**FLOOD MONITORING AND EARLY WARING SYSTEM**

**DEVELOPMENT PART -2[ PHASE\_\_4]**



**In this part you will continue building your project. Continue building the project by developing the early warning platform. Use web development technologies (e.g., HTML, CSS, JavaScript) to create a platform that displays real-time water level data and flood warnings.**

**Design the platform to receive and display water level data from IoT sensors and issue flood warnings when necessary.**

* **Creating an environmental monitoring platform that displays real-time temperature and humidity data from IoT devices requires a combination of front-end and back-end development.**
* **In this response, I'll guide you through the process using HTML, CSS, and JavaScript for the front-end and assume that you already have an IoT device providing the data.**



**HTML Structure:**

* **Start by creating the basic structure of your web page. Here's a simple example:**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Early Warning Platform</title>**

**<link rel="stylesheet" type="text/css" href="My\_prog1.css">**

**</head>**

**<body>**

**<div class="container">**

**<h1>Early Warning Platform</h1>**

**<div class="data-container">**

**<div class="data-box" id="water-level">**

**<h2>Water Level</h2>**

**<p id="water-level-value">0.00 m</p>**

**</div>**

**<div class="data-box" id="flood-warning">**

**<h2>Flood Warning</h2>**

**<p id="flood-warning-value">No Alert</p>**

**</div>**

**<div class="data-box" id="temperature">**

**<h2>Temperature</h2>**

**<p id="temperature-value">0.00°C</p>**

**</div>**

**<div class="data-box" id="humidity">**

**<h2>Humidity</h2>**

**<p id="humidity-value">0%</p>**

**</div>**

**</div>**

**</div>**

**<script src="My\_prog2.js"></script>**

**</body>**

**</html>**

**CSS Styling (My\_prog1.css):**

* **Style your page to make it visually appealing and responsive:**

**body**

**{**

**font-family: Arial, sans-serif;**

**background-color: #f4f4f4;**

**}**

**.container**

**{**

**max-width: 800px;**

**margin: 0 auto;**

**background-color: #fff;**

**padding: 20px;**

**border-radius: 5px;**

**box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);**

**}**

**.data-containe**

**{**

**display: flex;**

**flex-wrap: wrap;**

**justify-content: space-between;**

**}**

**.data-box**

**{**

**background-color: #f0f0f0;**

**padding: 15px;**

**margin: 10px;**

**border-radius: 5px;**

**text-align: center;**

**box-shadow: 0 0 5px rgba(0, 0, 0, 0.2);**

**}**

**h1**

**{**

**text-align: center;**

**}**

**h2**

**{**

**font-size: 20px;**

**margin-bottom: 10px;**

**}**

**p**

**{**

**font-size: 18px;**

**font-weight: bold;**

**}**

**JavaScript (My\_prog2.js):**

* **Use JavaScript to fetch real-time data from your IoT device and update the webpage.**

**function updateData()**

**{**

**// Generate random data for water level, flood warning, temperature, and humidity**

**const waterLevelValue = (Math.random() \* 10).toFixed(2);**

**const floodWarningValue = Math.random() < 0.2 ? "Alert" : "No Alert";**

**const temperatureValue = (Math.random() \* 40).toFixed(2);**

**const humidityValue = (Math.random() \* 100).toFixed(2);**

**document.getElementById("water-level-value").textContent = ` ${waterLevelValue} m`;**

**document.getElementById("flood-warning-value").textContent = floodWarningValue;**

**document.getElementById("temperature-value").textContent = ` ${temperatureValue}°C`;**

**document.getElementById("humidity-value").textContent = ` ${humidityValue}%`;**

**}**

**// Update data every 5 seconds (5000 milliseconds)**

**setInterval(updateData, 5000);**

**// Initial data update**

**updateData();**

**OUTPUT:**



**MODEL IMAGE FOR FLOOD ALERT:**



**Serving the Web Application:**

* **You can use a simple HTTP server or a framework like Express.js if you want to add a backend for more advanced features, but this example demonstrates a static frontend with random data generation.**
* **To run this project, save the HTML as "index.html," the CSS as "styles.css," and the JavaScript as "script.js" in the same directory.**
* **Open "index.html" in a web browser, and you should see a real-time early warning platform displaying random data for water level, flood warnings, temperature, and humidity. The data will update every 5 seconds, simulating real-time information.**
* **For a production environment, you would replace the random data generation with a connection to a data source providing actual real-time information.**
* **Additionally, you could enhance the platform with more advanced features like data visualization, alerts, and notifications based on the data.**

**FLOW CHART :**

