

Virtual Voice Assistant

A PROJECT REPORT

Submitted by

Abhishek Pawar: 202101102008

Amey Gaikwad: 202101103185

For the subject AI LAB

of

Third Year

IN

COMPUTER SCIENCE & ENGINEERING

Guided by

Mr. Nitin Suryawanshi

Ms. Priya Borade



Department of Computer Science and Engineering
**MGM's Jawaharlal Nehru Engineering College, Ch.
Sambhajinagar**

YEAR 2023- 2024

CERTIFICATE

This is to certify that the project report

“Virtual voice assistant”

Submitted by

Abhishek Pawar: 202101102008

Amey Gaikwad: 202101103185

is a bonafide work carried out by them under the supervision of Mr. Nitin Suryawanshi and Ms. Priya Borade it is approved for the subject Artificial intelligence in academic year 2023-2024 Part-II Semester VI at JNEC, MGM University, Ch. Sambhajinagar.

Date:

Mr. Nitin Suryawanshi
Ms. Priya Borade
Guide
Dept. of Computer Sci.

& Engineering

Dr. Deepa Deshpande
Head of Department
Dept. of Computer Sci.

& Engineering

Dr. H. H. Shinde
Principal

MGM's Jawaharlal Nehru Engineering College, Ch. Sambhajinagar

Table of Contents

1. Introduction	1
1.1 Purpose	
1.2 Scope	
1.3 Definitions, Acronyms and Abbreviations	
1.4 References	
1.5 Overview	
2. Overall Description	3
2.1 Product Perspective	
2.2 Product Functions	
2.3 Constraints	
3. Functional Requirements	4
4. Non Functional Requirements	5
5. Design	6
5.1 Level 0 Data Flow Diagram	
5.2 Level 1 Data Flow Diagram	
5.3 Sequence Diagram	
5.4 Use case Diagram	
5.5 Class Diagram	
5.6 Activity Diagram	
6. Implementation	10
7. Conclusion	17

1. Introduction

1.1 Purpose

This SRS describes the requirements and specifications of Voice Assistant, a voicecontrolled personal assistant. It explains the functional features of the Personal Assistant, along with interface details, user constraints and related considerations. This SRS is targeted towards the team of developers involved in the development of Voice Assistant as a first step in the developmental process. The intended audience also includes users of Voice Assistant.

1.2 Scope

Voice Interaction: The software shall support voice input for initiating commands, queries, and interactions.

Natural Language Understanding: It shall include natural language processing capabilities to interpret user requests and intents accurately.

Task Execution: The assistant should be able to perform various tasks such as setting reminders, sending messages, making calls, retrieving information, etc.

Integration: The software should integrate with external services and APIs to provide relevant responses and execute actions.

Personalization: It shall support user preferences and personalization options for tailoring the assistant's responses and behavior.

1.3 Definitions, Acronyms and Abbreviations

- VVA: Virtual Voice Assistant
- NLP: Natural Language Processing
- SR: Speech Recognition
- TE: Task Execution
- NL: Plain English
- SRS: Software Requirement Specification
- DFD: Data Flow Diagram
- IP: Internet Protocol
- UI: User Interface
- API: Application Programming Interface
- HTTP: Hypertext Transfer Protocol

- REST: Representational State Transfer

1.4 References

- [1] Jamie Alexandre, David Marx, Stephan Auerhahn, John Sabath, Arijit Basu, Wes Brown, Jacob Kelley, Frankwin Faber and Chris Allen. SDK for interacting with stability.ai APIs (e.g.stable diffusion inference). Dec 01, 2022. (<https://github.com/Stability-AI/stability-sdk>)

- [2] Anonymous.SendingEmailsviaGmailwithPython.Script Reference,Sep23,2020. (<https://scriptreference.com/sending-emails-via-gmail-with-python/>)

- [3] Amila Viraj. How To Build Your Own Chat Bot Using Deep Learning. Towards Data Science,Nov1,2020.(<https://towardsdatascience.com/how-to-build-your-own-chatbot-using-deep-learning-bb41f970e281>)

1.5 Overview

Voice Assistant is a virtual assistant that uses speech recognition, NLP and speech synthesis to take actions to help its users. These assistants have evolved quickly and can perform several complex tasks today. It has become synonymous with Google Home and Amazon Echo. This perception cannot be more wrong. Underlying technology powering both these smart speakers are the actual voice assistants, Voice and Amazon Alexa, respectively. Today voice assistants are not limited to smart speakers but are also available in cars, household devices, Smartphone's, and several apps.

