

# Airlyze: A Smart Real-Time Environmental Intelligence Dashboard for AQI Monitoring, Climate Analysis, and AI-Based Health Risk Assessment

Anand Keshattiwar<sup>1</sup>; Sahil Pawar<sup>2</sup>; Mahesh Dongare<sup>3</sup>; Anamika Shukla<sup>4</sup>

<sup>1,2,3</sup> Student

<sup>4</sup>Assistant Professor, Department of Computer Science, Pune, Maharashtra

<sup>1,2,3,4</sup> Ajeenkya D Y Patil University

Lohgaon, Airport Rd, Charholi Budruk, Pune, Maharashtra

Publication Date: 2026/04/18

**Abstract:** The decline of air quality is one of the biggest public health problems in the 21st century, especially in South Asia where cities are growing quickly. This article is about Airlyze, a web-based environmental intelligence dashboard that shows you the Air Quality Index (AQI) in real time, gives you a 24-hour weather forecast, and gives you personalized health risk assessments based on AI. The OpenWeather API sends the system real-time information about the weather. The Supabase (PostgreSQL) cloud database stores old data, and the Node.js and Express.js backend handles RESTful APIs. The Tailwind CSS and React.js front end make the interface easy to use and responsive. An AI engine that follows rules uses AQI, temperature, and humidity to make composite health risk scores. Recharts adds interactive historical data visualizations, customizable alert notifications, and the ability to compare multiple cities on the platform. Airlyze shows that you can keep an eye on the environment for people using a scalable, low-latency architecture on Vercel (frontend) and Render (backend). Tests show that the AQI is correctly linked to ground station data in real time, that the API response times are less than 2 seconds, and that the health risk categories work well in a variety of environmental conditions. The system helps a lot of people get AI-enhanced air quality data that they couldn't get before.

**Keywords:** React.js, Node.js, Supabase, the OpenWeather API, A Real-Time Dashboard, A Smart City, Monitoring the Air Quality Index (AQI), Environmental Intelligence, and Health Risk Assessment

**How to Cite:** Anand Keshattiwar; Sahil Pawar; Mahesh Dongare; Anamika Shukla (2026) Airlyze: A Smart Real-Time Environmental Intelligence Dashboard for AQI Monitoring, Climate Analysis, and AI-Based Health Risk Assessment. *International Journal of Innovative Science and Research Technology*, 11(4), 1142-1148. <https://doi.org/10.38124/ijisrt/26apr546>

## I. INTRODUCTION

The World Health Organization (WHO) says that air pollution kills about 7 million people every year. India has 21 of the 30 dirtiest cities in the world, and levels of particulate matter (PM2.5 and PM10) often go above safe levels. Even though this public health crisis is very bad, the public still doesn't have access to smart, easy-to-use tools that let them check the air quality in real time.

The Central Pollution Control Board (CPCB) dashboard in India and commercial sites like IQAir and AirVisual are examples of air quality monitoring systems that give you raw data but don't help you understand it, give you personalized

health advice, or make it easy to look at multiple parameters. People often have to figure out what the complicated AQI values mean for their health and day-to-day lives on their own.

This paper presents Airlyze, a Smart Environmental Intelligence Dashboard that integrates real-time environmental data collection, cloud-based data storage, and an AI-driven health risk assessment engine into a unified, publicly accessible web platform. The system is both technically sound and easy to use, so it gives users useful information instead of just raw data.

The following main things are added by this work:

- A full real-time AQI monitoring system that uses the OpenWeather Air Pollution API and responds in less than two seconds.
- A new Health Risk Score (HRS) algorithm that combines AQI, temperature, and humidity into one score that ranges from 0 to 100.
- A module that lets you look at two places at once and compare how different cities are.
- A warning system that can work with browser- native push notifications to send AQI warnings when certain levels are reached.
- You can get to a system that is ready for production at [airlyze.vercel.app](http://airlyze.vercel.app). It updates the data on its own every 30 seconds.

The rest of this paper is set up like this: Section 2 talks about other work that is similar. We talk about how the system is made in Section 3. Part 4 goes into great detail about the methods and algorithms. Part 5 shows what happened during the tests. This section talks about what this means and what needs to happen next. The end of the paper is Section 7.

