A
Project Report
On
"AQLive: Comprehensive Air Quality Visualization Platform"

(CE451 - Software Project Major)

Prepared by
Rishi Patel (20CE100)
Darshil Shukla (20CE137)

Under the Supervision of
Prof. Ronak N. Patel

Submitted to

Charotar University of Science & Technology (CHARUSAT)
for the Partial Fulfillment of the Requirements for the
Degree of Bachelor of Technology (B.Tech.)
in U & P U. Patel Department of Computer Engineering (CE)
for B.Tech Semester 8

Submitted at

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

Accredited with Grade A+ by NAAC

cspit
Chandubhai S Patel Institute of Technology

U & P U. PATEL DEPARTMENT OF COMPUTER ENGINEERING
Chandubhai S. Patel Institute of Technology (CSPIT)
Faculty of Technology & Engineering (FTE), CHARUSAT
At: Changa, Dist.: Anand, Pin: 388421.

May, 2024

A
Project Report
On
"AQLive: Comprehensive Air Quality Visualization Platform"

(CE451 - Software Project Major)

Prepared by

Rishi Patel(20CE100)

Darshil Shukla(20CE137)

Under the Supervision of

Prof. Ronak N. Patel

Submitted to

Charotar University of Science & Technology (CHARUSAT)
for the Partial Fulfillment of the Requirements for the
Degree of Bachelor of Technology (B.Tech.)
in U & P U. Patel Department of Computer Engineering (CE)
for B. Tech Semester 8

Submitted at

Accredited with Grade A+ by NAAC

U & P U. PATEL DEPARTMENT OF COMPUTER ENGINEERING

Chandubhai S. Patel Institute of Technology (CSPIT)

Faculty of Technology & Engineering (FTE), CHARUSAT

At: Changa, Dist: Anand, Pin: 388421.

May, 2024

DECLARATION BY THE CANDIDATES

We hereby declare that the project report entitled "AQLive: Comprehensive Air Quality Visualization Platform" submitted by us to Chandubhai S. Patel Institute of Technology, Changa in partial fulfilment of the requirements for the award of the degree of B. Tech Computer Engineering, from U & P U. Patel Department of Computer Engineering, CSPIT, FTE, is a record of bonafide CE451 Software Project Major (project work) carried out by us under the guidance of Mr. Ronak N. Patel. We further declare that the work carried out and documented in this project report has not been submitted anywhere else either in part or in full and it is the original work, for the award of any other degree or diploma in this institute or any other institute or university.

(Rishi Patel-20CE100)

(Darshil Shukla-20CE137)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Mr. Ronak N. Patel,
Assistant Professor,
U & P U. Patel Department of Computer Engineering,
Chandubhai S Patel Institute of Technology (CSPIT)
Faculty of Technology (FTE)
Charotar University of Science and Technology (CHARUSAT) - Changa.

Ahmedabad University

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that Darshil Shukla, B. Tech (CE) student of Chandubhai S. Patel Institute of Technology, CHARUSAT, Changa has successfully completed a project on "AQLive: Comprehensive Air Quality Visualization Platform" at Ahmedabad University from 18th December, 2023 to 26th April, 2024. Throughout the internship period, he consistently displayed a high level of initiative, adaptability, and enthusiasm. His contributions to our team, particularly in Air quality visualization and database management reflect his potential for future success in his chosen field.

With warm regards,

Date:26/04/2024

Aditya Vaishya

--
Aditya Vaishya, PhD
Assistant Professor, School of Arts and Sciences,
Faculty Associate, Global Centre for Environment and Energy,
Ahmedabad University, Navrangpura, Ahmedabad 380 009
India. https://ahduni.edu.in/aditya-vaishya
Ph: (O) +91 79 61911520

Ahmedabad University

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that Rishi Patel, B. Tech (CE) student of Chandubhai S. Patel Institute of Technology, CHARUSAT, Changa has successfully completed a project on "AQLive: Comprehensive Air Quality Visualization Platform" at Ahmedabad University from 18th December, 2023 to 26th April, 2024. Throughout the internship period, he consistently displayed a high level of initiative, adaptability, and enthusiasm. His contributions to our team, particularly in Air quality visualization and database management reflect his potential for future success in his chosen field.

With warm regards, Date:26/04/2024

Aditya vaishya

--
Aditya Vaishya, PhD
Assistant Professor, School of Arts and Sciences,
Faculty Associate, Global Centre for Environment and Energy,
Ahmedabad University, Navrangpura, Ahmedabad 380 009
India. https://ahduni.edu.in/aditya-vaishya
Ph: (O) +91 79 61911520

CHARUSAT

CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

अमृतं तु विद्या

Accredited with Grade A+ by NAAC

CERTIFICATE

This is to certify that the report entitled "AQLive: Comprehensive Air Quality Visualization Platform" is a bonafied work carried out by **Rishi Patel(20CE100) & Darshil Shukla(20CE137)** under the guidance and supervision of **Prof. Ronak N. Patel** & **Dr. Aditya Vaishya** for the subject **Software Project Major (CE451)** of 8th Semester of Bachelor of Technology in **Computer Engineering** at Chandubhai S. Patel Institute of Technology (CSPIT), Faculty of Technology & Engineering (FTE) – CHARUSAT, Gujarat. To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred by the examiner(s).

Under the supervision of,

Prof. Ronak N. Patel
Assistant Professor
U & P U. Patel Dept. of Computer Engineering
CSPIT, FTE, CHARUSAT, Changa, Gujarat

Dr. Aditya Vaishya Ph.D.
Assistant Professor
Mathematical and Physical Sciences division
School of Arts and Sciences
Ahmedabad University

Dr. Nikita Bhatt
Head - U & P U. Patel Department of Computer Engineering,
CSPIT, FTE, CHARUSAT, Changa, Gujarat.

Chandubhai S. Patel Institute of Technology (CSPIT)

Faculty of Technology & Engineering (FTE), CHARUSAT

At: Changa, Ta. Petlad, Dist. Anand, Pin: 388421. Gujarat.

ABSTRACT

This project showcases the development of a sophisticated environmental monitoring and analytics system designed to provide real-time insights into various environmental parameters. By utilizing Grafana as the primary visualization tool, InfluxDB for efficient data storage, and an ESP32 microcontroller interfacing with a range of sensors including the NEO-6M GPS Sensor, BME688 Sensor, and PMS Sensor, the system offers comprehensive monitoring capabilities. The aim of this system is to enable continuous monitoring and analysis of critical environmental metrics such as GPS location, air quality, and particulate matter levels. Key features encompass data acquisition, processing, storage, visualization, and alerting modules, ensuring a holistic approach to environmental data management. Security measures including robust authentication, encryption protocols, access control mechanisms, and data integrity checks are integrated to safeguard sensitive data and ensure system integrity. Regarding implementation, the project outlines the configuration of both hardware and software components, emphasizing compatibility and interoperability between different system elements. Adherence to coding standards is emphasized to ensure code quality, readability, and functionality throughout the development process. Ultimately, this project aims to provide a flexible, reliable, and user-friendly solution for environmental monitoring, facilitating informed decision-making and proactive response strategies in various areas including environmental conservation, urban planning, and public health management.

ACKNOWLEDGEMENT

We extend our heartfelt gratitude to Dr. Aditya Vaishya, Ph.D., from Ahmedabad University, for his invaluable guidance and mentorship as our external guide throughout the duration of this project. Dr. Vaishya's expertise and insightful feedback have been instrumental in shaping the direction and success of our work.

We also express our sincere appreciation to Mr. Ronak N. Patel for his unwavering support and guidance as our internal guide. His expertise, encouragement, and constructive criticism have greatly contributed to the development and refinement of our project.

Special thanks are due to Mr. Yash Dahima, a Ph.D. student, and Mr. Yogesh Patel, a Ph.D. student whose continuous guidance and support have been invaluable throughout the project duration. Their expertise and willingness to share knowledge have been instrumental in overcoming challenges and achieving milestones.

Lastly, we would like to thank Charotar University of Science and Technology for providing us with the opportunity to undertake this internship experience. The support provided by the university has been instrumental in our professional development and academic pursuits.

Sincerely,
Darshil Shukla and Rishi Patel.

Table of Content

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

In response to the growing demand for real-time environmental data monitoring and analysis, our project endeavors to develop a sophisticated monitoring and analytics system using state-of-the-art technologies. By harnessing the power of Grafana as a robust big data analytics tool, we aim to construct a solution capable of seamlessly integrating with various sensors, facilitating the collection and visualization of critical data points.

1.1 PROJECT SUMMARY

This project revolves around the creation of a dynamic dashboard using Grafana, customized to visualize data sourced from multiple sensors. The system architecture incorporates Influx DB as the underlying database, ensuring efficient storage and retrieval of sensor data. Central to our design are advanced hardware components including the ESP32 DEVKIT-V1 microcontroller, NEO-M8N GPS Sensor, Adafruit BME688 Sensor, and PMS7003 Sensor.

1.2 PURPOSE

The primary objective of this project is to develop a robust monitoring and analytics platform capable of processing real-time sensor data for diverse applications. By leveraging the capabilities of Grafana alongside advanced sensors and microcontroller technology, our objective is to empower users with actionable insights into environmental conditions, facilitating informed decision-making and proactive response strategies.

1.3 OBJECTIVE

The principal aim of the project is to:

Develop an intuitive and user-friendly dashboard interface using Grafana for seamless data visualization.

Establish an efficient data flow from sensors to the InfluxDB database via the ESP32 microcontroller.

Ensure precise and timely collection of environmental parameters, encompassing GPS location, atmospheric conditions, and particulate matter levels.

Enable flexibility and scalability to accommodate varied monitoring requirements across different domains.

1.4 SCOPE

The project scope encompasses:

1. Designing and configuring customized Grafana dashboards tailored to specific monitoring needs.
2. Integrating sensor data acquisition and transmission functionalities within the ESP32 microcontroller environment.
3. Implementing data storage and retrieval mechanisms using InfluxDB to ensure optimal data management.
4. Supporting real-time visualization and analysis of environmental data to facilitate informed decision-making processes.
5. Providing a framework for potential expansion and integration with additional sensors or data sources in the future.

1.5 TECHNOLOGY AND LITERATURE REVIEW

Technology review:

1. Grafana for Visualization: Grafana Cloud is a highly available, fast, fully-managed OpenSaaS logging, metrics, traces, and profiling platform that also provides incident management and our application monitoring service. It is everything you love about Grafana, hosted by Grafana Labs [1].

   Utilizing Grafana as the primary analytics platform offers several advantages, including its versatility and extensive visualization capabilities. Grafana's user-friendly interface allows for the creation of customizable dashboards and visualizations, enabling researchers to interpret sensor data effectively.
2. InfluxDB for Data Storage: InfluxDB serves as the backend database, providing high-performance storage and retrieval optimized for time-series data. Time series data is a sequence of data points indexed in time order. Data points typically consist

of successive measurements made from the same source and are used to track changes over time [2].

3. ESP32 DEVKIT-V1 Microcontroller: The ESP32 DEVKIT-V1 microcontroller is chosen as the interface for sensor data acquisition and transmission [3].
4. Sensor Selection for Comprehensive Monitoring: The design incorporates multiple sensors including the NEO-6M GPS Sensor [4], BME688 Sensor [5], and PMS Sensor [6] to ensure comprehensive monitoring of air quality parameters. Each sensor provides unique data points such as location, temperature, humidity, particulate matter, and gas concentrations, enabling a holistic understanding of air quality conditions.

Literature review:

1. Choosing database was the main concern for our development, so, Naqvi, S.N.Z., Yfantidou, S. and Zimányi, E., 2017. [7] explained about time series database that, A Time Series Database (TSDB) is a database type which is optimized for time series or time-stamped data. It is built specifically for handling metrics, events or measurements that are time-stamped. A TSDB is optimized for measuring change over time. A TSDB allows its users to create, enumerate, update, destroy and organize various time series in a more efficient manner. The key difference with time series data from regular data is that mostly you ask questions about it over time. Nowadays, the majority of the companies are generating a insanely large stream of metrics and events (time series data) and hence the need of a TSDBs is unavoidable. According to this, InfluxDB was the first choice for people who wanted to work with real-time data. Hence, making it a perfect choice for our development.

2. Hossain, M.A., Noor, N.U., Yeasmin, A., Ritu, M.J.C. and Sarker, S., 2023[8] worked on design and implementation of GPS controlled environment monitoring system based on internet of things. Reference was taken from this looking at the similarity of factors like use of ESP32, Neo M8M. Data was stored from the microcontroller to Firebase (database).

3. Elias Biondo, Thadeu Brito, Alberto Nakano & José Lima, 2022[9] worked on Real-Time Monitoring System Approach for Measuring Indoor Air Quality Using the Internet of Things, where their work consists of a real-time IAQ system to monitor thermal comfort and gas concentration. The system has a data acquisition stage, captured by the WSN with a set of sensors that measures the data and send it to be stored on the InfluxDB database and displayed on Grafana.

CHAPTER 2: PROJECT PLANNING

2.1: PROJECT PLANNING

2.1.1 Project Development Approach and Justification

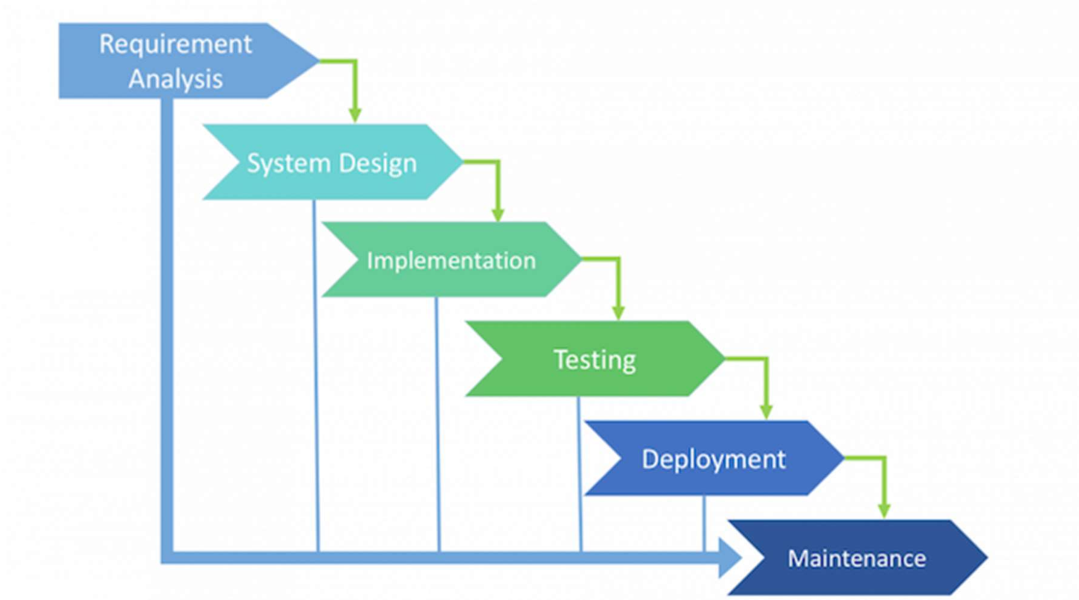


Fig 2.1. Iterative Waterfall Model
https://biplus.com.vn/iterative-model/

Iterative Waterfall methodology, though traditionally linear, can be adapted effectively for the development of a real-time air quality data visualization project on a Grafana dashboard. By breaking the project into distinct phases, each dedicated to specific aspects such as backend infrastructure and frontend development, the team can efficiently manage complex requirements while maintaining a structured approach. This method enables early risk identification and mitigation, allowing for prompt adjustments and improvements in subsequent iterations. While providing predictability and clarity in project planning and execution, Iterative Waterfall may lack the flexibility of Agile, necessitating careful management of evolving requirements and stakeholder feedback to ensure project success.

2.1.2 Project Effort and Time, Cost Estimation

Project Effort	2 people working on weekdays
Project Time	90-100 working days
Project Cost Estimation	Sensors+ Microcontroller cost = 8,000 INR(approx.). Development cost varies.

Table 2.1. Project Effort, time and cost estimation

2.1.3 Roles and Responsibilities

Both Rishi Patel(20CE100) and Darshil Shukla(20CE137) were tasked with the same responsibilities.

Here is the list of all the tasks assigned to the team:

- Design a front-end which delineates the variation of the provided air-quality data parameters (CSV Format).
- Write code for ESP-32 microcontroller which is equipped with 3 sensors namely, PMS, BME688 and NEO M8N.
- Optimize the code such that SD card and Screen can also function with the device.
- Select and set-up a database which can store the values from the device via code.
- Visualize the data stored onto the database in the big-data analytics tool named Grafana.
- Add multiple visualizations for all the parameters, that is, Bar chart, Line chart, Gauge, table, etc.
- Add a map module to Grafana which shows the real-time location of the device by rendering Latitude and Longitude values from the database.
- Test the system by replicating the device and visualizing all its data accurately.

2.1.4 Group Dependencies

Darshil Shukla (20CE137) and Rishi Patel (20CE100) are key team members in this project development. Their collaboration is crucial for successful execution of tasks and activities. They share responsibilities across phases, from requirement gathering to implementation and testing. Their combined skills, expertise, and dedication ensure smooth progress and effective problem-solving throughout the project lifecycle. Their collaboration strengthens outcomes and enhances quality.

2.2. PROJECT SCHEDULING

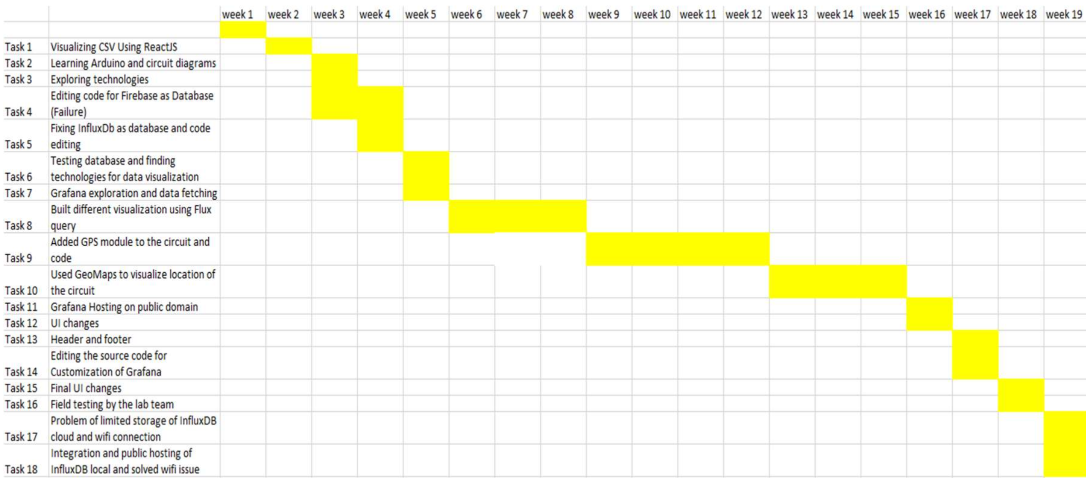


Fig 2.2 Gantt Chart

The project schedule, meticulously planned and documented using MS Excel, is allocated weekly from December 18, 2023, to April 26, 2024, with milestones and dependencies tracked.

CHAPTER 3: SYSTEM REQUIREMENTS STUDY

3.1 User Characteristics

Comprehending the attributes of the users is essential to customizing the system in a way that best suits their requirements. The following user traits are taken into account in our project:

1. Technical Proficiency: Users should be acquainted with the fundamentals of sensor data interpretation and data visualization tools such as Grafana.
2. Environmental Domain Knowledge: To accurately interpret the data, users are expected to have domain knowledge pertaining to environmental monitoring.
3. Use Cases: Users will work with the system to keep an eye on current environmental conditions and to make decisions based on the information that is presented. Requirements for Accessibility: The system interface should be user-friendly and adaptable to accommodate users with different levels of technical expertise.

3.2 Hardware and Software Requirements

The system's ability to function depends on certain hardware and software elements. Important prerequisites consist of:

1. Hardware Specifications: Data acquisition is accomplished by means of an ESP32 DEVKIT-V1 microcontroller, a NEO-6M GPS sensor, a BME688 sensor, and a PMS sensor.
2. Software Requirements: InfluxDB as the backend database, Arduino IDE for microcontroller programming, and Grafana for data visualization.
3. Compatibility Requirements: Guaranteeing that software tools and hardware components are compatible to enable smooth data flow and visualization.
4. Performance Benchmarks: Determining performance metrics to assess how well the system processes, stores, and visualizes data.

