https://doi.org/10.1051/e3sconf/202560100008

# From Data to Decisions: A Smart IoT and Cloud Approach to Environmental Monitoring

Mohammed Guerbaoui[1], Samira El Faiz [1], Abdelali Ed-Dahhak [1], Abdeslam Lachhab [1], Bachir BENHALA [4], Zakaria Bakziz[2], Ismail Ichou [2], Abdelouahed Selmani [3]

[1]MMCS Team, EST Meknes, Moulay Ismail University, Meknes, Morroco
[2]Faculty of Sciences Meknes, Moulay Ismail University, Meknes, Morocco
[3]S.A.R.S Team , ENSA of Safi, UCA University, Marrakech, Morroco
[4]Faculty of science Dhar EL Mahraz, Sidi Mohamed Ben Abdellah University, Fes, Morroco

**Abstract.** Environmental monitoring plays a crucial role in various domains, including agriculture, healthcare, and manufacturing, where optimal environmental conditions are essential for productivity and safety. In this project, we present a smart environmental monitoring system that leverages IoT (Internet of Things) technology and data analytics to monitor temperature and humidity levels in real-time. The system consists of a network of sensor nodes deployed in the target environment, comprising ESP32 microcontrollers and DHT11 sensors for data collection. The sensor nodes transmit data using the MQTT (Message Queuing Telemetry Transport) protocol to a cloud-based MQTT Broker hosted on HiveMQ Cloud. Data processing and visualization are handled by Node-RED, which subscribes to MQTT topics, processes incoming data streams, and stores them in a time-series database, InfluxDB Cloud. The collected data is then visualized in real-time using Grafana dashboards, which are embedded within a Flask web application, providing stakeholders with seamless access to actionable insights into environmental conditions. The smart environmental monitoring system offers numerous benefits, including improved decision-making, proactive maintenance, and enhanced productivity. Future enhancements could include the integration of additional sensors and the application of machine learning algorithms for predictive analytics. Overall, the project demonstrates the potential of IoT and data analytics in addressing real-world challenges related to environmental monitoring.

## 1 Introduction

The integration of IoT technologies into environmental monitoring systems has led to significant advancements in data collection and real-time monitoring capabilities, revolutionizing how we manage and interact with our environment. IoT-enabled solutions are increasingly applied in diverse fields, from smart homes to industrial automation, enabling more informed decision-making and efficient resource management. As noted by Zafar et al. (2018), IoT systems are designed to monitor vital physical phenomena, generating data that can be transmitted and stored in the cloud, from where it can be accessed and utilized through various applications [1] . This emphasizes the role of IoT in providing reliable, scalable, and real-time environmental data, which is crucial for timely interventions and informed decisions.

Moreover, the application of MQTT (Message Queuing Telemetry Transport) protocol has further enhanced the efficiency of IoT systems by enabling seamless communication between devices. According to Macheso et al. (2021), MQTT is particularly advantageous in IoT applications due to its lightweight nature and publish/subscribe model, which facilitates efficient data transmission between devices [2]. This protocol ensures that data from environmental sensors, such as those used for monitoring temperature and humidity, can be transmitted reliably to central systems for processing and analysis.
The growing adoption of IoT technologies in environmental monitoring reflects a broader trend towards automation and smart systems that can operate with minimal human intervention. Gupta et al. (2021) highlight that these advancements not only improve the efficiency of monitoring systems but also contribute to energy conservation and cost savings by enabling remote control and automation of environmental systems [3].

## 2 PROPOSED SYSTEM ARCHITECTURE

The diagram below outlines the proposed architecture for the Smart Environmental Monitoring System. The system leverages the NodeMCU ESP32 for data collection and transmission, utilizing the MQTT protocol to communicate with a cloud-based MQTT broker hosted on HiveMQ Cloud. From there, Node-RED processes the incoming data streams and stores them in InfluxDB Cloud, a time-series database optimized for managing high volumes of data. Finally, the data is visualized in real-time using Grafana dashboards, offering comprehensive insights into environmental conditions.

This architecture ensures seamless integration between hardware and software components, facilitating real-time monitoring and decision-making.

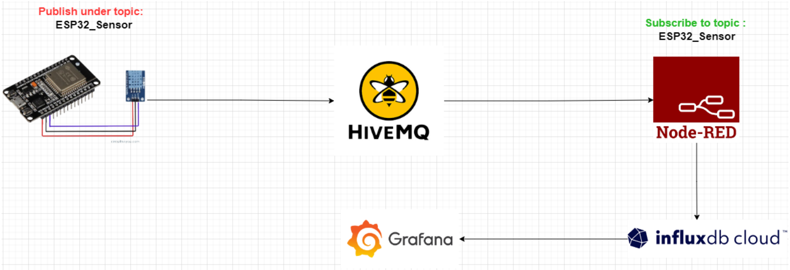


**Figure 1.** Overview of a System Architecture

### Flow of Data

Following the system architecture, the flow of data through the various components is critical to ensuring efficient data collection, transmission, processing, and visualization. Each component plays a vital role in enabling real-time monitoring and decision-making.

1. **Data Collection (ESP32 & DHT11 Sensors):** The process begins with the ESP32 microcontroller connected to DHT11 sensors, which measure temperature and humidity. The ESP32 module collects sensor data and prepares it for transmission.
2. **Data Transmission (MQTT Broker - HiveMQ Cloud):** The sensor data is transmitted using the MQTT protocol to the HiveMQ Cloud broker. The

ESP32 publishes data to specific MQTT topics, such as environmental readings (e.g., Node1/Temperature, Node1/Humidity) and the node status topic (e.g., Node1_Sensor/status, Node2_Sensor/status). This structure enables seamless communication between the devices and other components, allowing real-time data monitoring and status updates.

For a deeper understanding of the communication flow and topic structure, refer to Figure. 3

3. **Data Processing (Node-RED):** Once the data is published on the MQTT broker, Node-RED subscribes to the relevant topics and processes the incoming data. Node-RED acts as the integration hub, allowing for the routing, filtering, and transformation of data streams before storing the information in a time-series database.
4. **Data Storage (InfluxDB Cloud):** Processed data is stored in InfluxDB Cloud, a time-series database that efficiently handles high volumes of sensor data. Each data point is tagged with metadata (e.g., sensor ID, location) for easy retrieval and analysis, allowing for long-term data storage and historical trend analysis.
5. **Data Visualization (Grafana):** Finally, the stored data is visualized in real-time through Grafana dashboards. Grafana retrieves the time-series data from InfluxDB and provides users with interactive and customizable graphs, allowing them to monitor environmental conditions such as temperature and humidity. The dashboards also support real-time alerting, notifying users when data exceeds predefined thresholds.

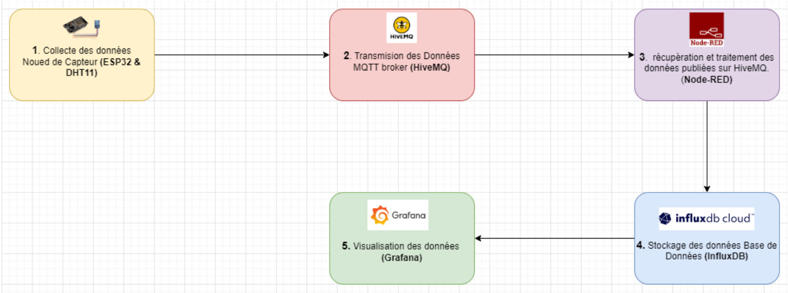


**Figure 2.** Data Flow Across System Components

## MQTT Architecture

The MQTT Architecture of the system outlines how data is communicated between the sensor nodes, the HiveMQ Cloud broker, and the clients that process and visualize the data. As shown in Figure 3, the architecture consists of:

1. **ESP32 Sensor Nodes:** These nodes collect environmental data (e.g., temperature and humidity) and publish it to specific topics on the MQTT broker. Additionally, they publish status messages to indicate whether a node is online or offline (e.g., Node1_Sensor/status).
2. **HiveMQ Cloud Broker:** The broker acts as the intermediary that manages the flow of data between the sensor nodes and the subscribing clients. Data published by the sensor nodes is forwarded to subscribers, such as Node-RED and a Paho MQTT Client.

3. **Node-RED:** This component subscribes to the data and status topics, processes the incoming data, and forwards it for storage and visualization.
4. **Paho MQTT Client:** Used by the web application to subscribe to the status topics (e.g., Node1_Sensor/status) and provide real-time monitoring of node health on the user interface.

This architecture enables the system to maintain real-time communication between the sensor nodes, the web application, and other critical components like Node-RED, allowing for efficient data monitoring, storage, and visualization.

