ESP8266 DHT11/DHT22 Temperature and Humidity Web Server

Introduction

The project aims to build a simple yet effective IoT-based temperature and humidity monitoring system using an ESP8266 NodeMCU and a DHT11 or DHT22 sensor. The sensor collects real-time temperature and humidity data, and the ESP8266 hosts a web server that displays this information on a web page. The project demonstrates the power of embedded systems combined with web technology for remote environmental monitoring.

Objective

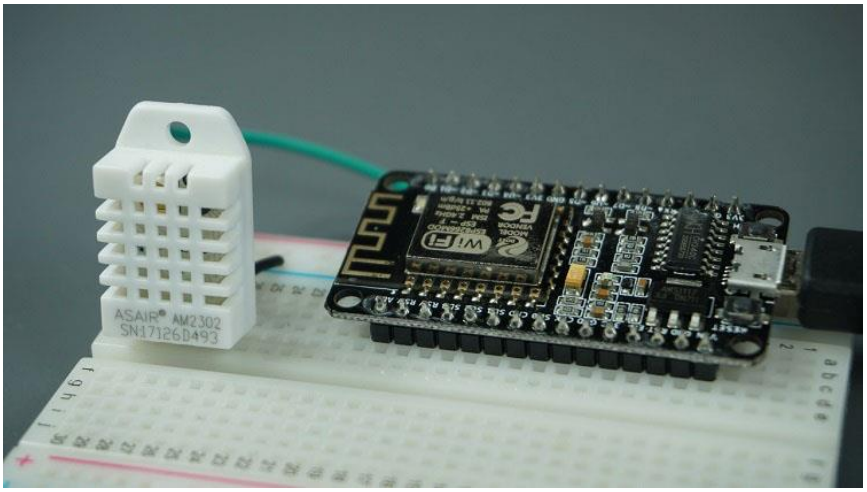
- To measure temperature and humidity using a DHT11/DHT22 sensor.
- To connect the ESP8266 NodeMCU to a Wi-Fi network.
- To display real-time sensor data on a web server interface hosted by the ESP8266.
- To understand how web technologies integrate with embedded systems.

Parts Required:

To build this project, you need the following parts:

Hardware Components

Component	Description
ESP8266 NodeMCU	Microcontroller with Wi-Fi
DHT11/DHT22	Temperature and humidity sensor
Breadboard	For easy circuit prototyping
Jumper Wires	To make electrical connections
Resistor (10kΩ)	Pull-up resistor for DHT signal
USB Cable	For programming and power

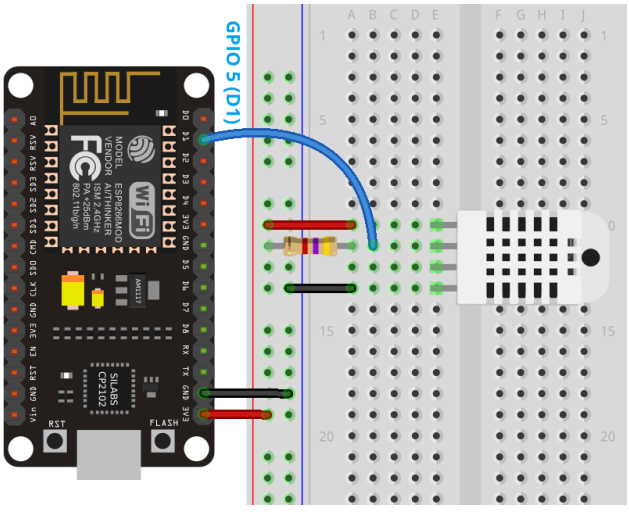


Software Requirements

- **Arduino IDE**
- **ESP8266 Board Package** (Installed via Board Manager)
- Libraries:
  - ESP8266WiFi.h
  - ESP8266WebServer.h
  - DHT.h

ESP8266 and DHT11/DHT22 Schematic Diagram

Before proceeding with the tutorial, wire the DHT11 or DHT22 temperature and humidity sensor to the ESP8266 as shown in the following schematic diagram.



Working Principle

1. Sensor Readings:
   The DHT sensor reads the surrounding temperature and humidity data.
2. Wi-Fi Connection:
   The ESP8266 connects to a predefined Wi-Fi network using provided SSID and password.
3. Web Server:
   A web server runs on the ESP8266. When accessed through a browser, it serves a HTML page displaying sensor data.
4. Auto-Refresh:
   The webpage auto-refreshes every few seconds to show the latest values, enabling real-time monitoring.

