International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

IJISAE

ISSN:2147-6799 www.ijisae.org Original Research Paper

Air Quality Monitoring System Development using IoT for Indoor Applications

Aneessha Acharya K[1], Sucheta V. Kolekar*[2]

Submitted: 01/11/2023 **Revised**: 20/12/2023 **Accepted**: 02/01/2024

Abstract: The third sustainable goal defined by the United Nations is about good health and well-being, in which air quality is a major concern in today's environment. The presence of harmful, if not noxious gases, degrades the quality of air present in our surroundings. The presence of this particulate matter affects the overall health of many individuals. Especially with the rise of COVID-19 and its variants in recent years, even breathing normally may become an arduous task for an individual. The pollutants in the air may facilitate worsening conditions for an affected individual. Hence, the present work aims to develop an indoor air quality monitoring system that alerts the user when the air quality worsens into harmful ranges such as good, moderate, and unhealthy. The present work utilizes multiple sensors to take readings of different environmental pollutants multiple times a day. These readings are uploaded to a cloud database. A mobile application was made to view these readings. The mobile application interfaces with the cloud database and gets the latest sensor data, which is viewable on the main screen. The application also visualizes the data collected in the past 24 hours. The readings for each day were aggregated, and the average was taken to be converted into an AQI index based on the US EPA standard. This was then compared against AQI data for the day provided by the Delhi Pollution Control Committee. The readings were in a suitable accuracy range for the present work, thus validating the monitoring system design. The analysis and alert time were considered for validating the application design and data analysis.

Keywords: *COVID-19, Internet of Things, Indoor Air Quality, Sensors, Cloud Computing, Good health and well-being.*

1. Introduction

In today's environment, pollution is rampant and ever-present in almost all-natural resources. Air quality has suffered due to particulate matter and gaseous compounds such as PM2.5, PM10, COx, NOx, etc. PM2.5 is generated through fuel burning or chemical reactions [1]. It can irritate the eyes, nose, and throat, causing coughing and shortness of breath with short-term exposure. In contrast, prolonged exposure can lead to permanent respiratory problems such as asthma, chronic bronchitis, and heart disease [2]. Depending on CO concentration and length of exposure, mild to severe poisoning may occur. Headache, dizziness, weakness, nausea, vomiting, and finally, loss of consciousness are symptoms of Carbon Monoxide poisoning [3]. Nitrogen oxides, emitted from motor engines, are deep lung irritants that can induce pulmonary edema if inhaled at high levels [4]. As such, these pollutants pose major risks to an individual's health and safety, in both short-term and long-term exposures.

A few years back, the rise of COVID-19 has caused respiratory distress in inflicted individuals, and pollutants may increase the mortality rate for affected individuals [5]. The Air Quality Monitoring System aims to act as an early warning system to warn of unhealthy air in an indoor space [6]. By using particulate matter sensors CO, and NO2 concentrations can be monitored in the room, and these readings alert the user of harmful levels of gases in the environment.

The Internet of Things (IoT) is a network of physical objects embedded with sensors, communication devices, and other technologies that communicate with other devices on the Internet to exchange data [7, 8]. Cloud computing provides a way of utilizing the resources of a remote system hosted on the internet rather than building and maintaining infrastructure for storage and computing for every work [9]. Cloud computing enables low-power IoT devices to focus on collecting data while the analysis is done on a remote system.

Previous implementations of Air Quality Monitoring systems in indoor spaces lacked the functionality of easily viewing previous sensor readings, measuring certain pollutant concentrations such as PM2.5 or PM10, or utilizing expensive sensors. The present work is unique as it bundles the major pollutant concentrations using relatively less expensive sensors, uploads them to the FireStore database, and provides a mobile application to view the visualizations of concentrations, and latest sensor readings, as well as provides an alert system in the case of harmful concentration levels.

[1] *Department of Instrumentation and Control Engineering, Manipal Academy of Higher Education, Manipal, India-576104*
ORCID ID : 0000-0001-8644-8882
[2] *Department of Information and Communication Technology, Manipal Academy of Higher Education, Manipal, India-576104*
ORCID ID: 0000-0003-2642-6088
** Corresponding Author Email: sucheta.kolekar@manipal.edu*

The present work includes developing an Air quality monitoring module that senses various environmental pollutants. These readings are stored in a cloud database; the present work involves developing a mobile application that provides a way to view sensor data. This may be used by users with temporary or chronic respiratory distress and people in environments that tend to be highly polluted to their benefit by getting alerted of harmful concentrations as soon as possible. The research's objectives include designing a portable device for indoor air quality monitoring using an MQ-135 gas sensor, DSM501A PM sensor, MICS-6814 gas sensor module, and Gas grove sensor to capture the data and store it in the cloud. A design and development of a mobile application for analyzing the data and generating visualizations and an alert system is also developed.

The rest of the paper is organized as follows. Section 2 presents the background information, section 3 describes the methodology, section 4 explains the implementation followed by the results in section 5. Finally, Section 6 summarizes and concludes the paper.

