Assignment 2: Individual Assignment (Practical) Report

Doan Minh Hieu - 104168106

1. Summary:

This project serves as a prototype of an IoT proof-of-concept for a smart home appliance, specifically a smart fan, utilizing my practical knowledge gained during tutorial sessions and through self-learning.

The proposed system involves an IoT node using an Arduino Uno with a DHT11 sensor for humidity and temperature measurement and an LM35 sensor for temperature readings. The setup also includes a 16x2 LCD for display and a small fan module. The Arduino communicates with a Raspberry Pi running on a virtual machine via serial communication to enable data storage and conditional automation based on temperature thresholds. Additionally, the Raspberry Pi on the virtual machine hosts a User Interface in PHP, displaying real-time data collected by the sensors. There are analysis features provided by the interface like average temperature, minimum temperature, maximum temperature, and so on.

Anyone can experience comfort with a smart fan's threshold-triggered automatic feature, current temperature and humidity notifications, and the convenience of remote control via a User Interface... Despite the simplicity of this project, this smart appliance concept has the potential to create a better, more connected lifestyle where technology improves our day-to-day activities.

2. Conceptual Design:

2.1. Block Diagram:

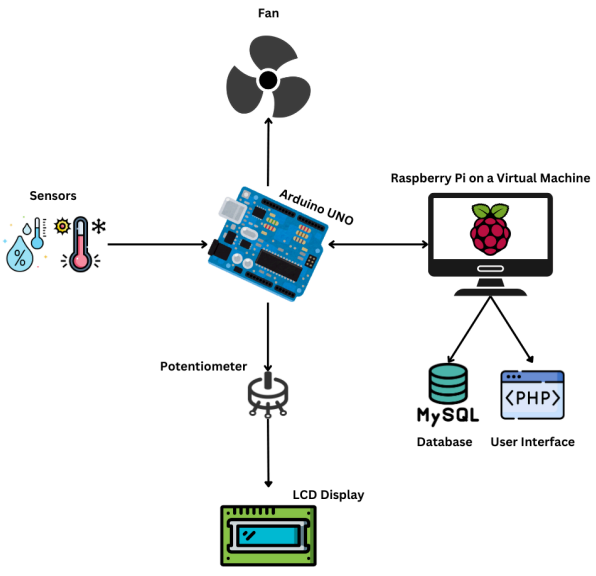

Fig. 1. The hardware/software used in the IoT Architecture

An Arduino Uno with sensors transmits environmental data to a Raspberry Pi (VM) for storage and analysis. A user interface in PHP allows for data visualization, historical analysis, and threshold control.

