



GMT

INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO
GALWAY-MAYO INSTITUTE OF TECHNOLOGY

22/04/2022

Environment Monitoring System

AONGHUS O DOMHNAILL

Bachelor of Engineering (Honours) in Software and Electronic Engineering

Supervisor

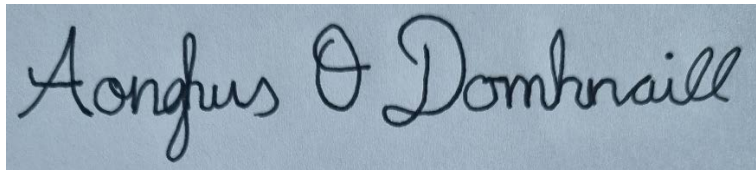
Brian O'Shea

Galway-Mayo Institute of Technology

Declaration

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Honours) in Software and Electronic Engineering at Galway-Mayo Institute of Technology.

This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

A handwritten signature in black ink on a light blue background. The signature reads "Aonghus O Domhnaill" in a cursive script. The 'A' is large and loops around the 'o' in 'Aonghus'. The 'O' is a simple circle. 'Domhnaill' is written in a fluid cursive style.

Acknowledgements

I would like to acknowledge Niall O’Keefe, Michelle Lynch and Paul Lennon for their help and feedback throughout the project.

Especially grateful to my project supervisor Brian O’Shea who has been extremely helpful.

Table of Contents

1	Summary.....	8
2	Poster.....	10
3	Introduction.....	11
4	MERN Stack.....	12
4.1	Mongo DB.....	12
4.2	React.js.....	12
4.3	Next.JS.....	12
4.4	Node.js.....	12
4.5	Express.js.....	12
5	Hardware.....	13
5.1	ESP32.....	13
5.2	DHT22.....	13
5.3	BH1750.....	13
5.4	CCS811.....	14
5.5	PH Sensor Module.....	14
5.6	TDS Module.....	15
5.7	SD Card.....	15
5.8	ESP32-CAM.....	15
6	Software.....	16
6.1	Arduino.....	16
6.2	FreeRTOS.....	16
6.3	Middleware.....	18
7	AWS.....	19

7.1 EC2 Server.....19

7.2 AWS IOT Core19

7.3 MQTT.....19

8 Grow20

8.1 CO2.....20

9 Project Architecture.....21

10 Results.....22

11 Project Plan23

12 Ethics.....24

13 Conclusion.....25

14 References26

1 Summary

The goal of the project was to effectively monitor the conditions in a growing environment and based on data received from sensors be able to adjust conditions through fans, heaters, or adding CO2. To create an environment most suitable to cultivate crops.

The scope of the project was mainly focused on deploying a full-stack web application using the MERN stack and using embedded C/C++ to write the code for ESP32.

On the frontend, React was used with Next.js framework that provides faster rendering, server-side rendering, built-in CSS, better image optimization, and API support.

Node.js was used as middleware in the backend. Express server-side framework ran inside here too for URL routing and handling HTTP requests and responses.

ESP32 publishes an MQTT message to a topic on AWS IOT Core. Node.js is subscribed to the topic and saves incoming data to the Mongo DB database. The web application receives data from the database using Nodejs and an API. Data is then displayed on a web page.

Throughout the project manage to accomplish most of the goals. The project can accurately take readings at certain intervals and after storing the data in the database with proper formatting, can display through a web application hosted on an AWS EC2 server.

Conclusions made over the course of the project have led to a greater understanding of full-stack development, embedded software and sensor logging applications. The integration of a web server, microcontroller, sensors and a database server can be easily adapted into many systems.

2 Poster

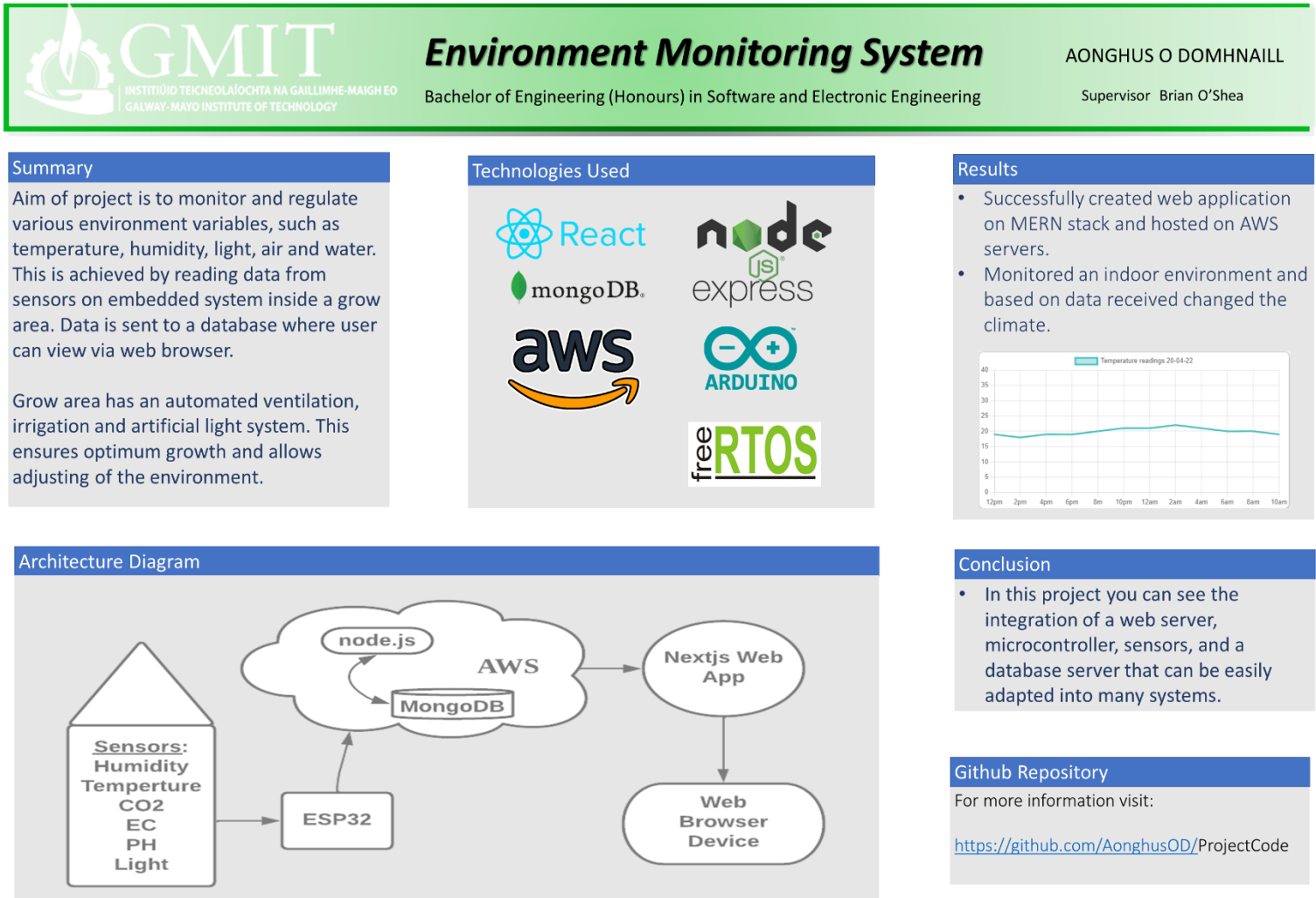


Fig 1: Poster

3 Introduction

Inspiration for this project came from an increase in growing food at home in recent times [1] , and an interest to have a mix of hardware and software to help further my understanding of both.

Imported produce must be harvested before it has ripened fully and later sprayed with ethylene gas to aid in the ripening process, resulting in less taste and less nutritious produce. There was a huge growth in growing food around lockdown and people are getting more conscious about sustainability.

However, our climate restricts certain crops and times of the year when it is possible to grow. There are also many issues such as unwanted climate change and diseases that have always plagued farmers. The best solution to these issues is in a controlled environment.

The goal of this project is to develop a hardware device with multiple sensors to monitor temperature, humidity, light, PH, EC and air in a growing area. From the data received devices such as fans, heaters, dehumidifiers and extractor fans can be programmed to counter undesired readings.

To accomplish these goals research into embedded C/C++ to write the code for sensors on ESP32, building a full-stack web application to display and manage data, and Amazon Web Services to host was required. In addition to, how to grow to produce vegetables in an indoor environment.

This report details technologies used in developing an application to monitor an environment and some methods used to change the environment.

4 MERN Stack

The MERN stack is a collection of technologies, all JavaScript-based that enables faster application development. [2]

4.1 Mongo DB

MongoDB is a NoSQL database that stores data in a collection. Each record compromises key-value pairs that are similar to JSON objects.

4.2 React.js

React.js was used to create a progressive web application. Complex interfaces were developed through simple components and connected to data on Nodejs. It was then rendered as HTML.

4.3 Next.JS

Next.js is a framework that provides faster rendering, server-side rendering, built-in CSS, better image optimization, and API support for React.js.

4.4 Node.js

Node.js is required to run code on the server side. It is the middleware that handles requests from web application and database.

4.5 Express.js

Express.js is the framework used to avoid creating many Node modules. The main uses in project were URL routing and handling HTTP requests and responses.

5 Hardware

5.1 ESP32



Fig 2: ESP32

Cores	2
RAM	512 kB
FLASH	16 MB
GPIO Pins	36
Power Consumption	160-260mA
Deep Sleep	.01mA

5.2 DHT22

The DHT22 is a humidity and temperature sensor. It uses a thermistor and humidity sensor to measure the air. It sends back digital signal on the data pin. It uses max current of 2.5mA and operating voltage Of 3-5V.

5.3 BH1750

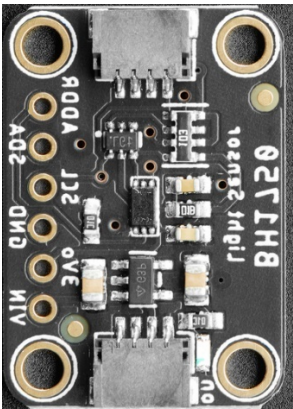


Fig 3: BH1750

The BH1750 is 16-bit Ambient Light sensor [8]. In a grow environment it is important to monitor the lux because based on if the lights are on or off you would have different thresholds because it would not be as hot in the tent. Also, with an adjustable light it can be very easy to leave the light turned down after working in grow area.

5.4 CCS811

The CCS811 is digital TVOC (air quality) and CO2 (carbon dioxide) sensor that operates over I2C bus [3]. A semaphore was required between CCS811 and BH1750 because they were both on I2C bus and were interfering with each other during run time.

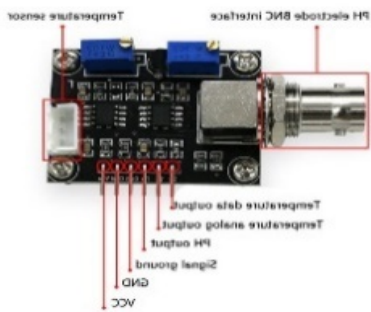


Fig 4: CCS811

Operating Voltage	3-5V
Max Current	30mA
Power Consumption	60mW
Operating Temperature	-5-50°C

5.5 PH Sensor Module

The pH sensor indicates the acidity or alkalinity of a solution. PH sensor is used to measure the pH of the reservoir. The sensor measures the electrochemical potential between a liquid inside a glass electrode and a liquid outside. There is a potentiometer to vary the value of the sensor. [4]



Operating Voltage	5V
Max Current	5-10mA
Power Consumption	0.5W

Fig 5: PH Module Board

5.6 TDS Module



Fig 6: TDS Module

TDS (Total Dissolved Solids) indicates how many milligrams of soluble solids dissolved in one liter of water [9]. This is helpful when growing crops because it lets you know how much nutrients is in the water. Ideally 900 – 1500 ppm is ideal for tomatoes depending on stage of growth.

5.7 SD Card

An SD Card module is used to store data. To reduce running time and power consumption readings can be scheduled at certain intervals and uploaded.

5.8 ESP32-CAM

An ESP32-CAM was scheduled to take 1 picture per day. Pictures can be viewed at github repository [5].



Fig 7: ESP32-CAM

Operating Voltage	5V
Power Consumption	180mA
Deep-Sleep Usage	6mA

6 Software

Code for ESP32 was written in embedded C/C++ using FreeRTOS to manage the tasks.

