The live demo of one of the applications can be found here: <https://www.youtube.com/watch?v=3AhD-r2PRLw>

While I was not able to generate a project for the N8N framework, I was able to find the project template thatI could have used in the link provided: <https://n8n.io/integrations/openweathermap/and/radar/>

Using the information from the page along with the applications I have created, I was able to discover the **Pros & Cons** of each framework:

| **Framework** | **Pros** | **Cons** |
| --- | --- | --- |
| **CrewAI** | - Supports **AI agents** for decision-making- Can automate **weather trend analysis** using multiple agents- Good for **AI-driven forecasting** | - Requires **Python coding**- Not ideal for **real-time ML predictions**- Needs **external data sources** for weather inputs |
| **N8N** | - **No-code/low-code** workflow builder- Best for **fetching & processing weather data** from APIs- Supports **automation & alerts** (e.g., weather notifications) | - **Limited AI capabilities**, mostly API-based- Not designed for **predictive modeling**- Relies on **external weather APIs** |
| **LangChain** | - **LLM-powered insights** for weather trends- Great for **natural language-based analysis**- Supports **historical data processing** with vector databases | - Requires **Python coding**- Needs **external ML models** for real-time forecasting- Not optimized for **high-speed automation** |