Weather Station Report

Dontae Whitely (620157609)
The University of the West Indies
ELET 2415 - Practices in Electronics Designs II
April 12, 2024

Table of Contents

Introduction……..……....2
Design Section……3
Development…….……13
Components and Equipment……..16
Conclusion……17
Refernce……18

Introduction

This report entails the design and implementation of a weather station, which is an Internet of Things (IoT) system. The Internet of Things, or IoT, is a network of physical devices. These devices are inter-connected and they can transfer data to one another without human intervention. The Internet of Things connects the physical and digital realms using a range of technologies. Physical objects are embedded with sensors, which can monitor things like motion, temperature, humidity, or really any change in the environment. The devices communicate via wired (for example, Ethernet) or wireless (for example, WiFi or cellular) networks. The device developed is a weather station that was tasked with capturing and saving certain meteorological data, such as “air temperature," "air pressure," "altitude," "humidity," "heat index," and "soil moisture." This data is then available for display at the physical weather station and remotely from any location through a web-based graphical user interface (GUI). All of these functionalities are achieved through the use of an ESP32 microcontroller, an array of sensors to acquire environmental variables, and a frontend and backend system to process, store, display, and transmit this data.

System Description and Specifications

This system possesses the capability to measure various meteorological variables, such as air temperature, humidity, air pressure, and more. Moreover, the system facilitates the viewing of these parameters at the physical weather station and remotely from any location through a web-based graphical user interface (GUI).

The Embedded System Weather Station:

● Reads and processes information from diverse sensors, incorporating a minimum set of parameters such as "air temperature," "air pressure," "altitude," "humidity," "heat index," and "soil moisture."

● Processes and transmits sensor data through a WiFi connection to section 2 of the system, which involves web-based data collection, processing, storage, and display.

Web-Based Data Collection, Processing, Storage, and Display System:

● Collects sensor data from section 1 of the system, which is the embedded system weather station.

● Processes and stores the collected data in an online database.

● Provides access to and displays the stored data through a user-friendly graphical user interface (GUI) in a user-friendly format.

Design Section

Hardware

Figure 1, shown below, is a block diagram used to outline all the components necessary to make the weather system operational. At the heart of the entire system is the ESP32 microcontroller, which is responsible for the intended functions. Also shown below are the necessary sensors that receive data from the environment and transmit it to the ESP32 for processing. The TFT SPI displays all information received from the ESP32.

