Arduino Uno R4 WiFi – Web Application Project

1. Project Overview

This project is a simple web-enabled monitoring system built using the **Arduino Uno R4 WiFi**. The system collects environmental data from two sensors and sends the processed data to a web application at fixed time intervals. The Arduino also periodically reports its operational status to confirm that it is running correctly.

The goal of the project is to demonstrate sensor integration, data processing, and WiFi-based communication using the Arduino Uno R4 WiFi.

---

2. Hardware Components

- **Arduino Uno R4 WiFi**
- **DHT11 Sensor** (Temperature and Humidity)
- **BMP280 Sensor** (Atmospheric Pressure and Temperature)
- Jumper wires
- Breadboard (optional)

---

3. Sensors Description

3.1 DHT11 Sensor

The DHT11 sensor is used to measure: - Ambient temperature - Relative humidity

The DHT11 is connected to the Arduino using a **GPIO (digital pin)**. It provides temperature data with low power consumption and is suitable for basic environmental monitoring.

3.2 BMP280 Sensor

The BMP280 sensor measures: - Atmospheric pressure - Temperature

The BMP280 is connected to the Arduino using the **I2C communication protocol**, which allows efficient data transfer using only two data lines (SDA and SCL).

---

4. Sensor Connections

Sensor	Connection Type	Arduino Interface
DHT11	GPIO	Digital Pin
BMP280	I2C	SDA / SCL

---

5. Data Processing

Both sensors provide temperature readings. To improve reliability, the system calculates a **final temperature value** as the average of the temperatures measured by the DHT11 and the BMP280.

Final Temperature Calculation: - Temperature from DHT11 - Temperature from BMP280 - Final temperature = (DHT11 temperature + BMP280 temperature) / 2

In addition to temperature, the system also records: - Humidity from the DHT11 - Atmospheric pressure from the BMP280

---

6. Communication and Data Transmission

The Arduino Uno R4 WiFi communicates with a remote server using **REST API calls over HTTP**. This client–server architecture enables reliable and scalable data exchange between the hardware device and the web application.

The Arduino acts as an **HTTP client**, sending JSON-formatted requests to the server endpoints exposed by the backend.

6.1 Alive Status Message

To ensure system reliability, the Arduino sends a **status (alive) message every 1 minute**. This message confirms that the device is powered on and functioning correctly.

6.2 Sensor Data Transmission

Environmental data are transmitted to the web application **every 3 minutes** using HTTP POST requests. Each data packet includes: - Average temperature - Humidity - Atmospheric pressure

---

7. Web Application

The web application is developed using **PHP** for both the **backend and frontend**.

7.1 Backend

The backend is responsible for: - Exposing REST API endpoints for the Arduino client - Receiving and validating sensor data - Receiving alive status messages - Storing data in the database

Sensor data received from the Arduino are stored in a **MySQL database**, allowing long-term storage and future analysis.

7.2 Frontend

The frontend, also developed in PHP, provides a web interface to: - Display the latest sensor readings - Visualize historical data - Monitor device connectivity using alive messages

---

8. Conclusion

This project demonstrates a simple and effective way to use the Arduino Uno R4 WiFi for environmental monitoring. By combining multiple sensors, averaging temperature values, and using periodic communication, the system provides reliable data suitable for web-based monitoring applications.