```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <DHT.h>

// ---------------- TELEGRAM SETTINGS ----------------
String BOT_TOKEN = "8272657678:AAGH8TcWHrJcC9xxc8eWEo8rvMNt7TJmTaA";
String CHAT_ID   = "6447411951";

// ---------------- WIFI SETTINGS ----------------
const char* WIFI_SSID = "iphone";
const char* WIFI_PASSWORD = "dhanush76";

// ---------------- PIN SETTINGS ----------------
#define DHTPIN 4
#define DHTTYPE DHT11
#define RELAY_PIN 15
#define SWITCH_PIN 18  // active LOW

DHT dht(DHTPIN, DHTTYPE);
WiFiClientSecure client;

// -------- EDITABLE LIMITS --------
float HUMIDITY_LIMIT = 75;
float TEMP_LIMIT     = 30;
// ---------------------------------

bool relayState = false;         // current relay state
unsigned long relayOnStart = 0;    // for 3-minute timer
```

```
unsigned long FORCE_ON_TIME = 3UL * 60UL * 1000UL;  // 3 minutes

// ---------------- URL ENCODE FUNCTION ----------------
String urlEncode(String str) {
 String encoded = "";
 char c;
 char buf[4];
 for (int i = 0; i < str.length(); i++) {
  c = str.charAt(i);
  if (isalnum(c)) {
   encoded += c;
  } else if (c == ' ') {
   encoded += "%20";
  } else if (c == '\n') {
   encoded += "%0A";
  } else {
   sprintf(buf, "%%%02X", (unsigned char)c);
   encoded += buf;
  }
 }
 return encoded;
}

// ---------------- TELEGRAM FUNCTION ----------------
void sendTelegram(String msg) {
 if (WiFi.status() != WL_CONNECTED) return;

 client.setInsecure();
```

```
 if (!client.connect("api.telegram.org", 443)) {
  Serial.println("❌ Telegram connect fail");
  return;
 }

 String encodedMsg = urlEncode(msg);  // ✅ encode everything
 String url = "/bot" + BOT_TOKEN + "/sendMessage?chat_id=" + CHAT_ID +
              "&text=" + encodedMsg;

 client.println("GET " + url + " HTTP/1.1");
 client.println("Host: api.telegram.org");
 client.println("Connection: close");
 client.println();

 Serial.println("Telegram Sent: " + msg);
}

// ---------------- SETUP ----------------
void setup() {
 Serial.begin(115200);

 pinMode(RELAY_PIN, OUTPUT);
 digitalWrite(RELAY_PIN, LOW);

 pinMode(SWITCH_PIN, INPUT_PULLUP); // switch to GND

 dht.begin();
```

```
 Serial.println("Connecting to WiFi...");
 WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
 while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
 }
 Serial.println("\nWiFi Connected!");
}

// ---------------- LOOP ----------------
void loop() {
 float h = dht.readHumidity();
 float t = dht.readTemperature();

 bool switchOn = (digitalRead(SWITCH_PIN) == LOW); // active LOW

 // ---------------- SENSOR READ FAILED ----------------
 if (isnan(h) || isnan(t)) {
  Serial.println(" ❌ DHT11 read failed");
  delay(1000);
  return;
 }

 bool dhtShouldTurnOn = (h < HUMIDITY_LIMIT || t > TEMP_LIMIT);

 // ---------------- RELAY ON LOGIC ----------------
 if (!relayState) {
  if (switchOn) {
```

```
    relayState = true;
    digitalWrite(RELAY_PIN, HIGH);
    relayOnStart = millis();
    sendTelegram("⭕ Switch ON → Relay ON\nTemp: " + String(t) + "°C\nHumidity: " + String(h) + "%");
  } else if (dhtShouldTurnOn) {
    relayState = true;
    digitalWrite(RELAY_PIN, HIGH);
    relayOnStart = millis();
    sendTelegram("🌡️ Auto ON (DHT)\nTemp: " + String(t) + "°C\nHumidity: " + String(h) + "%");
  }
 }

 // ---------------- RELAY OFF LOGIC ----------------
 if (relayState) {
  bool shouldStayOn = switchOn || dhtShouldTurnOn;

  if (!shouldStayOn) {
   // If relay was ON due to DHT, enforce timer
   if (millis() - relayOnStart >= FORCE_ON_TIME || !dhtShouldTurnOn) {
    relayState = false;
    digitalWrite(RELAY_PIN, LOW);
    sendTelegram("⚡ Relay OFF\nTemp: " + String(t) + "°C\nHumidity: " + String(h) + "%");
   }
  }
 }
```

```
  delay(1000);
}
```