iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XI **Month of publication:** November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75394

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

Lab Autonomy: IoT-Based Smart Laboratory Control and Monitoring System

Mr. Pradeep[1], Rishika[2], Aditya Kumar Choubey[3], Piyush Bhandari[4], Tarun[5]
[1, 2, 3, 4 ,5]*Department of Computer Science & Engineering, HMR Institute of Technology and Management*

Abstract: ***Manual laboratory operation is costly, slow, and difficult to audit. This paper introduces Lab Autonomy, an architecture that combines ESP32-based edge control, a secure Node.js WebSocket core, and a dual-database design (PostgreSQL for identity and audit; Influx DB for metrics) behind a Next.js dashboard. We also expose an OPC UA interface to align with industrial interoperability. In evaluations, round-trip command latency averaged 110 m-s under favorable network conditions, the control loop consistently met a sub-200 m-s target, and prototype scaling to 50 concurrent devices showed predictable, manageable latency growth with modeling up to 100 devices per server. Scenario modeling further indicates 15–18% energy savings from scheduling, occupancy cues, and closed-loop verification. These results indicate that low-cost edge hardware, when paired with persistent messaging and principled data separation, can deliver a secure and extensible platform for lab management.***

Keywords: IoT; laboratory automation; WebSocket; OPC UA; ESP32; time-series analytics; PostgreSQL; Influx DB; RBAC; Next.js.

I. INTRODUCTION

A. *Motivation and Context*

Laboratories in universities and research centers are still largely managed by on-site personnel with physical switches and ad-hoc routines. Such practices constrain flexibility, weaken situational awareness, and undermine energy stewardship. We propose Lab Autonomy, a system that blends embedded actuation, secure networking, and a unified web interface to deliver responsive, auditable control at scale.

B. *Gaps in Legacy Operations*

1) Inefficient Use of Power: Without automation, lights, fans, and HVAC frequently remain on. Our design targets 15–18% savings by combining schedules, occupancy sensing, and verified state feedback.
2) Limited Remote Reach: Physical presence remains a bottleneck. A browser-based dashboard with a persistent channel enables off-site control and monitoring.
3) Poor Observability: Manual logs are rare; structured traces are essential for future predictive analytics. Continuous telemetry and action logging close this gap.
4) Weak Accountability: Traditional setups seldom enforce identity, roles, and immutable records. We employ token-based access control and append-only logs to improve traceability.

C. *Objectives and Scope*

Our goals are fourfold: (i) provide sub-200 m-s end-to-end actuation feedback; (ii) split security-critical transactions from high-velocity metrics; (iii) offer a map-driven UI with real-time indicators; and (iv) expose OPC UA semantics to interoperate with SCADA/MES ecosystems.

II. ITERATURE REVIEW

A. *Edge Controllers and Low-Latency Actuation*

The ESP32 microcontroller is a pragmatic choice for Wi-Fi connectivity and modest on-device computation. In our system, ESP32 boards drive opto-isolated relays to switch mains loads while maintaining electrical isolation and safety margins. The firmware maintains persistent connectivity and executes commands deterministically.

B. Persistent Messaging for Control

Polling over HTTP introduces handshake overhead and latency variance unsuitable for timely actuation. WebSocket provides a single full-duplex channel, improving responsiveness and enabling immediate state confirmation from edge to UI.

C. Industrial Interoperability with OPC UA

OPC UA offers secure transport, rich information modeling, and broad industrial adoption. By mapping device states and sensor values to OPC UA nodes, data produced by low-cost edge devices becomes consumable by SCADA/MES layers without bespoke adapters.

D. Data Separation for Integrity and Throughput

Relational databases (PostgreSQL) provide ACID semantics for identities, roles, and audit logs, whereas time-series engines (Influx DB) are optimized for high-rate inserts and window aggregations. Functional separation prevents contention between security-critical transactions and analytics workloads.

E. Frontend Frameworks for Real-Time Dashboards

Next.js and React support composition of live widgets, server-assisted rendering, and incremental updates. These traits are well matched to a floor-map view with per-device state, alerts, and historical charts.

III. PROPOSED SYSTEM / METHODOLOGY

A. Four-Layer Architecture

1) Hardware and Edge Layer: ESP32 nodes actuate relays and sample sensors (DHT22 for temperature/humidity, ACS712 for current, PIR for occupancy). Firmware handles Wi-Fi, message parsing, and safe switching.

