COVID-19 Audio Assistant

Contents

Abstract

COVID-19 has drastically changed the way of human living in 2020. We have to stay cautious, careful and follow the standard operating procedures to stop the spread of the disease. There are travel restrictions for different countries. We might want to know the number of cases in different countries. Our goal is to easily help the users know the number of COVID cases in each country. Our project is designed in such a way that we can ask the bot the number of cases in a country and the bot will reply us the number of cases. To achieve this, we will extract data from a website that updates the number of cases in each country. We will use the microphone and speech recognition libraries to listen to the user. Based on the query, the bot will match the country and reply us the number of cases in the country. This method could be helpful to know the cases and even used in AI speech assistants to answer the question precisely.

Introduction

As technology advances, the different means by which people interact with technology also changes. Earlier, the text was the most commonly used way of communicating with Artificial Intelligence based assistants. But now, voice has taken over, and voice-based assistants have gained significant popularity. Voice assistants refer to digital assistants who listen to the user's voice commands and provide real-time responses to the user's queries.

Our project aims to provide relevant details regarding the COVID-19 pandemic, which has created havoc globally. As this issue persists, it is essential to know the overall impact, the number of reported cases, deaths by country, territory or conveyance. The COVID-19 voice assistant, which we developed with Python, assists us by answering COVID related queries by scraping a web site containing information regarding this pandemic. The web scraping tool ParseHub serves the purpose of extracting required data from web sites. This project also uses Google speech recognition API to identify voice commands and convert speech to text. Our voice assistant seeks answers to the user's query from the initially collected data. As the pandemic is still on, we also used the multithreading library in Python to keep the data updated.

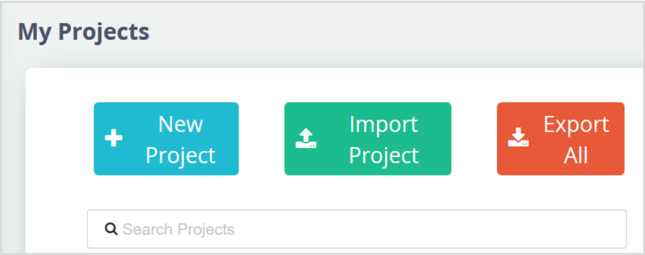
Methods

Data collection instruments and procedures

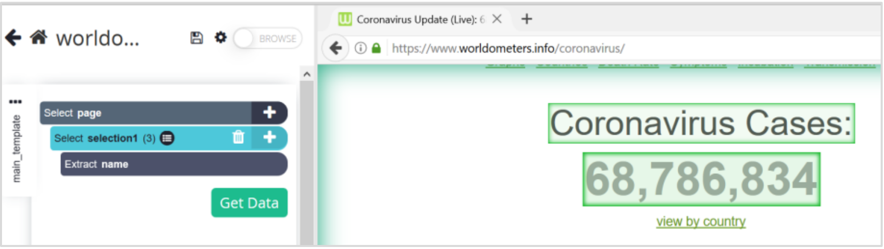
In this project, we have used ParseHub for web scraping purpose. ParseHub is a powerful and free web scraper tool, which seamlessly perform data extraction from different websites. For substantial data mining, ParseHub includes several features like image file scraping, automatic IP rotation, scheduled runs and data accessibility in multiple forms including JSON and csv format. The first step of web scraping process is installing ParseHub application.

The detailed web scraping process with ParseHub is as follows.

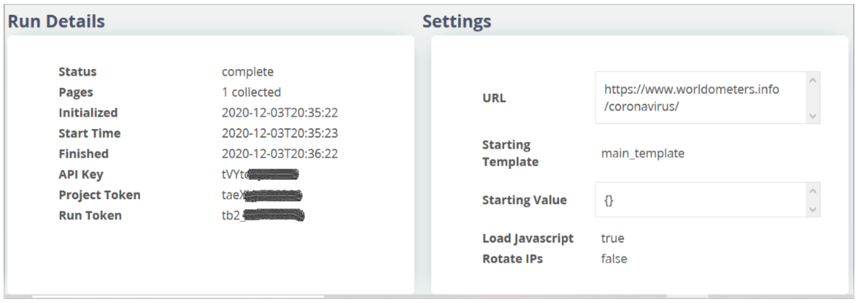
- Open ParseHub and select the option for a new project.



- In the space provided for pasting the website URL, paste the link to the Worldometers website.
- Click select the data that needs to be scraped from the website. It is possible to use the relative selection tool to connect different data.



- After selecting data, click 'Get Data' to retrieve it, after which the API key, Project token, and Run token is generated. These tokens are used while running the python code.



Run Details

Status	complete
Pages	1 collected
Initialized	2020-12-03T20:35:22
Start Time	2020-12-03T20:35:23
Finished	2020-12-03T20:36:22
API Key	tVYt[redacted]
Project Token	tae[redacted]
Run Token	tb2_[redacted]

Settings

URL	https://www.worldometers.info/coronavirus/
Starting Template	main_template
Starting Value	{}
Load Javascript	true
Rotate IPs	false

Libraries required

- **pyttsx3**

  This package pyttxs3 in python is used to convert text to speech.

- **Speech recognition**

  This speech recognition package is used to identify and convert the human's speech to text.

- **Pyaudio**

  PyAudio is used to play and record audio on a variety of platforms in python. It provides Python bindings for audio I/O library-PortAudio.

- **JSON**

  This JSON module is used to store and exchange data.

- **requests**

  This request module can be used to send all kinds of the HTTP request. It took URL as parameters and used to access a URL.

