**Weather Forecasting and Historical Data Application Overview**

This application provides a comprehensive solution for fetching and displaying current weather data, as well as historical weather data. Additionally, it supports exporting the data to CSV and JSON formats. The application leverages the Open Meteo API for weather forecasts and other APIs for historical weather data.

**Code Documentation**

**1. WeatherForecastingSystem Class**

* **Purpose:** Handles fetching and displaying current weather data, and supports exporting data to CSV and JSON formats.
* **Constructor:**

WeatherForecastingSystem(const string& apiKey, const string& baseUrl)

+ \*\*Parameters:\*\*

- `apiKey`: API key for authentication.

- `baseUrl`: Base URL of the weather forecast API.

* **Methods:**
  + **string fetchWeatherData(const string& location)**
    - Fetches weather data for the specified location.
    - **Parameters:**
      * **location**: City name or location for weather data.
    - **Returns:** Weather data in JSON format.
  + **void displayWeatherData(const string& data)**
    - Displays fetched weather data on the console.
    - **Parameters:**
      * **data**: Weather data in JSON format.
  + **void exportWeatherDataToCSV(const string& data, const string& filename)**
    - Exports weather data to a CSV file.
    - **Parameters:**
      * **data**: Weather data to be exported.
      * **filename**: Name of the CSV file.
  + **void exportWeatherDataToJSON(const string& data, const string& filename)**
    - Exports weather data to a JSON file.
    - **Parameters:**
      * **data**: Weather data to be exported.
      * **filename**: Name of the JSON file.

**2. HistoricalWeatherSystem Class**

* **Purpose:** Handles fetching and displaying historical weather data.
* **Constructor:**

1HistoricalWeatherSystem(const string& apiKey, const string& baseUrl, const string& historicalBaseUrl)

+ \*\*Parameters:\*\*

- `apiKey`: API key for authentication.

- `baseUrl`: Base URL of the weather API.

- `historicalBaseUrl`: Base URL of the historical weather data API.

* **Methods:**
  + **string fetchHistoricalWeatherData(const string& location, const string& date)**
    - Fetches historical weather data for the specified location and date.
    - **Parameters:**
      * **location**: City name or location for historical data.
      * **date**: Date in YYYY-MM-DD format.
    - **Returns:** Historical weather data in JSON format.
  + **void displayHistoricalWeatherData(const string& data)**
    - Displays fetched historical weather data on the console.
    - **Parameters:**
      * **data**: Historical weather data in JSON format.

**Setup and Running the Application**

**1. Prerequisites**

* Ensure you have the cURL library installed and properly configured.
* Include the nlohmann/json library for JSON handling.

**2. Configuration**

* Replace placeholder API keys in the main function with your actual API keys.
* Ensure the URLs for the forecast and historical weather APIs are correct.

**3. Compilation**

* Use a C++ compiler that supports C++11 or later.
* Compile the code with appropriate flags to link against the cURL and JSON libraries. For example:

g++ -o weather\_app main.cpp -lcurl -ljson

**4. Running the Application**

* Execute the compiled application from the command line:

/weather\_app

* The application will fetch current weather data for London, display it, and export it to CSV and JSON files. It will also fetch and display historical weather data for a specified date.

**Functionality**

* **Current Weather:**
  + Fetches current weather data from the Open Meteo API.
  + Displays weather data on the console.
  + Exports data to **current\_weather.csv** and **current\_weather.json**.
* **Historical Weather:**
  + Fetches historical weather data from the specified historical weather data API.
  + Displays historical weather data on the console.

**Sample Output**

* **Console Output:**

Weather Data: {"temperature\_2m\_max":22.5,"temperature\_2m\_min":15.8}

Historical Weather Data: {"temperature\_max":21.0,"temperature\_min":14.0}

* **CSV File (current\_weather.csv):**

weather\_data

{"temperature\_2m\_max":22.5,"temperature\_2m\_min":15.8}

* **JSON File (current\_weather.json):**

{"temperature\_2}