An Iot-Based Apartment Safety Management System

Dr. A Pandian[1], Shanmuganathan S.[2], Shruthi P.[3], Shyam MN[4], Srinithi R.[5]
[1]*M.E., Ph.D., Assistant Professor, Department of ECE, SRM Valliammai Engineering College Kattankulathur, Chengalpattu, India*
[2,3,4,5]*Department of ECE, SRM Valliammai Engineering College Kattankulathur, Chengalpattu, India*

***Abstract*—This project proposes an IoT-based apartment safety management system designed to provide Realtime detection and response to gas leakage and fire hazards in residential buildings. The system employs embedded sensor nodes consisting of gas, smoke, and temperature sensors interfaced with a microcontroller to continuously monitor environmental conditions in individual apartment units. Sensor data is transmitted wirelessly to a centralized monitoring system using the MQTT communication protocol. Based on predefined safety thresholds, the system identifies hazard conditions and initiates immediate responses such as local alarm activation and alert notifications to residents and administrators. A centralized web dashboard enables real-time visualization of apartment safety status and maintains incident records for monitoring and analysis. The proposed system improves response time, enhances safety, and ensures reliable operation through effective integration of embedded systems, wireless communication, and real-time monitoring.**

***Index Terms*—Internet of Things (IoT), Apartment Safety Management, Gas Leakage Detection, Fire Hazard Monitoring, MQTT Communication Protocol, Real-Time Monitoring, Embedded Sensor Networks.**

I. INTRODUCTION

With the rapid growth of urban residential infrastructure, ensuring safety in apartment environments has become a critical concern. Hazards such as gas leakage, smoke accumulation, poor air quality, and sudden temperature rise can lead to severe accidents including fire outbreaks, health issues, and property damage. Conventional safety systems often rely on standalone alarms or manual monitoring, which may not provide timely alerts, centralized supervision, or historical incident analysis. Therefore, there is a growing need for an intelligent, connected, and real-time apartment safety management system that can continuously monitor environmental conditions and initiate rapid responses when hazardous situations arise.

The Internet of Things (IoT) has emerged as a powerful technology for developing smart safety solutions by integrating sensors, embedded controllers, wireless communication, and cloud-based monitoring platforms. In apartment safety applications, IoT enables continuous data acquisition from multiple environmental sensors, remote transmission of data, centralized analysis, and real-time notification to users and administrators.

This project presents an IoT-based Apartment Safety Management System designed to monitor critical safety parameters such as temperature, gas concentration, smoke level, and air quality in residential apartment units. The proposed system uses an ESP32 microcontroller as the central processing and communication unit due to its low power consumption, integrated Wi-Fi capability, and suitability for embedded IoT applications. Environmental sensing is achieved using a DHT11 sensor for temperature monitoring, an MQ-2 sensor for gas and smoke detection, and an MQ-135 sensor for air quality assessment. These sensors continuously collect real-time data from the apartment environment and send the measured values wirelessly through the MQTT communication protocol to a cloud-based Firebase platform for storage and remote accessibility. The collected sensor data are compared against predefined threshold values to determine whether the apartment environment is in a safe or hazardous state. A centralized web dashboard provides real-time visualization of sensor readings, apartment safety status, and incident records, enabling continuous monitoring, analysis, and improved decision-making. The collected data are stored and interpreted in the Cloud platform based on the threshold values of each factor and according to this the safety level is measured.

In addition to cloud-based monitoring, the hardware setup provides immediate local indication through LED status signaling and an audible alarm mechanism. Under normal conditions, a green LED indicates that all monitored parameters are within safe limits. When any sensed value exceeds its threshold, the system changes the status indication to a red LED, activates a buzzer, and generates an alert notification. This dual-level response mechanism ensures that both local occupants and remote supervisors are informed without delay, thereby improving the overall reliability and effectiveness of the safety management process.

The main objective of this work is to design a low-cost, reliable, and scalable apartment safety system that combines real-time sensing, cloud connectivity, local alarm response, and centralized visualization. By integrating embedded hardware with MQTT-based communication and cloud-assisted data management, the proposed solution enhances residential safety, reduces response time during emergencies, and supports modern smart apartment infrastructure.

