COMP3901 Capstone Documentation

for

SmartGrow

Version 1.0

Prepared by

Abegail McCalla	**620157646**	**abegail.mccalla@mymona.uwi.edu**
Fabari Williams	**620154701**	**fabari.williams@mymona.uwi.edu**
Joel Foster	**620154514**	**joel.foster@mymona.uwi.edu**
Jeroen Wong	**620156810**	**jeroen.wong@mymona.uwi.edu**

Course Instructor:	**Dr. Carl Beckford**
Course:	**COMP3901—Group Capstone Project**
Group Supervisor:	**Prof. Daniel Coore**
Group Number:	**Six (6)**
Group Name:	**Defcon 4**
Due Date:	**May 15, 2025.**

TABLE OF CONTENTS

MOTIVATION

The Caribbean region is rich in agricultural potential, but Caribbean farmers are facing significant challenges in maintaining crop yields and reducing waste due to inefficient farming practices. Crop irrigation has become a challenge for farmers due to droughts. Regional models predict that significant parts of the Caribbean are likely to show an increase in temperature and a decrease in rainfall, leading to droughts, thus giving way to additional challenges in agricultural industries (Moulton et al., 2015). By implementing efficient irrigation and water management systems along with water-saving techniques, farmers can maximize crop productivity and minimize water wastage (Sil et al., 2024). As the Earth's climate continues to change, bringing about variations in temperature and precipitation patterns, agricultural methods must also adapt to assure food security, economic stability, and environmental sustainability. The increasing impacts of climate change on traditional agricultural landscapes are making it difficult for farmers to maintain steady and predictable crop yields. Rising temperatures, changing rainfall patterns, and an increase in the frequency of extreme weather events like heatwaves, floods, and droughts are all having an impact on global agricultural systems (Qwabe, 2024). Temperature management is a significant aspect of greenhouse farming because it substantially impacts plant growth, development, and overall health (Akpenpuun et al., 2023).

Light is a form of electromagnetic radiation, and its color is determined by its wavelength within the visible spectrum. Light wavelength greatly impacts photosynthesis and the rate of plant growth. Light with a wavelength shorter than 400 nm or longer than 700 nm was considered unimportant for photosynthesis due to its low quantum yield of carbon dioxide assimilation, applied as a single waveband (Liu et al., 2021). Understanding both Photosynthetically Active Radiation (PAR) and Photosynthetic Photon Flux Density (PPFD) lets the farmers get the right type of lighting that is suitable according to the requirements of the plant (Boyalite, Manufacturer of High Quality LED Grow Lights, 2025). The LEDs have many benefits. They allow wavelengths to be matched to plant photoreceptors to provide optimal production and influence plant morphology and composition (Olle et al., 2019). Monitoring tank water and fertilizer levels can be beneficial for farmers as they would be able to optimize water and fertilizer use, reduce energy costs, improve crop yields and enhance their decision-making. With frequent monitoring, the risks of under-irrigation are evaded, which results in appreciable water conservation, reduced energy consumption, and high crop production (Suresh et al., 2019). To address these challenges, this capstone project proposed the development of an Internet of Things (IoT)-powered, technological approach to farming, integrating innovative solutions to optimize crop growth, reduce waste, and promote sustainability. This project aims to design and implement an approach that integrates drip irrigation, temperature monitoring and regulation, optimized lighting along with water and fertilizer monitoring.

PROBLEM DEFINITION

Building on the agricultural challenges outlined in the motivation, integrating an IoT-powered technological approach into farming offers a transformative solution, enhancing both development and efficiency in modern agriculture. Poor plant watering, inadequate temperature regulation, and inefficient resource monitoring are crucial problems to be solved because they directly impact plant health, yield, and sustainability in agriculture. Underwatering can cause root damage, nutrient deficiencies, or wasted resources, while uncontrolled temperature fluctuations reduce photosynthesis efficiency and stress plants. Additionally, failing to monitor tank water and fertilizer levels can lead to shortages or excessive application, both of which harm crop growth and increase operational costs. By implementing optimized lighting, automated sensing, automated actuators, resource management and alert systems, farmers can optimize irrigation, temperature control, and nutrient distribution, ensuring improved efficiency, reduced waste, and healthier plants in smart greenhouse environments. Solving poor plant watering, temperature regulation, and resource monitoring issues using smart greenhouses is crucial for improving agricultural efficiency, sustainability, and crop health. By ensuring precise irrigation, temperature control, optimized lighting and automated tracking of tank water and fertilizer levels, farmers can reduce waste, optimize resource use, and maintain stable growing conditions. These advancements lead to higher yields, lower operational costs, and a more eco-friendly farming approach by minimizing excess water consumption and chemical runoff. Additionally, automated alerts and IoT-based monitoring reduce manual labor while enhancing decision-making, making smart greenhouses a transformative solution for modern agriculture.

