INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

ID151 INDEPENDENT PROJECT REPORT

ON

SPEECH BOT ON WEATHER FORECAST

Submitted to:

Dr. Mittapalle Kiran Reddy

Assistant Professor

Submitted by:

Kartavya Gupta

AD24B1028

7th April 2025

CERTIFICATE

This is to certify that Kartavya Gupta of Artificial Intelligence and Data Science branch in 1st year of INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, Racihur has performed and completed the project entitled “SPEECH BOT ON WEATHER FORECAST” under my guidance for the session 2024-2025. It is his original work and is in accordance with the guidelines of the institute.

____

Dr. Mittapalle Kiran Reddy

Assistant Professor

ACKNOWLEDGEMENT

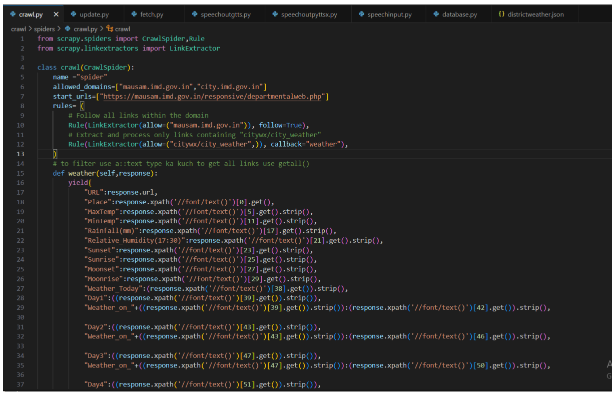
I, Kartavya Gupta, of Artificial Intelligence and Data Science branch in 1st year sincerely acknowledge my mentor Dr. Mittapalle Kiran Reddy for his invaluable guidance in selecting this extremely important topic for my project work. I express my sincere gratitude for his knowledgeable suggestions, comments, advice and guidance throughout. I would like to thank my fellow classmates,seniors who helped me during the course of project work.

INDEX

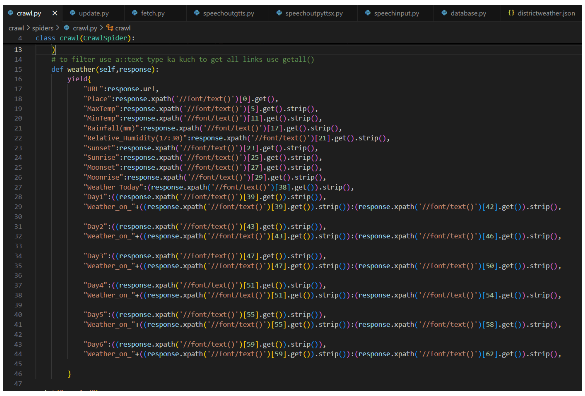
AIM

To create a speech bot which takes speech input of district name and gives overview of weather in speech using the government's official website for weather.

Web Scraping



```python
from scrapy.spiders import CrawlSpider,Rule
from scrapy.linkextractors import LinkExtractor

class crawl(CrawlSpider):
    name ="spider"
    allowed_domains=["mausam.imd.gov.in","city.imd.gov.in"]
    start_urls=["https://mausam.imd.gov.in/responsive/departmentalweb.php"]
    rules= (
        # Follow all links within the domain
        Rule(LinkExtractor(allow=("mausam.imd.gov.in")), follow=True),
        # Extract and process only links containing "citywx/city_weather"
        Rule(LinkExtractor(allow=("citywx/city_weather",)), callback="weather"),
    )
    # to filter use a::text type ka kuch to get all links use getall()
    def weather(self,response):
        yield{
            "URL":response.url,
            "Place":response.xpath('//font/text()')[0].get(),
            "MaxTemp":response.xpath('//font/text()')[5].get().strip(),
            "MinTemp":response.xpath('//font/text()')[11].get().strip(),
            "Rainfall(mm)":response.xpath('//font/text()')[17].get().strip(),
            "Relative_Humidity(17:30)":response.xpath('//font/text()')[21].get().strip(),
            "Sunset":response.xpath('//font/text()')[23].get().strip(),
            "Sunrise":response.xpath('//font/text()')[25].get().strip(),
            "Moonset":response.xpath('//font/text()')[27].get().strip(),
            "Moonrise":response.xpath('//font/text()')[29].get().strip(),
            "Weather_Today":(response.xpath('//font/text()')[38].get()).strip(),
            "Day1":((response.xpath('//font/text()')[39].get()).strip()),
            "Weather_on_"+((response.xpath('//font/text()')[39].get()).strip()):(response.xpath('//font/text()')[42].get()).strip(),

            "Day2":((response.xpath('//font/text()')[43].get()).strip()),
            "Weather_on_"+((response.xpath('//font/text()')[43].get()).strip()):(response.xpath('//font/text()')[46].get()).strip(),

            "Day3":((response.xpath('//font/text()')[47].get()).strip()),
            "Weather_on_"+((response.xpath('//font/text()')[47].get()).strip()):(response.xpath('//font/text()')[50].get()).strip(),

            "Day4":((response.xpath('//font/text()')[51].get()).strip()),
```



```python
class crawl(CrawlSpider):
    )
    # to filter use a::text type ka kuch to get all links use getall()
    def weather(self,response):
        yield{
            "URL":response.url,
            "Place":response.xpath('//font/text()')[0].get(),
            "MaxTemp":response.xpath('//font/text()')[5].get().strip(),
            "MinTemp":response.xpath('//font/text()')[11].get().strip(),
            "Rainfall(mm)":response.xpath('//font/text()')[17].get().strip(),
            "Relative_Humidity(17:30)":response.xpath('//font/text()')[21].get().strip(),
            "Sunset":response.xpath('//font/text()')[23].get().strip(),
            "Sunrise":response.xpath('//font/text()')[25].get().strip(),
            "Moonset":response.xpath('//font/text()')[27].get().strip(),
            "Moonrise":response.xpath('//font/text()')[29].get().strip(),
            "Weather_Today":(response.xpath('//font/text()')[38].get()).strip(),
            "Day1":((response.xpath('//font/text()')[39].get()).strip()),
            "Weather_on_"+((response.xpath('//font/text()')[39].get()).strip()):(response.xpath('//font/text()')[42].get()).strip(),

            "Day2":((response.xpath('//font/text()')[43].get()).strip()),
            "Weather_on_"+((response.xpath('//font/text()')[43].get()).strip()):(response.xpath('//font/text()')[46].get()).strip(),

            "Day3":((response.xpath('//font/text()')[47].get()).strip()),
            "Weather_on_"+((response.xpath('//font/text()')[47].get()).strip()):(response.xpath('//font/text()')[50].get()).strip(),

            "Day4":((response.xpath('//font/text()')[51].get()).strip()),
            "Weather_on_"+((response.xpath('//font/text()')[51].get()).strip()):(response.xpath('//font/text()')[54].get()).strip(),

            "Day5":((response.xpath('//font/text()')[55].get()).strip()),
            "Weather_on_"+((response.xpath('//font/text()')[55].get()).strip()):(response.xpath('//font/text()')[58].get()).strip(),

            "Day6":((response.xpath('//font/text()')[59].get()).strip()),
            "Weather_on_"+((response.xpath('//font/text()')[59].get()).strip()):(response.xpath('//font/text()')[62].get()).strip(),

        }
```

Government’s official website

"https://mausam.imd.gov.in/responsive/departmentalweb.php"

And its follow up webpages have been scraped using the ‘scrapy’ framework written in python, the classes imported were ‘CrawlSpider’, ‘Rule’ from module ‘spiders’ , ‘LinkExtractor’ from module ‘linkextractors’. A spider named ‘spider’ crawls over 1211 pages and gives a valid data of 175 districts of India.

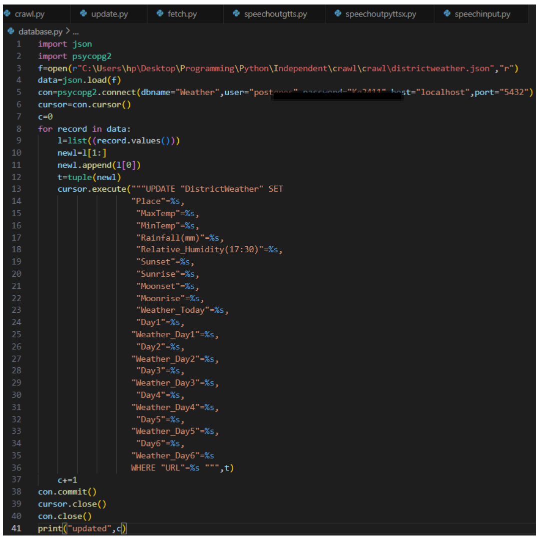
The data extracted is URL of webpage, District name , maximum, minimum temperature of the day, rainfall(in mm), relative humidity as on 17:30, time of sunrise, sunset , moonrise, moonset, and also weather forecast of next 6 days and its dates.

