GROUP ID: GE18

A PROJECT REPORT ON

IOT-BASED INDOOR AIR QUALITY MONITORING SYSTEM

SUBMITTED TO THE PIMPRI CHINCHWAD COLLEGE OF ENGINEERING
AN AUTONOMOUS INSTITUTE, PUNE
IN THE FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING
COMPUTER ENGINEERING (REGIONAL LANGUAGE)

SUBMITTED BY

SHRAVANI CHIDREWAR	**PRN: 121B1D035**
NEERA PATIL	**PRN: 121B1D045**

DEPARTMENT OF COMPUTER ENGINEERING (REGIONAL LANGUAGE)

PCET'S PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

Sector No. 26, Pradhikaran, Nigdi, Pimpri-Chinchwad, PUNE 411044

2024-2025

CERTIFICATE

This is to certify that the project report entitles

"IOT-BASED INDOOR AIR QUALITY MONITORING SYSTEM"

Submitted by

SHRAVANI CHIDREWAR PRN: 121B1D035
NEERA PATIL PRN: 121B1D045

are bonafide students of this institute and the work has been carried out by them under the supervision of **Dr. Sujata Kolhe** and it is approved for the partial fulfillment of the requirement of Pimpri Chinchwad College of Engineering an autonomous institute, for the award of the B. Tech. degree in Computer Engineering (Regional Language).

(Dr. Sujata Kolhe)
Guide,
Department of Computer Engineering
(Regional Language)

(Prof. Dr. Rachana Patil)
Head,
Department of Computer Engineering
(Regional Language)

(Mr. Mandar Sahasrabudhe)
External Guide,
Arklite Speciality lamps Pvt. Ltd

(Prof. Dr. G.N. Kulkarni)
Director,
Pimpri Chinchwad College of Engineering Pune – 44

Place: Pune
Date:

MEMORANDUM OF UNDERSTANDING

PIMPRI CHINCHWAD EDUCATION TRUST's
PIMPRI CHINCHWAD COLLEGE OF ENGINEERING
(An Autonomous Institute)

- N.B.A. Accredited Courses - Civil Engg, Comp. Engg., E&TC Engg., IT. Engg, Mech. Engg.
- NAAC Accredited with 'A' Grade
- An ISO 9001:2015 Certified Institute
- Approved by A.I.C.T.E., New Delhi, Govt. of Maharashtra, D.T.E., Mumbai
- Permanently Affiliated to Savitribai Phule Pune University, Pune

Ref. No.: PCCOE/AS&H/24-25/40 Date: 23/7/24

MEMORANDUM
OF
UNDERSTANDING

Between

PIMPRI CHINCHWAD EDUCATION TRUST'S
PIMPRI CHINCHWAD COLLEGE OF ENGINEERING
DEPARTMENT OF APPLIED SCIENCES AND HUMANITIES
Sector No.26, Pradhikaran, Nigdi,
Pune, Maharashtra 411 044

And

ARKLITE SPECIALITY LAMPS Pvt. Ltd
Gat No 2794, Kharabwadi Chakan Talegaon Road,
Tal Khed, Chakan, Pune, Maharashtra 410501.

ARKLITE® SPECIALITY LAMPS
AN ISO 9001:2015 CERTIFIED COMPANY

ARKLITE SPECIALITY LAMPS PVT. LTD. CHAKAN

PCET's Pimpri Chinchwad College of Engineering, An Autonomous Institute, Nigdi, Pune-44

ACKNOWLEDGEMENT

We express our sincere thanks to our **Guide Dr. Sujata Kolhe** for her constant encouragement and support throughout our project, especially for the useful suggestions given during the course of project and having laid down the foundation for the success of this work.

We would also like to thank our **Project Coordinator, Prof. Rohini Y. Sarode** for her assistance, genuine support and guidance from early stages of the project. We would like to thank **Prof. Dr. Rachana Y. Patil, Head of Computer Engineering (Regional Language) Department** for her unwavering support during the entire course of this project work. We are very grateful to our **Director, Prof. Dr. G.N. Kulkarni** for providing us with an environment to complete our project successfully. We also thank all the staff members of our college and technicians for their help in making this project a success.

We also thank all the web committees for enriching us with their immense knowledge. Finally, we take this opportunity to extend our deep appreciation to our family and friends, for all that they meant to us during the crucial times of the completion of our project.

SHRAVANI CHIDREWAR
NEERA PATIL

ABSTRACT

Indoor air quality (IAQ) is essential for ensuring the health, safety, and productivity of people in confined spaces like homes, offices, and industries. With increasing levels of air pollution and an ever-growing concern over its health effects, there is a need for sustained and affordable IAQ monitoring. Inadequate indoor air with contaminants such as VOCs (Volatile Organic Compounds), particulate matter (PM), and high temperatures or humidity levels may contribute to severe health issues such as respiratory diseases, allergies, and long-term sequelae.

This project suggests the development of a low-cost real-time IAQ monitoring system with sensors and a NodeMCU (ESP8266) microcontroller to measure temperature, humidity, VOCs, and PM levels. The sensor readings are sent wirelessly to the ThingSpeak cloud platform and accessed using API keys in a Flutter-based mobile app. The app displays the air quality data and computes an IAQ index that is simple to comprehend for users. Firebase provides secure authentication of users and storing historical data such that both real-time and trend analysis access is allowed.

With the integration of IoT, cloud solutions, and mobile technology, the system enables remote monitoring and alarms when pollutant concentrations cross healthy limits. With this solution, there is a cost-effective, scalable, and easy-to-use method of managing air quality. It helps build healthier indoor conditions, enabling the users to act in a timely manner and take informed decisions to ensure their own well-being.

KEYWORDS

Indoor Air Quality (IAQ), Air Pollution Monitoring, IoT-based Monitoring, Volatile Organic Compounds (VOCs), Particulate Matter (PM), Temperature and Humidity Sensing, Cloud-based Storage, ThingSpeak Channel, Arduino IDE, ESP8266 (NodeMCU), Sensor Calibration, Real-Time Data Visualization, Flutter App Development, Firebase Authentication, API Integration, Mobile-based Monitoring, Real-Time Alerts, Smart Environment, Air Quality Index (AQI)

TABLE OF CONTENTS

LIST OF ABBREVATIONS

ABBREVIATION	**ILLUSTRATION**
IAQ	INDOOR AIR QUALITY
IOT	INTERNET OF THINGS
VOC	VOLATILE ORGANIC COMPOUNDS
PM	PARTICULATE MATTER
AQI	AIR QUALITY INDEX
API	APPLICATION PROGRAMMING INTERFACE
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
MCU	MICROCONTROLLER UNIT
GUI	GRAPHICAL USER INTERFACE
HTTP	HYPERTEXT TRANSFER PROTOCOL
SDK	SOFTWARE DEVELOPMENT KIT
DB	DATABASE
JSON	JAVASCRIPT OBJECT NOTATION
LOC	LINES OF CODE
DHT	DIGITAL HUMIDITY AND TEMPERATURE SENSOR
PM2.5 / PM10	PARTICULATE MATTER 2.5 MICRONS / 10 MICRONS
RAM	RANDOM ACCESS MEMORY
SSD	SOLID STATE DRIVE
UI	USER INTERFACE

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 OVERVIEW

Air conditions keep worsening year by year with the expansion of civilization and mounting dirty emissions from industries and cars. While air is a necessary resource for living, most people do not care about the gravity of air pollution or only now realize the issue. Of the different kinds of polluting substances like water, land, thermal, and noise, air pollution is the worst and most perilous, leading to climate change and fatal diseases. Air pollution is an enormous issue and not limited to inhabitants of smog-filled cities, in such ways as global warming and destruction of the ozone layer, it can harm us all.

Based on the World Health Organization (WHO), 90 percent of the population currently inhales tainted air, and air pollution is the cause of death for 7 million people every year. Pollution's health impacts are extremely harsh that leads to stroke, lung cancer, and heart disease. Air quality monitoring and management are thus prime topics of concern. The impact of dirty air must be prevented and one of the options is by employing a system to control the indoor air quality.

To address this issue, our project proposes to create a real-time IAQ monitoring system based on temperature, humidity, VOC, and PM sensors. The suggested system can identify the air's quality level. The data will be uploaded to the ThingSpeak cloud platform, accessed through API keys, and presented in a Flutter-based Android app. Firebase will also store past IAQ data and handle user authentication. This system will help industries like Arklite monitor environmental conditions, ensuring a safer and more productive workspace.

1.2 MOTIVATION

Air pollution is now one of the greatest environmental issues, worsening every year with fast industrialization, city sprawl, and automobile emissions. Although air is a necessity for living, most people do not realize the long-term effect of bad air. Of all types of pollution—water, soil, thermal, and noise pollution—air pollution is the most lethal and can cause acute health risks, climate change, and ecological deterioration.

Health and Environmental Impact -

- Respiratory conditions like asthma, bronchitis, and lung infections
- Cardiovascular illnesses, including stroke and myocardial infarction
- Elevation of the risk of cancer, especially lung cancer
- Impacts on climate change through the production of greenhouse gases and destruction of the ozone layer

Indoor air pollution is just as dangerous as outdoor pollution, impacting workplace efficiency and general health. Poorly ventilated industries, offices, and factories can have high levels of volatile organic compounds (VOCs), particulate matter (PM), which can cause fatigue, dizziness, and long-term health problems. Conventional air quality monitoring used bulky, sophisticated analog equipment that needed a lot of manual labour to gather and process data. Digital sensors, over time, made the process more efficient but still did not have real-time data processing and access. The Internet of Things (IoT) transformed environmental monitoring by allowing automated, real-time data gathering and cloud-based processing. In response to the increasing issue of indoor air pollution, **o**ur project suggests a real-time IAQ monitoring system based on sensors for temperature, humidity, VOC, and PM concentrations.

The system will:

- Monitor IAQ parameters continuously and compute an IAQ index
- Upload sensor readings to the cloud (ThingSpeak) for instant access
- Fetch and display results in a simple Flutter-based Android application
- Save historical IAQ data with Firebase for trend analysis and decision-making

With the installation of this cost-effective, scalable, and real-time monitoring system, Arklite as well as other industries can be assured of optimum environmental conditions, enhanced worker health and safety, and regulatory compliance with air quality standards.

1.3 PROBLEM STATEMENT AND OBJECTIVES

1.3.1 Problem Statement

Ensuring good indoor air quality (IAQ) is crucial for maintaining a safe and healthy environment, particularly in industrial settings where air pollutants like VOCs, particulate matter, and humidity variations can pose serious health risks. Current monitoring technologies are typically antiquated, have no real-time analysis, and need manual intervention, so detecting and reacting to air quality changes in a timely manner is problematic. Without a smooth and automated system, industries struggle with regulatory compliance and protecting employee well-being.

To overcome these limitations, this project aims to develop a real-time IAQ monitoring system with continuous monitoring of air quality parameters utilizing temperature, humidity, VOC, and PM sensors. The sensor readings will be transferred to the ThingSpeak cloud, accessed through API keys, and represented using a Flutter-based Android app. In addition, Firebase will save past IAQ data and handle user authentication, allowing industries such as Arklite to track air quality trends and make better decisions for a healthier working environment.

1.3.2 Objectives

The primary objectives of this project are:

1. To design a real-time air quality monitoring system with temperature, humidity, VOC, and PM sensors for industrial and indoor applications.
2. To implement an IAQ index calculation approach that effectively translates sensor readings to determine indoor air pollution levels in real-time.
3. To integrate ThingSpeak for cloud-based data transmission and Firebase for securely storing historical IAQ data and user authentication information.
4. To develop an easy-to-use Flutter-based Android app that presents real-time and historical air quality information for simple interpretation by users.
5. To facilitate industries such as Arklite to track air quality, meet regulatory requirements, and have a healthier and safer working environment.

1.4 SCOPE OF THE WORK

The scope of the project consists of creating an indoor air quality (IAQ) real-time monitoring system utilizing IoT-based sensors and cloud. The system will gather environmental information, calculate the levels of IAQ, and display it using a user-friendly mobile application.

The scope of this project includes:

1. Real-time Air Quality Monitoring – Temperature, humidity, VOC, and PM levels will be constantly measured through proprietary sensors and real-time data updates will be provided.
2. Cloud-based Data Storage and Access – Sensor data will be sent to the ThingSpeak cloud platform, where data will be stored, processed, and accessed through API keys for advanced analysis.
3. Android Mobile App Development – An Android-based mobile app will be created to present real-time IAQ measurements, past data trends, and air quality warnings to users.
4. Data Storage and Login – Firebase integration will be used to retain previous IAQ data and secure user login for viewing customized air quality information.
5. Industrial Use – The system can be used in industries such as Arklite to monitor air conditions in the workplace in real time and maintain compliance with safety regulations as well as enhance the well-being of workers.
6. Scalability and Future Upgrades – The project is scalable to include more sensors, predictive analytics, and automation functionalities, including activation of air purifying systems according to IAQ levels.