II. LITERATURE SURVEY

The growing adoption of smart residential technologies has increased interest in IoT-based safety monitoring systems for homes and apartment buildings. Earlier studies in this area mainly focused on standalone gas leakage detectors or basic fire alarm units that operated independently without network connectivity. While such systems were capable of providing local warnings through buzzers or alarms, they were limited in terms of remote accessibility, data storage, and centralized supervision. Researchers gradually recognized that apartment safety requires not only immediate local alerting but also continuous monitoring, historical data analysis, and remote notification to occupants or administrators.

Recent studies have demonstrated that IoT-based gas leakage and fire safety systems play a significant role in improving residential protection by enabling continuous environmental monitoring and early warning generation. Research focused on smart gas leakage detection confirms that real-time sensing and alert mechanisms can effectively reduce the risk of accidents caused by combustible gas accumulation in domestic spaces [1]. Similarly, smart home safety systems that combine gas and fire detection using multiple sensors have shown improved hazard recognition accuracy and faster emergency response under abnormal conditions [2]. These findings highlight the growing importance of IoT-enabled safety platforms in modern residential environments.

In addition to gas leakage detection, indoor air quality monitoring has emerged as an essential component of smart safety systems. IoT-based research on environmental monitoring emphasizes that continuous measurement of air quality parameters helps maintain safer and healthier living conditions in enclosed spaces [3]. Multi-sensor emergency detection frameworks further demonstrate that the use of several distributed sensing elements can significantly improve the reliability of identifying smoke, gas leakage, and abnormal environmental variations in residential buildings [4]. Cloud-connected IoT monitoring systems have also shown strong potential for remote supervision and centralized data analysis, making them suitable for apartment-level safety management applications [5].

Several studies have also explored the use of wireless embedded platforms for gas and pollution monitoring. Real-time gas and air quality analysis systems have shown that cloud-integrated sensing nodes can detect pollution trends and provide timely alerts before the situation becomes hazardous [6]. IoT-based wireless air quality monitoring systems further confirm that low-cost embedded devices can be effectively used for continuous environmental surveillance in smart homes and apartments [7]. Similarly, IoT-based LPG and carbon monoxide monitoring systems have demonstrated the practical value of integrating gas sensors with wireless communication modules to detect domestic hazards at an early stage [8]. Advanced sensor fusion techniques have also been investigated, showing that combining multiple sensing modalities can improve gas detection accuracy while reducing false alarms in safety-critical applications [9].

Official Firebase documentation confirms that the platform supports continuous data synchronization, low-latency cloud storage, and real-time access to sensor readings, which are essential for centralized apartment safety dashboards [10], [11]. In addition, Firebase's web integration framework simplifies the development of browser-based dashboards for live visualization of environmental conditions and apartment-wise status reporting [12]. Security features such as database access rules also strengthen the reliability of cloud-based monitoring systems by ensuring controlled and secure handling of apartment safety data [13]. The ESP32 microcontroller is widely adopted in IoT applications due to its built-in Wi-Fi capability, efficient processing performance, and support for interfacing with multiple environmental sensors [14]. The MQ-2 sensor is suitable for detecting combustible gases and smoke, making it a practical choice for domestic gas leakage and fire-related hazard monitoring [15].

III. SYSTEM ARCHITECTURE

The proposed IoT-based apartment safety management system is organized into three major functional modules to ensure efficient sensing, communication, and response.

These modules are:

A. Sensing Module

The sensing module is responsible for continuously monitoring the environmental parameters associated with apartment safety. In the proposed system, three sensors are used: the DHT11 sensor for temperature measurement, the MQ-135 sensor for detecting air quality and harmful gas concentration, and the MQ-2 sensor for sensing smoke and combustible gases. These sensors are connected to the ESP32 microcontroller and provide real-time data regarding the apartment’s internal environmental conditions.

The collected sensor values are periodically sampled and used to identify abnormal situations such as high temperature, gas leakage, or smoke presence. Each sensor plays a specific role in hazard detection, and the use of multiple sensors ensures a more comprehensive safety monitoring mechanism.

B. Communication and Cloud Module

The communication and cloud module serves as the processing and data transmission layer of the proposed system. The ESP32 microcontroller acts as the central control unit, where sensor data from the DHT11, MQ-135, and MQ-2 sensors is collected and analyzed. Based on predefined threshold values, the ESP32 determines whether the sensed environment is in a safe or alert condition. Since the ESP32 has built-in Wi-Fi capability, it is well suited for real-time IoT-based monitoring applications.

After processing, the ESP32 transmits the sensor readings wirelessly using the MQTT protocol, which is selected for its lightweight and efficient publish–subscribe communication model. The transmitted data is stored in the Firebase cloud platform, where it is maintained for real-time access and incident logging. This cloud integration enables continuous synchronization of sensor data, supports remote monitoring, and provides a centralized storage mechanism for safety-related events.

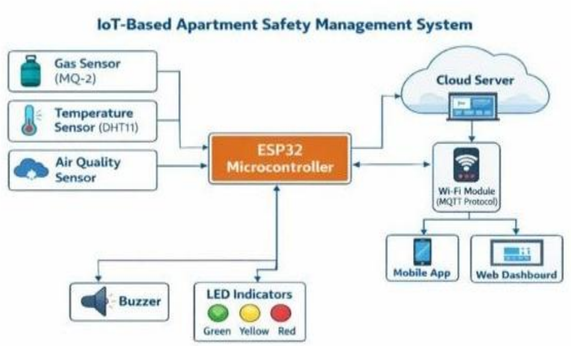
C. Alert and Visualization Module

The alert and visualization module provides both local and remote awareness of the apartment’s safety status. In the hardware setup, LED indicators are used to represent the current condition of the system. When all sensed values remain within the safe threshold range, the green LED glows to indicate normal operation.

If any monitored parameter exceeds its threshold value, the red LED is activated to indicate a hazardous or alert condition. In addition to visual indication, a buzzer is triggered whenever an unsafe condition is detected, providing an immediate audible warning to nearby occupants. Simultaneously, the sensor data and alert status are updated on a centralized web dashboard, which displays real-time values and maintains incident records for monitoring and analysis. This combination of local alerts and cloud-based visualization ensures faster response, improved user awareness, and effective safety management in apartment environments.

IV. METHODOLOGY

The methodology of the proposed IoT-based apartment safety management system is designed to provide continuous environmental monitoring, real-time hazard detection, wireless data transmission, and immediate alert generation. The complete operation of the system is carried out through a sequence of sensing, local processing, threshold evaluation, cloud communication, and user alert actions. The ESP32 microcontroller acts as the core processing unit that receives input from multiple sensors, compares the measured values with predefined safety limits, and determines whether the apartment environment is safe or hazardous. Based on this decision, the system updates the local hardware indicators and simultaneously transmits the information to the cloud for real-time dashboard visualization and incident recording.



The Block diagram is given in the above fig.1,

The proposed methodology is divided into the following major stages: (A) Sensor Data Acquisition, (B) Threshold-Based Hazard Analysis, (C) Local Alert Mechanism, (D) Cloud Communication and Data Logging, and (E) Real-Time Dashboard Monitoring and Notification Handling. Each stage contributes to the reliable functioning of the complete apartment safety framework.

A. Sensor Data Acquisition

The first stage of the methodology involves continuous collection of environmental parameters from the apartment unit using dedicated sensors. The DHT11 sensor is used to measure the ambient temperature, which helps in identifying abnormal temperature rise that may indicate fire-related risk. The MQ-135 sensor is used to sense air quality degradation and harmful gas concentration levels, making it suitable for monitoring indoor air safety conditions. The MQ-2 sensor is used to detect smoke and combustible gases, which are important indicators of gas leakage or fire hazards in residential spaces.