2. Overall Description

2.1 Product Perspective

From a product perspective, think of the Virtual Voice Assistant like your helpful friend in your device. It's designed to make your life easier by understanding and talking to you. Here's how:

- Voice Command Responsiveness
- Task Automation
- User Customization
- Privacy Protection

2.2 Product Functions

- The user is able to carry out some basic operations related to processes using voice commands in NL.
- The user can send emails, so that you can stay in touch with friends and colleagues with the help of voice commands in NL.
- The user see the current weather report of any city you specify, or, if you don't specify a city, it will use your IP address to give you the weather for your current location.
- The user will be able to play music by using the voice command in NL.
- VVA saves your chat history, so that you can refer it later.

2.3 Constraints

- The words being used in User speech should be restricted to the Vocabulary defined in the application. This vocabulary will at least contain the words of the English Dictionary.
- Ambient noise should be minimum when the user is trying to input a voice command.
- The user sets the system to passive state, if there is ambient noise and conversations going on which can be picked up by the audio input device.
- The user sets the system to active state before issuing voice commands.
- The user limits his speech to be Understandable. We define Understandable to include at least the set of all sentences correct under the English Grammar.
- The platform on which the system is running should at least have access to an audio input device such as a microphone for the most basic functionality.

3. Functional Requirements

Natural Language Processing (NLP) Libraries:NLP is a critical technology that enables machines to understand and interpret human language. Python provides various NLP libraries such as NLTK, Spacy, and TextBlob. These libraries use machine learning algorithms to perform various NLP tasks such as sentiment analysis, part-of-speech tagging, and more.

Speech Recognition Library: Speech recognition is a critical component of a voice assistant AI. It is the ability to convert spoken words into text. Python provides various speech recognition libraries such as SpeechRecognition, PocketSphinx, and Google Cloud Speech API. These libraries use machine learning algorithms to recognize speech and convert it into text.

Text-to-Speech Library:Text-to-speech is another essential component of a voice assistant AI. It is the ability to convert text into spoken words. Python provides various text-to-speech libraries such as pyttsx3, gTTS, and Amazon Polly. These libraries use machine learning algorithms to generate human-like speech from text.

Database Management System:A database management system is essential for storing and retrieving data in a voice assistant AI application. Python provides various database management systems such as MySQL, PostgreSQL, and SQLite. These systems offer various features such as data indexing, transaction management, and more.

OpenWeatherMap API:OpenWeatherMap is provides weather data through an API. The API provides global weather data, including current weather, forecasts, nowcasts, and historical weather data.

News API: News API is a simple HTTP REST API for searching and retrieving live articles from all over the web.

Wolframalpha API: Wolfram|Alpha is an answer engine that uses algorithms, knowledgebase, and Artificial Intelligence (AI) technology to calculate expert-level answers.

4. Non Functional Requirements

- **Security and safety:** Voice assistant safety and security involve data privacy, authentication, encryption for secure communication, protection against cyber threats, auditing, user education, ethical considerations, and regulatory compliance. These measures ensure user trust and protection from unauthorized access or data breaches.
- **Effectiveness and Efficiency:** Efficiency and effectiveness are crucial non-functional requirements for voice assistants. Efficiency refers to the ability of the assistant to perform tasks quickly and with minimal resources, such as processing time and memory usage. Effectiveness relates to the assistant's accuracy in understanding user queries and providing relevant responses or actions.
- **Maintenance services:** Maintenance services for a voice assistant encompass regular updates, bug fixes, performance optimizations, and server maintenance. These ensure the voice assistant remains reliable, efficient, and compatible with evolving technologies and user needs.
- **Good UI and User-friendly:** A voice assistant project with a good UI and user-friendly design is crucial as a non-functional requirement. It involves intuitive navigation, clear voice prompts, visually appealing interfaces, and seamless integration with other devices or platforms. Prioritizing user experience enhances engagement, satisfaction, and overall usability of the voice assistant, leading to greater adoption and success of the project.
- **Adaptability:** Voice assistant adaptability refers to its ability to integrate across devices, platforms, and environments seamlessly, supporting diverse languages, accents, and use cases while maintaining consistent performance and user experience.
- **Light on resources:** Light on resources means the voice assistant should operate efficiently without using excessive memory, processing power, or energy, ensuring smooth performance on various devices without draining battery life or causing slowdowns.