2. LITERATURE SURVEY

2.1 REVIEW OF RECENT LITERATURE

Smart sensors and cloud computing are used by Internet of Things (IoT)-based indoor air quality (IAQ) monitoring systems to detect and analyze pollutants such as aerosols, volatile organic compounds (VOCs), CO, CO_2, temperature, and humidity in real time. According to research, these devices, which are installed in offices and buildings at universities, enhance air quality control by facilitating automated reactions, real-time alerts, and remote access. Although cloud-based solutions improve accessibility and scalability, issues like sensor calibration and internet dependence still exist. According to studies, combining edge computing and AI-driven calibration can improve dependability and lessen network reliance, increasing the sustainability and efficiency of IoT-based IAQ monitoring.[1]

Using an Arduino Uno and a MQ135 sensor, the AirQMon system monitors indoor air quality in real time. Data is transmitted to ThingSpeak via an Ethernet shield, and a graphical depiction is provided by an Android app. By providing real-time alerts through a buzzer and mobile messages, this inexpensive and simple-to-install technology raises awareness of environmental hazards. Its reliance on Ethernet rather than Wi-Fi limits connectivity, and the lack of additional sensors like PM2.5 and VOC limits its capacity to offer a thorough assessment of air quality, despite its benefits, which include price and simplicity. For improved usefulness, future developments might include wireless communication and additional sensors.[2]

A multi-sensor configuration is used in the Low-Cost IoT-Based Indoor Air Quality Monitoring system to measure CO_2, VOC, temperature, humidity, and PM levels. Data is transmitted over an IoT platform for real-time display and analysis. It guarantees affordability and scalability while preserving a balance between cost and accuracy by employing a stationary monitoring device. By facilitating improved decision-making and spatial coverage through several connected devices, this system enhances environmental monitoring. But in order to keep accuracy, the sensor needs to be calibrated on a regular basis, and because of its short battery life, it needs a constant power source or energy management that is suited for extended use.[3]

The IoT-based Indoor Air Quality Monitoring System is made for smart houses and uses fuzzy control logic to improve air quality control. It monitors CO_2, CO, and fine particulate matter by integrating an Arduino Uno, an ESP8266 Wi-Fi module, and several sensors. Validated using MATLAB-based simulations, the system automates ventilation and air purification decisions. It guarantees smooth real-time monitoring by utilizing wireless connection and AI-based decision-making. However, its use in industrial settings is limited because it necessitates constant internet connectivity and is best suited for smart homes.[4]

Key environmental parameters may be tracked in real time with the IoT-Based Handheld Environmental and Air Quality Monitoring Station. Temperature, humidity, PM2.5, PM10, VOC, and CO levels are measured using sensor-based devices, and the data is sent over the ThingSpeak cloud platform via Wi-Fi and GSM. Real-time readings and historical trend analysis are made possible using a mobile application. After calibration, the device produces high-accuracy data and is inexpensive and simple to deploy. Its drawbacks include a short battery life that necessitates external power sources and the possibility of sensor accuracy changes that call for routine calibration.[5]

2.2 GAP IDENTIFICATION / COMMON FINDINGS IN LITERATURE

Recent advancements in IoT-based environmental monitoring have greatly enhanced real-time data collection, accessibility, and system scalability. However, several gaps and challenges still exist, particularly in the integration of low-cost sensor networks with reliable data interpretation and mobile accessibility. The following common conclusions and research gaps are highlighted in the study:

Common Findings:

1. Real-Time Monitoring:
    - All solutions provide real-time IAQ (Indoor Air Quality) monitoring using IoT-based networks.
    - Data is collected and analysed remotely for better environmental control.

2. Use of IoT and Cloud technology:
    - All systems use IoT-enabled sensors to measure environmental parameters like CO_2, VOC, PM levels, humidity, and temperature.
    - Cloud-based platforms (such as ThingSpeak) are used for data visualization, analysis, and storage.
3. Cost-Effectiveness and Scalability:
    - Compared to commercial solutions, these systems offer low-cost and scalable options for IAQ monitoring.
4. Automation and Decision-Making:
    - Fuzzy logic control (in Reference 2) and real-time AI-based decision-making improve air quality control and automation.
    - Data-driven insights enable proactive environmental monitoring.

Identified Research Gaps:

1. Limited battery life:
    - Most systems require continuous power supply or frequent battery recharging, limiting portability and long-term use.
2. Sensor accuracy and Calibration Issues:
    - Sensors require periodic calibration for accuracy, which adds maintenance costs and efforts.
    - Sensor readings may vary due to environmental factors, requiring advanced error correction techniques.
3. Limited scope of application:
    - Some solutions are designed only for smart homes (Reference 2) and need modifications for industrial or large-scale use.
    - Portable monitoring stations (Reference 1) lack automation in air quality control.
4. Need for Internet Connectivity:
    - Cloud-based data storage and analysis rely on continuous internet access, which may not always be available.
5. Lack of Integration with Advanced AI/ML Models:
    - Although IoT is used, the application of machine learning for predictive analysis is minimal.

- Future work could focus on integrating AI for advanced air quality forecasting and anomaly detection.

While existing research offers effective approaches for indoor air quality monitoring using IoT devices and cloud integration, there are still challenges in sensor calibration accuracy, real-time mobile visualization, scalability across locations, and long-term data reliability. This project aims to address these gaps by implementing calibrated multi-sensor integration, real-time IAQ index computation, cloud-based data storage, and a Flutter-based mobile interface for user-friendly visualization and remote monitoring.

3. SOFTWARE REQUIREMENTS SPECIFICATION

3.1 FUNCTIONAL REQUIREMENTS

The system must fulfill the following functional requirements to ensure effective Indoor Air Quality monitoring:

3.1.1 System Feature 1 – Real-Time Air Quality Monitoring

- The system must collect real-time Temperature, Humidity, VOC, and PM data using sensors.
- Sensor data should be processed and sent to ThingSpeak for cloud storage and analysis.

3.1.2 System Feature 2 – Android Mobile App for Live Data Display

- The app must fetch real-time air quality data from ThingSpeak API.
- Display the IAQ Index with categorized pollution levels (Good, Moderate, Poor).
- Show historical trends with graphical representation.

3.1.3 System Feature 3 – User Authentication & Data Storage

- Firebase Authentication must allow users to register/login.
- Firebase Realtime Database must store historical IAQ data per user for analysis.

3.1.4 System Feature 4 – IAQ Index Calculation and Interpretation

- The system must calculate the Indoor Air Quality (IAQ) index based on collected sensor data (Temperature, Humidity, VOC, PM).
- The mobile app must interpret the IAQ value and classify it into predefined levels (e.g., Good, Moderate, Poor).
- The calculated IAQ and its corresponding category should be clearly displayed in the app interface.

3.2 EXTERNAL INTERFACE REQUIREMENTS

3.2.1 User Interfaces

Flutter-based Android App with:

- A real-time dashboard showing IAQ data.
- Graphical charts for historical trends.
- A simple & intuitive UI for easy navigation.

3.2.2 Hardware Interfaces

- Sensors:
  - DHT11 (Temperature & Humidity)
  - ZP07 V4.0 (VOC Sensor)
  - PM2007 (PM Sensor)
- Microcontroller: ESP8266MOD with built-in Wi-Fi
- Power Source: USB or rechargeable battery
- Connection: Breadboard & jumper wires

3.2.3 Software Interfaces

- Arduino IDE: For programming ESP8266
- ThingSpeak: Cloud platform for data handling
- Firebase: User authentication & data storage
- Android Studio: For Flutter-based mobile app development
- Flutter & Dart: UI and app logic development

3.2.4 Communication Interfaces

- Wi-Fi Connection:
  - ESP8266 transmits sensor data via HTTP POST requests to ThingSpeak.
- Data shall be exchanged securely using HTTPS and REST APIs.

3.3 NON-FUNCTIONAL REQUIREMENTS

3.3.1 Performance Requirements

- The system must update sensor readings every 5-10 seconds.
- The mobile app should display real-time updates with minimal latency (<1s).
- Firebase should handle simultaneous user requests efficiently.

3.3.2 Security Requirements

- User authentication via Firebase Auth (email/password).
- Secure API calls using HTTPS encryption for ThingSpeak & Firebase.

3.4 SYSTEM REQUIREMENTS

3.4.1 Database Requirements

- Firebase Realtime Database -
  - Stores user profiles & authentication data.
  - Saves historical IAQ data for analysis.

3.4.2 Software Requirements (Platform Choice)

- Development Tools:
  - Arduino IDE (for ESP8266 programming)
  - Android Studio (for mobile app development)
  - Firebase Console (for database management)

- Languages & Frameworks:
  - C++ (for ESP8266 firmware)
  - Dart & Flutter (for Android app)
  - REST API (for communication)

3.4.3 Hardware Requirements

- Sensors: DHT11, ZP07 V4.0, PM2007
- Microcontroller: ESP8266MOD

- Communication Module: Wi-Fi
- Power Supply: USB/Rechargeable Battery
- Additional Components: Breadboard, Jumper Wires
- Processor: Intel Core i5 or higher
- RAM: 8GB or more

4. PROPOSED METHODOLOGY

4.1 PROPOSED ARCHITECTURE (BLOCK DIAGRAM)

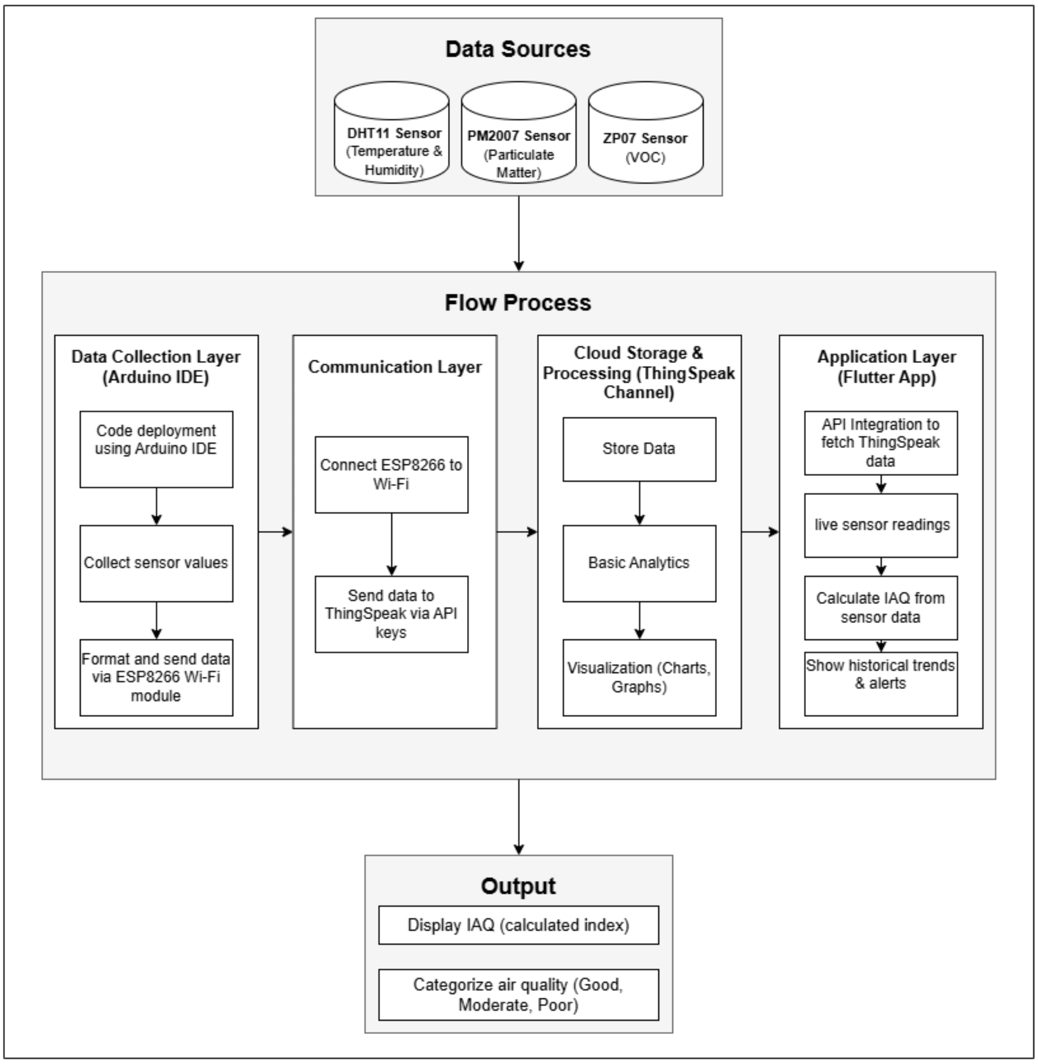


Fig 4.1 Proposed System Architecture

4.2 DATABASE DESIGN

The dataset used in this project is structured to monitor and evaluate indoor environmental conditions using real-time sensor data. The design supports API-based data fetching from ThingSpeak, IAQ (Indoor Air Quality) calculation, and historical trend analysis through mobile app visualization. The dataset comprises the following components:

1. Sensor Metadata:
    - DHT11 (Temperature and Humidity)
    - PM2007 (Particulate matter)
    - ZP07 (Volatile Organic Compounds)
2. Sensor Readings:
    - DHT11 (Temperature and Humidity)
    - PM2007 (Particulate matter)
    - ZP07 (Volatile Organic Compounds)
3. Data Processing and Feature Engineering:
    - Computation of IAQ using weighted formula
    - Mapping IAQ score to qualitative status
4. ThingSpeak Channel:
    - Acts as a cloud-based IoT platform
    - Each reading stored as field1 to field4
    - Accessible via ThingSpeak RestAPI
5. Visualization and App storage:
    - Stored locally in Firebase for app access
    - IAQ data used for graph generation
6. Firebase Firestore:
    - users Collection -
        - user_id (UID)
        - email, name, joined_on
    - Iaq_records Collection -
        - timestamp
        - temp, humidity, pm25, voc

- iaq_score, status

This design enables efficient retrieval, live monitoring, and historical analysis while maintaining a structured and extendable format for future enhancements (e.g., alert systems, room-wise comparison, AQI prediction using ML).

4.3 OVERVIEW OF PROJECT MODULES

Several useful components are integrated to create the Indoor Air Quality (IAQ) Monitoring and Alert System. Sensing environmental data, sending it to the cloud, determining IAQ levels, controlling users, and presenting insightful information via a mobile interface are just a few of the specialized duties that each module is in charge of. Together, these modules provide a dependable, user-focused, real-time IAQ monitoring experience.

4.3.1 Sensor Data Collection & Microcontroller Module

- Gathers environmental data using four sensors:
  - **DHT11** for measuring temperature and humidity
  - **PM2.5 sensor (PM2007)** for particulate matter
  - **ZP07** for volatile organic compounds (VOC)

- Using an ESP8266 NodeMCU, it periodically reads the values of analog and digital sensors
- Transforms sensor data into a readable format, and gets it ready for transmission.
- Reduces data loss during transmission and guarantees sensor reading synchronization.

4.3.2 Cloud Integration using ThingSpeak Module

- ESP8266 sends sensor data to ThingSpeak over HTTP POST requests.
- Each environmental parameter is stored in a separate field within a ThingSpeak channel.
- Enables real-time data display through public and private channels and API connection.

- Preserves a historical log of sensor data for analysis and trend identification in the future.

4.3.3 IAQ Computation & Analysis Module

- To obtain raw sensor data from ThingSpeak, use REST API endpoints.
- Determines the IAQ index using the weighted contributions of humidity, VOC, and PM2.5.
- Adheres to established IAQ thresholds and formulas (e.g., EPA, WHO).
- Provides meaningful labels and classifies IAQ values as Good, Moderate, Poor, or Hazardous.

4.3.4 Firebase Authentication & User Management Module

- Safe user signup and login are accomplished with Firebase Authentication
- Each user account is securely managed and individually identifiable by UID.
- Facilitates session monitoring and email/password-based access.
- Guarantees that individualized IAQ data and warnings are only accessible to authenticated users.

