**👩‍💻 Overview**

**Mellie** is a personalized, emoji-friendly AI chatbot designed to engage in natural language conversation, provide real-time weather information, and fetch factual summaries from Wikipedia. The bot is powered by OpenAI's **DialoGPT-medium** model and integrates external knowledge sources through the **Wikipedia** API and **OpenWeatherMap API**.

**⚙️ Technologies Used**

* **Programming Language**: Python 3
* **Libraries**:
  + transformers (Hugging Face): Loads the DialoGPT model
  + torch: Inference backend for the language model
  + wikipedia: Wikipedia query and summary API
  + requests: HTTP requests for external API calls (weather)
  + re: Regex operations for city extraction and input cleaning
  + dotenv: Environment variable loader (for API keys)

**🌐 Features**

**1. 🌤️ Weather Information**

* Extracts the city name from user input using regular expressions.
* Uses the **OpenWeatherMap API** to fetch real-time weather data.
* Returns temperature, humidity, description, and wind speed.
* Handles API errors and city not found scenarios gracefully.

**2. 📖 Wikipedia Summaries**

* Cleans the user query to remove filler phrases (e.g., "what is", "tell me about").
* Uses the wikipedia.summary() and wikipedia.search() functions to fetch summaries.
* Handles disambiguation and page errors using a fallback loop.
* Returns rich summaries for educational and factual queries.

**3. 🧑‍📚 Natural Conversation (DialoGPT)**

* Uses HuggingFace's **microsoft/DialoGPT-medium** model.
* Maintains context across conversation turns using chat\_history\_ids.
* Responds to casual greetings, questions, and conversation inputs.
* Prepends a custom intro prompt to make the bot friendly and personalized.

**4. ☺️ Personality & Emojis**

* Responds with emojis to sound cheerful and engaging.
* Greets the user by name (Layla) using hardcoded examples.
* Uses consistent tone and templates for good morning/evening greetings.

**🔍 Input Handling**

* Input text is matched against known phrases:
  + Greetings ("hello", "hi", etc.)
  + Gratitude ("thank you")
  + Farewells ("bye")
  + Keywords like "weather" and question words for Wikipedia
* Queries that don’t match weather or Wikipedia routes go to DialoGPT.

**📊 Flow Diagram (High Level)**

1. User enters text.
2. System checks for exit keywords.
3. Checks for gratitude or greetings and replies accordingly.
4. If "weather" in query → get\_weather()
5. If question word or "what is", "who is" → search\_wikipedia()
6. Otherwise, use DialoGPT for a natural response.

**📃 Environment Setup**

* .env file must contain:

WEATHER\_API\_KEY=your\_openweathermap\_key

* Required packages:

pip install transformers torch wikipedia python-dotenv requests

**⚠️ Known Issues / Limitations**

* Wikipedia search may return to broad or unrelated topics.
* DialoGPT is not trained to follow strict context or instructions.

**✅ Future Improvements**

* Add custom NLU (Natural Language Understanding) for better intent recognition.
* Build a local FAQ knowledge base for improved fact answering.
* Integrate more external APIs (e.g., time zones, movies, books).
* Dynamically detect and store user's name.
* Add fallback responses when Wikipedia results are weak.

**🤝 Contributors**

* **Laila Hassan ElHadidy** (Developer)
* Guided and supported by OpenAI tools