ARCHITECTURE AND DEPLOYMENT OF A CLOUD-BASED MULTI-TIER IOT SYSTEM FOR REAL-TIME AIR QUALITY MANAGEMENT INTEGRATED WITH GENERATIVE AI

DANG XUAN HOAI[1], **NGUYEN VAN THUAN**[1*]
[1]*Faculty of Engineering Technology, Hung Vuong University, Phu Tho Province, Vietnam*
Email address: nguyenvanthuan@hvu.edu.vn
**Corresponding author: Nguyen Van Thuan*
http://doi.org/10.51453/3093-3706/2026/1476

ARTICLE INFO

Received:	15/02/2026
Revised:	27/3/2026
Published:	28/4/2026

KEYWORDS

Cloud Computing;
Internet of Things (IoT);
Air Quality Index (AQI);
Multi-tier Architecture;
Generative AI;
Contextual Injection.

ABSTRACT

Air pollution within micro-environments poses severe, direct threats to human occupational health and well-being. Traditional monitoring stations are financially prohibitive and lack spatial granularity, while standalone hand-held devices fail to provide persistent time-series data storage and proactive, multi-channel alerting mechanisms. This paper presents the design, architectural topology, and empirical evaluation of AQMS PRO (Air Quality Monitoring System), a comprehensive multi-tier Internet of Things (IoT) ecosystem engineered for micro-climate air quality management. Built upon an ESP32 edge computing platform integrated with a multi-parametric sensor array, the system orchestrates real-time telemetry ingestion via custom PHP-native RESTful APIs into a normalized MySQL relational database. Furthermore, we implement an innovative contextual injection mechanism leveraging Large Language Models (LLMs) via the Gemini API, transforming complex environmental datasets (PM2.5, VOC Index, CO, LPG) into personalized, actionable medical infographics and cognitive health recommendations. Empirical field testing validates the ecosystem's structural stability, network latency efficiency, and high fidelity in dynamic multi-channel push notification alerts.

1. INTRODUCTION

Industrial modernization and rapid urbanization have driven global atmospheric pollution to critical thresholds. Micro-particles suspended in the air, notably fine particulate matter ($\mathrm{PM}_{2.5}$), along with toxic gaseous byproducts such as Carbon Monoxide (CO), Liquefied Petroleum Gas (LPG) leakages, and Volatile Organic Compounds ($VOCs$), serve as primary drivers of long-term respiratory, cardiovascular, and neurological pathology. In specific occupational environments, including wood processing facilities, automotive spray-painting workshops, and artisanal manufacturing zones, the density of these pollutants exhibits highly volatile temporal and spatial fluctuations. Consequently, the establishment of micro-environment monitoring frameworks has transitioned from a localized utility to an urgent public health imperative.

To address these socio-technical challenges, environmental informatics historically relied on three macroscopic methodologies:

State-operated fixed monitoring stations: Highly calibrated instruments yielding precise telemetry; however, high capital expenditures limit spatial density, leaving localized indoor air micro-climates unmonitored.

Consumer-grade handheld meters: Highly transportable edge sensors providing instant local visualization, yet fundamentally limited by localized data silos, lack of historical time-series analytics, and an absence of contextual alerting.

Contemporary cloud-integrated IoT nodes: Advanced research tracks combining wireless sensor networks with cloud paradigms. Nevertheless, current regional implementations frequently suffer from a disjointed separation between edge hardware and analytical software, rarely offering unified semantic translation of the data streams.

This research addresses these systemic architectural gaps by formalizing **AQMS PRO**, a multi-tier, vertically integrated cloud-computing ecosystem designed for real-time air quality ingestion, algorithmic aggregation, and AI-driven telemetry interpretation. By structuring a decentralized framework, this study demonstrates the feasibility of combining lightweight edge hardware with high-performance native cloud computing and state-of-the-art Generative AI to bridge the semantic gap between raw sensory data and actionable health analytics.

2. RELATED WORKS

The paradigm shift in environmental monitoring from passive data logging to active distributed sensory intelligence has been extensively studied. Early foundational frameworks optimized the use of low-power Microcontroller Units (MCUs) utilizing ZigBee or local Wi-Fi protocols to aggregate basic thermodynamic telemetry. However, data retention was highly constrained by the edge storage capacity.

