COMPUTERENGINEERINGWORKSHOP

S.E. (CIS) OEL REPORT Project

Group ID:

Muhammad Ahmed Raza	CS-23097
Hafiz Muhammad Owais Zafar	CS-23098
Muhammad Yousuf Mateen	CS-23135

BATCH: 2023

Department of Computer and Information Systems Engineering
NED University of Engg. & Tech.,
Karachi-75270 CONTENTS

Contents

PROBLEM DESCRIPTION

The Environmental Monitoring System Weather app (YRO Weather App) is a lightweight Command Line Interface (CLI)-based weather monitoring application designed for Linux users. Its main purpose is to provide real-time weather updates and alerts using the OpenWeather API. The app is built with a focus on simplicity, performance, and customization. Key features include real-time data retrieval, JSON parsing using `cJSON`, and critical condition alerts implemented via Linux system calls and syslog.

1. Key Objectives

1. Enable real-time weather monitoring through a simple CLI interface.
2. Provide alerts for hazardous weather conditions such as high temperature, high humidity, and strong winds.
3. Log events and data in a structured manner for future reference or analysis.

The problem addressed by this application is the lack of an efficient, terminal-based weather monitoring tool for Linux users that can provide both real-time updates and critical condition alerts.

METHODOLOGY

The Environmental Monitoring System Weather app (YRO Weather App) was developed following a systematic approach to ensure its reliability and usability across different Linux distributions. The methodology can be outlined in the following steps:

1. Development Environment and Setup:

- The application was developed in a Linux environment using the GCC compiler.
- Dependencies like `libcurl` for HTTP requests and `cJSON` for JSON parsing were installed.
- A shell script (`retrieve_data.sh`) was created to automate the build and execution process.

2. API Integration and Data Parsing:

- The OpenWeather API was used to fetch real-time weather data.
- API responses were parsed using the `cJSON` library to extract relevant details such as temperature, humidity, and wind speed.

3. Real-Time Alerts

- Alerts were implemented using Linux system calls. For example, an alert is triggered when the temperature exceeds 35°C, the humidity exceeds 80%, or the wind speed surpasses 15 m/s.
- Alerts are logged using syslog, providing a record of critical events.

4. Background Execution

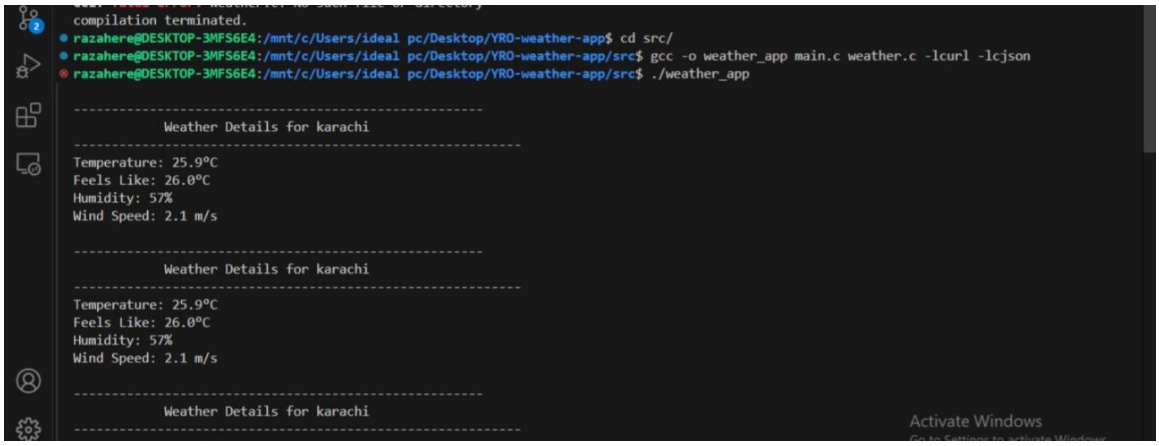
The application can run as a background process using `nohup`. This ensures it operates continuously without interruption, even if the user logs out of the system.

RESULTS

The Environmental Monitoring System Weather app (YRO Weather App) delivered significant results in terms of functionality and performance. It effectively met its objectives by providing the following outcomes:

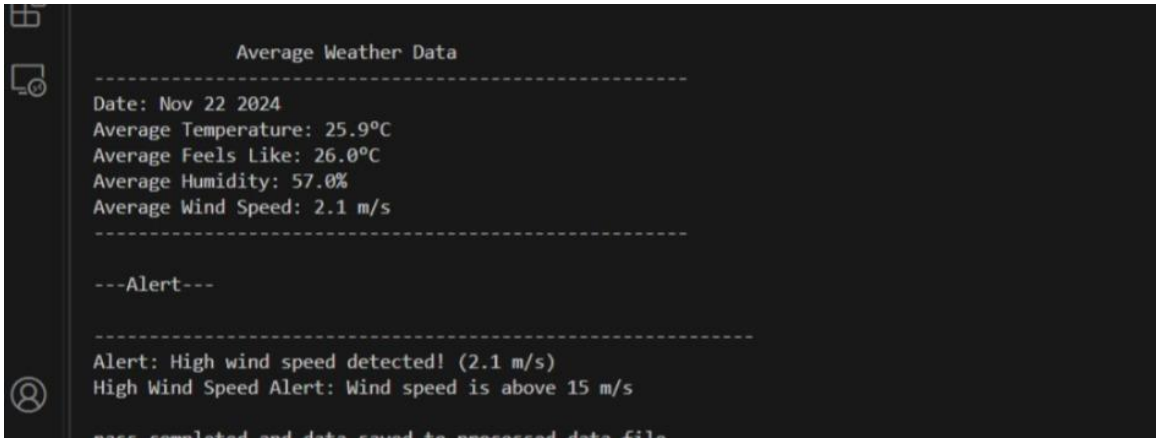
1. Real-Time Weather Data

- The application retrieves and displays weather information such as temperature, humidity, and wind speed accurately and in real-time.



2. Critical Alerts:

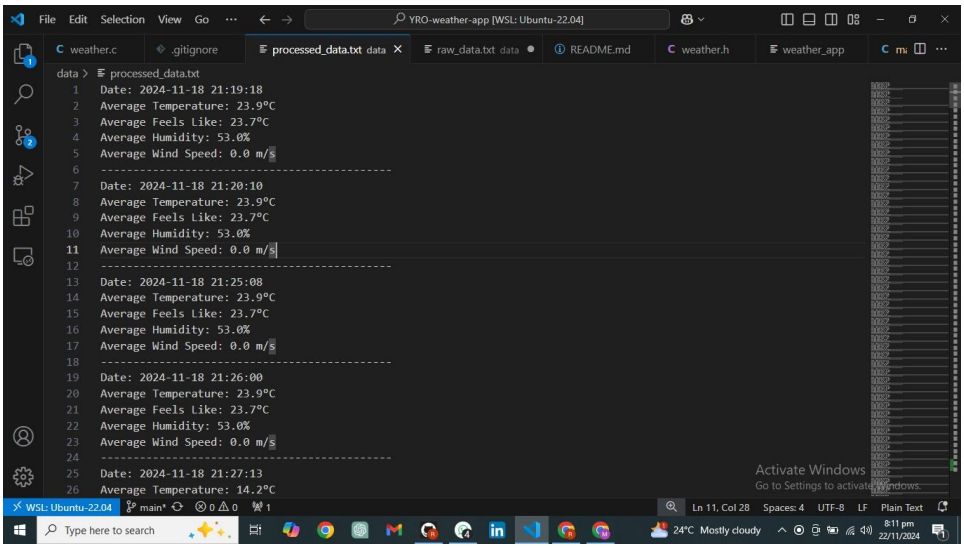
- Alerts for hazardous weather conditions (e.g., high temperature, humidity, or wind speed) were successfully implemented and logged. Users are notified immediately of critical events.



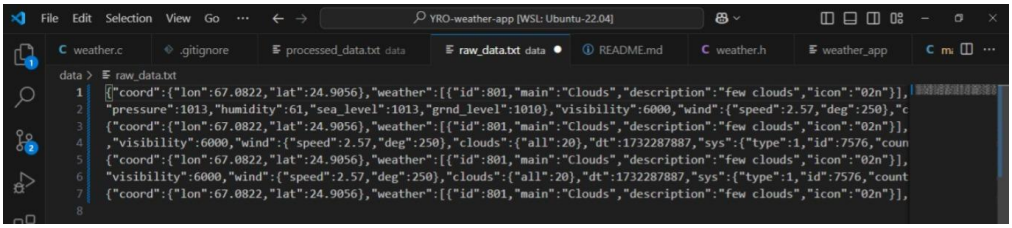
3. Data Storage and Logging:

- The app stores both raw and processed weather data in files for future reference.