2. Background

Indexes provide a simple measure of air quality, considering all factors and weights to determine a single value to quantify air quality. Currently, no official government indexes exist for indoor air quality. However, researchers have researched and developed some [10, 11]. There are also no internationally agreed upon AQI indices available [12]. These can be used to generate alerts when air quality drops to poor, or harmful levels. Table 1 shows the different harmful concentrations of different pollutants. The averaging period is the time over which the pollutant concentration is considered for calculating the AQI of the pollutant [13].

Table 1. Harmful concentrations and averaging periods of major pollutants

Pollutant	***Averaging Period***	***Unhealthy Concentrations***
PM2.5	24 hours	35 μg/m^3
CO	8 hours	9 ppm
NO_2	1 hour	10 ppb
PM10	24 hours	150 μg/m^3

According to multiple sources, the major pollutants present in Delhi are PM2.5, PM10, NO_2, and CO [14-17]. The concentration of these pollutants has been pushing the air quality into hazardous or unhealthy levels for both sensitive and normal people. Hence, the ever-present need for indoor air quality monitoring systems and air purifiers is increasing. Tigor Hamonangan Nasution et al. [18] give the design of an IAQ monitoring system, that uses a screen to display temperature, humidity, and concentrations of NO_2, NH3, CO, and dust levels using a GP2y1010AU0F dust sensor, BME280 sensor module, MQ-136 sensor module, MICS6814 sensor module, and an ESP32 microcontroller board. The data is sent to a cloud where visualizations can be done. Edoardo Conticini et al. conducted a study in Northern Italy regarding the correlation between the high SARS-COVID lethality and atmospheric pollution. The study shows that a person living in a region with higher pollution levels is prone to develop chronic respiratory illnesses and thus is suitable to any infective agent [5]. Prolonged exposure also increases the risk of respiratory inflammation and distress. This has led to a higher mortality rate for COVID-19 in Northern Italy [19, 20]. Sean McGrath et al. [21] have suggested implementing a personal air quality monitoring system that users can carry without inconvenience. The system measures values of CO, Ozone, and NO_2. It uploads them to a cloud database while utilizing low-power sensor modules and microcontrollers to limit battery consumption in the mobile system. M.A. Mokar et al. [22, 23] detail using Firebase Cloud messaging for sending alerts and instructions to a mobile application on a smartphone in the form of data messages. These messages can be sent to a specific mobile device or a group sharing a token. This provides a way to manage the functions of an application on multiple devices at the same time. This can be utilized to send alerts in our present work. Siavash Esfahani et al. [11] propose a low-cost, portable Internet of Things (IoT) Indoor Air Quality (IAQ) monitoring system. The system has 30 hours of battery life to monitor VOCs, CO2, PM2.5, and PM10. The system utilizes BME680 temperature and humidity sensor, CCS811 (AMS) to capture the concentration of VOCs, a Sensiron SCD30 Carbon Dioxide sensor, and an ESP32 micro-controller to capture readings from the environment and send them to a cloud server. The authors propose that this system would be viable as a node in a smart city for monitoring pollution concentrations [24].

3. System Design and Methodology

The system's design is split into three parts: the monitoring system hardware, the cloud storage service, and the mobile application. These three interfaces with each other to form the Air Quality Monitoring System.

A. Monitoring System

The major pollutants affecting the health of individuals are PM2.5, PM10, CO, and NO_2. The air quality monitoring system has to take readings for the concentrations of these pollutants. Hence, the microcontroller takes input from sensors to take readings for dust levels (PM2.5, PM10), and concentrations of CO and NO_2 in the atmosphere.

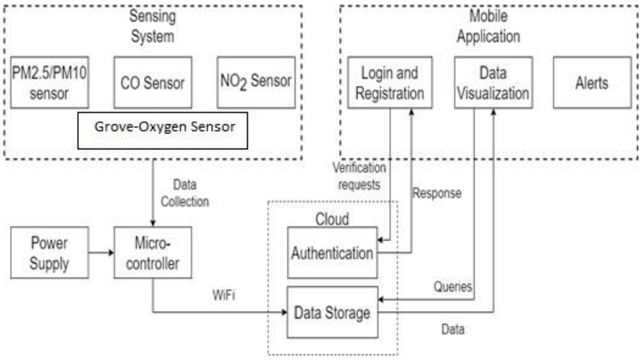

Fig. 1. The basic design of the Air quality monitoring system

The system is connected to a rechargeable power supply. Figure 1 shows the basic design of the monitoring system. The sensor readings that are read by the microcontroller are converted to concentrations for each pollutant. After conversion, they are sent to the cloud database for storage.

B. Cloud storage

The data read by the system has to be secure, and easily accessible to be of use. Hence, we require fast response times, preventive measures against data corruption, and restricted access to the database. A cloud service is chosen because it provides fast, reliable access from anywhere, guarantees server availability, and authentication services for users. As well as notifications and updates can be pushed to the user as and when required.

C. Mobile Application

A clean and simple representation of data is required to identify patterns, recognize risks, and take action on warnings, to this end a mobile application is developed that can interface with the cloud database and query the data. The application can be installed on the user's device. The user can register for the application using an email and a password. Access to the application is limited until the user verifies their email. This application enables visualizations of the recent data through graphs and user-interface displays. These graphs enable users to identify trends in pollutant concentrations concerning time. This is useful to identify if there is a certain cause of rising pollution levels at certain times of the day, or if the pollution levels have been constant, rising, or declining. The application also notifies the user if and when the pollutant concentrations reach harmful levels as defined by the AQI.

D. Materials

The materials used in the Air Quality Monitoring system are as follows:

- Raspberry Pi 4 microcontroller
- MICS6814 sensor module
- SM501A PM2.5 sensor module
- MCP3008 analog to digital converter

Raspberry Pi

In this work, we use a Raspberry Pi 4 microcontroller board that supports wireless and Bluetooth connections. The Raspberry Pi 4 has a 64-bit 1.5GHz quad-core Cortex-A72 processor and a USB C power port. The board is depicted in Figure 2, which has 40 GPIO pins with backward compatibility support with older boards. The microcontroller board can be programmed using Python. The different sensors and AQI conversions are supported using third-party libraries [25].

Fig. 2. Raspberry Pi 4

MICS-6814 Gas Sensor

The next component that enables reading pollutant concentrations is a MICS- 6814 sensor module to measure carbon monoxide levels and nitrogen dioxide levels. The MICS-6814 gas sensor is a Metal Oxide Sensor. The chip structure of MICS-6814 consists of a precision mechanical diaphragm and an embedded heating resistor with the sensing layer at the top as shown in Figure 3 [18]. It is a robust MEMS sensor for harsh environments There are 3 separate sensing elements integrated in the module. Three sensor chips comprise the MICS-6814, each with independent heaters and sensitive layers. The different sensor chips are used to read different gas concentrations. One sensor chip detects the presence of oxidizing gases (OX), the other sensor detects the presence of reducing gases (RED) and the last one detects NH3. The module has 5 pins, one for input voltage(5V), one each for each sensor output, and one is grounded [26].

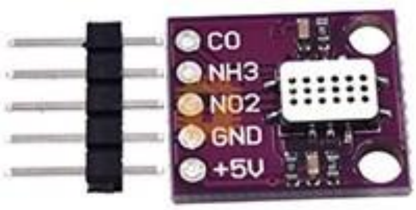

Fig. 3. MICS-6814 gas sensor module

DSM501A Dust Sensor

The next component used in the present work is a DSM501A PM dust sensor with dual outputs that can simultaneously detect both PM2.5 and PM10 concentrations. It is a low-cost LED sensor that detects particles as small as 1 /µm. The sensor must be kept from artificial airflow as it utilizes a heater to implement air intake in the sensing areas [27]. The image of the sensor is represented in Figure 4.

Fig. 4. DSM501A dust sensor

MCP3008 Analog to Digital Converter

The MCP3008 analog-to-digital converter converts analog outputs given by the sensors to digital inputs that the Raspberry Pi can read. The pin diagram for the IC can be seen in Figure 5. The ADC has 16 pins, where pins 1 to 8 are for input channels 0 to 7. Pins 9 and 14 are grounded, and pins 15, and 16 are for reference voltage for the conversion. Pin 10 enables or shuts the IC, and the other pins are used for SPI communication with the Raspberry Pi [28].

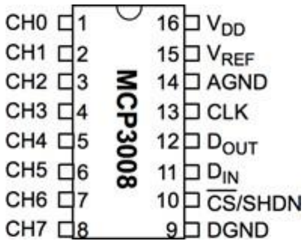

Fig. 5. Pin Diagram for MCP-3008

E. Circuit Design

The circuit design shows how the different components are connected and provides a guideline to form connections between the sensor modules and the Raspberry Pi microcontroller. The sensor modules require an input voltage of 5V, while the ADC requires a reference voltage of 3.3V to convert the readings. DSM501A and MICS-6814 require a 5V input voltage, so each has its Vcc connected to Pin 2 of the Raspberry Pi. The GND pins are connected to Pin 6 on the Raspberry Pi as it serves as a GND pin. The output pins for both sensor modules are connected to the input channel pins on the MCP3008 ADC IC. Pins 18, 23, 24, and 25 on the Raspberry Pi are connected to the ADC for SPI communication. The ADC requires a reference voltage for converting analog values to digital values, which is given by Pin 1 on the Raspberry Pi [28, 29].

The Circuit shown in Figure 6 connects the Raspberry Pi with the sensor modules with a MCP3008 ADC to convert analog values read by the sensor to digital values supported by Raspberry Pi. The Pi uses it’s inbuilt Wi-Fi module to connect to a router and transfer data to the cloud storage.

