PROJECT REPORT:VOICE ASSISTANT IN PYTHON

1. Introduction

This project implements a voice assistant using Python. It uses speech recognition to listen for commands, text-to-speech for replies, and integrates additional libraries for tasks like playing YouTube videos, fetching Wikipedia summaries, and telling jokes.

2. Objectives

- Build an interactive voice assistant - Enable voice-based commands - Use APIs and libraries to perform tasks like information retrieval and media playback

3. System Requirements

- Python 3.x - Internet connection - Microphone - Libraries: speech_recognition, pyttsx3, pywhatkit, wikipedia, pyjokes

4. Conclusion

This project demonstrates a functional voice assistant using Python. The assistant can be expanded further with more features such as smart home integration, reminders, or offline speech models.

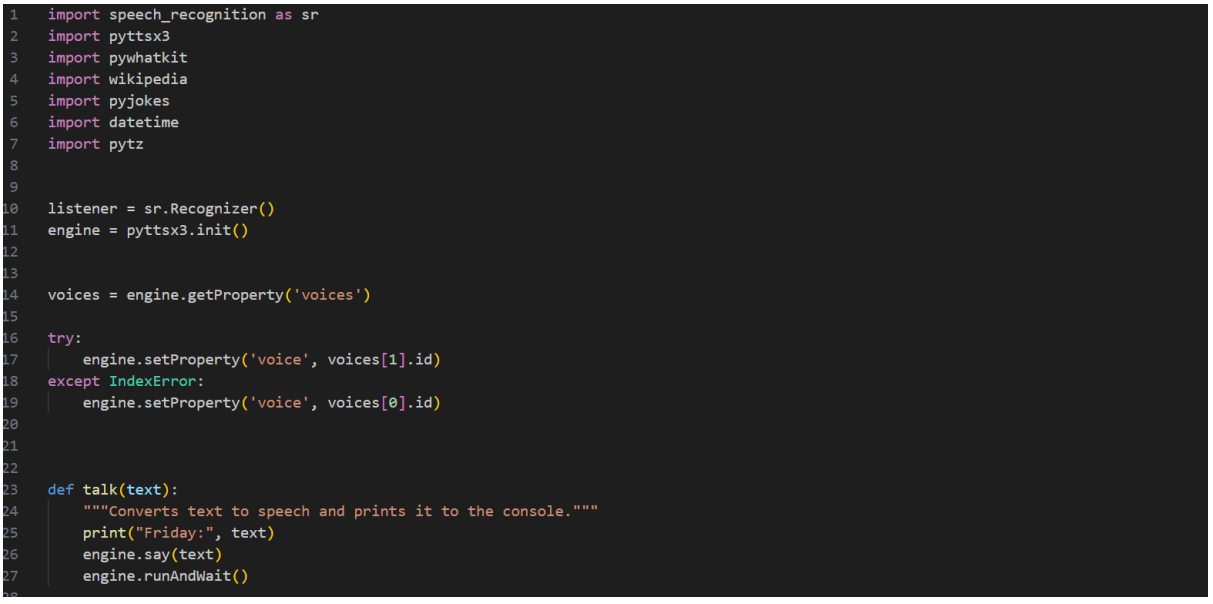
FRIDAY PYTHON CODE

```
import speech_recognition as sr
import pyttsx3
import pywhatkit
import wikipedia
import pyjokes
```

```
import datetime
import pytz


listener = sr.Recognizer()
engine = pyttsx3.init()


voices = engine.getProperty('voices')

try:
    engine.setProperty('voice', voices[1].id)
except IndexError:
    engine.setProperty('voice', voices[0].id)



def talk(text):
    """Converts text to speech and prints it to the console."""
    print("Friday:", text)
    engine.say(text)
    engine.runAndWait()

def take_command():
    """Listens to user's voice, recognizes the command using Google API, and processes it."""
    command = ""
    try:
        with sr.Microphone() as source:
            print("Listening...")
            listener.adjust_for_ambient_noise(source)
            voice = listener.listen(source)


            command = listener.recognize_google(voice)

            command = command.lower()
            if 'Friday' in command:
                command = command.replace('Friday', '').strip()
            print("You said:", command)

    except sr.UnknownValueError:

        print("Sorry, I didn't understand that.")
        talk("Sorry, I didn't catch that. Could you please repeat?")
        command = ""
    except sr.RequestError:

        print("Network error. Check your internet connection.")
```

```
        talk("I seem to have a network problem. Please check your internet connection.")
        command = ""
    except Exception as e:

        print(f"An unexpected error occurred: {e}")
        command = ""

    return command

def run_Friday():
    """Processes the recognized command and responds accordingly."""
    command = take_command()
    if not command:
        return


    if 'play' in command:
        song = command.replace('play', '').strip()
        talk("Playing " + song)
        pywhatkit.playonyt(song)


    elif 'time' in command:

        IST = pytz.timezone('Asia/Kolkata')

        now = datetime.datetime.now(IST)
        time = now.strftime('%I:%M %p')
        talk(f"The current time in India is {time}")


    elif 'date' in command:
        today = datetime.date.today().strftime('%A, %B %d, %Y')
        talk(f"Today is {today}")


    elif 'who is' in command or 'who the heck is' in command:
        person = command.replace('who the heck is', '').replace('who is', '').strip()
        try:

            info = wikipedia.summary(person, sentences=1, auto_suggest=False, redirect=True)
            print(info)
            talk(info)
        except wikipedia.exceptions.PageError:
            talk(f"Sorry, I couldn't find any information on {person}")
        except wikipedia.exceptions.DisambiguationError:
            talk(f"The term {person} is ambiguous. Please be more specific.")
```

```
    elif 'what is' in command or 'tell me about' in command or 'search for' in command:

        query = command.replace('what is', '').replace('tell me about', '').replace('search for',
'').strip()

        if query:
            talk(f"Searching the web for {query}")

            pywhatkit.search(query)
        else:
            talk("What would you like me to search for?")


    elif 'are you single' in command or 'married' in command:
        talk("I am in a relationship with Wi-Fi.")
    elif 'joke' in command:
        talk(pyjokes.get_joke())


    else:
        talk("I'm sorry, I am not programmed to handle that specific command yet.")


while True:
    run_Friday()
```



```
import speech_recognition as sr
import pyttsx3
import pywhatkit
import wikipedia
import pyjokes
import datetime
import pytz


listener = sr.Recognizer()
engine = pyttsx3.init()


voices = engine.getProperty('voices')

try:
    engine.setProperty('voice', voices[1].id)
except IndexError:
    engine.setProperty('voice', voices[0].id)



def talk(text):
    """Converts text to speech and prints it to the console."""
    print("Friday:", text)
    engine.say(text)
    engine.runAndWait()
```

```python
def take_command():
    """Listens to user's voice, recognizes the command using Google API, and processes it."""
    command = ""
    try:
        with sr.Microphone() as source:
            print("Listening...")
            listener.adjust_for_ambient_noise(source)
            voice = listener.listen(source)


            command = listener.recognize_google(voice)

            command = command.lower()
            if 'Friday' in command:
                command = command.replace('Friday', '').strip()
            print("You said:", command)

    except sr.UnknownValueError:

        print("Sorry, I didn't understand that.")
        talk("Sorry, I didn't catch that. Could you please repeat?")
        command = ""
    except sr.RequestError:

        print("Network error. Check your internet connection.")
        talk("I seem to have a network problem. Please check your internet connection.")
        command = ""
    except Exception as e:
```

```python
        print(f"An unexpected error occurred: {e}")
        command = ""

    return command

def run_Friday():
    """Processes the recognized command and responds accordingly."""
    command = take_command()
    if not command:
        return


    if 'play' in command:
        song = command.replace('play', '').strip()
        talk("Playing " + song)
        pywhatkit.playonyt(song)


    elif 'time' in command:

        IST = pytz.timezone('Asia/Kolkata')

        now = datetime.datetime.now(IST)
        time = now.strftime('%I:%M %p')
        talk(f"The current time in India is {time}")


```

```
    elif 'date' in command:
        today = datetime.date.today().strftime('%A, %B %d, %Y')
        talk(f"Today is {today}")


    elif 'who is' in command or 'who the heck is' in command:
        person = command.replace('who the heck is', '').replace('who is', '').strip()
        try:

            info = wikipedia.summary(person, sentences=1, auto_suggest=False, redirect=True)
            print(info)
            talk(info)
        except wikipedia.exceptions.PageError:
            talk(f"Sorry, I couldn't find any information on {person}")
        except wikipedia.exceptions.DisambiguationError:
            talk(f"The term {person} is ambiguous. Please be more specific.")


    elif 'what is' in command or 'tell me about' in command or 'search for' in command:

        query = command.replace('what is', '').replace('tell me about', '').replace('search for', '').strip()

        if query:
            talk(f"Searching the web for {query}")

            pywhatkit.search(query)
        else:
            talk("What would you like me to search for?")
```

```
    elif 'are you single' in command or 'married' in command:
        talk("I am in a relationship with Wi-Fi.")
    elif 'joke' in command:
        talk(pyjokes.get_joke())


    else:
        talk("I'm sorry, I am not programmed to handle that specific command yet.")


while True:
    run_Friday()
```