FloodMonitoringAndEarlyWarningSystem Using IOT With Sensors

ProjectDescription:

Continuebuildingtheprojectbydevelopingtheenvironmentalmonitoringplatform.

Use web development technologies (e.g., HTML, CSS, JavaScript) to create a platform thatdisplays real-time environmentaldata. Designtheplatformtoreceiveanddisplayreal-timetemperatureandhumiditydatafrom IoT devices.

Gps Sensor:

Ultrasonic sensorsare deployed on hundreds of coastal tide gauge platforms that provide tsunami and tropical storm surge warning data.

Benefits:

Timely detection of possible flood risks and floods. Highly reliable and available real-time data. Tailored solution that can be integrated with external developments at any level (device, connectivity, cloud or user application).

Application:

In this system we make use of a raspberry pi with water sensors, rain sensors to predict flood and alert respective authorities and sound instant alarm in nearby villages to instantly transmit information about possible floods using IOT. The water sensors are used to measure water level of 3 different location

1. Definethe Project Structure:

Beforeyoustartcoding, planyourproject'sstructure. CreatefoldersforHTML, CSS, JavaScript, and any other assets (images, icons, etc.). This helps keep your code organized.

1. DesigntheUser Interface:

Design a user-friendly interface to display the real-time data. Consider using a responsive design to make it accessible on different devices. You can use HTML for the structure, CSS for styling, and JavaScript for interactivity.

1. CSS Styling:

StyleyourplatformusingCSStomakeitvisuallyappealinganduser-friendly.Youcan customize fonts, colors, and layouts according to your preferences.

1. JavaScriptforReal-TimeData:

Use JavaScript to fetch real-time data from your IoT devices and update the HTML content accordingly. You can use technologies like WebSockets or AJAX to achieve real-time updates. Here's a basic example using JavaScript and AJAX:

1. BackendandIoTIntegration:

On the server-side, you'll need to create an endpoint to fetch data from your IoT devices. You can use a web framework like Node.js with Express to handle these requests. Make sure your IoT devices are set up to provide real-time data through an API.

1. TestingandDeployment:

Testyour platform thoroughly,bothlocallyandondifferentdevices,toensure it works as expected. After testing, deploy your platform to a web server so it's accessible to users.

1. Security:

Ensure that you implement proper security measures to protect your platform from unauthorized access and data breaches, especially if it's dealing with real-time data from IoT devices.

Language:HTML,CSS,JavaScript,PHPandMySql.

Hardware:Float switch for water level detector, inverter, rain gauge, GSM module, and microcontrollerdevelopment board

DEVELOPINGTHEREAL-TIMETRANSITINFORMATIONPLATFORM BY USING WEBTECHNOLOGY

HTMLSTRUCTURE:

<!DOCTYPEhtml>

<html>

<head>

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

</head>

<body>

<header>

<h1>EnvironmentalMonitoringPlatform</h1>

</header>

<main>

<sectionid="sensor-data">

<h2>Real-timeData</h2>

<divid="temperature">Temperature:<span></span>°C</div>

<divid="humidity">Humidity:<span></span>%</div>

</section>

</main>

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

</body>

</html>

JavaScriptforRealtime data:

// script.js

//Functiontofetchandupdatetemperatureandhumiditydata function updateData() {

//MakeanAJAXrequesttoyourIoTdeviceendpoint fetch('/api/environmental-data')

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

.then(data=>{

//UpdatetheHTMLwithreal-time data

document.querySelector('#temperaturespan').textContent=data.temperature+'°C'; document.querySelector('#humidity span').textContent = data.humidity + '%';

})

.catch(error=>{

console.error('Errorfetchingdata:',error);

});

}

//Updatethedataevery10seconds(adjustasneeded) setInterval(updateData, 10000);

updateData();//Callinitiallyto loaddata

DESIGNTHEPLATFORMTORECEIVEANDDISPLAYREAL-TIMEDATAFROMIOTSENSOR

