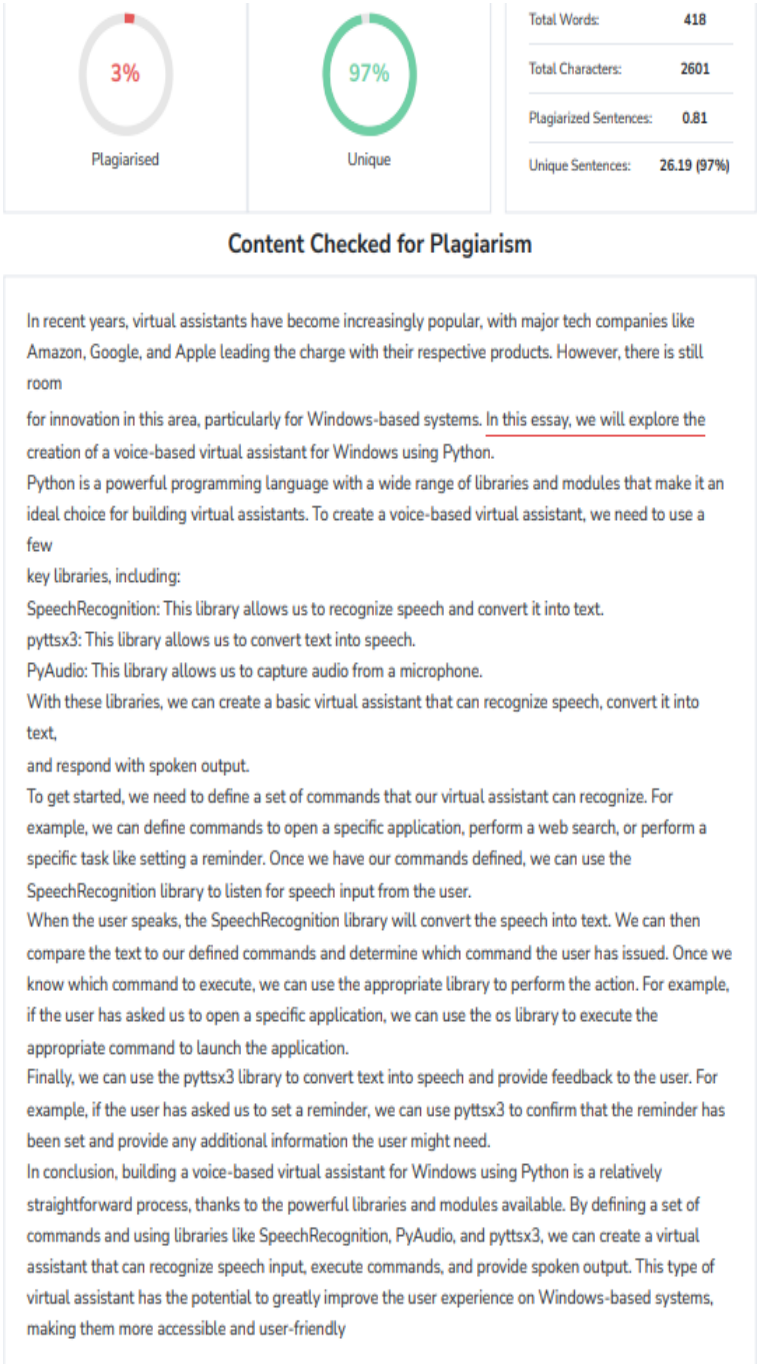


Figure 8.1: **Plagiarism Report**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
import pyttsx3 # pip install pyttsx3
import speech_recognition as sr # pip install  speechRecognition
import datetime
import wikipedia # pip install wikipedia
import webbrowser
import os
import smtplib

# set up text to speech engine
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)

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

def greet():
    hour = int(datetime.datetime.now().hour)
    if hour >=0 and hour <12:
        speak("Good Morning!")
    elif hour >=12 and hour <18:
        speak("Good Afternoon!")
    else:
        speak("Good Evening!")
    speak("I am your virtual assistant. How may I assist you?")

def take_command():
    # use speech recognition to take command
    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-in')
        print(f"User said: {query}\n")

    except Exception as e:
        print("I'm sorry, I didn't catch that. Can you please repeat?")
        return "None"
    return query

def send_email(to, subject, body):
    # set up email server and credentials
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('your_email_address', 'your_email_password')
    # compose email and send
    message = f'Subject: {subject}\n\n{body}'
    server.sendmail('your_email_address', to, message)
    server.close()

if __name__ == "__main__":
    greet()
    while True:
        query = take_command().lower()
        if 'wikipedia' in query:
            speak('Searching Wikipedia...')
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=2)
            speak("According to Wikipedia")
            print(results)
            speak(results)

        elif 'open youtube' in query:
            speak('Opening Youtube...')
            webbrowser.open("youtube.com")

        elif 'open google' in query:
            speak('Opening Google...')
            webbrowser.open("google.com")

        elif 'open stackoverflow' in query:
            speak('Opening StackOverflow...')
            webbrowser.open("stackoverflow.com")

        elif 'play music' in query:
            music_dir = 'C:\\Users\\YourUsername\\Music\\MusicFolderName'
            songs = os.listdir(music_dir)
            print(songs)
            os.startfile(os.path.join(music_dir, songs[0]))
```

