**ESP8266 CODE**

#include <ESP8266WiFi.h>

#include <ESP8266HTTPClient.h>

#include <SoftwareSerial.h>

SoftwareSerial espSerial(0, 1); // RX, TX

// WiFi credentials

const char\* ssid = "YOUR WIFI SSID";

const char\* password = "YOUR WIFI PASSWORD";

// Server details (use the actual IP address of your server)

const char\* serverurl = "http://192.168.37.253:8000/iot/esp8266\_endpoint";

void setup() {

espSerial.begin(9600);

// Initialize serial communication for debugging

Serial.begin(9600); // Set the baud rate to 9600

delay(100);

// Connect to Wi-Fi

WiFi.begin(ssid, password);

Serial.print("Connecting to ");

Serial.print(ssid);

Serial.println("...");

// Wait for connection

int retries = 0;

const int maxRetries = 30; // 30 retries = 15 seconds (500ms per retry)

while (WiFi.status() != WL\_CONNECTED && retries < maxRetries) {

delay(500);

Serial.print(".");

retries++;

}

if (WiFi.status() == WL\_CONNECTED) {

Serial.println();

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

} else {

Serial.println();

Serial.println("Failed to connect to WiFi");

}

}

void loop() {

if (WiFi.status() == WL\_CONNECTED && espSerial.available()) {

String data = espSerial.readStringUntil('\n');

WiFiClient client;

HTTPClient http;

http.begin(client, serverurl);

http.addHeader("Content-Type", "application/x-www-form-urlencoded");

// Send POST request

int httpResponseCode = http.POST(data);

// Check the returning code

if (httpResponseCode > 0) {

Serial.printf("HTTP POST code: %d\n", httpResponseCode);

if (httpResponseCode == HTTP\_CODE\_OK) {

String payload = http.getString();

Serial.println("Response:");

Serial.println(payload);

}

} else {

Serial.printf("HTTP POST request failed, error: %s\n", http.errorToString(httpResponseCode).c\_str());

}

// Close connection

http.end();

}

delay(1000);

}