Department of Computer Engineering

“KNOW THE AIR YOU BREEZE IN”

COMP413- Internet of Things

Instructor: Abdülkadir Köse

Team 1

Beyza Erdem- 2211051049

Dilara Cantanı - 2211011045

Elif Sinem Genç- 2211051007

11.01.2026

Table of Contents

1. Introduction

1.1 The Global Imperative for Air Quality Monitoring

There are an array of environmental issues that are brought about by urbanisation accelerating. Poor quality of air is one of the major challenges facing the well being, survival and economic competence of people. By 2050, the United Nations estimates that 68 percent of the world population will be living in cities. This population change creates an extra burden on the cities leading to high resource demand, increased pollution rates, and increased emissions. This will lead to an increased level of hazardous airborne contaminants, namely, particulate matter (PM), volatile organic compounds (VOCs), and nitrogen oxides (NOx), which will threaten human health.

Air pollution is a significant health hazard, which has a significant yearly cost of death. The WHO states that it is associated with lung cancer, cardiovascular disease, and infectious processes.[1] The residents of cities who are vulnerable to high levels of pollutants suffer lower comfort and low productivity. These negative influences are skewed towards people who have little financial capability in accessing healthcare services.

One way of alleviating these problems includes creation of smart, sustainable cities. This approach involves the use of modern technologies in order to improve the urban systems without compromising the environment. As an example, real-time monitoring of air quality parameters can be carried out using the Internet of Things (IoT).

1.2 The Problem Context: Kayseri, Turkey.

The chosen location of research is the city of Kayseri, which is a large industrial and business city in Central Anatolia, Turkey and where the air quality management presents a severe problem because of the unusual topography and meteorological peculiarities. The terrain is basin like with mountains to the east, southeast, and south which inhibit natural ventilation and air circulation. Consequently, the city has high chances of experiencing temperature inversions, especially during winter where warm air film ventures on the cold air that is closer to the surface resulting in residential, traffic, and industrial pollutants trapped in the urban canopy. Kayseri is characterized by air pollution that is mainly generated by three sources, namely residential heating, industrial activity, and vehicle emissions. Although there has been a gradual shift in coal to natural gas, residential heating is still a significant source of pollution, particularly in older residential neighbourhoods who are not well equipped in infrastructural support or have a low economy. The main cause of industrial emissions is three Organized Industrial Zones and a Free Trade Zone that emit such pollutants like VOCs and heavy metals. Also, a fast population increase and more vehicles on the road increases traffic emissions, specifically NOx and CO at peak times. The current air quality surveillance is based on highly accurate reference points which are run by authorities but due to the high cost, the system has a low spatial density which covers street-level and hot trends of pollution. Hence, an additional layer of monitoring, which remains inexpensive and is composed of a network of sensors, is necessary to obtain the spatiotemporal air quality data of high quality in real-time.

1.3 Project Objectives

The main purpose of the proposed study is to design, develop and test a full-scale internet of things air quality monitoring system that fits the special requirements of Kayseri. This system also tries to narrow the gap between the sparse reference stations and the requirement of the hyper-local environmental data.

The specific objectives are:

- **Prototype Development**: To design a strong sensor node based on the SGP30 (TVOC/eCO2) and CJMCU-680/BME680 (Temperature, Humidity, Pressure, Gas) sensors in combination with ESP32 microcontroller system.
- **Long-Distance Communication:** To establish a private LoRa (Long Range) Point-to-Point (P2P) network with 433MHz frequency, with Ra-02 modules being used to guarantee the full transmission of data over the urban landscape of Kayseri and not depending on the cellular network or any third-party WAN infrastructure.
- **Smart Anomaly Detection**: To go beyond mere data logging by implementing TinyML (Tiny Machine Learning) code on the edge device. This will allow the system to detect dangerous air quality incidents independently and send local alarms (Visual/Auditory).
- **Full-Stack Visualization:** To create a complete software stack, i.e. a PlatformIO-based firmware, a Python FastAPI backend with SQL database, and an interactive web dashboard using Leaflet.js to display pollution gradients throughout the city.
- **Sustainability Alignment:** In order to specifically align the technological solution with the United Nations Sustainable Development Goals (SDGs) 3, 11, and 13, it is important to show how engineering interventions can be used to achieve the global sustainability goals.

1.4 Scope of the Project

The breadth of this project will include the design of the entire system of monitoring, including the design of physical circuits and the deployment of cloud applications. Geographically, pilot deployment and web interface design will be concentrated on the province of Kayseri where the dashboard is directly mapped to it. The system is able to trace five important environmental parameters, such as, Atmospheric Pressure, Temperature, Relative Humidity, Volatile Organic Compounds (VOC) and Carbon Dioxide equivalent (eCO2). The project is designed against the background of an educational course in the field of IoT, with a focus on educational aspects of hardware interfacing, network protocols, and software engineering.

2. System Model

2.1 Theoretical Framework: The IoT Layered Architecture

The proposed system is designed based on the traditional three-layer IoT design: the Perception Layer, the Network Layer, and the Application Layer. Such a structured approach guarantees the modularity, scalability and maintenance ease.

2.1.1 The Perception Layer (Edge Nodes)

The bottommost layer is the Perception Layer which includes the physical hardware installed in the environment. This contains sensors (SGP30, BME680) that can translate the physical environmental phenomena into digital or analog signals. This layer has the ESP32 microcontroller as its brain. These nodes are not passive data loggers, but are instead, so-called, smart edges; they both acquire and pre-process data, but more importantly, infer using TinyML models. This layer also takes care of local instant feedback through the alarm system (Red/Green LEDs and Buzzer) which gives instant status messages to the citizens nearby regardless of network connectivity.

2.1.2 The Network Layer (Transmission)

NG Network Layer deals with reliable conveyance of data between the edge and processing center. Taking into consideration the size of Kayseri city and the need to use a low power consumption, the standard Wi-Fi is not applicable to the distributed sensor nodes since of restricted range and high energy cost. Cellular (NB-IoT/LTE-M) is covered, but it also brings about recurring subscription fees. Thus, the given project is based on LoRa (Long Range) technology.

- **Topology**: The system uses a Star topology which is put in practice by a custom-made Point-to-Point (P2P) protocol. The sensor nodes (Senders) are sending data packets to a central Gateway (Receivers).
- **Frequency**: The system has a frequency of 433 MHz which is also license free in Turkey (ISM). This frequency is selected strategically at 868 MHz or 2.4 GHz, where the frequency possesses better characteristics of propagation in urban settings, so that it can penetrate concrete structures and diffract around obstacles better.

2.1.3 The Application Layer (Processing and Visualization)

Application Layer converts raw data to actionable intelligence. It consists of: Gateway Bridge: ESP-WROOM-32 device that takes in LoRa packets and forwards them to the backend server through Serial/UART.

