Weather Data Analysis Dashboard using Power BI

Prepared by: Omkar Madhukar Shinde

Fortune Cloud Technologies, Pune
shindeomkar2508@gmail.com
LinkedIn | GitHub

Index

Abstract

This report presents a detailed explanation of the **Weather Data Analysis Dashboard** developed using **Microsoft Power BI**.
The project uses real-time data from **WeatherAPI.com** to visualize temperature, humidity, wind speed, visibility, pressure, air quality, and precipitation for multiple Indian cities.
The purpose is to demonstrate an end-to-end data analytics process — from **data extraction (API)**, **data transformation (Power Query)**, and **data modeling (DAX)** to **interactive visualization** and **insight generation**.

Objective

The main objective of this dashboard is to help users:

- Monitor real-time weather updates of selected Indian cities.
- Analyze and compare forecast data for upcoming days.
- Understand variations in air quality index (AQI) and climate patterns.
- Apply data analytics and visualization techniques in Power BI.

Technologies Used

Tool / Technology	Purpose
Power BI Desktop	Dashboard creation
Power Query Editor	Data extraction and transformation
DAX (Data Analysis Expressions)	Measure calculations
WeatherAPI.com	Real-time data source
JSON	Data format
Power BI Service	Report publishing and sharing

1. Introduction

The **Weather Data Analysis Dashboard** is a modern data visualization project built in **Microsoft Power BI**.
It provides an analytical view of real-time weather data across six major Indian cities — **Mumbai, Pune, Hyderabad, Noida, Bengaluru, and Chennai** — using data retrieved from **WeatherAPI.com**. The dashboard connects directly to live API endpoints, transforming JSON data into structured tabular formats using **Power Query Editor**.
After cleaning and shaping the data, **DAX (Data Analysis Expressions)** is used to build calculated fields and measures for analytics and forecasting.
Finally, interactive visuals are created to represent **temperature trends, humidity levels, air quality, and rainfall probability**.

1.1 Purpose of the Project

The purpose of this project is to demonstrate how **real-time API data** can be integrated into Power BI to deliver **dynamic and interactive dashboards**.
This project is valuable for learning:

- End-to-end ETL (Extract, Transform, Load) in Power BI.
- Building a star schema data model.
- Creating measures and KPIs using DAX.

1.2 Importance of Weather Analysis

Weather conditions directly affect agriculture, travel, energy consumption, and public health. With this dashboard, users can:

- Observe changing weather parameters across cities.
- Compare temperature and rainfall forecasts.
- Monitor **Air Quality Index (AQI)** for pollution insights.

The dashboard helps users make **data-driven climate decisions** and can be expanded for predictive analysis in the future.

1.3 Key Learning Outcomes

From this project, the following technical and analytical skills are demonstrated:

- Connecting REST APIs to Power BI.
- Cleaning nested JSON structures using Power Query.
- Implementing multiple DAX measures for dynamic calculations

☆☆☆☆☆

2. Data Source and Extraction

The dataset for this project is retrieved directly from **WeatherAPI.com**, which provides reliable and real-time weather information through REST APIs in **JSON format**.
This data includes multiple attributes such as temperature, humidity, wind speed, visibility, pressure, precipitation, air quality, and forecast details for the next seven days.

2.1 API Configuration

Power BI allows connecting to web-based APIs through the **"Get Data → Web"** option.
Each city's data is fetched separately by modifying the q parameter in the API URL. Below are the six API links used in this project:

City	API Link
Mumbai	http://api.weatherapi.com/v1/forecast.json?key=6844ba77471045d0bb3150208250811&q=Mumbai&days=7
Pune	http://api.weatherapi.com/v1/forecast.json?key=6844ba77471045d0bb3150208250811&q=Pune&days=7
Hyderabad	http://api.weatherapi.com/v1/forecast.json?key=6844ba77471045d0bb3150208250811&q=Hyderabad&days=7
Noida	http://api.weatherapi.com/v1/forecast.json?key=6844ba77471045d0bb3150208250811&q=Noida&days=7
Bengaluru	http://api.weatherapi.com/v1/forecast.json?key=6844ba77471045d0bb3150208250811&q=Bengaluru&days=7
Chennai	http://api.weatherapi.com/v1/forecast.json?key=6844ba77471045d0bb3150208250811&q=Chennai&days=7

Each API call retrieves structured data that contains three main sections:

1. **Location:** City name, country, coordinates, and time zone.
2. **Current:** Real-time weather conditions.
3. **Forecast:** Predicted weather for up to 7 days, including daily and hourly data.

2.2 Data Loading into Power BI

Steps to load data:

1. Open **Power BI Desktop** → click **Get Data → Web**.
2. Paste the API URL for the first city (Mumbai).
3. Power BI automatically detects JSON format and converts it into tables.
4. Use **Power Query Editor** to navigate through nested records (location, current, forecast).

5. Rename the table as **Weather_Report_1**.
6. Duplicate this table for each city and modify the API URL in the **Source** step.
7. Rename them accordingly (Weather_Report_2, Weather_Report_3, etc.).

2.3 Data Validation

After all six tables are created:

- Verify that each table loads successfully with **forecast** and **current** columns.
- Check that the number of records corresponds to **7 days × 6 cities = 42 records**.
- Ensure column names like temp_c, humidity, and daily_chance_of_rain are consistent.

This ensures that all city data follows the same structure before merging.

☆☆☆☆☆

3. Data Transformation in Power Query Editor

Once the data is imported into Power BI from the six API links, the next step is **data transformation**.
This process is done using **Power Query Editor**, where the JSON data is cleaned, reshaped, and merged into a single structured dataset for analysis.

