## **15 Git Commit Messages**

### **Core Setup (Menu-Based Version)**

1. **Initial commit: basic weather advisor structure using pyinputplus** *Set up the main menu loop, weather data fetch, and question parsing with console input.*
2. **Add parse\_weather\_question() for basic NLP detection of location and weather type** *Implements keyword extraction logic for questions like “What’s the temperature in Paris?”*
3. **Add generate\_weather\_response() to return human-readable weather messages** *Converts parsed question + API data into natural language response.*
4. **Add temperature visualisation with matplotlib line chart** *First visualisation function to plot average temperatures over forecast days.*
5. **Add precipitation visualisation using bar chart** *Adds second required visualisation to show rainfall chances per day.*

### **Transition to AI-Enhanced + Widget UI**

1. **Refactor console UI to interactive notebook interface using ipywidgets** *Replaces pyinputplus CLI menu with text boxes, buttons, and layout elements.*
2. **Create launch\_weatherwise\_interface() for a clean widget-based layout** *Main display container with text input, forecast range, and event handlers.*
3. **Link fetch button to get\_weather\_data() with output feedback** *Implements interactive data loading with success/failure messaging.*
4. **Connect temperature button to visualisation function in output box** *Temperature chart now displays inline when user clicks the button.*
5. **Connect precipitation button to rainfall visualisation in output panel** *Adds chart with rainfall values and date labels dynamically.*

### **AI-Driven Enhancements**

1. **Wire AI response button to generate\_weather\_response() with question input** *Allows free-form weather questions answered with intelligent feedback.*
2. **Improve visual appeal: add emojis, titles, and grid lines to plots** *Enhances UX with modern, emoji-labeled titles and cleaner graph visuals.*
3. **Improve error handling for missing API data or invalid location** *Prints user-friendly error messages if location doesn’t return valid JSON.*
4. **Add responsive layout: use VBox, HBox for better interface flow** *Groups buttons horizontally and sections vertically for readability.*
5. **Final polish: emoji headers, default values, placeholder text** *Makes the app friendly and fun with default location (e.g. Port Louis), placeholders, and emojis.*

**BEFORE -  
  
import os**

**import requests**

**import matplotlib.pyplot as plt**

**import pyinputplus as pyip**

**# WEATHER DATA FUNCTION**

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

**url = f"https://wttr.in/{location}?format=j1"**

**try:**

**response = requests.get(url)**

**response.raise\_for\_status()**

**return response.json()**

**except Exception as e:**

**print(f"Error retrieving weather data: {e}")**

**return {}**

**# PARSE WEATHER QUESTION FUNCTION**

**def parse\_weather\_question(question):**

**result = {"location": "", "attribute": ""}**

**if "temperature" in question.lower():**

**result["attribute"] = "temperature"**

**elif "rain" in question.lower() or "precipitation" in question.lower():**

**result["attribute"] = "precipitation"**

**else:**

**result["attribute"] = "general"**

**words = question.split()**

**for i, word in enumerate(words):**

**if word.lower() == "in" and i + 1 < len(words):**

**result["location"] = words[i + 1]**

**return result**

**# GENERATE RESPONSE FUNCTION**

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

**if not weather\_data:**

**return "Sorry, I couldn't retrieve weather data."**

**location = parsed\_question["location"]**

**attribute = parsed\_question["attribute"]**

**current = weather\_data.get("current\_condition", [{}])[0]**

**if attribute == "temperature":**

**return f"The current temperature in {location} is {current.get('temp\_C')}°C."**

**elif attribute == "precipitation":**

**return f"The current chance of precipitation in {location} is {current.get('precipMM')} mm."**

**else:**

**return f"In {location}, it's currently {current.get('weatherDesc')[0]['value']} with a temperature of {current.get('temp\_C')}°C."**

