AIR QUALITY MONITORING

Sakthivel. P(210521205045)

OBJECTIVES:

We aim to install , maintain a network of air quality sensors in key location and continuously monitor ,record air pollutants such as CO,NO2,SO2 and ozone using sensors. The objective is to provide real time data on air quality parameters with high accuracy and reliability.Moreover,it enables the establishment of early warning systems for potential air quality issues allowing communities to take preventive actions and minimize exposure harmful pollutants

AQM DEVICE SETUP:

* Placement:Choose a location that is representative of the air quality in environment,away from direct sources of pollution or airflow disturbance
* Power supply:The device has a stable power source, either batteries
* Connectivity:The device connects to a network,it has access to Wi-Fi
* Calibration: follow the manufacturer's instructions to calibrate the device properly before use
* Data Recording:Set up the device to record and store data over time.
* Monitoring: Monitor the readings regularly and take necessary actions
* Develop a maintenance plan that includes regular sensor calibration,device maintenance,and troubleshooting procedures

PLATEFORM DEVELOPMENTS FOR AQM:

Advancement in sensor technology: Development of more offordable and portable sensors for widespread monitoring

Integration of satellite data: Incorporation of satellite data into monitoring systems to provide a broader and more comprehensive view of air quality

Enhanced public awareness: Growing emphasis on educating the public about the importance of air quality monitoring and it's impact on human health and the environment

Regulatory changes: Implementation of stricter regulations and policies based on the insights obtained from advanced monitoring technology

Define Requirements:

Clearly define the objectives and requirements of the platform, including the types of data to be collected, the user interface, and any specific features like data visualization or alerts.

Select Technology Stack:

Choose the technology stack for development, including programming languages, frameworks, and databases. The choice depends on your project's needs and your development team's expertise.

Data Collection and Integration:

Set up data connectors to collect data from monitoring devices. Ensure data is gathered efficiently and accurately. This may involve integrating with IoT devices through APIs or communication protocols.

Data Storage:

Design a database structure to store air quality data securely. Consider factors like data redundancy, scalability, and data retention policies.

Data Processing and Analysis:

Develop algorithms and processes for data processing and analysis. This could include real-time data validation, aggregation, and the calculation of air quality indices.

PROJECT DETAILS OF AQM:

Designing an air quality monitoring project in detail involves careful planning, execution, and ongoing management. Below is a comprehensive outline of the steps to create an air quality monitoring project, from inception to operation:

Project Planning and Definition:

Project Objectives: Clearly define the objectives, including the pollutants to monitor, the project's scope (e.g., a city or specific industrial area), and the target audience (e.g., public, regulatory bodies)

Budget and Resources:

Determine the project budget and allocate resources for equipment, technology, and personnel.

Project Timeline:

Establish a project timeline with milestones and deadlines.

Site Selection:

Identify and select monitoring sites based on project objectives, ensuring representativeness of the area's air quality.

Equipment Procurement:

Select and purchase air quality monitoring devices, sensors, communication modules, and any required hardware.

Sensor Calibration and Installation:

Calibrate sensors to ensure data accuracy and consistency.Install sensors at selected monitoring sites following best practices.

Data Communication and Management:

Develop a data management plan, including data storage, transmission, and backup.Configure communication modules to transmit data to a central server or cloud platform.

Platform Development (if applicable):

If developing a platform, follow the steps outlined in the previous response for platform development.

Quality Control and Validation:

Establish a data quality control process, which includes outlier detection and error correction.Conduct validation tests to ensure the accuracy of the monitoring devices.

Regulatory Compliance:

Ensure the project complies with local, national, or international regulations and standards for air quality monitoring.

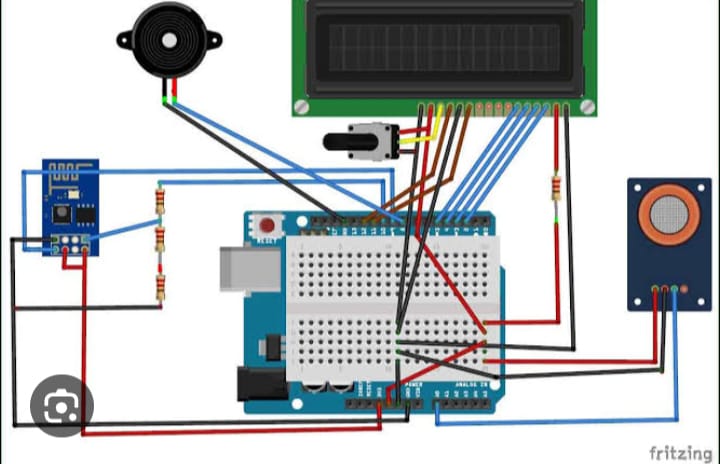
Data Visualization and Reporting:

Develop data visualization tools, including real-time dashboards and historical data reports.Create user-friendly interfaces for accessing air quality data.

Alerting System:

Implement an alerting system that notifies stakeholders and the public when air quality exceeds predefined thresholds.

Code for running in this circuit



html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Air Quality Monitoring Platform</title>

<style>

body {

font-family: Arial, sans-serif;

}

header {

background-color: #007ACC;

color: #FFF;

text-align: center;

padding: 20px;

}

main {

padding: 20px;

}

#air-quality-data {

border: 1px solid #ccc;

padding: 20px;

border-radius: 5px;

}

</style>

</head>

<body>

<header>

<h1>Real-time Air Quality Monitoring Platform</h1>

</header>

<main>

<div id="air-quality-data">

<!-- Air quality data will be displayed here -->

</div>

</main>

<script>

// Simulated real-time data

function generateRandomData() {

return {

pm25: (Math.random() \* 100).toFixed(2), // Simulated PM2.5 level in µg/m³

pm10: (Math.random() \* 150).toFixed(2), // Simulated PM10 level in µg/m³

co2: (Math.random() \* 1000).toFixed(2) // Simulated CO2 level in ppm

};

}

function updateData() {

const airQualityDataElement = document.getElementById('air-quality-data');

setInterval(() => {

const data = generateRandomData();

const timestamp = new Date().toLocaleTimeString();

// Update the data display

const newData = `<p>${timestamp}: PM2.5: ${data.pm25} µg/m³, PM10: ${data.pm10} µg/m³, CO2: ${data.co2} ppm</p>`;

airQualityDataElement.innerHTML = newData + airQualityDataElement.innerHTML;

}, 3000); // Update data every 3 seconds

}

window.addEventListener('load', () => {

updateData();

});

</script>

</body>

</html>

Output of the above program:

