

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA VẬT LÝ**



ĐỒ ÁN CHUYÊN NGÀNH

**HỆ THỐNG GIÁM SÁT CHẤT LƯỢNG MÔI
TRƯỜNG THÔNG MINH SỬ DỤNG CÔNG NGHỆ
IOT**

Giảng viên hướng dẫn : PGS. TS. Nguyễn Bách Khoa

Sinh viên thực hiện 1 : Nguyễn Minh Hiếu

MSSV 1 : [Điền MSSV của bạn]

Chuyên ngành : Khoa học dữ liệu

Khóa : 2020

Chương trình đào tạo : Chương trình đào tạo Chuẩn

Hà Nội - 2025

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA VẬT LÝ**



ĐỒ ÁN CHUYÊN NGÀNH

**HỆ THỐNG GIÁM SÁT CHẤT LƯỢNG MÔI
TRƯỜNG THÔNG MINH SỬ DỤNG CÔNG NGHỆ
IOT**

Giảng viên hướng dẫn : PGS. TS. Nguyễn Bách Khoa

Sinh viên thực hiện 1 : Nguyễn Minh Hiếu

MSSV 1 : [Điền MSSV của bạn]

Chuyên ngành : Khoa học dữ liệu

Khóa : 2020

Chương trình đào tạo : Chương trình đào tạo Chuẩn

Hà Nội - 2025

“Cái tôi và sự hiểu biết tỷ lệ nghịch với nhau. Hiểu biết càng nhiều cái tôi càng bé. Hiểu biết càng ít, cái tôi càng to. ”

Albert Einstein

Acknowledgements

Lời cảm ơn chân thành đến thầy/cô hướng dẫn, gia đình, bạn bè và những người đã hỗ trợ tôi trong quá trình thực hiện báo cáo này. Đặc biệt, tôi xin bày tỏ lòng biết ơn sâu sắc đến [Tên Giảng viên Hướng dẫn] đã tận tình chỉ bảo và truyền đạt kiến thức quý báu, giúp tôi hoàn thành nghiên cứu này.

Contents

Acknowledgements	ii
List of Figures	iv
List of Tables	v
List of Abbreviations	vi
Physical Constants	vii
List of Symbols	viii
MỞ ĐẦU	1
1 TỔNG QUAN VỀ HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ VÀ ĐỘ ẨM	2
1.1 Bối cảnh và sự cần thiết của giám sát môi trường trong nhà	2
1.2 Mục tiêu của hệ thống	2
1.3 Các thành phần chính của hệ thống	3
1.3.1 Phần cứng	3
1.3.2 Phần mềm	4
A LIỆT KÊ SOURCE CODE	6
A.1 Mã nguồn Firmware ESP32	6
A.2 Mã nguồn Backend (PHP)	10
A.2.1 Các file chính trong TH_DO_AN/	10
A.2.2 Các file API trong TH_DO_AN/api/	18
A.2.3 Các file API người dùng trong Login/api/users/	31
A.3 Mã nguồn Frontend (HTML/CSS/JavaScript)	35
A.3.1 Các file chính trong TH_DO_AN/	36
A.3.2 Các file trong Login/	75

List of Figures

List of Tables

List of Abbreviations

IoT	Internet of Things
MCU	Microcontroller Unit
API	Application Programming Interface
PM2.5	Particulate Matter 2.5
AQI	Air Quality Index
DHT	Digital Humidity and Temperature
NTP	Network Time Protocol
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
PHP	PHP: Hypertext Preprocessor
MySQL	MyStructured Query Language
CSS	Cascading Style Sheets
JS	JavaScript
XAMPP	X-platform Apache MySQL PHP Perl

Physical Constants

Điện áp hoạt động	$V_{op} = 5 \text{ V}$
Dải nhiệt độ DHT11	$T_{range} = 0 \text{ đến } 50 \text{ }^{\circ}\text{C}$
Dải độ ẩm DHT11	$H_{range} = 20\% \text{ đến } 90\%$
Tần suất cập nhật mặc định	$f_{update} = 30 \text{ giây}$

List of Symbols

T	Nhiệt độ	°C
H	Độ ẩm	%
V_{in}	Điện áp đầu vào	V
I_{out}	Dòng điện đầu ra	A
t_{stamp}	Thời gian cập nhật	giây

MỞ ĐẦU

Phần "MỞ ĐẦU" của báo cáo này trình bày tổng quan về hệ thống giám sát nhiệt độ và độ ẩm thông minh, một giải pháp được phát triển nhằm theo dõi chất lượng không khí trong môi trường nhà ở và văn phòng. Hệ thống này được thiết kế với mục tiêu cung cấp dữ liệu môi trường theo thời gian thực, hỗ trợ người dùng theo dõi lịch sử dữ liệu, và đưa ra cảnh báo kịp thời khi các thông số vượt ngưỡng cho phép.

Hệ thống nổi bật với khả năng triển khai đơn giản, chi phí thấp, và giao diện web thân thiện, dễ sử dụng, phù hợp cho các ứng dụng giám sát chất lượng không khí trong nhà. Báo cáo này sẽ đi sâu vào các khía cạnh từ bối cảnh và sự cần thiết, mục tiêu, đến thiết kế phần cứng và phần mềm, quy trình triển khai, cũng như kết quả hoạt động, các thách thức đã gặp phải và hướng phát triển trong tương lai.

Bố cục của báo cáo được tổ chức như sau:

- **Chương 1: Tổng quan về hệ thống giám sát nhiệt độ và độ ẩm** trình bày bối cảnh, sự cần thiết và các thành phần chính của hệ thống.
- **Chương 2: Thiết kế và triển khai hệ thống** mô tả chi tiết kiến trúc phần cứng và phần mềm, cùng với quy trình xây dựng và lắp đặt hệ thống.
- **Chương 3: Kết quả và đánh giá hệ thống** trình bày các dữ liệu thu thập được, phân tích hiệu quả hoạt động và các tính năng cảnh báo, điều khiển.
- **Chương 4: Kết luận và định hướng phát triển** tóm tắt những thành quả chính và đề xuất các hướng cải tiến, mở rộng hệ thống trong tương lai.
- **Phụ lục** bao gồm mã nguồn chi tiết của các module hệ thống.

Chapter 1 TỔNG QUAN VỀ HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ VÀ ĐỘ ẨM

1.1 Bối cảnh và sự cần thiết của giám sát môi trường trong nhà

Chất lượng không khí trong nhà đóng vai trò vô cùng quan trọng đối với sức khỏe và năng suất làm việc của con người. Con người dành phần lớn thời gian trong nhà, nơi chất lượng không khí có thể bị ảnh hưởng bởi nhiều yếu tố như nhiệt độ, độ ẩm, và các chất ô nhiễm. Nhiệt độ và độ ẩm không phù hợp không chỉ gây khó chịu mà còn tạo điều kiện cho vi khuẩn, nấm mốc phát triển, ảnh hưởng tiêu cực đến hệ hô hấp và sức khỏe tổng thể. Do đó, việc giám sát liên tục và chính xác các thông số này là cần thiết để duy trì một môi trường sống và làm việc lành mạnh. Hệ thống giám sát tự động, đặc biệt là ứng dụng công nghệ IoT, mang đến giải pháp hiệu quả để thu thập dữ liệu, phân tích xu hướng và kịp thời đưa ra cảnh báo khi các chỉ số vượt ngưỡng an toàn.

1.2 Mục tiêu của hệ thống

Mục tiêu tổng quát của dự án là thiết kế và triển khai một hệ thống giám sát nhiệt độ và độ ẩm trong nhà một cách hiệu quả, dễ sử dụng và chi phí thấp, góp phần nâng cao chất lượng môi trường sống và làm việc.

Các mục tiêu cụ thể bao gồm:

- Thu thập dữ liệu nhiệt độ và độ ẩm theo thời gian thực từ cảm biến **DHT11**.
- Xây dựng một giao diện web trực quan, cho phép người dùng dễ dàng theo dõi dữ liệu hiện tại, xem biểu đồ lịch sử và quản lý thiết bị.
- Cho phép người dùng thiết lập các ngưỡng cảnh báo cho nhiệt độ và độ ẩm, và hiển thị trạng thái cảnh báo trực tiếp trên giao diện.

- Cung cấp khả năng điều khiển bật/tắt chức năng đo của cảm biến thông qua giao diện web.
- Hỗ trợ xuất dữ liệu lịch sử ra định dạng CSV để tiện cho việc phân tích chuyên sâu.

1.3 Các thành phần chính của hệ thống

Hệ thống giám sát nhiệt độ và độ ẩm thông minh bao gồm các thành phần chính được phân loại thành phần cứng và phần mềm, cùng với luồng dữ liệu tương tác giữa chúng.

1.3.1 Phần cứng

- **Cảm biến DHT11:** Là loại cảm biến chính được sử dụng để đo nhiệt độ và độ ẩm môi trường. Cảm biến này có ưu điểm về chi phí thấp và dễ tích hợp.
- **Bộ vi điều khiển ESP32:** Đóng vai trò là bộ não của hệ thống, ESP32 chịu trách nhiệm đọc dữ liệu từ cảm biến **DHT11**, xử lý sơ bộ và truyền dữ liệu lên máy chủ thông qua kết nối Wi-Fi. ESP32 cũng điều khiển các **LED** thông báo trạng thái hoạt động.
- **Module Wi-Fi:** Tích hợp sẵn trong ESP32, được sử dụng để thiết lập kết nối không dây với mạng cục bộ, cho phép truyền dữ liệu đến máy chủ **localhost**.
- **LED thông báo:**
 - **LED hoạt động (màu xanh lá):** Sáng liên tục khi cảm biến đang đo và gửi dữ liệu.
 - **LED cảnh báo (màu đỏ):** Sẽ bật sáng khi nhiệt độ hoặc độ ẩm vượt quá ngưỡng đã thiết lập.
- **Nguồn điện Adapter 5V:** Cung cấp năng lượng cho toàn bộ hệ thống phần cứng hoạt động.

1.3.2 Phần mềm

- **Firmware (Nhúng trên ESP32):**

- Ngôn ngữ lập trình: C/C++ sử dụng Arduino IDE.
- Chức năng: Đọc dữ liệu từ **DHT11**, đồng bộ thời gian thông qua **NTPClient** để gán **timestamp** chính xác cho dữ liệu, gửi yêu cầu **HTTP GET** đến API điều khiển, tạo gói dữ liệu **JSON** và gửi qua **HTTP POST** đến máy chủ. Firmware cũng điều khiển các **LED** thông báo trạng thái.

- **Hệ thống Backend (Máy chủ cục bộ):**

- Nền tảng: XAMPP (bao gồm Apache làm web server và PHP làm ngôn ngữ xử lý phía máy chủ).
- Cơ sở dữ liệu: MySQL được sử dụng để lưu trữ dữ liệu nhiệt độ, độ ẩm, **timestamp** và **device_id** trong bảng `sensor_readings` và `device_settings`.
- Các **API** chính:
 - * `post_data.php`: Nhận dữ liệu **JSON** từ ESP32 và lưu vào MySQL.
 - * `get_dht_status.php`, `toggle_dht_status.php`: Quản lý trạng thái bật/tắt cảm biến.
 - * `save_alert_thresholds.php`, `get_alert_thresholds.php`: Quản lý các ngưỡng cảnh báo.
 - * `get_devices.php`: Cung cấp danh sách các thiết bị.
 - * `get_current_data.php`, `get_history_data.php`: Cung cấp dữ liệu hiện tại và lịch sử.
 - * `export_data_csv.php`: Xuất dữ liệu lịch sử ra CSV.

- **Hệ thống Frontend (Giao diện người dùng web):**

- Công nghệ: HTML, CSS và JavaScript.

- Chức năng: Hiển thị dữ liệu nhiệt độ và độ ẩm theo thời gian thực dưới dạng thẻ lớn, biểu đồ lịch sử tương tác (sử dụng Chart.js và Luxon), trang cài đặt ngưỡng cảnh báo, và khả năng điều khiển bật/tắt cảm biến. Giao diện cũng hiển thị thông tin kỹ thuật về cảm biến **DHT11**.
- Cảnh báo: Kích hoạt hiệu ứng nhấp nháy đỏ trên giao diện khi dữ liệu vượt ngưỡng.

Appendix A LIỆT KÊ SOURCE CODE

A.1 Mã nguồn Firmware ESP32

Mã nguồn **firmware** được nạp vào ESP32 để thực hiện các chức năng thu thập dữ liệu, điều khiển **LED** và truyền thông.

```
1 #include <WiFi.h>           // ưTh ệvin Wi-Fi cho ESP32
2 #include <HTTPClient.h>     // ưTh ệvin HTTP client cho ESP32
3 #include "DHT.h"           // ưTh ệvin DHT sensor
4 #include <Adafruit_Sensor.h> // ưTh ệvin Adafruit Unified Sensor (dependency ủa DHT)
5 #include <WiFiUdp.h>        // ưTh ệvin UDP cho NTP client
6 #include <NTPClient.h>      // ưTh ệvin NTP client
7 #include <ArduinoJson.h>    // ưTh ệvin ArduinoJson để ạo JSON payload
8
9 // --- ẮC HẸN ẠMNG VÀ SERVER ỦA ẠBN ---
10 const char* ssid = "Tang 4"; // Tên ạmng Wi-Fi ủa ạbn
11 const char* password = "88888888"; // ậMt ắkhủ Wi-Fi ủa ạbn
12
13 // Thay ếth ầbng ĐỊA ỈCH IP ỦA MÁY TÍNH ẠCHY XAMPP ỦA ẠBN
14 const char* serverAddress = "192.168.4.106";
15
16 // Đường ắdn đến file PHP để POST ữd ệliu ắcm ếbin
17 const char* postDataApiUrl = "/TH_DO_AN/post_data.php";
18 // Đường ắdn đến API để ắly ạtrng thắi ậbt/ắtt DHT ừt web
19 const char* getDhtStatusApiUrl = "http://192.168.4.106/TH_DO_AN/api/get_dht_status.php";
20
21 // --- ẮC HẸN ẮCM ẾBIN VÀ LED ---
22 #define DHTPIN 4           // Chân GPIO4 ủa ESP32 ếkt ốni ớvi chân Data ủa DHT11 (ươtng ứng D4)
23 #define DHTTYPE DHT11     // ạLoi ắcm ếbin DHT
24
25 #define LED_PIN 15        // Chân GPIO15 ủa ESP32 ếkt ốni ớvi đèn LED thông báo (ươtng ứng D15)
26
27 String deviceId = "ESP32_Sensor_02"; // ID duy ắnhտ cho ắcm ếbin này
28
29 // --- ỜTHI GIAN VÀ ẾBIN ẠTNG THẮI ---
30 const long postingInterval = 1000; // ầTn ắsut ừgi ữd ệliu ắcm ếbin (1 giắy)
31 unsigned long lastDataPostTime = 0; // ờThi đểim ầln ốcuì ừgi ữd ệliu
32
33 const long STATUS_CHECK_INTERVAL = 1000; // ầTn ắsut ểkim tra ạtrng thắi ậbt/ắtt DHT ừt server (1 giắy)
34 unsigned long lastStatusCheckTime = 0; // ờThi đểim ầln ốcuì ểkim tra ạtrng thắi
35 bool dht_enabled = true; // ếBin ứlu ạtrng thắi ậbt/ắtt ủa DHT (ắmc ặnh là ậbt)
36
37 // --- ĐỐI ỰỢTNG ---
38 DHT dht(DHTPIN, DHTTYPE); // ởKhi ạo đối ựợtng DHT
39
40 WiFiUDP ntpUDP;
41 // ởKhi ạo NTPClient (server: vn.pool.ntp.org, múi ờgi GMT+7, ắkhong ờthi gian ắcp ắnhտ 5 phút)
42 NTPClient timeClient(ntpUDP, "vn.pool.ntp.org", 7 * 3600, 300000);
43
```



```

44 // --- SETUP ---
45 void setup() {
46     Serial.begin(115200); // ở Khi động Serial Monitor để debug
47     dht.begin();         // ở Khi động cảm ến bin DHT
48
49     pinMode(LED_PIN, OUTPUT); // ế Thít ập chân LED là OUTPUT
50     digitalWrite(LED_PIN, LOW); // Đám ả bo LED ắt khi ở khi động
51
52     Serial.print("Connecting to WiFi ");
53     Serial.println(ssid);
54
55     WiFi.begin(ssid, password); // ắ Bt đầu ế kt ó ni Wi-Fi
56
57     int attempts = 0;
58     while (WiFi.status() != WL_CONNECTED && attempts < 20) {
59         delay(500);
60         Serial.print(".");
61         attempts++;
62     }
63
64     if (WiFi.status() == WL_CONNECTED) {
65         Serial.println("\nWiFi connected.");
66         Serial.print("Local IP: ");
67         Serial.println(WiFi.localIP());
68         timeClient.begin(); // ở Khi động NTP client sau khi có WiFi
69         // ắ Ly ạt ẳ ng thá i DHT ban đầu ngay sau khi ế kt ó ni WiFi thành công
70         getDhtStatus();
71     } else {
72         Serial.println("\nWiFi connection failed. Please check SSID and password.");
73     }
74 }
75
76 // --- HÀM Ắ LY TIMESTAMP ---
77 String getTimeStamp() {
78     // ậ Cp ậ nh t ở thi gian NTP
79     timeClient.update();
80
81     // ế Nu ư ch ả có ở thi gian NTP, ử th ậ li vài ầ n
82     if (!timeClient.isTimeSet()) {
83         Serial.println("NTP time not yet set, attempting to sync...");
84         for(int i = 0; i < 5 && !timeClient.isTimeSet(); ++i) {
85             delay(1000);
86             timeClient.update();
87         }
88         if (!timeClient.isTimeSet()) {
89             Serial.println("Still failed to get NTP time. Using fallback timestamp (may
be inaccurate).");
90             return ""; // ắ Tr ề v ỗ ẳ ng để ỗ b qua ệ vic ử gi ử d ệ liu
91         }
92     }
93
94     time_t rawtime = timeClient.getEpochTime();
95     struct tm * ti;
96     ti = localtime(&rawtime); // ế Chuy ển đ ổi sang ở gi địa ư ph ẳ ng
97     char buffer[80];
98     // Định ặ đ ẳ g ở thi gian theo ắ chun MySQL TIMESTAMP: YYYY-MM-DD HH:MM:SS
99     strftime(buffer, sizeof(buffer), "%Y-%m-%d %H:%M:%S", ti);
100     return String(buffer);
101 }
102
103 // --- HÀM Ắ LY Ặ ẳ NG THÁ I DHT Ừ T SERVER ---
104 void getDhtStatus() {
105     if (WiFi.status() == WL_CONNECTED) {
106         HTTPClient http;
107         http.begin(getDhtStatusApiUrl); // ế Kt ó ni đến API ắ ly ặ ẳ ng thá i

```

```

108     int httpCode = http.GET(); // ửGi GET request
109
110     if (httpCode > 0) {
111         String payload = http.getString();
112         Serial.print("DHT Status API Response: ");
113         Serial.println(payload);
114
115         payload.trim(); // ấCt ấkhong ấtrng ừt ồchui payload ẬI ỒCH
116
117         // API ấtr ềv "true" ấhoc "false"
118         if (payload.equals("true")) { // So sánh ựtrc ếtíp trên payload đã được ấct
119             dht_enabled = true;
120             Serial.println("DHT is ENABLED.");
121         } else {
122             dht_enabled = false;
123             Serial.println("DHT is DISABLED.");
124         }
125     } else {
126         Serial.printf("[HTTP GET] getDhtStatus failed , error: %s\n", http.
errorToString(httpCode).c_str());
127         Serial.println("Retaining previous DHT state due to API error.");
128     }
129     http.end(); // Đóng ết ốni
130 }
131 }
132
133 // --- LOOP ---
134 void loop() {
135     if (WiFi.status() == WL_CONNECTED) {
136         // Luôn ồc ấgng ấcp ậnh ờthi gian NTP trong loop
137         timeClient.update();
138
139         // ểKim tra ấtrng thái ậbt/ấtt DHT ừt server định ỳk
140         if (millis() - lastStatusCheckTime > STATUS_CHECK_INTERVAL) {
141             getDhtStatus(); // ọGi hàm để ấcp ậnh ấtrng thái dht_enabled
142             lastStatusCheckTime = millis();
143         }
144
145         // --- LOGIC ĐIỀU ỂKHIN LED VÀ ỬGI ỮD ỆLIU ---
146         if (dht_enabled) { // ếNu DHT được ẬBT
147             digitalWrite(LED_PIN, HIGH); // ẬBT LED LIÊN ỰC KHI ẾCH ĐỘ ĐO ĐANG ẬHOT
ĐỘNG
148
149             if (millis() - lastDataPostTime > postingInterval) { // Và đã đến ờthi gian ửgi
150                 lastDataPostTime = millis();
151
152                 // KHÔNG ẦCN digitalWrite(LED_PIN, HIGH) hay delay Ở ĐÂY ỪNA, LED ĐẢ SÁNG
LIÊN ỰC
153
154                 // Đọc ệnhit độ và độ ẩm
155                 float h = dht.readHumidity();
156                 float t = dht.readTemperature();
157
158                 // KHÔNG ẦCN digitalWrite(LED_PIN, LOW) Ở ĐÂY ỪNA, LED ỀS ẮTT KHI
dht_enabled = false
159
160                 // ểKim tra xem ệvic đọc có thành công không
161                 if (isnan(h) || isnan(t)) {
162                     Serial.println("Failed to read from DHT sensor!");
163                     // ếNu đọc ồli, LED ấvn ềs sáng liên ực (ếnu dht_enabled ấvn true)
164                     // ấhoc ấtt ếnu dht_enabled ểchuyyn false
165                     return; // Thoát ồkhi vòng ặlp ếnu đọc ồli
166                 }
167

```

```

168 Serial.print("Humidity: ");
169 Serial.print(h);
170 Serial.print(" %\t");
171 Serial.print("Temperature: ");
172 Serial.print(t);
173 Serial.println(" *C");
174
175 String timestamp = getTimeStamp(); // lấy timestamp
176 if (timestamp != "") { // chỉ gửi dữ liệu nếu có timestamp hợp lệ
177     String serverPath = "http://" + String(serverAddress) + String(
postDataApiUrl);
178
179     HTTPClient http;
180     http.begin(serverPath);
181     http.addHeader("Content-Type", "application/json");
182
183     StaticJsonDocument<256> jsonDocument;
184     jsonDocument["device_id"] = deviceId;
185     jsonDocument["temperature"] = t;
186     jsonDocument["humidity"] = h;
187     jsonDocument["timestamp"] = timestamp;
188
189     String httpRequestData;
190     serializeJson(jsonDocument, httpRequestData);
191
192     Serial.print("Sending POST request to: ");
193     Serial.println(serverPath);
194     Serial.print("Payload: ");
195     Serial.println(httpRequestData);
196
197     int httpResponseCode = http.POST(httpRequestData);
198
199     if (httpResponseCode > 0) {
200         Serial.print("HTTP Response code: ");
201         Serial.println(httpResponseCode);
202         String response = http.getString();
203         Serial.println(response);
204     } else {
205         Serial.print("Error code: ");
206         Serial.println(httpResponseCode);
207         Serial.print("HTTP Error: ");
208         Serial.println(http.errorToString(httpResponseCode).c_str());
209     }
210     http.end();
211 } else {
212     Serial.println("Skipping data send due to invalid timestamp.");
213 }
214 } // kết thúc if (millis() - lastDataPostTime > postingInterval)
215 // nếu dht_enabled là true thì gửi dữ liệu đến lúc gửi, LED vẫn sáng
216 } else { // nếu DHT bị ngắt (dht_enabled là false)
217     Serial.println("DHT sensor is currently disabled. Not reading/sending data.");
218     digitalWrite(LED_PIN, LOW); // ĐẢM BẢO LED NGẮT KHI DHT KHÔNG HOẠT ĐỘNG
219     delay(1000); // Thêm một delay ngắn để tránh spam Serial Monitor
220 }
221 } else { // nếu Wi-Fi không kết nối
222     Serial.println("WiFi not connected. Reconnecting...");
223     WiFi.begin(ssid, password);
224     delay(5000); // Đợi 5 giây trước khi thử lại
225 }
226 }

```

A.2 Mã nguồn Backend (PHP)

Các file PHP xử lý các yêu cầu từ ESP32 và giao diện web, quản lý cơ sở dữ liệu MySQL.