The above code is written under ‘spiders’ folder under default ‘crawl’ folder which is created with other python files when ‘scrapy’ is called for a project by ‘scrapy startproject crawl’.

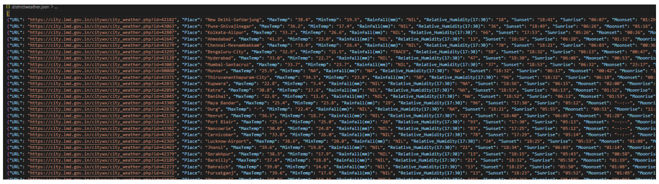
To run the above code , ‘scrapy crawl spider -o districtweather.json’ command is used.

After crawling the data is stored in ‘districtweather.json’ file as list of dictionaries.

SQL Connectivity



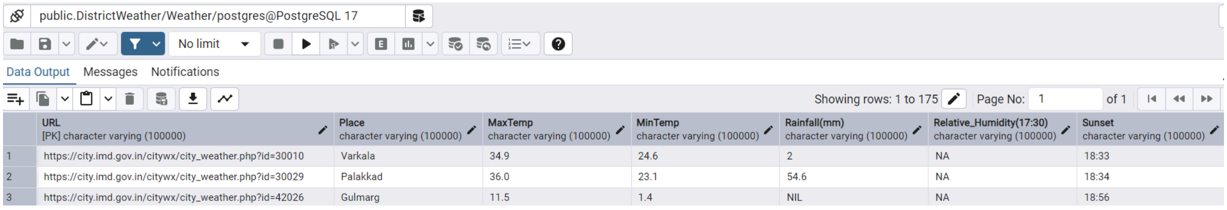
```
import json
import psycopg2
f=open(r"C:\Users\hp\Desktop\Programming\Python\Independent\crawl\crawl\districtweather.json","r")
data=json.load(f)
con=psycopg2.connect(dbname="Weather",user="post[redacted]st="localhost",port="5432")
cursor=con.cursor()
c=0
for record in data:
    l=list((record.values()))
    newl=l[1:]
    newl.append(l[0])
    t=tuple(newl)
    cursor.execute("""UPDATE "DistrictWeather" SET
                    "Place"=%s,
                     "MaxTemp"=%s,
                     "MinTemp"=%s,
                     "Rainfall(mm)"=%s,
                     "Relative_Humidity(17:30)"=%s,
                     "Sunset"=%s,
                     "Sunrise"=%s,
                     "Moonset"=%s,
                     "Moonrise"=%s,
                     "Weather_Today"=%s,
                     "Day1"=%s,
                    "Weather_Day1"=%s,
                     "Day2"=%s,
                    "Weather_Day2"=%s,
                     "Day3"=%s,
                    "Weather_Day3"=%s,
                     "Day4"=%s,
                    "Weather_Day4"=%s,
                     "Day5"=%s,
                    "Weather_Day5"=%s,
                     "Day6"=%s,
                    "Weather_Day6"=%s
                    WHERE "URL"=%s """,t)
    c+=1
con.commit()
cursor.close()
con.close()
print("updated",c)
```



```
[
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=42182", "Place": "New Delhi-Safdarjung", "MaxTemp": "38.4", "MinTemp": "19.5", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "18", "Sunset": "18:41", "Sunrise": "06:07", "Moonset": "01:29
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=43063", "Place": "Pune-Shivajinagar", "MaxTemp": "36.2", "MinTemp": "17.4", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "36", "Sunset": "18:49", "Sunrise": "06:26", "Moonset": "01:16",
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=42807", "Place": "Kolkata-Alipur", "MaxTemp": "33.2", "MinTemp": "26.6", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "66", "Sunset": "17:53", "Sunrise": "05:26", "Moonset": "00:26", "Mo
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=42647", "Place": "Ahmedabad", "MaxTemp": "41.3", "MinTemp": "23.0", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "13", "Sunset": "18:56", "Sunrise": "06:28", "Moonset": "01:32", "Moonris
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=43279", "Place": "Chennai-Meenambakkam", "MaxTemp": "33.9", "MinTemp": "26.4", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "70", "Sunset": "18:21", "Sunrise": "06:03", "Moonset": "00:36
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=43295", "Place": "Bengaluru-City", "MaxTemp": "32.9", "MinTemp": "21.5", "Rainfall(mm)": "TRACE", "Relative_Humidity(17:30)": "50", "Sunset": "18:32", "Sunrise": "06:13", "Moonset": "00:47", "
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=43128", "Place": "Hyderabad", "MaxTemp": "33.0", "MinTemp": "22.7", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "47", "Sunset": "18:30", "Sunrise": "06:08", "Moonset": "00:53", "Moonris
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=43003", "Place": "Mumbai-Santacruz", "MaxTemp": "33.7", "MinTemp": "23.7", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "37", "Sunset": "18:53", "Sunrise": "06:32", "Moonset": "23:17", "
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=90060", "Place": "Munnar", "MaxTemp": "25.9", "MinTemp": "NA", "Rainfall(mm)": "NA", "Relative_Humidity(17:30)": "NA", "Sunset": "18:32", "Sunrise": "06:17", "Moonset": "00:42", "Moonrise": "1
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=43371", "Place": "Thiruvananthapuram-City", "MaxTemp": "34.3", "MinTemp": "23.8", "Rainfall(mm)": "50", "Relative_Humidity(17:30)": "96", "Sunset": "18:32", "Sunrise": "06:18", "Moonset": "00:
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=42031", "Place": "Kupwara", "MaxTemp": "22.8", "MinTemp": "3.4", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "NA", "Sunset": "18:56", "Sunrise": "06:15", "Moonset": "02:00", "Moonrise":
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=42054", "Place": "Katra", "MaxTemp": "30.8", "MinTemp": "17.6", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "NA", "Sunset": "18:53", "Sunrise": "06:13", "Moonset": "01:52", "Moonrise":
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=42045", "Place": "Banihal", "MaxTemp": "22.8", "MinTemp": "11.6", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "NA", "Sunset": "18:52", "Sunrise": "06:12", "Moonset": "01:53", "Moonrise"
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=43309", "Place": "Maya Bandar", "MaxTemp": "25.4", "MinTemp": "23.8", "Rainfall(mm)": "29", "Relative_Humidity(17:30)": "96", "Sunset": "17:30", "Sunrise": "05:12", "Moonset": "--:--", "Moonri
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=42876", "Place": "Durg", "MaxTemp": "-", "MinTemp": "22.4", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "NA", "Sunset": "18:21", "Sunrise": "05:55", "Moonset": "00:51", "Moonrise": "11:
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=42139", "Place": "Meerut", "MaxTemp": "36.5", "MinTemp": "18.7", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "21", "Sunset": "18:40", "Sunrise": "06:05", "Moonset": "01:28", "Moonrise":
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=43333", "Place": "Port Blair", "MaxTemp": "25.6", "MinTemp": "25.0", "Rainfall(mm)": "26", "Relative_Humidity(17:30)": "93", "Sunset": "17:30", "Sunrise": "05:13", "Moonset": "--:--", "Moonris
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=43382", "Place": "Nancowrie", "MaxTemp": "30.0", "MinTemp": "24.8", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "83", "Sunset": "17:25", "Sunrise": "05:12", "Moonset": "--:--", "Moonris
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=43367", "Place": "Carnicobar", "MaxTemp": "33.0", "MinTemp": "26.0", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "78", "Sunset": "17:29", "Sunrise": "05:14", "Moonset": "--:--", "Moonri
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=42369", "Place": "Lucknow-Airport", "MaxTemp": "38.6", "MinTemp": "20.0", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "24", "Sunset": "18:25", "Sunrise": "05:53", "Moonset": "01:08", "M
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=42463", "Place": "Jhansi", "MaxTemp": "39.6", "MinTemp": "19.8", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "21", "Sunset": "18:34", "Sunrise": "06:03", "Moonset": "01:14", "Moonrise":
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=42379", "Place": "Gorakhpur", "MaxTemp": "38.3", "MinTemp": "17.9", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "13", "Sunset": "18:15", "Sunrise": "05:43", "Moonset": "00:58", "Moonris
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=42189", "Place": "Bareilly", "MaxTemp": "37.4", "MinTemp": "18.8", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "21", "Sunset": "18:32", "Sunrise": "05:58", "Moonset": "01:19", "Moonrise
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=42273", "Place": "Bahraich", "MaxTemp": "39.0", "MinTemp": "14.6", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "13", "Sunset": "18:23", "Sunrise": "05:50", "Moonset": "01:08", "Moonrise
{"URL": "https://city.imd.gov.in/citywx/city_weather.php?id=42372", "Place": "Fursatganj", "MaxTemp": "39.6", "MinTemp": "17.6", "Rainfall(mm)": "NIL", "Relative_Humidity(17:30)": "13", "Sunset": "18:23", "Sunrise": "05:52", "Moonset": "01:05", "Moonri
```