- **Backend Server**: Python-based application based on FastApi. It performs data parsing, storage in a SQL database and publication to an MQTT broker.
- **Frontend Interface**: A web-based dashboard that is developed using HTML, CSS, and JavaScript. It subscribes to MQTT stream to update in real-time and applies Leaflet.js to create geospatial heatmaps of the distribution of pollution.

2.2 System Architecture Diagram Description

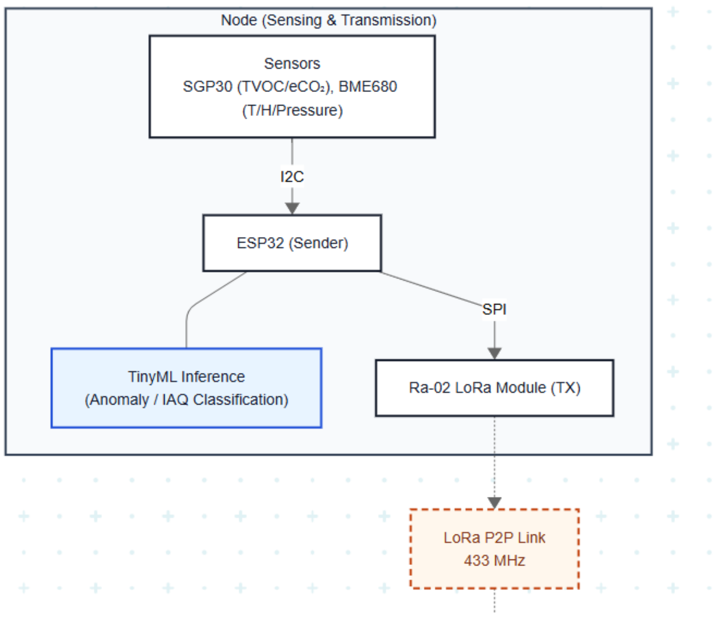


Figure 1-System Architecture Diagram Description-1

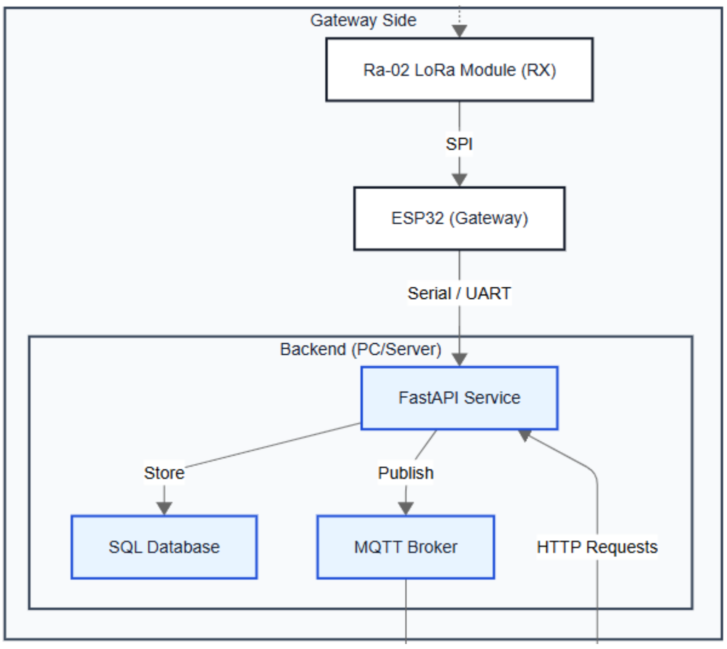


Figure 2-System Architecture Diagram Description-2

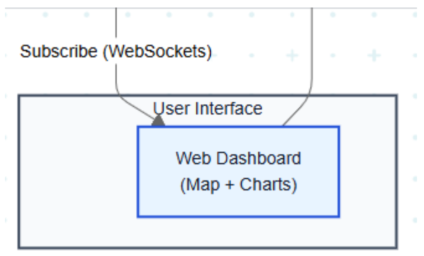


Figure 3-System Architecture Diagram Description-3

2.3 Operational Workflow

- **Sensing**: At each sampling time (i.e. 10-sec), ESP32 spins up and reads the SGP30 and BME680 sensors using I2C bus.
- **Processing**: The raw values are made normalized. The model is a TinyML which examines sensor data as a vector to identify anomalies. At the same time, the firmware determines whether the values of VOC or eCO2 have surpassed hard-coded safety conditions.
- **Actuation**: Safe State: Green LED on. Red LED on and Buzzer.
- **Transmission**: The payload of the data (Sensor IDs, Values, Anomaly Status) is packetized and sent over the Ra-02 module.
- **Ingestion**: The packet is sent to the Gateway that verifies the checksum and forwarded using Serial to the backend.
- **Storage & Broadcasting**: The backend saves the record in the SQL database to provide historical analysis and publishes the live data to a particular MQTT topic (e.g., kayseri/air/node1).
- **Visualization**: The MQTT message is transferred to the web frontend and shown on the map markers and the Dom elements (gauges, charts) instantly.

3. Hardware Design

3.1 Microcontroller Units (MCU): ESP32 Series

The project utilizes the ESP32 platform, with the sensor nodes using the generic ESP32 development board and the ESP-WROOM-32 gateway module.

- **Processing Capabilities**: ESP32 has a processor core of a dual-core Xtensa 32-bit LX6 capable of a clock frequency up to 240 MHz.6 This is highly important because it has a high computational performance. As one core effectively can handle the LoRa protocol requiring timing, and the sensor essentially, the other can be directed at the computationally intensive TinyML inferences, avoiding blocking of the system.
- **Memory**: The ESP32 has 520 KB of internal SRAM, meaning it has sufficient memory to store the TensorFlow Lite for Microcontrollers interpreter and the associated model weights in RAM without needing more memory.
- **Power Management**: The ESP32 has powerful power management capabilities such as the Deep Sleep mode with only a few microamps of power consumption that is likely to be important in the next version of the project that may require the introduction of solar energy.

3.2 Environmental Sensors

3.2.1 SGP30 Multi-Pixel Gas Sensor

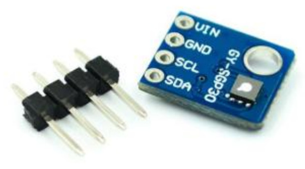


Figure 4-SGP30 Multi-Pixel Gas Sensor

The SGP30 is a digital multi-pixel metal-oxide (MOX) gas sensor that is used in indoor air quality monitoring. This is because it is selected due to its capability to identify a large variety of Volatile Organic Compounds (VOCs) and give a matching CO2 (eCO2) value.

- **Principles of Working**: This sensor has a micro-hotplate which heats a metal oxide sensing element. In the reduction of gases (such as VOCs) the oxygen present on the surface reacts with that gas thereby releasing electrons and lowering the electrical resistance. The SGP30 is a measure of this resistance change.
- **Multi-Pixel Technology**: In contrast to the conventional single-pixel sensor (e.g., CCS811), the SGP30 has a combination of sensing elements in one chip. This enables it to distinguish among the various types of gases and give a more stable TVOC signal, calibrated against Ethanol, and eCO2 signal calibrated against Hydrogen.[10]

Key Specifications:

- **TVOC Range:** 0 ppb to 60,000 ppb.
- **eCO2 Range:** 400 ppm to 60,000 ppm.