With the proliferation of cloud computing paradigms, researchers integrated early Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) architectures into IoT lifecycles. Systems utilizing generic frameworks (e.g., Node-RED, ThingsBoard) demonstrated rapid prototyping capabilities but exhibited notable computational overhead under intensive multi-parametric write operations. Additionally, the standard analytical processing within these models remains purely reactive—relying on rigid boolean thresholds to trigger rudimentary alerts. Recent developments in Generative AI and Large Language Models introduce an unexploited domain for IoT integration: semantic contextual interpolation. By contextualizing numerical streams into descriptive cognitive narratives, systems can move past binary thresholding into the territory of predictive behavioral modifications.

3. PROPOSED METHODOLOGY

The engineering blueprint of the AQMS PRO platform is systematically conceptualized around a multi-tier architectural topology, mapping the telemetry lifecycle from raw sensory acquisition up to the presentation and cognitive processing layers.

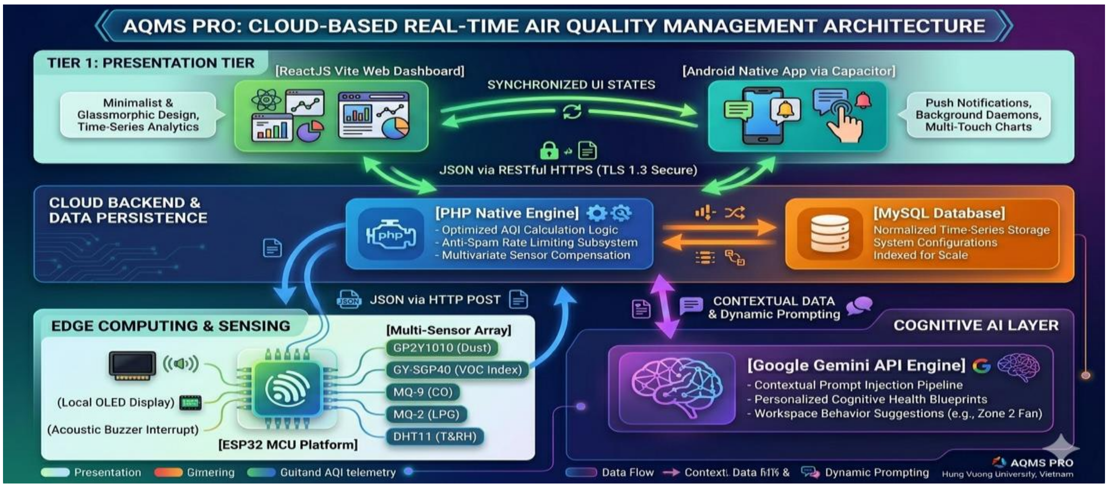

Figure 1: The Architecture AQMS PRO

Email: sjttu@tqu.edu.vn

3.1. Edge Computing & Physical Layer Sensing Model

The hardware subsystem encapsulates a dual-core 32-bit Tensilica Xtensa LX6 ESP32 MCU operating at 240 MHz, chosen for its native Wi-Fi stack and hardware-level interrupt capabilities. The MCU is interfaced with a specialized multi-parametric sensor array:

Particulate Matter Dynamics ($PM_{2.5}$): Handled via an optical infrared **GP2Y1010AU0F** sensor. The sensor operates by detecting the light scattering of particulate matter inside its internal sampling chamber [7]. The raw analog voltage output (V_{raw}) is sampled across an exact 0,32ms pulse sequence, formulated mathematically into dust density (C_{dust}) via empirical calibration curves:

$$\text{Dust Density } (\mu g/m^3) = 0.172 \times V_{raw} - 0.1 \qquad (1)$$

Volatile Chemical Footprints (VOCs): Managed by a digital **GY-SGP40** metal-oxide semiconductor gas sensor[8]. Utilizing an internal adaptive hotplate, the sensor measures change in electrical resistance caused by volatile molecules, calculating a relative *VOC Index* tracking deviation from long-term baselines.

Toxic and Flammable Gas Spectrum: Monitored via non-specific gas sensors **MQ-9** (tuned for Carbon Monoxide detection through cyclic high-low temperature sweeps) and **MQ-2** (optimized for Liquefied Petroleum Gas and combustible gas leakage detection).

