Real-Time IOT Data Display with Firebase web development.

1. Introduction:

This document provides a step-by-step guide to displaying real-time IOT data from Firebase in a web browser by phase 3. The code provided includes HTML, CSS, and JavaScript components that enable to visualize real-time data from IOT device.

2. Prerequisites:

Firebase Account: Create a Firebase account at Firebase Console.

Firebase Configuration: Obtain your Firebase configuration details, including API Key, Auth Domain, Database URL, Project ID, Storage Bucket, Messaging Sender ID, and App ID.

Arduino /ESP32 Setup: Ensure your IOT device is sending data to Firebase Real time Database under fields like humidity and temperature

3. HTML Structure (index.html):

The index.html file provides the structure for displaying real-time IOT data.

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Real-Time IoT Data</title>
    <h1> ESP32 environmental monitoring data </h1>
   <style>
      body {
         font-family: Arial, sans-serif;
         margin: 20px;
      }
      .data-container {
         margin-top: 20px;
      }
      .data-item {
         margin-bottom: 10px;
      }
body {
   font-family: Arial, sans-serif;
   margin: 20px;
   background-color: #f7f7f7;
}
.data-container {
  margin-top: 20px;
```

```
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0px 0px 10px 0px rgba(0, 0, 0, 0.1);
    background-color: #ffffff;
}

.data-item {
    margin-bottom: 15px;
    font-size: 18px;
    color: #333333;
}

.data-item strong {
    color: #007bff; /* Change this color to your preferred color */
}

#humidity, #temperature {
    font-weight: bold;
    color: #28a745; /* Change this color to your preferred color */
}

/* Responsive styles for smaller screens */
@media screen and (max-width: 768px) {
  .data-container {
```

```
        padding: 15px;
    }

    .data-item {
        font-size: 16px;
    }
}
    </style>
</head>
<body>
    <div class="data-container">
        <div class="data-item">
            <strong>Humidity:</strong> <span id="humidity"></span> %
        </div>
        <div class="data-item">
            <strong>Temperature:</strong> <span id="temperature"></span> Â°C
        </div>
    </div>

    <script src="https://www.gstatic.com/firebasejs/9.5.0/firebase-app-compat.js"></script>
    <script src="https://www.gstatic.com/firebasejs/9.5.0/firebase-database-compat.js"></script>
    <script>
        // Firebase configuration
```

```
    const firebaseConfig = {
apiKey: "AIzaSyCy2aZiFAz0PYkOEt2Wgwr1raIgn7QV97k",
authDomain: "esp32-environmental-monitoring.firebaseapp.com",
databaseURL: "https://esp32-environmental-monitoring-default-rtdb.firebaseio.com",
projectId: "esp32-environmental-monitoring",
storageBucket: "esp32-environmental-monitoring.appspot.com",
messagingSenderId: "494025608341",
appId: "1:494025608341:web:bec4872274bb5844db5f9c",
measurementId: "G-M08YFMBQNC"
    };

    firebase.initializeApp(firebaseConfig);

    const database = firebase.database();

    // Reference to the 'data' node in your Firebase database
    const dataRef = database.ref('data');

    // Update UI with real-time data from Firebase
    dataRef.on('value', (snapshot) => {
        const data = snapshot.val();
        if (data) {
            document.getElementById('humidity').textContent = data.humidity || 'N/A';
            document.getElementById('temperature').textContent = data.temperature || 'N/A';
```

```
        } else {
            document.getElementById('humidity').textContent = 'N/A';
            document.getElementById('temperature').textContent = 'N/A';
        }
    });
  </script>
</body>
</html>
```

Output: