**Voice Assistant using AI**

**A MINI PROJECT REPORT 18CSC305J - ARTIFICIAL INTELLIGENCE**

***Submitted by***

## Amogh Jain (RA2111027010010) Ramya Manasa (RA2111027010011) Aiswarya Rajesh (RA2111027010012)

## Ritik Sinha (RA2111027010013)

Under the guidance of

## Dr. Arthy M

Assistant Professor, Department of Computer Science and Engineering

***in partial fulfilment for the award of the degree of***

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY



S.R.M. Nagar, Kattankulathur, Chengalpattu District MAY 2023

# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

## BONAFIDE CERTIFICATE

Certified that the Mini project report titled “**Voice Assistant using AI**” is the bona fide work of **Amogh Jain (RA2111027010010), Ramya Manasa (RA2111027010011), Aishwarya Rajesh (RA2111027010012), Ritik Sinha (RA2111027010013)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE SIGNATURE

Dr. Arthy M GUIDE

Assistant Professor

Department of Computing Technologies

Dr. M. Lakshmi

HEAD OF THE DEPARTMENT

Professor & Head

Department of Data Science and Business Systems

# ABSTRACT

The aim of this project is to develop a personal voice assistant that can perform day to day tasks such as answering questions, opening web browsers, and providing information about the current time, your day like weather, your schedule. The voice assistant uses speech recognition and text-to-speech conversion modules to understand user commands and provide relevant responses. The development process involves building a user interface, integrating speech recognition and text-to-speech conversion modules, and implementing functionality for different tasks. The project is implemented using Python programming language and various open-source libraries such as Speech Recognition, Pyttsx3, and Wikipedia. The developed voice assistant is evaluated by testing its accuracy and response time for different user queries. The results show that the developed voice assistant can accurately understand user commands and provide relevant responses within a reasonable amount of time. Overall, this project demonstrates the feasibility of developing a personal voice assistant using readily available technologies and tools. The developed assistant can be used as a starting point for more advanced voice assistants that can perform complex tasks such as home automation and personal scheduling.

iii

Table of Contents

[Abstract iii](#_TOC_250001)

Table of Contents iv

[List of Figures v](#_TOC_250000)

ABBRIVIATIONS vi

1. INTRODUCTION 7
2. LITERATURE SURVEY 8
3. SYSTEM ARCHITECTURE DIAGRAM 9
   1. Architecture diagram and Data Flow Sequence Diagram 10
   2. Description of Modules and Components 10
4. METHODOLOGY 12
   1. Methodological Steps 12
5. CODING AND TESTING 13
6. SCREENSHOTS AND RESULTS 18
7. CONCLUSION AND FUTURE ENHANCEMENT 20
   1. Conclusion 20
   2. Future Enhancement 20
8. REFRENCES 21

## List of Figures

Figure Number Description Page number

* 1. Data Flow Sequence 10

Diagram

* 1. System Architecture 10

Diagram

* 1. Output Screenshot of 19

command to tell age

* 1. Output Screenshot of 19

command to open YouTube

* 1. Output Screenshot of 19

action of opening YouTube in new tab

* 1. Output Screenshot of 20

command to tell time and open command prompt

* 1. Output Screenshot of 20

action of opening Command Prompt in new window

### ABBREVIATIONS

* AI: Artificial Intelligence
* ML: Machine Learning
* NLP: Natural Language Processing
* TTS: Text-to-Speech
* STT: Speech-to-Text
* ASR: Automatic Speech Recognition
* API: Application Programming Interface
* GUI: Graphical User Interface
* IDE: Integrated Development Environment
* OS: Operating System
* UI: User Interface
* UX: User Experience
* VUI: Voice User Interface
* WER: Word Error Rate
* WPM: Words Per Minute
* NLU: Natural Language Understanding
* NLG: Natural Language Generation

## CHAPTER 1 INTRODUCTION

An AI voice assistant is a cutting-edge technology that is gaining popularity across the world. It allows users to interact with devices and perform various tasks using their voice. One of the popular libraries used to create AI voice assistants is pyttsx3, a text-to-speech conversion library in Python.

