BREATHE

COMBATING AIR POLLUTION

For Real-Time Air Quality Monitoring

Introduction

Breathe is a mobile and IoT application developed in MIT App Inventor that enables real time monitoring and prediction of air quality, integrating information from Sentinel-5P TROPOMI satellite mission and low-cost environmental sensors. The system collects, analyzes, and visualizes air pollution data to provide users with personalized health recommendations based on their environment and personal profile.

In addition to its mobile and IoT components, Breathe also includes an AI powered chatbot linked to ChatGPT, which allows users to interact naturally with the system, asking questions about air quality, understanding pollutant levels, and receiving guidance on how to stay safe in varying atmospheric conditions.

Furthermore, the project features a dedicated web platform that shares educational resources, real-time updates, and information about the project's technologies and environmental impact. This website aims to raise public awareness and promote community participation in air-quality monitoring.

The goal of Breathe is to protect public health, raise environmental awareness, and promote citizen participation in air quality measurement—combining satellite science, sensor technology, and artificial intelligence in a collaborative ecosystem.

Context and Motivation

Air pollution represents a global environmental and public health crisis. The World Health Organization (WHO) estimates that more than 99% of the world's population breathes air containing pollutant levels exceeding safe limits, leading to 7 million deaths annually from respiratory, cardiovascular, and neurological diseases.

The most critical pollutants include:

- Nitrogen Dioxide (NO_2)
- Formaldehyde (CH_2O or HCHO)
- Aerosol Index (AI)

- Particulate Matter (PM1.0, PM2.5, PM10)
- Ozone (O_3)

The Sentinel-5 Precursor (TROPOMI) Mission Provides near-real-time global monitoring of key atmospheric pollutants, including NO_2, O_3, SO_2, CO, formaldehyde, and aerosols, with a spatial resolution of up to 7×3.5 km². However, satellite data alone cannot fully represent street-level conditions where people actually breathe.

Our challenge is simple: help as many people as possible make safer, healthier choices every day. This project has two parts that work together: a mobile app that personalizes air-quality guidance, and a companion device (working name) that measures local air when there's no internet or cell signal.

The mobile app turns satellite observations from Sentinel-5P trompi, MERRA-2 atmospheric y OpenWeatherMap into practical, real-time advice. It shows current air quality, short-term trends, and clear recommendations tailored to each user. With your consent, you can add details like age, respiratory or heart conditions, pregnancy, outdoor work hours, or training goals. The app then adjusts alerts to your profile—sending notifications such as "Great air today—perfect for a run," or "Unhealthy for sensitive groups—limit outdoor time and consider a mask." You choose alert times and thresholds; your data is kept private and used only to improve your guidance

For communities without reliable signal or Wi-Fi, we are building an offline air-quality monitor (name TBD). It uses onboard sensors to read key pollutants and displays an easy-to-understand status on the device itself, so anyone nearby can see current conditions. When a phone is in range, it can share data locally (e.g., via Bluetooth) so people can still get personalized tips—no mobile data required.

System Architecture

The Breathe ecosystem is structured into three interconnected layers:

1. Breathe Node (IoT Sensors)
2. Python Server (Data Processing)
3. Mobile Application (MIT App Inventor)

Breathe Node – Personalized Sensing Device

The Breathe Node is a portable IoT device designed to capture localized air quality data in real time. It acts as the ground-level sensing component of the Breathe ecosystem, complementing satellite observations and meteorological datasets with high-resolution, on-site environmental measurements.

Built around an ESP32-WROOM-32UE microcontroller, the Breathe Node integrates three key sensors that detect both gaseous and particulate pollutants affecting air quality and human health.
This combination provides a more comprehensive environmental profile and enables the system to correlate ground-based readings with satellite data for validation and model refinement.

This project includes one production firmware for the real ESP32-WROOM-32UE board and two Python-based simulators that replicate the same functionality when physical hardware is unavailable.

The ESP32 firmware handles the following tasks:

- Connects to Wi-Fi and sends data securely using HTTP(S) POST requests.
- Reads particulate matter values (PM1.0, PM2.5, PM10) from the PMS5003 sensor via UART.
- Reads ozone (O_3) and methane (CH_4) levels from the MQ-131 and MQ-4 gas sensors through ADC1.
- Packages all readings into a JSON payload and transmits it to the configured Python Flask server endpoint.

