Developing an IoT-based air quality monitoring system :

1. \*Define Requirements\*:

Determine what pollutants (e.g., particulate matter, CO2, VOCs).

Decide the location and scale of your monitoring system (indoor, outdoor, city-wide).

1. \*Choose Sensors\*:
   * Consider sensors’ accuracy, power consumption, and calibration requirements.

3. \*\*Microcontroller and Communication:\*\*

- Choose a microcontroller (e.g., Arduino, Raspberry Pi) to interface with sensors.

- Integrate communication modules (Wi-Fi, Bluetooth, LoRa) for data transmission.

4. \*Connectivity and IoT Platform:\*\*

- Establish a connection to the internet using the communication module.

- Choose an IoT platform (like AWS IoT, Azure IoT, or ThingSpeak) to store and manage data.

5. \*\*Data Visualization:\*\*

- Develop a web or mobile application to visualize the air quality data.

- Implement real-time charts, graphs, and alerts for users to monitor air quality.

6. \*Data Analysis and Machine Learning\* :

- Implement algorithms for data analysis, trend prediction, or anomaly detection.

- Use machine learning techniques to predict air quality based on historical data.

7. \*User Interface and Notifications\*:

- Design a user-friendly interface displaying air quality information.

- Implement notifications (email, SMS) for users when air quality crosses predefined thresholds.

8. \*Power Management\*:

- Implement power-saving mechanisms to prolong the device’s battery life (if applicable).

9. \*Testing and Deployment\*:

- Test the system in various environmental conditions to ensure accuracy and reliability.

- Deploy the monitoring devices in the target locations.

10. \*Maintenance and Updates\*:

- Plan for regular maintenance to calibrate sensors and ensure accurate readings.

- Provide over-the-air updates for software improvements and bug fixes

Program:

```javascript

// Importing the necessary modules if you’re using Node.js

Const axios = require(‘axios’);

// Replace ‘YOUR\_API\_KEY’ with your actual OpenWeatherMap API key

Const apiKey = ‘YOUR\_API\_KEY’;

Const city = ‘New York’; // Replace with the city you want to monitor

Const apiUrl = `https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}`;

Axios.get(apiUrl)

.then(response => {

Const airQuality = response.data.main.aqi;

Console.log(`Air Quality in ${city}: ${getAirQualityStatus(airQuality)}`);

})

.catch(error => {

Console.error(‘Error fetching air quality data:’, error);

});

Function getAirQualityStatus(aqi) {

// AQI scale: 1 (Good) to 5 (Hazardous)

Switch (true) {

Case aqi <= 50:

Return ‘Good’;

Case aqi <= 100:

Return ‘Moderate’;

Case aqi <= 150:

Return ‘Unhealthy for Sensitive Groups’;

Case aqi <= 200:

Return ‘Unhealthy’;

Case aqi <= 300:

Return ‘Very Unhealthy’;

Default:

Return ‘Hazardous’;

}

}

```