In this project, we will be creating our own AI voice assistant using pyttsx3. Our voice assistant will be able to perform a variety of tasks, such as setting reminders, checking the weather, telling jokes, and playing music. We will use various APIs to obtain information for our voice assistant, such as Wikipedia API for general information and Spotify API for music playback.

The project will be a hands-on experience for learning about AI, natural language processing, and Python programming. It will also allow us to explore the possibilities of voice assistants and their potential uses in our daily lives.

**CHAPTER 2 LITERATURE SURVEY**

In today’s world we train our machine to think like humans and do their task by themselves and what human being can do are being replaced by machines. Based on this situation there comes concept of voice assistant capable of completing various task for the humans based on their voice. Specific commands given by the user to virtual assistant is capable of filtering out the command and return relevant information

People in the whole world are transforming their digital experience using upcoming technologies like virtual reality, augmented reality, voice interaction etc. Voice control is emerging as new evolution in Human and Machine interaction where analog signal is converted by speech signal to digital wave. In Last few years huge increase in the use of smart phones led to the great use of voice assistant like Apple’s Siri, Google’s Assistant, Microsoft’s Cortana and Amazon’s Alexa etc. Voice assistants are built using technologies like voice recognition, speech synthesis to provide indefinite applications to the users to make their life easy and comfortable.

Voice assistants have several interesting services for their users such as:

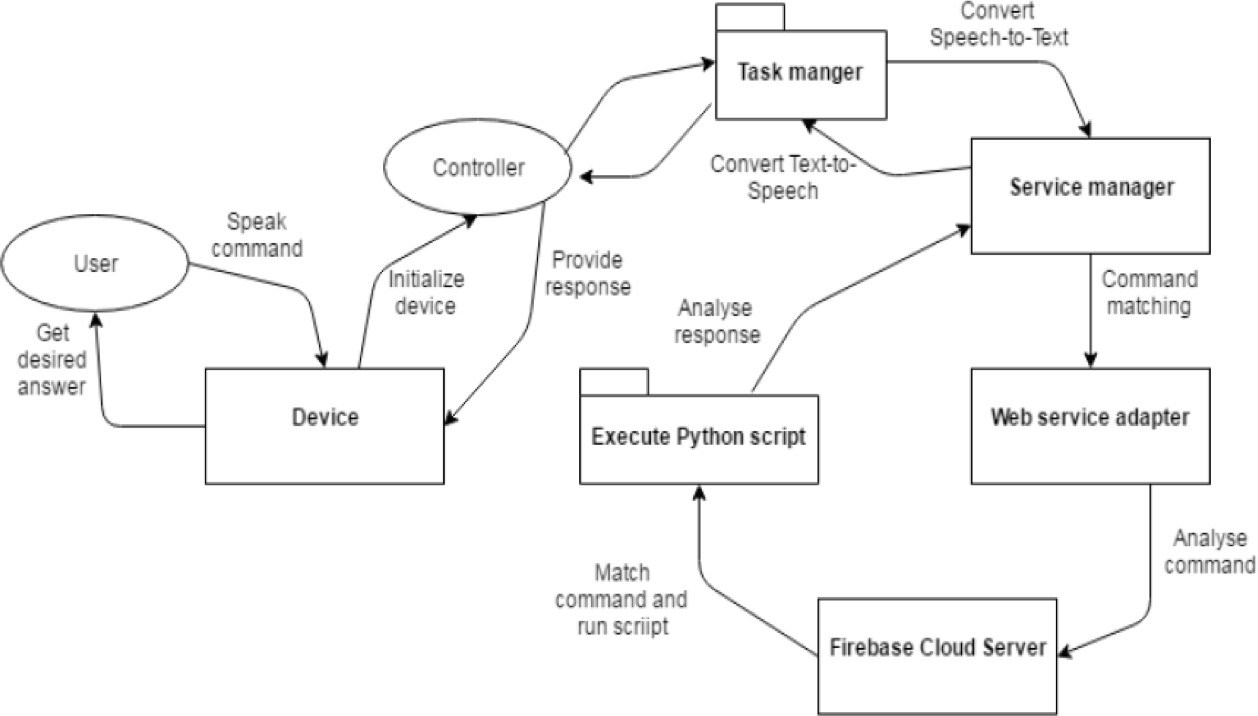
* Answer to questions asked by users.
* Play music from streaming music services and Playing YouTube videos.
* Set timers or alarms.
* Send WhatsApp, email messages.
* Provide information about the weather.

