**Python Code**

**Team Members:**

M. Sanjay-AP23110010185

K. Samveeth\_AP23110010176

T. Abhinav-AP23110010228

MD.Waseem-AP23110010174

**CODE:**

import requests import os import datetime

def fetch\_weather\_data(city\_name, api\_key):

base\_url = "[https://api.openweathermap.org/data/2.5/weather"](https://api.openweathermap.org/data/2.5/weather%22)

params = { "q": city\_name, "appid": api\_key, "units": "metric" }

try:  
 response = requests.get(base\_url, params=params, timeout=10)  
 response.raise\_for\_status()  
 return response.json()  
except requests.exceptions.HTTPError as http\_err:  
 print(f"HTTP error occurred for {city\_name}: {http\_err}")  
except requests.exceptions.ConnectionError:  
 print("Connection error. Please check your internet connection.")  
except requests.exceptions.Timeout:  
 print("Request timed out.")  
except requests.exceptions.RequestException as e:  
 print("An error occurred:", e)  
  
return None

def parse\_weather\_data(data):

if data is None or data.get("cod") != 200:

return None

weather\_info = {  
 "description": data["weather"][0]["description"],  
 "temperature": data["main"]["temp"],  
 "feels\_like": data["main"]["feels\_like"],  
 "temp\_min": data["main"]["temp\_min"],  
 "temp\_max": data["main"]["temp\_max"],  
 "humidity": data["main"]["humidity"],  
 "wind\_speed": data["wind"]["speed"],  
 "city": data["name"],  
 "country": data["sys"]["country"],  
 "timestamp": datetime.datetime.utcfromtimestamp(data["dt"]).strftime('%Y-%m-%d %H:%M:%S')  
}  
return weather\_info

def display\_weather(info): if info is None: print("Could not display weather information.") return

print(f"\nWeather in {info['city']}, {info['country']} at {info['timestamp']} UTC:")  
print(f"Condition : {info['description'].capitalize()}")  
print(f"Temperature : {info['temperature']}°C (Feels like: {info['feels\_like']}°C)")  
print(f"Min/Max Temp: {info['temp\_min']}°C / {info['temp\_max']}°C")  
print(f"Humidity : {info['humidity']}%")  
print(f"Wind Speed : {info['wind\_speed']} m/s")

def log\_to\_file(info, filename="weather\_log.txt"): if info is None: return

with open(filename, "a") as file:  
 file.write(f"Weather in {info['city']}, {info['country']} at {info['timestamp']} UTC:\n")  
 file.write(f"Condition : {info['description'].capitalize()}\n")  
 file.write(f"Temperature : {info['temperature']}°C (Feels like: {info['feels\_like']}°C)\n")  
 file.write(f"Min/Max Temp: {info['temp\_min']}°C / {info['temp\_max']}°C\n")  
 file.write(f"Humidity : {info['humidity']}%\n")  
 file.write(f"Wind Speed : {info['wind\_speed']} m/s\n")  
 file.write("-" \* 40 + "\n")

def main(): api\_key = "484bdd54ba5296ef242100b0de56a4a0" # Replace this with your real key

print("Welcome to the Weather App!")  
cities = []  
  
while True:  
 city = input("Enter a city name (or type 'done' to finish): ").strip()  
 if city.lower() == "done":  
 break  
 elif city:  
 cities.append(city)  
  
if not cities:  
 print("No cities entered. Exiting.")  
 return  
  
for city in cities:  
 print(f"\nFetching weather for {city}...")  
 data = fetch\_weather\_data(city, api\_key)  
 weather\_info = parse\_weather\_data(data)  
 display\_weather(weather\_info)  
 log\_to\_file(weather\_info)  
  
print("\nWeather information retrieved and saved to log (if enabled).")

if **name** == "**main**": main()