All three sensors are interfaced with the ESP32 microcontroller, which periodically reads their output values at fixed time intervals. The DHT11 provides temperature data in digital form, while the MQ-series sensors provide analog values corresponding to the concentration of detected gases or smoke in the surrounding environment. The ESP32 collects these values continuously and temporarily stores them for further analysis. This periodic sensing mechanism ensures uninterrupted monitoring of apartment conditions and allows the system to detect sudden environmental changes without delay.

B. Threshold-Based Hazard Analysis

After acquiring the sensor data, the next stage of the methodology is threshold-based hazard analysis. In this stage, the ESP32 compares each sensed value with a predefined threshold level that represents the safe operating limit for that particular parameter. The threshold values are selected based on experimental observation and practical operating requirements of the apartment environment. Separate threshold values are assigned for temperature, air quality/gas concentration, and smoke/gas detection.

If the measured values remain below their respective threshold limits, the system categorizes the apartment condition as safe. However, if the reading from any one of the sensors exceeds its threshold, the system interprets the condition as an alert or hazard state. This logic ensures that even a single abnormal factor is sufficient to trigger a warning, thereby improving the responsiveness of the system. The use of threshold comparison provides a simple and effective decision-making method that can be executed quickly on the ESP32 without requiring complex cloud-side processing.

This local analysis at the microcontroller level is an important part of the methodology because it allows the system to react immediately without waiting for server confirmation. Since apartment safety is time-critical, hazard detection must occur as close to the sensing point as possible. By performing threshold evaluation directly on the ESP32, the proposed system minimizes latency and ensures faster local response in emergency situations.

C. Local Alert Mechanism

Once the system determines the environmental condition, the next stage is local alert generation. In normal operation, when all sensor readings are within the safe threshold range, the green LED remains ON to indicate that the apartment environment is safe. This provides a simple and intuitive visual indication to occupants or maintenance personnel. The green LED acts as a status confirmation signal, showing that the sensors, ESP32, and monitoring logic are functioning correctly under safe conditions.

If any sensor value exceeds its threshold, the ESP32 immediately changes the hardware indication to an alert mode. In this condition, the red LED is activated to visually indicate danger, and the buzzer is triggered to provide an audible warning. The buzzer is particularly important because it can alert nearby residents even if they are not observing the LEDs or dashboard. This combination of red LED and buzzer ensures that the system provides both visual and audible emergency feedback.

The local alert mechanism is designed to operate instantly and independently of internet connectivity. Even if cloud communication is temporarily delayed or unavailable, the hardware-level alerts continue to function based on local sensor analysis. This makes the proposed system more reliable in practical apartment deployments where network interruptions may occur occasionally.

D. Cloud Communication and Data Logging

In addition to local processing and alerts, the proposed methodology includes cloud-based communication for remote monitoring and record maintenance. After each sensing cycle, the ESP32 transmits the latest sensor values and the current safety status wirelessly using the MQTT (Message Queuing Telemetry Transport) protocol. MQTT is selected because it is lightweight, efficient, and well suited for IoT applications involving frequent sensor updates and low-power embedded devices.

The ESP32 publishes the sensor data to the configured MQTT topics, allowing the information to be received by the cloud-connected monitoring platform. In this project, the transmitted data is stored in the Firebase cloud platform, which acts as the backend for real-time synchronization and historical record storage. Firebase maintains the latest values of temperature, air quality, smoke/gas levels, and system status. If a hazard condition is detected, the alert event is also stored as an incident entry for future review and analysis.

This cloud data logging capability is an important feature of the methodology because it extends the system beyond simple real-time detection. By preserving sensor history and incident records, the system supports long-term monitoring, fault analysis, and safety trend observation. Such records can be useful for apartment administrators or users to understand recurring safety issues and improve preventive maintenance.

E. Real-Time Dashboard Monitoring and Notification Handling:

The final stage of the methodology focuses on user interaction through real-time visualization and remote alert awareness. The centralized web dashboard continuously retrieves the latest data from the Firebase cloud and displays the real-time sensor readings in an organized format. The dashboard presents the current temperature, air quality/gas concentration, and smoke/gas levels along with the corresponding system status. This provides a centralized view of apartment safety conditions and allows users to monitor the environment remotely.