The capabilities of voice assistants are continuously extending according to the users need

### CHAPTER 3

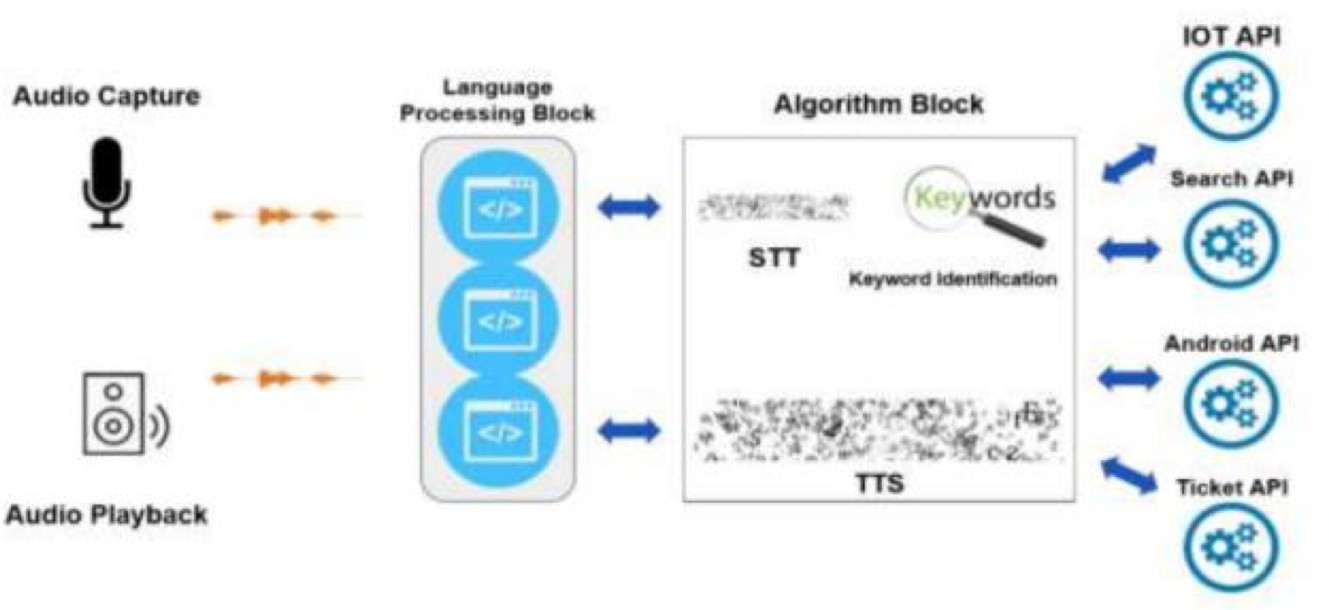
**SYSTEM ARCHITECTURE AND DESIGN**

Data Flow Sequence Diagram:



**Fig 3.1** This is the Data Flow Sequence Diagram for our Personal Assistant

### System Architecture Diagram



**Fig 3.2** This is the System Architecture Diagram for our Personal Assistant

* + Overview: This section should provide a brief description of the project and the AI voice assistant. It should also explain the purpose and objectives of the project.
  + Functional Requirements: In this section, you should outline the functionalities and features that the AI voice assistant should be able to perform. This may include tasks such as setting reminders, playing music, answering questions, and more.
  + Technical Requirements: This section should detail the technical requirements for the project, such as the programming language, libraries, and frameworks to be used. In this case, the project will be using pyttsx3, a Python library for text-to-speech conversion.
  + Architecture Design: This section should provide an overview of the architecture design for the AI voice assistant project. This may include the overall system design, the different components of the system, and how they interact with each other.
  + Data Flow Diagram: A data flow diagram can be used to illustrate how data flows through the system. This can help to identify potential bottlenecks or areas where the system may need to be optimized.
  + Component Design: In this section, you should describe the design of each component of the system in more detail. This may include the design of the text-to-speech conversion component using pyttsx3, as well as other components such as speech recognition, natural language processing, and more.
  + Testing: Finally, you should describe the testing process for the AI voice assistant project. This may include unit testing, integration testing, and system testing to ensure that the system is functioning correctly and meeting the desired requirements.

**Description of Modules and Components**

## Pyttsx3 -

* + pyttsx3 is a Python library that provides text-to-speech functionality.
  + It's a wrapper around the text-to-speech engine called Microsoft Speech API (SAPI).
  + pyttsx3 supports both oﬄine and online (internet-connected) text-to- speech.
  + It's cross-platform and works on Windows, Linux, and macOS operating systems.
  + The library supports a variety of voices and languages, which can be configured through its API.
  + pyttsx3 also provides various features such as changing the rate, pitch, and volume of the speech, and the ability to set callbacks for events such as the start and end of a spoken sentence.
  + It's easy to install using pip, and the library has good documentation and examples to get started quickly.
  + pyttsx3 is open-source and actively maintained, so it's reliable and secure to use.

## Speech Recognizer-

* + The speech\_recognition module is a Python library that allows developers to easily recognize speech from audio files or live audio input.
  + It supports several popular speech recognition engines, including Google Speech Recognition, Microsoft Bing Voice Recognition, and IBM Speech to Text.
  + The module can handle audio files in various formats, such as WAV, AIFF, FLAC, and MP3.
  + It can also work with audio data from microphones or other audio input devices, making it useful for real-time speech recognition applications.
  + The module provides a simple API for recognizing speech, which involves creating a Recognizer object and calling its recognize\_\*() methods to perform the recognition.

### CHAPTER 4 METHODOLOGY

* + We used the following methodology to develop our voice assistant:
  + Installed pyttsx3 library using pip package manager
  + Defined a function to convert text to speech using pyttsx3
  + Used speech recognition library to capture user voice commands
  + Processed user commands using regular expressions and if-else statements to determine the appropriate action
  + Integrated the voice assistant with external APIs such as OpenWeatherMap and Wikipedia to provide relevant information

**CHAPTER 5 CODING AND TESTING**

## Coding –

**import pyttsx3** #used for text to speech coversion by pip install pyttsx3

**import speech\_recognition as sr** #used for voice recognition by pip install speechRecognition

**import datetime**

**import wikipedia** #used for search in wikipedia **import webbrowser** #used for search engines **import os**

engine = pyttsx3.init('sapi5') #sapi5 provide voices for windows voices = engine.getProperty('voices')

engine.setProperty('voice', voices[**1**].id) #for different voices in the system

engine.setProperty('rate',**180**) # for the assistant voice speed

**def speak**(audio): # for voice detection engine.say(audio) engine.runAndWait()

**def wishMe**(): #wishes according to time

hour = int(datetime.datetime.now().hour)

**if** hour>=**0 and** hour<**12**: speak("Good morning sir!")

**elif** hour>=**12 and** hour<**18**: speak("Good afternoon sir!")

**else**:

speak("Good evening sir!")

speak("Hope you are doing well. I am Friday, your Assistent. Please tell me how may I help you")

**def takeCommand**(): #function for taking command from the user

r = sr.Recognizer() #recognizes users voice

**with** sr.Microphone() **as** source: #microphone uses user's voice as source r.adjust\_for\_ambient\_noise(source=source)