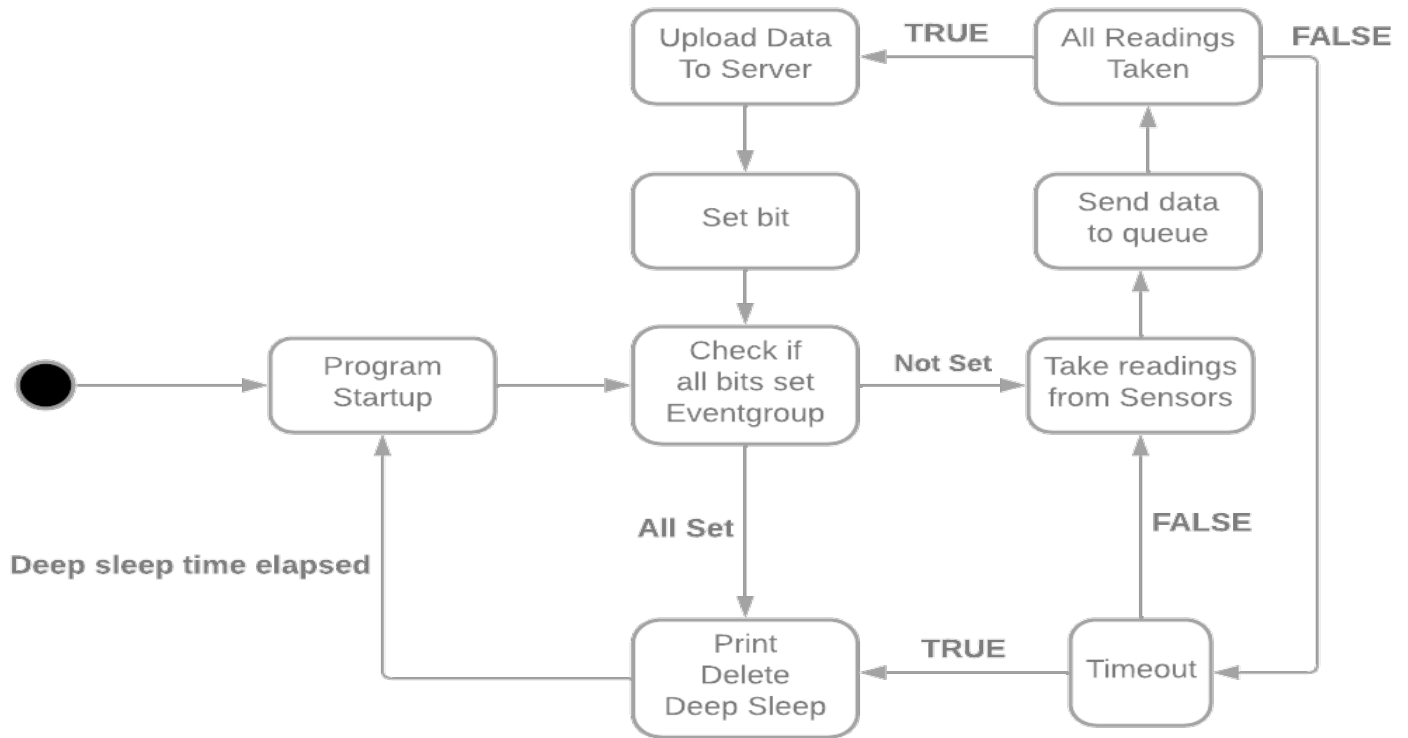


Fig 8: UML State Diagram

6.1 Arduino

The Arduino IDE was used to write & test code and upload it to the microcontroller. It had all the libraries for the sensors.

6.2 FreeRTOS

FreeRTOS was used to manage the tasks using semaphores, event-groups, and task priority. It also enabled the use of a second core in ESP32 which is not achievable with Arduino alone.

Each sensor has its own task. At run time the various sensors take readings and send the data as a struct through a queue.

```
typedef struct {
    uint8_t sensor;    //ID for switch statement
    uint16_t qData;    //16bit integer required for values above 255
    uint8_t qData2;
    float qData3;      //Float required for PH Values
} dataStruct;
```

Fig 9: Struct Declaration

Data is sent to the queue with the relevant ID.

```
ClimateData.sensor = CLIMATE_ID;    //ID for switch statement
ClimateData.qData = t;               //Data to be sent
ClimateData.qData2 = h;              //Data to be sent
xQueueSend(data_Queue, &ClimateData, 0); //Structure data queued by copy
```

