GOVERNMENT OF TAMILNADU

DIRECTORATE OF TECHNICAL EDUCATION, CHENNAI

NAAN MUDHALVAN SCHEME (TNSDC) SPONSORED

STUDENTS DEVELOPMENT PROGRAMME

ON

IoT AND ITS APPLICATIONS

"IOT BASED AIR QUALITY MONITORING SYSTEM"

HOST INSTITUTION

XXXX

COIMBATORE - 04

TRAINING PARTNER

ENTHU TECHNOLOGY SOLUTIONS INDIA PVT LTD

NAME	ROLL NO
Name 1	
Name 2	
Name 3	
Name 4	
Name 5	
Name 6	

Table of contents

IOT BASED AIR QUALITY MONITORING SYSTEM

ABSTRACT:

In recent years, the quality of air has become a significant concern due to the increasing levels of pollution and its adverse effects on human health and the environment. This project, IoT-based Air Quality Monitoring System, aims to develop a cost-effective and efficient solution for monitoring air quality in real-time. The system utilizes an MQ135 gas sensor to detect harmful gases such as ammonia, benzene, and alcohol, which are key indicators of air pollution. Additionally, a DHT11 sensor is integrated to measure environmental factors like temperature and humidity, providing a comprehensive assessment of the air quality. The collected data is transmitted to a cloud platform using IoT technology, enabling remote monitoring and analysis. This real-time data can be accessed from anywhere, allowing users to stay informed about the air quality in their environment and take necessary actions to mitigate potential health risks. The system's ability to monitor and report air quality continuously makes it a valuable tool for individuals and organizations concerned with environmental safety and public health. Overall, this project demonstrates the potential of IoT in environmental monitoring, offering a scalable and accessible solution to the global challenge of air pollution.

INTRODUCTION:

Air quality monitoring has become increasingly important in today's world, where pollution levels are on the rise, and the effects of poor air quality are more evident. The need for real-time, accurate, and accessible air quality data has never been more critical. This project, titled IoT-based Air Quality Monitoring System, aims to address this need by leveraging the Internet of Things (IoT) to create a system that continuously monitors and reports air quality parameters. The system is equipped with an MQ135 gas sensor and a DHT11 sensor. The MQ135 is responsible for detecting various harmful gases such as ammonia, benzene, and alcohol, which are common indicators of air pollution. Meanwhile, the DHT11 sensor monitors environmental parameters like temperature and humidity, providing a more comprehensive overview of the air quality. By integrating these sensors with an IoT platform, the system can send real-time data to the cloud, allowing for continuous monitoring from anywhere with an internet connection. This capability not only ensures that users are always informed about the air quality in their surroundings but also aids in identifying pollution trends and taking timely actions to mitigate health risks.

HARDWARE & SOFTWARE COMPONENTS USED:

DHT11/DHT22 Sensor

MQ135 Sensor

ESP32 Microcontroller

Breadboard

Resistors (10K Ω)

USB Cable (B-Type)

Jumper wires

Arduino IDE

Thingzmate cloud platform

HARDWARE COMPONENTS DESCRIPTION:

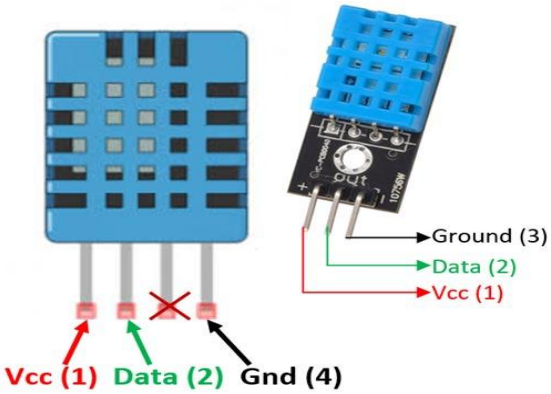
i) DHT11:

The DHT11 is a widely used temperature and humidity sensor that provides reliable measurements at a low cost. It combines a capacitive humidity sensor and a thermistor to measure the surrounding air and outputs a digital signal on the data pin. The sensor is commonly used in various applications such as weather monitoring, home automation systems, and environmental monitoring.

