IoT Dashboard

Mubeen Khan and Damiano Visalli

Final Project Presentation

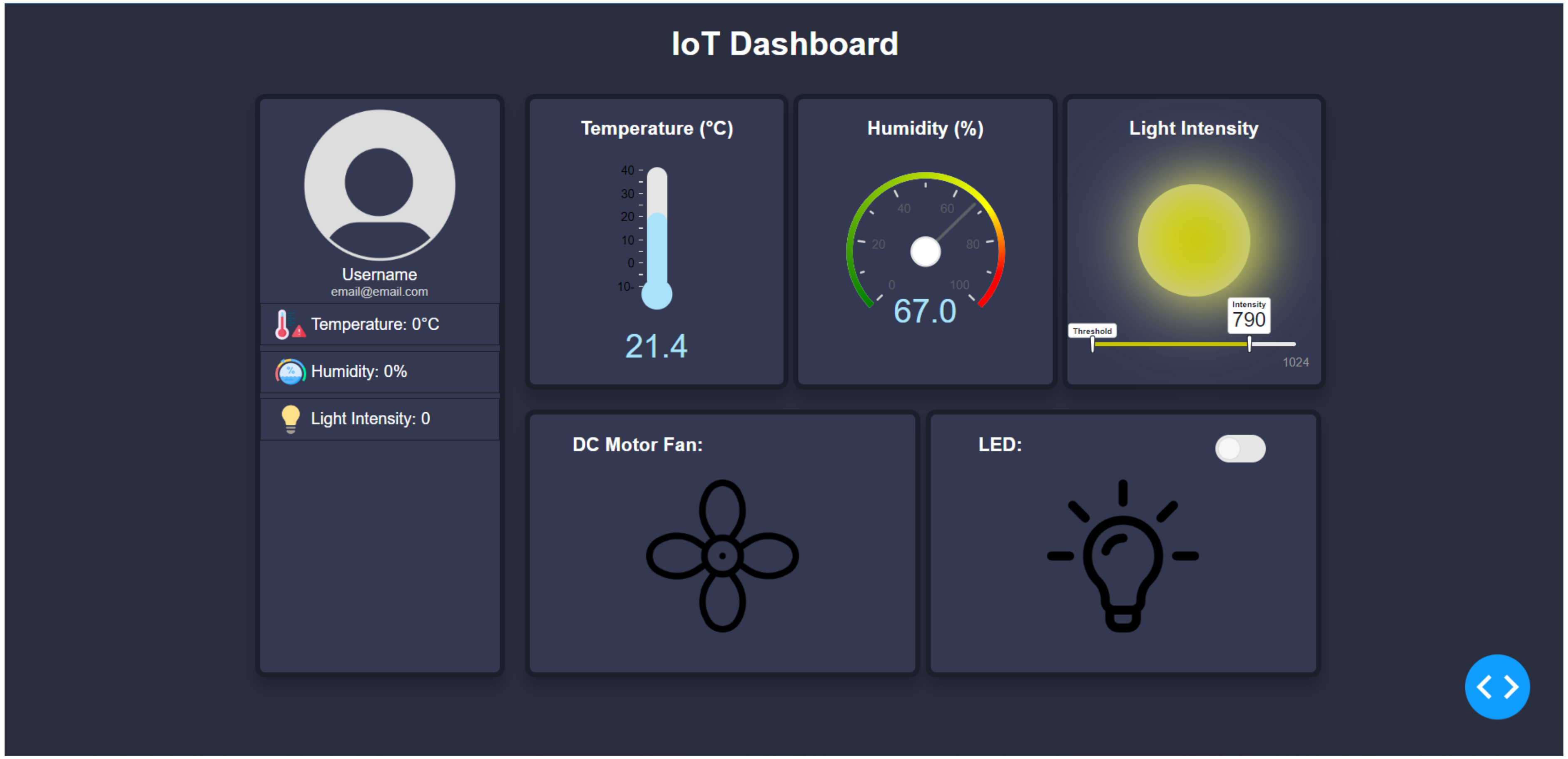
Agenda

- Project introduction
- Dashboard and Project Wiring
- Dashboard Design
- Members Roles for each phase
- Activities for each Phase
- Challenges and achievements

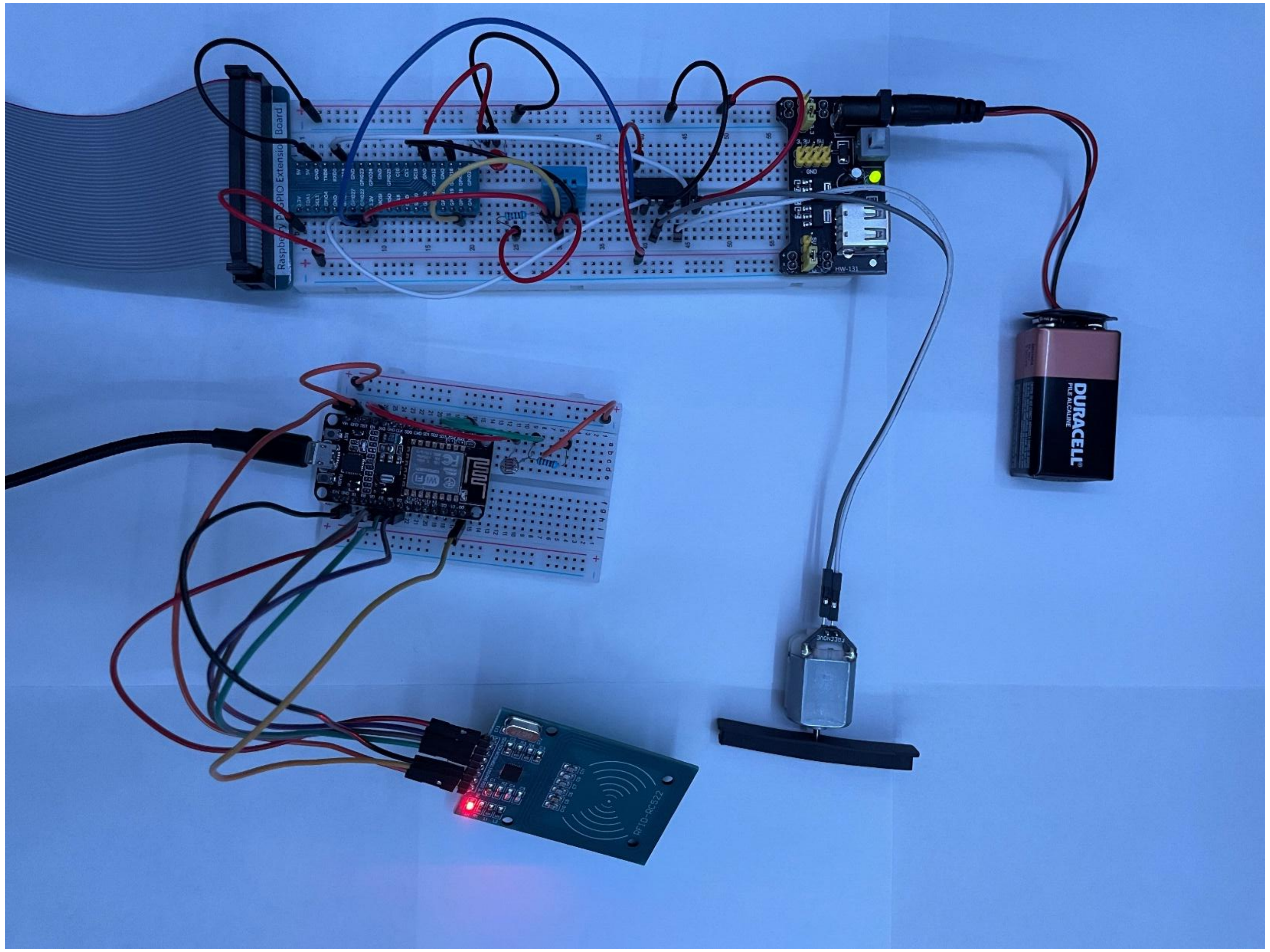
Project introduction

- The goal of this project is to build and design a web-based IoT dashboard.
- The system is built using different technologies and devices we have learned in this course and implement them in one single system.
- We utilized what we learned in this course to build an application that will capture data and make decisions based on the data.