Fig 10: Sending data to the Queue from the task

A switch statement manages which task sent the data, then data is saved to the SD card and sets a bit for the event-group.

```
switch (received_Data.sensor) {
    case CLIMATE_ID:
    {
        objArrayData["TEMP"] = received_Data.qData;
        objArrayData["HUM"] = received_Data.qData2;
        saveJsonToAFile(&doc, filename);
        xEventGroupSetBits(SwitchEventGroup, climateBit);
    }
}
```

Fig 11: Data is received and saved to file

When all the data is saved to an SD card it then publishes an MQTT message to the topic on AWS IOT Core and sets the final bit on the event-group.

```
char jsonBuffer[512];
serializeJson(data, jsonBuffer); // print to client
client.publish(AWS_IOT_PUBLISH_TOPIC, jsonBuffer);
```

Fig 12: Send a message to topic from file

After all, bits are set it prints the file, then deletes it and goes into a deep sleep for as many seconds that were predefined at startup.

```
if (saveSD_EventBits & ( airBit | luxBit | phBit | tdsBit | climateBit)) {
  printFile(filename);
  SD.remove(filename);
  Serial.flush();
  esp_deep_sleep_start();
}
```

Fig 13: Event Bits Are Set

6.3 Middleware

On the backend, NodeJS subscribes to the topic and inserts the data with Mongoose function insertMany into the Mongo DB database.

```
dataModel.insertMany(myobj3, function(err, res) {
  if (err) throw err;
  console.log("1 record inserted");
});
```

Fig 14: Mongoose Function insertMany

To fetch the data from the database Express.js uses a POST request through a REST API and sends that data to the web application to be processed.

```
router.post("/getdata", function (req, res, next) {
  dataModel.find().then(function (docs) {
    docs = JSON.stringify(docs);
    res.status(200).json(docs);
  });
});
```

Fig 15: Express Function Post

7 AWS

AWS (Amazon Web Services) is a collection of web services that help launch an app or website. They have a wide range of options and servers to suit everyone's needs while being very cost-effective.

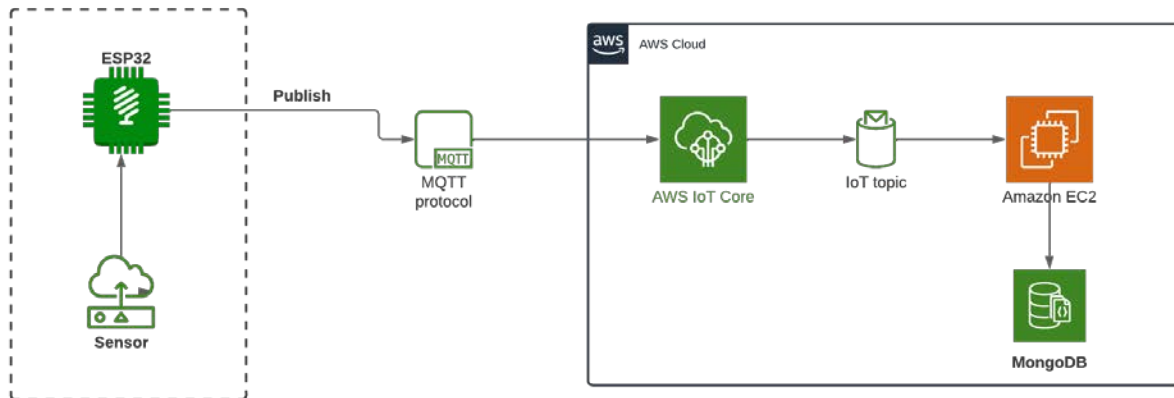


Fig 16: AWS Architecture Diagram

7.1 EC2 Server

Amazon EC2 (Elastic Compute Cloud) platform is used to host the Frontend, backend, and Mongo DB database. It is hosted on an EU West server.

7.2 AWS IOT Core

The AWS IoT Core message broker provides secure, bi-directional communication for Internet-connected devices and supports devices and clients to publish and subscribe to messages that use MQTT protocols.

7.3 MQTT

IoT (Internet of Things) describes the network of physical objects from small sensors, to devices that control large systems. There are few protocols to connect such devices to the internet for example TCP/IP, HTTP, and HTTPS. However, they are not optimized for low power and efficiency like MQTT (Message Queue Telemetry Transport).

8 Grow

An indoor grow environment was created to get real data to monitor. Artificial lighting was used, a 250W metal halide for early growth and a 1000W LED for mid to finish using only 120W electricity. An extractor fan and a circulating fan were used to control the air. The system is controlled by a series of timers.