```
86
87          elif 'what time is it' in query:
88              strTime = datetime.datetime.now().strftime("%H:%M:%S")
89              speak(f"Sir, the time is {strTime}")
90
91          elif 'send email' in query:
92              try:
93                  speak("What should I say?")
94                  body = take_command()
95                  to = "receiver_email_address"
96                  subject = "Subject of the email"
97                  send_email(to, subject, body)
98                  speak("Email has been sent!")
99              except Exception as e:
100                 print(e)
101                 speak("Sorry, I was not able to send the email at this moment.")
102
103         elif 'exit' in query:
104             speak("Goodbye!")
105             exit()
```

9.2 Poster Presentation

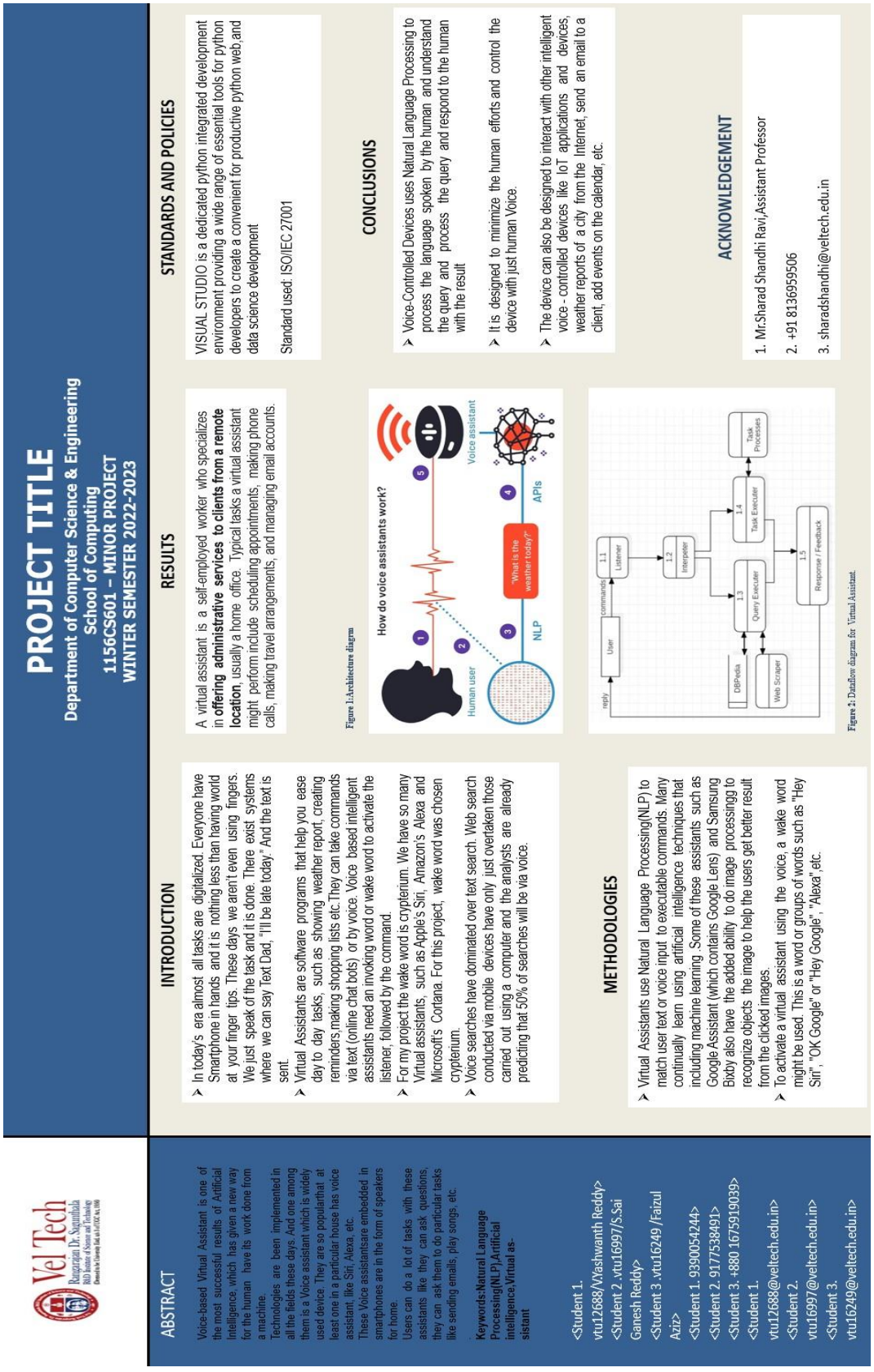


Figure 9.1: **Poster presentation**

References

[1] Douglas O,shaughnessy, senior member, IEEE, “Interacting With Computers by Voice: Automatic Speech Recognition and Synthesis” proceedings of THE IEEE, VOL. 91, NO. 9, SEPTEMBER 2019

[2] F. M. Sultan, A. Alhajj and M. Alhajj, ”Intelligent Personal Assistant: A Review of Voice-based Virtual Assistant for Windows,” 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Seoul, Korea (South), 2020.

[3] S.K. Bhatia and S. K. Khanna, ”Development of a Voice-Enabled Chatbot Using Google Dialogflow,” 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020.

[4] M. Shahzad and M. C. Patel, ”Voice Recognition Based Intelligent Personal Assistant for Windows,” 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2019.

[5] T. H. Le and H. V. Nguyen, ”Voice Recognition-Based Personal Assistant System for Windows Using Raspberry Pi,” 2019 IEEE International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019.

[6] V. J. Pillai and V. Sudha, ”Smart Personal Assistant (SPA): An Innovative Application for Windows,” 2018 International Conference on Information and Communication Technology for Intelligent Systems (ICTIS), Ahmedabad, India, 2018.

[7] S. J. Patel and M. C. Patel, ”Voice Recognition Based Intelligent Personal Assistant for Windows,” 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2019.

[8] S. Chen and G. Gupta, ”Jarvis: A Voice-Based Personal Assistant Using Machine Learning,” 2018 9th International Conference on Computing, Commu- nication and Networking Technologies (ICCCNT), Bangalore, India, 2020.

[9] S. T. Shivam ”An Artificial Intelligence Based Voice Recognition Personal Assistant System for Windows,” 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 2019.

[10] R. J. Piyush and P. Kumar, ”Voice-Based Virtual Assistant Using Artificial Intelligence and IoT for Home Automation,” 2020 5th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimavaram, India, 2020.

