



Weather Analytics Power BI Dashboard – Real-Time Weather, AQI & Forecast Insights



Abstract

This project presents a fully interactive Weather Analytics Dashboard designed using Microsoft Power BI, integrating real-time data from WeatherAPI. The dashboard visualizes temperature, humidity, wind speed, visibility, air quality index (AQI), UV index, and multi-day weather forecasts. It uses API-driven data, Power Query M transformations, custom DAX measures, and a modern glassmorphism UI, making it ideal for data analytics practice, real-world BI analysis, and portfolio showcase.

Introduction

Weather analytics plays a crucial role in environmental monitoring, agriculture, travel planning, and public safety. This project demonstrates how Power BI can be used to create a dynamic, visually appealing, and data-driven weather monitoring system by consuming real-time API data. The dashboard is designed to help users stay informed with real-time weather metrics and forecast trends for better decision-making.

Objectives of the Project

- 1 To integrate a real-time WeatherAPI into Power BI
- 2 To visualize key weather metrics (Temp, Humidity, Wind, UV, Pressure, Rain)
- 3 To display AQI indicators with severity-based color coding
- 4 To analyze 7-day forecast trends
- 5 To design a modern, clean, and user-friendly UI
- 6 To create reusable DAX measures for KPIs
- 7 To develop a portfolio-ready, learning-focused dashboard

Data Source & Extraction

5.1 Data Source

WeatherAPI (<https://www.weatherapi.com/>)

Endpoints used:

- /current.json
- /forecast.json

5.3 Data Refresh

- Manual refresh supported
- Can be extended for scheduled refresh if premium workspace is available

5.2 Data Extraction Method

Data is fetched using Power Query M Language:

- API URL constructed using parameters
- Data imported in JSON format
- Parsed and normalized into tables
- Current Weather Table
- Forecast Day Table
- Forecast Hour Table
- Air Quality Table

Tools & Technologies Used

Tool / Tech	Purpose
Power BI Desktop	Dashboard design, modeling & visualization
DAX Measures	KPIs, AQI categories, computed metrics
Power Query M	Fetching & transforming API data
WeatherAPI	Real-time data source
JSON Parsing	Structuring the API response
Glassmorphism UI	Modern design aesthetic

Data Modeling

The model includes three main tables:

- **Current**
 - Current temp, humidity, pressure, visibility
 - AQI parameters (PM10, PM2.5, O3, SO2, CO, NO2)
- **Forecast_Day**
 - Day-wise avg temperature
 - Chance of rain
 - Sunrise/Sunset
- **Forecast_Hour**
 - Hourly temperature trends
 - Weather conditions every hour

Relationships are kept simple due to API structure.

DAX Measures Used

(Summarized for the report — full list can be added as an appendix)

Temperature KPIs

```
Curr_Temp_C = SUM('Current'[current.temp_c]) & " °C"
```

Last Updated

```
last_update = "Last Updated, " & FORMAT(FIRSTNONBLANK('Current'[current.last_updated], ""), "dd mmm")
```

AQI Category Logic

```
AQI_Status_Text = VAR AQI = ROUND(SELECTEDVALUE('Current'[current.air_quality.pm10]),0) RETURN SWITCH( TRUE(), AQI <= 50, "Good", AQI <= 100, "Moderate", AQI <= 150, "Unhealthy for Sensitive", AQI <= 200, "Unhealthy", AQI <= 300, "Very Unhealthy", "Hazardous" )
```

AQI Color Measures

Includes PM10, SO2, PM2.5, NO2, CO, O3.

Dashboard Features Explained

9.1 Real-Time Weather Panel

Shows:

- Temperature
- Weather condition (Mist, Cloudy, Sunny etc.)
- Location
- Last updated timestamp
- Switchable city buttons

9.2 Weather Metrics Cards

- Humidity
- Wind Speed
- Visibility
- Pressure
- Precipitation
- UV Index

9.3 Forecast Temperature Chart

- Line graph
- 7-day trend
- Avg temperature labels

9.4 Air Quality Index Panel

- Circular AQI gauge
- Color-coded pollutant levels
- AQI health recommendation

9.5 Rain Probability

- Bar representation for each day
- Shows chance of rain in percentage

9.6 Sunrise & Sunset Panel

- Auto-fetched from API
- Clean illustrative icons

UI/UX Design Choices

- Glassmorphism theme
- Rounded cards
- Soft-shadow effects
- Weather-themed icons
- Clean typography
- Hierarchical visual grouping
- Dark background for better contrast

This enhances clarity and makes the dashboard visually appealing.

Results & Insights

The final dashboard provides:

- ✓ Accurate real-time weather status
- ✓ AQI health insights with severity colors
- ✓ Clear trend analysis for upcoming temperatures
- ✓ Location-based weather comparison
- ✓ Rain likelihood for planning
- ✓ Modern and intuitive BI user experience

Use Cases

This dashboard can be used by:

- Students learning Power BI
- Data analysts practicing API integration
- Weather researchers
- UI/UX focused BI developers
- Portfolio building for job interviews

Limitations

- Free WeatherAPI key may limit requests
- Scheduled refresh requires Power BI Pro/Premium
- API structure may change over time

Future Enhancements

- ♦ Add map visualization for locations
- ♦ Integrate historical weather analytics
- ♦ Add heat index & wind chill factors
- ♦ Multi-country weather comparison
- ♦ Automate scheduled refresh
- ♦ Add alerts & notifications

Conclusion

The Weather Analytics Power BI Dashboard successfully demonstrates how external API data can be imported, transformed, and visualized using modern BI techniques. The project is an effective blend of data engineering, data visualization, and UI/UX design, making it an ideal portfolio project and a strong demonstration of analytical skills. Appendix: Complete DAX Measures Used in the Dashboard This appendix includes all DAX measures used to calculate KPI values, weather indicators, AQI logic, and color codes for pollutants.