Update database code

districtweather.json(175 rows)



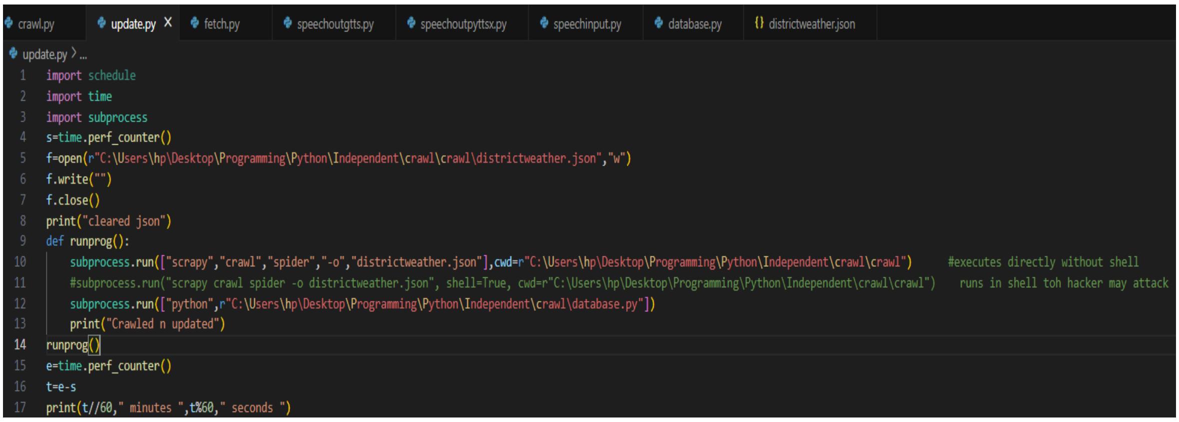
	URL [PK] character varying (100000)	Place character varying (100000)	MaxTemp character varying (100000)	MinTemp character varying (100000)	Rainfall(mm) character varying (100000)	Relative_Humidity(17:30) character varying (100000)	Sunset character varying (100000)
1	https://city.imd.gov.in/citywx/city_weather.php?id=30010	Varkala	34.9	24.6	2	NA	18:33
2	https://city.imd.gov.in/citywx/city_weather.php?id=30029	Palakkad	36.0	23.1	54.6	NA	18:34
3	https://city.imd.gov.in/citywx/city_weather.php?id=42026	Gulmarg	11.5	1.4	NIL	NA	18:56

Database of PostgresSQL (175 rows , 23 cols)

Initially a table 'DistrictWeather' is created under 'Weather' database with 23 columns mentioned above.

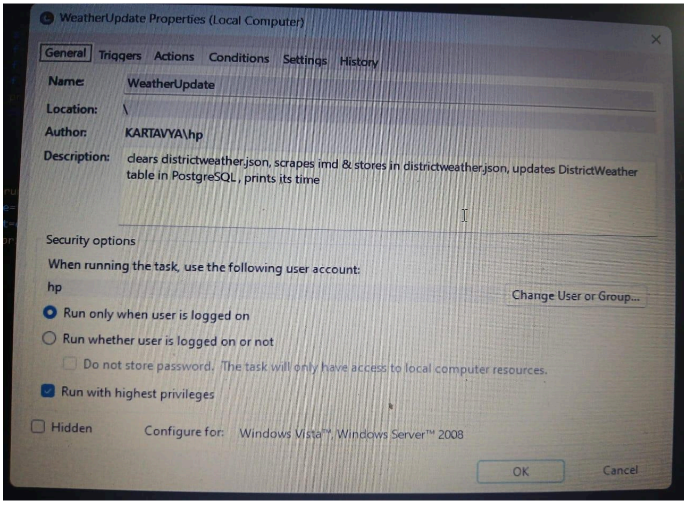
In database.py , json, psycopg2 modules are imported , the data from districtweather.json is loaded and by the help of postgres connectivity using psycopg2 , the data of all 22 columns is updated for all rows using the identification through 'URL' as primary key , using tuple(converting explicitly from list) as argument for each field in PostgresSQL using a loop.

Update Process and Task Scheduler



```python
import schedule
import time
import subprocess
s=time.perf_counter()
f=open(r"C:\Users\hp\Desktop\Programming\Python\Independent\crawl\crawl\districtweather.json","w")
f.write("")
f.close()
print("cleared json")
def runprog():
    subprocess.run(["scrapy","crawl","spider","-o","districtweather.json"],cwd=r"C:\Users\hp\Desktop\Programming\Python\Independent\crawl\crawl")    #executes directly without shell
    #subprocess.run("scrapy crawl spider -o districtweather.json", shell=True, cwd=r"C:\Users\hp\Desktop\Programming\Python\Independent\crawl\crawl")   runs in shell toh hacker may attack
    subprocess.run(["python",r"C:\Users\hp\Desktop\Programming\Python\Independent\crawl\database.py"])
    print("Crawled n updated")
runprog()
e=time.perf_counter()
t=e-s
print(t//60," minutes ",t%60," seconds ")
```

update.py



Name: WeatherUpdate

Location: \

Author: KARTAVYA\hp

Description: clears districtweather.json, scrapes imd & stores in districtweather.json, updates DistrictWeather table in PostgreSQL , prints its time

Security options

When running the task, use the following user account:

hp

● Run only when user is logged on

○ Run whether user is logged on or not

□ Do not store password. The task will only have access to local computer resources.

☑ Run with highest privileges

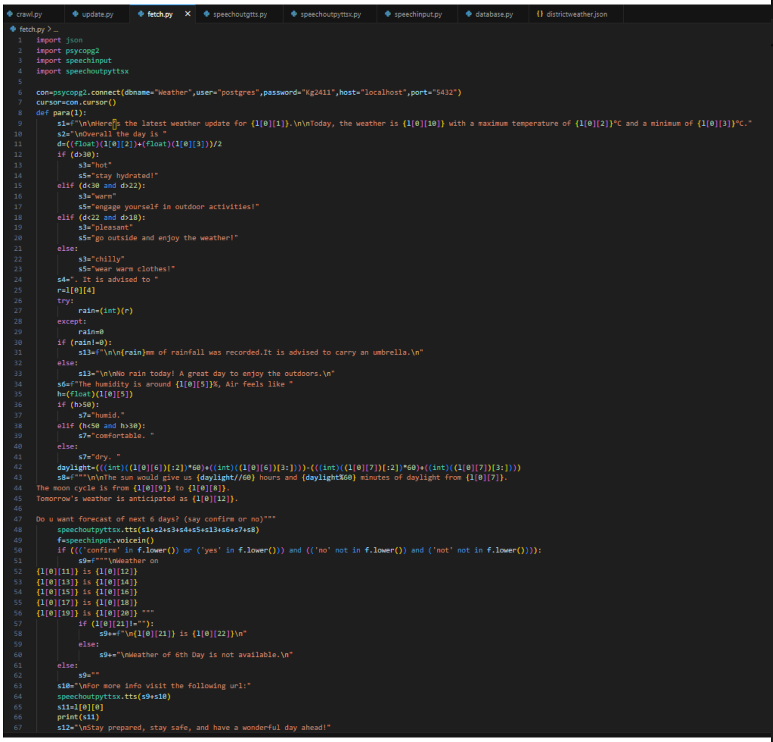
□ Hidden   Configure for: Windows Vista™, Windows Server™ 2008

Task Scheduler

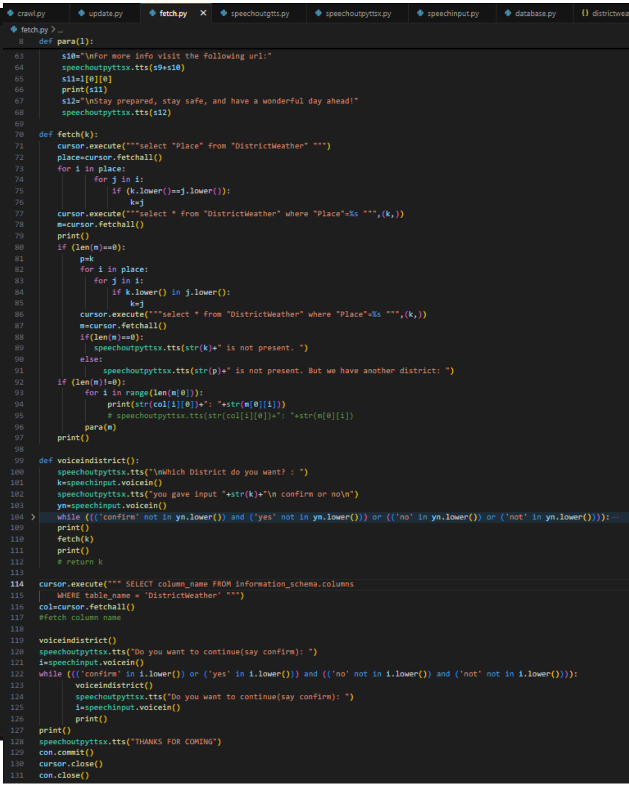
In update.py , ‘schedule’,’time’,’subprocess’ modules are imported. This code overwrites districtweather.json with empty string to have new data after crawling, the function runprog() runs the crawl.py at first, then database.py . it also gives the final execution time.