Dashboard

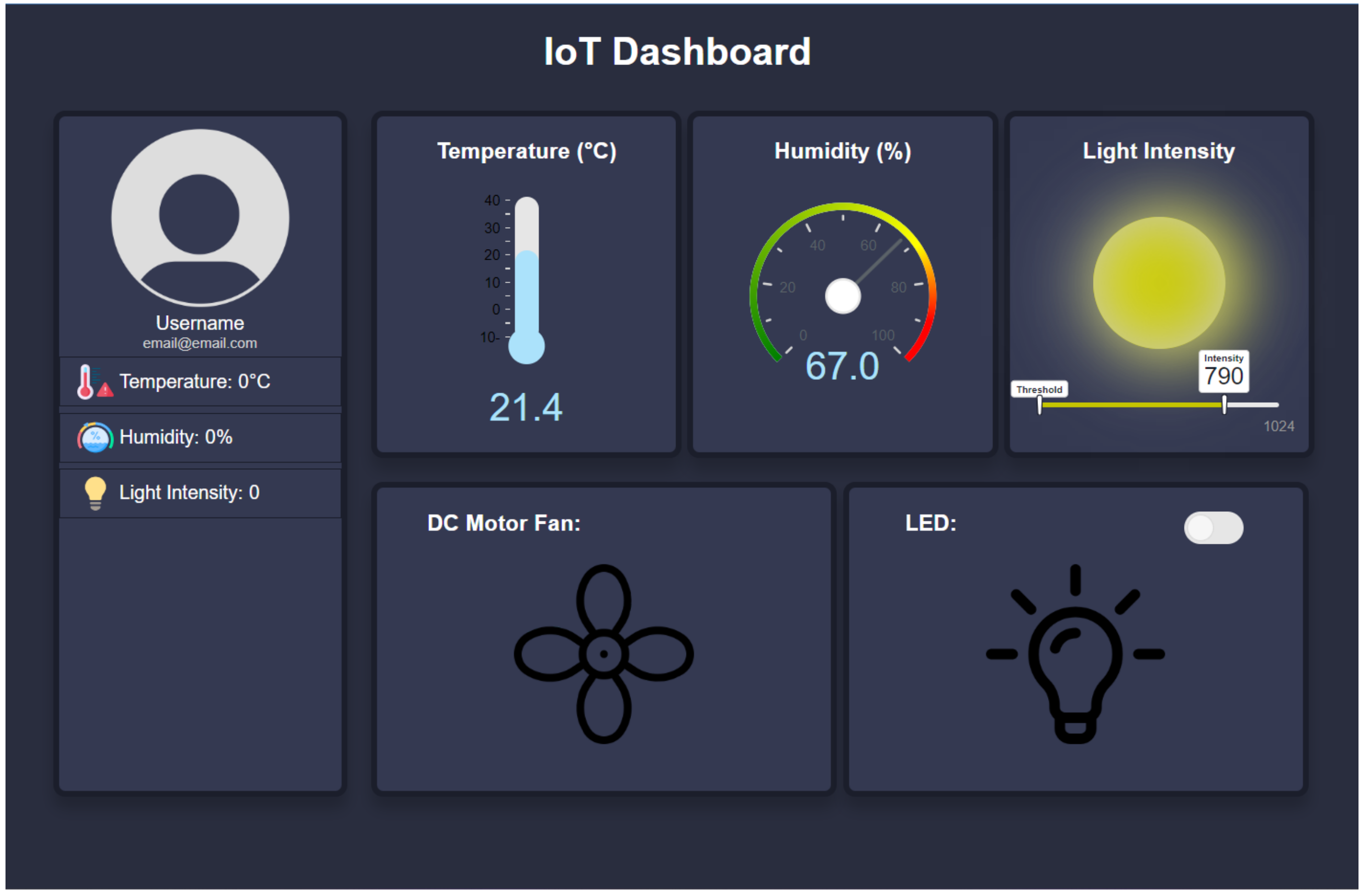

IoT Dashboard
Username
email@email.com
Temperature: 0°C
Humidity: 0%
Light Intensity: 0
Temperature (°C)
40
30
20
10
0
10-
21.4
Humidity (%)
0
20
40
60
80
100
67.0
Light Intensity
Intensity
790
Threshold
1024
DC Motor Fan:
LED:

Project Wiring



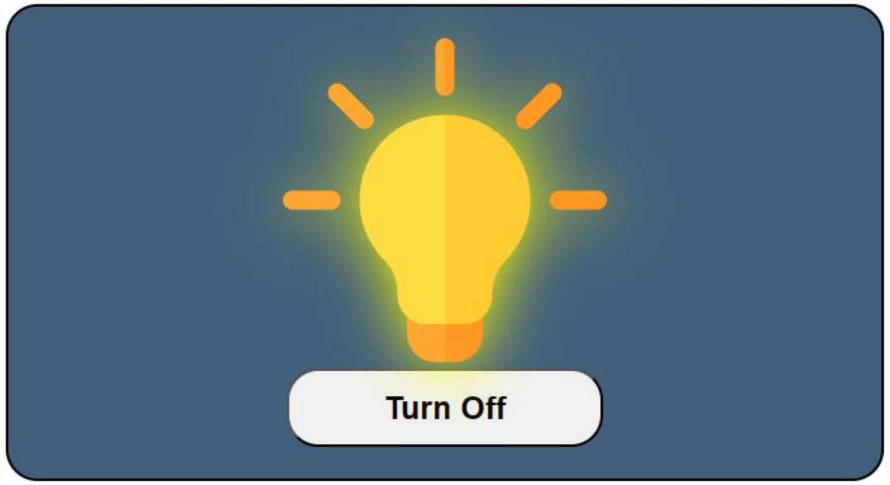
Dashboard Design

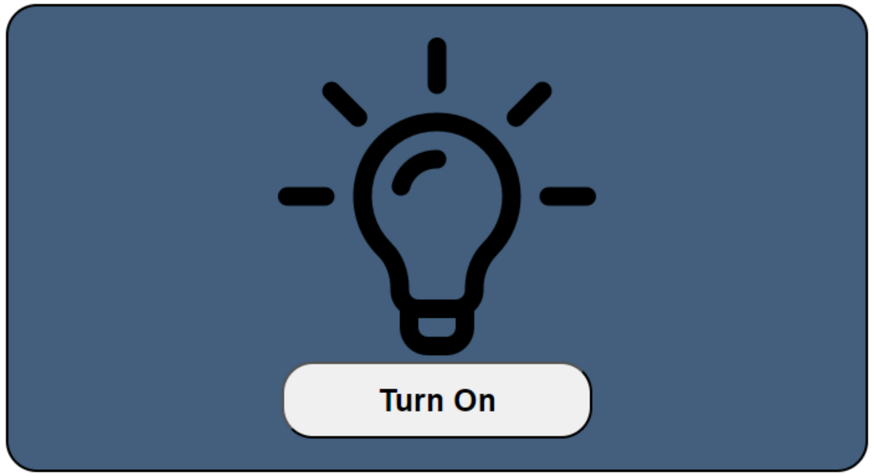
- Color Pallet:
  - Inspired by the themes in VScode
  - Palenight Theme
- Layout:
  - The top side displays all the data captured from the sensors
  - The bottom side represent the hardware's that can be manipulated using the sensors data (DC motor and LED)
  - Left side has the user information and thresholds.



Member roles: Phase 1

- Mubeen:
  - Did all of phase 1, because it was a short and simple task.
  - Main objective was to understand how Dash works and how each feature will call its method using a callback.