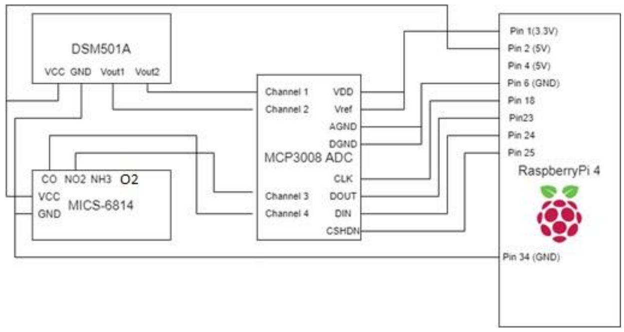

Fig. 6. Circuit diagram for the Air quality Sensor

F. Reading Sensor Data

After the circuit is prepared, the readings must be taken from the GPIO pins of the Raspberry Pi. The ADC has 4 channels in use and the output from the sensors is multiplexed, the Raspberry Pi can read each input separately and then convert each input into its corresponding concentration values. These values are then collected and sent to the cloud database for storage.

Conversion to Concentrations

The MICS-6814 sensor module returns readings in terms of resistance in kOhms. These values must be converted to concentration values in ppm, or ppb [30]. For MICS-6814 sensor the formulae for calculating different values are represented in equations 1 and 2.

$$CO_{ppm} = 10^{(-0.25*\log10(Rs/R0)+0.64)} \quad (1)$$

$$NO_2 = 10^{(\log10(Rs/R0) - 0.8129)} \quad (2)$$

where COppm is the concentration of CO, in ppm NO_2 is concentration of NO_2, in ppm, Rs is Sensor reading, R0 is Sensor reading in baseline air

The DSM501A dust sensor can be used to measure the dust levels, however it uses the PWM system, hence the values can be read by measuring the ratio between time spent in high state to the time spent in a low state, then using the datasheet, the for-mula for conversion to µg/m3 is given by equation 3:

$$PMx = 1.1*R3 - 3.8*R2+520*R+0.62 \quad (3)$$

where PMx is Concentration of Particulate Matter in µg/m3 R is ratio of time spent in high state to time spent in low state.

G. Firebase

Firebase is a google developed platform that is used for creating web and mobile applications. It provides various interfaces to help streamline back-end development and access for an application and many of these have been used for the present work. It has features that enable tracking analytics, reporting, and fixing crashes. The tools provided by Firebase enable developers to focus on building the application instead of man-aging back-end operations [31, 32]. The tools are all hosted int the cloud and are easily scalable with little to no effort. Client SDKs are provided by Firebase to interact directly with these tools and require no middleware. Algorithm 1 represents the gathering the data and uploading then to the cloud database.

H. Firebase Authentication

Firebase provides an authentication service that can register users with various different login methods and authenticate them when they try to log in to the application [33]. Firebase authentication SDK allows the developer to set up sign in methods, create their own Sign-in UI flows and pass the user's credentials to the Firebase SDK for registration as well as verification. Firebase provides an in-built user email verification system that can be used to send an email to verify a user. As shown in Figure 7, the front end takes user credentials from the user, and passes these credentials to Firebase Authentication SDK, Firebase backend services verify the credentials and send a response back to the application. The developer may choose what to do with the response as per the application.

Algorithm 1 Getting sensor readings

procedure GETTING SENSOR READINGS( )
 Import Libraries
 LowPulsePM 25 = 0
 LowPulsePM 10 = 0
 sampleTime = 5 X 60 X 1000 true
 Sleep;
 PM25 reading = pulse width on channel
 PM10 reading = pulse width on channel
 NO_2 reading = voltage input from channel
 CO reading = voltage input from channel 4
 $LowPulsePM25 = LowPulsePM25 + PM25reading$
 $LowPulsePM10 = LowPulsePM10 + PM10reading$
 $ratio = \frac{LowPulsePM25}{sampleTime*10}$
 $PM25 = 1.1 * ratio^3 - 3.8 * ratio^2 + 520 * ratio + 0.62$
 $ratio = \frac{LowPulsePM10}{sampleTime*10}$
 $PM10 = 1.1 * ratio^3 - 3.8 * ratio^2 + 520 * ratio + 0.62$
 $NO_2 = \frac{NO_2reading*20000}{6.5}$
 $CO = 10^{\frac{log_{10}(COreading*150000/3.5)}{-0.845}}$
 Put readings in a Dictionary Upload to cloud
end procedure

I. Firebase Firestore

Firebase Firestore is a real-time, cloud based, NoSQL database [34]. It is a document-oriented database where instead of tables and rows, data is stored in the form of collections of documents. Multiple key-value pairs are used to store data in each document. Firestore is optimized to manage large collections of small documents [35]. Documents support many different kinds of data types such as strings, numbers, dates, etc. One can create sub-collections within documents and build hierarchical structures as the database grows. The data can be retrieved by simple queries at the document level and filters through the application and real time listeners can be set on the database to retrieve data as soon as a new record is added in the form of a database snapshot with only the recently added data.