Interface: I2C (Address 0x58).[11]

- **Baseline Management**: MOX sensors have a serious drawback in drift. The SGP30 has an internal baseline which is that of clean air. This calibration should be saved in the EEPROM of the ESP32 and reloaded each time it is booted down otherwise the sensor will recalibrate each time it is booted up, resulting in a false first reading.[12]

3.2.2 CJMCU-680 (BME680)

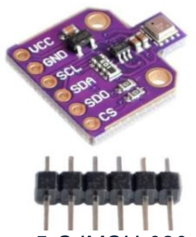


Figure 5-CJMCU-680 (BME680)

The BME680 is a 4-in-1 integrated environmental sensor measuring Temperature, Humidity, Barometric Pressure, and Gas Resistance.

- **Importance of Sensor Fusion**: Although the SGP30 is of primary use as a gas sensor, the BME680 is a two-fold use device. First, it gives meteorological information (Pressure, Temp, Humidity) and itself such information constitutes valuable environmental parameters. Second, and more crucially, humidity and temperature are great effects on MOX gas sensors. The high humidity may be confused with the high concentration of gasses. The system uses the fine-tuning of humidity measurements on the BME680 by feeding the compensation algorithms with the data feeds of the SGP30 to obtain accurate values of the SGP30 in the changing weather conditions of Kayseri.

Key Specifications:

- **Pressure Accuracy:** RMS Noise 0.12 Pa (approx. 1.7 cm altitude resolution).
- **Humidity Response Time:** 8 seconds.
- **Interface:** I2C (Address 0x77 or 0x76).
- **Power Consumption:** Extremely low (3.7 µA at 1Hz), supporting energy-efficient operation.

3.3 Wireless Communication: Ra-02 LoRa Module

Figure 6-Ra-02 LoRa Module

Ra-02 is an example of a wireless transmission module that is created on the basis of Semtech SX1278 wireless transceiver.

- **Modulation Technology**: It uses the Long Range (LoRa) modulation of spread spectrum. It is a proprietary form of chirp Spread Spectrum, known as Chirp modulation technique. It represents data in chirps a signal of different frequencies. This enables a receiver to demodulate signals that are much lower than the noise floor, and it has outstanding sensitivity (-148 dBm).
- **Frequency Band**: The module will be used at 433 MHz. The 433 MHz propagation character is superior to that of the higher frequencies in the environment of the urban development of Kayseri. Longer wavelengths have a higher ability to bend around buildings and go through walls, which is essential to a P2P network that does not rely on a dense gateway infrastructure.
- **Antenna**: The module has an IPX connector(U.FL) connector. The project makes use of external 433MHz antenna. The gain and quality of this antenna is the most important factor which determines the link budget and the effective range.

3.4 Circuit Configuration and Interfacing

The hardware integration involves interfacing the sensors and the LoRa module with the ESP32. Both sensors share the I2C bus, while the LoRa module uses the SPI bus. The node is physically designed to have small wiring by ensuring that there is minimal interference.

Component	Pin Name	ESP32 GPIO	Description
SGP30	VCC	3.3V	Power Supply
SGP30	GND	GND	Ground
SGP30	SDA	GPIO 21	I2C Data Line
SGP30	SCL	GPIO 22	I2C Clock Line

BME680	VCC	3.3V	Power Supply
BME680	GND	GND	Ground
BME680	SDA	GPIO 21	Shared I2C Bus
BME680	SCL	GPIO 22	Shared I2C Bus
Ra-02	VCC	3.3V	Power Supply
Ra-02	GND	GND	Ground
Ra-02	MISO	GPIO 19	SPI Master In Slave Out
Ra-02	MOSI	GPIO 23	SPI Master Out Slave In
Ra-02	SCK	GPIO 18	SPI Clock
Ra-02	NSS (CS)	GPIO 27	Chip Select
Ra-02	RST	GPIO 14	Reset
Ra-02	DIO0	GPIO 26	Interrupt Request (Done)
Visual Alarm	Anode	GPIO 33	Red LED (via Resistor)
Status	Anode	GPIO 32	Green LED (via Resistor)
Audio Alarm	Pos	GPIO 25	Buzzer (PWM driven)

Table 1 - Pin Configuration

Note: Logic level shifting is not required as both the ESP32 and the sensors/modules operate at 3.3V logic.

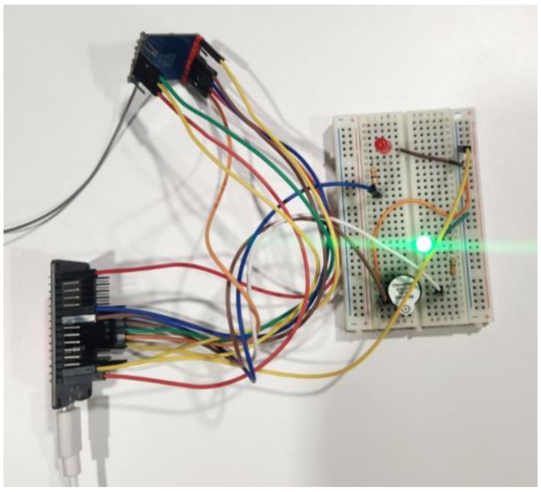


Figure 7-Gateway Circuit-1

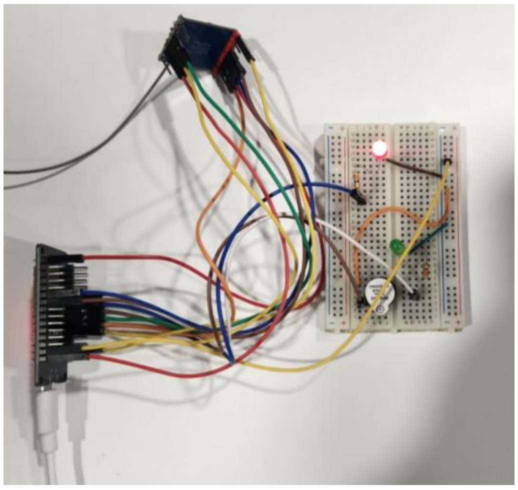


Figure 8- Gateway Circuit-2

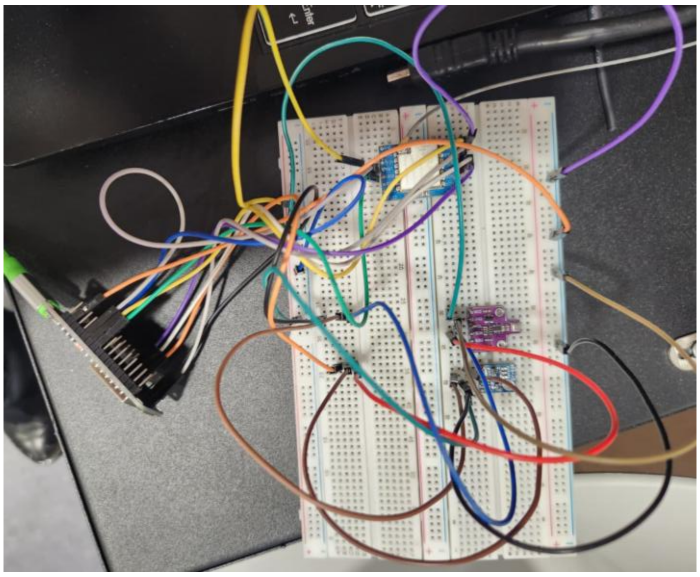


Figure 9- Sensor Node Circuit

4. Software Design

The software architecture follows a modern three-tier design pattern consisting of a database layer, backend services layer, and frontend presentation layer. This separation of concerns enables scalability, maintainability, and independent development of each component.