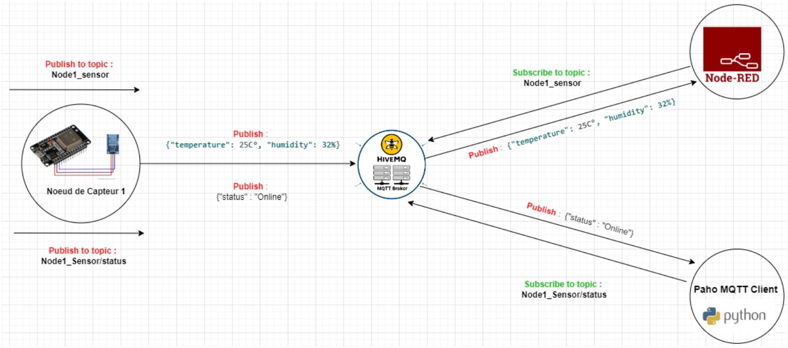


**Figure 3.** MQTT Communication Flow in the Smart Environmental Monitoring System

## 2.1 SYSTEM Hardware

### *2.1.1 NodeMCU ESP32 :*

The NodeMCU ESP32 is an open-source microcontroller board developed by Espressif, designed specifically for IoT applications. It builds upon the success of its predecessor, the ESP8266, but offers significant enhancements in both processing power and functionality. At its core, the ESP32 chip provides Wi-Fi and Bluetooth connectivity, making it versatile for various wireless communication needs. It features a Tensilica Xtensa 32-bit LX6 microprocessor, which is notably more powerful than the processor in the ESP8266, allowing for more complex and resource-intensive IoT applications.
The ESP32 module includes 128 KB of RAM and 4 MB of Flash memory, offering ample space for storing sensor data and executing programs. This capacity makes the ESP32 an ideal choice for projects that require real-time data processing and the handling of multiple sensor inputs. Furthermore, the board's TCP/IP stack enables seamless connectivity with cloud services and MQTT brokers, further enhancing its applicability in IoT-based environmental monitoring, smart home systems, and agricultural automation. The ESP32's balance of processing power, low-cost design, and wireless capabilities make it the backbone of our system's sensor nodes.

### *2.1.2 DHT 11 :*

The DHT11 is a basic, low-cost digital sensor used for measuring temperature and humidity. Although it does not offer the high precision of more advanced sensors, it is perfectly suited for general environmental monitoring applications where accurate, real-time data on ambient conditions is required. The DHT11 sensor is easily integrated with the NodeMCU ESP32 via its digital output, which transmits temperature and humidity data to the microcontroller. The simplicity of this sensor, combined with its low power consumption, makes it an efficient choice for continuous monitoring in projects such as weather stations or smart greenhouse systems.

## 2.2 SYSTEM SOFTWARE

### *2.2.1 Arduino IDE*

The Arduino Integrated Development Environment (IDE) is a widely-used open-source platform that enables users to write and upload code to microcontroller boards such as the ESP32. The IDE is compatible with major operating systems, including Windows, macOS, and Linux, making it accessible to a broad range of developers. It provides a straightforward interface for writing code in C/C++, and its built-in libraries make integrating with the ESP32 simple. The Serial Monitor feature of the Arduino IDE is invaluable for debugging, allowing users to track real-time data from the DHT11 sensor and other components, thus ensuring smooth data flow and execution.

### *2.2.2 MQTT Broker*

The MQTT Broker, in this case, HiveMQ Cloud, acts as the intermediary that facilitates the communication between IoT devices in our system. The MQTT protocol is particularly well-suited for IoT applications because of its lightweight design, which makes it efficient for low-bandwidth networks. HiveMQ Cloud allows devices, such as the ESP32, to publish sensor data to specific topics and subscribe to topics for receiving data. This publish-subscribe model ensures efficient, real-time communication between the sensor nodes and the central server. HiveMQ Cloud provides several advantages, including easy deployment, high availability, and scalability, which are crucial for managing the increasing volume of data generated by multiple sensor nodes in an IoT ecosystem.

### *2.2.3 Node-Red*

Node-RED is a visual programming tool that simplifies the integration of various devices, services, and APIs within IoT applications. Its browser-based flow editor enables users to create complex workflows with minimal coding. In our system, Node-RED subscribes to MQTT topics for incoming sensor data, processes the data in real-time, and routes it to both the database (InfluxDB) and visualization platform (Grafana). One of the key advantages of Node-RED is its flexibility in integrating different technologies, making it an essential component for handling the real-time data flow and control logic of the system.

### *2.2.4 InfluxDB*

InfluxDB is a cloud-native, time-series database that is ideal for storing the high volumes of time-stamped data generated by IoT devices. Its scalability and efficiency make it well-suited for handling the continuous stream of data from the ESP32 nodes. The time-series nature of the data allows for historical analysis and pattern recognition over time, which is critical for environmental monitoring systems. With InfluxDB, we can retain long-term data, execute complex queries, and optimize the storage for rapid data retrieval, ensuring that the system remains responsive even as the dataset grows.

### *2.2.5 Grafana*

Grafana is an open-source analytics and visualization platform that provides real-time insights into the data stored in InfluxDB. Grafana allows users to create custom dashboards that display key metrics, such as temperature and humidity, in a variety of formats including graphs, heatmaps, and gauges. This enables users to monitor environmental conditions in real-time and set alert thresholds that trigger notifications when sensor data exceeds predefined limits. Grafana's ability to integrate seamlessly with time-series databases like InfluxDB and its flexible visualization options make it the perfect tool for delivering actionable insights to end-users.

## 3 Litterature Review

The rapid development of the Internet of Things (IoT) has paved the way for various applications across fields such as home automation, environmental monitoring, and healthcare. The integration of real-time data processing and wireless sensor networks (WSNs) enables the continuous monitoring of physical parameters and the remote control of devices. Several studies have explored different architectures and technologies to implement such systems. While these contributions are valuable, there are notable areas where improvements can be made, particularly in terms of scalability, real-time analytics, security, and the inclusion of a user-friendly web interface, which our proposed system addresses.

Home automation has been an area of significant interest due to the convenience it provides in remotely controlling home appliances and optimizing energy consumption. For example, Agarwal et al. presented a system that utilizes Node-RED and MQTT for controlling home devices [4]. This system excels in using MQTT, a lightweight protocol, which facilitates efficient communication between devices. However, it remains limited in scope, focusing on home environments with a small number of connected devices. Similarly, Gupta et al. also explored home automation using ESP8266 microcontrollers and MQTT, but encountered similar limitations [3]. Although these systems are low-cost and user-friendly, they lack the scalability needed for more complex environments such as agriculture, where multiple sensors require real-time monitoring and control.

The use of IoT for environmental monitoring has also been explored in various studies, but these systems often fall short when it comes to providing advanced data analytics and real-time visualizations. Zafar et al. developed an environmental monitoring system that uses an Arduino UNO board and ThingSpeak for cloud storage [1]. This system allows for remote monitoring of temperature and humidity through an Android application, but it lacks the advanced real-time data visualization necessary for making timely decisions in critical environments. Winarno et al. also developed a smart home system using ESP8266 microcontrollers, which included monitoring temperature, humidity, and security factors like gas leaks [5]. However, like Zafar's work, it does not incorporate more sophisticated tools like Grafana for visualizing data trends, which limits its application in more complex environments where real-time analytics and decision-making are crucial.

As IoT systems continue to evolve, the importance of securing these systems becomes more pronounced, especially when they are deployed in critical applications like healthcare and agriculture. Stoyanova et al. highlight the growing need for security in IoT systems, particularly the challenges of securing data transmission and storage [6]. While their work provides valuable insights into the vulnerabilities of IoT systems, it focuses more on forensics and less on the real-time protection of data during transmission, which is critical for systems handling sensitive environmental data.

Compared to these previous works, our proposed system offers several improvements that address these limitations. First, while the systems developed by Agarwal et al. [4] and Gupta et al. [3] are effective in small-scale home automation, our system is designed for broader, more dynamic environments. We use ESP32 microcontrollers, which provide greater processing power and enable real-time monitoring of multiple sensors across larger areas, such as agricultural fields. Additionally, by integrating Node-RED, InfluxDB, and Grafana, our system allows for real-time data visualization and storage, providing users with the ability to monitor and act on environmental changes as they occur.

Where Zafar et al. [1] and Winarno et al. [5] relied on simpler cloud storage solutions like ThingSpeak, our system uses InfluxDB and Grafana to offer more advanced data analytics and long-term storage capabilities. This combination not only allows users to visualize data in real time but also to analyze trends over time, providing a more comprehensive understanding of the monitored environment. This makes our system more scalable and adaptable, particularly in sectors like agriculture, where multiple environmental parameters need to be continuously monitored and controlled.

In addition to scalability and real-time analytics, our system also addresses the security concerns highlighted by Stoyanova et al. [6]. We incorporate TLS encryption for secure data transmission over the MQTT protocol, ensuring that the data collected from sensor nodes is protected against unauthorized access during transmission. This added layer of security is crucial, especially in applications where sensitive environmental data is being monitored and acted upon.

A key feature of our system that distinguishes it from previous works is the inclusion of a web application that tracks the status of each sensor node and provides real-time visualizations of data. While most existing systems rely on mobile applications or simple dashboards for device control, our web application offers a more comprehensive user interface. It allows users to monitor the status of each sensor node in real time—indicating whether nodes are online or offline—and visualize key environmental data such as temperature and humidity. This integration provides users with immediate insights into the conditions of the monitored environment, empowering them to take timely actions when necessary. Moreover, the system's capability to track and display node statuses adds an additional layer of reliability, as it helps ensure that users are aware of any potential issues with the sensor network.

In conclusion, while previous works have made significant contributions to IoT-based home automation and environmental monitoring, they often fall short in addressing the challenges of scalability, real-time data analytics, security, and user interaction. Our proposed system not only overcomes these limitations but also provides a more comprehensive solution by integrating advanced data visualization, real-time analytics, secure communication protocols, and a user-friendly web interface for monitoring node statuses. This makes our system more versatile and capable of addressing a broader range of applications, from agriculture to healthcare and beyond.

## 4 METHODOLOGY

### 4.1 Node Sensor

The figure [4] shows the implemented prototype for the Sensor Node (ESP32 and DHT11). The Node ESP32 is programmed to connect to the HiveMQ cloud MQTT broker. This meant setting it up to use MQTT for communication and providing the broker's address and login details. Once connected, the ESP32 gathers sensor data and sends it to the broker using specific topics. This lets other devices or systems subscribe to the data and receive updates instantly. Additionally, it's worth noting that the ESP32 is powered by a USB cable.

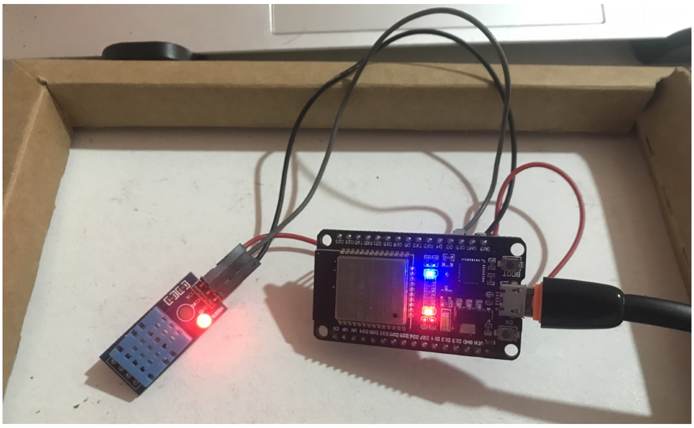


**Figure 4.** Sensor Node Prototype

## 4.2 Node-Red

Figure [5]. This Node-RED flow orchestrates data from IoT sensors, focusing on two distinct MQTT topics, Node1_Sensor and Node2_Sensor. Messages arriving on these topics are first intercepted by MQTT input nodes, where they undergo initial processing. The Generate Payload function node plays a crucial role by structuring incoming messages, extracting essential data such as temperature and humidity from each sensor node. Subsequently, a switch node categorizes the data based on its originating topic, channeling it to dedicated paths for further handling.

Each stream of processed data is then directed to an InfluxDB output node, where it is stored in the "Greenhouse_Monitoring" bucket. Within this bucket, data from Node1_Sensor is stored under the "Node1_data" measurement, while data from Node2_Sensor is stored under the "Node2_data" measurement. This structured approach ensures efficient storage and organization of sensor data, facilitating subsequent analysis and application within the monitoring system.

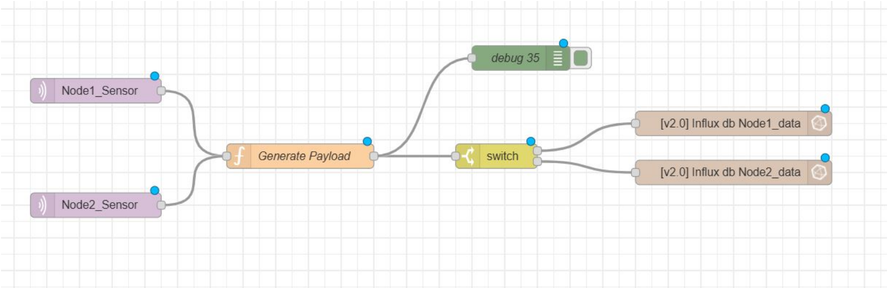
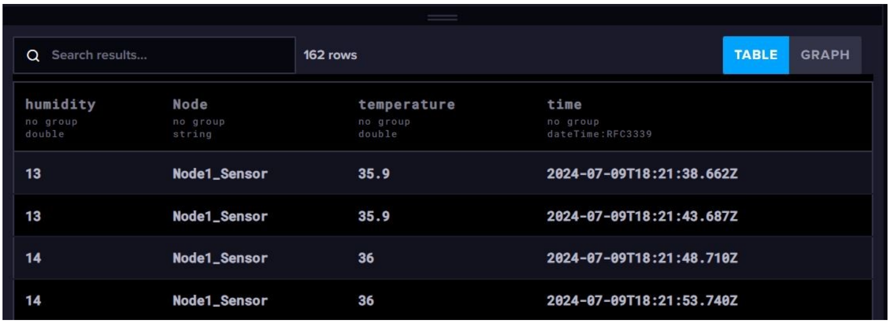


**Figure 5.** Node-red flow for influxDB

### 4.3 Cloud InfluxDB

IfluxDB offers two primary query languages for interacting with and manipulating the stored time-series data: Flux and SQL. These languages allow us to retrieve, analyze, and customize the vast amount of sensor information. As depicted in Figure [6], we can leverage an SQL query to extract specific data from the "Node1_data" measurement within InfluxDB. This query enables us to define a timeframe for retrieving data and filter it based on specific fields, ensuring we only retrieve entries with non-null values in those chosen fields. This focused approach streamlines data analysis and visualization.



| humidity<br>no group<br>double | Node<br>no group<br>string | temperature<br>no group<br>double | time<br>no group<br>dateTime:RFC3339 |
|---|---|---|---|
| 13 | Node1_Sensor | 35.9 | 2024-07-09T18:21:38.662Z |
| 13 | Node1_Sensor | 35.9 | 2024-07-09T18:21:43.687Z |
| 14 | Node1_Sensor | 36 | 2024-07-09T18:21:48.710Z |
| 14 | Node1_Sensor | 36 | 2024-07-09T18:21:53.740Z |

**Figure 6.** InfluxDB cloud Data Base

### 4.4 Grafana vizualisation

The figure [7] showcases a Grafana dashboard designed to visualize real-time temperature data from Node1_sensor. On the left, a gauge displays the current temperature, providing an immediate overview, while the graph on the right tracks temperature variations over time, allowing users to observe trends and fluctuations. The dashboard retrieves data using Flux queries from the Node1_data measurement within the Greenhouse_Monitoring bucket. In addition to real-time monitoring, the dashboard also includes an alert system, indicated by the red, yellow, and blue lines on the graph, which trigger notifications when the temperature crosses predefined thresholds. Similar dashboards are implemented for other nodes, ensuring consistent monitoring across the system.

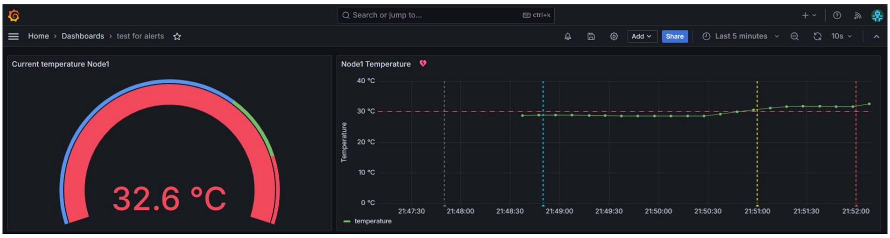


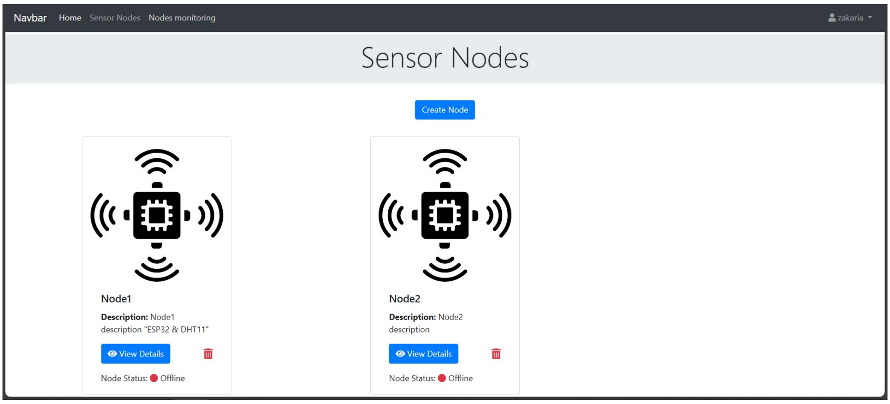
**Figure 7.** Real-Time Temperature Monitoring for Node1 on Grafana Dashboard

## 5 Implementation of Flask Application for Real-Time Node Visualization

The implementation of our Flask application for real-time node visualization showcases the seamless integration of sensor data monitoring and dynamic data presentation. This application is designed to offer an intuitive interface for users to monitor environmental parameters collected by sensor nodes, specifically focusing on Node1 in our example.

Our Flask application facilitates real-time visualization of sensor data through a simple and easily navigable web interface. By leveraging Flask's capabilities, we dynamically render templates and manage routes, ensuring users experience efficient and responsive interaction with the system. As depicted in figure[8], the route "/Nodes" presents an overview of all the sensor nodes, displaying their descriptions, current status, and providing options to visualize detailed data or update node information.
The dashboard prominently features real-time updates of temperature and humidity levels figure[9] and figure[10], displayed adjacent to their respective graphs. These graphs are Grafana visualizations embedded within the application, providing both instant insights and historical context, aiding in comprehensive data analysis. The data for these visualizations is fetched using Flux queries from the "Node1_data" measurement within the Greenhouse_Monitoring bucket, ensuring that the information is both current and relevant.
Moreover, the application indicates the node's status whether it is online, offline, or unknown using clear visual cues, allowing users to quickly assess the operational state of the sensor node. This real-time status update is a crucial feature, providing immediate feedback on the connectivity and functionality of the node.

In essence, our Flask application not only simplifies the process of real-time data visualization but also enhances the user experience by providing a cohesive and user-friendly interface for monitoring environmental data from IoT sensors. The integration of real-time status updates and dynamic data presentation underscores the application's capability to deliver valuable insights, enabling stakeholders to make informed decisions based on up-to-date information.



Node1
**Description:** Node1 description "ESP32 & DHT11"
Node Status: ● Offline

Node2
**Description:** Node2 description
Node Status: ● Offline

**Figure 8.** Overview of Sensor Nodes with Real-Time Status and Management Features in Flask Application

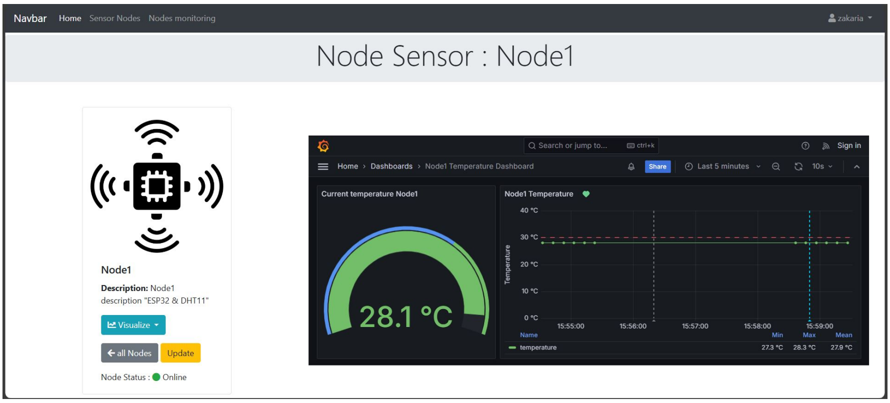


**Figure 9.** Node1 Temperature Monitoring Dashboard

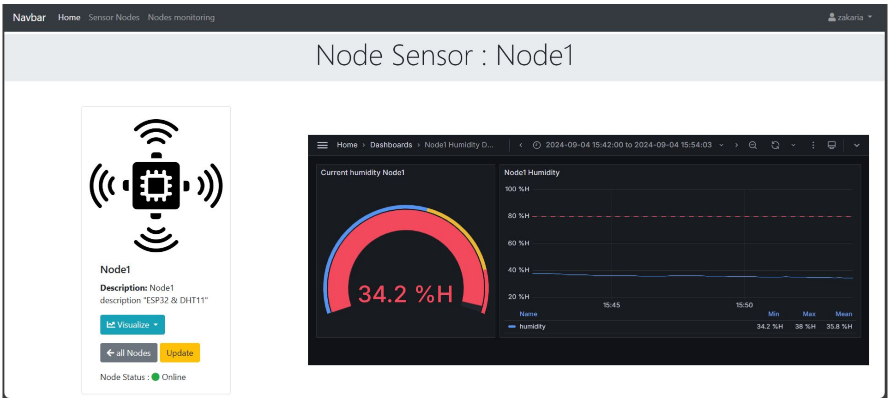


**Figure 10.** Node1 Humidity Monitoring Dashboard

## 6 CONCLUSION

This article demonstrates the successful integration of IoT technology and data analytics to create a smart environmental monitoring system capable of real-time tracking of temperature and humidity levels. By utilizing sensor nodes, the MQTT protocol, cloud-based services, and powerful visualization tools like Grafana, we have built a robust framework that not only enables real-time monitoring but also simplifies data visualization through a user-friendly Flask web application. The seamless integration of Grafana dashboards into the web interface, combined with real-time node status updates, ensures a comprehensive and accessible monitoring experience for stakeholders.

Looking to the future, there are numerous opportunities to expand and enhance the system. Incorporating additional sensors to monitor environmental factors such as air quality, soil moisture, and light intensity will provide a more holistic view of the monitored environment, contributing to smarter decision-making. Furthermore, integrating machine learning algorithms opens the door to predictive analytics, enabling the system to forecast trends and suggest proactive actions to mitigate potential issues.

By continuously innovating and refining this system, we aim to provide a scalable, adaptable solution that supports more sustainable environmental practices and meets the evolving needs of diverse sectors, from agriculture to urban environments. The ongoing evolution of this project will play a key role in fostering resilient ecosystems through intelligent environmental monitoring.

## References

[1] S. Zafar, G. Miraj, R. Baloch, D. Murtaza, K. Arshad, An iot based real-time environmental monitoring system using arduino and cloud service, Engineering, Technology & Applied Science Research **8**, 3238 (2018). 10.48084/etasr.2144

[2] P. Macheso, T.D. Mandab, S.W. Chisale, N. Dzupire, J. Mlathoa, D. Mukanyiligira, Design of ESP8266 Smart Home Using MQTT and Node-RED, in *2021 1st International Conference on Advances in Information Technology (ICAIS)* (IEEE, 2021), p. 9396027

[3] S. Gupta, Saurav, V. Patira, N. Jain, V. Kumar, Home automation solution using node-red and mqtt, International Journal of Advanced Research in Computer and Communication Engineering **10**, 335 (2021). 10.17148/IJARCCE.2021.101263

[4] A. Agarwal, R. Singh, A. Gehlot, G. Gupta, M. Choudhary, Iot enabled home automation through nodered and mqtt, International Journal of Control theory and Applications **10**, 217 (2017).

[5] A. Winarno, B.P. Sembodo, M. Affandi, Design and construction of smart house prototype based on internet of things (iot) using esp8266, BEST: Journal of Applied Electrical & Science Technology **4**, 11 (2022).

[6] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, E.K. Markakis, A survey on the internet of things (iot) forensics: Challenges, approaches, and open issues, IEEE Communications Surveys & Tutorials **22**, 1191 (2020). 10.1109/COMST.2019.2962586

[7] O.B.M. Magtibay, Applying mqtt protocol and node-red programming in designing iot based home automation, International Journal of Advanced Science and Convergence **1**, 7 (2019). 10.22662/IJASC.2019.1.3.007

[8] M. Esposito, A. Belli, L. Palma, P. Pierleoni, Design and implementation of a framework for smart home automation based on cellular iot, mqtt, and serverless functions, Sensors **23**, 4459 (2023). 10.3390/s23094459

[9] V. Sasikala, S. Cherishma, T.P. Reddy, Design and implementation of smart home control using ni lab view, International Journal of Electronics and Communication Engineering and Technology (IJECET) **10**, 17 (2019).

[10] D. Indra, Tasmil, Herman, S.H. Mansyur, E.I. Alwi, Design web-based electrical control system using raspberry pi, Journal of Information Technology and Its Utilization **2**, 1 (2019).

[11] B. Shireesha, M. Eruri, Home appliances controlling using raspberry pi on webpage, International Journal for Modern Trends in Science and Technology **2**, 140 (2016).

[12] E.N. Jannatheran, V.Y.K. Loung, C.Y. Leow, Iot smart home lighting system using arduino uno and esp8266, IEEE pp. 1–8 (2021).

[13] A. Zare, M.T. Iqbal, Low-Cost ESP32, Raspberry Pi, Node-Red, and MQTT Protocol Based SCADA System, in *2020 IEEE International Conference on Industrial Technology (ICIT)* (IEEE, 2020), pp. 1–7

[14] T.S. Gunawan, I.R.H. Yaldi, M. Kartiwi, N. Ismail, N.F. Za'bah, H. Mansor, A.N. Nordin, Prototype design of smart home system using internet of things, Indonesian Journal of Electrical Engineering and Computer Science **7**, 107 (2017). 10.11591/ijeecs.v7.i1.pp107-115

[15] D.P. Sharma, A. Baldeo, C. Phillip, Raspberry pi based smart home for deployment in the smart grid, International Journal of Computer Applications **119**, 6 (2015). 10.5120/21002-3637

[16] R. Akhter, S.A. Sofi, Precision agriculture using iot data analytics and machine learning, Journal of King Saud University-Computer and Information Sciences **34**, 5602 (2022). 10.1016/j.jksuci.2021.05.013

[17] D.P. Guragain, B. Shrestha, I. Bajracharya, A low-cost centralized iot ecosystem for enhancing oyster mushroom cultivation, Journal of Agriculture and Food Research **15**, 100952 (2024).

[18] F. Irwanto, U. Hasan, E. Saputra Lays, N.J. De La Croix, D. Mukanyiligira, L. Sibomana, T. Ahmad, Iot and fuzzy logic integration for improved substrate environment management in mushroom cultivation, Smart Agricultural Technology **7**, 100427 (2024). 10.1016/j.atech.2024.100427

[19] V.S. Reddy, S. Harivardhagini, G. Sreelakshmi, IoT and Cloud Based Sustainable Smart Irrigation System, in *E3S Web of Conferences* (EDP Sciences, 2024), Vol. 472, p. 01026

[20] A. Hafian, M. Benbrahim, M.N. Kabbaj, Iot-based smart irrigation management system using real-time data., International Journal of Electrical & Computer Engineering (2088-8708) **13** (2023).

[21] Y. Tace, M. Tabaa, S. Elfilali, C. Leghris, H. Bensag, E. Renault, Smart irrigation system based on iot and machine learning, Energy Reports **8**, 1025 (2022).