Application

Description automatically generated with low confidence  
**PREPARED BY**

S.GNANANIVIN

S.PUGAZHENTHI

K.GOKULKUMARAN



**SMART WATER MANAGEMENT**

**PLANNING:**

Creating a platform to display real-time water level data and issue flood warnings involves a combination of frontend web development (HTML, CSS, JavaScript) for the user interface and backend technologies to receive data from IoT sensors and issue warnings. In this example, I'll focus on the front-end part. For the backend, you would typically need a server, database, and IoT communication system (e.g., MQTT) as discussed earlier. Here's a simple frontend design for your use case,

**Step 1: HTML Structure:**

Create an HTML file to structure your platform,

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Water Level Monitoring</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>

<h1>Water Level Monitoring Platform</h1>

</header>

<section class="data-section">

<h2>Real-time Water Level</h2>

<div class="water-level">

<p id="water-level">Loading...</p>

</div>

</section>

<section class="warning-section">

<h2>Flood Warning</h2>

<div class="flood-warning">

<p id="flood-warning">No Warnings</p>

</div>

</section>

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

</body>

</html>

**Step 2: CSS Styling:**

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

}

header {

background-color: #3498db;

color: white;

text-align: center;

padding: 20px;

}

.data-section, .warning-section {

text-align: center;

margin: 20px;

}

h2 {

margin: 10px 0;

}

.water-level, .flood-warning {

background-color: #f1f1f1;

border: 1px solid #ccc;

padding: 10px;

font-size: 24px;

margin: 10px;

}

**Step 3: JavaScript for Real-time Data:**

Create a JavaScript file (**script.js**) to update the water level and flood warning data in real-time.

function updateWaterLevel() {

// Simulate water level data (replace with actual data)

const waterLevel = Math.random() \* 100;

document.getElementById("water-level").textContent = `Water Level: ${waterLevel.toFixed(2)} inches`;

}

function updateFloodWarning() {

// Simulate flood warning (replace with actual data)

const hasFloodWarning = Math.random() < 0.1; // 10% chance of warning

const warningText = hasFloodWarning ? "Flood Warning: Take precautions!" : "No Warnings";

document.getElementById("flood-warning").textContent = warningText;

}

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

setInterval(updateWaterLevel, 5000);

setInterval(updateFloodWarning, 5000);

// Initial update

updateWaterLevel();

updateFloodWarning();

This JavaScript code simulates updates for water level and flood warnings and sets an interval for periodic updates. You should replace the simulated data with real data from your backend system.

Remember that this is just the front-end part. The real-time data would be fetched from your backend system connected to the IoT sensors and flood monitoring equipment. You'll need to set up the backend to provide this data to the frontend through an API or WebSocket connection.