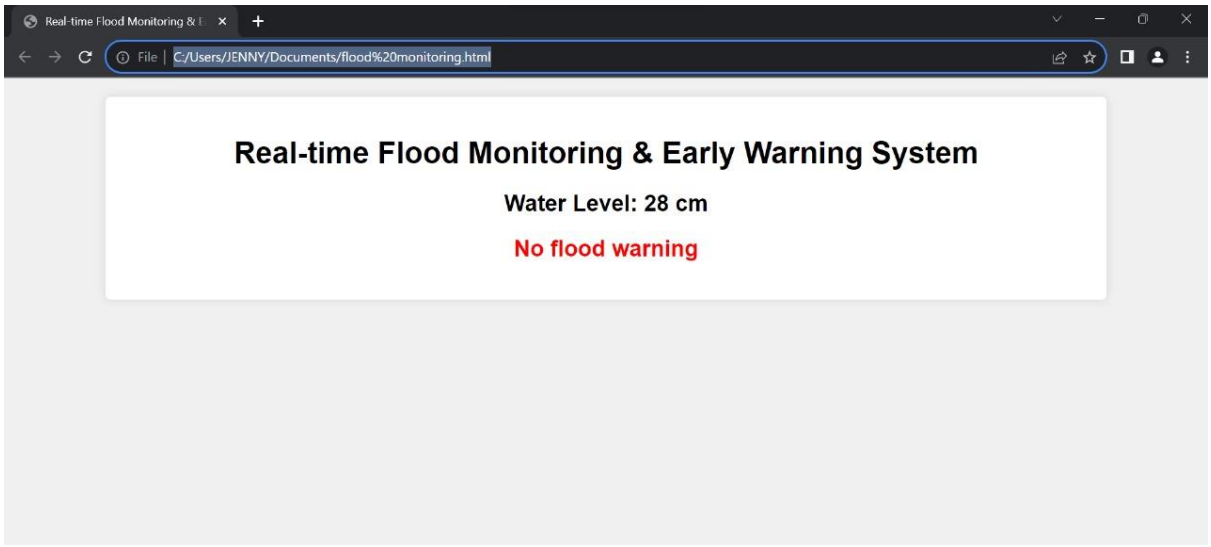
FLOOD MONITORING SYSTEM AND EARLY WARNING

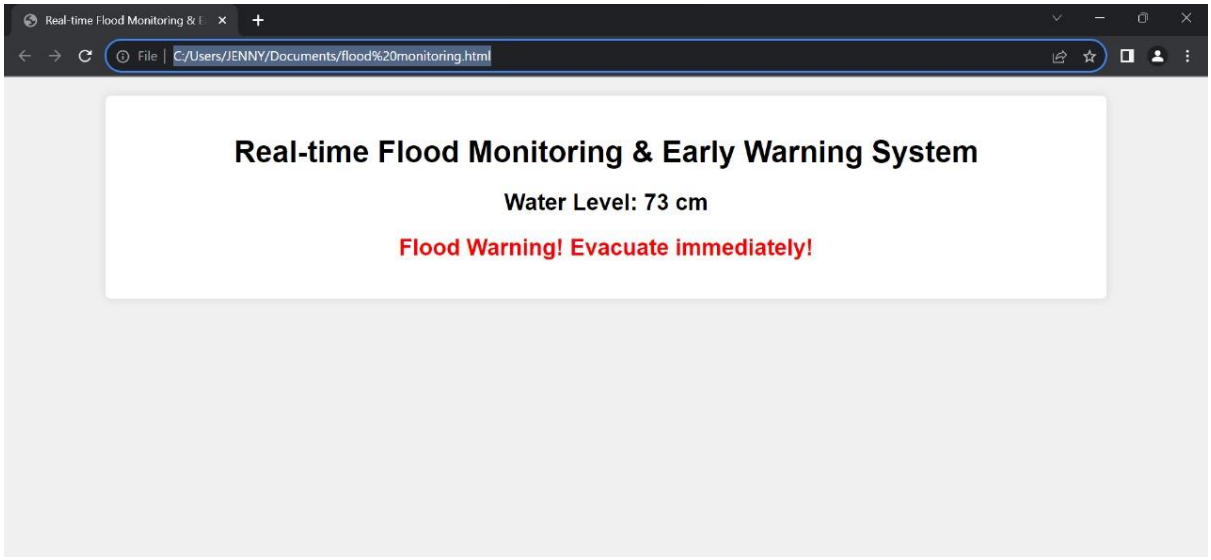
PLATFORM DEVELOPMENT:

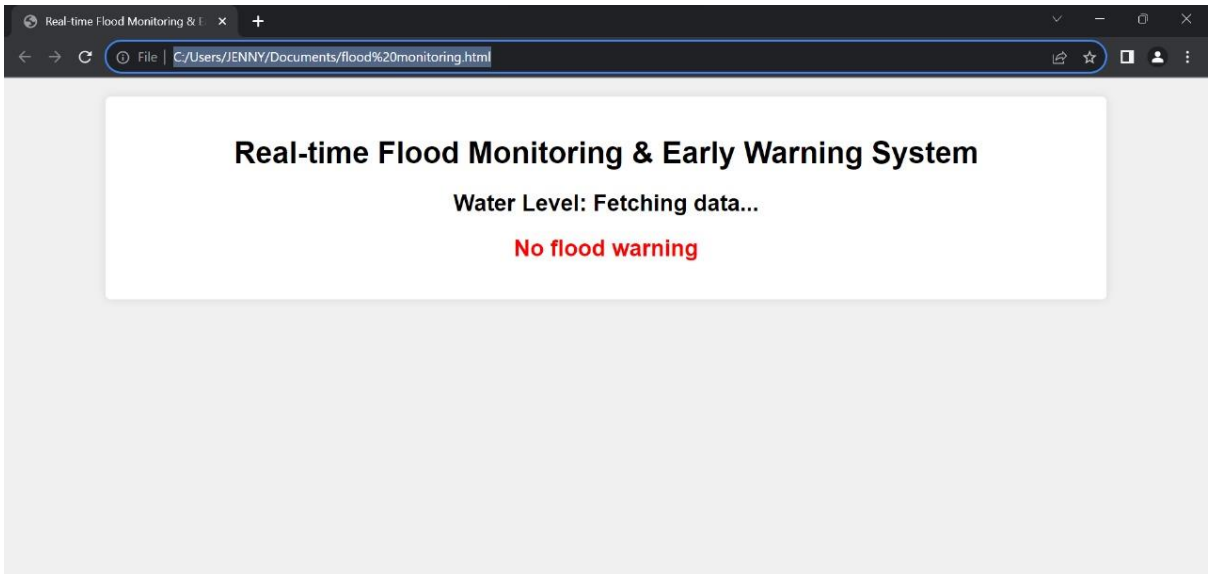
To use web development technologies (e.g., HTML, CSS, JavaScript) to create a platform that displays real-time water level data and flood warnings.

To design the platform to receive and display water level data from IoT sensors and issue flood warnings when necessary.

OUTPUTS:







CODE IMPLEMENTATION:

```
<!DOCTYPE html>
<html>
<head>
<title>Real-time Flood Monitoring & Early Warning System</title>
<style>
/* Basic styling for demo purposes - replace with your own styles */
body {
```

```
font-family: Arial, sans-serif;
background-color: #f0f0f0;
margin: 0;
padding: 0;
}
#data-container {
width: 80%;
margin: 20px auto;
padding: 20px;
background-color: #fff;
border-radius: 5px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
h1, h2 {
text-align: center;
}
#warning {
color: red;
}
</style>
</head>
<body>
<div id="data-container">
<h1>Real-time Flood Monitoring & Early Warning System</h1>
<h2>Water Level: <span id="waterLevel">Fetching data...</span></h2>
<h2 id="warning">No flood warning</h2>
</div>

<script>
// Simulating function to fetch real-time data (replace this with your actual data source)
```

```
function getRealTimeData() {
// Simulated data - replace with your data-fetching mechanism
return Math.floor(Math.random() * 100); // Random water level between 0 to 100 (for demonstration)
}

// Function to update real-time data and warnings
function updateData() {
const waterLevel = getRealTimeData();
document.getElementById('waterLevel').textContent = ${waterLevel} cm;

if (waterLevel > 70) {
document.getElementById('warning').textContent = 'Flood Warning! Evacuate immediately!';
} else {
document.getElementById('warning').textContent = 'No flood warning';
}
}

// Update data and warnings every 5 seconds (for demonstration)
setInterval(updateData, 5000);
</script>
</body>
</html>
```