API Based Weather Dashboard

This Power BI Weather Dashboard visualizes real-time and 5-day weather forecast data for selected Indian cities. It utilizes data from the OpenWeatherMap API and displays key metrics such as temperature, humidity, pressure, wind speed, visibility, and weather conditions.

Data Sources

Document where the data comes from.

Example:

- **Source:** OpenWeatherMap API
- **Endpoints Used:**
    - Current Weather: https://api.openweathermap.org/data/2.5/weather
    - 5-Day Forecast: https://api.openweathermap.org/data/2.5/forecast
- **Authentication:** API Key (kept secure)
- **Frequency of Refresh:** On demand / periodic (define if scheduled)

Data Model Overview

Summarize your tables and their key columns.

Tables:

- CurrentWeather
    - City, Temperature, FeelsLike, Humidity, Pressure, Weather, IconURL, WindSpeed, Visibility, Sunrise, Sunset, UpdatedAt
- Forecast
    - City, DateTime, Temp, TempMin, TempMax, Humidity, Pressure, Weather, IconURL, WindSpeed, Rain, Visibility

Relationships:

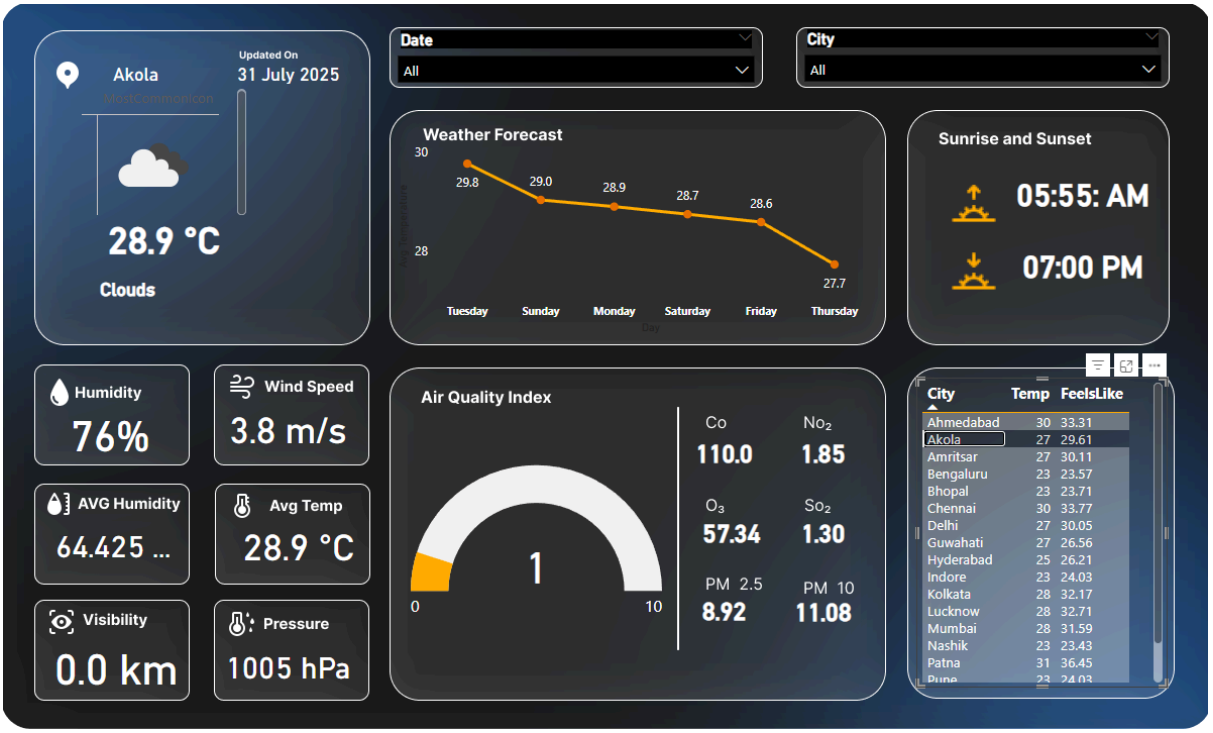
- If any dimension tables (e.g. city list), describe them here.