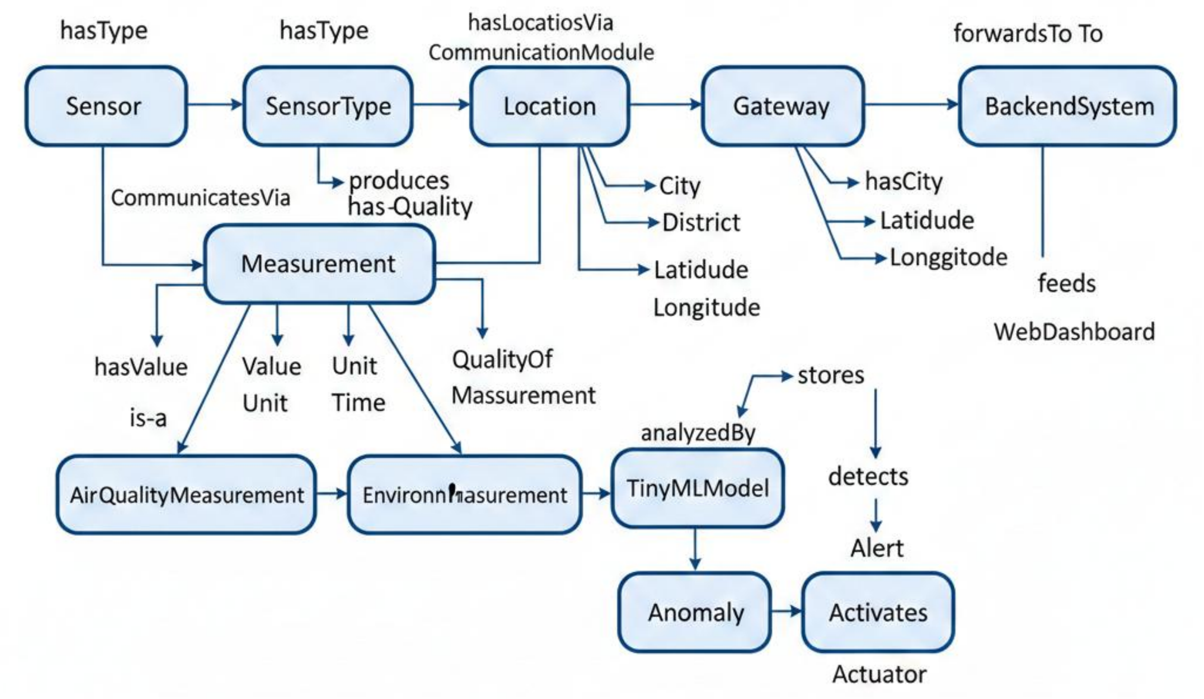


Figure 10-Project Data Flow Scheme

4.1 Database Technologies

The system utilizes SQLite as the primary database management system, chosen for its lightweight nature, zero-configuration deployment, and excellent performance for read-heavy workloads typical of IoT applications. The database schema is designed using SQLAlchemy ORM (Object-Relational Mapping), providing a Pythonic interface for database operations and ensuring type safety.

4.1.1 Database Schema

The database consists of two primary tables:

- **Devices Table:** Stores metadata for each sensor node including device ID, name, geographic coordinates (latitude/longitude), city, and district. This enables geographic visualization and filtering on the web interface.
- **Measurements Table:** Stores time-series sensor data with indexed device_id and timestamp columns for efficient queries. Each record contains environmental measurements (temperature, humidity, pressure), air quality metrics (TVOC, eCO_2), LoRa communication metrics (RSSI, SNR), TinyML predictions, anomaly flags, and computed air quality status.

Field Name	Data Type	Description
device_id	String(64)	Unique sensor node identifier
ts	DateTime	Measurement timestamp (UTC)
temp_c, hum_rh, pressure_hpa	Float	Environmental parameters from BME680
tvoc_ppb, eco2_ppm	Integer	Air quality metrics from SGP30
pred_eco2_60m, pred_tvoc_60m	Integer	TinyML 60-minute predictions
anom_eco2, anom_tvoc	Boolean	Anomaly detection flags
status	String(16)	Air quality status: NORMAL/WARN/HIGH
alert	Boolean	Delta-based sudden change detection

Table 2-Measurements Table Schema

Composite indexes are created on (device_id, ts) tuples to optimize time-range queries, which are frequent in the dashboard's historical data visualization. The database automatically manages record insertion timestamps and handles timezone-aware datetime objects using UTC.

4.2 Frontend Web Application

The web interface provides an intuitive, real-time dashboard for monitoring air quality across multiple sensor locations. Built with vanilla JavaScript, HTML5, and CSS3, the frontend emphasizes performance and browser compatibility without requiring heavy frameworks.

4.2.1 Key Features

- **Interactive Map Visualization:** Implemented using Leaflet.js library, the map displays sensor locations with color-coded markers indicating air quality status. Users can zoom, pan, and click markers to view detailed sensor information.
- **Real-Time Updates:** The dashboard polls the backend API every 5 seconds to fetch latest measurements, ensuring users see current conditions without manual refresh.
- **Heat Map Layer:** Leaflet.heat plugin creates a gradient overlay showing air quality intensity across the geographic area, useful for identifying pollution hotspots.
- **Historical Charts:** Chart.js renders time-series line graphs for the past 2 hours of data, allowing users to observe trends and patterns.
- **Alert History:** A dedicated panel displays recent air quality alerts with timestamps, affected parameters, and severity levels.
- **Geographic Filtering:** Dropdown menus enable filtering by city and district, dynamically updating the map to show only relevant sensors.

