# COMPUTER ENGINEERING WORKSHOP

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

**Project Group ID:**

|  |  |
| --- | --- |
| FATIMA MUSHTAQ (CS-23011)  LAIBA KHAN (CS-23022)  MUHAMMAD USMAN (CS-23034) |  |

**BATCH:** 2023

**Department of Computer and Information Systems Engineering**

**NED University of Engg. & Tech.,**

**Karachi-75270**

**CONTENTS**

**S.No. Page No.**

1. **Problem Description** 1

1. **Methodology**

1. **Results**

# PROBLEM DESCRIPTION

The project involves designing and implementing an **Integrated Environmental Monitoring System** using C programming. This system interacts with a free API to retrieve real-time environmental data such as temperature and humidity, processes the data, and generates reports. Additionally, it aims to enhance programming skills by integrating advanced C programming concepts like pointers, dynamic memory allocation, and modular code structure.

**Key Objectives:**

* Interact with APIs for environmental data retrieval.
* Process, store, and automate data handling using files and shell scripts.
* Optimize the system using pointers and dynamic memory allocation.
* Trigger real-time alerts for critical environmental conditions via Linux system calls.
* Modularize code using custom header files.

# METHODOLOGY

The methodology followed for the development of the Integrated Environmental Monitoring System is summarized below:

1. **API Interaction:**
   * The curl library is used to send HTTP requests and fetch environmental data from a free API.
2. **Data Parsing:**
   * The fetched JSON data is parsed using the cJSON library to extract key environmental parameters such as temperature and humidity.
3. **Data Handling:**
   * Raw data is stored in files using C file-handling techniques.
   * Processed data is calculated and saved separately for efficient reporting.
4. **Automation:**
   * Shell scripts automate the tasks of fetching, processing, and saving data, minimizing manual intervention.
5. **Optimization:**
   * Pointers and dynamic memory allocation enhance the system’s performance when processing large datasets.
6. **Real-Time Alerts:**
   * Linux system calls are employed to trigger alerts for critical environmental readings.
7. **Modular Design:**
   * The code is modularized using header files to ensure readability and maintainability.

# RESULTS

The project outcomes are as follows:

1. Successfully retrieved real-time environmental data using the curl library.
2. Parsed JSON data efficiently with the cJSON library.
3. Stored raw and processed data in files for analysis and reporting.
4. Automated data retrieval and alerts with shell scripting.
5. Real-time alerts accurately notified users of critical environmental conditions.
6. Optimized and modularized code enhanced performance and maintainability.





