Tourist Weather Advisor Chatbot
Phase 3: Finalization

Kadzuwa Kuyeri
Matriculation number: 92121883

IU International University of Applied Sciences
MSc in Artificial Intelligence

Project: AI Use Case (DLMAIPAIUC01)
Portfolio

Professor **Tim Schlippe**

November 01, 2025

1. Project Overview

This project presents *WeatherBot*, a conversational chatbot designed to provide real-time weather forecasts and personalized activity recommendations based on meteorological conditions. The WeatherBot chatbot aims to help tourists and residents decide whether a day's weather is suitable for outdoor activities. By combining a simple rule-based decision layer with a conversational NPL model, the chatbot translates weather information into practical recommendations ("Go" or "No-Go").

Developed using **Python**, WeatherBot combines natural language understanding (NLP) and live data integration from the OpenWeatherMap API, using Streamlit for its user interface, enabling users to inquire about weather in any city worldwide.

The main goal of this project is to design a chatbot capable of interpreting user questions in natural language and returning meaningful, actionable advice. This solution addresses the challenge of interpreting weather data intuitively, allowing users to simply ask questions like *"Can I go hiking in Munich today?"* and receive a direct response with weather conditions and a recommendation.

2. Methodology and System Design

The development utilized a **hybrid approach** combining ChatterBot, a machine learning conversational engine, with spaCy for natural language processing (NLP) to extract city and activity intents. Python served as the programming language, with PyCharm IDE in a virtual environment ensuring reproducibility. Phase 2 implemented a prototype using simulated weather data for 11 Mozambican cities, trained with 10 question-answer pairs and rule-based logic evaluated conditions. In Phase 3, optimization integrated the **OpenWeatherMap API** for live forecasts, enhanced the recommendation engine with rain probability and wind speed, and deployed a **Streamlit web interface** for accessibility.

The chatbot architecture integrates three layers (System Flow Diagram included in the folder):

- **User Interface (Streamlit):** A simple, web-based conversational environment enabling interactive communication.
- **Processing Layer (Python + NLP):** The system extracts cities and intents using SpaCy, then requests data from the OpenWeatherMap API.
- **Weather Intelligence Layer:** Applies rule-based logic (temperature, condition, humidity, wind speed) to generate personalized activity recommendations.

3. Implementation

Development was performed using PyCharm as the IDE within a virtual environment. All required Python packages (Streamlit, SpaCy, Requests) were installed through the interpreter. The OpenWeatherMap API was used for real-time weather data retrieval. WeatherBot offers consistent responses and gracefully handles unknown or misspelled city inputs.

- **ChatterBot** was chosen for its simplicity and educational suitability, enabling quick prototype creation.
- **spaCy** provides industrial-grade NLP while remaining lightweight.
- **Rule-based logic** ensures clear, explainable decisions suitable for early-stage models.

The modular design ensures flexibility and scalability, allowing for the integration of other APIs (e.g., air quality or UV index).

4. Results and Reflection

The bot successfully combines conversational design and weather reasoning for a personalized experience. The enhanced version of WeatherBot successfully expands coverage to global cities and integrates real-time environmental parameters, offering more accurate and contextual responses. Tests with different cities demonstrated that the bot correctly identifies location names, retrieves corresponding weather data, and generates logical suggestions. The Streamlit interface improved user engagement, transforming WeatherBot from a simple terminal-based assistant into a more accessible, visual, and user-friendly application. Minor limitations include the dependence on API key validity and occasional delays in API response times.

The hybrid approach proved effective, balancing ML and rule-based logic. Phase 1 concept was tested and expanded, with Phase 2 prototype laying the groundwork for Phase 3 web app. The journey from concept to final product highlights the importance of iterative development, and technical adaptability in creating successful AI applications.

The complete project implementation and documentation are publicly available at: https://github.com/KaddyKuyeri/WeatherBot-by-Kadzuwa-Kuyeri

References

ChatterBot Machine learning, conversational dialog engine (2025). ChatterBot (Version 1.2.8) [Computer software documentation]. https://docs.chatterbot.us/

Explosion AI. (2025). spaCy: Industrial-strength NLP. https://spacy.io/

OpenWeather. (2025). Current weather data – API documentation. https://openweathermap.org/current

OpenWeatherMap (2024). Weather API Documentation. https://openweathermap.org/api

Python Software Foundation. (2025). General Python FAQ: General Information [Documentation]. https://docs.python.org/3/faq/general.html#general-information

Streamlit Inc. (2025). Streamlit Python library documentation. https://docs.streamlit.io/