Code:

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <DHT.h>
// Wi-Fi credentials
const char* ssid = "nodemcu";
const char* password = "12345678";
// DHT setup
#define DHTPIN 5          // D1 = GPIO5
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
// Web server
ESP8266WebServer server(80);
// Wi-Fi connection
void connectToWiFi() {
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
```

```
  Serial.print("🔗 Connecting to Wi-Fi");
  int tries = 0;
  while (WiFi.status() != WL_CONNECTED && tries < 30) {
    Serial.print(".");
    delay(500);
    tries++;
  }
  Serial.println();
  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("✅ Wi-Fi Connected!");
    Serial.print("🌍 IP Address: ");
    Serial.println(WiFi.localIP());
  } else {
    Serial.println("❌ Failed to connect to Wi-Fi. Check SSID/password.");
  }
}
// HTML Web Interface
void handleRoot() {
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();
  String html = "<!DOCTYPE html><html><head>"
                "<meta http-equiv='refresh' content='5'/>"
                "<meta name='viewport' content='width=device-width, initial-scale=1.0'/>"
                "<title>Temperature & Humidity Monitor</title>"
                "<style>"
```

```
        "body { margin: 0; font-family: 'Segoe UI', sans-serif; background: linear-gradient(to right, #e0f7fa, #fce4ec); display: flex; flex-direction: column; align-items: center; justify-content: flex-start; min-height: 100vh; padding: 20px; }"
        ".container { max-width: 600px; width: 100%; }"
        ".card { background-color: #fff; border-radius: 15px; box-shadow: 0 6px 20px rgba(0,0,0,0.1); padding: 20px; margin: 15px 0; text-align: center; }"
        ".card h2 { color: #333; margin-bottom: 10px; }"
        ".value { font-size: 2.2em; font-weight: bold; margin-top: 10px; }"
        ".temp { background: linear-gradient(135deg, #ffebee, #ffcdd2); }"
        ".humid { background: linear-gradient(135deg, #e1f5fe, #b3e5fc); }"
        ".college { background: linear-gradient(135deg, #ede7f6, #d1c4e9); }"
        ".students { font-size: 1em; margin-top: 10px; width: 100%; border-collapse: collapse; }"
        ".students th, .students td { border: 1px solid #ccc; padding: 10px; }"
        ".students th { background-color: #f3e5f5; }"
        ".students td { background-color: #fafafa; }"
        "@media (max-width: 600px) { .value { font-size: 1.8em; } }"
        "</style></head><body>";
  html += "<div class='container'>";
  html += "<div class='card college'>";
  html += "<h2>Priyadarshini J. L. College of Engineering</h2>";
  html += "<h3>Dept. of Electronics & Telecommunication</h3>";
  html += "<h3>Temperature & Humidity Sensor</h3>";
  html += "</div>";
  if (isnan(temperature) || isnan(humidity)) {
    html += "<div class='card'><p><strong>⚠ Error reading from DHT sensor.</strong></p></div>";
  } else {
    html += "<div class='card temp'>";
```

```
    html += "<h2> Temperature</h2>";
    html += "<div class='value'>" + String(temperature, 1) + " °C</div>";
    html += "</div>";
    html += "<div class='card humid'>";
    html += "<h2>Humidity</h2>";
    html += "<div class='value'>" + String(humidity, 1) + " %</div>";
    html += "</div>";
  }
  html += "<div class='card'>";
  html += "<h2> Student Details</h2>";
  html += "<table class='students'>";
  html += "<tr><th>Name</th><th>Roll No.</th></tr>";
  html += "<tr><td>Reshma Satpute</td><td>06</td></tr>";
  html += "<tr><td>Vedant Pathrabe</td><td>28</td></tr>";
  html += "<tr><td>Lavanya Gabhane</td><td>40</td></tr>";
  html += "</table>";
  html += "</div>";
  html += "</div></body></html>";
  server.send(200, "text/html", html);
}
// Setup
void setup() {
  Serial.begin(115200);
  delay(1000);
  Serial.println("🚀 Booting...");
  dht.begin();
  delay(2000);
```