Member roles: Phase 2

- Mubeen:
  - Worked on sending and reading emails
  - Reading DHT11 temperature and humidity data
  - Displaying DHT11 data on dashboard
  - CSS for dashboard
  - Worked on Motor wiring
  - Worked on turning motor on and off

Member roles: Phase 3

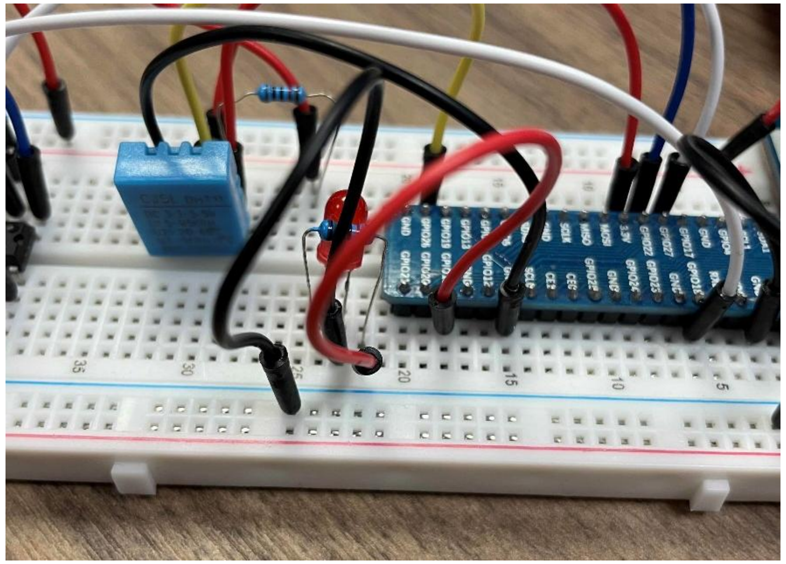
- Mubeen:
  - Retrieve light intensity from the Photoresistor using ESP8266
  - Publish data to the broker on Arduino
  - Subscribe to broker using the topic on python
  - CSS for dashboard
  - Send data retrieved from the ESP8266 to the app
  - Display and update data on the dashboard

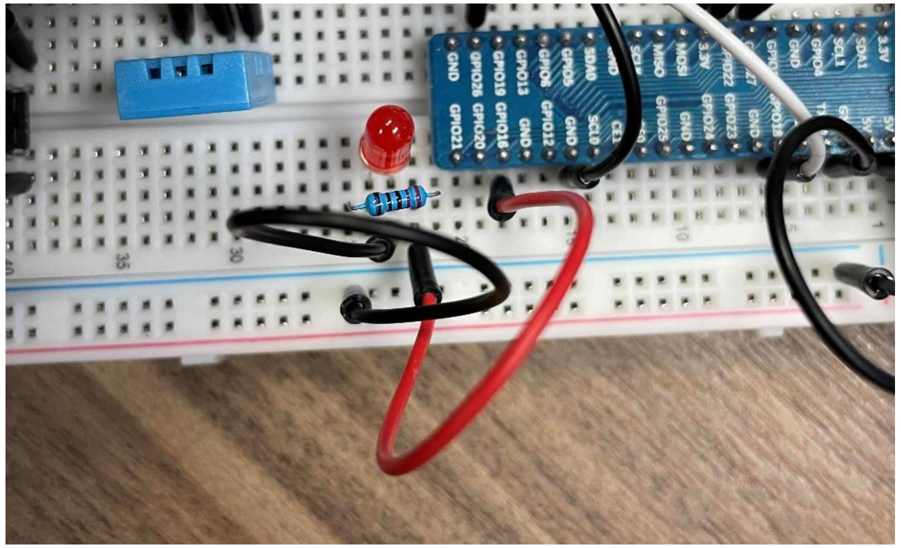
Member roles: Phase 4

- Mubeen:
  - Read data from RFID tag and publish it to the broker on Arduino
  - Subscribe to broker using the topic on python
    - While also subscribing to the Photoresistor topic from phase 3
  - Created the SQLite database
  - Sending email to the user when they Tap the RFID tag on the RDIF reader
- Damiano:
  - Used RFID tag ID to retrieve user information from database
  - Display the user's data on the dashboard

Activities for Phase 1

- In Phase 1, we implemented a switch/button on the dashboard that turn an LED on and off on both the dashboard and the breadboard.





Activities for Phase 1
(Components and Libraries)

Components used:

- LED
- Resistor (220Ω)
- Wires
- Breadboard
- Raspberry-Pi

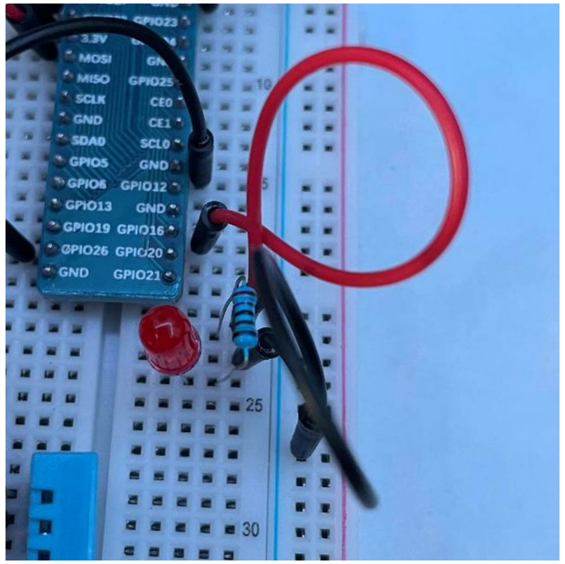
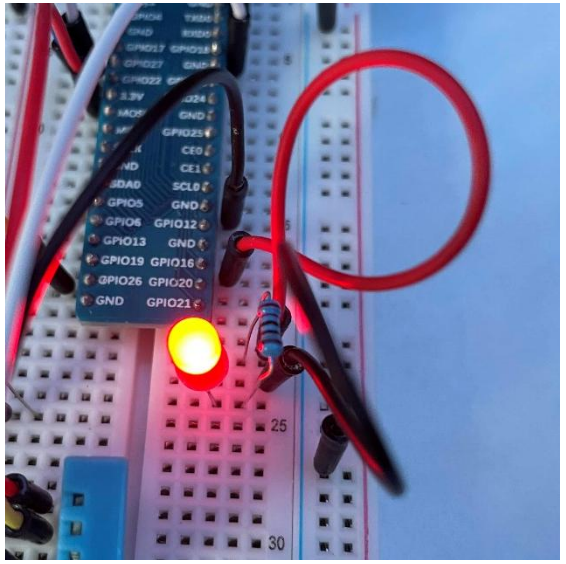
Libraries imported:

- Dash
- RPI.GPIO

Activities for Phase 1
(Breadboard)

- When the switch/button on the dashboard is turned on, the LED on the breadboard will turn on.

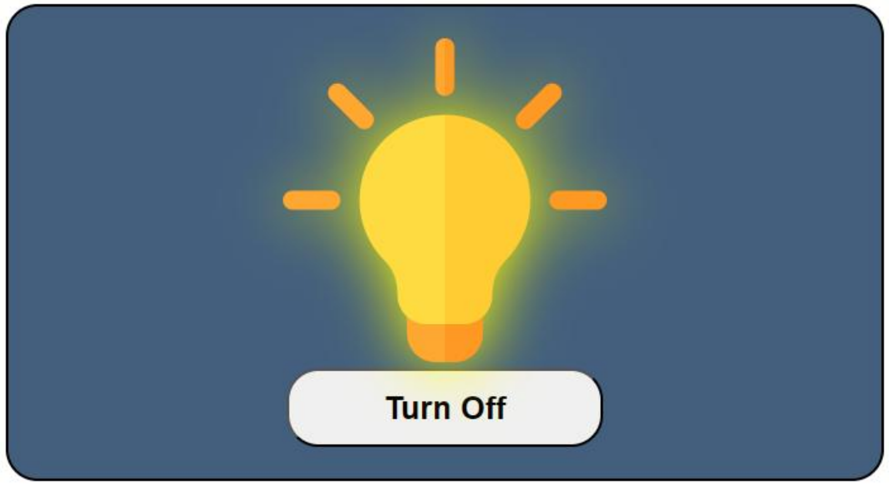
- When the switch/button is off, the LED on the breadboard will turn off.

