AN IOT BASED AIR QUALITY MONITORING SYSTEM

A Mini-Project Report Submitted by

Aniket Ajit Pai	Chris M	Chinmaya S Ram
4NM19CS020	4NM19CS046	4NM19CS044

Under The Guidance of

Dr. Venugopala P S

Associate Professor

Department of Computer Science and Engineering

As a part of the elective subject Internet of Things (18CSE21)

In The

Bachelor of Engineering in Computer Science & Engineering

From

Visvesvaraya Technological University, Belagavi

At

NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)

Nitte – 574 110, Karnataka, India

(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC

Phone: 08258 – 281039 / 281263, Fax: 08258 – 281265

ACKNOWLEDGEMENT

We take this opportunity to express our deep sense of gratitude to our guide Dr. Venugopal P S, Associate Professor, Department of Computer Science and Engineering, NMAMIT, Nitte.

We convey our sincere regards to Dr. Jyothi Shetty, Professor & Head, Department of Computer Science and Engineering, NMAMIT, Nitte for encouraging us during the project tenure.

Our thanks to Dr. Niranjan N. Chiplunkar, Principal, NMAMIT, Nitte for his inspiration and support while carrying out the project work.

I cordially acknowledge and give due credit to my teammate and family for the constant support they have extended to us and the faith they have shown in me during the tenure of this mini-project.

Aniket Ajit Pai	Chris M	Chinmaya S Ram
4NM19CS020	4NM19CS046	4NM19CS044

ABSTRACT

Internet of Things (IoT) technology has brought about a revolution to every field of the common man's life by making everything smart and intelligent. IoT can be proved very useful in the field of detecting pollution and air quality. The level of pollution is increasing rapidly due to various factors like industries, urbanization, excessive vehicle uses which can affect human health. An IOT Based Air Pollution Monitoring System is used to monitor the Air Quality over a cloud server using Internet facilities. This system makes it easy for an individual to perceive the details of the environment he is present in, thereby notifying them regarding the conditions of the environment, and also alerting them if in case, the pollutant levels are high to leave the place to be safe.

The objective of the project is to design an IoT-based model that enables the user to obtain details regarding the status of the surroundings they are present in. The model will have four sensors MQ-6, MQ-135, Particulate Matter 2.5, and DHT-11. Each of these sensors will be connected to an Arduino which will further be connected to a Raspberry Pi to achieve communication between both the devices through Universal Asynchronous Receiver/Transmitter (UART) Protocol. The system will be always active receiving data from the sensors and sending it to the Raspberry Pi. The Raspberry Pi device will further send the data using the API Write Key to the Thingspeak cloud, where channels are already created to receive the data. This data will be read using an app made with Flutter, where an API Read Key will retrieve the data from the cloud and analysis of data from each sensor will be produced via a graph.

The system can be installed anywhere, but mostly in industries and houses where gases are mostly to be found and gives an alert message when the system crosses threshold limit for a particular sensor data.

TABLE OF CONTENTS

INTRODUCTION

The Internet of Things (IoT) is the interconnection of physical gadgets, vehicles, home machines. It includes various things implanted with science, programming, sensors, actuators, and properties. Installed framework tackles each issue in a situation, however, arranged to inter-operate among the present web foundation. An individual can control questions remotely over a system. This besides makes open doors for direct incorporation of the physical world into PC-based frameworks, which brings about improved productivity, exactness, and monetary benefit, notwithstanding a decrease in human intervention.

Air pollution is the biggest problem of every nation, whether it is developed or developing. Health problems have been growing at faster rate especially in urban areas of developing countries where industrialization and growing number of vehicles leads to release of lot of gaseous pollutants. Harmful effects of pollution include mild allergic reactions such as irritation of the throat, eyes and nose as well as some serious problems like bronchitis, heart diseases, pneumonia, lung and aggravated asthma.

Poor air quality poses a significant risk to the vulnerable section of the society such as children, asthmatic, pregnant women, and the elderly persons. As per WHO statics, millions of premature death cases are reported due to air pollution every year worldwide. Due to this, in recent years, the air quality of the cities has become one of the major causes of concern around the world. Thus, it is necessary to constantly monitor the air quality index of a city to make it smart and livable.

