```
#include "DHT.h"
#include <Wire.h>
#include <RH_NRF24.h>

// ---------------------- Pin Definitions ----------------------
#define DHTPIN 2
#define DHTTYPE DHT11
#define LDR_PIN A1
#define SOIL_PIN A0
#define LED_PIN LED_BUILTIN

// ---------------------- Sensor & NRF Setup ----------------------
DHT dht(DHTPIN, DHTTYPE);
RH_NRF24 nrf24(7, 8); // CE, CSN

// Soil calibration values (change as per your calibration)
const int DRY_VALUE = 700;
const int WET_VALUE = 300;

// ---------------------- Helper Functions ----------------------
void blinkLED(int onTime, int times, int offTime = 500) {
  for (int i = 0; i < times; i++) {
    digitalWrite(LED_PIN, HIGH);
    delay(onTime);
    digitalWrite(LED_PIN, LOW);
    delay(offTime);
  }
  Serial.println();
}

void sendData(int brightness, int soilMoisture, float temp, float humidity) {
  Serial.println("\nSending data via NRF24...");

  uint8_t data[6];
  data[0] = 1;                    // Node ID
  data[1] = soilMoisture;         // Soil Moisture (0-100)
  data[2] = (uint8_t)temp;        // Temperature (approx.)
  data[3] = (uint8_t)humidity;    // Humidity (approx.)
  data[4] = (uint8_t)brightness;  // Brightness (0-100)
  data[5] = 0;                    // Reserved

  for (int i = 0; i < 5; i++) {
    delay(500);
    nrf24.send(data, sizeof(data));
    nrf24.waitPacketSent();

    uint8_t buf[RH_NRF24_MAX_MESSAGE_LEN];
    uint8_t len = sizeof(buf);
```

```
    if (nrf24.waitAvailableTimeout(1000)) {
      if (nrf24.recv(buf, &len)) {
        Serial.print("Master Acknowledged: ");
        Serial.println((char*)buf);
        break;
      } else {
        Serial.println("Receive failed");
      }
    } else {
      Serial.println("No response from master.");
    }
  }

  Serial.println("Transmission complete.");
  blinkLED(2000, 3, 500);
}

// ---------------------- Setup ----------------------
void setup() {
  Serial.begin(9600);
  dht.begin();
  Wire.begin();

  pinMode(LED_PIN, OUTPUT);
  pinMode(LDR_PIN, INPUT);
  pinMode(SOIL_PIN, INPUT);

  Serial.println("Slave Node 1 Starting...");

  if (!nrf24.init()) {
    Serial.println("NRF24 initialization failed!");
  }

  if (!nrf24.setChannel(3)) {
    Serial.println("Failed to set NRF24 channel.");
  }

  if (!nrf24.setRF(RH_NRF24::DataRate2Mbps, RH_NRF24::TransmitPower0dBm)) {
    Serial.println("Failed to set NRF24 RF parameters.");
  }

  Serial.println("NRF24 ready. Starting transmission...");
  blinkLED(800, 5, 300);
}

// ---------------------- Main Loop ----------------------
void loop() {
  // Read LDR
  int ldrRaw = analogRead(LDR_PIN);
```

```
  int brightness = round((ldrRaw / 1023.0) * 200);
  Serial.print("Brightness: ");
  Serial.print(brightness);
  Serial.println(" %");

  // Read Soil Moisture
  int soilRaw = analogRead(SOIL_PIN);
  int soilMoisture = map(soilRaw, DRY_VALUE, WET_VALUE, 0, 100);
  soilMoisture = constrain(soilMoisture, 0, 100);
  Serial.print("Soil Moisture: ");
  Serial.print(soilMoisture);
  Serial.println(" %");

  // Read Temperature and Humidity
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT11 sensor!");
  } else {
    Serial.print("Temperature: ");
    Serial.print(temperature);
    Serial.println(" 掳C");

    Serial.print("Humidity: ");
    Serial.print(humidity);
    Serial.println(" %");

    // Transmit data via NRF24
    sendData(brightness, soilMoisture, temperature, humidity);
  }

  delay(1000);
}
```