2) Communication and Middleware: A Node.js server (w s) terminates secure WebSocket sessions, authenticates users, enforces role policies, routes commands to devices, and accepts telemetry. An OPC UA mapping layer exposes standardized nodes for enterprise tools.

3) Data Persistence: PostgreSQL stores users, roles, sessions, and an append-only log of actions. Influx DB ingests time-stamped states and environmental metrics for trend analysis.

4) Frontend Presentation: A Next.js dashboard renders a lab map with per-device toggles and live status. Historical panels query time-series data for analytics and anomaly inspection.

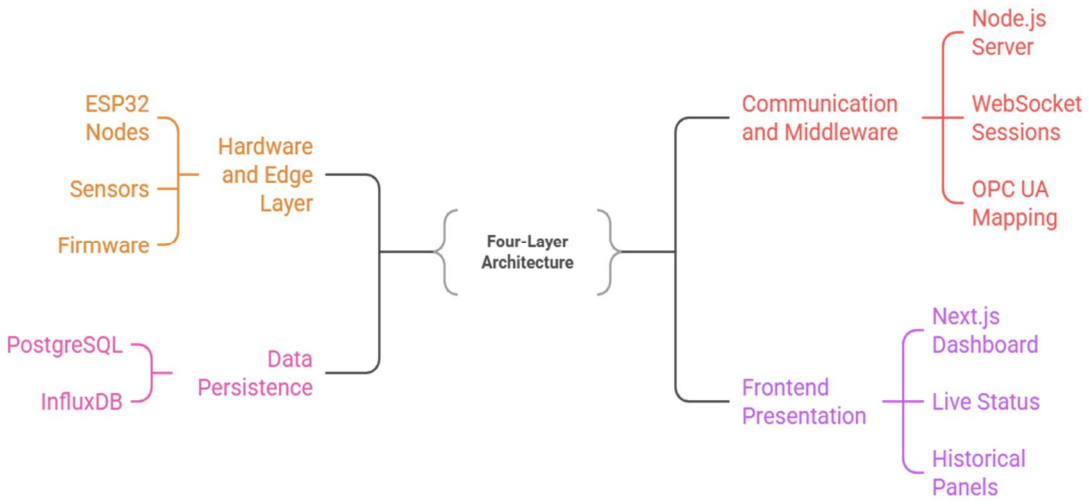


Fig. 1. System architecture of Lab Autonomy

B. Bi-Directional Communication

WebSocket enables low-latency control and feedback among the browser, middleware, and ESP32 clients. Under normal conditions, average round-trip latency is ~110 m-s, while core control latency remains beneath 200 m-s. OPC UA broadens applicability by standardizing access to data and methods.

C. Data Responsibilities and Flows

Action requests traverse an authenticated path to the target device. Upon actuation, the device returns its state and metrics. The middleware persists a signed audit record in PostgreSQL and streams metrics to Influx DB. The verified state is then emitted to subscribed clients to keep UI and physical state aligned.

IV. IMPLEMENTATION

A. Hardware Configuration

ESP32 Dev Kits run C++ (Arduino toolchain). Relay boards (5 V, opto-isolated) provide galvanic separation and switch the live conductor of mains circuits. Sensor channels report thermal, humidity, current, and occupancy signals used for automation policies and dashboards.

B. Middleware and Routing Logic

A Node.js/Express service backed by the w s library hosts a wss:// endpoint. Downlink messages encode device-specific actions; uplink messages carry state and metric payloads. Role-based checks are evaluated prior to emission, and malformed or unauthorized requests are dropped with structured errors.

C. Datastores and Schemas

PostgreSQL persists the Users, Roles, Sessions, and Logs tables with foreign keys and timestamps. Influx DB uses measurement, tags, fields, and time to represent device states and sensor series. This split maintains transactional guarantees while supporting efficient time-window queries.

