 **KGiSL Institute of Technology**

(Affiliated to ANNA University, Chennai and Approved by AICTE, New Delhi)

365, KGiSL Campus, Thudiyalur Road, Saravanampatti Coimbatore – 641035

**Department of Artificial**

**Intelligence and Data Science**



**NAAN MUDHALVAN -INTERNET OF THINGS**

PROJECT TITLE : ENVIRONMENTAL MONITORING IN PARKS

REGISTER NUMBER : 711721243026

NAME : GANESAN.M

GMAIL ID : mganesanmganesan611@gmail.com

MENTOR : MOHANKUMAR.M

EVALUATOR : AKILANDESWARI.M

**ENVIRONMENTAL MONITORING IN PARKS**

**PROBLEM STATEMENT:**

In this phase of the project, you will continue to develop an environmental monitoring platform using web development technologies, including HTML, CSS, and JavaScript. The goal is to create a user-friendly platform that can receive and display real-time temperature and humidity data from IoT (Internet of Things) devices. Here's a more detailed breakdown of the tasks:

**1. Design the User Interface (UI):**

- Create a visually appealing and intuitive user interface for the environmental monitoring platform. Consider using HTML and CSS to structure and style your web application.

- Design a dashboard that can accommodate real-time data visualization, with sections for temperature and humidity data.

**2. Set Up the Front-End:**

- Develop the front-end of the platform using HTML, CSS, and JavaScript. You can use libraries and frameworks like React or Vue.js to simplify development.

- Create components or sections for displaying real-time temperature and humidity data.

**3. Real-Time Data Integration:**

- Establish a connection to your IoT devices that collect temperature and humidity data. You may need to use technologies like WebSockets, MQTT, or REST APIs to fetch real-time data from the IoT sensors.

- Use JavaScript to update the UI in real-time as new data becomes available. This can be achieved through asynchronous data retrieval and DOM manipulation.

**4. Data Visualization:**

- Implement data visualization techniques to display temperature and humidity data. You can use libraries like Chart.js or D3.js to create interactive charts and graphs.

- Ensure that the data is presented in a user-friendly and easily understandable format, such as line charts or gauges.

**5. User Interaction:**

- Allow users to interact with the platform. Provide options for customizing the view, such as selecting a specific time range, setting alerts, or switching between different IoT sensors.

**6.Responsive Design:**

- Ensure that your platform is responsive and compatible with various devices (e.g., desktops, tablets, and smartphones). Use CSS to create a layout that adapts to different screen sizes.

**7. Error Handling and Security:**

- Implement error handling to gracefully manage issues that may arise with data retrieval or IoT device connectivity.

- Pay attention to security considerations, especially if sensitive data is involved. Use secure protocols and access controls to protect the system.

**8. Testing and Debugging:**

- Thoroughly test your platform to ensure it functions as expected. Test it with real IoT devices to verify data accuracy.

- Debug and address any issues or bugs that may arise during testing.

**9. Documentation:**

- Create documentation for your platform, explaining how it works, how to use it, and any integration instructions for IoT devices.

**10. Deployment:**

- Choose a web hosting service or server to deploy your environmental monitoring platform. Ensure that it's accessible to authorized users.

- Set up any necessary infrastructure for scaling and reliability, especially if you anticipate a large number of users or IoT devices.

Throughout this project, it's important to maintain a focus on user experience and data accuracy. Regularly update and improve the platform based on user feedback and evolving requirements, as environmental monitoring often requires adjustments over time to remain effective and relevant.

**Sample Code:**

**HTML (INDEX.HTML):**

<!DOCTYPE html>

<html>

<head>

<title>Environmental Monitoring Platform</title>

<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<div class="container">

<h1>Real-time Environmental Data</h1>

<div id="temperature">Temperature: <span id="temp-value">--</span>°C</div>

<div id="humidity">Humidity: <span id="humid-value">--</span>%</div>

</div>

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

</body>

</html>

**CSS (STYLE.CSS):**

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

text-align: center;

}

.container {

background-color: #fff;

border-radius: 10px;

padding: 20px;

margin: 20px auto;

width: 300px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);

}

h1 {

margin: 0;

}

#temperature, #humidity {

margin-top: 10px;

}

**JAVA SCRIPT (SCRIPT.JS):**

function updateData() {

// Replace the following lines with code to fetch real-time data from IoT devices

const temperatureData = getRandomValue(10, 30).toFixed(2);

const humidityData = getRandomValue(20, 70).toFixed(2);

document.getElementById("temp-value").textContent = temperatureData + "°C";

document.getElementById("humid-value").textContent = humidityData + "%";

}

function getRandomValue(min, max) {

return Math.random() \* (max - min) + min;

}

// Update data every 5 seconds (adjust as needed)

setInterval(updateData, 5000);

**CONCLUSION:**

In conclusion, using web development technologies to create a real-time environmental monitoring platform provides a flexible and accessible solution for tracking and analyzing environmental data. While the code examples provided here are basic, they serve as a foundation for more sophisticated and practical implementations, integrating with IoT devices and ensuring data accuracy and security. Such platforms are invaluable in various fields, including agriculture, climate monitoring, and building management.