The update.py is scheduled to run everyday at 11:00 through Task Scheduler , if the task is missed (a constraint that internet connection is must or other) , it would run with highest privilege.

Fetch Process

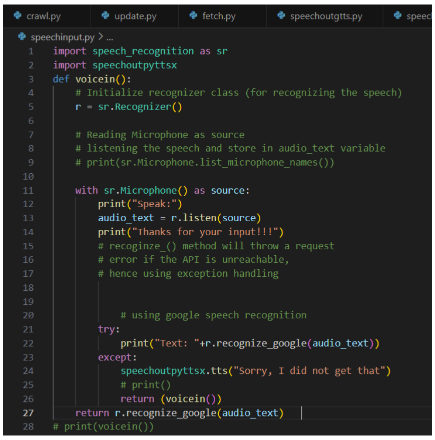


```python
import json
import psycopg2
import speechinput
import speechoutpyttsx

con=psycopg2.connect(dbname="Weather",user="postgres",password="Kg2411",host="localhost",port="5432")
cursor=con.cursor()
def para(l):
    s1=f"\n\nHere's the latest weather update for {l[0][1]}.\n\nToday, the weather is {l[0][10]} with a maximum temperature of {l[0][2]}°C and a minimum of {l[0][3]}°C."
    s2="\nOverall the day is "
    d=((float)(l[0][2])+(float)(l[0][3]))/2
    if (d>30):
        s3="hot"
        s5="stay hydrated!"
    elif (d<30 and d>22):
        s3="warm"
        s5="engage yourself in outdoor activities!"
    elif (d<22 and d>18):
        s3="pleasant"
        s5="go outside and enjoy the weather!"
    else:
        s3="chilly"
        s5="wear warm clothes!"
    s4=". It is advised to "
    r=l[0][4]
    try:
        rain=(int)(r)
    except:
        rain=0
    if (rain!=0):
        s13=f"\n\n{rain}mm of rainfall was recorded.It is advised to carry an umbrella.\n"
    else:
        s13="\n\nNo rain today! A great day to enjoy the outdoors.\n"
    s6=f"The humidity is around {l[0][5]}%, Air feels like "
    h=(float)(l[0][5])
    if (h>50):
        s7="humid."
    elif (h<50 and h>30):
        s7="comfortable. "
    else:
        s7="dry. "
    daylight=(((int)((l[0][6])[:2])*60)+((int)((l[0][6])[3:])))-(((int)((l[0][7])[:2])*60)+((int)((l[0][7])[3:])))
    s8=f"""\n\nThe sun would give us {daylight//60} hours and {daylight%60} minutes of daylight from {l[0][7]}.
The moon cycle is from {l[0][9]} to {l[0][8]}.
Tomorrow's weather is anticipated as {l[0][12]}.

Do u want forecast of next 6 days? (say confirm or no)"""
    speechoutpyttsx.tts(s1+s2+s3+s4+s5+s13+s6+s7+s8)
    f=speechinput.voicein()
    if ((('confirm' in f.lower()) or ('yes' in f.lower())) and (('no' not in f.lower()) and ('not' not in f.lower()))):
        s9=f"""\nWeather on
{l[0][11]} is {l[0][12]}
{l[0][13]} is {l[0][14]}
{l[0][15]} is {l[0][16]}
{l[0][17]} is {l[0][18]}
{l[0][19]} is {l[0][20]} """
        if (l[0][21]!=""):
            s9+=f"\n{l[0][21]} is {l[0][22]}\n"
        else:
            s9+="\nWeather of 6th Day is not available.\n"
    else:
        s9=""
    s10="\nFor more info visit the following url:"
    speechoutpyttsx.tts(s9+s10)
    s11=l[0][0]
    print(s11)
    s12="\nStay prepared, stay safe, and have a wonderful day ahead!"
```

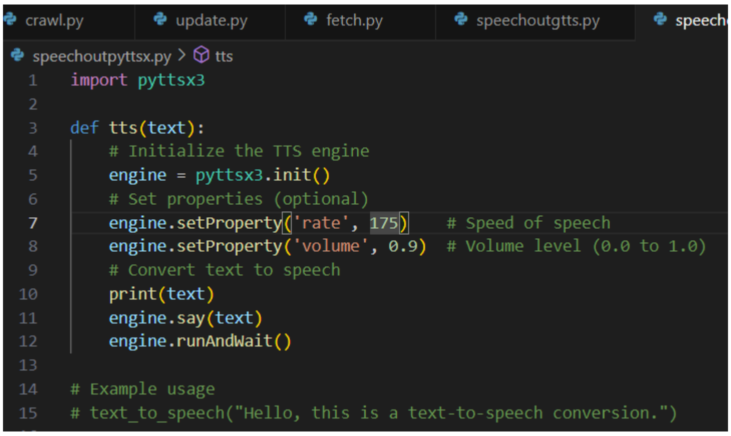


```python
def para(l):
    s10="\nFor more info visit the following url:"
    speechoutpyttsx.tts(s9+s10)
    s11=l[0][0]
    print(s11)
    s12="\nStay prepared, stay safe, and have a wonderful day ahead!"
    speechoutpyttsx.tts(s12)

def fetch(k):
    cursor.execute("""select "Place" from "DistrictWeather" """)
    place=cursor.fetchall()
    for i in place:
            for j in i:
                if (k.lower()==j.lower()):
                    k=j
    cursor.execute("""select * from "DistrictWeather" where "Place"=%s """,(k,))
    m=cursor.fetchall()
    print()
    if (len(m)==0):
        p=k
        for i in place:
            for j in i:
                if k.lower() in j.lower():
                    k=j
        cursor.execute("""select * from "DistrictWeather" where "Place"=%s """,(k,))
        m=cursor.fetchall()
        if(len(m)==0):
            speechoutpyttsx.tts(str(k)+" is not present. ")
        else:
             speechoutpyttsx.tts(str(p)+" is not present. But we have another district: ")
    if (len(m)!=0):
         for i in range(len(m[0])):
             print(str(col[i][0])+": "+str(m[0][i]))
             # speechoutpyttsx.tts(str(col[i][0])+": "+str(m[0][i])
         para(m)
    print()

def voiceindistrict():
    speechoutpyttsx.tts("\nWhich District do you want? : ")
    k=speechinput.voicein()
    speechoutpyttsx.tts("you gave input "+str(k)+"\n confirm or no\n")
    yn=speechinput.voicein()
    while ((('confirm' not in yn.lower()) and ('yes' not in yn.lower())) or (('no' in yn.lower()) or ('not' in yn.lower()))):···
    print()
    fetch(k)
    print()
    # return k

cursor.execute(""" SELECT column_name FROM information_schema.columns
    WHERE table_name = 'DistrictWeather' """)
col=cursor.fetchall()
#fetch column name

voiceindistrict()
speechoutpyttsx.tts("Do you want to continue(say confirm): ")
i=speechinput.voicein()
while ((('confirm' in i.lower()) or ('yes' in i.lower())) and (('no' not in i.lower()) and ('not' not in i.lower()))):
        voiceindistrict()
        speechoutpyttsx.tts("Do you want to continue(say confirm): ")
        i=speechinput.voicein()
        print()
print()
speechoutpyttsx.tts("THANKS FOR COMING")
con.commit()
cursor.close()
con.close()
```

fetch.py



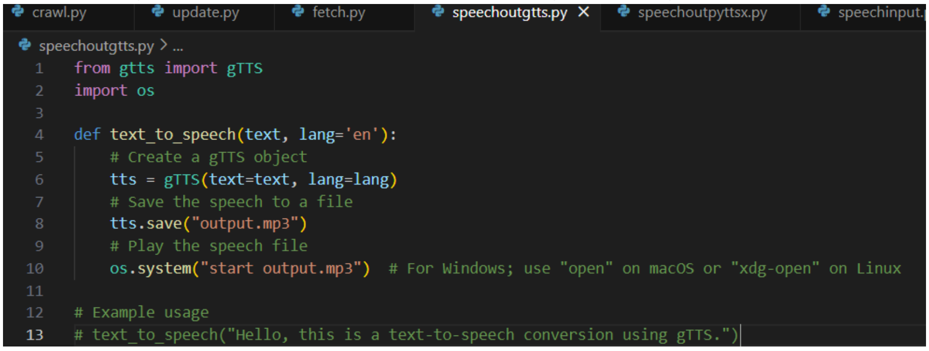
```python
import speech_recognition as sr
import speechoutpyttsx
def voicein():
    # Initialize recognizer class (for recognizing the speech)
    r = sr.Recognizer()

    # Reading Microphone as source
    # listening the speech and store in audio_text variable
    # print(sr.Microphone.list_microphone_names())

    with sr.Microphone() as source:
        print("Speak:")
        audio_text = r.listen(source)
        print("Thanks for your input!!!")
        # recoginze_() method will throw a request
        # error if the API is unreachable,
        # hence using exception handling


            # using google speech recognition
        try:
            print("Text: "+r.recognize_google(audio_text))
        except:
            speechoutpyttsx.tts("Sorry, I did not get that")
            # print()
            return (voicein())
    return r.recognize_google(audio_text)
# print(voicein())
```



```python
import pyttsx3

def tts(text):
    # Initialize the TTS engine
    engine = pyttsx3.init()
    # Set properties (optional)
    engine.setProperty('rate', 175)    # Speed of speech
    engine.setProperty('volume', 0.9)  # Volume level (0.0 to 1.0)
    # Convert text to speech
    print(text)
    engine.say(text)
    engine.runAndWait()

# Example usage
# text_to_speech("Hello, this is a text-to-speech conversion.")
```