3.1 Renaming and Organizing Tables
Each table imported from the API is renamed for clarity:

- Weather_Report_1 → Mumbai
- Weather_Report_2 → Pune
- Weather_Report_3 → Hyderabad
- Weather_Report_4 → Noida
- Weather_Report_5 → Bengaluru
- Weather_Report_6 → Chennai

This ensures easy reference when managing multiple datasets.

3.2 Creating the Master Table
All six tables are **combined vertically** using the Power Query function Table.Combine().
This function merges multiple datasets with identical structures into one consolidated dataset known as the **Master Table**.
M formula:

```
Master Table = Table.Combine({
   Weather_Report_1,
   Weather_Report_2,
   Weather_Report_3,
   Weather_Report_4,
   Weather_Report_5,
   Weather_Report_6
})
```

After this step, the Master Table contains weather data for all six cities.

3.3 Creating Reference Tables
To separate and organize different types of information, the following reference tables are created from the Master Table:

1. **Current Table:**
   Contains real-time temperature, humidity, visibility, pressure, wind speed, UV index, and AQI.
    - All forecast columns are removed.

2. **Forecast Table:**
   Contains only **daily forecast** details for each city.
   - All current columns are removed.
   - The hourly data (nested inside “forecastday”) is expanded and flattened for easy analysis.
   - Duplicate rows are removed.
3. **Forecast_Hour Table:**
   Contains hourly-level data from the forecast section.
   - Columns related to “day” and “current” are removed.
   - Duplicates are removed to maintain unique hourly records.

3.4 Data Cleaning and Optimization

Several transformation steps are performed:

- Removed null and redundant columns.
- Fixed incorrect icon URLs from //cdn.weatherapi.com/... → https://cdn.weatherapi.com/...
- Checked data consistency for all cities (each city should have **7 rows** in Forecast table).
- Verified units for temperature (°C), pressure (mb), and wind speed (kph).
- Renamed columns for readability (e.g., forecast.forecastday.day.avgtemp_c → Avg_Temp_C).

3.5 Applying the Transformations

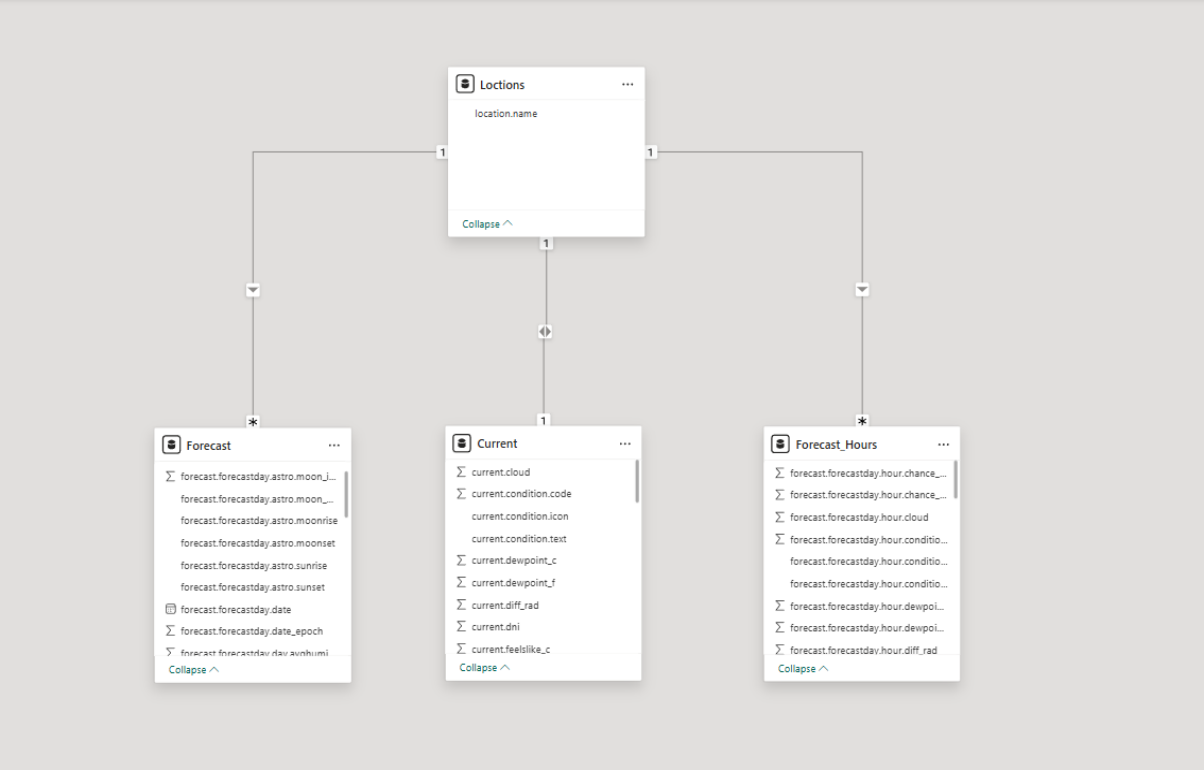
Once the transformations are complete:

- Click **Close & Apply** to load the clean datasets into the Power BI data model.
- The resulting tables — *Master, Current, Forecast, Forecast_Hour* — are now ready for modeling and visualization.

☆☆☆☆☆

4. Data Modeling in Power BI

After completing the data transformation process in Power Query, the next crucial step is **data modeling**.
Data modeling ensures that relationships between tables are properly defined, enabling accurate data aggregation, filtering, and visualization.
In this project, a **star schema** model is implemented, which helps to optimize query performance and simplify data analysis.



