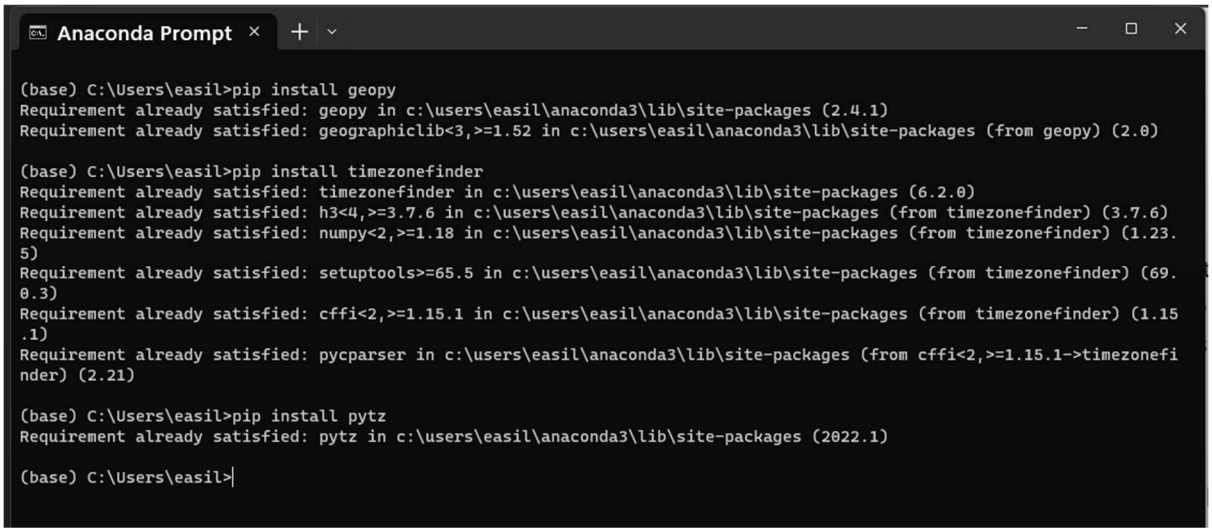
Problem Statement:

Develop a weather application that allows users to input a city name and obtain real-time weather information for that location. The application should display current weather conditions, temperature, humidity, wind speed, and pressure. Additionally, it should provide a visual representation of weather information through icons and labels.

Features:

Install this libraries in Anaconda prompt before running the Jupiter note book

Pip install geopy
Pip install timezonefinder
Pip install pytz

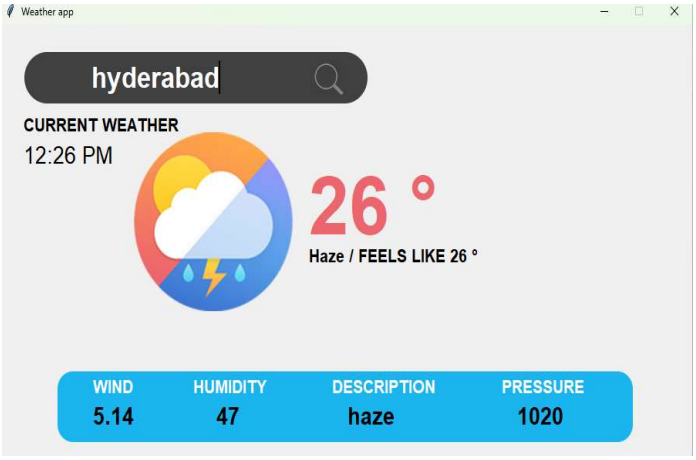


1. **Search Functionality:**
    - Users can input the name of a city in the search box.
    - Clicking on the search icon retrieves and displays the current weather information for the specified city.
2. **Weather Information Display:**
    - Current time and date are displayed at the top of the application.
    - Weather details such as temperature, condition, humidity, wind speed, and pressure are presented.
    - Descriptive labels for each weather parameter are included for user understanding.
3. **Visual Elements:**
    - Icons and images related to search, weather condition, and application branding are integrated.
    - A bottom box with a distinct design adds aesthetics to the application.
4. **Error Handling:**

- If the user enters an invalid city name or encounters any other error during the weather data retrieval process, an error message is displayed using a messagebox.

Implementation:

1. **Search Functionality:**
    - Utilizes the **geopy** library to fetch the geographical coordinates (latitude and longitude) of the entered city.
    - Retrieves the timezone based on the coordinates to display the current time.
    - Uses the OpenWeatherMap API to get real-time weather data for the specified city.



2. **Weather Information Display:**
    - Parses the JSON response from the OpenWeatherMap API to extract relevant weather information.
    - Displays temperature in both Celsius and Fahrenheit for user convenience.
    - Utilizes labels to display weather conditions, humidity, wind speed, and pressure.
3. **Visual Elements:**
    - Incorporates images and icons for search, branding, and weather condition representation.
    - Implements a bottom box with a unique design for an enhanced visual appeal.
4. **Error Handling:**
    - Catches exceptions during the data retrieval process and displays an error message if the user inputs an invalid city name or encounters any other issue.