2.2. Flowchart:

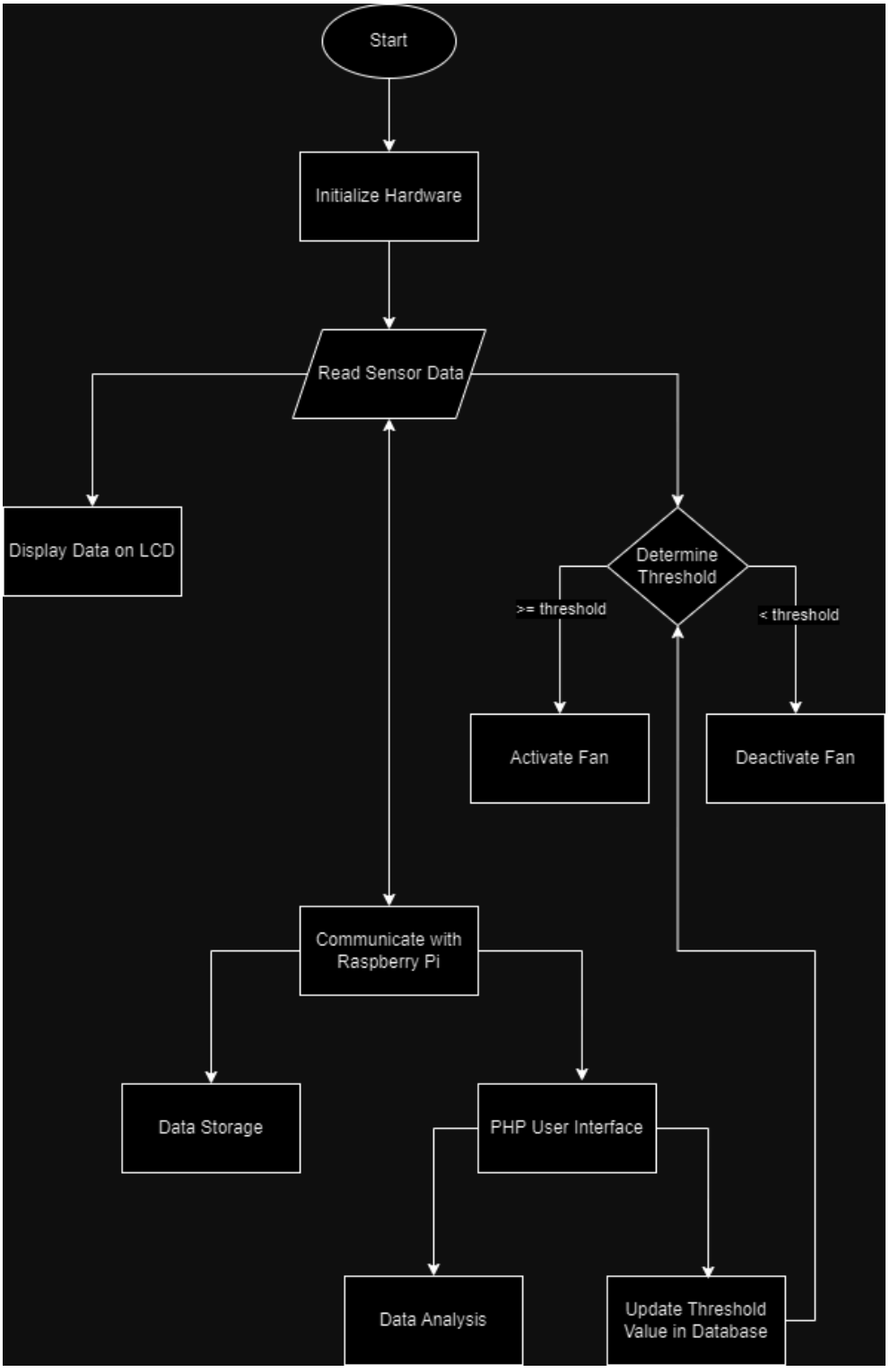

Fig. 2. The processes involved in the IoT project

This flowchart illustrates an IoT system for a smart fan, as well as temperature and humidity monitoring. An Arduino Uno collects data from DHT11 and LM35 sensors, displaying it on an LCD. The system then compares temperature readings to a user-defined threshold stored in a database. If exceeded, a fan is activated via the Arduino. Sensor data is then transmitted to a Raspberry Pi for storage and analysis, including calculating temperature averages and min/max values. The user interface in PHP on the Raspberry Pi allows for threshold adjustments and data analysis.

3. Implementation:

3.1. Sensors:

3.1.1. DHT11 Sensor:

A digital sensor measures both humidity and temperature. This sensor requires three pins on an Arduino Uno: VCC for power, OUT for data output, and GND for ground, with readings processed using the DHT sensor library and transmitted via serial communication to the Raspberry Pi on Virtual Box.

3.1.2. LM35 Sensor:

An analog sensor measures temperature. This sensor uses three pins when connected to an Arduino Uno for power (VCC), analog output (OUT), and ground (GND), with readings processed without the need for any special library and transmitted via serial communication to the Raspberry Pi on Virtual Box.

3.2. Actuators:

3.2.1. Small Fan Module:

A device activates to increase airflow. This fan module uses the GND pin for power negative (ground), the V pin (5V) for power positive, and the S pin (PWM input/digital input) for control on the Arduino board.

3.2.2. 16x2 LCD Display:

A display module is used to show real-time sensor data. This LCD is connected to the Arduino using digital I/O pins for power, data transfer, contrast control, and backlight functionality, with data displayed using the LiquidCrystal library.

3.3. Software/Libraries:

3.3.1. Libraries:

DHT Sensor library: Simplifies the process of obtaining temperature and humidity data from the DHT11 sensor.
LiquidCrystal library: Displays information on the 16x2 LCD screen.
Serial library: Enables data transfer between the Arduino and Raspberry Pi.
MySQLdb library: Provides a connection interface to interact with MySQL databases from Python script.
Datetime and Time library: Provides functions for manipulating dates and times or measuring time intervals.

4. Resources:

Learn Electronics - 16x2 LCD Pinout Diagram https://in.pinterest.com/pin/16x2-lcd-pinout-diagram-description-arduino-examples-and-applications--316096467598470833/

How to Interface 16x2 LCD with Arduino https://projecthub.arduino.cc/shreyas_arbatti/how-to-interface-16x2-lcd-with-arduino-a1a0f1

LM35 Temperature Sensor https://www.ti.com/lit/gpn/LM35

DHT11 Temperature Sensor https://components101.com/sensors/dht11-temperature-sensor

pyserial API https://media.readthedocs.org/pdf/pyserial/latest/pyserial.pdf

Python User-Defined Functions https://www.geeksforgeeks.org/python-functions/

Serial Communication Functions (Arduino Reference) https://www.arduino.cc/en/Reference/serial