When a threshold violation occurs, the dashboard reflects the alert state immediately by showing the abnormal values and recording the event in the incident log. This allows both residents and administrators to identify when and why the hazard occurred. Since the system stores the incident records, the dashboard can also be used for historical monitoring and analysis of previous unsafe conditions. This feature improves the usefulness of the system beyond simple alarm generation.

In addition to visualization, the methodology includes alert notification handling, where unsafe conditions are flagged and communicated through the monitoring platform. This ensures that the hazard is not only indicated locally by the red LED and buzzer but is also made available remotely through the cloud-connected interface. Thus, the proposed methodology integrates local emergency awareness with centralized digital monitoring to create a complete apartment safety management framework.

F. Step-by-Step Working Procedure

The overall working procedure of the proposed system can be summarized as follows:

1. The ESP32 initializes all connected sensors, Wi-Fi connectivity, MQTT communication, Firebase cloud connection, LEDs, and buzzer.
2. The DHT11 sensor reads the current apartment temperature.
3. The MQ-135 sensor reads the air quality and gas concentration level.
4. The MQ-2 sensor reads the smoke and combustible gas level.
5. The ESP32 collects all sensor readings and stores them temporarily.
6. The collected values are compared with predefined threshold values for each parameter.
7. If all values are below threshold, the system status is set to SAFE.
8. In the safe state, the green LED glows, the red LED

remains OFF, and the buzzer remains OFF.

9. If any one of the sensor values exceeds its threshold, the system status is set to ALERT.
10. In the alert state, the green LED is turned OFF, the red LED glows, and the buzzer is activated.
11. The ESP32 publishes the latest sensor values and current system status using MQTT.
12. The transmitted data is stored in the Firebase cloud platform.
13. The web dashboard fetches the updated values from Firebase and displays them in real time.
14. If an alert occurs, the dashboard records the event in the incident history for monitoring and analysis.
15. The system repeats the sensing and monitoring cycle continuously at regular intervals.

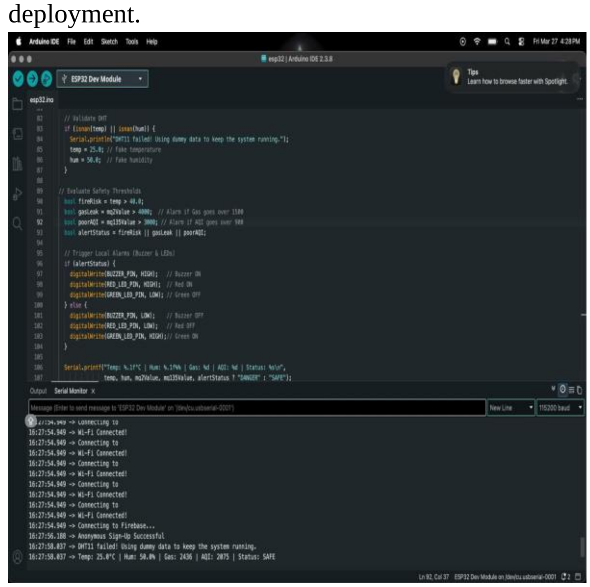
V. RESULT AND DISCUSSION

The proposed IoT-Based Apartment Safety Management System was successfully implemented and tested using the ESP32 microcontroller, DHT11 temperature sensor, MQ-135 air quality sensor, and MQ-2 gas/smoke sensor along with LED indicators and a buzzer for local alert generation. The system was evaluated under both normal operating conditions and simulated hazardous environmental conditions to verify its sensing accuracy, threshold-based decision logic, and real-time alert response. During testing, the ESP32 continuously acquired sensor values, compared them against predefined thresholds, and transmitted the data wirelessly to the cloud platform for dashboard visualization and incident monitoring.

Under normal operating conditions, the system demonstrated stable and reliable performance. As observed from the project output shown in the attached serial monitor reference, the system successfully established Wi-Fi and Firebase connectivity, and valid environmental readings were continuously displayed. A sample reading recorded during testing showed a temperature of 25.0°C, humidity of 50.0%, gas sensor value of 2436, and air quality value of 2075, with the system status identified as SAFE. In this condition, the green LED remained ON, while the red LED and buzzer remained OFF, confirming that all monitored parameters were within the acceptable safety range defined in the program logic.