3.3 Assumptions and Dependencies

Certain assumptions influence the system's design and operation. These include:

1. **Stable Environmental Conditions:** Assuming consistent environmental conditions to ensure reliable sensor data collection.
2. **Data Availability:** Dependency on continuous and accurate data streams from sensors for real-time monitoring and analysis.
3. **External Dependencies:** Integration with third-party services or APIs for supplementary data sources or functionalities.
4. **Regulatory Compliance:** Adherence to regulatory standards governing data privacy and environmental monitoring practices.
5. **Project Timeline:** Assumption of adherence to the project schedule, considering potential delays or unforeseen challenges during development and deployment phases.

Dependencies include:

- *Plantower_PMS7003: Custom library, version not applicable.*
- *HardwareSerial: Part of the Arduino Core library, version depends on the Arduino IDE version.*
- *TinyGPS++: Version 1.0.2.*
- *Wire: Part of the Arduino Core library, version depends on the Arduino IDE version.*
- *WiFi: Part of the Arduino Core library, version depends on the Arduino IDE version.*
- *InfluxDbClient: Version 0.0.10.*
- *InfluxDbCloud: Version 0.0.3.*
- *WiFiManager: Version 0.16.0.*
- *Adafruit_BME680: Version 2.0.3.*
- *Adafruit_GFX: Version 1.10.7.*
- *Adafruit_SSD1306: Version 2.5.0.*
- *FS: Part of the Arduino Core library, version depends on the Arduino IDE version.*
- *SD: Part of the Arduino Core library, version depends on the Arduino IDE version.*
- *SPI: Part of the Arduino Core library, version depends on the Arduino IDE version.*

CHAPTER 4: SYSTEM ANALYSIS

4.1 Study of Current System

Currently, prominent platforms like OpenAQ.com, iqair.com, accuweather.com, and airnow.gov offer robust systems for monitoring air quality parameters, drawing data from diverse sources including government monitoring stations and citizen science initiatives. These platforms provide real-time and historical data on various pollutants, ensuring accuracy and reliability while offering user-friendly interfaces for easy interpretation. They stand out for their wide coverage, real-time updates, and data download options, making them invaluable resources for informed decision-making and public health awareness.

4.2 Problem and Weaknesses of Current System

Operating costs for these platforms are high, hindering coverage, especially in under-resourced areas. Moreover, these platforms struggle to incorporate data from affordable and portable sensors like PMS, BME688, and NEO M8N, limiting their ability to provide a comprehensive air quality overview.

4.3 Requirements of New System

4.3.1 Functional Requirements

The new system must address the following functional requirements:

1. **Real-Time Data Monitoring**: Implement mechanisms to enable the real-time monitoring of environmental parameters, ensuring prompt detection of anomalies or deviations.
2. **Automated Data Collection**: Introduce automated data collection processes to streamline data acquisition and minimize errors associated with manual methods.
3. **Centralized Data Storage**: Establish a centralized database solution, such as InfluxDB, to efficiently store and manage sensor data, facilitating easy access and retrieval.

4. **Interactive Data Visualization**: Develop interactive and customizable dashboards using Grafana, allowing users to visualize and analyze environmental data effectively.

4.3.2 Non-Functional Requirements

The new system should also meet the following non-functional requirements:

1. **Scalability**: Ensure that the system architecture is scalable to accommodate future growth and integration with additional sensors or data sources.
2. **Reliability**: Maintain high system reliability and availability to support continuous monitoring operations without downtime.
3. **Performance**: Optimize system performance to handle large volumes of data and provide responsive user interactions, even during peak usage periods.
4. **Security**: Implement robust security measures to protect sensitive environmental data from unauthorized access, ensuring data integrity and confidentiality.
5. **User Experience**: Design an intuitive and user-friendly interface to enhance user experience and facilitate efficient data interpretation, catering to users with varying technical backgrounds.

4.4 Feasibility Study

4.4.1 Does the System Contribute to the Overall Objectives of the Organization?

Yes, the proposed framework aligns closely with the organization's overarching goals, especially in supporting environmental monitoring capabilities and facilitating data-informed decision-making. Through highlights like real-time checking, mechanized information collection, and interactive visualization of natural information, the framework enables the organization to form well-informed choices with respect to asset allotment, chance administration, and compliance with controls. Furthermore, the system's versatility and flexibility guarantee its capacity to bolster the organization's advancing needs and vital activities in natural maintainability and administration.

4.4.2 Can the System Be Implemented Using the Current Technology and within the Given Cost and Schedule Constraints?

Yes, it is possible to implement the system because it makes use of well-established technologies that are widely available and supported in the industry, such as ESP32 microcontrollers, Influx DB local, and Grafana. System component development is made easier with the use of open-source tools and libraries. The system's deployment costs are reasonable all around, with the lowest initial costs coming from software licensing, hardware purchases, and development work. The investment is justified by the long-term advantages of improved data analysis and decision-making. The project is designed to support different stages of system development, testing, and deployment. It runs from December 18, 2023, to April 26, 2024. In order to minimize schedule risks and ensure steady progress, daily iterations and milestone assessments are implemented.

4.4.3 Can the System Be Integrated with Other Systems Already in Place?

Yes, the system's design allows seamless integration with other existing systems within the organization. Through standardized APIs and data exchange protocols, interoperability with complementary systems can be achieved. This integration enhances data interoperability and facilitates holistic decision-making processes across various organizational functions.

4.5 Activity/Process in New System

The new system introduces streamlined activities and processes to enhance environmental monitoring and data analysis capabilities. These include real-time data acquisition from sensors, automated data processing and storage in a centralized database, interactive visualization of environmental parameters using customizable dashboards, and timely alerting mechanisms for critical events or threshold breaches.

4.6 Features of New System

The new system offers a range of features designed to meet the diverse needs of users and stakeholders. Key features include:

1. Real-time Data Monitoring: Enables continuous monitoring of environmental parameters, providing instant insights into changing conditions.
2. Automated Data Collection: Automates the process of data acquisition from sensors, reducing manual intervention and minimizing errors.
3. Centralized Data Storage: Utilizes a centralized database solution, facilitating efficient storage and retrieval of sensor data for analysis.

4. Interactive Data Visualization: Provides customizable dashboards with interactive visualizations, allowing users to explore and analyze data intuitively.
5. Alerting Mechanism: Notifies users of critical events or threshold breaches through timely alerts, enabling proactive responses.
6. Integration Capability: Supports seamless integration with existing systems through standardized APIs, enhancing data interoperability and decision-making processes.