As mentioned in the motivation, the general objective of this capstone project is to address the traditional agricultural challenges mentioned prior by proposing the development of a smart greenhouse system, using an IoT-based technological approach to farming, that integrates automated drip irrigation, temperature regulation, optimized LED lighting, and real-time monitoring of water and fertilizer levels to enhance agricultural efficiency in tropical or subtropical regions. Given the challenges posed by climate change, including rising temperatures and irregular rainfall, farmers require advanced solutions to optimize crop growth while minimizing waste. By leveraging IoT-based sensing and automation, this system ensures precise irrigation, stable temperature monitoring and regulation, optimized lighting and efficient resource management, reducing operational costs and environmental impact. The significance of this project lies in its ability to promote sustainable farming practices, improve food security, and provide a scalable technological framework for adapting to changing agricultural conditions.

Given crops that can thrive in tropical or subtropical regions such as cereals like rice and maize, pulses such as chickpea, kidneybeans, pigeonpeas, mothbeans, mungbean, blackgram and lentil, fruits such as pomegranate, banana, mango, grapes, watermelon, muskmelon, apple, orange, papaya and coconut and **c**ommercial crops such as cotton, jute and coffee, the specific objective of this capstone project is to develop a system that can maintain optimal growing condition for these crops over their lifetimes, by maintaining optimal soil moisture and temperature levels, allowing the farmer to change the colour of the lighting, and monitor resources such as water and fertilizer levels by alerting the farmer when the levels get to 10% capacity. For efficacy, the problem is broken into soil quality maintenance (soil moisture regulation), air quality maintenance (temperature regulation and optimal lighting) and resource monitoring (monitoring tank water and tank fertilizer levels). Information on the optimal growing conditions for these crops will be taken from a dataset of crop recommendations found on Kaggle, which is a platform for data science and machine learning. The table below contains the variables that the system should monitor and the optimal values as well as allowable deviation ranges that the system should maintain for the respective crops over their lifetimes.

Table 1 showing the approximate temperature and soil moisture optimal values and allowable deviations for the respective crops

Crop	**Temperature (°C)**	**Min Temperature (°C)**	**Max Temperature (°C)**	**Soil Moisture (mm)**	**Min Soil Moisture (mm)**	**Max Soil Moisture (mm)**
rice	23.68933221	20.0454142	26.92995077	236.1811136	182.5616319	298.5601175
maize	22.38920391	18.04185513	26.54986394	84.76698766	60.65171481	109.7515385
chickpea	18.87284675	17.02498456	20.99502153	80.05897726	65.11365631	94.78189594

kidneybeans	20.11508469	15.33042636	24.92360104	105.9197775	60.27552528	149.7441028
pigeonpeas	27.74176223	18.31910448	36.97794384	149.4575638	90.05422663	198.8298806
mothbeans	28.19492048	24.01825377	31.99928579	51.19848705	30.92014047	74.44330654
mungbean	28.52577474	27.01470397	29.9145443	48.4036009	36.12042927	59.87232071
blackgram	29.97333968	25.09737391	34.9466155	67.88415118	60.41790253	74.91559514
lentil	24.5090524	18.06486101	29.94413861	45.6804542	35.03484812	54.9393771
pomegranate	21.83784172	18.07132963	24.96273236	107.5284424	102.5184759	112.4750941
banana	27.37679831	25.01018457	29.90888522	104.6269804	90.10978128	119.84797
mango	31.20877015	27.00315545	35.99009679	94.70451504	89.29147581	100.8124659
grapes	23.84957512	8.825674745	41.94865736	69.61182889	65.01095312	74.91506217
watermelon	25.59176724	24.04355803	26.98603693	50.78621894	40.12650421	59.75980023
muskmelon	28.66306576	27.02415146	29.94349168	24.68995207	20.21126747	29.86681385
apple	22.63094241	21.0365275	23.99686172	112.6547793	100.1173443	124.9831618
orange	22.7657255	10.01081312	34.90665289	110.4749687	100.1737964	119.6946577
papaya	33.72385874	23.0124018	43.67549305	142.6278389	40.35153141	248.8592986

coconut	27.40989217	25.00872392	29.8690834	175.6866458	131.0900076	225.6323656
cotton	23.9889579	22.00085141	25.99237426	80.39804312	60.65381719	99.93100821
jute	24.95837583	23.09433785	26.98582182	174.7927975	150.2355238	199.8362913
coffee	25.54047682	23.05951896	27.92374437	158.0662949	115.1564012	199.4735636