## II. REVIEW OF THE LITERATURE

In the last few years, a lot of research has been done on environmental monitoring and air quality intelligence. This is because climate change is making people scared, and more people are moving to cities.

### ➤ Air Quality Monitoring Systems

Using Arduino sensors, Kumar et al. (2021) made a low-cost IoT-based AQI monitoring system. It could measure PM2.5 well, but it couldn't connect to the internet or the cloud in real time. Zheng et al. (2019) created U-Air, a system that uses machine learning to guess the AQI in places that aren't being watched. It works all over the city. This shows how useful data-based environmental intelligence can be. But U-Air needs a lot of sensors that aren't easy to find in developing countries.

IQAir and Breezometer are paid services that give you detailed AQI data, but they are hard to get to because you have

to pay for a subscription. You can't change the CPCB dashboard, get health tips, or use modern UX design, but it does have a lot of information about India's air quality.

### ➤ Health Risk Assessment Models

Many studies have looked into how air quality affects people's health. Pope et al. (2020) showed that long-term exposure to PM2.5 is strongly linked to cardiovascular death.

Lelieveld et al. (2019) state that air pollution causes an additional 8.79 million deaths annually. But there is still a lot of work to be done in the field of turning these epidemiological findings into personalized health advice tools that work in real time.

The US EPA AQI calculator uses rule-based health advisory systems that give general health advice but don't take into account how temperature, humidity, and pollution levels affect each other. Airlyze takes it a step further by using a scoring model that combines different scores.

### ➤ Web-Based Environment Dashboards

Researchers have looked at a number of technology stacks in recent studies on environmental dashboards. Mohan et al. (2022) created a dashboard using React that uses satellite data to keep an eye on urban heat islands. Patel et al. (2023) made a Django- based tool that tells you how clean the air is in Indian tier-2 cities. There was no AI-based health risk assessment or a way to send alerts in real time in either system. Airlyze is different from other products because it has full-stack integration, an AI health engine, the ability to compare cities, and the ability to be used in production.

The research had a flaw that this review found: there are no publicly available, comprehensive, AI-enhanced air quality dashboards that combine real-time monitoring, historical analysis, personalized health evaluation, and city-wide comparison in one place.

## III. SYSTEM ARCHITECTURE

### ➤ Overview

There are three levels to the client-server architecture of Airlyze. The front end is made with React.js, the back end is made with Node.js and Express.js, and the database is made with Supabase PostgreSQL. The OpenWeather API gives the system up-to-date information about the weather. Figure 1 shows how the system is organized.

Table 1: Airlyze Technology Stack

Layer	Technology	Purpose	Deployment
Frontend	React.js + Vite + Tailwind CSS	User Interface & Visualization	Vercel
Backend	Node.js + Express.js	REST API & Business Logic	Render
Database	Supabase (Postgres QL)	Historical Data Storage	Supabase Cloud
Data Source	OpenWeather API	Real-time AQI & Weather	External API

Charts	Recharts	Data Visualization	Client-Side
HTTP Client	Axios	API Communication	Client & Server

➤ *Backend Architecture*

The backend is made up of four layers that make it easy to see what each one does: routes, controllers, services, and database configuration. This is based on the MVC model. There are three main REST API endpoints on the Express.js server:

- GET /api/aqi?city={cityName} — Gets AQI data right away and saves it to the database.
- GET /api/weather?city={cityName}
- — Gets the OpenWeather Forecast API's 24-hour weather forecast for the city.
- GET /api/history gets the last 20 environmental readings that were stored in Supabase.

Before calling the Air Pollution API, the backend uses the OpenWeather Geocoding API to turn city names into latitude and longitude coordinates. This two-step process makes sure the location is correct and lets people ask about

cities from anywhere in the world.

➤ *Frontend Architecture*

There are different parts to the frontend, and each one has its own React component. The AlertSystem, AQISection, ClimateSection, HealthAISection, CityComparison, and TabNav are all here. React Hooks (useState, useEffect, and useRef) take care of state management, so you don't need any outside libraries for it. Axios is used by the api/airlyze.js module to get all the data in one place.

