Jarvis code storage \_day 7

*"""  
today I will give the violet the book of knowledge, the google.it will look at that and reply me .I will make this as the google protocol  
"""*import pywhatkit  
import datetime  
import time  
import speech\_recognition as sr  
import pyttsx3  
import wikipedia  
from word2number import w2n  
  
engine = pyttsx3.init('sapi5')  
voices = engine.getProperty('voices')  
r = sr.Recognizer()  
  
for voice in voices:  
 if "zira" in voice.name.lower():  
 engine.setProperty('voice', voice.id)  
 female\_voice\_found = True  
 break  
  
def speak(audio):#speaks using this  
 engine.say(audio)  
 engine.runAndWait()  
  
def command(argument="listening to your voice........."):#gets the command to text and speaks can be used to get the speech and use it for other properties  
 with sr.Microphone() as source:  
 print(argument)  
 r.pause\_threshold=0.6  
 r.energy\_threshold=300  
  
 try:  
 audio = r.listen(source, timeout=15, phrase\_time\_limit=30)  
 except sr.WaitTimeoutError:  
 print("Listening timed out.")  
 speak("I didn’t hear anything.")  
 return "None"  
  
  
 try:  
 print("Recognizing...")  
 query=r.recognize\_google(audio).lower()  
 print(f"\nYou said: {query}\n")  
 speak(query)  
 return query  
  
 except sr.UnknownValueError:  
 print("Could not understand audio")  
 return "None"  
 except sr.RequestError:  
 print("Could not request results from Google Speech Recognition service")  
 return "None"  
  
def call\_assistant(call\_word="violet"):#program initiator  
 with sr.Microphone() as source:  
 print("initialize the assistant....please speak the code ")  
 pause\_threshold =1  
 try:  
 audio = r.listen(source,timeout=5)  
 except sr.WaitTimeoutError:  
 return None  
  
 try:  
 print("just a sec.....")  
 query2=r.recognize\_google(audio,language='en-in').lower()  
 print(f"\nYou said: {query2}\n")  
 return query2  
 except:  
 return None  
def greetings():#greetings and other things in future like time  
 now=datetime.datetime.now()  
 hour=now.hour  
 if(0<=hour<12):  
 speak("good morning christy.")  
 elif(12<=hour<18):  
 speak("good afternoon christy.")  
 else:  
 speak("good evening christy.")  
 speak("i am violet. your personal assistant.")  
 speak("how can i help you today?")  
def search\_summary(contxt,sentences):  
 try:  
 summary = wikipedia.summary(contxt, sentences=sentences)  
 return summary  
 except Exception as e:  
 print("Something went wrong:",e)  
  
def google\_search(query):  
 try:  
 speak(f"searching {query}...")  
 pywhatkit.search(query)  
 except Exception as e:  
 print("Something went wrong:",e)  
  
def youtube\_search(query):  
 try:  
 speak(f"playing {query}...")  
 pywhatkit.playonyt(query)  
 except Exception as e:  
 print("Something went wrong:",e)  
def safe\_command(prompt="please say it again"):# in here the safe command ensures that the protocol initiator gives one more chance to the user  
 query= command("listening the protocol you want to use........")  
 if query == "None":  
 print(prompt)  
 speak(prompt)  
 query = command()  
 return query  
def ultra\_modified\_safe\_command(retries=2,fall\_back="None"):  
 for i in range(retries+1):  
 query=command(f"attempt {i+1},listening to the protocol you want to use........")#the protocol name  
 if query!="None" and query.strip()!="":  
 return query#protocol name is obtained then this loop stops  
 speak("i didn't got that")  
 print("i didn't got that.")  
 speak("you have not spoken the protocol that you want yet. ")  
 print("you have not spoken the protocol that you want yet.")  
  
 return fall\_back  
  
  
  
def protocol\_initiator():#protocols initiated by calling this function  
 speak("which protocol do you want to use?")  
 try:  
 protocol=ultra\_modified\_safe\_command()  
 if "wikipedia" in protocol:  
 speak("what would you like to search in wikipedia?")  
 information=command("listening to your search query in wikipedia protocol.....")  
 speak("how many sentences would you like to search in wikipedia?")  
 try:  
 num\_query=command()  
 print("Heard number as:", num\_query)  
 no=w2n.word\_to\_num(num\_query)  
 except:  
 print("i couldnt find anything using 4 as default")  
 no=4  
 summary=search\_summary(information,no)  
 print(summary)  
 speak(summary)  
 if "python search" in protocol:  
 speak("what would you like to search in python?")  
 search=command("listenig the search query in python search protocol........")  
 try:  
 google\_search(search) #pykit function called  
 except Exception as e:  
 print("Something went wrong:",e)  
 if "play youtube" in protocol:  
 speak("what would you like to play in youtube?")  
 search=command("listening to the search query in play youtube protocol..........")  
 try:  
 youtube\_search(search)  
 except Exception as e:  
 print("Something went wrong:",e)  
 except Exception as e:  
 print("Something went wrong:",e)  
  
  
  
  
  
  
try:  
 while True:  
 result = call\_assistant()  
 k=False  
 count=0  
 if result is not None and "violet" in result:  
 while True:#violet command loop runs main portion of the time  
 if count<1:  
 greetings()  
 count+=1  
  
 commander = command()  
 if "initiate protocol" in commander:  
 protocol\_initiator()#calls the protocol initiator if it returns none the  
  
  
  
  
 if commander == "stop":  
 speak("ok you said stop. ")  
 speak("so see you again christy. ")  
 speak("and wishing you all the best with devika. ")  
 k=True  
 break  
 time.sleep(1)  
 if(k==True):  
 break  
  
 elif(result is None):  
 speak("you are not saying any thing dear")  
 time.sleep(2)#2 seconds  
 elif result is not None:  
 speak("say my name please")  
 time.sleep(1)  
  
except Exception as e:  
 print("Something went wrong",e)  
 speak("sorry.an error occured.")

Jarvis code storage\_day 8