**Live Temperature and Humidity Monitoring using ESP32 and DHT11 on Web Server.**

**1. Objective**

To develop a real-time IoT system using ESP32 and DHT11 sensor that continuously monitors and displays ambient temperature and humidity data on a web page hosted by the ESP32. The data is auto-updated in real-time using AJAX, eliminating the need for page refreshes.

**2. Components Required**

| **Component** | **Quantity** |
| --- | --- |
| ESP32 Dev Board | 1 |
| DHT11 Sensor | 1 |
| Breadboard | 1 |
| Jumper Wires | 3–4 |
| USB Cable (for ESP32) | 1 |
| Computer (for coding) | 1 |

**3. Circuit Connections:**

| **DHT11 Pin** | **Connect To (ESP32)** |
| --- | --- |
| VCC | 3.3V |
| GND | GND |
| DATA | GPIO 4 (D4) |

**Note**: Avoid using 5V with DHT11 on ESP32. Always use **3.3V**.

**4. Software Requirements**

* Arduino IDE
* ESP32 Board Package (via Board Manager)
* Libraries:
  + DHT sensor library by Adafruit
  + Adafruit Unified Sensor
  + WebServer (already included with ESP32 core)

**5. Functional Overview**

* ESP32 connects to a Wi-Fi network.
* Reads temperature and humidity data from the DHT11 sensor.
* Hosts a web server on port 80 (default).
* Displays data on a custom HTML webpage.
* Uses JavaScript with AJAX to fetch new data every 2 seconds from the /data route without reloading the page.

**6. Source Code**

#include <WiFi.h>

#include <WebServer.h>

#include <DHT.h>

// Replace with your Wi-Fi credentials

const char\* ssid = "NSUT\_WIFI";

const char\* password = "";

// DHT setup

#define DHTPIN 4

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

// WebServer on port 80

WebServer server(80);

// HTML + JavaScript (AJAX auto-update)

void handleRoot() {

  String html = R"rawliteral(

    <!DOCTYPE html>

    <html>

    <head>

      <title>ESP32 DHT11 Monitor</title>

      <meta name="viewport" content="width=device-width, initial-scale=1">

      <style>

        body { font-family: Arial; text-align: center; padding: 30px; }

        h2 { font-size: 22px; }

        #data { font-size: 20px; margin-top: 20px; }

      </style>

    </head>

    <body>

      <h2>Temperature & Humidity Monitor</h2>

      <div id="data">Loading...</div>

      <script>

        setInterval(() => {

          fetch("/data")

            .then(response => response.text())

            .then(data => {

              document.getElementById("data").innerHTML = data;

            });

        }, 2000);  // refresh every 2 seconds

      </script>

    </body>

    </html>

  )rawliteral";

  server.send(200, "text/html", html);

}

void handleData() {

  float temperature = dht.readTemperature();

  float humidity = dht.readHumidity();

  if (isnan(temperature) || isnan(humidity)) {

    server.send(500, "text/plain", "Sensor Error");

    return;

  }

  String response = "Temperature: " + String(temperature) + " °C<br> Humidity: " + String(humidity) + " %";

  server.send(200, "text/html", response);

}

void setup() {

  Serial.begin(115200);

  dht.begin();

  // Connect to Wi-Fi

  WiFi.begin(ssid, password);

  Serial.print("Connecting to WiFi");

  while (WiFi.status() != WL\_CONNECTED) {

    delay(1000);

    Serial.print(".");

  }

  Serial.println("\n WiFi Connected");

  Serial.print("IP Address: ");

  Serial.println(WiFi.localIP());

  // Define routes

  server.on("/", handleRoot);

  server.on("/data", handleData);

  server.begin();

  Serial.println("Web Server Started");

}

void loop() {

  server.handleClient();  // Handle HTTP requests

}

**7. How It Works**

**setup()**

* Initializes Serial monitor, DHT sensor, and Wi-Fi connection.
* Starts the WebServer and defines routes:
  + / → serves the HTML page
  + /data → serves the latest temperature and humidity values

**loop()**

* Calls server.handleClient() to keep the web server running and responsive.

**handleRoot()**

* Sends an HTML page that uses JavaScript to fetch and display live sensor data from /data.

**handleData()**

* Reads sensor values and sends them as an HTML fragment back to the client when /data is requested.

**8. Output Example**

When accessed in a browser:

Temp: 27.3 °C

Humidity: 62 %

* Auto-updates every 2 seconds.
* No full-page reload required.

**9. Applications**

* Smart agriculture systems
* Smart home climate monitoring
* Weather stations
* Classroom IoT demonstration

**10. Troubleshooting Tips**

| **Issue** | **Solution** |
| --- | --- |
| The web page is not loading. | Ensure the correct IP from the Serial Monitor is used in the browser. |
| "Sensor error" is shown on the screen | DHT11 is not connected properly or is defective |
| Compilation error (DHT.h missing) | Install Adafruit DHT & Unified Sensor libraries |
| ESP32 not connecting to Wi-Fi | Double-check SSID and password; ensure 2.4GHz Wi-Fi |