**ENVIRONMENTAL MONITORING USING IOT**

**PHASE 4**

**REAL TIME DATA DISPLAY IN WEB PAGE**

**AIM**

To create a platform that displays real-time environmental data using web develop technologies

**SOFTWARE USED**

* Notepad++
* LANGUAGES USED
* HTML
* CSS
* JavaScript

**BROWSER FOR DISPLAY**

* Microsoft Edge

**CODES**

**HTML CODE**

<!DOCTYPE html>

<html>

<head>

<title>ENVIRONMENTAL MONITORING USING IOT</title>

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

</head>

<body>

<h1>REAL TIME ENVIRONMENTAL DATA DISPLAY</h1>

<div id="data-display">

<p>Loading IoT data...</p>

</div>

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

</body>

</html>

**CSS CODE**

body {

font-family: Arial, sans-serif;

text-align: center;

margin: 20px;

}

h1 {

font-size: 24px;

margin-bottom: 20px;

}

.data-container {

display: inline-block;

margin: 10px;

}

.data-label {

font-weight: bold;

margin-bottom: 5px;

}

.data-value {

font-size: 18px;

}

**JAVASCRIPT CODE**

const channelID = '2306875';

const apiUrl = `https://api.thingspeak.com/channels/${channelID}/feeds.json?results=1`;

fetch(apiUrl)

.then((response) => response.json())

.then((data) => {

if (data.feeds.length > 0) {

const lastEntry = data.feeds[0];

const field1Data = lastEntry.field1;

const field2Data = lastEntry.field2;

dataDisplay.innerHTML = `

<p>Temperature: ${field1Data}'C</p>

<p>Humidity: ${field2Data}g/m^3</p>

`;

} else {

dataDisplay.innerHTML = '<p>No data available.</p>';

}

})

.catch((error) => {

console.error('Error fetching data from ThingSpeak:', error);

dataDisplay.innerHTML = '<p>Error fetching ThingSpeak data.</p>';

});

}

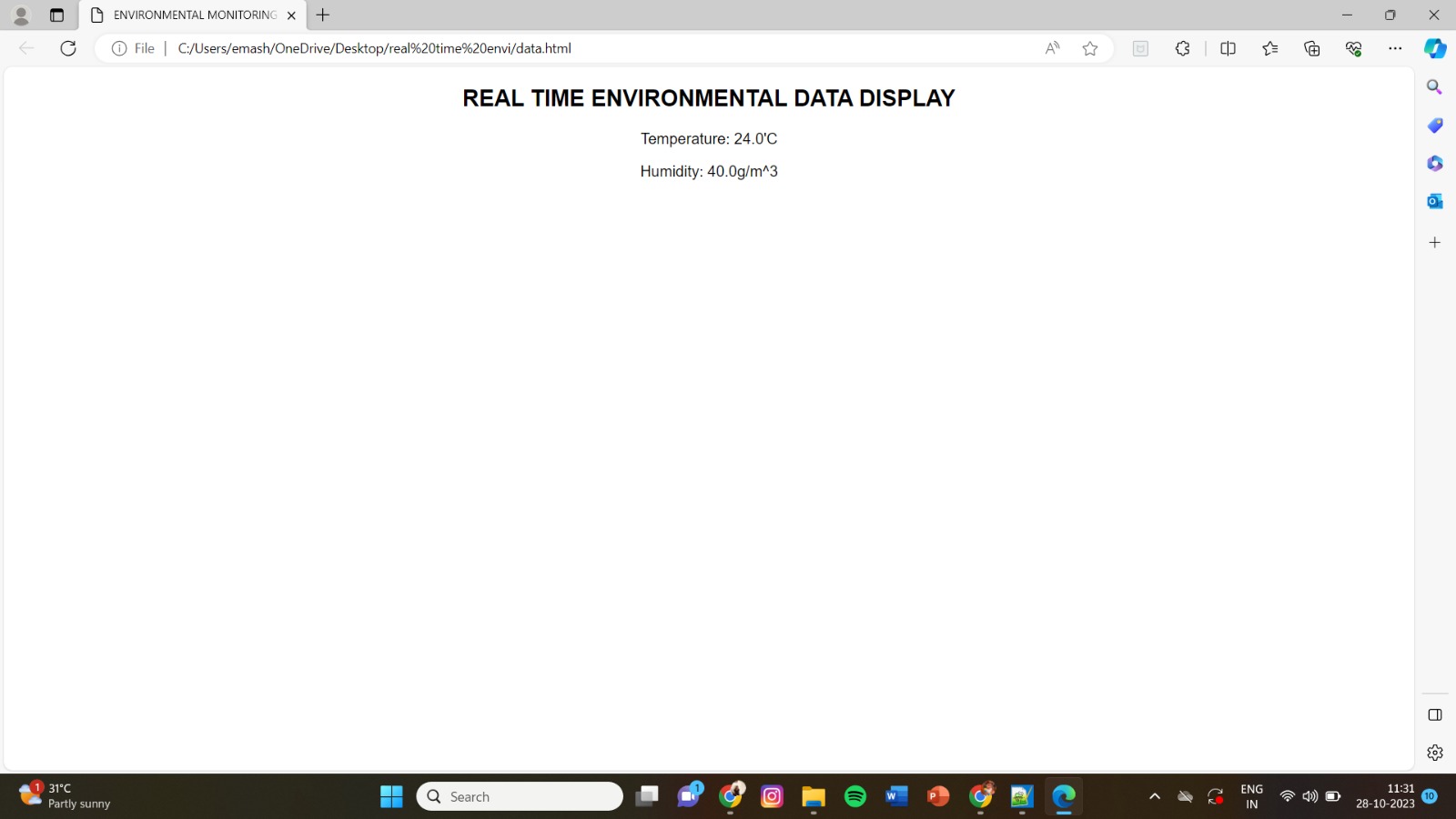
// Fetch ThingSpeak data initially

fetchThingSpeakData();

// Fetch ThingSpeak data periodically (e.g., every 30 seconds)

setInterval(fetchThingSpeakData, 30000); // 30,000 milliseconds = 30 seconds

**WEB DISPLAY**



**THANK YOU**