PROPOSED SOLUTION

To address the issues that were defined in the problem definition prior, this capstone project proposed the development of an IoT-based technological approach to farming, integrating sensors, data analytics and actuators to produce an innovative solution to optimize crop growth, reduce waste, and promote sustainability. Specifically, this project aims to design and implement an approach that integrates drip irrigation, temperature monitoring and regulation, optimized lighting along with water and fertilizer monitoring. In terms of data input, the system will allow the user to select a crop from the list of given crops mentioned in the problem definition. To maintain accuracy, the system will only allow the user to select one crop to be monitored at a time. The system will then read live sensor data for various environmental parameters, such as soil moisture, temperature, tank water level and tank fertilizer level. The system will also allow the farmer to enter the brightness or color of the RGB LED strips. The input data will be stored in the MongoDB COMP3901 database greenhouse collection.

In terms of data processing, the live data collected by the sensors will be compared with the pre-existing Kaggle data in Table 1 in the problem definition, which outlines the fundamental environmental conditions necessary for crop growth. The Kaggle data will be stored in the MongoDB COMP3901 database crops collection. The system will also process the LED color entered by the farmer in terms of RGB ratio (the ratio of red to green to blue) to determine the LED color. In terms of output, if the data being collected by the sensors deviates outside of the allowable deviation ranges, the system will enable actuators such as heaters, fans, a valve and level warnings to correct the deviation. The system will regulate the environmental parameters, as mentioned prior, for optimal crop growth by activating actuators to correct deviations outside of the allowable ranges mentioned in the problem definition. The proposed solution aims to address the three parts that the problem is broken into, as mentioned in the problem definition. Therefore, the solution consists of soil quality maintenance (soil moisture regulation), air quality maintenance (temperature regulation and optimal lighting) and resource monitoring (monitoring tank water and tank fertilizer levels).

In terms of soil quality maintenance, the soil's moisture will be monitored and regulated, using drip irrigation, for optimal plant growth conditions so that plants may thrive. If the soil moisture deviates outside of the allowable minimum value, the system will open the valve for 2.5 seconds to allow water to flow into the tube, which is within the soil, and drip into the soil so that the plants can get the water. If the deviation is not corrected after five minutes, the system will reopen the valve for 2.5 seconds. Therefore, the system will repeat this process at five-minute intervals until the deviation has been corrected.

In terms of air quality maintenance, the temperature will be monitored and regulated for optimal plant growth conditions so that the plants may thrive. Higher temperatures can hold more moisture, increasing the relative humidity if the moisture content remains constant. Conversely, at lower temperatures, the air holds less moisture, leading to lower relative humidity. As temperatures rise, the heat index also increases, especially when combined with high humidity. These three elements are interconnected; therefore, humidity and heat index are also monitored. However, temperature will be directly manipulated. If the temperature deviates outside of the allowable minimum value, the system will enable three heaters, which are strategically placed inside the greenhouse to maximize efficacy, to increase the temperature. The heaters will remain on until the deviation is corrected. If the temperature deviates outside of the allowable maximum value, the system will enable two fans, which are strategically placed on either side of the greenhouse to maximize efficacy, to decrease the temperature. The fans will remain on until the deviation is corrected. Natural sunlight will be the main source of energy for the crops during the day; however, the system will allow the farmer to turn on the programmable RGB LED strips. The farmer should be able to adjust the color and brightness of the LED light at their convenience. RGB LEDs are highly beneficial for greenhouse farming, especially when natural sunlight is unavailable. Their ability to provide customized light spectrums helps optimize plant growth, flowering, and energy efficiency. LEDs provide precise light wavelengths to enhance plant oxygen production, thus optimizing photosynthesis. LEDs consume less power, making them energy efficient, which lowers carbon emissions from energy sources. Farmers can adjust LED intensity based on plant needs and air quality, resulting in climate control. LEDs produce far less heat than traditional incandescent or halogen bulbs.

In terms of resource monitoring, the tank water level and tank fertilizer level will be monitored so that the plants may always have access to the resources needed to thrive. If the tank water level or tank fertilizer level deviates outside of the allowable minimum value (less than or equal to 10% capacity), the system will alert the farmer that they are running out of water or fertilizer, respectively. The system will do this by flashing an alert message. If the water level is less than or equal to 10% capacity, the system will flash the message “Water Level is Critically Low.” If the fertilizer level is less than or equal to 10% capacity, the system will flash the message “Fertilizer Level is Critically Low.” Implementing this system will reduce water waste and water consumption, increase crop yields and quality, and enable a real-time monitoring and decision-support system for farmers.

CONSTRAINTS, LIMITATIONS AND EXCLUSIONS

In terms of constraints, the system must be completed by the demonstration date (May 13, 2025). The system must ensure proper air quality for optimal plant growth by monitoring and regulating temperature and allowing the farmer to manipulate the brightness and color of the RGB LEDs. The system must ensure proper soil quality for optimal plant growth by monitoring and regulating soil moisture. The system must ensure proper resource monitoring and alert the farmer when the tank water and tank fertilizer levels are less than or equal to 10% capacity.

Limitations such as Kaggle's allowable ranges for environmental parameters, sensor accuracy (how close the sensor's reading is to the true or actual value), sensor sensitivity (how much a sensor's output changes per unit change in the measured quantity), sensor resolution (the smallest change in input that produces a detectable change in output), sensor hysteresis (the difference in output at the same input level when the input is increasing vs. decreasing), sensor drift (the slow change in the sensor's baseline output over time, unrelated to input changes), sensor linearity (the degree to which the sensor output is directly proportional to its input across its range), actuator response time and effectiveness as well as system scalability and adaptability to different crop types and environmental conditions may pose as inherent restrictions for this project.

In terms of exclusions, AI disease detection and cloud integration are not included in the project due to time constraints and skill limitations. Even though only temperature and soil moisture data were used from the Kaggle dataset, optimal values and allowable deviation ranges were also provided for other parameters such as soil pH and soil npk (nitrogen, phosphorus and potassium). The scope of this project does not include monitoring and regulating soil pH or soil npk due to the cost for suitable sensors and actuators being outside of the project's budget.

RELEVANT TECHNICAL BACKGROUND

The study entitled "Greenhouse Technology is Once Again Washing the Caribbean—Can We Ride the Wave This Time Around?" explored the reintroduction of greenhouse technology in the Caribbean, with a particular focus on Trinidad and Tobago and Jamaica, emphasizing its potential to modernize agriculture and boost crop productivity. It highlighted various structural improvements, such as enhanced ventilation systems and insect protection, which are essential for maintaining optimal environmental conditions within greenhouses. These insights closely align with SmartGrow's emphasis on temperature regulation and optimized lighting, as proper climate management plays a crucial role in heat control and photosynthesis efficiency. Furthermore, the study identified key challenges faced by greenhouse farmers, including limited technical support, financial constraints, and excessive heat and humidity, reinforcing the need for automated sensing and climate control solutions, which are features that are integrated within SmartGrow.

The research also examined historical attempts at greenhouse farming in the Caribbean, including trials with soilless culture systems and nutrient film techniques (NFT) designed to improve water efficiency and crop yield. These findings directly support the drip irrigation and resource monitoring components of SmartGrow, as they underscore the importance of precise water and nutrient management in greenhouse environments. Additionally, the study discussed pesticide reduction strategies using insect screens by minimizing the reliance on chemical treatments. While the study offered valuable insights into structural adaptations for controlled environments, it did not directly indicate the presence of advanced automation technologies, such as IoT-powered sensing, dynamic climate regulation, or real-time data processing.

SmartGrow can be used to improve this existing system as it integrates continuous sensor monitoring, automated irrigation, temperature control, and dynamic actuator responses, ensuring real-time environmental adjustments without requiring manual intervention. The study's lack of mention of these advanced automation capabilities suggests that traditional Caribbean greenhouse designs still depend on human oversight for irrigation and climate management, whereas SmartGrow reduces human inefficiencies through intelligent automation. By incorporating IoT-powered automation, SmartGrow significantly enhances efficiency, scalability, and sustainability, making it far more adaptive than conventional greenhouse models.

Moreover, while traditional greenhouse designs rely on predefined structural configurations, SmartGrow utilizes real-time data analytics and automated actuators, allowing for dynamic environmental adjustments based on crop requirements. This approach not only optimizes resource distribution and climate conditions but also ensures continuous monitoring of soil moisture, temperature, tank water, and fertilizer levels, further improving greenhouse efficiency. By addressing the technological gaps present in conventional Caribbean greenhouse systems, SmartGrow delivers a highly responsive, energy-efficient, and scalable agricultural solution, advancing modern greenhouse technology beyond static environmental controls.

COMPONENTS AND MATERIALS

SOFTWARE:

➢ **Backend:** Flask API (routes), MQTT (broker), PyMongo (database handling)

Flask over other web frameworks such as Django and FastAPI

Flask:

- Middleware: Flask serves as a backend API that connects your Vue.js frontend to the MongoDB database It exposes endpoints (like /api/mmar/soil/<start>/<end>,/api/crop/update/<crop> etc.) that Vue can call via fetch.
- Simplicity and Flexibility: Flask is lightweight and easy to set up, making it ideal for small to medium-sized projects and quick prototypes.
- Microframework: Unlike Django, which is more opinionated and comes with many built-in features, Flask gives you the flexibility to pick and choose components.
- Customization: You can easily extend Flask with custom middleware and add-ons, providing more control over your application's architecture.