4.7 Use Case Diagram

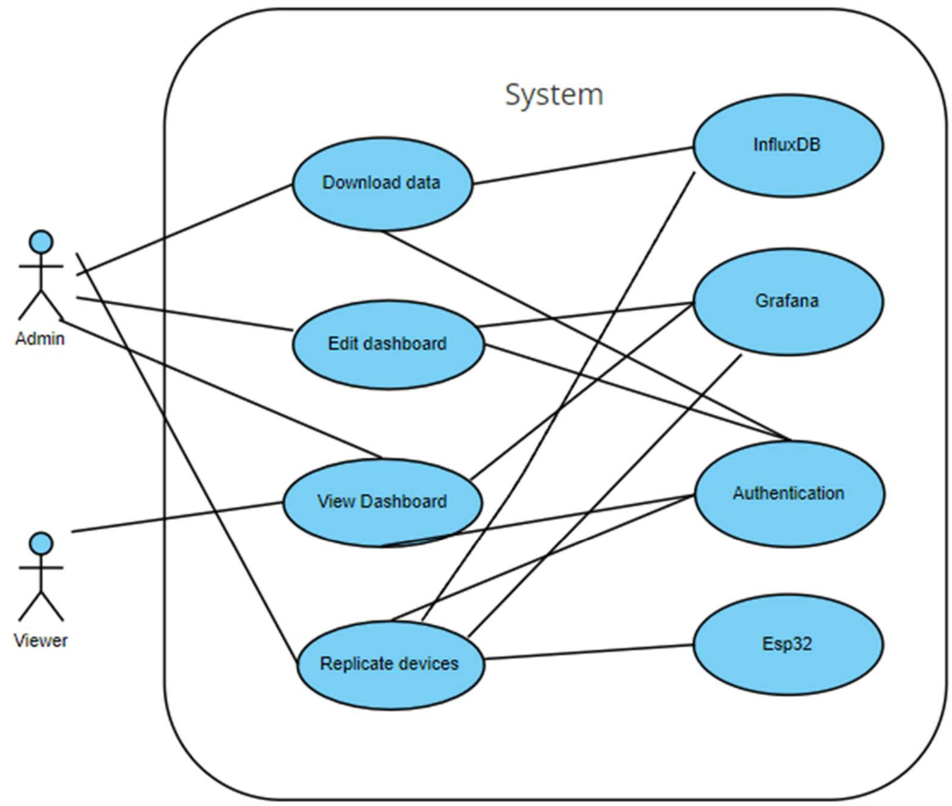


Fig 4.1. Use Case Diagram

4.9 Sequence Diagram

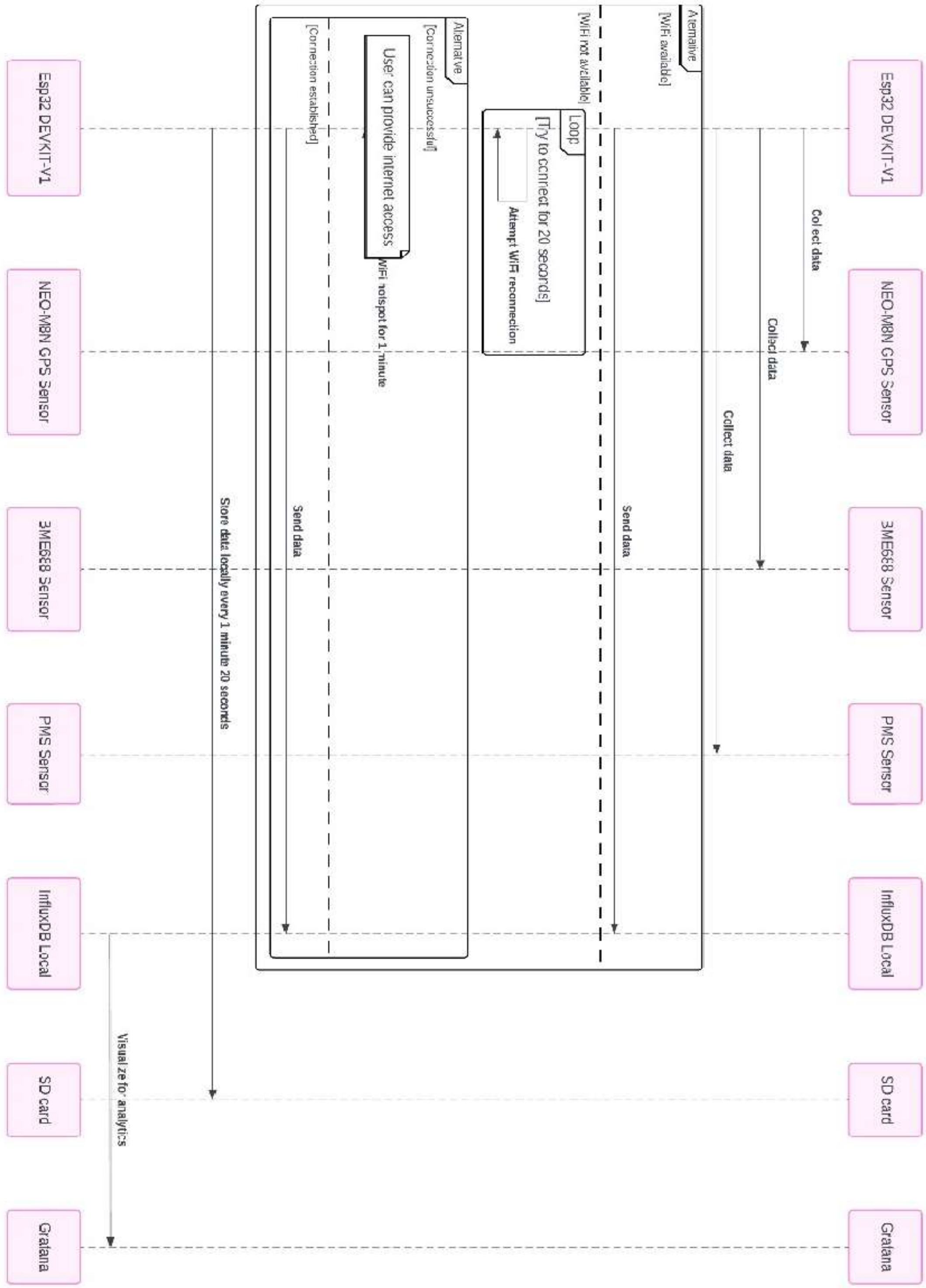


Fig 4.3. Sequence Diagram

Chapter 5: System Design

5.1 System Application Design

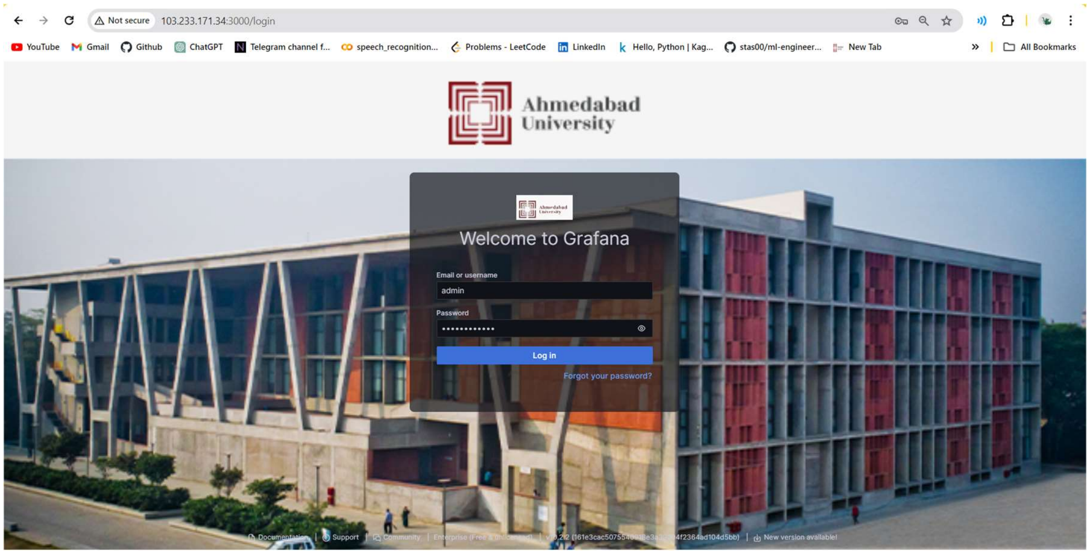


Fig 5.1 Grafana Login Page

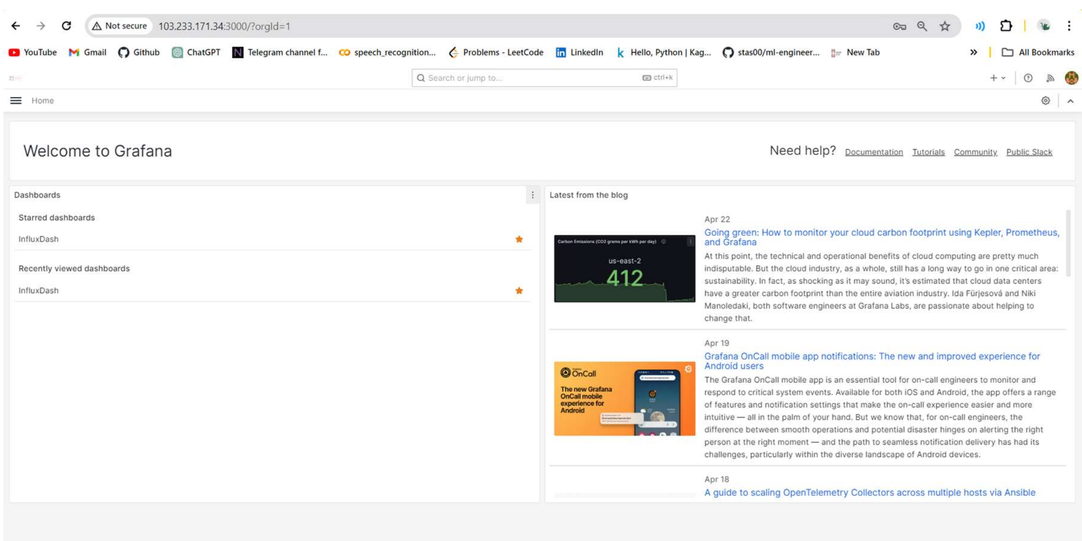


