2025-05-21

Supplementary Material: A Project Report on “An Affordable IoT and Cloud-Based System for Real-Time Indoor Air Quality Monitoring in Educational Environments: A Case Study from Bangladesh”

Zihan, Tahfizul Hasan

Supplementary Material: A Project Report on "IoT-Driven Cloud-based Approach for Comprehensive Indoor Air Quality Monitoring Device in Educational Environments"

Prepared by: Tahfizul Hasan Zihan (Undergraduate, CSE), Noureen Islam (Lecturer C, CSE), Noor-E Sadman (Lecturer C, CSE), Hasin Mahir (Undergraduate, CSE).

Acknowledgements

We gratefully acknowledge the unwavering support of FabLab IUB for tools and workspace support. The guidance received from the members of this lab made the rapid prototyping of our sensor nodes possible. Special thanks go to Dr. Ferdows Zahid for his insightful technical advice and to Dr. Mahady Hasan for his constant encouragement and constructive feedback throughout the project.

Table of Contents

Chapter 1: Overview

The main concern of this project is to focus on IoT based devices and to distinguish if the IoT based device is as equal as the industrial graded device, by looking into the data and evaluating accuracy of the device that we have made, and provide a low cost solution. Collation of data is determined to be done to examine the accuracy of the data retrieved from the IoT based device. Errors are measured and compared to justify that our IoT based device is as good as the industrial ones. Sidewise, heading to the goal of finding an inexpensive solution, we will be learning more about the advanced air monitoring technology used by AirVisual Pro and analyze the data to determine the accuracy of the device we built.

This document provides additional technical details to complement the main manuscript. It includes a detailed bill of materials (BOM) for the low-cost indoor air quality (IAQ) sensor nodes, descriptions of the sensor node's physical design and the classroom deployment setup (including exact dimensions and placement heights), a schematic layout of the classroom with sensor locations and ventilation features, an overview of the software system (particularly the MQTT–Node-RED data pipeline with JSON processing logic and database insertion), and a summary of debugging challenges encountered during deployment (with their resolutions). Where relevant, insights from earlier project stages and manuscript versions are incorporated to enrich the methodology and design narrative.

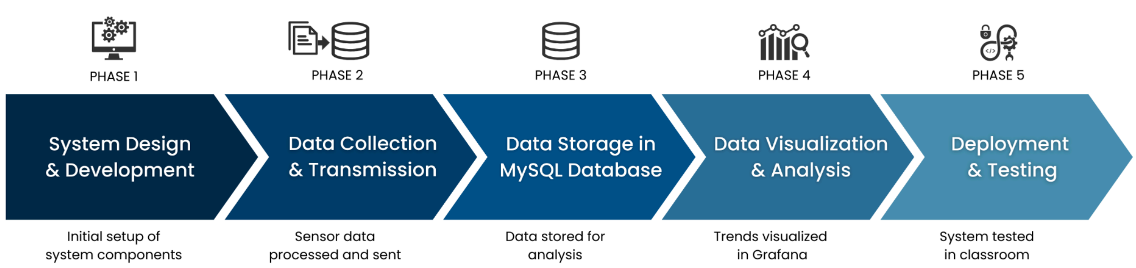

Figure 1: Step-by-step development process of the IoT-based indoor air quality monitoring system.

1.1 Cost Breakdown for IAQ Sensor Nodes

Each sensor node was constructed using affordable, commercially available components. Table 1 below presents the full BOM, listing each component, the specific model or description, and the approximate unit price in USD. The total cost per node comes to about $94.35, which demonstrates the cost-effectiveness of the system compared to professional-grade IAQ monitors (often costing several times more). All prices reflect the market rate at the time of development. Minor items such as a custom PCB, wiring, and enclosure hardware are included in the total.

Table 1: Cost Breakdown for each IAQ sensor node, listing all components, models, and approximate prices.

Component	Model/Description	Price (USD)
Power Module	Hi-Link AC–DC 220V-to-5V Module (20 W)	$2.50
DC Fan (5 V)	5 V axial cooling fan (32 dB quiet)	$1.00
Temperature & Humidity Sensor	Seeed Grove SHT35 (I2C probe)	$13.95
Particulate Matter Sensor	Plantower PMS5003 (PM_1.0/2.5/10)	$20.70
CO_2 Sensor	MH-Z19B NDIR module	$23.20
Microcontroller + WiFi	Keyestudio **Arduino Mega 2560** + ESP8266 ("Mega Plus WiFi")	$25.00
PCB and Circuit Board	Custom prototyped PCB	$3.00
Wiring and Connectors	Jumper wires, connectors (misc.)	$2.00
Enclosure & Fittings	Device box, mounting hardware	$3.00
Total per node	—	**$94.35**