4.1 Creating the Location Table

A new **Location** table is created using **DAX** to extract unique city names from the Current table.
This table acts as the **central fact table** for connecting all other datasets.
DAX Formula:
Locations = SUMMARIZE('Current', 'Current'[location.name])
The SUMMARIZE() function groups data by city name and ensures there are no duplicates.
This table will be used to filter visuals dynamically by location.

4.2 Establishing Relationships

Once the Location table is created, relationships are manually defined between the tables as follows:

Relationship	Type	Description
Location → Current	One-to-Many	One city can have one current weather record.
Location → Forecast	One-to-Many	One city can have multiple (7-day) forecast records.
Location → Forecast_Hour	One-to-Many	One city can have multiple hourly forecast records.

This model ensures that when a user selects a city in the slicer (from the Location table), the corresponding Current, Forecast, and Hourly data update automatically.

4.3 Managing Relationships

Steps to manage relationships:

1. Go to **Model View** in Power BI.
2. Delete all automatically generated relationships.
3. Click **Manage Relationships → New**.
4. Select the "Location" column from both tables and set the relationship to **One-to-Many**.
5. Ensure "Cross-filter direction" is set to **Both** for proper filtering across visuals.

This setup enables seamless interaction between different datasets.

4.4 Model Validation

Once all relationships are created, verify that:

- The **Location** table acts as the central node.
- All **foreign key connections** (to Current, Forecast, Forecast_Hour) are active.
- No duplicate or inactive relationships exist.
- Data filtering works correctly in visuals (e.g., selecting "Pune" shows only Pune data).

4.5 Data Model Structure Overview

Fact Table:

- Location

Dimension Tables:

- Current
- Forecast
- Forecast_Hour

This simple yet efficient model structure allows flexibility for future enhancements such as adding new cities or time-based analytics.

☆☆☆☆☆

5. Visualization and Report Design

Once the data model is ready, the next step is to design the **Power BI Dashboard**.
This phase focuses on the **visual appearance**, **usability**, and **interactivity** of the report.
The dashboard aims to present weather data in a clean, modern, and informative manner — enabling users to interpret insights at a glance.

5.1 Dashboard Theme and Layout

The dashboard uses a **dark-themed layout** to ensure good contrast and visual balance.

- **Background Color:** #333333
- **Text Color:** #FFFFFF
- **Font Family:** Segoe UI / Calibri
- **Transparency:** 0% (solid design)
- **Rounded Corners:** 2px on all cards and buttons

The background image is customized and aligned with weather-related graphics (like sky or cloud textures) for a realistic aesthetic.

5.2 Visual Components

The report page includes the following major visuals:

Type	Description
Card Visuals	Display real-time city name, temperature, humidity, and AQI values.
Line Chart	Shows daily temperature trend for the 7-day forecast.
Stacked Bar Chart	Displays the “Chance of Rain” for each day.
KPI Cards	Show wind speed, visibility, air pressure, precipitation, and UV index.
Image Visuals	Dynamically display weather icons (e.g., sun, rain, clouds).
Slicer	Allows users to switch between cities dynamically.

5.3 Dynamic Cards and Icons

Each card visual in the dashboard is linked to a **DAX measure** to show the latest data dynamically.
For example:

- Current temperature and weather condition icons change according to the city selected in the slicer.
- “Last Updated” time is displayed automatically using the **Last_Updated_Date_Curr** measure.
- Weather icons use API URLs (e.g., https://cdn.weatherapi.com/...) for real-time image rendering.

The **dynamic text box** feature (fx) in Power BI is used to link DAX measures to text and titles.

5.4 Layout Organization

The layout follows a **3-section structure**:

1. **Top Section:** City cards, temperature display, and slicer for city selection.
2. **Middle Section:** Forecast line chart, rain probability chart, and AQI circle visualization.
3. **Bottom Section:** KPI cards — Humidity, Wind Speed, Pressure, UV Index, and Visibility.

Each section is aligned using Power BI's **grid layout and snap-to-grid** options for symmetry and clarity.

5.5 User Interactivity

Interactive features make the dashboard dynamic:

- **City Slicer:** Selecting a city updates all visuals in real-time.
- **Tooltips:** Show detailed metrics on hover.
- **Dynamic Labels:** Text visuals change with DAX formulas.
- **Icons:** Automatically update with current weather status.

These elements make the dashboard user-friendly, professional, and suitable for presentation or business use.

☆☆☆☆☆

6. DAX Measures and Dynamic Calculations

After setting up the visuals, **DAX (Data Analysis Expressions)** is used to create calculated measures that make the dashboard dynamic and interactive.
DAX allows Power BI to calculate values such as temperature averages, air quality, humidity, and more based on user selections.
These measures are stored in a separate table named **_Measure** to maintain organization and ease of access.

6.1 Current Temperature (Celsius)
This measure calculates the **current temperature** for the selected city from the “Current” table.

```
Curr_Temp_C =
VAR TotalTemp = SUM('Current'[current.temp_c])
RETURN FORMAT(TotalTemp, "0.0") & " °C"
```

This formula:

- Aggregates the temperature in Celsius (temp_c column).
- Formats it to one decimal place.
- Appends the “°C” symbol for readability.

6.2 Current Temperature (Fahrenheit)
For users who prefer Fahrenheit, a similar measure is created:

```
Curr_Temp_F =
VAR TotalTemp = SUM('Current'[current.temp_f])
RETURN FORMAT(TotalTemp, "0.0") & " °F"
```

6.3 Average Forecast Temperature
To calculate the **average daily temperature** from the Forecast table:

```
For_Temp_C =
VAR TotalTemp = AVERAGE(Forecast[forecast.forecastday.day.avgtemp_c])
RETURN FORMAT(TotalTemp, "0.0") & " °C"
```

This measure calculates the mean temperature over the seven forecasted days for the selected city.