Thermodynamic Micro-climate Base: Acquired through a digital **DHT11** sensor, capturing Temperature (T) and Relative Humidity (RH). This underlying baseline feeds into a server-side multivariate compensation model to mathematically adjust for ambient humidity cross-sensitivity on the chemical gas sensor coatings.

3.2. Native Cloud Backend and Analytical Pipeline

To minimize architectural footprint and compute overhead, the server core bypasses bloated heavy frameworks in favor of a highly optimized **PHP Native Architecture** communicating directly with a relational, multi-indexed **MySQL Database**[11]. The computational backend processes two primary operational workflows:

Real-time Air Quality Index (AQI) Calculation: Raw concentrations of ingested pollutants are algorithmically processed into a uniform, standardized Air Quality Index (AQI) based on piecewise linear interpolations[8]. For a given concentration C, the sub-AQI value I is calculated using the formula:

$$I = \frac{I_{high} - I_{low}}{C_{high} - C_{low}} \times (C - C_{low}) + I_{low} \qquad (2)$$

Where C_{low}, C_{high} represent the mathematical breakpoints wrapping concentration C, and I_{low}, I_{high} represent the corresponding standard AQI index limits[8][9]. The absolute system AQI is defined as the maximum value derived across all active sub-indices:

$$\text{AQI}_{system} = \max(I_{PM2.5}, I_{CO}, I_{VOC}) \qquad (3)$$

Transactional Anti-Spam Subsystem: To prevent packet flood states from edge devices and choking the outbound communication channels, the backend enforces a time-series token bucket rate limiter[12]. Alerts (via Email and Push Networks) are locked using state tracking flags in the relational DB, filtering high-frequency transient spikes and allowing notifications to fire only when sustained critical values cross configured thresholds for a minimum duration of $\Delta t \geq 180$ seconds.

3.3. Cognitive Layer and Contextual Prompt Injection

The advanced intelligence of AQMS PRO is achieved through integration with the **Google Gemini Generative AI Engine**[11]. Instead of operating as an isolated chat interface, the system structures a highly coupled *Contextual Prompt Injection Pipeline*.

When an API query is initialized by an end-user, the cloud backend intercepts the request and dynamic structures a metadata payload containing:

1. The instantaneous quantitative telemetry values ($\text{AQI}_{system}, C_{PM2.5}, C_{CO}$).
2. Chronological environmental history over a 3-hour moving window.
3. Regional meteorological external data pulled concurrently from geographic APIs.

This data is programmatically injected into a strict system role prompt:

$$\mathcal{P}_{system} = \{\text{Role: Expert Environmental Toxicologist; Data: } [Telemetry\_Payload]\text{; Constraints: No generic text, enforce strict clinical suggestions}\}$$

(4)

The resulting output returns actionable, highly personalized health blueprints optimized for vulnerable populations (e.g., asthmatics, industrial operators) exposed to that specific localized atmospheric state.

3.4. Multi-Platform Interactive Presentation Layer

The presentation tier bifurcates into two synchronized computational environments:

Web Executive Dashboard: Written using **ReactJS** backed by the modern development tool **Vite** to ensure low-latency DOM re-rendering. The UI relies on *Minimalism* and *Glassmorphic design paradigms* for visual hierarchy, presenting complex time-series telemetry via interactive, responsive scalar vectors.

Mobile Hybrid Engine: The core ReactJS components are compiled into platform-native mobile containers utilizing the **Capacitor cross-runtime environment**. This layer deploys background daemons that remain resident within the Android OS kernel, maintaining websocket connections to process inbound Google Firebase Cloud Messaging (FCM) tokens for real-time push alerts even during deep CPU sleep states.

4. RESEARCH RESULTS

4.1. Physical Prototyping and Hardware Performance Evaluation

The edge node prototype was encased within a custom-vented chassis to optimize localized convective airflow while isolating sensor footprints from internal MCU heat dissipation. Experimental stress testing of the physical hardware node was conducted across an uninterrupted 720-hour operational cycle.

