

# **COMPUTER ENGINEERING WORKSHOP**

## **S.E. (CIS) OEL REPORT**

**Project Group ID:** <https://github.com/Umer2547/Cew-OEL-Project-.git>

Syed Muhammad Umer Iqbal CS-23031

Ather Hasan CS-23041

**BATCH:** 2023

**Department of Computer and Information Systems Engineering**

**NED University of Engg. & Tech.,**

**Karachi-75270**

## Table of Contents

Problem Description.....	3
Methodology .....	3
Key Learnings and Challenges .....	4
Results .....	5
Test Case Runs .....	1

## Problem Description

The development of the Weather Application addresses the critical need for accessible and reliable real-time weather data. This project was initiated to harness contemporary API technologies to fetch and process meteorological data from the OpenWeather API, focusing on essential environmental parameters such as temperature, humidity, wind speed, and atmospheric pressure. The goal is to provide a scalable, robust solution for various sectors needing precise weather forecasting and analysis, ranging from agriculture to urban planning.

### Core Objectives of the Weather Application System

1. **Real-Time Data Retrieval:** Retrieve real-time environmental data from the OpenWeather API to ensure up-to-date weather information is available for users.
2. **Data Storage and Retrieval:** Transfer and store both raw and processed data in files, organized in a manner that allows easy access and retrieval whenever necessary.
3. **Automation Through Shell Scripting:** Utilize efficient shell scripting to automate the tasks associated with data acquisition and analysis, ensuring the system operates with minimal manual intervention.
4. **Efficient Memory Management:** Implement the system using pointers and dynamic memory allocation to optimize memory usage and enhance performance.
5. **Real-Time Alerts:** Issue real-time alerts based on critical weather states, such as extreme temperatures or high wind speeds, to inform users immediately of significant environmental changes.

### Scope of the Project

1. **Data Retrieval:** The system continuously updates real-time environmental data, such as temperature and humidity, from a freely accessible API at specified intervals.
2. **Data Processing:** Processes raw data to derive useful information, such as mean values and thresholds for triggering alerts, which are crucial for monitoring and responding to environmental conditions.
3. **Data Reporting:** Organizes and stores the processed data in files for easy access and reference, facilitating detailed reporting and analysis for users and decision-makers.

## Methodology

### System Overview:

The Weather Application is designed to automate the retrieval and processing of real-time weather data, utilizing a structured approach that combines modern software engineering practices with efficient data handling techniques. The system architecture is divided into distinct components, each responsible for a specific aspect of the application's functionality, ensuring modularity and ease of maintenance.

### Component Details:

1. **API Interaction Components:**
  - a. The application uses the cURL library to make API calls to OpenWeather, a publicly accessible weather data source that provides comprehensive environmental data.
  - b. It features advanced error handling to manage issues such as network disruptions or data fetch errors, ensuring the application remains robust and reliable.
  - c. The API responses are received in JSON format, which are then standardized within the application to facilitate uniform data processing.
2. **Data Handling Components:**

- a. This component is tasked with parsing the JSON responses from the API to extract crucial weather parameters like temperature, humidity, and wind speed.
  - b. A series of data transformation functions convert this raw data into meaningful insights, calculating averages and identifying trends over designated time periods.
  - c. It also manages all file operations, categorizing data into 'raw' for immediate needs and 'processed' for archival and trend analysis, ensuring efficient data storage and retrieval.
3. **Main Application Flow:**
- a. This central component orchestrates the overall operation of the application, managing the sequence of data fetching, processing, and storage.
  - b. It ensures seamless communication between the API interaction and data handling components, maintaining a continuous loop of operations to fetch and process data at predefined intervals.
4. **Automation and Scheduling:**
- a. A dedicated shell script is configured to execute the application continuously, which allows the system to operate autonomously without manual intervention.
  - b. The script ensures that the application is persistently active, fetching and updating weather data every 600 seconds, thereby providing real-time updates.
5. **Alert Conditions and Triggers:**
- a. The application is configured with thresholds for triggering alerts:
    - i. **High Temperature Alert:** Triggered when the temperature exceeds 30°C.
    - ii. **High Wind Speed Alert:** Activated when wind speed surpasses 10 meters per second.
    - iii. **Low Visibility Alert:** Issued when visibility drops below 1000 meters.
  - b. These conditions are adjustable based on user requirements or geographical specificity, allowing flexibility in alert management.
6. **Notification Mechanism:**
- a. When an alert condition is met, the main application coordinates with the data handling component to generate a notification.
  - b. Notifications are implemented using system calls that interact with the operating system's native notification system or through external communication services like email or SMS, depending on the system configuration.

## **Key Learnings and Challenges**

### **Key Learnings:**

1. **Real-Time Data Integration:** We learned integrating and handling real-time data using cURL and JSON, which proved crucial for maintaining dynamic updates within the application.
2. **Automated System Operations:** Our ability to script for continuous operations enhanced the reliability and efficiency of the application, ensuring it performed consistently under various conditions.
3. **Advanced Alert Mechanisms:** The project deepened our understanding of real-time alert systems, enabling us to implement responsive mechanisms that adjust to specific environmental thresholds effectively.