Fig 17: Grow Area

8.1 CO₂

Yeast with sugar was added to water to create CO₂ which turned out to be very effective at raising CO₂ levels. Although it only lasts for a few days can be time-consuming and costly. A CO₂ dispenser would be a better alternative in a controlled environment.

9 Project Architecture

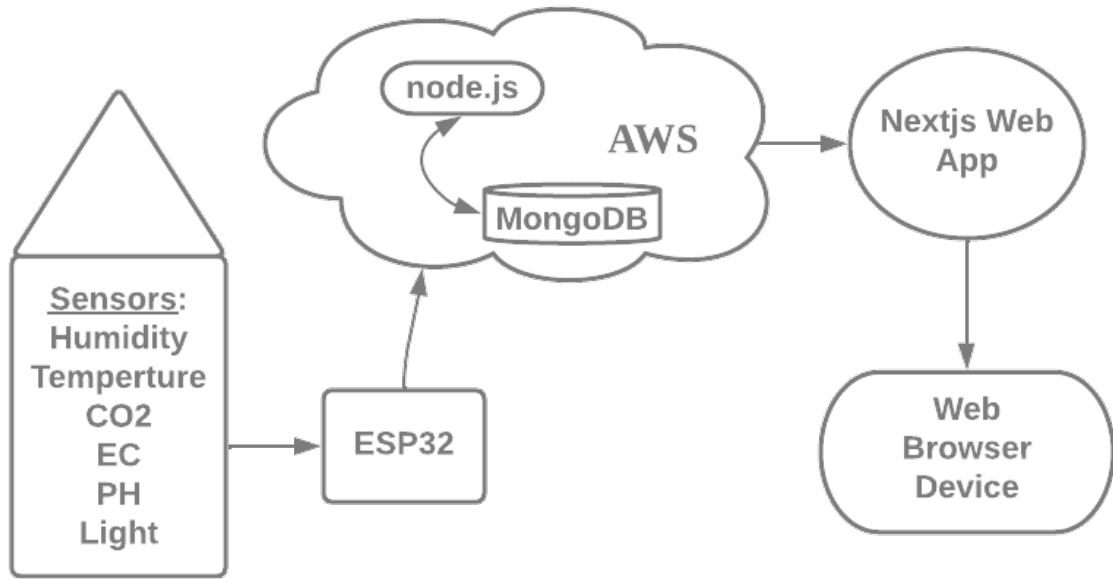


Figure 18: Architecture Diagram

10 Results

After taking readings from the growing area, it could be seen that there was a decline in temperature when the lights were off. It wasn't too significant but it was a mild night and it would have been worse during colder nights. Originally lights were off between 12 am and 6 am.

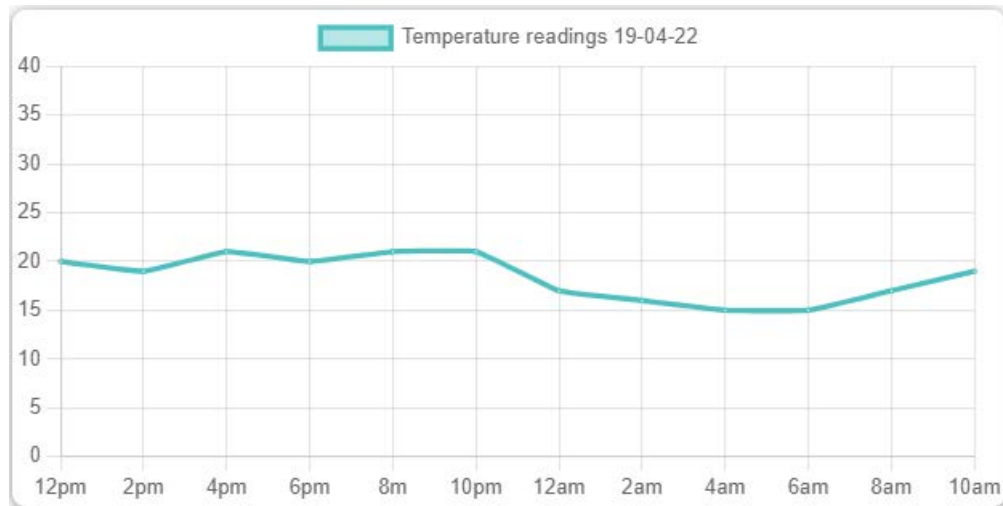


Fig 19: Reading taken on 19/04/2022

The light schedule was changed to counter the decline in temperature, they were switched off from 11 am to 5 pm. Results were a more consistent temperature.

