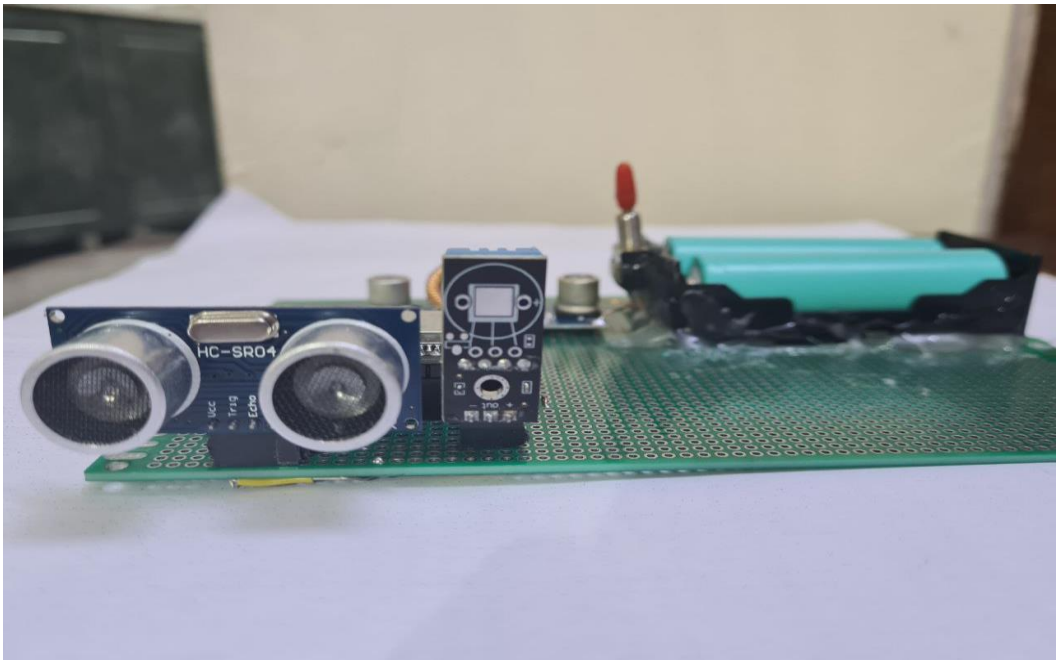
PROJECT IMPLEMENATION REPORT

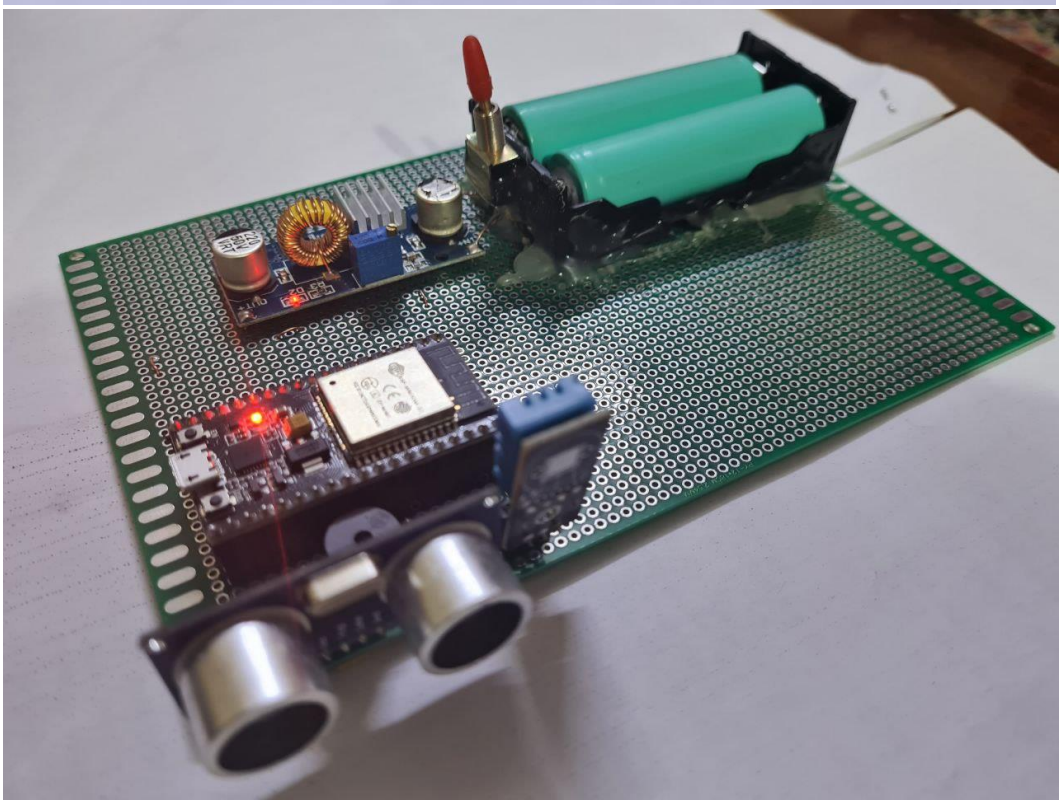
ITE 412

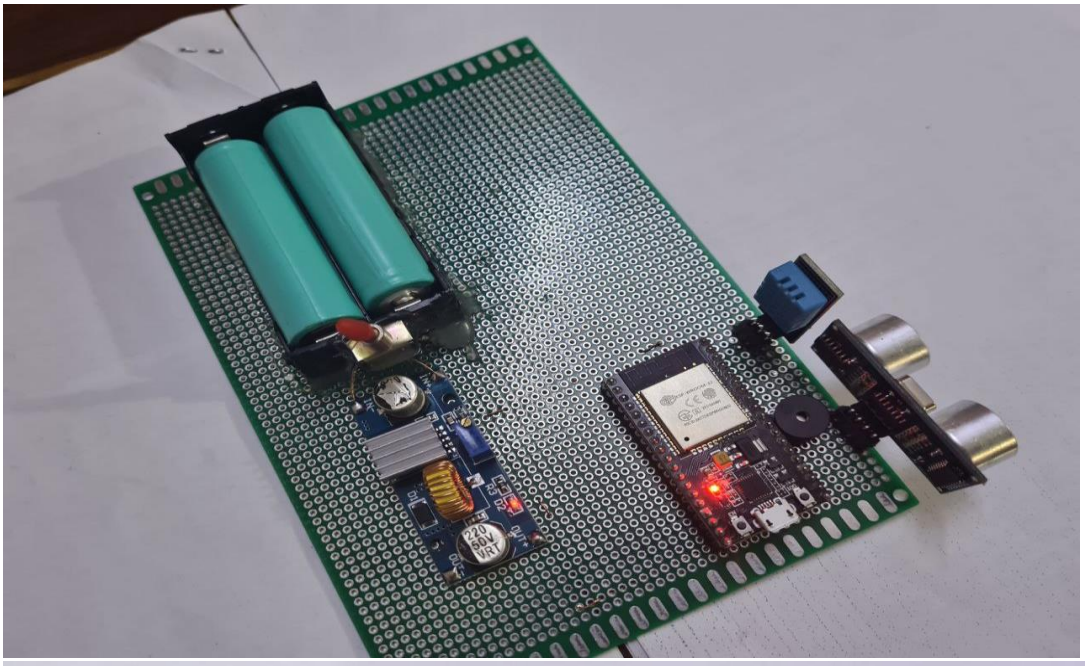
Supervised By Dr. Yad Tahir

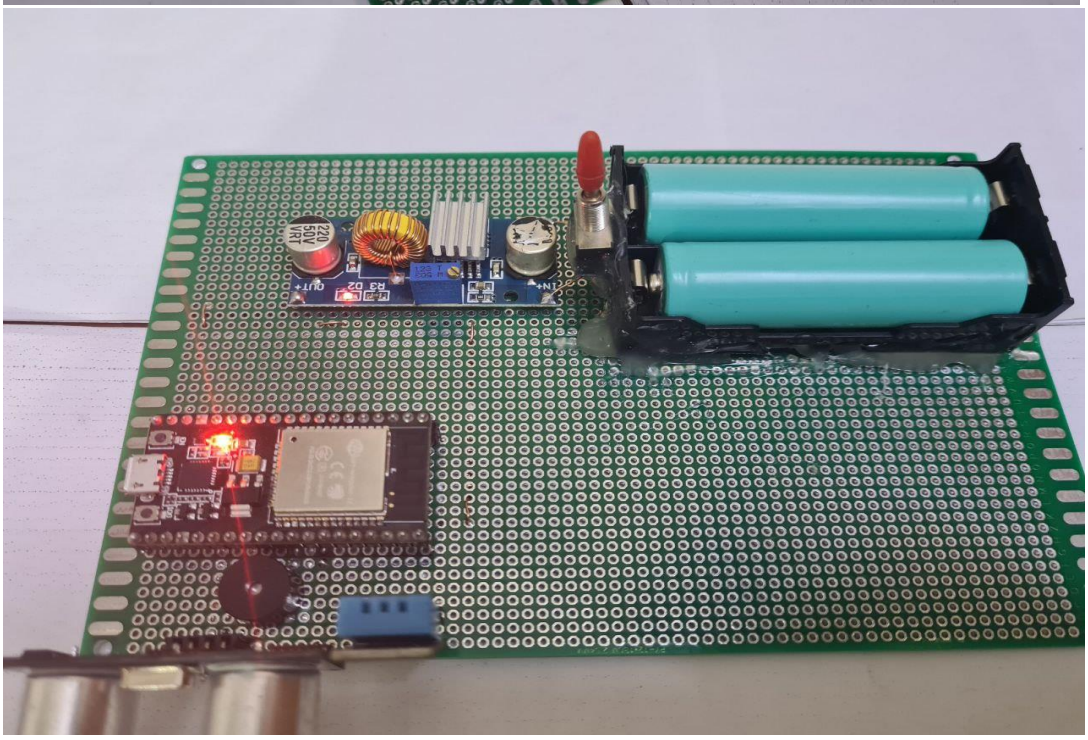
Ahmed Basim Naser ab16367@auis.edu.krd
Salih Saad ss17039@auis.edu.krd

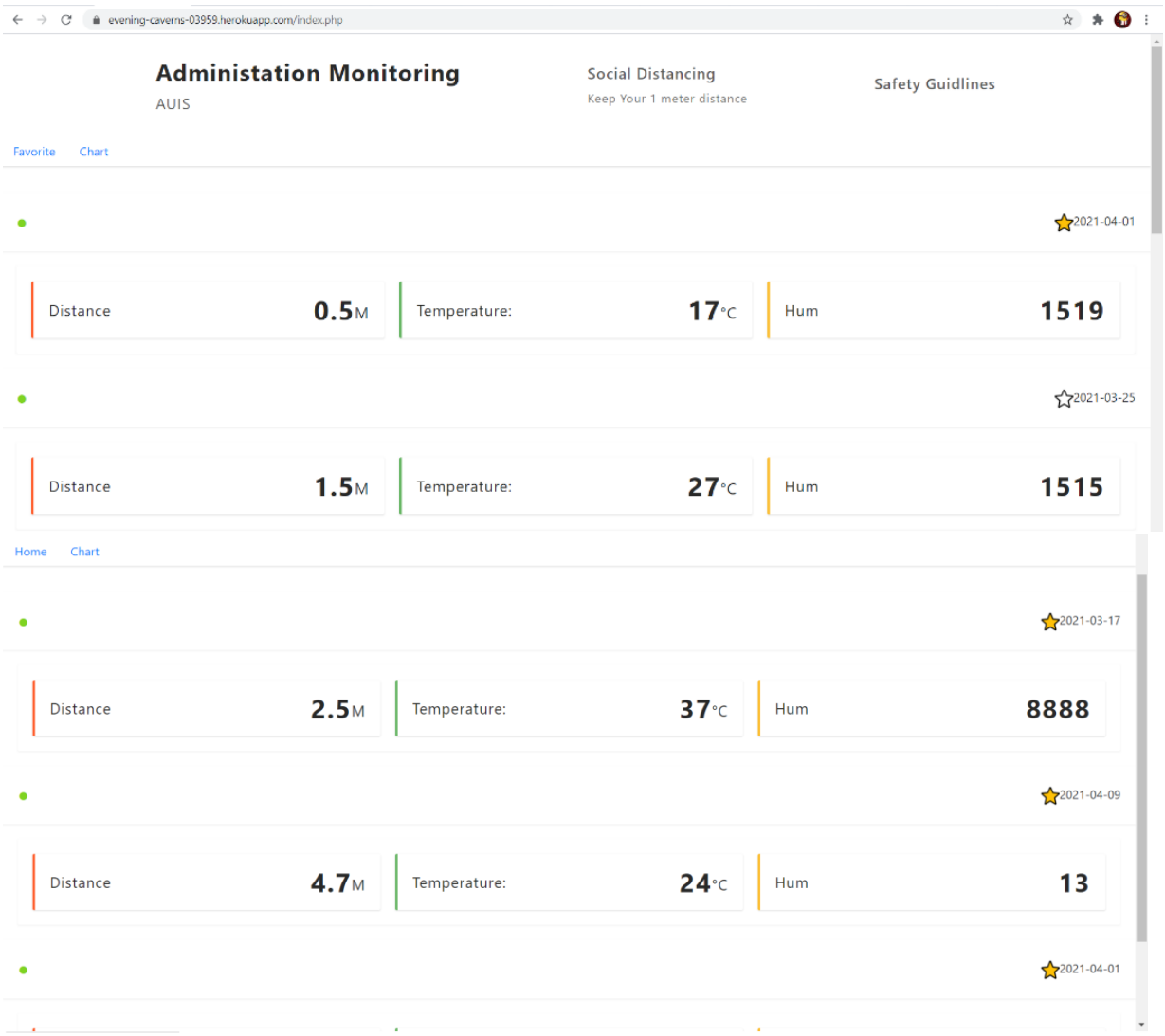
Demo and User Interface Screenshots:

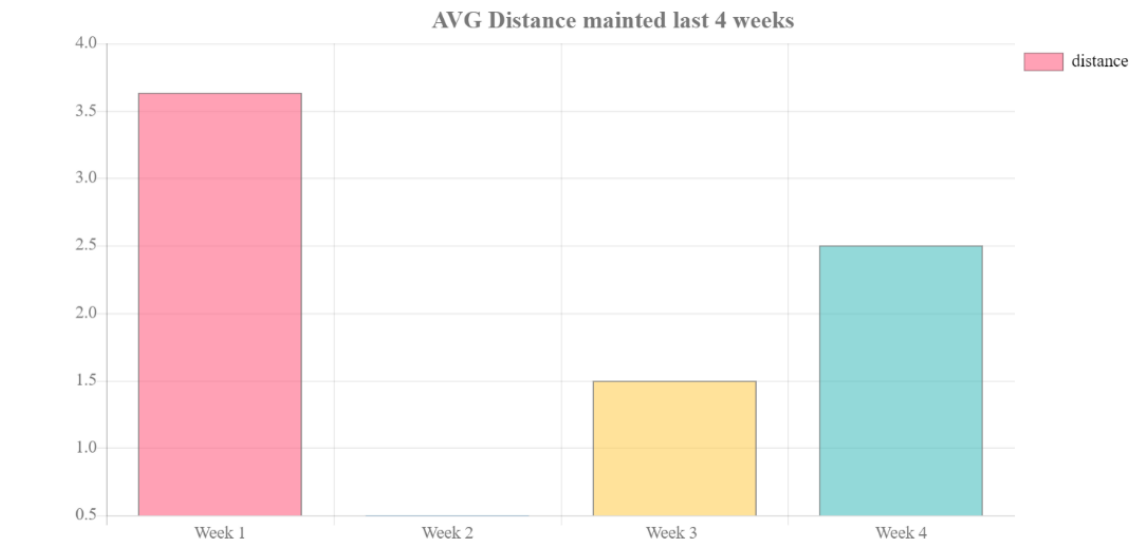









evening-caverns-03959.herokuapp.com/index.php
Administation Monitoring
AUIS
Social Distancing
Keep Your 1 meter distance
Safety Guidlines
Favorite
Chart
2021-04-01
Distance 0.5M
Temperature: 17°C
Hum 1519
2021-03-25
Distance 1.5M
Temperature: 27°C
Hum 1515
Home
Chart
2021-03-17
Distance 2.5M
Temperature: 37°C
Hum 8888
2021-04-09
Distance 4.7M
Temperature: 24°C
Hum 13
2021-04-01


AVG Distance mainted last 4 weeks
distance
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
Week 1
Week 2
Week 3
Week 4

Problems Occurred During the Implementation Phase:

In the course of this project, we have faced my unpredicted and predicted problems and we implemented solutions to solve them while maintaining the same quality of performance and implementation. The problems can be categorized into two major categories, which are software and hardware**.**