5. Design

5.1 Level 0 Data Flow Diagram

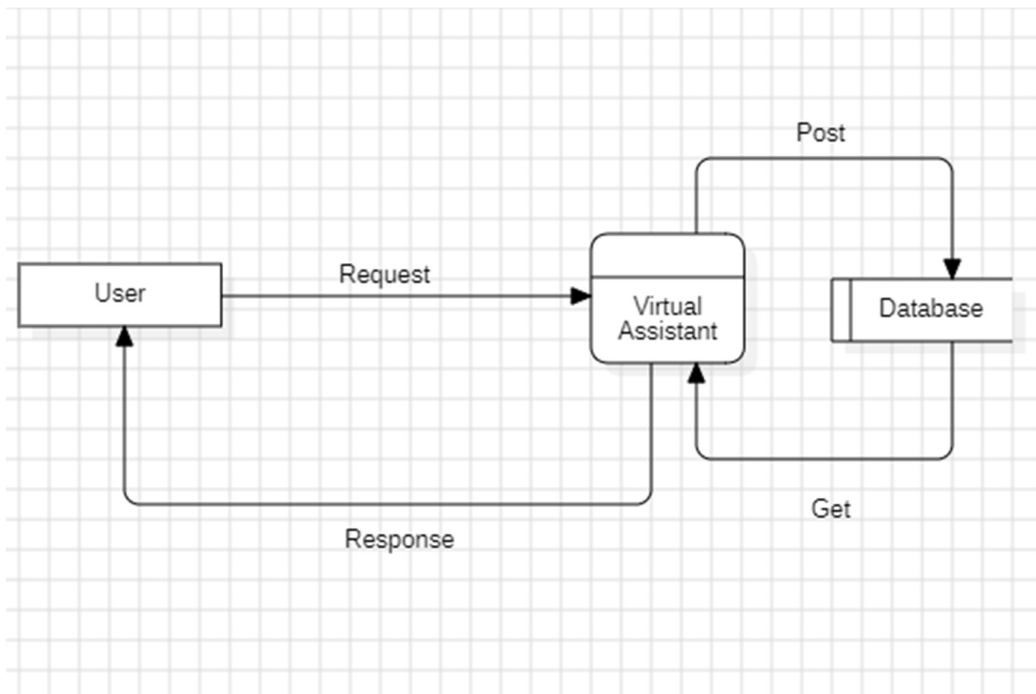