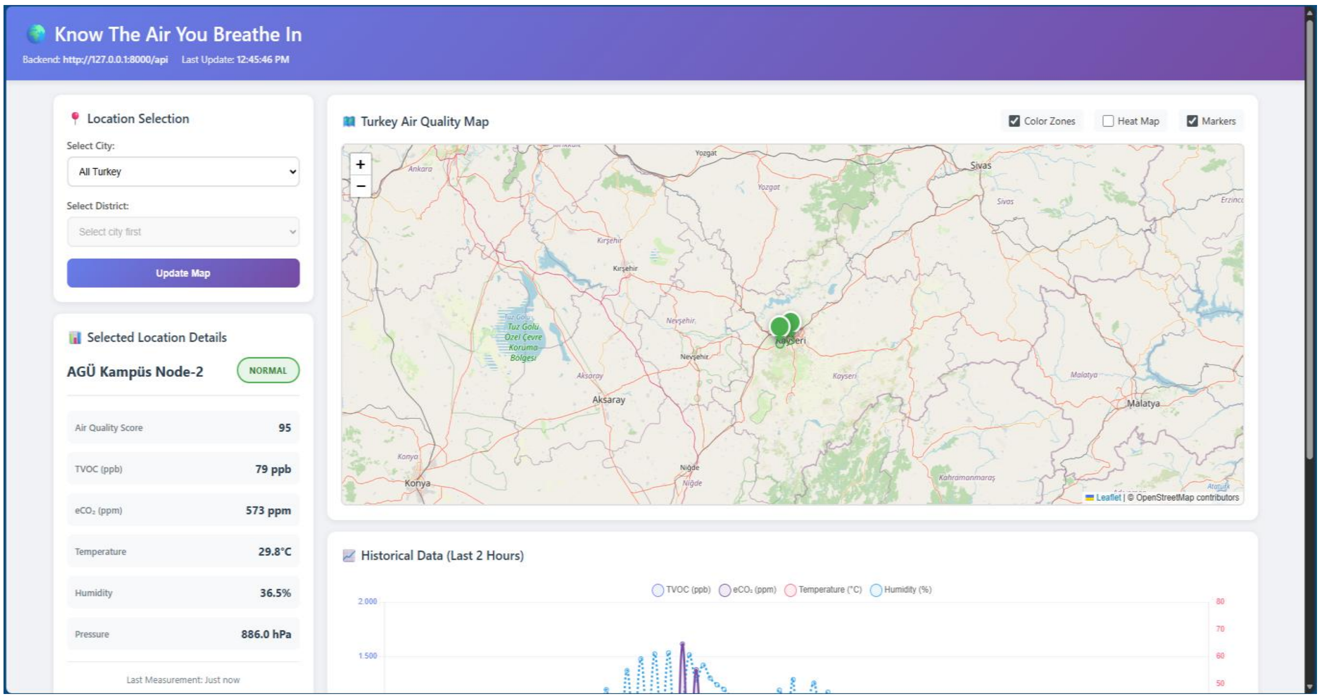
4.2.2 Technical Architecture

The frontend follows the Model-View-Controller (MVC) pattern, with app.js serving as the controller that coordinates between API calls (model) and DOM manipulation (view). Key JavaScript modules include:

- **API Communication:** Fetch API handles all HTTP requests to the backend with error handling and retry logic.
- **Map Management:** Encapsulated functions manage Leaflet map initialization, marker creation/updates, and layer toggling.
- **Chart Rendering:** Chart.js configuration dynamically updates datasets based on API responses, handling missing data gracefully.
- **Event Handlers:** User interactions (clicks, filters, refreshes) trigger appropriate API calls and UI updates.

The CSS stylesheet implements a responsive design with flexbox and grid layouts, ensuring usability across desktop, tablet, and mobile devices. Custom animations provide visual feedback for data updates and loading states.

Web Page Pictures:


Know The Air You Breathe In

Backend: http://127.0.0.1:8000/api Last Update: 12:45:46 PM

Location Selection

Select City: All Turkey

Select District: Select city first

Update Map

Selected Location Details

AGÜ Kampüs Node-2 NORMAL

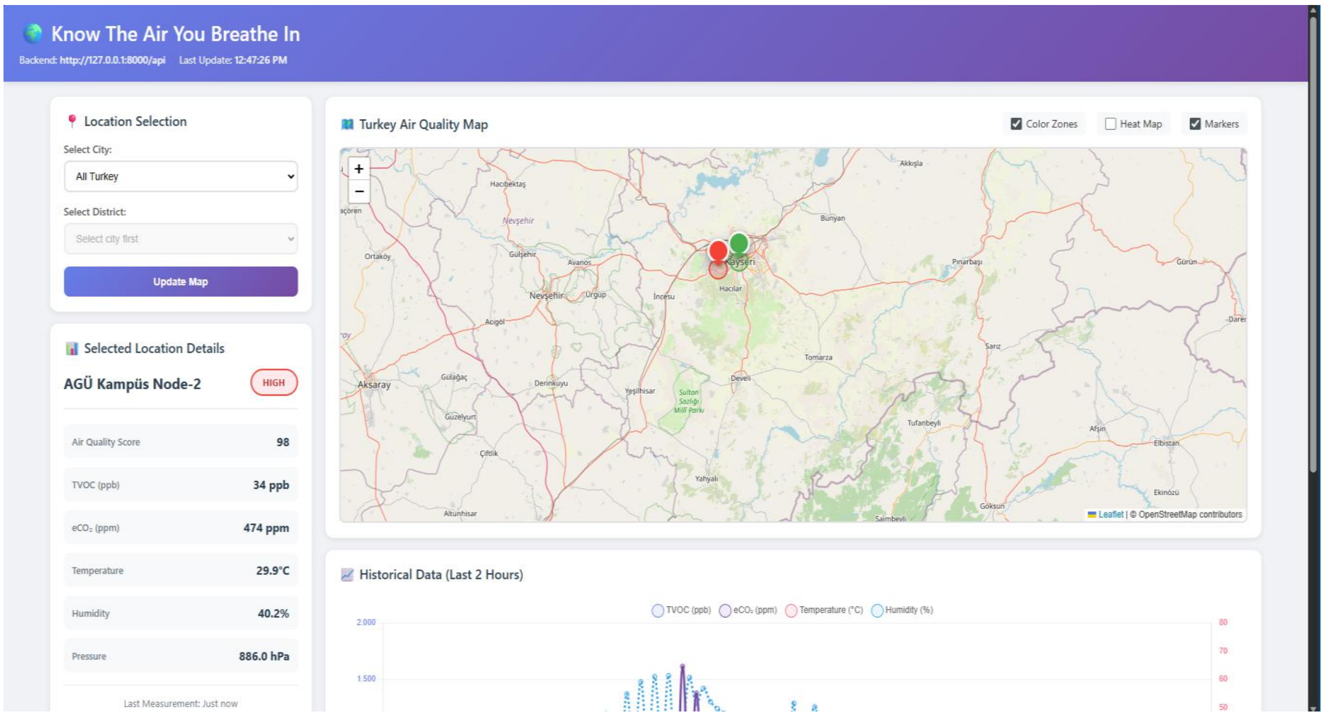
Air Quality Score	95
TVOC (ppb)	79 ppb
eCO₂ (ppm)	573 ppm
Temperature	29.8°C
Humidity	36.5%
Pressure	886.0 hPa

