|  |
| --- |
| import speech\_recognition as sr # recognise speech |
|  | import playsound # to play an audio file |
|  | from gtts import gTTS # google text to speech |
|  | import random |
|  | from time import ctime # get time details |
|  | import webbrowser # open browser |
|  | import yfinance as yf # to fetch financial data |
|  | import ssl |
|  | import certifi |
|  | import time |
|  | import os # to remove created audio files |
|  |  |
|  | class person: |
|  | name = '' |
|  | def setName(self, name): |
|  | self.name = name |
|  |  |
|  | def there\_exists(terms): |
|  | for term in terms: |
|  | if term in voice\_data: |
|  | return True |
|  |  |
|  | r = sr.Recognizer() # initialise a recogniser |
|  | # listen for audio and convert it to text: |
|  | def record\_audio(ask=False): |
|  | with sr.Microphone() as source: # microphone as source |
|  | if ask: |
|  | speak(ask) |
|  | audio = r.listen(source) # listen for the audio via source |
|  | voice\_data = '' |
|  | try: |
|  | voice\_data = r.recognize\_google(audio) # convert audio to text |
|  | except sr.UnknownValueError: # error: recognizer does not understand |
|  | speak('I did not get that') |
|  | except sr.RequestError: |
|  | speak('Sorry, the service is down') # error: recognizer is not connected |
|  | print(f">> {voice\_data.lower()}") # print what user said |
|  | return voice\_data.lower() |
|  |  |
|  | # get string and make a audio file to be played |
|  | def speak(audio\_string): |
|  | tts = gTTS(text=audio\_string, lang='en') # text to speech(voice) |
|  | r = random.randint(1,20000000) |
|  | audio\_file = 'audio' + str(r) + '.mp3' |
|  | tts.save(audio\_file) # save as mp3 |
|  | playsound.playsound(audio\_file) # play the audio file |
|  | print(f"kiri: {audio\_string}") # print what app said |
|  | os.remove(audio\_file) # remove audio file |
|  |  |
|  | def respond(voice\_data): |
|  | # 1: greeting |
|  | if there\_exists(['hey','hi','hello']): |
|  | greetings = [f"hey, how can I help you {person\_obj.name}", f"hey, what's up? {person\_obj.name}", f"I'm listening {person\_obj.name}", f"how can I help you? {person\_obj.name}", f"hello {person\_obj.name}"] |
|  | greet = greetings[random.randint(0,len(greetings)-1)] |
|  | speak(greet) |
|  |  |
|  | # 2: name |
|  | if there\_exists(["what is your name","what's your name","tell me your name"]): |
|  | if person\_obj.name: |
|  | speak("my name is Alexis") |
|  | else: |
|  | speak("my name is Alexis. what's your name?") |
|  |  |
|  | if there\_exists(["my name is"]): |
|  | person\_name = voice\_data.split("is")[-1].strip() |
|  | speak(f"okay, i will remember that {person\_name}") |
|  | person\_obj.setName(person\_name) # remember name in person object |
|  |  |
|  | # 3: greeting |
|  | if there\_exists(["how are you","how are you doing"]): |
|  | speak(f"I'm very well, thanks for asking {person\_obj.name}") |
|  |  |
|  | # 4: time |
|  | if there\_exists(["what's the time","tell me the time","what time is it"]): |
|  | time = ctime().split(" ")[3].split(":")[0:2] |
|  | if time[0] == "00": |
|  | hours = '12' |
|  | else: |
|  | hours = time[0] |
|  | minutes = time[1] |
|  | time = f'{hours} {minutes}' |
|  | speak(time) |
|  |  |
|  | # 5: search google |
|  | if there\_exists(["search for"]) and 'youtube' not in voice\_data: |
|  | search\_term = voice\_data.split("for")[-1] |
|  | url = f"https://google.com/search?q={search\_term}" |
|  | webbrowser.get().open(url) |
|  | speak(f'Here is what I found for {search\_term} on google') |
|  |  |
|  | # 6: search youtube |
|  | if there\_exists(["youtube"]): |
|  | search\_term = voice\_data.split("for")[-1] |
|  | url = f"https://www.youtube.com/results?search\_query={search\_term}" |
|  | webbrowser.get().open(url) |
|  | speak(f'Here is what I found for {search\_term} on youtube') |
|  |  |
|  | # 7: get stock price |
|  | if there\_exists(["price of"]): |
|  | search\_term = voice\_data.lower().split(" of ")[-1].strip() #strip removes whitespace after/before a term in string |
|  | stocks = { |
|  | "apple":"AAPL", |
|  | "microsoft":"MSFT", |
|  | "facebook":"FB", |
|  | "tesla":"TSLA", |
|  | "bitcoin":"BTC-USD" |
|  | } |
|  | try: |
|  | stock = stocks[search\_term] |
|  | stock = yf.Ticker(stock) |
|  | price = stock.info["regularMarketPrice"] |
|  |  |
|  | speak(f'price of {search\_term} is {price} {stock.info["currency"]} {person\_obj.name}') |
|  | except: |
|  | speak('oops, something went wrong') |
|  | if there\_exists(["exit", "quit", "goodbye"]): |
|  | speak("going offline") |
|  | exit() |
|  |  |
|  |  |
|  | time.sleep(1) |
|  |  |
|  | person\_obj = person() |
|  | while(1): |
|  | voice\_data = record\_audio() # get the voice input |
|  | respond(voice\_data) # respond |