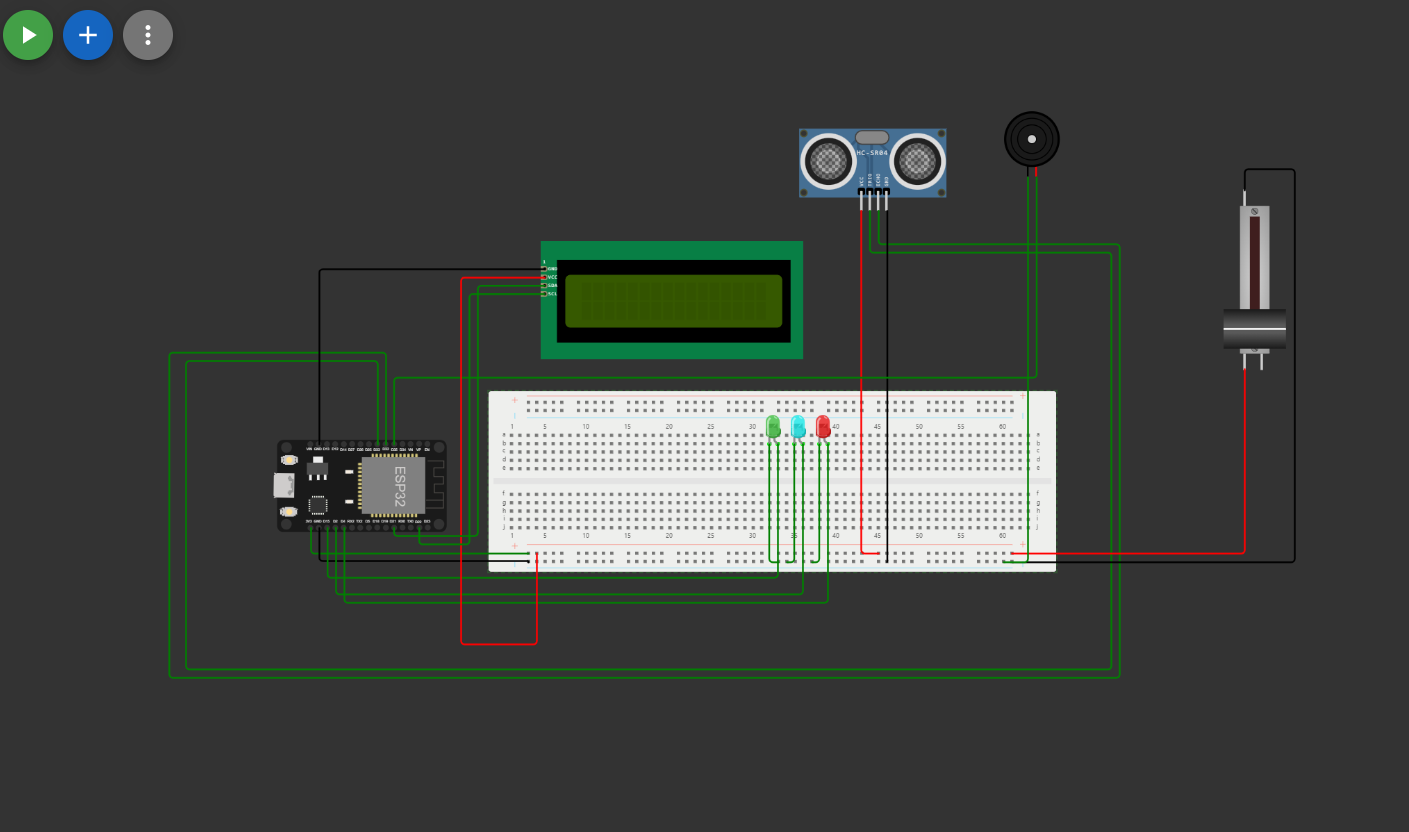
**Flood Monitoring and Early Warnings**

**Phase-4**

**Introduction:**

Creating a platform to display real-time water level data and issue flood warnings involves several components. Here's a simplified example of how you can build such a platform using web development technology.



**HTML Structure:**

Create the basic HTML structure for my platform. You'll need elements to display the water level data, flood warnings, and possibly a map to visualize the affected areas.

**Example of my written code:**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Real-time Water Level and Flood Warning Platform</title>**

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

**</head>**

**<body>**

**<div id="water-level">**

**<h1>Water Level: <span id="water-level-value">0 cm</span></h1>**

**</div>**

**<div id="flood-warning">**

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

**<p id="warning-message">No flood warnings at the moment.</p>**

**</div>**

**<div id="map">**

**<!-- Embed a map if necessary -->**

**</div>**

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

**</body>**

**</html>**

**CSS Styling:**

Apply CSS styles to make your platform visually appealing and user-friendly.