i. Processed Data:

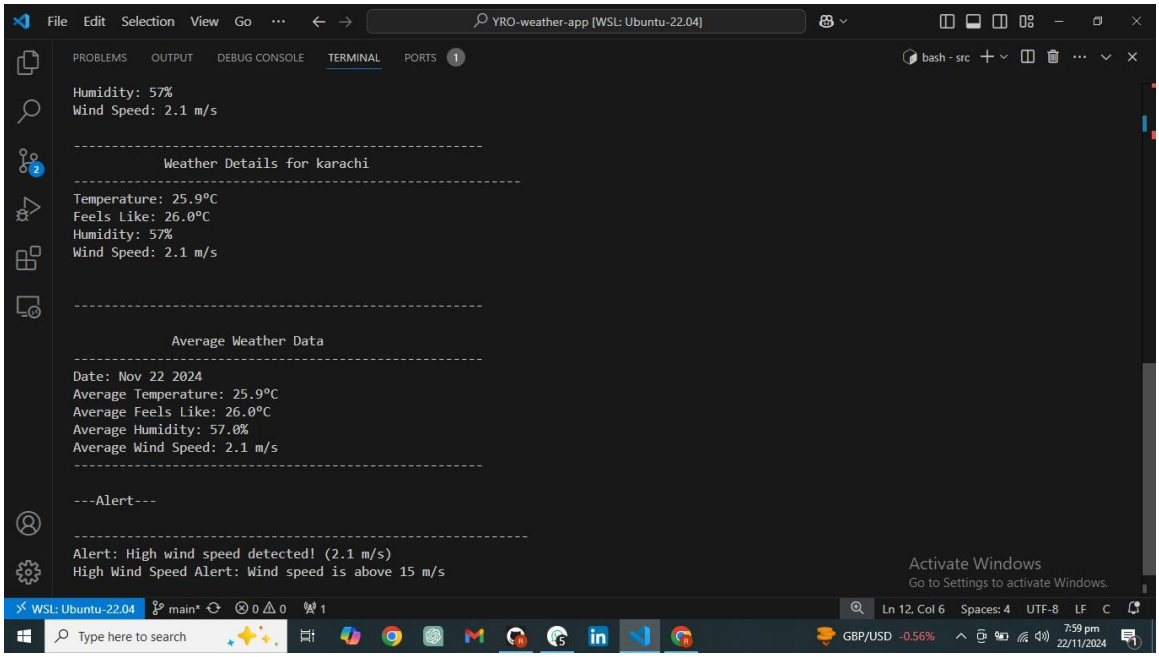


ii. Raw Data:



4. Seamless Integration:

- The application integrates well with Linux systems, using syslog for logging and supporting background execution through `nohup`.



5. Performance Analysis:

- The app was tested in various scenarios, including normal and extreme weather conditions. It performed reliably with minimal resource usage, making it ideal for lightweight systems.

DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING

BACHELORS IN COMPUTER SYSTEMS ENGINEERING

Course Code: CS-219

Course Title: Computer Engineering Workshop

Open Ended Lab

SE Batch 2023, Fall Semester 2024

Grading Rubric

TERM PROJECT

Group Members:

Student No.	Name	Roll No.
S1	**Muhammad Ahmed Raza**	**CS-23097**
S2	**Hafiz Muhammad Owais Zafar**	**CS-23098**
S3	**Muhammad Yousuf Mateen**	**CS-23135**

CRITERIA AND SCALES				Marks Obtained		
				S1	S2	S3
Criterion1: Has the student implemented an efficient and scalable solution for data retrieval, processing, and reporting?						
0	1	2	3			

The student has not even implemented a basic solution that meets the project's requirements.	The student has implemented a basic solution that meets the project's requirements but may lack optimization in certain aspects.	The student has implemented a proficient and well-optimized solution.	The student has implemented an exceptionally efficient and scalable solution.			
Criterion 2: Has student demonstrated a strong understanding of C programming fundamentals?						
0	1	2	3			
The student doesn't have basic understanding of C programming fundamentals.	The student exhibits a basic understanding of C programming fundamentals.	The student demonstrates a strong understanding of C programming fundamentals.	The student demonstrates an exceptional understanding of C programming fundamentals.			
Criterion 3: How well written is the report?						
0	1	2	3			
The submitted report is unfit to be graded.	The report is partially acceptable.	The report is complete and concise.	The report is exceptionally written.			
			Total Marks:			