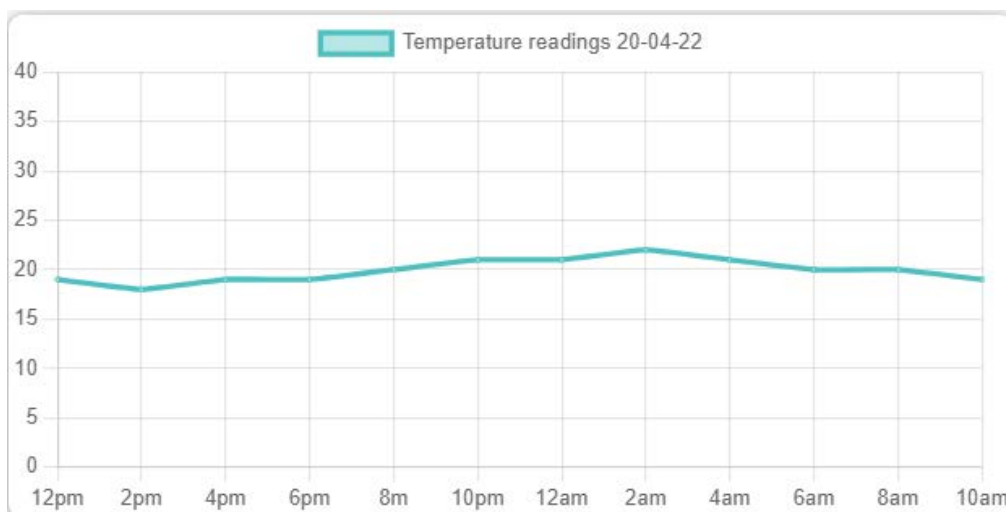
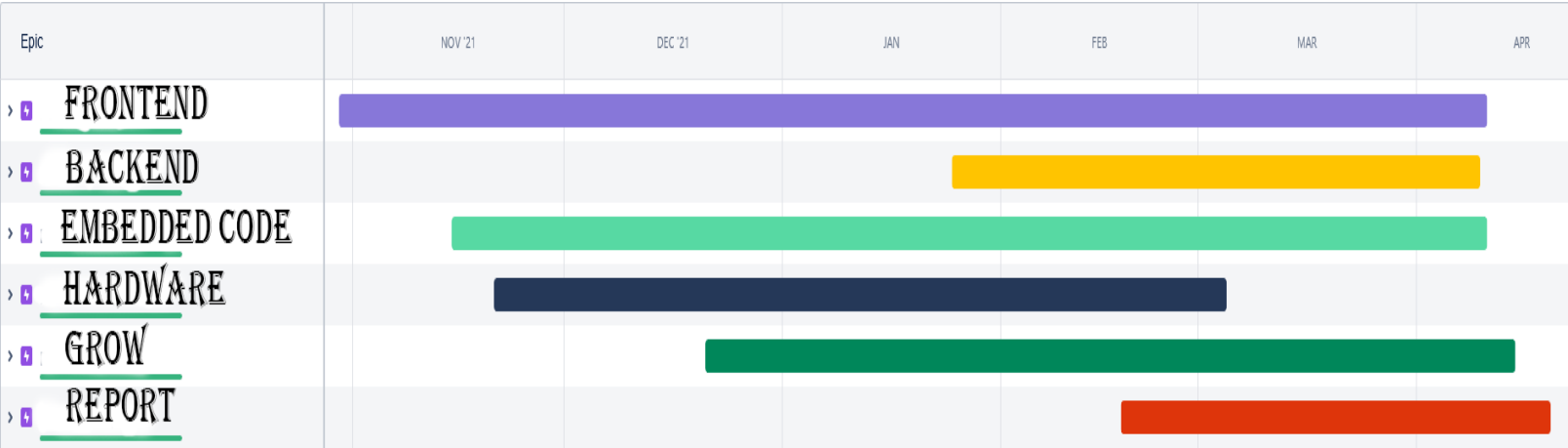


Fig 20: Reading taken on 20/04/2022

11 Project Plan

The project was documented using JIRA and Microsoft team’s notebooks. They were both updated weekly.