Fig5.1: Level 0 Data Flow Diagram

5.2 Level 1 Data Flow Diagram

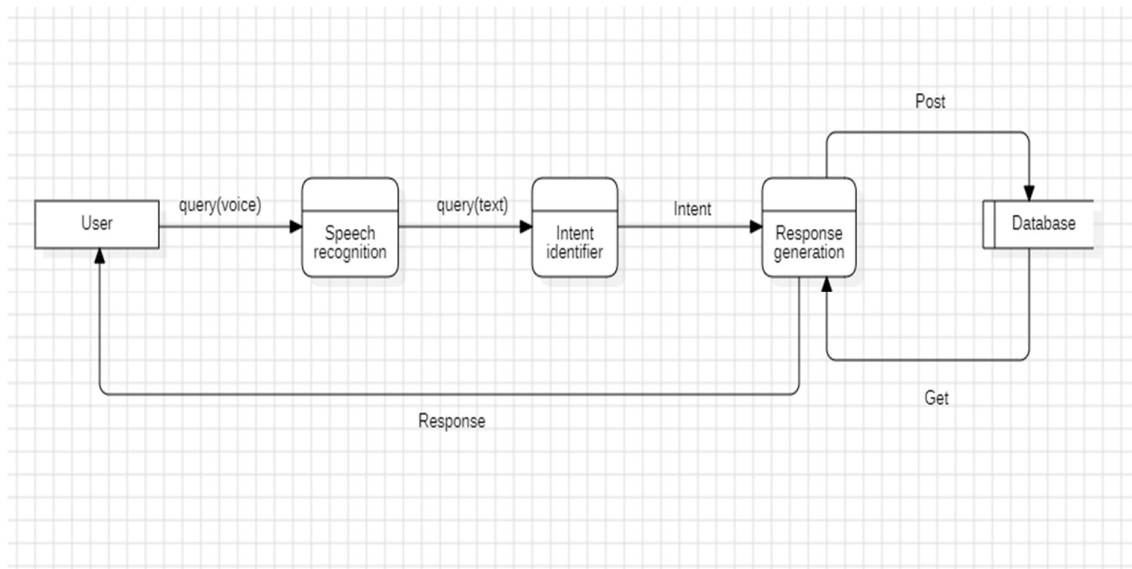
