Air Quality Management

Phase:4-Development

Building the project by developing the data-sharing platform.

Data-sharing platform:

After building the initial data display platform using Arduino or similar IoT devices, the next levels of development for a data-sharing platform typically involve enhancing the platform's functionality, scalability, and security

Data Storage and Database Integration:

Implement a database to store historical air quality data. This will allow you to analyze trends over time.

Real-Time Data Streaming:

Enable real-time data streaming capabilities to provide users with immediate access to the latest air quality information.

User Authentication and Authorization:

Implement user authentication and authorization to ensure that only authorized users can access the data-sharing platform.

Data Visualization:

Enhance the user interface with interactive data visualizations such as charts, graphs, and maps to provide a richer user experience.

Geospatial Integration:

Integrate geospatial features to allow users to view air quality data on a map, which can be especially useful for monitoring air quality in different locations.

Alerting and Notifications:

Implement alerting and notification systems that inform users when air quality reaches certain thresholds or when specific gases are detected at harmful levels.

Data Analysis and Reporting:

Provide data analysis tools that allow users to generate reports and insights from the collected data.

Scalability and Performance:

Ensure that the platform can handle a growing volume of data and users. Consider using cloud-based solutions for scalability.

Security and Privacy:

Implement strong security measures to protect data integrity and user privacy. This includes encryption, access controls, and compliance with data protection regulations.

Collaboration and Sharing Features:

Add features that allow users to collaborate and share data with others, fostering a community of users interested in air quality.

Continuous Monitoring and Maintenance:

Regularly monitor and maintain the platform to ensure data accuracy and system reliability.

Documentation and Support:

Create user and developer documentation to assist users and other developers in using and integrating your platform.

Compliance with Environmental Standards:

Ensure that your platform adheres to environmental standards and regulations related to air quality monitoring.

Creating a platform that displays real-time air quality data. Design the platform to receive and display air quality data sent by the IOT devices.

Code that has been used for creating a web page are:

HTML (index.html): This file contains the structure and content of your web page.

JavaScript (app.js): This file contains the JavaScript code that simulates air quality data and displays it on your web page.

CSS (styles.css): This file contains the styling rules to control the appearance of your web page.

HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="styles.css">
    <title>Air Quality Data Platform</title>
</head>
<body>
    <div class="container">
        <h1>Air Quality Data Platform</h1>
        <button id="simulateButton">Simulate IoT Data</button>
        <div id="dataDisplay"></div>
    </div>
    <script src="app.js"></script>
</body>
</html>
```

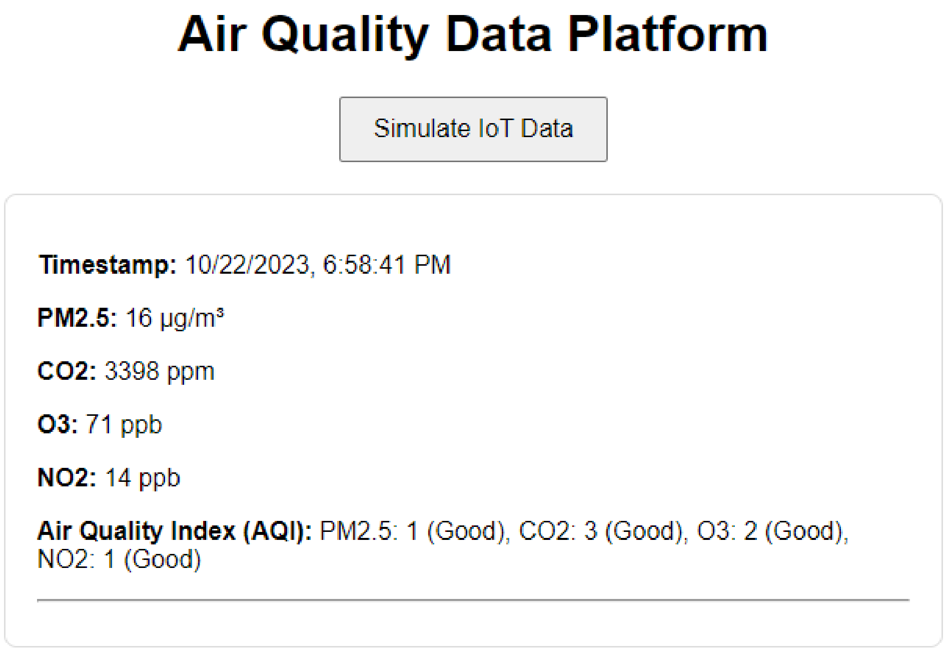
css:

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}

.container {
    max-width: 600px;
    margin: 50px auto;
    text-align: center;
}

button {
    padding: 10px 20px;
    font-size: 16px;
    cursor: pointer;
}

#dataDisplay {
    margin-top: 20px;
    padding: 20px;
    border: 1px solid #ddd;
    border-radius: 8px;
    text-align: left;
}
```