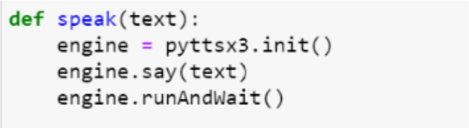
Implementation

- **Import the necessary libraries**

```
import requests
import json
import pyttsx3
import speech_recognition as sr
import re
import threading
import time
```

- **Set up the speech engine**

Here, we convert text to speech. The speak function accepts the text as its argument and uses the pyttsx3 module to initialize the engine. runAndWait() function invokes callbacks for engine notifications.

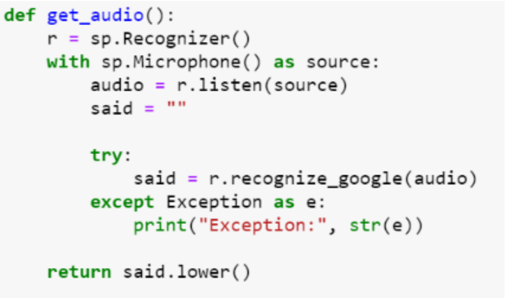


```python
def speak(text):
    engine = pyttsx3.init()
    engine.say(text)
    engine.runAndWait()
```

- **Set up a function for voice assistant**

Designed a function get_audio for the COVID-19 voice assistant to recognize and to accept human speech. Here, the microphone receives the human voice, and the recognizer identifies the voice to answer the query of human.

The exception handling is included to handle the exception during the run time.



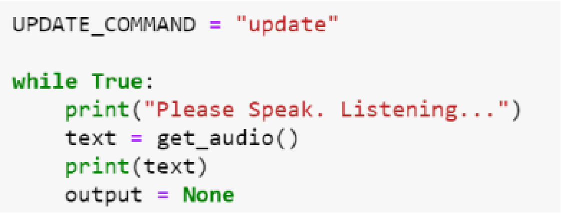
```python
def get_audio():
    r = sp.Recognizer()
    with sp.Microphone() as source:
        audio = r.listen(source)
        said = ""

        try:
            said = r.recognize_google(audio)
        except Exception as e:
            print("Exception:", str(e))

    return said.lower()
```

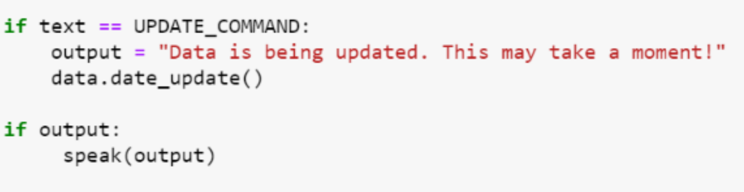
- **To get updated information from the website**

When the console says, “Please speak. Listening…”, say “how many coronavirus cases in Canada,” The assistant would reply with the number of COVID19 cases across Canada. We can say, “update” and the assistant will update the values.



```python
UPDATE_COMMAND = "update"

while True:
    print("Please Speak. Listening...")
    text = get_audio()
    print(text)
    output = None
```

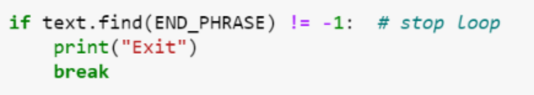


```
if text == UPDATE_COMMAND:
    output = "Data is being updated. This may take a moment!"
    data.date_update()

if output:
     speak(output)
```

- **To exit from running application**

To exit from the application, we need to input the “stop” command through the microphone.

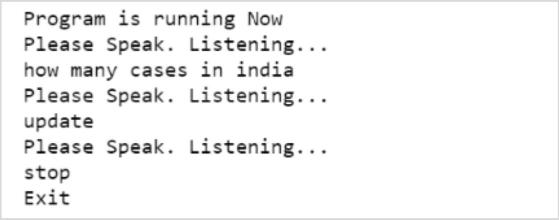


```
if text.find(END_PHRASE) != -1:  # stop loop
    print("Exit")
    break
```

Results

Developed a COVID-19 voice assistant, which helps the public by providing updated and reliable COVID-related information such as the number of coronavirus cases in different countries and total deaths. It delivers updated information as new data is added to the Worldometers website.

Appended the screenshot and mp3 audio, which depicts the received output.



```
Program is running Now
Please Speak. Listening...
how many cases in india
Please Speak. Listening...
update
Please Speak. Listening...
stop
Exit
```

result.mp3

Conclusions and Future Work

The report concludes that voice assistants have revolutionized the whole aspect of the conversation. We have successfully implemented a COVID-19 voice assistant to provide COVID updates by using the ParseHub online web scrapping tool, and it scrapes data from the Worldometers website. It fetches the number of corona cases, death cases count, total corona cases, country list, and country-wise COVID cases with the support of API key, project token, and run token. We also attached an output with an mp3 audio file corresponding to the human queries and an interactive agent's replies.

Apart from the program, we have improved our project developing skills and learned coding skills, and gained experience in a team's platform. We started with brainstorming ideas, getting connected in Teams, and ended up with an excellent COVID-19 voice assistant model.

We can develop this project by having some design improvements, additional functions, and interface improvements to web scrape from any websites and respond based on the user voice input. Since this is a Future emerging technology, we can use it in every sector, such as medical, banking, and teaching.

References

1. https://www.parsehub.com/docs/ref/api/v2/
2. https://www.worldometers.info/coronavirus/
3. https://www.lfd.uci.edu/~gohlke/pythonlibs/#pyaudio
4. https://towardsdatascience.com/how-to-build-your-own-ai-personal-assistant-using-python-f57247b4494b