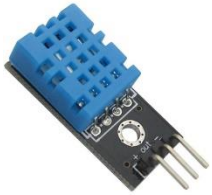
- March 1st, problem connecting to WIFI using esp8266 and Arduino uno. To solve this problem, I used a different WIFI module that is called NodeMCU-32S[ESP32-S] WiFi instead of esp8266 and Arduino uno.
- March 1st, deploying the website online has created some problems because there were some libraries which were excluded from the php package that were essential and critical for our project such as PostgreSQL. In addition to that, the php composer were not able to find the php package so after a long research, it was solved by adding the path to the environment variables of the operating system.
- March 5th, the code from cannot be uploaded from the Arduino IDE to the ESP32-S. The problem solved by Pressing the EN and 100 buttons on the ESP32-S to reset it while uploading the code.
- March 7th, the first issue was establishing a connection to a PostgreSQL database using PDO. The problem arose because I used libraries such as pg_connect() to connect but they were not compatible with PDO. The errors logs were not helpful because it showed internal server errors. After researching, I concluded that using PDO would solve the problem because unlike what I have done in the past, (which is connecting to a local database) connecting to a public database which is provided by Heroku requires the usage of PDO.
- March 8th, problem with connecting ultrasonic sensor with the ESP32-S. The reason why this problem happened is that the module I have doesn't have the pins positions printed on the microcontroller. But in the end, I found the correct pinout and I managed to connect the ultrasonic sensor with the ESP32-S.
- March 15th Problem with connecting to the web server because the server is secured Https. The problem addressed by using the library WiFiClientSecure instead of using the library WifiClient.

- April 2nd, Problem was implementing the favorite feature. Basically, any resource (a single row) would be considered favorite if there was a full yellow star next to it. Likewise, it would be considered unfavorite if there was a white blank star next to it. The JavaScript algorithm was made using toggle (add class (x) to an element if the class doesn't exist in that element => if that class exist => delete the class from the element) but problems started to arise when connecting the backend (database) logic with the frontend. The main problem was that by default, all resource were unfavorite, both in database and frontend. So, when a resource was changed from unfavorite to favorite, the database would change accordingly and so would the frontend. However, when reloading the page, all elements would be displayed as unfavorite in the frontend (while blank star next to them) while displaying the correct output in the database. To address this problem, the class value would have to be the same value as the database. Then when loading the page, JavaScript would check if the element is favorite and if so, it would automatically show the full star image next to it.
- April 6th, the problem was inserting the received data from the robot into a database. For some reason, the data was null until put in an html page. At the beginning, I tried to use a php script to fetch the data and insert it into the database but the fields were blank. To address this issue, the data was put into an html page using the file_put_content() and then retrieved from the html page using JavaScript then sent to a php script which would eventually insert the data into the database.
- April 6th, failed to send multiple Post requests to the server. The problem addressed by Sending all the data (Distance, Temperature, and Humidity) in one Post request.

The Hardware Components:

- HC-SR04 Ultrasonic sensor to measure the distance.

- DHT 11 Temperature and humidity sensor.

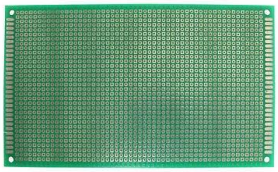


- NodeMCU-32S[ESP32-S] WiFi

- Piezo Active Buzzer

- Wires

- Veroboard



- DC-DC XL4015 Step Down Power Module

- Lithium-Ion Battery (2)

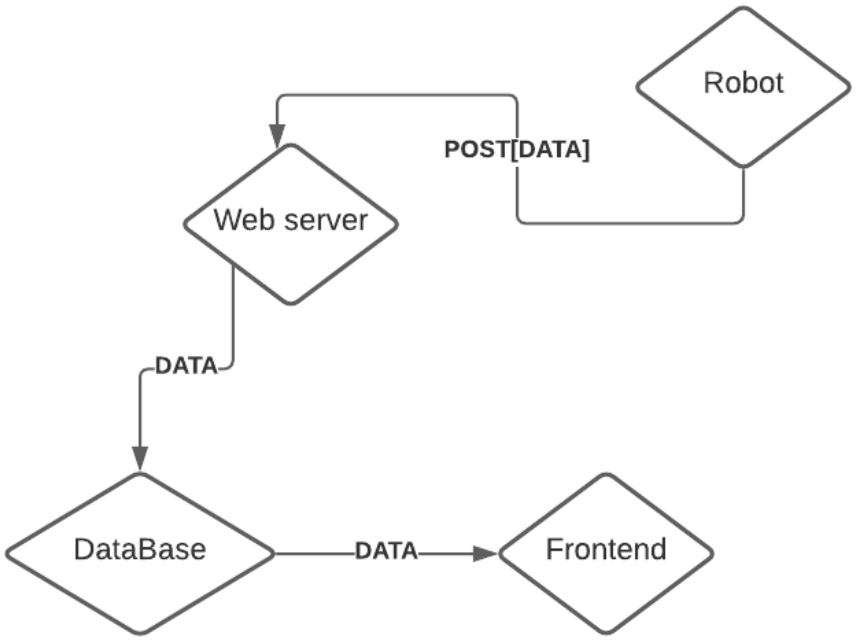
The Software Components:

- Arduino IDE
- PHP
- Composer
- PGSQL
- Apache
- Git

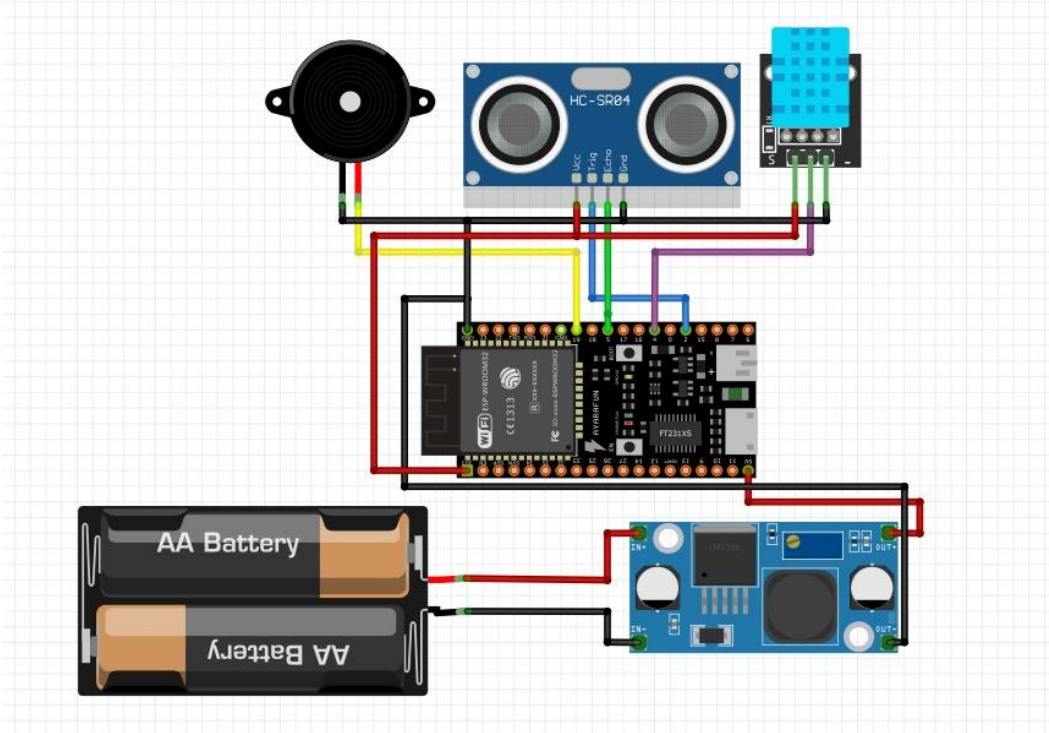
Third Party Libraries:

- WiFi
- HTTPClient
- WiFiClientSecure
- DHT
- JQUERY
- Bootstrap
- Chart.js
- AJAX

Architecture Diagram:



The Wiring Diagram:



The Enhancement Phase Plan:

- Create a box or container from plastic.
- Place the created circuit inside the container or the box to create a social distancing device that can be carried by hand.
- Add some LED lights, for example green LED light if the distance for the social distancing is maintained, and red LED light when the distance become less than the distance required for social distancing. This could help the deaf people who are using the device because they can't hear the buzzer sound.
- Allow the user to select different time periods for the chart. For example, last year, last month, last 4 months etc...
- Allow the user to filter the data by date, id, favorite, etc...
- Increase the quality of the code and software architecture.
- Add htaccess file.