Key Features:

- **Temperature Range**: The DHT11 can measure temperatures from 0°C to 50°C with an accuracy of ±2°C.
- **Humidity Range**: It can measure humidity from 20% to 90% RH with an accuracy of ±5% RH.
- **Digital Output**: The DHT11 provides a digital output, making it easy to interface with microcontrollers like the ESP32.

- **Low Power Consumption**: The sensor operates at a voltage of 3.3V to 5V and consumes very little power, making it suitable for battery-powered applications.
- **Slow Sampling Rate**: The DHT11 has a sampling rate of 1 Hz (one reading per second), which is adequate for most environmental monitoring tasks but may not be suitable for real-time applications requiring fast response.



ii) MQ135 Sensor:

The MQ135 is a versatile gas sensor commonly used in air quality monitoring and environmental projects. It detects various harmful gases and pollutants, making it suitable for applications where air quality is a concern.

Key Features:

- **Multi-Gas Detection:** Capable of detecting CO2, NH3, benzene, alcohol, smoke, and other gases.
- **High Sensitivity:** Detects gas concentrations in the range of 10 to 1000 ppm.
- **Analog Output:** Provides an analog voltage output that corresponds to gas concentration levels.
- **Simple Interface:** Easy to connect with VCC, GND, and AOUT pins for power, ground, and analog output, respectively.
- **Heater Coil:** Includes a built-in heater that requires a warm-up period for accurate readings.

- **Low Cost:** Affordable and widely available for hobbyist and professional projects.
- **Operating Voltage:** Operates at 5V, compatible with most microcontrollers.
- **Wide Detection Range:** Suitable for detecting both low and high concentrations of gases.
- **Long Lifespan:** Durable with a long operational life under normal conditions.
- **Compact Size:** Small and easy to integrate into various devices and systems.

Applications:

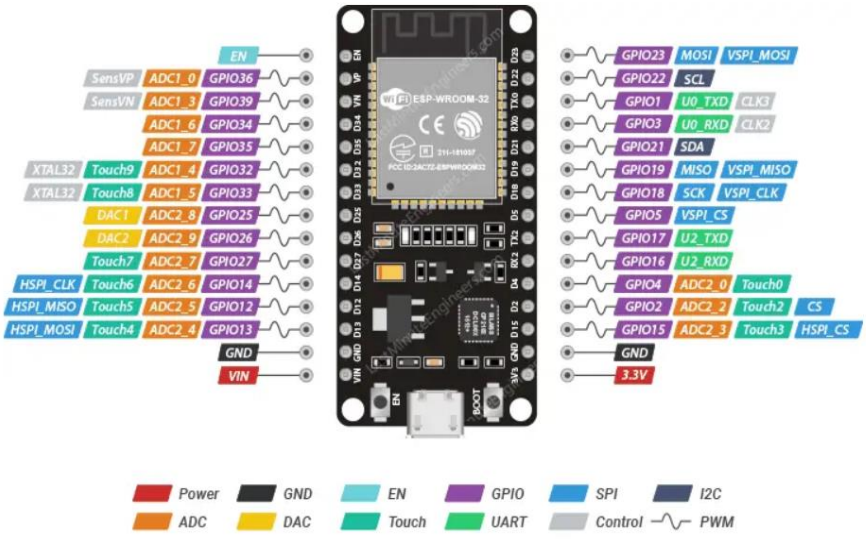
- **Air Quality Monitoring:** Used in devices for indoor and outdoor air quality assessment.
- **Safety Devices:** Incorporated in safety systems to detect harmful gas levels.
- **HVAC Systems:** Monitors air quality in heating, ventilation, and air conditioning systems.
- **Environmental Monitoring:** Employed in environmental stations for pollution tracking.
- **Smart Homes:** Integrated into smart home systems for automatic air quality control.

iii) ESP32:

The ESP32 is a highly versatile microcontroller developed by Espressif Systems, designed for a wide range of applications, particularly in the Internet of Things (IoT) space. It is renowned for its combination of high performance, integrated wireless connectivity, and a rich set of features, all at a low cost. The ESP32 is commonly used in projects that require both Wi-Fi and Bluetooth capabilities, making it suitable for smart home devices, sensor networks, and wearable technology.

