**PROJECT : ENVIRONMENTAL MONITORING**

**PHASE 4: DEVELOPMENT PART-2**

Building an environmental monitoring platform to display real-time temperature and humidity data from IoT devices involves creating a web-based interface. We used HTML, CSS, and JavaScript to achieve this. Here's a step-by-step guide to get started:

**1. Set Up the Project**:

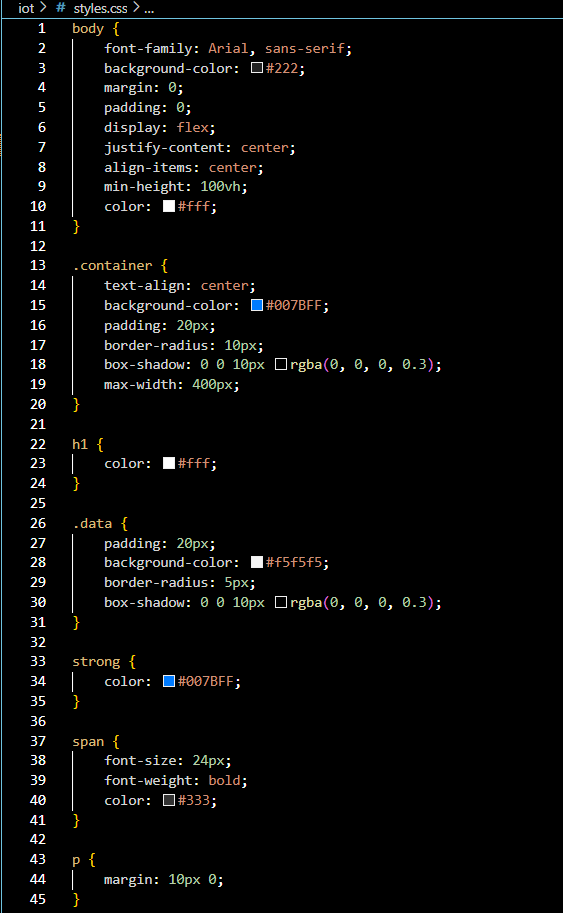
Before we start coding, we assured that we have a basic development environment set up. We used Visual Studio Code Editor and a web server to host our application. It involves configuring our development environment and tools to ensure a smooth and efficient development process.

**2. Create the HTML Structure**

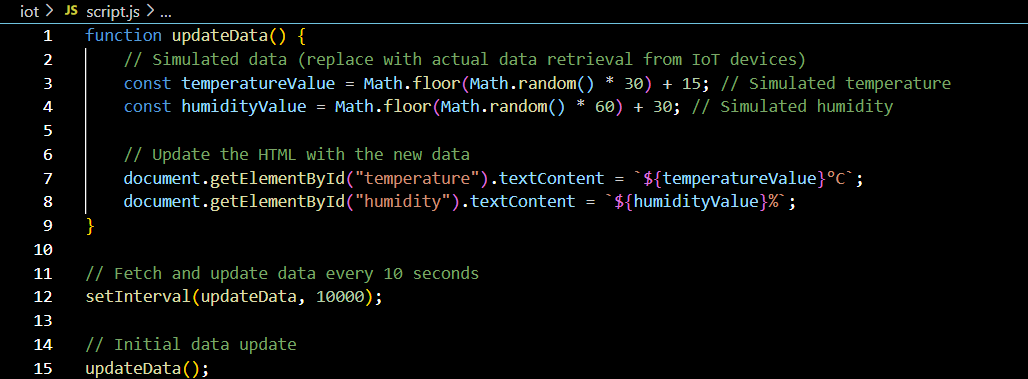
Start by creating the HTML structure for our environmental monitoring platform. Here's a simple example:

**3. Style Your Page (CSS)**:

We created a CSS file (styles.css) to style our page. This is where we added colours, fonts, and layout adjustments to make the platform visually appealing.