4.3.5 IAQ History Storage using Firebase Firestore Module

- Keeps sensor summaries, timestamps, and computed IAQ values under each user profile.
- Stores structured documents in a cloud NoSQL database called Firebase Firestore.
- Makes it possible to retrieve previous IAQ records for comparison, reporting, and visualization.
- Facilitates real-time syncing and data access across several devices.

4.3.6 Mobile Application Visualization Module

- Developed with the Flutter framework to provide a responsive, tidy user interface.
- Uses color-coded cards, charts, and gauges to show sensor data and IAQ results.
- Contains graph-based visualization for changes in IAQ on an hourly, daily, or weekly basis.
- Implements data refresh using ThingSpeak APIs and Firebase listeners.

4.4 TOOLS AND TECHNOLOGY USED

A wide range of hardware elements, software tools, and cloud platforms are used in the creation of the IAQ Monitoring and Alert System. These tools guarantee smooth user interaction across several modules as well as data collecting, processing, and storage.

4.4.1 Hardware Tools

- ESP8266 NodeMCU: A Wi-Fi-enabled microcontroller for data transfer and sensor interface.
- DHT11 Sensor: Determines humidity and temperature.
- PM2007 (PM2.5 Sensor): This device measures the amount of fine particulate matter in the air.
- ZP07 Sensor: Indicates the amount of volatile organic compounds (VOC) present.
- Resistors, breadboards, and jumper wires are used to connect sensors and set up circuits.

4.4.2 Software Tools

- Arduino IDE: This tool is used to upload firmware and program the ESP8266 microcontroller.
- The Flutter mobile application was developed using Android Studio.
- Visual Studio Code: For modifying code and troubleshooting the logic of mobile apps.

4.4.3 Programming Languages

- C/C++: Used to program microcontrollers based on Arduino.
- Dart: A programming language used in Flutter to create mobile applications.
- JSON & HTTP: For ThingSpeak and the mobile app to communicate via API.

4.4.4 Cloud Platforms & APIs

- ThingSpeak: Used to store, visualize, and retrieve sensor values from ESP8266 in real-time.
- Firebase Authentication: Oversees safe user registration, login, and session monitoring.

- Firebase Firestore: Holds user metadata, historical logs, and computed IAQ values.

4.4.5 Libraries & Packages

- Flutter SDK's Firebase Core, Auth, and Firestore: These are used to integrate Firebase services into apps.
- http (Flutter package): To send ThingSpeak API requests.
- Flutter Charts/Gauge Widgets: These are used in mobile user interfaces to visualize data in real time.

4.5 MATHEMATICAL MODEL

Pollutant sensor data, including Particulate Matter (PM2.5) and Volatile Organic Compounds (VOC), among others, are used to calculate Indoor Air Quality (IAQ). Using established breakpoint algorithms, each of these numbers is transformed into an AQI (Air Quality Index) scale before being added together to get the final IAQ index.

1. **Variables and Parameters**
   - $PM_{2.5}$ = Concentration of Particulate Matter in $\mu g/m^3$
   - VOC = Concentration of Volatile Organic Compounds in ppb
   - AQI_{PM} = AQI derived from PM2.5
   - AQI_{VOC} = AQI derived from VOC
   - IAQ = Final Indoor Air Quality Index

2. **AQI Calculation Formula**

   The standard AQI for any pollutant is calculated using the formula:

$$AQI = \frac{(I_{Hi} - I_{Lo})}{(BP_{Hi} - BP_{Lo})} \times (C - BP_{Lo}) + I_{Lo}$$

   Where:

   - C= Concentration of pollutant
   - BP_{Hi}, BP_{Lo}= AQI breakpoints corresponding to the pollutant concentration range
   - I_{Hi} , I_{Lo}= AQI index range corresponding to the breakpoints

3. **IAQ Index Calculation**

$$IAQ = \max(AQI_{PM}, AQI_{VOC})$$

This guarantees that the final IAQ index is determined by the pollutant that poses the greatest health risk.

4. **IAQ Category Mapping**

IAQ Index	Category
0–50	Good
51–100	Moderate
101–150	Unhealthy for Sensitive Groups
151–200	Unhealthy
201–300	Very Unhealthy
301–500	Hazardous

Table 4.1 IAQ Category Mapping

4.6 ALGORITHM DETAILS

4.6.1 IAQ Index Calculation Algorithm

- Converts raw results to an AQI scale using the usual AQI breakpoint formula and pollutant concentration (such as PM2.5 and VOC).
- Chooses the highest value to represent the final Indoor Air Quality Index after comparing several pollutant AQIs.
- For user interpretation, IAQ is then mapped to a health category (Good, Moderate, Unhealthy, etc.).

4.6.2 Data Normalization / Threshold Mapping

- Guarantees that sensor results from various units—such as ppb for VOC and µg/m^3 for PM—are scaled consistently.
- Aids in accurately comparing the quantities of pollutants by applying breakpoint thresholds.
- Avoids incorrect air quality classification brought on by variations in raw sensor values.

4.6.3 Real-Time Data Parsing Algorithm (Flutter App)

- Collects sensor values and parses JSON data obtained from the ThingSpeak API.
- Uses IAQ calculation logic and provides real-time pollution status updates to the user interface.
- Prior to display, make sure the data is formatted properly (e.g., decimal points, units).

4.6.4 Time-Series Visualization Logic

- Uses Flutter chart library to transfer IAQ values to graph coordinates after collecting them over time.
- Aids in each user's comprehension of hourly or daily patterns in air quality.
- A dynamic and understandable visual history is guaranteed by a seamless user interface transition.

4.7 COMPLEXITY OF PROJECT

4.7.1 Computational Complexity

- **Sensor Data Processing Time:**
  - Real-time data fetching from ThingSpeak takes approximately 1–2 seconds.
  - IAQ calculation is instantaneous (within milliseconds).
- **Memory Usage:**
  - 50–100 MB RAM usage on mobile during runtime for UI and HTTP parsing.
- **Big-O Notation:**
  - Data fetching: O(1) (single API call).
  - IAQ computation: O(1) (direct arithmetic calculation with fixed inputs).

4.7.2 Algorithmic Complexity

- **IAQ Formula:**
  - Constant-time arithmetic operations: O(1).
- **Optional Classification Logic:**

- Threshold-based categorization: O(1).

4.7.3 Implementation Complexity

- **Lines of Code:**
  - Estimated 400–600 lines (Flutter UI, ThingSpeak API integration, IAQ logic).
- **Number of Dependencies:**
  - 6–8 packages (http, flutter_dotenv, provider, charts_flutter, etc.).
- **Integration Complexity:**
  - Integration of:
    - Sensor data from Arduino via ThingSpeak,
    - API fetch into Flutter,
    - UI update with IAQ and visualization.
- **Code Modularity:**
  - Separated into components for:
    - Data services (API handling),
    - UI rendering,
    - IAQ computation and classification.

4.7.4 Resource Complexity

- **Hardware Requirements:**
  - ESP8266 NodeMCU for IoT connectivity
  - Sensors: DHT11 (Temp & Humidity), PM2007 (PM2.5/PM10), ZP07 (VOC), CO_2 sensor
  - Smartphone (Android device) to run the Flutter application
- **Storage Requirements:**
  - Minimal local storage on mobile device (~20MB APK size)
  - Cloud storage via ThingSpeak (sufficient for real-time sensor data logging)
  - No heavy dataset or model storage required
- **Scalability:**
  - Modular Flutter codebase allows easy UI and logic extensions

- ThingSpeak supports channel expansion or migration to more robust platforms (e.g., Firebase, AWS IoT) for future scaling
- Additional sensors or locations can be integrated with minimal code changes

4.8 SDLC MODEL TO BE APPLIED

Chosen Model: Agile Methodology

Fig 4.2 SDLC Model

Agile Methodology in This Project

Agile methodology was selected for the Indoor Air Quality (IAQ) Monitoring System project due to its iterative, flexible, and team-oriented nature, which aligns perfectly with the hardware-software integration and evolving requirements of our system. The project involves sensor data acquisition (DHT11, PM2007, ZP07 sensor), microcontroller programming, API communication with ThingSpeak, and a Flutter-based mobile application. These modules are best handled in short, manageable sprints under the Agile model.

1. **Sprint-Based Development**

- Each module—sensor interfacing, Wi-Fi (ESP8266) communication, ThingSpeak API setup, and Flutter UI development—was broken down into focused sprints.
- This allowed us to develop, test, and validate one component at a time while ensuring steady progress.

2. Flexibility to Accommodate Changes

- Hardware-based projects often require dynamic changes due to sensor limitations, API response issues, or real-time performance tuning.
- Agile allows us to adapt to such changes without disrupting the overall workflow, ensuring continuous improvement throughout the lifecycle.

3. Component-Wise Backlog and Prioritization

- We created user stories for each functional aspect—like "send DHT11 readings to ThingSpeak" or "display AQI color-coded on the app."
- These stories helped us build a clear backlog, prioritize high-impact features first (e.g., real-time data fetching and display), and track development efficiently.

4. Collaborative Communication

- With team members working on different parts (Arduino code, API setup, Flutter UI), Agile stand-ups and regular reviews helped us stay in sync.
- Problems like sensor noise, data delay, or UI bugs were quickly addressed through team coordination.

5. Testing and Validation in Every Sprint

- After each sprint, we tested sensor readings, cloud integration, and UI display.
- This reduced the chances of system failure during integration, and also helped improve reliability—critical for real-time IAQ monitoring systems.

6. **User-Focused Improvements**

   - We gathered continuous feedback on the app interface, sensor performance, and real-time response to improve the system's usability.
   - Agile's user-focused nature ensured the final product was both functional and user-friendly.

7. **Cross-Functional Collaboration**

   - Developers and testers worked together, ensuring smooth integration of different system components.

Justification:

- The project involves distinct modules—sensor interfacing, data transmission, cloud storage, and mobile app visualization—making Agile's sprint-based approach ideal for focusing on one task at a time while maintaining overall progress.
- IoT projects often face hardware limitations, sensor calibration needs, or API constraints. Agile allowed quick adaptation to such changes without overhauling the entire system, ensuring smoother iteration and refinement.
- Team members worked on hardware (ESP8266), cloud APIs (ThingSpeak), and app development (Flutter), making Agile’s emphasis on communication and coordination essential to avoid delays and integration issues.
- Regular sprint testing ensured that real-time data, cloud connectivity, and UI components were functioning correctly, reducing risk during final integration. Agile supported early bug detection and feature refinement.
- Agile enabled early feedback collection from users on data display, app interface, and interpretation of IAQ levels. This helped improve the usability and practicality of the final system.