speechinput.py

speechoutpyttsx.py



```python
from gtts import gTTS
import os

def text_to_speech(text, lang='en'):
    # Create a gTTS object
    tts = gTTS(text=text, lang=lang)
    # Save the speech to a file
    tts.save("output.mp3")
    # Play the speech file
    os.system("start output.mp3")  # For Windows; use "open" on macOS or "xdg-open" on Linux

# Example usage
# text_to_speech("Hello, this is a text-to-speech conversion using gTTS.")
```

speechoutgtts.py

Fetch Process

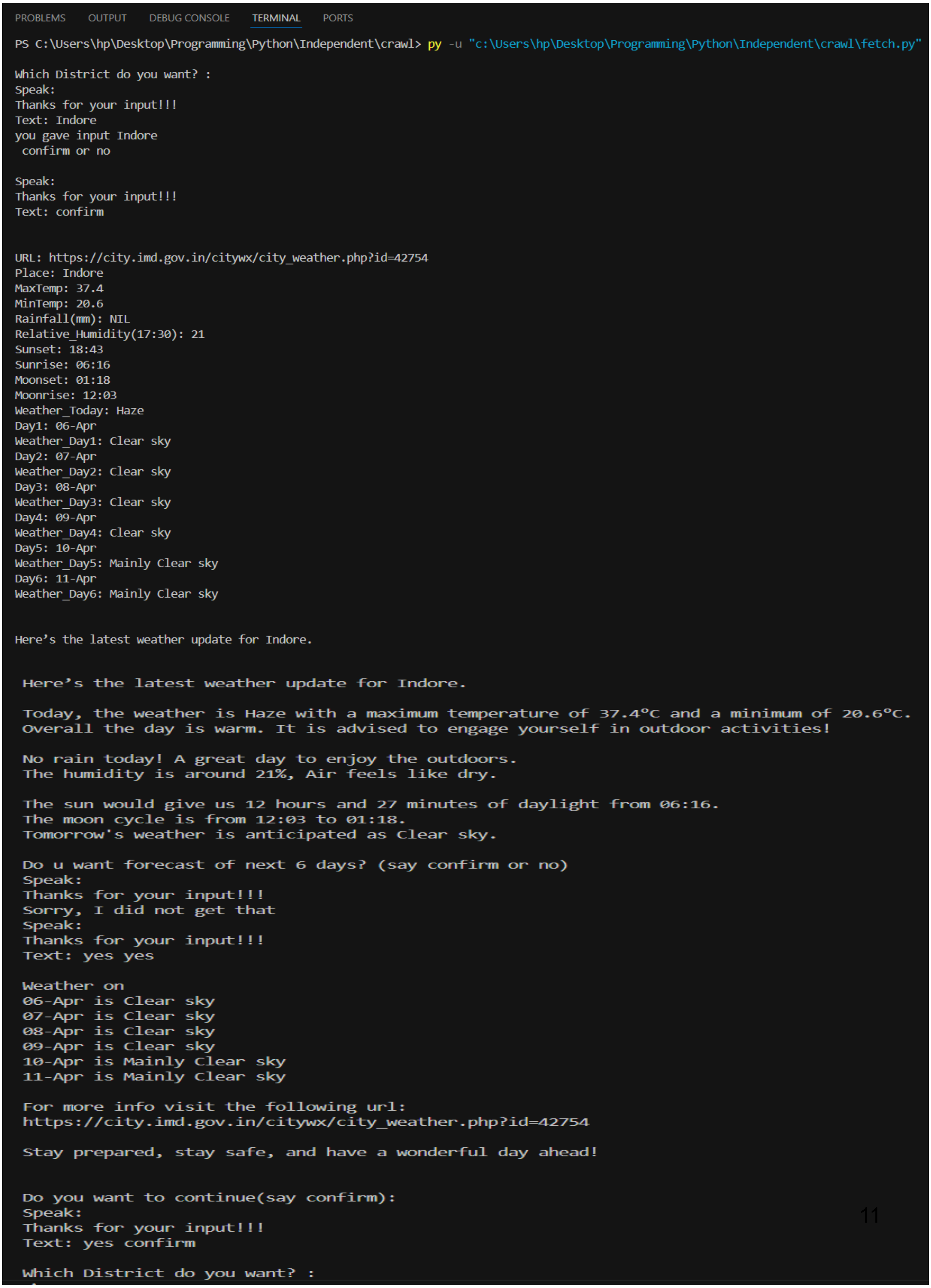
In fetch.py , ‘psycopg2’ module, ‘speechinput’ , ‘speechoutpyttsx’ python files are imported.
In **speechoutpyttsx.py** , ‘pyttsx3’ module is imported which runs the speech engines locally, **tts()** function prints the text and reads it aloud whose speed, volume can be adjusted.
In speechinput.py ‘speech_recognition’ module is imported to recognise the audio input given by the user , ‘speechoutpyttsx’ python file is imported. **voicein()** function takes input audio and recognises the audio through the recognizer. If it didn’t understand a speech message is given , and recursion takes place , at last the audio input text is returned.
In speechoutgtts.py , ‘gTTs’ class from ‘gtts’ module ,‘os’ are imported the text_to_speech() function gives audio output but requires an internet connection.

In fetch.py , which is the user interface ,
voiceindistrict() function takes speech input of district by asking in speech , also verifies the input from user , and runs a loop of input until verified, when verified, it calls fetch() function by giving input as argument.
fetch() function runs a query to get list of all places in table, another query to get list of all values associated to that place, if not found it checks the district as substring in ‘Place’ and get values of that district, else gives speech output as not found. If found, it prints the column_name with it corresponding values and it calls para() function giving the list of data as argument.
para() function gives the forecast as speech output including some advises based on if-else statements, also asks if forecast of next 6 days is needed through speech and takes input through speech, if yes then gives forecast of 6 days in speech.

Firstly, a connection to postgresSQL is established, then a query is run to get all column names of the table, then voiceindistrict() function is called which executes the above explanation. After its execution , it asks for continue through speech and takes input in speech, if yes then voiceindistrict() is again called otherwise , the program ends by greeting ‘Thanks’. All speech process through tts(),voicein(). If a positive and negative input(ex. “yes no”, “not confirm”, “yes not”, “yes no not confirm”) are given then negative (“no”) dominates.