Fig 5.2 Grafana Homepage

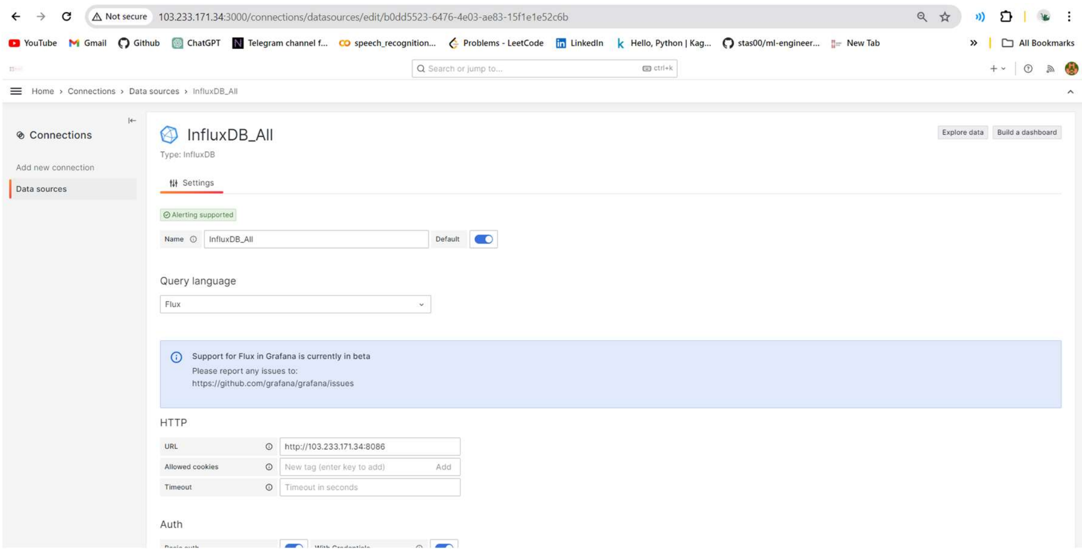


Fig 5.3 Grafana Data Source

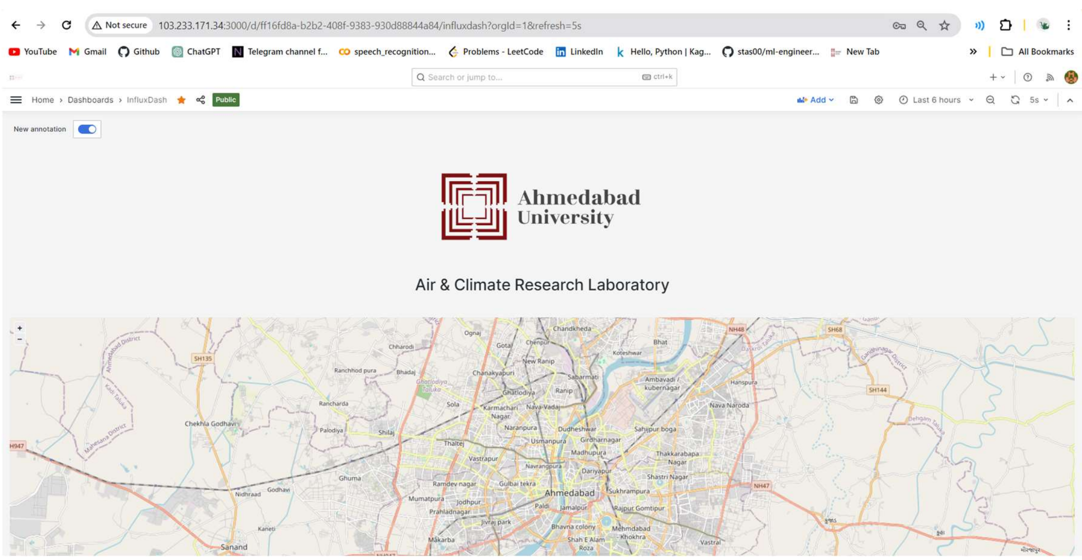


Fig 5.4 Dashboard Homepage

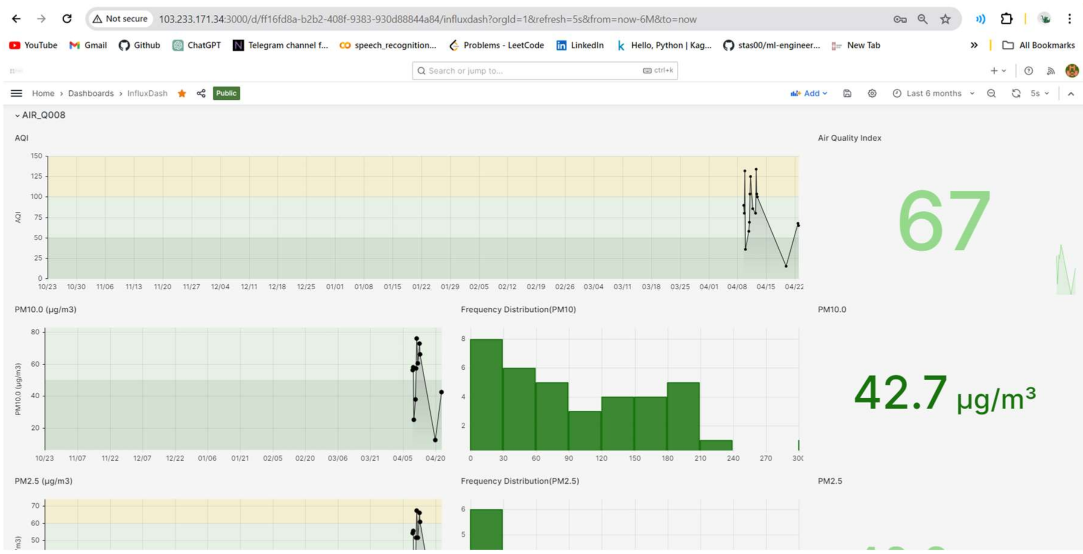


Fig 5.5 Sensor data visualization

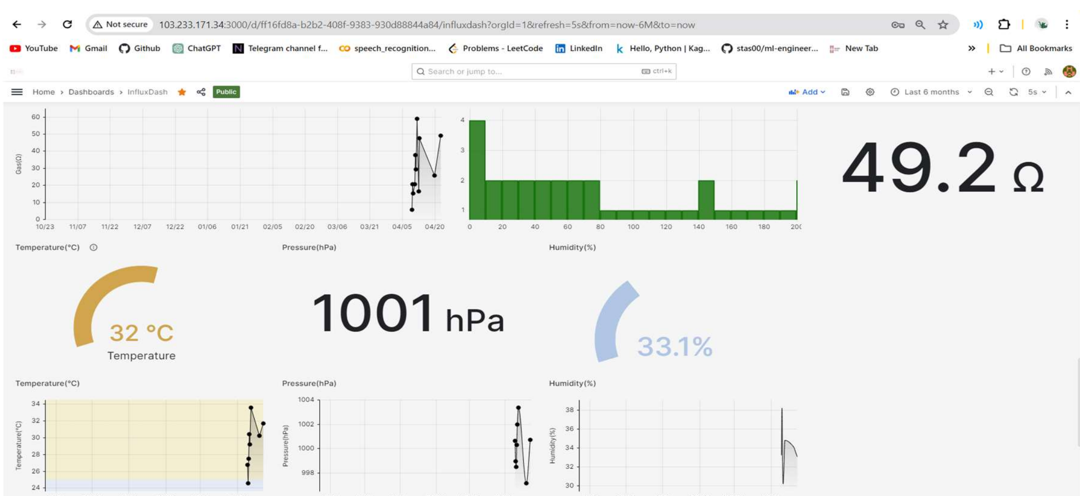
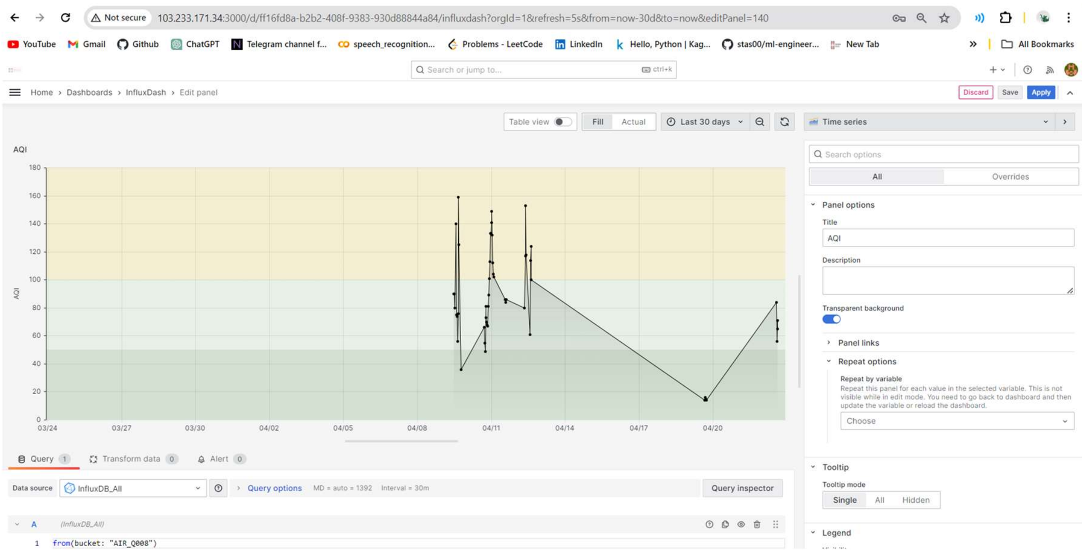