A.2.1 Các file chính trong TH_DO_AN/

```
1 <?php
2 // Bao gồm file cấu hình chung của ứng dụng (bao gồm cài đặt ôli và CORS)
3 require_once 'app_config.php';
4
5 // Đặt Content-Type cho phản hồi này là JSON.
6 // Đảm bảo dòng này nằm sau require_once 'app_config.php' để tránh ôli "Headers
  already sent"
7 header("Content-Type: application/json");
8
9 // Bao gồm file kết nối cơ sở dữ liệu
10 // Đảm bảo file 'db_connect.php' tồn tại trong cùng thư mục (TH_DO_AN/)
11 // và thiết lập biến $conn (kết nối mysqli)
12 require_once 'db_connect.php';
13
14 // Kiểm tra xem biến $conn đã được thiết lập từ db_connect.php chưa
15 // và kiểm tra kết nối có thành công không
16 if (!isset($conn) || $conn->connect_error) {
17     error_log("Database connection error in post_data.php: " . ($conn->connect_error
18     ?? 'Connection object not set.));
19     echo json_encode(['status' => 'error', 'message' => 'ôli kết nối CSDL: ' . ($conn
20     ->connect_error ?? 'Không thể kết nối.')] );
21     exit();
22 }
23
24 // Thiết lập charset cho kết nối để hỗ trợ tiếng Việt
25 // Dòng này đã có trong db_connect.php, nên có thể bỏ qua ở đây để tránh dư thừa.
26 // $conn->set_charset("utf8mb4");
27
28 // Nhận dữ liệu POST thô (raw POST data)
29 $json = file_get_contents('php://input');
30 $data = json_decode($json);
31
32 // Kiểm tra xem dữ liệu JSON có hợp lệ không
33 if ($data === null && json_last_error() !== JSON_ERROR_NONE) {
34     error_log("Invalid JSON received in post_data.php: " . json_last_error_msg() . "
35     . Raw data: " . $json . "");
36     echo json_encode(["status" => "error", "message" => "Dữ liệu JSON không hợp lệ."
37     ]);
38     $conn->close();
39     exit();
40 }
41
42 // Lấy dữ liệu từ JSON payload
43 // Sử dụng null coalescing operator (??) để tránh ôli nếu trường không tồn tại
44 $device_id = $data->device_id ?? '';
45 $temperature = $data->temperature ?? '';
46 $humidity = $data->humidity ?? '';
47 $timestamp = $data->timestamp ?? null; // Lấy timestamp từ ESP, có thể là null
48
49 // Kiểm tra các trường dữ liệu cần thiết
50 if (empty($device_id) || empty($temperature) || empty($humidity)) {
```

```

47     error_log("Missing required data in post_data.php. Device ID: '$device_id', Temp
: '$temperature', Humidity: '$humidity'");
48     echo json_encode(["status" => "error", "message" => "ẾThiu ữd ệliu ấbt ộbuc:
device_id, temperature, ặhoc humidity."]);
49     $conn->close();
50     exit();
51 }
52
53 // ểChun ỏi ệnhit độ và độ ẩ sang ểkiu ós ựthc
54 $temperature = floatval($temperature);
55 $humidity = floatval($humidity);
56
57 // ểNu timestamp không được cung ấp ặhoc ỗrng, ửs ụngng ờthi gian ệhin ậti ủa máy ủch
58 if (empty($timestamp)) {
59     $timestamp = date('Y-m-d H:i:s');
60 }
61
62 // ẩChun ậ câu ệlnh SQL để chèn ữd ệliu
63 // ửS ụngng Prepared Statements để tránh SQL Injection
64 $stmt = $conn->prepare("INSERT INTO sensor_readings (device_id, temperature,
humidity, reading_time) VALUES (?, ?, ?, ?)");
65 if ($stmt === false) {
66     error_log("Prepare failed in post_data.php: " . $conn->error);
67     echo json_encode(["status" => "error", "message" => "ỗLi ệh ốthng. Vui lòng ửth
ặli sau."]);
68     $conn->close();
69     exit();
70 }
71
72 // Bind các tham số và ựthc thi câu ệlnh
73 // "sdds" ỉngha là: s=string (device_id), d=double (temperature), d=double (humidity)
, s=string (reading_time)
74 $stmt->bind_param("sdds", $device_id, $temperature, $humidity, $timestamp);
75
76 if ($stmt->execute()) {
77     echo json_encode(["status" => "success", "message" => "ữD ệliu ẩcm ếbin đã được
ứlu thành công."]);
78 } else {
79     error_log("Execute failed in post_data.php: " . $stmt->error);
80     echo json_encode(["status" => "error", "message" => "ỗLi khi ứlu ữd ệliu ẩcm ếbin
."]);
81 }
82
83 $stmt->close();
84 $conn->close();
85 ?>

```

```

1 <?php
2 // Bao ồgm file ấcu hình chung ủa ứng ụngng (bao ồgm cài đặt ỗli và CORS)
3 require_once 'app_config.php'; // Đẩm ẩbo đường ẩđn này đúng óvi ịv trí ủa
app_config.php
4
5 // Đặt Content-Type cho ẩphn ồhi này là JSON
6 // Đẩm ẩbo dòng này ẩm sau require_once 'app_config.php' để tránh "Headers already
sent"
7 header("Content-Type: application/json");
8
9 // Bao ồgm file ểkt ốni ợc ós ữd ệliu
10 require_once 'db_connect.php';
11
12 // ểKim tra xem ếbin $conn đã được ếthit ẩlp ừt db_connect.php ựcha
13 // và ểkim tra ểkt ốni có thành công không
14 if (!isset($conn) || $conn->connect_error) {
15     error_log("Database connection error in get_current_data.php: " . ($conn->
connect_error ?? 'Connection object not set.));

```

```

16     echo json_encode([ 'status' => 'error', 'message' => 'ỖLi ết ốni CSDL: ' . ($conn
17     ->connect_error ?? 'Không ết ết ốni.'))];
18     exit();
19 }
20 // ấLy device_id ừt tham ố GET, tên ấphi ớkhp ớvi main.js
21 $device_id_filter = isset($_GET['device_id']) ? $_GET['device_id'] : null;
22
23 // Xây ựng truy ấvn SQL
24 // ấLy ệnhit độ, độ ẩm và ờthi gian đọc ừt ảbng sensor_readings
25 $sql = "SELECT temperature, humidity, reading_time FROM sensor_readings ";
26
27 // ếNu có device_id_filter, thêm điều ệkin WHERE
28 if ($device_id_filter) {
29     $sql .= "WHERE device_id = ? ";
30 }
31
32 // ấSp ếp theo ID ảgim ảdn ể( ấly ảbn ghi ớmi ấnh) và ớgii ảhn 1 ảbn ghi
33 $sql .= "ORDER BY id DESC LIMIT 1";
34
35 // ấChun ịb câu ệlnh SQL
36 $stmt = $conn->prepare($sql);
37
38 if ($stmt === false) {
39     // Ghi log ỗli thay vì ểhin ịth ựtrc ếtip trong môi ườtrng ảsn ấphm
40     error_log("Prepare failed in get_current_data.php: " . $conn->error);
41     echo json_encode([ "status" => "error", "message" => "ỖLi ệh ốthng. Vui lòng ừth
42     ảli sau."]);
43     $conn->close();
44     exit();
45 }
46
47 // ếNu có device_id_filter, ấgn tham ố vào câu ệlnh
48 if ($device_id_filter) {
49     $stmt->bind_param("s", $device_id_filter); // "s" cho ểkiu string
50 }
51
52 // ựThc thi câu ệlnh
53 $stmt->execute();
54 $result = $stmt->get_result();
55
56 // ểKim tra và ấtr ềv ết ấqu
57 if ($result && $result->num_rows > 0) {
58     $row = $result->fetch_assoc();
59     echo json_encode([
60         "status" => "success",
61         // ấTr ềv ộmt đối ựtng 'reading' ể ớkhp ớvi main.js
62         "reading" => [
63             "temperature" => floatval($row["temperature"]),
64             "humidity" => floatval($row["humidity"]),
65             "reading_time" => $row["reading_time"] // Đả ảbo tên ườtrng ớkhp ớvi JS
66         ]
67     ]);
68     exit(); // Thêm exit() sau khi ừgi JSON
69 } else {
70     // ấTr ềv thông báo ỗli ếnu không tìm ấthy ừđ ệliu
71     echo json_encode([
72         "status" => "error",
73         "message" => "Không tìm ấthy ừđ ệliu." . ($device_id_filter ? " cho ếthit ịb "
74         . htmlspecialchars($device_id_filter) : "")
75     ]);
76     exit(); // Thêm exit() sau khi ừgi JSON
77 }
78
79 // Đóng câu ệlnh và ết ốni database (dòng này ế không được ựthc thi ếnu exit() đã
80 được ọgi ướtrc đó)

```

```

78 $stmt->close();
79 $conn->close();
80 ?>

1 <?php
2 // Bao gồm file cấu hình chung của ứng dụng (bao gồm cài đặt ôli và CORS)
3 require_once 'app_config.php';
4
5 session_start(); // Khởi đầu session để ứng dụng thông tin phiên nếu có logic xác thực
6
7 // Đặt Content-Type cho phản hồi này là JSON
8 header("Content-Type: application/json");
9
10 // Bao gồm file kết nối cơ sở dữ liệu
11 require_once 'db_connect.php';
12
13 // Kiểm tra xem biến $conn đã được thiết lập từ db_connect.php chưa
14 // và kiểm tra kết nối có thành công không.
15 if (!isset($conn) || $conn->connect_error) {
16     error_log("Database connection error in get_history_data.php: " . ($conn->
17         connect_error ?? 'Connection object not set.'));
18     echo json_encode(['status' => 'error', 'message' => 'Ôli kết nối CSDL: ' . ($conn
19         ->connect_error ?? 'Không thể kết nối.')] );
20     exit();
21 }
22
23 // Lấy deviceId từ request. Đặt mặc định là null.
24 $deviceId = isset($_GET['device_id']) ? $_GET['device_id'] : null;
25
26 // Nếu deviceId không được cung cấp, trả về ôli ngay lập tức
27 if (empty($deviceId)) {
28     echo json_encode(['status' => "error", "message" => "Thiếu ID thiết bị."]);
29     $conn->close(); // Đóng kết nối CSDL trước khi thoát
30     exit(); // Kết thúc script
31 }
32
33 // Lấy khoảng thời gian từ request.
34 $period = isset($_GET['period']) ? $_GET['period'] : '24h';
35 // Lấy tham số interval tối đa (n giờ phút) từ request. Null nếu không có.
36 $interval_minutes = isset($_GET['interval']) ? intval($_GET['interval']) : null;
37
38 $sql_condition = ""; // Điều kiện WHERE cho SQL query
39 $params = [$deviceId]; // Device ID luôn là tham số đầu tiên
40 $types = "s"; // Kiểu dữ liệu cho deviceId
41
42 // Các điều kiện cho SELECT và GROUP BY
43 $effectiveInterval = 60; // Mặc định 1 giờ (60 phút)
44 // Thay đổi cách ở khi tạo startDateTime và endDateTime ở đây
45 $startDateTime = null;
46 $endDateTime = null;
47
48 // Lấy thời gian hiện tại từ server (đã được đặt múi giờ)
49 $currentTime = new DateTime();
50
51 // Xử lý logic không thời gian và interval
52 // Xử lý logic không thời gian và interval
53 switch ($period) {
54     case '1h':
55         // Làm tròn thời gian hiện tại xuống phút gần nhất
56         $endDateTime = clone $currentTime; // Clone để không ảnh hưởng đến
57             $currentTime gốc
58         $endDateTime->setTime($endDateTime->format('H'), $endDateTime->format('i'),
59             0);
60         $startDateTime = (clone $endDateTime)->sub(new DateInterval('PT1H')); // Trước
61             1 giờ từ thời điểm đã làm tròn

```

```

58     $effectiveInterval = 1; // 1 phút
59     break;
60     case '6h':
61         $endDateTime = clone $currentTime;
62         $endDateTime->setTime($endDateTime->format('H'), (floor($endDateTime->format
('i') / 15) * 15), 0);
63         $startDateTime = (clone $endDateTime)->sub(new DateInterval('PT6H'));
64         $effectiveInterval = 15; // 15 phút
65         break;
66     case '12h':
67         $endDateTime = clone $currentTime;
68         $endDateTime->setTime($endDateTime->format('H'), (floor($endDateTime->format
('i') / 30) * 30), 0);
69         $startDateTime = (clone $endDateTime)->sub(new DateInterval('PT12H'));
70         $effectiveInterval = 30; // 30 phút
71         break;
72     case '24h':
73         // Làm tròn thời gian hiện tại xuống giờ gần nhất
74         $endDateTime = clone $currentTime;
75         $endDateTime->setTime($endDateTime->format('H'), 0, 0); // Làm tròn phút và
giây về 00
76         $startDateTime = (clone $endDateTime)->sub(new DateInterval('PT24H')); //
ừ Tr 24 giờ từ thời điểm đã làm tròn
77         $effectiveInterval = 60; // 1 giờ
78         break;
79     case '7d':
80         // Làm tròn thời gian hiện tại xuống 6 giờ gần nhất
81         $endDateTime = clone $currentTime;
82         $endDateTime->setTime(floor($endDateTime->format('H') / 6) * 6, 0, 0);
83         $startDateTime = (clone $endDateTime)->sub(new DateInterval('P7D')); // ừ Tr
7 ngày từ thời điểm đã làm tròn
84         $effectiveInterval = 6 * 60; // 6 giờ
85         break;
86     case '30d':
87         // Làm tròn thời gian hiện tại xuống đầu ngày gần nhất
88         $endDateTime = clone $currentTime;
89         $endDateTime->setTime(0, 0, 0);
90         $startDateTime = (clone $endDateTime)->sub(new DateInterval('P30D')); // ừ Tr
30 ngày từ thời điểm đã làm tròn
91         $effectiveInterval = 24 * 60; // 1 ngày
92         break;
93     case 'custom':
94         $startDate = isset($_GET['startDate']) ? $_GET['startDate'] : null;
95         $endDate = isset($_GET['endDate']) ? $_GET['endDate'] : null;
96
97         if ($startDate && $endDate) {
98             try {
99                 $startDateTime = new DateTime($startDate . " 00:00:00");
100                 $endDateTime = new DateTime($endDate . " 23:59:59");
101
102                 if ($startDateTime > $endDateTime) {
103                     throw new Exception("Ngày bắt đầu không ểh sau ngày ết thúc.");
104                 }
105
106                 $intervalDiff = $startDateTime->diff($endDateTime);
107                 $totalMinutes = $intervalDiff->days * 24 * 60 + $intervalDiff->h *
60 + $intervalDiff->i;
108
109                 if ($totalMinutes <= 60) {
110                     $effectiveInterval = 1;
111                 } elseif ($totalMinutes <= 6 * 60) {
112                     $effectiveInterval = 15;
113                 } elseif ($totalMinutes <= 24 * 60) {
114                     $effectiveInterval = 30;
115                 } elseif ($totalMinutes <= 7 * 24 * 60) {
116                     $effectiveInterval = 60;

```



```

117         } elseif ($totalMinutes <= 30 * 24 * 60) {
118             $effectiveInterval = 6 * 60;
119         } else {
120             $effectiveInterval = 24 * 60;
121         }
122
123         if ($interval_minutes !== null && $interval_minutes > 0) {
124             $effectiveInterval = $interval_minutes;
125         }
126
127         if ($effectiveInterval <= 0) $effectiveInterval = 1;
128
129         } catch (Exception $e) {
130             error_log("Invalid date format for custom period in get_history_data
.php: " . $e->getMessage());
131             echo json_encode(["status" => "error", "message" => "Định ạng ngày
không ợp ệl: " . $e->getMessage()]);
132             $conn->close();
133             exit();
134         }
135
136     } else {
137         echo json_encode(["status" => "error", "message" => "ăKhong ờthi gian tùy
ichnh yêu ầu ngày ắbt đầu và ếkt thúc."]);
138         $conn->close();
139         exit();
140     }
141     break;
142     default: // ườTrng ợhp period không ợhp ệl, ắmc định 24h
143         $sendDateTime = clone $currentTime;
144         $sendDateTime->setTime($sendDateTime->format('H'), 0, 0);
145         $startDateTime = (clone $sendDateTime)->sub(new DateInterval('PT24H'));
146         $effectiveInterval = 60;
147         break;
148     }
149
150     // ắChc ắchn $startDateTime và $sendDateTime đã được định ĩngha
151     if ($startDateTime === null || $sendDateTime === null) {
152         echo json_encode(["status" => "error", "message" => "ỗLi xác định ăkhong ờthi
gian."]);
153         $conn->close();
154         exit();
155     }
156
157     $startSQL = $startDateTime->format('Y-m-d H:i:s');
158     $endSQL = $sendDateTime->format('Y-m-d H:i:s');
159
160     // ỬSA ĐỐI: Tính toán totalSecondsInPeriod chính xác
161     $intervalDiff = $startDateTime->diff($endDateTime);
162     $totalSecondsInPeriod = $intervalDiff->days * 24 * 3600 + $intervalDiff->h * 3600 +
        $intervalDiff->i * 60 + $intervalDiff->s;
163
164     // ỬSA ĐỐI: Tính toán max_intervals chính xác ớhn và dùng cho LIMIT
165     $secondsPerInterval = $effectiveInterval * 60;
166     if ($secondsPerInterval == 0) $secondsPerInterval = 1; // Tránh chia cho 0
167     $num_intervals_to_generate = (int)ceil($totalSecondsInPeriod / $secondsPerInterval);
168
169     // Dùng ợm ốt ớ hàng ớln để ạo ốt (ví ụđ: 1000 hàng ừt INFORMATION_SCHEMA.COLUMNS)
170     // ắhoc ặbn có ếth ạo ợm ắbng numbers ớnh trong DB ếnu ắcn ềnhu ớhn
171     $limit_numbers_series = $num_intervals_to_generate + 5; // ợCng thêm 5 làm ăkhong ặđn
172     if ($limit_numbers_series < 60 && $effectiveInterval == 1) { // ếNu là 1h, ắcn ít
        ắnh 60 ếđm
173         $limit_numbers_series = 65; // 60 phút + ặđn
174     } elseif ($limit_numbers_series < 24 && $effectiveInterval == 60) { // ếNu là 24h,
        ắcn ít ắnh 24 ếđm

```

```

175     $limit_numbers_series = 29; // 24 ờgi + đên
176 }
177 // óGii ảhn không quá óln để tránh ấvn đề ệhiu ấsut óvi INFORMATION_SCHEMA
178 if ($limit_numbers_series > 2000) $limit_numbers_series = 2000;
179
180 // == ẮBT ĐẦU ẦPHN SQL ÓMI ĐỂ ẠTO ỒCHUI ỜTHI GIAN VÀ LEFT JOIN (ừS ụng
181 $sql = "
182 SELECT
183     -- Làm tròn timestamp để đảm bảo ókhp óvi các ómc nhóm
184     DATE_FORMAT(FROM_UNIXTIME(FLOOR(UNIX_TIMESTAMP(time_series.interval_start) / ({
185         $effectiveInterval} * 60)) * ({ $effectiveInterval} * 60)), '%Y-%m-%d %H:%i:%s')
186     AS timestamp,
187     AVG(sr.temperature) AS temperature,
188     AVG(sr.humidity) AS humidity
189 FROM (
190     SELECT
191         -- ấLy ờthi gian ắbt đầu làm tròn óxung ómc interval ầgn ấnh
192         FROM_UNIXTIME(FLOOR(UNIX_TIMESTAMP(?) / ({ $effectiveInterval} * 60)) * ({
193             $effectiveInterval} * 60))
194         -- ộCng thêm ós phút ủa ừtng interval (n.n là cách tham ếchiu ộct 'n' ừt
195         ảbng ảo 'n')
196         + INTERVAL (n.n * { $effectiveInterval}) MINUTE AS interval_start
197     FROM
198         (SELECT @n := -1) AS init_n, -- ởKhi ạo ếbin session @n
199         (SELECT @n := @n + 1 AS n FROM INFORMATION_SCHEMA.COLUMNS LIMIT {
200             $limit_numbers_series}) AS n
201     -- INFORMATION_SCHEMA.COLUMNS là ộmt ảbng ệh óthng óln để ạo ós hàng
202     -- ặHoc dùng ảbng ós ủa riêng ảbn ềnu có
203 ) AS time_series
204 LEFT JOIN sensor_readings sr ON
205     sr.device_id = ? AND
206     FROM_UNIXTIME(FLOOR(UNIX_TIMESTAMP(sr.reading_time) / ({ $effectiveInterval} *
207         60)) * ({ $effectiveInterval} * 60)) = time_series.interval_start
208 WHERE
209     -- Đảm bảo ờthi gian ạo ra ằm trong ảkhong truy ấvn ban đầu
210     time_series.interval_start BETWEEN ? AND ?
211 GROUP BY
212     timestamp
213 ORDER BY
214     timestamp ASC;
215 ";
216
217 // ậCp ảnh t tham ós binding
218 // Tham ós: $startSQL (cho FROM_UNIXTIME), $deviceId, $startSQL (cho BETWEEN),
219 $endSQL (cho BETWEEN)
220 $params_sql_bind = [$startSQL, $deviceId, $startSQL, $endSQL];
221 $types_sql_bind = "sss"; // string (startSQL), string (deviceId), string (startSQL)
222 , string (endSQL)
223
224 // --- DEBUG LOGS ---
225 error_log("DEBUG SQL: Period=" . $period . ", DeviceID=" . $deviceId . ", Interval="
226     . $effectiveInterval . " minutes");
227 error_log("DEBUG SQL: Start=" . $startSQL . ", End=" . $endSQL);
228 error_log("DEBUG SQL: totalSecondsInPeriod=" . $totalSecondsInPeriod . ",
229     num_intervals_to_generate=" . $num_intervals_to_generate . ",
230     limit_numbers_series=" . $limit_numbers_series);
231 error_log("DEBUG SQL: Query: " . $sql);
232 error_log("DEBUG SQL: Bind Params: " . json_encode($params_sql_bind));
233 // --- END DEBUG LOGS ---
234
235 $stmt = $conn->prepare($sql);
236
237 if ($stmt == false) {
238     error_log("Prepare failed in get_history_data.php: " . $conn->error . " SQL: " .
239         $sql);

```

```

228     echo json_encode(["status" => "error", "message" => "Ồi hệ thống. Vui lòng thử
    lại sau. (Prepare Failed)"]);
229     $conn->close();
230     exit();
231 }
232
233 // Binding các tham số
234 // áBt ôbuc ử dụng call_user_func_array cho số lượng tham số động
235 $bind_names = [$types_sql_bind];
236 for ($i = 0; $i < count($params_sql_bind); $i++) {
237     $bind_names[] = &$params_sql_bind[$i];
238 }
239 call_user_func_array([$stmt, 'bind_param'], $bind_names);
240
241
242 $stmt->execute();
243 $result = $stmt->get_result();
244
245 $data = [];
246 if ($result->num_rows > 0) {
247     while($row = $result->fetch_assoc()) {
248         $data[] = [
249             "timestamp" => $row["timestamp"], // ử dụng timestamp từ LEFT JOIN
250             "temperature" => ($row["temperature"] !== null) ? (float) $row["
            temperature"] : null,
251             "humidity" => ($row["humidity"] !== null) ? (float) $row["humidity"] :
            null
252         ];
253     }
254     echo json_encode(["status" => "success", "data" => $data, "interval_minutes" =>
        $effectiveInterval]); // áTr về interval
255     exit();
256 } else {
257     // áTr về status success nhưng data trống nếu không tìm thấy bản ghi
258     echo json_encode(["status" => "success", "message" => "Không tìm thấy dữ liệu
        lịch sử cho thiết bị: " . htmlspecialchars($deviceId) . " trong khoảng thời gian
        đã chọn.", "data" => [], "interval_minutes" => $effectiveInterval]);
259     exit();
260 }
261
262 $stmt->close();
263 $conn->close();
264 ?>

```

```

1 <?php
2 // test_db.php
3 $servername = "localhost";
4 $username = "root";
5 $password = "";
6 $dbname = "dht_sensor_db";
7
8 // áTo kết nối
9 $conn = new mysqli($servername, $username, $password, $dbname);
10
11 // ểKim tra kết nối
12 if ($conn->connect_error) {
13     die("Kết nối thất bại: " . $conn->connect_error);
14 }
15 echo "Kết nối database thành công!";
16 $conn->close();
17 ?>

```

```

1 <?php
2 // db_connect.php - File xử lý kết nối cơ sở dữ liệu
3
4 // ưLu ý: Các cấu hình hiển thị lỗi và CORS đã được di chuyển vào app_config.php
5

```

```

6 // Thông tin kết nối cơ sở dữ liệu
7 $servername = "localhost"; // Tên máy chủ
8 $username = "root"; // Tên người dùng MySQL
9 $password = ""; // Mật khẩu MySQL
10 $dbname = "dht_sensor_db"; // Tên cơ sở dữ liệu
11
12 // Kết nối
13 $conn = new mysqli($servername, $username, $password, $dbname);
14
15 // Kiểm tra kết nối
16 if ($conn->connect_error) {
17     // Không đặt header('Content-Type: application/json'); ở đây nữa.
18     // Việc này sẽ được thực hiện trong từng file API sau khi app_config.php được
19     // include.
20
21     // Ghi lỗi vào log của máy chủ
22     error_log("Kết nối database thất bại: " . $conn->connect_error); // Ghi lỗi vào
23     // log của máy chủ
24
25     // Đặt mã lỗi HTTP và trả về JSON cho phía client
26     http_response_code(500);
27     echo json_encode(["status" => "error", "message" => "Lỗi kết nối cơ sở dữ liệu.
28     Vui lòng thử lại sau."]);
29     exit();
30 }
31
32 // Thiết lập ký tự cho kết nối để (hỗ trợ encoding UTF và Unicode)
33 $conn->set_charset("utf8mb4");
34
35 ?>

```

```

1 <?php
2 // TH_DO_AN/app_config.php - Cấu hình chung cho ứng dụng
3
4 // --- Cấu hình hiển thị lỗi PHP ---
5 // Trong môi trường Production, nên tắt hiển thị lỗi trực tiếp và chỉ ghi vào log.
6 // Đối với môi trường phát triển (development), bạn có thể bật lại (ini_set('
7     display_errors', 1);)
8 ini_set('display_errors', 1); // Tắt hiển thị lỗi trên trình duyệt
9 ini_set('display_startup_errors', 1); // Tắt hiển thị lỗi khi khởi động
10 error_reporting(E_ALL); // Bật tất cả các loại lỗi để ghi vào log (ví dụ: php_error.
11     log)
12
13 // --- Bỏ qua cấu hình CORS ---
14 // Đường dẫn này gọi định config_cors.php nằm ở thư mục cha của TH_DO_AN/ (tức là
15     httdocs/)
16 require_once __DIR__ . '/../config_cors.php';
17
18 // --- Các cài đặt chung khác của ứng dụng (nếu có) ---
19 // Ví dụ: múi giờ mặc định cho PHP
20 date_default_timezone_set('Asia/Ho_Chi_Minh');
21
22 ?>

```

A.2.2 Các file API trong TH_DO_AN/api/

```

1 <?php
2 // Bỏ qua file cấu hình chung của ứng dụng (bao gồm cài đặt lỗi và CORS)
3 require_once '../app_config.php';
4
5 // Đặt Content-Type cho phản hồi này là JSON
6 header('Content-Type: application/json');
7
8 // Bỏ qua file kết nối cơ sở dữ liệu
9 // Vì get_devices.php nằm trong TH_DO_AN/api/ và db_connect.php nằm trong TH_DO_AN/

```

```

10 // nên đường dẫn cần đi ra ngoài ômt áp để đến db_connect.php
11 require_once '../db_connect.php';
12
13 // Kiểm tra kết nối sau khi include db_connect.php
14 if (!isset($conn) || $conn->connect_error) {
15     error_log("Database connection error in get_devices.php: " . ($conn->
16         connect_error ?? 'Connection object not set.'));
17     echo json_encode(['success' => false, 'message' => 'Ồi kết nối CSDL: ' . ($conn
18         ->connect_error ?? 'Không thể kết nối.')] );
19     exit();
20 }
21
22 $response = ['success' => false, 'devices' => [], 'message' => ''];
23
24 try {
25     // Lựa chọn 1: KHUYẾN DÙNG - sử dụng bảng 'devices' riêng với 'device_id' và '
26     device_name'
27     // Đã COMMENT OUT toàn bộ phần này vì bảng 'devices' không tồn tại.
28     /*
29     $sql_devices = "SELECT device_id, device_name FROM devices ORDER BY device_name
30     ASC";
31     $stmt_devices = $conn->prepare($sql_devices);
32     if ($stmt_devices === false) {
33         throw new Exception("Ồi prepare statement: " . $conn->error);
34     }
35     $stmt_devices->execute();
36     $result_devices = $stmt_devices->get_result();
37
38     $devices = [];
39     while ($row = $result_devices->fetch_assoc()) {
40         $devices[] = $row;
41     }
42
43     $response['success'] = true;
44     $response['devices'] = $devices;
45     $response['message'] = 'Ài danh sách thiết bị thành công.';
46
47     $stmt_devices->close();
48     */
49
50     // Lựa chọn 2: SỬ DỤNG ẾU KHÔNG CỐ BẢNG 'devices' RIÊNG
51     // Nếu bạn ÍCH vào device_id trong bảng sensor_readings hoặc device_settings
52     // Chúng ta sẽ lấy device_id từ bảng device_settings theo thông tin bạn đã cung
53     cấp
54     $sql_devices = "SELECT DISTINCT device_id FROM sensor_readings ORDER BY
55     device_id ASC";
56     $stmt_devices = $conn->prepare($sql_devices);
57     if ($stmt_devices === false) {
58         throw new Exception("Ồi prepare statement: " . $conn->error);
59     }
60     $stmt_devices->execute();
61     $result_devices = $stmt_devices->get_result();
62
63     $devices = [];
64     while ($row = $result_devices->fetch_assoc()) {
65         $devices[] = [
66             'device_id' => $row['device_id'],
67             'device_name' => 'Thiết bị: ' . $row['device_id'] // Tên hiển thị từ
68             ID
69         ];
70     }
71     $response['success'] = true;
72     $response['devices'] = $devices;
73     $response['message'] = 'Ài danh sách thiết bị thành công.';
74     $stmt_devices->close();
75 }

```

```

69 } catch (Exception $e) {
70     $response['message'] = 'ỒLi truy vấn cơ sở dữ liệu: ' . $e->getMessage();
71     error_log("ỒLi trong get_devices.php: " . $e->getMessage()); // Ghi log lỗi vào
    server log
72 } finally {
73     if (isset($conn) && $conn instanceof mysqli && !$conn->connect_error) { // Đảm
        bảo $conn là đối tượng MySQLi.php để tắt khi đóng
74         $conn->close();
75     }
76 }
77
78 echo json_encode($response);
79 exit(); // Đảm bảo exit() để dừng script sau khi gửi JSON
80 ?>

```

```

1 <?php
2 // Bao gồm file cấu hình chung của ứng dụng (bao gồm cài đặt lỗi và CORS)
3 // Đường dẫn này chỉ định export_data_csv.php nằm trong TH_DO_AN/api/
4 // và app_config.php nằm trong TH_DO_AN/ (thư mục cha của api/)
5 require_once __DIR__ . '/../app_config.php';
6
7 // Thiết lập header cho file CSV
8 header('Content-Type: text/csv');
9 header('Content-Disposition: attachment; filename="sensor_data_export.csv"');
10
11 // Bao gồm file kết nối cơ sở dữ liệu
12 // Đường dẫn này chỉ định db_connect.php nằm trong TH_DO_AN/ (thư mục cha của api/)
13 require_once __DIR__ . '/../db_connect.php';
14
15 // Kiểm tra xem biến $conn đã được thiết lập từ db_connect.php và có thành công không.
16 if (!isset($conn) || $conn->connect_error) {
17     // Ghi lỗi vào log và dừng script với thông báo đơn giản.
18     error_log("Database connection failed in export_data_csv.php: " . ($conn->
        connect_error ?? 'Connection object not set.'));
19     die("ỒLi kết nối cơ sở dữ liệu. Vui lòng thử lại sau.");
20 }
21
22 // Lấy period và deviceId từ tham số GET
23 $period = $_GET['period'] ?? '24h';
24 // ỦA ĐỔI DÒNG NÀY: Thay 'deviceId' bằng 'device_id' (thực thường) để khớp với main.
    js
25 $deviceId = isset($_GET['device_id']) ? $_GET['device_id'] : null;
26
27 $startDate = $_GET['startDate'] ?? null;
28 $endDate = $_GET['endDate'] ?? null;
29
30 // Xử lý deviceId riêng (theo yêu cầu ảnh hưởng với các API khác)
31 if (empty($deviceId)) {
32     error_log("Missing device ID in export_data_csv.php.");
33     die("ỒLi: Thiếu ID thiết bị để xuất dữ liệu."); // Dòng này sẽ không còn được kích
        hoạt nếu device_id được truyền đúng
34 }
35
36 // Logic cho truy vấn SQL
37 $sql = "SELECT reading_time, temperature, humidity FROM sensor_readings WHERE
    device_id = ?"; // Đổi tên bảng và cột
38 $params = [$deviceId];
39 $types = "s"; // s for string (device_id)
40
41 $currentTime = new DateTime(); // Khi tạo DateTime để tính khoảng thời gian
42 $interval = '';
43
44 switch ($period) {
45     case '1h':
46         $interval = '1 HOUR';

```

```

47     break;
48     case '6h':
49         $interval = '6 HOUR';
50         break;
51     case '12h':
52         $interval = '12 HOUR';
53         break;
54     case '24h':
55         $interval = '24 HOUR';
56         break;
57     case '7d':
58         $interval = '7 DAY';
59         break;
60     case '30d':
61         $interval = '30 DAY';
62         break;
63     case 'custom':
64         if ($startDate && $endDate) {
65             // Validate and sanitize dates
66             $start = new DateTime($startDate);
67             $end = new DateTime($endDate);
68             if ($start > $end) {
69                 die("ôLi: Ngày bắt đầu không ểth sau ngày ểkt thúc."); // ậCp ậnh
thông báo ôli
70             }
71             $sql .= " AND reading_time BETWEEN ? AND ?"; // Đổi tên ộct
72             $params[] = $start->format('Y-m-d H:i:s');
73             $params[] = $end->format('Y-m-d H:i:s');
74             $types .= "ss"; // Add two more 's' for string dates
75         } else {
76             die("ôLi: ấKhong ờthi gian tùy ichnh yêu ầu ngày bắt đầu và ểkt thúc.");
// ậCp ậnh thông báo ôli
77         }
78         break;
79     default:
80         $interval = '24 HOUR'; // ắMc định 24 ờgi ếuu không ớkhp
81         break;
82 }
83
84 if ($interval !== '') {
85     $sql .= " AND reading_time >= DATE_SUB(NOW(), INTERVAL $interval)"; // Đổi tên
ộct
86 }
87
88 $sql .= " ORDER BY reading_time ASC"; // Đổi tên ộct
89
90 $stmt = $conn->prepare($sql);
91
92 if (!$stmt) {
93     error_log("Failed to prepare statement in export_data_csv.php: " . $conn->error
. " SQL: " . $sql);
94     die("ôLi: Không ểth ắchun ịb câu ệlnh SQL. Vui lòng ửth ặli sau."); // ậCp ậnh
thông báo ôli
95 }
96
97 // ửS ụng call_user_func_array để bind_param ớvi ốs ượlng tham ốs động
98 if (!empty($params)) {
99     $bind_names = [$types];
100     for ($i = 0; $i < count($params); $i++) {
101         $bind_names[] = &$params[$i];
102     }
103     call_user_func_array([$stmt, 'bind_param'], $bind_names);
104 }
105
106 $stmt->execute();
107 $result = $stmt->get_result();

```

```

108
109 $output = fopen('php://output', 'w'); // Mở output stream
110
111 // Ghi tiêu đề CSV (ập nhĩn cho ườgĩ dũg)
112 fputcsv($output, ['Thĩ gĩn đợc', 'ệNhiĩ độ (C)', 'Độ ẩm (%)']); // ặHợc ữgĩ ếting
    Anh ếũ ốmun
113
114 // Ghi ữ đệliu
115 while ($row = $result->fetch_assoc()) {
116     // Đẩm ẩbợ ẩly đũg cắ đợc theo tẽn ốĩ
117     fputcsv($output, [$row['reading_time'], $row['temperature'], $row['humidity']]);
118 }
119
120 fclose($output); // Đổg output stream
121
122 $stmt->close();
123 $conn->close();
124 exit(); // Đẩm ẩbợ exit() đễ ửdũg script
125 ?>

```

```

1 <?php
2 // Bợ ồgĩ file ẩcũ hĩnh chung ẩcũ ứg ựdũg (bợ ồgĩ cặ đặ ốli vầ CORS)
3 require_once '../app_config.php';
4
5 // Đặ Content-Type cho ẩphĩ ồĩ nầy lầ JSON
6 header('Content-Type: application/json');
7
8 // Bợ ồgĩ file ếkt ốĩ đợc ử đệliu
9 require_once '../db_connect.php';
10
11 // ểKim trỏ xem ếbĩn $conn đầ đợc ếthĩt ẩp ừ đb_connect.php vầ cầ hầnh cồg kồg.
12 if (!isset($conn) || $conn->connect_error) {
13     error_log("Database connection failed in save_alert_thresholds.php: " . ($conn->
14         connect_error ?? 'Connection object not set.));
15     echo json_encode(['status' => 'error', 'message' => 'ốLi ếkt ốĩ CSDL: ' . ($conn
16         ->connect_error ?? 'Kồg ểth ếkt ốĩ.));
17     exit();
18 }
19
20 // ểKim trỏ ườphũg ứthc yẽu ẩcũ
21 if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
22     echo json_encode(['status' => 'error', 'message' => 'Invalid request method.]);
23     exit();
24 }
25
26 // ẩly ữ đệliu JSON ừ request body
27 $input = file_get_contents('php://input');
28 $data = json_decode($input, true);
29
30 // ểKim trỏ ữ đệliu đầ vầo
31 if (json_last_error() !== JSON_ERROR_NONE) {
32     echo json_encode(['status' => 'error', 'message' => 'Invalid JSON input.]);
33     exit();
34 }
35
36 // ểKim trỏ cắ ườtrũg ẩb đợc
37 if (!isset($data['device_id'], $data['temp_threshold'], $data['humidity_threshold'])
38 ) {
39     echo json_encode(['status' => 'error', 'message' => 'Missing required fields (
40         device_id, temp_threshold, humidity_threshold).]);
41     exit();
42 }
43
44 $deviceId = $data['device_id'];
45 $tempThreshold = floatval($data['temp_threshold']);
46 $humidityThreshold = floatval($data['humidity_threshold']);

```



```

43
44 // Kiểm tra giá trị hợp lệ
45 if (!is_numeric($tempThreshold) || !is_numeric($humidityThreshold)) {
46     echo json_encode(['status' => 'error', 'message' => 'Threshold values must be
47         numbers.']);
48     exit();
49 }
50 $success_count = 0;
51 $error_messages = [];
52
53 // --- SQL để cập nhật ngưỡng nhiệt độ ---
54 $setting_name_temp = 'temp_threshold';
55 $bind_temp_value = (string)$tempThreshold; // Chuyển đổi float sang string để lưu vào
56     setting_value (varchar)
57 $stmt_temp = $conn->prepare("INSERT INTO device_settings (device_id, setting_name,
58     setting_value) VALUES (?, ?, ?) ON DUPLICATE KEY UPDATE setting_value = ?");
59 if ($stmt_temp === false) {
60     error_log("Prepare failed for temp_threshold in save_alert_thresholds.php: " .
61         $conn->error);
62 } else {
63     $error_messages[] = 'Lỗi hệ thống khi chuẩn bị truy vấn nhiệt độ.';
64     $stmt_temp->bind_param("sss", $deviceId, $setting_name_temp, $bind_temp_value,
65         $bind_temp_value);
66     if ($stmt_temp->execute()) {
67         $success_count++;
68     } else {
69         error_log("Execute failed for temp_threshold in save_alert_thresholds.php: "
70             . $stmt_temp->error);
71         $error_messages[] = 'Lỗi khi lưu ngưỡng nhiệt độ: ' . $stmt_temp->error;
72     }
73     $stmt_temp->close();
74 }
75
76 // --- SQL để cập nhật ngưỡng độ ẩm ---
77 $setting_name_humidity = 'humidity_threshold';
78 $bind_humidity_value = (string)$humidityThreshold; // Chuyển đổi float sang string
79     để lưu vào setting_value (varchar)
80 $stmt_humidity = $conn->prepare("INSERT INTO device_settings (device_id,
81     setting_name, setting_value) VALUES (?, ?, ?) ON DUPLICATE KEY UPDATE
82     setting_value = ?");
83 if ($stmt_humidity === false) {
84     error_log("Prepare failed for humidity_threshold in save_alert_thresholds.php: "
85         . $conn->error);
86 } else {
87     $error_messages[] = 'Lỗi hệ thống khi chuẩn bị truy vấn độ ẩm.';
88     $stmt_humidity->bind_param("sss", $deviceId, $setting_name_humidity,
89         $bind_humidity_value, $bind_humidity_value);
90     if ($stmt_humidity->execute()) {
91         $success_count++;
92     } else {
93         error_log("Execute failed for humidity_threshold in save_alert_thresholds.
94             php: " . $stmt_humidity->error);
95         $error_messages[] = 'Lỗi khi lưu ngưỡng độ ẩm: ' . $stmt_humidity->error;
96     }
97     $stmt_humidity->close();
98 }
99
100 $conn->close();
101
102 // --- Trả về phản hồi cuối cùng ---
103 if ($success_count == 2) {

```

```

96     echo json_encode(['status' => 'success', 'message' => 'Các ườngng ảnh báo đã
    được ập ậht thành công.']);
97 } else {
98     echo json_encode(['status' => 'error', 'message' => 'Có ỗili khi ập ậht ườngng
    ảnh báo: ' . implode(';', $error_messages)]);
99 }
100 exit(); // Đảm bảo exit() để ừdng script
101 ?>

1 <?php
2 // Bao ồm file ầu hình chung ủa ứng ụngng (bao ồm cài đặt ỗili và CORS)
3 require_once '../app_config.php';
4
5 // Đặt Content-Type cho ảphn ời này là JSON
6 header('Content-Type: application/json');
7
8 // Bao ồm file ết ốni ợc ởs ữd ệliu
9 require_once '../db_connect.php';
10
11 // ểKim tra xem ếbin $conn đã được ếthit ập ừt db_connect.php và có thành công không.
12 if (!isset($conn) || $conn->connect_error) {
13     error_log("Database connection failed in save_email_settings.php: " . ($conn->
14     connect_error ?? 'Connection object not set.));
15     echo json_encode(['status' => 'error', 'message' => 'ỗiLi ết ốni CSDL: ' . ($conn
16     ->connect_error ?? 'Không ểth ết ốni.']));
17     exit();
18 }
19
20 // ểKim tra ươphng ứthc yêu ầu (ích ấph ậnhn POST)
21 if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
22     echo json_encode(['status' => 'error', 'message' => 'Invalid request method.']);
23     exit();
24 }
25
26 // ấLy ữd ệliu JSON ừt request body
27 $input = file_get_contents('php://input');
28 $data = json_decode($input, true);
29
30 // ểKim tra ữd ệliu đầu vào
31 if (json_last_error() !== JSON_ERROR_NONE) {
32     echo json_encode(['status' => 'error', 'message' => 'Invalid JSON input.']);
33     exit();
34 }
35
36 // ểKim tra các ườtrng ấbt ộbuc
37 if (!isset($data['device_id'], $data['enable_email_alerts'], $data['email_address'])) {
38     echo json_encode(['status' => 'error', 'message' => 'Missing required fields (
39     device_id, enable_email_alerts, email_address).']);
40     exit();
41 }
42
43 $deviceId = $data['device_id'];
44 $enableEmailAlerts = (bool)$data['enable_email_alerts']; // ểChuyển đổi ề boolean
45 $emailAddress = $data['email_address'];
46
47 // ểKim tra định ảdng email ếnu ậbt ảnh báo
48 if ($enableEmailAlerts && !filter_var($emailAddress, FILTER_VALIDATE_EMAIL)) {
49     echo json_encode(['status' => 'error', 'message' => 'Invalid email address
50     format.']);
51     exit();
52 }
53
54 $success_count = 0;
55 $error_messages = [];
56

```

```

53 // --- SQL để cập nhật địa chỉ email ---
54 $setting_name_email = 'email_address';
55 $bind_email_value = (string)$emailAddress; // chuyển đổi sang string
56
57 $stmt_email = $conn->prepare("INSERT INTO device_settings (device_id, setting_name,
    setting_value) VALUES (?, ?, ?) ON DUPLICATE KEY UPDATE setting_value = ?");
58 if ($stmt_email === false) {
59     error_log("Prepare failed for email_address in save_email_settings.php: " .
        $conn->error);
60     $error_messages[] = 'Lỗi khi thiết lập địa chỉ email.';
61 } else {
62     $stmt_email->bind_param("sss", $deviceId, $setting_name_email,
        $bind_email_value, $bind_email_value);
63     if ($stmt_email->execute()) {
64         $success_count++;
65     } else {
66         error_log("Execute failed for email_address in save_email_settings.php: " .
            $stmt_email->error);
67         $error_messages[] = 'Lỗi khi lưu địa chỉ email: ' . $stmt_email->error;
68     }
69     $stmt_email->close();
70 }
71
72 // --- SQL để cập nhật trạng thái bật/tắt cảnh báo email ---
73 $setting_name_enable_alerts = 'enable_email_alerts';
74 $bind_enable_alerts_value = (string)($enableEmailAlerts ? 1 : 0); // chuyển boolean
    sang '1' hoặc '0' (string)
75
76 $stmt_enable_alerts = $conn->prepare("INSERT INTO device_settings (device_id,
    setting_name, setting_value) VALUES (?, ?, ?) ON DUPLICATE KEY UPDATE
    setting_value = ?");
77 if ($stmt_enable_alerts === false) {
78     error_log("Prepare failed for enable_email_alerts in save_email_settings.php: " .
        $conn->error);
79     $error_messages[] = 'Lỗi khi thiết lập trạng thái cảnh báo email.';
80 } else {
81     $stmt_enable_alerts->bind_param("sss", $deviceId, $setting_name_enable_alerts,
        $bind_enable_alerts_value, $bind_enable_alerts_value);
82     if ($stmt_enable_alerts->execute()) {
83         $success_count++;
84     } else {
85         error_log("Execute failed for enable_email_alerts in save_email_settings.php: " .
            $stmt_enable_alerts->error);
86         $error_messages[] = 'Lỗi khi lưu trạng thái cảnh báo email: ' .
            $stmt_enable_alerts->error;
87     }
88     $stmt_enable_alerts->close();
89 }
90
91 $conn->close();
92
93 // --- Trả về phản hồi cuối cùng ---
94 if ($success_count == 2) {
95     echo json_encode(['status' => 'success', 'message' => 'Cài đặt email đã được cập
        nhật thành công.']);
96 } else {
97     echo json_encode(['status' => 'error', 'message' => 'Có lỗi khi cập nhật cài đặt
        email: ' . implode(' ', $error_messages)]);
98 }
99 exit(); // Đảm bảo exit() để dừng script
100 ?>

```

```

1 <?php
2 // Bao gồm file cấu hình chung của ứng dụng (bao gồm cài đặt lỗi và CORS)
3 // Đường dẫn này ghi định save_update_frequency.php nằm trong TH_DO_AN/api/

```

```

4 // và app_config.php nằm trong TH_DO_AN/ (ưth ưmc cha ưca api/)
5 require_once '../app_config.php';
6
7 // áRt quan ợtng: Đặt header để báo cho trình ệduyt ếbit đây là JSON
8 // ẢPHI ĐẢM ẢO DÒNG NÀY VÀ session_start() ẦNM SAU require_once app_config.php
9 header('Content-Type: application/json');
10 session_start(); // áBt đầu session ếnu ận ứs ụng session cho ệvic xác ựthc ặhoc ưlu
    ứtr
11
12 $response = ['status' => 'error', 'message' => 'ồLi không xác định ừt server.'];
13
14 // Bao ềgm file ếkt ốni ợc ớs ứđ ệliu chung
15 // Đường ấdn này ặgi định db_connect.php nằm trong TH_DO_AN/ (ưth ưmc cha ưca api/)
16 require_once '../db_connect.php';
17
18 // ểKim tra ếkt ốni sau khi include db_connect.php
19 if (!isset($conn) || $conn->connect_error) {
20     error_log("Database connection failed in save_update_frequency.php: " . ($conn->
21         connect_error ?? 'Connection object not set.'));
22     echo json_encode(['status' => 'error', 'message' => 'ồLi ếkt ốni CSDL: ' . ($conn
23         ->connect_error ?? 'Không ểth ếkt ốni.')] );
24     exit();
25 }
26
27 // ừX lý preflight OPTIONS request (ếNu ưcha được ừx lý trong config_cors.php cho
    POST)
28 // (ườThng thì config_cors.php đã ừx lý cho átt ắc methods, nên đạon này có ểth ỏb
    qua ếnu ận áchc áchn)
29 /*
30 if ($_SERVER['REQUEST_METHOD'] === 'OPTIONS') {
31     http_response_code(200);
32     exit();
33 }
34 */
35
36 // ếThit ậlp charset cho ếkt ốni (Dòng này đã có trong db_connect.php, có ểth ỏb qua
    ở đây)
37 // $conn->set_charset('utf8mb4');
38
39 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
40     $input = file_get_contents('php://input');
41     $data = json_decode($input, true); // true để átr ềv ảmng ếkt ợhp
42
43     // ểKim tra ồli JSON decode
44     if (json_last_error() !== JSON_ERROR_NONE) {
45         $response['message'] = 'ườ ệliu JSON ừgi ừt client không ợhp ệl: ' .
46         json_last_error_msg();
47         echo json_encode($response);
48         exit();
49     }
50
51     // ỂKIM TRA ẮC 'frequency' VÀ 'device_id'
52     elseif ($data && isset($data['frequency']) && isset($data['device_id'])) {
53         $frequency = intval($data['frequency']); // Đảm ảbo là ốs nguyên
54         // KHÔNG ẦCN real_escape_string vì đã dùng Prepared Statements
55         $device_id = $data['device_id'];
56
57         // ểKim tra tính ợhp ệl ưca ầtn ástut (ví ựđ: ảphi là ốs nguyên ươdng)
58         if ($frequency <= 0) {
59             $response['message'] = 'ầTh ástut ảphi là ộmt ốs nguyên ươdng.';
60             echo json_encode($response);
61             exit();
62         }
63
64         // ểKim tra device_id không ờrng
65         elseif (empty($device_id)) {

```

```

61     $response['message'] = 'ID ếthit ịb không được để ốtrng.';
62     echo json_encode($response);
63     exit();
64 }
65 else {
66     // ưLu ầtn ấsút vào CSDL
67     $setting_name = 'update_frequency';
68     // ỦS ỤDNG ẮBNG 'device_settings' VÀ THÊM ỘCT 'device_id' VÀO TRUY ẮVN
69     $stmt = $conn->prepare("INSERT INTO device_settings (device_id,
        setting_name, setting_value) VALUES (?, ?, ?) ON DUPLICATE KEY UPDATE
        setting_value = ?");
70
71     if ($stmt === false) { // ểKim tra ỗli prepare statement
72         error_log("Prepare failed in save_update_frequency.php: " . $conn->
        error);
73         $response['message'] = 'ỗLi ệh ốthng khi ắchun ịb truy ắvn.';
74         echo json_encode($response);
75         $conn->close();
76         exit();
77     }
78
79     $bind_frequency = (string)$frequency; // Ép ểkiu rõ ràng thành string
80     $stmt->bind_param("ssss", $device_id, $setting_name, $bind_frequency,
        $bind_frequency);
81
82     if ($stmt->execute()) {
83         $response = ['status' => 'success', 'message' => 'ầTn ấsút ắcp ậnh
        cho ếthit ịb ' . htmlspecialchars($device_id) . ' đã được ưlu: ' . $frequency . '
        giây.'];
84     } else {
85         error_log("Execute failed in save_update_frequency.php: " . $stmt->
        error);
86         $response = ['status' => 'error', 'message' => 'ỗLi khi ưlu ầtn ấsút
        vào CSDL: ' . $stmt->error]; // Chi ếtit ỗli ừt statement
87     }
88     $stmt->close();
89 }
90 } else {
91     $response['message'] = 'ừ ệliu ầtn ấsút ắhoc ID ếthit ịb không ợhp ệl (ếthiu
        ườtrng "frequency" ắhoc "device_id" ắhoc định ặdnng sai).';
92     echo json_encode($response);
93     exit();
94 }
95 } else {
96     $response['message'] = 'Yêu ầcu không ợhp ệl (ích ắchp ậnhn ươphng ứthc POST).';
97     echo json_encode($response);
98     exit();
99 }
100
101 $conn->close(); // Đóng ểkt ốni CSDL (dòng này ếs được ỗb qua ếnu exit() đã được ợgi
        ướtrng đó)
102 echo json_encode($response); // ắTr ềv ắphn ồhi JSON ốcuì cùng
103 exit(); // Đắmb ắbo exit() sau khi ửgi ắphn ồhi ốcuì cùng
104 ?>

```

```

1 <?php
2 require_once '../app_config.php';
3 header('Content-Type: application/json');
4 require_once '../db_connect.php';
5
6 if (!isset($conn) || $conn->connect_error) {
7     error_log("Database connection failed in get_alert_thresholds.php: " . ($conn->
        connect_error ?? 'Connection object not set.));
8     echo json_encode(['status' => 'error', 'message' => 'ỗLi ểkt ốni CSDL.']);
9     exit();
10 }

```

```

11
12 if ($_SERVER['REQUEST_METHOD'] === 'GET') {
13     $deviceId = $_GET['device_id'] ?? null;
14
15     if (empty($deviceId)) {
16         echo json_encode(['status' => 'error', 'message' => 'Thiếu ID thiết bị.']);
17         exit();
18     }
19
20     $thresholds = [
21         'temp_threshold' => null,
22         'humidity_threshold' => null
23     ];
24
25     try {
26         $stmt = $conn->prepare("SELECT setting_name, setting_value FROM
device_settings WHERE device_id = ? AND (setting_name = 'temp_threshold' OR
setting_name = 'humidity_threshold')");
27         if ($stmt === false) {
28             throw new Exception("Prepare failed: " . $conn->error);
29         }
30         $stmt->bind_param("s", $deviceId);
31         $stmt->execute();
32         $result = $stmt->get_result();
33
34         while ($row = $result->fetch_assoc()) {
35             $thresholds[$row['setting_name']] = $row['setting_value'];
36         }
37
38         echo json_encode(['status' => 'success', 'thresholds' => $thresholds]);
39
40         $stmt->close();
41     } catch (Exception $e) {
42         error_log("Error in get_alert_thresholds.php: " . $e->getMessage());
43         echo json_encode(['status' => 'error', 'message' => 'Lỗi truy vấn: ' . $e->
getMessage()]);
44     } else {
45         echo json_encode(['status' => 'error', 'message' => 'Server đang ngừng hoạt động.']);
46     }
47 }
48
49 $conn->close();
50 exit();
51 ?>

```

```

1 <?php
2 // Bao gồm file cấu hình chung của ứng dụng (bao gồm cài đặt lỗi và CORS)
3 require_once '../app_config.php';
4
5 // Đặt Content-Type cho phản hồi này là text/plain vì ESP32 đang mong đợi "true" hoặc
"false"
6
7 // Nếu bạn muốn trả về JSON cho ESP32, hãy thay đổi thành 'application/json'
8 // và sửa code ESP32 để parse JSON. Hiện tại, tôi giữ nguyên như đã thảo luận.
9 header('Content-Type: text/plain');
10
11 // Bao gồm file kết nối cơ sở dữ liệu
12 require_once '../db_connect.php';
13
14 // Kiểm tra kết nối
15 if (!isset($conn) || $conn->connect_error) {
16     error_log("Database connection failed in get_dht_status.php: " . ($conn->
connect_error ?? 'Connection object not set.'));
17
18     // Trả về 'true' để không làm gián đoạn phản ứng nếu có lỗi DB, hoặc một giá trị
19     // xác định an toàn.
20     echo 'true'; // Mặc định là bất cứ khi nào có lỗi kết nối CSDL

```

```

18     exit();
19 }
20
21 // áLy device_id từ tham số GET để (ứng thích với ESP32)
22 // Nếu không có, abn cần một device_id mặc định khác từ lý do.
23 // Trong trường hợp này, ESP32 sẽ gửi request mà không có tham số GET,
24 // nên chúng ta sẽ lấy giá trị mặc định đã nêu.
25 // HOC: Nếu ESP32 gửi device_id qua GET: $deviceId = $_GET['device_id'] ?? '
    default_device_id';
26 // Hiện tại, ESP32 chưa abn không gửi device_id, nên chúng ta sẽ lấy từ cài đặt chung.
27 // Nếu abn có nhiều ESP32, abn cần thiết bị gửi device_id trong request.
28
29 // áLy trạng thái từ CSDL
30 $setting_name = 'dht_enabled';
31 $deviceId = isset($_GET['device_id']) ? $_GET['device_id'] : null; // áLy device_id
    từ GET nếu có
32
33 $stmt = null;
34 $result = null;
35 $row = null;
36
37 try {
38     if ($deviceId) {
39         // Nếu có device_id được cung cấp, lấy cài đặt riêng cho thiết bị đó
40         $stmt = $conn->prepare("SELECT setting_value FROM device_settings WHERE
device_id = ? AND setting_name = ?");
41         if ($stmt == false) {
42             throw new Exception("Prepare failed for device_specific: " . $conn->
error);
43         }
44         $stmt->bind_param("ss", $deviceId, $setting_name);
45     } else {
46         // Fallback: lấy cài đặt chung nếu không có device_id khác abn ích có một
    thiết bị
47         // Trong trường hợp này, gđ định ESP32_Sensor_02 là device_id mặc định.
48         // Đây là cách hoạt động hiện tại của firmware ESP32 của abn.
49         $default_device_id = "ESP32_Sensor_02"; // Thay bằng ID mặc định của abn nếu
    có
50         $stmt = $conn->prepare("SELECT setting_value FROM device_settings WHERE
device_id = ? AND setting_name = ?");
51         if ($stmt == false) {
52             throw new Exception("Prepare failed for default_device: " . $conn->
error);
53         }
54         $stmt->bind_param("ss", $default_device_id, $setting_name);
55     }
56
57     $stmt->execute();
58     $result = $stmt->get_result();
59     $row = $result->fetch_assoc();
60
61     if ($row) {
62         // Trả về 'true' khác 'false' (string) dựa trên giá trị từ DB
63         echo $row['setting_value'];
64     } else {
65         // Nếu không tìm thấy cài đặt, mặc định là 'true'
66         echo 'true';
67         // Ghi log nếu cài đặt không tìm thấy
68         error_log("DHT status setting not found for device_id: " . ($deviceId ?? "
default_device_id") . ". Defaulting to 'true'.");
69     }
70 } catch (Exception $e) {
71     // Trả về 'true' để không làm gián đoạn phản ứng nếu có lỗi
72     echo 'true';
73     error_log("Error in get_dht_status.php: " . $e->getMessage()); // Ghi log lỗi
    vào server log

```

```

74 } finally {
75     if ($stmt) {
76         $stmt->close();
77     }
78     if (isset($conn) && $conn instanceof mysqli && !$conn->connect_error) {
79         $conn->close();
80     }
81 }
82 exit();
83 ?>

```

```

1 <?php
2 // Bao ògm file ácu hình chung ùca ứng ụngng (bao ògm cài ðặt òli và CORS)
3 require_once '../app_config.php';
4
5 // áRt quan ợtng: Ðặt header ðể báo cho trình ệduyt ébit ðây là JSON
6 header('Content-Type: application/json');
7 session_start(); // áBt ðầu session ểnu ạbn ửs ụngng session cho ệvic xác ựthc ặhoc ứlu
   ừtr
8
9 $response = ['status' => 'error', 'message' => 'ỏLi không xác ðịnh ừt server.'];
10
11 // Bao ògm file ékt óni ợc ởs ửd ệliu chung
12 require_once '../db_connect.php';
13
14 // ểKim tra ékt óni sau khi include db_connect.php
15 if (!isset($conn) || $conn->connect_error) {
16     error_log("Database connection failed in toggle_dht_status.php: " . ($conn->
17         connect_error ?? 'Connection object not set.));
18     echo json_encode(['status' => 'error', 'message' => 'ỏLi ékt óni CSDL: ' . ($conn
19         ->connect_error ?? 'Không ểth ékt óni.']));
20     exit();
21 }
22
23 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
24     $input = file_get_contents('php://input');
25     $data = json_decode($input, true); // true ðể ảtr ềv ảmng ékt ợhp
26
27     // ểKim tra òli JSON decode
28     if (json_last_error() !== JSON_ERROR_NONE) {
29         $response['message'] = 'ửD ệliu JSON ửgi ừt client không ợhp ệl: ' .
30             json_last_error_msg();
31         echo json_encode($response);
32         exit();
33     }
34
35     // ểKim tra các ườtrng ábt ộbuc
36     elseif ($data && isset($data['status']) && isset($data['device_id'])) {
37         $status = (bool)$data['status']; // ểChuyển ðối ềv boolean true/false
38         $device_id = $data['device_id']; // ID ùca ểthit ịb
39
40         // ểKim tra device_id không ỏrng
41         if (empty($device_id)) {
42             $response['message'] = 'ID ểthit ịb không ðược ðể ỏtrng.';
43             echo json_encode($response);
44             exit();
45         } else {
46             // ứLu ảtrng thái vào CSDL trong ảbng device_settings
47             $setting_name = 'dht_enabled';
48             $bind_status_value = $status ? 'true' : 'false'; // ứLu ướdi ạdng ỏchui '
               true'/'false'
49
50             $stmt = $conn->prepare("INSERT INTO device_settings (device_id,
51                 setting_name, setting_value) VALUES (?, ?, ?) ON DUPLICATE KEY UPDATE
52                 setting_value = ?");
53
54             if ($stmt === false) { // ểKim tra òli prepare statement

```



```

49         error_log("Prepare failed in toggle_dht_status.php: " . $conn->
error);
50         $response['message'] = 'ỒLi ệh óthng khi ắchun ịb truy ắvn.';
51         echo json_encode($response);
52         $conn->close();
53         exit();
54     }
55
56     $stmt->bind_param("ssss", $device_id, $setting_name, $bind_status_value,
$bind_status_value);
57
58     if ($stmt->execute()) {
59         $response = ['status' => 'success', 'message' => 'ạTrng thái DHT cho
ếthit ịb ' . htmlspecialchars($device_id) . ' đã được ắp ắnh thành: ' . (
$status ? 'ắBt' : 'ắTt')];
60     } else {
61         error_log("Execute failed in toggle_dht_status.php: " . $stmt->error
);
62         $response = ['status' => 'error', 'message' => 'ỒLi khi ứu ạtrng
thái DHT vào CSDL: ' . $stmt->error]; // Chi ếtit ồli ừ statement
63     }
64     $stmt->close();
65 }
66 } else {
67     $response['message'] = 'ừ ệliu ạtrng thái ắhoc ID ếthit ịb không ợhp ệl (
ếthiu ừtrng "status" ắhoc "device_id" ắhoc ắnh ắng sai).';
68     echo json_encode($response);
69     exit();
70 }
71 } else {
72     $response['message'] = 'Yêu ắc không ợhp ệl (ích ắchp ắnhn ườphng ứthc POST).';
73     echo json_encode($response);
74     exit();
75 }
76
77 $conn->close();
78 echo json_encode($response);
79 exit();
80 ?>

```

A.2.3 Các file API người dùng trong Login/api/users/

```

1 <?php
2 // Bao ồgm file ắc hình chung ắc ứng ứng (bao ồgm cài ắt ồli và CORS)
3 require_once '../.../TH_DO_AN/app_config.php'; // ỀIU ỈCHNH ỀỜNG ẮĐN NÀY CHO
CHÍNH XÁC
4
5 session_start(); // Session ắphi ắbt đầu SAU KHI header đã được ứgi ởbi app_config.
php ếnu session ID là trong cookie
6 header('Content-Type: application/json'); // Ắt header ắphn ồhi JSON
7
8 // ởKhi ắto ắphn ồhi ắmc ắnh là ứch ắng ắnhp
9 $response = ['status' => 'success', 'loggedIn' => false];
10
11 // ểKim tra ếnu các ếbin session ắc ườngi dùng đã được ếthit ắlp
12 if (isset($_SESSION['user_id']) && isset($_SESSION['username']) && isset($_SESSION['
role'])) {
13     $response['loggedIn'] = true;
14     $response['user_id'] = $_SESSION['user_id'];
15     $response['username'] = $_SESSION['username'];
16     $response['role'] = $_SESSION['role'];
17     // Có ểth thêm các thông tin khác ếnu ắcn cho frontend
18 }
19

```

```

20 echo json_encode($response);
21 exit(); // Đảm bảo thoát sau khi xử lý phản hồi
22 ?>

1 <?php
2 // Bao gồm file cấu hình chung của ứng dụng (bao gồm cài đặt ôli và CORS)
3 require_once '../..../TH_DO_AN/app_config.php'; // ĐIỀU KIỆN ĐƯỜNG LỐI NÀY CHO
   CHÍNH XÁC
4
5 session_start(); // Khởi đầu session để lưu trữ trạng thái đăng nhập. API SAU
   require_once app_config.php để tránh Headers already sent.
6 header('Content-Type: application/json'); // Đặt header phản hồi JSON. API SAU
   require_once app_config.php.
7
8 // Bao gồm file kết nối database chung
9 // Đường dẫn tương đối từ Login/api/users/ đến TH_DO_AN/db_connect.php
10 require_once '../..../TH_DO_AN/db_connect.php';
11
12 // Kiểm tra kết nối sau khi include db_connect.php (db_connect.php đã có exit() nếu
   ôli)
13 if (!isset($conn) || $conn->connect_error) {
14     error_log("Database connection error in login.php: " . ($conn->connect_error ??
   'Connection object not set.'));
15     echo json_encode(['status' => 'error', 'message' => 'Ôli kết nối CSDL: ' . ($conn
   ->connect_error ?? 'Không thể kết nối.')] );
16     exit();
17 }
18
19 // Mã hóa dữ liệu JSON từ request body
20 $data = json_decode(file_get_contents('php://input'), true);
21
22 $username = $data['username'] ?? '';
23 $password = $data['password'] ?? '';
24
25 // Kiểm tra dữ liệu đầu vào
26 if (empty($username) || empty($password)) {
27     echo json_encode(['status' => 'error', 'message' => 'Vui lòng nhập tên đăng nhập
   và mật khẩu.']);
28     exit();
29 }
30
31 // Chuẩn bị câu lệnh để lấy thông tin người dùng theo username
32 $stmt = $conn->prepare("SELECT id, username, password, role FROM users WHERE
   username = ?");
33
34 if ($stmt === false) {
35     error_log("Ôli chuẩn bị câu lệnh trong login.php: " . $conn->error);
36     echo json_encode(['status' => 'error', 'message' => 'Ôli hệ thống. Vui lòng thử
   lại sau.']);
37     exit();
38 }
39
40 $stmt->bind_param("s", $username); // Gán username vào câu lệnh
41 $stmt->execute(); // Thực thi câu lệnh
42 $result = $stmt->get_result(); // Lấy kết quả
43
44 if ($result->num_rows == 1) { // Nếu tìm thấy một người dùng
45     $user = $result->fetch_assoc(); // Lấy thông tin người dùng
46
47     // Xác minh mật khẩu đã hash với mật khẩu nhập vào
48     if (password_verify($password, $user['password'])) {
49         // Đăng nhập thành công, lưu thông tin người dùng vào session
50         $_SESSION['user_id'] = $user['id'];
51         $_SESSION['username'] = $user['username'];
52         $_SESSION['role'] = $user['role'];

```

```

53     $_SESSION['loggedin'] = true; // ÁT QUAN ỢTRNG ĐỂ index.php CỐ ỂTH ỀKIM
    TRA
54
55     echo json_encode([
56         'status' => 'success',
57         'message' => 'Đã đăng nhập thành công!',
58         'user_id' => $user['id'],
59         'username' => $user['username'],
60         'role' => $user['role']
61     ]);
62     exit(); // Thêm exit()
63 } else {
64     echo json_encode(['status' => 'error', 'message' => 'Tên đăng nhập học ật
    ả khu không đúng.']);
65     exit(); // Thêm exit()
66 }
67 } else {
68     echo json_encode(['status' => 'error', 'message' => 'Tên đăng nhập học ật ả khu
    không đúng.']);
69     exit(); // Thêm exit()
70 }
71
72 $stmt->close(); // ữNhg dòng này ẽs không được ực thi ếnu exit() đã được ọi ở trên
73 $conn->close(); // Tuy nhiên, ữgi ảli cho ườtrng ợhp ồlung code không ọi exit ởsm.
74 ?>

```

```

1 <?php
2 // Bao ồm file ầu hình chung ủa ứng ụng (bao ồm cài đặt ỗli và CORS)
3 // Đường ẫn này ảgi định logout.php ằm trong Login/api/users/
4 // và app_config.php ằm trong TH_DO_AN/
5 // ảCn đi lên 3 ấp (../../..) để đến htdocs/ sau đó đi óxung TH_DO_AN/
6 require_once '../..../TH_DO_AN/app_config.php';
7
8 // Đặt header Content-Type cho ảphn ồhi JSON
9 header('Content-Type: application/json');
10
11 // ữX lý preflight OPTIONS request (ảphi ĐẶT ƯỚTRC ọi logic khác và session_start
    ())
12 if ($_SERVER['REQUEST_METHOD'] === 'OPTIONS') {
13     http_response_code(200);
14     exit();
15 }
16
17 // iCh ấphp ậnhn ườphng ực POST cho đăng ấxut
18 if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
19     http_response_code(405); // Method Not Allowed
20     echo json_encode(['status' => 'error', 'message' => 'Yêu ầu không ợhp ệl. iCh
    ấphp ậnhn ườphng ực POST cho đăng ấxut.']);
21     exit();
22 }
23
24 session_start(); // ấBt đầu session sau khi ểkim tra method để( tránh ạto session
    không ầcn ểthit cho OPTIONS request)
25
26 // Xóa ấtt ắc các ểbin session
27 $_SESSION = array();
28
29 // Xóa cookie session ỏkhi trình ệduyt
30 // Điều này ất quan ợtrng để đảm ảbo cookie phiên không còn ồtn ạti
31 if (ini_get("session.use_cookies")) {
32     $params = session_get_cookie_params();
33     setcookie(session_name(), '', time() - 42000, // Đặt ờthi gian ểht ảhn trong quá
    ứkh
34         $params['path'], $params['domain'],
35         $params['secure'], $params['httponly']

```

```

36     );
37 }
38
39 // hủy session trên máy chủ
40 session_destroy();
41
42 echo json_encode(['status' => 'success', 'message' => 'Đăng xuất thành công.']);
43 exit(); // Đảm bảo thoát sau khi xử lý phản hồi
44 ?>

```

```

1 <?php
2 // Bao gồm file cấu hình chung của ứng dụng (bao gồm cài đặt lỗi và CORS)
3 require_once '../.../TH_DO_AN/app_config.php';
4
5 // Đặt header để trình duyệt biết đây là phản hồi JSON
6 header('Content-Type: application/json');
7
8 // Bao gồm file kết nối database chung
9 require_once '../.../TH_DO_AN/db_connect.php';
10
11 // Kiểm tra kết nối sau khi include db_connect.php
12 if (!isset($conn) || $conn->connect_error) {
13     error_log("Database connection error in register.php: " . ($conn->connect_error
14     ?? 'Connection object not set.));
15     echo json_encode(['status' => 'error', 'message' => 'Lỗi kết nối CSDL: ' . ($conn
16     ->connect_error ?? 'Không thể kết nối.']));
17     exit();
18 }
19
20 // Xử lý preflight OPTIONS request (phải ĐẶT ƯỚC ợm logic và session_start() nếu
21 // có)
22 if ($_SERVER['REQUEST_METHOD'] === 'OPTIONS') {
23     http_response_code(200);
24     exit();
25 }
26
27 // Kiểm tra phương thức yêu cầu
28 if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
29     http_response_code(405); // Method Not Allowed
30     echo json_encode(['status' => 'error', 'message' => 'Yêu cầu không hợp lệ. Chỉ
31     chấp nhận phương thức POST cho đăng ký.']);
32     exit();
33 }
34
35 // Giải mã dữ liệu JSON từ request body
36 $data = json_decode(file_get_contents('php://input'), true);
37
38 $username = $data['username'] ?? '';
39 $password = $data['password'] ?? '';
40
41 // Kiểm tra dữ liệu đầu vào (Validation mạnh mẽ)
42 if (empty($username) || empty($password)) {
43     echo json_encode(['status' => 'error', 'message' => 'Tên đăng nhập và mật khẩu
44     không được để trống.']);
45     exit();
46 }
47
48 // --- SERVER-SIDE VALIDATION THÊM CHO USERNAME ---
49 if (strlen($username) < 3 || strlen($username) > 30 || !preg_match('/^[a-zA-Z0-9_]+$
50 /', $username)) {
51     echo json_encode(['status' => 'error', 'message' => 'Tên đăng nhập phải từ 3-30
52     ký tự và chỉ chứa chữ cái, số, gạch dưới.']);
53     exit();
54 }
55
56 // --- SERVER-SIDE VALIDATION THÊM CHO PASSWORD ---

```

```

50 if (strlen($password) < 6) { // Ớtỉ ểthiu 6 ký ựt (ảphi ớkhp ớvi JS frontend)
51     echo json_encode(['status' => 'error', 'message' => 'ậMt ắkhu ảphi có ít ắnhт 6
    ký ựt.']);
52     exit();
53 }
54 // Có ểth thêm ểkim tra độ úphc ật ậMt ắkhu (nâng cao):
55 // if (!preg_match('/[A-Z]/', $password) || !preg_match('/[a-z]/', $password) || !
    preg_match('/[0-9]/', $password) || !preg_match('/[^A-Za-z0-9]/', $password)) {
56 //     echo json_encode(['status' => 'error', 'message' => 'ậMt ắkhu ảphi ứcha ít
    ắnhт ộmt ữch hoa, ộmt ữch ườthng, ộmt ốs và ộmt ký ựt đặc ệbit.']);
57 //     exit();
58 // }
59
60 // Hash ậMt ắkhu ướtrc khi ứlu vào database để ắbo ậMt
61 $password_hash = password_hash($password, PASSWORD_DEFAULT);
62
63 // ắMc định vai trò là 'user' cho ườngi dùng đăng ký
64 $role = 'user';
65
66 // ắChun ậ câu ệlnh SQL để tránh SQL Injection
67 $stmt = $conn->prepare("INSERT INTO users (username, password, role) VALUES (?, ?,
    ?)");
68
69 // ểKim tra xem câu ệlnh có được ắchun ậ thành công không
70 if ($stmt === false) {
71     error_log("ốLi ắchun ậ câu ệlnh trong register.php: " . $conn->error);
72     echo json_encode(['status' => 'error', 'message' => 'ốLi ệh ốthng. Vui lòng ứth
    ặli sau.']);
73     exit();
74 }
75
76 // ắGn tham ốs vào câu ệlnh đã ắchun ậ
77 $stmt->bind_param("sss", $username, $password_hash, $role);
78
79 // ựThc thi câu ệlnh
80 if ($stmt->execute()) {
81     echo json_encode(['status' => 'success', 'message' => 'Đăng ký thành công!']);
82     exit();
83 } else {
84     // ểKim tra ốli ếnu username đã ồtn ặti (ốli mã 1062)
85     if ($conn->errno == 1062) {
86         echo json_encode(['status' => 'error', 'message' => 'Tên đăng ậnhp đã ồtn ặti
    . Vui lòng ợchn tên khác.']);
87         exit();
88     } else {
89         error_log("ốLi ựthc thi câu ệlnh INSERT trong register.php: " . $stmt->error);
90         echo json_encode(['status' => 'error', 'message' => 'Đăng ký ắthт ặbi: Đã
    ắxy ra ốli không xác định.']);
91         exit();
92     }
93 }
94
95 // Đóng câu ệlnh và ểkt ốni database (các dòng này ếs không được ựthc thi ếnu exit()
    đã được ợgi ở trên)
96 $stmt->close();
97 $conn->close();
98 ?>

```

A.3 Mã nguồn Frontend (HTML/CSS/JavaScript)

Các file xây dựng giao diện người dùng web.

A.3.1 Các file chính trong TH_DO_AN/

```
1 <?php
2 // TH_DO_AN/index.php - Trang ười ứng dụng sau khi đăng ập
3
4 // Bao ồm file ầu hình chung ủa ứng dụng (bao ồm cài đặt ỗi và CORS)
5 require_once 'app_config.php'; // Đảm ả app_config.php ằm cùng ười
6
7 session_start(); // ấBt đầu session để truy ập thông tin đăng ập
8
9 // ểKim tra xem ười dùng đã đăng ập ư
10 if (!isset($_SESSION['loggedin']) || $_SESSION['loggedin'] !== true) {
11     // ểNu ư đăng ập ặ session không ợp ệ, ểchuyển ường ềv trang đăng ập
12     header('Location: ../Login/login_register.html');
13     exit();
14 }
15
16 ?>
17 <!DOCTYPE html>
18 <html lang="vi">
19 <head>
20     <meta charset="UTF-8">
21     <meta name="viewport" content="width=device-width, initial-scale=1.0">
22
23     <title>ệH ốThng Giám Sát ệNhit Độ và Độ Ẩm</title>
24     <link rel="stylesheet" href="main.css">
25     <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
26     <script src="https://cdn.jsdelivr.net/npm/luxon@3.x.x/build/global/luxon.min.js"></script>
27     <script src="https://cdn.jsdelivr.net/npm/chartjs-adapter-luxon@1.x.x"></script>
28
29 </head>
30 <body>
31     <header>
32         <div class="header-content-wrapper">
33             <div class="header-left-empty">
34
35             <div class="header-right-section">
36                 <div class="auth-buttons">
37                     <button id="logout-button">Đăng ấXut</button>
38                     <span id="welcome-message" style="color: black; margin-right: 10px;">
39                         >Xin chào, <span id="username-display">?php echo htmlspecialchars($_SESSION['
40                         username']); ?</span>!</span>
41                     </div>
42
43                     <nav id="main-nav">
44                         <ul>
45                             <li data-section="current-data" class="active"><a href="#">ữD
46                             ệliu ệHin ặTi</a></li>
47                             <li data-section="history-data"><a href="#">ịLch ữS ữD ệLiu</a></li>
48                             <li data-section="alerts"><a href="#">ấCnh Báo</a></li>
49                             <li data-section="user-management" class="?php echo ($_SESSION
50                             ['role'] == 'admin') ? '' : 'hidden'; ?>"><a href="#">ấQun Lý ườNgì Dùng</a></li>
51                             <li data-section="settings"><a href="#">Thông tin ệthit ịb</a></li>
52                         </ul>
53                     </nav>
54                 </div>
55             </div>
56 </header>
57
58 <main>
59     <section id="current-data" class="active-section">
```

```

56         <div class="current-data-header-controls">
57             <div class="device-selection-main">
58                 <label for="device-selector" class="block text-gray-700 text-sm
font-bold mb-2">ỌChn ếthit ịb:</label>
59                 <select id="device-selector" class="shadow appearance-none
border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none
focus:shadow-outline">
60                     <option value="">-- ọChn ếthit ịb --</option>
61                 </select>
62             </div>
63
64             <div class="dht-toggle-control">
65                 <h3>Đềiu ểkhin ẩCm ếbin DHT</h3>
66                 <button id="toggleDhtButton" class="dht-on">Đang đo</button>
67             </div>
68         </div>
69         <h2>ũD ệliu ệHin ạTi</h2>
70         <div class="current-data-grid">
71             <div class="data-widget temperature-widget">
72                 <div class="widget-header">ệNhit độ</div>
73                 <div class="widget-content">
74                     <span id="currentTemperatureValue" class="data-value">N/A</
span>
75                     <span id="currentTemperatureUnit" class="data-unit"></span>
76                 </div>
77             </div>
78             <div class="data-widget humidity-widget">
79                 <div class="widget-header">Độ ẩm</div>
80                 <div class="widget-content">
81                     <span id="currentHumidityValue" class="data-value">N/A</span>
82                     <span id="currentHumidityUnit" class="data-unit">%</span>
83                 </div>
84             </div>
85         </div>
86         <div class="data-item timestamp-item">
87             <span class="label">ậCp ậnhit lúc:</span>
88             <span id="lastUpdatedTime" class="value">N/A</span>
89         </div>
90
91         <div class="settings-grid">
92         <div class="setting-group">
93             <h3>Đơn ịV Đo</h3>
94             <label for="temperatureUnitSelect">Đơn ịv ệnhit độ:</label>
95             <select id="temperatureUnitSelect">
96                 <option value="C">°C</option>
97                 <option value="F">°F</option>
98             </select>
99         </div>
100
101         <div class="setting-group">
102             <h3>ầTn ấSut ậCp ậNht</h3>
103             <label for="update-frequency">ầTn ấsut (giây):</label>
104             <input type="number" id="update-frequency" value="5">
105             <button id="save-frequency">Save</button>
106         </div>
107     </div>
108     <div id="current-data-message" class="info-message"></div>
109 </section>
110
111     <section id="history-data" class="hidden-section">
112         <h2>ịLch ẩS ẩD ệLiu</h2>
113         <div class="controls">
114             <label for="history-period">ọChn ẩkhong ờthi gian:</label>
115             <select id="history-period">
116                 <option value="1h">l ờgi qua</option>

```

```

117         <option value="24h">24 ờgi qua</option>
118         <option value="7d">7 ngày qua</option>
119         <option value="30d">30 ngày qua</option>
120         <option value="custom">Tùy ỉchnh</option>
121     </select>
122     <div id="custom-period" class="hidden">
123         <label for="start-date">ừT ngày:</label>
124         <input type="date" id="start-date">
125         <label for="end-date">Đến ngày:</label>
126         <input type="date" id="end-date">
127         <button id="apply-custom-period">Áp ụngng</button>
128     </div>
129     <button id="export-csv">ắXut CSV</button>
130 </div>
131 <div id="history-chart">
132     <canvas id="temperatureHumidityChart"></canvas>
133 </div>
134 <div id="history-table">
135     <table class="data-table">
136         <thead>
137             <tr>
138                 <th>ờThi gian</th>
139                 <th>ệNhit độ (<span id="table-temperature-unit-header">°C
140
141                 <th>Độ ẩm (%)</th>
142             </tr>
143         </thead>
144         <tbody id="history-table-body">
145             </tbody>
146         </table>
147     </div>
148     <div id="history-data-message" class="info-message"></div>
149 </section>
150 <section id="alerts" class="hidden-section">
151     <h2>ắCnh Báo</h2>
152     <div class="alert-settings">
153         <h3>ếThit ậlp ưỡngng</h3>
154         <div class="setting-item">
155             <label for="temp-threshold">ưỡNgng ệnhit độ (°C):</label>
156             <input type="number" id="temp-threshold">
157         </div>
158         <div class="setting-item">
159             <label for="humidity-threshold">ưỡNgng độ ẩm (%):</label>
160             <input type="number" id="humidity-threshold">
161         </div>
162         <button id="save-thresholds">ữLu ưỡNgng</button>
163     </div>
164     <div class="alert-status">
165         <h3>ạTrng thái ắcnh báo</h3>
166         <div id="temperature-alert" class="alert">ệNhit độ: <span id="temp-
167 alert-status">Bình ườthng</span></div>
168         <div id="humidity-alert" class="alert">Độ ẩm: <span id="humidity-
169 alert-status">Bình ườthng</span></div>
170     </div>
171     <div class="notification-settings">
172         <h3>Thông báo</h3>
173         <label>
174             <input type="checkbox" id="enable-email-alerts"> ửGi thông báo
175             qua Email
176         </label>
177         <div id="email-settings" class="hidden">
178             <label for="email-address">Địa ỉch Email:</label>
179             <input type="email" id="email-address">
180             <button id="save-email-settings">ữLu Email</button>
181         </div>

```



```

179     </div>
180 </section>
181
182 <section id="settings" class="hidden-section">
183     <h2>Thông Tin Kỹ Thuật về Sản phẩm</h2>
184     <div class="component-info">
185         <h3>DHT11 - Cảm biến nhiệt độ và độ ẩm</h3>
186         <p>DHT11 là một cảm biến nhiệt độ và độ ẩm thường được sử dụng đi kèm
        với một NTC chuyên dụng để đo nhiệt độ và một bộ vi điều khiển 8 bit để xử lý ra các
        giá trị nhiệt độ và độ ẩm sử dụng bộ điều khiển. </p>
187         <div class="image-container">
188             
190             </div>
191             <h4>Định dạng sơ đồ chân DHT11 và cấu hình:</h4>
192             <ul>
193                 <li><strong>VCC:</strong> Nguồn điện 3.5V đến 5.5V</li>
194                 <li><strong>Data:</strong> Đầu ra analog nhiệt độ và Độ ẩm thông qua
195                 bộ điều khiển</li>
196                 <li><strong>Ground:</strong> Kết nối với mặt đất của mạch</li>
197             </ul>
198             <h4>Thông số kỹ thuật DHT11:</h4>
199             <ul>
200                 <li><strong>Điện áp hoạt động:</strong> 3.5V đến 5.5V</li>
201                 <li><strong>Dòng hoạt động:</strong> 0,3mA (ở 60uA (chênh lệch 0))</li>
202                 <li><strong>Đầu ra:</strong> Bộ điều khiển</li>
203                 <li><strong>Phạm vi nhiệt độ:</strong> 0 ° C đến 50 ° C</li>
204                 <li><strong>Phạm vi độ ẩm:</strong> 20% đến 90%</li>
205                 <li><strong>Độ phân giải:</strong> Nhiệt độ và Độ ẩm đều là 16-
206                 bit</li>
207                 <li><strong>Độ chính xác:</strong> ± 1 ° C và ± 1%</li>
208             </ul>
209         </div>
210         <div class="component-info">
211             <h3>Module Vi điều khiển ESP32</h3>
212             <p>ESP32 là một bộ vi điều khiển thuộc danh mục vi điều khiển trên
            chip công suất thấp và tiết kiệm chi phí. Module hỗ trợ các cảm biến ESP32 đều
            tích hợp Bluetooth và Wi-Fi tích hợp kép, làm cho nó có tính linh hoạt cao, mạnh mẽ
            và đáng tin cậy cho nhiều ứng dụng. Nó là lựa chọn lý tưởng cho vi điều khiển NodeMCU
            ESP8266 phổ biến và cung cấp hiệu suất và tính năng vượt trội. Bộ vi điều khiển
            ESP32 được sản xuất bởi Espressif Systems và được sử dụng rộng rãi trong nhiều ứng
            dụng khác nhau như IoT, robot và tự động hóa. </p>
213             <p>ESP32 cũng được thiết kế để tiêu thụ điện năng thấp, lý tưởng cho
            các ứng dụng chạy bằng pin. Nó có hệ thống quản lý năng lượng cho phép nó hoạt
            động ở chế độ ngủ và kích hoạt khi cần thiết, điều này có thể kéo dài tuổi thọ
            pin rất nhiều. </p>
214             <div class="image-container">
215                 
217                 </div>
218             <h4>Sơ đồ mạch:</h4>
219             <div class="image-container">
220                 
222                 </div>
223             </div>
224         </section>
225
226 <section id="user-management" class="hidden-section">
227     <h2>Quản Lý Người Dùng</h2>
228 </section>
229
230 </main>
231
232 <footer id="bottom-header"> <div class="bottom-header-content">

```

```

229     <p>&copy; 2025 ệH ốThng Giám Sát ệNhit Độ và Độ Ẩm</p>
230   </div>
231 </footer>
232
233   <script src="main.js"></script>
234 </body>
235 </html>

```

```

1  html {
2    width: 100%;
3    height: 100%;
4    box-sizing: border-box;
5  }
6  *, *::before, *::after {
7    box-sizing: inherit;
8  }
9
10 body {
11   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
12   margin: 0;
13   width: 100vw; /* ếChim đúng 100% ềchiu ợrng khung nhìn */
14   overflow-x: hidden; /* ẶNGN THANH ỘCUN NGANG */
15   background-color: #f8f9fa;
16   color: #495057;
17   display: flex;
18   flex-direction: column;
19   min-height: 100vh;
20   /* box-sizing: border-box; // Đã được ửx lý ởbi * selector */
21 }
22
23 /* Các ếbin Font Awesome (ếnu ậbn ửs ụngng Font Awesome) */
24 :root {
25   --fa-style-family-brands: 'Font Awesome 6 Brands';
26   --fa-font-brands: normal 400 1em/1 'Font Awesome 6 Brands';
27   --fa-font-regular: normal 400 1em/1 'Font Awesome 6 Free';
28   --fa-style-family-classic: 'Font Awesome 6 Free';
29   --fa-font-solid: normal 900 1em/1 'Font Awesome 6 Free';
30 }
31
32 /* Trong main.css */
33
34 header {
35   /* Các ộthuc tính background ắy ừt ạon ậbn cung ắp */
36   background: url(Images/bg_menu-scaled.jpg) no-repeat center; /* Ắm ắo đường ắdn
    ảnh ược ộb ắa ậbn là đúng */
37   background-size: 100%; /* ửSa ổi: Ắt kích ướthc ềnn là 100% */
38   /* background-position: center center; // ửGi nguyên ếnu ậbn ốmun ảnh ềnn được ắcn
    ửgia */
39   /* background-repeat: no-repeat; // Đã bao ồgm trong shorthand 'background' */
40
41   /* Các ộthuc tính khác ắy ừt ạon ậbn cung ắp */
42   color: #444; /* ửSa ổi: Màu ữch ốtng ếth ắa header */
43   /* font: 14px "roboto" !important; // ắCn ậthn ớvi !important, có ếth gây khó
    ắkhn khi debug */
44   /* ếNu ậbn ốmun ắp ụngng font này cho toàn ộb header, hãy thêm vào đây */
45   font-family: "roboto", sans-serif;
46   font-size: 14px;
47
48   /* Các ộthuc tính ửgi nguyên ừt ắcu hình ướtrc đó ắa ậbn để duy trì ốb ược
    Flexbox */
49   padding: 0.8rem 20px;
50   box-shadow: 0 0.5rem 1rem rgba(0, 0, 0, 0.1);
51   width: 100vw; /* Ắm ắo header ếchim đúng 100% ềchiu ợrng khung nhìn */
52   display: flex;
53   flex-direction: column;
54   align-items: center;

```

```

55
56 /* Các ôthuc tính dành riêng cho webkit (mobile/trình ệduyt ục ểth) */
57 -webkit-text-size-adjust: 100%;
58 -webkit-tap-highlight-color: transparent;
59
60 /* box-sizing: border-box; // Đã được ửx lý ởbi * selector */
61 /* display: block; // Header đã là display:flex, không ần đổi thành block */
62 }
63
64 /* Container chính bao ọc 1/3 và 2/3 */
65 .header-content-wrapper {
66     display: flex; /* ểBin nó thành Flex container ngang */
67     width: 100%; /* ểChim toàn ộb ềchiu ộrng ủa header */
68     align-items: flex-start; /* ểCh ỉchnh ộni dung lên phía trên ủa các ộct */
69     padding: 10px 0; /* ểKhông cách bên trong wrapper */
70 }
71
72 /* Khu ực 1 TRÁI (1/3), ốtrng */
73 .header-left-empty {
74     flex: 0 0 33.33%; /* Không co (0), không giãn (0), ểchim đúng 1/3 ềchiu ộrng */
75     /* background-color: rgba(255, 0, 0, 0.05); /* Dùng để debug, sau đó xóa */
76     /* min-height: 50px; /* ậN có ểth ần ểhit ập ộmt ềchiu cao ốti ểthiu */
77 }
78
79 /* Khu ực 2 ẬPHI (2/3), ứa auth-buttons và nav */
80 .header-right-section {
81     flex: 1 1 66.67%; /* Co (1), giãn (1), ểchim 2/3 ềchiu ộrng còn ại */
82     display: flex; /* ểBin nó thành Flex container ọc */
83     flex-direction: column; /* ểXp ồchng auth-buttons và nav */
84     align-items: flex-end; /* ểCh ỉchnh ộni dung (auth-buttons và nav) sang ậphi */
85     /* background-color: rgba(0, 255, 0, 0.05); /* Dùng để debug, sau đó xóa */
86 }
87
88 #welcome-message {
89     color: white;
90     margin-right: 10px;
91     white-space: nowrap; /* ểĐảm ảo ắv ắb không ốxung dòng */
92     overflow: hidden; /* ểRt quan ộtrng: Ắn ắphn tràn ủa ắv ắb */
93     text-overflow: ellipsis; /* THÊM DÒNG NÀY: Để thêm ắdu ... ển ắv ắb quá dài */
94     max-width: calc(100% - 70px); /* ểGii ắh ềchiu ộrng ốti ủa ắthông báo chào
95                                     (100% ủa auth-buttons ừtr đi ắkhong 70px cho
96                                     nút ắđng ắxut) */
97     display: block; /* ểCh display: block ặhoc inline-block để overflow và text-
98     overflow ắhot động */
99 }
100
101 #username-display {
102     color: rgb(0, 0, 0); /* Màu ứch cho tên ườngi dùng, làm ốni ắbt */
103     font-weight: bold;
104     font-size: 1rem; /* ểĐi ỉchnh kích ướthc font cho phù ộhp */
105 }
106
107 .auth-buttons {
108     display: flex;
109     justify-content: flex-end; /* ểCh ộni dung bên trong nó sang ậphi */
110     align-items: center;
111     width: 100%; /* ểRt quan ộtrng: để nó kéo dài ểht ềchiu ộrng ủa header-right-
112     section */
113     margin-bottom: 10px; /* ểKhông cách ớvi menu nav bên ướdi */
114     white-space: nowrap; /* ểNg không cho ứch ốxung dòng */
115 }
116
117 .auth-buttons button {
118     padding: 0.5rem 1rem;

```

```

116     margin-left: 0.5rem;
117     background-color: #007bff;
118     color: white;
119     border: none;
120     border-radius: 0.25rem;
121     cursor: pointer;
122     transition: background-color 0.3s ease;
123 }
124
125 .auth-buttons button:first-child {
126     margin-left: 0;
127 }
128
129 .auth-buttons button:hover {
130     background-color: #0056b3;
131 }
132
133 nav#main-nav {
134     width: 100%; /* Quan trọng: để nó kéo dài hết chiều rộng của header-right-section */
135     background-color: rgba(255, 255, 255, 0.2); /* Nên cho menu */
136     border-radius: 5px;
137     padding: 5px 0;
138 }
139
140 nav#main-nav ul {
141     display: flex;
142     list-style: none;
143     padding: 0;
144     margin: 0;
145     justify-content: space-around; /* Chia đều các mục menu */
146     width: 100%; /* Đảm bảo ul kéo dài hết chiều rộng của nav */
147 }
148
149 nav#main-nav ul li {
150     margin: 0;
151 }
152
153 nav#main-nav ul li a {
154     color: black; /* Màu mặc định cho các mục menu */
155     text-decoration: none;
156     transition: color 0.3s ease;
157     font-size: 0.9rem;
158     font-weight: 500;
159     padding: 5px 10px; /* Thêm padding để dễ bấm */
160     display: block; /* Để padding hoạt động */
161 }
162
163 nav#main-nav ul li a:hover,
164 nav#main-nav ul li.active a {
165     color: #f8c291; /* Màu khi hover hoặc active */
166     font-weight: bold;
167 }
168
169 /* == KẾT THÚC ĐIỀU KIỆN CÁC QUYẮC == */
170
171
172 main {
173     padding: 20px;
174     flex-grow: 1;
175     max-width: 1200px;
176     width: 90%;
177     margin: 0 auto;
178 }
179
180 main section {

```

```

181     background-color: white;
182     padding: 20px;
183     margin-bottom: 20px;
184     border-radius: 8px;
185     box-shadow: 0 0.125rem 0.25rem rgba(0, 0, 0, 0.075);
186 }
187
188 h2, h3 {
189     color: #343a40;
190     margin-bottom: 15px;
191 }
192
193 .data-item {
194     margin-bottom: 15px;
195     font-size: 1.2rem;
196 }
197
198 .label {
199     font-weight: bold;
200     margin-right: 10px;
201     color: #343a40;
202 }
203
204 .value {
205     font-weight: normal;
206     color: #007bff;
207 }
208
209 .unit {
210     margin-left: 5px;
211     color: #6c757d;
212 }
213
214 .controls {
215     margin-bottom: 20px;
216     display: flex;
217     flex-wrap: wrap;
218     align-items: center;
219     gap: 10px;
220 }
221
222 .controls label {
223     margin-right: 5px;
224     font-weight: bold;
225 }
226
227 .controls select, .controls input[type="date"] {
228     padding: 0.75rem;
229     border: 1px solid #ced4da;
230     border-radius: 0.25rem;
231 }
232
233 .controls button {
234     padding: 0.75rem 1.5rem;
235     background-color: #28a745;
236     color: white;
237     border: none;
238     border-radius: 0.25rem;
239     cursor: pointer;
240     transition: background-color 0.3s ease;
241 }
242
243 .controls button:hover {
244     background-color: #218838;
245 }
246
247 .alert-settings .setting-item, .setting-group label {

```

```

248     display: block;
249     margin-bottom: 10px;
250     font-weight: bold;
251 }
252
253 .alert-settings input[type="number"], .setting-group input[type="number"],
254 .alert-settings input[type="email"], #user-management input[type="text"],
255 #user-management input[type="password"], .setting-group select {
256     padding: 0.75rem;
257     width: 250px;
258     border: 1px solid #ced4da;
259     border-radius: 0.25rem;
260     margin-bottom: 15px;
261 }
262
263 .alert-status .alert {
264     padding: 1rem;
265     margin-bottom: 10px;
266     border-radius: 0.25rem;
267     font-weight: bold;
268 }
269
270 .alert-status #temperature-alert {
271     background-color: #ffc107;
272     color: #212529;
273 }
274
275 .alert-status #humidity-alert {
276     background-color: #17a2b8;
277     color: white;
278 }
279
280 .hidden-section {
281     display: none;
282 }
283
284 .active-section {
285     display: block;
286 }
287
288 #history-chart, /* ữGi nguyên #history-chart để nó được ăn ữgia cùng óvi #history-
289   table */
290 #history-table {
291     margin-top: 20px; /* ảKhông cách ừt ằphn ừt phía trên (ví ụđ: ểbiu đồ) */
292     max-width: 900px; /* óGii ảhn ềchiu ộrng óti đa ủa container ảbng */
293     margin-left: auto; /* ẮCN ỮGIA ỀNN ẢBNG THEO ỀCHIU NGANG */
294     margin-right: auto; /* ẮCN ỮGIA ỀNN ẢBNG THEO ỀCHIU NGANG */
295     padding: 15px; /* ảKhông cách bên trong container ảbng */
296     border: 1px solid #ddd; /* Đường ềvin ệnh cho container */
297     border-radius: 8px; /* Bo tròn góc */
298     box-shadow: 0 2px 5px rgba(0,0,0,0.1); /* Đổ bóng ệnh */
299     background-color: #f8f9fa; /* Màu ềnn cho container ảbng */
300 }
301
302 #user-management label {
303     display: block;
304     margin-bottom: 8px;
305     font-weight: bold;
306 }
307
308 #user-management button {
309     padding: 0.75rem 1.5rem;
310     background-color: #007bff;
311     color: white;
312     border: none;
313     border-radius: 0.25rem;

```

```

313     cursor: pointer;
314     margin-top: 15px;
315     transition: background-color 0.3s ease;
316 }
317
318 #user-management button:hover {
319     background-color: #0056b3;
320 }
321
322 #device-management label {
323     display: block;
324     margin-bottom: 8px;
325     font-weight: bold;
326 }
327
328 #device-management select {
329     padding: 0.75rem;
330     width: 250px;
331     border: 1px solid #ced4da;
332     border-radius: 0.25rem;
333     margin-bottom: 15px;
334 }
335
336 footer {
337     text-align: center;
338     padding: 0.5rem 0;
339     background-color: #a1d4dc;
340     color: white;
341     border-radius: 0.25rem;
342     margin-top: 20px;
343     box-shadow: 0 -0.125rem 0.25rem rgba(0, 0, 0, 0.075);
344 }
345
346 /* CSS cho ôb ỳc ứli ủa các widget ỹ ệliu ệhin ạti */
347 .current-data-grid {
348     display: grid;
349     grid-template-columns: 1fr 1fr;
350     gap: 30px;
351     margin-bottom: 20px;
352 }
353
354 .data-widget {
355     border-radius: 5px;
356     overflow: hidden;
357     box-shadow: 0 0.1rem 0.2rem rgba(0, 0, 0, 0.05);
358 }
359
360 .widget-header {
361     background-color: #1163d0;
362     color: white;
363     padding: 10px;
364     text-align: center;
365     font-weight: bold;
366     font-size: 1.8rem;
367 }
368
369 .widget-content {
370     background-color: white;
371     padding: 40px;
372     text-align: center;
373     display: flex;
374     justify-content: center;
375     align-items: baseline;
376     flex-direction: row;
377 }
378
379 .data-value {

```

```

380     font-weight: bold;
381     color: #14f458;
382     font-size: 4rem;
383     margin-right: 5px;
384     margin-bottom: 0;
385 }
386
387 .data-unit {
388     color: #000000;
389     font-size: 2rem;
390 }
391
392 /* Điều chỉnh kiểu dáng cho ume timestamp */
393 .timestamp-item {
394     font-size: 1rem;
395     color: #495057;
396     text-align: center;
397 }
398
399 .timestamp-item .label {
400     font-weight: bold;
401     margin-right: 5px;
402 }
403
404 .timestamp-item .value {
405     font-weight: normal;
406     color: #6c757d;
407 }
408
409 /* ===== */
410 /* Styles for History Chart Section */
411 /* ===== */
412
413 #history-chart {
414     max-width: 900px;
415     margin: 20px auto;
416     padding: 15px;
417     border: 1px solid #ddd;
418     border-radius: 8px;
419     box-shadow: 0 2px 5px rgba(0,0,0,0.1);
420     height: 400px;
421     display: flex;
422     justify-content: center;
423     align-items: center;
424 }
425
426 #temperatureHumidityChart {
427     width: 100% !important;
428     height: 100% !important;
429 }
430
431 /* Styles cho thông báo khi không có dữ liệu */
432 .no-data-message, .error-message {
433     text-align: center;
434     color: #555;
435     font-size: 1.1em;
436     padding: 20px;
437 }
438
439 .error-message {
440     color: red;
441     font-weight: bold;
442 }
443
444 .hidden {
445     display: none !important;
446 }

```



```

447
448 /* Optional: Media Queries for better responsiveness */
449 @media (max-width: 768px) {
450     header {
451         padding: 0.5rem 10px;
452     }
453
454     /* Đảm bảo luôn là flex-direction: row để ữngi trên cùng ột hàng */
455     .header-content-wrapper { /* ỬA CÁI NÀY THAY VÌ .header-top */
456         flex-direction: column; /* ểXp ềchng các ột 1/3 và 2/3 */
457         align-items: flex-start;
458     }
459
460     .header-left-empty {
461         /* ỏB flex: 0 0 33.33%; để nó không ểchim không gian trên mobile */
462         flex: auto; /* Cho phép nó co giãn */
463         width: 100%; /* ăHoc auto */
464         min-height: 0; /* Đảm bảo không ểchim ềchiu cao không ần ếthit */
465     }
466
467     .header-right-section {
468         flex-direction: column; /* ểXp ềchng auth-buttons và nav */
469         align-items: center; /* ăCn ữgia auth-buttons và nav */
470         width: 100%; /* ểChim toàn ột ềchiu ộrng */
471         flex: auto; /* Cho phép co giãn */
472     }
473
474     .auth-buttons {
475         margin-top: 10px;
476         width: 100%;
477         justify-content: center; /* ăCn ữgia trên mobile */
478     }
479
480     nav#main-nav {
481         width: 100%;
482         margin-left: 0;
483         margin-top: 10px;
484     }
485
486     nav#main-nav ul {
487         flex-direction: column;
488         align-items: center;
489     }
490
491     nav#main-nav ul li {
492         margin-bottom: 10px;
493     }
494
495     main {
496         width: 95%; /* ăM ộrng main trên màn hình ỏnh */
497         padding: 10px;
498     }
499
500     .current-data-grid {
501         grid-template-columns: 1fr; /* Stack columns on small screens */
502         gap: 20px; /* ăGim gap cho màn hình ỏnh */
503     }
504
505     .widget-content {
506         padding: 20px; /* ăGim padding cho ộni dung widget */
507     }
508
509     .data-value, .data-unit {
510         font-size: 3rem; /* ăGim kích ướthc font cho giá ịtr và đơn ịv */
511     }
512

```

```

513     .controls {
514         flex-direction: column;
515         align-items: flex-start;
516     }
517
518     .controls select, .controls input[type="date"], .controls button {
519         width: 100%;
520         margin-right: 0;
521         margin-bottom: 10px;
522     }
523
524     .alert-settings input[type="number"], .setting-group input[type="number"],
525     .alert-settings input[type="email"], #user-management input[type="text"],
526     #user-management input[type="password"], .setting-group select,
527     #device-management select {
528         width: 100%;
529     }
530 }
531
532 .settings-grid {
533     display: flex;
534     justify-content: space-around;
535     align-items: flex-start;
536     flex-wrap: wrap;
537     gap: 20px;
538     margin-top: 20px;
539 }
540
541 .settings-grid .setting-group {
542     flex: 1;
543     min-width: 280px;
544     background-color: #f8f9fa;
545     padding: 15px;
546     border-radius: 8px;
547     box-shadow: 0 0.125rem 0.25rem rgba(0, 0, 0, 0.075);
548 }
549
550 /* Điều chỉnh thêm cho input và button trong setting-group để chúng hiển thị tốt hơn
551 */
552 .settings-grid .setting-group input[type="number"],
553 .settings-grid .setting-group select {
554     width: calc(100% - 16px);
555     max-width: 250px;
556     box-sizing: border-box;
557 }
558
559 .settings-grid .setting-group button {
560     width: auto;
561     padding: 0.75rem 1.5rem;
562     margin-top: 10px;
563     background-color: #007bff;
564     color: white;
565     border: none;
566     border-radius: 0.25rem;
567     cursor: pointer;
568     transition: background-color 0.3s ease;
569 }
570
571 .settings-grid .setting-group button:hover {
572     background-color: #0056b3;
573 }
574
575 #bottom-header {
576     display: flex;
577     flex-direction: column;
578     align-items: stretch;
579     background: rgb(151, 221, 228);

```

```

579     color: white;
580     padding: 1rem 20px;
581     box-shadow: 0 -0.5rem 1rem rgba(0, 0, 0, 0.1);
582     min-height: 50px;
583     margin-top: 20px;
584 }
585
586 .bottom-header-content {
587     text-align: center;
588     padding: 10px 0;
589 }
590
591 #history-table {
592     display: block; /* Đảm bảo nó là một khối riêng biệt để auto margin */
593     height: auto; /* Cho phép chiều cao tự động co giãn theo nội dung bảng */
594     overflow-x: auto; /* Alert quan trọng: Thêm thanh cuộn ngang nếu bảng quá rộng trên
595     màn hình nhỏ */
596 }
597
598 /* Trong main.css (sau quy tắc #history-table hoặc @media đó) */
599 #history-table h3 {
600     text-align: center; /* Căn giữa tiêu đề */
601     margin-bottom: 15px; /* Không cách với tiêu đề */
602     color: #343a40; /* Màu chữ */
603 }
604
605 /* Trong main.css */
606 .history-data-table { /* Đây là class áp dụng cho thẻ <table> */
607     width: 100%; /* Đảm bảo bảng chiếm toàn bộ chiều rộng của #history-table
608     container */
609     border-collapse: collapse; /* Gộp các đường biên của ô */
610     margin-top: 20px; /* Không cách với tiêu đề "Ước lượng thô" (nếu có) */
611     font-size: 0.9rem; /* Điều chỉnh kích thước font cho toàn bộ bảng */
612 }
613
614 /* Định nghĩa tiêu đề cột (<th>) và ô dữ liệu (<td>) */
615 .history-data-table th,
616 .history-data-table td {
617     border: 1px solid #e9ecef; /* Màu đường biên nhạt hơn cho các ô */
618     padding: 12px 15px; /* Tăng padding để có thêm không gian giữa nội dung và đường
619     biên */
620
621     text-align: center; /* Căn giữa nội dung trong tất cả các ô (th và td) */
622     white-space: nowrap; /* Ngăn chữ tràn/ổn định nội dung, giúp bảng gọn hơn (nếu
623     có nội dung ngang nếu tràn) */
624 }
625
626 /* Định nghĩa riêng cho hàng tiêu đề của bảng */
627 .history-data-table th {
628     background-color: #f1f3f5; /* Màu header bảng */
629     font-weight: bold; /* Chữ đậm */
630     color: #495057; /* Màu chữ header */
631     text-transform: uppercase; /* Chữ hoa cho tiêu đề */
632     letter-spacing: 0.05em; /* Không cách giữa các chữ cái */
633 }
634
635 /* Định nghĩa riêng cho các ô dữ liệu */
636 .history-data-table td {
637     color: #343a40; /* Màu chữ mặc định cho ô dữ liệu */
638 }
639
640 /* Thay đổi màu nền cho các hàng lẻ để dễ đọc hơn (striped rows) */
641 .history-data-table tbody tr:nth-child(even) {
642     background-color: #f8f9fa;
643 }

```

```

641 /* Thay đổi màu nền khi di chuột qua hàng */
642 .history-data-table tbody tr:hover {
643     background-color: #e2e6ea;
644 }
645
646 /* Điều chỉnh độ rộng tương đối của từng cột (có thể tinh chỉnh lại các giá trị % này)
    */
647 .history-data-table th:nth-child(1), /* Cột Thời gian */
648 .history-data-table td:nth-child(1) {
649     width: 35%;
650 }
651 .history-data-table th:nth-child(2), /* Cột Nhiệt độ */
652 .history-data-table td:nth-child(2) {
653     width: 30%;
654 }
655 .history-data-table th:nth-child(3), /* Cột Độ ẩm */
656 .history-data-table td:nth-child(3) {
657     width: 35%;
658 }
659
660 /* Để bố cục "Chọn thiết bị" và "Điều khiển cảm biến DHT" trên cùng một hàng */
661 .current-data-header-controls {
662     display: flex;
663     justify-content: space-between; /* Đẩy các phần tử ra hai bên */
664     align-items: flex-end; /* Căn chỉnh các mục theo phía dưới */
665     margin-bottom: 20px; /* Thêm khoảng cách phía dưới nếu cần */
666     flex-wrap: wrap; /* Cho phép các phần tử xếp chồng trên màn hình nhỏ */
667 }
668
669 .device-selection-main {
670     flex: 1; /* Cho phép phần chọn thiết bị chiếm không gian */
671     max-width: 300px; /* Giới hạn chiều rộng để không quá lớn */
672     margin-right: 20px; /* Khoảng cách giữa hai phần tử */
673 }
674
675 .dht-toggle-control {
676     text-align: right; /* Căn nút sang phải trong khung của nó */
677 }
678
679 .dht-toggle-control h3 {
680     margin-bottom: 5px; /* Giảm khoảng cách giữa tiêu đề và nút */
681     font-size: 1em; /* Điều chỉnh kích thước tiêu đề nếu cần */
682     color: #555;
683 }
684
685 /* Styles cho nút bật/tắt */
686 #toggleDhtButton {
687     padding: 10px 20px;
688     font-size: 16px;
689     border: none;
690     border-radius: 5px;
691     cursor: pointer;
692     transition: background-color 0.3s ease;
693     white-space: nowrap; /* Ngăn không cho chữ xếp chồng dòng */
694 }
695
696 #toggleDhtButton.dht-on {
697     background-color: #28a745; /* Màu xanh lá cây cho trạng thái bật */
698     color: white;
699 }
700
701 #toggleDhtButton.dht-off {
702     background-color: #dc3545; /* Màu đỏ cho trạng thái tắt */
703     color: white;
704 }
705

```

```

706 /* Optional: Media Queries for better responsiveness */
707 @media (max-width: 768px) {
708     .current-data-header-controls {
709         flex-direction: column; /* Ếp ồchng trên màn hình ỏnh */
710         align-items: flex-start;
711     }
712     .device-selection-main {
713         width: 100%; /* ẾChim toàn ộp ềchiu ộrng */
714         max-width: none;
715         margin-right: 0;
716         margin-bottom: 15px; /* ảKhông cách ữgia select và nút */
717     }
718     .dht-toggle-control {
719         width: 100%;
720         text-align: left; /* ăCn trái trên mobile */
721     }
722     #toggleDhtButton {
723         width: 100%; /* Nút ếchim toàn ộp ềchiu ộrng */
724     }
725 }
726
727 /* Trong main.css */
728
729 /* Định ãngha ệhiu ứng ấnhp nháy cho ấcnh báo */
730 @keyframes warning-glow {
731     0%, 100% {
732         box-shadow: 0 0 5px rgba(255, 0, 0, 0.5), /* Bóng ờm ệnh ban đầu */
733                     0 0 5px rgba(255, 0, 0, 0.5) inset;
734         border-color: #1163d0; /* Màu ềvin bình ườthng ủa widget, ặhoc màu ặbn ốmun */
735     }
736     50% {
737         box-shadow: 0 0 20px red, /* Bóng ờm ặmnh ớhn ở ữgia chu ỳk */
738                     0 0 20px red inset;
739         border-color: red; /* Màu ềvin ấcnh báo */
740     }
741 }
742
743 /* Class để áp ựdng ệhiu ứng cho các widget khi có ấcnh báo */
744 .data-widget.warning-active {
745     animation: warning-glow 1s infinite alternate; /* 'alternate' để animation ượmt
746     /* Có ểth thêm border-color ban đầu cho ặtrng thái ấcnh báo ếnu ốmun ỏni ặbt ớhn */
747     border: 4px solid #1163d0; /* ữGi nguyên border ệhin ặti ặhoc làm nó rõ ràng ớhn */
748 }
749
750 /* ẾNu ặbn ốmun widget ếchuyn sang màu ềnn ấcnh báo (tùy ợchn) */
751 /* .data-widget.warning-active .widget-header {
752     background-color: red !important;
753 } */

```

```

1 // --- ếBin toàn ựcc để ứlu ữtr ữđ ệliu ốgc ---
2 let currentRawCelsiusTemperature = null;
3 let currentHumidity = null;
4 let lastUpdatedTimestamp = null;
5 let updateIntervalId = null; // ếBin để ứlu ID ủa setInterval
6
7 // ếBin toàn ựcc cho ếbiu đồ
8 let temperatureHumidityChart; // Đối ượtrng Chart.js
9
10 let deviceSelector; // Khai báo deviceSelector ở ặphm vi toàn ựcc
11
12 // ếBin ặtrng thái DHT (ữgi nguyên)
13 let isDhtEnabled = true;

```

```

14
15 // ếBin cho ưỡngng ấcnh báo
16 let tempThreshold = null;
17 let humidityThreshold = null;
18
19 // main.js
20
21 // ... (các hàm ệhin có: updateToggleButtonUI, getInitialDhtStatus,
    convertCelsiusToFahrenheit, updateAuthUI, loadDeviceList, v.v.) ...
22
23 // Hàm ấti ưỡngng ấcnh báo cho ếthit ỉb đã ọchn
24 async function loadAlertThresholds(deviceId) {
25     const tempThresholdInput = document.getElementById('temp-threshold');
26     const humidityThresholdInput = document.getElementById('humidity-threshold');
27
28     if (!deviceId) {
29         tempThreshold = null;
30         humidityThreshold = null;
31         if (tempThresholdInput) tempThresholdInput.value = '';
32         if (humidityThresholdInput) humidityThresholdInput.value = '';
33         console.warn("Không có Device ID để ấti ưỡngng ấcnh báo.");
34         return;
35     }
36
37     try {
38         const response = await fetch(`api/get_alert_thresholds.php?device_id=${
deviceId}`, { credentials: 'include' });
39         if (!response.ok) throw new Error(`HTTP error! status: ${response.status}`);
40         const data = await response.json();
41
42         if (data.status === 'success' && data.thresholds) {
43             tempThreshold = parseFloat(data.thresholds.temp_threshold) || null;
44             humidityThreshold = parseFloat(data.thresholds.humidity_threshold) ||
null;
45
46             if (tempThresholdInput) tempThresholdInput.value = tempThreshold !==
null ? tempThreshold : '';
47             if (humidityThresholdInput) humidityThresholdInput.value =
humidityThreshold !== null ? humidityThreshold : '';
48             console.log(`ưỡNgng ấcnh báo cho ${deviceId} đã ấti: ệNhit độ=${
tempThreshold}, Độ ẩ=${humidityThreshold}`);
49         } else {
50             tempThreshold = null;
51             humidityThreshold = null;
52             if (tempThresholdInput) tempThresholdInput.value = '';
53             if (humidityThresholdInput) humidityThresholdInput.value = '';
54             console.warn(`Không tìm ấthy ưỡngng ấcnh báo cho ${deviceId} ặhoc có ỗli:
${data.message || ''}`);
55         }
56     } catch (error) {
57         console.error('ỗLi khi ấti ưỡngng ấcnh báo:', error);
58         tempThreshold = null;
59         humidityThreshold = null;
60         if (tempThresholdInput) tempThresholdInput.value = '';
61         if (humidityThresholdInput) humidityThresholdInput.value = '';
62     }
63 }
64
65 // Hàm ểkim tra và áp ụng ệhiu ứng ấcnh báo
66 function checkAndApplyAlerts(temperature, humidity) {
67     const tempWidget = document.querySelector('.temperature-widget');
68     const humWidget = document.querySelector('.humidity-widget');
69
70     if (!tempWidget || !humWidget) return;
71
72     // ểKim tra ưỡngng ệnhit độ

```

```

73     if (tempThreshold !== null && typeof temperature === 'number' && !isNaN(
temperature)) {
74         // Ếnũ ệnhit độ ượvt ưỡngng ẶHOC ệnhit độ áthp ohn ưỡngng ếnu ặn ốmun ắcnh báo
ắc 2 ềchiu
75         if (temperature > tempThreshold) { // Ví ụđ: ỉch ắcnh báo khi ượvt quá
76             tempWidget.classList.add('warning-active');
77         } else {
78             tempWidget.classList.remove('warning-active');
79         }
80     } else {
81         tempWidget.classList.remove('warning-active'); // ỜG ỏb ếnu không có ưỡngng
ắhoc ữđ ệliu không ợhp ệl
82     }
83
84     // ỂKĩm tra ưỡngng độ ẩm
85     if (humidityThreshold !== null && typeof humidity === 'number' && !isNaN(
humidity)) {
86         if (humidity > humidityThreshold) { // Ví ụđ: ỉch ắcnh báo khi ượvt quá
87             humWidget.classList.add('warning-active');
88         } else {
89             humWidget.classList.remove('warning-active');
90         }
91     } else {
92         humWidget.classList.remove('warning-active'); // ỜG ỏb ếnu không có ưỡngng
ắhoc ữđ ệliu không ợhp ệl
93     }
94 }
95
96 // THÊM HÀM NÀY: Hàm để ắcp ậnhit UI ủa nút ắbt/ắtt DHT
97 function updateToggleButtonUI() {
98     const toggleDhtButton = document.getElementById('toggleDhtButton');
99     if (!toggleDhtButton) return; // Đắmb ắo nút ồtn ặti ướtrc khi ốc ắngng ắcp ậnhit
100
101     if (isDhtEnabled) {
102         toggleDhtButton.classList.remove('dht-off'); // Xóa class ắtt
103         toggleDhtButton.classList.add('dht-on'); // Thêm class ắbt
104         toggleDhtButton.textContent = 'Đắng đo';
105     } else {
106         toggleDhtButton.classList.remove('dht-on'); // Xóa class ắbt
107         toggleDhtButton.classList.add('dht-off'); // Thêm class ắtt
108         toggleDhtButton.textContent = 'Đắ ắtt';
109     }
110 }
111
112 // Hàm để ắly ặtrng thắi DHT ban đầu ừt server khi ắti trang
113 async function getInitialDhtStatus(deviceId) { // deviceId ếs được ềtruyն ừt ơni ợgi
114     if (!deviceId) {
115         console.warn("getInitialDhtStatus: Device ID is empty. Cannot fetch initial
status.");
116
117         isDhtEnabled = true; // ẶMc địnđ là ắbt ếnu không có ID để trắnh ồli.
118         updateToggleButtonUI();
119         return;
120     }
121     try {
122         const response = await fetch(`api/get_dht_status.php?device_id=${deviceId}`,
{credentials: 'include'});
123         const statusText = await response.text();
124         isDhtEnabled = (statusText.trim() === 'true');
125         updateToggleButtonUI();
126         console.log(`Initial DHT status for ${deviceId}: ${isDhtEnabled ? 'Enabled'
: 'Disabled'}`);
127     } catch (error) {
128         console.error('Error getting initial DHT status:', error);
129         isDhtEnabled = true;
130         updateToggleButtonUI();
131     }

```

```

131 }
132 // --- Hàm ểchuyển đổi ừt Celsius sang Fahrenheit ---
133 function convertCelsiusToFahrenheit(celsius) {
134     if (typeof celsius !== 'number' || isNaN(celsius)) {
135         return null; // ảTr ềv null ểnu đầu vào không ợp ệl
136     }
137     return (celsius * 1.8) + 32;
138 }
139
140 // --- Hàm ửx lý ảtrng thái đăng ậnhpđã/ng ấxut Đ(Ả ỬSA ĐỔI THEO YẾU ẦCU) ---
141 function updateAuthUI(isLoggedIn, username = '', role = '') { // THÊM THAM ỚS `role`
142     const logoutButtonHeader = document.getElementById('logout-button');
143     const welcomeMessageEl = document.getElementById('welcome-message');
144     const usernameDisplayEl = document.getElementById('username-display');
145     const userManagementTab = document.querySelector('li[data-section="user-management"]');
146     // const deviceManagementTab = document.querySelector('li[data-section="device-management"]'); // ểNu có
147
148     // ỈCH ỂKIM TRA CÁC ẦPHN ỬT ỰTHC ỰS ỒTN ẬTI TRÊN INDEX.HTML
149     if (logoutButtonHeader && welcomeMessageEl && usernameDisplayEl) {
150         if (isLoggedIn) {
151             logoutButtonHeader.classList.remove('hidden'); // ệHin nút Đăng ấXut
152             welcomeMessageEl.classList.remove('hidden'); // ệHin dòng Chào ừmng
153             usernameDisplayEl.textContent = username;
154         } else {
155             logoutButtonHeader.classList.add('hidden'); // Ẩn nút Đăng ấXut
156             welcomeMessageEl.classList.add('hidden'); // Ẩn dòng Chào ừmng
157             usernameDisplayEl.textContent = '';
158         }
159     }
160
161     // ểHin ỉthả/n tab ảqun lý ườngi dùng ựda trên vai trò 'admin'
162     if (userManagementTab) {
163         if (isLoggedIn && role === 'admin') { // ỉCh ệhin khi đã đăng ậnhp và là admin
164             userManagementTab.classList.remove('hidden');
165         } else {
166             userManagementTab.classList.add('hidden');
167         }
168     }
169 }
170
171 // Hàm ảti danh sách ểthit ịb ừt backend và điền vào select
172 async function loadDeviceList() {
173     const deviceSelector = document.getElementById('device-selector');
174     if (!deviceSelector) {
175         console.error("Không tìm ấthy ầphn ửt #device-selector.");
176         return;
177     }
178
179     // Xóa ấtt ắc các option ửc, ừtr option ặmc định
180     deviceSelector.innerHTML = '<option value="">-- ọChn ểthit ịb --</option>';
181
182     try {
183         const response = await fetch('api/get_devices.php', {credentials: 'include'});
184         if (!response.ok) {
185             throw new Error(`HTTP error! status: ${response.status}`);
186         }
187         const data = await response.json();
188
189         if (data.success && data.devices && data.devices.length > 0) {
190             data.devices.forEach(device => {
191                 const option = document.createElement('option');

```



```

192         option.value = device.device_id;
193         option.textContent = device.device_name || device.device_id;
194         deviceSelector.appendChild(option);
195     });
196
197     // Sau khi tải danh sách thiết bị, chọn thiết bị cuối cùng đã chọn
198     const lastSelectedDeviceId = localStorage.getItem('lastSelectedDeviceId')
199 );
200     if (lastSelectedDeviceId && Array.from(deviceSelector.options).some(
201 option => option.value === lastSelectedDeviceId)) {
202         deviceSelector.value = lastSelectedDeviceId;
203     } else {
204         // Nếu không có lastSelectedDeviceId hoặc nó không hợp lệ,
205         // chọn ID của thiết bị đầu tiên từ dữ liệu nhận được
206         deviceSelector.value = data.devices[0].device_id;
207     }
208
209     // QUAN TRỌNG: KÍCH AHOT ỨS ỆKIN 'change' SAU KHI CHỌN THIẾT BỊ
210     // Việc này sẽ làm cho deviceSelector.addEventListener('change') được
211     // và đó là nơi updateCurrentData() và loadHistoryData() sẽ được gọi.
212     if (deviceSelector.value) { // Khi kích hoạt nếu đã có giá trị được chọn
213         console.log("loadDeviceList: Đã chọn thiết bị xác định. Kích hoạt ứs
214     ệkin change.");
215         deviceSelector.dispatchEvent(new Event('change'));
216
217         // Lấy trạng thái DHT ban đầu cho thiết bị vừa được chọn
218         // Hàm này sẽ gọi updateToggleButtonUI()
219         getInitialDhtStatus(deviceSelector.value);
220     } else {
221         // Trường hợp này xảy ra nếu deviceSelector.value bị đặt thành một giá
222         // trị không có trong danh sách
223         console.warn("loadDeviceList: Không có thiết bị nào được chọn xác định
224     sau khi tải.");
225         clearCurrentDataDisplay();
226         clearHistoryDataDisplay('Không có thiết bị nào được cấu hình. Vui
227     lòng thêm thiết bị.');
```

```

246         currentDataMessageEl.classList.add('error-message');
247     }
248     clearCurrentDataDisplay();
249     clearHistoryDataDisplay(`Ồi khi ắti ỉlch ứs ếthit ỉb: ${error.message}.`);
250     // ĐÃ ỦSA: Khi có ồli, nút ắphi ở ặtrng tháit Đ"ã ắtt"
251     isDhtEnabled = false;
252     updateToggleButtonUI();
253 }
254 }
255
256 // Hàm ắcp ậnhit ứđ ệliu ệhin ặti
257 async function updateCurrentData(deviceId) {
258     const currentTemperatureValueEl = document.getElementById('
259     currentTemperatureValue');
260     const currentTemperatureUnitEl = document.getElementById('currentTemperatureUnit
261     ');
262     const currentHumidityValueEl = document.getElementById('currentHumidityValue');
263     const lastUpdatedTimeEl = document.getElementById('lastUpdatedTime');
264     const currentDataMessageEl = document.getElementById('current-data-message');
265     const temperatureUnitSelectEl = document.getElementById('temperatureUnitSelect');
266
267     // ếNu không có deviceId, xóa ểhin ỉth và ừđng ắcp ậnhit
268     if (!deviceId) {
269         clearCurrentDataDisplay();
270         if (currentDataMessageEl) {
271             currentDataMessageEl.textContent = 'Vui lòng ợchn ộmt ếthit ỉb.';
272             currentDataMessageEl.classList.add('no-data-message');
273             currentDataMessageEl.classList.remove('info-message', 'error-message');
274         }
275         if (updateIntervalId) {
276             clearInterval(updateIntervalId);
277             updateIntervalId = null;
278         }
279         checkAndApplyAlerts(null, null); // ốG ỏb ợmi ắcnh báo khi không có ếthit ỉb
280         return;
281     }
282
283     // ểHin ỉth thớng báo đặng ắti
284     if (currentDataMessageEl) {
285         currentDataMessageEl.textContent = 'Đặng ắti ứđ ệliu ệhin ặti...';
286         currentDataMessageEl.classList.remove('error-message', 'no-data-message');
287         currentDataMessageEl.classList.add('info-message');
288     }
289
290     try {
291         const response = await fetch(`get_current_data.php?device_id=${deviceId}`, {
292             credentials: 'include'});
293         if (!response.ok) {
294             throw new Error(`HTTP error! status: ${response.status}`);
295         }
296         const data = await response.json();
297
298         // ừX lý khi ứđ ệliu thặnh cớng
299         if (data.status === "success" && data.reading) {
300             if (currentDataMessageEl) {
301                 currentDataMessageEl.textContent = '';
302                 currentDataMessageEl.classList.remove('info-message', 'error-message', 'no-data-message');
303             }
304
305             currentRawCelsiusTemperature = parseFloat(data.reading.temperature);
306             currentHumidity = parseFloat(data.reading.humidity);
307             lastUpdatedTimestamp = data.reading.reading_time;

```

```

306     const selectedUnit = localStorage.getItem('selectedTemperatureUnit') ||
    'celsius';
307     if (temperatureUnitSelectEl) {
308         // Đảm bảo giá trị của select khớp với localStorage
309         temperatureUnitSelectEl.value = (selectedUnit === 'fahrenheit') ? 'F
    ' : 'C';
310     }
311
312     // Gọi hàm displayTemperature để cập nhật nhiệt độ và đơn vị
313     displayTemperature(currentRawCelsiusTemperature, selectedUnit,
    currentTemperatureValueEl, currentTemperatureUnitEl);
314
315     if (currentHumidityValueEl) currentHumidityValueEl.textContent =
    currentHumidity.toFixed(1);
316     if (lastUpdatedTimeEl) {
317         const dateTime = luxon.DateTime.fromSQL(lastUpdatedTimestamp);
318
319         if (dateTime.isValid) {
320             lastUpdatedTimeEl.textContent = dateTime.toFormat('HH:mm:ss dd/
    MM/yyyy');
321         } else {
322             console.error("Ồi phân tích thời gian Luxon:", dateTime.
    invalidReason, dateTime.invalidExplanation);
323             lastUpdatedTimeEl.textContent = 'Ồi Định dạng thời Gian';
324         }
325     }
326
327     // THÊM: Kiểm tra và áp dụng hiệu ứng ảnh báo với dữ liệu mới nhất
328     checkAndApplyAlerts(currentRawCelsiusTemperature, currentHumidity);
329
330     // Khi động lại interval cập nhật tự động với tần suất đã lưu
331     const savedFrequency = localStorage.getItem('savedUpdateFrequency');
332     const defaultFrequency = 5;
333     const initialFrequency = Math.max(1, parseInt(savedFrequency) ||
    defaultFrequency);
334     startAutoUpdate(initialFrequency, deviceId);
335
336     } else { // Xử lý khi API thành công nhưng không có dữ liệu hoặc có lỗi logic
337         currentRawCelsiusTemperature = null;
338         currentHumidity = null;
339         lastUpdatedTimestamp = null;
340
341         if (currentDataMessageEl) {
342             currentDataMessageEl.textContent = data.message || 'Không có dữ liệu
    hiện tại cho thiết bị này.';
343             currentDataMessageEl.classList.remove('info-message', 'error-message
    ');
344             currentDataMessageEl.classList.add('no-data-message');
345         }
346         clearCurrentDataDisplay();
347         if (updateIntervalId) {
348             clearInterval(updateIntervalId);
349             updateIntervalId = null;
350         }
351         checkAndApplyAlerts(null, null); // Gọi hàm ảnh báo khi không có dữ
    liệu
    }
352 }
353 } catch (error) { // Xử lý khi có lỗi mạng hoặc lỗi server
354     console.error('Ồi khi tải dữ liệu hiện tại:', error);
355     currentRawCelsiusTemperature = null;
356     currentHumidity = null;
357     lastUpdatedTimestamp = null;
358
359     if (currentDataMessageEl) {
360         currentDataMessageEl.textContent = `Ồi khi tải dữ liệu hiện tại: ${error
    .message}. Vui lòng thử lại.`;
    }

```

```

361         currentDataMessageEl.classList.remove('info - message', 'no-data - message')
362     ;
363     currentDataMessageEl.classList.add('error - message');
364 }
365 clearCurrentDataDisplay();
366 if (updateIntervalId) {
367     clearInterval(updateIntervalId);
368     updateIntervalId = null;
369 }
370 checkAndApplyAlerts(null, null); // ỜG Ờb ợmi ảnh báo khi có ỗli
371 }
372 }
373 // Hàm ểhìn ỉth ệnhit độ ựda trên đơn ịv đã ợchn (thêm element ểhìn ỉth đơn ịv)
374 function displayTemperature(celsius, unit, displayElement, unitDisplayElement) {
375     if (!displayElement) return;
376
377     if (celsius === null || isNaN(celsius)) {
378         displayElement.textContent = 'N/A';
379         if (unitDisplayElement) unitDisplayElement.textContent = ''; // Ắn đơn ịv
380         return;
381     }
382
383     if (unit === 'fahrenheit') {
384         const fahrenheit = convertCelsiusToFahrenheit(celsius);
385         displayElement.textContent = fahrenheit !== null ? fahrenheit.toFixed(1) : '
N/A';
386         if (unitDisplayElement) unitDisplayElement.textContent = '°F';
387     } else { // ảmc định là celsius
388         displayElement.textContent = celsius.toFixed(1);
389         if (unitDisplayElement) unitDisplayElement.textContent = '°C';
390     }
391 }
392
393 // Hàm để ởkhi ạto/ắtt/ậbt ập ậnh ữd ệliu định ỳk Đ(Ả ỦSA ĐỔI)
394 function startAutoUpdate(frequency, deviceId) { // THÊM THAM ỚS deviceId
395     if (updateIntervalId) {
396         clearInterval(updateIntervalId);
397     }
398     // ỉCh ởkhi ạto interval ếnu có deviceId ợhp ệl
399     if (deviceId) { // Bầy ờgi deviceId đã được ềtruyv vào
400         updateIntervalId = setInterval(() => {
401             updateCurrentData(deviceId); // ợGi ớvi deviceId
402             }, frequency * 1000); // ểChuyv giây sang mili giây
403             console.log(`ẮBt đầu ập ậnh ựt động cho ếthit ịb ${deviceId} ớvi ầtn ấsut ${
frequency} giây.`);
404         } else {
405             console.warn("Không có Device ID để ập ậnh ữd ệliu định ỳk. Đã ừdng ập ậnh
ựt động.");
406             updateIntervalId = null; // Đảmb ảo ếbin là null ếnu không có interval
407         }
408     }
409
410 // Hàm để ởkhi ạto ặhoc ập ậnh ểbiu đồ
411 function createOrUpdateChart(labels, temperatures, humidities, timeUnit,
selectedTemperatureUnit) {
412     const ctx = document.getElementById('temperatureHumidityChart');
413     if (!ctx) {
414         console.error("Không tìm ấthy ầphn ừt canvas ểbiu đồ ớvi ID '
temperatureHumidityChart'");
415         return;
416     }
417     const chartContext = ctx.getContext('2d');
418
419     // Xác định nhữn cho ựt ệnhit độ và dataset

```

```

420     const temperatureLabel = selectedTemperatureUnit === 'fahrenheit' ? '°Nhit độ (°F)' : '°Nhit độ (°C)';
421     const temperatureUnitSymbol = selectedTemperatureUnit === 'fahrenheit' ? '°F' : '°C';
422
423     if (temperatureHumidityChart) {
424         // ậCp ậnhit ửd ệliu cho ểbiu đồ ệhinh có
425         temperatureHumidityChart.data.labels = labels;
426         temperatureHumidityChart.data.datasets[0].label = temperatureLabel; // ậCp
         ậnhit label ệnhit độ
427         temperatureHumidityChart.data.datasets[0].data = temperatures;
428         temperatureHumidityChart.data.datasets[1].data = humidities;
429
430         // Đảm bảo spanGaps là false khi ậcp ậnhit ểnu ốmun thay đổi động
431         temperatureHumidityChart.data.datasets[0].spanGaps = false; // THÊM DÒNG NÀY
432         temperatureHumidityChart.data.datasets[1].spanGaps = false; // THÊM DÒNG NÀY
433
434
435         // ậCp ậnhit đơn iv ựtrc X
436         temperatureHumidityChart.options.scales.x.time.unit = timeUnit;
437
438         // ậCp ậnhit displayFormats ựda trên timeUnit
439         if (timeUnit === 'minute') {
440             temperatureHumidityChart.options.scales.x.time.displayFormats = { minute
: 'HH:mm' };
441         } else if (timeUnit === 'hour') {
442             temperatureHumidityChart.options.scales.x.time.displayFormats = { hour:
'HH:mm DD/MM' };
443         } else if (timeUnit === 'day') {
444             temperatureHumidityChart.options.scales.x.time.displayFormats = { day: '
DD/MM' };
445         }
446
447         // ậCp ậnhit nhân ựtrc Y
448         // Không ầcn ậcp ậnhit ựtrc ểtip nhân ựtrc Y ửna vì đã có 2 ựtrc riêng
449         temperatureHumidityChart.options.scales.yTemperature.title.text = `°Nhit độ (
${temperatureUnitSymbol})`; // ậCp ậnhit nhân ựtrc ệnhit độ
450         temperatureHumidityChart.options.scales.yHumidity.title.text = `Độ ẩm (%)`;
451         // ậCp ậnhit nhân ựtrc độ ẩm
452
453         temperatureHumidityChart.update();
454     } else {
455         // ậTo ểbiu đồ ốmi
456         temperatureHumidityChart = new Chart(chartContext, {
457             type: 'line',
458             data: {
459                 labels: labels,
460                 datasets: [
461                     {
462                         label: temperatureLabel, // ửS ựdng label động
463                         data: temperatures,
464                         borderColor: 'rgb(255, 99, 132)',
465                         backgroundColor: 'rgba(255, 99, 132, 0.2)',
466                         fill: false,
467                         tension: 0.1,
468                         yAxisID: 'yTemperature', // GÁN DATASET NÀY ỚVI ỰTRC Y ỆNHIT
ĐỘ
469                     },
470                     {
471                         label: 'Độ ẩm (%)',
472                         data: humidities,
473                         borderColor: 'rgb(54, 162, 235)',
474                         backgroundColor: 'rgba(54, 162, 235, 0.2)',

```

```

475         fill: false ,
476         tension: 0.1 ,
477         yAxisID: 'yHumidity', // GÁN DATASET NÀY Ở VI ỰTRC Y ĐỘ ẨM
478         spanGaps: false // THÊM DÒNG NÀY: Để không ốni các ảkhong
    ốtrng (NULL)
    }
    ],
    },
    options: {
        responsive: true ,
        maintainAspectRatio: false ,
        scales: {
            x: {
                type: 'time' ,
                time: {
                    unit: timeUnit ,
                    tooltipFormat: 'HH:mm:ss DD/MM/YYYY' ,
                    displayFormats: {
                        minute: 'HH:mm' ,
                        hour: 'HH:mm DD/MM' ,
                        day: 'DD/MM' ,
                    }
                },
                title: {
                    display: true ,
                    text: 'ờThi gian'
                }
            },
        },
    },
    // ĐINH NGHĨA ỰTRC Y CHO ỆNHIT ĐỘ (Bên trái)
    yTemperature: {
        type: 'linear' , // ạLoi ựtrc là ếtuyt tính
        position: 'left' , // Đặt ở bên trái
        beginAtZero: false , // Không ắbt đầu ừt 0
        title: {
            display: true ,
            text: `ệNhit độ (${temperatureUnitSymbol})` // Nhân ựtrc
    ệnhit độ
        },
        min : 0 ,
        max : 100 ,
        grid: {
            drawOnChartArea: false // iCh ẽv grid cho ựtrc này, không
    ẽv qua toàn ộb ểbiu đồ
        }
    },
    // ĐINH NGHĨA ỰTRC Y CHO ĐỘ ẨM (Bên ảphi)
    yHumidity: {
        type: 'linear' , // ạLoi ựtrc là ếtuyt tính
        position: 'right' , // Đặt ở bên ảphi
        beginAtZero: true , // Độ ẩm ườthng ắbt đầu ừt 0
        title: {
            display: true ,
            text: 'Độ ẩm (%)' // Nhân ựtrc độ ẩm
        },
        min : 0 ,
        max : 100 ,
        // Có ểth thêm grid: { drawOnChartArea: false } ếuu không
    ốmun grid cho ắc 2 ựtrc
        // ưNhg ườthng thì grid ỉch ẽv ừt ựtrc trái.
        grid: {
            drawOnChartArea: false // Không ẽv grid cho ựtrc này để
    tránh trùng ắlp
        }
    },
    },
    },

```

```

535         plugins: {
536             tooltip: {
537                 callbacks: {
538                     title: function(context) {
539                         const timestamp = luxon.DateTime.fromMillis(context
[0].parsed.x);
540                         if (timeUnit === 'minute') {
541                             return timestamp.toFormat('HH:mm:ss dd/MM');
542                         } else if (timeUnit === 'hour') {
543                             return timestamp.toFormat('HH:mm dd/MM/yyyy');
544                         } else if (timeUnit === 'day') {
545                             return timestamp.toFormat('dd/MM/yyyy');
546                         }
547                         return timestamp.toFormat('HH:mm:ss dd/MM/yyyy');
548                     },
549                     label: function(context) {
550                         let label = context.dataset.label || '';
551                         if (label) {
552                             label += ': ';
553                         }
554                         if (context.parsed.y !== null) {
555                             if (context.dataset.label.includes('Nhiệt độ')) {
556                                 label += context.parsed.y.toFixed(1); // Làm
tròn 1 chữ số thập phân
557                             } else {
558                                 label += context.parsed.y.toFixed(1);
559                             }
560                         } else {
561                             label += 'Không có dữ liệu';
562                         }
563                         return label;
564                     }
565                 }
566             }
567         }
568     });
569 });
570 }
571 }
572
573 async function loadHistoryData(period, deviceId, startDate = null, endDate = null) {
574     const historyTableBody = document.getElementById('history-table-body');
575     const chartCanvas = document.getElementById('temperatureHumidityChart');
576     const historyDataMessageEl = document.getElementById('history-data-message');
577     const historyTableContainer = document.getElementById('history-table');
578     const historyChartContainer = document.getElementById('history-chart');
579
580     // Hiển thị thông báo đang tải
581     if (historyDataMessageEl) {
582         historyDataMessageEl.textContent = 'Đang tải dữ liệu...';
583         historyDataMessageEl.classList.remove('error-message', 'no-data-message');
584         historyDataMessageEl.classList.add('info-message');
585         historyDataMessageEl.style.display = 'block';
586     }
587
588     // Ẩn ảnh và biểu đồ trong khi tải
589     if (historyTableContainer) historyTableContainer.style.display = 'none';
590     if (historyChartContainer) historyChartContainer.style.display = 'none';
591
592     if (!deviceId) {
593         clearHistoryDataDisplay('Vui lòng chọn một thiết bị để xem lịch sử dữ liệu.');
594         return;
595     }
596
597     try {
598         let apiUrl = `get_history_data.php?device_id=${deviceId}&period=${period}`;

```

```

599     if (period === 'custom' && startDate && endDate) {
600         apiUrl += `&start_date=${startDate}&end_date=${endDate}`;
601     }
602
603     const response = await fetch(apiUrl, {credentials: 'include'}); // THÊM
604     credentials: 'include'
605     if (!response.ok) {
606         throw new Error(`HTTP error! status: ${response.status}`);
607     }
608     const data = await response.json();
609     const selectedTemperatureUnit = localStorage.getItem('
610     selectedTemperatureUnit') || 'celsius';
611
612     if (data.status === "success" && data.data && data.data.length > 0) {
613         if (historyDataMessageEl) {
614             historyDataMessageEl.textContent = '';
615             historyDataMessageEl.classList.remove('info-message', 'error-message
616             ', 'no-data-message');
617             historyDataMessageEl.style.display = 'none';
618         }
619
620         const readings = data.data;
621
622         let unitDisplay;
623         const startLuxon = startDate ? luxon.DateTime.fromISO(startDate) : null;
624         const endLuxon = endDate ? luxon.DateTime.fromISO(endDate) : null;
625
626         if (period === '1h' || (period === 'custom' && startLuxon && endLuxon &&
627         endLuxon.diff(startLuxon, 'hours').hours <= 2)) {
628             unitDisplay = 'minute';
629         } else if (period === '24h' || (period === 'custom' && startLuxon &&
630         endLuxon && endLuxon.diff(startLuxon, 'hours').hours <= 24)) {
631             unitDisplay = 'hour';
632         } else {
633             unitDisplay = 'day';
634         }
635     }
636
637     // ừSa đổi DÒNG NÀY: ừS ụngng fromSQL() để phân tích ổchui ởthi gian
638     const timeLabels = readings.map(r => luxon.DateTime.fromSQL(r.timestamp)
639     .toMillis());
640
641     const chartTemperatures = readings.map(r => {
642         const tempC = parseFloat(r.temperature);
643         return selectedTemperatureUnit === 'fahrenheit' ?
644         convertCelsiusToFahrenheit(tempC) : tempC;
645     });
646     const chartHumidities = readings.map(r => parseFloat(r.humidity));
647
648     const tableReadings = readings.map(r => ({
649         reading_time: r.timestamp, // Đây là ổchui ởthi gian để ểhin ịth
650         trong ảbng
651         temperature: parseFloat(r.temperature),
652         humidity: parseFloat(r.humidity)
653     }));
654
655     createOrUpdateChart(timeLabels, chartTemperatures, chartHumidities,
656     unitDisplay, selectedTemperatureUnit);
657     populateHistoryTable(tableReadings);
658
659     // ểHin ịth ảli ảbng và ểbiu đồ
660     if (historyTableContainer) historyTableContainer.style.display = 'block'
661     ;
662     if (historyChartContainer) historyChartContainer.style.display = 'block'
663     ;
664     } else {

```



```

654         // Không có dữ liệu
655         if (temperatureHumidityChart) {
656             temperatureHumidityChart.destroy();
657             temperatureHumidityChart = null;
658         }
659         clearHistoryDataDisplay(data.message || 'Không có dữ liệu lịch sử cho
        không ở thi gian học ethic này. ');
660     }
661 } catch (error) {
662     console.error('Ồi khi tải lịch sử dữ liệu:', error);
663     if (temperatureHumidityChart) {
664         temperatureHumidityChart.destroy();
665         temperatureHumidityChart = null;
666     }
667     clearHistoryDataDisplay(`Ấy ra ời khi tải dữ liệu lịch sử: ${error.
        message}.`);
668 }
669 }
670
671 // --- HÀM POPULATE HISTORY TABLE (ỮNG NGUYÊN) ---
672 function populateHistoryTable(readingsForTable) {
673     const historyTableBody = document.getElementById('history-table-body');
674     const tableUnitHeaderEl = document.getElementById('table-temperature-unit-header
        ');
675
676     if (!historyTableBody) {
677         console.error('Ồi: Không tìm thấy phần tử tbody của bảng lịch sử ở vi ID '
        history-table-body'.');
678         return;
679     }
680
681     historyTableBody.innerHTML = '';
682
683     const selectedUnit = localStorage.getItem('selectedTemperatureUnit') || 'celsius
        ';
684     if (tableUnitHeaderEl) {
685         tableUnitHeaderEl.textContent = selectedUnit === 'fahrenheit' ? '°F' : '°C';
686     }
687
688     if (!readingsForTable || readingsForTable.length === 0) {
689         const row = historyTableBody.insertRow();
690         const cell = row.insertCell();
691         cell.colSpan = 3;
692         cell.className = 'no-data-message';
693         cell.textContent = 'Không có dữ liệu lịch sử để hiển thị trong bảng.';
694         return;
695     }
696
697     readingsForTable.forEach(reading => {
698         const row = historyTableBody.insertRow();
699
700         const timeCell = row.insertCell();
701         const tempCell = row.insertCell();
702         const humidityCell = row.insertCell();
703
704         timeCell.textContent = luxon.DateTime.fromSQL(reading.reading_time).toFormat
        ('HH:mm:ss dd/MM/yyyy');
705
706         const rawTemperature = parseFloat(reading.temperature);
707         if (rawTemperature !== null && !isNaN(rawTemperature)) {
708             let displayTemperature;
709             if (selectedUnit === 'fahrenheit') {
710                 displayTemperature = convertCelsiusToFahrenheit(rawTemperature);
711                 tempCell.textContent = displayTemperature !== null ?
        displayTemperature.toFixed(1) : 'N/A';
712             } else {

```

```

713         tempCell.textContent = rawTemperature.toFixed(1);
714     }
715     } else {
716         tempCell.textContent = 'N/A';
717     }
718
719     const rawHumidity = parseFloat(reading.humidity);
720     if (rawHumidity !== null && !isNaN(rawHumidity)) {
721         humidityCell.textContent = rawHumidity.toFixed(1);
722     } else {
723         humidityCell.textContent = 'N/A';
724     }
725 });
726 }
727
728 // --- Hàm ợtr giúp để clear ểhin ỉth ữđ ệliu ệhin ạti (ỮGI NGUYỄN) ---
729 function clearCurrentDataDisplay() {
730     const currentTemperatureValueEl = document.getElementById('
731     currentTemperatureValue');
732     const currentHumidityValueEl = document.getElementById('currentHumidityValue');
733     const lastUpdatedTimeEl = document.getElementById('lastUpdatedTime');
734     const currentTemperatureUnitEl = document.getElementById('currentTemperatureUnit
735     ');
736
737     if (currentTemperatureValueEl) currentTemperatureValueEl.textContent = 'N/A';
738     if (currentHumidityValueEl) currentHumidityValueEl.textContent = 'N/A';
739     if (lastUpdatedTimeEl) lastUpdatedTimeEl.textContent = 'N/A';
740     if (currentTemperatureUnitEl) currentTemperatureUnitEl.textContent = ''; // Xóa
741     đơn ịv
742 }
743
744 // --- Hàm ợtr giúp để clear ểhin ỉth ỉlch ửđ ữđ ệliu (ỮGI NGUYỄN) ---
745 function clearHistoryDataDisplay(message = 'Vui lòng ợchn ểthit ịb để xem ỉlch ửđ.') {
746     const historyDataMessageEl = document.getElementById('history-data-message');
747     const historyTableBody = document.getElementById('history-table-body');
748     const historyChartContainer = document.getElementById('history-chart');
749     const historyTableContainer = document.getElementById('history-table');
750
751     if (historyDataMessageEl) {
752         historyDataMessageEl.textContent = message;
753         historyDataMessageEl.classList.remove('info-message', 'error-message');
754         historyDataMessageEl.classList.add('no-data-message');
755         historyDataMessageEl.style.display = 'block'; // Luôn ểhin ỉth thông báo
756     }
757     if (historyTableBody) {
758         historyTableBody.innerHTML = '<tr><td colspan="3" class="no-data-message">'
759         + message + '</td></tr>';
760     }
761
762     // ửHy ểbiu đồ ểnu nó đang ồtn ạti
763     if (temperatureHumidityChart) {
764         temperatureHumidityChart.destroy();
765         temperatureHumidityChart = null;
766     }
767
768     // Ẩn chart và table containers
769     if (historyChartContainer) historyChartContainer.style.display = 'none';
770     if (historyTableContainer) historyTableContainer.style.display = 'none';
771 }
772
773 document.addEventListener('DOMContentLoaded', async function() {
774     // --- 1. ấLy tham ểchiu đến các ầphn ửđ DOM ---
775     const navLinks = document.querySelectorAll('#main-nav ul li');
776     const sections = document.querySelectorAll('main section');
777     const logoutButtonHeader = document.getElementById('logout-button');
778
779     const historyPeriodSelect = document.getElementById('history-period');
780     const customPeriodDiv = document.getElementById('custom-period');

```

```

775     const startDateInput = document.getElementById('start-date');
776     const endDateInput = document.getElementById('end-date');
777     const applyCustomPeriodButton = document.getElementById('apply-custom-period');
778     const exportCsvButton = document.getElementById('export-csv');
779
780     deviceSelector = document.getElementById('device-selector'); // Đảm bảo
deviceSelector được gán ở đây
781
782     // Các hành xử cho việc hiển thị dữ liệu hiển thị
783     const updateFrequencyInput = document.getElementById('update-frequency');
784     const saveFrequencyButton = document.getElementById('save-frequency');
785     const temperatureUnitSelectEl = document.getElementById('temperatureUnitSelect')
;
786
787     // --- 2. ở Khi tạo trạng thái đơn vị nhiệt độ ---
788     const savedUnit = localStorage.getItem('selectedTemperatureUnit');
789     if (savedUnit && temperatureUnitSelectEl) {
790         temperatureUnitSelectEl.value = (savedUnit === 'fahrenheit') ? 'F' : 'C';
791     } else if (temperatureUnitSelectEl) {
792         localStorage.setItem('selectedTemperatureUnit', 'celsius');
793         temperatureUnitSelectEl.value = 'C';
794     }
795
796     // *** ÁT ĐẦU ÓKHI ẾKIM TRA ĐĂNG ẬNH ***
797     try {
798         const response = await fetch('../Login/api/users/check_login.php', {
credentials: 'include'});
799         if (!response.ok) {
800             console.error(`HTTP error! status: ${response.status}, statusText: ${
response.statusText}`);
801             throw new Error(`HTTP error! status: ${response.status}`);
802         }
803         const data = await response.json();
804
805         if (data.loggedIn) {
806             updateAuthUI(true, data.username, data.role);
807             await loadDeviceList(); // ảTi danh sách ếthit ịb ich khi đã đăng ậnhp
808
809             // ở Khi tạo ầtn ấsút ậcp ậnhb ban đầu (sau khi ếthit ịb đã được ảti)
810             const savedFrequency = localStorage.getItem('savedUpdateFrequency');
811             const defaultFrequency = 5;
812             const initialFrequency = Math.max(1, parseInt(savedFrequency) ||
defaultFrequency);
813             if (updateFrequencyInput) {
814                 updateFrequencyInput.value = initialFrequency;
815             }
816             // Logic startAutoUpdate ban đầu đã được ếchuyển vào loadDeviceList()
817             // thông qua event 'change' ủa deviceSelector.
818             // KHÔNG ẦCN startAutoUpdate ở đây để tránh trùng ặlp.
819
820             } else {
821                 updateAuthUI(false);
822                 clearCurrentDataDisplay();
823                 clearHistoryDataDisplay('Vui lòng đăng ậnhp để xem ữd ệliu. ');
824                 alert('Phiên đăng ậnhp đã ếht ậnh ặhoc ậbn ưcha đăng ậnhp. Vui lòng đăng
ậnhp ặli. ');
825                 window.location.href = '../Login/login_register.html';
826             }
827         } catch (error) {
828             console.error('ồLi khi ếkim tra ậtrng thái đăng ậnhp ừt main.js:', error);
829             alert('ồLi ếkt óni để xác ựthc. Vui lòng ữth ặli. ');
830             updateAuthUI(false);
831             clearCurrentDataDisplay();
832             clearHistoryDataDisplay('ồLi ảti ữd ệliu. Vui lòng ữth ặli. ');
833             window.location.href = '../Login/login_register.html';
834         }

```

```

835 // *** ÉKT THỨC ÓKHI ÉKIM TRA ĐĂNG ẬNH ***
836
837 // -----
838 // ÁBT ĐẦU CÁC EVENT LISTENERS VÀ LOGIC KHÁC (SAU ÓKHI ĐĂNG ẬNH)
839 // -----
840
841 // --- LOGIC CHO NÚT ẬBT/ẮTT DHT ---
842 const toggleDhtButton = document.getElementById('toggleDhtButton');
843 if (toggleDhtButton) {
844     toggleDhtButton.addEventListener('click', async () => {
845         isDhtEnabled = !isDhtEnabled; // Đảo ượcngc ạtrng tháí UI ngay ập út
846         updateToggleButtonUI(); // ập ậnh ểhìn ịth nút
847
848         const currentDeviceId = deviceSelector ? deviceSelector.value : '';
849
850         if (!currentDeviceId) {
851             alert('Vui lòng ọchn ộmt ếthit ịb ướtrc khi ậbt/ắtt ắcm ếbin.');
```

ếthit ịb

```

852             isDhtEnabled = !isDhtEnabled; // Revert ặli ạtrng tháí ếnu không có
853             updateToggleButtonUI();
854             return;
855         }
856
857         try {
858             const response = await fetch('api/toggle_dht_status.php', { // DỪNG
ĐƯỜNG ẶDN ỚNG ĐỐI
859                 method: 'POST',
860                 headers: {
861                     'Content-Type': 'application/json',
862                 },
863                 body: JSON.stringify({ status: isDhtEnabled, device_id:
currentDeviceId }),
864                 credentials: 'include'
865             });
866
867             const data = await response.json();
868             if (data.status === 'success') {
869                 console.log(data.message);
870             } else {
871                 console.error('API Error:', data.message);
872                 isDhtEnabled = !isDhtEnabled;
873                 updateToggleButtonUI();
874                 alert('Có ỗli khi ập ậnh ạtrng tháí DHT: ' + data.message);
875             }
876         } catch (error) {
877             console.error('Network error:', error);
878             isDhtEnabled = !isDhtEnabled;
879             updateToggleButtonUI();
880             alert('ỖLi ếkt ốni đến server. Vui lòng ửth ặli.');
```

}}

```

881         });
882     });
883 }
884
885 // --- ÉKT THỨC LOGIC CHO NÚT ẬBT/ẮTT DHT ---
886
887 // --- ửX lý ểchuyển đổi ữgia các tab điều ướhng ---
888 navLinks.forEach(link => {
889     link.addEventListener('click', function(e) {
890         e.preventDefault();
891         navLinks.forEach(nav => nav.classList.remove('active'));
892         this.classList.add('active');
893
894         const targetSectionId = this.dataset.section;
895         sections.forEach(section => {
896             if (section.id === targetSectionId) {
897                 section.classList.remove('hidden-section');
```

```

897         section.classList.add('active-section');
898     } else {
899         section.classList.add('hidden-section');
900         section.classList.remove('active-section');
901     }
902 });
903
904 const currentDeviceId = deviceSelector ? deviceSelector.value : '';
905 const currentPeriod = historyPeriodSelect ? historyPeriodSelect.value :
'24h';
906
907 if (targetSectionId === 'history-data') {
908     if (updateIntervalId) {
909         clearInterval(updateIntervalId);
910         updateIntervalId = null;
911     }
912     // ĐÃ ỦSA: Đảm bảo nút bật/ắt DHT được đặt về trạng thái tắt khi
không ở tab hiển thị
913     isDhtEnabled = false;
914     updateToggleButtonUI();
915
916     if (currentDeviceId) {
917         loadHistoryData(currentPeriod, currentDeviceId, startDateInput.
value, endDateInput.value);
918     } else {
919         clearHistoryDataDisplay('Vui lòng chọn một thiết bị để xem lịch sử
. ');
920     }
921 } else if (targetSectionId === 'current-data') {
922     if (currentDeviceId) {
923         const defaultFrequency = 5;
924         startAutoUpdate(parseInt(updateFrequencyInput.value) ||
defaultFrequency, currentDeviceId);
925         updateCurrentData(currentDeviceId);
926         // ĐÃ ỦSA: álli trạng thái DHT khi quay lại tab Current Data
getInitialDhtStatus(currentDeviceId);
927     } else {
928         clearCurrentDataDisplay();
929     }
930     clearHistoryDataDisplay('');
931 } else { // Các tab khác không liên quan đến việc lưu trữ
932     if (updateIntervalId) {
933         clearInterval(updateIntervalId);
934         updateIntervalId = null;
935     }
936     clearCurrentDataDisplay();
937     clearHistoryDataDisplay('');
938
939     // ĐÃ ỦSA: cập nhật trạng thái nút khi rời khỏi tab lưu trữ
isDhtEnabled = false;
940     updateToggleButtonUI();
941 }
942 });
943 });
944
945 // --- xử lý nút Đăng ấXut ---
946 if (logoutButtonHeader) {
947     logoutButtonHeader.addEventListener('click', async () => {
948         try {
949             const response = await fetch('../Login/api/users/logout.php', {
950                 method: 'POST', credentials: 'include' });
951             if (!response.ok) {
952                 throw new Error(`HTTP error! status: ${response.status}`);
953             }
954             const result = await response.json();
955             if (result.status === 'success') {
956                 alert('Đăng ấxut thành công!');

```

```

957         if (updateIntervalId) {
958             clearInterval(updateIntervalId);
959             updateIntervalId = null;
960         }
961         clearCurrentDataDisplay();
962         clearHistoryDataDisplay('Vui lòng đăng nhập để xem lịch sử. ');
963         if (deviceSelector) {
964             deviceSelector.value = "";
965             deviceSelector.innerHTML = '<option value="">-- Chọn thiết bị
--</option>';
966         }
967         updateAuthUI(false);
968         // Đã USA: Cập nhật trạng thái nút khi đăng xuất
969         isDhtEnabled = false;
970         updateToggleButtonUI();
971
972         window.location.href = '../Login/login_register.html';
973     } else {
974         alert('Đăng xuất thất bại: ' + (result.message || 'Không xác định
'));
975     }
976 } catch (error) {
977     console.error(' lỗi khi đăng xuất:', error);
978     alert(' lỗi khi đăng xuất. Vui lòng thử lại. ');
979 }
980 });
981 }
982
983 // --- Xử lý thay đổi thiết bị ---
984 if (deviceSelector) {
985     deviceSelector.addEventListener('change', function() {
986         const selectedDeviceId = this.value;
987         localStorage.setItem('lastSelectedDeviceId', selectedDeviceId);
988
989         if (selectedDeviceId) {
990             updateCurrentData(selectedDeviceId);
991             // Đã USA: Khi người dùng nhấn báo cho thiết bị mới
992             loadAlertThresholds(selectedDeviceId);
993
994             const currentPeriod = historyPeriodSelect ? historyPeriodSelect.
value : '24h';
995             const activeSectionId = document.querySelector('.active-section').
id;
996             if (activeSectionId === 'history-data') {
997                 loadHistoryData(currentPeriod, selectedDeviceId, startDateInput.
value, endDateInput.value);
998             }
999             const defaultFrequency = 5;
1000             startAutoUpdate(parseInt(updateFrequencyInput.value) ||
defaultFrequency, selectedDeviceId);
1001             // Đã USA: Khi thay đổi trạng thái DHT khi đổi thiết bị
1002             getInitialDhtStatus(selectedDeviceId);
1003         } else {
1004             clearCurrentDataDisplay();
1005             clearHistoryDataDisplay('Vui lòng chọn thiết bị để xem lịch sử. ');
1006             if (updateIntervalId) {
1007                 clearInterval(updateIntervalId);
1008                 updateIntervalId = null;
1009             }
1010             // Đã USA: Cập nhật trạng thái nút khi không có thiết bị được chọn
1011             isDhtEnabled = false;
1012             updateToggleButtonUI();
1013         }
1014     });
1015 }
1016

```

```

1017 // --- xử lý thay đổi không ở thời gian lịch ---
1018 if (historyPeriodSelect) {
1019     historyPeriodSelect.addEventListener('change', function() {
1020         const selectedPeriod = this.value;
1021         if (selectedPeriod === 'custom') {
1022             if (customPeriodDiv) customPeriodDiv.classList.remove('hidden');
1023         } else {
1024             if (customPeriodDiv) customPeriodDiv.classList.add('hidden');
1025             const currentDeviceId = deviceSelector ? deviceSelector.value : '';
1026             if (currentDeviceId) {
1027                 loadHistoryData(selectedPeriod, currentDeviceId, null, null);
1028             } else {
1029                 clearHistoryDataDisplay('Vui lòng chọn thiết bị để xem lịch sử.');
```

```

1030             }
1031         }
1032     });
1033 }
1034
1035 // --- xử lý nút "Áp dụng" cho không ở thời gian tùy ý ---
1036 if (applyCustomPeriodButton) {
1037     applyCustomPeriodButton.addEventListener('click', function() {
1038         const currentDeviceId = deviceSelector ? deviceSelector.value : '';
1039         if (startDateInput && endDateInput && currentDeviceId) {
1040             if (startDateInput.value && endDateInput.value && startDateInput.
value <= endDateInput.value) {
1041                 loadHistoryData('custom', currentDeviceId, startDateInput.value,
endDateInput.value);
1042             } else {
1043                 alert('Vui lòng chọn ngày bắt đầu và ngày kết thúc hợp lệ.');
```

```

1044             }
1045         } else {
1046             alert('Vui lòng chọn đầy đủ ngày bắt đầu, ngày kết thúc và thiết bị.')
```

```

1047         };
1048     });
1049 }
1050
1051 // --- xử lý nút "Xuất CSV" ---
1052 if (exportCsvButton) {
1053     exportCsvButton.addEventListener('click', async function() {
1054         const currentDeviceId = deviceSelector ? deviceSelector.value : '';
1055         const selectedPeriod = historyPeriodSelect ? historyPeriodSelect.value :
```

```

'24h';
1056         let start = null;
1057         let end = null;
1058
1059         if (selectedPeriod === 'custom' && startDateInput && endDateInput) {
1060             start = startDateInput.value;
1061             end = endDateInput.value;
1062             if (!start || !end || start > end) {
1063                 alert('Vui lòng chọn ngày bắt đầu và ngày kết thúc hợp lệ cho
không ở thời gian tùy ý.');
```

```

1064                 return;
1065             }
1066         }
1067
1068         if (!currentDeviceId) {
1069             alert('Vui lòng chọn một thiết bị để xuất dữ liệu.');
```

```

1070             return;
1071         }
1072
1073         try {
1074             // ĐÃ XỬA: DỪNG ĐƯỜNG LỐI NGUYỄN ĐỐI
1075             let exportUrl = `api/export_data_csv.php?device_id=<span class="math
-inline">{\currentDeviceId}&period=</span>{selectedPeriod}`;
1076             if (start && end) {
```

```

1077         exportUrl += '&start_date=<span class="math-inline">\{start\}&
end\_date\=</span>\{end\} `;
1078     }
1079
1080     window.location.href = exportUrl;
1081
1082     } catch (error) {
1083         console.error('ỖLi khi áxut CSV:', error);
1084         alert('Đã áxy ra ỗli khi áxut ũđ ệliu CSV.');
```

```

1085     }
1086     });
1087 }
1088
```

```

1089 // --- ữX lý thay đổi đơn vị ệnhit độ (C/F) ---
1090 if (temperatureUnitSelectEl) {
1091     temperatureUnitSelectEl.addEventListener('change', function() {
1092         const selectedUnit = this.value === 'F' ? 'fahrenheit' : 'celsius';
1093         localStorage.setItem('selectedTemperatureUnit', selectedUnit);
1094
1095         const currentDeviceId = deviceSelector ? deviceSelector.value : '';
1096         if (currentDeviceId) {
1097             updateCurrentData(currentDeviceId);
1098         }
1099
1100         const activeSectionId = document.querySelector('.active-section')?.id;
1101         if (activeSectionId === 'history-data') {
1102             const currentPeriod = historyPeriodSelect ? historyPeriodSelect.
value : '24h';
1103             if (currentDeviceId) {
1104                 loadHistoryData(currentPeriod, currentDeviceId, startDateInput.
value, endDateInput.value);
1105             }
1106         }
1107     });
1108 }
1109
```

```

1110 // --- ữX lý ứlu àtn ásut ácp ậnhit ---
1111 if (saveFrequencyButton) {
1112     saveFrequencyButton.addEventListener('click', function() {
1113         const newFrequency = parseInt(updateFrequencyInput.value);
1114         if (newFrequency && newFrequency >= 1) {
1115             localStorage.setItem('savedUpdateFrequency', newFrequency);
1116             alert('ầTn ásut ácp ậnhit đã được ứlu: ' + newFrequency + ' giây.');
```

```

1117             const currentDeviceId = deviceSelector ? deviceSelector.value : '';
1118             if (currentDeviceId) {
1119                 const defaultFrequency = 5;
1120                 startAutoUpdate(newFrequency, currentDeviceId);
1121             }
1122         } else {
1123             alert('ầTn ásut ácp ậnhit ấphi là ộmt ốs nguyên ươdng.');
```

```

1124             updateFrequencyInput.value = initialFrequency;
1125         }
1126     });
1127 }
1128
1129 // --- ữX lý ứlu ườngng ảcnh báo ---
1130 const saveThresholdsButton = document.getElementById('save-thresholds');
1131 const tempThresholdInput = document.getElementById('temp-threshold');
1132 const humidityThresholdInput = document.getElementById('humidity-threshold');
```

```

1133
1134 if (saveThresholdsButton && tempThresholdInput && humidityThresholdInput) {
1135     saveThresholdsButton.addEventListener('click', async () => {
1136         const currentDeviceId = deviceSelector ? deviceSelector.value : '';
1137         if (!currentDeviceId) {
1138             alert('Vui lòng ọchn ộmt ếthit ịb để ứlu ườngng.');
```

```

1139             return;