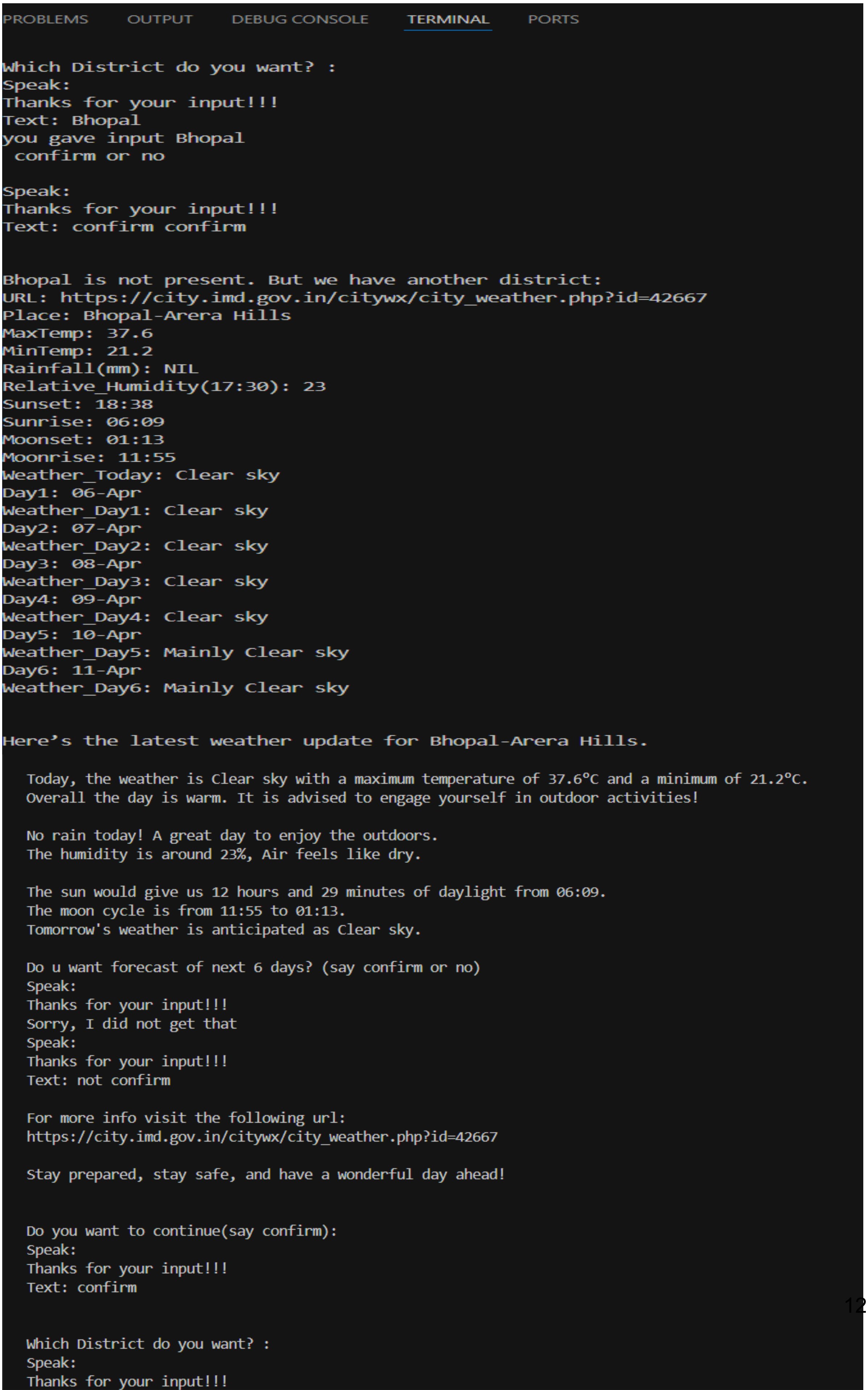
RESULT / OUTPUT

Case 1:
General input , giving 6 days forecast too



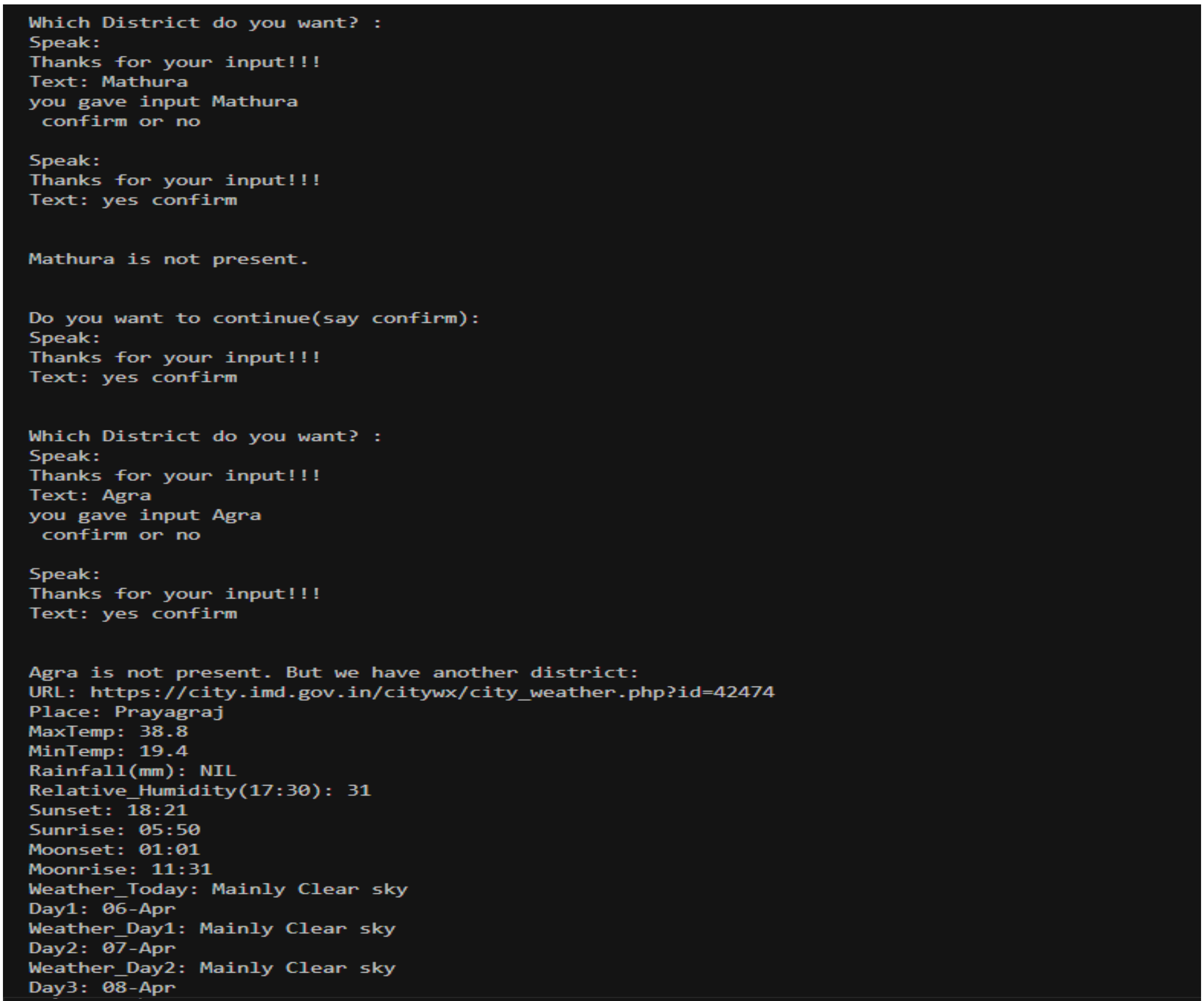
```
PS C:\Users\hp\Desktop\Programming\Python\Independent\crawl> py -u "c:\Users\hp\Desktop\Programming\Python\Independent\crawl\fetch.py"

Which District do you want? :
Speak:
Thanks for your input!!!
Text: Indore
you gave input Indore
 confirm or no

Speak:
Thanks for your input!!!
Text: confirm


URL: https://city.imd.gov.in/citywx/city_weather.php?id=42754
Place: Indore
MaxTemp: 37.4
MinTemp: 20.6
Rainfall(mm): NIL
Relative_Humidity(17:30): 21
Sunset: 18:43
Sunrise: 06:16
Moonset: 01:18
Moonrise: 12:03
Weather_Today: Haze
Day1: 06-Apr
Weather_Day1: Clear sky
Day2: 07-Apr
Weather_Day2: Clear sky
Day3: 08-Apr
Weather_Day3: Clear sky
Day4: 09-Apr
Weather_Day4: Clear sky
Day5: 10-Apr
Weather_Day5: Mainly Clear sky
Day6: 11-Apr
Weather_Day6: Mainly Clear sky


Here's the latest weather update for Indore.


 Here's the latest weather update for Indore.

 Today, the weather is Haze with a maximum temperature of 37.4°C and a minimum of 20.6°C.
 Overall the day is warm. It is advised to engage yourself in outdoor activities!

 No rain today! A great day to enjoy the outdoors.
 The humidity is around 21%, Air feels like dry.

 The sun would give us 12 hours and 27 minutes of daylight from 06:16.
 The moon cycle is from 12:03 to 01:18.
 Tomorrow's weather is anticipated as Clear sky.

 Do u want forecast of next 6 days? (say confirm or no)
 Speak:
 Thanks for your input!!!
 Sorry, I did not get that
 Speak:
 Thanks for your input!!!
 Text: yes yes

 Weather on
 06-Apr is Clear sky
 07-Apr is Clear sky
 08-Apr is Clear sky
 09-Apr is Clear sky
 10-Apr is Mainly Clear sky
 11-Apr is Mainly Clear sky

 For more info visit the following url:
 https://city.imd.gov.in/citywx/city_weather.php?id=42754

 Stay prepared, stay safe, and have a wonderful day ahead!


 Do you want to continue(say confirm):
 Speak:
 Thanks for your input!!!
 Text: yes confirm

 Which District do you want? :
```

Case 2:
Desired place is present but including other areas, not giving 6 days forecast