**# TEMPERATURE VISUALISATION FUNCTION**

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

**forecast = weather\_data.get("weather", [])**

**dates = [day["date"] for day in forecast]**

**temps = [int(day["avgtempC"]) for day in forecast]**

**fig, ax = plt.subplots()**

**ax.plot(dates, temps, marker='o')**

**ax.set\_title("Average Temperature Over Days")**

**ax.set\_ylabel("Temperature (°C)")**

**ax.set\_xlabel("Date")**

**if output\_type == 'figure':**

**return fig**

**plt.show()**

**# PRECIPITATION VISUALISATION FUNCTION**

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

**forecast = weather\_data.get("weather", [])**

**dates = [day["date"] for day in forecast]**

**precipitation = [float(day["hourly"][0]["precipMM"]) for day in forecast]**

**fig, ax = plt.subplots()**

**ax.bar(dates, precipitation)**

**ax.set\_title("Precipitation Over Days")**

**ax.set\_ylabel("Precipitation (mm)")**

**ax.set\_xlabel("Date")**

**if output\_type == 'figure':**

**return fig**

**plt.show()**

**# MAIN MENU FUNCTION**

**def weather\_menu():**

**print("🌦️ Welcome to WeatherWise Advisor!")**

**location = pyip.inputStr("Enter a city or location: ")**

**forecast\_days = pyip.inputInt("How many forecast days (1-5)? ", min=1, max=5)**

**data = get\_weather\_data(location, forecast\_days)**

**while True:**

**print("\nOptions:")**

**print("1. Ask a natural language weather question")**

**print("2. Show temperature chart")**

**print("3. Show precipitation chart")**

**print("4. Exit")**

**choice = pyip.inputMenu(["1", "2", "3", "4"], numbered=True)**

**if choice == "1":**

**q = pyip.inputStr("Ask your weather question (e.g., 'What is the temperature in Port Louis, Mauritius?'): ")**

**parsed = parse\_weather\_question(q)**

**parsed["location"] = location**

**print(generate\_weather\_response(parsed, data))**

**elif choice == "2":**

**create\_temperature\_visualisation(data)**

**elif choice == "3":**

**create\_precipitation\_visualisation(data)**

**elif choice == "4":**

**print("Goodbye!")**

**break**

**weather\_menu()**

**AFTER -  
  
  
import os**

**import requests**

**import matplotlib.pyplot as plt**

**import ipywidgets as widgets**

**from IPython.display import display, clear\_output**

**# WEATHER DATA FUNCTION**

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

**url = f"https://wttr.in/{location}?format=j1"**

**try:**

**response = requests.get(url)**

**response.raise\_for\_status()**

**return response.json()**

**except Exception as e:**

**return {"error": str(e)}**

**# PARSE WEATHER QUESTION FUNCTION**

**def parse\_weather\_question(question):**

**result = {"location": "", "attribute": ""}**

**if "temperature" in question.lower():**

**result["attribute"] = "temperature"**

**elif "rain" in question.lower() or "precipitation" in question.lower():**

**result["attribute"] = "precipitation"**

**else:**

**result["attribute"] = "general"**

**words = question.split()**

**for i, word in enumerate(words):**

**if word.lower() == "in" and i + 1 < len(words):**

**result["location"] = words[i + 1]**

**return result**

**# GENERATE RESPONSE FUNCTION**

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

**if "error" in weather\_data:**

**return f"Error: {weather\_data['error']}"**

**location = parsed\_question["location"]**

**attribute = parsed\_question["attribute"]**

**current = weather\_data.get("current\_condition", [{}])[0]**

**if attribute == "temperature":**

**return f"🌡️ The current temperature in {location} is {current.get('temp\_C')}°C."**

**elif attribute == "precipitation":**

**return f"🌧️ The current chance of precipitation in {location} is {current.get('precipMM')} mm."**

**else:**

**return f"🌤️ In {location}, it's currently {current.get('weatherDesc')[0]['value']} with a temperature of {current.get('temp\_C')}°C."**

**# TEMPERATURE VISUALISATION FUNCTION**

**def create\_temperature\_visualisation(weather\_data):**

**forecast = weather\_data.get("weather", [])**

**dates = [day["date"] for day in forecast]**

**temps = [int(day["avgtempC"]) for day in forecast]**

**fig, ax = plt.subplots()**

**ax.plot(dates, temps, marker='o', linestyle='-', linewidth=2)**

**ax.set\_title("📈 Temperature Trend")**

**ax.set\_ylabel("Temperature (°C)")**

**ax.set\_xlabel("Date")**

**plt.grid(True)**

**plt.show()**

**# PRECIPITATION VISUALISATION FUNCTION**

**def create\_precipitation\_visualisation(weather\_data):**

**forecast = weather\_data.get("weather", [])**

**dates = [day["date"] for day in forecast]**

**precipitation = [float(day["hourly"][0]["precipMM"]) for day in forecast]**

**fig, ax = plt.subplots()**

**ax.bar(dates, precipitation, color='skyblue')**

**ax.set\_title("🌧️ Precipitation Forecast")**

**ax.set\_ylabel("Precipitation (mm)")**

**ax.set\_xlabel("Date")**

**plt.grid(True)**

**plt.show()**

**# INTERFACE**

**def launch\_weatherwise\_interface():**

**output = widgets.Output()**

**location\_input = widgets.Text(value='Port Louis', description='Location:', style={'description\_width': 'initial'})**

**forecast\_input = widgets.BoundedIntText(value=3, min=1, max=5, description='Days (1-5):', style={'description\_width': 'initial'})**

**question\_input = widgets.Textarea(value='', placeholder='Ask a weather question...', layout=widgets.Layout(width='100%', height='60px'))**

**fetch\_button = widgets.Button(description='Fetch Weather', button\_style='info')**

**temp\_button = widgets.Button(description='Show Temperature Chart', button\_style='success')**

**rain\_button = widgets.Button(description='Show Precipitation Chart', button\_style='primary')**

**ai\_button = widgets.Button(description='Ask AI about Weather', button\_style='warning')**

**data\_store = {}**

**def on\_fetch\_clicked(b):**

**with output:**

**clear\_output()**

**print("🌍 Fetching weather data...")**

**loc = location\_input.value**

**days = forecast\_input.value**

**data\_store["weather"] = get\_weather\_data(loc, days)**

**if "error" in data\_store["weather"]:**

**print(f"❌ Error: {data\_store['weather']['error']}")**

**else:**

**print(f"✅ Weather data for {loc} retrieved successfully!")**

**def on\_temp\_clicked(b):**

**with output:**

**clear\_output()**

**if "weather" in data\_store:**

**create\_temperature\_visualisation(data\_store["weather"])**

**else:**

**print("⚠️ Please fetch the weather data first.")**

**def on\_rain\_clicked(b):**

**with output:**

**clear\_output()**

**if "weather" in data\_store:**

**create\_precipitation\_visualisation(data\_store["weather"])**

**else:**

**print("⚠️ Please fetch the weather data first.")**

**def on\_ai\_clicked(b):**

**with output:**

**clear\_output()**

**if "weather" not in data\_store:**

**print("⚠️ Please fetch the weather data first.")**

**return**

**question = question\_input.value**

**parsed = parse\_weather\_question(question)**

**parsed["location"] = location\_input.value**

**response = generate\_weather\_response(parsed, data\_store["weather"])**

**print(f"🤖 AI Response:\n{response}")**

**fetch\_button.on\_click(on\_fetch\_clicked)**

**temp\_button.on\_click(on\_temp\_clicked)**

**rain\_button.on\_click(on\_rain\_clicked)**

**ai\_button.on\_click(on\_ai\_clicked)**

**display(widgets.VBox([**

**widgets.HTML("<h2>🌦️ <b>WeatherWise: Intelligent Weather Assistant</b></h2>"),**

**location\_input,**

**forecast\_input,**

**widgets.HBox([fetch\_button, temp\_button, rain\_button]),**

**question\_input,**

**ai\_button,**

**output**

**]))**

**# launch the app**

**launch\_weatherwise\_interface()**

**RESULTS BEFORE :   
  
🌦️ Welcome to WeatherWise Advisor!**

**Enter a city or location: How many forecast days (1-5)?**

**Options:**

**1. Ask a natural language weather question**

**2. Show temperature chart**

**3. Show precipitation chart**

**4. Exit**

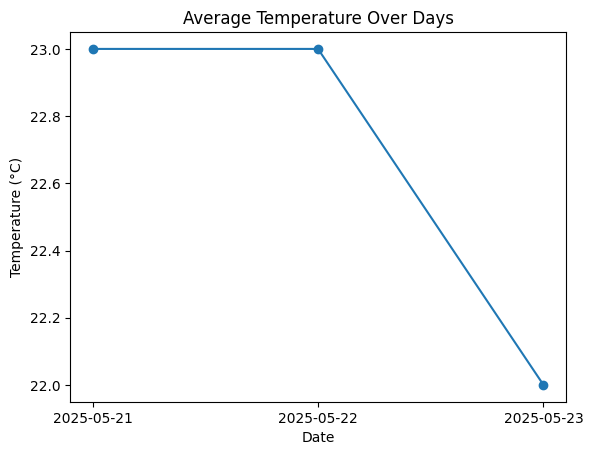
**Please select one of the following:**

**1. 1**

**2. 2**

**3. 3**

**4. 4**

****

**Options:**

**1. Ask a natural language weather question**

**2. Show temperature chart**

**3. Show precipitation chart**

**4. Exit**

**Please select one of the following:**

**1. 1**

**2. 2**

**3. 3**

**4. 4**

**Ask your weather question (e.g., 'What is the temperature in Port Louis, Mauritius?'): The current temperature in port louis is 25°C.**