Django:

- Complexity: While powerful, Django comes with a steeper learning curve and more boilerplate code.
- Batteries-Included: Django includes a lot of built-in features (like ORM, authentication, etc.) that may be overkill for simpler projects.

FastAPI:

- Asynchronous Capabilities: FastAPI is optimized for building APIs with asynchronous capabilities, but it may be more complex for beginners.
- More Features: FastAPI provides automatic interactive API documentation, which is useful but may add unnecessary complexity for smaller projects.

MQTT over other messaging protocols such as HTTP and AMQP (Advanced Message Queuing Protocol)

MQTT:

- Resilient: Built to withstand frequent network disruption and intermittent, slow, or poor quality networks
- Space Efficient: Our project uses MQTT with a Quality of Service (QoS) level of 0, which enables best-effort message delivery without requiring acknowledgments. This makes the setup simpler and conserves memory and processing resources on the ESP32, making it well-suited for space-constrained embedded systems.
- Lightweight: MQTT is designed for low-bandwidth, high-latency networks, making it ideal for IoT devices and constrained environments.
- Publish/Subscribe Model: MQTT uses a pub/sub model, which is efficient for many-to-many communication scenarios and real-time data streaming.
- Scalability: MQTT brokers can handle thousands of concurrent clients with minimal overhead.

HTTP:

- Request/Response Model: HTTP uses a request/response model, which is less efficient for real-time communication and many-to-many messaging scenarios.
- Bandwidth Usage: HTTP has higher bandwidth usage due to its stateless nature and headers.

AMQP:

- Complexity: AMQP is more complex and heavier compared to MQTT. It is designed for high-reliability enterprise messaging but may be overkill for simpler IoT applications.
- Overhead: AMQP requires more resources and is not as lightweight as MQTT, making it less suitable for constrained devices.

PyMongo over other database libraries such as SQLAlchemy and Django ORM

PyMongo:

- Integrated Authentication and Security: PyMongo supports MongoDB's built-in authentication, TLS/SSL encryption, and user role-based access.
- Pythonic Interface: PyMongo provides a seamless and Pythonic way to interact with MongoDB, making it easy for Python developers to use.

- Flexibility: MongoDB, being a NoSQL database, allows for flexible schema designs and easy scaling.
- Rich Query Language: PyMongo supports MongoDB's powerful query language, enabling complex queries and aggregations on large datasets.

SQLAlchemy:

- Relational Database Focus: SQLAlchemy is designed for relational databases and is more complex when dealing with unstructured or semi-structured data.
- ORM Overhead: SQLAlchemy's ORM adds an additional layer of abstraction, which may lead to performance overhead and complexity.

Django ORM:

- Tight Integration: Django ORM is tightly integrated with the Django framework, making it less flexible for projects that do not use Django.
- Learning Curve: Django ORM has a steeper learning curve and may be overkill for simple CRUD operations on MongoDB.

➢ **Frontend:** Vuejs, Vuetify – HTML, CSS and JavaScript

Vue.js over other frontend frameworks and libraries such as React and Angular

Vue.js:

- Ease of Learning: Vue.js has a gentle learning curve, making it easier for beginners to grasp compared to React or Angular.
- Simplicity: Vue.js provides a clear and straightforward syntax, which is intuitive for both new and experienced developers.
- Flexibility: Vue.js is versatile and can be incrementally adopted. You can use as much or as little of it as you need, making it suitable for a wide range of projects.
- Reactivity System: Vue’s reactivity system is efficient and easy to work with, enabling seamless state management and reactivity.

React:

- Steeper Learning Curve: React’s use of JSX and the need to learn additional concepts like hooks can be more challenging for newcomers.
- More Boilerplate: React often requires more boilerplate code and additional configuration compared to Vue.js.

Angular:

- Complexity: Angular is a full-fledged framework with a steeper learning curve and more complexity, which can be overwhelming for smaller projects.
- Opinionated: Angular is more opinionated and less flexible in terms of structuring your application compared to Vue.js.

Vuetify over other UI libraries/frameworks such as Bootstrap and Material-UI

Vuetify:

- Integration with Vue.js: Vuetify is specifically designed for Vue.js, providing seamless integration and a consistent development experience.
- Material Design: Vuetify implements Google's Material Design specifications, offering a wide range of high-quality, pre-built components.
- Ease of Use: Vuetify's components are easy to use and customize, allowing developers to create visually appealing UIs with minimal effort.