To validate the hazard detection capability, abnormal environmental conditions were artificially simulated by exposing the MQ-2 and MQ-135 sensors to smoke/gas sources and by increasing the temperature level beyond the defined threshold. When one or more sensor readings exceeded their threshold values, the ESP32 correctly classified the environment as ALERT/DANGER. During this condition, the green LED was turned OFF, the red LED was activated, and the buzzer was triggered immediately. At the same time, the updated alert status and sensor values were transmitted to the cloud through MQTT and reflected in the centralized dashboard. This confirms that the system can provide both instant local warning and remote real-time monitoring, which is essential for apartment safety applications.

The experimental results indicate that the proposed system can effectively distinguish between safe and hazardous conditions using a multi-sensor threshold-based approach. The integration of local alert mechanisms with cloud-based data logging and dashboard visualization improves the practical usability of the system when compared with standalone alarm devices. Although the DHT11 occasionally produced invalid readings during certain test cycles, the program logic-maintained continuity by assigning fallback values, thereby preventing system interruption. Overall, the results demonstrate that the developed prototype is capable of reliable environmental monitoring, rapid hazard indication, and centralized apartment safety supervision at low cost and with good scalability for smart residential deployment.



From the above fig.2, it is observed that the proposed system reliably differentiates between safe and hazardous environmental states based on predefined threshold values. During normal operating conditions, all sensor readings remained below their threshold limits, resulting in a SAFE status with green LED indication and no buzzer activation. Under simulated hazardous conditions, whenever the temperature, gas/smoke level, or air quality value exceeded the threshold, the system correctly transitioned to ALERT mode, activated the red LED and buzzer, and updated the cloud dashboard accordingly. These results confirm the effectiveness of the proposed multi-parameter sensing and threshold-based response mechanism for apartment safety monitoring.

Table 1 below presents the environmental sensor monitoring results obtained from the developed prototype under both normal and hazardous conditions. Under normal operation, the recorded values remained within the predefined safe limits, resulting in a SAFE system status. During simulated hazardous conditions, the sensed values exceeded their corresponding warning thresholds, causing the system to classify the environment as hazardous. In such cases, the red LED and buzzer were activated, and the alert status was transmitted to the cloud dashboard for real-time monitoring and incident logging. The observed results confirm the effectiveness of the proposed threshold-based apartment safety monitoring system.

Table 1. Environmental Sensor Monitoring Results

Parameter	Normal Value	Warning Threshold	Recorded Hazard Value
Temperature (°C)	25	40	43
Humidity (%)	50	70	78
Gas concentration (ppm)	2400	4000	4500
Smoke level (ppm)	2100	3000	3500

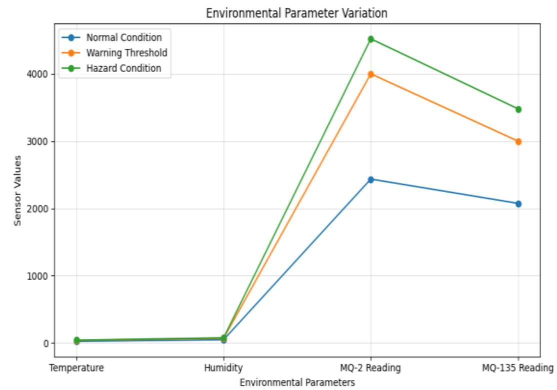


Fig.3 Environmental Sensor Monitoring Results

The above fig. illustrates the variation of the monitored environmental parameters under normal, warning threshold, and hazardous operating conditions. The graph compares the recorded values of temperature, humidity, MQ-2 sensor reading, and MQ-135 sensor reading. Under normal conditions, all parameters remain below their predefined threshold values, indicating a safe apartment environment. Under hazardous conditions, the recorded values exceed the corresponding warning thresholds, which causes the system to enter the alert state. This graphical representation clearly demonstrates the effectiveness of the threshold-based monitoring logic used in the proposed system for differentiating between safe and unsafe environmental conditions.