1.2 Schematic Diagram

The overall cost of $94.35 per node makes widespread deployment feasible in resource-constrained educational environments. Each node uses a Hi-Link AC–DC isolated power converter to step down mains 220 V AC to 5 V DC, supplying a Keyestudio Arduino Mega 2560 Plus WiFi board. This microcontroller board integrates an ESP8266 WiFi module, allowing the node to publish data wirelessly. The environmental sensors include an optical dust sensor (Plantower PMS5003) for particulate matter (PM1.0, PM2.5, PM10), an NDIR-based CO_2 sensor (MH-Z19B), and a digital temperature/humidity sensor (Grove SHT35). A small 5 V DC axial fan (rated at only ~32 dB noise) is installed in each enclosure to actively draw air through the sensor chamber. This active sampling ensures more representative air quality readings and also serves to dissipate heat from the electronics, improving sensor stability. All core components were selected to balance affordability, local availability, and sufficient accuracy for indoor environmental monitoring. The total parts cost of under $100 per device is significantly lower than that of typical commercial IAQ monitoring units, yet still provides multi-pollutant sensing and real-time data capabilities.

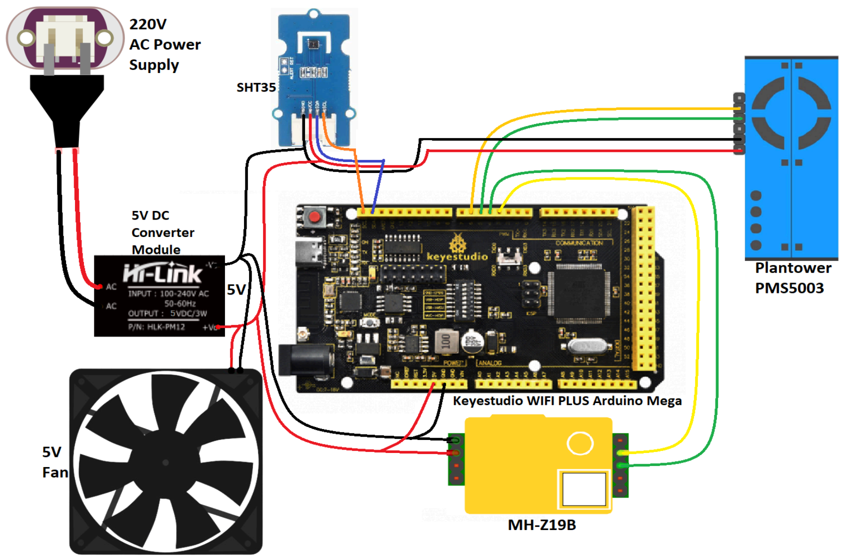

Figure 2. Circuit schematic diagram illustrating sensor integration, microcontroller connections, and power supply design.

1.3 Circuit Diagram

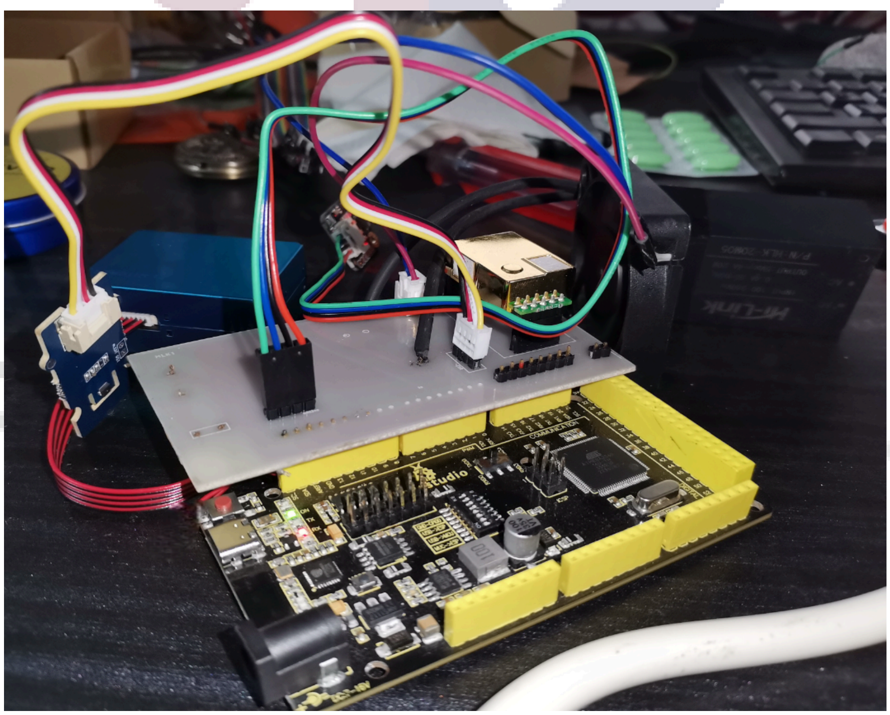

Figure 3. Assembled prototype circuit with complete wiring and all modules properly connected for testing and calibration.

1.4 PCB Design

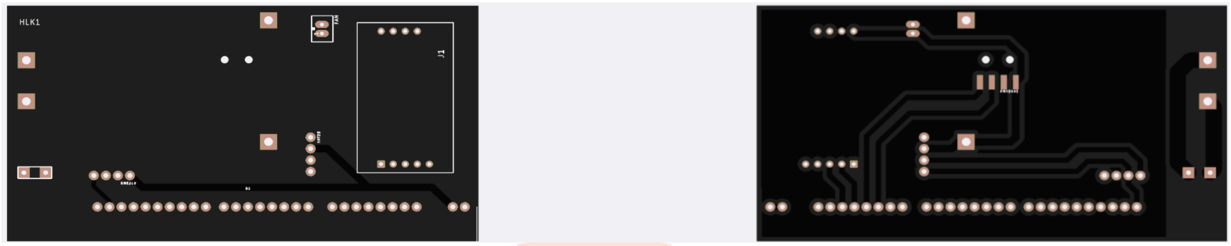

Figure 4. Custom PCB layout designed for the sensor node, showing component placement and routing for compact integration.

1.5 Node Architecture Overview

Each IAQ node follows a three-tier architecture that links **edge sensing**, **cloud processing**, and **user interaction** (Figure 5).

1. **Edge layer – Sensor node**: An Arduino Mega 2560 with an on-board ESP8266 handles sampling from the MH-Z19B CO_2, Plantower PMS5003 PM, and SHT35 temperature-humidity sensors. A 5 V axial fan drives a steady airflow across the sensor chamber while the Hi-Link AC–DC module supplies isolated power.
2. **Communication layer – Wi-Fi & MQTT**: Every 10s the ESP8266 publishes a JSON payload to a campus Wi-Fi router, which forwards the data via the lightweight MQTT protocol to a cloud server.
3. **Cloud / application layer**: The server ingests the messages with Node-RED, writes them to a MySQL time-series database, and streams them to a Grafana dashboard for live visualisation. A parallel Node-RED flow evaluates threshold rules and dispatches WhatsApp alerts when CO_2 or PM exceed preset limits, enabling real-time intervention.

This modular stack keeps the embedded code minimal, lets the heavy processing run in the cloud, and is easily scaled by adding more nodes or dashboards.

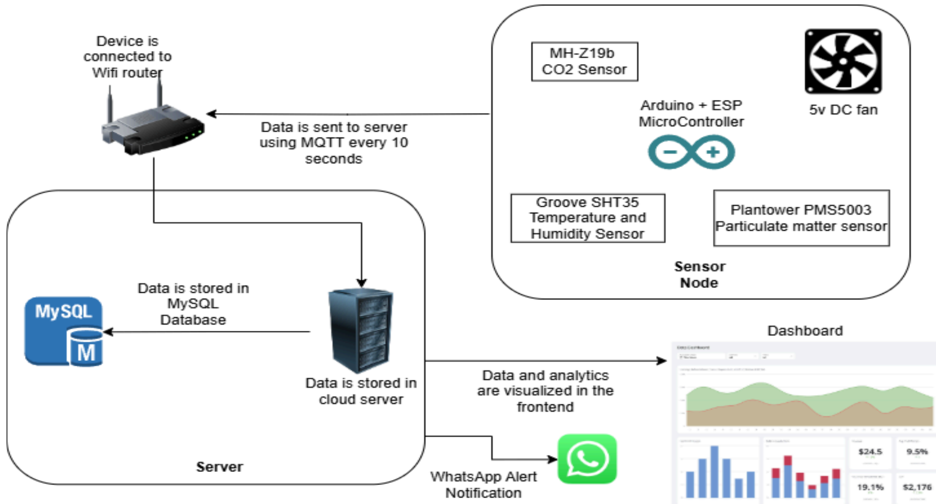

Figure 5. End-to-end system architecture of the IoT-driven IAQ monitoring platform.

Chapter 2: Sensor Node Design and Classroom Deployment

2.1 Device Enclosure: Physical Design

Each sensor node's components are housed inside a robust, lockable enclosure to protect against tampering or accidental damage in the classroom. The enclosure is a box of approximately 18 × 18 × 13 cm in size, with a custom 3D-printed ABS inner mount holding the sensors and circuitry. The enclosure material (metal for the outer lockbox with vents, and ABS plastic for internal mounts) was chosen to be durable and to allow airflow. The axial intake fan is mounted on one side of the box to continually pull room air over the CO_2 and dust sensors. The fan's airflow (a gentle, near-silent stream) ensures that each node is sampling the ambient air rather than air that might be heated by the electronics. A small perfboard PCB inside connects the sensors to the Arduino Mega, and the Hi-Link AC–DC module is shielded and isolated in the enclosure to minimize electromagnetic interference to the WiFi module. For safety and ease of deployment, the node's enclosure is fixed inside a wall-mounted iron cage with a lock – this metal cage (with mesh openings for ventilation) has the same external dimensions (~18 cm cube) and prevents theft or vandalism while still allowing air to circulate freely through the sensors.