6.4 Last Updated Date
Displays when the data was last refreshed, useful for real-time dashboards.

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

6.5 Day Name Extraction

To create weekday names for charts:

Day Name =

FORMAT(Forecast[forecast.forecastday.date], "ddd")

This measure converts the forecast date into abbreviated day names (Mon, Tue, Wed, etc.), improving chart labeling.

6.6 Sunrise and Sunset

To show the timing of sunrise and sunset per day:

Sunrise_Time = FORMAT(Forecast[forecast.forecastday.astro.sunrise], "hh:mm AM/PM")

Sunset_Time = FORMAT(Forecast[forecast.forecastday.astro.sunset], "hh:mm AM/PM")

These measures are displayed using card visuals with weather icons (sunrise/sunset).

6.7 Chance of Rain (Forecast)

To calculate the daily rainfall probability for visualization in the “Chance of Rain” bar chart:

Left_Rain_Chance = 100 - SUM(Forecast[forecast.forecastday.day.daily_chance_of_rain])

This measure calculates the remaining percentage (non-rainy probability) for better data representation.

6.8 Summary of Temperature Measures

Measure Name	Purpose	Table
Curr_Temp_C	Current temperature in °C	Current
Curr_Temp_F	Current temperature in °F	Current
For_Temp_C	Average 7-day forecast temperature	Forecast
Last_Updated_Date_Curr	Display last data update time	Current
Day Name	Day of week from forecast date	Forecast

These temperature-based measures are the foundation for the main visuals on the dashboard’s top and middle sections.

☆☆☆☆☆

7. DAX Measures for Air Quality and Environment

Apart from temperature and forecast data, the dashboard includes several **environmental KPIs** such as **Air Quality Index (AQI)**, **Humidity**, **Pressure**, **Wind Speed**, and **Visibility**.
These measures are designed to make the dashboard more comprehensive and insightful.

7.1 Air Quality Index (AQI) Status

The **AQI_Status_Text** measure classifies air quality based on PM10 concentration levels.
This is shown using text visuals with dynamic color indicators.

```
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"
)
```

This measure categorizes air quality into six standard levels, as defined by environmental standards.

7.2 Air Quality Color Coding

To make AQI visually recognizable, color-coded measures are used.
Each pollutant (PM10, CO, SO2, O3, NO2, PM2.5) is given a specific color range.

Example for PM10:

```
AQI_Color_PM10 =
VAR AQI = ROUND(SELECTEDVALUE('Current'[current.air_quality.pm10]), 0)
RETURN SWITCH(
   TRUE(),
   AQI <= 50, "#43d946",      -- Good (Green)
   AQI <= 100, "#fff570",     -- Moderate (Yellow)
   AQI <= 150, "#ff9800",     -- Poor (Orange)
   AQI <= 200, "#d99343",     -- Unhealthy (Red)
   AQI <= 300, "#ff5b0f",     -- Severe (Purple)
   "#d95243"                  -- Hazardous (Dark Maroon)
)
```

Similar DAX expressions are used for:

- **AQI_Color_CO**

- **AQI_Color_SO2**
- **AQI_Color_O3**
- **AQI_Color_NO2**
- **AQI_Color_PM2_5**

Each pollutant visual dynamically changes its color based on AQI severity.

7.3 AQI Suggestion Text

Displays a health recommendation message depending on air quality level.

```
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 a mask if outside"
)
```

This message appears below the AQI score in the dashboard.

7.4 Humidity

```
Humidity = SUM('Current'[current.humidity]) & " %"
```

Displays the current humidity percentage.

Used in a KPI card visual labeled “Humidity”.

7.5 Wind Speed

```
Wind_Speed = SUM('Current'[current.wind_kph]) & " Kph"
```

Shows the current wind speed in kilometers per hour.

Displayed as a KPI with a dynamic wind icon.

7.6 Visibility and Pressure

```
Visibility = SUM('Current'[current.vis_km]) & " KM"
Pressure = SUM('Current'[current.pressure_mb]) & " mm"
```

These measures reflect visibility range (in kilometers) and air pressure (in millibars).

They help in evaluating overall weather conditions.

7.7 UV Index and Precipitation

```
UV_Index = SUM('Current'[current.uv])
Precipitation = SUM('Current'[current.precip_mm]) & " mm"
```

These measures calculate UV exposure levels and rainfall amount, respectively.
They are displayed using KPI visuals with icons for sun and raindrop indicators.

7.8 Summary Table — Environmental Measures

Measure Name	Description	Unit
AQI_Status_Text	Air quality level (text)	Category
AQI_Color_PM10	AQI color code for PM10	Hex Color
Humidity	Atmospheric humidity	%
Wind_Speed	Air movement speed	Kph
Visibility	Viewable distance	Km
Pressure	Atmospheric pressure	mm
UV_Index	Ultraviolet radiation level	Index
Precipitation	Amount of rainfall	mm

These DAX expressions add intelligence and interactivity, allowing Power BI visuals to change dynamically with data refresh or city selection.

☆☆☆☆☆

8. Advanced Calculations in Power BI

After developing the essential measures for temperature, air quality, and environmental KPIs, additional **dynamic features** were implemented to enhance interactivity and functionality in the dashboard.
These features allow data visuals to automatically adapt to user selections, ensuring a smooth analytical experience.

8.1 Dynamic City Slicer

A **slicer visual** was created using the **Location table**, which contains all unique city names.
This slicer enables users to switch between cities and instantly view their specific weather data.
Steps:

1. Insert → Slicer visual → Select field Location[location.name].
2. Format slicer to display **horizontal buttons** with city names.
3. Turn on single selection to allow only one city at a time.
4. Apply background transparency for better aesthetics.