Measures & Calculations

Document all key DAX measures used, their logic, and units.

Measure Name	Formula (DAX)	Description	Unit
Avg Temp	AVERAGE('Forecast'[Temp])	Average temperature	°C
Avg Humidity	AVERAGE('Forecast'[Humidity])	Avg humidity	%
Avg Wind Speed	AVERAGE('Forecast'[WindSpeed])	Wind speed	m/s
Visibility (km)	'Forecast'[Visibility] / 1000	Visibility converted from meters	km
Sunrise IST	SELECTEDVALUE('CurrentWeather'[Sunrise]) + TIME(5,30,0)	Adjust UTC to IST	Time only

Dashboard image



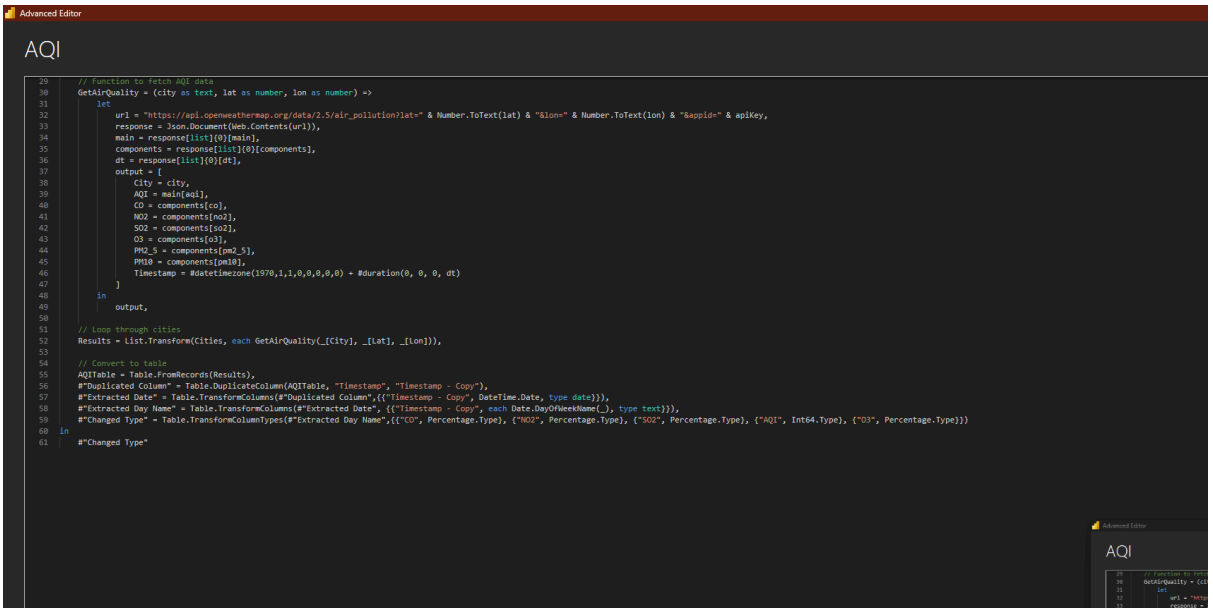
Script for the Api Collection

Weather Data

Display Options

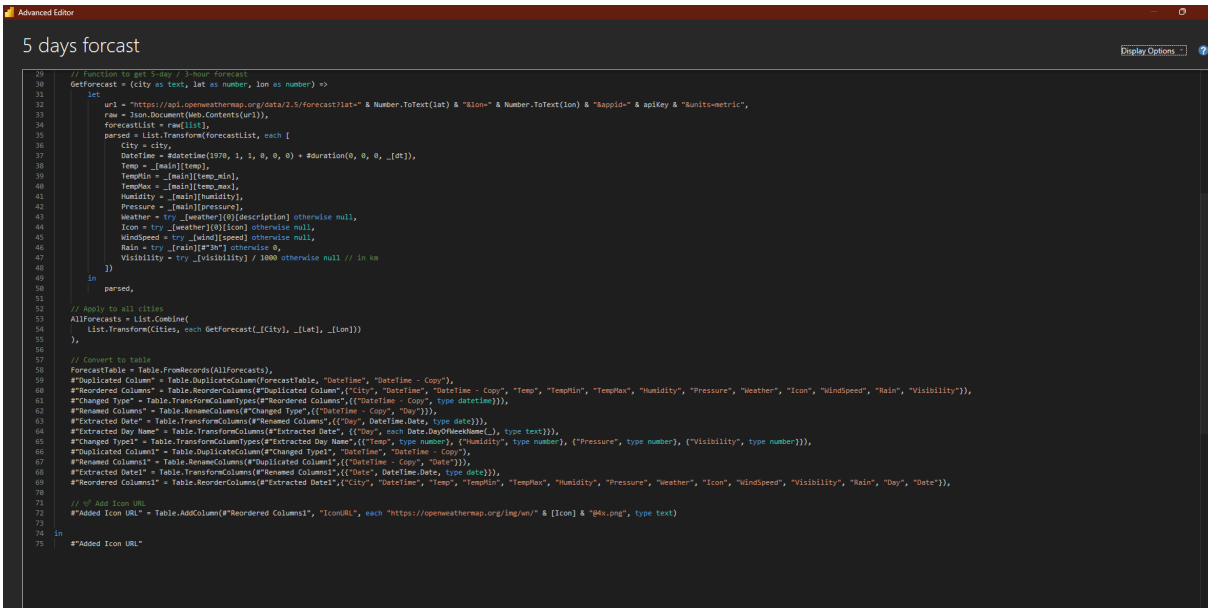
```
let
    // City list with UTC offsets (in hours)
    Cities = {
        [City="Mumbai", Offset=5.5],
        [City="Delhi", Offset=5.5],
        [City="Bengaluru", Offset=5.5],
        [City="Hyderabad", Offset=5.5],
        [City="Ahmedabad", Offset=5.5],
        [City="Chennai", Offset=5.5],
        [City="Kolkata", Offset=5.5],
        [City="Surat", Offset=5.5],
        [City="Pune", Offset=5.5],
        [City="Jaipur", Offset=5.5],
        [City="Akola", Offset=5.5],
        [City="Patur", Offset=5.5],
        [City="Lucknow", Offset=5.5],
        [City="Bhopal", Offset=5.5],
        [City="Indore", Offset=5.5],
        [City="Patna", Offset=5.5],
        [City="Nashik", Offset=5.5],
        [City="Vadodara", Offset=5.5],
        [City="Guwahati", Offset=5.5],
        [City="Amritsar", Offset=5.5]
    },

    // Your OpenWeatherMap API key
    apiKey = "4e7c77db2faedff7828cdb76796e85d3",

    // Function to fetch weather data
    GetWeather = (city as text, offset as number) =>
        let
            url = "https://api.openweathermap.org/data/2.5/weather?q=" & city & "&appid=" & apiKey & "&units=metric",
            response = Json.Document(Web.Contents(url)),

            main = response[main],
            weather = response[weather]{0},
            wind = response[wind],
            visibility = try response[visibility] otherwise null,
            sys = response[sys],
            dt = response[dt],

            sunriseUTC = #datetime(1970, 1, 1, 0, 0, 0) + #duration(0, 0, 0, sys[sunrise]),
            sunsetUTC  = #datetime(1970, 1, 1, 0, 0, 0) + #duration(0, 0, 0, sys[sunset]),
            UpdatedAt = #datetimezone(1970, 1, 1, 0, 0, 0, 0, 0) + #duration(0, 0, 0, dt),


            // Convert UTC to local using city-specific offset
            SunriseLocal = DateTimeZone.SwitchZone(DateTimeZone.From(sunriseUTC), offset),
            SunsetLocal  = DateTimeZone.SwitchZone(DateTimeZone.From(sunsetUTC), offset),


            output = [
                City = city,
