Html:

```
<!DOCTYPE html>
<html lang=”en”>
<head>
    <meta charset=”UTF-8”>
    <meta name=”viewport”
content=”width=device-width, initial-
scale=1.0”>
    <title>Early Warning Platform</title>
    <link rel=”stylesheet”
href=”styles.css”>
</head>
<body>
    <header>
        <h1>Early Warning Platform</h1>
    </header>
    <section id=”data-display”>
        <div id=”water-level”>
            <h2>Current Water Level</h2>
```

```
            <p id=”current-
level”>Loading…</p>
        </div>
    </section>
    <section id=”warning-system”>
        <div id=”flood-warning”>
            <h2>Flood Warning</h2>
            <p id=”warning-status”>No
Warning</p>
        </div>
    </section>
</body>
</html>
Css:
Body {
    Font-family: Arial, sans-serif;
    Margin: 0;
    Padding: 0;
    Background-color: #f0f0f0;
}
```

```
Header {
    Background-color: #0074D9;
    Color: white;
    Text-align: center;
    Padding: 20px;
}

H1 {
    Font-size: 24px;
}

Section {
    Background-color: white;
    Padding: 20px;
    Margin: 20px;
    Border: 1px solid #ccc;
    Border-radius: 5px;
}
```

```
H2 {
    Font-size: 20px;
    Margin: 0;
}

P {
    Font-size: 16px;
}

#water-level {
    Background-color: #DFF6F0;
}

#warning-system {
    Background-color: #FDEBD0;
}
```

Html with javascript:

```
<!DOCTYPE html>
<html lang=”en”>
<head>
```

```
    <meta charset=”UTF-8”>
    <meta name=”viewport” content=”width=device-width, initial-scale=1.0”>
    <title>Early Warning Platform</title>
    <link rel=”stylesheet” href=”styles.css”>
</head>
<body>
    <header>
        <h1>Early Warning Platform</h1>
    </header>
    <section id=”data-display”>
        <div id=”water-level”>
            <h2>Current Water Level</h2>
            <p id=”current-level”>Loading…</p>
        </div>
    </section>
    <section id=”warning-system”>
```

```
        <div id=”flood-warning”>
            <h2>Flood Warning</h2>
            <p id=”warning-status”>No
Warning</p>
        </div>
    </section>

    <script>
        // Simulate real-time data updates
        Function updateData() {
            Const waterLevel =
Math.floor(Math.random() * 100); //
Random water level (for demonstration)

Document.getElementById(“current-
level”).textContent = waterLevel + “ cm”;

            // Simulate flood warning based on
water level
            If (waterLevel > 75) {
```

```
Document.getElementById(“warning-status”).textContent = “Flood Warning! Take Action!”;

Document.getElementById(“warning-status”).style.color = “red”;
            } else {

Document.getElementById(“warning-status”).textContent = “No Warning”;

Document.getElementById(“warning-status”).style.color = “green”;
            }
        }

        // Update data every 5 seconds (for demonstration)
        setInterval(updateData, 5000);
```

```
    </script>
</body>
</html>
```

IoT Flood Monitoring Platform

Creating an online platform for flood monitoring and an early warning system using IoT sensors, HTML, CSS, and JavaScript is a complex project. Here's a high-level overview of the components and steps involved:

IoT Sensors:

Install various types of sensors, such as water level sensors, rain gauges, weather stations, and perhaps even cameras, at flood-prone locations.

These sensors should be capable of collecting data and transmitting it to a central server.

Central Server:

Set up a central server to receive, process, and store data from the sensors.
Choose a suitable technology stack for server-side development, such as Node.js, Python, or Java.

Database:

Use a database system (e.g., MySQL, PostgreSQL, or NoSQL databases) to store sensor data.
Design the database schema to efficiently store and retrieve data.

Data Processing:

Develop algorithms to process the incoming data, detect anomalies, and assess flood risk.
Calculate relevant metrics and thresholds for flood warnings.

User Interface (UI):

Create a web-based UI using HTML, CSS, and JavaScript to visualize the data and provide a user-friendly interface for monitoring.
Use libraries like Leaflet or Google Maps for mapping and data visualization.

Real-time Updates:

Implement real-time updates in your UI using technologies like WebSockets to display data as it arrives.
Early Warning System:

Develop a system that can send alerts and warnings to relevant authorities and the public based on the data analysis.
Implement a notification mechanism, which can include email, SMS, or push notifications.

User Authentication and Authorization:

Implement user authentication to ensure that only authorized personnel can access certain features and data.

Mobile Responsiveness:

Ensure that your web platform is responsive, so it can be accessed and used on mobile devices.

Testing and Validation:

Rigorously test the system under different conditions, including simulations of flood events.
Validate the accuracy and reliability of sensor data and warning mechanisms.

Deployment:

Deploy the platform on a web server or cloud platform like AWS, Azure, or Google Cloud.

Documentation and Training:

Provide clear documentation for users and administrators.
Train relevant personnel on using and maintaining the system.

Continuous Monitoring and Maintenance:

Regularly update the system to ensure it remains accurate and reliable.

Monitor sensor health and replace or maintain them as needed.
This project is a significant undertaking that requires expertise in IoT, web development, data analysis, and possibly machine learning for accurate flood predictions. You may also need to collaborate with domain experts in meteorology and hydrology to develop precise warning systems.