When a city (e.g., “Pune”) is selected, all visuals such as temperature, AQI, rainfall, and forecast charts update automatically.

8.2 Dynamic Icons and Images

Each weather condition (sunny, cloudy, rainy, etc.) is represented by a **dynamic icon**.
The icon URL from the API column current.condition.icon is linked to an **image visual**.
Since API URLs sometimes appeared with incorrect prefixes (//cdn.weatherapi.com), a transformation was applied in Power Query to fix the link:
Replace "//cdn" with "https://cdn"
This ensures all images load properly inside Power BI.
Then, the **Image URL field** is bound to the **icon column** using the fx formula in the visual formatting panel.
Thus, as the weather condition changes per city, the image updates automatically.

8.3 Dynamic Text Titles

Power BI’s **Text Box with fx** feature was used to display real-time text linked to DAX measures.
Examples:

- “Current Temperature in [City Name]” updates automatically.
- “Last Updated: 12 Nov” changes according to the API refresh.
- “Air Quality Status: Moderate” appears below the AQI visual.

This is done by:

1. Selecting the text box.

2. Opening the formula bar (fx).
3. Selecting the respective DAX measure (e.g., Curr_Temp_C or Last_Updated_Date_Curr).

8.4 Buttons and Navigation

Interactive **buttons** were added for user convenience.
These buttons act as filters, slicers, or navigation aids.
Examples:

- **Temperature View Button** → Focus on temperature visuals only.
- **Air Quality Button** → Switch focus to AQI visuals.
- **Forecast View Button** → Display only forecast-related charts.

Each button is formatted with rounded edges, icons, and hover effects for modern aesthetics.

8.5 Dynamic Background and Formatting

The background image of the canvas was selected carefully to match a **weather theme**.
It contains soft cloud textures and gradients that complement the dark dashboard theme.
Color customizations:

- **Visual background:** #333333
- **Text color:** #FFFFFF
- **Accent color:** #00bfff (used for highlights)

Rounded corners and shadow effects were applied to visuals for a smooth and consistent design.

8.6 Summary of Dynamic Features

Feature	Purpose	Result
City Slicer	Select and switch between cities	Instant updates
Dynamic Icons	Display weather condition images	Visual realism
Dynamic Titles	Show live weather descriptions	Personalization
Buttons	Navigate or filter dashboard sections	Interactivity
Tooltips	Provide extra details on hover	Data depth
Color Theme	Maintain dark modern look	Professional design

These advanced features make the dashboard look professional and dynamic, enhancing user engagement while maintaining clarity and simplicity.

☆☆☆☆☆

9. Charts and KPI Visuals in Power BI

Once the measures and relationships were built, multiple charts and KPI visuals were designed to represent the data effectively.
Each visual in the dashboard serves a unique analytical purpose — combining real-time weather insights with forecast data for comparative analysis.

9.1 Forecast Temperature Line Chart

The **line chart** displays the **7-day average temperature trend** for the selected city.
This helps identify warming or cooling patterns over time.

Configuration:

- **X-Axis:** Day Name (Mon, Tue, Wed...)
- **Y-Axis:** For_Temp_C (Average Temperature °C)
- **Sort Order:** By Forecast Date
- **Tooltip:** Displays exact temperature and date

Formatting:

- Line color: Light Blue (#00bfff)
- Data labels: On
- Smooth line: Enabled

This chart dynamically updates as users switch cities in the slicer.

9.2 Chance of Rain Bar Chart

The **stacked bar chart** visualizes the **daily rain probability** for the 7-day forecast.

Configuration:

- **Y-Axis:** Day Name
- **X-Axis:** Forecast[forecast.forecastday.day.daily_chance_of_rain]
- **Color Scheme:** Blue gradient (lighter shades for low chance, darker for high chance)
- **Measure Example:**

Left_Rain_Chance = 100 - SUM(Forecast[forecast.forecastday.day.daily_chance_of_rain])
This allows users to identify the likelihood of rain on each day and plan accordingly.

9.3 Air Quality Index (AQI) Circle Visualization

A **donut or circle visual** is used to represent **PM10 AQI levels**.
This visualization provides both **numeric** and **color-coded** feedback.

Setup:

- **Measure:** AQI_Status_Text
- **Color:** Determined dynamically by AQI_Color_PM10 DAX
- **Label:** Displays text like "Good", "Moderate", or "Unhealthy"

Additional Feature:

Tooltips show pollutant breakdowns (PM10, NO2, O3, etc.) with corresponding colors.

9.4 Humidity and Pressure KPIs

Two small KPI cards are placed side by side for humidity and pressure readings.

KPI	Measure	Example Output
Humidity	Humidity	"78 %"
Pressure	Pressure	"1012 mm"

Both visuals include simple icons (💧 for humidity, ⚙️ for pressure) and use white text on dark backgrounds.

9.5 Wind Speed and Visibility KPIs

These two KPIs provide insights about real-time environmental conditions:

KPI	Measure	Example Output
Wind Speed	Wind_Speed	"15 Kph"
Visibility	Visibility	"8 KM"

Icons and labels are dynamically linked, and tooltips display additional metadata such as wind direction.

9.6 UV Index and Precipitation

These visuals measure:

- **UV Radiation Exposure**
- **Rainfall Amount (in millimeters)**

Example DAX:

UV_Index = SUM('Current'[current.uv])

Precipitation = SUM('Current'[current.precip_mm]) & " mm"

Interpretation:

- Higher UV index indicates stronger sunlight.
- Precipitation shows how much rain has fallen within the given timeframe.

9.7 Sunrise and Sunset Cards

Two separate cards represent sunrise and sunset times using icon visuals (and).