Key Features:

- **Dual-Core Processor**: Features a dual-core Tensilica LX6 microprocessor running up to 240 MHz.
- **Connectivity**: Includes Wi-Fi (802.11 b/g/n) and Bluetooth (Classic and BLE).
- **Memory**: Typically comes with 520 KB of SRAM and supports external flash memory.
- **I/O Pins**: Offers numerous GPIO (General Purpose Input/Output) pins with various functionalities.
- **Peripherals**: Includes ADC, DAC, PWM, SPI, I2C, UART, and more.
- **Power Management**: Equipped with low-power modes for energy efficiency.



SOFTWARE DESCRIPTION:

i) ARDUINO IDE:

The Arduino Integrated Development Environment (IDE) is a powerful and user-friendly software application designed to facilitate the programming and uploading of code to Arduino microcontroller boards. It serves as the primary interface for developing, compiling, and debugging Arduino sketches (programs), making it an essential tool for anyone working with Arduino hardware.

- **Simple Interface:** User-friendly and intuitive design, suitable for beginners and experts.
- **Cross-Platform:** Available for Windows, macOS, and Linux.
- **Board Support:** Compatible with various Arduino boards (Uno, Nano, Mega, etc.).
- **Built-in Libraries:** Extensive libraries for sensors, displays, motors, and more.
- **Serial Monitor:** Tool for real-time communication and debugging via serial data.
- **Sketch Management:** Easily manage, save, and organize Arduino sketches (programs).
- **Library Manager:** Browse, install, and manage external libraries effortlessly.
- **Basic Debugging Tools:** Includes Serial Monitor and error indicators.
- **Easy Compilation & Uploading:** Simple process to compile and upload code to the board.
- **Community Support:** Backed by a large, active community with ample resources.
- **Extensible:** Supports third-party plugins for additional features.
- **Beginner-Friendly:** Ideal for those new to microcontrollers and electronics.
- **Open Source:** Free to use, modify, and share, encouraging innovation.
- **Continuous Updates:** Regular improvements and new features from the Arduino team.

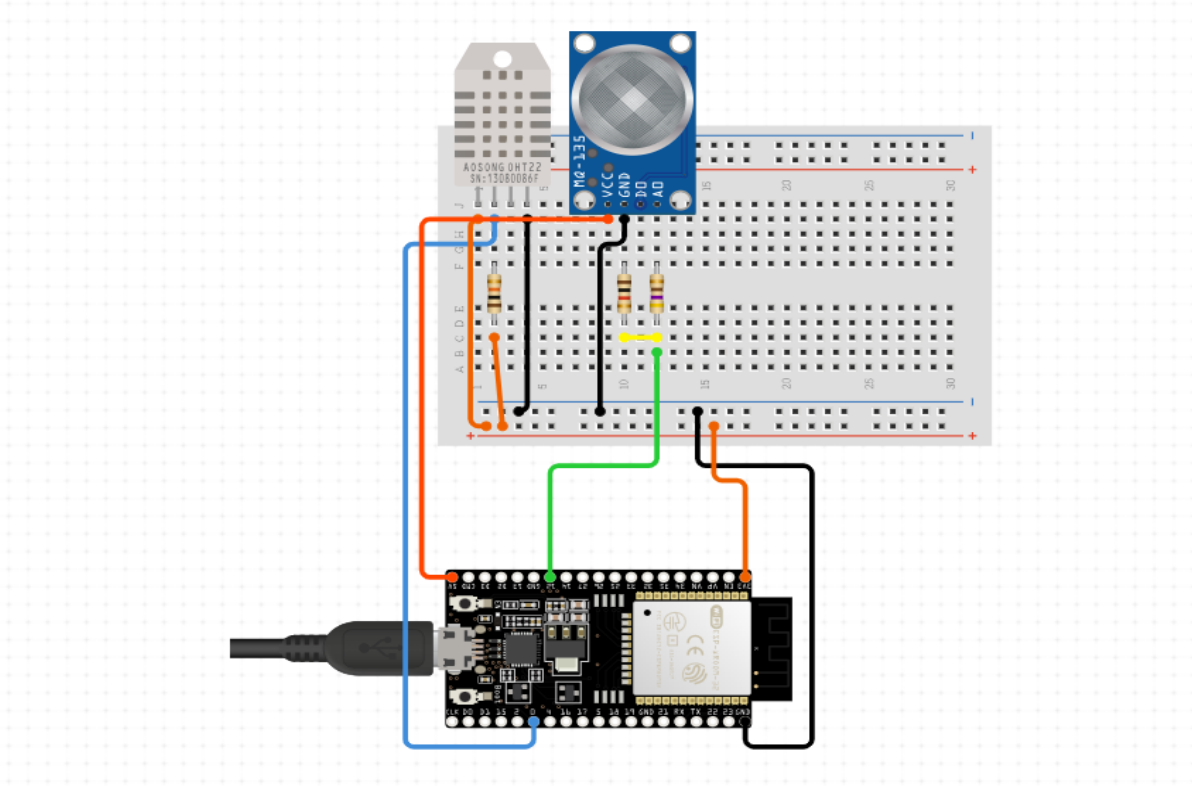
- **Versatile Applications:** Suitable for a wide range of projects, from simple to complex.