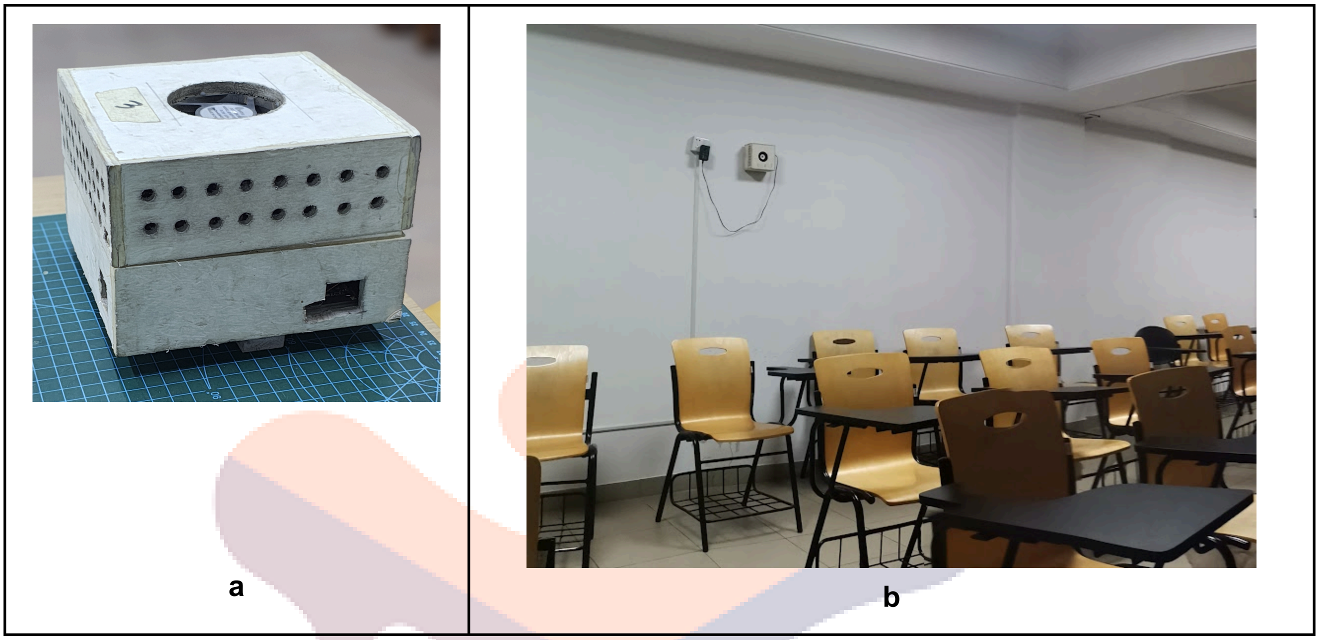

Figure 5. (a) Exterior view of the finalized device node enclosure; (b) Device node securely deployed in a classroom environment for real-time IAQ monitoring.

2.2 Classroom Environmental Characteristics

The IAQ system is deployed in a closed, lecture-style university classroom measuring 12.30 m × 6.95 m with a 3.10 m floor-to-slab height (a false ceiling at 2.90 m creates a shallow plenum). The room has no operable windows or mechanical ventilation; its only fresh-air path is a 21.4 cm × 46.8 cm louvered grille cut into the lower section of the east-wall door (shown in Figure 6). A fixed curtain-wall window and a wall-mounted AC unit occupy the north wall, but the AC recirculates indoor air and was used only intermittently during monitoring. Consequently, the classroom behaves as a minimally ventilated enclosed space, allowing clear observation of CO_2 accumulation and particulate build-up during occupancy.

Figure 6. Classroom entrance door featuring a built-in louvered ventilation grille, with a sensor node installed just above it for capturing air exchange data.

3.3 Sensor Node Placement and Mounting

Ten identical IAQ nodes now form a full-room sensing grid (Table 2 and Figure 7).
The first four units (D-1 to D-4) have been on the walls for more than five months, giving us a solid baseline of long-term trends. To sharpen the spatial picture, six new nodes (D-5 to D-10) were added at the remaining compass points only recently.