**Options:**

**1. Ask a natural language weather question**

**2. Show temperature chart**

**3. Show precipitation chart**

**4. Exit**

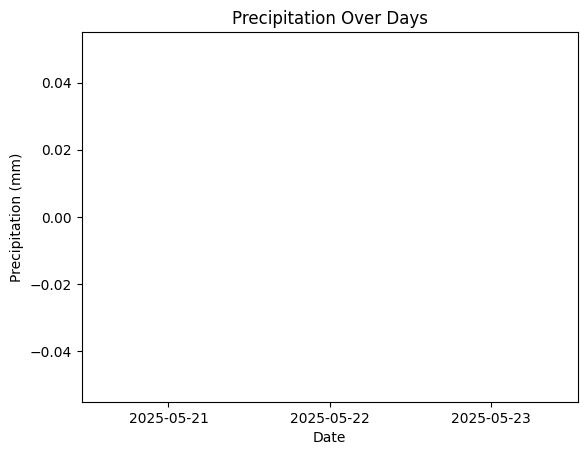
**Please select one of the following:**

**1. 1**

**2. 2**

**3. 3**

**4. 4**

****

**Options:**

**1. Ask a natural language weather question**

**2. Show temperature chart**

**3. Show precipitation chart**

**4. Exit**

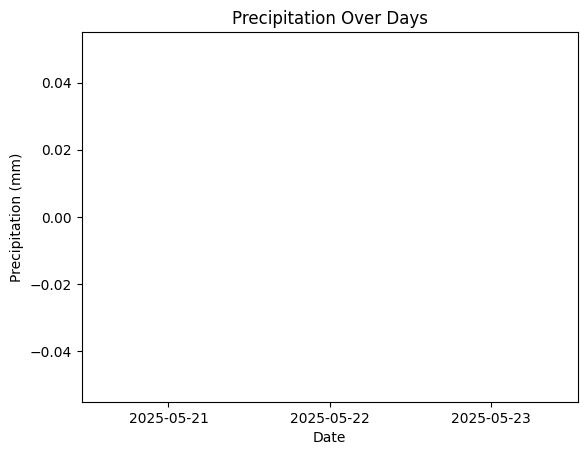
**Please select one of the following:**

**1. 1**

**2. 2**

**3. 3**

**4. 4**

****

**Options:**

**1. Ask a natural language weather question**

**2. Show temperature chart**

**3. Show precipitation chart**

**4. Exit**

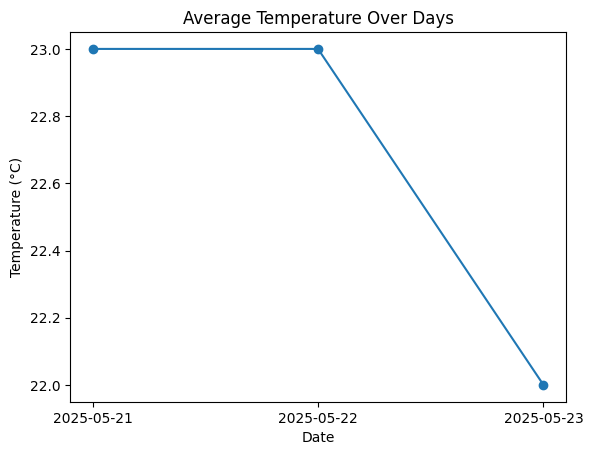
**Please select one of the following:**

**1. 1**

**2. 2**

**3. 3**

**4. 4**

****

**RESULTS AFTER :**

## **🌦️ WeatherWise: Intelligent Weather Assistant**

**Location:**

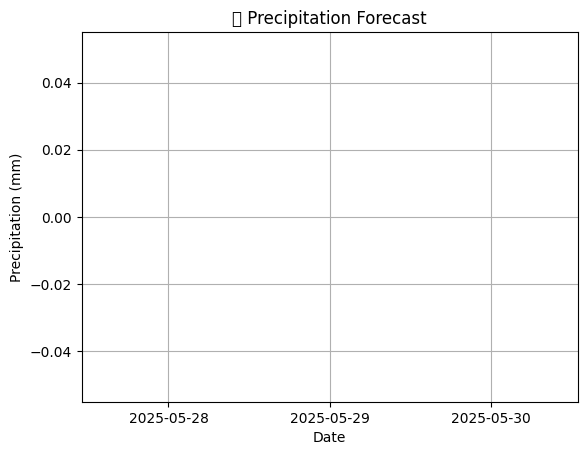
**Days (1-5):**

**Fetch WeatherShow Temperature ChartShow Precipitation Chart**

**Ask AI about Weather**

**/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 127783 (\N{CLOUD WITH RAIN}) missing from font(s) DejaVu Sans.**

**fig.canvas.print\_figure(bytes\_io, \*\*kw)**

****

