Simple Chat Bot : Documentation

High-Level Design (HLD)

Purpose

The **ChatBot** is a CLI-based interactive chatbot that provides two main services:

1. Fetching real-time weather information for a given city using the WeatherAPI.
2. Performing basic arithmetic operations.

The bot is designed to enhance user interaction by logging conversations and operations performed, and it uses exception handling to manage errors gracefully.

Architecture Components

1. **User Input Handling:** The chatbot receives input from users, guiding them through the options of weather lookup or calculation.
2. **Logging:** All user interactions and errors are logged to a file specific to each user's session.
3. **API Integration:** The bot fetches weather data from a third-party API (WeatherAPI) and parses the response for display.
4. **Error Handling:** The bot provides user-friendly messages for invalid input and handles API or computation errors gracefully.

Low-Level Design (LLD)

Classes and Methods

AiChatBot: Main chatbot class.

- **Attributes:**
    - **name:** The user's name, captured at initialization.
    - **log_file_name:** Dynamically named log file for each user session.
- **Methods:**
    - **__init__:** Initializes the chatbot, capturing the user's name and setting up logging.
    - **greet:** Main method to present user choices and route to appropriate functionality.
    - **ask_weather:** Prompts for a city name, fetches weather data, and displays the information.

- **ask_calculation:** Prompts for numbers and operations, performs the calculation, and displays the result.
- **get_weather_data:** Retrieves weather information from the API and handles API errors.
- **calculate:** Performs the selected arithmetic operation.
- **log_conversation:** Logs user interactions and errors.

Example Workflow

1. User initiates the chatbot.
2. AiChatBot welcomes the user and logs the interaction.
3. User selects a service (e.g., "Get weather").
4. AiChatBot fetches and displays weather information, logging all relevant data and user interaction.

System Design

Environment and Setup

- **Requirements:**
  - **requests**: For API calls.
  - **dotenv:** To securely load API keys from environment variables.
  - **logging:** For conversation and error logging.
- **Environment Variables:**
  - **WEATHER_API_KEY:** WeatherAPI key loaded from a .env file.

Logging

Logs are stored in **chatbot_log_<User_Name>.txt** with details of each user interaction and any errors.

Error Handling

- **Invalid Input:** Prompts the user to enter correct data.
- **API Failure:** Informs the user of connectivity or data retrieval issues.
- **Mathematical Errors:** Division by zero is handled with a specific error message.

API Request and Response

Weather API Endpoint

- **Endpoint:** http://api.weatherapi.com/v1/current.json
- **Request:** GET
- **Parameters:**
    - **key:** Your WeatherAPI key.
    - **q:** City name (provided by user input)

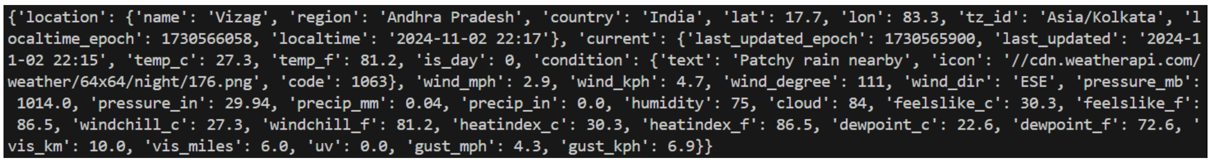
Example Request

GET http://api.weatherapi.com/v1/current.json?key=YOUR_API_KEY&q=Vizag

Response Structure

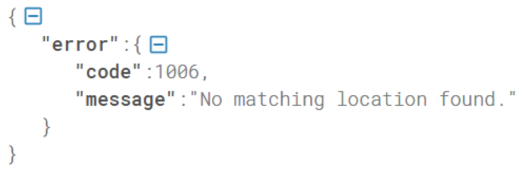
Success:

- **location:** Contains city, region, and country.
- **current:** Weather conditions including temperature, humidity, wind speed, and description**.**



```
{'location': {'name': 'Vizag', 'region': 'Andhra Pradesh', 'country': 'India', 'lat': 17.7, 'lon': 83.3, 'tz_id': 'Asia/Kolkata', 'l
ocaltime_epoch': 1730566058, 'localtime': '2024-11-02 22:17'}, 'current': {'last_updated_epoch': 1730565900, 'last_updated': '2024-1
1-02 22:15', 'temp_c': 27.3, 'temp_f': 81.2, 'is_day': 0, 'condition': {'text': 'Patchy rain nearby', 'icon': '//cdn.weatherapi.com/
weather/64x64/night/176.png', 'code': 1063}, 'wind_mph': 2.9, 'wind_kph': 4.7, 'wind_degree': 111, 'wind_dir': 'ESE', 'pressure_mb':
 1014.0, 'pressure_in': 29.94, 'precip_mm': 0.04, 'precip_in': 0.0, 'humidity': 75, 'cloud': 84, 'feelslike_c': 30.3, 'feelslike_f':
 86.5, 'windchill_c': 27.3, 'windchill_f': 81.2, 'heatindex_c': 30.3, 'heatindex_f': 86.5, 'dewpoint_c': 22.6, 'dewpoint_f': 72.6, '
vis_km': 10.0, 'vis_miles': 6.0, 'uv': 0.0, 'gust_mph': 4.3, 'gust_kph': 6.9}}
```

- **Error:** Returns an error message indicating the issue (e.g., invalid API key or location not found).



```
{
    "error":{
        "code":1006,
        "message":"No matching location found."
    }
}
```

Fields Description

- **location.name:** City name.
- **location.region:** Region within the country.
- **location.country:** Country name.
- **current.temp_c:** Current temperature in Celsius.
- **current.condition.text**: Brief weather description.

- **current.humidity:** Humidity percentage.
- **current.wind_kph**: Wind speed in km/h.

User Guide

Using the Chatbot

1. **Startup:** Run the chatbot program. It will prompt you for your name.
2. **Choose a Service:**
   - **Option 1:** Enter a city name to get the current weather.
   - **Option 2:** Enter two numbers and an operation (addition, subtraction, multiplication, division) to perform a calculation.
   - **Option 3:** Exit the program.
3. **View Logs:** Check chatbot_log_<Your_Name>.txt for a record of your interactions, including weather requests and calculations.