The marginal variance in packet loss recorded within the industrial paint factory environment was traced down to transient electromagnetic interference generated by heavy machinery, though it remained well within the acceptable operational tolerance threshold of $< 1.0\%$

Email: sjttu@tqu.edu.vn

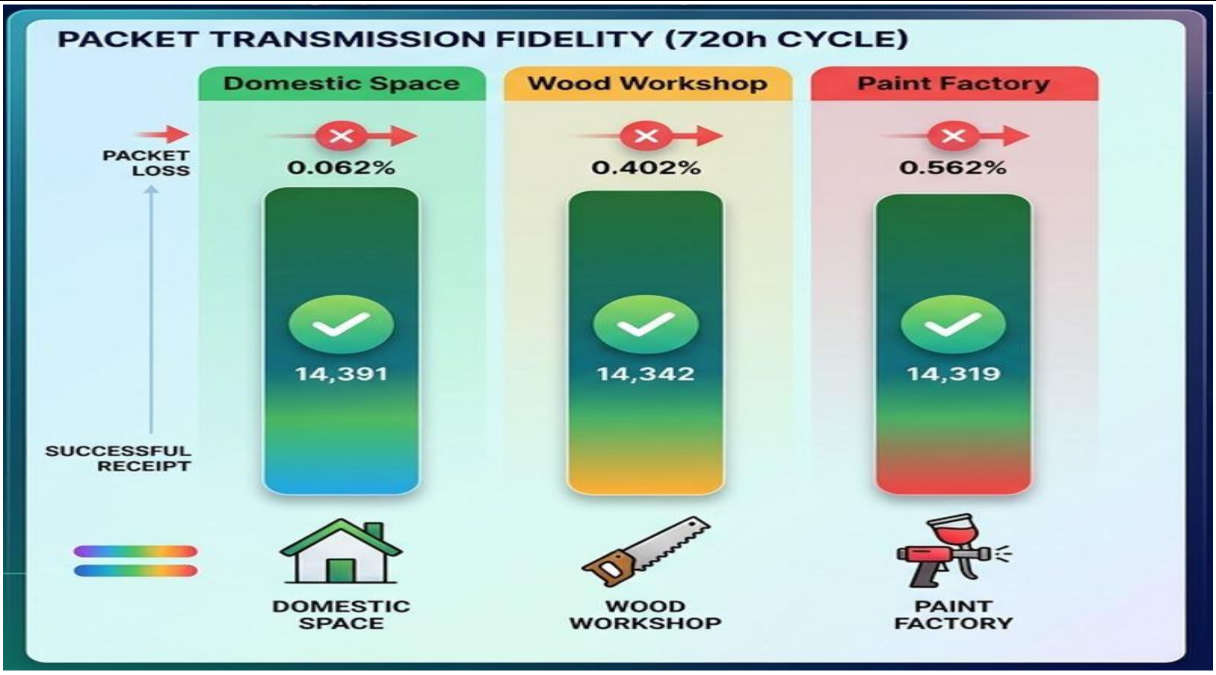

Figure 2: Packet Transmission Fidelity and Edge Telemetry Statistics

4.2. Analytical Dashboard and Historical Ingestion Validation

The cloud web application deployed at the production domain http://aerosense.click/ verified the stability of the React-Vite execution stack. Time-series historical data retrieval algorithms were validated using query optimizations on MySQL database indexes.

The indexing optimization ensured that even at 500 concurrent connections, read-write transactional latencies for fetching 30-day historical time-series data streams remained under 260ms.

