**COMPUTER ENGINEERING WORKSHOP**

# S.E. (CIS) OEL REPORT

**Project Group Members:**

MUHAMMAD UBAID CS-22079

SYEDA HIBRA TARIQ CS-22102

ALISHBA GUL AHMED CS-22106

# BATCH: 2022

**Jan** 2024

**Department of Computer and Information Systems Engineering**

**NED University of Engg. & Tech., Karachi-75270**

# CONTENTS

**S.No. Page No.**

1. **Problem Description** 1

1. **Code Overview** 1-3

1. **Dependencies**  3-4

1. **Future Improvements** 4

1. **Results** 4-5

1. **Conclusion**  5

**CHAPTER 1**

# PROBLEM DESCRIPTION

A C-written command-line application called the Weather Information Retrieval System uses the WeatherAPI to retrieve real-time weather data. It uses jansson to parse JSON answers and libcurl to send HTTP queries. The application gathers meteorological information for a certain area, analyses it, and outputs the results to the console. It also adds the data to a text file for historical documentation.

**CHAPTER 2**

# CODE OVERVIEW

The program consists of the following components:

* **struct MemoryStruct:**

Structure to store memory information during HTTP request.

* **makeHttpRequest Function:** 
  + Constructs the API URL with the provided API key and location. o Uses libcurl to perform an HTTP GET request to the WeatherAPI. o Handles response data and HTTP status code.
  + Returns the API response as a dynamically allocated string.
* **processWeatherData Function:** 
  + Placeholder function for any future data processing logic.
  + Currently, it is called after each API response but has no specific functionality.

* **WriteCallback Function:** 
  + Callback function used by libcurl to handle the response data. o Dynamically allocates memory to store the response. o Appends the response data to the allocated memory.

* **Main Function:** 
  + Defines the WeatherAPI URL, API key, and location. o Calls makeHttpRequest to fetch weather data. o Parses the JSON response using jansson. o Prints relevant weather information to the console. o Appends weather information to "raw\_data.txt." o Frees memory allocated during API response processing.
  + The main function reads mean temperature, mean wind speed, and mean visibility from the "processed\_data.txt" file. It checks if the mean visibility is below a specified threshold (4.0 km). If visibility is below the threshold, a low visibility alert is triggered:
    1. A message is printed to the console, advising precautionary measures.
    2. The alert is appended to the "alert\_log.txt" file with a timestamp.
    3. An email notification is sent to a specified recipient using the curl command.
* **File Handling:**

The program opens and reads the "processed\_data.txt" file to retrieve mean weather values.

The file pointer is reset to the beginning of the file after reading the values.

* **Alert System:**

If mean visibility is below the threshold, an alert is triggered with informative messages.

An entry is appended to the "alert\_log.txt" file, recording the alert and its timestamp.

* **Email Notification:**

If the visibility is low, an email is sent to a specified recipient using the curl command.

The email includes information about the low visibility condition.

* **Script Components:** 
  1. **API Calls Loop:**
  + API Call Limit: The script iterates a specified number of times (limit=3) to make API calls.
  + API Call Interval: After each API call, the script waits for a specified interval (interval=60 seconds) before making the next call. o Logging: The script appends the API call timestamp and other information to the "raw\_data.txt" file.
  + API Execution: The cew executable is responsible for making the API call.
  1. **Data Processing:** 
     + Processing Executable: After completing the API calls, the script executes the proccess\_data executable. o Data Saving: The processed data is saved to the "proccessed\_data.txt" file.
  2. **Email Execution:** 
     + Email Executable: The script concludes by executing the email executable.

**CHAPTER 3**

# DEPENDENCIES

The program relies on the following libraries:

* **libcurl:** Used for making HTTP requests.
* **jansson:** Used for parsing JSON responses.
* Ensure these libraries are installed on the system before compiling and running the program.

