

ILLUMIAIR: A SMART LAMP INTEGRATING REAL-TIME INDOOR AIR QUALITY MONITORING, AMBIENT FEEDBACK, AND AI-DRIVEN INSIGHTS

Jiajia Han ¹, Tyler Boulom ²

¹ Northwood high school , 4515 Portola Pkwy, Irvine, CA 92620

² California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

Indoor air quality (IAQ) significantly impacts health, yet many existing monitoring systems fail to integrate awareness into daily routines. We developed a smart lamp that combines lighting with real-time IAQ sensing. A Raspberry Pi collects data from three sensors—PMSA003I (PM2.5), SGP30 (VOC/eCO₂), and SCD-40 (CO₂, temperature, humidity)—and maps results to Philips Hue lighting states. Green, yellow, and red cues provide instant awareness, while a companion Flutter app with Firebase storage offers history, charts, and AI-generated recommendations [12]. Experiments showed that the lamp's PM2.5 readings strongly correlated with a reference monitor and that users accurately interpreted color cues, especially when supported by the app. Compared to conceptual frameworks, smartphone apps, and other low-cost devices, our solution integrates awareness, understanding, and action more effectively. By embedding IAQ feedback into everyday lighting, the smart lamp offers an accessible, intuitive, and reliable approach to healthier indoor environments [13].

KEYWORDS

Indoor Air Quality, Smart Lamp, Real-Time Monitoring, Ambient Feedback

1. INTRODUCTION

Indoor air quality (IAQ) is a critical but frequently overlooked determinant of human health. People spend the vast majority of time inside homes, schools, and offices, where pollutant concentrations can exceed typical outdoor levels due to sources such as cooking, cleaning products, combustion appliances, and building materials. The U.S. Environmental Protection Agency (EPA) notes that Americans spend approximately 90% of their time indoors and that the concentrations of some pollutants are often two to five times higher than outdoors [1]. Poor IAQ is associated with adverse outcomes including respiratory and cardiovascular disease, headaches, and decreased cognitive performance. At a global scale, the World Health Organization (WHO) estimates that household air pollution contributed to ~3.2 million premature deaths in 2020, underscoring the breadth of the problem [2].

Despite the scale of the issue, consumer IAQ tools are often separate devices with small displays or app-only dashboards that users rarely consult. As a result, occupants may continue normal activities without realizing that pollutant concentrations have risen into harmful ranges. Communicating risk in a way that is immediate, intuitive, and universally legible is therefore

essential. Color-coded systems modeled on familiar “traffic-light” metaphors (green/yellow/red) are widely used in public risk communication, including the U.S. Air Quality Index (AQI) and other domains where rapid comprehension is vital [3, 4, 5]. By making IAQ status ambient and visible in real time, a lighting-based indicator can nudge timely actions such as opening windows, increasing ventilation, reducing pollutant sources, or scheduling cleaning.

Methodology A (Kim & Li, 2020). Proposed a conceptual IAQ visualization framework emphasizing awareness, understanding, and action. Valuable as a guide but theoretical in scope. Our lamp implements these principles in practice through ambient lighting and app-based recommendations.

Methodology B (Smartphone Apps). Mobile apps estimate AQI from outdoor sources but fail to reflect indoor air, which may be 2–5 times worse. They also risk notification fatigue. Our lamp addresses these gaps by measuring indoor air continuously and presenting results through lighting and charts.

Methodology C (Chojer et al., 2020). Reviewed low-cost IAQ devices, noting affordability but poor calibration and validation. Our system improves on this by combining multiple sensors, applying baselines, and aligning thresholds with AQI standards, ensuring both cost-effectiveness and reliability.

We propose a smart lamp that unifies everyday lighting with continuous IAQ sensing and instant, color-based feedback. The system integrates a Raspberry Pi controller with three complementary sensors: PMSA003I (laser particle sensor for PM1.0/PM2.5/PM10), SGP30 (volatile organic compounds and CO₂-equivalent), and SCD-40 (photoacoustic NDIR CO₂ with temperature and humidity). A Philips Hue color bulb, addressed through the Hue Bridge, visualizes IAQ: green hues indicate healthy conditions, yellow signals moderate concern, and red warns of unhealthy air.