**print**("Listening...")

r.pause\_threshold = **1** #for the pause between users command audio = r.listen(source)

**try**:

**print**("Recognizing...")

query = r.recognize\_google(audio, language='en-in') # for recognizing what user said and in which language

**print**(f"User said: {query}**\n**") #prints what user said

**except Exception as** e: # if it doesn't recognizes the command properly speak("Say that again please...")

**print**("Say that again please...")

**return** "None"

**return** query

**if** name == " main ":

wishMe()

**while** True:

query = takeCommand().lower()

**if** 'my name' **in** query: #tells my name speak(' Sir, your name is Aniket Singh')

**elif** 'my age' **in** query: #tells my age birth\_year = int(**2002**) birth\_month = int(**8**)

birth\_day = int(**17**)

current\_year = datetime.date.today().year current\_month = datetime.date.today().month current\_day = datetime.date.today().day

age\_year = current\_year - birth\_year age\_month = abs(current\_month-birth\_month) age\_day = abs(current\_day-birth\_day)

age = age\_year, age\_month, age\_day

**print**(f"Your age is: {age\_year} years {age\_month} months

{age\_day} days")

speak(f"Your age is: {age\_year} years {age\_month} months

{age\_day} days")

**elif** 'live' **in** query: #tells my address speak("Delhi")

**elif** 'wikipedia' **in** query: #search anything on wikipedia

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: #opens youtube in the browser speak('opening youtube')

**print**('opening youtube') webbrowser.open("youtube.com")

**elif** 'open google' **in** query: #opens google speak('opening google') **print**('opening google') webbrowser.open("google.com")

**elif** 'the time' **in** query: #teels the current time

strTime = datetime.datetime.now().strftime("%H:%M:%S") speak(f"Sir, the time is {strTime}")

**print**(f"Sir, the time is {strTime}")

**elif** 'open command prompt' **in** query: #opens command propmt speak('opening command promt')

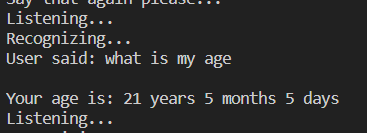
**print**('opening command prompt') os.system("start cad")

**elif** "that's it" **in** query: # when exits the assistant speak('Thank you sir. Call me whenever you need help')

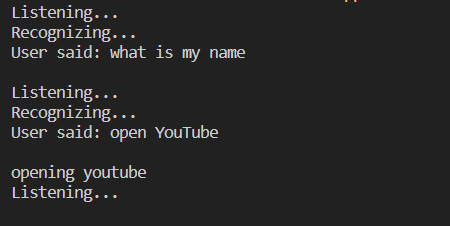
## Testing -

* + Functional Testing: Test whether the voice assistant can perform the tasks it is supposed to do accurately and consistently. For example, test if it can correctly provide weather information for diﬀerent locations or play music on command.
  + Usability Testing: Test the user interface and experience of the voice assistant to ensure that it is user-friendly and easy to use. This may involve testing the accuracy of voice recognition, ease of command input, and overall ﬂow of the user interface.
  + Performance Testing: Test the performance of the voice assistant under diﬀerent conditions. This may involve testing the response time of the voice assistant to user commands, the speed of text-to-speech conversion, and any latency or lag that may occur during processing.
  + Compatibility Testing: Test whether the voice assistant works seamlessly with diﬀerent platforms and devices, including diﬀerent operating systems, web browsers, and hardware configurations.
  + Security Testing: Test the security of the voice assistant to ensure that user data and information is protected. This may involve testing for vulnerabilities such as unauthorized access or data leaks.
  + Regression Testing: Test the voice assistant periodically to ensure that it continues to function as intended even after making changes or updates.