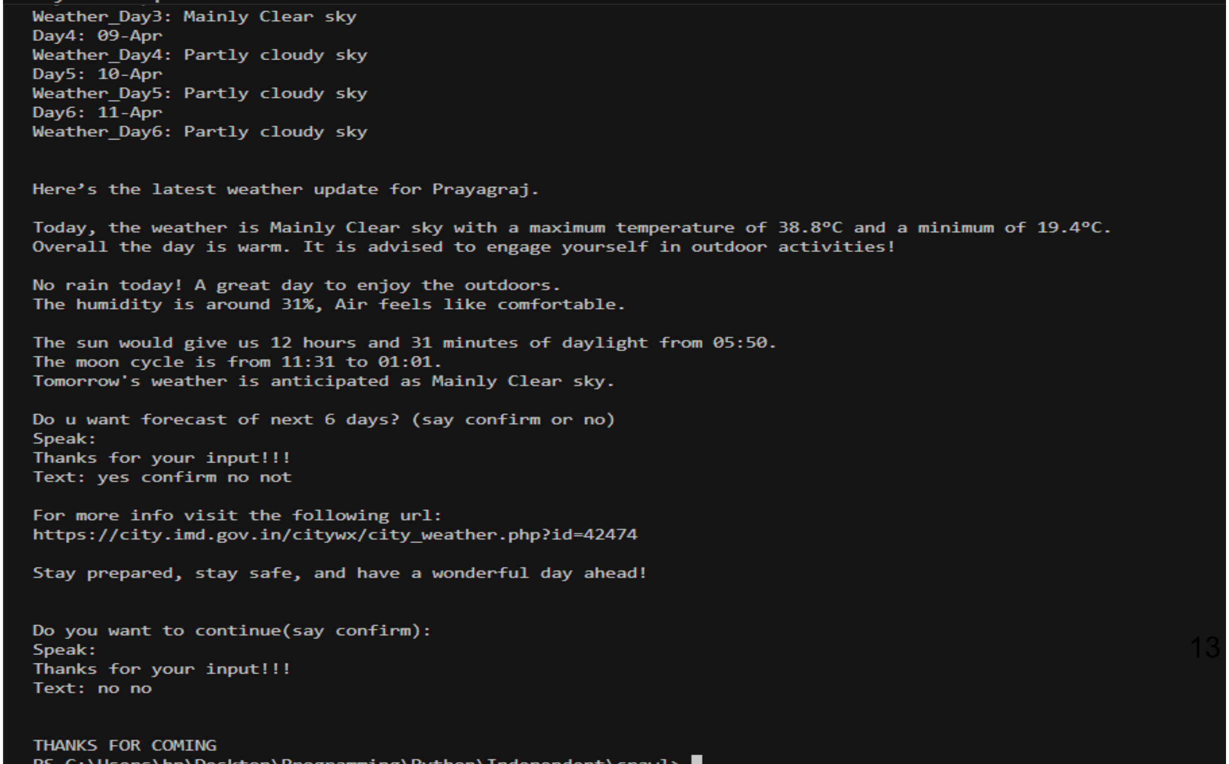


```
Which District do you want? :
Speak:
Thanks for your input!!!
Text: Bhopal
you gave input Bhopal
 confirm or no

Speak:
Thanks for your input!!!
Text: confirm confirm


Bhopal is not present. But we have another district:
URL: https://city.imd.gov.in/citywx/city_weather.php?id=42667
Place: Bhopal-Arera Hills
MaxTemp: 37.6
MinTemp: 21.2
Rainfall(mm): NIL
Relative_Humidity(17:30): 23
Sunset: 18:38
Sunrise: 06:09
Moonset: 01:13
Moonrise: 11:55
Weather_Today: Clear sky
Day1: 06-Apr
Weather_Day1: Clear sky
Day2: 07-Apr
Weather_Day2: Clear sky
Day3: 08-Apr
Weather_Day3: Clear sky
Day4: 09-Apr
Weather_Day4: Clear sky
Day5: 10-Apr
Weather_Day5: Mainly Clear sky
Day6: 11-Apr
Weather_Day6: Mainly Clear sky


Here’s the latest weather update for Bhopal-Arera Hills.

  Today, the weather is Clear sky with a maximum temperature of 37.6°C and a minimum of 21.2°C.
  Overall the day is warm. It is advised to engage yourself in outdoor activities!

  No rain today! A great day to enjoy the outdoors.
  The humidity is around 23%, Air feels like dry.

  The sun would give us 12 hours and 29 minutes of daylight from 06:09.
  The moon cycle is from 11:55 to 01:13.
  Tomorrow's weather is anticipated as Clear sky.

  Do u want forecast of next 6 days? (say confirm or no)
  Speak:
  Thanks for your input!!!
  Sorry, I did not get that
  Speak:
  Thanks for your input!!!
  Text: not confirm

  For more info visit the following url:
  https://city.imd.gov.in/citywx/city_weather.php?id=42667

  Stay prepared, stay safe, and have a wonderful day ahead!


  Do you want to continue(say confirm):
  Speak:
  Thanks for your input!!!
  Text: confirm


  Which District do you want? :
  Speak:
  Thanks for your input!!!
```

Case 3:
Desired place is not present even not as substring.
Case 4:
Desired place is not present but is present as substring.
Case 5:
Don’t want to continue



```
Which District do you want? :
Speak:
Thanks for your input!!!
Text: Mathura
you gave input Mathura
 confirm or no

Speak:
Thanks for your input!!!
Text: yes confirm


Mathura is not present.


Do you want to continue(say confirm):
Speak:
Thanks for your input!!!
Text: yes confirm


Which District do you want? :
Speak:
Thanks for your input!!!
Text: Agra
you gave input Agra
 confirm or no

Speak:
Thanks for your input!!!
Text: yes confirm


Agra is not present. But we have another district:
URL: https://city.imd.gov.in/citywx/city_weather.php?id=42474
Place: Prayagraj
MaxTemp: 38.8
MinTemp: 19.4
Rainfall(mm): NIL
Relative_Humidity(17:30): 31
Sunset: 18:21
Sunrise: 05:50
Moonset: 01:01
Moonrise: 11:31
Weather_Today: Mainly Clear sky
Day1: 06-Apr
Weather_Day1: Mainly Clear sky
Day2: 07-Apr
Weather_Day2: Mainly Clear sky
Day3: 08-Apr
```



```
Weather_Day3: Mainly Clear sky
Day4: 09-Apr
Weather_Day4: Partly cloudy sky
Day5: 10-Apr
Weather_Day5: Partly cloudy sky
Day6: 11-Apr
Weather_Day6: Partly cloudy sky


Here’s the latest weather update for Prayagraj.

Today, the weather is Mainly Clear sky with a maximum temperature of 38.8°C and a minimum of 19.4°C.
Overall the day is warm. It is advised to engage yourself in outdoor activities!

No rain today! A great day to enjoy the outdoors.
The humidity is around 31%, Air feels like comfortable.

The sun would give us 12 hours and 31 minutes of daylight from 05:50.
The moon cycle is from 11:31 to 01:01.
Tomorrow's weather is anticipated as Mainly Clear sky.

Do u want forecast of next 6 days? (say confirm or no)
Speak:
Thanks for your input!!!
Text: yes confirm no not

For more info visit the following url:
https://city.imd.gov.in/citywx/city_weather.php?id=42474

Stay prepared, stay safe, and have a wonderful day ahead!


Do you want to continue(say confirm):
Speak:
Thanks for your input!!!
Text: no no


THANKS FOR COMING
PS C:\Users\hp\Desktop\Programming\Python\Independent\cncvl>
```

Limitations and Future Scope

1. Data of only 175 districts is present. Output can be optimised by taking data of nearby districts present in the database by calculating minimum distance through latitudes and longitudes.
2. Data is scraped everyday and updated. In Future, it can be enhanced by directly taking data of the particular place entered by the user by web scraping.
3. Unnecessary pages crawling can be ignored to reduce the execution time of scraping.
4. More advanced websites can be used to extract data.
5. Mainly, a UI can be made for the speech bot.
6. Other features such as a suggestion bot can be implanted.
7. The forecast of all days , everyday can be stored to