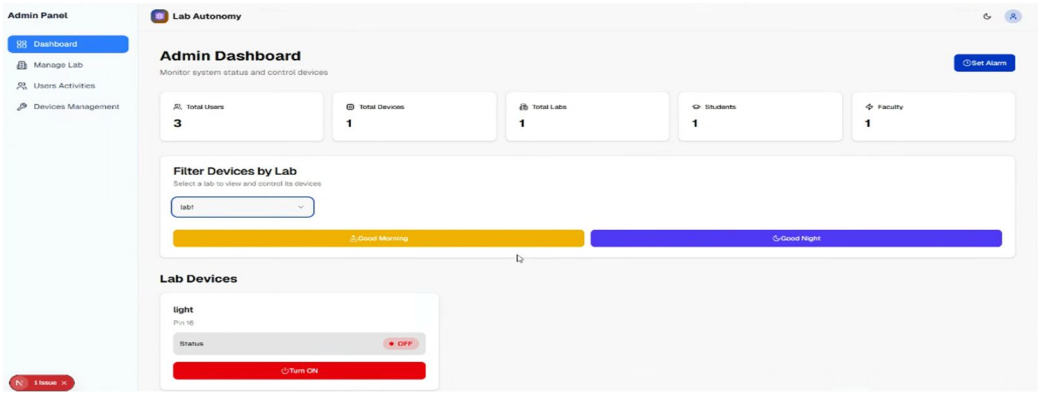


Fig. 2. Frontend Snippet

D. Frontend and User Experience

The dashboard emphasizes spatial cognition via an interactive map. Live badges reflect device states; charts visualize usage and environment trends. Client-side WebSocket subscriptions update components immediately after confirmed writes.

Fig. 2. Prototype Implementation.

V. RESULTS AND DISCUSSION

A. *Latency and Responsiveness*

We measure end-to-end latency from button press to physical actuation and UI confirmation. The mean was 110 m-s core control latency remained under 200 m-s. Persistent transport and minimized parsing on the edge contribute to tight feedback.

B. *Concurrency and Scaling Behavior*

With 50 simultaneous devices, latency growth was near linear, rising from ~120 m-s at light load to ~550 m-s. Modeling indicates that up to 100 devices per server remain feasible before architectural sharding becomes attractive.

C. *Integrity, Observability, and Analytics*

Segregating logs from metrics preserved authentication performance even under heavy telemetry. Typical time-range queries against Influx DB executed in ~15 m-s, sufficient for smooth charts and threshold alerts without starving transactional work.

D. *Operational Impact*

Combining schedules, occupancy, and verified state reduced waste in modeled scenarios by 15–18%. Immutable logs in PostgreSQL enhance accountability and form a foundation for compliance reporting.

E. *Limitations and Threats to Validity*

Results reflect controlled network conditions; highly congested or lossy links will degrade latency. Hardware variation and relay wear-out can affect actuation consistency. Future deployments should incorporate redundancy, watchdogs, and periodic calibration.

VI. CONCLUSION AND FUTURE WORK

A. *Contributions*

Lab Autonomy shows that persistent messaging, principled data separation, and OPC UA semantics can elevate laboratory operations with modest hardware. The system delivers sub-200 m-s feedback, clean audit trails, and a path to industrial interoperability.

B. *Future Directions*

We plan to

1) Train predictive models for maintenance using accumulated telemetry;
2) Introduce adaptive energy policies linked to weather and occupancy; and
3) Complete richer OPC UA mappings to integrate directly with SCADA/MES at scale.

