**Weather Report AI Automation**

**Purpose**

This automation identifies the **daily high and low temperatures** for a **user-selected location** from a form submission and stores the weather data in Airtable for record-keeping and reporting.

**Overview**

This AI Agent is designed to dynamically extract weather data (temperature highs and lows) for one of several predefined cities based on a form submission. It then stores this data along with the submitted location in an Airtable database.

**How It Works**

This workflow operates as follows:

1. **Form Submission**

Captures user input from a dropdown where the user selects a location (e.g., Toronto, Pune, Dubai, etc.). It triggers the entire workflow.

1. **Switch Node**

Checks the submitted location and forwards the flow to the correct branch based on match rules (equals: Toronto, equals: Pune, etc.).

1. **Set Nodes for Coordinates**

Each branch sets the respective city’s latitude and longitude manually. These nodes are named by city and contain:

**Tech Stack and Tools Used**

* n8n – Automation builder
* Form Trigger Node – Captures the user's selected city
* Switch Node – Routes logic based on selected location
* Set Node(s) – Assign latitude and longitude for selected city
* HTTP Request Node (Weather Pull) – Fetches weather data from [Open-Meteo.com](https://open-meteo.com)
* Code Node – Extracts the high and low temperatures for today
* Airtable Node – Records the data into an Airtable base

You can use alternate tools as per your preference, such as Make.com or Zapier, depending on your platform of choice.

**Setup Steps**

**Estimated Setup Time**: 10 minutes

Before running the workflow, complete the following configurations:

**1. Airtable Login and API Token**

* Sign in to [Airtable](https://airtable.com/)
* Generate a personal access token here:  
  <https://airtable.com/create/tokens/new>
* Use this token to create new Airtable credentials in n8n:
  + Go to **Credentials** > **New** > **Airtable Personal Token**
  + Paste your token and configure the base ID and table name
* Ensure your Airtable base has columns for: location, date, high\_temperature, low\_temperature

**2. Open-Meteo API URL**

* This workflow uses [Open-Meteo](https://open-meteo.com) to fetch hourly temperature data.
* You don’t need authentication for Open-Meteo; it’s a free and public API.

**3. n8n Configuration**

* If self-hosted, ensure you’re running the latest version of n8n with API access enabled.
* Recommended: Save your workflows in version control if building long-term projects.

**4. Timezone Awareness (Optional)**

* Open-Meteo returns data in UTC. If your use case requires local timezone conversions, adjust the API

**Workflow Breakdown (Node-by-Node)**

**1. Form Submission**

Captures user input from a dropdown where the user selects a location (e.g., Toronto, Pune, Dubai, etc.). It triggers the entire workflow.

**2. Switch Node**

Checks the submitted location and forwards the flow to the correct branch based on match rules (equals: Toronto, equals: Pune, etc.).

**3. Set Nodes for Coordinates**

Each branch sets the respective city’s **latitude and longitude** manually.

**4. Weather Pull (HTTP Request Node)**

Uses the latitude and longitude from the selected city to query Open-Meteo's API.

**5. Temperature High and Low (Code Node)**

Parses the hourly temperature data to calculate today’s high and low.

**6. Airtable Connection**

Saves the final output to Airtable:

* Location
* Date
* High temperature
* Low temperature

**Use Cases (Top 5)**

1. **Weather Reporting Dashboards** – Automate data capture and storage for daily updates.
2. **Classroom Projects or Demos** – Teach how conditional logic and APIs work in automation.
3. **Event Planning Tools** – Trigger based on weather forecast to update event organizers.
4. **Travel Agencies** – Store and review weather patterns for popular destinations.
5. **Remote Team Assistants** – Help teams across geographies see local weather summaries.