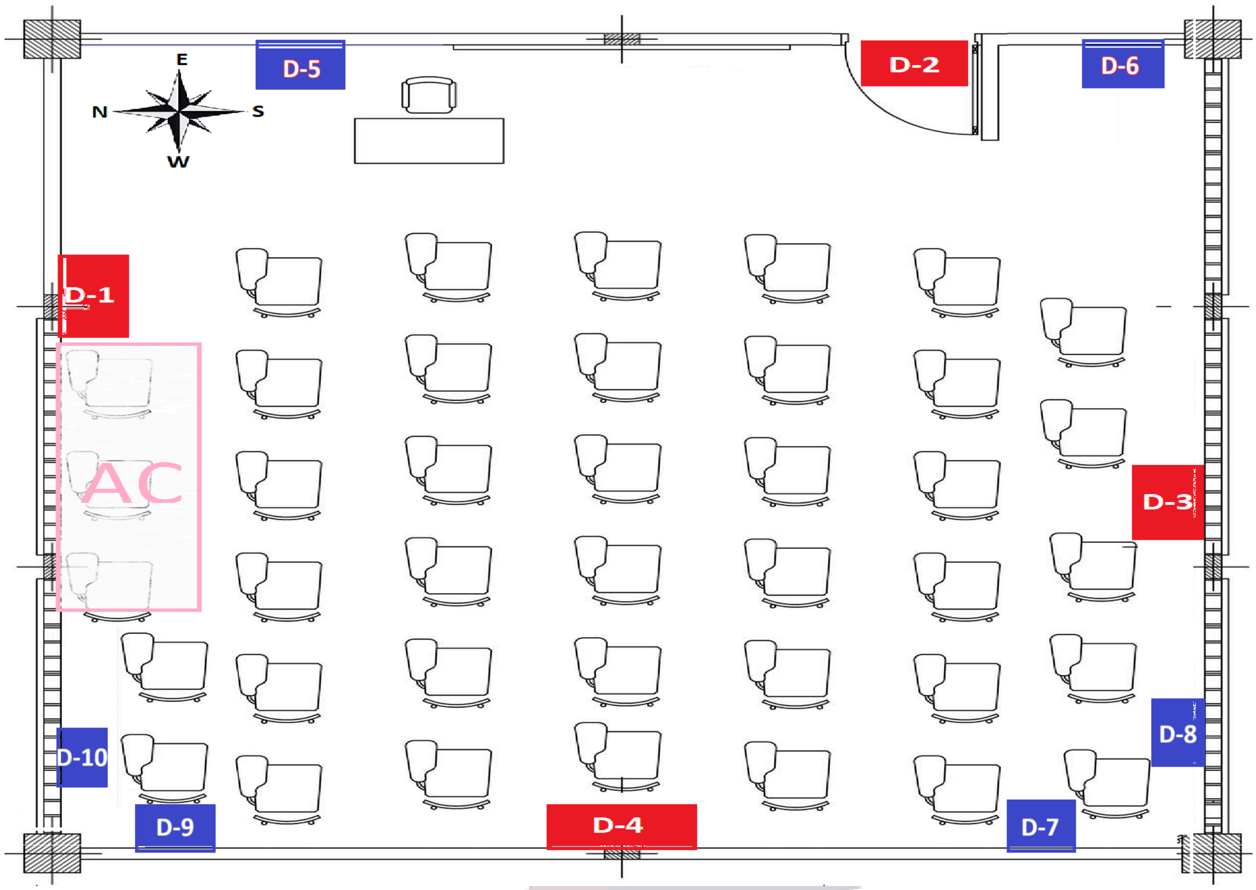

Figure 7. Classroom layout showing the deployment locations of previous four sensor nodes along with six more newly installed sensor nodes to capture spatial IAQ variations.

All devices are wall-mounted between roughly 2.0 m and 2.2 m above the floor—just below the false-ceiling line—so they stay clear of occupants yet remain in the breathing zone. The east-facing doorway (the room's sole opening) anchors the layout.

Table 2. Updated locations of the ten IAQ sensor nodes after the latest expansion.

Device ID	Compass tag	Wall description / notable feature
D-1	North	Fixed to the north wall, tucked just under the split-AC and beside the large curtained window.
D-2	East (door)	Perched directly above the only entry door and its louvered grille on the east wall
D-3	South	Mid-span of the south wall.
D-4	West	Centred on the west wall, facing the doorway across the room

D-5	East-North	Northern section of the east wall, near the front of the classroom above the whiteboard area
D-6	East-South	Southern section of the east wall, close to the south-east corner and door hinge side
D-7	West-South	Snug in the south-west corner on the west wall, just above the skirting trunking
D-8	South-West	Lower half of the south wall, a short distance from the west corner
D-9	West-North	Upper west wall, midway between the window bay and the rear corner
D-10	North-West	In the north-west corner, adjacent to both the fixed window and the AC unit