Around the world, governments are building the smart cities to keep a check on these problems and provide a healthy life for its inhabitants. The Indian government is in the process to build 100 smart cities by 2050. These cities will utilize advanced communication network, wireless sensor networks (WSNs), and intelligent system to solve future challenges and create new services. Although the Indian cities have installed real-time air quality monitoring systems in Delhi and other cities, low-cost IoT enabled WSN technology is the future for the coming smart cities around the world.

Real-time monitoring of the air quality requires the live data transfer between the devices over the internet and it can be visualized using an Android Application. It reduces the mobilization of system hardware at different locations, which is cost efficient as only one-time installation cost is involved.

PROBLEM STATEMENT

An IoT-based Air Quality Monitoring System which can monitor the quality and status of the environment.

PURPOSE & REQUIREMENTS SPECIFICATION

Purpose:

To design a real-time air quality monitoring system which collects the air quality information which is sent to the cloud, and which is further retrieved to an application from the cloud.

Requirements Specification:

Data collection requirement: The air quality monitoring system should be active at all times, and be able to read data from the sensors and send it to the cloud at regular intervals for storage.

System management requirement: The system should provide remote monitoring functions.

Application Deployment Requirement: The application will be deployed locally on the device, which will also be remotely available.

User Interface Requirement: The application will be deployed locally on the device, which will also be remotely available.