```
  connectToWiFi();
  server.on("/", handleRoot);
  server.begin();
  Serial.println("🌐 Web server started.");
}
// Main loop
void loop() {
  if (WiFi.status() != WL_CONNECTED) {
    connectToWiFi();
  }
  server.handleClient();
}
```

Advantages

- Real-time environmental monitoring.
- Simple and cost-effective design.
- No need for additional cloud service – works locally.
- Responsive and user-friendly interface**.**

Limitations

- Web interface is only accessible within the local network unless port forwarding or cloud integration is done.
- DHT11 has limited accuracy and range; DHT22 is recommended for better performance.
- Sensor data may be delayed by 1-2 seconds due to refresh rate.

Future Scope

The integration of ESP8266 with DHT11/DHT22 sensors to create a web-based temperature and humidity monitoring system holds significant potential for future advancements. As the demand for smart environmental monitoring grows, this project can be extended and scaled in various ways:

1. Cloud Integration:
   By connecting the ESP8266 to cloud platforms like ThingSpeak, Firebase, or Blynk, data can be stored, analyzed, and visualized in real-time from anywhere in the world. This allows long-term data logging and trend analysis.

2. Mobile App Control:
   Developing a companion mobile app can make it easier for users to access real-time sensor data, receive alerts, and even control connected appliances based on temperature/humidity thresholds.

3. IoT Automation and Smart Home Applications:
   The system can be integrated into smart home networks to automate fans, air conditioners, humidifiers, or dehumidifiers. This enables automatic climate control based on real-time sensor data.