3.4 Sensor Calibration and Quality Assurance

To ensure reliable data, each sensor underwent calibration or quality checks before deployment:

MH-Z19B CO_2 Sensor: Each CO_2 sensor was manually calibrated using the “fresh air” method. The sensor was placed outside in open air for several minutes to reset its baseline, aligning its zero point to the global background level (~400 ppm). This step helps eliminate sensor drift and ensures that the indoor CO_2 readings reflect real concentrations (e.g., 1000 ppm actually means 1000 ppm).
PMS5003 Particulate Sensor: Each PMS5003 was powered on for at least 30 seconds before logging data, letting the internal laser stabilize and minimizing noise. We also ran a HEPA-filtered “zero” check to confirm no false positives when no particles were present. This practice, recommended in recent studies, helps maintain accuracy and consistency, especially for $PM_{2.5}$ values below 50 µg/m³.
SHT35 Temperature/Humidity Sensor: The SHT35 sensors were factory-calibrated, but we additionally checked a sample batch against a reference thermometer and hygrometer. All units were within ±0.2 °C and ±2% RH of the reference, so no further adjustment was needed. Recent study shows these sensors can maintain mean errors under 0.05 °C and 1.2% RH under controlled conditions.
In addition to our standard sensor calibration steps, we validated the performance of our custom-built sensor nodes against a reference-grade instrument, the **IQAir AirVisual Pro**. This validation process involved parallel deployment of both our sensors and the reference device in the same environment and over the same time periods, allowing direct comparison of their outputs. By analyzing the degree of agreement between our sensors and the AirVisual Pro for key parameters such as $PM_{2.5}$, CO_2, and temperature, we were able to establish a practical baseline correlation.

Chapter 4: Software System and Data Pipeline

4.1 Sensor Data Acquisition and MQTT Transmission

Each IAQ sensor node collects data every 10 seconds, capturing measurements for CO_2 (ppm), particulate matter ($PM_{1.0}$, $PM_{2.5}$, PM_{10} in µg/m³), temperature (°C), and humidity (%RH). The Arduino Mega microcontroller aggregates these measurements into a lightweight JSON payload along with the device identifier and a timestamp (from either a real-time clock or a local counter). The payload is wirelessly transmitted via the integrated ESP8266 Wi-Fi module, utilizing the MQTT protocol due to its lightweight nature and efficient publish/subscribe messaging structure. Each node publishes to distinct MQTT topics (e.g., IAQ/deviceID/sensor) through the campus Wi-Fi network. A standard university router handles connectivity, with a dedicated mobile hotspot as a fallback to maintain reliability (details of network management are discussed in the troubleshooting section).

4.2 Cloud-Based Data Handling: Node-RED Processing

The MQTT messages sent from sensor nodes are received by a Mosquitto MQTT broker, hosted on a custom built cloud server in real-time. On this server, the open-source, flow-based programming environment Node-RED is used to ingest and process incoming data streams.
The primary functions of the Node-RED workflow include:

MQTT Subscription: Subscribing to MQTT topics to receive incoming sensor messages.
JSON Parsing: Decoding JSON-formatted strings into structured JavaScript objects.
Data Validation and Enrichment: Adding a reliable server-side timestamp (if missing from the device), validating sensor readings, and preparing data for database insertion.
Database Insertion: Writing structured data to a MySQL database with device identifiers, timestamps, and measurement values stored in clearly structured tables.

Node-RED was selected due to its simplicity, minimal code requirement, visual debugging capabilities, and ease of modification.

4.3 Database Storage and Real-time Visualization

The processed sensor data is stored in a MySQL time-series database on the same cloud server, chosen for robust handling of large volumes of timestamped sensor records and efficient querying for subsequent analysis. Data stored in MySQL is continuously visualized through Grafana, providing real-time dashboards showing live-updating graphs of key IAQ parameters such as CO_2, $PM_{2.5}$, temperature, and humidity. These visual dashboards enable immediate, intuitive understanding of current indoor air quality trends and conditions.
Live Data Dashboard:

A live dashboard showing real-time data & Graphs from the deployed device nodes is available at:
https://iaq-fablab-iub.netlify.app/

A semester long dataset used for analysis in our study can be accessed here:
Google Drive link

4.4 Automated Real-time Alert System

To proactively manage air quality, an automated alert system was implemented. A secondary Node-RED flow continuously monitors sensor data, activating alerts when pre-set thresholds are exceeded—for example, sustained CO_2 levels above 1000 ppm or $PM_{2.5}$ levels exceeding 150 µg/m³. Upon threshold breaches & for missing Nodes from Node-red, alerts are instantly delivered via the WhatsApp API, notifying the research team and facility manager, enabling rapid response actions such as opening doors or increasing ventilation.

4.5 Detailed Node-RED Flow Configuration

For transparency and reproducibility, Listing below provides the JSON export of the Node-RED configuration used in this project (see Figure 8 for visual reference). This structured flow includes:

MQTT Input Node: Subscribes to each device-specific MQTT topic.
JSON Parsing Node: Converts incoming MQTT payloads to structured JavaScript objects.
Custom Function Node (Process Data): Adds server-side timestamps and performs data preparation tasks.
MySQL Output Node: Executes database insertion queries into the configured remote database.
Here is the simplified example Node-RED JSON configuration (Figure 8 illustrates the whole Node-RED graphical flow).

Listing 1. Node-RED JSON flow:

```
[
  {
    "id": "e1f2a3b4c5d6",
    "type": "tab",
    "label": "Sensor Data",
    "disabled": false,
    "info": ""
  },
  {
    "id": "c0f1a2b3d4e5",
    "type": "mqtt in",
```

```
    "z": "e1f2a3b4c5d6",
    "name": "",
    "topic": "IAQ/device1/sensor",
    "qos": "2",
    "datatype": "auto",
    "broker": "mqtt_broker_config",
    "x": 150,
    "y": 100,
    "wires": [
      [
        "f1e2d3c4b5a6"
      ]
    ]
  },
  {
    "id": "f1e2d3c4b5a6",
    "type": "json",
    "z": "e1f2a3b4c5d6",
    "name": "",
    "property": "payload",
    "action": "",
    "pretty": false,
    "x": 350,
    "y": 100,
    "wires": [
      [
        "d4c3b2a1e5f6"
      ]
    ]
  },
  {
    "id": "d4c3b2a1e5f6",
    "type": "function",
    "z": "e1f2a3b4c5d6",
    "name": "Process Data",
        "func": "// Access sensor data\nvar deviceid = msg.payload.deviceid;\nvar temp = msg.payload.temp;\nvar hum = msg.payload.hum;\nvar pressure = msg.payload.pressure;\nvar altitude = msg.payload.altitude;\nvar pm1 = msg.payload.pm1;\nvar pm25 = msg.payload.pm25;\nvar pm10 = msg.payload.pm10;\nvar co2 = msg.payload.co2;\n\n// Example: Add timestamp\nmsg.payload.timestamp = new Date().toISOString();\n\n// Example: Log temperature above threshold\n\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
```

```
    "finalize": "",
    "libs": [],
    "x": 550,
    "y": 100,
    "wires": [
      [
        "b5a4e3d2c1f0"
      ]
    ]
  },
  {
    "id": "b5a4e3d2c1f0",
    "type": "debug",
    "z": "e1f2a3b4c5d6",
    "name": "Debug Output",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 750,
    "y": 100,
    "wires": []
  },
  {
    "id": "mqtt_broker_config",
    "type": "mqtt-broker",
    "name": "MQTT Broker",
    "broker": "103.237.39.27",
    "port": "1883",
    "clientid": "",
    "usetls": false,
    "protocolVersion": "4",
    "keepalive": "60",
    "cleansession": true,
    "birthTopic": "",
    "birthQos": "0",
    "birthPayload": "",
    "birthMsg": {},
    "closeTopic": "",
    "closeQos": "0",
```

```
      "closePayload": "",
      "closeMsg": {},
      "willTopic": "",
      "willQos": "0",
      "willPayload": "",
      "willMsg": {},
      "sessionExpiry": ""
    }
]
```


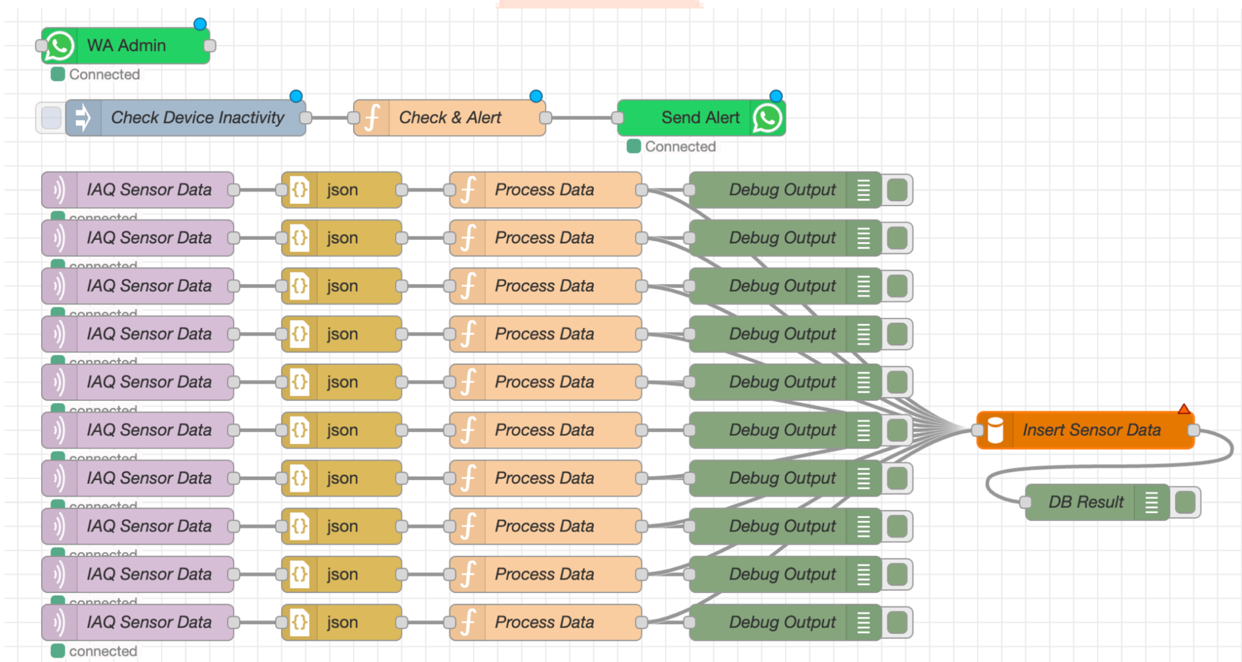

Figure 8. Node-RED data flow configuration used for MQTT data ingestion, JSON parsing, and insertion into the MySQL time-series database.

Chapter 5: Deployment Challenges and Debugging Insights

5.1 Data Transmission Issues and Network Overload

When we initially set up the IAQ monitoring system, With four nodes the network handled traffic comfortably, but expanding to ten revealed frequent data losses and delays. Sometimes the sensor data wouldn't reach the server, or messages arrived late, causing gaps in our dataset.
We discovered the main reason was network congestion: the classroom Wi-Fi was getting crowded, with multiple devices competing to send data simultaneously. The metal cases around our devices and thick classroom walls further weakened the Wi-Fi signals, especially for devices located far from the door and router.

MQTT QoS 0 → 1 test: Changing from “fire-and-forget” (QoS 0) to acknowledge delivery (QoS 1) cut message loss from ≈8 % to <1 %, at the cost of ~15 % extra traffic. While rotating the cages and later adding a tiny external whip antenna increased RSSI by ~8 dB on the worst-case node and eliminated most dropouts.
We experimented with different solutions like adjusting MQTT settings, trying external antennas, and tweaking Wi-Fi configurations. Eventually, the best fix was simply dividing our sensors between two separate Wi-Fi networks. This dramatically reduced interference and stabilized data transmissions, significantly improving reliability.

5.2 Power Supply and Signal Quality Issues

We initially suspected the small power modules we used might be causing electrical noise and interfering with the sensors or Wi-Fi signals. To test this, we shielded one of the modules and added extra capacitors to smooth out the power supply. While these steps slightly improved the stability, they didn't completely solve our Wi-Fi problems, indicating that the primary issue was indeed with Wi-Fi signal strength rather than power stability.However, ensuring clean and stable power remains crucial for long-term reliability.

5.3 Sensor Maintenance and Reliability

Over months of continuous operation, regular sensor maintenance became essential to ensure accuracy. Every two weeks, we carefully cleaned the sensors to remove dust, especially the particulate matter (PM) sensors. We also checked the CO_2 sensor apertures to ensure they were unobstructed. This simple maintenance routine kept the sensors consistent and reliable.
Occasional unusual readings typically turned out to be brief local events (like cleaning activities or someone standing near a sensor), rather than sensor malfunctions. Regular checks and maintenance significantly reduced false alarms and data anomalies.

Chapter 6: Conclusion

This document has shown how we designed, built, and deployed our low-cost IoT-based indoor air quality monitoring system. Each node costs around $95, which makes it suitable for widespread use in classrooms, even with limited budgets. We've shared comprehensive details about our design choices, hardware components, and software setup.
Field data revealed CO_2 spikes above 4 000 ppm near the north wall during packed lectures and persistent $PM_{2.5}$ peaks at the doorway, proving that pollutants pool unevenly in a sealed room. These hotspots make a strong case for demand-controlled ventilation (DCV) for example, triggering an exhaust fan or simply propping the door open whenever CO_2 climbs past 1 000 ppm.
Despite encountering challenges such as Wi-Fi network congestion, signal interference, and regular maintenance requirements; we successfully overcame these through practical solutions like splitting network traffic and performing routine sensor cleaning. Our experience clearly

shows that air quality in an enclosed classroom varies significantly across the room, highlighting the importance of strategic sensor placement for accurate monitoring.

We hope these insights will guide others in developing similar affordable systems, making real-time air quality monitoring more accessible and effective in educational settings.