Network credentials, server URL, and transmission intervals are fully configurable, making this firmware suitable for field deployment on real hardware.

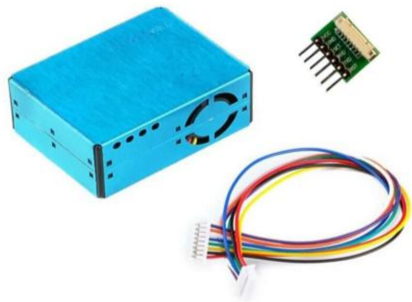
Because the actual ESP32 hardware is not always available during development, two Python simulators were created to mimic the device behavior:

1. Single-Device Simulator: Generates realistic data for PMS5003, MQ-131, and MQ-4 sensors, posting them at defined intervals to the server as if they came from a single ESP32. Users can adjust parameters such as URL, device ID, data count, and timing.
2. Multi-Device Simulator: Extends the same logic to multiple virtual nodes operating simultaneously—either concurrently or in a round-robin pattern—to stress-test the API, database, and dashboards with several simulated devices (e.g., IDs 001…015).

Both simulators strictly follow the same API contract as the ESP32 firmware, sending JSON objects with top-level keys (PMS5003, MQ131, MQ4) and a Content-Type: application/json header.
This unified design allows seamless switching between simulated and real clients without modifying the server, enabling developers to test, debug, and validate the entire data pipeline before deploying the physical hardware.

It includes the following components:

Component	Description	Measured Variable
ESP32-WROOM-32U	Microcontroller that manages sensor data, performs local processing, and transmits information via Wi-Fi.	–
MQ-131	Semiconductor sensor for detecting ozone (O_3) concentrations in the atmosphere.	O_3 (Ozone)
MQ-4	Gas sensor for detecting methane (CH_4) and other light hydrocarbons.	CH_4 (Methane)
PMS5003	Laser-based particulate sensor that measures PM1.0, PM2.5, and PM10 with high precision.	Particulate Matter