**CHAPTER 4**

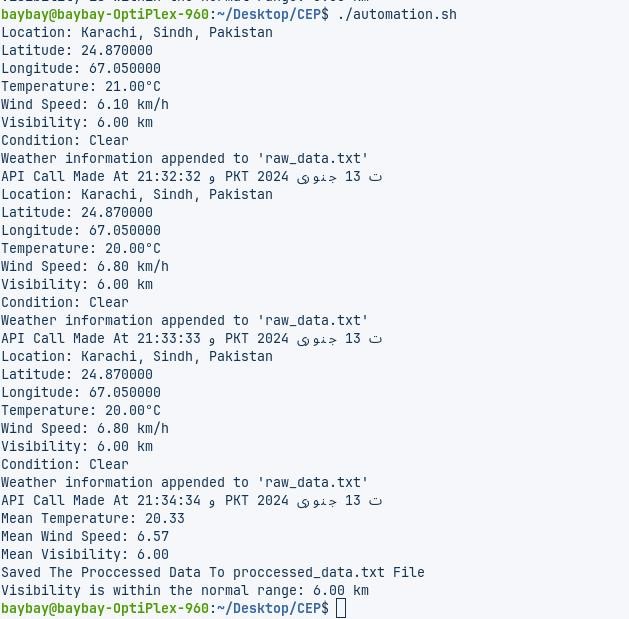
# FUTURE IMPROVEMENTS

* Implement more robust error handling for HTTP requests and JSON parsing.
* Enhance the processWeatherData function for additional data processing.
* Allow user input for location, API key, and other parameters.
* Currently working on a specific location, in future can be implemented for different locations.

**CHAPTER 5**

# RESULTS

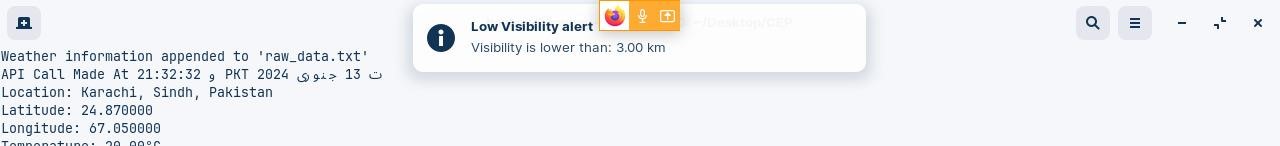
**API’S WEATHER INFO:**



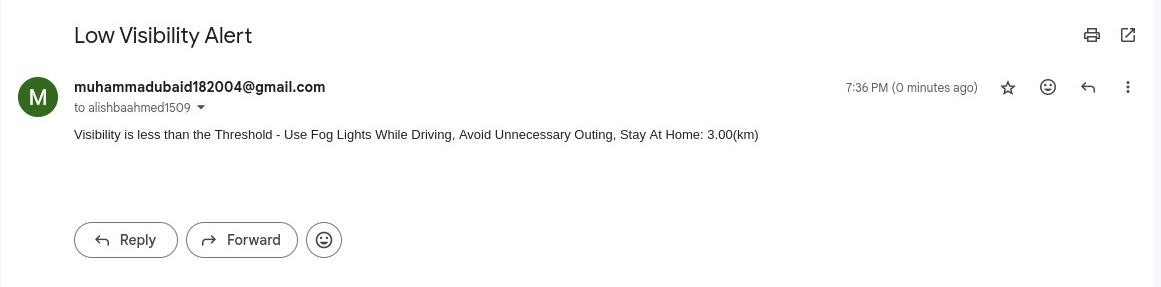
**FILE’S COMPILATION:**



**NOTIFICATION ALERT:**



**ALERT ON EMAIL:**



**CHAPTER 5**

# CONCLUSION

The Weather Information Retrieval System successfully retrieves and displays real-time weather information using the WeatherAPI. It demonstrates the use of libcurl for HTTP requests and jansson for JSON parsing in a C programming environment. The program serves as a foundation for further improvements and customization based on specific project requirements.