NEW
HTML
RUN
index.html
styles.css
script
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width
6 <title>Early Warning Platform</title
7 <link rel="stylesheet" href="styles.
8 </head>
9 <body>
10 <header>
11 <h1>Early Warning Platform</h1>
12 </header>
13 <section id="data-display">
14
15 <h2>Current Water Level</h2>
16 <p id="current-level">Loadin
17
18 </section>
19 <section id="warning-system">
20
21 <h2>Flood Warning</h2>
22 <p id="warning-status">No Wa
23
24 </section>
25 </body>
26 </html>

Early Warning Platform

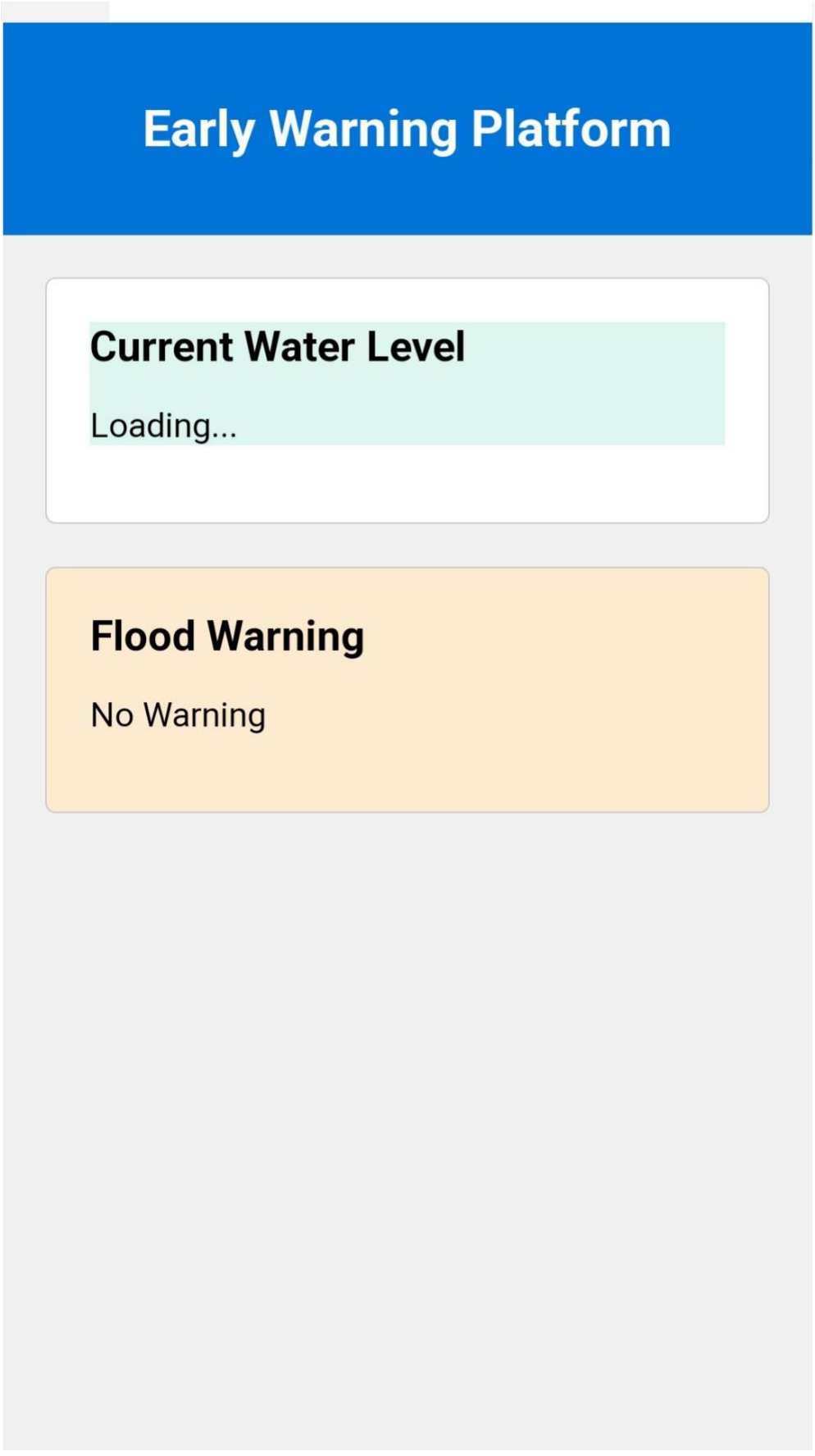
Current Water Level

Loading...

Flood Warning

No Warning

```css
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f0f0f0;
}

header {
    background-color: #0074D9;
    color: white;
    text-align: center;
    padding: 20px;
}

h1 {
    font-size: 24px;
}

section {
    background-color: white;
    padding: 20px;
    margin: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

h2 {
    font-size: 20px;
    margin: 0;
}

p {
    font-size: 16px;
}

#water-level {
    background-color: #DFF6F0;
}

#warning-system {
    background-color: #FDEBD0;
}
```


Early Warning Platform
Current Water Level
Loading...
Flood Warning
No Warning

styles.css script.js +

```
1  html with javascript:
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width
7      <title>Early Warning Platform</title
8      <link rel="stylesheet" href="styles.
9  </head>
10 <body>
11     <header>
12         <h1>Early Warning Platform</h1>
13     </header>
14     <section id="data-display">
15         <div id="water-level">
16             <h2>Current Water Level</h2>
17             <p id="current-level">Loadin
18         </div>
19     </section>
20     <section id="warning-system">
21         <div id="flood-warning">
22             <h2>Flood Warning</h2>
23             <p id="warning-status">No Wa
24         </div>
25     </section>
26
27     <script>
28         // Simulate real-time data updat
29         function updateData() {
30             const waterLevel = Math.floo
31             document.getElementById("cur
32
33             // Simulate flood warning ba
34             if (waterLevel > 75) {
35                 document.getElementById(
36                 document.getElementById(
37             } else {
38                 document.getElementById(
```

html with javascript:

Early Warning Platform

Current Water Level

45 cm

Flood Warning

No Warning