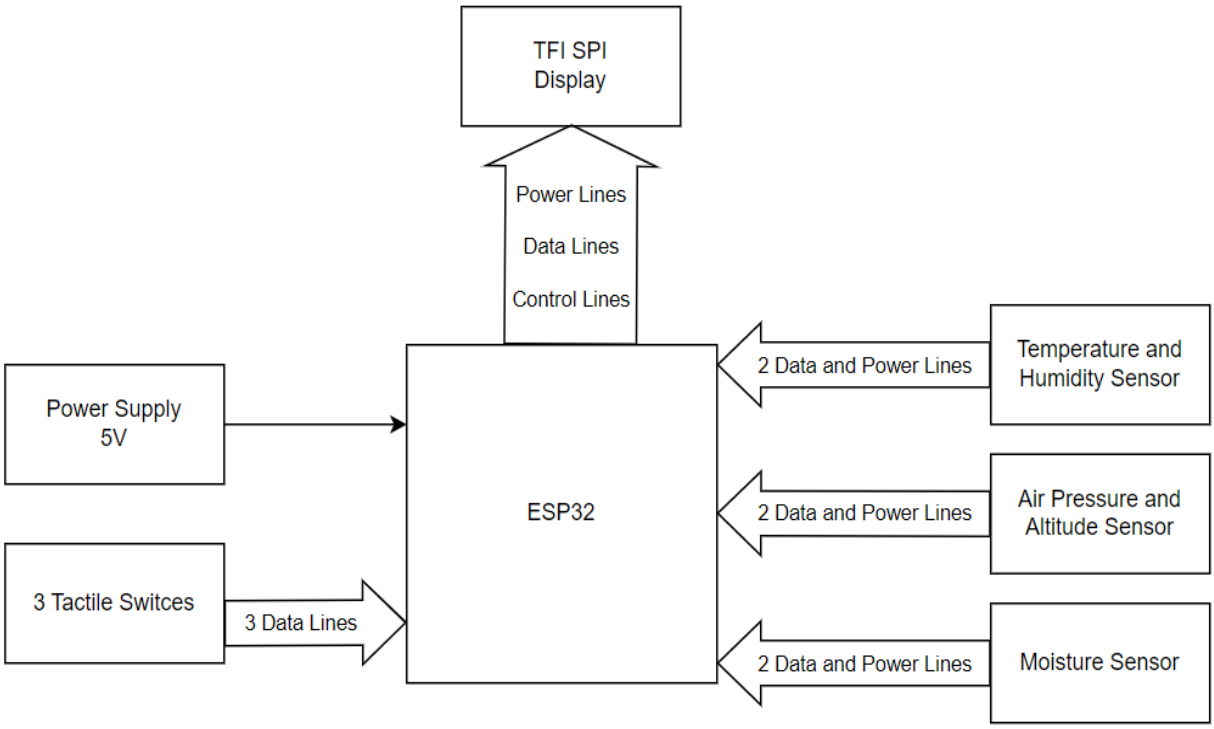


Figure 1

Figure 2: ***Weather Station Hardware Schematic***

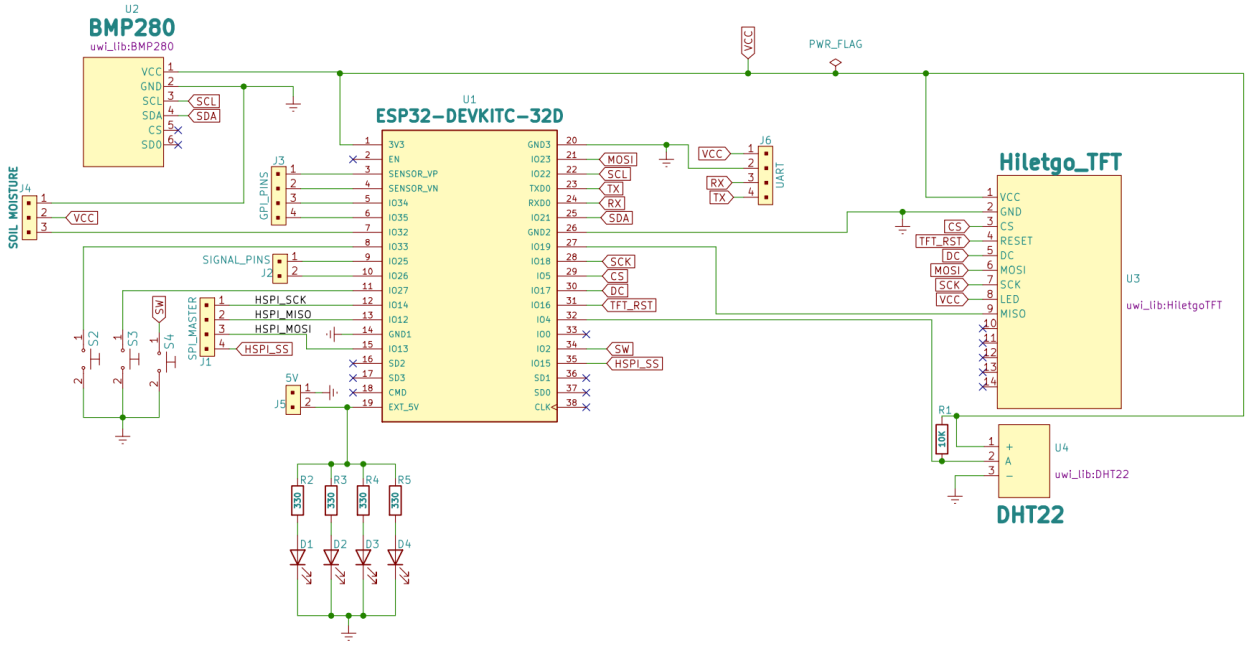
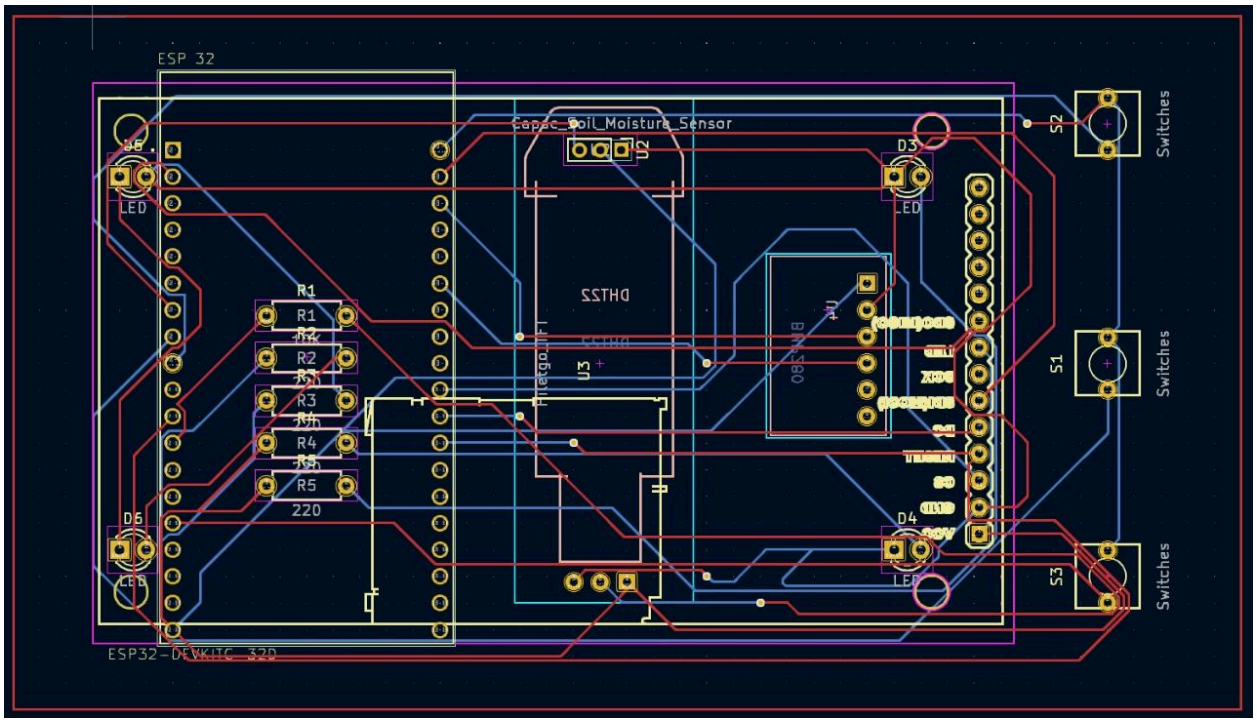


Figure 3: Weather Station PCB



Backend

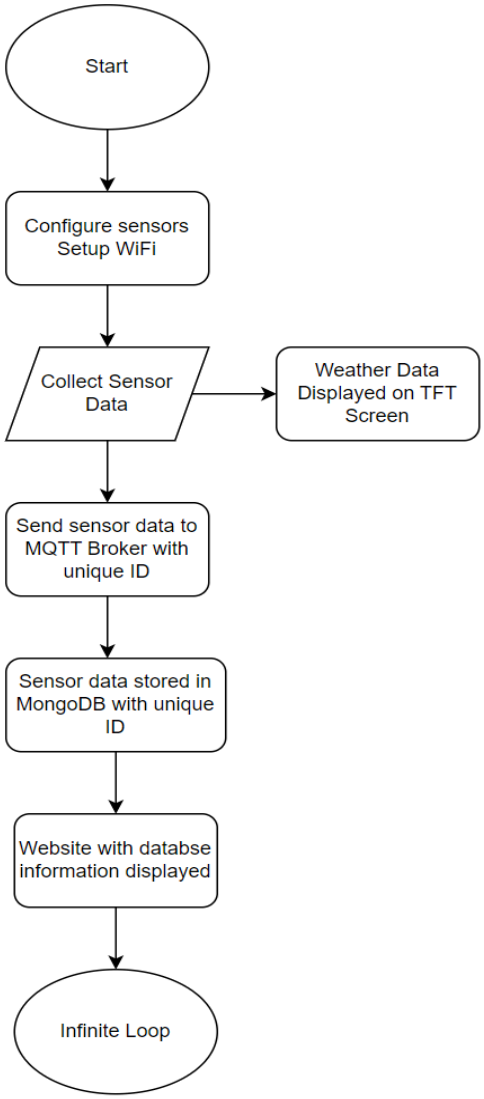


Figure 4

The backend allows for communication between the hardware and the frontend. It accomplishes this task through the use of two primary interfaces. These interfaces are Message Queuing Telemetry Transport (MQTT) and MongoDB. Message Queuing Telemetry Transport (MQTT) acts as a communication hub. Devices publish sensor data, and the MQTT broker, Mosquitto, securely manages this data, ensuring only devices that are authorized can publish and only intended parties can receive it. Our ESP32 device utilizes MQTT to publish the weather system data. MongoDB, a document-based database, serves as the data storage medium. The formatted data is stored in JSON-like documents within MongoDB, ready to be accessed for further analysis and visualization. This stored data is then available for graphicalized viewing for users of the system through the frontend. The backend essentially connects the different interfaces, allowing them to communicate.

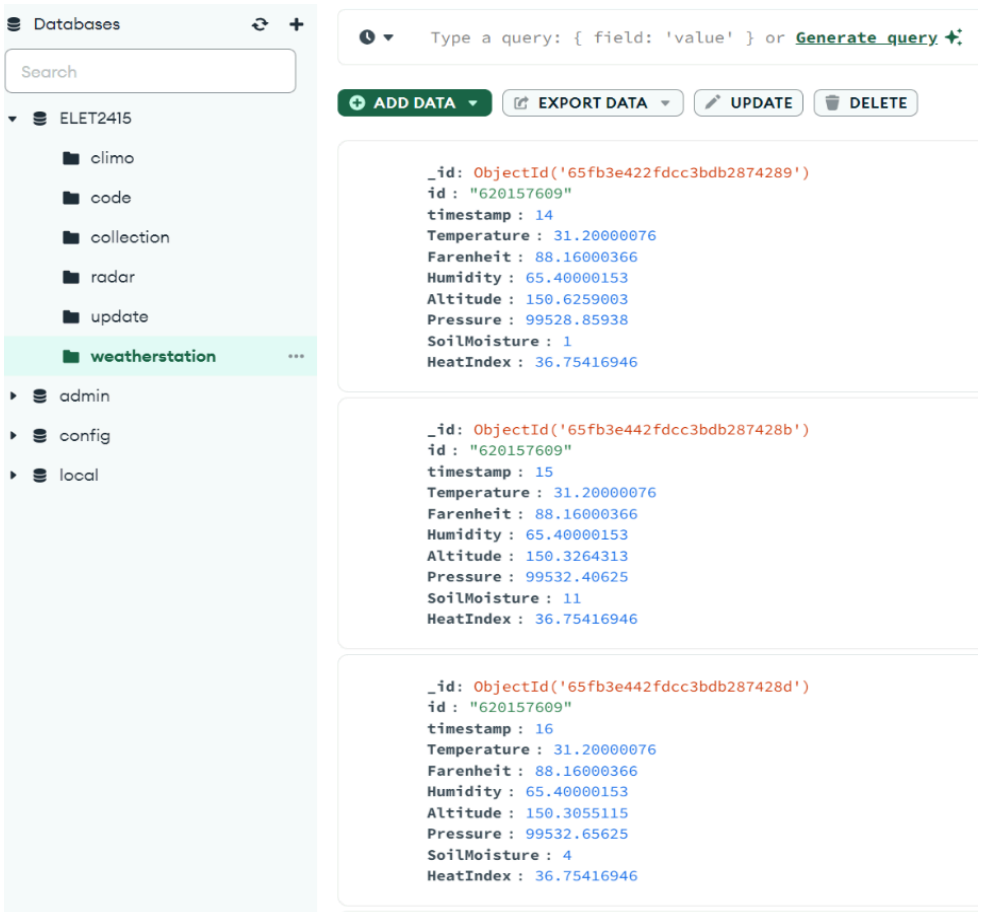
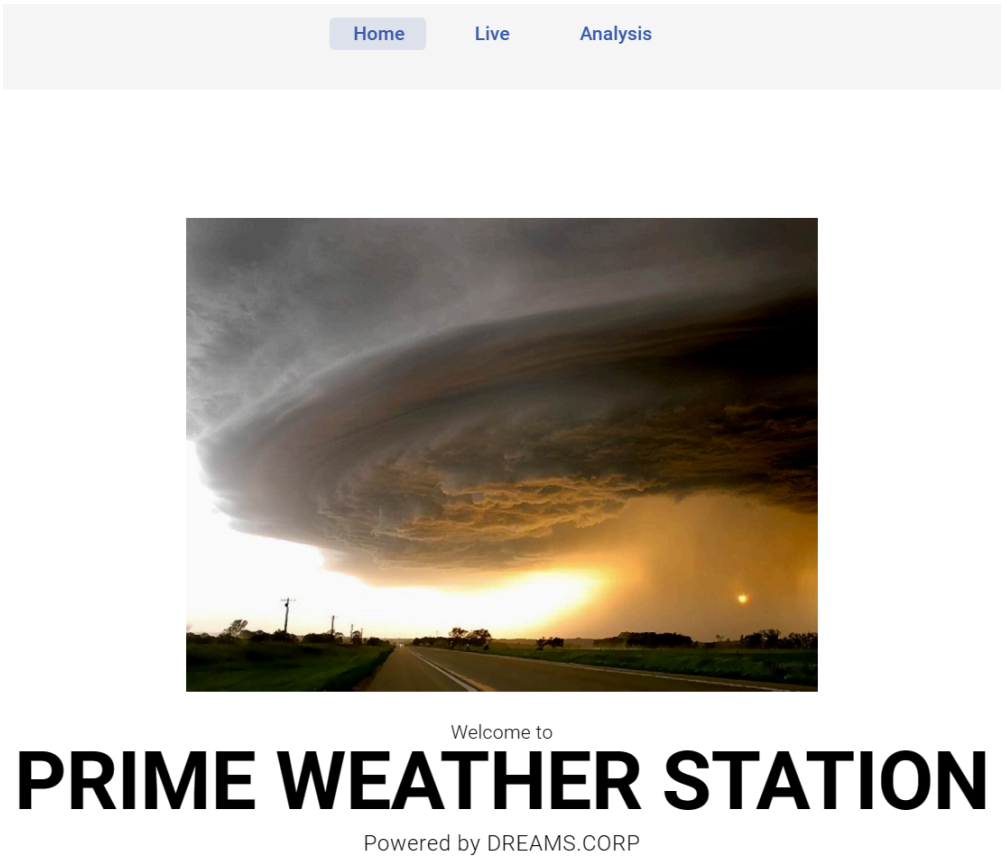


Figure 5: MongoDB (Weather Station database)

Frontend

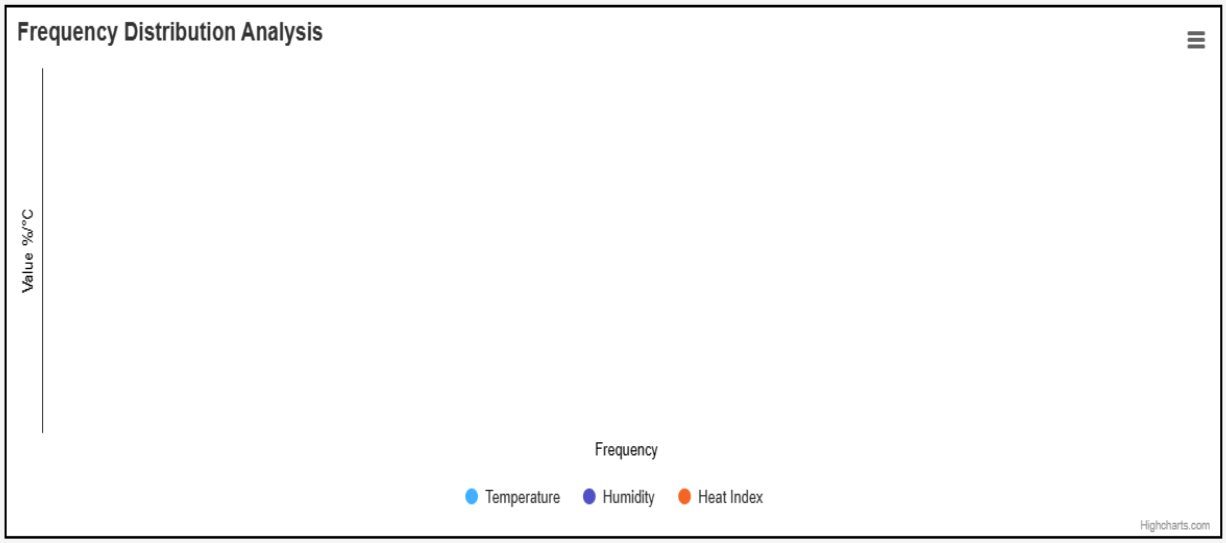
The front end of an IoT system is all about user interaction. It's the stage where the collected data is transformed into a user-friendly format, allowing users to visualize trends, monitor devices, and interact with the system. The system facilitates the viewing of the different sensor information remotely from any location through a web-based graphical user interface (GUI).

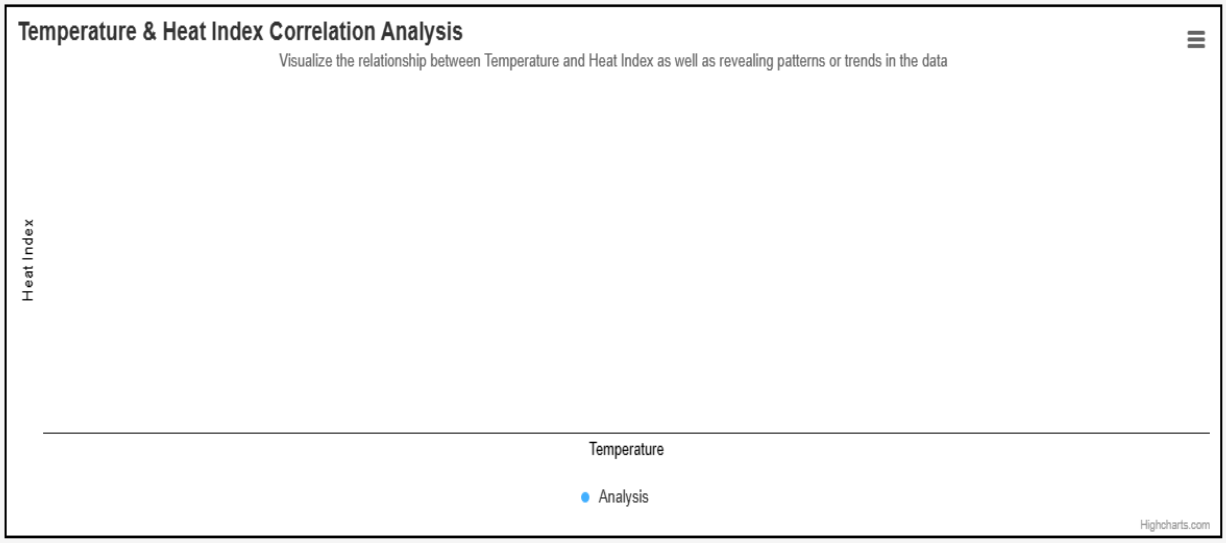


Weather Station Home Page

Weather Station Live Page

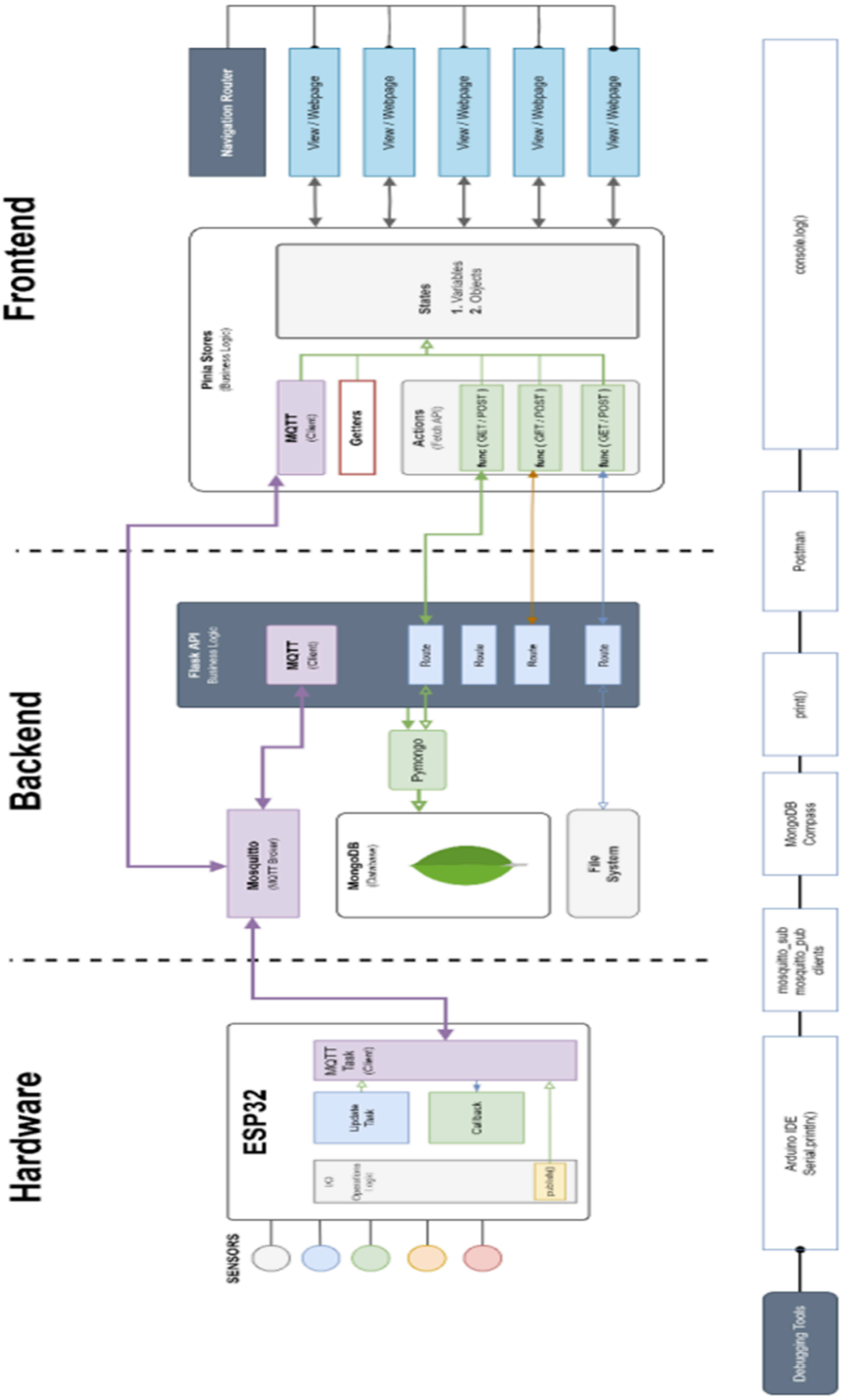

Home
Live
Analysis
Enter date range for Analysis
Start date
End date
ANALYZE
Temperature
0 0 0
0
Humidity
0 0 0
0
Air Temperature and Heat Index Analysis
The heat index, also known as the apparent temperature, is a measure that combines air temperature and relative humidity to assess how hot it feels to the human body. The relationship between heat index and air temperature is influenced by humidity levels. As humidity increases, the heatindex also rises, making the perceived temperature higher than the actual air temperature.
Air Temperature & Heat Index
Time
Temperature
Heat Index
Highcharts.com
Humidity Analysis
Humidity
Time
Humidity
Highcharts.com


Frequency Distribution Analysis
Value %/°C
Frequency
Temperature
Humidity
Heat Index
Highcharts.com


Temperature & Heat Index Correlation Analysis
Visualize the relationship between Temperature and Heat Index as well as revealing patterns or trends in the data
Heat Index
Temperature
Analysis
Highcharts.com

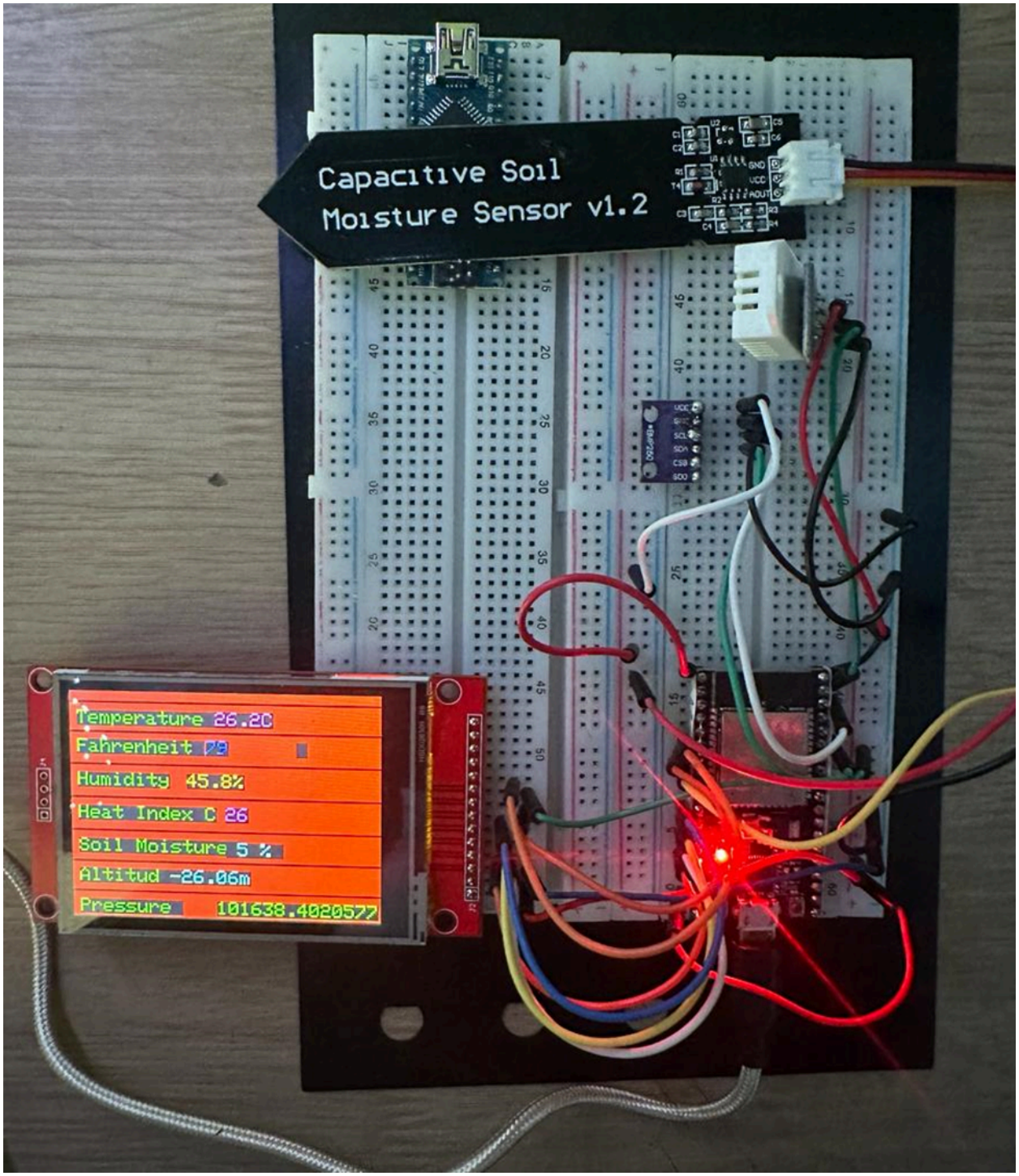

Humidity & Heat Index Correlation Analysis
Visualize the relationship between Humidity and Heat Index as well as revealing patterns or trends in the data
Heat Index
Humidity
Analysis
Highcharts.com

Weather Station Analysis Page


Frontend
Backend
Hardware
Navigation Router
View / Webpage
View / Webpage
View / Webpage
View / Webpage
View / Webpage
Pinia Stores
(Business Logic)
States
1. Variables
2. Objects
MQTT
(Client)
Getters
Actions
(Fetch API)
func (GET / POST)
func (GET / POST)
func (GET / POST)
Flask API
Business Logic
MQTT
(Client)
Route
Route
Route
Route
Pymongo
Mosquitto
(MQTT Broker)
MongoDB
(Database)
File
System
ESP32
MQTT
Task
(Client)
Update
Task
Callback
I/O
Operations
Logic
publish()
SENSORS
Debugging Tools
Arduino IDE
Serial.println()
mosquitto_sub
mosquitto_pub
clients
MongoDB
Compass
print()
Postman
console.log()