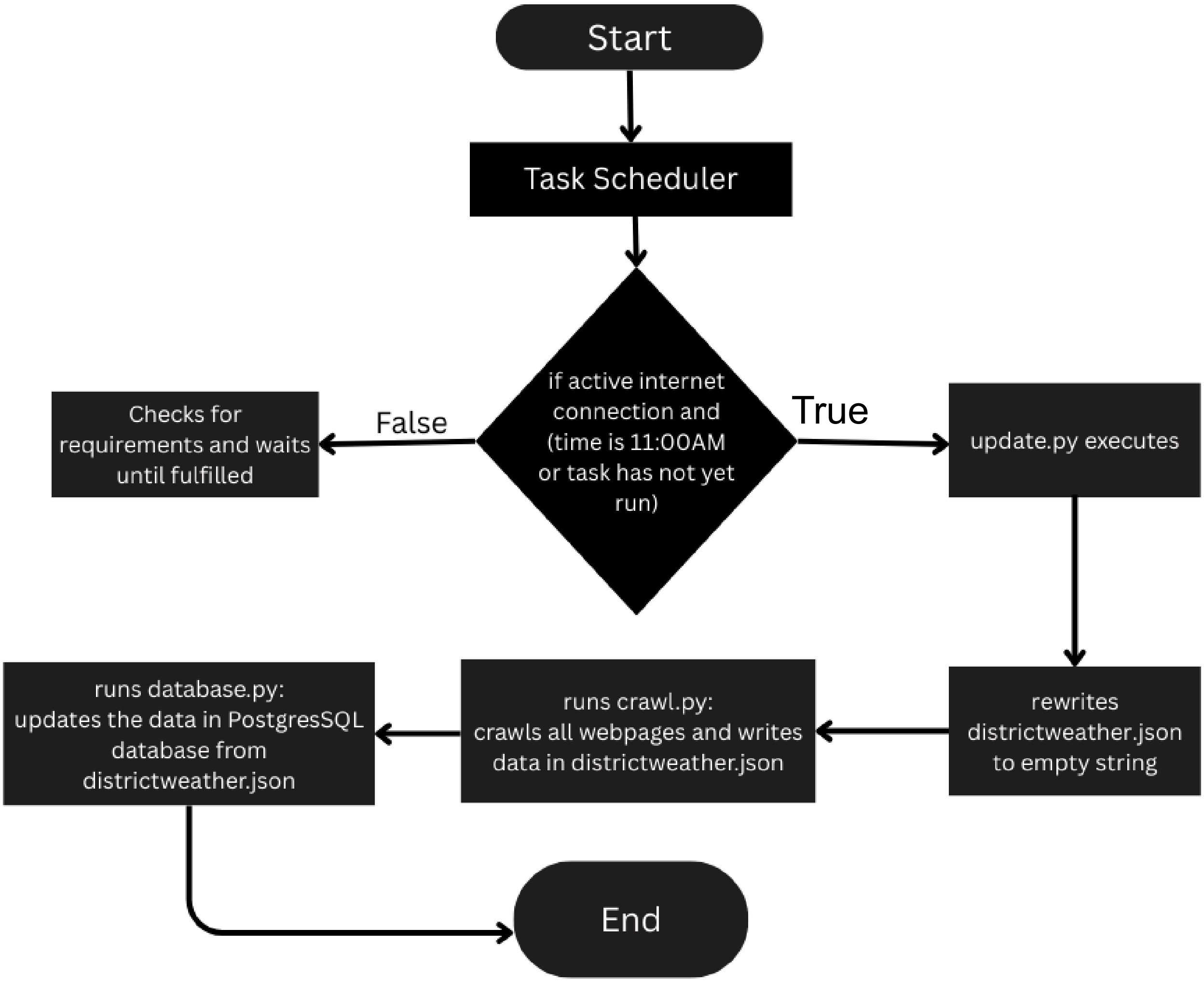
   a) check the accuracy of the forecast of the website.

   b) use NLP and predict the weather for more days, thus, can make our own weather forecaster.

   c) show the variation in graphical forms.

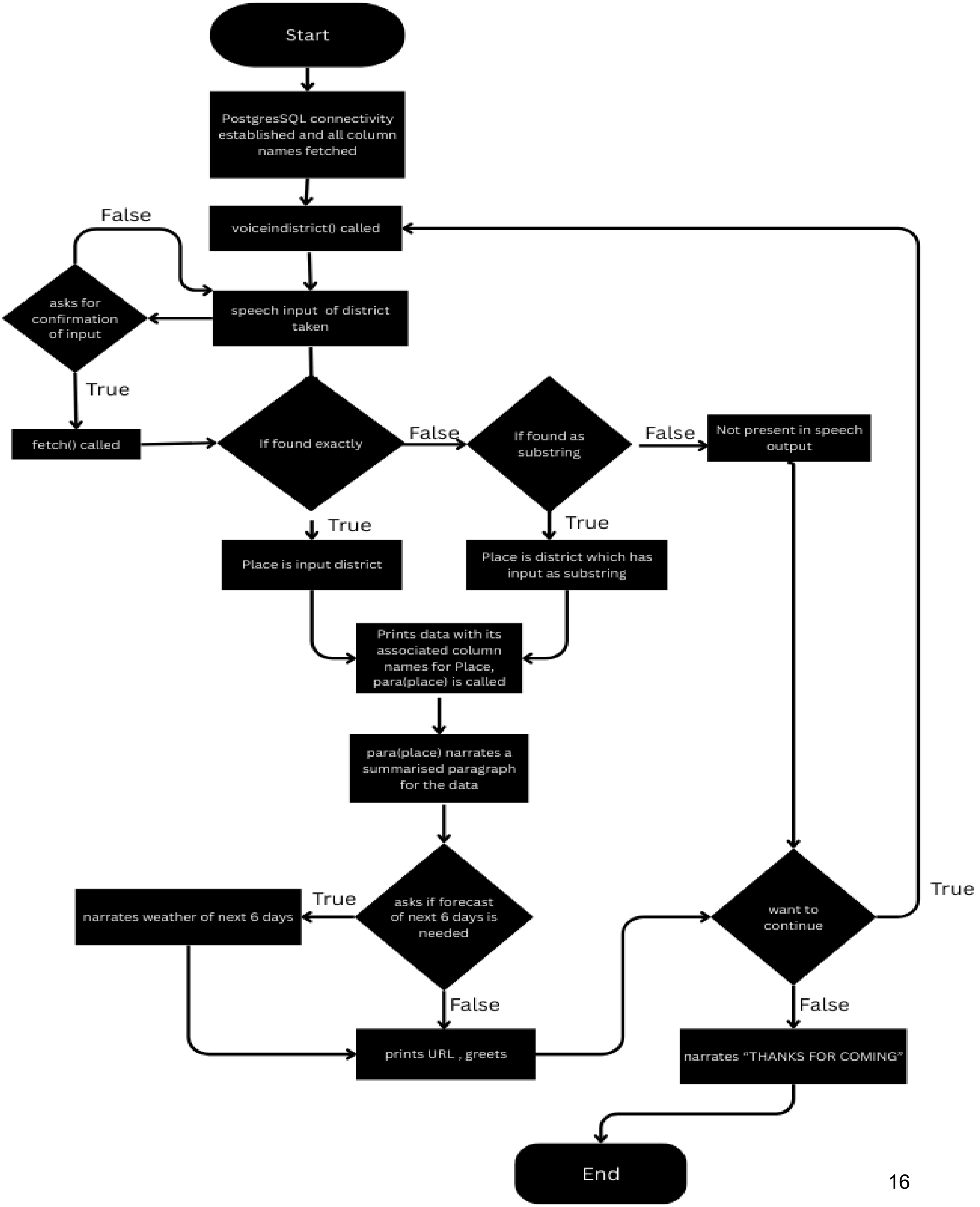
Flow and Conclusion

Task Scheduler:


Start
Task Scheduler
if active internet connection and (time is 11:00AM or task has not yet run)
False
Checks for requirements and waits until fulfilled
True
update.py executes
rewrites districtweather.json to empty string
runs crawl.py: crawls all webpages and writes data in districtweather.json
runs database.py: updates the data in PostgresSQL database from districtweather.json
End

Flow and Conclusion

MAIN PROCESS , fetch.py():



Start
PostgresSQL connectivity established and all column names fetched
voiceindistrict() called
False
asks for confirmation of input
speech input of district taken
True
fetch() called
If found exactly
False
If found as substring
False
Not present in speech output
True
Place is input district
True
Place is district which has input as substring
Prints data with its associated column names for Place, para(place) is called
para(place) narrates a summarised paragraph for the data
narrates weather of next 6 days
True
asks if forecast of next 6 days is needed
False
prints URL , greets
want to continue
True
False
narrates "THANKS FOR COMING"
End

Flow and Conclusion

1. A web scraper scrapes the webpages of government website daily.
2. The data is updated in database of PostgresSQL.
3. The data is fetched through input taken in speech and gives all data pertaining to that place in speech(if present).
4. This speech bot makes weather forecast easily accessible to the users.

Bibliography

- Coding Web Crawler in Python with Scrapy (https://youtu.be/m_3gjHGxIJc?si=YpS6WGfazcJLyd0d)
- https://chatgpt.com/c/67e52d37-35a4-8002-ad7f-f672735d8682
- https://chatgpt.com/c/67dd3cfd-392c-8002-8c9c-c7a5be69ce49
- https://chatgpt.com/c/67e6b426-24f0-8002-8e2f-0e1ea5abd253
- https://chatgpt.com/c/67e7a41d-f44c-8002-884a-7b7a3f0ab7f8
- https://chatgpt.com/c/67e9210a-690c-8002-a3e8-88aa038425d0
- https://www.geeksforgeeks.org/python-speech-recognition-module/
- https://www.geeksforgeeks.org/python-text-to-speech-by-using-pyttsx3/