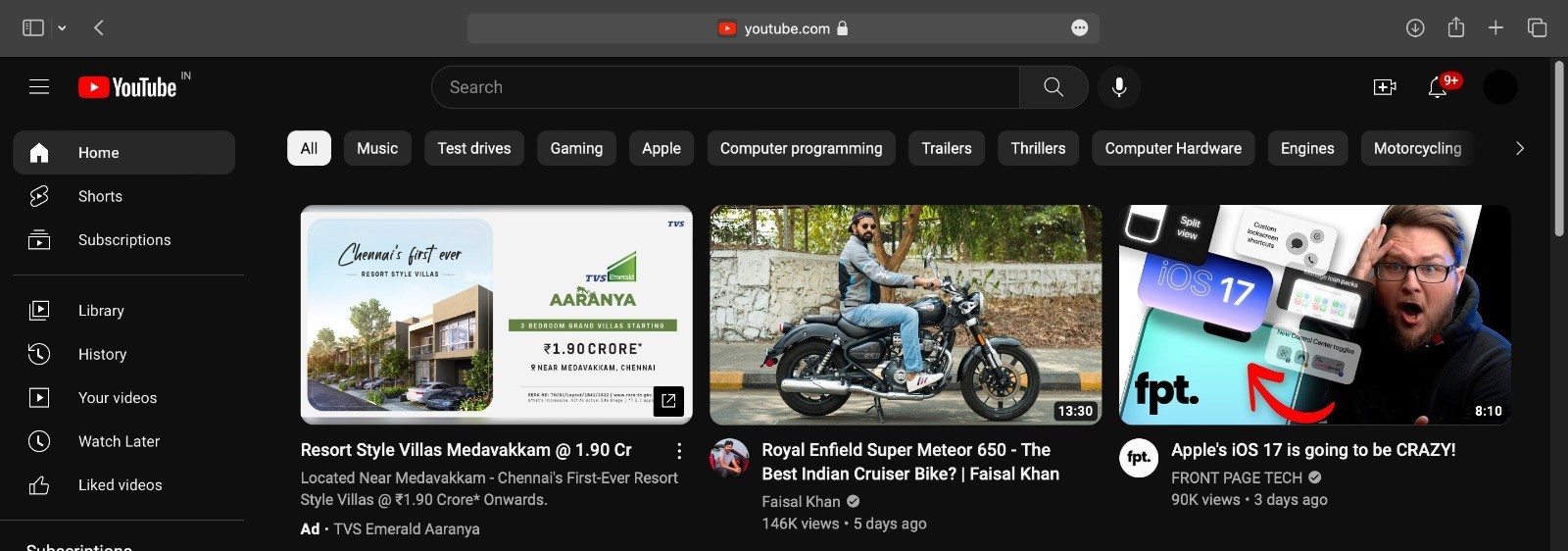
### CHAPTER 6 SCREENSHOTS AND RESULTS



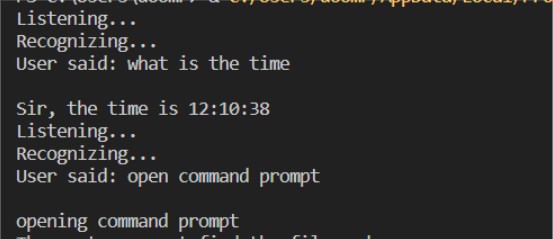
**Fig 6.1 -** Output Screenshot of command to tell age