4. Multiple Sensor Support:
   Adding support for multiple sensors across different rooms or environments can help create a distributed sensing network. This is useful for large-scale environmental monitoring like greenhouses, warehouses, or industrial settings.

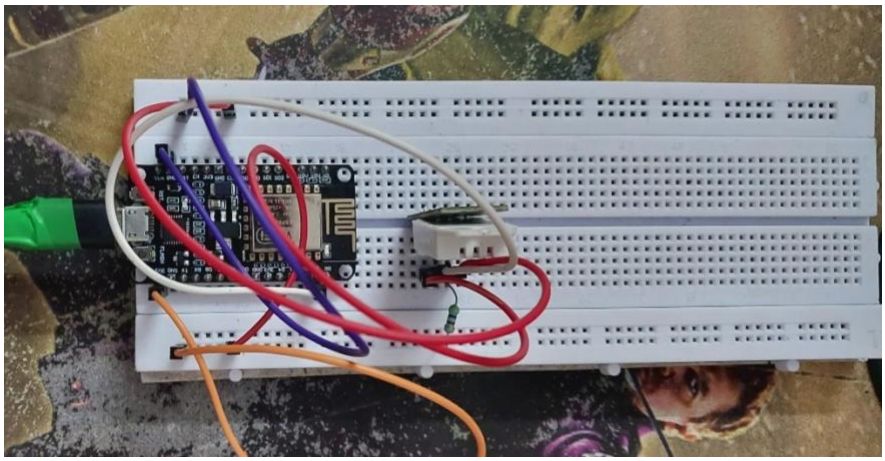
5. Battery Power and Energy Optimization:
   Future versions can be optimized for low power consumption and run on batteries or solar panels, enabling use in remote or outdoor environments.

6. Advanced Web Interface and Analytics:
   The basic web server can be enhanced with features like graphical data charts, downloadable reports, user authentication, and device settings, improving the usability and functionality of the system.

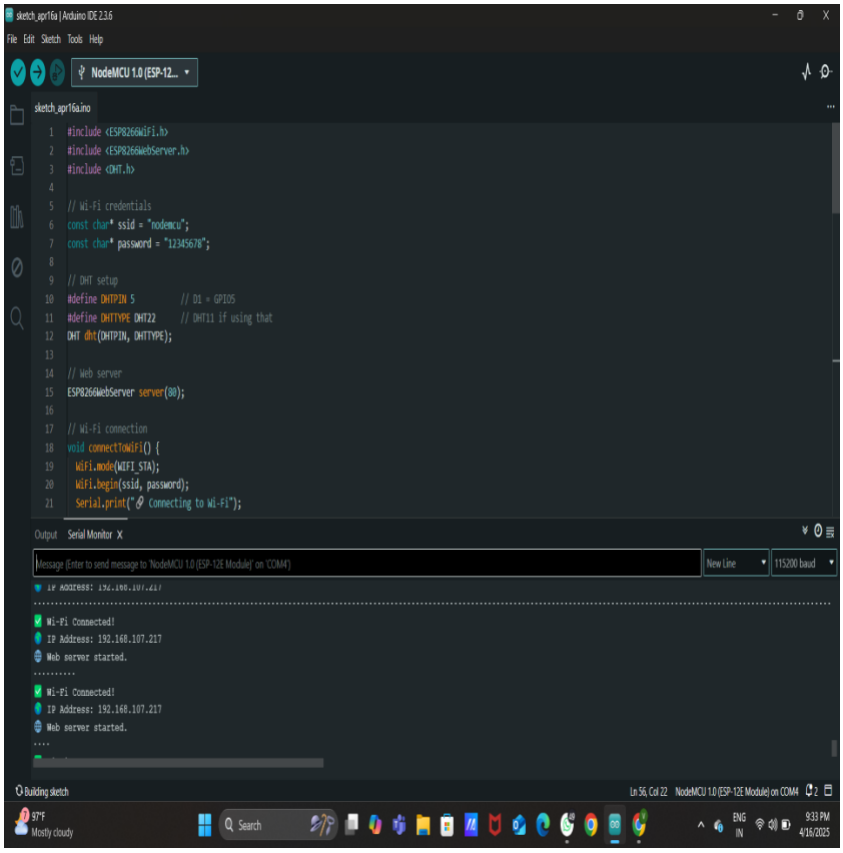
7. Integration with AI/ML:
   Incorporating basic AI models for predictive analytics (e.g., forecasting humidity or temperature changes) can make the system more intelligent and proactive in controlling the environment.

Output:

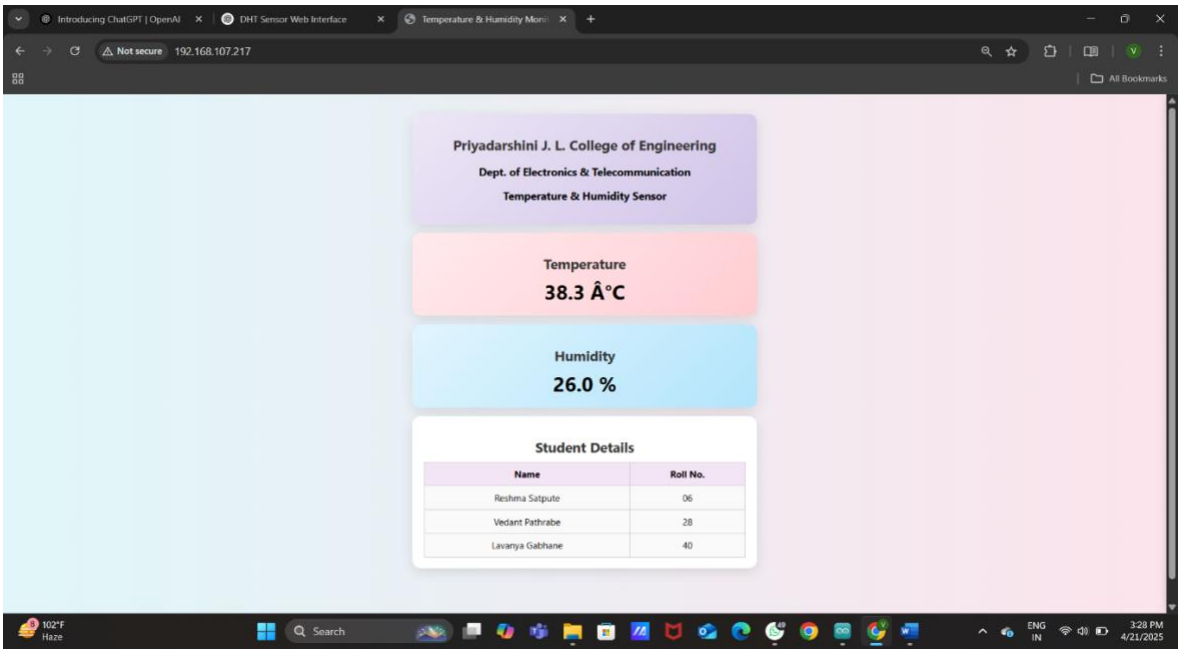
Breadboard connection:



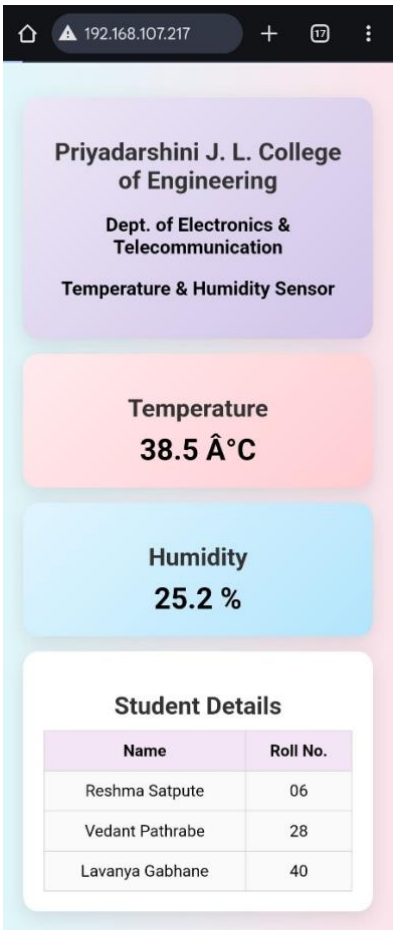
Web server interface:



Web interface of PC



Web interface of mobile



Conclusion:

This project successfully demonstrates how to use the ESP8266 microcontroller and a DHT sensor to create a standalone web-based temperature and humidity monitoring system. It serves as a practical foundation for more advanced IoT applications such as home automation, weather stations, or cloud-based data logging.