Bootstrap:

- Generic: While Bootstrap is versatile, it is not specifically tailored for Vue.js, which may result in more work to integrate.
- Less Customizable: Bootstrap components may require more customization to fit the Material Design look and feel.

Material-UI:

- React-Focused: Material-UI is designed primarily for React, making it less suitable for Vue.js projects.
- Learning Curve: Material-UI can be more complex to set up and use compared to Vuetify, especially for Vue.js developers.

HTML, CSS, and JavaScript over other web technologies such as Preprocessors and Complex Build Tools

HTML, CSS, and JavaScript:

- Ubiquity: These are the core technologies of the web, supported by all browsers and used in virtually every web project.
- Ease of Learning: HTML, CSS, and JavaScript are relatively easy to learn and form the foundation of web development.

- Flexibility: These technologies allow for a wide range of customization and creativity without the constraints of additional abstractions or preprocessors.

Preprocessors (e.g., SASS, LESS):

- Additional Learning Curve: While preprocessors offer powerful features, they require additional learning and setup.
- Dependency: Using preprocessors adds a layer of dependency, which may complicate the development process.

Complex Build Tools (e.g., Webpack, Gulp):

- Complexity: Build tools can add significant complexity to a project, especially for beginners or small projects.
- Configuration Overhead: Setting up and maintaining build tools requires additional effort and expertise.

➢ **Database:** MongoDB (Cloud platform for data storage and analysis)

MongoDB over other relational databases such as MySQL and PostgreSQL

- Integrated Aggregation Framework: Enables complex data processing and transformation within the database.
- MongoDB is chosen for its flexible schema design, horizontal scalability, and ease of use, making it more suitable than relational databases like MySQL and PostgreSQL.

MongoDB over other noSQL databases such as Cassandra and CouchDB

- MongoDB offers rich querying and indexing capabilities, making it more user-friendly than other NoSQL databases like Cassandra and CouchDB.

MongoDB over other in-memory databases such as Redis and Memcached

- MongoDB provides durable data storage with complex data modeling, making it more suitable for long-term data storage compared to in-memory databases like Redis and Memcached.

HARDWARE:

➢ **2, DHT22 Sensor (Temperature, humidity & heat index sensor)**

DHT22 sensor over DHT11 sensor

- Temperature Range and Accuracy: DHT22 measures temperatures from -40 to 80°C with an accuracy of ±0.5°C while DHT11 measures temperatures from 0 to 50°C with an accuracy of ±2°C2.
- Humidity Range and Accuracy: DHT22 measures humidity from 0 to 100% with an accuracy of ±2-5% while DHT11 measures humidity from 20 to 80% with an accuracy of ±5%.
- Resolution: DHT22 provides 16-bit resolution for both temperature and humidity while DHT11 provides 8-bit resolution for both temperature and humidity.

➢ **2, Capacitive Soil Moisture Sensor**

Capacitive Soil Moisture Sensors over Resistive Soil Moisture Sensors

Capacitive Soil Moisture Sensors

- How They Work: Measure soil moisture levels by detecting changes in capacitance caused by the presence of water in the soil.
- Advantages:
  - Corrosion Resistance: Made of non-corrosive materials, ensuring a longer lifespan compared to resistive sensors.
  - Accuracy: Provide more stable and accurate readings as they are less affected by soil salinity or electrolysis.
  - Durability: Suitable for long-term use in various soil conditions.
  - Low Power Consumption: Ideal for battery-powered or IoT applications.
  - Are often considered superior to other types of soil moisture sensors due to their durability, accuracy, and resistance to environmental challenges.

Resistive Soil Moisture Sensors

- How They Work: Use two probes to measure the resistance of the soil, which changes with moisture levels.
- Disadvantages Compared to Capacitive Sensors:
  - Corrosion: The probes are prone to corrosion due to electrolysis, especially in wet or saline soils.
  - Short Lifespan: Require frequent replacement due to wear and tear.

        - Less Accurate: Affected by soil salinity and temperature, leading to inconsistent readings.

➢ **1, ESP32 microcontroller**

ESP32 over Arduino Nano

Processing Power:

- ESP32: Features a dual-core Xtensa LX6 processor with a clock speed of up to 240 MHz, making it significantly faster and more capable of handling complex tasks.
- Arduino Nano: Uses an ATmega328P microcontroller with a clock speed of 16 MHz, which is sufficient for basic tasks but limited for more demanding applications.

Connectivity:

- ESP32: Comes with built-in Wi-Fi and Bluetooth (including BLE), making it ideal for IoT projects and wireless communication.
- Arduino Nano: Lacks built-in Wi-Fi or Bluetooth, requiring external modules for wireless connectivity.