String Object (Arduino Reference) https://www.arduino.cc/reference/en/language/variables/data-types/string

Submitting AJAX Forms with jQuery https://www.digitalocean.com/community/tutorials/submitting-ajax-forms-with-jquery

Runway Web Application (for the background image) https://app.runwayml.com/dashboard

Color Hunt - Color Palette (for the User Interface) https://colorhunt.co/palette/ffc7c7ffe2e2f6f6f68785a2

5. Appendix:

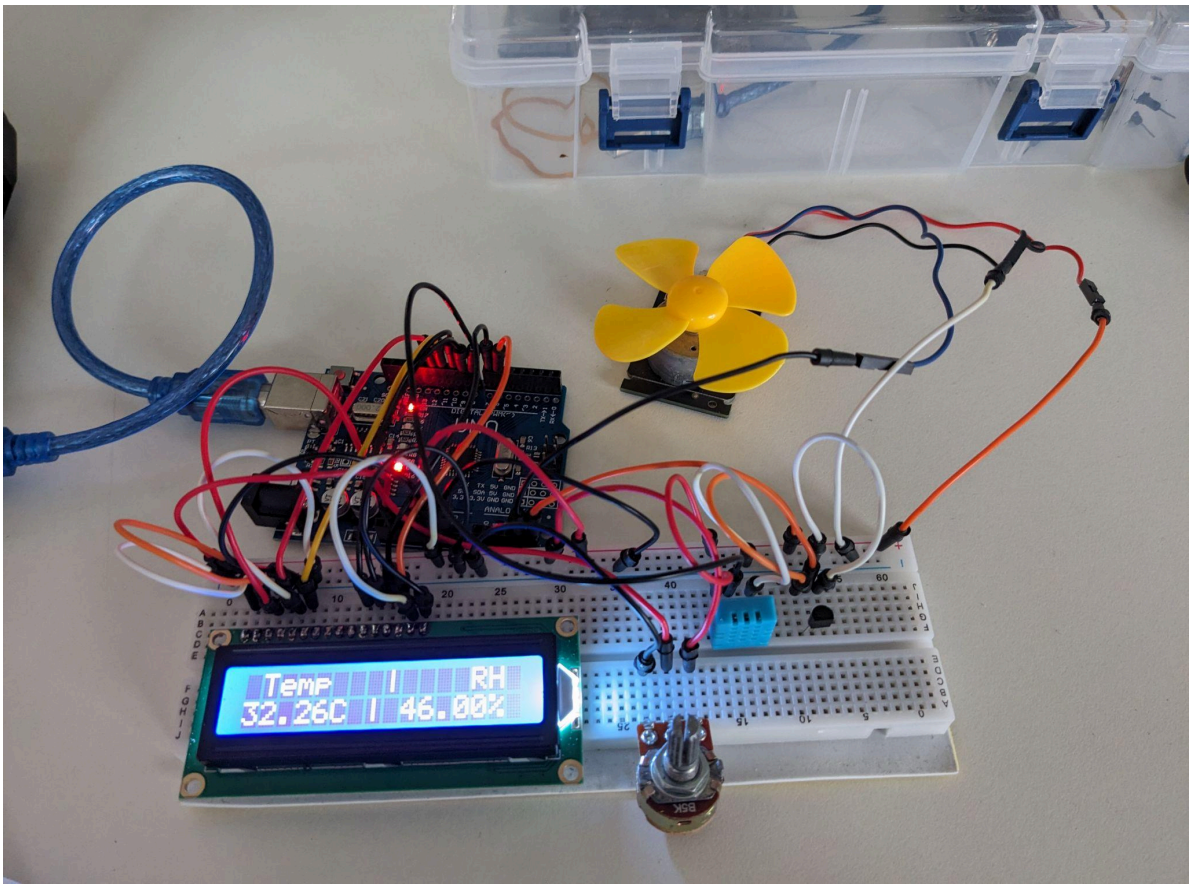

Fig. 3. My IoT Node Setup and Application

```
.
├── arduino.cpp
├── rpi.py
├── style.css
├── update_treshold.php
├── weather.gif
└── weather.php
```

Fig. 4. My directory list of files

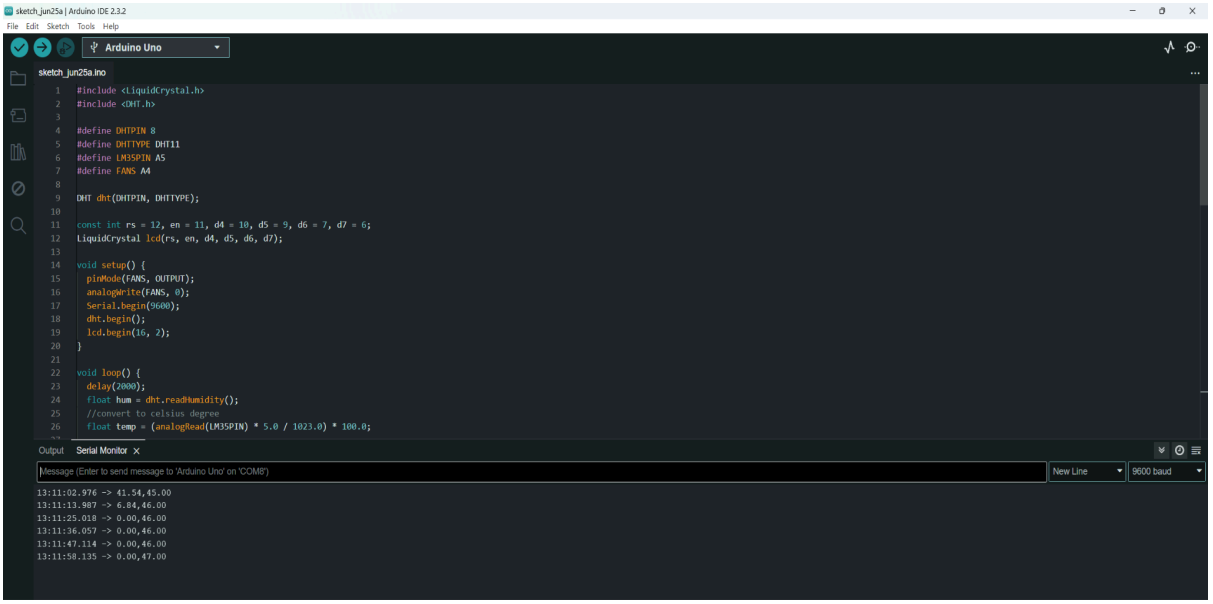

Fig. 5. Serial Monitor showing the data output from the Arduino Uno

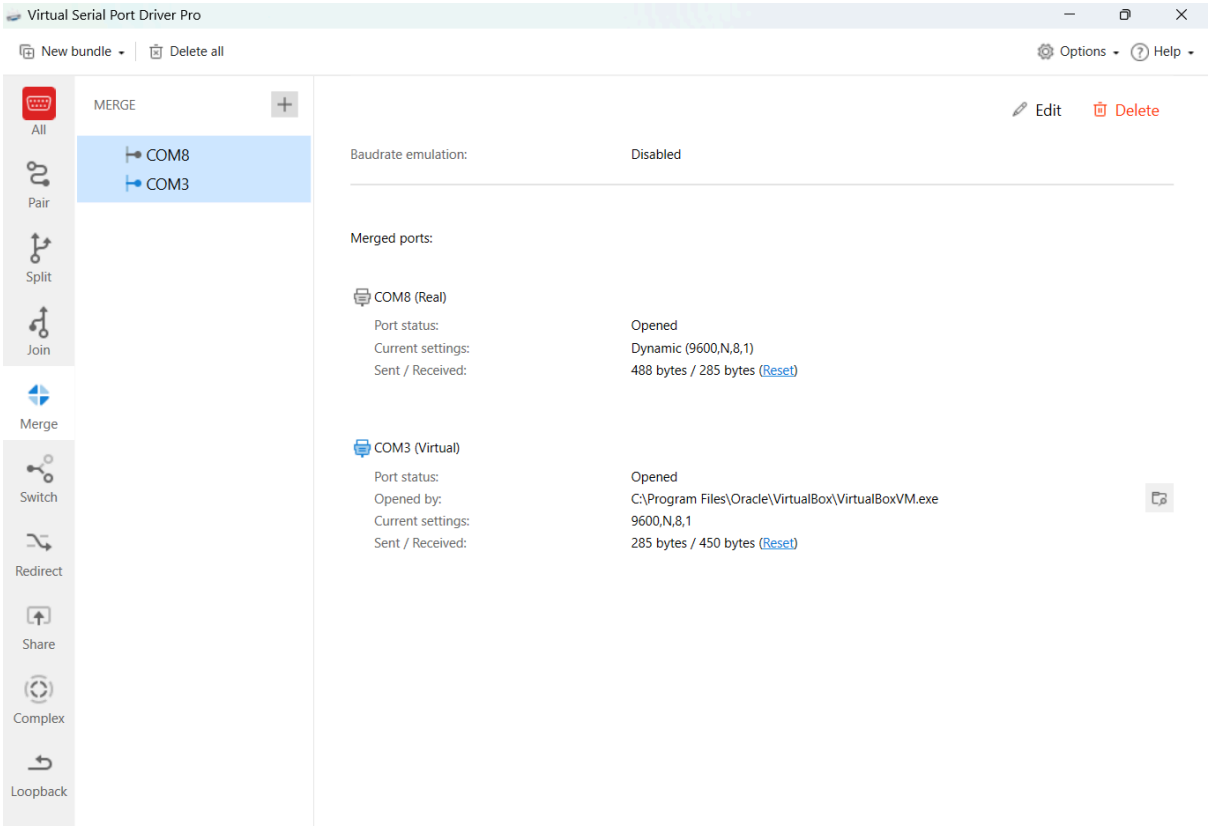

Fig. 6. Emulating a serial connection through virtual COM ports between the Arduino Uno and the Raspberry Pi on VM