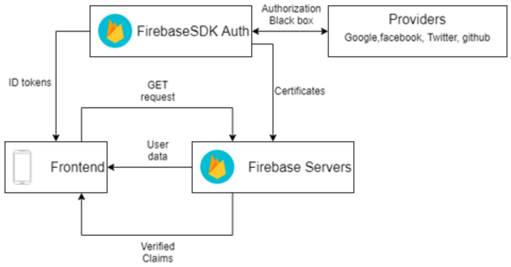

Fig. 7. Authentication using Firebase

J. Software Design

An android mobile application is created to allow the user to access the sensor readings. The application requires basic security to stop malicious or unwanted access to the data [36]. To this end, Firebase's Authentication SDK can be utilized for authenticating users trying to login to the application. If a user doesn't have an account, the authentication SDK can be used to make an account for the user. To avoid someone entering a non-existent email, or an email that isn't their own, an email can be sent to the user to verify that the email provided belongs to them. Once the user is verified, they are now free to log in and use the application.

The data stored on the cloud database, in this case Firestore, has to be retrieved and visualized for further analysis. The data retrieval must be quick to improve user experience as well as handle multiple requests at once. The application uses the Fire-based Firestore SDK to retrieve the most recent data documents from the NoSQL database in the form of a query snapshot. This data can then be represented into a line graph visualization for the benefit of the user [37].

The application requires a quick response to data indicating harmful levels of pollutant concentrations, as in extreme cases, these might lead to fatal scenarios, this requires a means of analyzing the data as soon as it is updated, and generating an alert after checking its values. Firebase's Cloud messaging service and cloud functions can have used to generate functions that are triggered on updates to the database, which then send alerts and notifications if a pollutant's concentration is above a certain threshold and reaching harmful levels. These alerts serve as a warning system to the user.

4. Results

T Various results of subsystems and comparative analysis of capture air quality values are discussed in this section. The overall setup is depicted in Figure 8.

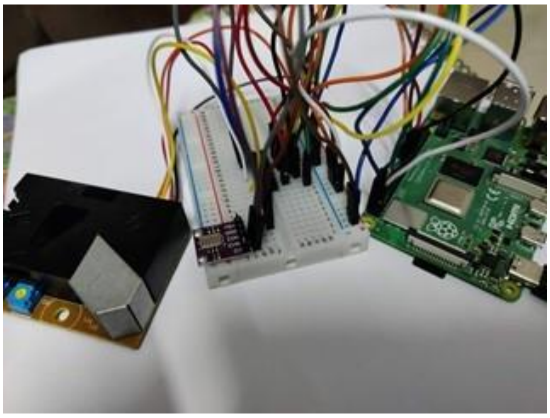

Fig. 8. Air quality monitoring system set up

A. Monitoring System

To verify the accuracy of the monitoring system, it was kept outside for a period of 30 days and measurements of pollutant concentrations were taken every hour. The readings were uploaded, along with the system timestamp at the moment of taking the reading, to the Firestore database. The readings were averaged for a day and converted to AQI values. The AQI values were plotted against official data for outdoor pollutant AQIs provided by Delhi Pollution Control Committee (Government of NCT of Delhi).

The graph displayed in Figure 9 shows a plot of the AQI (US EPA standard) values of the concentrations for carbon monoxide read by the monitoring system compared against the official AQI values for carbon monoxide provided.

Figure shows a plot of the AQI (US EPA standard) values of the concentrations for Nitrogen dioxide read by the monitoring system compared against the official AQI values provided. Figure 11, and Figure 12 show the same for concentrations of PM2.5 and PM10 respectively

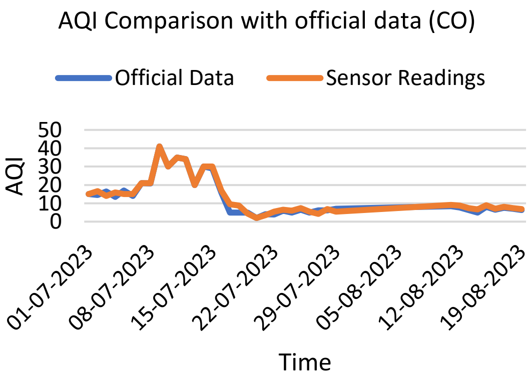

Fig. 9. Graph for CO AQI comparison

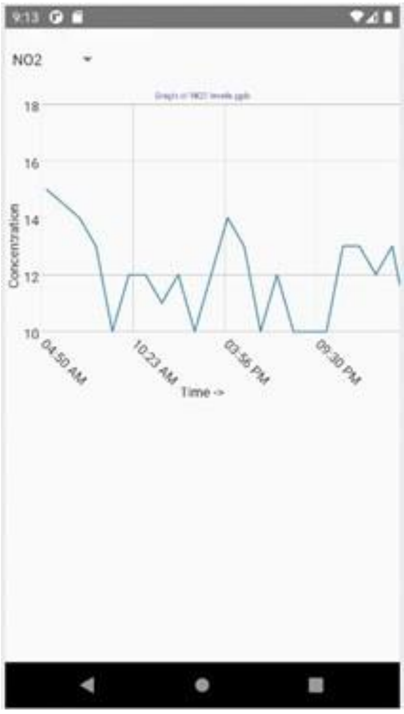

Fig. 10. Graph for NO_2 AQI comparison

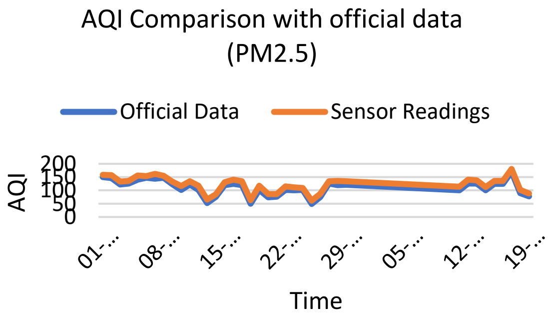

Fig. 11. Graph for PM2.5

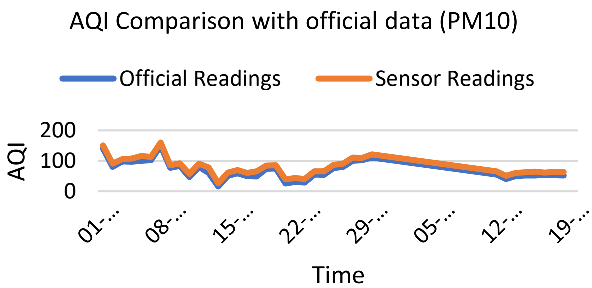

Fig. 12. Graph for PM10

B. Mobile Application and Displaying data

The mobile application provides a user interface where a user can easily view the data available in the cloud database. The application features a login screen that inter-faces with Firebase authentication services, a main screen that is used to display the most recently uploaded data, a graphs screen where graphs of the pollutant concentration in the last 24 hours is available, and an alerts screen that notifies the user if the concentration levels entered the harmful range at any point.

C. Login and Sign Up

The mobile application features a login screen (Figure 13) which provides the user a way to sign in to the application or sign up for one. The supported sign in method is email and password. Once the user enters their credentials, the credentials are sent to the Firebase Authentication service to authenticate the user. If the authentication succeeds, the user is taken to the main screen. A new user can sign up using the sign-up button provided on the login screen. As shown in Figure 14, the user must enter a name, email, and password to register for the application. Once registered, the user will receive a verification email from Firebase, which includes a link to verify their email. This is so fake email addresses are not used to register for the application.

Fig. 13. Login Screen

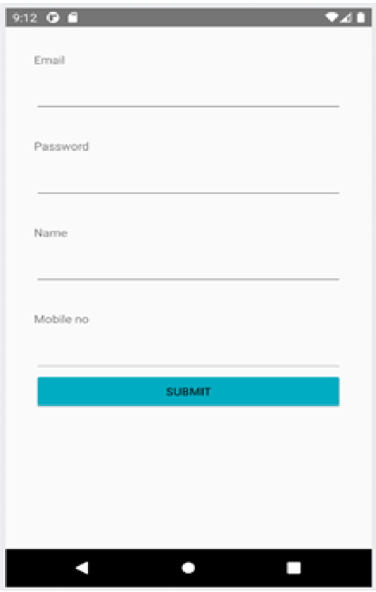

Fig. 14. New user registration Screen

D. Main Screen

Once signed in, the user lands on the main screen of the application (Figure 15). In in the background the application queries Firestore database for the latest readings. These readings are then displayed on the main screen, and are colour coded to represent their severity. This provides a user an idea of what the current levels of different pollutants are in the room.

Fig. 15. The latest readings are available on the main screen

E. Visualization

The mobile application also queries the sensor readings for the last 24 hours to enable visualizations for the readings. Figure 16 displays the graph of CO concentration from the last 24 hours plotted against time.

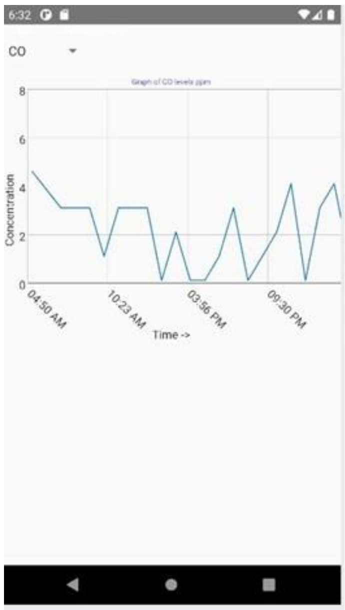

Fig. 16. Graph for CO concentrations in ppm

Figure 17 displays the graph of NO2 concentration from the last 24 hours plotted against time. Figure 18 displays the graph of PM2.5 concentration from the previous 24 hours plotted against time. These Visualizations help the user keep track of the pollutant concentrations for the past day and act accordingly.