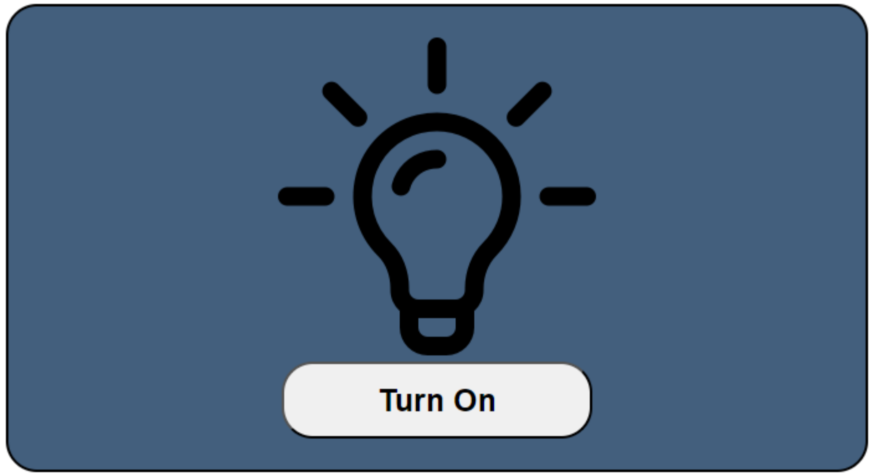



Activities for Phase 1 (Dashboard)

- When the switch/button is on, the image on the dashboard will change to an image that represents an ON LED.

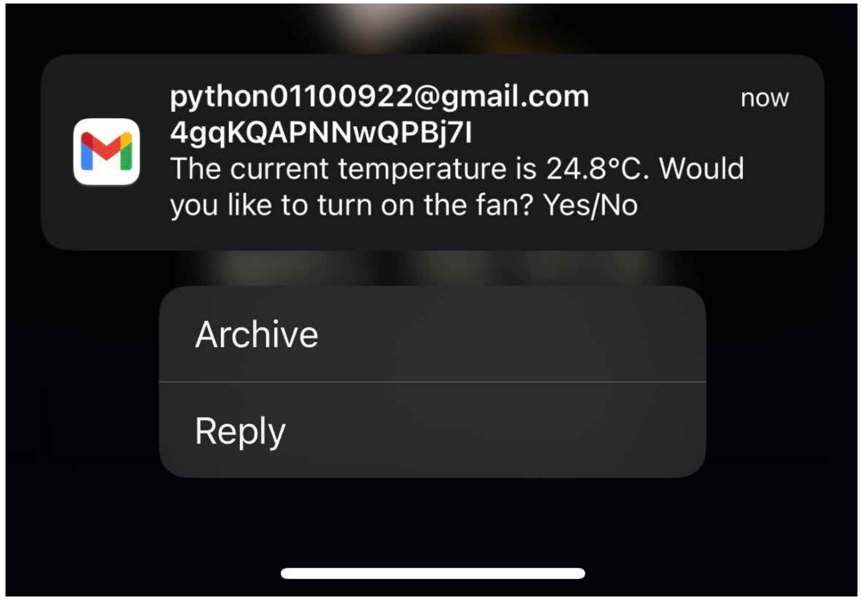
- When the switch/button is off, the image on the dashboard will change to an image that represents an OFF LED.



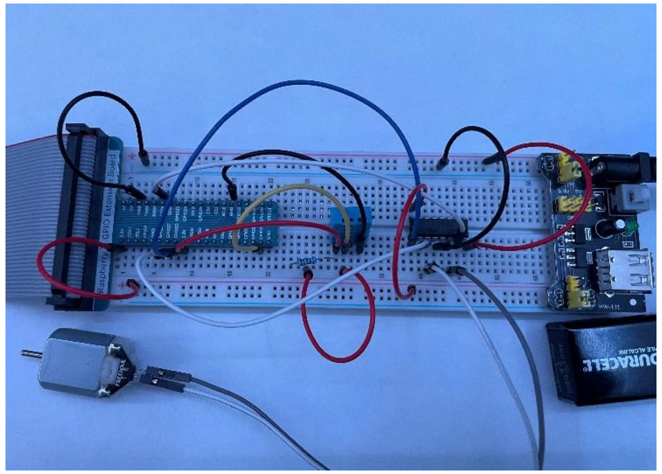
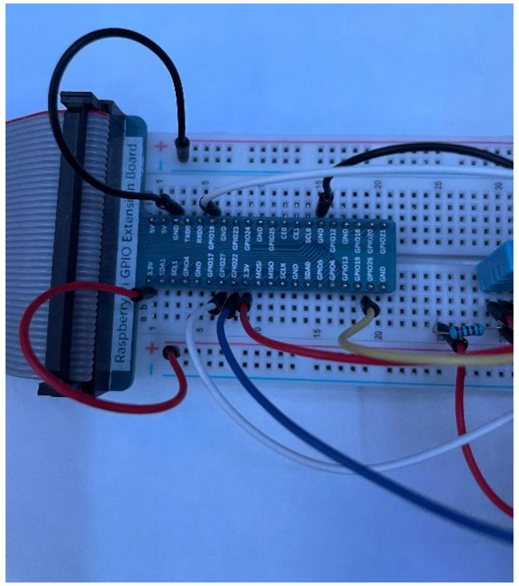
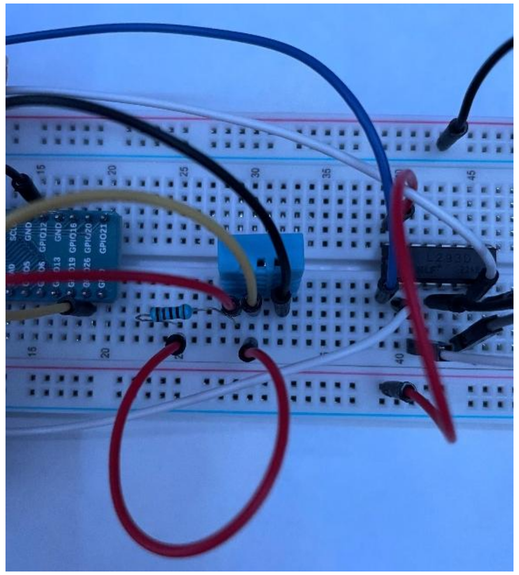
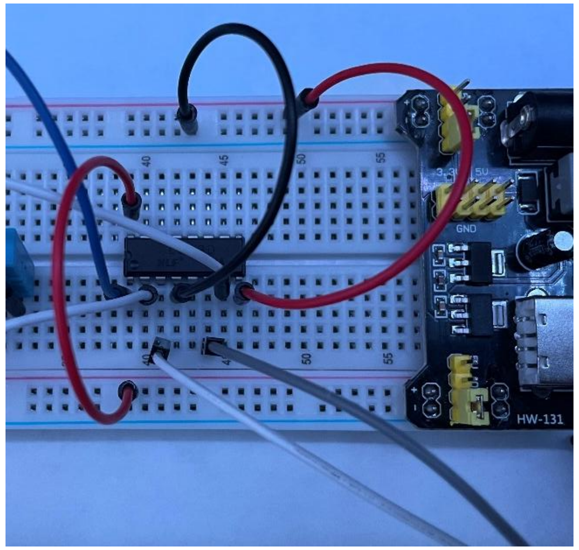