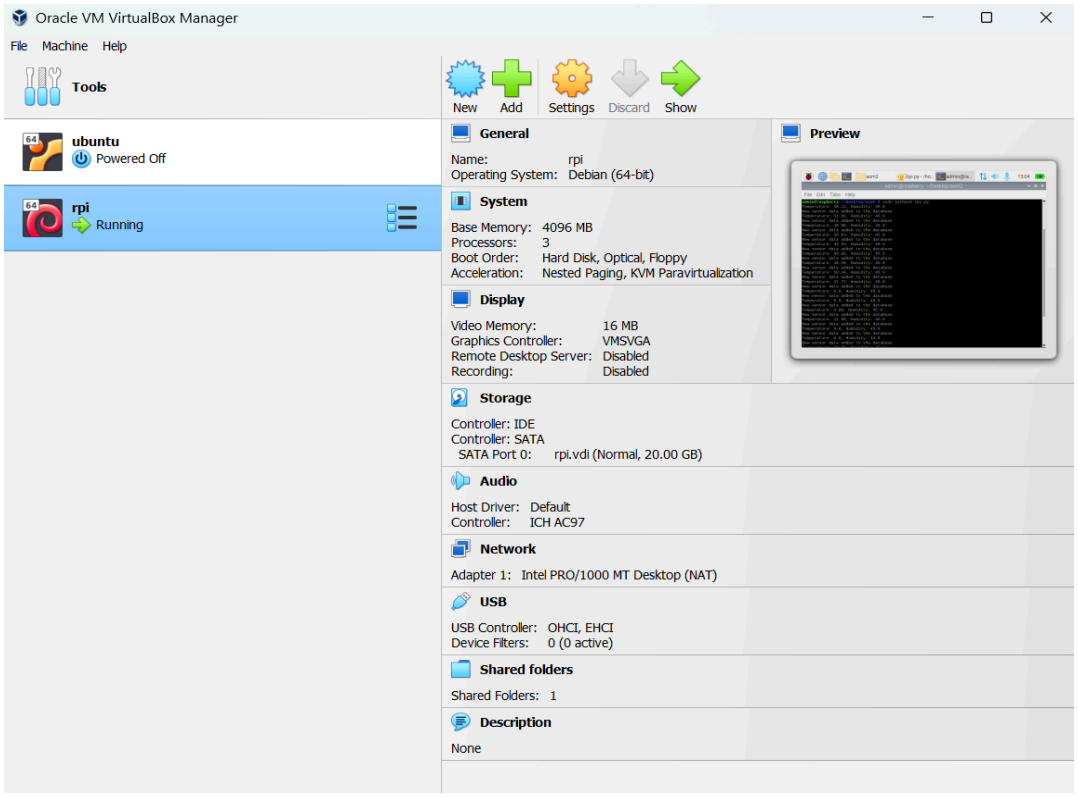

Fig. 7. Raspberry Pi on VM

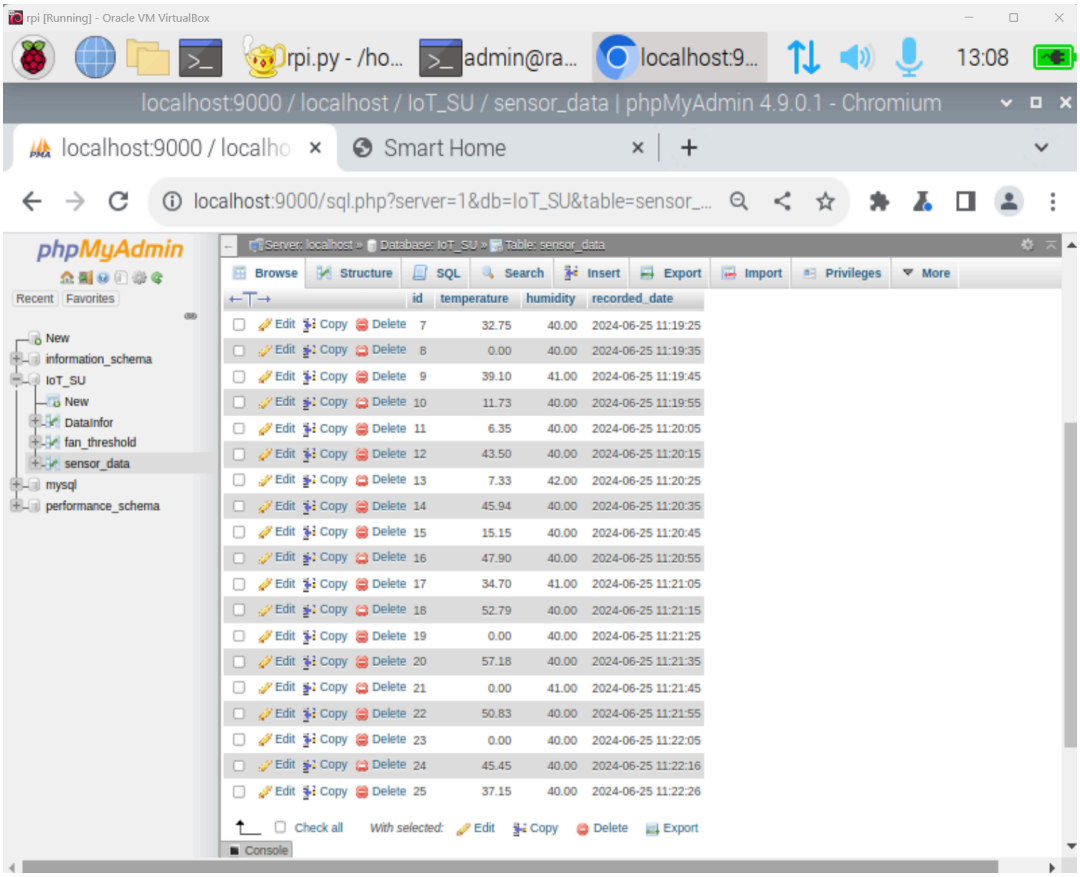

Fig. 8. The sensor_data table

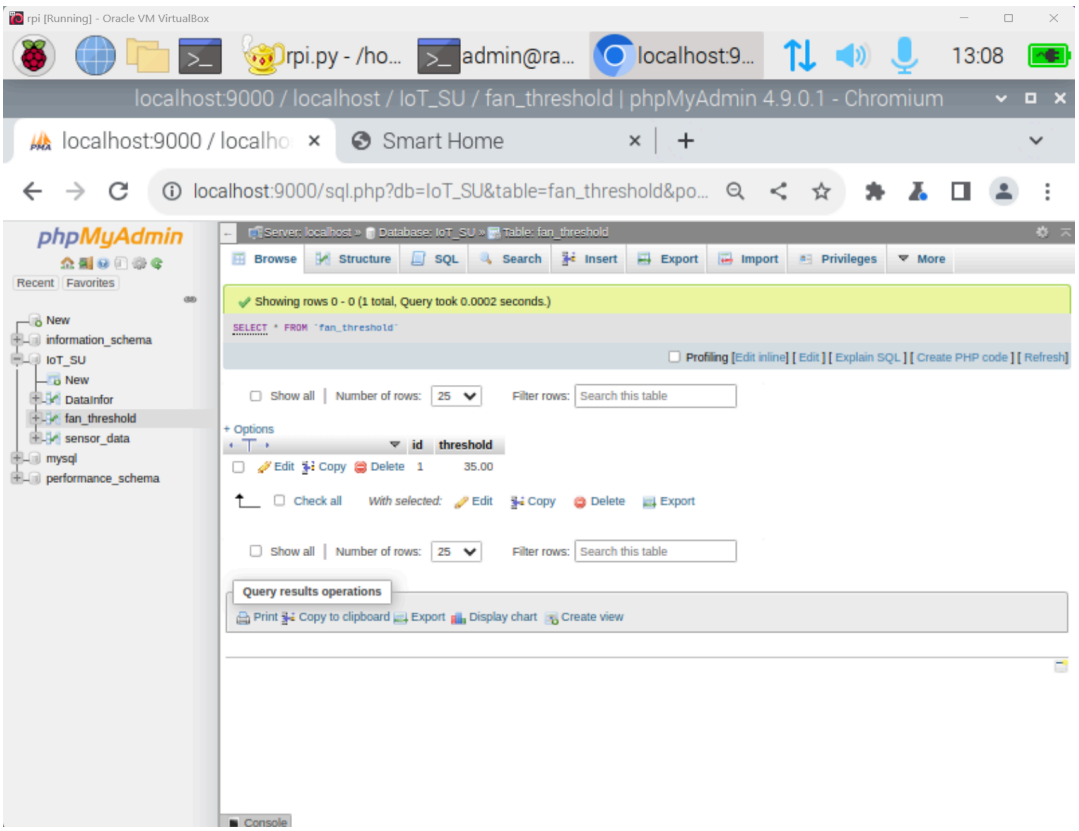

Fig. 9. The fan_threshold table

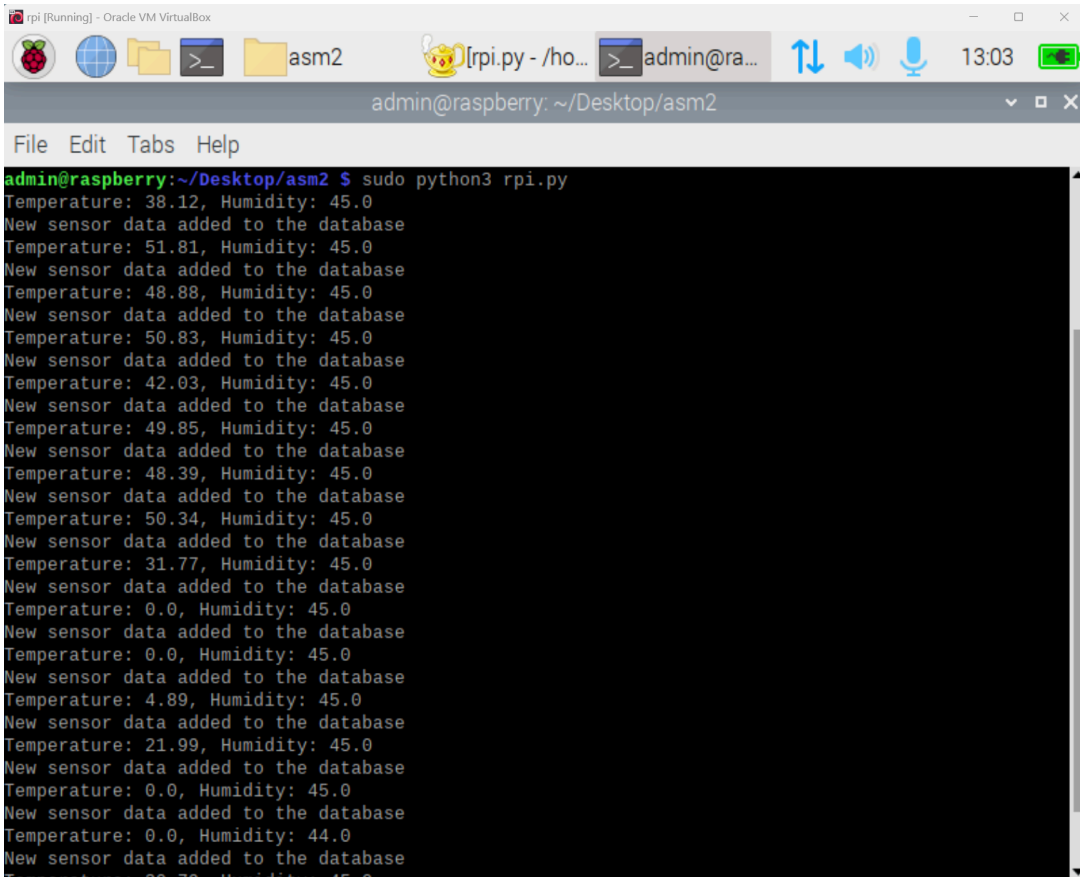

Fig. 10. The Python script to store the sensor data on the database and control the fan

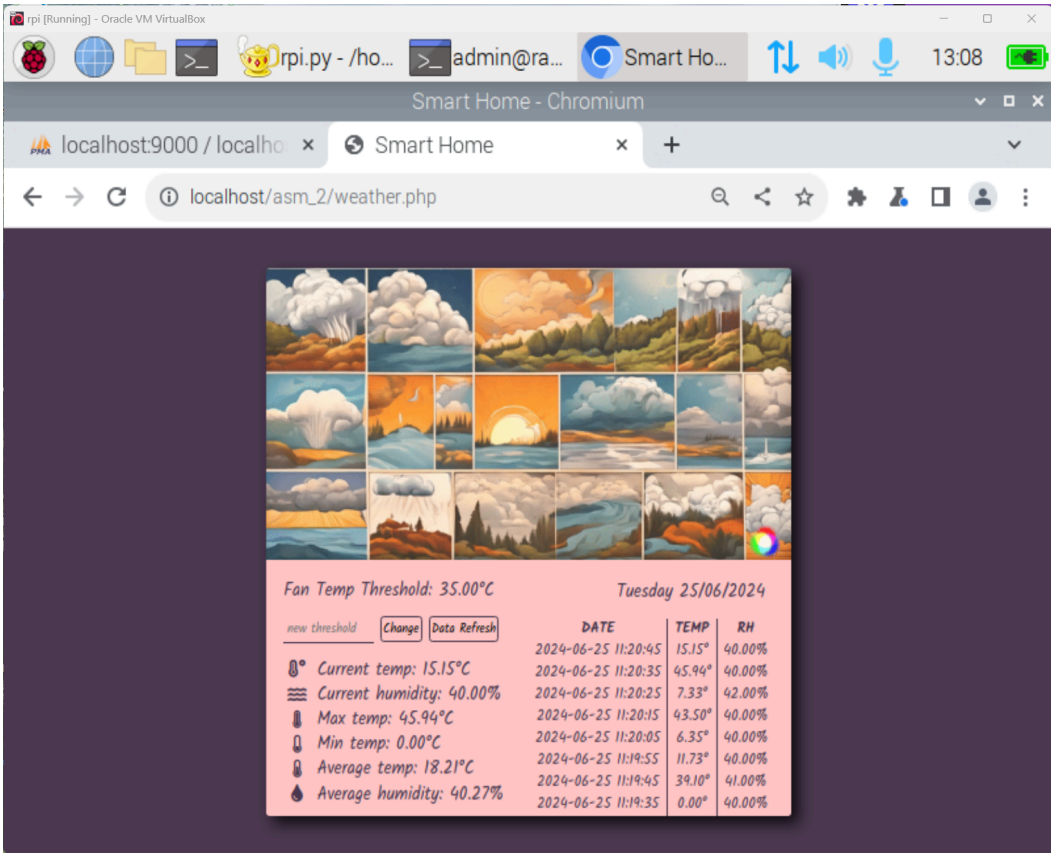

Fig. 11. The User Interface displaying information from the database

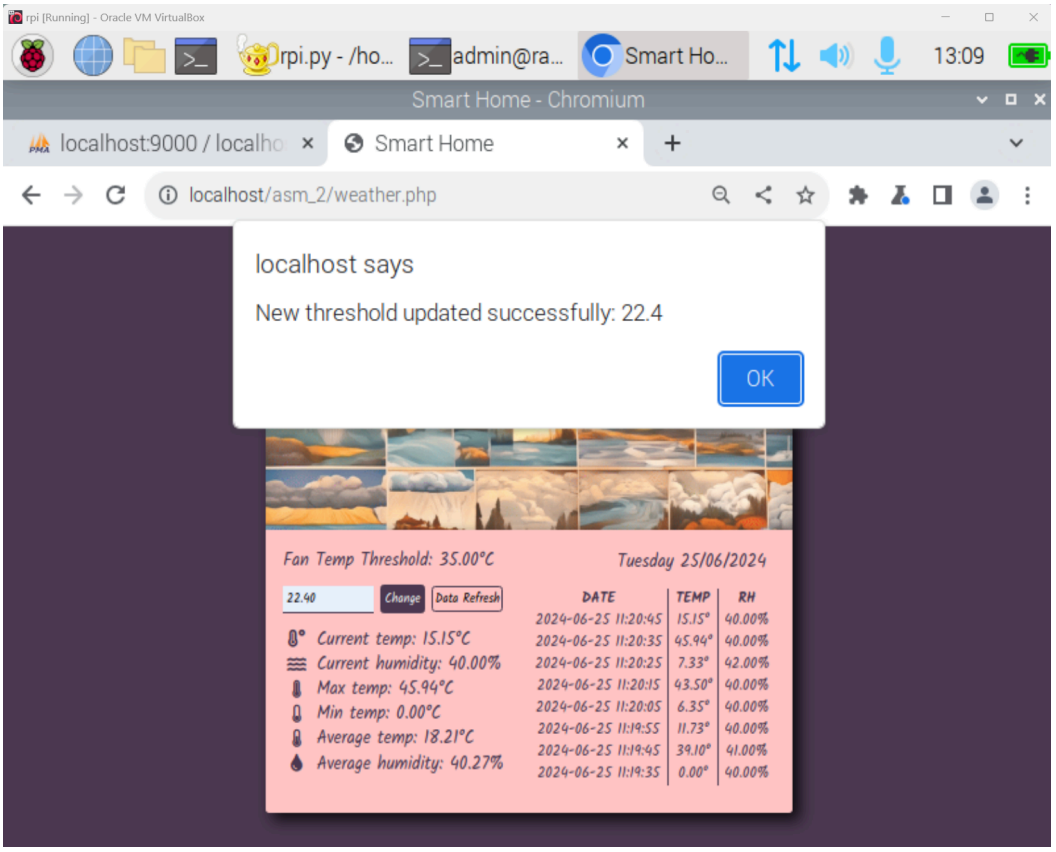

Fig. 12. User changing the threshold of the fan