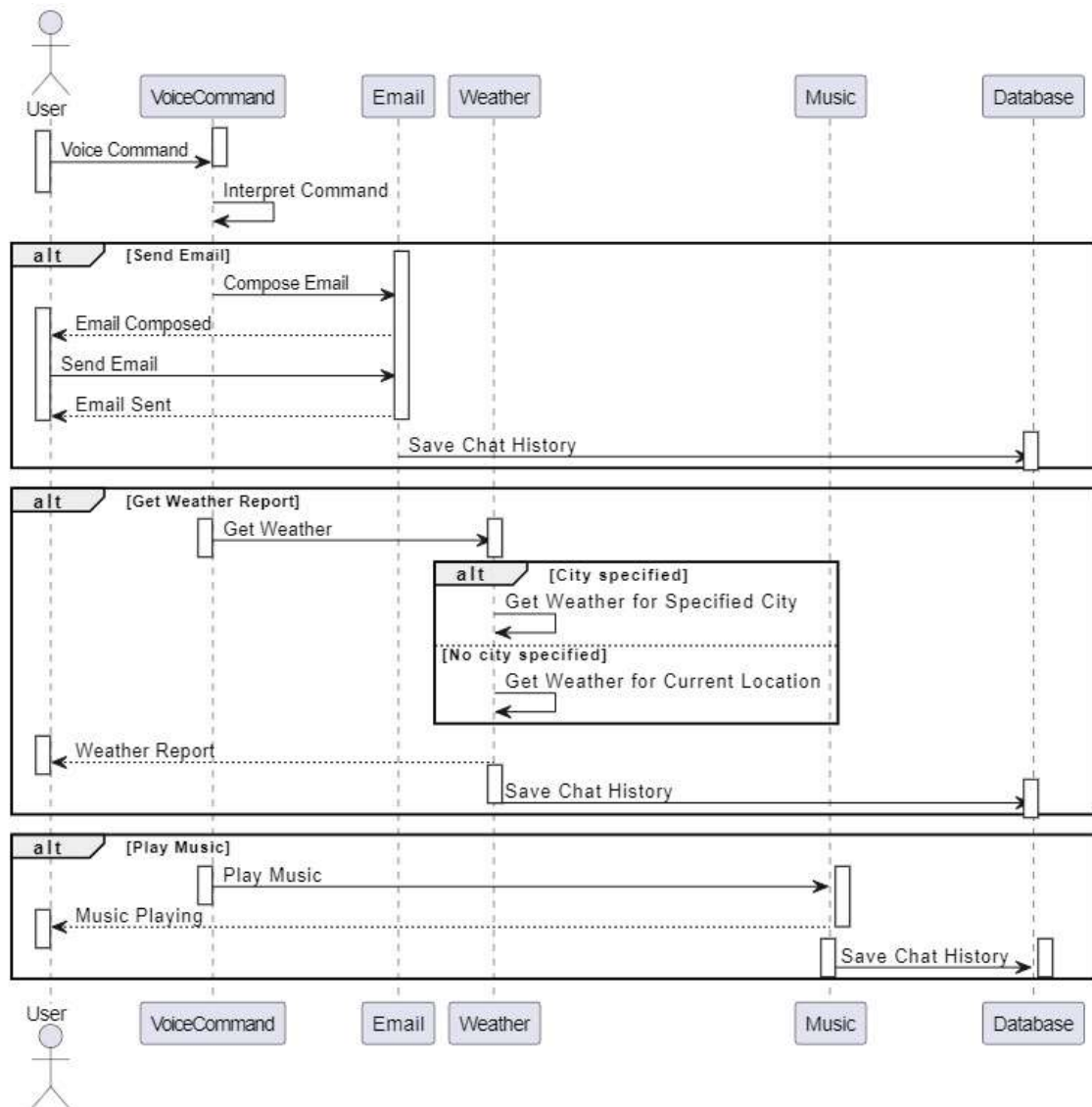
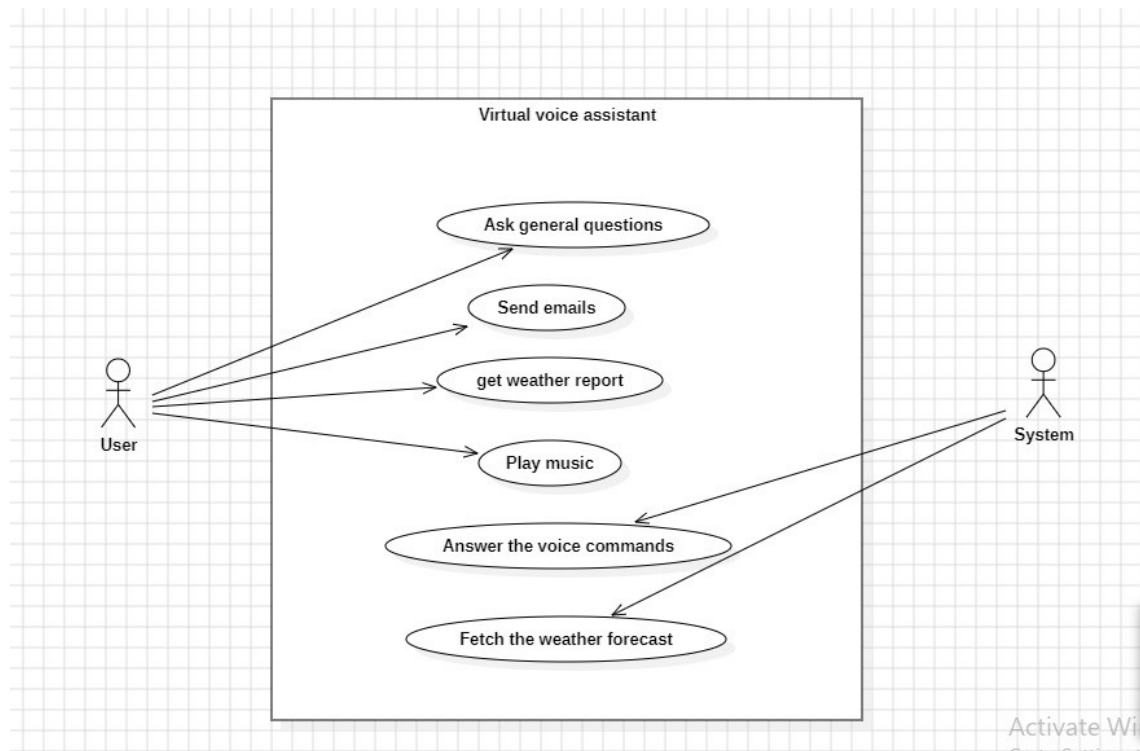