Fig 5.6 Sensor Data Visualization 2



\

Fig 5.7 Edit graph section

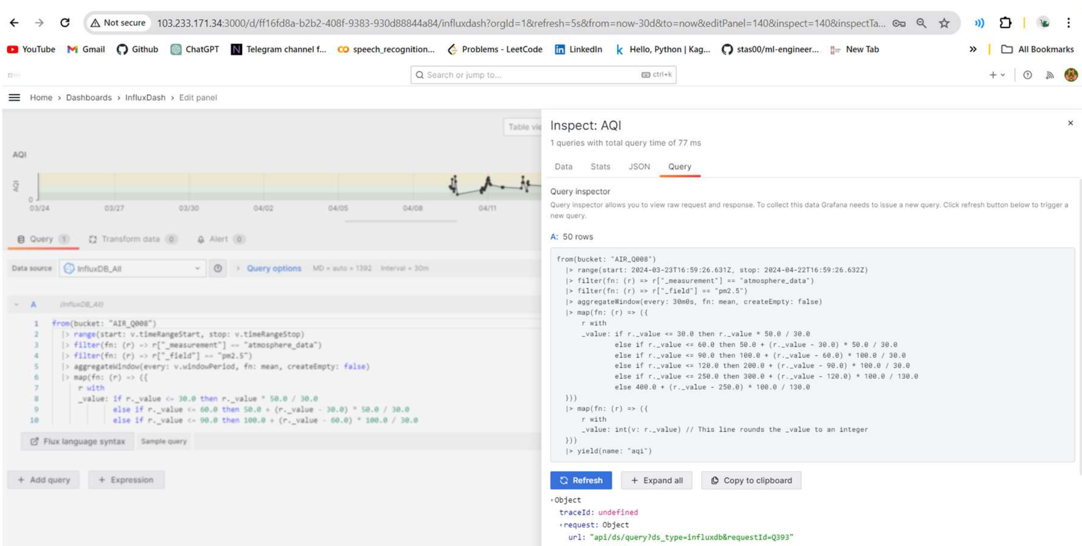


Fig 5.8 Inspect query section

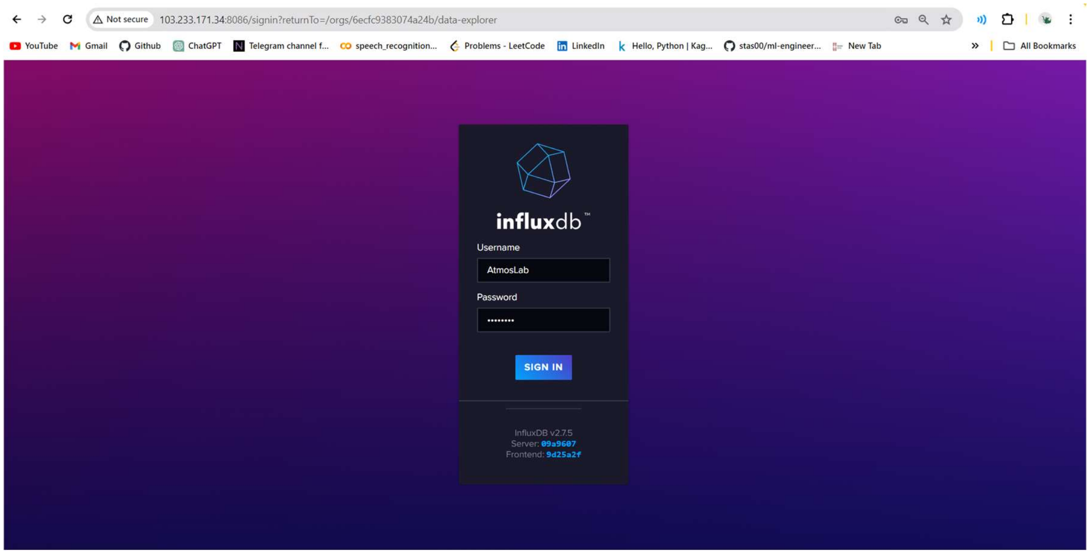


Fig 5.9 InfluxDB login page

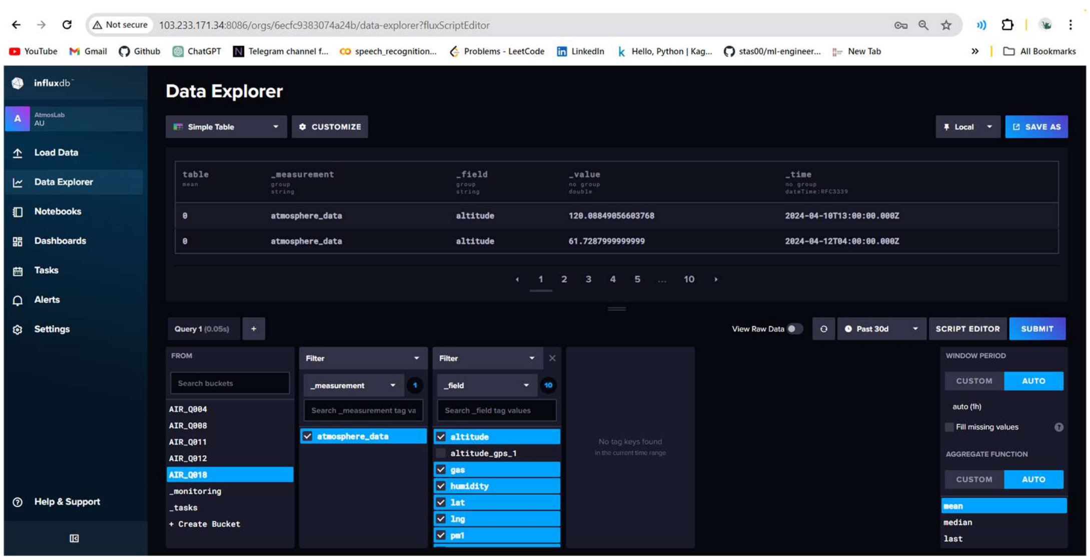


Fig 5.10 InfluxDB table view

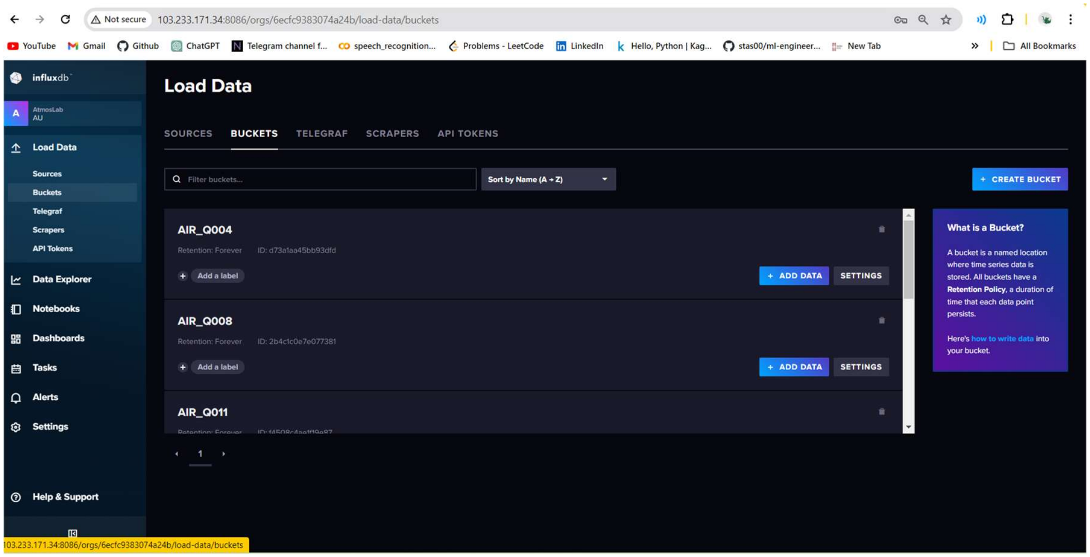


Fig 5.11 InfluxDB bucket creation/view

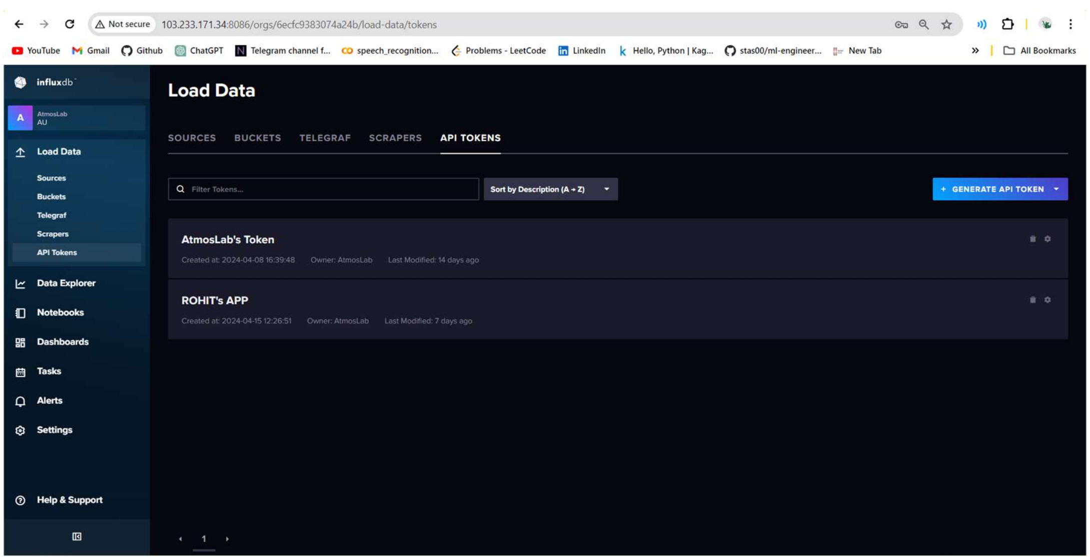


Fig 5.12 InfluxDB API token section

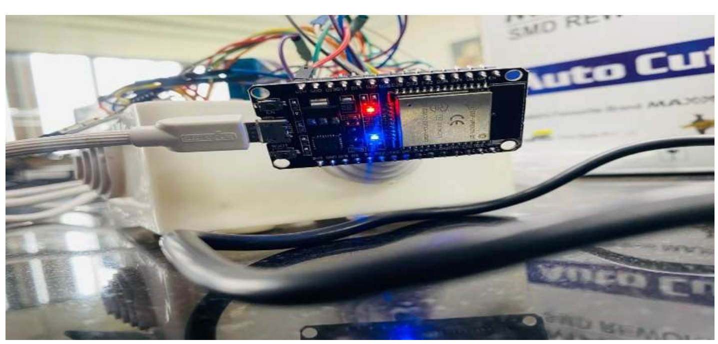


Fig 5.13 ESP32- DEV Kit V1

Fig 5.14 ESP32 DevKit V1 Pin diagram

https://www.upesy.com/blogs/tutorials/esp32-pinout-reference-gpio-pins-ultimate-guide

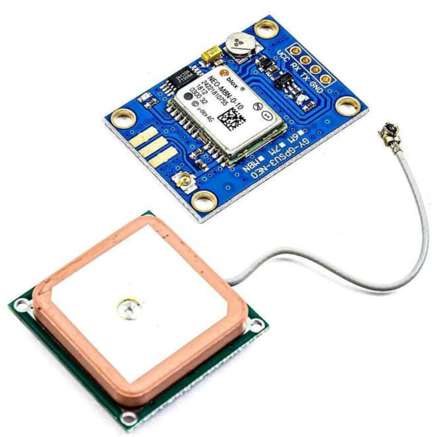


Fig 5.15 NEO M8N GPS

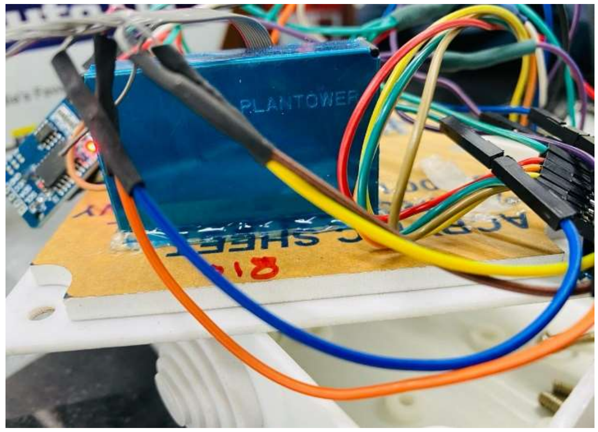


Fig 5.16 Particulate Matter Sensor(PMS)

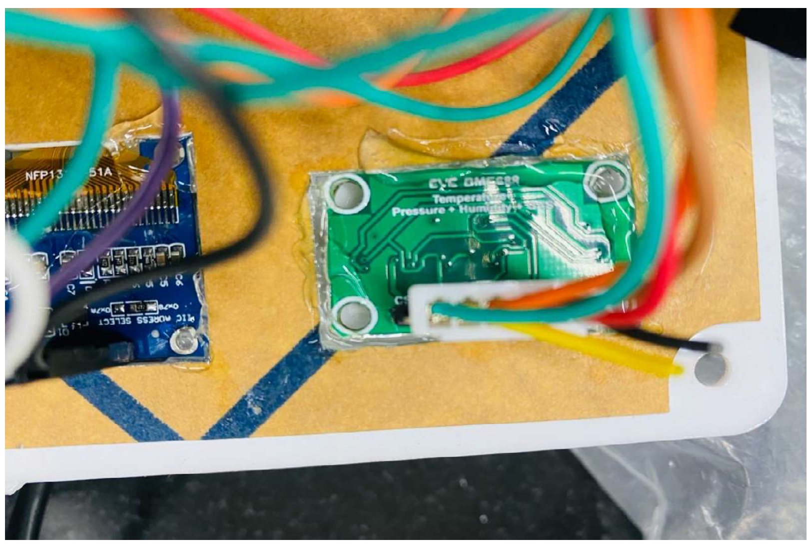


Fig 5.17. BME688

5.1.1 Method Pseudo Code

```
Setup:
  InitializeSerialCommunication() {
    Initialize serial communication for debugging and sensors.
  }

  ConnectToWiFi() {
    // Check if already connected to WiFi
    if (WiFi.status() == WL_CONNECTED) {
      return;
    }

    // Attempt to connect using saved credentials for 20 seconds
    unsigned long wifiStartTime = millis();
```

```
        while (WiFi.status() != WL_CONNECTED && millis() - wifiStartTime < 20000) {
            WiFi.begin();
            delay(500);
        }

        // If WiFi connection is established, return
        if (WiFi.status() == WL_CONNECTED) {
            return;
        }

        // Start WiFi configuration portal and run for 1 minute
        WiFiManager;
        wifiManager.setTimeout(60);
        if (!wifiManager.startConfigPortal("AIR_Q004", "AIR_Q004")) {
            // Handle connection failure or timeout
            return;
        }
    }

    InitializeSensors() {
        Initialize sensors (BME680, PMS7003, GPS).
    }

    InitializeInfluxDBClient() {
        Initialize InfluxDB client and conFig connection parameters.
    }

    InitializeSDCard() {
        Initialize SD card for data logging.
        Check if data file exists on SD card, create it if not and add headers.
    }

Loop:
    ReadSensorData() {
        Read GPS data if available and extract latitude, longitude, altitude, and number of satellites.
        Read air quality data from PMS7003 sensor.
        Read environmental data from BME680 sensor.
    }

    FormatData() {
        Format data into InfluxDB line protocol.
    }

    LogDataToSDCard() {
        Log data to SD card.
    }

    SendDataToInfluxDB() {
        If WiFi is connected:
```

```
        Send data to InfluxDB Cloud.
    }

    DisplayDataOnScreen() {
        Display formatted data on OLED screen.
    }

    ReconnectToWiFi() {
        If not already connected to WiFi:
            Connect to WiFi.
    }

Helper Functions:
    GetDateTimeFromGPS() {
        Function to get date and time from GPS data.
    }

    SaveWiFiConfiguration() {
        Function to handle saving WiFi configuration.
    }

    CheckFileExistsOnSDCard() {
        Function to check if data file exists on SD card and create it if not.
    }

    AppendDataToSDCard() {
        Function to append data to SD card.
    }
```