4.9 UML DIAGRAMS

1. Use Case Diagram

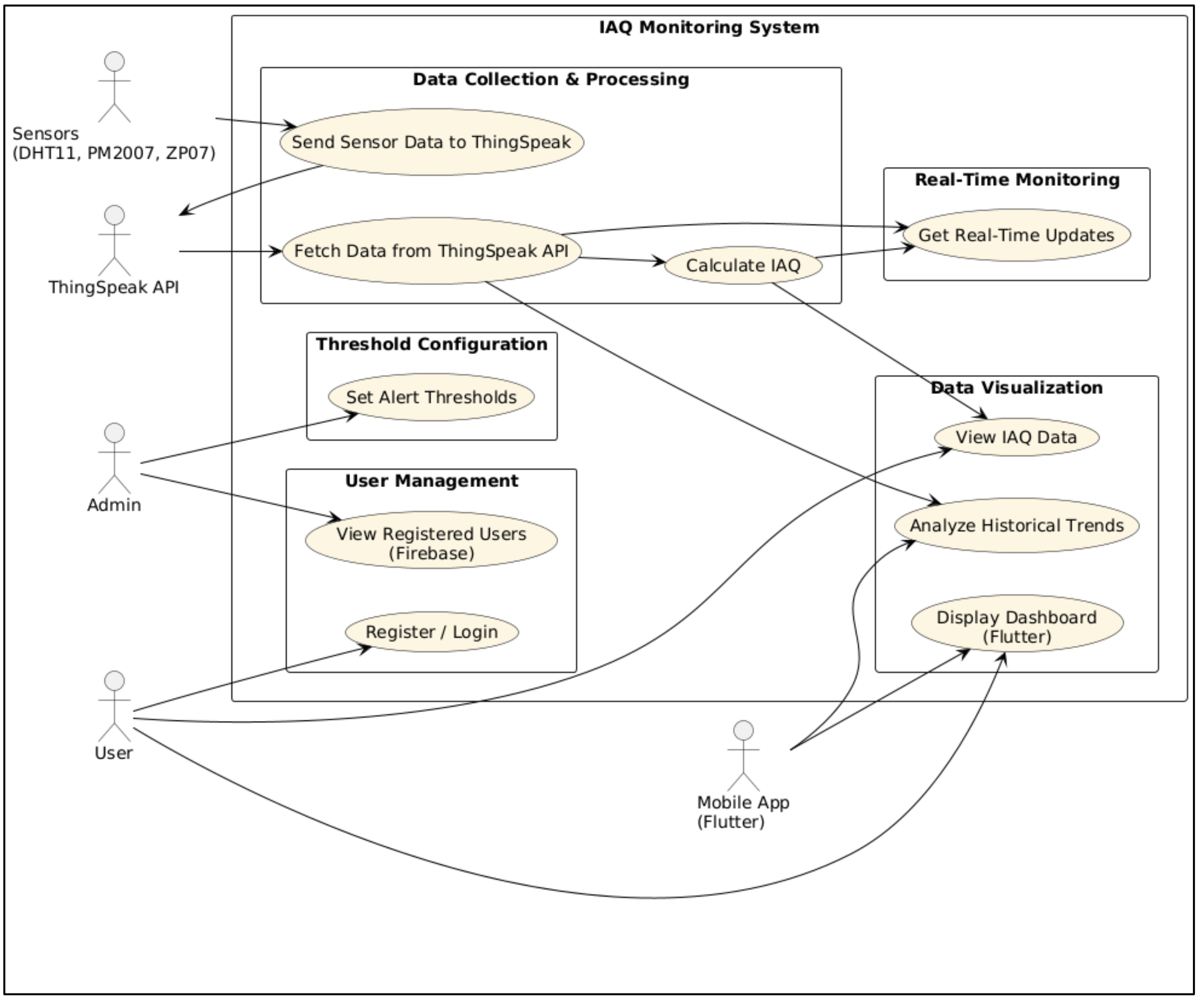


Fig 4.3 Use Case Diagram

2. Class Diagram

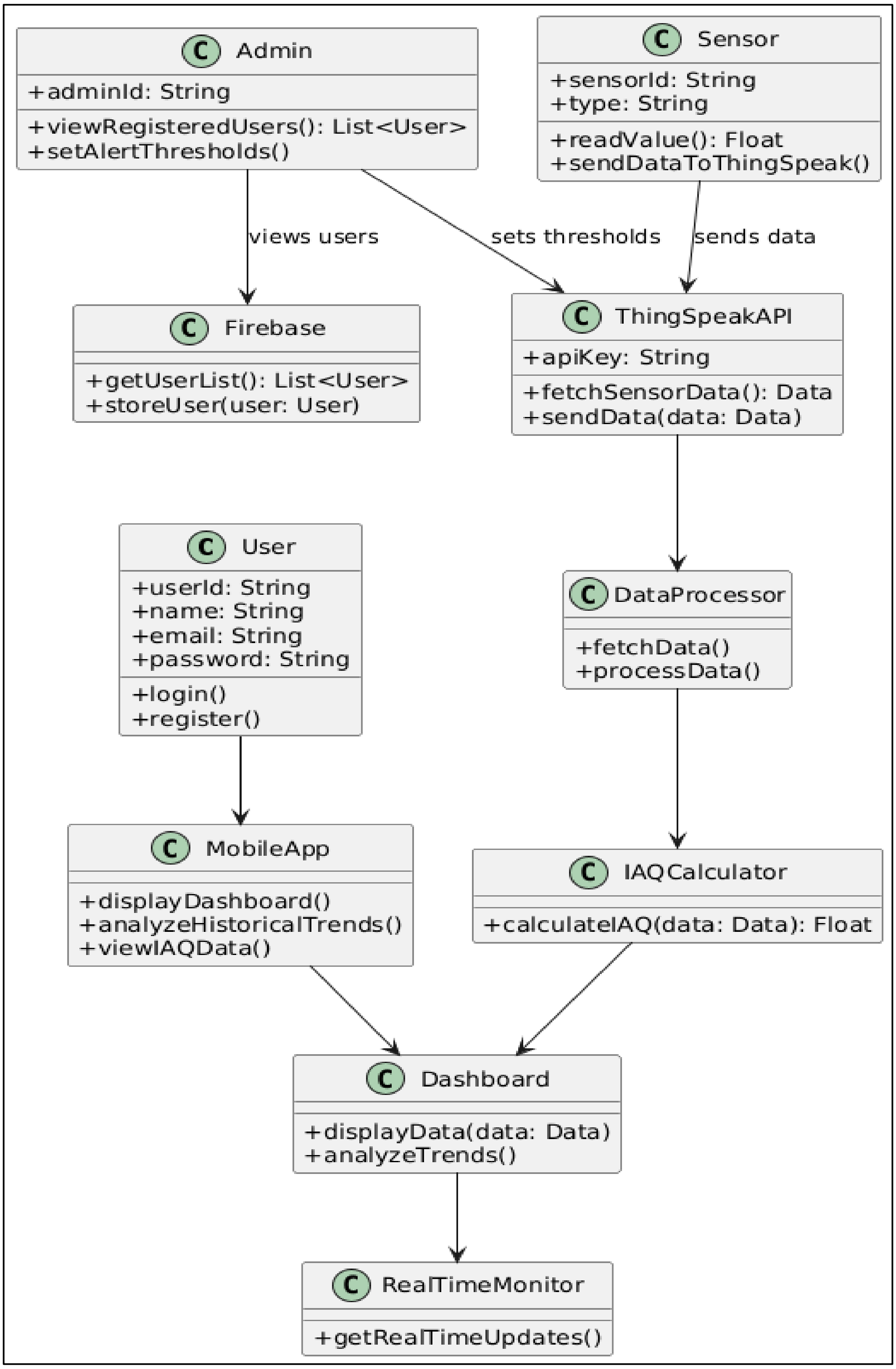


Fig 4.4 Class Diagram

3. Component Diagram

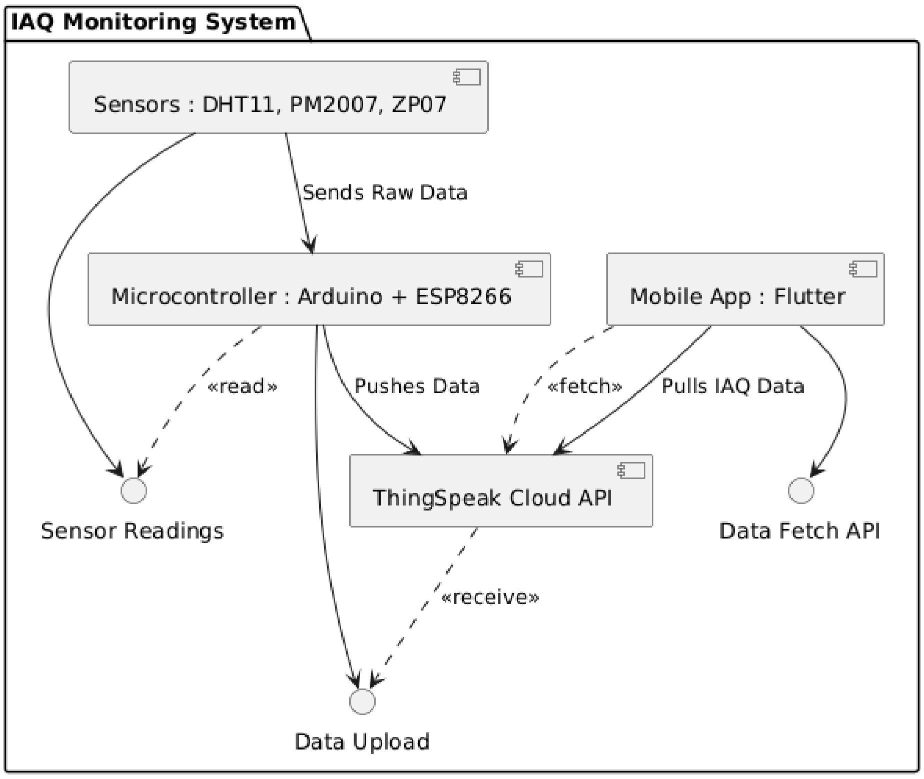


Fig 4.5 Component Diagram

4. Deployment Diagram

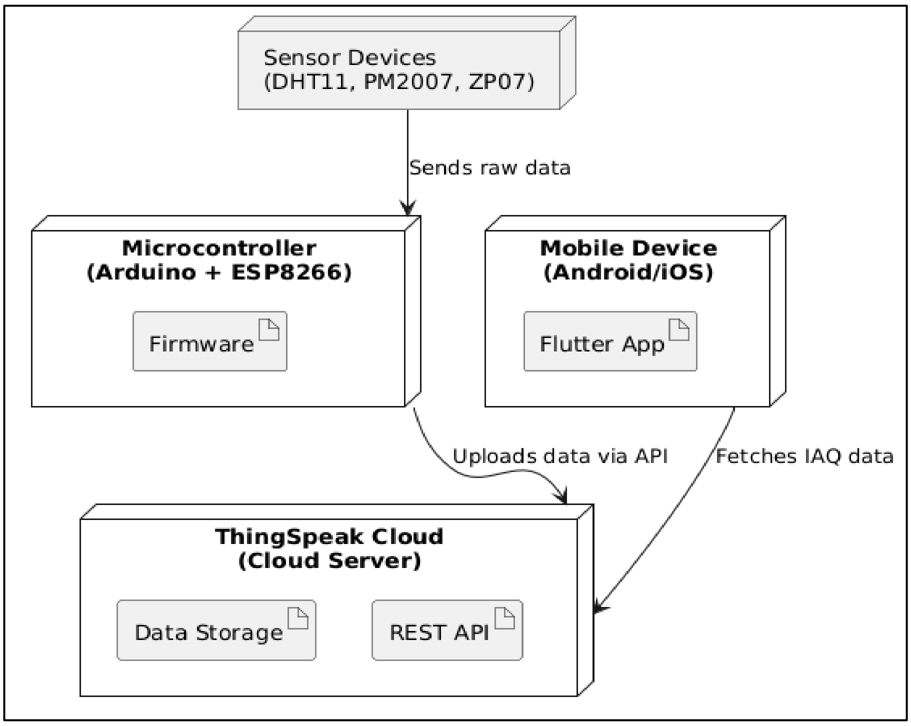


Fig 4.6 Deployment Diagram

5. Activity Diagram

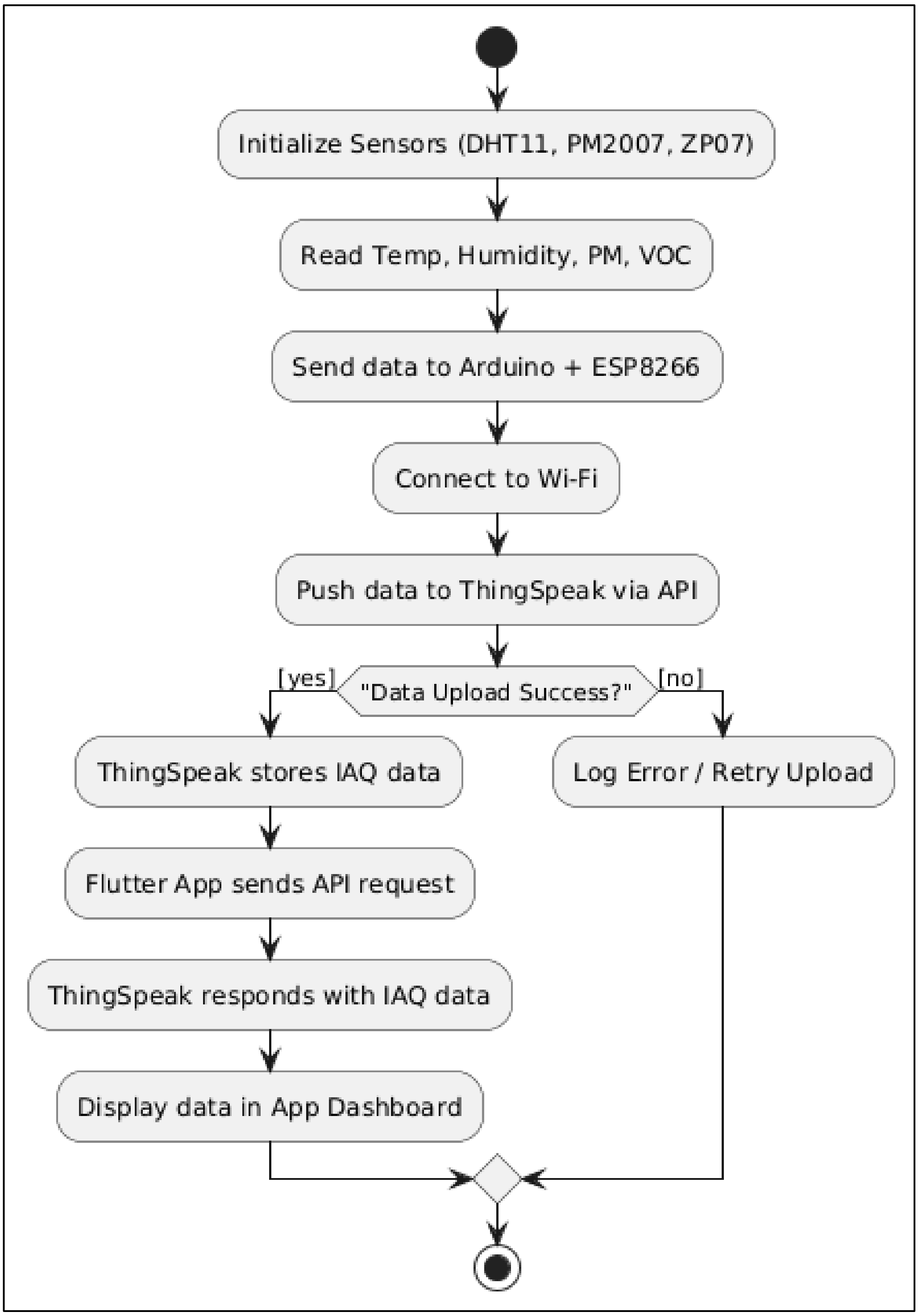


Fig 4.7 Activity Diagram

6. Sequence Diagram

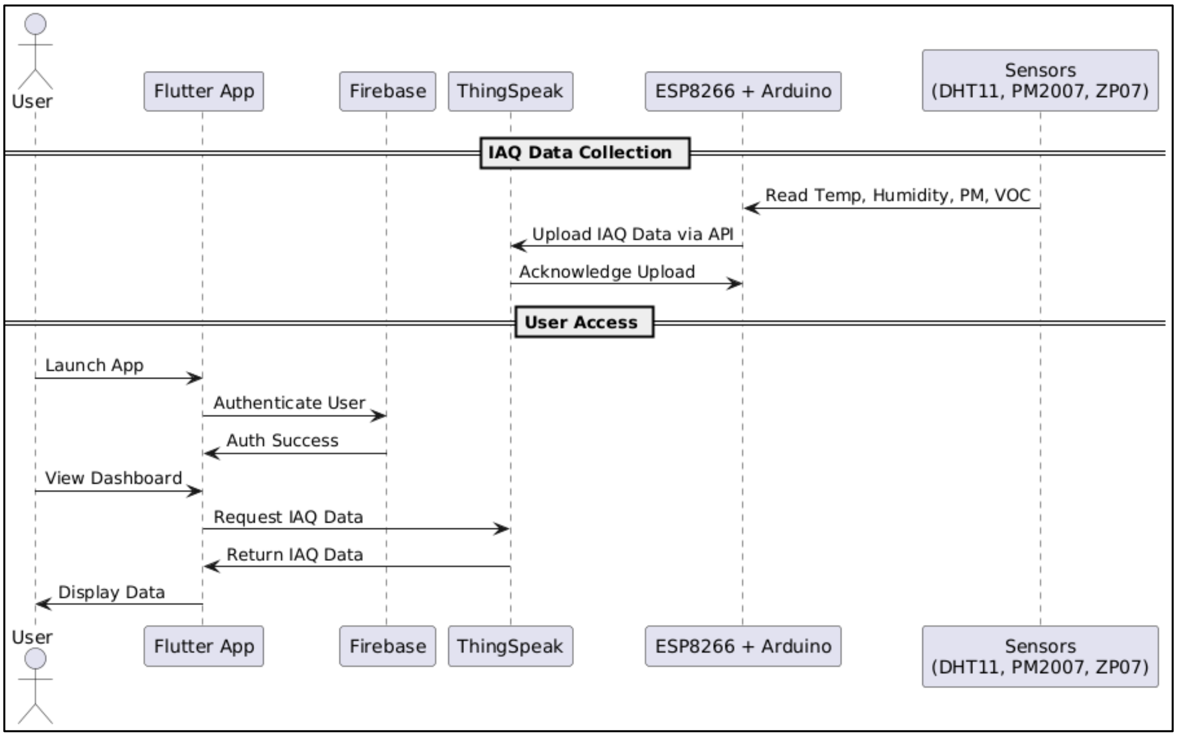


Fig 4.8 Sequence Diagram

5. PROJECT PLAN

5.1 PROJECT COST ESTIMATE

5.1.1 COCOMO Model

1. Assumptions:

- Project Type: Organic (small, simple, familiar — perfect fit for academic + IoT + app project)
- Estimated Lines of Code (LoC):
    - Arduino code: ~600 LoC
    - Flutter app (Dart): ~1500 LoC
    - Firebase integration: ~400 LoC
    - Total = 2500 LoC ≈ 2.5 KLoC