Activities for Phase 2

- In Phase 2, we captured the Temperature and Humidity data from the DHT11 sensor and displayed it on the dashboard.
- When the temperature surpasses the threshold, the system will send an email to the user asking whether they want to turn the fan/motor ON.
- If the user replies with anything other than "yes" or they don't reply at all, the fan/motor will remain OFF


python01100922@gmail.com
4gqKQAPNNwQPBj7I
The current temperature is 24.8°C. Would you like to turn on the fan? Yes/No

Activities for Phase 2
(Wiring)

Activities for Phase 2
(Components and Libraries)

Components introduced:

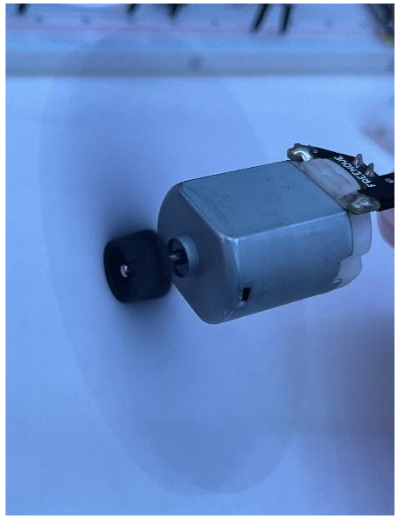
- DHT11 Temperature and Humidity sensor
- Resistors (10k Ω)
- DC Motor
- L293D motor driver
- HW-131 for power supply (9v battery)
- Wires

Libraries imported:

- dash_daq
- Freenove_DHT
- smtplib (send emails)
- imaplib (access emails)
- email (manage email messages)
- secrets
- string
- time

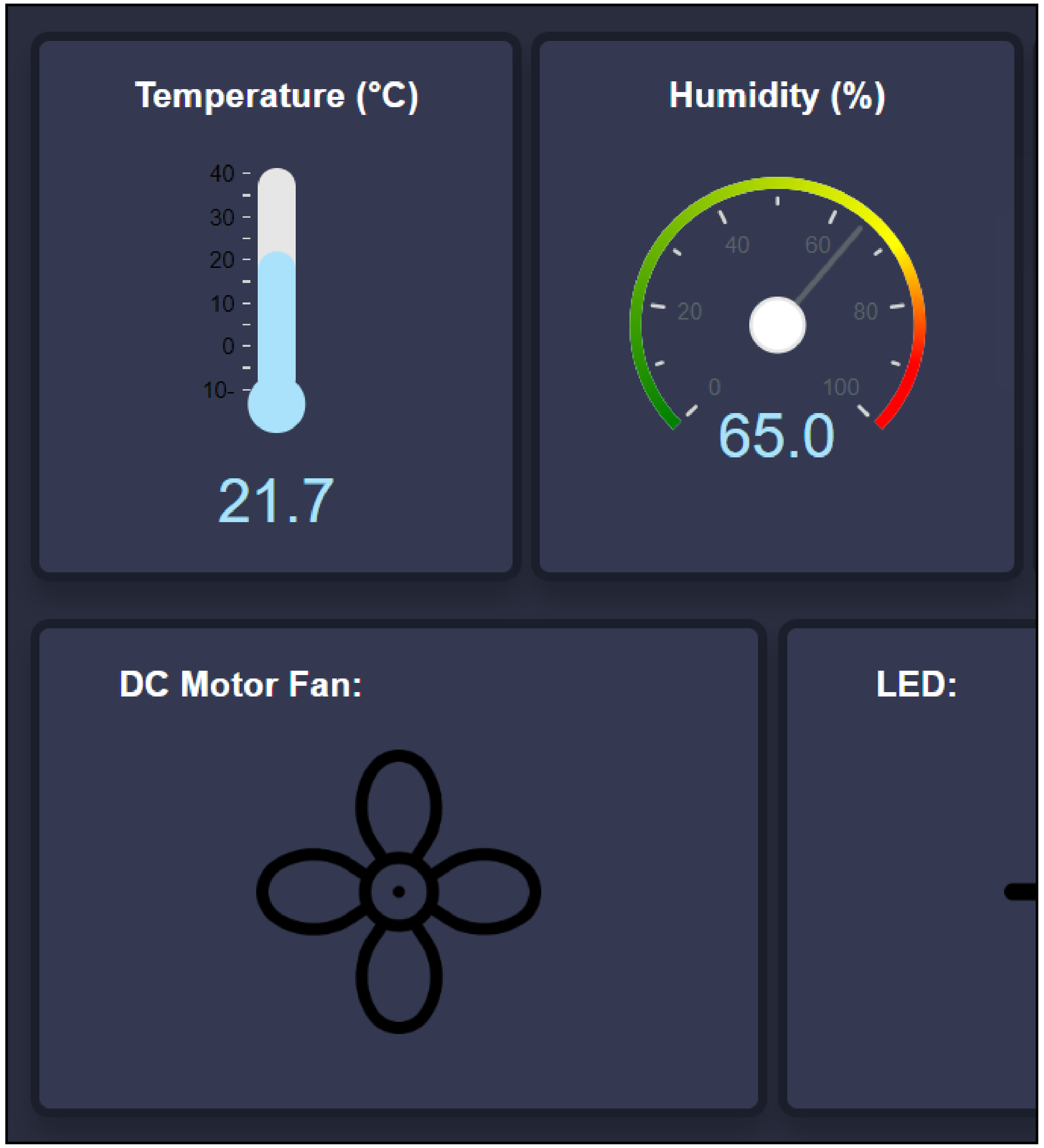
Activities for Phase 2
(Breadboard)

- When the temperature surpasses the threshold, the user receives an email. If they reply yes, the motor will turn on making the fan spin.
- The fan will only turn off once the temperature goes below the threshold.
- Anything other than "yes" will leave the fan off



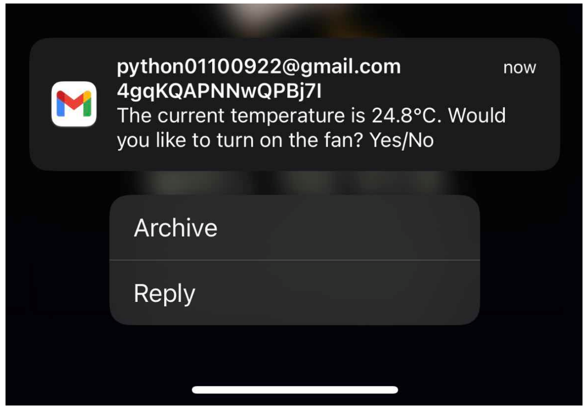
Activities for Phase 2
(Dashboard)

- We are using a Dash component called "Interval" which updates the app in real-time without having to reload the page or click any button.
- We used this component to read the temperature and humidity from the DHT11 sensor and update the values on the dashboard.

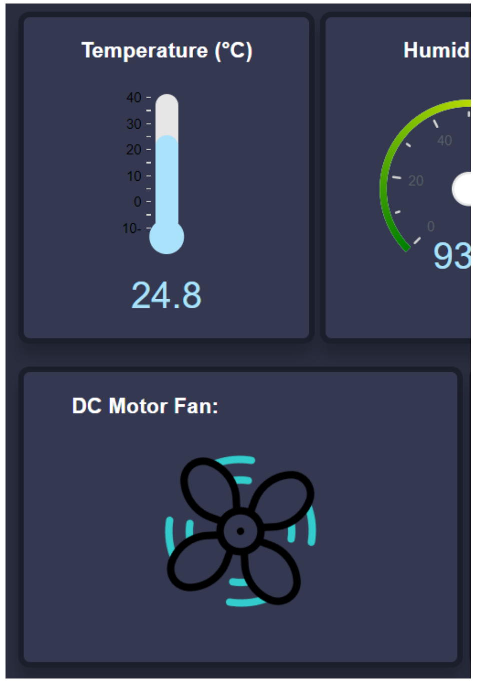


Activities for Phase 2 (Const.) (Dashboard)

- When the user allows the system to open the fan, the fan image on the dashboard will start to spin