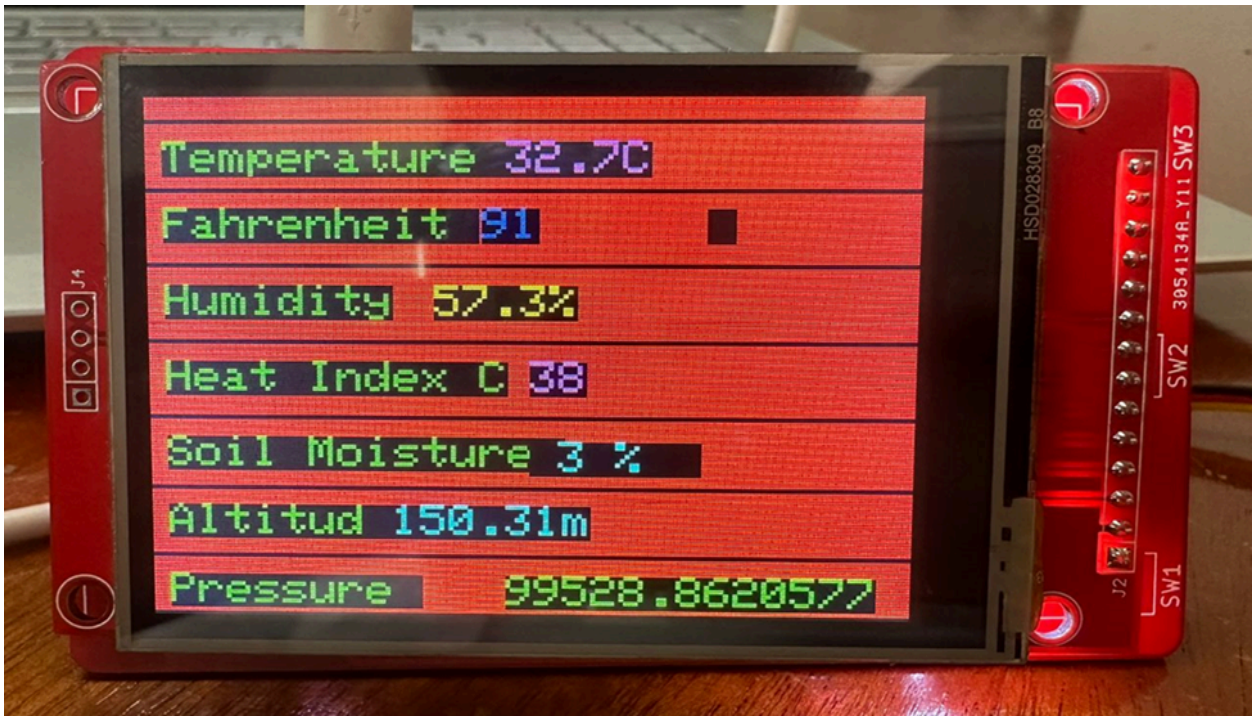
Development

The development period for the weather station lasted for a period of 12 weeks. The development period consisted of weeks worth of research and development, equipment and components acquisition as well as the weather station construction.

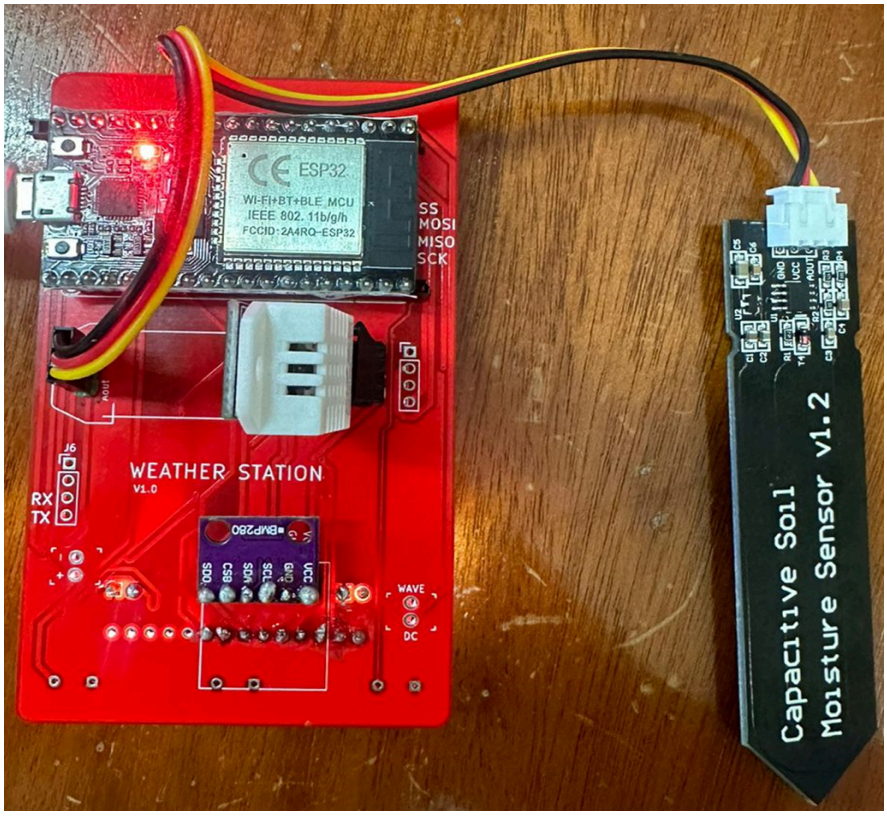


Bread Board during development period.