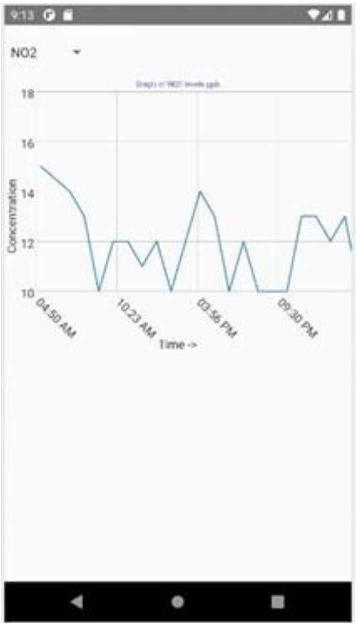

Fig. 17. Graph for NO_2 Concentrations in ppb

F. Alerts

The app features an alert system that notifies users when a pollutant's concentration becomes unhealthy. This also sends a notification to the user and serves as a warning system. The alerts page can be viewed in Figure 19. The alerts screen records past warnings generated and serves as a historical record of pollutant concentrations.

Fig. 18. Graph for PM2.5 concentration in $\mu g/m^3$

Fig. 19. Alert screen

5. Conclusion and Future Scope

The proposed system provides an indoor air quality monitoring setup using Raspberry Pi, along with gas and dust sensors that can give accurate and reliable readings, as shown by the results. It uses the Internet of Things and Cloud concepts to facilitate low power consumption on the device node. It detects some of the major pollutants in today's atmosphere and uploads the readings to the Firebase Firestore NoSQL cloud database. The values uploaded can be compared against safe ranges for specific pollutants. If the values are not in a safe range, the user is alerted through notifications, which are also displayed in-app. This allows the users to keep track of their surrounding environment and monitor pollution levels. and act accordingly. The future scope of the present work involves enabling scaling on the system nodes to support indoor areas that may be larger where the current system might not give entirely accurate readings, as well as implementing Bluetooth device registration for the application as a Quality of Life improvement. The system can also be enhanced by the addition of more sensors reading other environmental information such as temperature and humidity, Oxygen levels, and pollutant concentrations.

Author contributions

Aneesha Acharya K[1]: Conceptualization, Methodology, Software, Field study

Sucheta V. Kolekar[2]: Data curation, Writing-Original draft preparation, Software, Validation., Field study Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] D. P. M. Junior, C. Bueno, and C. M. da Silva, "The effect of urban green spaces on reduction of particulate matter concentration," Bulletin of Environmental Contamination and Toxicology, vol. 108, no. 6, pp. 1104-1110, 2022.

[2] M. Bentayeb et al., "Indoor air pollution and respiratory health in the elderly," Journal of Environmental Science and Health, Part A, vol. 48, no. 14, pp. 1783-1789, 2013.

[3] L. D. Prockop and R. I. Chichkova, "Carbon monoxide intoxication: an updated review," Journal of the neurological sciences, vol. 262, no. 1-2, pp. 122-130, 2007.

[4] A. Ghorani-Azam, B. Riahi-Zanjani, and M. Balali-Mood, "Effects of air pollution on human health and practical measures for prevention in Iran," Journal of research in medical sciences: the official journal of Isfahan University of Medical Sciences, vol. 21, 2016.

[5] E. Conticini, B. Frediani, and D. Caro, "Can atmospheric pollution be considered a co-factor in extremely high level of SARS-CoV-2 lethality in Northern Italy?" Environmental pollution, vol. 261, p. 114465, 2020.

[6] A. Shore, "Indoor Air Pollution: Environmental Inequality Inside," in Synthetic Planet: Routledge, 2013, pp. 25-50.

[7] M. U. Farooq, M. Waseem, S. Mazhar, A. Khairi, and T. Kamal, "A review on internet of things (IoT)," International journal of computer applications, vol. 113, no. 1, pp. 1-7, 2015.

[8] O. Bello and S. Zeadally, "Communication issues in the Internet of Things (IoT)," in Next-Generation Wireless Technologies: 4G and Beyond: Springer, 2013, pp. 189-219.

[9] K. Hwang, J. Dongarra, and G. C. Fox, Distributed and cloud computing: from parallel processing to the internet of things. Morgan kaufmann, 2013.

[10] A. Tiele, S. Esfahani, and J. Covington, "Design and development of a low-cost, portable monitoring device for indoor environment quality," Journal of Sensors, vol. 2018, 2018.

[11] S. Esfahani, P. Rollins, J. P. Specht, M. Cole, and J. W. Gardner, "Smart city battery operated IoT based indoor air quality monitoring system," in 2020 IEEE SENSORS, 2020: IEEE, pp. 1-4.

[12] K. Kanchan, A. K. Gorai, and P. Goyal, "A review on air quality indexing system," Asian Journal of Atmospheric Environment, vol. 9, no. 2, pp. 101-113, 2015.

[13] G. Marques, C. R. Ferreira, and R. Pitarma, "Indoor air quality assessment using a CO 2 monitoring system based on internet of things," Journal of medical systems, vol. 43, pp. 1-10, 2019.

[14] A. Kumar and P. Goyal, "Forecasting of daily air quality index in Delhi," Science of the Total Environment, vol. 409, no. 24, pp. 5517-5523, 2011.

