AIR QUALITY MONITORING

Phase 4:

Introduction

pollution has become a pressing concern in the modern world, leading to numerous health and environmental issues. To address this problem, we propose designing a platform that can receive and display air quality data sent by IoT devices. This platform aims to provide real-time and comprehensive information on air pollution levels in different locations. By collecting data from various IoT devices strategically placed across different areas, we can create a comprehensive picture of air quality and empower individuals, communities, and policymakers to make informed decisions to mitigate pollution and improve the quality of life. The platform will offer an intuitive and user-friendly interface to visualize air quality data, enabling users to access and understand information easily. With this platform, we can take a significant step towards achieving cleaner and healthier air for all.

Objcetives:

1. Data Collection: The platform will act as a central receiver for air quality data transmitted by IoT devices placed at various locations. It will effectively collect and store this data in a secure and organized manner.

2. Real-time Updates: The platform will provide real-time updates on air quality, ensuring that users have access to the most recent information. This will enable them to react promptly and take necessary actions based on current air quality conditions.

3. Visualization: The platform will use cutting-edge data visualization techniques to present air quality information in an intuitive and user-friendly manner. Users will be able to grasp the data effortlessly through visually appealing charts, maps, and graphs, making it easier to interpret air quality patterns and trends.

4. Alerts and Notifications: The platform will allow users to set personalized alerts and notifications based on specific air quality parameters. Users will be instantly notified whenever air quality levels reach or exceed certain thresholds, enabling them to mitigate potential health risks and take appropriate action.

5. Data Analytics: The platform will incorporate data analytics capabilities to detect patterns, trends, and correlations within the air quality data. It will provide insights into long-term air quality trends, identify pollution hotspots, and help users understand the potential causes and impacts of air pollution.

6. Data Sharing: The platform will facilitate data sharing amongst users and relevant authorities, promoting collaboration and collective efforts in improving air quality. It will support open data initiatives, allowing researchers, policymakers, and environmental organizations to access the data for further analysis and decision-making.

Overall, the air quality platform aims to empower individuals and communities by equipping them with accurate and easily accessible air quality data. By providing a clear understanding of air pollution conditions, the platform enhances environmental awareness and fosters a proactive approach towards mitigating air pollution for a healthier and sustainable future.

1. HTML Structure:

Start by creating the basic structure of your web page using HTML. Include sections for the real-time data display, device status indicators, and any other necessary information. For example:

```html

<html>

<head>

<title>Air Quality Monitoring</title>

<!-- Link CSS stylesheets and necessary JavaScript files -->

<link rel="stylesheet" href="styles.css">

<script src="script.js"></script>

</head>

<body>

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

<div id="data-display">

<!-- Data will be dynamically updated here -->

</div>

<div id="device-status">

<!-- Device status indicators will be dynamically updated here -->

</div>

</body>

</html>

```

2. CSS Styling:

Enhance the appearance of your platform using CSS. Style the different sections, elements, and apply appropriate color schemes. Customize the design to match your preferences and ensure it is user-friendly.

3. JavaScript Functionality:

Use JavaScript to handle real-time updates and data retrieval from your IoT devices. Implement AJAX or WebSocket techniques to establish a connection and receive data from the devices. Update the data on the web page dynamically without refreshing the entire page.

```javascript

// WebSocket example

const socket = new WebSocket('wss://your-iot-device-endpoint');

socket.onmessage = function (event) {

const data = JSON.parse(event.data);

updateDataDisplay(data);

};

function updateDataDisplay(data) {

// Update the data-display div with the received air quality data

document.getElementById('data-display').innerHTML = `

<h2>Current Air Quality:</h2>

<ul>

<li>Temperature: ${data.temperature}°C</li>

<li>Humidity: ${data.humidity}%</li>

<li>PM2.5: ${data.pm25} µg/m³</li>

<li>CO2: ${data.co2} ppm</li>

<!-- Add more data fields as per your requirement -->

</ul>

`;

}

```

4. IoT Device Integration:

Ensure your IoT devices are configured to send air quality data to your web platform. The devices should regularly transmit the data to the web server using established protocols (e.g., HTTP, MQTT, WebSocket). Implement the necessary backend infrastructure to receive and process the incoming data.

5. Backend Processing and Storage:

Set up a backend server (e.g., Node.js, PHP) to receive data sent by the IoT devices. Process the received data as per your requirements (e.g., store it in a database, perform data analytics, or trigger alerts). Connect your backend server with the frontend to transmit the received data to the web platform.

These steps will help you create a platform that receives and displays real-time air quality data from IoT devices. Remember to adjust and customize the code as per your specific needs and the protocols used by your IoT devices.