Before:

import requests

import matplotlib.pyplot as plt

import pyinputplus as pyip

import os

BASE\_URL = "https://wttr.in"

def welcome\_message():

print("="\*60)

print("🤖 Welcome to WeatherWise – Your AI-Powered Weather Assistant")

print("="\*60)

print("I’m your assistant. Ask me anything like:")

print("- 'Will it rain tomorrow in Port Louis?'")

print("- 'What is the temperature today in Rose Hill?'")

print("- Or check visual forecasts for any city!")

print()

def get\_weather\_data(location, forecast\_days=5):

"""

Retrieve weather data for a specified location using wttr.in API.

"""

try:

url = f"{BASE\_URL}/{location}?format=j1"

response = requests.get(url, timeout=10)

response.raise\_for\_status()

data = response.json()

if 'weather' not in data or 'current\_condition' not in data:

print("⚠️ AI Notice: The data seems incomplete. Try another city.")

return None

forecast = data.get('weather', [])[:forecast\_days]

current = data.get('current\_condition', [{}])[0]

return {

'location': location,

'current': current,

'forecast': forecast

}

except Exception as e:

print(f"🚨 AI Alert: Could not retrieve weather info for {location}. Error: {e}")

return None

def create\_temperature\_visualisation(weather\_data, output\_type='display'):

dates = [day['date'] for day in weather\_data['forecast']]

max\_temps = [int(day['maxtempC']) for day in weather\_data['forecast']]

min\_temps = [int(day['mintempC']) for day in weather\_data['forecast']]

plt.figure(figsize=(8, 4))

plt.plot(dates, max\_temps, label='Max Temp (°C)', marker='o')

plt.plot(dates, min\_temps, label='Min Temp (°C)', marker='o')

plt.title(f"🌡️ Temperature Trend for {weather\_data['location'].title()}")

plt.xlabel("Date")

plt.ylabel("Temperature (°C)")

plt.legend()

plt.grid(True)

if output\_type == 'display':

plt.show()

else:

return plt.gcf()

def create\_precipitation\_visualisation(weather\_data, output\_type='display'):

dates = [day['date'] for day in weather\_data['forecast']]

rain\_chances = [int(day['hourly'][4]['chanceofrain']) for day in weather\_data['forecast']]

plt.figure(figsize=(8, 4))

plt.bar(dates, rain\_chances, color='skyblue')

plt.title(f"☔ Chance of Rain for {weather\_data['location'].title()}")

plt.xlabel("Date")

plt.ylabel("Chance of Rain (%)")

plt.ylim(0, 100)

plt.grid(True, axis='y')

if output\_type == 'display':

plt.show()

else:

return plt.gcf()

def parse\_weather\_question(question):

keywords = {

'temperature': ['temperature', 'hot', 'cold', 'warm', 'cool'],

'rain': ['rain', 'precipitation', 'wet', 'drizzle'],

'today': ['today', 'now', 'currently'],

'tomorrow': ['tomorrow', 'next day', 'following day']

}

weather\_type = None

for key in ['temperature', 'rain']:

if any(word in question.lower() for word in keywords[key]):

weather\_type = key

break

if not weather\_type:

weather\_type = 'temperature'

time\_period = 'today' if any(word in question.lower() for word in keywords['today']) else 'tomorrow'

location = question.lower().split(' in ')[-1].strip().title() if ' in ' in question.lower() else 'Rose Hill'

return {

'type': weather\_type,

'time': time\_period,

'location': location

}

def generate\_weather\_response(parsed\_question, weather\_data):

forecast = weather\_data['forecast'][0] if parsed\_question['time'] == 'today' else weather\_data['forecast'][1]

if parsed\_question['type'] == 'temperature':

max\_temp = forecast['maxtempC']

min\_temp = forecast['mintempC']

return (f"🤖 Here's what I found for {parsed\_question['location']}:\n"

f"The temperature {parsed\_question['time']} will range between {min\_temp}°C and {max\_temp}°C.")

else:

rain = forecast['hourly'][4]['chanceofrain']

return (f"🌧️ Rain forecast for {parsed\_question['location']}:\n"

f"There's a {rain}% chance of rain {parsed\_question['time']}. Don’t forget your umbrella!")

def main\_menu():

welcome\_message()

while True:

print("\nChoose an option:")

choice = pyip.inputMenu([

'Check Weather by City',

'Ask a Weather Question',

'Show Temperature Visualisation',

'Show Precipitation Visualisation',

'Exit'

], numbered=True)

if choice == 'Check Weather by City':

city = pyip.inputStr("Enter city name: ")

data = get\_weather\_data(city)

if data:

print(f"🌤️ Current weather in {city.title()}: {data['current']['temp\_C']}°C, {data['current']['weatherDesc'][0]['value']}")

elif choice == 'Ask a Weather Question':

question = pyip.inputStr("Ask a weather-related question (e.g., Will it rain tomorrow in Sydney?): ")

parsed = parse\_weather\_question(question)

data = get\_weather\_data(parsed['location'])

if data:

response = generate\_weather\_response(parsed, data)

print(response)

elif choice == 'Show Temperature Visualisation':

city = pyip.inputStr("Enter city name: ")

data = get\_weather\_data(city)

if data:

create\_temperature\_visualisation(data)

elif choice == 'Show Precipitation Visualisation':

city = pyip.inputStr("Enter city name: ")

data = get\_weather\_data(city)

if data:

create\_precipitation\_visualisation(data)

elif choice == 'Exit':

print("🫡 Thank you for using WeatherWise. Stay informed and stay safe!")

break

if \_\_name\_\_ == '\_\_main\_\_':

main\_menu()

After:

import requests

import matplotlib.pyplot as plt

import pyinputplus as pyip

import os

BASE\_URL = "https://wttr.in"

def welcome\_message():

print("="\*60)

print("🤖 Welcome to WeatherWise – Your AI-Powered Weather Assistant")

print("="\*60)

print("I’m your assistant. Ask me anything like:")

print("- 'Will it rain tomorrow in Port Louis?'")

print("- 'What is the temperature today in Rose Hill?'")

print("- Or check visual forecasts for any city!")

print()

def get\_weather\_data(location, forecast\_days=5):

"""

Retrieve weather data for a specified location using wttr.in API.

"""

try:

url = f"{BASE\_URL}/{location}?format=j1"

response = requests.get(url, timeout=10)

response.raise\_for\_status()

data = response.json()

if 'weather' not in data or 'current\_condition' not in data:

print("⚠️ AI Notice: The data seems incomplete. Try another city.")

return None

forecast = data.get('weather', [])[:forecast\_days]

current = data.get('current\_condition', [{}])[0]

return {

'location': location,

'current': current,

'forecast': forecast

}

except Exception as e:

print(f"🚨 AI Alert: Could not retrieve weather info for {location}. Error: {e}")

return None

def create\_temperature\_visualisation(weather\_data, output\_type='display'):

dates = [day['date'] for day in weather\_data['forecast']]

max\_temps = [int(day['maxtempC']) for day in weather\_data['forecast']]

min\_temps = [int(day['mintempC']) for day in weather\_data['forecast']]

plt.figure(figsize=(8, 4))

plt.plot(dates, max\_temps, label='Max Temp (°C)', marker='o')

plt.plot(dates, min\_temps, label='Min Temp (°C)', marker='o')

plt.title(f"🌡️ Temperature Trend for {weather\_data['location'].title()}")

plt.xlabel("Date")

plt.ylabel("Temperature (°C)")

plt.legend()

plt.grid(True)

if output\_type == 'display':

plt.show()

else:

return plt.gcf()

def create\_precipitation\_visualisation(weather\_data, output\_type='display'):

dates = [day['date'] for day in weather\_data['forecast']]

rain\_chances = [int(day['hourly'][4]['chanceofrain']) for day in weather\_data['forecast']]

plt.figure(figsize=(8, 4))

plt.bar(dates, rain\_chances, color='skyblue')

plt.title(f"☔ Chance of Rain for {weather\_data['location'].title()}")

plt.xlabel("Date")

plt.ylabel("Chance of Rain (%)")

plt.ylim(0, 100)

plt.grid(True, axis='y')

if output\_type == 'display':

plt.show()

else:

return plt.gcf()

def parse\_weather\_question(question):

keywords = {

'temperature': ['temperature', 'hot', 'cold', 'warm', 'cool'],

'rain': ['rain', 'precipitation', 'wet', 'drizzle'],

'today': ['today', 'now', 'currently'],

'tomorrow': ['tomorrow', 'next day', 'following day']

}

weather\_type = None

for key in ['temperature', 'rain']:

if any(word in question.lower() for word in keywords[key]):

weather\_type = key

break

if not weather\_type:

weather\_type = 'temperature'

time\_period = 'today' if any(word in question.lower() for word in keywords['today']) else 'tomorrow'

location = question.lower().split(' in ')[-1].strip().title() if ' in ' in question.lower() else 'Rose Hill'

return {

'type': weather\_type,

'time': time\_period,

'location': location

}

def generate\_weather\_response(parsed\_question, weather\_data):

forecast = weather\_data['forecast'][0] if parsed\_question['time'] == 'today' else weather\_data['forecast'][1]

if parsed\_question['type'] == 'temperature':

max\_temp = forecast['maxtempC']

min\_temp = forecast['mintempC']

return (f"🤖 Here's what I found for {parsed\_question['location']}:\n"

f"The temperature {parsed\_question['time']} will range between {min\_temp}°C and {max\_temp}°C.")

else:

rain = forecast['hourly'][4]['chanceofrain']

return (f"🌧️ Rain forecast for {parsed\_question['location']}:\n"

f"There's a {rain}% chance of rain {parsed\_question['time']}. Don’t forget your umbrella!")

def main\_menu():

welcome\_message()

while True:

print("\nChoose an option:")

choice = pyip.inputMenu([

'Check Weather by City',

'Ask a Weather Question',

'Show Temperature Visualisation',

'Show Precipitation Visualisation',

'Exit'

], numbered=True)

if choice == 'Check Weather by City':

city = pyip.inputStr("Enter city name: ")

data = get\_weather\_data(city)

if data:

print(f"🌤️ Current weather in {city.title()}: {data['current']['temp\_C']}°C, {data['current']['weatherDesc'][0]['value']}")

elif choice == 'Ask a Weather Question':

question = pyip.inputStr("Ask a weather-related question (e.g., Will it rain tomorrow in Sydney?): ")

parsed = parse\_weather\_question(question)

data = get\_weather\_data(parsed['location'])

if data:

response = generate\_weather\_response(parsed, data)

print(response)

elif choice == 'Show Temperature Visualisation':

city = pyip.inputStr("Enter city name: ")

data = get\_weather\_data(city)

if data:

create\_temperature\_visualisation(data)

elif choice == 'Show Precipitation Visualisation':

city = pyip.inputStr("Enter city name: ")

data = get\_weather\_data(city)

if data:

create\_precipitation\_visualisation(data)

elif choice == 'Exit':

print("🫡 Thank you for using WeatherWise. Stay informed and stay safe!")

break

if \_\_name\_\_ == '\_\_main\_\_':

main\_menu()