5.2 Input/ Output and Interface Design

5.2.1 Samples of Forms, Reports and Interface

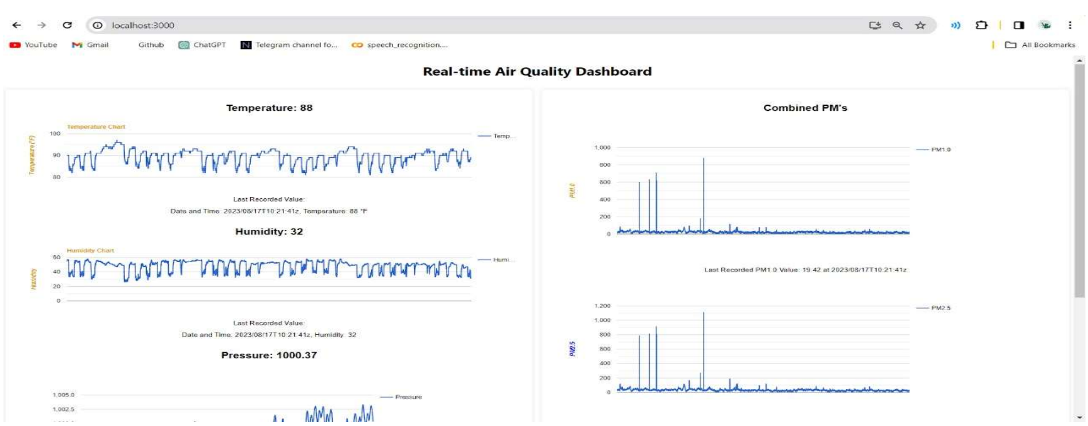


Fig 5.18. Sample Interface 1

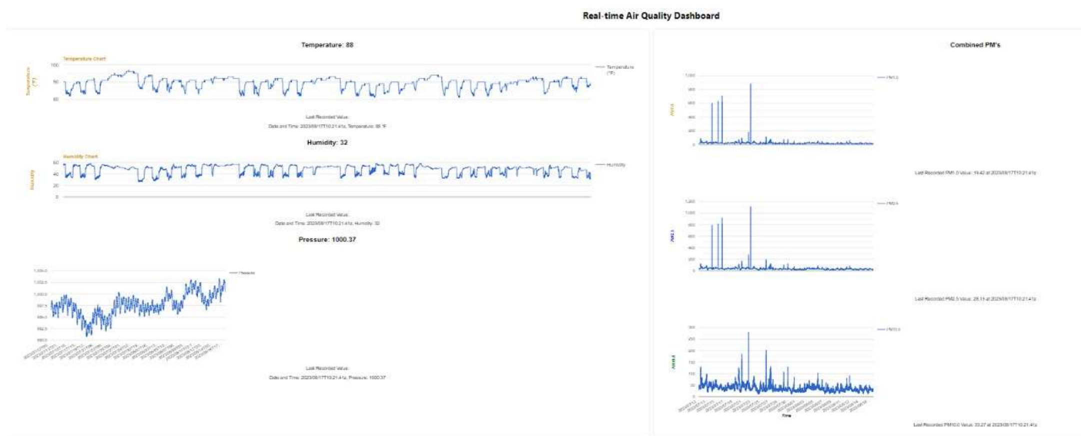


Fig 5.19. Sample Interface 2

5.2.2 State Diagram

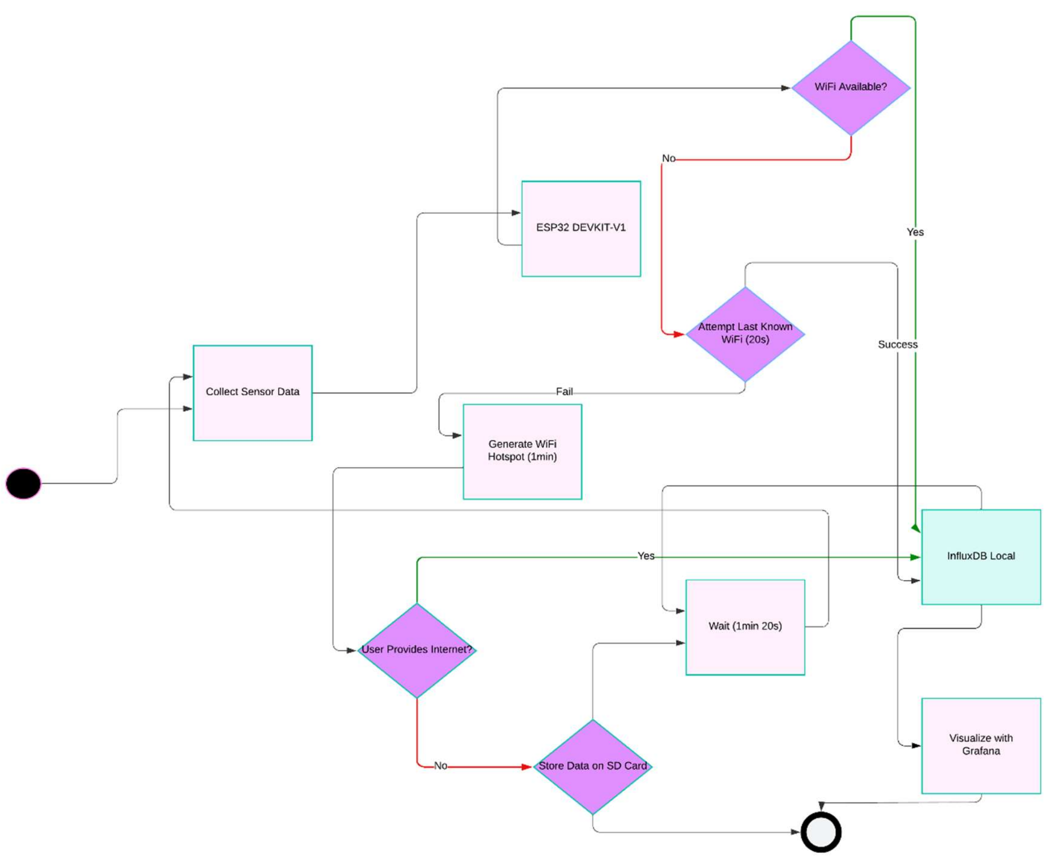


Fig 5.20. State Diagram

Chapter 6: Implementation Planning

6.1 Implementation Environment

Essential facets of the Implementation Environment phase encompass:

- Hardware configuration involves selecting and adjusting essential components like microcontrollers and sensors.
- Software installation is crucial for setting up IDEs, libraries, and dependencies.
- Network settings are needed for smooth communication between system parts, requiring configuration of Wi-Fi, Ethernet, etc.
- Database configuration involves setting up DBMS like InfluxDB and designing schemas for data integrity.
- Cloud integration allows for storage, processing, and analysis, requiring setup of platforms and APIs.
- Development tools, like Git, simplify coding and collaboration.
- A test environment is essential for verifying system operation and performance through rigorous testing methodologies.

6.2 Modules Specification

Data Acquisition Module: This module is responsible for gathering real-time environmental data from various sensors deployed in the field. It interacts directly with the sensors, retrieves sensor readings, and prepares the data for further processing.

Data Storage Module: The data storage module manages the storage of acquired sensor data in a structured format. It interacts with the database system, InfluxDB, to efficiently store the data while ensuring data integrity and reliability.

Data Visualization Module: This module is tasked with transforming raw sensor data into visually informative representations. It utilizes Grafana, a powerful analytics and visualization platform, to create interactive dashboards and charts for users to monitor environmental parameters.

User Interaction Module: The user interaction module facilitates communication between users and the system. It handles user authentication, authorization, and session management, allowing users to log in securely, view sensor data, and customize dashboard settings based on their permissions.

InfluxDB Interface Module: This module serves as an intermediary between the system and the InfluxDB database. It encapsulates functionalities for writing sensor data to the database and retrieving stored data as needed. It ensures seamless integration with the database system for efficient data management.

Grafana Interface Module: The Grafana interface module acts as a bridge between the system and the Grafana platform. It enables the creation and customization of Grafana dashboards, configuring data sources, defining visualization layouts, and managing dashboard settings.

Sensor Interface Modules: These modules are responsible for interfacing with individual sensor types used in the system, such as the PMS7003 particulate matter sensor, BME680 environmental sensor, and NEO-6M GPS sensor. Each module handles data acquisition, sensor calibration, and data processing specific to its sensor type.

WiFi Connectivity Module: The WiFi connectivity module manages the system's wireless communication capabilities. It handles WiFi network configuration, connection establishment, and data transmission over WiFi networks, ensuring seamless connectivity with external systems.

SD Card Logging Module: This module provides functionality for logging sensor data to an SD card. It ensures redundancy and data backup by storing sensor readings locally on the SD card, thereby enhancing data reliability and resilience against network failures or interruptions.

Error Handling Module: The error handling module detects and manages error conditions that may occur during system operation. It includes mechanisms for logging errors, notifying system administrators, and implementing fault tolerance strategies to maintain system reliability and robustness.

6.3 Security Features

Several security measures are implemented:

1. Restricted Database Access: Only authorized administrators are granted access to query and modify the contents of the database. This access control mechanism

prevents unauthorized users from tampering with the air quality data stored in the database, maintaining its integrity.