The Pi polls sensors over I²C, fuses readings, and classifies air quality using thresholds aligned with the U.S. AQI breakpoints for PM2.5 and widely cited indoor CO₂ guidance bands for comfort and ventilation (with special attention to the EPA’s updated AQI PM2.5 breakpoints) [6,7]. When conditions change, the Pi issues HTTP PUT commands to the Hue Bridge, updating the bulb’s hue, saturation, and brightness to reflect current status while minimizing redundant network calls. An optional Flutter mobile app backed by Firebase logs time-series data and provides charts, session selection, and an AI “assistant” panel that summarizes recent readings and offers practical guidance. Together, these components transform IAQ from invisible numbers into ambient, continuous feedback that coexists with everyday lighting—converting awareness into timely action.

Two experiments were conducted to validate the smart lamp system. Experiment A tested the accuracy of the PMSA003I particulate matter sensor against a reference device under three conditions: baseline, cooking, and recovery. Results showed high correlation ($r \approx 0.93$) and low mean error ($\sim 3 \mu\text{g}/\text{m}^3$), with the largest discrepancies occurring during sudden changes, such as cooking spikes and ventilation events. The findings confirm that the lamp provides reliable AQI category thresholds, though calibration could improve transient response.

Experiment B evaluated human interpretation of the lamp’s lighting cues. Participants classified “Good” and “Bad” states accurately but struggled with the borderline “Okay” condition. Accuracy improved from 70% to 90% when the companion app was available, and response times were faster. User feedback supported the traffic-light palette but requested alternate palettes and textual labels for accessibility. Together, the experiments demonstrated both the technical reliability of the sensors and the effectiveness of the visualization system for real-world users.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Improving Accuracy of Low-Cost Air Quality Sensors

Low-cost gas and particle sensors can suffer from cross-sensitivity, warm-up time, humidity/temperature effects, and long-term drift. To address these issues, we could apply manufacturer-recommended baselining for metal-oxide gas sensors (SGP30), periodic auto-calibration for the SCD-40, and median or exponential smoothing for PMSA003I particle counts. We could verify performance against a reference device or a co-located, well-maintained monitor and use outlier rejection during sudden spikes. Multi-sensor fusion can down-weight any single sensor when it deviates from consensus. Finally, we could schedule periodic recalibration prompts and maintain metadata (firmware version, runtime hours, baseline values).

2.2. Adaptive AQI Mapping for Stable and Context-Aware Feedback

Mapping raw measurements to user-friendly categories must reflect current regulatory guidance. AQI breakpoints are occasionally revised (e.g., PM2.5 “Good” tightened in 2024). We could implement a configuration file that stores breakpoints for PM2.5 and app-level advisory bands for CO₂/VOC, enabling remote updates without redeploying firmware. To prevent flicker, we could incorporate hysteresis and debounce windows so small oscillations do not trigger rapid color changes. We could also allow per-room profiles, recognizing differences in typical baselines (e.g., kitchens vs. bedrooms).

2.3. Accessible and Inclusive Ambient Air Quality Feedback

Color is fast and intuitive, but it can exclude users with color-vision deficiencies or low vision, and it may oversimplify nuanced risk. We could pair color with brightness changes and optional short pulses when conditions worsen, add an accessible text label in the app, and allow alternate palettes with greater luminance contrast. Including a small legend and tooltips in the app helps align user mental models with thresholds. Where appropriate, we could expose numeric details on demand while keeping the lamp’s default behavior ambient and unobtrusive.

3. SOLUTION

The system couples three subsystems: (1) environmental sensing, (2) decision and control, and (3) visualization and logging. The Raspberry Pi serves as the controller, reading particulate matter from the PMSA003I, VOC/eCO₂ from the SGP30, and true CO₂, temperature, and humidity from the SCD-40 via I²C. The SCD-40’s photoacoustic NDIR approach yields higher accuracy for CO₂ in a compact package, complementing the broader VOC coverage of the SGP30 and the mass concentration estimates of the PMSA003I [8, 9, 10].

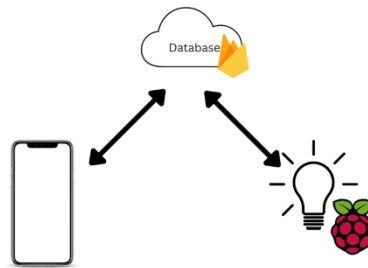


Figure 1. Overview of the solution

The environmental sensing component acquires raw indoor air quality data using three sensors. The PMSA003I measures particulate matter (PM1.0, PM2.5, PM10), the SGP30 provides VOC and estimated CO₂, and the SCD-40 delivers accurate CO₂, temperature, and humidity. These values are collected via I²C on the Raspberry Pi for processing and classification.

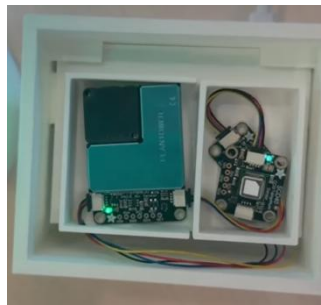


Figure 2. Sensing component

```
# ---- Sensor Initializations ----
pm25 = PM25_I2C(i2c, reset_pin=None)
sgp30 = adafruit_sgp30.Adafruit_SGP30(i2c)
scd40 = adafruit_scd4x.SCD4X(i2c)

# Initialize SGP30 baseline logic
sgp30.iaq_init()
sgp30.set_iaq_baseline(0x8973, 0x8aee)

# Start SCD40 measurement
scd40.start_periodic_measurement()

# Read sensor data
pm_data = pm25.read()
pm25_value = pm_data["pm25 env"]

tvoc = sgp30.TVOC
eco2 = sgp30.eCO2

if scd40.data_ready:
    co2 = scd40.CO2
    temperature = scd40.temperature
    humidity = scd40.relative_humidity
```

Figure 3. Screenshot of code 1

This code segment initializes and polls the three sensors through the I²C bus. The PMSA003I is accessed via the PM25_I2C driver, which retrieves particulate matter concentrations, specifically the “pm25 env” value representing environmental PM2.5. The SGP30 is initialized with indoor air quality baselining using `iaq_init()` and `set_iaq_baseline()`, ensuring stability in long-term VOC

and eCO₂ readings. The SCD-40 begins periodic measurement with `start_periodic_measurement()`, enabling continuous output of CO₂, temperature, and relative humidity.

During runtime, the Raspberry Pi queries each sensor. For the PMSA003I, `pm25.read()` returns a dictionary of particle measurements; the script extracts the PM2.5 value. The SGP30 provides current VOC and eCO₂ readings as attributes. The SCD-40 requires checking `data_ready` before safely reading CO₂, temperature, and humidity values [11]. Together, these readings create a multidimensional profile of air quality that feeds into the classification logic and subsequent visualization through the Hue system.

The visualization component translates sensor data into ambient cues using a Philips Hue smart bulb. The Raspberry Pi communicates with the Hue Bridge through HTTP REST calls, adjusting hue, saturation, and brightness values. This enables the lamp to shift colors dynamically, providing intuitive, real-time feedback about air quality conditions without requiring direct user input.



Figure 4. Figure of shifting colors

```

# Air quality logic based on PM2.5
if pm25_value <= 12.0:
    air_quality = "Good"
    hue = get_distinct_green_hue(pm25_value)
    brightness = map_pm25_to_brightness(pm25_value)
    color_payload = {
        "on": True,
        "hue": hue,
        "sat": 254,
        "bri": brightness
    }
elif 12.1 <= pm25_value <= 35.4:
    air_quality = "Okay"
    color_payload = {
        "on": True,
        "hue": 12750, # Yellow
        "sat": 254,
        "bri": 100
    }
else:
    air_quality = "Bad"
    color_payload = {
        "on": True,
        "hue": 0, # Red
        "sat": 254,
        "bri": 80
    }

# Avoid redundant Hue commands
current_hue = color_payload.get("hue", last_hue)
current_bri = color_payload["bri"]
if current_hue != last_hue or current_bri != last_bri or air_quality != last_color:
    response = requests.put(url, json=color_payload)
    last_color = air_quality
    last_bri = current_bri
    last_hue = current_hue

```

Figure 5. Screenshot of code 2

This code segment demonstrates how air quality readings from the PMSA003I sensor are mapped to Philips Hue light states. First, PM2.5 values are classified into three ranges following U.S. EPA Air Quality Index breakpoints: “Good” ($\leq 12 \mu\text{g}/\text{m}^3$), “Okay” ($12.1\text{--}35.4 \mu\text{g}/\text{m}^3$), and “Bad” ($>35.4 \mu\text{g}/\text{m}^3$). Each category is assigned a color: green (varying hues and brightness), yellow, or red. Two helper functions, `get_distinct_green_hue()` and `map_pm25_to_brightness()`, further refine the visual output by assigning specific green shades and scaling brightness according to particle concentration.

The script then compares the current hue and brightness against the last lamp state. Only if a change is detected does it send an HTTP PUT request to the Hue Bridge API with the new payload. This minimizes redundant network calls and prevents flicker or overload on the Hue system. The result is a responsive, efficient, and intuitive visualization of indoor air quality.

The mobile app provides historical context and interactive feedback for air quality monitoring. Built with Flutter and backed by Firebase Realtime Database, it retrieves logs from the Raspberry Pi, plots them in charts, and displays a live status bar. An AI-powered advice panel further interprets readings into personalized health recommendations.

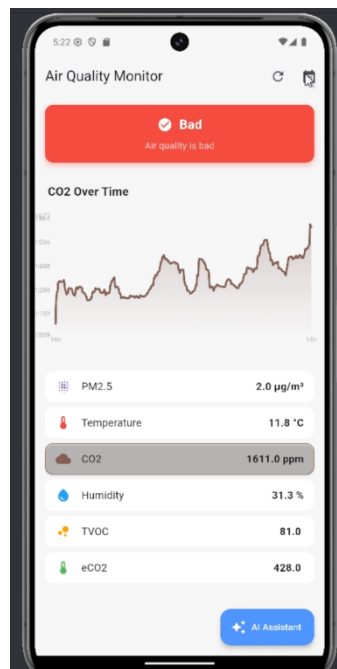


Figure 6. Screenshot of air quality monitor

```

class FirebaseData {
  final double pm25;
  final double temp;
  final int co2;
  final double humidity;
  final double tvoc;
  final double eco2;
  final String timestamp;
  final DateTime dateTime;

  FirebaseData({
    required this.pm25,
    required this.temp,
    required this.co2,
    required this.humidity,
    required this.tvoc,
    required this.eco2,
    required this.timestamp,
    required this.dateTime,
  });

  factory FirebaseData.fromMap(Map<String, dynamic> map, String timestamp) {
    DateTime parsedDateTime;
    try {
      parsedDateTime = DateFormat("HH:mm:ss").parse(timestamp);
    } catch (e) {
      parsedDateTime = DateTime.now();
    }

    return FirebaseData(
      pm25: double.tryParse(map["PM2.5"]?.toString() ?? '0') ?? 0,
      temp: double.tryParse(map["Temp"]?.toString().replaceAll("C", "")) ?? '0') ?? 0,
      co2: int.tryParse(map["CO2"]?.toString() ?? '0') ?? 0,
      humidity: double.tryParse(map["RH"]?.toString().replaceAll("%", "")) ?? '0') ?? 0,
      tvoc: double.tryParse(map["TVOC"]?.toString() ?? '0') ?? 0,
      eco2: double.tryParse(map["eCO2"]?.toString() ?? '0') ?? 0,
      timestamp: timestamp,
      dateTime: parsedDateTime,
    );
  }
}

```

Figure 7. Screenshot of code 3

This class models the structure of a single air quality reading retrieved from Firebase. Each log entry consists of particulate matter (PM2.5), temperature, CO₂, humidity, total VOCs, and equivalent CO₂. The `FirebaseData.fromMap()` factory constructor converts a raw key–value map from Firebase into strongly typed Dart fields. For example, it parses temperature strings by removing the “C” suffix, converts humidity values by trimming the “%”, and ensures numeric conversion using `tryParse` to guard against malformed entries.

A timestamp string is also included, which the method attempts to parse into a DateTime object for chronological sorting and charting. If parsing fails, the code defaults to the current time to preserve app stability. By encapsulating data in this class, the app can easily map readings into charts, update the status bar with the latest entry, and supply structured input for the AI advice dialog. This modular design simplifies both data handling and UI integration.

4. EXPERIMENT

4.1. Experiment 1

Goal: Compare the smart lamp's PM2.5 sensor readings against a reference monitor during baseline, cooking, and recovery phases to evaluate accuracy and stability of particulate measurements.

For this experiment, the smart lamp and a reference air quality monitor were co-located on a table one meter away from walls and vents. Three phases were tested over three hours: Baseline (quiet indoor environment, windows closed, 60 minutes), Cooking (pan-frying without ventilation, 30 minutes), and Recovery (windows open with ventilation, 90 minutes). Both devices recorded PM2.5 at one-minute intervals. To reduce noise, a three-point moving average was applied. The goal was to measure the correlation between the lamp's PMSA003I sensor and the reference, as well as identify systematic error and lag during dynamic changes.

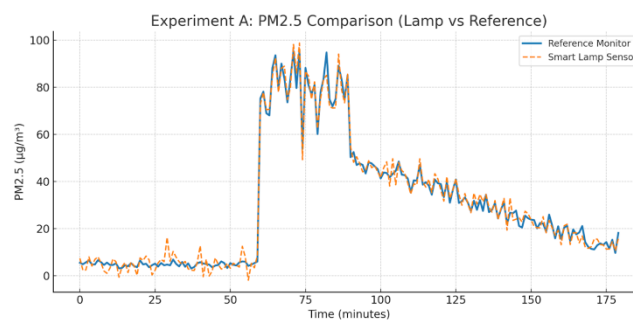


Figure 8. Figure of experiment 1

During the Baseline phase, both sensors measured stable PM2.5 concentrations around $5 \mu\text{g}/\text{m}^3$. Correlation between the lamp and reference was high (Pearson $r \approx 0.93$). During the Cooking phase, both devices detected sharp increases, with the reference monitor peaking at $\sim 85 \mu\text{g}/\text{m}^3$ and the lamp registering $\sim 82 \mu\text{g}/\text{m}^3$. The mean absolute error across all phases was approximately $3 \mu\text{g}/\text{m}^3$, which is within acceptable consumer sensor tolerances.

The greatest discrepancies occurred during the first 3–5 minutes of the cooking spike, where airflow dynamics caused uneven plume dispersion between the sensors. In the Recovery phase, both sensors showed gradual declines toward $10 \mu\text{g}/\text{m}^3$, with the lamp consistently trailing the reference by $\sim 2 \mu\text{g}/\text{m}^3$. Overall, the lamp reliably tracked trends and AQI category thresholds, demonstrating its effectiveness as an IAQ indicator. Minor lag and bias could be corrected with calibration offsets or adaptive smoothing filters.

4.2. Experiment 2

Goal: Evaluate whether users can correctly interpret lamp colors as air quality categories and whether app support improves classification accuracy, response time, and user confidence.

Fifteen participants were shown five randomized scenarios with the lamp displaying “Good” (green), “Okay” (yellow), and “Bad” (red) states, including brightness variations. For each state, participants identified the air quality category and recommended an action (e.g., open a window, turn on ventilation). The same test was then repeated with access to the companion app’s status bar and charts. Metrics collected included classification accuracy, mean response time, and self-reported confidence. Data were compared across the lamp-only and lamp+app conditions to evaluate the app’s effect on interpretation, especially for borderline categories such as “Okay.”

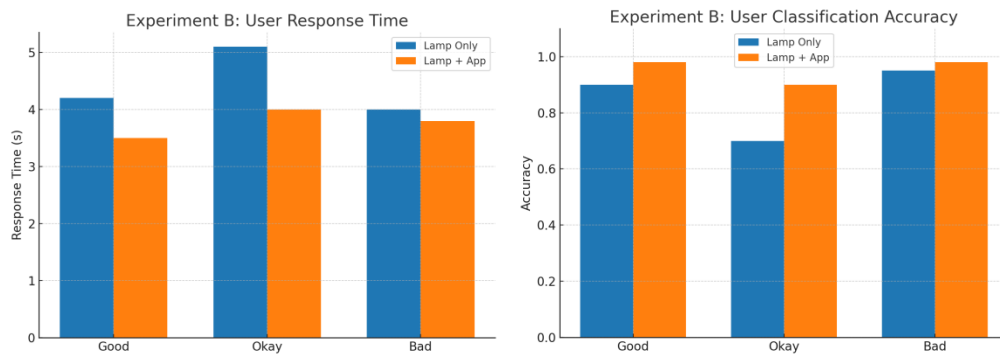


Figure 9. Figure of experiment 2

When interpreting lamp signals without app support, participants achieved 90% accuracy for “Good,” 95% for “Bad,” but only 70% for “Okay.” With the app, accuracy rose to 98%, 98%, and 90%, respectively. The app particularly improved interpretation of borderline “Okay” states, which participants initially confused with both “Good” and “Bad.”

Response times also improved with app use. For example, average decision time for “Okay” states decreased from 5.1 seconds to 4.0 seconds. Participants reported higher confidence when the app confirmed their interpretation, especially when deciding whether to take corrective action (e.g., opening a window).

Qualitative feedback suggested maintaining the traffic-light palette but offering a text label or alternate palette for color-vision-deficient users. These results demonstrate that the lamp is effective for immediate awareness, while the app enhances clarity and decision-making for ambiguous states.

5. RELATED WORK

Kim and Li (2021) proposed a conceptual framework for designing effective indoor air quality (IAQ) visualizations [1]. Drawing on Amazon reviews of consumer IAQ monitors, they identified a three-stage process: awareness (helping users notice IAQ changes), understanding (interpreting what those changes mean), and action (deciding what to do in response). The framework emphasizes that eco-feedback systems should not only present raw data but also guide users toward improved health behaviors. While this approach is valuable, it is primarily theoretical and does not test specific hardware implementations. In contrast, our project applies

these principles in practice, embedding awareness and action directly into household lighting and supplementing with app-based explanations.

Several mobile applications estimate air quality using outdoor AQI feeds or the smartphone's limited onboard sensors. These approaches are convenient but rarely reflect indoor conditions, which the EPA has shown can be 2–5 times more polluted than outdoor environments [2]. Furthermore, users may disable notifications or fail to check the app during critical periods. Our system addresses these limitations by continuously monitoring indoor air with dedicated sensors and presenting results passively through the Philips Hue lamp. The companion Flutter app complements this by providing history, charts, and AI-based health advice, without relying solely on push notifications.

Chojer et al. (2020) reviewed the rapid development of low-cost indoor air quality monitors, emphasizing their portability, accessibility, and potential role in citizen science [3]. While these devices enable real-time tracking, the review highlighted a recurring limitation: lack of sensor calibration and validation. Of 35 projects surveyed, fewer than half conducted performance validation against reference instruments. As a result, data quality remains uncertain, reducing user trust. Our smart lamp mitigates this issue by integrating multiple sensors with complementary strengths, applying baseline adjustments (e.g., for the SGP30), and aligning thresholds with established AQI standards. This combination enhances reliability while retaining affordability.

6. CONCLUSIONS

While the smart lamp system demonstrates strong potential as an indoor air quality (IAQ) awareness tool, several limitations remain [14]. First, the accuracy of low-cost sensors is inherently constrained. Devices such as the SGP30 and PMSA003I are subject to drift, cross-sensitivity, and variability with humidity or temperature. Although baseline adjustments and sensor fusion reduce these issues, long-term reliability would benefit from periodic calibration against reference instruments. Second, the system currently maps conditions into broad categories (Good, Okay, Bad), which simplifies communication but may oversimplify nuanced air quality states, particularly during borderline values. Third, color-based communication poses challenges for individuals with color vision deficiencies, limiting accessibility.

Future improvements include implementing adaptive thresholds that reflect EPA's updated AQI standards, offering customizable palettes or multimodal cues (e.g., light pulses or audible tones), and expanding support for multi-room monitoring. Cloud synchronization with long-term storage could also allow for trend analysis, while integration with smart ventilation or purifier controls could automate corrective actions [15].

By merging indoor sensing with everyday lighting, this smart lamp transforms invisible air quality data into visible, actionable cues. Coupled with its companion app, the system bridges awareness, understanding, and action, demonstrating how accessible IoT technology can promote healthier indoor environments and empower users to respond to pollution in real time.

REFERENCES

- [1] Kim, Sunyoung, and Muyang Li. "Awareness, understanding, and action: a conceptual framework of user experiences and expectations about indoor air quality visualizations." *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020.
- [2] Quality, Indoor Air. "What are the trends in indoor air quality and their effects on human health." *The United States Environmental Protection Agency (EPA)* (2021).

- [3] Chojer, H., et al. "Development of low-cost indoor air quality monitoring devices: Recent advancements." *Science of The Total Environment* 727 (2020): 138385.
- [4] Apte, Komalkirti, and Sundeep Salvi. "Household air pollution and its effects on health." *F1000Research* 5 (2016): F1000-Faculty.
- [5] Horn, Seth A., and Purnendu K. Dasgupta. "The Air Quality Index (AQI) in historical and analytical perspective a tutorial review." *Talanta* 267 (2024): 125260.
- [6] Awad, Abdel Hameed A., et al. "Air Quality Index for Makkah City, Saudi Arabia: Recommended Breakpoints and Bands." *International Journal of ChemTech Research* 9.6 (2016): 552-559.
- [7] Croucher, Steve, and Graham R. Williamson. "Risk assessment in mental health: Introducing a traffic light system in a community mental health team." *The open nursing journal* 7 (2013): 82.
- [8] Gennarelli, Gianluca, et al. "A microwave resonant sensor for concentration measurements of liquid solutions." *IEEE Sensors Journal* 13.5 (2013): 1857-1864.
- [9] Conway, Bevil R., and Margaret S. Livingstone. "A different point of hue." *Proceedings of the National Academy of Sciences* 102.31 (2005): 10761-10762.
- [10] Alfano, Vincenzo, and Salvatore Ercolano. "Navigating Pandemic Risks." *Journal of Insurance Issues* 47.2 (2024): 158-172.
- [11] Rebber, Matthias, Christoph Willa, and Dorota Koziej. "Organic–inorganic hybrids for CO₂ sensing, separation and conversion." *Nanoscale horizons* 5.3 (2020): 431-453.
- [12] Tashildar, Aakanksha, et al. "Application development using flutter." *International Research Journal of Modernization in Engineering Technology and Science* 2.8 (2020): 1262-1266.
- [13] Geng, Yang, et al. "An intelligent IEQ monitoring and feedback system: development and applications." *Engineering* 18 (2022): 218-231.
- [14] Saini, Jagriti, Maitreyee Dutta, and Gonalo Marques. "A comprehensive review on indoor air quality monitoring systems for enhanced public health." *Sustainable environment research* 30.1 (2020): 6.
- [15] Uppoor, Sandesh, Michail D. Flouris, and Angelos Bilas. "Cloud-based synchronization of distributed file system hierarchies." *2010 IEEE International Conference on Cluster Computing Workshops and Posters (Cluster Workshops)*. IEEE, 2010.