```



```

1140     }
1141
1142     const newTempThreshold = parseFloat(tempThresholdInput.value);
1143     const newHumidityThreshold = parseFloat(humidityThresholdInput.value);
1144
1145     // ểKim tra tính ợhp ệl ủa ưỡngng
1146     if (isNaN(newTempThreshold) && tempThresholdInput.value !== '') {
1147         alert('ườNgng ệnhit độ không ợhp ệl. ');
1148         return;
1149     }
1150     if (isNaN(newHumidityThreshold) && humidityThresholdInput.value !== '')
1151     {
1152         alert('ườNgng độ ẩm không ợhp ệl. ');
1153         return;
1154     }
1155
1156     // ửX lý ườtrng ợhp ườngi dùng xóa giá ịtr để ỡg ỏb ưỡngng
1157     const finalTempThreshold = tempThresholdInput.value === '' ? null :
newTempThreshold;
1158     const finalHumidityThreshold = humidityThresholdInput.value === '' ?
null : newHumidityThreshold;
1159
1160     try {
1161         const response = await fetch('api/save_alert_thresholds.php', {
1162             method: 'POST',
1163             headers: {
1164                 'Content-Type': 'application/json',
1165             },
1166             body: JSON.stringify({
1167                 device_id: currentDeviceId,
1168                 temp_threshold: finalTempThreshold,
1169                 humidity_threshold: finalHumidityThreshold
1170             }),
1171             credentials: 'include'
1172         });
1173
1174         const data = await response.json();
1175         if (data.status === 'success') {
1176             alert('ườNgng ảnh báo đã được ứlư thành công!');
1177             tempThreshold = finalTempThreshold; // ậCp ậnhт ếbin toàn ược
ườcc
1178             humidityThreshold = finalHumidityThreshold; // ậCp ậnhт ếbin toàn
ườcc
1179             checkAndApplyAlerts(currentRawCelsiusTemperature,
currentHumidity); // ểKim tra ặli ảnh báo
1180         } else {
1181             alert('ỗLi khi ứlư ưỡngng: ' + (data.message || 'Không xác định. '
1182             ));
1183         }
1184         } catch (error) {
1185             console.error('ỗLi ảnhт khi ứlư ưỡngng:', error);
1186             alert('ỗLi ếkt ốni đến server khi ứlư ưỡngng. Vui lòng ửth ặli. ');
1187         }
1188     });
1189
1190     // ởKhi ặto ặtrng thái ban đầu ủa custom-period div
1191     if (historyPeriodSelect && customPeriodDiv) {
1192         if (historyPeriodSelect.value === 'custom') {
1193             customPeriodDiv.classList.remove('hidden');
1194         } else {
1195             customPeriodDiv.classList.add('hidden');
1196         }
1197     }
1198 }); // ếKt thúc DOMContentLoaded đây là ảbn hoàn ịchnh ủa hàm DOM

```

```

1 <!DOCTYPE html>
2 <html lang="vi">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Hệ Thống Giám Sát Nhiệt Độ và Độ Ẩm</title>
7     <link rel="stylesheet" href="main.css">
8     <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
9     <script src="https://cdn.jsdelivr.net/npm/luxon@3.x.x/build/global/luxon.min.js">
10     </script>
11     <script src="https://cdn.jsdelivr.net/npm/chartjs-adapter-luxon@1.x.x"></script>
12 </head>
13 <body>
14     <header>
15         <div class="header-top">
16
17             <div class="auth-buttons">
18                 <button id="logout-button" class="hidden">Đăng ấXut</button>
19                 <span id="welcome-message" class="hidden">Xin chào, <span id="
20 username-display"></span>!</span>
21             </div>
22         </div>
23
24         <nav id="main-nav">
25             <ul>
26                 <li data-section="current-data" class="active"><a href="#">ƯĐ ệliu
27 ệHin ậTi</a></li>
28                 <li data-section="history-data"><a href="#">ịLch ửS ữĐ ệLiu</a></li>
29                 <li data-section="alerts"><a href="#">ấCnh Báo</a></li>
30                 <li data-section="user-management" class="hidden"><a href="#">ảQun
31 Lý ườNgì Dừn</a></li>
32                 <li data-section="settings"><a href="#">Thông tin ếhit ịb</a></li>
33             </ul>
34         </nav>
35     </header>
36
37     <main>
38         <section id="current-data" class="active-section">
39             <div class="device-selection-main">
40                 <label for="device-selector" class="block text-gray-700 text-sm font
41 -bold mb-2">ợChn ếhit ịb:</label>
42                 <select id="device-selector" class="shadow appearance-none border
43 rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none
44 focus:shadow-outline">
45                     <option value="">-- ợChn ếhit ịb --</option>
46                 </select>
47             </div>
48             <h2>ƯĐ ệliu ệHin ậTi</h2>
49             <div class="current-data-grid">
50                 <div class="data-widget temperature-widget">
51                     <div class="widget-header">ệNhit độ</div>
52                     <div class="widget-content">
53                         <span id="currentTemperatureValue" class="data-value">N/A</
54 span>
55                         <span id="currentTemperatureUnit" class="data-unit"></span>
56                     </div>
57                 </div>
58                 <div class="data-widget humidity-widget">
59                     <div class="widget-header">Độ ẩm</div>
60                     <div class="widget-content">
61                         <span id="currentHumidityValue" class="data-value">N/A</span>
62                         <span id="currentHumidityUnit" class="data-unit">%</span>
63                     </div>
64                 </div>
65             </div>
66         </section>
67     </main>
68 </body>
69 </html>

```

```

58     </div>
59     <div class="data-item timestamp-item">
60         <span class="label">Cập nhật lúc:</span>
61         <span id="lastUpdatedTime" class="value">N/A</span>
62     </div>
63
64     <div class="settings-grid">
65         <div class="setting-group">
66             <h3>Ấp đặt cập nhật</h3>
67             <label for="update-frequency">Ấp đặt (giờ):</label>
68             <input type="number" id="update-frequency" value="5">
69             <button id="save-frequency">Save</button>
70         </div>
71
72         <div class="setting-group">
73             <h3>Đơn vị Đo</h3>
74             <label for="temperatureUnitSelect">Đơn vị nhiệt độ:</label>
75             <select id="temperatureUnitSelect">
76                 <option value="C">°C</option>
77                 <option value="F">°F</option>
78             </select>
79         </div>
80     </div>
81     <div id="current-data-message" class="info-message"></div>
82 </section>
83
84 <section id="history-data" class="hidden-section">
85     <h2>Lịch sử Dữ liệu</h2>
86     <div class="controls">
87         <label for="history-period">Chọn khoảng thời gian:</label>
88         <select id="history-period">
89             <option value="1h">1 giờ qua</option>
90             <option value="24h">24 giờ qua</option>
91             <option value="7d">7 ngày qua</option>
92             <option value="30d">30 ngày qua</option>
93             <option value="custom">Tùy chỉnh</option>
94         </select>
95         <div id="custom-period" class="hidden">
96             <label for="start-date">Từ ngày:</label>
97             <input type="date" id="start-date">
98             <label for="end-date">Đến ngày:</label>
99             <input type="date" id="end-date">
100             <button id="apply-custom-period">Áp dụng</button>
101         </div>
102         <button id="export-csv">Xuất CSV</button>
103     </div>
104     <div id="history-chart">
105         <canvas id="temperatureHumidityChart"></canvas>
106     </div>
107     <div id="history-table">
108         <table class="data-table">
109             <thead>
110                 <tr>
111                     <th>Thời gian</th>
112                     <th>Nhiệt độ (<span id="table-temperature-unit-header">°C
113
114                     <th>Độ ẩm (%)</th>
115                 </tr>
116             </thead>
117             <tbody id="history-table-body">
118             </tbody>
119         </table>
120     </div>
121     <div id="history-data-message" class="info-message"></div>
122 </section>

```

```

123 <section id="alerts" class="hidden-section">
124 <h2>Ảnh Báo</h2>
125 <div class="alert-settings">
126 <h3>Thiết lập ngưỡng</h3>
127 <div class="setting-item">
128 <label for="temp-threshold">Ngưỡng nhiệt độ (°C):</label>
129 <input type="number" id="temp-threshold">
130 </div>
131 <div class="setting-item">
132 <label for="humidity-threshold">Ngưỡng độ ẩm (%):</label>
133 <input type="number" id="humidity-threshold">
134 </div>
135 <button id="save-thresholds">Lưu ngưỡng</button>
136 </div>
137 <div class="alert-status">
138 <h3>Trạng thái cảnh báo</h3>
139 <div id="temperature-alert" class="alert">Nhiệt độ: <span id="temp-
alert-status">Bình thường</span></div>
140 <div id="humidity-alert" class="alert">Độ ẩm: <span id="humidity-
alert-status">Bình thường</span></div>
141 </div>
142 <div class="notification-settings">
143 <h3>Thông báo</h3>
144 <label>
145 <input type="checkbox" id="enable-email-alerts"> Gửi thông báo
qua Email
146 </label>
147 <div id="email-settings" class="hidden">
148 <label for="email-address">Địa chỉ Email:</label>
149 <input type="email" id="email-address">
150 <button id="save-email-settings">Lưu Email</button>
151 </div>
152 </div>
153 </section>
154
155 <section id="settings" class="hidden-section">
156 <h2>Thông Tin Kỹ Thuật Sản Phẩm</h2>
157 <div class="component-info">
158 <h3>DHT11 - Cảm biến nhiệt độ và độ ẩm</h3>
159 <p>DHT11 là một cảm biến nhiệt độ và độ ẩm thường được ứng dụng đi kèm
với một NTC chuyên dụng để đo nhiệt độ và một bộ vi điều khiển 8 bit để xử lý các
giá trị nhiệt độ và độ ẩm rồi đưa ra kết quả.</p>
160 <div class="image-container">
161 
162 </div>
163 <h4>Định nghĩa sơ đồ chân DHT11 và cấu hình:</h4>
164 <ul>
165 <li><strong>VCC</strong> Nguồn điện 3.5V đến 5.5V</li>
166 <li><strong>Data</strong> Đầu ra về nhiệt độ và Độ ẩm thông qua
mạch vi xử lý</li>
167 <li><strong>Ground</strong> Kết nối với mặt đất của mạch</li>
168 </ul>
169 <h4>Thông số kỹ thuật DHT11:</h4>
170 <ul>
171 <li><strong>Điện áp hoạt động</strong> 3.5V đến 5.5V</li>
172 <li><strong>Dòng hoạt động</strong> 0,3mA (ở 60uA (sai lệch 5%)</li>
173 <li><strong>Đầu ra</strong> Mạch vi xử lý</li>
174 <li><strong>Phạm vi nhiệt độ</strong> 0 ° C đến 50 ° C</li>
175 <li><strong>Phạm vi độ ẩm</strong> 20% đến 90%</li>
176 <li><strong>Độ phân giải</strong> Nhiệt độ và Độ ẩm đều là 16-
bit</li>
177 <li><strong>Độ chính xác</strong> ± 1 ° C và ± 1%</li>
178 </ul>
179 </div>

```

```

180
181         <div class="component-info">
182             <h3>Ổ Gii ệthi Vi điều ểkhin ESP32</h3>
183             <p>ESP32 là ộmt ộb vi điều ểkhin ộthuc danh ụmc vi điều ểkhin trên
chip công ấsut ấthp và ếtít ệkim chỉ phí. ầHu ểht ấtt ắc các ếbin ểth ESP32 đều
tích ộhp Bluetooth và Wi-Fi ếch độ kép, làm cho nó có tính linh ậhot cao, ạmnh ễn
và đáng tin ậcy cho ềnhieu ứng ựng. Nó là ựs ểk ừtha ứca vi điều ểkhin NodeMCU
ESP8266 ộph ếbin và cung ấcp ệhiu ấsut và tính ắnnng ốtt ộhn. ộB vi điều ểkhin
ESP32 được ắsn ấxut ởbi Espressif Systems và được ửs ựng ộngng rãi trong ềnhieu ứng
ựng khác nhau ỳnh IoT, robot và ựt động hóa.</p>
184             <p>ESP32 ừcng được ếthit ểk để tiêu ựth đẹin ắnnng ấthp, lý ửởng cho
các ứng ựng ậchy ầbng pin. Nó có ệh ốthng ắqun lý ắnnng ựngng cho phép nó ậhot
động ở ếch độ ừng và ịch ứthc ậdy khi ắcn ếthit, điều này có ểth kéo dài ốtui ộth
pin ắrt ềnhieu.</p>
185             <div class="image-container">
186                 
187             </div>
188             <h4>ổS đồ ạmch:</h4>
189             <div class="image-container">
190                 
191             </div>
192         </div>
193     </section>
194
195     <section id="user-management" class="hidden-section">
196         <h2>ắQun Lý ườNgi Dừng</h2>
197     </section>
198
199 </main>
200
201 <footer id="bottom-header"> <div class="bottom-header-content">
202     <p>&copy; 2025 ệH ốThng Giám Sát ệNhit Độ và Độ Ắm</p>
203 </div>
204 </footer>
205
206 <script src="main.js"></script>
207 </body>
208 </html>

```

A.3.2 Các file trong Login/

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Đăng ậNhập / Đăng Ký</title>
7     <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='
stylesheet '>
8     <link rel="stylesheet" href="auth_styles.css"> </head>
9 <body>
10     <div class="auth-wrapper">
11         <form action="#" class="auth-form" id="login-form">
12             <h1 class="form-title">Đăng ậNhập</h1>
13
14             <div class="input-group">
15                 <i class='bx bxs-user'></i>
16                 <input type="text" id="username-login" placeholder="Tên đăng ậnhp"
required>
17             </div>
18             <div class="input-group">
19                 <i class='bx bxs-lock-alt'></i>
20                 <input type="password" id="password-login" placeholder="ậMt ắkhu"
required>

```

```

21     </div>
22
23     <div class="remember-forgot-box">
24         <label>
25             <input type="checkbox" id="remember-me">
26             Ghi ớnh ấnhp
27         </label>
28         <a href="#" class="forgot-password-link">Quên ậmt ắkh?</a>
29     </div>
30
31     <button type="submit" class="auth-btn" id="login-submit-btn">Đắnh ậNh</
button>
32     <p class="auth-switch">
33         ưCh có tài ắkhon?
34         <a href="#" id="show-register-form-link">Đắnh ký</a>
35     </p>
36     <p id="login-status-message" class="status-message"></p>
37 </form>
38
39 <form action="#" class="auth-form hidden" id="register-form">
40     <h1 class="form-title">Đắnh Ký</h1>
41
42     <div class="input-group">
43         <i class='bx bxs-user '></i>
44         <input type="text" id="username-register" placeholder="Tên đắnh ậnhp"
required>
45     </div>
46     <div class="input-group">
47         <i class='bx bxs-lock-alt '></i>
48         <input type="password" id="password-register" placeholder="ậMt ắkh"
required>
49     </div>
50     <div class="input-group">
51         <i class='bx bxs-lock-alt '></i>
52         <input type="password" id="confirm-password-register" placeholder="
Xác ậnhn ậmt ắkh" required>
53     </div>
54
55     <button type="submit" class="auth-btn" id="register-submit-btn">Đắnh Ký<
/button>
56     <p class="auth-switch">
57         Đắ có tài ắkhon?
58         <a href="#" id="show-login-form-link">Đắnh ậnhp</a>
59     </p>
60     <p id="register-status-message" class="status-message"></p>
61 </form>
62 </div>
63
64 <script src="auth_script.js"></script> </body>
65 </html>

```

```

1 * {
2     margin: 0;
3     padding: 0;
4     box-sizing: border-box;
5     font-family: 'Poppins', sans-serif;
6 }
7
8 body {
9     min-height: 100vh;
10    display: flex;
11    align-items: center;
12    justify-content: center;
13    background-image: linear-gradient(to top, #4CAF50, #2196F3);
14    background-size: cover;
15    background-position: center;
16    background-attachment: fixed;

```

```

17 }
18
19 .auth-wrapper { /* Thêm wrapper để ẩn ùgia form */
20     display: flex;
21     justify-content: center;
22     align-items: center;
23     width: 100%;
24     min-height: 100vh;
25     padding: 20px;
26 }
27
28 .auth-form {
29     background: linear-gradient(to top, #3a7bd5, #00d2ff);
30     border: none;
31     padding: 30px;
32     border-radius: 16px;
33     text-align: center;
34     color: white;
35     width: 100%;
36     max-width: 400px; /* óGii ăhn èchiu ộrng óti đa */
37     box-shadow: 0px 10px 25px rgba(0, 0, 0, 0.3);
38 }
39
40 .form-title {
41     font-size: 40px;
42     margin-bottom: 40px;
43     color: white;
44 }
45
46 .input-group {
47     margin: 20px 0;
48     position: relative;
49 }
50
51 .input-group input {
52     width: 100%;
53     background: rgba(255, 255, 255, 0.25);
54     border: none;
55     padding: 12px 12px 12px 45px;
56     border-radius: 99px;
57     outline: 3px solid transparent;
58     transition: 0.3s;
59     font-size: 17px;
60     color: white;
61     font-weight: 600;
62 }
63
64 .input-group input::placeholder {
65     color: rgba(255, 255, 255, 0.9);
66     font-size: 17px;
67     font-weight: 500;
68 }
69
70 .input-group input:focus {
71     outline: 3px solid rgba(255, 255, 255, 0.5);
72 }
73
74 .input-group input::-ms-reveal {
75     filter: invert(100%);
76 }
77
78 .input-group .bx {
79     position: absolute;
80     left: 15px;
81     top: 50%;
82     transform: translateY(-50%);
83     font-size: 28px;

```

```

84     color: white;
85 }
86
87 .remember-forgot-box {
88     display: flex;
89     justify-content: space-between;
90     margin: 20px 0;
91     font-size: 15px;
92     color: white;
93 }
94
95 .remember-forgot-box label {
96     display: flex;
97     gap: 8px;
98     cursor: pointer;
99 }
100
101 .remember-forgot-box input[type="checkbox"] {
102     accent-color: white;
103     cursor: pointer;
104 }
105
106 .remember-forgot-box a {
107     color: white;
108     text-decoration: none;
109 }
110
111 .remember-forgot-box a:hover {
112     text-decoration: underline;
113 }
114
115 .auth-btn {
116     width: 100%;
117     padding: 10px 0;
118     background: #2F9CF4;
119     border: none;
120     border-radius: 99px;
121     color: white;
122     font-size: 16px;
123     font-weight: 600;
124     cursor: pointer;
125     transition: 0.3s;
126     box-shadow: 0px 5px 15px rgba(0, 0, 0, 0.2);
127 }
128
129 .auth-btn:hover {
130     background: #0887EC;
131 }
132
133 .auth-switch {
134     margin-top: 15px;
135     font-size: 15px;
136     color: white;
137 }
138
139 .auth-switch a {
140     color: white;
141     text-decoration: none;
142     font-weight: 500;
143 }
144
145 .auth-switch a:hover {
146     text-decoration: underline;
147 }
148
149 /* Các ớlp ệtin ích (hidden) */
150 .hidden {

```



```

151     display: none !important;
152 }
153
154 /* Status messages */
155 .status-message {
156     margin-top: 15px;
157     padding: 10px;
158     border-radius: 5px;
159     font-weight: bold;
160     text-align: center;
161     display: none;
162 }
163
164 .status-message.error {
165     background-color: rgba(255, 99, 71, 0.2);
166     border: 1px solid tomato;
167     color: #FF6347;
168 }
169
170 .status-message.success {
171     background-color: rgba(60, 179, 113, 0.2);
172     border: 1px solid MediumSeaGreen;
173     color: MediumSeaGreen;
174 }
175
176 .status-message.info {
177     background-color: rgba(70, 130, 180, 0.2);
178     border: 1px solid SteelBlue;
179     color: SteelBlue;
180 }

```

```

1 // auth_script.js
2
3 // áLy tham ếchíu đến các form và các ầphn ửt thông báo
4 const loginForm = document.getElementById('login-form');
5 const registerForm = document.getElementById('register-form');
6 const showRegisterFormLink = document.getElementById('show-register-form-link');
7 const showLoginFormLink = document.getElementById('show-login-form-link');
8
9 const usernameLoginInput = document.getElementById('username-login');
10 const passwordLoginInput = document.getElementById('password-login');
11 const loginStatusMessage = document.getElementById('login-status-message');
12
13 const usernameRegisterInput = document.getElementById('username-register');
14 const passwordRegisterInput = document.getElementById('password-register');
15 const confirmPasswordRegisterInput = document.getElementById('confirm-password-register');
16 const registerStatusMessage = document.getElementById('register-status-message');
17
18 // Hàm ểhìn ịth thông báo
19 function showStatusMessage(element, message, type = 'info') {
20     element.textContent = message;
21     element.className = 'status-message'; // Đặt ặli các class ềv ặmc định
22     element.classList.add(type); // Thêm class ặloi thông báo (error, success, info)
23     element.style.display = 'block'; // ểHìn ịth thông báo
24
25     // Xóa thông báo sau 5 giây
26     setTimeout(() => {
27         element.style.display = 'none';
28         element.textContent = '';
29         element.classList.remove(type); // Xóa class ặloi thông báo
30     }, 5000);
31 }
32
33 // Hàm để ểhìn ịth form Đăng ký và ắn form Đăng ậnhp
34 function displayRegisterForm() {
35     loginForm.classList.add('hidden');

```

```

36     registerForm.classList.remove('hidden');
37     // Xóa thông báo tức khi chuyển form
38     showStatusMessage(loginStatusMessage, '', 'info');
39     showStatusMessage(registerStatusMessage, '', 'info');
40 }
41
42 // Hàm để hiển thị form Đăng nhập và ẩn form Đăng ký
43 function displayLoginForm() {
44     registerForm.classList.add('hidden');
45     loginForm.classList.remove('hidden');
46     // Xóa thông báo tức khi chuyển form
47     showStatusMessage(loginStatusMessage, '', 'info');
48     showStatusMessage(registerStatusMessage, '', 'info');
49 }
50
51 // --- Xử lý sự kiện chuyển đổi form ---
52 if (showRegisterFormLink) {
53     showRegisterFormLink.addEventListener('click', function(e) {
54         e.preventDefault();
55         displayRegisterForm();
56     });
57 }
58
59 if (showLoginFormLink) {
60     showLoginFormLink.addEventListener('click', function(e) {
61         e.preventDefault();
62         displayLoginForm();
63     });
64 }
65
66 // --- Xử lý đăng nhập ---
67 if (loginForm) { // Kiểm tra nếu form tồn tại
68     loginForm.addEventListener('submit', async function(e) {
69         e.preventDefault(); // Ngăn form gửi đi theo cách truyền thống
70
71         const username = usernameLoginInput.value.trim();
72         const password = passwordLoginInput.value.trim();
73
74         showStatusMessage(loginStatusMessage, 'Đang xử lý...', 'info');
75
76         if (!username || !password) {
77             showStatusMessage(loginStatusMessage, 'Vui lòng nhập tên đăng nhập và mật
78             khẩu.', 'error');
79             return;
80         }
81
82         try {
83             // Đường dẫn từ Login/login_register.html đến Login/api/users/login.php
84             const response = await fetch('api/users/login.php', {
85                 method: 'POST',
86                 headers: {
87                     'Content-Type': 'application/json'
88                 },
89                 body: JSON.stringify({ username, password }),
90                 credentials: 'include' // Để gửi cookie session
91             });
92             const data = await response.json();
93
94             if (response.ok && data.status === 'success') {
95                 console.log("auth_script.js: Đăng nhập thành công. ID gửi từ server:
96                 ", data);
97
98                 // XÓA DÒNG NÀY: localStorage.setItem('loggedInUser', data.username)
99                 ;
100
101                 // XÓA DÒNG NÀY: console.log("auth_script.js: Đã lưu loggedInUser
102                 vào localStorage:", localStorage.getItem('loggedInUser'));

```

```

97         showStatusMessage(loginStatusMessage, 'Đăng nhập thành công! Đang
    ểchuyển ướhng...', 'success');
98         // ểChuyển ướhng đến trang dashboard chính (index.php) sau khi đăng
    ậnhp thành công
99         setTimeout(() => {
100             // Đảm bảo ểchuyển ướhng đến PHP gateway để ểkim tra session
101             window.location.href = '../TH_DO_AN/index.php'; // ỂCHUYỂN ỚHNG
    ỚTI TH_DO_AN/index.php
102         }, 1500); // ờCh 1.5 giây để ườngi dùng ấthý thông báo
103     } else {
104         console.log("auth_script.js: Đăng nhập ấtht ặbi. ấPhn ồhi:", data);
105         showStatusMessage(loginStatusMessage, 'ồLi đăng ậnhp: ' + (data.
    message || 'Tên đăng ậnhp ặhoc ậmt ấkhu không đúng. '), 'error');
106     }
107     } catch (error) {
108         console.error('ồLi network khi đăng ậnhp:', error);
109         showStatusMessage(loginStatusMessage, 'ồLi ểkt ốni. Vui lòng ữth ặli.', '
    error');
110     }
111     });
112 }
113
114
115 // --- ửX lý đăng ký ---
116 if (registerForm) { // ểKim tra ếnu form ồtn ặti
117     registerForm.addEventListener('submit', async function(e) {
118         e.preventDefault(); // ấNgn form ửgi đi theo cách ềtruyền ốthng
119
120         const username = usernameRegisterInput.value.trim();
121         const password = passwordRegisterInput.value.trim();
122         const confirmPassword = confirmPasswordRegisterInput.value.trim();
123
124         showStatusMessage(registerStatusMessage, 'Đang ửx lý...', 'info');
125
126         if (!username || !password || !confirmPassword) {
127             showStatusMessage(registerStatusMessage, 'Vui lòng điền đầy đủ các ườtrng
    .', 'error');
128             return;
129         }
130         if (password !== confirmPassword) {
131             showStatusMessage(registerStatusMessage, 'ậMt ấkhu xác ậnhn không ớkhp!',
    'error');
132             return;
133         }
134         if (password.length < 6) { // Yêu ầcu ậmt ấkhu ốti ểthiu 6 ký ựt
135             showStatusMessage(registerStatusMessage, 'ậMt ấkhu ấphi có ít ấnh 6 ký
    ựt.', 'error');
136             return;
137         }
138
139         try {
140             // Đường ấdn ừt Login/login_register.html đến Login/api/users/register.
    php
141             const response = await fetch('api/users/register.php', {
142                 method: 'POST',
143                 headers: {
144                     'Content-Type': 'application/json'
145                 },
146                 body: JSON.stringify({ username, password }),
147                 credentials: 'include' // Quan ợtrng để ửgi cookie session
148             });
149             const data = await response.json();
150
151             if (response.ok && data.status === 'success') {

```

```

152         showStatusMessage(registerStatusMessage, 'Đăng ký thành công! ạn có
        ểh đăng ậnhp ngay bây ờgi.', 'success');
153         // Sau khi đăng ký thành công, ựt động ểchuyn sang form đăng ậnhp
154         setTimeout(() => {
155             displayLoginForm();
156             // Xóa ữd ệliu input sau khi đăng ký thành công
157             usernameRegisterInput.value = '';
158             passwordRegisterInput.value = '';
159             confirmPasswordRegisterInput.value = '';
160         }, 1000); // Ớđi 1 giây để ườngi dùng ấthy thông báo
161     } else {
162         showStatusMessage(registerStatusMessage, 'ỗLi đăng ký: ' + (data.
        message || 'Tên đăng ậnhp đã ồtn ạti ặhoc có ỗli. '), 'error');
163     }
164     } catch (error) {
165         console.error('ỗLi network khi đăng ký:', error);
166         showStatusMessage(registerStatusMessage, 'ỗLi ểkt ốni. Vui lòng ữth ặli.'
        , 'error');
167     }
168     });
169 }
170
171 // Hàm ểkim tra ạtrng thái đăng ậnhp khi trang ắti (ỉCh dùng ếnu ốmun ựt động ểchuyn
        ớhng khi đã đăng ậnhp)
172 // ếNu ườngi dùng đã đăng ậnhp, ựt động ểchuyn ớhng đến dashboard.
173 document.addEventListener('DOMContentLoaded', async () => {
174     try {
175         // Đường ấdn ừt Login/login_register.html đến Login/api/users/check_login.php
176         const response = await fetch('api/users/check_login.php', {credentials: '
        include'});
177         const data = await response.json();
178         if (data.loggedIn) {
179             console.log("ườNgì dùng đã đăng ậnhp. ểChuyn ớhng đến dashboard.");
180             window.location.href = '../TH_DO_AN/index.php'; // ểChuyn ớhng đến
            TH_DO_AN/index.php
181         } else {
182             console.log("ườNgì dùng ưcha đăng ậnhp. ểHin ịth form đăng ậnhp.");
183             displayLoginForm(); // ểHin ịth form đăng ậnhp khi ắti trang
184         }
185     } catch (error) {
186         console.error('ỗLi khi ểkim tra ạtrng thái đăng ậnhp ban đầu:', error);
187         // ếNu có ỗli, ấvn ểhin ịth form đăng ậnhp để ườngi dùng có ểth ữth ặli
188         displayLoginForm();
189     }
190 });

```