```
#define BLYNK_TEMPLATE_ID "TMPL3VJxQUV-f"
#define BLYNK_TEMPLATE_NAME "FIRE DETECTION"
#define BLYNK_AUTH_TOKEN "ge-I6bhy6EmeHe-dygd_RzxS6XMBXRwW"

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>

// Wi-Fi credentials
char ssid[] = "Mahesh_3g";
char pass[] = "789251837";

// Sensor and actuator pins
const int MQ2_PIN       = 34;  // MQ2 analog pin
const int DHT_PIN       = 23;  // DHT11 data pin
const int BUZZER_PIN    = 4;   // Buzzer pin
const int GREEN_LED_PIN = 18;  // Wi-Fi status
const int RED_LED_PIN   = 19;  // Fire or sensor failure

DHT dht(DHT_PIN, DHT11);

// Thresholds
const int GAS_THRESHOLD = 400;
const float TEMP_THRESHOLD = 38;
const float HUMIDITY_LOW_THRESHOLD = 10;

BlynkTimer timer;

// Flags
bool fireAlertSent = false;
bool dhtFailSent = false;
bool mq2FailSent = false;
bool deviceEnabled = true; // Remote ON/OFF flag

// Remote ON/OFF control from Blynk V6
BLYNK_WRITE(V6) {
  deviceEnabled = param.asInt();  // 1 = ON, 0 = OFF
}

void setup() {
  Serial.begin(115200);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(GREEN_LED_PIN, OUTPUT);
  pinMode(RED_LED_PIN, OUTPUT);
```

```
  dht.begin();
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  timer.setInterval(2000L, sendSensorData); // every 2 seconds
}

void sendSensorData() {
  if (!deviceEnabled) {
    Serial.println("[INFO] Device disabled from Blynk.");
    digitalWrite(BUZZER_PIN, LOW); // ensure buzzer is off
    digitalWrite(RED_LED_PIN, LOW); // turn off red LED
    return; // skip sensor reading
  }

  int gasValue = analogRead(MQ2_PIN);
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  Serial.print("MQ2 Gas: "); Serial.println(gasValue);
  Serial.print("Temp: "); Serial.print(temperature);
  Serial.print(" °C  Humidity: "); Serial.print(humidity); Serial.println(" %");

  // Send values to Blynk
  Blynk.virtualWrite(V0, temperature);
  Blynk.virtualWrite(V1, humidity);
  Blynk.virtualWrite(V2, gasValue);

  bool dhtFail = isnan(temperature) || isnan(humidity);
   bool mq2Fail = gasValue >= 4095;
  bool fireDetected = gasValue > GAS_THRESHOLD && temperature > TEMP_THRESHOLD;

  //  Fire condition
  if (fireDetected) {
    digitalWrite(BUZZER_PIN, HIGH);
    digitalWrite(RED_LED_PIN, HIGH); // Solid red LED
    if (!fireAlertSent) {
      Blynk.logEvent("gas", " Flammable Gas Detected!");
      Blynk.logEvent("temp", " High Temperature!");
      fireAlertSent = true;
    }
  } else {
    digitalWrite(BUZZER_PIN, LOW);
    fireAlertSent = false;
  }
```

```
  //  Sensor health LED status
  if (dhtFail || mq2Fail) {
    digitalWrite(RED_LED_PIN, millis() / 300 % 2); // Blink red LED
  } else if (!fireDetected) {
    digitalWrite(RED_LED_PIN, LOW); // Turn off red LED
  }

  // ✅ Sensor health display on Blynk
  if (dhtFail) {
    if (!dhtFailSent) {
      Blynk.logEvent("sensor_dht_fail", "❌ DHT Sensor Failure");
      dhtFailSent = true;
    }
    Blynk.virtualWrite(V3, "❌ FAIL");
  } else {
    dhtFailSent = false;
    Blynk.virtualWrite(V3, "✅ OK");
  }

  if (mq2Fail) {
    if (!mq2FailSent) {
      Blynk.logEvent("sensor_mq2_fail", "❌ MQ2 Sensor Failure");
      mq2FailSent = true;
    }
    Blynk.virtualWrite(V4, "❌ FAIL");
  } else {
    mq2FailSent = false;
    Blynk.virtualWrite(V4, "✅ OK");
  }
}

void loop() {
  Blynk.run();
  timer.run();

  //  Green LED logic
  if (!deviceEnabled) {
    digitalWrite(GREEN_LED_PIN, LOW); // OFF when device is disabled
  } else if (WiFi.status() == WL_CONNECTED) {
    digitalWrite(GREEN_LED_PIN, HIGH); // Solid ON when connected
  } else {
    digitalWrite(GREEN_LED_PIN, millis() / 500 % 2); // Blinking when not connected
  }
```

```
}
```