COMPUTER ENGINEERING WORKSHOP

S.E. (CIS) OEL REPORT

Project Members:

HUZAIFA DANIYAL	CS-116
HAMZA NAVEED	CS-130

BATCH: 2022

13 January, 2024

Department of Computer and Information Systems Engineering

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

Table of Contents

Integrated Environmental Monitoring System

Introduction:

The implemented system serves as an integrated environmental monitoring solution, written in C, that interacts with a free API to retrieve real-time environmental data. This report provides an overview of the major components and functionalities of the system, including data retrieval, processing, analysis, storage, reporting, automation, optimization, and real-time alerts.

Data Retrieval:

In this program we have used Curl library to interact with the WeatherAPI from api.weatherapi.com to retrieve real-time environmental data for a fixed city, Karachi.

Data Processing and Analysis:

We have used the cJSON library, extracting relevant information such as the current temperature. ProcessWeatherdata and temperaturedata are responsible for handling JSON data.

.

Data Storage:

Both raw and processed Temperature data are stored in separate files named rawData.json and processedData.txt, respectively. The saveToFile function handles the file writing operations, ensuring data persistence.

Report Generation:

We have generated a temperature report, including the highest, lowest, and average temperatures. The report is saved to a file temperatureReport.txt for future reference.

Real-Time Alerts:

Alerts are displayed on the console and appended mail.txt file, and an email notification is sent to the stakeholders using the Gmail SMTP server.

Usage and Execution:

Execute the program to fetch real-time data from the api.weatherapi.com API.
Raw data is saved to rawData.json file.
Temperature report is saved to temperatureReport.txt file.
There is also a processeData.txt text file which save the processing data.
Stakeholders mail data are saved to mail.txt file.
Loop.sh bash file does the iterations for running the code for 24 hours and sending mail.

Automation and Integration:

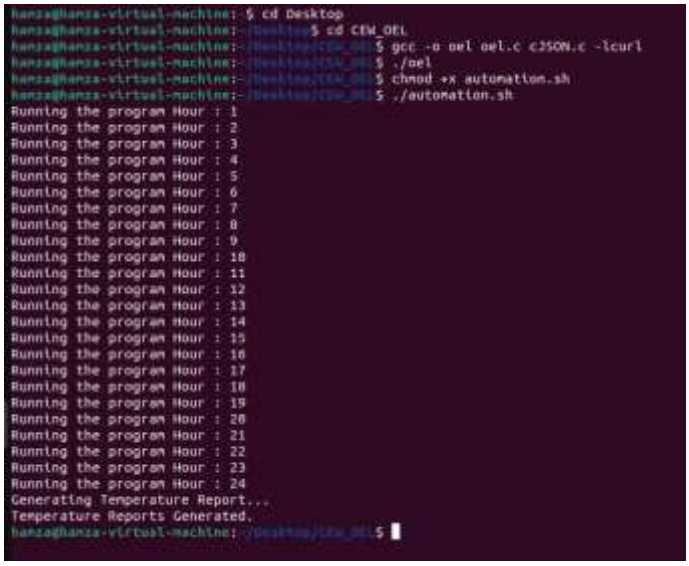
Utilizing shell scripts, the system automates tasks such as data retrieval, processing, and report generation. A job is scheduled to execute the C program every hour for 24 hours, ensuring that the environmental data is continuously updated and analyzed at regular intervals. This automation enhances the system's efficiency and ensures timely monitoring of environmental conditions

Documentation and Code Organization:

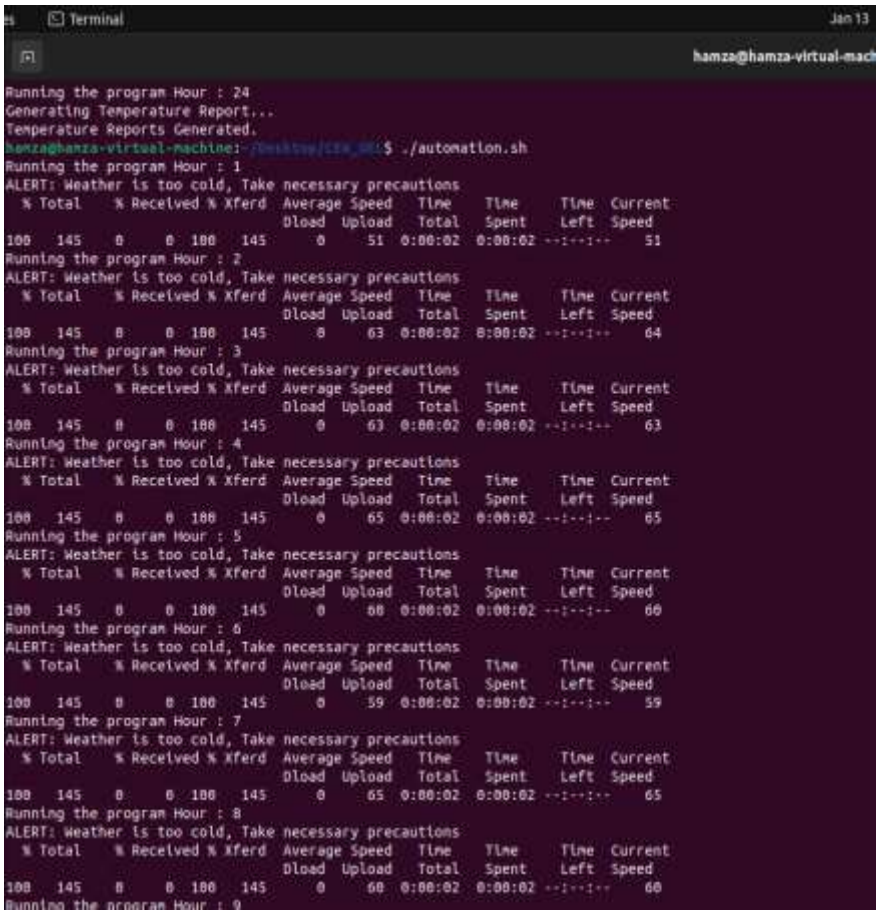
Adequate comments are present in the C code, explaining key functionalities and procedures. The Bash script includes comments to clarify each section's purpose. A detailed report provides an overview of the entire system.

Result:

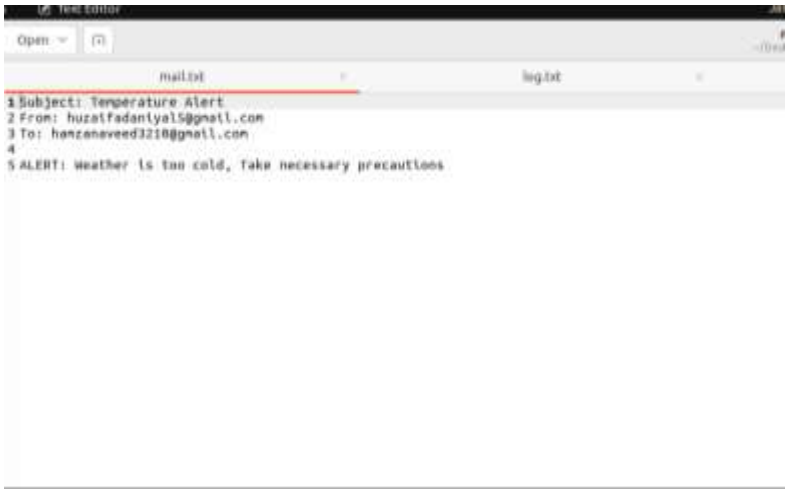
TERMINAL:



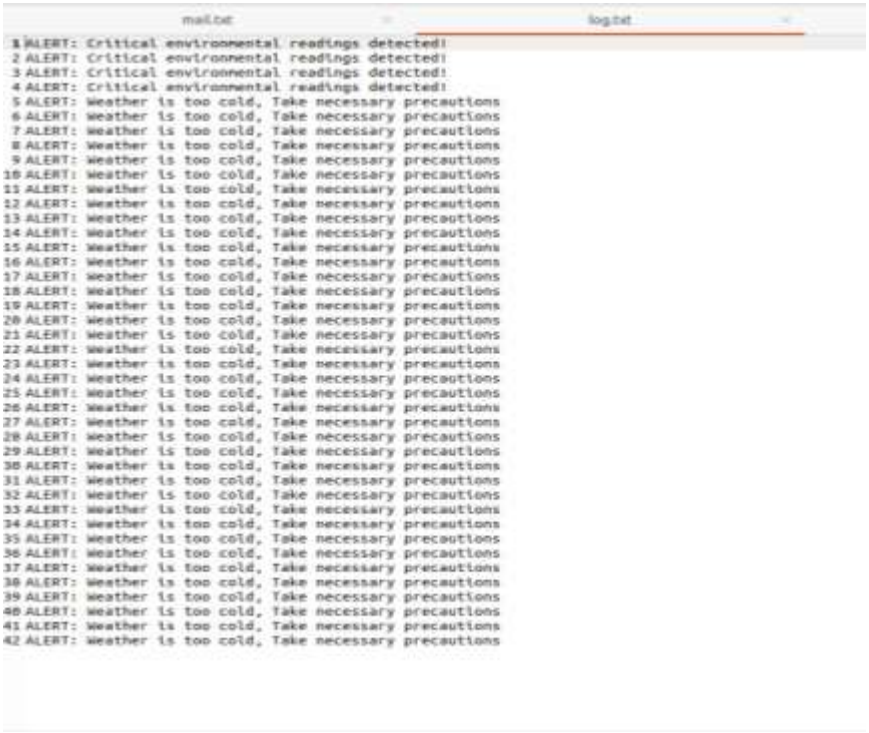
TEMPERATURE REPORT:



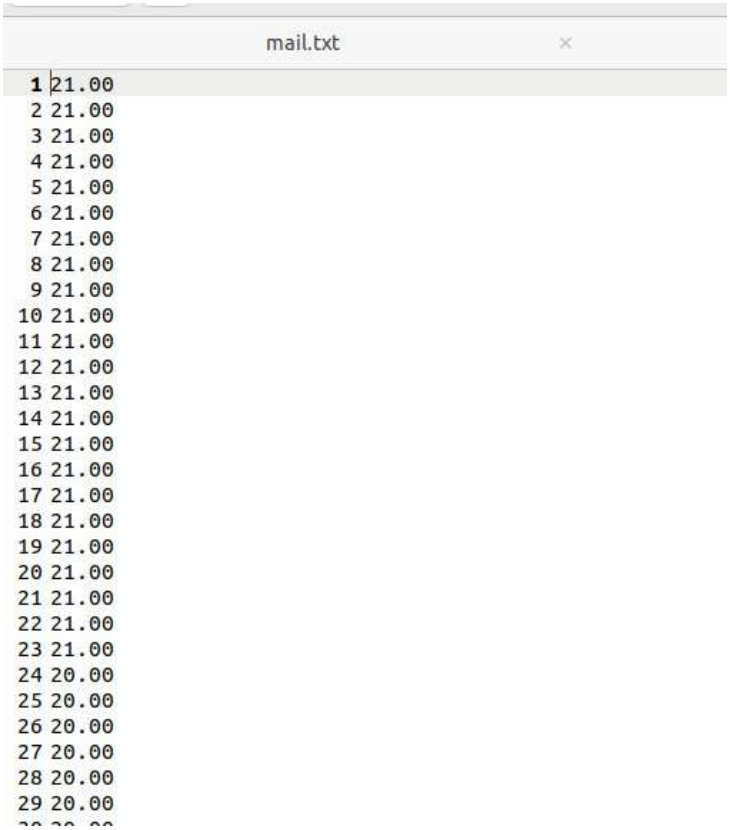
MAIL:



LOG FILE:



PROCESSED DATA:



Conclusion:

The implemented environmental monitoring system successfully retrieves, processes, analyzes, stores, and reports real-time environmental data. Automation and integration with a Bash script enhance the system's usability, and real-time alerts ensure prompt notifications for critical conditions. Further optimizations, code modularization, and comprehensive documentation can be considered for future enhancements.