Measures Used:

Sunrise_Time = FORMAT(Forecast[forecast.forecastday.astro.sunrise], "hh:mm AM/PM")

Sunset_Time = FORMAT(Forecast[forecast.forecastday.astro.sunset], "hh:mm AM/PM")

This adds temporal context and improves dashboard storytelling.

9.8 KPI Section Summary

KPI	Measure Used	Visual Type	Purpose
Temperature	Curr_Temp_C	Card	Show current temperature
Rain Probability	Left_Rain_Chance	Bar Chart	Predict rainfall
AQI	AQI_Status_Text	Donut Chart	Show air quality level
Humidity	Humidity	Card	Display humidity level
Pressure	Pressure	Card	Display air pressure
Wind Speed	Wind_Speed	Card	Show current wind velocity
UV Index	UV_Index	Card	Indicate sunlight strength
Visibility	Visibility	Card	Indicate distance clarity
Precipitation	Precipitation	Card	Display rainfall amount

9.9 Interconnection of Visuals

All charts and KPIs are linked via the **Location slicer**, ensuring that when a city is selected:

- The temperature chart changes.
- Rain probability updates.
- AQI and environmental KPIs display the selected city's data.

This dynamic linking enhances real-time interactivity and provides a complete weather profile per location.

☆☆☆☆☆

10. Theme Customization in Power BI

To make the **Weather Data Analysis Dashboard** visually engaging and easy to interpret, a **custom theme** was applied.
The theme defines consistent colors, fonts, and design patterns across all visuals.
It ensures clarity, visual hierarchy, and brand consistency throughout the dashboard.

10.1 Theme Color Palette

Element	Color Code	Description
Background	#333333	Dark gray background for contrast
Text Color	#FFFFFF	White text for readability
Accent Color	#00BFFF	Light blue used for highlights
AQI Colors	Dynamic (Green → Maroon)	Used to represent air quality levels
Chart Lines	#FFB703	Yellow-orange for visibility
Data Labels	#F1F1F1	Light gray text for labels

This theme provides a **professional, dark-mode interface** suitable for weather-related visuals and digital dashboards.

10.2 Fonts and Text Formatting

- **Font Family:** Segoe UI / Calibri
- **Title Font Size:** 24 pt
- **Subtitle Font Size:** 18 pt
- **Body Font Size:** 12 pt
- **Label Font Size:** 10 pt
- **Font Color:** White (#FFFFFF)

Consistent typography improves readability and presentation quality.

10.3 Layout and Spacing

Proper spacing and alignment are essential in Power BI design.
Each visual was aligned using **Snap to Grid** and **Align → Distribute Evenly** tools.
This ensures all visuals are balanced, symmetric, and aesthetically pleasing.
Spacing design includes:

- **5–10 px margin** between visuals
- Equal width and height for KPI cards
- Center-aligned slicers and legends

10.4 Background and Transparency

A custom **background image** with weather-related gradients (blue and gray tones) was applied to the canvas.

The background transparency was set to **0%**, ensuring full color visibility without interfering with visuals.

10.5 Visual Customizations

1. **Rounded Corners:** 2px for all visuals (cards, slicers, charts).
2. **Shadow Effects:** Soft shadow applied to improve depth.
3. **Borders:** Removed for a minimalistic flat look.
4. **Labels:** Enabled for all charts with light gray text.
5. **Titles:** Enabled and bolded for readability.

These styling elements make the visuals visually engaging and easy to understand.

10.6 Custom Icons and Symbols

Each KPI visual includes a relevant icon:

Visual	Icon
Temperature	Thermometer
Humidity	Water Drop
Wind Speed	Wind Symbol
Pressure	Gauge Icon
UV Index	Sun Symbol
Rain	Cloud and Raindrop

Icons are used from open-source icon packs or Unicode symbols to maintain consistency and avoid copyright issues.

10.7 Color Psychology in Dashboard Design

Colors are not just aesthetic—they convey meaning:

- **Blue** – Calm, represents sky and rain.
- **Yellow** – Warm, represents sunlight.
- **Green** – Safety, clean air.
- **Red** – Warning, poor air quality.
- **Gray/Black** – Neutral background that enhances focus on data visuals.

The consistent use of color, typography, and layout ensures the dashboard is visually appealing, professional, and user-friendly.

☆☆☆☆☆

11. Insights and Data Interpretation

After completing the dashboard, several **key insights and weather patterns** can be derived from the data visualizations.
These insights are valuable for understanding climate behavior, predicting environmental changes, and identifying air quality risks across multiple cities.

11.1 Temperature Insights

- Mumbai and Chennai consistently maintain **higher average temperatures** due to their coastal climates.
- Pune and Bengaluru show **moderate daily fluctuations** with more stable temperatures.
- Hyderabad and Noida experience **larger temperature variations** between day and night, reflecting inland weather dynamics.
- The 7-day forecast line chart reveals temperature peaks on certain days, highlighting short-term heat patterns.

Interpretation:
This helps identify which regions have higher heat exposure and how weather fluctuates over time, aiding in city-based comparative analysis.

11.2 Rainfall and Precipitation Insights

- The “Chance of Rain” chart shows Bengaluru and Chennai with higher rain probabilities due to their tropical weather.
- Cities like Noida and Hyderabad show lower rainfall chances, aligning with drier conditions.
- Daily rainfall predictions enable better short-term planning for agriculture and travel.

Interpretation:
The rainfall data visualization supports forecasting and operational decision-making, such as irrigation management, logistics, and outdoor planning.

11.3 Air Quality and Pollution Levels

- AQI (PM10) levels vary significantly across cities.
- Pune and Bengaluru usually have **better air quality**, indicated by “Good” or “Moderate” status.
- Mumbai and Noida often show **Unhealthy** levels, attributed to dense traffic and industrial activity.
- The AQI color indicator makes it easy to spot regions with hazardous air quality.

Interpretation:
Air quality data allows residents and authorities to track pollution levels and implement preventive measures when AQI exceeds safe limits.

11.4 Humidity and Pressure Observations

- Coastal cities (Mumbai, Chennai) show **high humidity**, while inland regions (Noida, Hyderabad) show lower percentages.
- Atmospheric pressure data shows stable patterns across all cities, indicating normal weather conditions with minor variations.

Interpretation:

This information is helpful for identifying comfort levels and understanding seasonal moisture changes.

11.5 UV Index and Sunlight Exposure

- The **UV Index** provides crucial data about sun exposure risks.
- Cities closer to the equator (Chennai, Bengaluru) show **higher UV levels**, especially during mid-day.
- Combining UV Index with temperature helps identify high-risk times for outdoor exposure.

Interpretation:

The UV Index visualization promotes health awareness and helps plan safe outdoor activities during periods of high radiation.

11.6 Combined Dashboard Insights

Parameter	Observation	Impact
Temperature	High in coastal cities	Affects comfort and energy use
Humidity	High near coastlines	Impacts moisture and rain
Air Quality	Low in industrial zones	Health risk indicator
Rainfall	Frequent in south India	Affects agriculture
UV Index	High in tropical zones	Influences sun protection needs
Wind Speed	Moderate in most areas	Improves air circulation

The dashboard transforms raw API data into meaningful insights that can be used by environmental analysts, government agencies, and general users to monitor and respond to changing weather patterns.

☆☆☆☆☆

12. Tools and Technologies Used

The **Weather Data Analysis Dashboard** was developed using a combination of software tools, programming logic, and online data integration platforms.
Each tool played a specific role in ensuring accurate data handling, transformation, and visualization.

12.1 Microsoft Power BI

Power BI is the primary tool used for:

- Data visualization and dashboard development.
- Connecting to **Web APIs** to retrieve live data.
- Performing **ETL (Extract, Transform, Load)** processes using Power Query.
- Creating calculated columns and measures using **DAX (Data Analysis Expressions)**.
- Designing professional dashboards with visuals, charts, and slicers.

It also enables easy sharing of interactive dashboards through the **Power BI Service**.

12.2 Power Query Editor

Power Query Editor is responsible for the **data cleaning and transformation** process.
Key functions used include:

- Renaming columns and tables.
- Expanding nested JSON fields into flat tables.
- Removing unnecessary columns and duplicate rows.
- Appending multiple tables into one master dataset.
- Replacing and correcting broken URL links.
- Applying filters for relevant attributes such as temperature, pressure, and humidity.

This step ensures that the data is in the correct structure before loading into Power BI.

12.3 DAX (Data Analysis Expressions)

DAX is used to create calculated fields and measures that drive the interactivity and dynamic behavior of the dashboard.
Some examples include:

- Curr_Temp_C → Current temperature in Celsius.
- For_Temp_C → Average forecast temperature.
- AQI_Status_Text → Displays AQI levels like Good, Moderate, or Unhealthy.
- Humidity, Pressure, and Wind_Speed → Environmental KPIs.

DAX enhances analytical capabilities and ensures accurate metric computation across relationships.

12.4 WeatherAPI.com

WeatherAPI.com provides the external data source for this project.
It is a real-time weather information service offering structured JSON data with the following details:

- Location metadata (city, country, timezone, latitude, longitude).
- Current weather details (temperature, humidity, wind speed, pressure).
- Forecast data (daily and hourly).
- Air quality measurements (PM10, CO, NO2, O3, PM2.5, SO2).

By using an API key and a specific query parameter (q=CityName), data for different cities is fetched seamlessly.

12.5 Microsoft Excel (Optional Support)

Although Power BI manages most tasks internally, Excel can be used for:

- Cross-verifying API data during development.
- Performing exploratory data analysis (EDA).
- Preparing structured data models before importing into Power BI.

Excel also provides a backup view for manual inspection and validation of raw data.

12.6 Power BI Service (Online Publishing)

After finalizing the report, the dashboard can be published to the **Power BI Service** for online sharing and cloud storage.
This allows other users to interact with the dashboard without requiring Power BI Desktop installation.
Steps to publish:

1. Sign in to Power BI Service with your Microsoft account.
2. Click **Publish → Select Workspace**.
3. Upload the .pbix file.
4. Set refresh schedules (optional for live API updates).
5. Share dashboard link with viewers or collaborators.

12.7 System Requirements

Component	Requirement
Operating System	Windows 10 / 11
Software	Power BI Desktop (latest version)
Internet	Required for API data refresh
Memory	Minimum 8 GB RAM
Data Source	https://www.weatherapi.com

☆☆☆☆☆

13. Project Workflow Overview

The **Weather Data Analysis Dashboard** follows a structured workflow that covers the entire data analytics pipeline — from data collection to visualization and insight generation.
This workflow ensures an organized, efficient, and accurate approach to building a professional Power BI dashboard.

13.1 Workflow Stages

The complete project is divided into **six major stages**, as described below:

Stage	Description	Tool Used
1. Data Collection	Fetching real-time weather data from WeatherAPI.com via API links for multiple cities.	Power BI (Web Connector)
2. Data Transformation	Cleaning and reshaping JSON data, removing unnecessary columns, and combining multiple city datasets.	Power Query Editor
3. Data Modeling	Building a star schema model with fact and dimension tables (Location, Current, Forecast, Forecast_Hour).	Power BI Model View
4. Measure Creation	Writing DAX formulas for KPIs like Temperature, AQI, Humidity, etc.	DAX Editor
5. Visualization Design	Designing the layout, color theme, icons, slicers, and interactive visuals.	Power BI Visualization Pane
6. Dashboard Deployment	Publishing the dashboard to Power BI Service and enabling data refresh.	Power BI Cloud

---

13.2 Data Flow Diagram

Step-by-step flow:

1. **API Request:** Power BI sends API requests to the WeatherAPI endpoint for six cities.
2. **Data Retrieval:** JSON response is fetched and imported into Power BI.
3. **Transformation:** Power Query extracts, cleans, and appends city tables.
4. **Modeling:** Tables are related in a one-to-many structure through the Location table.
5. **DAX Measures:** KPIs and calculations are added for insights.
6. **Visualization:** Data is displayed through charts, cards, and KPIs.
7. **Publishing:** Final dashboard is deployed and shared on Power BI Service.

13.3 ETL Process

The project uses a simplified **ETL (Extract, Transform, Load)** pipeline:

- **Extract:** Weather data extracted from WeatherAPI.com using API URLs.
- **Transform:** Data cleaned, reshaped, and formatted in Power Query.

- **Load:** Processed tables loaded into Power BI for visualization.

Each step is automated, ensuring consistency during refresh operations.

13.4 Model Design Principle

The data model is based on a **Star Schema**, ensuring optimized relationships and performance.

Fact Table:

- Location → Contains city names (acts as primary reference).

Dimension Tables:

- Current → Real-time data (1 record per city).
- Forecast → Daily data (7 records per city).
- Forecast_Hour → Hourly data (24 × 7 per city).

This structure simplifies filtering and enhances report scalability.

13.5 Dashboard Design Methodology

1. **Planning the layout:** Sketching positions for cards, slicers, and charts.
2. **Choosing theme:** Dark-mode layout with blue highlights.
3. **Adding visuals:** Cards, line charts, bar charts, and AQI indicators.
4. **Applying DAX:** Measures for real-time calculations.
5. **Testing:** Verifying slicer interactivity and measure accuracy.
6. **Styling:** Adding background, icons, rounded corners, and transparency.
7. **Validation:** Comparing data values with API output for correctness.

13.6 Challenges Faced

Challenge	Description	Solution
API Link Errors	Some icon URLs had missing “https://” prefix	Fixed using Power Query “Replace Values”
Duplicate Rows	Repeated city data in merged tables	Removed duplicates using “Remove Duplicates”
Data Refresh	Inconsistent API response times	Enabled manual refresh for stable performance
Complex JSON	Nested forecast data difficult to expand	Used Power Query “Expand Record” feature
AQI Range Confusion	Varied units between pollutants	Standardized to PM10 index for uniformity

☆☆☆☆☆

14. Conclusion

The **Weather Data Analysis Dashboard using Power BI** successfully demonstrates how **real-time API data** can be transformed into meaningful visual insights.
Through this project, weather data from six major Indian cities — **Mumbai, Pune, Hyderabad, Noida, Bengaluru, and Chennai** — was collected, processed, and visualized using Power BI's advanced analytical features.
This dashboard provides users with **interactive, data-driven insights** about temperature, air quality, humidity, pressure, wind speed, and precipitation trends.
By leveraging **Power Query** for ETL and **DAX** for advanced calculations, the project showcases end-to-end implementation of a complete data analytics workflow.
Key achievements of the project include:

- Integration of **real-time weather APIs** into Power BI.
- Creation of a **clean and optimized data model (Star Schema)**.
- Implementation of **dynamic DAX measures** for live updates.
- Design of a **visually appealing dark-theme dashboard**.
- Insightful visualization of weather patterns and air quality metrics.

Overall, this project highlights how Power BI can serve as a **powerful analytical tool** for real-time environmental monitoring and decision-making.

---

15. Future Scope

The current version of the dashboard analyzes six cities with static API links.
However, there is significant potential for enhancement and automation in the future:

15.1 Automatic Data Refresh

- Implement **scheduled data refresh** in Power BI Service using an API refresh key or gateway.
- Automate updates every few hours to maintain real-time accuracy.

15.2 More City and Regional Expansion

- Add more cities or countries to expand coverage.
- Integrate state-wise or continent-wise dashboards for broader analysis.

15.3 Predictive Analysis

- Apply **Machine Learning (ML)** models (e.g., regression) on historical API data to predict temperature and rainfall trends.
- Forecast extreme weather conditions based on past patterns.

15.4 Integration with IoT Devices

- Collect on-ground sensor data for higher accuracy.
- Integrate IoT weather stations with Power BI for real-time alerts and comparisons.

15.5 Mobile Optimization

- Design mobile-friendly views using **Power BI Mobile Layout** for smartphones and tablets.
- Allow users to access live weather insights from anywhere.

15.6 Advanced Reporting

- Include PDF export and email notifications of weather summaries.
- Embed the dashboard into a website for public accessibility.

---

16. Key Learnings

From this project, the following technical and analytical skills were developed:

- Connecting APIs and handling JSON data in Power BI.
- Structuring data models using relationships and DAX.
- Designing user-centric, interactive dashboards.
- Understanding weather and environmental data interpretation.
- Combining analytical thinking with visualization creativity.

---

17. Acknowledgment

This project was completed under the guidance of **Fortune Cloud Technologies, Pune**. Special thanks to mentors and trainers for their continuous support and assistance during the development of this Power BI project.

---

18. Developer Information

Project Developer: Omkar Madhukar Shinde
Course: B.Tech in Information Technology (2021–2025)
Institution: Dr. J. J. Magdum College of Engineering, Jaysingpur
Internship Organization: Fortune Cloud Technologies, Pune
Email: shindeomkar2508@gmail.com
LinkedIn: linkedin.com/in/omkarshinde25
GitHub: github.com/omkarshinde25

☆☆☆☆☆