



# **CEW ENGINEERING WORKSHOP**

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

### **PROJECT GROUP:**

M.Hamza Raza Khan (CS-23059)

Jibran Saleem (CS-23061)

Syed Anas Azim (CS-23051)

***BATCH 2023***

**Department of Computer and Information  
Systems Engineering**

NED University of Engineering & Technology, Karachi-75270

## **Contents**

<b>S.No</b>		<b>Page No.</b>
1.	Problem Description.....	2
2.	Methodology.....	3
3.	Results.....	6

### **PROBLEM DESCRIPTION**

Construct an integrated environmental monitoring system in C, covering a range of fundamental concepts and practical applications. The project involves interacting with a free API that provides real-time environmental data. The system's core functionalities include data retrieval, processing and reporting. Requirements of the project include:

- Interact with a free API to retrieve real-time environmental data (e.g., temperature, humidity).
- Store raw and processed data in files.
- Create shell scripts to automate tasks such as data retrieval and processing.
- Utilize pointers and dynamic memory allocation in the C program to optimize data manipulation and enhance efficiency
- Implement real-time alerts using Linux system calls to notify relevant personnel of critical environmental readings.
- Use header files to modularize the C code and enhance code readability.

## **METHODOLOGY:**

To enhance **readability**, **maintainability**, and **scalability**, the program was organized into multiple file:

**Header Files(.h):** Defines the function prototypes and interfaces for use in other modules.

**Source Files(.c):** Handles API interaction, Data processing , Alert generation

**Main Program(learning.c):** Acts as the central control unit, coordinating the various modules.

### **Function Of each File:**

#### **1. api.c:**

Handles API interaction with Open Weather Map to fetch weather information. Its key components are memory management , curl integration ,call back function and error handling

#### **2. api.h:**

Defines the function prototypes and interfaces for use in other modules. Ensures modularity by separating interface declarations from implementation.

#### **3. for\_normal\_run.sh:**

Automates the compilation and execution process.

#### **4. json.c:**

Handles JSON parsing using the cJSON library. Extract weather data (temperature, "feels like" temperature, humidity) and performs checks for specific conditions (e.g., high or low temperatures).

#### **5. json.h:**

Header file containing the function prototype for JSON.C

#### **6. learning.c:**

Main program logic. Retrieves data from the weather API, writes raw JSON data to file, and calls functions to process it.

#### **7. out.sh:**

Bash script designed to execute a program, save its output, and manage the number of output files generated. Simply used for Automated program execution, Output file management and Log archiving.

### **System Workflow:**

#### **1. Data Retrieval:**

A **Weather API Module** (implemented with libcurl) fetches weather data from an external API. The data includes parameters like temperature, humidity, and pressure. The data is parsed using cJSON for JSON handling.

## 2. Data Processing:

The raw data is processed through a **Data Processing Module** that structures and prepares it for analysis. This involves algorithms to identify trends and patterns.

## 3. Anomaly Detection:

An **Anomaly Checker Module** analyzes the processed data to detect outliers or abnormal trends, which could indicate environmental issues.

## 4. Report Generation:

A **Report Generator Module** creates detailed reports summarizing the findings from the data analysis. These reports highlight key weather metrics and detected anomalies.

## 5. Automation:

Shell scripts like `CEWWeatherAPI.sh` automate the execution of these modules. For example, this script runs the data retrieval, processing, and report generation in sequence.

## 6. Alerts and Notifications:

Real-time alerts are sent to users via email. Logs of anomalies are also maintained for reference.

## 7. File Management:

The system handles output file storage and retention. Older files are removed to maintain efficiency, as specified in configuration parameters.

## RESULTS:

### 1. Processed Data:

```
|||||Karachi|||||
Current temperature : 23.90°C
feels like : 24.25°C
Humidity : 73
```

### 2. Alerts:

 notify-send Just now

Low Temperature  
Temperature is lower than 5°C

 notify-send Just now

High Temperature  
Temperature is higher than 30°C