COMPUTER ENGINEERING WORKSHOP

S.E. (CIS) OEL REPORT

Project Group ID:

ALISHA ATHER	CS-01
MEHAK DUSEJA	CS-29
MUHAMMAD MAAZ WALI KHAN	CS-36

BATCH: 2023

Department of Computer and Information Systems Engineering

NED University of Engg. & Tech.,

Karachi-75270

Table of Contents

ENVIRONMENTAL MONITORING SYSTEM

INTRODUCTION:

The Environmental Monitoring System is a software project designed to keep an eye on the weather in Karachi, providing real-time insights into environmental conditions. Using data from an external weather API, the program fetches, processes, and reports on key metrics like temperature, pressure, and humidity. With built-in automation and notification features, the system makes monitoring effortless and proactive.

KEY FEATURES:

- **Fetching Weather Data with CURL**
  The program uses the powerful CURL library to communicate with the OpenWeatherMap API. It retrieves real-time weather information for Karachi, including:
  i. Temperature (converted to Celsius).
  ii. Atmospheric Pressure.
  iii. Humidity Levels.

- **Instant Desktop Notifications**
  The system integrates with Ubuntu's notification system to display alerts on your desktop. If weather thresholds are crossed (e.g., high temperature, low humidity), users are immediately notified. Notifications are clear and actionable, allowing users to react in real time.

- **Automated Monitoring**
  This program doesn't require manual input once set up. It fetches data, processes it, and checks for alerts on its own. By automating the workflow, it ensures continuous monitoring without user intervention.

- **Clear and Modular Code**
  The program's code is structured for readability and maintainability:
  A dedicated header file (data.h) organizes the core data structures (WeatherData) and function prototypes. Using structs ensures weather data is logically grouped and easy to manage.

- **Parsing JSON Responses**
  The program uses the cJSON library to make sense of the raw JSON data from the API. It extracts key information like temperature, pressure, and humidity, converting it into meaningful insights.

- **Easy-to-Understand Reports**
  Processed data is saved to a file (processed_data.txt) for later analysis. Reports include averages for temperature, pressure, and humidity over time.

- **Struct Usage**

The project employs a struct named WeatherData to represent and store weather-related information for Karachi, including temperature, pressure, and humidity. Structs enhance code organization and encapsulation of related data, contributing to better maintainability.

CHALLENGES FACED:

- **Desktop Notifications:**

Ensuring notify-send worked reliably required testing in a desktop session and resolving environment-specific issues.

- **Temperature Conversion:**

The API provides temperatures in Kelvin, which needed to be accurately converted to Celsius for user-friendly reporting.
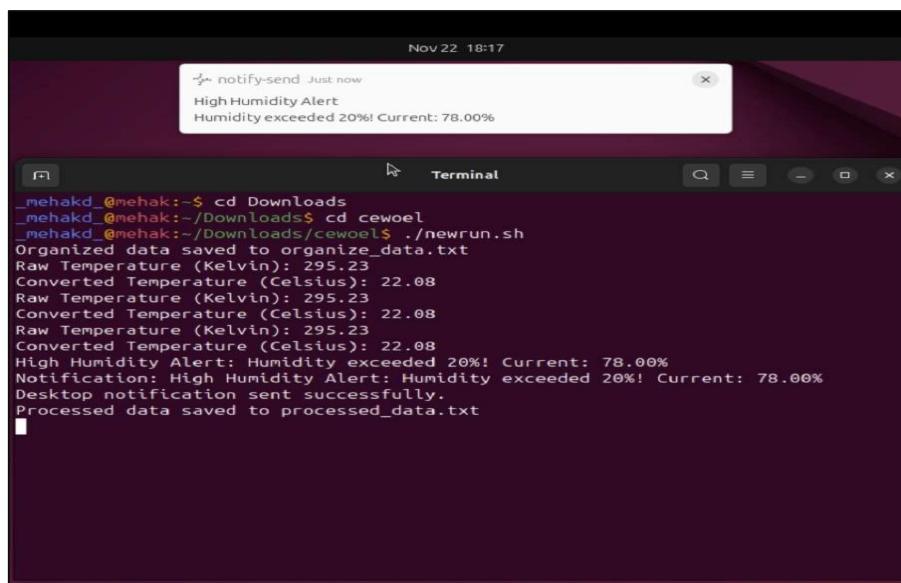
- **Automated Processing:**

Setting up the program to handle repeated API calls and data processing without user intervention was tricky but rewarding.

CONCLUSION:

This project combines automation, real-time data handling, and actionable alerts into a single, easy-to-use program. It's tailored specifically for Karachi, making it a valuable tool for staying informed about local weather conditions. Whether you're a casual user or someone relying on environmental data for decision-making, this system keeps you updated effortlessly.

RESULTS: