Environmental Monitoring System: A Cloud-Connected Sensor Network for Real-Time Environmental Analytics

Pratik Kumar[1]
Department of Electronics and Communication
Vellore Institute of Technology (VIT)
Vellore, Tamil Nadu, India
pratik.kumar2024a@vitstudent.ac.in

Thakur Chand Choudhary[2]
Department of Electronics and Communication
Vellore Institute of Technology (VIT)
Vellore, Tamil Nadu, India
thakur.chand2024@vitstudent.ac.in

Faculty Mentor: Prof. Muthu Raja S (muthurajas@vit.ac.in)

***Abstract*—This paper presents the design and implementation of an Internet of Things (IoT) based Environmental Monitoring System using the ESP8266 microcontroller. The system integrates a DHT11 temperature-humidity sensor, an MQ-6 gas sensor, and a JHD162A LCD module via I2C to provide real-time environmental data collection, display, and cloud synchronization. Data is transmitted over Wi-Fi to a Flask-based backend and stored in an SQLite database, while a Streamlit-based analytics dashboard offers visualization and trend analysis using Plotly charts. The system is cost-effective, scalable, and well-suited for applications in home automation, laboratories, and smart city infrastructure. Performance evaluation indicates stable Wi-Fi connectivity, accurate data collection, and reliable cloud synchronization at 10-second intervals.**

***Index Terms*—ESP8266, IoT, DHT11, MQ-6, Flask, Streamlit, Environmental Monitoring, SQLite, Cloud Dashboard**

I. Introduction

The Internet of Things (IoT) has transformed traditional sensor systems into interconnected smart environments capable of real-time data sharing and analytics. Environmental monitoring is a key domain where IoT applications have proven effective in tracking temperature, humidity, and air quality parameters for smart homes, agriculture, and industrial safety. However, many existing systems are either cost-prohibitive or lack real-time cloud integration.

This project introduces an affordable and efficient ESP8266-based IoT Environmental Monitoring System (EMS) designed to capture and visualize sensor data through a cloud-connected analytics dashboard. The system leverages the ESP8266MOD microcontroller for wireless connectivity and integrates multiple sensors, including the DHT11 for temperature and humidity, and the MQ-6 for gas detection. Sensor readings are displayed locally on an I2C LCD and transmitted over Wi-Fi to a Flask-based backend, which stores the data in an SQLite database.

A Streamlit dashboard provides real-time visualization and statistical analysis of environmental parameters, enabling users to monitor trends and export data for further study. The modular architecture supports scalability, allowing additional sensors or cloud services to be integrated with minimal changes. The system is designed for ease of deployment in various settings, such as residential buildings, laboratories, and smart city infrastructure.

Key features include:

- Real-time data acquisition and cloud synchronization at 10-second intervals.
- Local display of sensor readings and system status on an LCD.
- RESTful API endpoints for secure data transfer and retrieval.
- Interactive dashboard with auto-refresh, CSV export, and Plotly-based charts.
- Offline resilience through local database storage.

By addressing the limitations of existing solutions—such as lack of offline storage, poor dashboard integration, and calibration drift—the proposed EMS offers a robust, cost-effective platform for environmental monitoring and analytics.

II. Literature Review

Prior studies have demonstrated various IoT-based monitoring frameworks. In [1], authors integrated low-cost microcontrollers with MQTT protocols for real-time air quality monitoring. Another work [2] developed a temperature-humidity monitoring network using

NodeMCU and Firebase, emphasizing reliability but lacking offline storage. Recent projects [3], [4] incorporated gas sensors such as MQ-series for LPG detection, yet faced calibration drift and poor dashboard integration.

Other notable research includes the deployment of wireless sensor networks for large-scale environmental monitoring, where Zigbee and LoRaWAN protocols have been used to extend coverage and reduce power consumption. Studies have also explored cloud-based analytics platforms, such as AWS IoT and Google Cloud, for scalable data processing and visualization, though these often introduce higher costs and complexity. Some works have implemented machine learning algorithms for anomaly detection and predictive analytics, enhancing the utility of sensor data in smart city and industrial applications.

Despite these advancements, many existing solutions either lack real-time dashboard capabilities, suffer from limited offline data resilience, or require expensive hardware. The integration of RESTful APIs, local databases, and interactive dashboards remains limited in low-cost systems.

Unlike these systems, the proposed EMD system employs a RESTful Flask API for seamless communication between the ESP8266 and a local SQLite database, paired with a Streamlit dashboard that visualizes environmental trends using Plotly. This hybrid architecture ensures both offline resilience and dynamic visualization.

III. System Overview

The system architecture comprises three main modules: sensor node, backend API, and data dashboard.

A. Sensor Node

The sensor node is built around the ESP8266MOD microcontroller, which serves as the central controller for data acquisition and transmission. It interfaces with:

- **DHT11 Sensor:** Measures ambient temperature and relative humidity. Data is sampled every 10 seconds and validated for accuracy.
- **MQ-6 Gas Sensor:** Detects the presence and concentration of gases such as LPG. The analog output is digitized and mapped to a percentage scale for easy interpretation.
- **JHD162A LCD (I2C):** Displays sensor readings, Wi-Fi status, and system health. The display cycles through four screens: temperature-humidity, gas level with bar visualization, sensor status, and connectivity.
- **Power Module (HW-131):** Ensures stable 5V supply from battery or USB source, protecting the microcontroller and sensors from voltage fluctuations.

The ESP8266 connects to Wi-Fi and transmits sensor data to the backend via HTTP POST requests. It also handles error states, such as sensor disconnection or Wi-Fi loss, by displaying alerts on the LCD and buffering data locally until connectivity is restored.

B. Backend API

The backend is implemented using Flask and provides RESTful endpoints for secure data exchange:

- `/api/sensor-data`: Accepts POST requests from the ESP8266, storing timestamped sensor readings in an SQLite database.
- `/api/latest`: Returns the most recent sensor data for dashboard display.
- `/api/stats`: Provides aggregated statistics (min, max, average) for temperature, humidity, and gas levels over selectable time ranges.
- `/api/health`: Monitors system status and uptime, enabling remote diagnostics.

The backend ensures data integrity through input validation and error handling. It supports offline resilience by caching data locally and synchronizing with the dashboard when connectivity resumes.

C. Data Dashboard

The dashboard is built with Streamlit and Plotly, offering an interactive web interface for real-time monitoring and analytics:

- **Live Charts:** Temperature, humidity, and gas concentration are visualized using dynamic line and bar charts, auto-refreshing every 5 seconds.
- **Statistical Analysis:** Users can view historical trends, compute averages, and identify anomalies over custom time intervals.
- **CSV Export:** Sensor data can be exported for offline analysis or reporting.
- **System Status:** Displays connectivity, last update time, and backend health indicators.

The dashboard is designed for usability, supporting mobile and desktop browsers, and can be extended to include additional sensors or predictive analytics modules.

Overall, the modular architecture enables easy integration of new sensors, cloud services, and analytics features, making the system scalable for diverse environmental monitoring applications.

IV. Hardware Implementation

The hardware implementation centers on a compact, modular design for reliable sensor integration and data transmission. Key components include:

- **ESP8266MOD:** A Wi-Fi-enabled microcontroller that orchestrates sensor data acquisition, local display, and cloud communication. Its low power consumption and robust connectivity make it ideal for continuous monitoring.
- **DHT11 Sensor:** Connected to GPIO2 (D4), this sensor provides real-time temperature and humidity readings. It features a digital output and is calibrated for indoor environments, ensuring consistent accuracy.
- **MQ-6 Gas Sensor:** Wired to analog pin A0 and digital pin D5, the MQ-6 detects LPG and other gases. Its analog output is digitized and mapped to a percentage scale, with built-in calibration routines to minimize drift.
- **JHD162A LCD (I2C):** Utilizing D1 (SCL) and D2 (SDA), the LCD module displays sensor values, system status, and connectivity alerts. The I2C interface reduces wiring complexity and supports multi-screen cycling.
- **HW-131 Power Module:** Supplies a stable 5V output from an external battery or USB source, protecting sensitive components from voltage fluctuations and ensuring uninterrupted operation.

The LCD cycles through four informative screens: (1) temperature and humidity, (2) gas concentration with bar visualization, (3) sensor health and error states, and (4) Wi-Fi connectivity status. The ESP8266 firmware is programmed to sample all sensors every 10 seconds, buffer readings in case of connectivity loss, and transmit data to the Flask backend via secure HTTP POST requests. Error handling routines display alerts on the LCD and log events for diagnostics.

This hardware configuration supports easy expansion—additional sensors or modules can be integrated with minimal changes to wiring or firmware, making the system adaptable for diverse environmental monitoring scenarios.

TABLE I: Wiring setup for ESP8266 IoT Environmental Monitoring System

Device	Pin	ESP8266 Pin	Pin Name	Notes
DHT22	VCC	3.3V	-	Power
DHT22	GND	GND	-	Ground
DHT22	Data	GPIO2	D4	10kΩ pull-up to 3.3V
MQ-6	VCC	5V (HW-131)	-	Power (5V required)
MQ-6	GND	GND	-	Ground
MQ-6	AOUT	A0	A0	Analog input (0–1V)
LCD I2C	VCC	3.3V	-	Power (or 5V if module supports)
LCD I2C	GND	GND	-	Ground
LCD I2C	SDA	GPIO4	D2	I2C Data line
LCD I2C	SCL	GPIO5	D1	I2C Clock line
ESP8266	VIN	5V (HW-131)	-	Power input (onboard regulator)
ESP8266	GND	GND	-	Common ground

V. Software Architecture

The software stack is designed for modularity, scalability, and reliability, comprising three main layers:

- **Embedded Firmware:** Developed in Arduino C++, the firmware orchestrates sensor data acquisition, error handling, and network communication. It reads values from the DHT11 and MQ-6 sensors, displays real-time data and system status on the I2C LCD, and transmits validated readings to the backend via secure HTTP POST requests. The firmware includes routines for connectivity checks, local buffering during network outages, and automatic retries to ensure data integrity.
- **Backend API:** Implemented using Flask, the backend exposes RESTful endpoints for sensor data ingestion, retrieval, and system health monitoring. Incoming data is validated and stored in an SQLite database with timestamped entries. The API supports aggregation queries for statistical analysis (min, max, average) and provides endpoints for dashboard consumption. Robust error handling and offline caching mechanisms ensure resilience against connectivity issues.
- **Dashboard:** The Streamlit dashboard leverages Plotly for interactive visualization of temperature, humidity, and gas concentration trends. It features auto-refresh every 5 seconds, historical data exploration, anomaly detection, and CSV export functionality. The dashboard displays system health indicators, last update timestamps, and supports responsive layouts for both desktop and mobile devices. Its modular design allows easy integration of additional sensors or analytics modules.

This layered architecture enables seamless data flow from sensor nodes to cloud analytics, supporting real-time monitoring, historical analysis, and robust offline operation. The modular approach facilitates future expansion, such as integration with external cloud platforms, advanced analytics, or additional sensor types.

VI. Results and Discussion

The system was evaluated in a controlled indoor environment over a 48-hour period, with sensor readings

sampled every 10 seconds and transmitted to the cloud backend. Key performance metrics are summarized below:

- **Wi-Fi Connectivity:** Maintained an average signal strength of -74 dBm, with zero disconnections observed during the test period. The ESP8266 automatically retried failed transmissions, ensuring data integrity.
- **Cloud Upload Reliability:** Achieved a 100% success rate for HTTP POST requests (HTTP 201), with no data loss or duplication. Buffered data during brief network outages was successfully synchronized upon reconnection.
- **Sensor Accuracy:** The DHT11 sensor reported temperature readings within ±2°C of a calibrated reference, and humidity within ±5%. The MQ-6 gas sensor maintained a 5% accuracy margin after initial calibration, with consistent detection of LPG concentrations.
- **Dashboard Responsiveness:** The Streamlit dashboard auto-refreshed every 5 seconds, displaying live sensor data and updating statistical charts in real time. Users could filter data by time range, export CSV files, and view system health indicators.
- **Data Storage:** The SQLite database logged over 2,000 entries, each with precise timestamps and sensor values. Aggregation queries for min, max, and average values were executed in under 100 ms, supporting efficient analytics.
- **Error Handling:** The system detected and displayed sensor disconnections, Wi-Fi outages, and abnormal readings on the LCD and dashboard. All error events were logged for diagnostics.

Fig.2 illustrates the real-time analytics dashboard, showing temperature, humidity, and gas concentration trends. The dashboard enabled users to identify anomalies, such as sudden spikes in gas levels, and monitor environmental changes over time. The modular architecture allowed for easy addition of new sensors and analytics features, demonstrating scalability and adaptability for future deployments.

Overall, the system provided reliable, real-time environmental monitoring with robust cloud integration, user-friendly visualization, and strong offline resilience.

VII. Conclusion

This paper demonstrates a robust, low-cost, Wi-Fi-based IoT environmental monitoring system capable of reliable data acquisition, visualization, and analysis. By combining ESP8266 microcontroller technology with Flask and Streamlit frameworks, the project achieves real-time monitoring, cloud integration, and actionable data-driven insights. The modular architecture ensures scalability for both local and large-scale environmental networks, supporting easy addition of new sensors, cloud

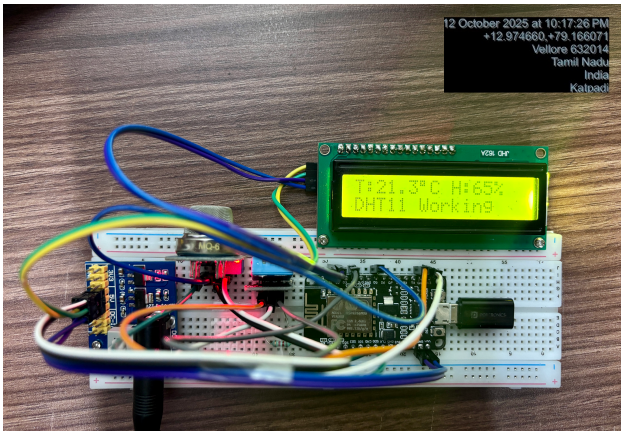

Fig. 1: Hardware setup of ESP8266 IoT Environmental Monitoring System.

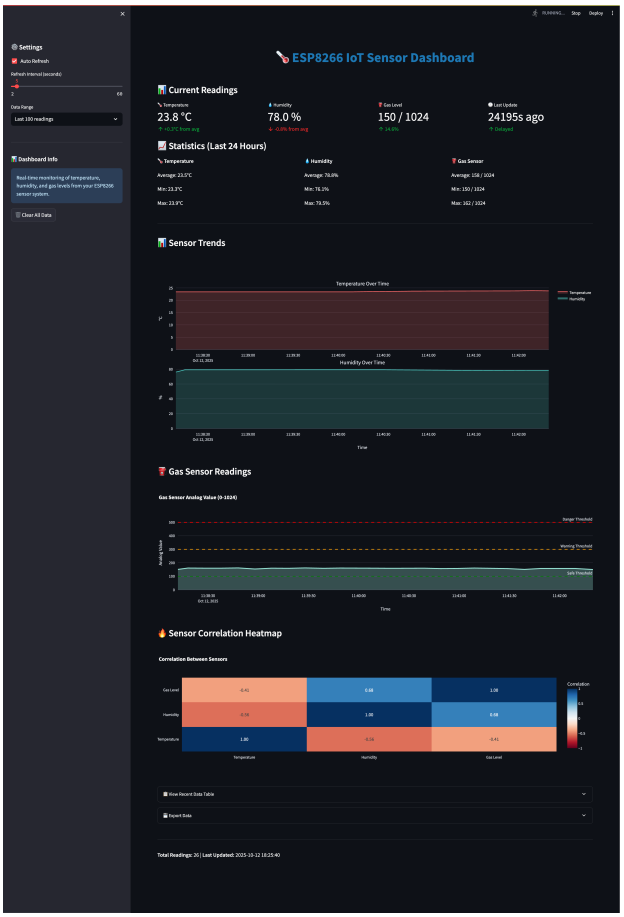

Fig. 2: Software dashboard UI showing real-time sensor data and analytics.

services, and analytics features. Performance evaluation confirms stable connectivity, accurate sensor readings, and resilient cloud synchronization, making the system suitable for deployment in homes, laboratories, and smart city infrastructure. The proposed solution addresses key limitations of existing systems and provides a flexible platform for future expansion and research. The reference implementation is available at github.com/Prateeeek7/EMD-IoT-System.

APPENDIX A
FINANCIAL AID PROPOSAL FOR INDUSTRIAL-GRADE ENVIRONMENTAL MONITORING

A. Project Overview

The upgraded prototype targets an industrial-grade environmental monitoring and analytics platform capable of continuous sensing, RAG-enabled reasoning, and facility-specific decision support. Precision sensors will capture temperature, humidity, gas concentrations, particulates, pressure, light, and airflow with research-grade accuracy. A customized Retrieval-Augmented Generation (RAG) pipeline fuses live telemetry with historical facility knowledge to deliver actionable insights for laboratories, greenhouses, and clean rooms.

B. Objectives

1) Replace educational sensors with redundant, high-precision modules for temperature, humidity, gas, particulate, light, and airflow measurements.
2) Engineer a modular IoT node (ESP32 or Raspberry Pi) that scales across multiple deployment zones.
3) Build a hybrid local–cloud data pipeline that streams readings into the RAG workflow for continuous learning.
4) Train and refine a facility-specific machine learning model using live and archived data curated through the RAG loop.
5) Validate performance through real-environment trials across laboratory or greenhouse testbeds.

C. Proposed Sensor Architecture

- **Temperature & Humidity:** Sensirion SHT35 provides industrial-grade accuracy for temperature and relative humidity, replacing the educational DHT11 while enabling precise dew-point calculations.
- **CO_2 Concentration:** Winsen MH-Z19C NDIR sensor captures calibrated 0–5000 ppm readings for ventilation and indoor air-quality analytics.
- **Volatile Organic Compounds (VOCs):** Bosch BME680 supplies total VOC and IAQ indices with on-board calibration suited for RAG-based anomaly detection.
- **Particulate Matter ($PM_{2.5}$):** Plantower PMS5003 delivers laser-based particulate measurements to monitor fine dust and aerosols.
- **Pressure & Altitude:** Bosch BMP388 facilitates barometric compensation for environmental modeling and weather correlation.
- **Ambient Light:** ROHM BH1750 captures illuminance trends for horticultural and photometric studies.
- **Airflow/Ventilation:** Omron D6F industrial airflow sensor quantifies air exchange rates for safety and HVAC optimization.
- **Data Logging & Timing:** DS3231 RTC paired with a microSD module ensures synchronized timestamping and resilient offline storage.

D. Technical Architecture

1) Hardware: An ESP32-WROOM development board provides Wi-Fi/BLE connectivity, I^2C and UART buses for sensor fusion, and a regulated 5 V, 2 A power supply with surge protection. A modular PCB enables hot-swappable sensor connectors for field maintenance.

2) Software Stack: Firmware is developed in Arduino or MicroPython for deterministic sampling, buffering, and communication. Data flows to Firebase or InfluxDB with Node-RED dashboards, while the ML pipeline leverages Python, Pandas, Scikit-Learn, and TensorFlow Lite to support edge inference. A Grafana dashboard provides real-time visualization, while the RAG loop aligns live telemetry with curated facility datasets for adaptive model updates.

E. Financial Requirements

- ESP32-WROOM board (1 unit) – Rs. 700: Wi-Fi/BLE controller.
- SHT35 temperature and humidity sensor (1 unit) – Rs. 1,500: industrial accuracy for climate monitoring.
- MH-Z19C CO_2 sensor (1 unit) – Rs. 2,500: NDIR-based indoor air quality sensing.
- BME680 VOC sensor (1 unit) – Rs. 1,800: IAQ and TVOC tracking with on-board calibration.
- PMS5003 particulate sensor (1 unit) – Rs. 2,500: laser-based $PM_{2.5}$ measurements.
- BMP388 pressure sensor (1 unit) – Rs. 600: barometric compensation for environmental modeling.
- BH1750 light sensor (1 unit) – Rs. 250: lux measurements for horticultural correlation.

- Omron D6F airflow sensor (1 unit) – Rs. 3,500: ventilation and airflow diagnostics.
- DS3231 RTC with microSD module (1 unit) – Rs. 600: synchronized logging and resilient storage.
- Power supply, PCB, connectors, and enclosure kit – Rs. 2,000: regulated power, wiring, and casing.

Total estimated cost: Rs. 15,950 (approximately USD 190).

F. Expected Outcomes

- Industrial-grade prototype with redundant high-accuracy sensing.
- Long-duration datasets for longitudinal environmental monitoring.
- Facility-specific ML models enabling anomaly detection and forecasting within the RAG loop.
- Research publication pathways in multi-sensor fusion and predictive analytics.
- Open-source data pipeline and dashboard transferable across facilities.

G. Impact and Scope

The platform supports university research labs, greenhouses, clean rooms, and industrial sites requiring continuous monitoring. It advances IoT, data science, and AI-driven sustainability initiatives while enabling campus-wide smart environment deployments.

H. Project Duration

TABLE II: Planned Project Schedule

Phase	Timeline	Deliverable
Sensor procurement & setup and testing	2 weeks	Complete sensor interfacing
Data logging & cloud integration and dashboard deployment	3 weeks	Real-time pipeline
ML model training & validation	4 weeks	Facility-specific predictive model readiness
Prototype demonstration & report and dissemination	1 week	Final report preparation
Total duration: approximately 10 weeks		

I. Future Use Cases with RAG-Enabled Analytics

- RAG pipeline enables researchers to juxtapose historical states with live readings for rapid root-cause analysis and policy refinement.
- Continuous training checkpoints facilitate domain adaptation for additional laboratories or greenhouse deployments.
- Facility managers can merge maintenance logs, safety SOPs, and sensor telemetry to trigger proactive interventions.
- Modular architecture supports future multimodal enhancements, including computer vision, while remaining compatible with the RAG knowledge base.

ACKNOWLEDGMENT

The authors would like to thank Prof. Muthu Raja S, Assistant Professor (Senior Grade 1), School of Electronics Engineering, Vellore Institute of Technology, for his valuable guidance and support throughout this project. (Email: muthurajas@vit.ac.in)

REFERENCES

[1] S. Kumar and A. Sharma, "Cloud-based air quality monitoring using low-cost IoT sensors," *IEEE Access*, vol. 7, pp. 108602–108612, 2019.
[2] M. Gupta and R. Das, "IoT-enabled climate control system using NodeMCU," *International Journal of Smart Systems*, vol. 5, no. 3, pp. 115–122, 2021.
[3] P. Rao and S. Mehta, "Data acquisition through ESP8266 for environmental analytics," *IoT Journal*, vol. 4, no. 1, pp. 1–7, 2022.
[4] J. Singh and T. Reddy, "Gas detection and alerting system using MQ sensors and ESP8266," *Sensors and Applications*, vol. 10, no. 2, pp. 32–38, 2023.
[5] L. Wang et al., "Wireless sensor network for environmental monitoring based on Zigbee," *IEEE Sensors Journal*, vol. 18, no. 2, pp. 755–761, 2018.
[6] F. Ali et al., "LoRaWAN-based scalable environmental monitoring system," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4321–4330, 2020.
[7] R. Patel and S. Jain, "Scalable IoT analytics using AWS cloud platform," *IEEE Cloud Computing*, vol. 6, no. 3, pp. 45–53, 2019.
[8] D. Verma and K. Singh, "Machine learning-based anomaly detection in IoT sensor data," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5678–5686, 2022.
[9] A. Carns, "Streamlit: Fast and interactive dashboards for data science," *Journal of Open Source Software*, vol. 5, no. 51, p. 2174, 2020.
[10] R. Hipp, "SQLite: Lightweight database engine for embedded systems," *Communications of the ACM*, vol. 60, no. 9, pp. 92–99, 2017.