12 Ethics

Not only is there an increase on imports and higher costs, importing food has a negative impact on the environment [6]. Imported produce must be harvested before it has ripened fully and later sprayed with ethylene gas to aid in the ripening process [7], resulting in less taste and less nutritious produce. These are just a few of the ethical considerations.

13 Conclusion

Overall, the main project goal of developing a full-stack application that receives data from a microcontroller was successful.

Growing optimum vegetables in an indoor environment was unsuccessful, due mainly because the device was not ready for use for most of the growth cycle and the climate could not be monitored or adjusted.

The crop that was grown did not yield any significant fruit due to the environment not being adequate.

There are many opportunities for an application with the integration of a web server, microcontroller, sensors and a database server, it can be easily adapted into many systems.

14 References

- ["Huge Increase In People Growing," [Online]. Available:
1 [https://www.independent.ie/regionals/sligochampion/news/pandemic-saw-huge-rise-in-](https://www.independent.ie/regionals/sligochampion/news/pandemic-saw-huge-rise-in-people-growing-their-own-food-41177387.html)
] [people-growing-their-own-food-41177387.html](https://www.independent.ie/regionals/sligochampion/news/pandemic-saw-huge-rise-in-people-growing-their-own-food-41177387.html).
- ["What is MERN stack?," [Online]. Available: <https://www.mongodb.com/mern-stack>.
2
]
- [Adafruit. [Online]. Available: <https://learn.adafruit.com/adafruit-ccs811-air-quality-sensor>.
3
]
- ["DF Robot," [Online]. Available: [https://ie.farnell.com/dfrobot/sen0161/analog-ph-sensor-](https://ie.farnell.com/dfrobot/sen0161/analog-ph-sensor-4-meter-kit-arduino/dp/2946120?CMP=KNC-GIE-GEN-KWL-MAN-L5Pages-New-Structure-Test745&mckv=_dc|pcrid|560385016212|&gclid=Cj0KCQjwgYSTBhDKARIsAB8KuKs9yTDJPQzmr1UIBGvhLJefA_dgNd4Oc2zvweChCjvayqxJzZhcp9IaA)
4 [meter-kit-arduino/dp/2946120?CMP=KNC-GIE-GEN-KWL-MAN-L5Pages-New-Structure-](https://ie.farnell.com/dfrobot/sen0161/analog-ph-sensor-4-meter-kit-arduino/dp/2946120?CMP=KNC-GIE-GEN-KWL-MAN-L5Pages-New-Structure-Test745&mckv=_dc|pcrid|560385016212|&gclid=Cj0KCQjwgYSTBhDKARIsAB8KuKs9yTDJPQzmr1UIBGvhLJefA_dgNd4Oc2zvweChCjvayqxJzZhcp9IaA)
] [Test745&mckv=_dc|pcrid|560385016212|&gclid=Cj0KCQjwgYSTBhDKARIsAB8KuKs9yTDJPQzmr1UIBGvhLJefA_dgNd4Oc2zvweChCjvayqxJzZhcp9IaA](https://ie.farnell.com/dfrobot/sen0161/analog-ph-sensor-4-meter-kit-arduino/dp/2946120?CMP=KNC-GIE-GEN-KWL-MAN-L5Pages-New-Structure-Test745&mckv=_dc|pcrid|560385016212|&gclid=Cj0KCQjwgYSTBhDKARIsAB8KuKs9yTDJPQzmr1UIBGvhLJefA_dgNd4Oc2zvweChCjvayqxJzZhcp9IaA).
- [A. O. Domhnaill. [Online]. Available:
5 <https://github.com/AonghusOD/ProjectCode/tree/master/PictureFromTent>.
]
- ["Impact Importing Food On Environment," [Online]. Available:
6 <https://www.pnas.org/doi/10.1073/pnas.1718153115>.
]
- ["Sprayed With Ethylene Gas," [Online]. Available:
7 <https://www.hindawi.com/journals/ijfs/2019/2520179/>.
]

["BH1750 Ambient Light Sensor," [Online]. Available: <https://learn.adafruit.com/adafruit-bh1750-ambient-light-sensor>.
8 bh1750-ambient-light-sensor.
]

[D. Robot. [Online]. Available: <https://www.dfrobot.com/product-1662.html>.
9
]