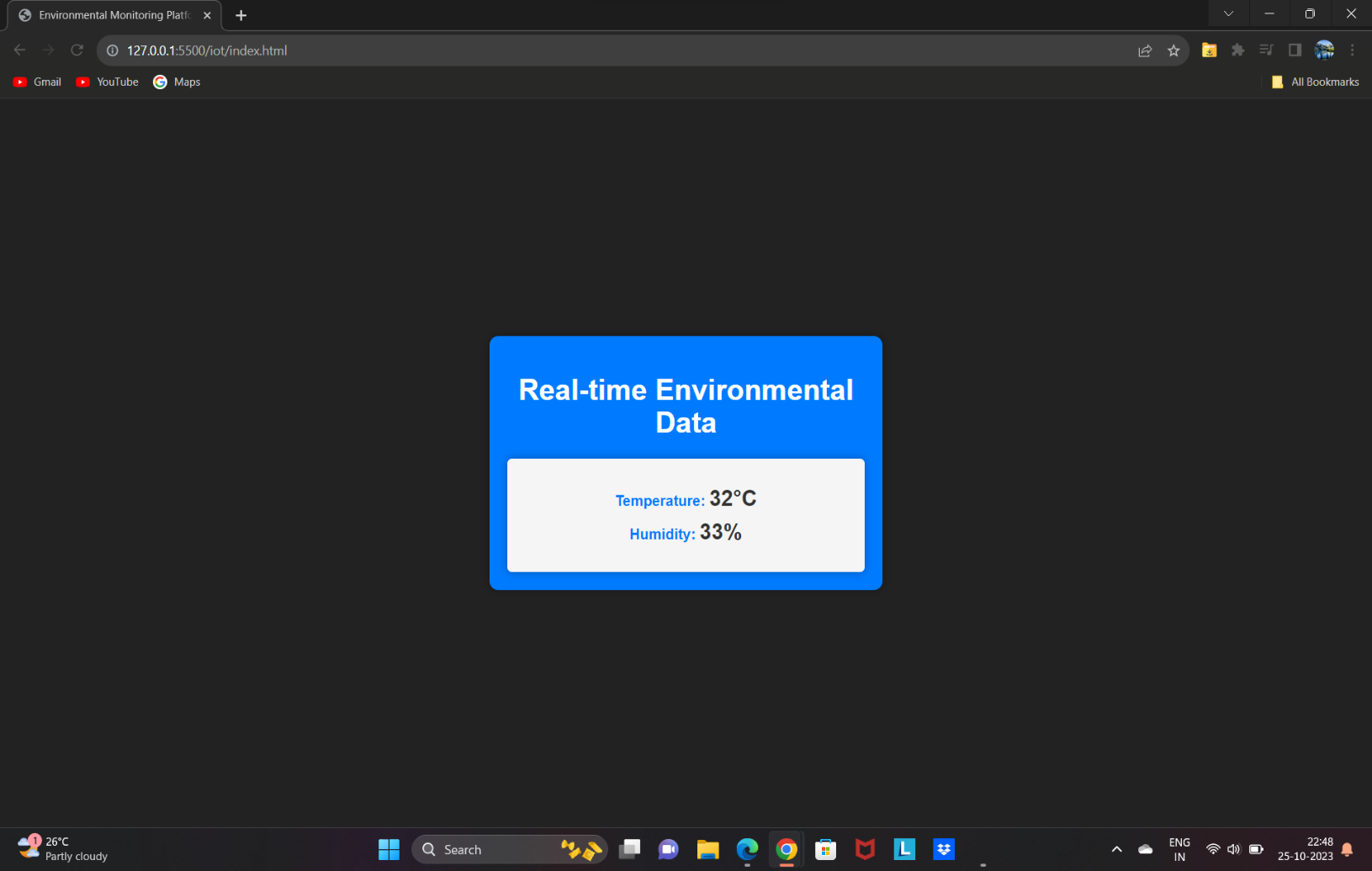


**4. JavaScript for Real-time Data (script.js)**:

Now, we needed JavaScript to fetch and display real-time data from IoT devices. We used AJAX or fetch API to get data from a server where IoT devices send updates. For this example, we used a simulated function:

**5. Server-Side Data Handling**:

In a real-world scenario, we set up a server that receives data from IoT devices and provides this data to the web interface via an API. This is beyond the scope of this simple example but is an essential step for a production-ready environmental monitoring platform.



**6. Deployment**:

Once we have created our web application, we have deploy it to a web server or a cloud platform to make it accessible to others. We might also consider using a service like AWS, Google Cloud, or Azure to handle the data and server-side operations.

**Conclusion:**

In summary, this project focuses on creating a web-based environmental monitoring platform that displays real-time temperature and humidity data from IoT devices. It utilizes HTML, CSS, and JavaScript to build a user-friendly interface. However, it's a starting point, and in real-world applications, we had need to consider security, data handling, and scalability while using actual IoT device data. This project contributes to better environmental understanding and monitoring.