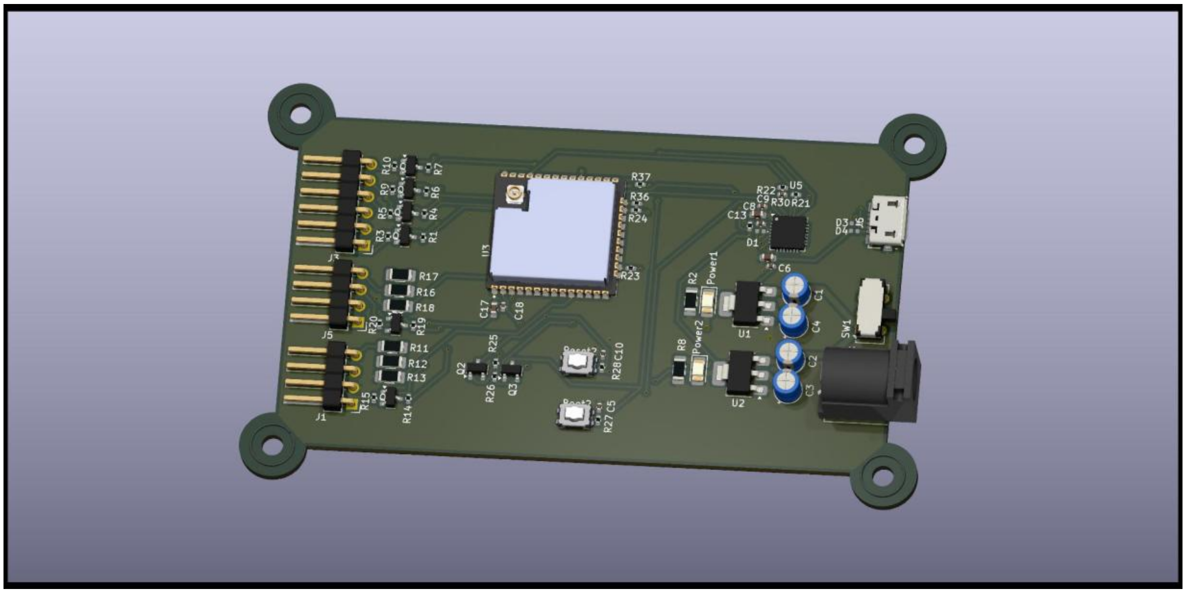
PMS5003

MQ-131

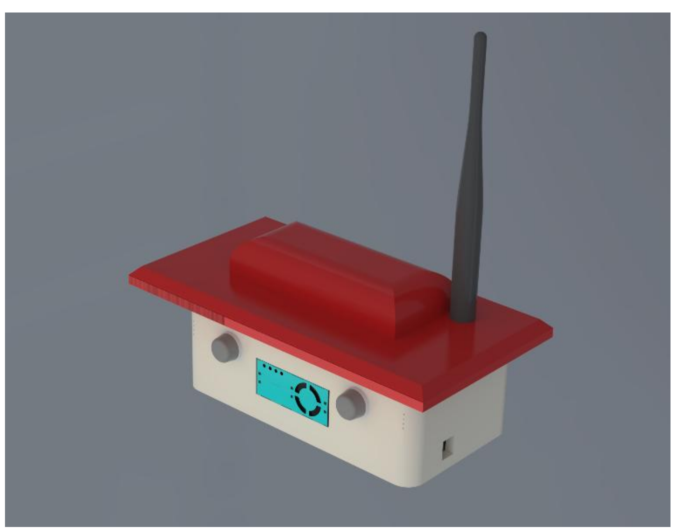
MQ-4

The Air Quality Monitoring PCB is a custom-designed electronic board built around the ESP32-WROOM-32U microcontroller. It serves as a compact and efficient hardware platform for collecting and transmitting environmental data over Wi-Fi. The PCB integrates multiple sensor interfaces and power management components to ensure stable operation and accurate measurements. The design prioritizes:

Ease of assembly: plug-and-play sensor sockets. Signal integrity: proper routing and isolation for analog readings. Wireless communication: through the integrated Wi-Fi module of the ESP32. Compact form factor: suitable for enclosure integration and prototyping.



In parallel with the functional development of the system, a process of prototyping and industrial design was executed to define the final look of the Breathe Node. The three-dimensional rendering was completed, which allows for a precise visualization of the device's ergonomics and aesthetics. The image showcasing the final prototype design is presented below.



Server Layer

The Breathe server, developed in Python, serves as the core processing unit of the system, responsible for integrating, analyzing, and managing all environmental data that feeds the mobile application.

Its main role is to receive real-time information from multiple sources such as IoT sensors (Breathe Nodes) and open environmental databases, and OpenWeatherMap to generate accurate and personalized air quality assessments.

The server operates using a Flask rest api architecture, enabling efficient and structured communication between the backend and the mobile application.

Data exchange between the devices and the server is performed through HTTP requests.

Server Workflow

1. Data Reception: The server receives environmental measurements in JSON format from the Breathe Nodes, including variables such as gas concentrations (CO_2, NH_3, CO), particulate matter (PM2.5), temperature, and humidity.

2. Integration with External Sources: Using the earthaccess library, the server connects to the NASA TEMPO mission and downloads satellite products containing atmospheric concentrations of NO_2, SO_2, O_3, and formaldehyde (HCHO). At the same time, it retrieves meteorological data (temperature, humidity, pressure, wind speed) from OpenWeatherMap, and urban air quality.

3. Data Processing and Analysis: With the help of xarray, numpy, and pandas, the server processes satellite files, merges them with local sensor readings to generate a comprehensive environmental profile.
From these combined datasets, the server computes the air quality (AQ) and determines the dominant pollutants in the user's vicinity.

4. Personalized Output Generation: Based on the user's profile and GPS coordinates, the server filters data within an approximate 50 km radius of the user's location. It then returns a json response to the mobile application containing:

    - The current AQ value

    - The dominant pollutants detected in the air
    - A personalized health recommendation (e.g., “Wear a mask” or “Avoid outdoor activities”).

5. Continuous Feedback and Learning: The information collected by users and their Breathe Nodes is stored in the server’s database, where it contributes to refining predictive models. Over time, this allows the system to self-improve, increasing accuracy as more users participate in data collection.

Mobile Application Layer

The Breathe mobile application serves as an intelligent interface for the Python-based server, enabling users to receive filtered and personalized air quality information according to their location and individual profile.

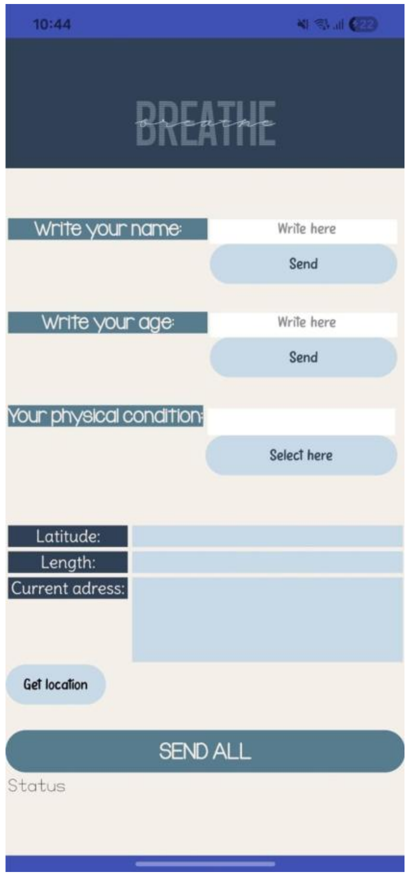
When the user launches the app, they provide basic personal details and GPS access. The server then aggregates information from satellite, Breathe Node sensors, and meteorological APIs to generate localized air quality (AQ) data and personalized health recommendations.

In addition, the mobile app integrates an AI-powered chatbot connected to ChatGPT, designed to provide users with a friendly and educational experience. This virtual assistant can:

- Explain air quality indexes and pollutant data in simple language.
- Recommend protective measures based on the user’s health profile.
- Answer questions about environmental conditions and the effects of pollution.
- Offer environmental education and daily tips to reduce exposure to contaminants.

Through this conversational assistant, users can interact with the system in real time, making the information more accessible, engaging, and easy to understand.

.



Prototyping the application

General Data Flow

1. The Breathe Node captures environmental data and sends it to the Python server.
2. The Python server processes the data, retrieves complementary information from external APIs, and computes the Air Quality.
3. The mobile app (MIT App Inventor) receives the server's response and updates its interface with the latest values, status, and recommendations.

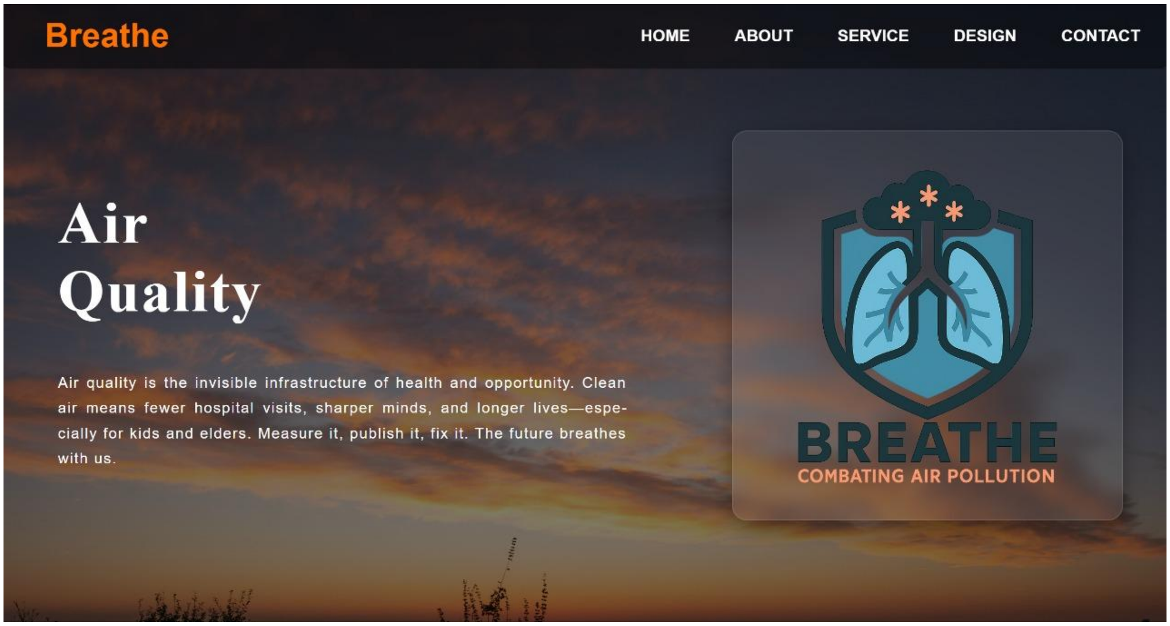
Breathe Web Platform

In addition to the mobile and IoT components, the Breathe project also includes a web platform designed to present, explain, and promote the initiative to a broader audience. This website serves as an informational and educational space, offering insight into the project's objectives, technologies, and environmental impact.

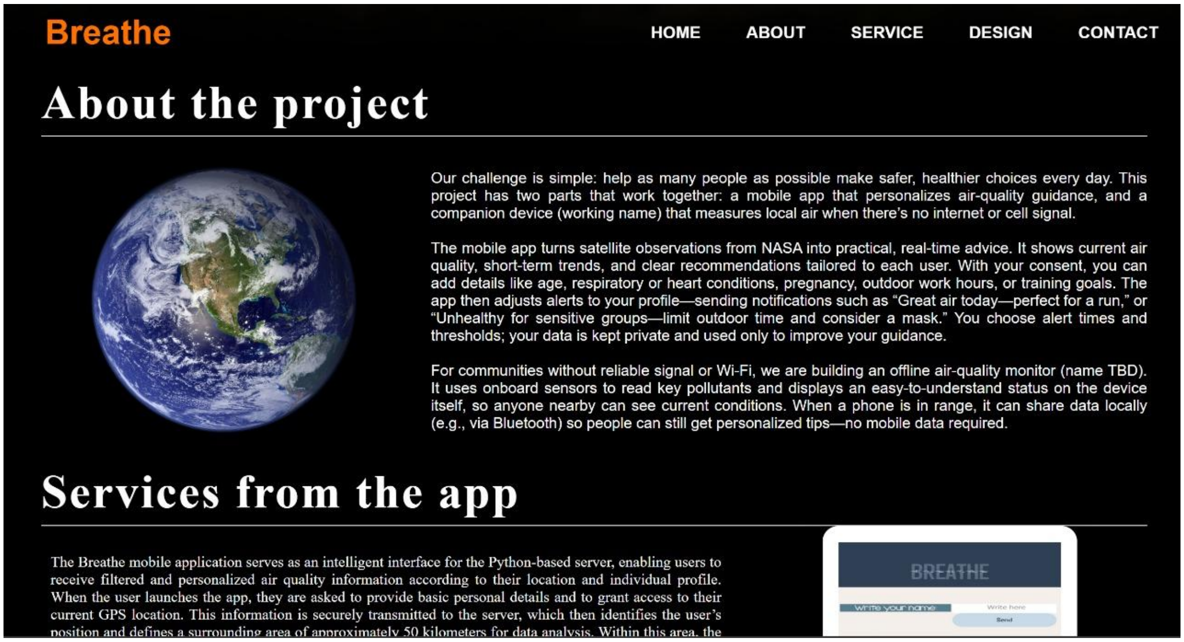
Developed using HTML y CSS the webpage summarizes how Breathe integrates data from satellite, IoT ground sensors, and meteorological APIs to provide real-time air quality insights.

The site not only documents the project but also serves as a public outreach tool, helping raise awareness about air pollution and how citizen-driven technology can contribute to environmental monitoring.

Through this platform, visitors can learn about the project's progress, view live updates, and understand how Breathe bridges the gap between scientific research and community action.



Main face



Secondary face

Future Vision and Potential Impact

The project's next steps are focused on the following key areas:

- Enhancing Predictive Models with Artificial Intelligence: As the Breathe Node network expands, the database will be continuously enriched. We plan to implement machine learning algorithms to analyze this historical data and cross-reference it with satellite and meteorological data. The objective is to develop models capable of predicting high-pollution events hours in advance, allowing users and authorities to take preventive action.

- Expanding the Citizen Network and Geographic Coverage: We will promote the adoption of Breathe Node devices to create an increasingly dense and extensive hyperlocal monitoring network. This will not only improve the accuracy of the global model but also enable the identification of street-level pollution hotspots that satellite data alone cannot capture.

- Integration with Public Policy and Open Science: We aim for the data collected by the Breathe community to become a valuable resource for academic institutions, environmental organizations, and local governments. An open-data portal is planned to allow researchers and urban planners to use the information for scientific studies and for designing public policies that improve air quality in cities.

Breathe is not just a monitoring tool but a dynamic ecosystem with significant growth potential. The long-term goal is to transform how citizens interact with their environment and empower them to become agents of change.