**Challenges:**

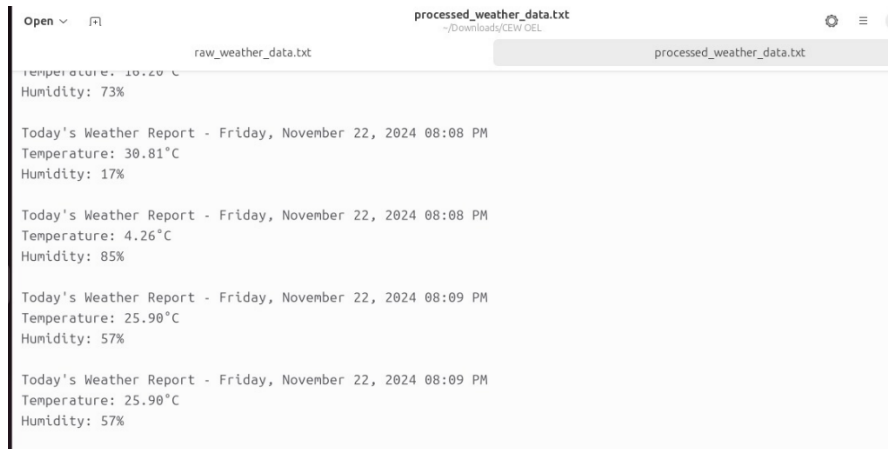
4. **Optimization Challenges:** We encountered difficulties in optimizing the application to efficiently manage resource constraints, which required us to balance performance with speed and accuracy.
5. **Adaptation to User Needs:** Adapting the system to meet diverse user requirements and geographic specifics presented a complex challenge, necessitating iterative testing and customization.

**Results**

The Weather Application has successfully met its design and functionality objectives, demonstrating robust performance in real-time environmental data retrieval, processing, and reporting. The system effectively gathers and updates weather data at specified intervals, utilizing advanced data processing techniques to extract meaningful metrics such as average temperature and humidity. Automated alerts have proven functional and timely, notifying users of critical weather conditions as they occur. Moreover, the application's use of dynamic memory allocation and efficient shell scripting has enhanced its performance and reliability, ensured minimal resource wastage and optimized operational efficiency. Overall, the project has not only achieved its intended goals but has also laid a strong foundation for future enhancements and scalability.

# Test Case Runs

## 1. Processed Data File

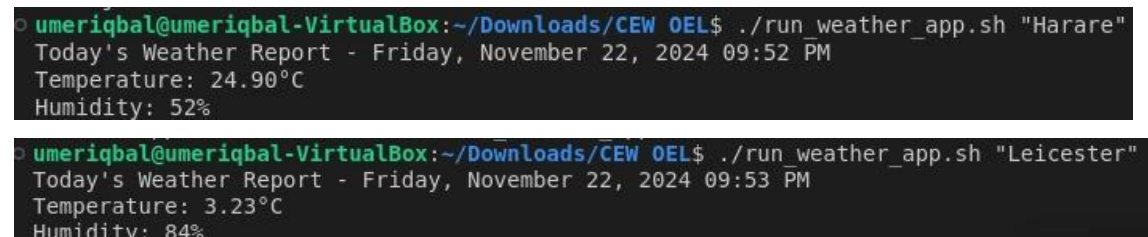
A screenshot of a text editor window titled 'processed\_weather\_data.txt'. The editor shows the contents of a file named 'raw\_weather\_data.txt'. The text contains several weather reports for Friday, November 22, 2024, at 08:08 PM and 08:09 PM. Each report includes temperature and humidity values.

```
temperature: 10.20°C  
Humidity: 73%  
  
Today's Weather Report - Friday, November 22, 2024 08:08 PM  
Temperature: 30.81°C  
Humidity: 17%  
  
Today's Weather Report - Friday, November 22, 2024 08:08 PM  
Temperature: 4.26°C  
Humidity: 85%  
  
Today's Weather Report - Friday, November 22, 2024 08:09 PM  
Temperature: 25.90°C  
Humidity: 57%  
  
Today's Weather Report - Friday, November 22, 2024 08:09 PM  
Temperature: 25.90°C  
Humidity: 57%
```

## 2. Notification Alerts



## 3. Output At Terminal

A screenshot of a terminal window showing the execution of a script named 'run\_weather\_app.sh'. The script is run twice, once for 'Harare' and once for 'Leicester'. The output shows the date, time, temperature, and humidity for each location.

```
umeriqbal@umeriqbal-VirtualBox:~/Downloads/CEW OEL$ ./run_weather_app.sh "Harare"  
Today's Weather Report - Friday, November 22, 2024 09:52 PM  
Temperature: 24.90°C  
Humidity: 52%  
  
umeriqbal@umeriqbal-VirtualBox:~/Downloads/CEW OEL$ ./run_weather_app.sh "Leicester"  
Today's Weather Report - Friday, November 22, 2024 09:53 PM  
Temperature: 3.23°C  
Humidity: 84%
```