COMPUTER ENGINEERING WORKSHOP

S.E. (CIS) OEL REPORT

Project Group ID: 03

NAME OF MEMBER #1 - Zainab CS-23150
NAME OF MEMBER #2 – Fathima Raihana CS-23151

BATCH: 2023

Department of Computer and Information Systems Engineering

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

CONTENTS

S.No.

PROBLEM DESCRIPTION

Monitoring environmental conditions, such as temperature and humidity, is crucial for various purposes, including agriculture, healthcare, industrial processes, and ensuring safety in everyday life. Sudden changes in these conditions can lead to serious consequences, such as crop damage, equipment failures, or health risks. While there are systems available to monitor these factors, they often face several limitations:

1. **Real-Time Monitoring:** Many existing systems struggle to provide up-to-the-minute data. This can delay important decisions when rapid action is required.
2. **Data Analysis and Logging:** It is not enough to collect raw data; the data must be analyzed to identify trends, detect unusual values (anomalies), and be stored for later review or reporting.
3. **Alert Mechanisms:** A reliable way to notify users of critical conditions is often missing. For example, users should be instantly alerted when the temperature is too high or humidity exceeds safe levels.
4. **Cost and Complexity:** Many monitoring solutions are expensive or too complicated for general use, requiring advanced setups or technical expertise.
5. **Integration with Online Services:** Modern systems need to work with external tools, like APIs for fetching data from online sources and email services for sending alerts.

This project aims to overcome these limitations by developing a simple yet effective temperature and humidity monitoring system. The system will:

- **Collect Data Automatically:** Retrieve real-time temperature and humidity information from an online API.
- **Process and Analyze Data:** Extract meaningful insights, such as the current temperature, humidity levels, and potential risks.
- **Log Information:** Keep a detailed record of the data, including timestamps, for future reference or compliance purposes.
- **Send Alerts:** Automatically send email notifications when the temperature or humidity crosses pre-set thresholds, helping users take timely action.

This project ensures that users can monitor environmental conditions in real-time, stay informed about critical changes, and respond proactively to avoid risks.

Methodology

This project focuses on creating a real-time temperature and humidity monitoring system. The following steps outline the approach:

1. Requirements Analysis

- **Problem:** The need for real-time environmental monitoring.
- **Goal:** Automate data retrieval, process and analyze it, and trigger alerts when thresholds are crossed.

2. System Design

- **Modules:**
    - **Data Retrieval:** Fetches data from an external weather API.
    - **Data Processing:** Extracts and processes temperature, humidity, and dew point.
    - **Alert System:** Sends email alerts if values exceed predefined limits.
    - **Logging:** Logs temperature and humidity readings for analysis.

3. Data Retrieval and API Integration

- Integrated with a weather API to retrieve data in JSON format.
- Used the cURL library to make HTTP requests to fetch real-time data.

4. Data Processing

- Parsed the JSON data using the json-c library.
- Extracted temperature, humidity, and calculated dew point.
- Implemented calculations for temperature thresholds and anomaly detection.

5. Alert System

- Integrated the libcurl library to send email notifications when critical thresholds are reached.

6. Data Logging

- Logged temperature and humidity data with timestamps in a text file (temperature_log.txt) for future analysis.

7. Testing and Validation

- Conducted tests to ensure accurate data retrieval, processing, and alert functionality.
- Verified the system's performance during real-time monitoring and logging.

8. Obtaining Output

To run the system and get real-time temperature and humidity data:

1. **Setup:**

    - Ensure that the required libraries (e.g., libcurl, json-c) are installed.
    - Configure the API URL and email credentials in the appropriate files (e.g., data_retrieve.c, email_alert.c).

2. **Running the System:**

    - Compile the project using **make** command in the terminal.
    - Run the compiled executable. The system will start fetching temperature and humidity data from the external API at regular intervals.
    - Output includes:

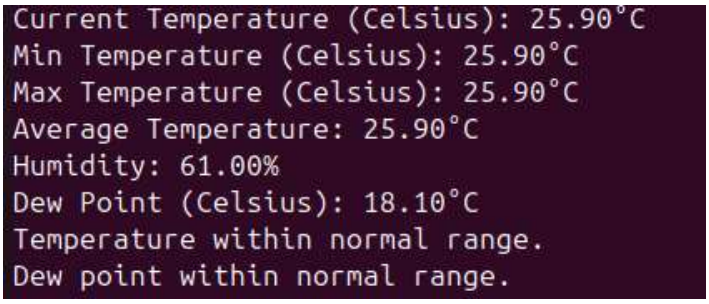
        - **Real-time readings**: Displayed on the console (current temperature, humidity, dew point).
        - **Logs**: Recorded in a file (temperature_log.txt) with timestamped data.
        - **Alerts**: If the temperature or humidity exceeds the set threshold, an email alert is sent.

Result

The system was successfully implemented to monitor temperature and humidity in real-time. The following outcomes were observed during testing:

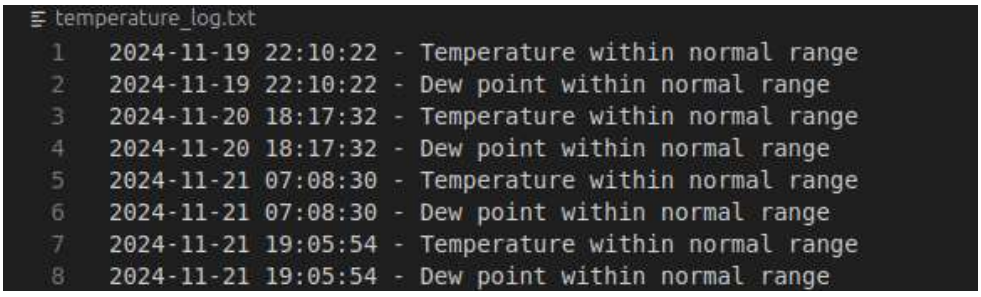
1. Real-Time Data Monitoring:

- The system correctly retrieved temperature and humidity data from the API at regular intervals.



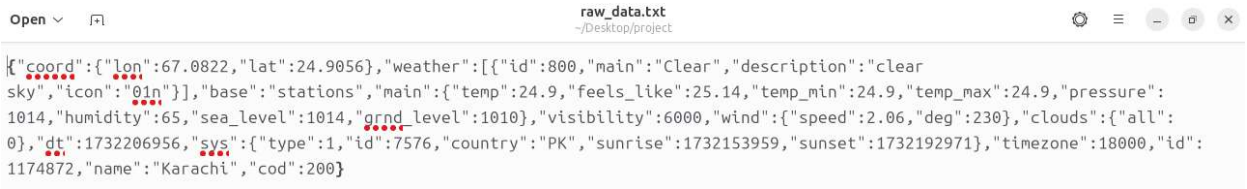
2. Data Logging:

- Data was logged successfully to `temperature_log.txt` with timestamped entries.



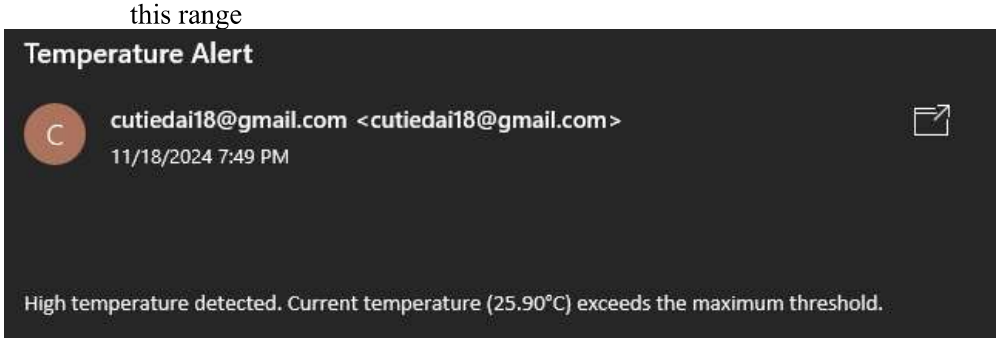
4. Storing Raw Data:

- The system retrieves weather data from the API and logs the raw JSON response in a file called `raw_data.txt`. This file contains the complete raw response from the API, which includes various weather parameters such as temperature, humidity, pressure, and weather description



3. Email Alerts:

- When the temperature exceeded 35°C, an email alert was sent successfully to the configured recipient.
- When the temperature dropped below 10°C, an email alert was triggered and sent to the recipient.
    - ❖ To test the functionality of the email alerts, I temporarily modified the temperature range to trigger alerts between 10°C and 25°C. This adjustment was made to check whether the system could successfully send email alerts within this range



- If the dew point exceeded 20°C or fell below 10°C, the system sent an email alert to notify the recipient of the change.