Genie - Desktop Assistant

21CSS101J – PROGRAMMING FOR PROBLEM SOLVING

Mini Project Report

Submitted by

Sanskar [Reg. No.: RA2211003011110]
B.Tech. CSE - CORE

Srishti Chordia [Reg. No.: RA2211003011118]
B.Tech. CSE - CORE

SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Under Section 3 of UGC Act, 1956)
S.R.M. NAGAR, KATTANKULATHUR – 603 203
KANCHEEPURAM DISTRICT

December 2022

TABLE OF CONTENTS

Problem Statement

Concept

Technology has advanced in the twenty-first century. Humans are increasingly compelled to communicate with computers. Today, we teach our computers to complete tasks on their own. As a result, the concept of a virtual assistant was born.

Voice assistants are defined as the software agents which interpret or converts human speech and it responds through synchronized voices, i.e., Siri (Apple Inc.), Alexa (Amazon), Cortana (Microsoft), so forth these are most popular voice assistants.

People who are blind, elderly or physically disabled can work with the device through a virtual assistant. As a result, these disabled people can now also communicate. With the voice assistant, we are therefore moving to the next stage of technological innovation, when we will be able to converse with our machines.

Our virtual assistant is a desktop assistant that uses speech recognition. It can understand and carry out audio instructions given by the user. We don't have to worry about using input devices like keyboard and mouse, so we will use them less. It also saves the user a lot of time. This is a desktop assistant written in python. And accept voice input and provide voice and text output.

Methodology/Procedure

Libraries used-

1. pyttsx3
2. query as query
3. speech_recognition as sr
4. datetime
5. pyaudio
6. Wikipedia
7. Web-browser
8. os
9. smtplib

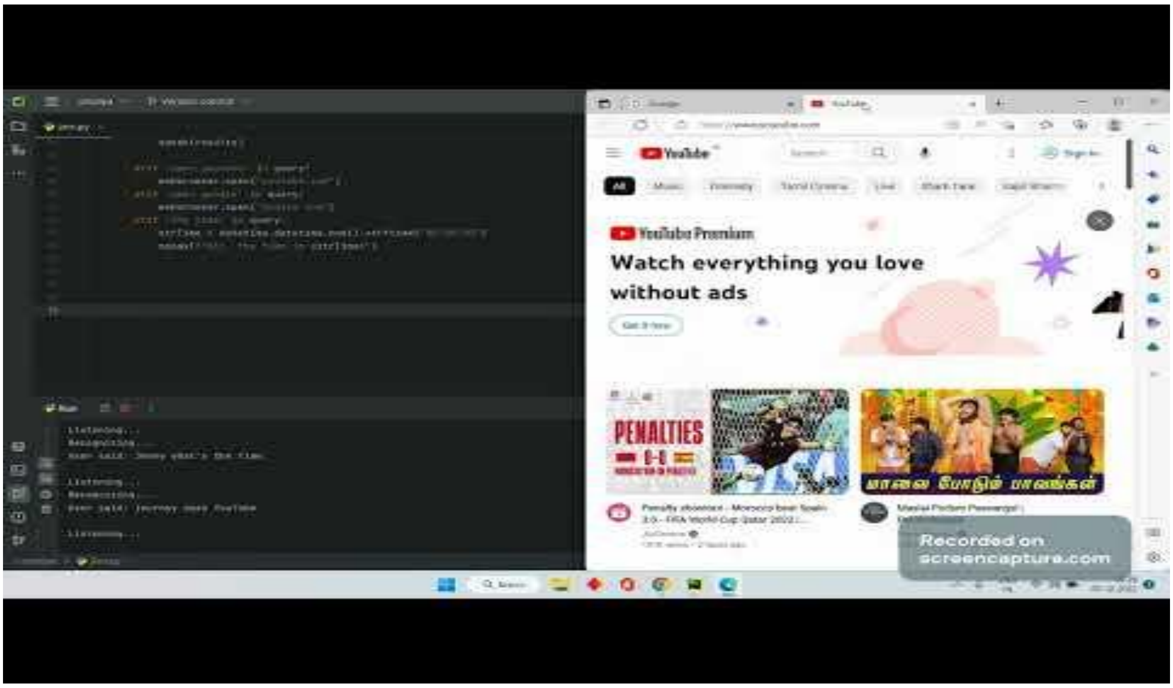
Coding (Python)

```
import pyttsx3
import query as query
import speech_recognition as sr
import datetime
import pyaudio
import wikipedia
import webbrowser
import os
import smtplib

engine = pyttsx3.init('sapi5')

voices = engine.getProperty('voices')   #getting details
of current voice

engine.setProperty('voice', voices[1].id)
def speak(audio):
    engine.say(audio)
    engine.runAndWait() #Without this command, speech will
not be audible to us.
def wishme():
    hour = int(datetime.datetime.now().hour)
    if hour>=0 and hour<=12:
       speak("Good morning")
    elif hour>=12 and hour<16:
        speak("Good Afternoon!")

    else:
        speak("Good Evening!")
    speak("Hi I am Genie. How may I help you")


def takeCommand():
    # It takes microphone input from the user and returns
string output
```

```
    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')  # Using google for voice recognition.
        print(f"User said: {query}\n")  # User query will
be printed.

    except Exception as e:
        # print(e)
        print("Say that again please...")  # Say that
again will be printed in case of improper voice
        return "None"  # None string will be returned
    return query



if __name__ == "__main__":
    wishme()
    while True:
    #-if 1:
        query = takeCommand().lower() #Converting user
query into lower case

        # Logic for executing tasks based on query
        if 'wikipedia' in query:  #if wikipedia found in
the query then this block will be executed
            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:
            webbrowser.open("youtube.com")
        elif 'open google' in query:
            webbrowser.open("google.com")
        elif 'the time' in query:
            strTime = 
datetime.datetime.now().strftime("%H:%M:%S")
            speak(f"Sir, the time is {strTime}")
```

Result

Genie:- The AI assistant



```
            speak(results)

        elif 'open youtube' in query:
            webbrowser.open("youtube.com")
        elif 'open google' in query:
            webbrowser.open("google.com")
        elif 'the time' in query:
            strTime = datetime.datetime.now().strftime("%H:%M:%S")
            speak(f"Sir, the time is {strTime}")
```

```
Listening...
Recognizing...
User said: Jenny what's the time

Listening...
Recognizing...
User said: Journey open YouTube

Listening...
```

Conclusion

Desktop Voice Assistant has been designed with ease of use as the main feature. The Assistant works properly to perform some tasks given by user. This Voice assistant, in today's life style will be more effective in case of saving time, compared to that of previous days. The popularity of voice activated virtual assistants, as well as their future potential, were examined in this study which performs operations in audio format as directed by the user. This desktop assistant can open YouTube, Google, search Wikipedia. It can show date and time and can stream music.