Last Measurement: Just now

Turkey Air Quality Map ✓ Color Zones ☐ Heat Map ✓ Markers

Historical Data (Last 2 Hours)

Figure 11-Web Page-1


Know The Air You Breathe In

Backend: http://127.0.0.1:8000/api Last Update: 12:47:26 PM

Location Selection

Select City: All Turkey

Select District: Select city first

Update Map

Selected Location Details

AGÜ Kampüs Node-2 HIGH

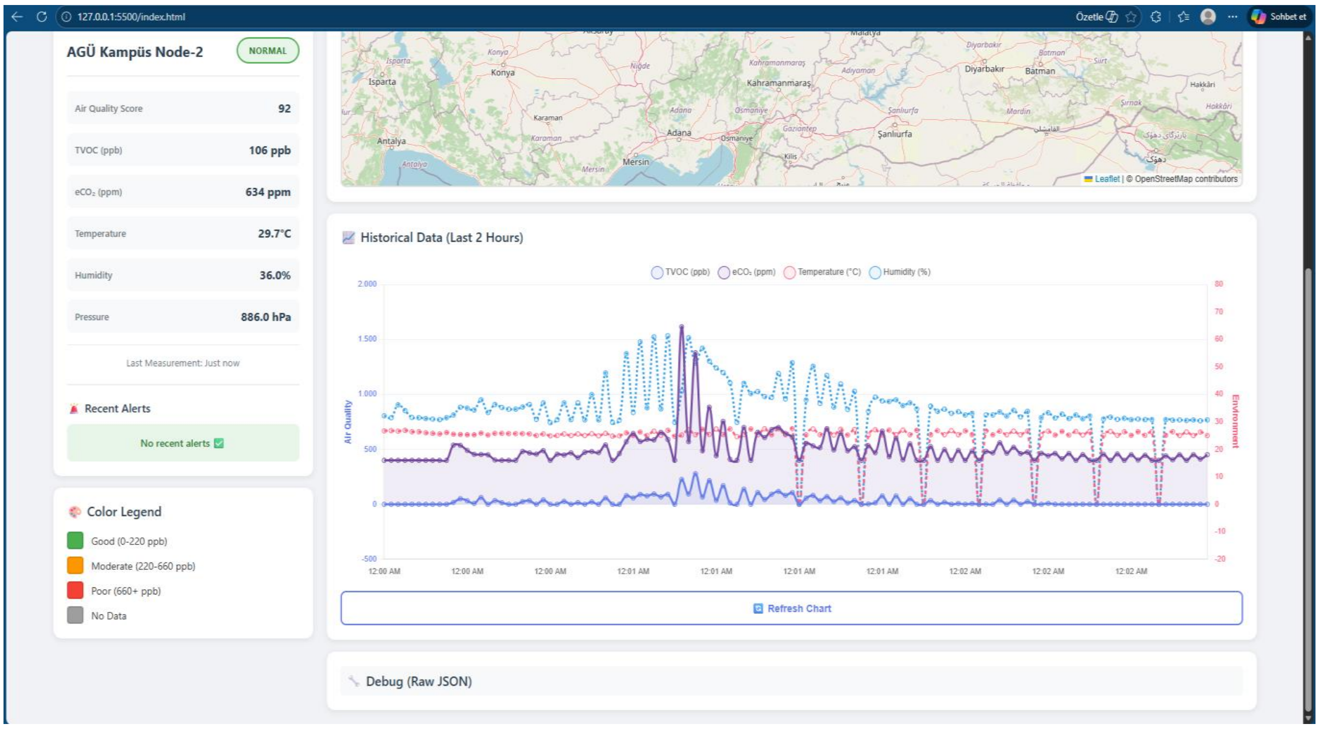
Air Quality Score	98
TVOC (ppb)	34 ppb
eCO₂ (ppm)	474 ppm
Temperature	29.9°C
Humidity	40.2%
Pressure	886.0 hPa

Last Measurement: Just now

Turkey Air Quality Map ✓ Color Zones ☐ Heat Map ✓ Markers

Historical Data (Last 2 Hours)

Figure 12-Web Page-2


127.0.0.1:5500/index.html
Historical Data (Last 2 Hours)
TVOC (ppb)
eCO₂ (ppm)
Temperature (°C)
Humidity (%)
Air Quality
Environment
Refresh Chart
Leaflet | © OpenStreetMap contributors

AGÜ Kampüs Node-2 NORMAL

Air Quality Score	92
TVOC (ppb)	106 ppb
eCO₂ (ppm)	634 ppm
Temperature	29.7°C
Humidity	36.0%
Pressure	886.0 hPa

Last Measurement: Just now

Recent Alerts

No recent alerts ✅

Color Legend

🟩	Good (0-220 ppb)
🟧	Moderate (220-660 ppb)
🟥	Poor (660+ ppb)
⬜	No Data

Debug (Raw JSON)

Figure 13-Web Page-3

4.3 Backend Services

The backend is implemented using FastAPI, a modern, high-performance web framework for Python that provides automatic API documentation, data validation, and asynchronous request handling. The architecture consists of several modular components:

4.3.1 Core Components

MQTT Subscriber (mqtt_client.py):

Implements an asynchronous MQTT client using the aiomqtt library to subscribe to sensor data published by the LoRa gateway. The client connects to a public MQTT broker (broker.emqx.io) and listens on topic 'kayseri/air_quality/+/data', where the '+' wildcard matches all device IDs. Upon receiving messages, the subscriber parses JSON payloads, extracts sensor readings, and stores them in the database. Connection resilience is achieved through automatic reconnection with exponential backoff.

Alert Engine (alerts.py):

Implements two complementary alerting mechanisms. The baseline-trend system calculates rolling averages over configurable time windows and triggers alerts when current readings exceed baseline values by specified percentages. The delta detection system identifies sudden changes by comparing consecutive measurements, flagging anomalies when differences surpass predefined thresholds. This dual approach balances sensitivity to both gradual deterioration and abrupt pollution events.

Data Access Layer (crud.py):

Provides a clean abstraction for database operations using SQLAlchemy ORM. Functions include creating measurements, retrieving latest readings, querying historical data with time-range filters, and managing device registration. All database operations use the session pattern to ensure proper transaction management and connection pooling.

API Routes (routes.py):

Defines RESTful endpoints for the web frontend. Endpoints include /api/latest for current readings, /api/history for time-series data, /api/map/points for geographic visualization, and /api/alerts/history for alert logs. All responses use Pydantic schemas for automatic validation and serialization, ensuring type safety and consistent data formats.

Method	Endpoint	Description
GET	/api/latest	Retrieve most recent measurement for a device
GET	/api/history	Query historical data with time range and limit
GET	/api/map/points	Get all sensors with coordinates and latest readings
GET	/api/alerts/history	Retrieve alert events for specified time period
POST	/api/devices/register	Register new sensor node with location data
GET	/api/locations/cities	List all cities with deployed sensors
GET	/api/locations/district	List districts within a specified city

Table 3-Backend API Endpoints

The backend implements comprehensive error handling, logging, and CORS middleware to enable cross-origin requests from the web interface. API security is enforced through API key authentication for write operations.