ii) THINGZMATE:

Thingzmate is a platform designed to streamline the development of IoT (Internet of Things) solutions by providing tools, services, and infrastructure that enable rapid prototyping, deployment, and management of IoT devices and applications. It typically offers features like device management, data visualization, cloud integration, and APIs, making it easier for developers to build, connect, and monitor IoT projects.

- **Device Management:** Centralized control of IoT devices, including remote updates.
- **Data Visualization:** Real-time data dashboards and insights.
- **Cloud Integration:** Seamless connection to cloud services for scalability.
- **APIs & SDKs:** Tools for easy integration with existing apps.
- **Security:** Robust encryption and authentication for device and data protection.
- **Rapid Prototyping:** Tools for quick IoT project deployment.
- **Scalability:** Supports projects of all sizes.
- **User-Friendly Interface:** Accessible for all skill levels.
- **Customizable Solutions:** Tailor IoT systems to specific needs.
- **Support:** Access to technical help and a user community.

CIRCUIT DIAGRAM:



CODE:

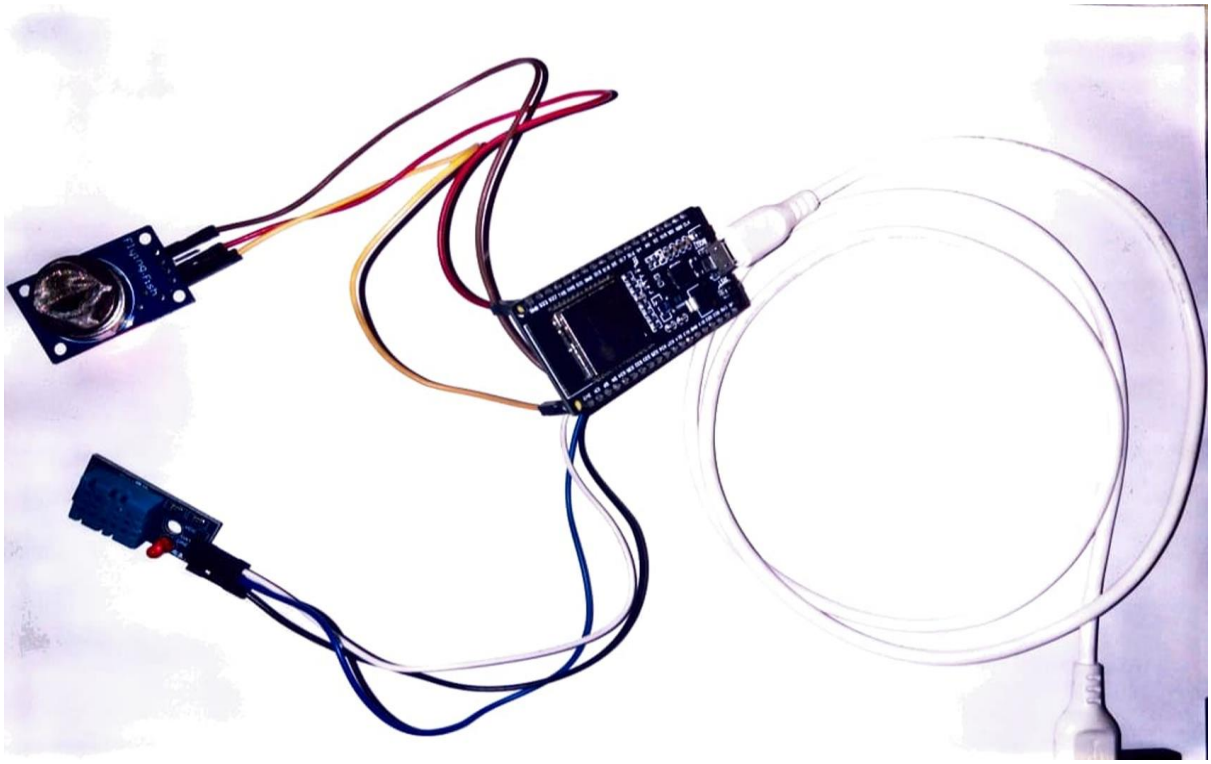
```
#include <WiFi.h>
#include <HTTPClient.h>