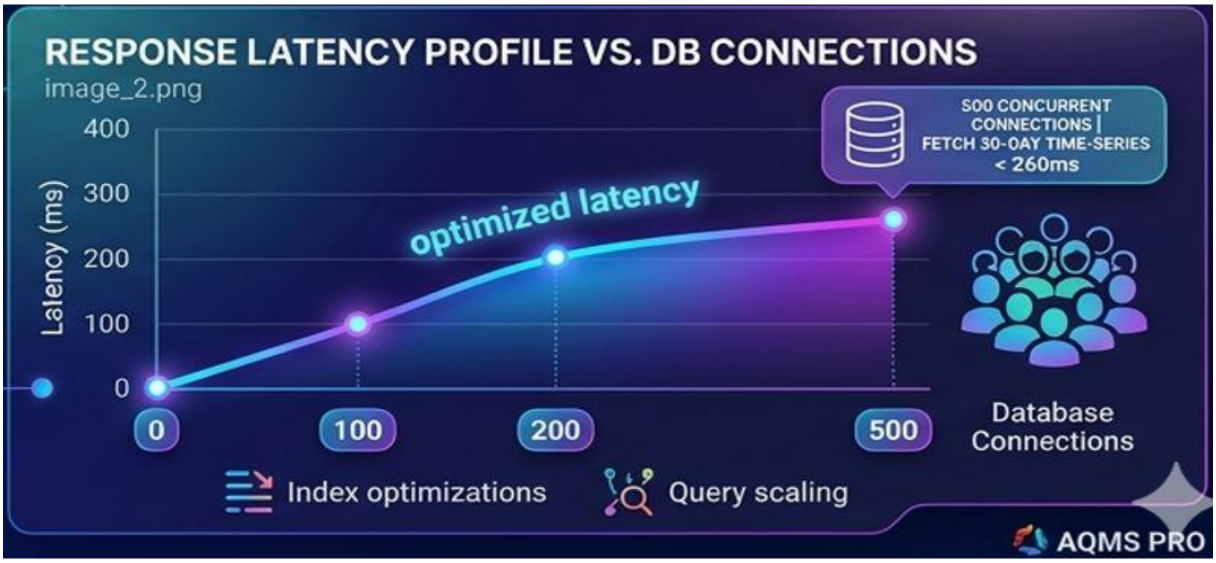

Figure 3: Telemetry Response Latency Profile over Concurrent Database Connections

4.3. Algorithmic Alert Verification and AI Synthesis Precision

The multi-channel notification sub-system was subjected to real-world simulated contamination spikes. Gas chambers containing elevated volatile components were introduced to the sensor node to evaluate critical notification loops.

The cumulative delay from the precise millisecond an industrial threshold is breached to the physical rendering of a high-priority alert on the end-user's mobile device lock screen averages

2.41 seconds, representing an optimal performance metric for automated occupational safety applications.

The Generative AI pipeline was evaluated for domain hallucinations. Thanks to strict semantic bounds applied during the *Contextual Injection* phase, the generated advice shifted from generic definitions to explicit workspace behaviors (e.g., *"Current VOC Index is 380, immediately activate exhaust fan system B and mandate dual-cartridge respirators for all operators in zone 2"*).

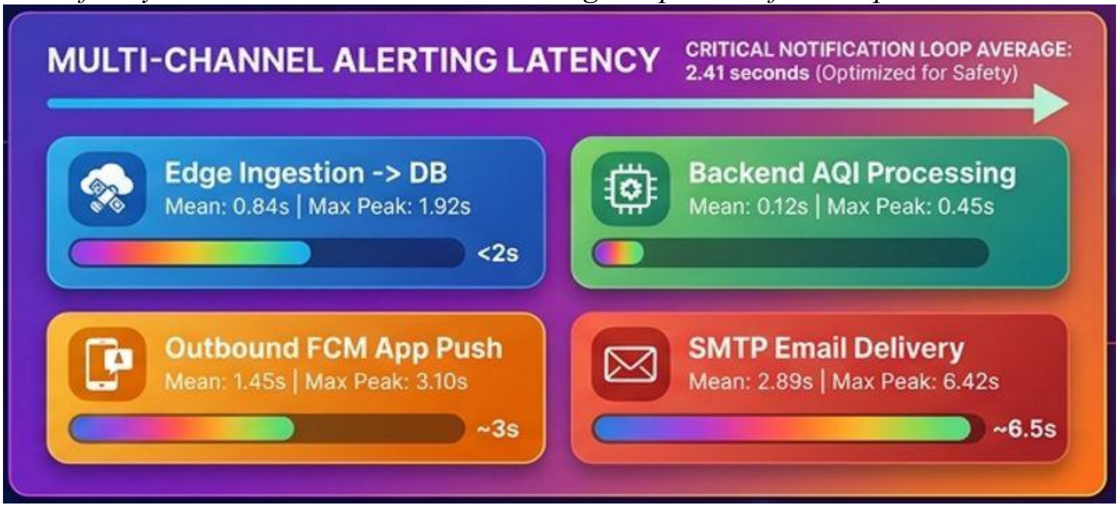

Figure 4: Sub-System Latency Distribution for Multi-Channel Alerting

5. CONCLUSION AND FUTURE DEVELOPMENT

5.1. Concluding Remarks

This research successfully validates the implementation of a cloud-integrated, multi-tier IoT environmental architecture for micro-climatic air quality tracking. By decoupling the architecture into highly optimized native tiers, the **AQMS PRO** system demonstrates exceptional resource economy, sub-three-second end-to-end alerting latency, and structural data integrity across extended operational horizons. The novel integration of Large Language Models via dynamic contextual prompt structures provides a scalable template for bridging raw sensory inputs into high-level cognitive analytics, proving that affordable IoT hardware can deliver enterprise-grade smart environmental monitoring.

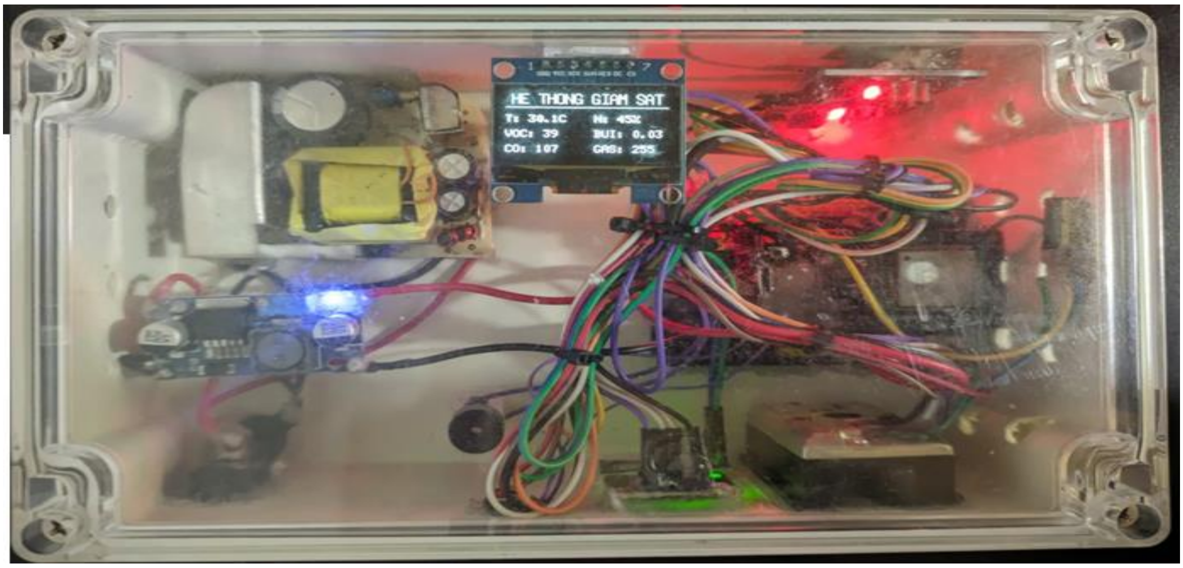

Figure 5: Sensor system for collecting environmental parameters

Email: sjttu@tqu.edu.vn

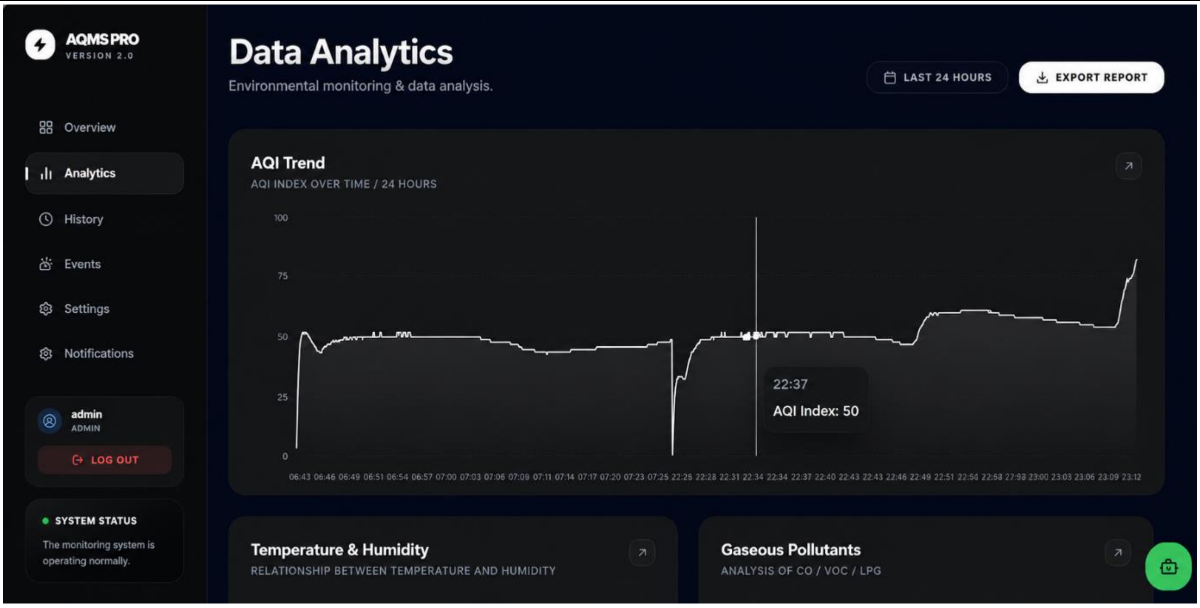

Figure 6: The Air Quality Monitoring System PRO

5.2. Future Research Trajectories

While the current infrastructure exhibits high stability, future milestones will focus on:

1. *Edge-level Predictive Analytics:* Implementing lightweight Long Short-Term Memory (LSTM) machine learning neural networks directly within the backend architecture to model and forecast AQI trends up to 24 hours in advance.

2. *Energy Minimization Topologies:* Re-engineering the firmware execution pipeline to utilize advanced deep sleep cycles within the ESP32 chip alongside photovoltaic harvesting micro-circuits, enabling autonomous remote deployments independent of standard electrical grids.

REFERENCES

1. Government of Vietnam, “Decree No. 13/2023/ND-CP on Personal Data Protection,” 2023.

2. R. Buyya, C. Vecchiola, and S. T. Selvi, Mastering Cloud Computing: Foundations and Applications Programming, Morgan Kaufmann, 2013.

3. A. McEwen and H. Cassimally, Designing the Internet of Things, Wiley, 2013.

4. S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed., Pearson, 2021.

5. C. Alippi, “Intelligence in Cyber-Physical Embedded Systems,” IEEE Transactions on Instrumentation and Measurement, vol. 55, no. 4, pp. 1110-1119, 2006.

6. Ministry of Information and Communications, Vietnam, “Annual Report on National Cybersecurity and IoT Standards Development,” 2025.

7. Nguyen, H. V., and Tran, D. T., “Development of low-cost multi-parametric sensor nodes using ESP32 for real-time micro-climate tracking,” IEEE Sensors Journal, vol. 24, no. 8, pp. 11204-11215, 2024.

8. Wang, J., and Liu, X., “Characterization and dynamic humidity compensation models for metal-oxide semiconductor gas sensors in industrial environments,” Analytica Chimica Acta, vol. 1241, pp. 340-352, 2025.

9. Al-Khafajiy, M., and Baker, T., “Performance optimization of relational databases for high-frequency time-series IoT data ingestion,” Journal of Cloud Computing, vol. 12, no. 1, pp. 45-58, 2023.

10. Fielding, R. T., “Architectural Styles and the Design of Network-based Software Architectures: Re-evaluated for Modern RESTful Cloud APIs,” ACM Transactions on Internet Technology, vol. 24, no. 2, pp. 15-32, 2024.

11. Brown, T. B., et al., “Contextual Injection and Dynamic Prompt Engineering in Cyber-Physical Systems: Converting Sensor Telemetry into Cognitive Action,” Artificial Intelligence Review, vol. 57, no. 3, pp. 289-311, 2024.

12. Martinez, C., and Zhang, Y., “Bridging the Semantic Gap in IoT Ecosystems through Large Language Models: A Health and Safety Perspective,” International Journal of Medical Informatics, vol. 182, pp. 105-119, 2025.

13. K. Saito and J. Miller, “Hybrid Containerization with Capacitor: Maintaining Kernel-level Background WebSockets for Real-time OS Push Notifications,” IEEE Software, vol. 43, no. 2, pp. 64-72, 2026.

14. Flanagan, D., JavaScript & Frameworks: Deep Dive into Component-Based Architectures with React and Vite, O'Reilly Media, 2025.