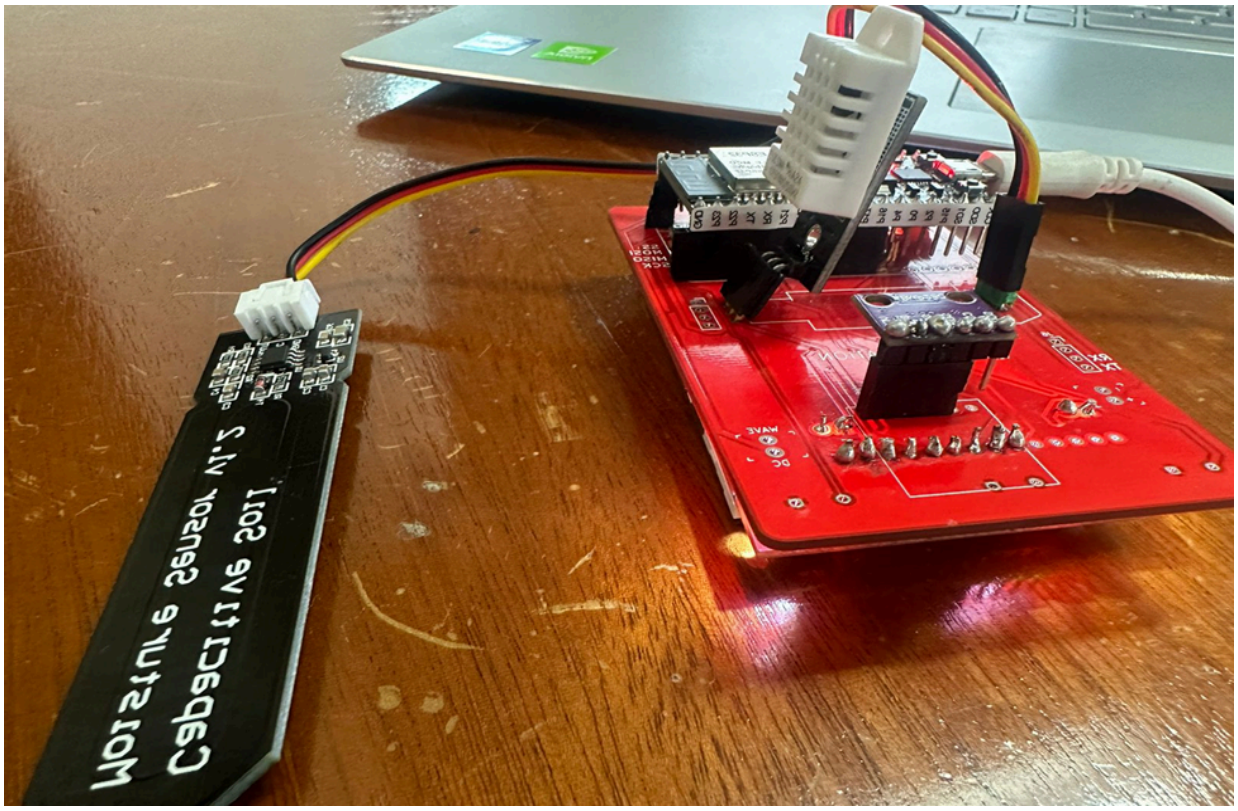
Final Product- Front View



View of the back



Side View



Components and Equipment

Below is a list of the components and equipment used to design and develop the weather station.

Components:	**1**	ESP 32 Module	$2,200.00
	1	Weather Station Blank PCB	$500.00
	1	Soil Moisture Sensor	$1,200.00
	1	DHT22 Humidity Sensor	$1,200.00
	1	Pressure Sensor	$850.00
	1	2.8” TFT Display Module	$3,200.00
	1	10k ohm Resistor	$100.00
	4	220 ohm Resistor	$200.00
	4	LED	$350.00
Equipment:		Computer with the latest version of: ● Microsoft VS Code Editor ● Postman ● KiCad Suite ● Arduino IDE ● Eclipse Mosquitto MQTT Broker ● Node.js JavaScript Engine MongoDB Database	
	1	Micro USB Cable	

Total cost of Weather Station: $9800

Conclusion

This project has been instrumental in the development of my technical abilities, specifically those related to designing and developing solutions, using the fundamentals of IoT. Throughout the development period of this weather station, I have garnered a profound understanding of Internet of Things devices and their impact on the environment around them. These devices work seamlessly within the environment, essentially connecting the physical and digital realms with the overarching goal of making daily tasks as simple as possible. I must also make mention that, it is through this assignment that I have learned how versatile the ESP32 microcontroller is, being able to be programmed to work in multiple different configurations to carry out different tasks. Many challenges were encountered during this experiment, such as having to learn the correct syntax for various html, css, and JavaScript components, replacing defective components, ensuring that a proper connection is made between the hardware, backend, and frontend, and generally having to understand how to code and configure for the ESP32. Although these tasks were at times challenging, it was nevertheless a challenge that I have deep and resounding admiration for what it has done for the growth of my technical abilities. The weather station constructed was able to meet the requirements outlined. However, it should be noted that improvements could have been made in the overall system design, such as having an eloquent-looking enclosure for the weather system itself. Despite that, the skill set that I have attained regarding configuring and creating an IoT device will remain with me as I leave this university and enter the professional world.

References

DontaeW. “DontaeW/Weather-Station.” *GitHub*, 28 Mar. 2024, github.com/DontaeW/Weather-Station.git. Accessed 11 Apr. 2024.

IBM. “What Is the Internet of Things?” *Www.ibm.com*, 2024, www.ibm.com/topics/internet-of-things.

KiCad, v7.0.10

Schulze, Jessica. “What Is the Internet of Things (IoT)? With Examples.” *Coursera*, 16 June 2023, www.coursera.org/articles/internet-of-things.

“Tajay59 - Overview.” *GitHub*, github.com/tajay59. Accessed 8 Apr. 2024.

“What Is IoT: The Internet of Things Explained | McKinsey.” *Www.mckinsey.com*, www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-the-internet-of-things.

“What Is the Internet of Things? | IBM.” *Www.ibm.com*, www.ibm.com/topics/internet-of-things.