✓ 1. Temperature Measures

- Current Temperature (°C) `Curr_Temp_C = SUM('Current'[current.temp_c]) & " °C"`
- Current Temperature (°F) `Curr_Temp_f = SUM('Current'[current.temp_f]) & " °F"`
- Average Forecast Day Temperature `For_Temp_C = AVERAGE(Forcast_Day[forecast.forecastday.day.avgtemp_c]) & " °C"`
- Average Daily Temp (Raw) `avg_day_Temp_c = SUM(Forcast_Day[forecast.forecastday.day.avgtemp_c])`
- Hourly Avg Temp `avg_hour_Temp_c = SUM(Forcast_Hour[forecast.forecastday.hour.temp_c])`

✓ 2. Date / Time Measures

- Last Updated `last_update = "Last Updated, " & FORMAT(FIRSTNONBLANK('Current'[current.last_updated], ""), "dd mmm")`

✓ 3. Air Quality – Main AQI Measures

- AQI Status (Good / Moderate / Poor / etc.) `AQI_Status_Text = VAR AQI = ROUND(SELECTEDVALUE('Current'[current.air_quality.pm10]),0) RETURN SWITCH(TRUE(), AQI <= 50, "Good", AQI <= 100, "Moderate", AQI <= 150, "Unhealthy for Sensitive", AQI <= 200, "Unhealthy", AQI <= 300, "Very Unhealthy", "Hazardous")`
- AQI Health Suggestion `AQI_Suggestion = VAR AQI = SELECTEDVALUE('Current'[current.air_quality.pm10]) RETURN SWITCH(TRUE(), AQI <= 50, "Air is clean and healthy", AQI <= 100, "Acceptable air quality, stay active", AQI <= 150, "Sensitive groups should reduce outdoor time", AQI <= 200, "Limit prolonged outdoor exertion", AQI <= 300, "Avoid outdoor activity if possible", "Stay indoors, wear mask if outside")`

✓ 4. AQI Color Measures

- PM10 Color `AQI_Color_PM10 = VAR AQI = ROUND(SELECTEDVALUE('Current'[current.air_quality.pm10]),0) RETURN SWITCH(TRUE(), AQI <= 50, "#43d946", AQI <= 100, "#fff570", AQI <= 150, "#ff9800", AQI <= 200, "#d99343", AQI <= 300, "#ff5b0f", "#d95243")`
- PM2.5 Color `AQI_Color_PM2_5 = VAR AQI = ROUND(SELECTEDVALUE('Current'[current.air_quality.pm2_5]),0) RETURN SWITCH(TRUE(), AQI <= 50, "#43d946", AQI <= 100, "#fff570", AQI <= 150, "#ff9800", AQI <= 200, "#d99343", AQI <= 300, "#ff5b0f", "#d95243")`
- SO₂ Color `AQI_Color_SO2 = VAR AQI = ROUND(SELECTEDVALUE('Current'[current.air_quality.so2]),0) RETURN SWITCH(TRUE(), AQI <= 50, "#43d946", AQI <= 100, "#fff570", AQI <= 150, "#ff9800", AQI <= 200, "#d99343", AQI <= 300, "#ff5b0f", "#d95243")`
- NO₂ Color `AQI_Color_NO2 = VAR AQI = ROUND(SELECTEDVALUE('Current'[current.air_quality.no2]),0) RETURN SWITCH(TRUE(), AQI <= 50, "#43d946", AQI <= 100, "#fff570", AQI <= 150, "#ff9800", AQI <= 200, "#d99343", AQI <= 300, "#ff5b0f", "#d95243")`
- O₃ Color `AQI_Color_O3 = VAR AQI = ROUND(SELECTEDVALUE('Current'[current.air_quality.o3]),0) RETURN SWITCH(TRUE(), AQI <= 50, "#43d946", AQI <= 100, "#fff570", AQI <= 150, "#ff9800", AQI <= 200, "#d99343", AQI <= 300, "#ff5b0f", "#d95243")`
- CO Color `AQI_Color_CO = VAR AQI = ROUND(SELECTEDVALUE('Current'[current.air_quality.co]),0) RETURN SWITCH(TRUE(), AQI <= 50, "#43d946", AQI <= 100, "#fff570", AQI <= 150, "#ff9800", AQI <= 200, "#d99343", AQI <= 300, "#ff5b0f", "#d95243")`

✓ 5. Additional AQI Helper Measures

- PM10 Remaining (for donut visual) `PM10_Left = 300 - SUM('Current'[current.air_quality.pm10])`

✓ 6. Weather Metrics

- Humidity `Humidity = SUM('Current'[current.humidity]) & " %"`
- Wind Speed `Wind_Speed = SUM('Current'[current.wind_kph]) & " Kph"`
- Visibility `Visibility = SUM('Current'[current.vis_km]) & " KM"`
- Pressure `Pressure = SUM('Current'[current.pressure_mb]) & " mm"`
- UV Index `UV_Index = SUM('Current'[current.uv])`
- Precipitation (mm) `Precipitation = SUM('Current'[current.precip_mm]) & " mm"`

✓ 7. Rain Probability

- Leftover (%) for progress bar `Left_Rain_Chance = 100 - SUM(Forcast_Day[forecast.forecastday.day.daily_chance_of_rain])`