                DateTime = DateTime.Date(DateTimeZone.FixedUtcNow() + #duration(0, offset, 0, 0)),
                Temperature = main[temp],
                FeelsLike = main[feels_like],
                Humidity = main[humidity],
                Pressure = main[pressure],
                Weather = weather[main],
                Description = weather[description],
                WindSpeed = wind[speed],
                Visibility = visibility,
                Sunrise = SunriseLocal,
                Sunset = SunsetLocal,
                UpdatedAt = UpdatedAt,
                FetchDate = DateTime.Date(DateTimeZone.FixedUtcNow() + #duration(0, 0, offset * 60, 0))

            ]
        in
            output,

    // Loop through all cities and fetch data
    Results = List.Transform(Cities, each GetWeather(_[City], _[Offset])),

    // Convert to table
    TableResult = Table.FromRecords(Results),

    // Extract time only from Sunrise
    #"Duplicated Column" = Table.DuplicateColumn(TableResult, "Sunrise", "Sunrise - Copy"),
    #"Renamed Columns" = Table.RenameColumns(#"Duplicated Column",{{"Sunrise - Copy", "Sunrise - Time"}}),
    #"Extracted Time" = Table.TransformColumns(#"Renamed Columns",{{"Sunrise - Time", DateTime.Time, type time}}),
    #"Inserted Local Time" = Table.AddColumn(#"Extracted Time", "Local Time", each DateTimeZone.ToLocal([Sunrise]), type datetimezone),
    #"Reordered Columns" = Table.ReorderColumns(#"Inserted Local Time",{"City", "Temperature", "Humidity", "Pressure", "Weather", "Descr
    #"Duplicated Column1" = Table.DuplicateColumn(#"Reordered Columns", "Sunset", "Sunset - Copy"),
    #"Renamed Columns1" = Table.RenameColumns(#"Duplicated Column1",{{"Sunset - Copy", "Sunset - Time"}}),
    #"Calculated Local Time" = Table.TransformColumns(#"Renamed Columns1",{{"Sunset - Time", DateTimeZone.ToLocal, type datetimezone}}),
    #"Changed Type" = Table.TransformColumnTypes(#"Calculated Local Time",{{"Temperature", type number}, {"Humidity", type number}}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Changed Type",{{"Temperature", Int64.Type}, {"WindSpeed", type number}}),
    #"Removed Duplicates" = Table.Distinct(#"Changed Type1", {"UpdatedAt"}),
    #"Duplicated Column2" = Table.DuplicateColumn(#"Removed Duplicates", "UpdatedAt", "UpdatedAt - Copy"),
    #"Renamed Columns2" = Table.RenameColumns(#"Duplicated Column2",{{"UpdatedAt - Copy", "UpdatedAt  Date"}}),
    #"Extracted Date" = Table.TransformColumns(#"Renamed Columns2",{{"UpdatedAt  Date", DateTime.Date, type date}}),
    #"Removed Columns" = Table.RemoveColumns(#"Extracted Date",{"DateTime"})
in
    #"Removed Columns"
```