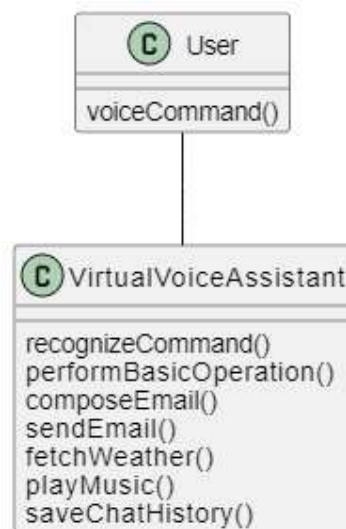
Fig 5.2: Level 1 Data Flow Diagram



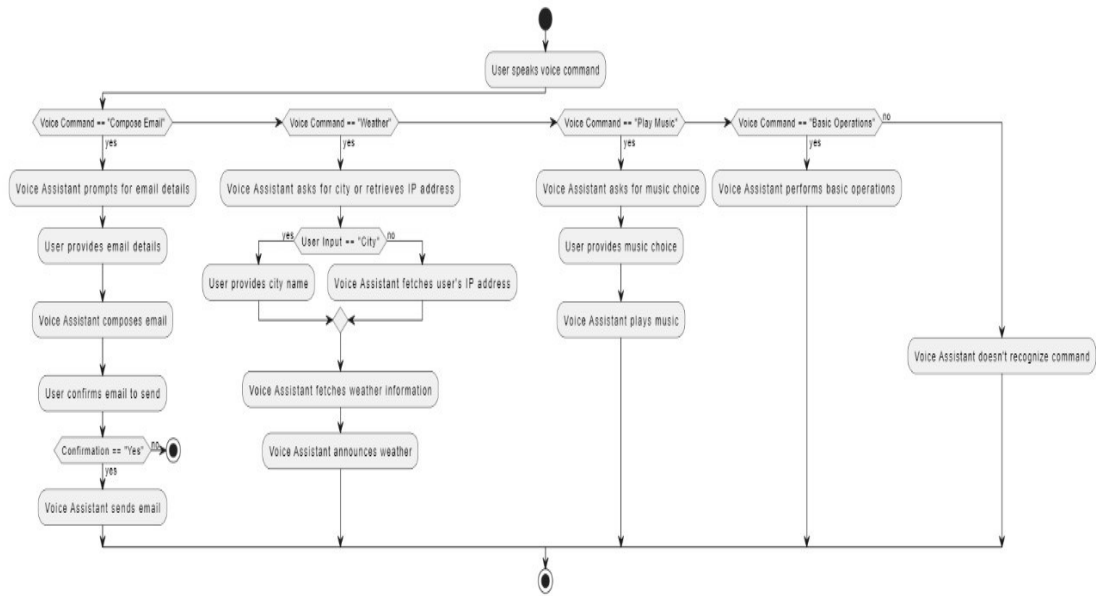
5.3 Sequence diagram



5.4 Use case diagram



5.5 Class diagram



5.6 Activity diagram

6.Implementation

Code :

```
import pyttsx3 as py
import speech_recognition as sr
import randfacts
import datetime

from selenium_web import infow
from YouTube import music
from News import news
from jokes import *
from weather import *
from wishme import *
from gemini import gemini

engine = py.init()
rate = engine.getProperty('rate') #Speed of the voice
engine.setProperty('rate',130) #change the speed rate to 180(default id 200)
voices = engine.getProperty('voices')
engine.setProperty('voice',voices[1].id) #change the voice at 0 index male voice
and 1 index female voice

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

today_date = datetime.datetime.now()
```

```

r = sr.Recognizer() #Recognizer use microphone

speak("Hello sir, Good " + wish() + " i'm your voice assistant.")
speak("How are You?")

with sr.Microphone() as source:

    r.energy_threshold = 10000 #It increace the spectrum of voice
    r.adjust_for_ambient_noise(source, 1.2) #cancle all noises around you
    print("listening")
    audio = r.listen(source)
    text_found = False
    while not text_found:
        text = r.recognize_google(audio)
        text_found = True

    print(text)

if "what" and "about" and "you" in text:
    speak("I am also having a good day sir")

first = True

while True:
    if first:
        first = False
        speak("What can I do for you?")
    else:

```

```
speak("Anything else I can help you with?")
```

```
with sr.Microphone() as source:
```

```
    r.energy_threshold = 10000
```

```
    r.adjust_for_ambient_noise(source,1.5)
```

```
    print("listening...")
```

```
    audio = r.listen(source)
```

```
    text_found = False
```

```
    while not text_found:
```

```
        try:
```

```
            text2 = r.recognize_google(audio)
```

```
            text_found = True
```

```
        except:
```

```
            speak("Please repeat!")
```

```
            print("Please repeat!")
```

```
    print(text2)
```

```
if "information" in text2:
```

```
    speak("You need information related to which topic?")
```

```
    with sr.Microphone() as source:
```

```
        r.energy_threshold = 10000
```

```
        r.adjust_for_ambient_noise(source, 1.5)
```

```
        print("listening...")
```

```
        audio = r.listen(source)
```

```
        infor = r.recognize_google(audio)
```

```
speak("Searching {} in wikipedia".format(infor))
```

```
print("Searching {} in wikipedia".format(infor))
```

```
assist = infow()
```

```
assist.get_info(infor)
```

```
assist.wait_to_close()
```

```
elif "play" and "video" in text2:
```

```
    speak("You want me to play which video?")
```

```
    with sr.Microphone() as source:
```

```
        r.energy_threshold = 10000
```

```
        r.adjust_for_ambient_noise(source, 1.5)
```

```
        print("listening...")
```

```
        audio = r.listen(source)
```

```
        video = r.recognize_google(audio)
```

```
    speak("Playing {} in YouTube".format(video))
```

```
    print("Playing {} in YouTube".format(video))
```

```
assist = music()
```

```
assist.play(video)
```

```
assist.wait_to_close()
```

```
elif "news" in text2:
```

```
    print("Sure sir, Now I will read news for you")
```

```
    speak("Sure sir, Now I will read news for you")
```

```
arr = news()
```



```
for i in range(len(arr)):
```

```
    print(arr[i])
```

```
    speak(arr[i])
```

```
elif "fact" in text2:
```

```
    print("Sure sir ")
```

```
    speak("Sure sir ")
```

```
x = randfacts.get_fact()
```

```
print(x)
```

```
speak("Did you know that, "+x)
```

```
elif "joke" in text2:
```

```
    print("Sure sir")
```

```
    speak("Sure sir")
```

```
arr = joke()
```

```
print(arr[0])
```

```
speak(arr[0])
```

```
print(arr[1])
```

```
speak(arr[1])
```

```
elif "weather" in text2:
```

```
    print("Sure sir")
```

```
    speak("Sure sir")
```

```
print("Temperature in Aurangabad is "+str(temp())+" degree celcius" " and  
with " +str(des()))
```

```
speak("Temperature in Aurangabad is " + str(temp()) + " degree celcius" "  
and with " + str(des()))
```

```
elif "time" in text2:
```

```
print("Sure sir")
```

```
speak("Sure sir")
```

```
print("Today is "+today_date.strftime("%d")+ " of "  
+today_date.strftime("%B")+ " And its currenty "  
+(today_date.strftime("%I:"))+(today_date.strftime("%M"))+" ")
```

```
speak("Today is " + today_date.strftime("%d") + " of" +  
today_date.strftime("%B") + " And its currenty " + (today_date.strftime("%I"))  
+ (today_date.strftime("%M")) + (today_date.strftime("%D")))
```

```
else:
```

```
if "no thanks" in text2:
```

```
exit()
```

```
response = gemini(text2)
```

```
print(response)
```

```
speak(response)
```

Output:

```
Terminal: Local x + v
PS C:\VoiceAssistantProject\VoiceAssistantProject> python main.py
listening
I am good how are you
listening...
tell me some news
Sure sir, Now I will read news for you
Number 1: [Removed].
Number 2: Tesla is settling with the family of the Apple engineer who died in an Autopilot crash.
Number 3: The Supercut app brings good Netflix to the Vision Pro.
Number 4: An adult version of scented markers exists... as thermal paste.
Number 5: Ask Jerry: Will Google help me access a loved one's data after they pass away?.
listening...
tell me a joke
Sure sir
Why did the octopus beat the shark in a fight?
Because it was well armed.
listening...
tell me today's weather
Sure sir
Temperature in Aurangabad is 39.49 degree celcius and with clear sky
listening...
play video
listening...
Playing cricket highlights England cricket in YouTube
DevTools listening on ws://127.0.0.1:64345/devtools/browser/a5587a86-204d-440f-b90b-fc7efd5ffe4b
Press Enter to close the browser...
```

7. Conclusion

We've created a helpful tool capable of answering questions, fetching weather updates, cracking jokes, sharing the date and time, delivering news, and even accessing YouTube and Wikipedia through Selenium WebDriver.

This project demonstrates our proficiency in Python programming and our ability to integrate various APIs and web scraping techniques. By incorporating voice commands, we've made the assistant user-friendly and accessible.

The use of Selenium WebDriver for web content access adds versatility to the assistant, ensuring it can handle a variety of tasks effectively.

Overall, this project showcases our skills in software development, API integration, web scraping, and user interface design. It's a practical implementation of a voice assistant that can be further improved and expanded upon in the future.