2. Basic COCOMO Formula (Organic Type):

- Effort (E) = $2.4 \times (2.5)^{1.05} \approx 6.3$ person-months
- Development Time (D) = $2.5 \times (6.3)^{0.38} \approx 3.6$ months
- Average Team Size = E / D = 6.3 / 3.6 ≈ 2 people

3. Cost Estimation:

- Assuming ₹25,000/month per developer

  Total Estimated Cost = 6.3 × ₹25,000 = ₹1,57,500

5.1.2 Computational Costs

- Processing Power:
    - A basic setup using Arduino IDE with ESP8266 does not require heavy processing units.
    - Estimated Cost: ₹1,500 – ₹3,000

- Memory Requirements:
    - Minimum: 8GB RAM (16GB recommended)
    - Estimated Cost: ₹10,000 – ₹20,000

- Storage Needs:
    - SSD: 500GB for data caching, API logs, and offline model/data handling
    - Estimated Cost: ₹8,000 – ₹15,000
- Network Bandwidth:
    - Required: Minimum 100 Mbps connection for real-time API calls to ThingSpeak/Firebase
    - Monthly Cost Estimate: ₹3,000 – ₹10,000

5.1.3 Hardware Costs

- Microcontroller & Communication:
    - NodeMCU ESP8266 Board: ₹250 – ₹400
- Sensors Used:
    - DHT11 (Temperature & Humidity): ₹120 – ₹200
    - PM2.5 Sensor (PMS7003/PM2007): ₹3500 – ₹4000
    - ZP07 (VOC Sensor): ₹600 – ₹1,000
- Additional Components:
    - Breadboard, jumper wires, resistors, power supply: ₹300 – ₹500

Total Hardware Cost: ₹4,500 – ₹5,500

5.1.4 Software Performance Costs

- Algorithm Complexity:
    - The system uses basic data interpretation without heavy computation or ML, making it cost-efficient.
    - Estimated Cost: ₹0 for training; ₹2,000 – ₹5,000/month for API access (Firebase, ThingSpeak)
- Database Query Performance:
    - Firebase handles real-time read/write of sensor data with minimal latency.
    - Estimated Cost: ₹3,000 – ₹10,000 (for Firebase usage and API quotas)

- Cloud Hosting & API Calls:
    - Firebase and ThingSpeak are used to host and visualize sensor data through APIs.
    - Estimated Cost: ₹5,000 – ₹10,000/month (depending on data points and update frequency)

5.2 RISK MANAGEMENT

5.2.1 Risk Identification

Possible threats in the IoT-Based Handheld Environmental and Air Quality Monitoring Station are:

1. Hardware Failures

- Malfunctions in sensors (DHT11, ZP07, PM2007) based on environmental conditions.
- ESP8266 connectivity issue resulting in data transmission failure.
- Power cuts disrupting continuous monitoring.

2. Data Transmission Risks

- Wi-Fi connection problems resulting in real-time loss of data.
- API failure or ThingSpeak rate limits impacting cloud data storage.
- Delayed updates for data affecting real-time visualization.

3. Software & Security Risks

- Mobile application crashes or bugs that result in the wrong representation of IAQ data.
- Security breaches allowing unauthenticated access to user data.
- Corruption of data in Firebase or ThingSpeak resulting in incorrect historical records.

4. Operational & User Risks

- Users interpreting IAQ data wrongly, resulting in improper health decisions.
- Challenges in sensor calibration, compromising measurement accuracy.
- High power consumption necessitating frequent battery replacement.

5.2.2 Risk Analysis

Risk Category	Likelihood	Impact	Severity	Mitigation Plan
Sensor Failures	Medium	High	High	Periodic sensor calibration and maintenance.
Wi-Fi connectivity issues	High	Medium	High	Store temporary data locally and sync when online.
API failure	Medium	High	High	Implement fallback storage and retry mechanisms.
Mobile app bugs	Medium	Medium	Medium	Conduct thorough testing and debugging.
Unauthorized access	Low	High	High	Secure authentication with Firebase Auth & HTTPS
Power supply issues	High	High	High	Use rechargeable battery and power-saving modes
Data corruption	Low	Medium	Medium	Implement regular backups for Firebase data

Table 5.1 Risk Analysis

5.2.3 Overview of Risk Mitigation, Monitoring and Management

1. Risk Mitigation (Preventive Measures)

- Hardware Redundancy: Use quality-rated sensors and include watchdog timers or restart logic in NodeMCU to recover from unresponsive states.
- Stable Power Supply: Integrate battery backup or power regulation circuitry to prevent outages from disrupting data capture.
- API & Cloud Optimization: Implement local data caching to buffer sensor readings during internet outages and avoid breaching ThingSpeak API rate limits.

2. Risk Monitoring (Ongoing Checks)

- System Health Logging: Periodically log sensor response, connectivity status, and update timestamps in Firebase for tracking anomalies.
- Error Notification: Use in-app alerts or logs to notify users of Wi-Fi disconnection, data loss, or failed API calls.
- Version Control & Testing: Continuously test mobile app builds and backend endpoints to detect bugs and performance issues early.

3. Risk Management (Response & Control Strategy)

- Regular Calibration Protocol: Schedule periodic sensor calibration checks to ensure data accuracy, especially for VOC and PM sensors.
- Secure Authentication: Use Firebase Auth with email/password or OTP verification to prevent unauthorized access to user profiles and IAQ data.
- User Education: Provide tooltips or an IAQ interpretation guide in the app to help users correctly understand pollution levels and suggested actions.

5.3 PROJECT SCHEDULE

5.3.1 Project Task Set

1. **Requirement Analysis**
    - Identified hardware components including ESP8266 (NodeMCU), DHT11, PM2007and ZP07 VOC sensor.
    - Specified software requirements such as ThingSpeak for cloud storage, Firebase for real-time app sync, and Flutter for mobile interface.
    - Outlined system goals—real-time environmental monitoring, IAQ index calculation, and data visualization.

2. **Hardware Assembly & Integration**
    - Assembled all sensors with the ESP8266 microcontroller on a compact board with voltage regulation.
    - Verified wiring, data pin configurations, and ensured reliable sensor output via serial monitor.
    - Added OLED display (optional) for onboard sensor reading preview.

3. **Sensor Calibration & Testing**
   - Calibrated sensors in different conditions (e.g., indoors, near traffic, closed rooms) for accuracy tuning.
   - Compared sensor outputs with standard air quality references (AQI standards).
   - Documented sensor drift and added auto-correction factors in firmware if needed.

4. **Firmware Development (ESP8266/Arduino)**
   - Wrote Arduino code to read sensor values and send data to ThingSpeak via Wi-Fi.
   - Added error handling and retry mechanisms for data upload failures.
   - Included IAQ index formula in firmware for quick on-device evaluation.

5. **Cloud & Database Integration**
   - Integrated ThingSpeak for live environmental data logging and visualization.
   - Used Firebase Realtime Database to sync IAQ and sensor data with the mobile app.
   - Ensured secure API key usage and real-time data refresh via Firebase listeners.

6. **Frontend Development (Flutter)**
   - Designed an intuitive mobile interface displaying live IAQ index, sensor readings, and color-coded air quality levels.
   - Used ThingSpeak and Firebase APIs to fetch and update data in real-time.
   - Created charts to show historical trends and implemented UI alerts for poor air quality levels.

7. **Backend Development**
   - Built backend logic in Dart to fetch and parse JSON data from Firebase and ThingSpeak.
   - Managed user authentication (if required) and database connectivity for app personalization.
   - Added background sync service to pull latest readings at regular intervals.

8. **Testing & Evaluation**
    - Performed unit testing on Flutter widgets and integration testing between ESP8266, ThingSpeak, and Firebase.
    - Evaluated system performance metrics like data lag, mobile app sync time, and battery efficiency.
    - Conducted stress testing by simulating continuous sensor input.

9. **Deployment & Field Testing**
    - Deployed the handheld unit in home, office, and factory environments to observe real-time variations.
    - Monitored IAQ output reliability and conducted survey-based feedback from trial users.
    - Identified edge-case issues (e.g., Wi-Fi loss) and implemented fixes.

10. **Documentation & Reporting**
    - Created comprehensive documentation including block diagrams, circuit schematic, flowcharts, and data sample screenshots.
    - Compiled weekly progress reports, final project report, and user manual for app and hardware usage.
    - Created a poster and presentation slides for project demonstration and viva.

5.3.2 Timeline Chart

Phase	Tasks	Duration
Research and Planning	• Literature review on IAQ & sensors • Finalize sensor selection & requirements	2 weeks
Hardware Setup	• Connect and test sensors with Arduino • Calibrate sensors for accurate readings	2 weeks
System Development	• Program ESP8266 for data transmission • Send sensor data to ThingSpeak	3 weeks
App Development	• Build Flutter frontend to fetch/display data • Integrate ThingSpeak APIs in app	3 weeks
Testing and Debugging	• Validate real-time data accuracy • Test app responsiveness and edge cases	2 weeks
Deployment and Review	• Final deployment of system • Optimize performance & power usage • Prepare documentation	2 weeks

Table 5.2 Project Plan (Timeline)

5.4 SUSTAINABILITY OF PROJECT

1. **Environmental Monitoring for Public Health**

- Continuously tracks indoor air quality to help reduce respiratory and allergic diseases by identifying poor air quality zones.
- Raises public awareness by providing real-time data, encouraging users to adopt eco-friendly practices such as better ventilation or reducing indoor pollutants.

2. **Low-Power and Reusable Hardware**

- Utilizes ESP8266 and efficient sensors that operate on minimal power, reducing overall energy consumption.
- Promotes electronic sustainability by allowing hardware reuse and encouraging sensor calibration over replacement, minimizing e-waste.

3. Cloud-Based Data Storage

- Uses ThingSpeak and Firebase to store and visualize data remotely, eliminating the need for paper-based logs and reducing hardware dependency.
- Ensures long-term accessibility and backup of environmental data, enabling trends analysis and predictive modelling without physical storage.

4. Scalability and Cost-Effectiveness

- The modular design allows easy integration of additional sensors or expansion to other buildings, enhancing scalability.
- Leverages open-source platforms and low-cost components, making the solution budget-friendly and ideal for large-scale institutional or community deployment.

5. Educational and Research Applications

- Acts as a practical learning tool for students to understand IoT, sensor networks, and data visualization, promoting STEM education.
- Provides a research-ready framework that can be adapted for academic studies, environmental policy trials, or smart city integration projects.

6. SOFTWARE TESTING

6.1 TYPES OF TESTING

6.1.1 Unit Testing

Unit testing focuses on verifying individual modules of the system in isolation. For this project, functions like sensor data reading (DHT11, PM2007, ZP07), IAQ index calculation, and Flutter API fetch methods are tested separately. This ensures that each block performs its expected task correctly. Errors can be quickly identified and fixed at the component level. Arduino code functions and Flutter service classes are ideal candidates for this type of testing.

6.1.2 Integration Testing

Integration testing ensures smooth interaction between modules such as sensor data capture, microcontroller data transmission, cloud storage (ThingSpeak), and Flutter data retrieval. For example, if ESP8266 sends faulty data, the app must handle and display it appropriately. This step ensures the components work well together as a unit. It is especially crucial in IoT projects where hardware-software interaction is critical.

6.1.3 System Testing

System testing is performed to validate the entire IAQ Monitoring System as a complete and functioning product. It covers everything from powering the NodeMCU and collecting real-time sensor data to pushing it to ThingSpeak and fetching it in the mobile app. This tests both hardware and software integration in real-world scenarios. It ensures the end-to-end system works as intended and meets user requirements.

6.1.4 Regression Testing

Regression testing is done after updates or feature additions to ensure existing functionalities are not broken. For instance, after improving the IAQ index formula or modifying the app UI, previous features like live data display and Firebase connectivity are re-tested. This ensures stability is maintained throughout the development cycle. It's essential to avoid introducing new bugs while fixing old ones.

6.1.5 Security Testing

Security testing ensures that the system is protected from unauthorized access. This includes validating Firebase authentication, secure data transmission, and restricted access to user-specific data. The goal is to prevent data leakage, hacking, or misuse of stored sensor readings. Proper security is especially important for systems involving personal or environmental data.

6.1.6 Compatibility Testing

Compatibility testing verifies that the mobile app functions correctly on different Android versions and screen sizes. It ensures that UI elements, charts, and data fields adjust properly across devices. This helps make the system accessible to a broader user base. It also includes checking performance on lower-end phones.

6.1.7 Usability Testing

Usability testing focuses on user experience with the Flutter mobile application. It involves checking the clarity of IAQ levels, color-coded warnings, chart readability, and navigation ease. Feedback from potential users is gathered to refine UI design. A user-friendly interface is key to making the app effective for all audiences.