Memory:

- ESP32: Offers up to 520 KB of SRAM and 4 MB of flash memory, with options for additional PSRAM in some models.
- Arduino Nano: Limited to 2 KB of SRAM and 32 KB of flash memory, which can be restrictive for larger programs.

Power Efficiency:

- ESP32: Supports multiple power-saving modes, making it suitable for battery-powered and energy-efficient projects.
- Arduino Nano: Consumes more power relative to its capabilities and lacks advanced power management features.

I/O and Peripherals:

- ESP32: Provides a wide range of peripherals, including capacitive touch sensors, ADC, DAC, PWM, SPI, I2C, UART, and more.

- Arduino Nano: Offers fewer peripherals and lacks advanced features like capacitive touch or DAC.

- **2, Ultrasonic radar sensors**
- **2, Programmable RGB LED strips (LED control)**
- **1, Breadboard**
- **2, 5V DC fans**
- **1, 110V AC heater**
- **2, 12V DC heaters**
- **3, 5V DC relays**
- **1, 12V DC water valve**
- **Transistor**
- **Wires**
- **Protection gear (mask and gloves)**
- **Plexiglas**

HARDWARE DESIGN

7.1 Hardware Block Diagram

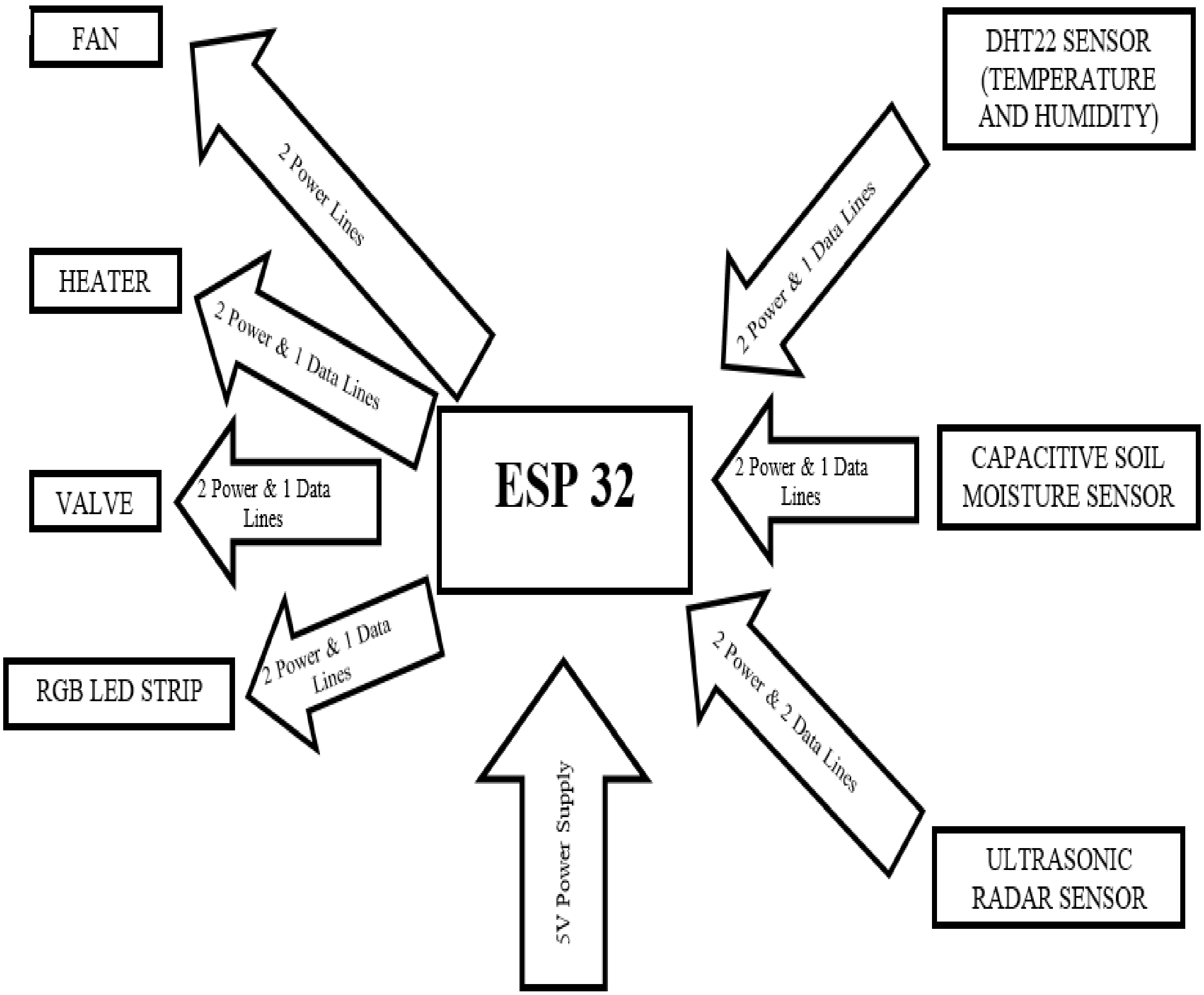


Figure 1 shows the hardware block diagram for the system.