AQI




```
// Function to fetch AQI data
GetAirQuality = (city as text, lat as number, lon as number) =>
    let
        url = "https://api.openweathermap.org/data/2.5/air_pollution?lat=" & Number.ToText(lat) & "&lon=" & Number.ToText(lon) & "&appid=" & apiKey,
        response = Json.Document(Web.Contents(url)),
        main = response[list]{0}[main],
        components = response[list]{0}[components],
        dt = response[list]{0}[dt],
        output = [
            City = city,
            AQI = main[aqi],
            CO = components[co],
            NO2 = components[no2],
            SO2 = components[so2],
            O3 = components[o3],
            PM2_5 = components[pm2_5],
            PM10 = components[pm10],
            Timestamp = #datetimezone(1970,1,1,0,0,0,0,0) + #duration(0, 0, 0, dt)
        ]
    in
        output,

// Loop through cities
Results = List.Transform(Cities, each GetAirQuality(_[City], _[Lat], _[Lon])),

// Convert to table
AQITable = Table.FromRecords(Results),
#"Duplicated Column" = Table.DuplicateColumn(AQITable, "Timestamp", "Timestamp - Copy"),
#"Extracted Date" = Table.TransformColumns(#"Duplicated Column",{{"Timestamp - Copy", DateTime.Date, type date}}),
#"Extracted Day Name" = Table.TransformColumns(#"Extracted Date", {{"Timestamp - Copy", each Date.DayOfWeekName(_), type text}}),
#"Changed Type" = Table.TransformColumnTypes(#"Extracted Day Name",{{"CO", Percentage.Type}, {"NO2", Percentage.Type}, {"SO2", Percentage.Type}, {"AQI", Int64.Type}, {"O3", Percentage.Type}})
in
#"Changed Type"
```