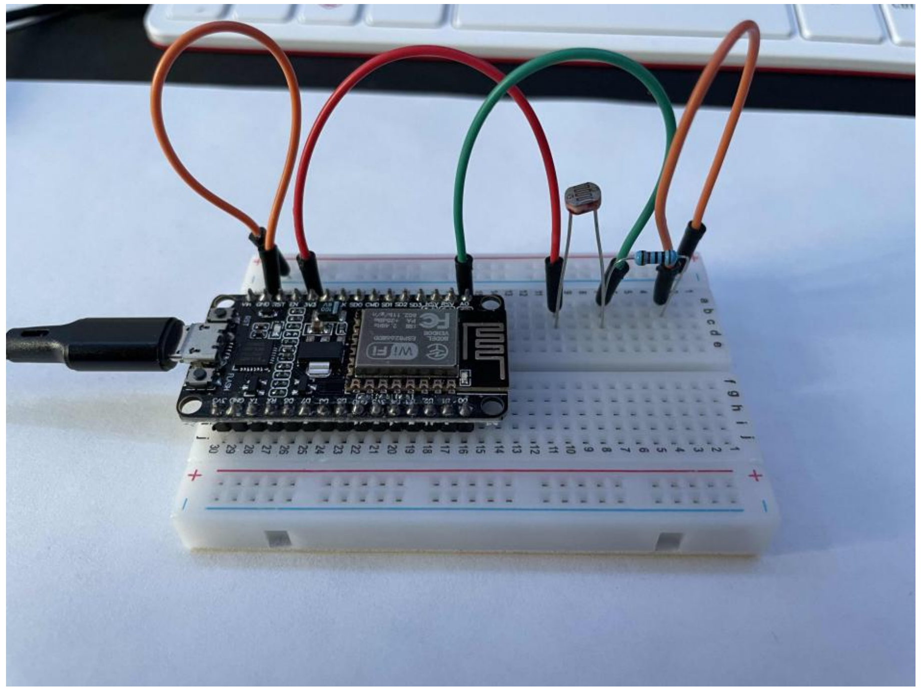


python01100922@gmail.com now
4gqKQAPNNwQPBj7l
The current temperature is 24.8°C. Would you like to turn on the fan? Yes/No



Activities for Phase 3

- In Phase 3, we used the ESP8266 and the Photoresistor to capture the light intensity in a room and publish that data to the Broker.
- That data is then be used to manipulate an LED depending on the threshold.
    - We are using the same LED setup from phase 1, but removed the switch that turns it ON and OFF.
- Data exchange was done using MQTT publish-subscribe communication



Activities for Phase 3
(Components and Libraries)

Components introduced :

- LED
- Photoresistor
- ESP8266
- Resistor (10kΩ and 220Ω)
- Wires

Libraries imported:

Python:

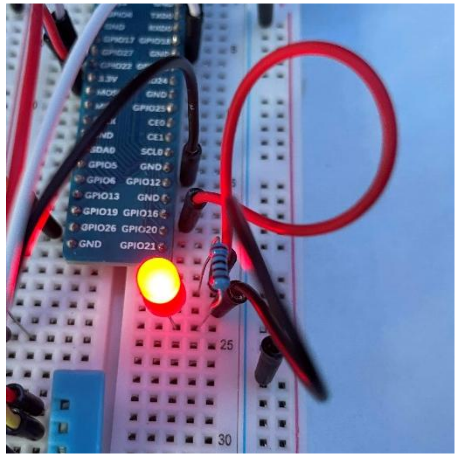
- paho.mqtt.client

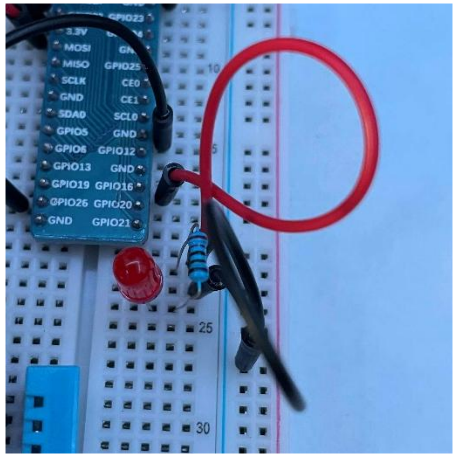
Arduino:

- ESP8266WiFi.h
- PubSubClient.h
- Arduino.h

Activities for Phase 3
(Breadboard)

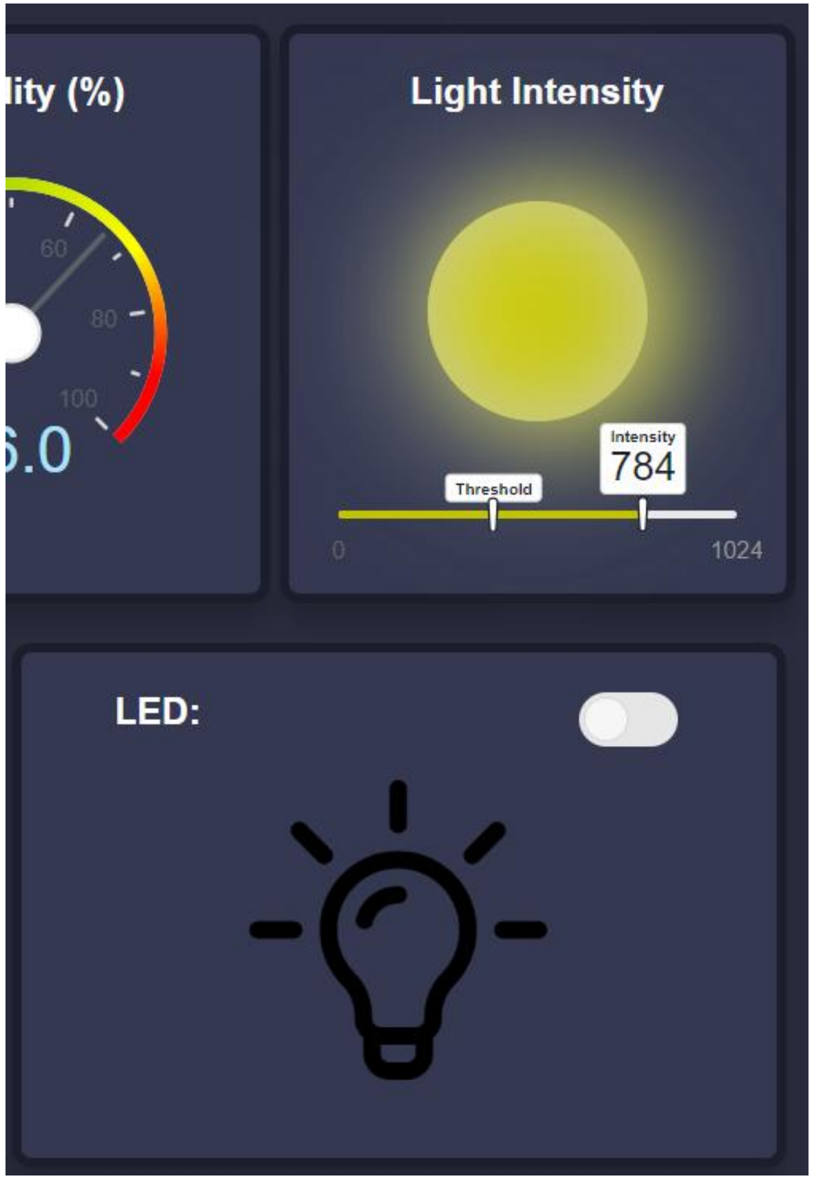
- The Photoresistor is connected to the ESP8266 and measures the light intensity in the room. When the light intensity goes under the threshold, the LED will turn on.
- The LED will turn off when the light intensity is above the threshold.