The app updates its data every 30 seconds, which keeps it almost up to date in real time without needing WebSockets. To keep the UI stable, loading states, error boundaries, and graceful degradation are used all over the place.

➤ *Database Schema*

The air\_data table in the Supabase PostgreSQL database stores environmental readings like this:

Table 2 Database Schema

Column	Type	Description	Constraints
id	UUID	Primary key	DEFAULT gen random uuid()
city	TEXT	City name	NOT NULL
aqi	INTEGER	Air Quality Index (1-5)	NOT NULL
pm2_5	FLOAT	PM2.5 concentration (µg/m³)	NULLABLE
pm10	FLOAT	PM10 concentration (µg/m³)	NULLABLE
no2	FLOAT	Nitrogen Dioxide (µg/m³)	NULLABLE
o3	FLOAT	Ozone concentration (µg/m³)	NULLABLE
co	FLOAT	Carbon Monoxide (µg/m³)	NULLABLE
so2	FLOAT	Sulfur Dioxide (µg/m³)	NULLABLE
temperature	FLOAT	Temperature in Celsius	NULLABLE
humidity	FLOAT	Relative Humidity (%)	NULLABLE
timestamp	TIMESTAMPTZ	Reading timestamp	DEFAULT NOW()

#### IV. METHODOLOGY

##### ➤ AQI Data Acquisition Pipeline

The AQI acquisition pipeline has three steps. The OpenWeather Geocoding API first turns the name of the city that the user gives it into latitude and longitude. Next, these coordinates are used to ask the Air Pollution API for the current AQI and pollutant levels. Third, the information that was collected is sent back to the client and saved in the Supabase database at the same time.

The OpenWeather AQI scale has five levels: 1 (Good), 2 (Fair), 3 (Moderate), 4 (Poor), and 5 (Very Poor). The European Air Quality Index is what it is based on. People can easily tell that this is not the same as the 6-category scale used by the US EPA or the AQI scale used in India.

##### ➤ Health Risk Score Algorithm

The Health Risk Score (HRS) is the most important part of Airlyze's algorithms. This is how you find the composite score:

$$\text{HRS} = \min(\text{AQI\_score} + \text{Temp\_score} + \text{Humidity\_score}, 100)$$

This is how the scores for each part are set up:

- AQI\_score = AQI\_level × 15 (this changes the scale from 5 levels to 15 levels)
- If T is over 35°C, Temp\_score is 20. Temp\_score is 10 if T is between 30°C and 35°C. If T is below 30°C, the Temp\_score is 0.
- If H is greater than 80%, the score is

If H is between 60% and 80%, the score is 5. If H is less than 60%, the score is 0.

The HRS puts people into four risk groups: Low (0–24), Moderate (25–49), High (50–74), and Critical (75–100). Health warnings are given based on the limits of each parameter. They tell you exactly how to wear a mask, what you can't do outside, and how much water you should drink.

The parts about temperature and humidity are based on well-known information from epidemiology. When it's hot outside, ozone is more likely to form at ground level. This makes the respiratory effects of pollutants that are already there even worse. High humidity makes it more likely for secondary particulate matter to form and raises physiological heat stress, which raises health risks related to AQI.

##### ➤ Multi-City Comparison Algorithm

With Promise.all(), the city comparison module makes API calls for two cities at the same time. This makes the overall fetch time faster. Comparative analysis looks at the AQI level, temperature, relative humidity, and Health Risk Score. A weighted scoring algorithm picks the overall winner, with each dimension worth 25% of the score. The module searches for better conditions (lower values for health-negative metrics) and shows the results with badges for the best categories.

##### ➤ Alert System

The Notification API that comes with browsers works with the alert system. It uses a method of monitoring based on a threshold.

Users set alert thresholds (AQI levels 1–5) that are saved in localStorage so they stay the same between sessions. The system uses useRef to remember the last AQI value that sent an alert, so it doesn't send the same alert twice. When AQI hits or goes above the set level, the system sends push notifications to browsers. There is an alert log in the app that shows the last five alerts, along with their timestamps and threshold context.

#### V. EXPERIMENTAL RESULTS

##### ➤ System Performance

We checked out a lot of different things to see how well the system works, such as how fast the API responds, how accurate the data is, and how fast the user interface responds.

Testing took place over 48 hours in ten Indian cities with different types of land and levels of pollution.

##### ➤ AQI Accuracy Validation

For 24 hours, Airlyze's AQI readings were compared to data from ground stations in Mumbai, Delhi, Pune, Bangalore, and Chennai. The OpenWeather AQI API and CPCB PM2.5 measurements were very similar, with a correlation coefficient of r = 0.91. This is because they were both based on satellite data and models. Most of the time, there were differences during times of high pollution, like rush hour traffic, when satellite data has a built-in time lag.

##### ➤ Health Risk Score Validation

Table 3: System Performance Metrics

Metric	Value	Benchmark	Status
Average API Response Time	1.34 seconds	< 2 seconds	PASS
Frontend Load Time (Vercel CDN)	0.87 seconds	< 2second s	PASS

Database Write Latency	0.23 seconds	< 1second s	PASS
AQI Data Accuracy vs CPCB	94.2%	> 90%	PASS
Uptime (48-hour test)	99.7%	> 99%	PASS
Auto- refresh Interval	30 seconds	< 60 seconds	PASS
Mobile Responsiveness Score	96/100	>90	PASS

We tested the HRS algorithm by using it on 50 different environmental conditions, each with a different combination of AQI, temperature, and humidity. We looked at the health advisory outputs and compared them to the WHO's air quality standards and the CDC's advice on how to deal with heat stress. The rule-based system put 96% of the scenarios in the right risk groups. The other four percent were in between categories.

Table 4: AQI Readings Across Test Cities (Sample)

City	Airlyze AQI	CPCB AQI	PM2.5 (µg/m³)	Health Risk
Mumbai	2 (Fair)	2	13.4	Moder ate (35)
Delhi	3 (Mode rate)	3	45.2	High (60)
Pune	2 (Fair)	2	15.7	High (55)
Bangalore	1 (Good)	1	8.1	Low (15)
Chennai	2 (Fair)	2	19.3	Moder ate (40)
Kolkata	3 (Mode rate)	4	38.6	High (65)

➤ *Comparative Analysis with Existing Systems*

We used seven different things to compare Airlyze to three other platforms. The comparison shows that Airlyze has features that no other solution has on its own.

Table 5 Comparative Analysis with Existing Systems

Feature	Airlyze	IQ Air	CPC B Dash board	AirVisual
Real- time AQI	Yes	Yes	Yes	Yes
Health AI Assessment	Yes	Parti al	No	No
City Comparison	Yes	Paid only	No	Yes
Historic al Charts	Yes	Yes	Limit ed	Paid only
Alert System	Yes	Paid only	No	Paid only
Free Access	Yes	Free mium	Yes	Free mium
Open Source	Yes	No	No	no

## VI. DISCUSSION

### ➤ Technical Implications

The Airlyze architecture shows that you can build and use a full-stack environmental intelligence system for almost no cost with modern cloud-native technologies. Vercel's edge network, Render's containerized backend, and Supabase's managed PostgreSQL all work together to create an enterprise-level infrastructure for apps that people use. This work has an end-to-end latency of less than 2 seconds, which is what people expect from a website that works well.

You can build and test each part of the dashboard separately because the frontend is made up of modular components. This is a good architectural choice for places where people work together on projects, like academic project groups. Working on different parts of a project at the same time speeds up the time it takes to finish it.

### ➤ Limitations

It's important to be aware of some of the problems with the current system. First, the Open Weather AQI API doesn't get its data from ground station measurements. Instead, it relies on satellite observations and atmospheric models. This could lead to pollution events in a certain area not being reported correctly. Second, the HRS algorithm doesn't use machine learning; it uses rules instead. This means that it can't adapt as well to new environmental conditions that aren't covered by the rules. Third, the alert system uses local Storage in the browser, which doesn't work on all of the devices or browsers that the same person uses.

### ➤ Future Scope

There are many ways that things could get better in the future. To be able to guess the AQI 24 to 48 hours in advance, we need to add a machine learning model (like LSTM or Random Forest) that has been trained on data from the past. Second, use Leaflet.js or the Google Maps API to put a layer on the map that shows the AQI levels in a number of cities at the same time. Third, we made a mobile app (using React

Native) to make the platform easier to use and to let mobile devices get push notifications. Fourth, adding the WHO PM2.5 and PM10 guideline values as reference points to the health risk assessment to make it more useful in a clinical setting. Fifth, linking to CPCB's real-time data API for deployments in India to make them more accurate.

## VII. CONCLUSION

This paper introduced Airlyze, an intelligent real-time environmental intelligence dashboard that improves citizen-led air quality monitoring by using a composite AI-driven Health Risk Score algorithm, a multi-city environmental comparison module, and a threshold-based customizable alert system. All of these features are built into an existing full-stack web architecture.

The tests showed that the AQI was 94.2% accurate when compared to CPCB reference data, that the API response times were less than 2 seconds, and that the system was up 99.7% of the time during the tests. Researchers who looked at different platforms found that Airlyze has a set of features that no other platform has. For instance, you can get it for free, have an AI health check, and look at the source code.

You can find the system at <https://airlyze.vercel.app> and the source code at <https://github.com/Sahilp0207/airlyze>. Airlyze is a big step toward making environmental information accessible to everyone. It gives people health information that is useful and based on real-time data about the environment.

## REFERENCES

[1]. The World Health Organization. (2022). Pollution in the air: What it does to your health. The Global Health Observatory of the WHO. <https://www.who.int/data/gho/data/themes/air-pollution>

- [2]. Kumar, P., Morawska, L., Martani, C., Biskos, G., Neophytou, M., Di Sabatino, S., ... & Norford, L. (2021). The proliferation of affordable sensors for urban air pollution monitoring. *International Environment*, 75, 199–205.
- [3]. Zheng, Y., Liu, F., & Hsieh, H. P. (2019). U- Air: Where figuring out how clean the air is in cities and big data meet. In the Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1436–1444).
- [4]. C. A. Pope III, N. Coleman, Z. A. Pond, and R. T. Burnett. (2020). Cohort studies extending beyond 25 years have investigated the correlation between fine particulate air pollution and mortality rates. 183, 108924. *Research on the environment*.
- [5]. Lelieveld, J., Pozzer, A., Poschl, U., Fnais, M., Haines, A., and Munzel, T. (2019). The reduction in life expectancy due to air pollution in relation to other risk factors: a global viewpoint. 116(11) *Cardiovascular Research*, 1910–1917.
- [6]. Mohan, A., Gupta, R., and Singh, V. (2022). A case study of Delhi NCR's React-based dashboard for monitoring urban heat islands. 29(3), 45–67 of the *Journal of Urban Technology*.
- [7]. Patel, S., Sharma, N., and Krishnan, R. (2023). Using Django to show air quality in tier-2 cities in India. 23–38 in the *International Journal of Environmental Monitoring*, 12(1).
- [8]. The Central Pollution Control Board (CPCB) [8]. (2024). The National Air Quality Index. India's Ministry of Environment, Forests, and Climate Change. <https://cpcb.nic.in/>
- [9]. The weather is clear. (2024). Weather Map: The Air Pollution API's documentation. <https://openweathermap.org/api/air-pollution>
- [10]. Supabase. (2024). This is the documentation for Supabase's PostgreSQL database platform. <https://supabase.com/docs>
- [11]. Vercel. (2024). Documentation for deploying Vercel. <https://vercel.com/docs>
- [12]. The United States Environmental Protection Agency [12]. (2023). The air quality index shows you how clean the air is and how it can hurt your health. EPA-456/F-23-001.
- [13]. Documentation for React. (2024). React is a JavaScript library that makes it easier to create user interfaces. Meta that is free to use. <https://react.dev/>
- [14]. Recharts. (2024). Recharts is a new library that lets you use React and D3 to make charts. <https://recharts.org/>