5 Days Forecast




```
// Function to get 5-day / 3-hour forecast
GetForecast = (city as text, lat as number, lon as number) =>
    let
        url = "https://api.openweathermap.org/data/2.5/forecast?lat=" & Number.ToText(lat) & "&lon=" & Number.ToText(lon) & "&appid=" & apiKey & "&units=metric",
        raw = Json.Document(Web.Contents(url)),
        forecastList = raw[list],
        parsed = List.Transform(forecastList, each [
            City = city,
            DateTime = #datetime(1970, 1, 1, 0, 0, 0) + #duration(0, 0, 0, _[dt]),
            Temp = _[main][temp],
            TempMin = _[main][temp_min],
            TempMax = _[main][temp_max],
            Humidity = _[main][humidity],
            Pressure = _[main][pressure],
            Weather = try _[weather]{0}[description] otherwise null,
            Icon = try _[weather]{0}[icon] otherwise null,
            WindSpeed = try _[wind][speed] otherwise null,
            Rain = try _[rain][#"3h"] otherwise 0,
            Visibility = try _[visibility] / 1000 otherwise null // in km
        ])
    in
        parsed,

// Apply to all cities
AllForecasts = List.Combine(
    List.Transform(Cities, each GetForecast(_[City], _[Lat], _[Lon]))
),

// Convert to table
ForecastTable = Table.FromRecords(AllForecasts),
#"Duplicated Column" = Table.DuplicateColumn(ForecastTable, "DateTime", "DateTime - Copy"),
#"Reordered Columns" = Table.ReorderColumns(#"Duplicated Column",{"City", "DateTime", "DateTime - Copy", "Temp", "TempMin", "TempMax", "Humidity", "Pressure", "Weather", "Icon", "WindSpeed", "Rain", "Visibility"}),
#"Changed Type" = Table.TransformColumnTypes(#"Reordered Columns",{{"DateTime - Copy", type datetime}}),
#"Renamed Columns" = Table.RenameColumns(#"Changed Type",{{"DateTime - Copy", "Day"}}),
#"Extracted Date" = Table.TransformColumns(#"Renamed Columns",{{"Day", DateTime.Date, type date}}),
#"Extracted Day Name" = Table.TransformColumns(#"Extracted Date", {{"Day", each Date.DayOfWeekName(_), type text}}),
#"Changed Type1" = Table.TransformColumnTypes(#"Extracted Day Name",{{"Temp", type number}, {"Humidity", type number}, {"Pressure", type number}, {"Visibility", type number}}),
#"Duplicated Column1" = Table.DuplicateColumn(#"Changed Type1", "DateTime", "DateTime - Copy"),
#"Renamed Columns1" = Table.RenameColumns(#"Duplicated Column1",{{"DateTime - Copy", "Date"}}),
#"Extracted Date1" = Table.TransformColumns(#"Renamed Columns1",{{"Date", DateTime.Date, type date}}),
#"Reordered Columns1" = Table.ReorderColumns(#"Extracted Date1",{"City", "DateTime", "Temp", "TempMin", "TempMax", "Humidity", "Pressure", "Weather", "Icon", "WindSpeed", "Visibility", "Rain", "Day", "Date"}),

// 🌐 Add Icon URL
#"Added Icon URL" = Table.AddColumn(#"Reordered Columns1", "IconURL", each "https://openweathermap.org/img/wn/" & [Icon] & "@4x.png", type text)

in
#"Added Icon URL"
```