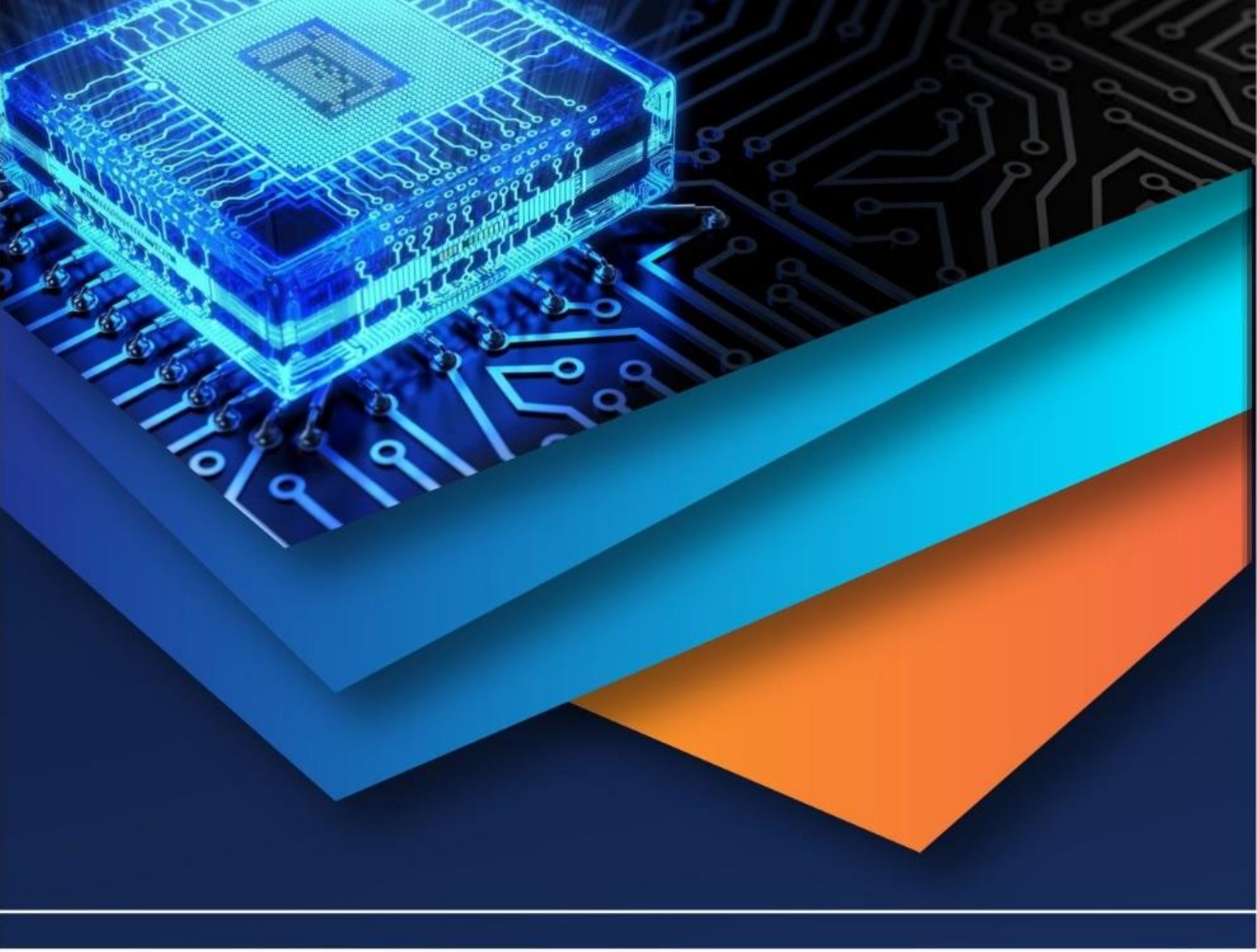
VII. ACKNOWLEDGMENT

The success and outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. I wish to express our gratitude to Prof. (Dr.) V. C Pandey Director, HMRITM Delhi for providing the facilities of the Institute and for his encouragement during the course of this work. I also express our deep indebtedness to our mentor, Mr. Pradeep (Assistant Professor, CSE) for his guidance and constant supervision as well as for providing necessary information regarding the project in addition to offering her consistent support in completing the project. Finally, we wish to thank our family members and our friends who have always been very supportive and encouraging.

REFERENCES

[1] Sharma et al., “IoT-Based Real-Time Automation using ESP32 Microcontroller,” IEEE Access, 2022.
[2] R. Singh and N. Patel, “Implementation of OPC UA for Secure Industrial IoT Communication,” Springer IoT Journal, 2023.
[3] M. Rahman, “Time-Series Data Handling in IoT Systems using Influx DB,” Elsevier Procedia Computer Science, 2021.
[4] V. Gupta et al., “Real-Time Data Streaming using WebSocket and Node.js for IoT,” in Proc. IEEE Internet of Things Conf., 2022.
[5] Y. Zhou, “Design and Optimization of IoT Dashboards using React and Next.js,” ACM Computing Surveys, 2023.
[6] K. Bhattacharya, “Dual-Database Architecture for Scalable IoT Applications,” Journal of Emerging Technologies, 2022.
[7] A. P. L. Karupaiya et al., “IoT-based Smart Laboratory System,” Int. J. Eng. Res. & Technol. (IJERT), vol. 9, no. 1, 2020.
[8] Espressif Systems, ESP32 Technical Reference Manual, 2023.
[9] Influx Data, Influx DB Documentation, 2024.
[10] OPC Foundation, Introduction to OPC UA, 2023.

[11] Node.js Foundation, "WebSocket Implementation using w s," 2024.
[12] Springer, "Suitability of Influx DB for IoT Applications," 2023.
[13] GitHub Repositories, "ESP32 OPC UA Client and Server Implementations," 2024.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY