CS-219
COMPUTER ENGINEERING WORKSHOP

OPEN
ENDED LAB

Group Members:
SUMAIKA IMRAN CS-22009
SHEIKH MALAIKA MUSTAFA CS-22006
QURATULAIN CS-22018

Submitted To:
Miss MAHNOOR MALIK

DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING
BACHELORS IN COMPUTER SYSTEMS ENGINEERING
Course Code: CS-219
Course Title: Computer Engineering Workshop
Open Ended Lab
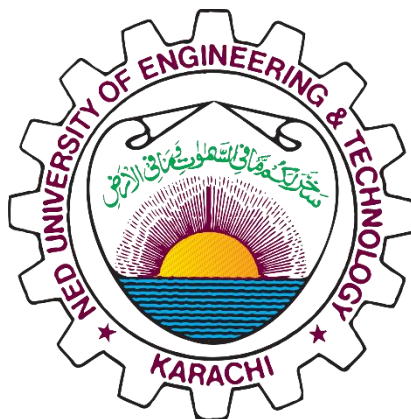**SE Batch 2022, Fall Semester 2023**
Grading Rubric
TERM PROJECT

Group Members:

Student No.	Name	Roll No.
S1		
S2		
S3		

CRITERIA AND SCALES				Marks Obtained		
				S1	S2	S3
Criterion1: Has the student implemented an efficient and scalable solution for data retrieval, processing, and reporting?						
0	1	2	3			
The student has not even implemented a basic solution that meets the project's requirements.	The student has implemented a basic solution that meets the project's requirements but may lack optimization in certain aspects.	The student has implemented a proficient and well-optimized solution.	The student has implemented an exceptionally efficient and scalable solution.			
Criterion 2: Has student demonstrated a strong understanding of C programming fundamentals?						
0	1	2	3			
The student doesn't have basic understanding of C programming fundamentals.	The student exhibits a basic understanding of C programming fundamentals.	The student demonstrates a strong understanding of C programming fundamentals.	The student demonstrates an exceptional understanding of C programming fundamentals.			
Criterion 3: How well written is the report?						
0	1	2	3			
The submitted report is unfit to be graded.	The report is partially acceptable.	The report is complete and concise.	The report is exceptionally written.			
			Total Marks:			

TABLE OF CONTENTS:

PROBLEM DEFINITION:

"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, analysis, and reporting."

1. INTRODUCTION:

The purpose of this report is to provide an overview of the Integrated Environmental Monitoring System project. This project aims to develop an automated system that retrieves real-time environmental data from a free API, processes the data, generates reports, and sends alerts for anomalies. The system utilizes various technologies and techniques to ensure efficient data handling and analysis.

2. SYSTEM ARCHITECTURE AND IMPLEMENTATION DETAILS:

The Integrated Environmental Monitoring System is comprised of the following key components:

a. Data Retrieval:

To access real-time environmental data, the system generates an API key from a website that provides free access to environmental data. This key serves as an authentication mechanism for retrieving data through HTTP requests. The system uses the curl library to fetch the data from the API endpoint. The retrieved data is in the JSON format and contains information such as temperature, humidity, and air quality.

b. Data Storage:

The system maintains separate files for storing different types of data. The raw environmental data retrieved from the API is saved in a JSON file, this file serves as a backup and historical record of the retrieved data. while the processed values are stored in a text file. This approach ensures data integrity and allows for easy retrieval and analysis.

c. Data Processing and Analysis:

After retrieving the environmental data, the system focuses on processing and analyzing the temperature values. It parses the JSON data to extract the temperature values and saves them in a separate text file for further processing. The system calculates the average, minimum, and maximum temperature values for a specific time period using the data stored in the text file. These processed temperature values are saved in three separate text files: one for average temperatures, one for minimum temperatures, and one for maximum temperatures. This approach allows for easy retrieval and analysis during report generation.

d. Anomaly Detection and Alerting:

To ensure effective monitoring, the system continuously analyzes the processed temperature values for anomalies. Anomalies can include a significant increase in average temperature or extreme values for the minimum or maximum temperature. If an anomaly is detected, the system triggers an email alert to notify the designated user. The email alert provides details about the anomaly, enabling timely decision-making and intervention.,

e. Reporting System:

The system generates daily reports summarizing the processing and analysis performed on the temperature values. The report includes the average, minimum, and maximum temperature for the day, as well as any detected anomalies. The report is saved in a text file which serves as a historical record of the system's operations. A job is scheduled using a crone tab to automatically call the program once a day and generate the report.

f. Automation and Integration:

To automate the system, a job scheduler such as the Crone tab is utilized. A scheduled job triggers the execution of the system once a day. This ensures that data retrieval, processing, and report generation tasks are performed autonomously.

g. Code Organization and Readability:

The system's code is organized into logical modules, enhancing code maintainability and readability. Header files are utilized to modularize the code, allowing for efficient collaboration among team members. The use of scripting enables the automation and scheduling of the system's execution.

3. CHALLENGES FACED:

During the implementation of the Integrated Environmental Monitoring System, several challenges were encountered. One notable challenge was faced when integrating a library to send emails in C for the first time. The details of this challenge and its resolution are outlined below:

- Implementing email functionality within the system using a C library posed a challenge, as it was the first time working with email integration in the C programming language. The lack of familiarity with the library's syntax, functions, and configuration options required additional research and learning.
- Another challenge was parsing data from the retrieved JSON file. The system needed to extract specific information, such as temperature values, from the JSON data structure. we faced difficulties in understanding the JSON parsing techniques and implementing them correctly in their code.
- Job scheduling was also a significant challenge faced during the implementation of the system. We needed to schedule the program execution to generate reports and send email alerts automatically once a day. We used a crone tab to schedule the execution, but

configuring the crone job correctly and ensuring its proper functioning required experimentation and troubleshooting.

4. CONCLUSION:

The Integrated Environmental Monitoring System project aims to provide an automated solution for retrieving, processing, and analyzing real-time environmental data. By utilizing a free API, the system fetches the data, calculates temperature statistics, detects anomalies, and generates reports. The system's automation and integration enable autonomous execution, ensuring continuous monitoring and timely reporting of environmental conditions.