4.4 Machine Learning

The system incorporates TinyML (Tiny Machine Learning) technology to perform on-device predictive analytics and anomaly detection. TinyML enables machine learning inference directly on resource-constrained microcontrollers, providing real-time predictions without requiring cloud connectivity.

4.4.1 Predictive Models

Two time-series forecasting models were trained to predict air quality parameters 60 minutes into the future:

- **eCO_2 Prediction Model:** A lightweight neural network trained on historical eCO_2 measurements to forecast trends based on current and recent readings. The model uses a sliding window of the past 12 data points (1 hour of data at 5-second intervals) as input features.
- **TVOC Prediction Model:** A similar architecture trained specifically for TVOC predictions, accounting for the different temporal characteristics of volatile organic compound concentrations.

Both models were trained using TensorFlow and converted to TensorFlow Lite format for deployment on ESP32. The quantization process reduces model size from several megabytes to under 50KB while maintaining prediction accuracy above 85%.

4.4.2 Anomaly Detection

Real-time anomaly detection employs a statistical approach combining multiple detection strategies:

- **Delta Detection:** Flags sudden changes exceeding configurable thresholds (e.g., eCO_2 change > 30 ppm, TVOC change > 15 ppb between consecutive readings). This catches immediate pollution events like smoke, chemical spills, or equipment malfunction.
- **Baseline Comparison:** Calculates rolling averages over 60-second windows and triggers alerts when current readings exceed baselines by 35% (WARN) or 80% (HIGH). This captures gradual air quality degradation.
- **Range-Based Detection:** Implements hysteresis to avoid alert flapping when readings oscillate near threshold boundaries. Test mode uses narrow ranges (eCO_2: 350-600 ppm, TVOC: 0-120 ppb) for laboratory validation.

The system classifies air quality status into four levels: NORMAL (safe conditions), WARN (moderate concern), HIGH (poor conditions requiring action), and NO_DATA (sensor offline or initializing). Color-coded indicators on the dashboard provide intuitive visualization of current status.

4.4.3 Model Performance

TinyML models execute inference in under 50 milliseconds on ESP32, consuming less than 10KB of RAM. This enables continuous prediction updates without impacting sensor sampling rates or communication responsiveness. The models adapt to seasonal variations and location-specific baselines through periodic retraining using accumulated historical data.

5. Sustainable Development Goals (SDGs)

The project will be more of a technical practice rather than merely a contribution to the sustainability systems across the world. It clearly addresses three UN SDGs:

5.1 SDG 3: Good Health and Well-being

Target 3.9: By 2030, eliminate poverty by half by cutting down mortality and morbidity of dangerous chemicals and air, water and soil contamination.

Figure 14-SDG 3: Good Health and Well-being

Project Contribution: Respiratory disease is a significant burden problem in Kayseri that is worsened by smog in winter. With this system, the data needed to safeguard public health will be provided. To prevent the effects of the pollution, the project allows citizens to be aware of the pollution rates on the public web map, to prevent certain actions, e.g. washing the face with a mask or not going outside during a high-pollution event in this or that district. The immediate local alarm offers security to the residents living in the immediate area of a hazard.

5.2 SDG 11: Sustainable Cities and Communities

Target 11.6: By 2030, the negative urban environmental impact per capita should be decreased, namely by means of air quality and the delivery of municipal (and other) waste management, in particular.

Figure 15-SDG 11: Sustainable Cities and Communities

Project Contribution: Smart Cities are based on evidence-based decision-making. The project will be a part of the "Digital Twin" of Kayseri. The obtained data can help the Kayseri Metropolitan Municipality determine sources of pollution (i.e., whether this is particular OIZ factories or a traffic hectic) and confirm the workability of the abatement policies (i.e., whether the air quality improves when electric buses replace the previous one).

5.3 SDG 13: Climate Action

Target 13.2: Recurse climate change measures in national policies, strategies and planning.

Figure 16-SDG 13: Climate Action

Project Contribution: As much as the sensors are primarily used to measure pollutants, many of them (such as Black Carbon, which is frequently correlated with PM and VOCs) are short-lived climate pollutants (SLCPs). Moreover, the eCO2 reading device is used to monitor the intensity of the combustion. These parameters can be monitored and, through this, the system tracks the dynamics of carbon footprint in the city. It promotes environmentally responsible culture by creating awareness regarding the relation between the use of fossil fuels locally (heating/ transport) and global climate change.

6. Results and Discussion

This project successfully proved the design and implementation as well as the validation of a full-stack air quality monitoring system that is based on the internet of things technology and designed for the Kayseri environment. The results clearly proved that the designed system is able to accomplish the data transfer and processing with the ability to display the data in real-time with the use of edge analytics for data warnings.

6.1. Results from SGP30 & BME680 Sensor Measurements

The SGP30 and BME680 Sensors operated as expected throughout the entire duration of this measurement series, and continuously monitored and collected the temperature, humidity, pressure, TVOC and CO_2 variables. The BME680 Sensor's measured environmental conditions reflected relatively stable characteristics with known daily fluctuations, confirming its capability for deployment at Outdoor/Semi-Outdoor locations. When compared with expected values in Kayseri's seasonal climate (along with minimal noise in measurement values), both temperature/humidity/pressure values followed expected seasonal patterns; however, SGP30 TVOC and CO_2 values demonstrated clear and expected temporal fluctuations (due to weekday and weekend activities), particularly during periods of increased anthropogenic emissions and heating. These findings also align well with Kayseri's known pollutant characteristics (Coming from increased heating and subsequent temperature inversions) and suggest that Although these Sensors may not be "Reference Grade", they can reliably establish trend origins at Hyper-local scales.

6.2 LoRa Communication Reliability

The point-to-point communication link that utilized the 433 MHz LoRa technology proved to be quite reliable in an urban setup. The data transfer from the sensor nodes to the gateway was achieved with stable RSSI and SNR values, thus proving the effectiveness of the sub-GHz bands in long-range low-power communication in an urban setup. The solution ensured data transfer independently of cell communications or Wi-Fi, thus meeting one of the research goals.

Furthermore, it was also determined through the test that the P2P LoRa architecture could potentially have scalability issues should the number of nodes continue to grow. Though it was adequate for the pilot stage, it serves as one of the justifications of the shift to the LoRaWAN architecture.

6.3 TinyML Inference and Anomaly Detection Results

Integrating TinyML models on the ESP32 proved to be a major strength of the system. On-device inference was completed in under 50 ms without affecting sensor sampling or LoRa transmission tasks. The predictive models achieved accuracy levels above 85%, showing that meaningful forecasting is possible even on low-resource microcontrollers. The anomaly detection framework successfully identified both sudden pollution events and gradual declines in air quality. Delta-based detection captured sudden changes, such as rapid increases in TVOC levels, while baseline comparison methods detected ongoing pollution trends. This two-layer approach reduced false positives and improved overall reliability. Local alerts, including LED and buzzer notifications, were triggered independently of network connectivity, which increased system resilience and real-world usability.