Java script:

```
document.addEventListener('DOMContentLoaded', () => {
    const simulateButton = document.getElementById('simulateButton');
    const dataDisplay = document.getElementById('dataDisplay');

    simulateButton.addEventListener('click', () => {
        const simulatedData = generateSimulatedData();
        displayData(simulatedData);
    });

    function generateSimulatedData() {
        const timestamp = new Date().toLocaleString();
        const airQualityData = {
            PM25: Math.floor(Math.random() * 50),
            CO2: Math.floor(Math.random() * 5000),
            O3: Math.floor(Math.random() * 100),
            NO2: Math.floor(Math.random() * 50)
        };

        return { timestamp, airQualityData };
    }

    function displayData(data) {
        const newDataElement = document.createElement('div');
        newDataElement.innerHTML = `<p><strong>Timestamp:</strong> ${data.timestamp}</p>
                                    <p><strong>PM2.5:</strong> ${data.airQualityData.PM25} µg/m³</p>
                                    <p><strong>CO2:</strong> ${data.airQualityData.CO2} ppm</p>
                                    <p><strong>O3:</strong> ${data.airQualityData.O3} ppb</p>
                                    <p><strong>NO2:</strong> ${data.airQualityData.NO2} ppb</p>
                                    <p><strong>Air Quality Index (AQI):</strong>
${getSpecificAirQuality(data.airQualityData)}</p>
                                    <hr>`;

        dataDisplay.prepend(newDataElement);
    }

    function getSpecificAirQuality(airQualityData) {
        const pm25AQI = calculateAQI(airQualityData.PM25, [0, 12, 35, 55, 150]);
        const co2AQI = calculateAQI(airQualityData.CO2, [0, 400, 1000, 2000, 5000]);
        const o3AQI = calculateAQI(airQualityData.O3, [0, 54, 70, 85, 100]);
        const no2AQI = calculateAQI(airQualityData.NO2, [0, 25, 50, 100, 200]);

        const pm25Level = getAirQualityLevel(pm25AQI);
        const co2Level = getAirQualityLevel(co2AQI);
        const o3Level = getAirQualityLevel(o3AQI);
        const no2Level = getAirQualityLevel(no2AQI);

        return `PM2.5: ${pm25AQI} (${pm25Level}), CO2: ${co2AQI} (${co2Level}), O3:
${o3AQI} (${o3Level}), NO2: ${no2AQI} (${no2Level})`;
    }

    function calculateAQI(concentration, levels) {
        for (let i = 0; i < levels.length - 1; i++) {
            const [low, high, lowIndex, highIndex] = [levels[i], levels[i + 1], i, i + 1];
            if (concentration >= low && concentration <= high) {
                return Math.round(((highIndex - lowIndex) / (high - low)) * (concentration - low) +
lowIndex);
            }
        }
        return 0; // Default to the lowest level
    }

    function getAirQualityLevel(aqi) {
        if (aqi <= 50) {
            return "Good";
        } else if (aqi <= 100) {
            return "Moderate";
        } else if (aqi <= 150) {
            return "Unhealthy for Sensitive Groups";
        } else if (aqi <= 200) {
            return "Unhealthy";
        } else if (aqi <= 300) {
            return "Very Unhealthy";
        } else {
            return "Hazardous";
        }
    }
});
```

The above code gives the website as:



Air Quality Data Platform

Timestamp: 10/22/2023, 6:58:41 PM

PM2.5: 16 µg/m^3

CO2: 3398 ppm

O3: 71 ppb

NO2: 14 ppb

Air Quality Index (AQI): PM2.5: 1 (Good), CO2: 3 (Good), O3: 2 (Good), NO2: 1 (Good)

The webhost that has been responsible for launching our webpage which has air quality is as follows:

https://shruthisrini.neocities.org/AQM/

we have used neocities as the platform to post my webpage as of to be viewed by everyone