The experimental results of the proposed IoT-Based Apartment Safety Management System demonstrate the successful acquisition, transmission, and storage of environmental monitoring data from the apartment unit. The real-time database output shows that the system continuously records key safety parameters including temperature, humidity, gas level, and air quality, along with the corresponding alert status and last update time.

In the observed output for Apartment1, the measured values were temperature = 25°C, humidity = 50%, gas level = 2095, and air quality = 1170, which confirms that the ESP32-based sensing node was able to accurately collect sensor readings and upload them to the Firebase cloud platform through the configured wireless communication framework.

The recorded sensor values indicate that the apartment was operating under normal environmental conditions at the time of observation. The measured temperature of 25°C and humidity of 50% fall within a safe indoor comfort range for residential environments. Similarly, the gas level (2095) and air quality value (1170) remained below the predefined hazard thresholds established in the system logic. As a result, the alert_status field in the database was observed as false, indicating that no abnormal condition such as gas leakage, smoke accumulation, or excessive temperature rise was detected. This confirms that the threshold-based decision-making algorithm embedded in the ESP32 performs correctly in distinguishing safe operating conditions from hazardous states.

Another important result observed from the Firebase output is the presence of the last_update_time field, which verifies the continuous synchronization of sensor data between the hardware node and the cloud database. This field confirms that the sensing node periodically refreshes environmental values, thereby enabling real-time monitoring and ensuring that the centralized dashboard reflects the most recent apartment status. The successful cloud update mechanism validates the reliability of the communication architecture and proves that the system can maintain a live safety record for each apartment unit. Such continuous data logging is highly beneficial for maintaining incident history, monitoring environmental trends, and supporting future safety analysis.

Overall, the obtained results confirm that the proposed system effectively integrates embedded sensing, wireless cloud communication, and real-time safety monitoring into a single apartment protection framework. Under normal conditions, the system reliably indicates a safe state through the absence of alerts, while the architecture is designed to immediately activate the buzzer, change the LED indication from green to red, and issue notifications whenever any monitored parameter exceeds its threshold. Hence, the output validates the practical feasibility of the developed prototype and demonstrates its suitability for enhancing residential safety through timely hazard detection and rapid response.

VI. CONCLUSION

This paper presented the design and implementation of an IoT-Based Apartment Safety Management System for real-time monitoring and early detection of hazardous environmental conditions in residential apartments. The proposed system integrates an ESP32 microcontroller with DHT11, MQ-2, and MQ-135 sensors to continuously monitor temperature, humidity, gas level, and air quality within an apartment unit. The sensed data is transmitted wirelessly to the cloud and stored in Firebase, enabling real-time access through a centralized web dashboard. Based on predefined threshold values, the system is capable of identifying abnormal conditions and initiating immediate responses such as buzzer activation, visual alert indication using LEDs, and alert notifications. The experimental results confirm that the developed prototype can reliably distinguish between safe and unsafe conditions while maintaining continuous cloud-based data updates.

The implementation results demonstrate that the proposed system effectively improves apartment safety by combining embedded sensing, wireless communication, and real-time cloud monitoring into a unified safety framework. The system not only provides local alert mechanisms through hardware indicators but also supports remote supervision and incident tracking through a centralized dashboard. This makes it suitable for multi-apartment residential environments where timely hazard detection and quick response are critical for minimizing risks caused by gas leakage, smoke accumulation, or sudden temperature rise.

In future work, the system can be further enhanced by integrating automatic gas valve shutoff, mobile application support, SMS or push notification services, and additional sensors such as flame or motion detectors for broader safety coverage. The proposed model can also be extended into a scalable smart apartment or smart building safety platform, capable of monitoring multiple apartment units simultaneously. Overall, the developed system offers a practical, low-cost, and scalable solution for improving residential safety using IoT technology.

VII. FUTURE SCOPE

The proposed IoT-Based Apartment Safety Management System can be further enhanced by incorporating advanced safety and automation features to improve its practical applicability in large-scale residential environments. Future developments may include the integration of an automatic gas valve shutoff mechanism to immediately isolate gas supply during leakage detection, as well as SMS, mobile application, or push notification support for faster alert delivery to residents and building administrators. Additional sensing elements such as flame sensors, motion detectors, and carbon monoxide sensors can be included to expand hazard coverage and improve detection accuracy. The system can also be extended to support multi-apartment centralized monitoring with apartment-wise analytics, predictive risk assessment, and AI-based anomaly detection for smarter decision-making. These improvements would transform the proposed prototype into a more scalable, intelligent, and fully automated apartment safety platform suitable for real-world smart building deployments.

REFERENCES

[1] N. N., B. Harshini, and M. P. Mahalakshmi, "Smart gas leakage detection using IoT," *Int. Res. J. Adv. Eng. Hub (IRJAEH)*, vol. 3, no. 3, pp. 687–692, Mar. 2025, doi: 10.47392/IRJAEH.2025.0096.

[2] B. Deepika, S. Sivakami, S. Sneha, and B. Sujitha, "Early detection system for gas leakage and fire in smart homes using IoT," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 10, no. 3, pp. 102–108, May 2024, doi: 10.32628/CSEIT24102123.

[3] A. M. Simamora, A. Denih, and M. I. Suriansyah, "Indoor air quality detection robot model based on the Internet of Things (IoT)," *arXiv preprint*, arXiv:2505.19600, May 2025.

[4] A. Sterz, M. Sommer, K. Lüttge, and B. Freisleben, "Improving residential safety by multiple sensors on multiple nodes for joint emergency detection," *arXiv preprint*, arXiv:2502.15705, Jan. 2025.

[5] A. Hasib and A. S. M. A. S. Akib, "Cloud-enabled IoT system for real-time environmental monitoring and remote device control using Firebase," *arXiv preprint*, arXiv:2601.17414, Jan. 2026.

[6] A. Sarkar, D. Ghosh, K. Ganguly, S. Ghosh, and S. Saha, "Exploring IoT for real-time CO_2 monitoring and analysis," *arXiv preprint*, arXiv:2308.03780, Aug. 2023.

[7] M. E. Karar, A. M. Al-Masaad, and O. Reyad, "Gasduino—Wireless air quality monitoring system using Internet of Things," *arXiv preprint*, arXiv:2005.04126, May 2020.

[8] O. Flores-Cortez, R. Cortez, and B. González, "Design and implementation of an IoT-based LPG and CO gases monitoring system," *arXiv preprint*, arXiv:2107.07406, Jul. 2021.

[9] P. Narkhede, R. Walambe, S. Mandaokar, P. Chandel, K. Kotecha, and G. Ghinea, "Gas detection and identification using multimodal artificial intelligence-based sensor fusion," *arXiv preprint*, arXiv:2106.06180, Jun. 2021.

[10] Google Firebase, "Firebase Realtime Database: Store and sync data in realtime," 2026. [Online]. Available: Firebase official documentation. Accessed: Mar. 27, 2026.

[11] Google Firebase, "Firebase Realtime Database documentation," 2026. [Online]. Available: Official Realtime Database documentation. Accessed: Mar. 27, 2026.

[12] Google Firebase, "Installation and setup in JavaScript | Firebase Realtime Database," 2026. [Online]. Available: Official web setup guide. Accessed: Mar. 27, 2026.

[13] Google Firebase, "Get started with Firebase security rules | Realtime Database," 2026. [Online]. Available: Official security rules guide. Accessed: Mar. 27, 2026.

[14] Espressif Systems, "About ESP32," *ESP-IDF Programming Guide*, ver. 5.5.3, 2026. [Online]. Available: Official ESP32 documentation. Accessed: Mar. 27, 2026.

[15] Hanwei Electronics Group Corporation, "MQ-2 gas sensor datasheet," 2026. [Online]. Available: MQ-2 datasheet archive / technical data. Accessed: Mar. 27, 2026.