[11] S. Das, S. Sengupta and S. Chatterjee, ”Braina: An AI-Based Virtual Assistant for Windows,” 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2019

[12] S. T. Haque, M. H. Khan and S. M. Islam, ”Development of a Voice Recognition-Based Personal Assistant for Windows Operating System,” 2021 International Conference on Electrical, Computer and Communication Engineering (ECCE), Chattogram, Bangladesh, 2021.

[13] P. C. Nwosu and I. U. Ogah ”Design and Implementation of a Windows-based Voice Assistant using Natural Language Processing and Artificial Intelligence” in the Proceedings of the 2nd International Conference on Computer Science and Information Technology,in Dubai, UAE, 2019.

[14] R. K. Sharma and M. Sharma, ”A Comparative Study of Voice Assistants for Windows using Artificial Intelligence” in the Proceedings of the 2nd International Conference on Computing and Communication Systems,in Gurgaon, India, 2020.

[15] S. Kumar and S. Mittal, ”Development of a Personalized Virtual Assistant for Windows using AI” in the Proceedings of the International Conference on Computing, Communication and Security, in Greater Noida, India, 2020.

General Instructions

- Cover Page should be printed as per the color template and the next page also should be printed in color as per the template
- **Wherever Figures applicable in Report , that page should be printed in color**
- Dont include general content , write more technical content
- Each chapter should minimum contain 3 pages
- Draw the notation of diagrams properly
- Every paragraph should be started with one tab space
- Literature review should be properly cited and described with content related to project
- All the diagrams should be properly described and dont include general information of any diagram
- Example Use case diagram - describe according to your project flow
- All diagrams,figures should be numbered according to the chapter number
- Test cases should be written with test input and test output
- All the references should be cited in the report
- **Strictly dont change font style or font size of the template, and dont customize the latex code of report**
- **Report should be prepared according to the template only**
- **Any deviations from the report template,will be summarily rejected**
- **Number of Project Soft Binded copy for each and every batch is (n+4) copies as given in the table below**
- **Attach the CD in last Cover page of the Project Report with CD cover and details of batch like Title,Members name and VTU No ,Batch No should be written in Marker pen**
- For **Standards and Policies** refer the below link https://law.resource.org/pub/in/manifest.in.html
- Plagiarism should be less than 15%

General Instructions

Docs1.jpg