6.2 TEST CASES AND TEST RESULTS

6.2.1 Unit Testing

ID	Description	Input	Expected Output	Status
TC01	Successful user registration	Valid name, email, and password	Account is created successfully	Pass
TC02	Registration with invalid email	Invalid email format	Error message: "Invalid email format"	Pass
TC03	Login with correct credentials	Valid email and password	User is redirected to dashboard	Pass
TC04	Login with incorrect password	Valid email, wrong password	Error message: "Incorrect credentials"	Pass
TC05	Display sensor data on dashboard	App launch with sensors active	Sensor values are displayed in real-time	Pass
TC06	IAQ calculation with valid input	Temperature, humidity, VOC, PM values	IAQ Index and Category shown correctly	Pass
TC07	Profile update with new email	Updated email address	Profile updated confirmation	Pass
TC08	Change password from profile	New password input	Password changed successfully	Pass
TC09	Navigation to History page	Click on "History" in side menu	Redirects to History screen	Pass
TC10	Add record to history after IAQ calculation	Perform IAQ calculation	New record appears in history list	Pass

Table 6.1 Unit Testing

6.2.2 Integration Testing

ID	Description	Input	Expected Output	Status
IT01	Register and autologin flow	Valid registration details	User registered and redirected to dashboard	Pass
IT02	Login and load real time sensor data	Valid login credentials	Dashboard loads with real-time sensor readings	Pass
IT03	IAQ Calculation updates dashboard and history	Calculate IAQ after sensor data loads	IAQ displayed and entry added in History	Pass
IT04	Update profile and verify dashboard name/email updates	Update profile name and email	New name/email reflected on dashboard/profile	Pass
IT05	Change password and validate new login	Change password, log out, then login with new password	Login successful with new password	Pass
IT06	Calculate IAQ → View History entry	Perform IAQ calc → Navigate to History	Recent IAQ entry shown in History	Pass
IT07	View Profile info after login	Login → Go to Profile screen	Shows correct user details (name, email)	Pass
IT08	Logout and ensure user is redirected to login screen	Tap Logout from sidebar menu	Redirected to login screen	Pass
IT09	Login with another account and verify separate history/profile	Login with a different user	Different profile info and history shown	Pass
IT10	Ensure IAQ calculation only works after sensor data is loaded	Click "Calculate IAQ" without sensor data	Show error or no action until sensor values are ready	Pass

Table 6.2 Integration Testing

6.2.3 System Testing

ID	Description	Input	Expected Output	Status
ST01	End-to-end user flow: Register → Login → Monitor IAQ	New user details, valid IAQ inputs	User can complete the full cycle and see IAQ result	Pass
ST02	Validate real-time sensor data updates on dashboard	Launch dashboard with active sensors	Temperature, humidity, VOC, and PM values update in real-time	Pass
ST03	IAQ calculation based on latest sensor readings	Live sensor values	IAQ Index and Category calculated and displayed correctly	Pass
ST04	View full IAQ history for logged-in user	User with multiple IAQ calculations	History screen shows all previous records accurately	Pass
ST05	Profile update reflects across system	Change name/email in Profile screen	New info reflected on sidebar and login details	Pass
ST06	Password change updates credential and allows secure relogin	New password input	System accepts new password on next login	Pass
ST07	User logout redirects to login and prevents access to internal pages	Tap logout	Redirects to login; blocks unauthorized access to dashboard/history	Pass
ST08	Ensure system handles invalid inputs (e.g. empty fields, wrong email)	Leave fields blank or input bad formats	Shows validation errors and prevents form submission	Pass
ST09	Navigation between all screens works smoothly	Navigate via sidebar menu (Home, History, Profile)	Each section loads correctly and with expected data	Pass
ST10	System performs consistently under typical usage	Register/login, monitor IAQ, update profile, view history	No crashes, hangs, or inconsistencies observed	Pass

Table 6.3 System Testing

6.2.4 Regression Testing

ID	Description	Input	Expected Output	Status
REG01	Validate login still works after backend update	Valid email and password	User logs in successfully	Pass
REG02	Ensure IAQ calculation works after updating sensor logic	Valid sensor data inputs	IAQ Index calculated and displayed correctly	Pass
REG03	Verify real-time sensor data still displays on dashboard after update	Open dashboard with active sensors	Real-time values displayed correctly	Pass
REG04	Check profile update still reflects after UI changes	Update name/email	Updated info appears on dashboard and profile	Pass
REG05	Ensure logout still clears session after session management update	Click logout	User is redirected to login; session cleared	Pass
REG06	IAQ history log remains functional after database optimization	Perform IAQ calculation	New entry added to History	Pass
REG07	Navigation works after UI refactoring	Click through menu: Home → History → Profile	Each screen loads correctly	Pass
REG08	Registration still functional after validation logic changes	Valid registration info	New account created and user logged in	Pass
REG09	Check error messages still appear on invalid input after form updates	Submit empty login form	Validation message: "Fields cannot be empty"	Pass
REG10	Ensure IAQ colour category display still works after style changes	IAQ Index: 2.0	UI shows “Good” with correct colour label	Pass

Table 6.4 Regression Testing

6.2.5 Security Testing

ID	Description	Input	Expected Output	Status
SEC01	Prevent login with incorrect credentials	Wrong email or password	Show "Invalid credentials" message; no access granted	Pass
SEC02	Passwords should be stored in encrypted format	Register new user	Password is hashed/encrypted in the backend database	Pass
SEC03	Session should expire after logout	Login → Logout → Try accessing dashboard via back button	Redirected to login screen	Pass
SEC04	Input validation to prevent SQL Injection	Enter ' OR 1=1 -- in login fields	Input rejected or sanitized	Pass
SEC05	No access to user history without login	Open history URL directly in browser without login	Redirect to login or show "Unauthorized access"	Pass
SEC06	Enforce password strength rules	Use weak password like 123 or abcd during registration	Show validation message: "Password too weak"	Pass
SEC07	Ensure HTTPS is used for API calls	Network monitor during API requests	All API endpoints use https://	Pass
SEC08	Restrict access based on user role (if applicable)	Normal user tries accessing admin endpoint (if any)	Access denied or forbidden	Pass
SEC09	Secure logout should clear session token	Logout → Inspect session/cookies	Session token is invalidated or removed	Pass
SEC10	Prevent Cross-Site Scripting (XSS) in input fields	Input <script>alert('XSS')</script> in profile fields	Input is sanitized and no script is executed	Pass

Table 6.5 Security Testing

6.2.6 Compatibility Testing

ID	Description	Input / Test Condition	Expected Output	Status
COM01	App functionality on Android 10	Install and run app on Android 10	App installs, launches, and works as expected	Pass
COM02	App functionality on Android 14	Install and run app on Android 14	All features work without crash or UI distortion	Pass
COM03	App UI compatibility with small screen (5")	Open app on 5" screen phone	UI elements are responsive and scrollable if needed	Pass
COM04	App UI compatibility with large screen (6.8"+)	Open app on large screen smartphone	UI adapts properly, no pixelation or misalignment	Pass
COM05	App behaviour on low network connectivity	Switch to 2G or weak Wi-Fi during AQI calculation	App handles delay gracefully or shows retry option	Pass
COM06	Functionality on iOS device (if applicable / hybrid app)	Launch app on iPhone (iOS 16)	App runs with full feature set and proper UI	N/A / Pass
COM07	Landscape vs Portrait orientation	Rotate device on dashboard and history screen	Layout adjusts smoothly without cutting off content	Pass
COM08	Browser compatibility for web version (if applicable)	Open web dashboard in Chrome, Firefox, Edge	All browsers render UI consistently	N/A / Pass
COM09	Dark mode support	Enable system dark mode	App elements adjust colours accordingly	Pass
COM10	App behaviour during background foreground switch	Minimize and reopen app	App resumes in same state without crashing	Pass

Table 6.6 Compatibility Testing

6.2.7 Usability Testing

ID	Description	Input / Interaction	Expected Output / Behaviour	Status
UT01	First-time user registration flow clarity	User signs up without help	Steps are intuitive, minimal, and clearly labelled	Pass
UT02	Login screen ease of use	Enter email and password	Form is clean, focused, with helpful error messages	Pass
UT03	Dashboard readability and sensor value display	View sensor values	Clear labels, colour coded AQI info, easy to understand	Pass
UT04	Button size and spacing for touch usability	Try tapping menu icons/buttons	No accidental taps, buttons are easy to press	Pass
UT05	Navigation through sidebar and tabs	Navigate: Dashboard → History → Profile	User can move between screens with minimal effort	Pass
UT06	AQI calculation process intuitiveness	View data → Tap "Calculate AQI"	User understands what AQI means and how to trigger calculation	Pass
UT07	Profile update process simplicity	Try updating email/name	Minimal steps, user gets visual confirmation	Pass
UT08	Error handling clarity	Leave login field empty / type wrong input	Clear, specific, user-friendly error messages shown	Pass
UT09	Font and colour accessibility	Scan dashboard for text clarity	Text is legible, good contrast, colour friendly UI	Pass
UT10	Overall satisfaction with app flow (feedback prompt)	After completing a task, app shows a rating/feedback option	User can rate or submit feedback easily	Pass

Table 6.7 Usability Testing

7. RESULTS AND DISCUSSION

7.1 OUTCOMES

An Internet of Things (IoT)-based indoor air quality monitoring system was successfully created to detect important environmental factors as temperature, humidity, PM2.5, PM10, and volatile organic compounds (VOCs). The system used sensors that were interfaced with a microcontroller (ESP32, for example) and sent data to a cloud platform in real time. Additionally, a web-based dashboard was included for ongoing remote monitoring.

Key outcomes include:

1. **Real-time Environmental Monitoring**

   - Particulate matter (PM), volatile organic compounds (VOC), temperature, humidity, and CO_2 can all be measured accurately with calibrated sensors.
   - A weighted scoring system is used to calculate the instant IAQ (Indoor Air Quality) index.
   - Assists users in acting quickly in the event of bad air quality.

2. **User-Friendly Mobile Application**

   - Designed using Flutter, this interface is beautiful and responsive.
   - Three main screens divided into: Profile (user and device data), History (chart of historical data), and Home (sensor data with IAQ).
   - Most Android devices run seamlessly due to the lightness of the design.

3. **Historical Data Storage and Visualization**

   - The graph-based history page allows for the comparison of dates and the identification of recurrent pollution patterns.
   - It also shows trends over time for IAQ data.
   - Encourages long-term thinking and well-informed decision-making.

4. **Location-Based Insights**

- Variations in IAQ levels observed across different environments:
  - Lab/Classroom: Moderate levels of VOC and PM due to poor ventilation and use; humidity varied depending on airflow or air conditioning, but temperature was relatively stable.
  - Industrial Area: There were elevated PM and VOC levels, along with increased temperatures induced by machinery and equipment; humidity levels tended to be lower.
  - Open Ground: Low and consistent PM and VOC levels showed the best air quality. The temperature and humidity were more realistic and consistent, in line with the ambient environment.
  - Homes: VOC levels rose when cooking or cleaning. Appliances, ventilation quality, and room insulation all contribute to temperature and humidity.
- Demonstrates how surroundings and human activity affect air quality.

5. **Encouragement for Health-Conscious Decisions**

- Provides customers with the option to use purifiers or ventilate rooms to enhance the quality of the air.
- Enhances the visibility of indoor pollution, which tends to be ignored.
- May help in the early identification of harmful air conditions, especially for sensitive groups (children, elderly, and asthmatics).

7.2 RESULT ANALYSIS AND VALIDATIONS

7.2.1 Result Analysis

The system was set up in a variety of settings, such as houses, open spaces, laboratories, classrooms, and industrial locations. The system's ability to identify environmental changes was validated by the different sensor readings obtained by each setting:

- Because of their small spaces and high occupancy, classrooms and labs have moderate levels of VOC and PM. The humidity changed with ventilation or air conditioning use, but the temperature stayed constant.

- High PM and VOC levels from emissions and machinery were observed in industrial areas, confirming sensor responsiveness in pollutant-heavy environments.
- With natural temperature and humidity variations, Open Grounds continuously delivered low PM and VOC measurements, reflecting cleaner ambient air.
- There were noticeable VOC increases in homes during cleaning and cooking tasks, as well as variations in humidity depending on the use of appliances (such as stoves and fans).

The pollution levels were accurately classified as Good, Moderate, or Poor using the app's IAQ calculating tool. By visualizing pollutant behaviour over time, the historical trends gave users useful information.

7.2.2 Validation and Evaluation

1. **Sensor Accuracy Evaluation**

   PM2.5/PM10 Readings:

   - Over the course of 48 hours, sensor readings (by PM2007) were verified against an air quality meter of commercial quality.
   - Variations stayed between ±5–7 µg/m³, which is suitable for low-cost sensor applications and shows excellent dependability in detecting pollution trends.

   Temperature & Humidity Accuracy:

   - The sensor precisely recorded changes in the ambient conditions, particularly when using the air conditioner and cooking.
   - DHT11 displayed consistent readings with variation margins of ±1.5°C and ±3% relative humidity.

   VOC Detection Reliability:

   - The ZP07 sensor validated trend accuracy but not absolute ppm values by successfully detecting relative changes in VOC content (such as spikes brought on by cleaning sprays or scents).

2. **Behavioral Pattern Validation**

   - PM levels significantly increased during high occupancy, cleaning, and dust-related activities, which is consistent with what is expected in confined areas.

- When using chemical aerosols, cooking oil fumes, and enclosed spaces with inadequate ventilation, VOC levels increased.
- The humidity rose while cooking and fell precipitously after using the air conditioner, confirming sensitivity to environmental cues.

3. System Stability & Performance

ThingSpeak API Integration:

- No data loss was observed during continuous operation over Wi-Fi.
- API requests responded within acceptable time limits, confirming stable cloud communication and retrieval.

Mobile Application Performance:

- IAQ calculations and visualizations on the Flutter app were consistently accurate across different tests.
- UI responsiveness and API syncs were tested across devices and yielded smooth, real-time feedback.

Firebase Data Verification:

- Firebase Realtime Database reliably stored historical IAQ data with user linkage.
- Authentication module consistently validated login/registration without delay or failure.

4. User Feedback & Usability

Usability Testing:

- Users found the UI clean and intuitive across all three pages (Home, History, Profile).
- Graphs helped users easily visualize and understand historical IAQ variations.