[15] F. Afghan and S. Patidar, "Health impacts assessment due to PM2. 5, PM10 and NO2 exposure in National Capital Territory (NCT) Delhi," Pollution, vol. 6, no. 1, pp. 115-126, 2020.

[16] R. Goel, S. Gani, S. K. Guttikunda, D. Wilson, and G. Tiwari, "On-road PM2. 5 pollution exposure in multiple transport microenvironments in Delhi," Atmospheric Environment, vol. 123, pp. 129-138, 2015.

[17] S. K. Sahu and S. H. Kota, "Significance of PM2. 5 air quality at the Indian capital," Aerosol and air quality research, vol. 17, no. 2, pp. 588-597, 2017.

[18] T. H. Nasution, A. Hizriadi, K. Tanjung, and F. Nurmayadi, "Design of indoor air quality monitoring systems," in 2020 4rd International conference on electrical, telecommunication and computer engineering (ELTICOM), 2020: IEEE, pp. 238-241.

[19] C.-C. Ho, S.-C. Hung, and W.-C. Ho, "Effects of short- and long-term exposure to atmospheric pollution on COVID-19 risk and fatality: analysis of the first epidemic wave in northern Italy," Environmental research, vol. 199, p. 111293, 2021.

[20] M. Goumenou et al., "COVID-19 in Northern Italy: An integrative overview of factors possibly influencing the sharp increase of the outbreak," Molecular medicine reports, vol. 22, no. 1, pp. 20-32, 2020.

[21] S. McGrath, C. Flanagan, L. Zeng, and C. O'leary, "IoT personal air quality monitor," in 2020 31st Irish signals and systems conference (ISSC), 2020: IEEE, pp. 1-4.

[22] M. A. Mokar, S. O. Fageeri, and S. E. Fattoh, "Using firebase cloud messaging to control mobile applications," in 2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), 2019: IEEE, pp. 1-5.

[23] J. Esquiagola, M. Manini, A. Aikawa, L. Yoshioka, and M. Zuffo, "Monitoring indoor air quality by using iot technology," in 2018 IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), 2018: IEEE, pp. 1-4.

[24] Z. Liu, G. Wang, L. Zhao, and G. Yang, "Multi-points

indoor air quality monitoring based on internet of things," IEEE Access, vol. 9, pp. 70479-70492, 2021.

[25] J. Sundell, "On the history of indoor air quality and health," Indoor air, vol. 14, no. s 7, pp. 51-58, 2004.

[26] W.-T. Sung and S.-J. Hsiao, "Building an indoor air quality monitoring system based on the architecture of the Internet of Things," EURASIP Journal on Wireless Communications and Networking, vol. 2021, pp. 1-41, 2021.

[27] S. N. Azemi, K. W. Loon, A. Amir, and M. Kamalrudin, "An IoT-Based Alarm Air Quality Monitoring System," in Journal of Physics: Conference Series, 2021, vol. 1755, no. 1: IOP Publishing, p. 012035.

[28] J. Saini, M. Dutta, and G. Marques, "A comprehensive review on indoor air quality monitoring systems for enhanced public health," Sustainable environment research, vol. 30, no. 1, pp. 1-12, 2020.

[29] M. Benammar, A. Abdaoui, S. H. Ahmad, F. Touati, and A. Kadri, "A modular IoT platform for real-time indoor air quality monitoring," Sensors, vol. 18, no. 2, p. 581, 2018.

[30] K. Chan et al., "Low-cost electronic sensors for environmental research: Pitfalls and opportunities," Progress in Physical Geography: Earth and Environment, vol. 45, no. 3, pp. 305-338, 2021.

[31] R. Mumtaz et al., "Internet of things (Iot) based indoor air quality sensing and predictive analytic—A COVID-19 perspective," Electronics, vol. 10, no. 2, p. 184, 2021.

[32] L. Moroney, "An Introduction to Firebase," The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform, pp. 1-24, 2017.

[33] C. Khawas and P. Shah, "Application of firebase in android app development-a study," International Journal of Computer Applications, vol. 179, no. 46, pp. 49-53, 2018.

[34] M. Tanna and H. Singh, Serverless Web Applications with React and Firebase: Develop real-time applications for web and mobile platforms. Packt Publishing Ltd, 2018.

[35] R. Payne, Beginning app development with flutter: Create cross-platform mobile apps. Springer, 2019.

[36] R. S. Hadikusuma, L. Lukas, and E. M. Rizaludin, "Methods of Stealing Personal Data on Android Using a Remote Administration Tool with Social Engineering Techniques," Ultimatics: Jurnal Teknik Informatika, vol. 15, no. 1, pp. 44-49, 2023.

[37] V. Y. Lim, L. M. M. Peralta, L. H. Rubel, S. Jiang, J. B. Kahn, and B. Herbel-Eisenmann, "Keeping pace with innovations in data visualizations: A commentary for mathematics education in times of crisis," ZDM–Mathematics Education, vol. 55, no. 1, pp. 109-118, 2023.