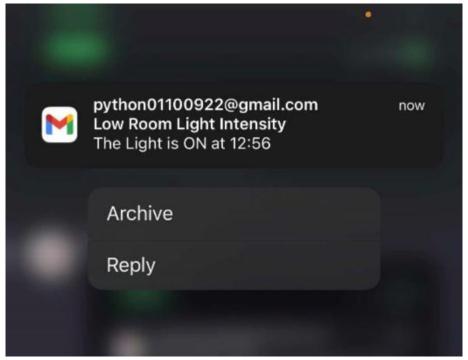
Activities for Phase 3
(Dashboard)

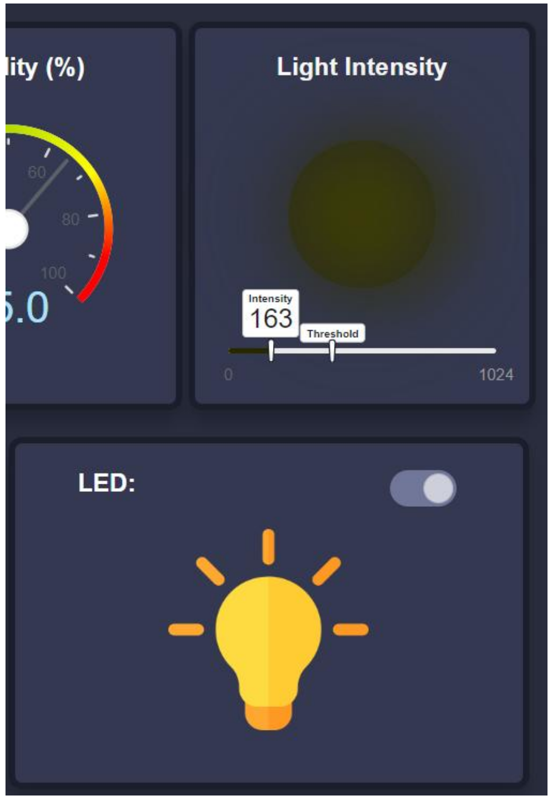
- The light intensity value and the threshold are both displayed on the slider.
- While the intensity value remains above the threshold, the LED will stay off.
- Intensity value is being updated using the “interval” component



Activities for Phase 3 (Const.) (Dashboard)

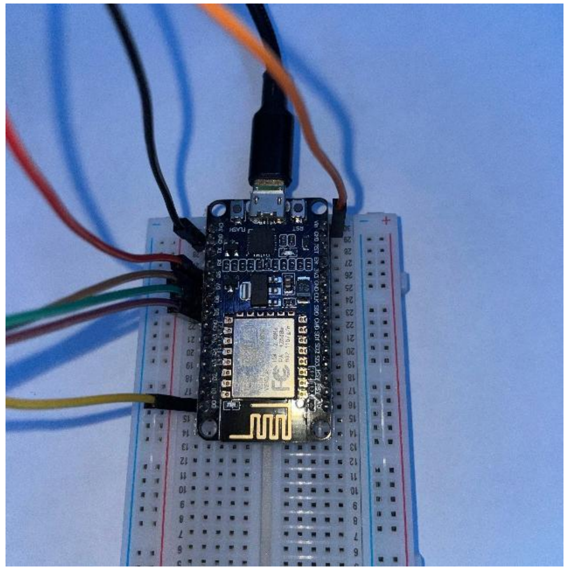
- When the light intensity goes under the threshold, the icon representing the intensity turns darker and the image representing the LED will change.

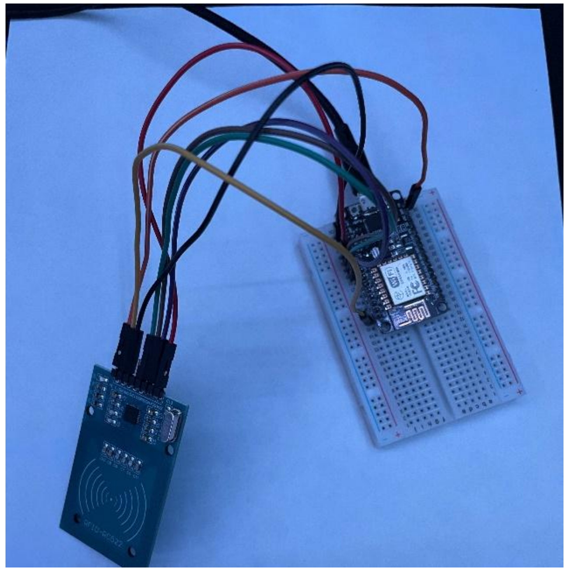

python01100922@gmail.com
now
Low Room Light Intensity
The Light is ON at 12:56

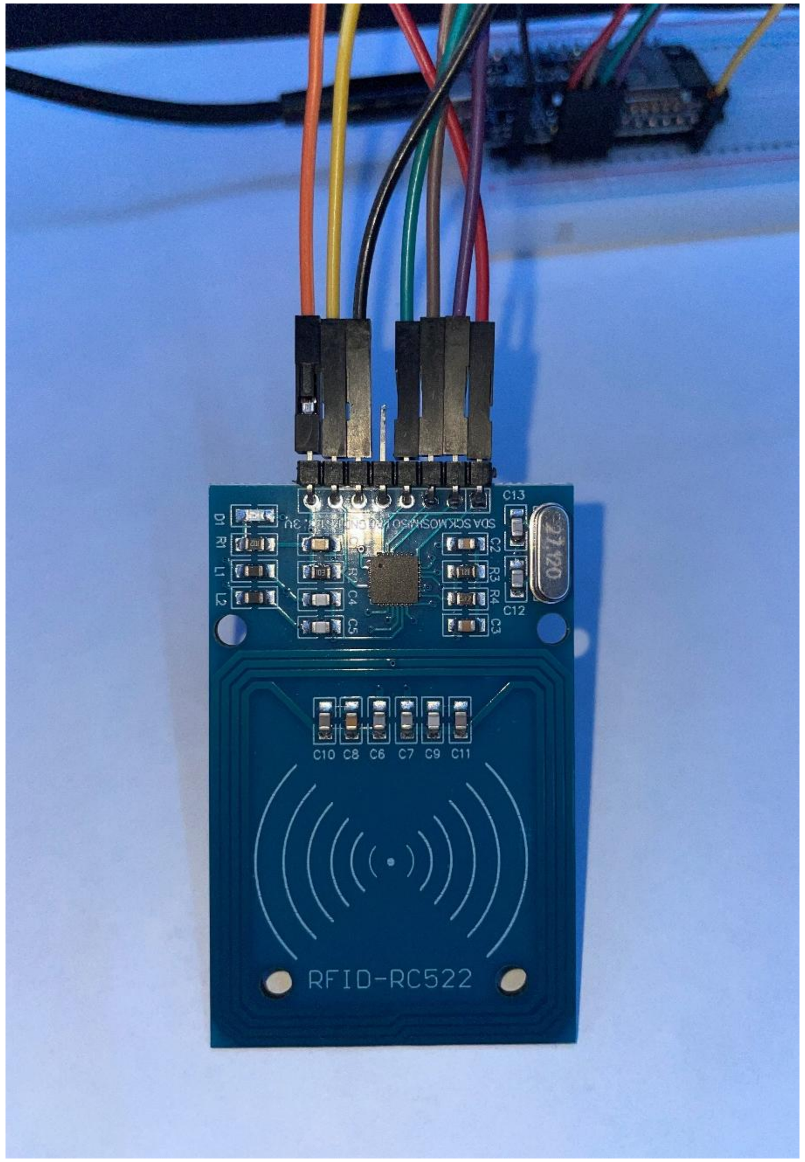


Activities for Phase 4

- In Phase 4, we used the RFID reader to capture the tag’s ID and update the user's information on the dashboard.
- The User’s information will be stored in a database (SQLite)
- Data exchange between the application and the ESP8266 is done through the MQTT publish-subscribe communication protocol







Activities for Phase 4
(Components and Libraries)

Components introduced :

- RC522 RFID Module
- ESP8266
- RFID Tags
- Wires

Libraries imported:

Python:

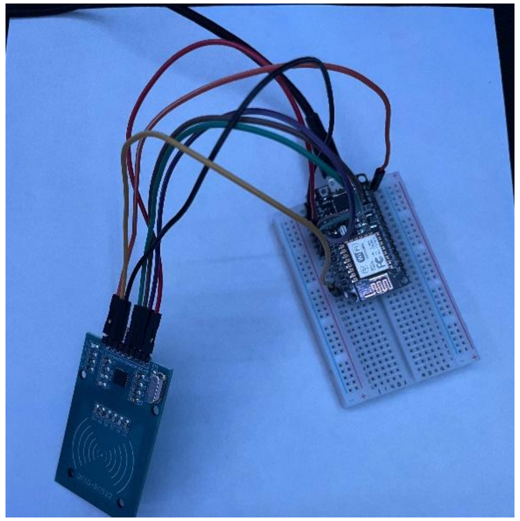
- SQLite3

Arduino:

- SPI.h
- MFRC522.h

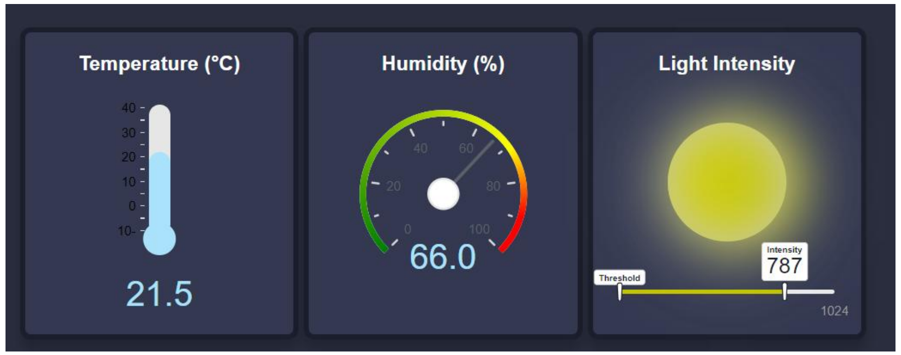
Activities for Phase 4
(Breadboard)

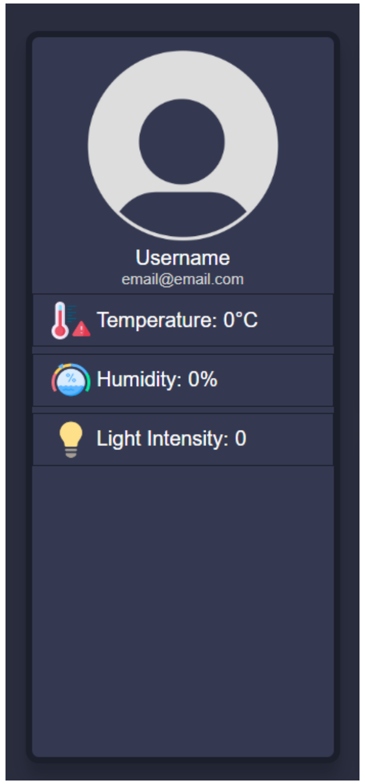
- The values of the RFID Tags are being read by an RFID reader that's connected to the ESP8266.
- The data is sent to the application through the ESP8266 using the MQTT protocol.



Activities for Phase 4
(Dashboard)

- When no RFID tag has been tapped on the RFID reader yet, the information on the application will be set to default.
- Furthermore, when a tag that does not exist in the database is tapped, the information on the application will be set back to default.





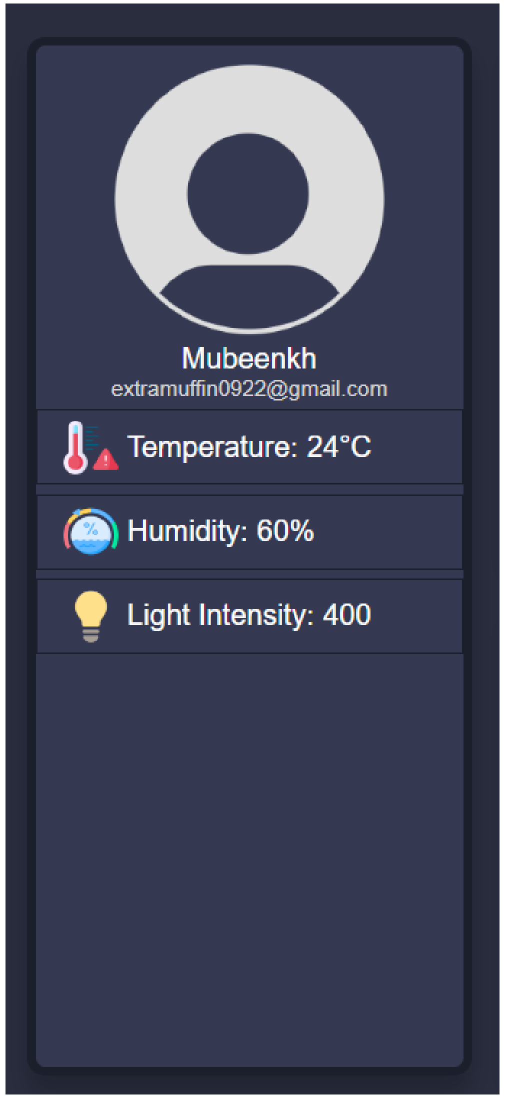
Temperature: 0°C

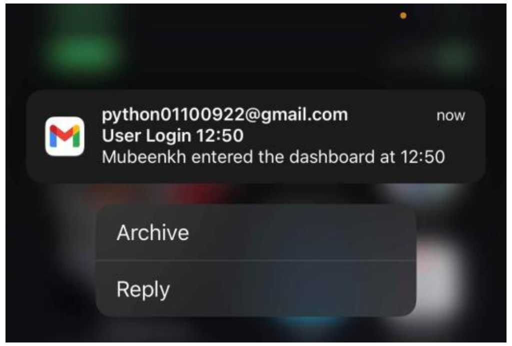
Humidity: 0%

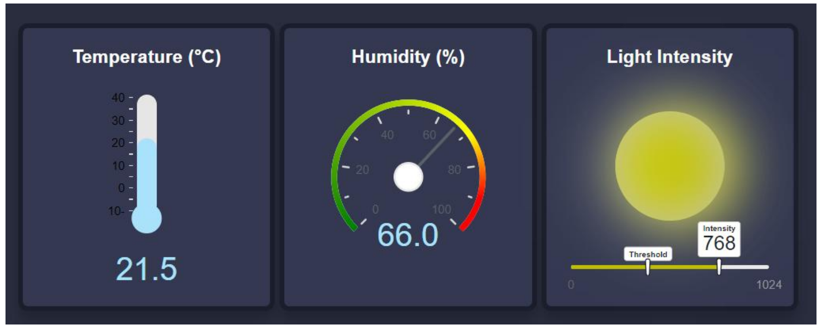
Light Intensity: 0

Activities for Phase 4 (Const.) (Dashboard)

- After tapping a tag that is in the database, the information of that user will be displayed in the dashboard.
- Furthermore, an email will be sent to the user letting them know what time they have accessed the dashboard.


Mubeenkh
extramuffin0922@gmail.com
Temperature: 24°C
Humidity: 60%
Light Intensity: 400


python01100922@gmail.com now
User Login 12:50
Mubeenkh entered the dashboard at 12:50



Challenges and achievements

(Conclusion)

- We had a lot of trouble trying to implement the Bluetooth feature, since the libraries we needed were not installing. Furthermore, the Bluetooth feature wasn't completed according to phase 4 requirement, so we did not include it in the dashboard.
- The RFID reader would not work most of the time. At first, it did not read any information from the tags. Then, it would read the data from the tags on the Arduino IDE, but nothing would be received or update on the application.

Thank You