2. User Authentication: Prior to accessing the dashboard, website users are required to undergo authentication procedures. This ensures that only authenticated users with valid credentials can access the features and functionalities of the dashboard, enhancing overall system security.

6.4 Coding Standard

While developing our project, it's very important to manage the coding standards and we have tried to follow as much its possible. By following the coding standards, we can ensure that the code will be well organized and easy to understand. Coding standards followed in our project are mentioned below.

We used comments to describe the purpose and behaviour of the code. This makes it easier for the developers to understand how the code works and how it should be used.

We have followed Single Responsibility Principle so that each class or function have single responsibility.

Chapter 7: Testing

7.1 Testing Plan

The testing plan covers the following aspects of the system:

1. Sensor data acquisition and processing.
2. Data storage and retrieval from the Influx DB database.
3. Visualization and dashboard creation using Grafana.
4. User interaction functionalities.
5. Integration with external systems and services.
6. Error handling and fault tolerance mechanisms.

7.2 Testing Strategy

For the real-time air quality data visualization project, the testing strategy will be tailored to ensure the reliability, functionality, and performance of the system.

1. Unit testing will focus on verifying the functionality of essential components such as sensor data collection, database storage, visualization, user authentication, and interaction features.

2. Integration testing will validate the seamless interaction and data flow between these components, ensuring accurate and efficient transmission of data from sensors to the database and then to the visualization dashboard.

3. System testing will involve comprehensive testing of the entire system to evaluate its behaviour and performance in a real-world environment. This includes testing end-to-end functionalities such as sensor data acquisition, storage, visualization, user interaction, and error handling.

4. Regression testing will be conducted to detect and prevent any regressions or unintended side effects introduced by code changes, maintaining the stability and reliability of existing functionalities across successive iterations and updates.

5. Performance testing will assess the system's responsiveness, scalability, and resource utilization under various load conditions. This includes measuring response times for data acquisition, storage, retrieval, and visualization tasks to ensure optimal performance.

6. User acceptance testing (UAT) will involve end-users to validate the system's compliance with requirements and gather feedback on usability and satisfaction. UAT sessions will provide valuable insights into user expectations and areas for improvement, ensuring that the system meets user needs effectively.

7.3 Test Suites Design

Test Case Description	Expected Result	Status
Data Acquisition		
1. Verify sensor data collection from designated sensors	Sensor data is collected accurately	Pass
2. Test accuracy of sensor readings	Sensor readings match known standards	Pass
3. Check consistency and reliability of sensor data	Sensor data remains consistent and reliable over time	Pass
Data Storage		
4. Ensure accurate storage of sensor data in Influx DB	Sensor data is stored accurately in the database	Pass
5. Test implementation of data retention policies	Historical data is retained as per specified policies	Pass
6. Validate database scalability for large data volumes	Database can handle large volumes of incoming data	Pass
Data Visualization		

7. Confirm accurate display of sensor data on Grafana	Sensor data is displayed accurately on the dashboard	Pass
8. Test responsiveness and update frequency of dashboard	Dashboard updates promptly with new incoming data	Pass
9. Validate visual layout and formatting of the dashboard	Dashboard components are clear and readable	Pass
User Interaction		
10. Verify secure user login to access the dashboard	Users can log in securely to access the dashboard	Pass
11. Test user permissions for editing and viewing rights	Administrators have editing rights, viewers have viewing rights	Pass
12. Validate user interactions on the dashboard	Zooming, panning, and filtering data functions correctly	Pass
Integration		
13. Test integration between sensor data acquisition and InfluxDB	Data flows seamlessly between modules	Pass
14. Validate integration between InfluxDB and Grafana	Data is visualized accurately on the Grafana dashboard	Pass
15. Check compatibility and interoperability of software components	Components work together without issues	Pass
Error Handling		
16. Test system response to sensor data transmission errors	System handles errors gracefully and continues operation	Pass
17. Verify appropriate display of error messages on the dashboard	Users are notified of any data retrieval or visualization errors	Pass
18. Check system's ability to recover from database or dashboard service failures	System recovers and resumes operation without data loss	Pass
Performance		
19. Measure response time for data acquisition, storage, and visualization tasks	Tasks are completed within specified timeframes	Pass

20. Test dashboard performance under different load conditions	Dashboard remains responsive with large datasets and multiple users	Pass
21. Validate system meets specified performance requirements	System performance meets design specifications	Pass

Table 7.1: Test suits design

Chapter 8 : Conclusion and Discussion

8.1 Self-Analysis of Project Viabilities

We carried out a comprehensive feasibility analysis of the project, taking into account resource-related, technological, and financial aspects. Conclusions are listed below:

Technical viability: Initially, the project had difficulties configuring the ESP32 microcontroller to interface with the NEO-M8N GPS sensor and troubleshooting signal problems. Nevertheless, we were able to effectively incorporate the sensor into the system through testing and debugging.

Financial viability: The project ran into issues with the amount of free buckets and the length of the data retention period when using InfluxDB 2.0 Cloud. So, in order to better meet project needs, we switched to InfluxDB local, which provided infinite storage and retention.

Resource Viability: It became clear that code needed to be optimized to make the most of the ESP32's limited storage space.

8.2 Problems Encountered and Possible Solutions

Challenges:

1. The integration of the NEO-M8N GPS sensor with the ESP32 microcontroller initially posed challenges, with signal reception being irregular and unreliable.
2. Cloud-based InfluxDB 2.0 had limitations on accessible buckets and data storage time.
3. Code optimization was crucial due to storage constraints, and adding an SD card module and display screen required further coding adjustments.

Solutions:

1. Extensive research, documentation review, and testing were conducted for effective integration.
2. Antenna modifications and signal strength monitoring systems were implemented to increase stability.
3. InfluxDB 2.0 Cloud was switched to InfluxDB local for limitless retention and storage.
4. Optimization strategies like efficient algorithms and variable compression were used to reduce code size.
5. SD card modules and display panels were added to the system architecture for new feature development.

8.3 Summary of Project Work

Accomplishments:

1. Effectively created an environmental monitoring system with the use of data analytics tools, microcontroller technology, and sophisticated sensors.

2. Overcame technological obstacles in the areas of database selection, signal stability, and sensor integration by using smart problem-solving and adaptability techniques.

3. Put code optimization techniques into practice to guarantee effective resource use and easily adapt to new needs.

4. Developed a dependable and expandable system that can gather, store, visualize, and communicate with users in real-time to facilitate informed decision-making in environmental monitoring applications.

Lessons Learned:

1. Importance of thorough research and experimentation in resolving technical challenges.

2. Flexibility and adaptability in adjusting project plans and strategies to overcome constraints.

3. Significance of code optimization in maximizing resource utilization and system performance.

4. Value of iterative development and continuous refinement to meet evolving project requirements effectively.

Future Directions:

1. Explore opportunities for further enhancing system functionalities, such as implementing predictive analytics and automated decision-making algorithms.
2. Continuously monitor and optimize system performance to ensure scalability and reliability in diverse environmental conditions.
3. Foster collaboration with stakeholders and domain experts to identify new use cases and expand the system's applicability in environmental monitoring and management.

Chapter 9: Limitation and Future Enhancement

Limitations:

1. Hardware Constraints: The project may face limitations due to hardware constraints, such as the limited processing power and memory of the ESP32 microcontroller. This could restrict the complexity of algorithms and data processing capabilities.
2. Data Transmission Range: The system's reliance on WiFi connectivity may limit its effectiveness in remote or isolated locations with poor network coverage. Implementing alternative communication methods, such as cellular or satellite communication, could mitigate this limitation.
3. Sensor Accuracy and Reliability: The accuracy and reliability of sensor data may vary based on environmental factors and sensor quality. Calibrating sensors and implementing error detection mechanisms can help address this limitation to some extent.
4. Data Security: The system's reliance on WiFi and cloud-based storage solutions introduces potential security vulnerabilities. Implementing robust encryption protocols and access control mechanisms is essential to safeguard sensitive environmental data.
5. Scalability: As the system collects more data over time, scalability may become a concern, especially in terms of data storage and processing capabilities. Regular evaluation and optimization of system architecture and resources are necessary to ensure scalability.

Future Enhancements:

1. Predictive Analytics: Incorporating machine learning algorithms to analyse historical data and predict future environmental trends could enhance the system's decision-making capabilities.
2. Diverse Sensor Integration: Expanding the range of sensors to include additional parameters such as noise levels, air quality indices, and radiation levels would provide a more comprehensive understanding of environmental conditions.

3. Fault Tolerance and Redundancy: Implementing fault-tolerant mechanisms and redundant systems to ensure continuous operation and data integrity, even in the event of hardware failures or network disruptions.

References

[1] “Technical documentation | Grafana Labs,” *Grafana Labs*. [Online]. Available: https://grafana.com/docs/ [Accessed: Apr. 25, 2024]

[2] “Get started with InfluxDB | InfluxDB OSS v2 Documentation,” *InfluxData Inc.* [Online]. Available: https://docs.influxdata.com/influxdb/v2/get-started/ [Accessed: Apr. 25, 2024]

[3] “esp32 wroom 32,” *espressif*. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-da_datasheet_en.pdf. [Accessed: Apr. 25, 2024]

[4] “NEO-M8 series,” *u-blox*. [Online]. Available: https://content.u-blox.com/sites/default/files/products/documents/NEO-M8_ProductSummary_UBX-16000345.pdf. [Accessed: Apr. 25, 2024]

[5] “BME688.” [Online]. Available: https://www.hestore.hu/prod_getfile.php?id=15427. [Accessed: Apr. 25, 2024]

[6] “Digital universal particle concentration sensor.” [Online]. Available: https://download.kamami.pl/p564008-PMS7003%20series%20data%20manua_English_V2.5.pdf. [Accessed: Apr. 25, 2024]

[7] “Naqvi, S.N.Z., Yfantidou, S. and Zimányi, E., 2017. Time series databases and influxdb. *Studienarbeit, Université Libre de Bruxelles*, *12*, pp.1-44.”

[8] “Hossain, M.A., Noor, N.U., Yeasmin, A., Ritu, M.J.C. and Sarker, S., Design and implementation of GPS controlled environment monitoring system based on internet of things.”

[9] E. Biondo, T. Brito, A. Nakano, and J. Lima, “A WSN Real-Time Monitoring System Approach for Measuring Indoor Air Quality Using the Internet of Things,” *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 76–90, 2023, doi: 10.1007/978-3-031-25222-8_7. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-25222-8_7