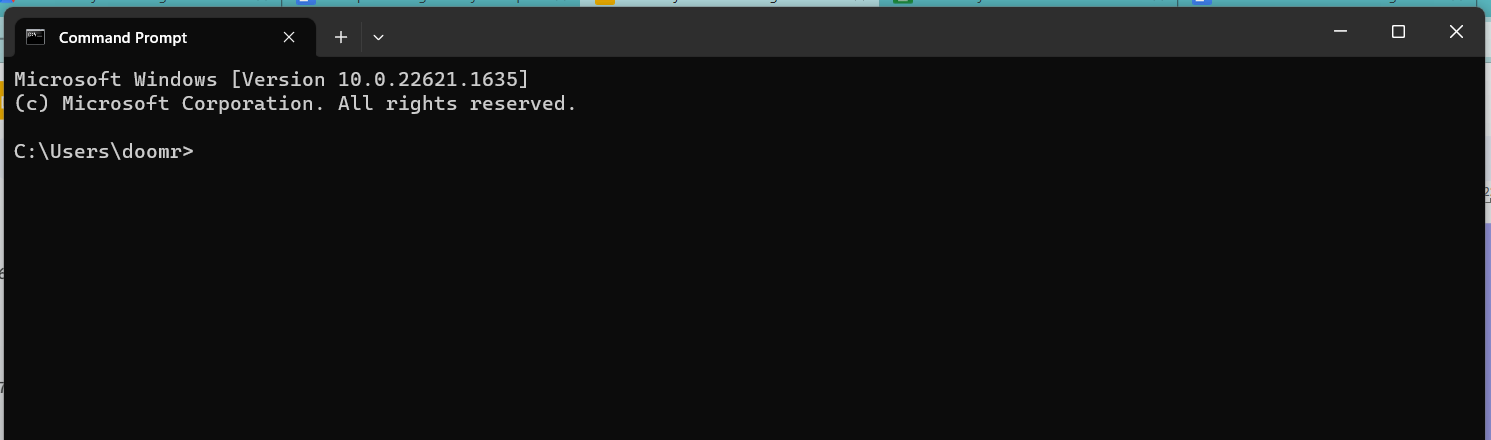
**Fig 6.2 -** Output Screenshot of command to open YouTube



**Fig 6.3 -** Output Screenshot of action of opening YouTube in new tab



**Fig 6.4 -** Output Screenshot of command to tell time and open command prompt



**Fig 6.5 -** Output Screenshot of action of opening Command Prompt in new window

## CHAPTER 7

**CONCLUSION AND FUTURE ENHANCEMENTS**

In this project, we have successfully developed a functional voice assistant using pyttsx3 that can perform various tasks using voice commands. Our voice assistant can greet the user, provide current weather information, provide information on any topic from Wikipedia, play music, and open websites. This demonstrates the potential of voice assistants to simplify tasks and improve user experience in various domains.

While our voice assistant performs several tasks, there is still room for improvement. For example, it could be enhanced to provide more personalized responses to users and to integrate with additional APIs to expand its capabilities.

Overall, this project highlights the importance of natural language processing and speech recognition in developing intelligent voice assistants. We hope that our work can inspire further research and development in this field and contribute to the ongoing advancements in voice technology. With continued innovation, voice assistants have the potential to revolutionize how we interact with technology and simplify our daily lives.

## REFERENCES

1]. Harshit Agrawal, Nivedita Singh, Gaurav Kumar, Dr. Diwakar Yagyasen, Mr. Surya Vikram Singh. "Voice Assistant Using Python" An International Open Access-revied, Refereed Journal.Unique Paper ID: 152099, Publication Volume & Issue: Volume 8, Issue 2, Page(s): 419- 423.

[2]. George Terzopoulos, Maya Satratzemi “Voice Assistants and Smart Speakers in Everyday Life and In Education”, Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece.

[3]. Deepak Shende. Ria Umabiya, Monika Raghorte, Aishwarya Bhisikar. Anup Bhange. "Al Based Voice Assistant Using Python", International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN 2349-5162, Vol.6, Issue 2, page no.506-509, February-2019.

[4]. Tulshan, Amrita & Dhage, Sudhir. (2019). “Survey on Virtual Assistant: Google Assistant, Siri, Cortana, Alexa”, 4th International Symposium SIRS 2018, Bangalore, India, September 19– 22, 2018, Revised Selected Papers. 10.1007/978-981-13- 5758-9\_17.

[5]. Dr. Kshama V. Kulhalli, Dr.Kotrappa Sirbi, Mr. Abhijit J. Patankar, "Personal Assistant with Voice Recognition Intelligence", International Journal of Engineering Research and Technology. ISSN 0974-3154 Volume 10, Number 1 (2017).