

DEPARTMENT OF BIOMEDICAL ENGINEERING

AIR QUALITY MONITORING

PHASE 3 DOCUMENT SUBMISSION



TEAM MEMBERS

SOWMIYA M – 732721121049

PAVITHRA C T – 732721121038

SWETHA G K – 732721121054

KANISHKA S – 732721121021

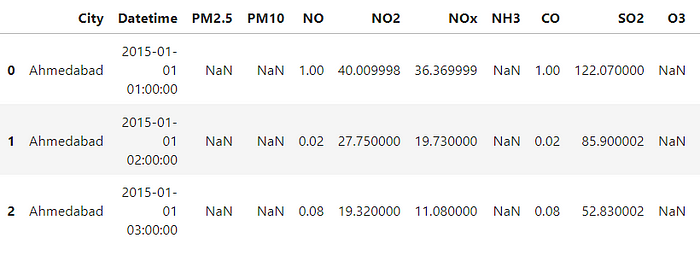
NISHANTHINI T – 732721121034

**INTRODUCTION:**

Air is a growing issue these days. It is necessary to monitor air quality and keep it under control for a better future and healthy living for all. Here we propose an air quality monitoring system that allows us to monitor and check live air quality in a particular areas through IOT. System uses air sensors to sense presence of harmful gases/compounds in the air and constantly transmit this data to microcontroller. The sensors interact with microcontroller which processes this data and transmits it over internet. This allows authorities to monitor air pollution in different areas and take action against it.

**DATASET:**

The dataset contains air quality data and AQI (Air Quality Index) at hourly and daily level of various stations across multiple cities in India. This project deals with exploration of the data and modelling to predict the classes namely : Good, Satisfactory, moderate, poor, very poor or severe cases of the air quality.



**DATA PREPROCESSING:**

// Define the analog pins for the MQ135 and MQ7 sensors

int mq135\_analog\_pin = A0;

int mq7\_analog\_pin = A1;

void setup() {

Serial.begin(9600);

}

void loop() {

// Read the analog values from the MQ135 and MQ7 sensors

int mq135\_sensor\_value = analogRead(mq135\_analog\_pin);

int mq7\_sensor\_value = analogRead(mq7\_analog\_pin);

// Convert the analog values to voltages

float mq135\_voltage = (mq135\_sensor\_value / 1024.0) \* 5.0;

float mq7\_voltage = (mq7\_sensor\_value / 1024.0) \* 5.0;

// Calculate the resistance of the sensors using a voltage divider formula

float mq135\_resistance = ((5.0 - mq135\_voltage) / mq135\_voltage) \* 10.0;

float mq7\_resistance = ((5.0 - mq7\_voltage) / mq7\_voltage) \* 10.0;

// Use the resistance values to estimate air quality for MQ135 and MQ7

int air\_quality\_mq135 = getAirQuality(mq135\_resistance);

int air\_quality\_mq7 = getAirQuality(mq7\_resistance);

// Print the results

Serial.print("MQ135 Air Quality: ");

Serial.print(air\_quality\_mq135);

Serial.println(" ppm");

Serial.print("MQ7 Air Quality: ");

Serial.print(air\_quality\_mq7);

Serial.println(" ppm");

delay(2000); // Delay for 2 seconds between readings

}

// Function to estimate air quality based on sensor resistance

int getAirQuality(float sensor\_resistance) {

// You will need to calibrate your sensor based on its datasheet

// These values are for reference and may need adjustments

if (sensor\_resistance <= 25) {

return 10000; // Very poor air quality

} else if (sensor\_resistance <= 50) {

return 8000; // Poor air quality

} else if (sensor\_resistance <= 100) {

return 5000; // Moderate air quality

} else if (sensor\_resistance <= 200) {

return 2000; // Good air quality

} else {

return 1000; // Excellent air quality

}

}

**OUTPUT:**





**CONCLUSION:**

An IoT-based air pollution monitoring system is a revolutionary solution that can provide accurate and real-time data about the air quality in a particular area. It can help identify the sources of pollution and take necessary measures to reduce it, protecting the environment and human health.