Creating a platform for real-time water level data and flood warnings using web development technologies involves several components and coding. Here's a simplified example using HTML, CSS, and JavaScript:

HTML (index.html):
```html
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
    <div class="header">
        <h1>Water Level Monitoring</h1>
    </div>
    <div class="data-container">
        <h2>Real-Time Water Level Data</h2>
        <div id="water-level-data">
            <!-- Real-time data will be displayed here -->
        </div>
    </div>
    <div class="alerts">
        <h2>Flood Alerts</h2>
        <div id="flood-alerts">
            <!-- Flood alerts will be displayed here -->
        </div>
    </div>
    <script src="script.js"></script>
</body>
</html>
```

CSS (style.css):
```css
body {
    font-family: Arial, sans-serif;
}

.header {
    background-color: #3498db;
    color: #fff;
    text-align: center;
    padding: 20px;
}

.data-container, .alerts {
    margin: 20px;
    padding: 10px;
    border: 1px solid #ddd;
}

.data-container h2, .alerts h2 {
    color: #3498db;
}

```

JavaScript (script.js):

```javascript
// Simulated real-time data from IoT sensors
const sensorData = {
    sensor1: 4.2,
    sensor2: 3.8,
    sensor3: 5.1
};

// Simulated flood warning logic
function checkFloodWarnings(data) {
    const warnings = [];
    for (const sensor in data) {
        if (data[sensor] > 4.0) {
            warnings.push(`Flood alert for ${sensor}: Water level is ${data[sensor]}
meters.`);
        }
    }
    return warnings;
}

// Function to update real-time data and flood alerts
function updateDataAndAlerts() {
    const waterLevelData = document.getElementById('water-level-data');
    const floodAlerts = document.getElementById('flood-alerts');

    // Update real-time water level data
    let dataHTML = '<ul>';
    for (const sensor in sensorData) {
        dataHTML += `<li>${sensor}: ${sensorData[sensor]} meters</li>`;
    }
    dataHTML += '</ul>';
    waterLevelData.innerHTML = dataHTML;

    // Check for flood warnings
    const warnings = checkFloodWarnings(sensorData);
    let alertsHTML = '<ul>';
    for (const warning of warnings) {
        alertsHTML += `<li>${warning}</li>`;
    }
    alertsHTML += '</ul>';
    floodAlerts.innerHTML = alertsHTML;
}

// Update data and alerts every 5 seconds (simulated real-time)
setInterval(updateDataAndAlerts, 5000);

// Initial update
updateDataAndAlerts();
```

This is a simple example that simulates real-time water level data and flood warnings. In a real-world scenario, you would replace the simulated data and logic with actual IoT sensor data and flood detection algorithms. Additionally, you may want to consider a server-side component to handle data and issue alerts more effectively.