User Interpretation:

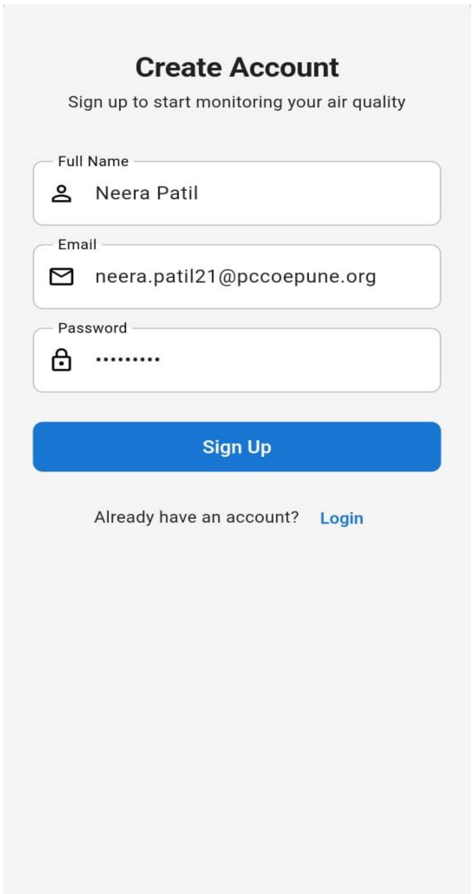
- IAQ categorization (Good, Moderate, Poor) helped users quickly assess air conditions and take corrective action.
- Feedback indicated that the Calculate IAQ feature was a useful and understandable tool.

5. Scalability & Extendibility

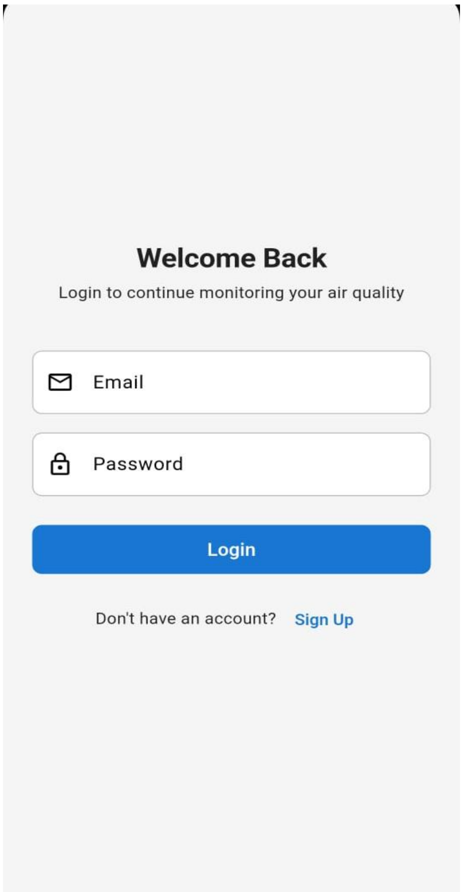
- System is modular and supports adding more sensors or locations with minimal code changes.
- Data handling via ThingSpeak and Firebase showed no performance degradation with prolonged testing.
- App is deployable on various Android devices, ensuring future compatibility and broader usability.

Overall, the system demonstrated consistent behaviour, and the sensor values correlated well with real-time indoor events, validating the effectiveness of the implementation.

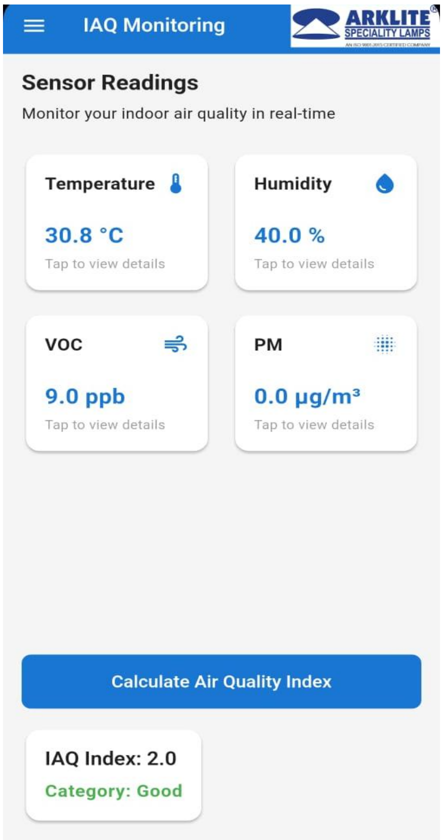
7.3 SCREENSHOTS



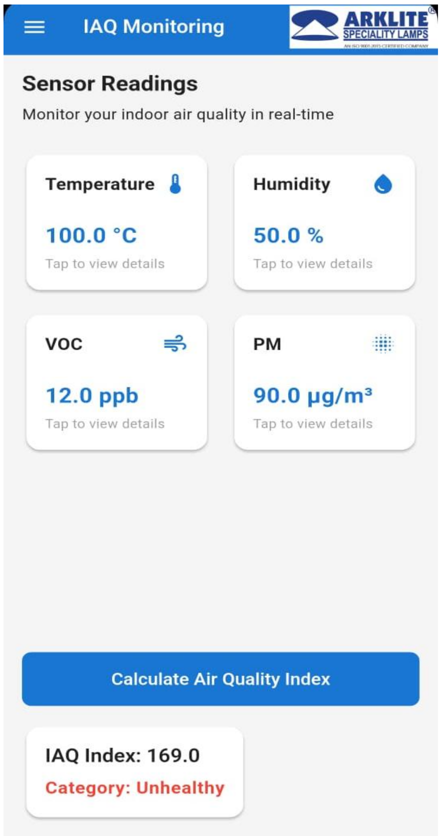
Sign up page



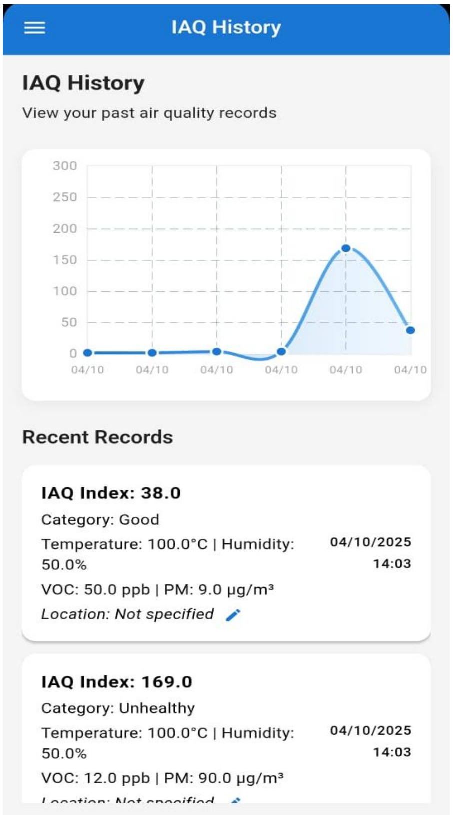
Log in page



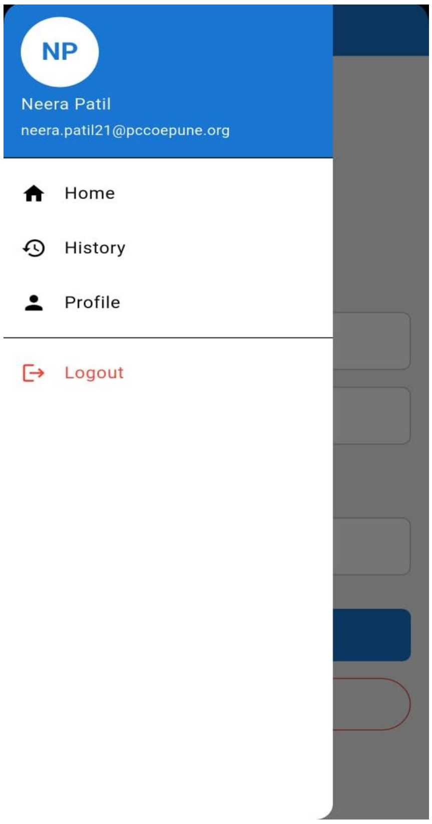
Home Page (Good IAQ)



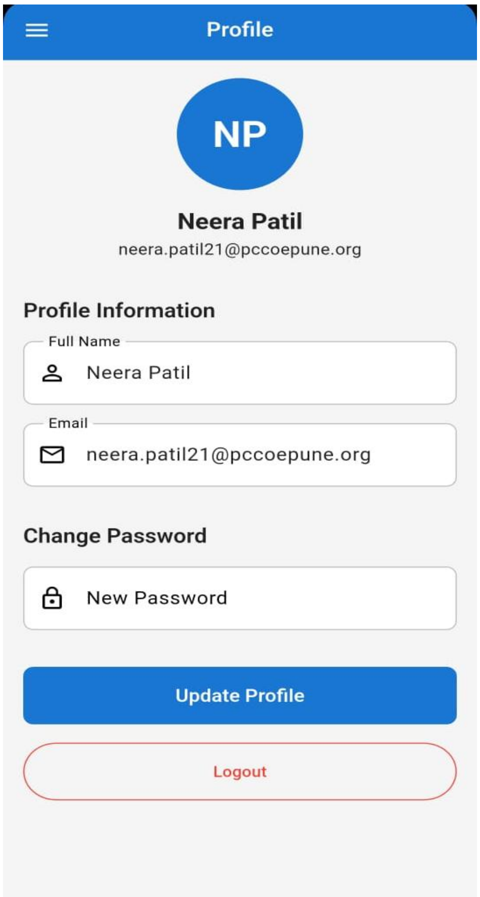
Home Page(Unhealthy IAQ)



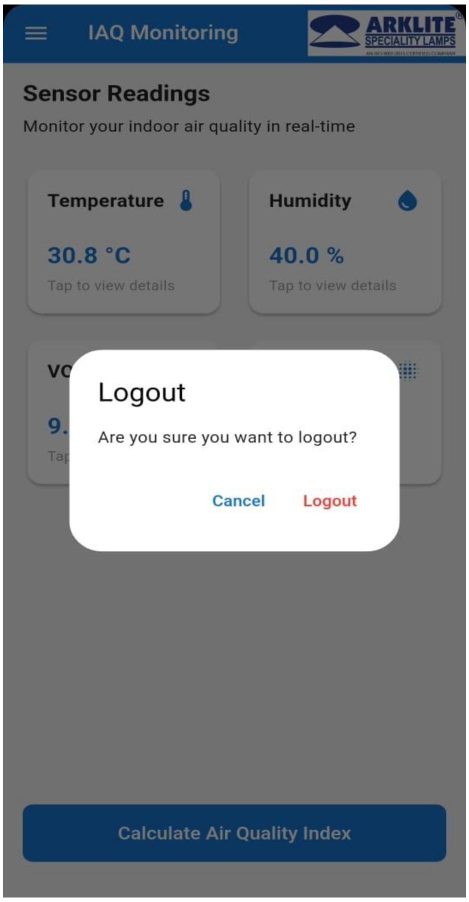
History Page



Sidebar



Profile Page



Logout

8. CONTRIBUTION TO SUSTAINABLE DEVELOPMENT GOALS

8.1 INTRODUCTION TO SDGs

The United Nations created the 17 Sustainable Development Goals (SDGs) in 2015 as a component of the 2030 Agenda for Sustainable Development. The most pressing global issues, including poverty, health, climate change, education, innovation, and environmental sustainability, are intended to be addressed by these aims. Governments, businesses, organizations, and individuals may work together to create a more resilient, just, and equitable future by using the SDGs as a common framework.

Focusing on striking a balance between environmental preservation, economic growth, and social advancement, each objective is interrelated. Sustainable engineering and academic initiatives are essential to achieving these objectives because they apply creative and approachable solutions to pressing issues.

This project, which focuses on an inexpensive, Internet of Things-based indoor air quality monitoring system, most directly supports several SDGs:

- SDG 3: Good Health and Well-being – by raising awareness of air pollution and enabling timely action to reduce exposure to harmful indoor pollutants.

- SDG 9: Industry, Innovation, and Infrastructure – through the use of smart environmental sensors, cloud integration, and mobile-based data visualization for health-conscious infrastructure.

- SDG 11: Sustainable Cities and Communities – by supporting air quality tracking in homes, schools, and industries, encouraging cleaner and safer indoor environments.

The project supports environmental awareness, health-driven innovation, and accessible technology, all of which contribute to the SDGs' global objective.

8.2 MAPPING OF THE PROJECT TO RELEVANT SDGs

Through the use of IoT, cloud computing, and mobile applications, this project directly contributes to the Sustainable Development Goals (SDGs) of the UN by monitoring indoor air quality and fostering healthier surroundings.

The following are the main objectives addressed:

1. **SDG 3: Good Health and Well-Being**
   - In order to assist users in identifying and minimizing exposure to hazardous air conditions, the system tracks pollutants such as PM2.5, VOCs, temperature, and humidity.
   - By allowing people, schools, and businesses to take preventive measures, real-time IAQ testing lowers the risk of allergies, respiratory disorders, and long-term health difficulties.
   - By graphically classifying IAQ (Good, Moderate, Poor) for ease of comprehension, the software promotes health awareness.

2. **SDG 9: Industry, Innovation, and Infrastructure**
   - An inexpensive and scalable smart environment solution is demonstrated by the combination of inexpensive sensors (DHT11, ZP07, and PM2007) with an ESP8266 microcontroller and a Flutter-based mobile application.
   - Promotes creativity in environmental monitoring for homes and businesses alike.
   - Cloud-based platforms (Firebase, ThingSpeak) encourage efficient, accessible, and remotely managed, lightweight infrastructure.

3. **SDG 11: Sustainable Cities and Communities**
   - Allows people and organizations to monitor and maintain healthy indoor air quality, which encourages sustainable living.
   - Uses real-time air monitoring in homes, businesses, and schools to help urban populations become more resilient.
   - Promotes data-driven decision-making for improved human health and building management.

8.3 PROJECT IMPACT ASSESSMENT

Public health, digital innovation, and environmental awareness are all significantly impacted by the creation and deployment of a real-time Indoor Air Quality (IAQ) Monitoring System that uses inexpensive IoT sensors, cloud integration (ThingSpeak & Firebase), and a Flutter-based mobile application. In addition to producing indirect societal and educational effects, the system provides quantifiable short- and long-term benefits.

1. Short-Term Impact

- Real-time environmental monitoring helps consumers take prompt action to improve indoor conditions by enabling the instant detection of poor air quality (such as excessive PM or VOC levels).
- Enhanced Awareness of Users provides a categorized IAQ index (Good, Moderate, Poor) through an easy-to-use smartphone interface, educating users about pollution levels and environmental health.
- The solution, which was developed with open-source tools and reasonably priced hardware (DHT11, ZP07, PM2007, ESP8266), is scalable and reasonably priced for small businesses, homes, and educational institutions.
- Data Visualization and Logging gives consumers a systematic historical perspective of air quality trends, assisting them in making well-informed decisions about infrastructure or lifestyle modifications (such as better ventilation).