COMPONENTS REQUIRED

- ❋ 1 x Raspberry Pi Microprocessor

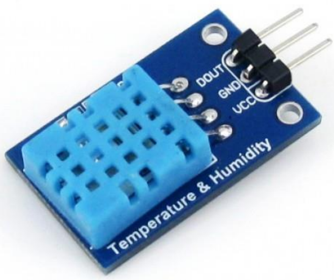
- ❋ 1 x Arduino Uno

- ❋ 1 x MQ135 Gas Sensor

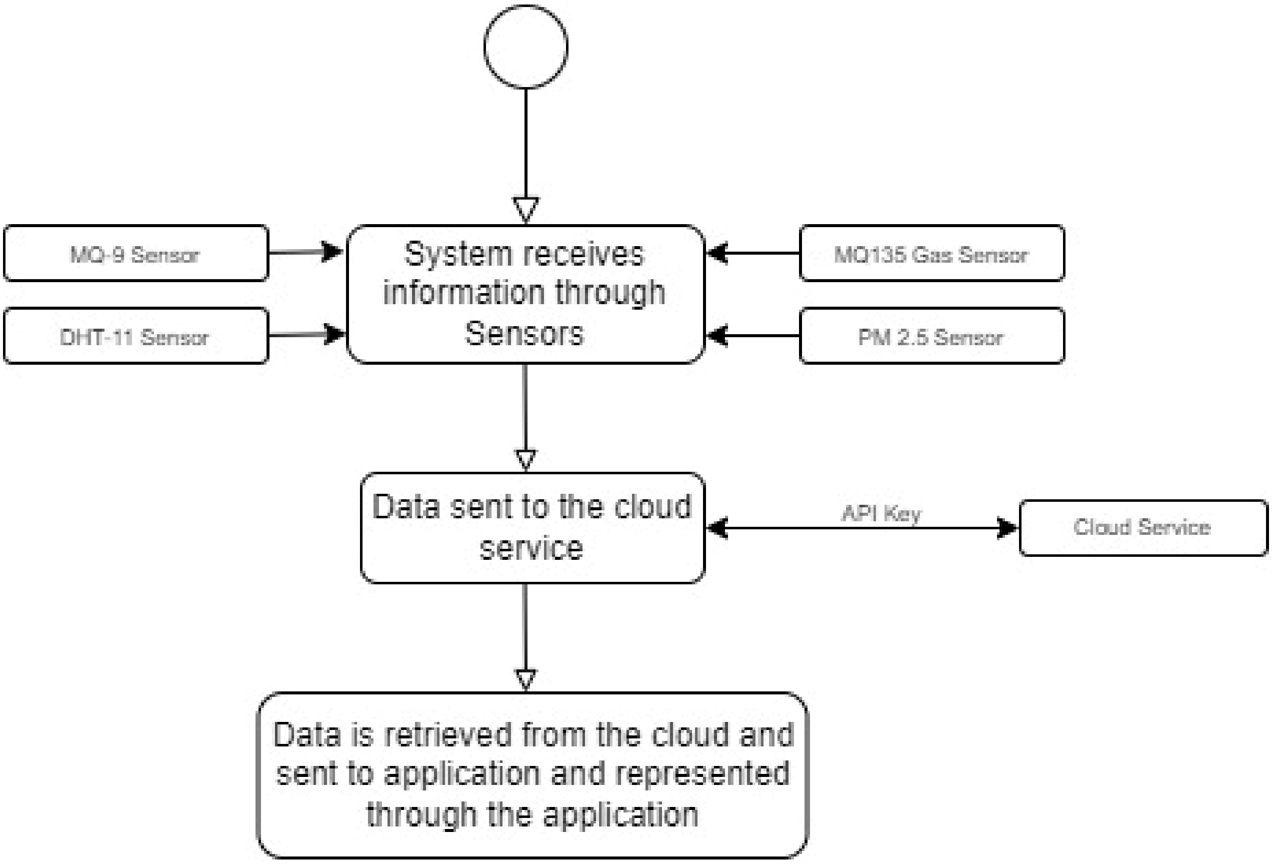
❋ 1 x MQ-6 LPG Gas Sensor

❋ 1 x PM2.5 Air Quality Sensor

❋ 1 x DHT-11 Digital Temperature and Humidity Sensor



PROCESS SPECIFICATION



The IoT based Air Quality Management System takes the data received from the sensors connected to an Arduino board as input. The gathered data is then sent to a Raspberry Pi microprocessor.

The data is analyzed and sent to the cloud using the API Write Key and is updated constantly whenever there is a change in the air quality.

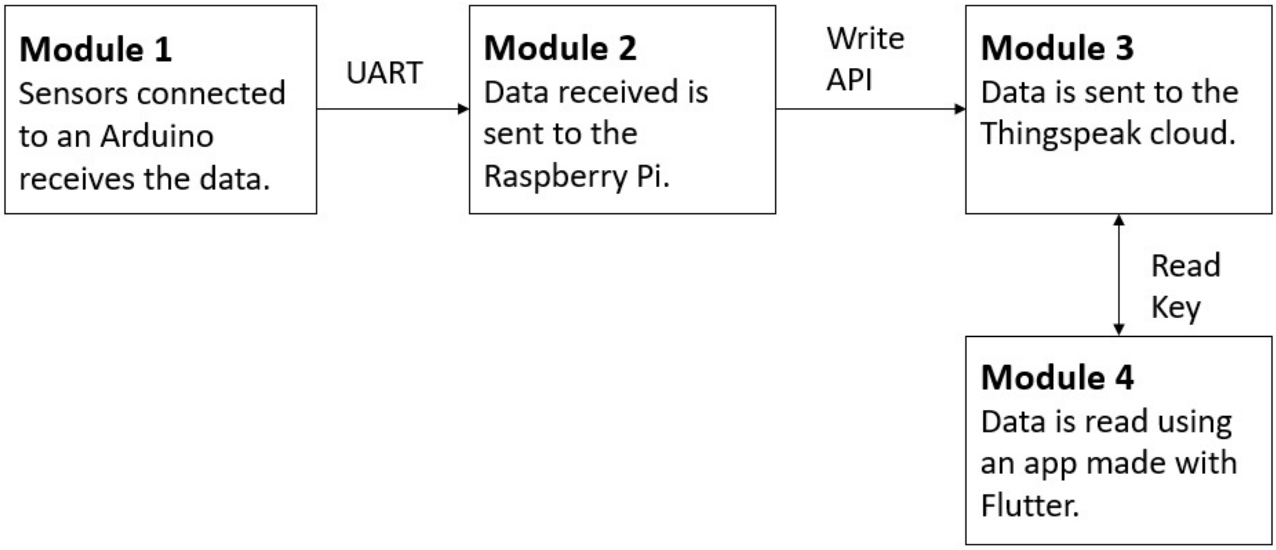
Data is then retrieved from the cloud using the API Read Key and then sent to an application where analysis of data from each sensor will be produced via a graphical/pictorial representation.

