Flood Monitoring And Early Warning

Phase4 :Development Part2

Index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Water Level Monitoring</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <header>
        <h1>Water Level Monitoring Platform</h1>
    </header>
    <div class="container">
        <div class="sensor-list">
            <h2>Sensor Locations</h2>
            <ul id="sensor-list">
        <!-- Sensor locations will be added dynamically through JavaScript --
    >
      </ul>
   </div>
    <div class="data-display">
      <h2>Real-Time Water Levels</h2>
      <div id="water-level-chart">
        <!-- Water level chart or map goes here -->
      </div>
      <div id="warning">
        <h3>Flood Warning</h3>
        <p id="warning-message">No flood warning at the moment.</p>
        <button id="test-warning-button">Test Flood Warning</button>
      </div>
    </div>
  </div>
  <script src="script.js"></script>
</body>
```

```
</html>
```

Style.css:

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f4f4f4;
}

header {
    background-color: #333;
    color: #fff;
    text-align: center;
    padding: 1em 0;
}.container {
    display: flex;
    justify-content: space-around;
    padding: 20px;
```

```
}
.sensor-list {
    width: 30%;
    background-color: #fff;
    padding: 20px;
    border: 1px solid #ccc;
}
.sensor-list ul {
    list-style: none;
    padding: 0;
}
.sensor-list li {
    margin: 10px 0;
    cursor: pointer;
}
.data-display {
    width: 70%;
    background-color: #fff;
    padding: 20px;

    border: 1px solid #ccc;

}#water-level-chart {

    /* Add styles for the water level chart or map here */}

h2 {

    font-size: 20px;

}#warning {

    border: 2px solid red;

    background-color: #ffcccc;

    padding: 10px;

    border-radius: 5px;

    margin-top: 20px;

}

#test-warning-button {

    background-color: red;

    color: white;

    border: none;

    padding: 5px 10px;
    border-radius: 3px;

    cursor: pointer;

}

#test-warning-button:hover

 {

 background-color: darkred;

}
```

Script.js:

```
document.addEventListener("DOMContentLoaded", function () {

    let useRealData = true; // Set to true to fetch real data, or false to simulate
         data

    // Function to fetch real sensor data

    function fetchSensorData() {

        fetch('https://your-real-data-source.com/sensors')

            .then(response => response.json())

            .then(data => {

                updateSensorDisplay(data); // Update the sensor display with real
         data
```

```
            })

            .catch(error => {

                console.error('Error fetching real sensor data: ' + error);

                // If there's an error, fall back to simulating data

                useRealData = false;

                simulateSensorData();

            });

    }

// Function to simulate sensor data

    function simulateSensorData() {

        const sensors = [

            { name: "Sensor 1", location: "Location A", waterLevel: Math.random()
          * 1 },

            { name: "Sensor 2", location: "Location B", waterLevel: Math.random()
          * 1 },

        ];

        updateSensorDisplay(sensors);

    }
```

```
// Function to update the sensor display with data
function updateSensorDisplay(sensors) {
    const sensorList = document.getElementById("sensor-list");
    // Clear the existing sensor list
    sensorList.innerHTML = ";
    sensors.forEach(sensor => {
        const li = document.createElement("li");
        li.textContent = sensor.name;
        li.addEventListener("click", () => displayWaterLevel(sensor));
        sensorList.appendChild(li);
    });
}
// Display water level data for a selected sensor
function displayWaterLevel(sensor) {
    const waterLevelChart = document.getElementById("water-level-chart");
    waterLevelChart.innerHTML = `<h3>${sensor.name} -
     ${sensor.location}</h3><p>Water Level: ${sensor.waterLevel}</p>`;
}
```

```
// Add functionality to test flood warning button
   const testWarningButton = document.getElementById("test-warning-
       button");
   const warningMessage = document.getElementById("warning-message")
       testWarningButton.addEventListener("click", () => {
      // Simulate a flood warning
      warningMessage.innerText = "Flood warning! Evacuate immediately!";
      // You would trigger the buzzer or alarm here (hardware implementation is
       not shown)
   });
  // Determine whether to use real data or simulate data
   if (useRealData) {
      fetchSensorData(); // Fetch real data
   } else {
      simulateSensorData(); // Simulate data
   }
});
```