2. Long-Term Impact

- Encouragement of Intelligent Health Facilities enhances public well-being by promoting the use of smart indoor spaces in companies, educational institutions, and urban regions.

- Long-term use of the system can reduce the incidence of chronic respiratory disorders and create better indoor environments, particularly for vulnerable populations including the elderly, children, and asthmatics.
- Future enhancements like additional sensors, predictive analytics, or smart HVAC integration are supported by the cloud-integrated and flexible architecture.
- Institutions or municipalities can utilize long-term data to evaluate trends in air quality and make plans for policy or infrastructure upgrades.

3. Indirect Impact

- Support for Sustainable Urban Living (SDG 11) makes communities and residential areas more sustainable and livable by assisting in the monitoring and enhancement of indoor settings.
- Empowerment through Technology (SDG 9) exemplifies how to apply intelligent, networked technology to address practical environmental issues while encouraging creativity and digital transformation.
- Educational and Research Value Inspires future ideas by serving as a useful teaching tool for educators and students studying subjects like embedded systems, environmental science, and the Internet of Things.
- Health & Social Equity (SDG 3) makes air quality monitoring accessible and affordable, particularly in high-risk or low-income areas where conventional monitoring methods are not available.

9. CONCLUSION AND FUTURE SCOPE

9.1 CONCLUSIONS

A cost-effective, real-time air quality monitoring solution, the IoT-Based Handheld Environmental and Air Quality Monitoring Station tracks temperature, humidity, volatile organic compounds (VOCs), and particulate matter (PM2.5 & PM10) by integrating multiple environmental sensors with ESP8266 Wi-Fi connectivity. The system sends sensor data to ThingSpeak for cloud storage, and a Flutter-based mobile app allows users to view both historical and real-time data. Secure user access and data storage are guaranteed using Firebase authentication. By successfully offering practical insights on indoor air quality, this project enables people and organizations to take preventative action against poor air quality. The system provides a scalable platform for future enhancements while improving accessibility and usability through the utilization of cloud computing and the Internet of Things.

Furthermore, the system's ability to adjust to diverse indoor circumstances was confirmed by its consistent sensor performance across a range of contexts, including residences, classrooms, and industrial sites. The presentation of the IAQ index and historical patterns made it simple for consumers to understand pollution levels and make wise choices. The system may be expanded to add features like automated alarms, multi-location tracking, or interaction with smart ventilation systems because to its modular architecture and generally accessible components. All things considered, the project effectively combines cloud-based data management, mobile technologies, and environmental sensing to produce a dependable and approachable IAQ monitoring system.

9.2 FUTURE SCOPE

1. AI-Driven Air Quality Prediction

- Implement machine learning algorithms to analyze historical data and predict future air quality trends.
- Provide personalized recommendations on improving indoor air quality based on data-driven insights.

- Enable automated alerts to warn users when air quality drops below a safe level.

2. Smart Home and IoT Integration

- Integrate with smart home systems such as air purifiers, HVAC (Heating, Ventilation, and Air Conditioning) systems, and smart windows for automated air quality improvement.
- Enable compatibility with Google Assistant, Alexa, or Apple HomeKit for voice control and automation.

3. Expansion of Sensor Network

- Incorporate CO_2, NO_x (Nitrogen Oxides), Ozone (O_3), and Carbon Monoxide (CO) sensors for a more comprehensive air quality assessment.
- Improve sensor accuracy and calibration to meet industry standards for environmental monitoring.
- Support outdoor air quality monitoring with additional weather-resistant components.

4. Enhanced Mobile Application Features

- Introduce an interactive dashboard with real-time air quality visualization, historical trends, and detailed analytics.
- Develop a community-driven air quality monitoring network, allowing users to share air quality data from different locations.
- Implement multi-device support to enable monitoring across various platforms such as iOS, Windows, and web applications.

5. Industrial and Smart City Applications

- Deploy in factories, warehouses, and industrial plants to monitor air pollution levels and comply with environmental regulations.
- Collaborate with municipal corporations to build a city-wide IoT air monitoring network that integrates data from multiple sensors across different locations.

9.3 APPLICATIONS

1. Indoor Air Quality Monitoring

- Helps households, offices, and educational institutions maintain healthy indoor air conditions.
- Supports individuals with asthma, allergies, or other respiratory diseases by providing real-time air quality insights.
- Can be used in hospitals, gyms, shopping malls to ensure proper ventilation and air safety.

2. Industrial & Workplace Safety

- Used in factories, chemical plants, and construction sites to monitor toxic gases and particulate matter.
- Helps companies comply with Occupational Safety and Health Administration (OSHA) regulations by tracking air quality.
- Enables hazardous material detection to prevent workplace accidents due to air contamination.

3. Smart Cities & Environmental Monitoring

- Deployed in urban areas to track pollution levels, helping policymakers implement better air quality control measures.
- Supports government agencies and environmental organizations in climate research and pollution tracking.
- Can be integrated into public transportation systems (buses, metro stations) to assess air quality exposure in different locations.

4. Healthcare & Medical Research

- Assists healthcare professionals in studying respiratory diseases linked to poor air quality.
- Provides data for epidemiological research on air pollution’s impact on cardiovascular and lung diseases.

- Can be used by pharmaceutical companies to study air quality effects on certain medications.

5. Disaster Management & Emergency Response

- Helps detect toxic gas leaks, wildfires, or chemical hazards in emergency situations.
- Supports first responders, firefighters, and disaster management teams with real-time air quality updates.
- Can be used in mines, tunnels, and other enclosed environments where air quality is critical for survival.

Appendix A

Paper Publication Details

Submissions   Contact Chairs   Help Center   Select Your Role : **Author**   ICCUBEA2025   shravani Chidrewar

Submission Summary

Print   Email

Conference Name	9th International Conference on Control Communication, Computing and Automation
Track Name	Internet of Things and Computer Networks
Paper ID	545
Paper Title	IOT Based Indoor Air Quality
Abstract	Indoor air quality (IAQ) significantly impacts health, safety, and productivity in enclosed spaces. Rising air pollution and its associated health risks highlight the need for affordable, continuous IAQ monitoring. This work proposes a low-cost, real-time IAQ monitoring system using a NodeMCU (ESP8266) microcontroller integrated with sensors to measure temperature, humidity, VOCs, and particulate matter. Data is transmitted to the ThingSpeak cloud platform and visualized through a Flutter-based mobile application, which also computes a user-friendly IAQ index. Firebase is utilized for secure authentication and historical data storage, enabling both real-time insights and trend analysis. The system leverages IoT, cloud, and mobile technologies to enable remote monitoring and alerts when pollutant levels exceed safe thresholds—offering a scalable, user-centric solution for healthier indoor environments. Keywords: Indoor Air Quality, IoT Monitoring, VOCs, Particulate Matter, Cloud Storage, ThingSpeak, Mobile App, NodeMCU
Created	4/19/2025, 11:29:21 AM
Last Modified	4/19/2025, 11:29:21 AM
Authors	**shravani Chidrewar** (PCCOE) <shravanichidrewar01@gmail.com>
Submission Files	IAQ RESEARCH PAPER(SHRAVANI NEERA).pdf (747.2 Kb, 4/19/2025, 11:29:15 AM)

9th International Conference on Control Communication, Computing and Automation : Submission (545) has been created. Inbox

Microsoft CMT <noreply@msr-cmt.org>
to me

11:29 (1 minute ago)

Hello,

The following submission has been created.

Track Name: Internet of Things and Computer Networks

Paper ID: 545

Paper Title: IOT Based Indoor Air Quality

Abstract:
Indoor air quality (IAQ) significantly impacts health, safety, and productivity in enclosed spaces. Rising air pollution and its associated health risks highlight the need for affordable, continuous IAQ monitoring. This work proposes a low-cost, real-time IAQ monitoring system using a NodeMCU (ESP8266) microcontroller integrated with sensors to measure temperature, humidity, VOCs, and particulate matter. Data is transmitted to the ThingSpeak cloud platform and visualized through a Flutter-based mobile application, which also computes a user-friendly IAQ index. Firebase is utilized for secure authentication and historical data storage, enabling both real-time insights and trend analysis. The system leverages IoT, cloud, and mobile technologies to enable remote monitoring and alerts when pollutant levels exceed safe thresholds—offering a scalable, user-centric solution for healthier indoor environments.
Keywords: Indoor Air Quality, IoT Monitoring, VOCs, Particulate Matter, Cloud Storage, ThingSpeak, Mobile App, NodeMCU

Created on: Sat, 19 Apr 2025 05:59:21 GMT

Last Modified: Sat, 19 Apr 2025 05:59:21 GMT

Authors:
- shravanichidrewar01@gmail.com (Primary)

Secondary Subject Areas: Not Entered

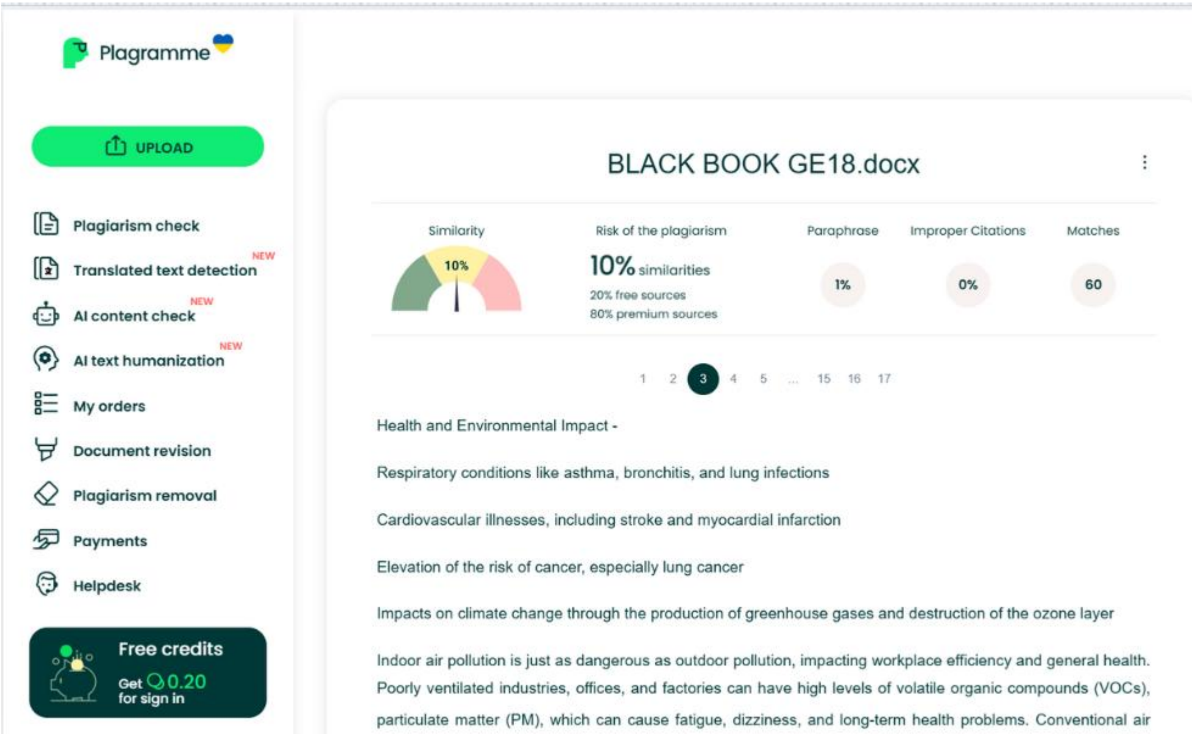
Submission Files:
IAQ RESEARCH PAPER(SHRAVANI NEERA).pdf (747 Kb, Sat, 19 Apr 2025 05:59:15 GMT)

Submission Questions Response: Not Entered

Thanks,
CMT team.

Appendix B

Plagiarism Report



REFERENCES

[1] M. N. M. Aashiq, W. T. C. C. Kurera, M. G. S. P. Thilekaratne, A. M. A. Saja, M. R. M. Rouzin, N. Neranjan, and H. Yassin, "An IoT Based Handheld Environmental and Air Quality Monitoring Station," *Acta IMEKO*, vol. 12, no. 3, pp. 1–8, Sep. 2023.

[2] W.-T. Sung and S.-J. Hsiao, "Building an Indoor Air Quality Monitoring System Based on the Architecture of the Internet of Things," *Journal of Wireless Communications and Networking*, vol. 2021, no. 153, 2021.

[3] H. Othman, R. Azari, and T. Guimarães, "Low-Cost IoT-Based Indoor Air Quality Monitoring," *Technology | Architecture + Design*, vol. 8, no. 2, pp. 250–270, 2024.

[4] J. M. S. Waworundeng and W. H. Limbong, "AirQMon: Indoor Air Quality Monitoring System Based on Microcontroller, Android, and IoT," *Cogito Smart Journal*, vol. 6, no. 2, pp. 251–261, Dec. 2020.

[5] J. Jo, B. Jo, J. Kim, S. Kim, and W. Han, "Development of an IoT-Based Indoor Air Quality Monitoring Platform," *Journal of Sensors*, vol. 2020, pp. 1–14, Jan. 2020.

[6] M. W. P. Maduranga, K. N. M. H. H. Kosgahakumbura, and G. H. C. J. Karunarathna, "Design of an IoT Based Indoor Air Quality Monitoring System," Proceedings of the National Engineering Research Symposium, vol. 1, pp. 117–122, 2020.

[7] J. Saini, M. Dutta, and G. Marques, "Indoor Air Quality Monitoring Systems Based on Internet of Things: A Systematic Review," *International Journal of Environmental Research and Public Health*, vol. 17, no. 14, pp. 4942, Jul. 2020. https://doi.org/10.3390/ijerph17144942.

[8] L. Zhao, Y. Yang, and Z. Wu, "Review of Communication Technology in Indoor Air Quality Monitoring System and Challenges," *Proceedings of the International Conference on Communication Technology*, vol. 1, pp. 1–6, 2020.

[9] A. Gecali, D. V. Origines Jr., and N. P. Sobejana, "Android Based Air Quality Monitoring System," *Proceedings of the International Conference on Information Technology and Engineering Applications*, vol. 1, pp. 1–6, 2020.