#include <DHT.h>
#define WIFI_SSID "Galaxy M016f42"
#define WIFI_PASSWORD "xskk0825"
// MQ-135 sensor settings
#define MQ135PIN 12 // Pin where the MQ-135 sensor is connected (ADC pin)
float ammoniaThreshold = 50;  // in ppm
float benzeneThreshold = 200; // in ppm
float alcoholThreshold = 100; // in ppm
// DHT sensor settings
#define DHTPIN 0 // Pin where the DHT sensor is connected
#define DHTTYPE DHT11 // DHT 22 (AM2302) or DHT11
DHT dht(DHTPIN, DHTTYPE);
// Server settings
const char *serverUrl = "https://console.thingzmate.com/api/v1/device-types/aqm2/devices/aqm3/uplink"; // Replace with your server endpoint
String AuthorizationToken = "Bearer 7dd1eb72e4c8fc439e4f1484d9f48739";
void setup() {
  Serial.begin(115200);
  dht.begin();
```

```
  delay(4000); // Delay to let serial settle
  // Connect to WiFi
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("Connected to WiFi");
}
void loop() {
  // Read DHT sensor values
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();
  // Check if any reads failed and exit early (to try again).
  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  // Read MQ-135 sensor value
  int sensorValue = analogRead(MQ135PIN);
  float voltage = sensorValue * (3.3 / 1023.0);
```