SERVICE SPECIFICATIONS

Functionalities Performed:

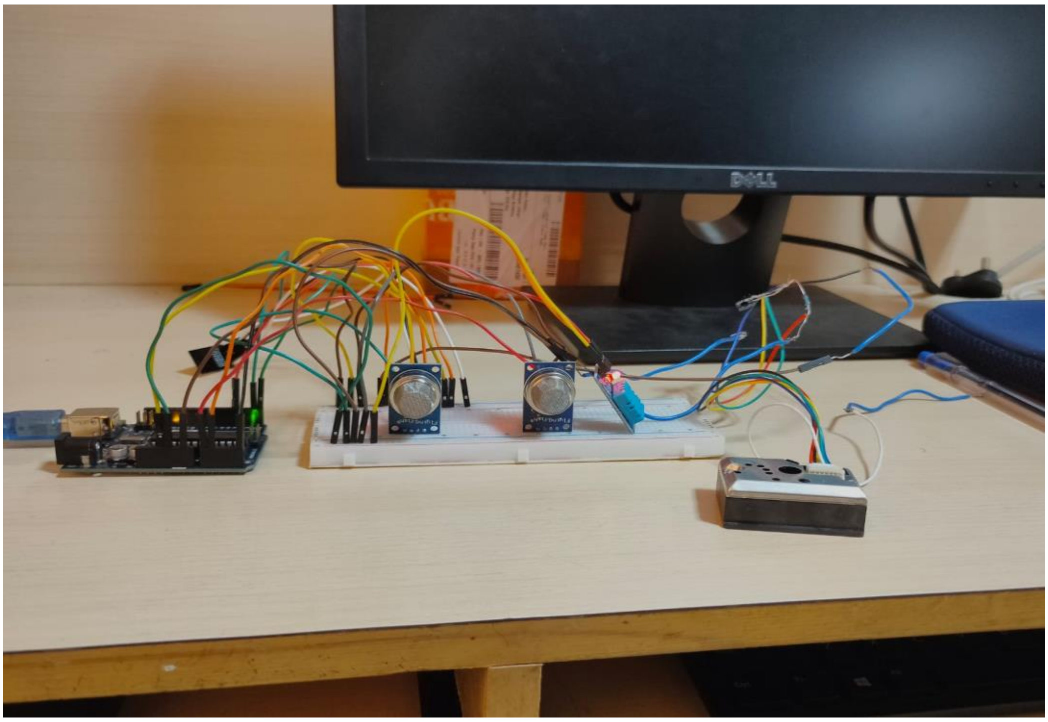
- ✓ The Arduino constantly reads data through the sensors.
- ✓ The data is sent from Arduino to Raspberry Pi.
- ✓ The data is further stored on the cloud with the help of Raspberry Pi.
- ✓ A Flutter app keeps track of the changing sensor values and displays it to the users.

Control flow:

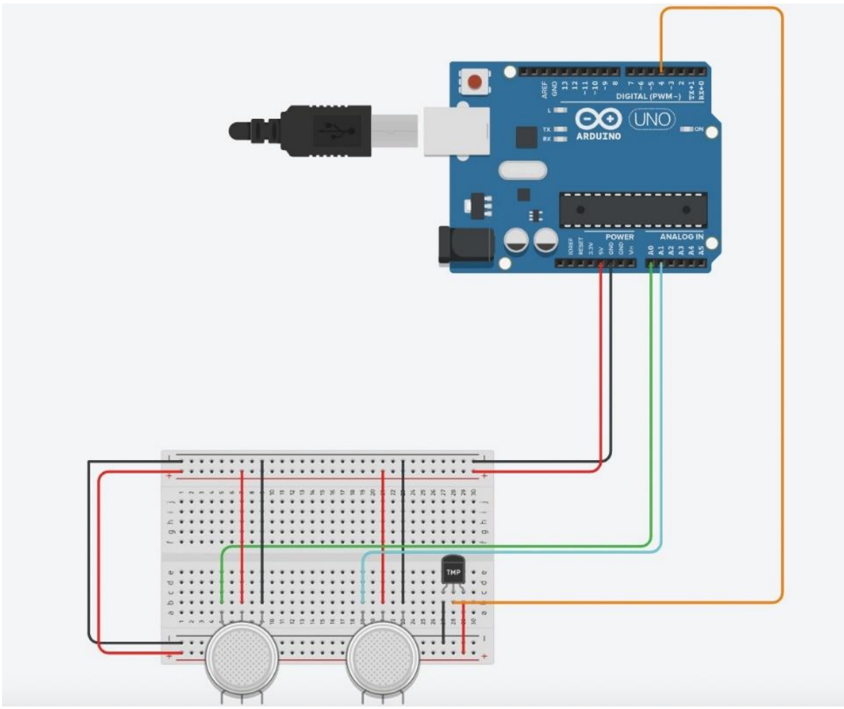


FUNCTIONAL VIEW SPECIFICATION

i. Demonstration of the project setup



ii. Arduino Circuit Connection



a) Using the Thonny IDE in the Raspberry Pi OS the following can be achieved:
   - Takes the data from Raspberry Pi .
   - Sends the data to the Thingspeak Channel.

```
import serial
import time
import thingspeak
import threading
import requests
import json
import urllib.request
import time
import RPi.GPIO as GPIO


channel_id = 1752477
write_key = '6CQD9T0EAK439XTI'


lst = []

ser=serial.Serial("/dev/ttyACM0",9600)  #change ACM number as found from ls
/dev/tty/ACM*
ser.baudrate=9600
GPIO.setmode(GPIO.BOARD)

while True:
    read_ser=ser.readline().decode("utf-8")
    read_ser = int(read_ser)
    lst.append(read_ser)
    while len(lst) == 5:
        print(lst)
        NEW_URL5 =
'https://api.thingspeak.com/update?api_key=6CQD9TOEAK439XTI&field1={}&field2={}&f
ield3={}&field4={}&field5={}'.format(lst[0],lst[1],lst[2],lst[3],lst[4])
        print(NEW_URL5)
        data = urllib.request.urlopen(NEW_URL5)
         print(data)
        lst.clear()
        time.sleep(16)
```

```
if __name__ == "__main__":
    channel = thingspeak.Channel(id = channel_id, api_key = write_key)
```

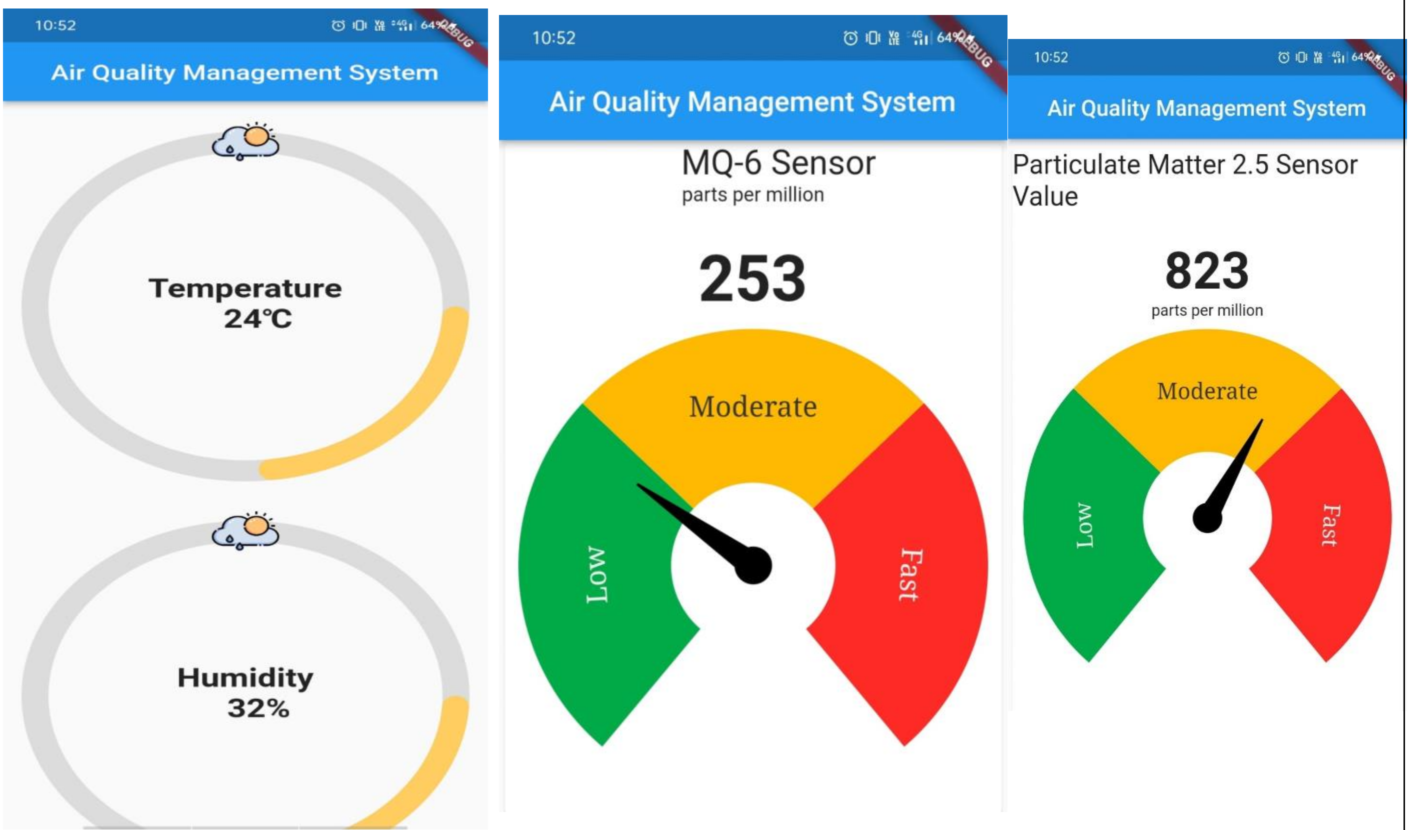
b) Arduino Uno code (C++) is dumped on to the hardware for the following functionalities:

- Reads the sensor values.

```
#include <dht11.h>
#define DHT11PIN 4
dht11 DHT11;
int A[5] ={};

int x, sensorValue;
int dhtValue = 0;
int measurePin=A5;  //PM2.5

int samplingTime =280;
int deltaTime =40;
int sleepTime =9680;

float voMeasured = 0;
float calcVoltage= 0;
float dustDensity= 0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(A0, INPUT);
  pinMode(A1, INPUT);

}

void loop() {
  // put your main code here, to run repeatedly:
  delayMicroseconds(samplingTime);
  voMeasured = analogRead(measurePin);
  delayMicroseconds(deltaTime);
  delayMicroseconds(sleepTime);           //PM2.5
  calcVoltage = voMeasured*(5.0/1024.0);
  dustDensity = 170*calcVoltage-0.1;

  x = analogRead(A0); // MQ-6 Data A0
  sensorValue = analogRead(A1);   //MQ-135 Data A1

  // MQ6 - MQ135 - PM2.5 - DHT11  HUMIDITY - TEMP
  A[0] = x;
  delay(500);

  //Serial.println("MQ-135 Value: ");
  //Serial.println(sensorValue);
  A[1] = sensorValue;
  delay(500);
```

```
//Serial.println("PM2.5 Value: ");
//Serial.println (dustDensity);
A[2] = dustDensity;
delay(500);

//DHT sensor
//Serial.println();

int chk = DHT11.read(DHT11PIN);

//Serial.print("Humidity (%): ");
//Serial.println((float)DHT11.humidity, 2);
A[3] = (int)DHT11.humidity;
//Serial.print("Temperature (C): ");
//Serial.println((float)DHT11.temperature, 2);
A[4] = (int)DHT11.temperature;

delay(500);

for(int i = 0; i<5; i++){
  Serial.println(A[i]);
  delay(500);
}


}
```

c) Application screenshot



OPERATIONAL VIEW SPECIFICATION

A. The Arduino is active at all times.

B. The sensors connected to the Arduino produce data which is further sent to the Raspberry Pi using UART.

C. The Raspberry Pi sends these data to the Thingspeak cloud.

D. When sufficient data is collected, a mobile app can be used in order to retrieve the data from the Thingspeak cloud using the APIs.

E. The system performs analysis of the data sensor and updates the cloud for every usage.

CONCLUSION

- This project is a basic implementation of retrieving data from the sensors and sending them to the cloud, which is then retrieved by the android app.
- We have used Arduino to take the inputs from the sensors, and Raspberry Pi to upload the data to the cloud; There are 5 data values, MQ-6 sensor, MQ-135 sensor, DHT Temperature, DHT Humidity, and PM2.5 Sensor
- We have implemented an IoT based Air Quality Management System for live monitoring.
- The system has efficiency and accuracy in fetching the live data from users.
- Also, the system gives an alert when any of the given data retrieved from the sensor are way above the threshold value.
- This IoT based Air Quality Management System will always provide a way for individuals to monitor the air quality and take good care of their health.

FUTURE SCOPE

- Can be introduced as a part of smart cities model.
- More sensors can be added to retrieve data.
- A Wi-Fi module can be introduced instead of using Raspberry Pi.
- This model can use actuators in industries, to perform various functions.