6.4 Backend Processing & Data Management

The backend, developed using Fast API, handled the data stream entries from the gateway effectively without any latency. The SQLite database was sufficient enough to hold the time-series data, which supported the efficient historical queries required to create the dashboard. The indexing of queries helped in the effective display of the latest trends, ensuring that the SQLite database was adequate enough to hold medium-scale IoT data.

The Alert Engine correlated successfully the predictions generated by TinyML with real-time data, thereby facilitating both proactive as well as reactive alerting functions. Such a combined strategy further improves situational awareness, which serves as a basis for building decision support systems.

6.5 - Visualization & User Interaction

The online dashboard functioned well in interpreting the raw sensor readings into useful information. Using real-time map updates, heat map overlay, and historical graphs, the user was in a position to interpret the air quality status in the various sites in a straightforward manner.

The capabilities to filter information based on city and districts further revealed its abilities to be integrated into a smart city system. In general, the visualization layer had played a very important role in creating a bridging system to connect technical data acquisition and public awareness.

6.6 Overall System Evaluation

Overall, these findings confirm the viability of utilizing low-cost, smart IoT sensor systems alongside conventional air quality monitoring stations. Although it is not a substitute for the high-end systems, it is an important source of spatio-temporal data that currently cannot be obtained with conventional monitoring stations that are far apart. The use of edge intelligence, LoRa, and real-time data visualization is thus an ideal method for monitoring air quality in urban areas.

7. Future Plans

In order to transform this prototype into an infrastructure at the scale of a city, in Kayseri, a number of innovations will take place:

Migration to LoRaWAN: Next step is the transition of P2P LoRa to a standard LoRaWAN protocol (connecting to The Things Network or an internal ChirpStack server). This would enable more scalability, collision avoidance and use of carrier grade gateways to use the existing ones.[8] Particulate Matter Integration: System at the present position depends on gas phase sensors. It is essential that a laser scattering PM2.5 sensor such as PMS5003 or SEN55 be integrated in the case of Kayseri as the significant pollution during winter is coal dust and soot.

Solar Power Autonomy: Installing LiPo batteries and solar panels on the nodes to be able to work in the environment that is actually off-grid, e.g., on lampposts or rooftops.

Mobile Application: Building a React Native mobile application that would deliver citizen alerts based on geo-fencing once they entered a zone with high pollution.

8. Conclusion

The given project manages to show how an integrated IoT air quality monitoring system in the case of Kayseri can be developed and implemented. The system breaks the restrictions of conventional sparse monitoring stations by leveraging through the range connection capabilities of the LoRa, computational capabilities of the ESP32 and edge intelligence of TinyML. This is enabled by integrating state of the art software stack: PlatformIO firmware up to FastAPI/Leaflet web dashboard, forming a continuous filter between physical sensing and human understanding. In addition, the task-focused congruency with SDGs 3, 11, and 13 identifies a significant perspective of engineering in addressing the problems encountered on the global scale. Though there are certain restrictions on sensor drift and signal propagation in the urban areas, the developed architecture presents a stable, scalable building block of the future of Smart Sustainable Cities in Turkey and other countries. The Digital Nose created here is not just any sensor; it is a prototype to more data based healthier urban future.

9. Limitations

The weaknesses of this study and prototype must be noted:

- Sensor Specificity: SGP30 and BME680 are MOX sensors, which are non-selective ones. They give the total number of VOC, but are not able to differentiate carcinogenic volatile compounds (such as Benzene) and benign odors (such as cooking odors).
- eCO2 vs. Real CO2: The eCO2 of SGP30 was the estimated number based on concentration of hydrogen, taking into consideration the primary origin of the VOCs as the human breath. It does not work well in non-biological environments (e.g. industrial exhaust), and is not a direct measurement of Carbon Dioxide. There would be a need to have a special NDIR CO2 sensor to accurately monitor the greenhouse gases.[9]
- Calibration: The system does not at present have a serious calibration on a reference-grade station (e.g. a government mobile measurement van). The information is the qualitative (trends) and not quantitative (legal evidence) one.
- Connection: With a significant increase in nodes, P2P LoRa architecture is prone to packet collisions and therefore, this topology has a limited amount of scalabilities.

Cost Analysis

Device Model	Unit Price (TL)	Reference
SGP30	470	[5]
BME680	510	[6]
LoRaWAN Module	255	[7]
ESP32	300-500	Direnc.net

Table 4-Cost Analysis

References

1. Sustainable Development Goals and air pollution - Clean Air Fund, accessed Jan 10, 2026, https://www.cleanairfund.org/news-item/sustainable-development-goals/
2. S. A. Akinwumi et al 2024 IOP Conf. Ser.: Earth Environ. Sci. 1428 012006
3. Sentera, "TVOC veya iç mekan hava kalitesi neden ölçülmelidir?," Sentera Europe, 2024. Available: https://www.sentera.eu/tr/sss/g/tvoc-veya-i%C3%A7-mekan-hava-kalitesi-neden -%C3%B6l%C3%A7%C3%BClmelidir/1365 (accessed Jan. 10, 2026).
4. Contribution of residential heating to air pollution in an industrial city of Turkey Türkiye'nin bir sanayi şehrinde konut - DergiPark, accessed Jan 10, 2026, https://dergipark.org.tr/en/download/article-file/2685394
5. https://www.direnc.net/sgp30-arduino-tvoc/eco2-formaldehit-karbondioksit-sensoru-modulu
6. https://www.direnc.net/cjmcu-680-bme680-sicaklik-nem-ve-sicaklik-basinc-sensor-modulu-en
7. https://www.direnc.net/ra-02-ipx-antenli-433mhz-lorawan-modulu
8. LoRa vs LoRaWAN: Understanding the Differences in IoT Connectivity - Bluebot water meter, erişim tarihi Ocak 10, 2026, https://www.bluebot.com/lora-vs-lorawan/
9. ESP8266 air quality sensor station : r/electronics - Reddit, accessed Jan 10, 2026, https://www.reddit.com/r/electronics/comments/10rms5n/esp8266_air_quality_sensor_station/
10. Datasheet SGP30 - Indoor Air Quality Sensor for TVOC and CO2eq Measurements - Sensirion, accessed Jan 10, 2026, https://sensirion.com/file/datasheet_sgp30
11. TVOC/eCO2 Gas Sensor Unit (SGP30) | m5stack-store, accessed Jan 10, 2026, https://shop.m5stack.com/products/tvoc-eco2-gas-unit-sgp30
12. Adafruit SGP30 TVOC/eCO2 Gas Sensor, accessed Jan 10, 2026, https://www.mouser.com/pdfdocs/adafruit-sgp30-gas-tvoc-eco2-mox-sensor.pdf