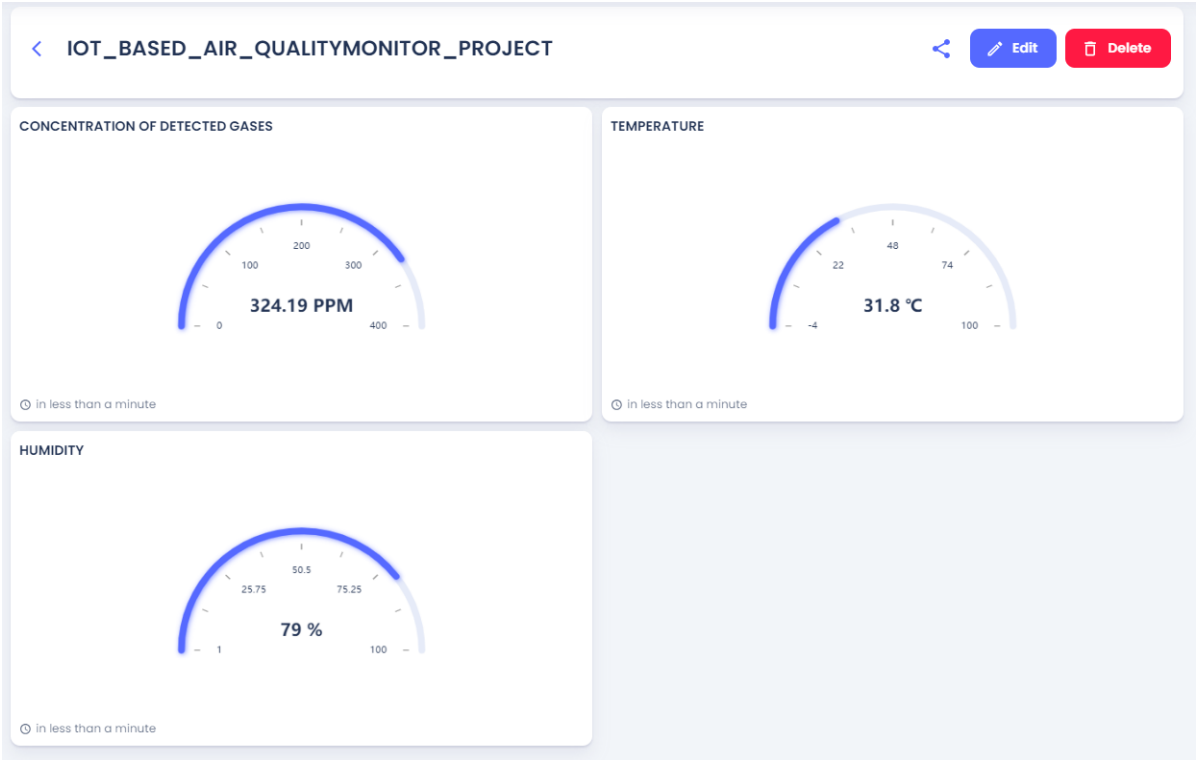
```
  float ppm = voltage * 100; // Example conversion to PPM
  // Determine the gas detection status
  String gasMessage = "No gas detected";
  if (ppm >= ammoniaThreshold && ppm < alcoholThreshold) {
    gasMessage = "Ammonia detected";
  }
  else if (ppm >= alcoholThreshold && ppm < benzeneThreshold) {
    gasMessage = "Alcohol detected";
  }
  else if (ppm >= benzeneThreshold) {
    gasMessage = "Benzene detected";
  }
  // Create JSON payload
   String payload = "{\"ppm\":" + String(ppm) + ",\"message\":\"" + gasMessage +
"\",\"temperature\":" + String(temperature) + ",\"humidity\":" + String(humidity) + "}";
  // Debug: Print payload
  Serial.print("Sending payload: ");
  Serial.println(payload);
  // Send data to server
  HTTPClient http;
  http.begin(serverUrl);
  http.addHeader("Content-Type", "application/json");
```

```
  http.addHeader("Authorization", AuthorizationToken); // Authorization token
  // Send POST request
  int httpResponseCode = http.POST(payload);
  if (httpResponseCode > 0) {
    String response = http.getString();
    Serial.println("HTTP Response code: " + String(httpResponseCode));
    Serial.println(response);
  } else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
  }
  http.end(); // Free resources
  delay(2000); // Wait for 2 seconds before sending the next request
}
```

OUTPUT RESULTS:



CLOUD OUTPUT



CONCLUSION:

The IoT-based Air Quality Monitoring System effectively demonstrates how modern technology can be leveraged to monitor environmental conditions in real-time. By integrating sensors like the DHT11 and MQ-135 with a cloud platform, the system provides accurate and accessible data on temperature, humidity, and harmful gas concentrations. This real-time monitoring capability empowers users to stay informed about air quality, enabling timely responses to potential health hazards. The project underscores the importance of continuous environmental monitoring in promoting public health and safety, while also showcasing the potential of IoT solutions in addressing global challenges related to air pollution.