#include <WiFi.h>

#include <WiFiClient.h>

#include <WebServer.h>

#include <ESPmDNS.h>

#include <DHT.h>

const char \*ssid = " Mohammad’s iphone ";

const char \*password = "12345678";

WebServer server(80);

DHT dht(26, DHT11);

void handleRoot() {

  char msg[1500];

  snprintf(msg, 1500,

           "<html>\

  <head>\

    <meta http-equiv='refresh' content='4'/>\

    <meta name='viewport' content='width=device-width, initial-scale=1'>\

    <link rel='stylesheet' href='https://use.fontawesome.com/releases/v5.7.2/css/all.css' integrity='sha384-fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr' crossorigin='anonymous'>\

    <title>ESP32 DHT Server</title>\

    <style>\

    html { font-family: Arial; display: inline-block; margin: 0px auto; text-align: center;}\

    h2 { font-size: 3.0rem; }\

    p { font-size: 3.0rem; }\

    .units { font-size: 1.2rem; }\

    .dht-labels{ font-size: 1.5rem; vertical-align:middle; padding-bottom: 15px;}\

    </style>\

  </head>\

  <body>\

      <h2>ESP32 DHT Server!</h2>\

      <p>\

        <i class='fas fa-thermometer-half' style='color:#ca3517;'></i>\

        <span class='dht-labels'>Temperature</span>\

        <span>%.2f</span>\

        <sup class='units'>&deg;C</sup>\

      </p>\

      <p>\

        <i class='fas fa-tint' style='color:#00add6;'></i>\

        <span class='dht-labels'>Humidity</span>\

        <span>%.2f</span>\

        <sup class='units'>&percnt;</sup>\

      </p>\

  </body>\

</html>",

           readDHTTemperature(), readDHTHumidity()

          );

  server.send(200, "text/html", msg);

}

void setup(void) {

  Serial.begin(115200);

  dht.begin();

  WiFi.mode(WIFI\_STA);

  WiFi.begin(ssid, password);

  Serial.println("");

  // Wait for connection

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

    Serial.print(".");

  }

  Serial.println("");

  Serial.print("Connected to ");

  Serial.println(ssid);

  Serial.print("IP address: ");

  Serial.println(WiFi.localIP());

  if (MDNS.begin("esp32")) {

    Serial.println("MDNS responder started");

  }

  server.on("/", handleRoot);

  server.begin();

  Serial.println("HTTP server started");

}

void loop(void) {

  server.handleClient();

  delay(2);//allow the cpu to switch to other tasks

}

float readDHTTemperature() {

  // Sensor readings may also be up to 2 seconds

  // Read temperature as Celsius (the default)

  float t = dht.readTemperature();

  if (isnan(t)) {

    Serial.println("Failed to read from DHT sensor!");

    return -1;

  }

  else {

    Serial.println(t);

    return t;

  }

}

float readDHTHumidity() {

  // Sensor readings may also be up to 2 seconds

  float h = dht.readHumidity();

  if (isnan(h)) {

    Serial.println("Failed to read from DHT sensor!");

    return -1;

  }

  else {

    Serial.println(h);

    return h;

  }

}