**Example of CSS code of my written HTML code:**

body {

font-family: Arial, sans-serif;

}

#water-level {

text-align: center;

margin: 20px;

}

#flood-warning {

text-align: center;

margin: 20px;

}

Java script for real-time data:

By using JavaScript to fetch and display real-time water level data and issue flood warnings.

**Example of JAVA Script code of my written HTML code:**

// Simulated real-time data (replace with actual IoT sensor data)

let waterLevel = 0;

function updateWaterLevel() {

// Simulated data update

waterLevel += Math.random() \* 10;

document.getElementById('water-level-value').textContent = waterLevel + ' cm';

// Check for flood conditions (set your threshold)

if (waterLevel > 50) {

document.getElementById('warning-message').textContent = 'Flood warning! Evacuate the area.';

document.getElementById('flood-warning').classList.add('warning-active');

} else {

document.getElementById('warning-message').textContent = 'No flood warnings at the moment.';

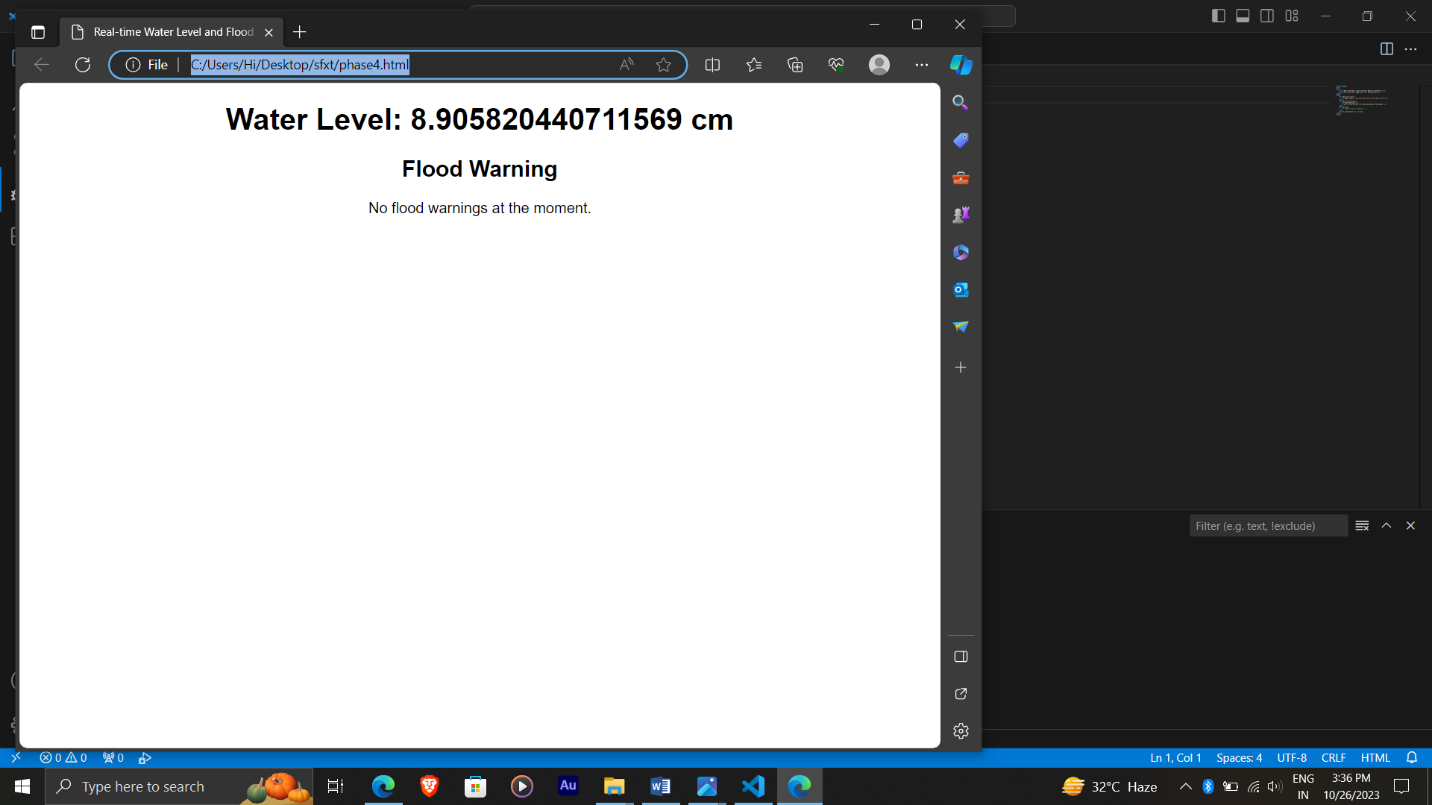
document.getElementById('flood-warning').classList.remove('warning-active');

}

}

setInterval(updateWaterLevel, 5000); // Update data every 5 seconds

**OUTPUT:**



WEBSITE



Finally:

Finally I have attached my HTML,CSS,JS files to my github account with this Explanation document.