The ESP is interfaced with the DHT22 temperature sensor, the capacitive soil moisture sensor, the fan, the heater, the valve and the RGB LED strip. The ESP32 microcontroller unit will take the data input from the sensors, compare it with the existing Kaggle data then send instructions to the actuators accordingly.

SOFTWARE DESIGN

8.1 User Functional and Non-Functional Requirements

Table 2 shows the user functional and non-functional requirements

Requirement Number	User Functional Requirements	User Non-Functional Requirements
1	The user shall be able to select a crop from the provided list.	The system must have an intuitive interface for selecting crops
2	The user shall be able to alter the brightness and color of the RGB LEDs.	The system shall allow the user to adjust LED settings and ensure ease of use for the user.
3	The user shall receive warnings when water or fertilizer levels are critically low.	Information must be presented concisely to avoid user frustration or confusion.
4	The user shall be able to view the data collected by the sensors on webpages	The system must ensure that users can easily read and navigate the webpage. Fonts, colors, contrast, and layout must be optimized for readability.

8.2 System Functional and Non-Functional Requirements

Table 3 shows the system functional and non-functional requirements

Requirement Number	System Functional Requirements	System Non-Functional Requirements
1	The system shall allow the user to select a crop from the menu and update that crop's selection status to true in the MongoDB COMP3901 database crops collection	System must support multiple crop profiles but allow monitoring of only one crop at a time
2	The system shall allow the user to enter the brightness and color of their desire for the RGB LEDs	The system must provide a wide range and spectrum of preset colours and brightness levels that the user can select from
3	The system shall perceive its environment by detecting changes in soil moisture and temperature parameters as well as the changes in the levels of the tank resources using sensors.	The system shall ensure that environmental perception is highly reliable, with the DHT22 sensor maintaining an accuracy of at least ±0.5°C for temperature, the soil moisture sensor providing a stable and accurate reading, as it is less affected by soil salinity or electrolysis and the ultrasonic sensor remaining operable under standard operating conditions. The system shall ensure that the sensors have a sample rate of at least 0.5 Hz
4	The system shall store the values of the parameters and levels of the tank resources in MongoDB COMP3901 database	The system shall ensure that the data is stored in a neat and readable format

	greenhouse collection	
5	The system shall compare the MongoDB COMP3901 database greenhouse collection parameter values with the pre-existing optimal parameter values and allowable deviation ranges that are stored in the MongoDB COMP3901 database crops collection	The system shall ensure that the comparison of greenhouse collection parameter values with pre-existing optimal values and allowable deviation ranges is executed within an average processing time of ≤500ms, ensuring real-time decision-making without delays in automation.
6	The system shall enable the actuators, such as heaters, fans and a valve, to correct any deviations in the parameters, as specified in the proposed solution.	The system must ensure that the actuators respond within 5 seconds from when the deviation occured The system must open the valve for 2.5 second at 5-minute intervals whenever the soil moisture deviates below the minimal allowable value until the soil moisture deviation is corrected
7	The system shall alert the user when the levels of the tank resources are less than or equal to 10% capacity	The system shall notify the user within 5 seconds when water or fertilizer levels is less than or equal to 10% capacity
8	The system shall adjust the brightness and color of the RGB LEDs based on the user's input.	System must update the RGB LEDs within 5 seconds based on user input

8.3 Use Case Specifications and Activity Diagrams

Requirement 1: Select A Crop

Table 4 showing the use case specification for requirement 1

Use case name	**Select A Crop**	
Related Requirements	Requirement 1	
Goal in Context	Allows the farmer to select a crop from the menu	
Preconditions	The farmer has fully implemented the system	
Successful End Condition	A crop has successfully been selected and the system updates the crop's selection status to true in the MongoDB COMP3901 database crops collection	
Failed End Condition	A crop has unsuccessfully been selected and the crop's selection status remains false in the MongoDB COMP3901 database crops collection	
Primary Actors	The farmer	
Trigger	The farmer selects the option to select a crop	
Main Flow	Step	Action
	1	The farmer selects the option to select a crop
	2	The system shall allow the farmer to select a crop from the menu
	3	The system shall update the crop's selected status to true in the MongoDB COMP3901 database crops collection
Extensions	Steps	Branching Action

	3.1	The system shall not update the crop's selected status to true in the MongoDB COMP3901 database crops collection

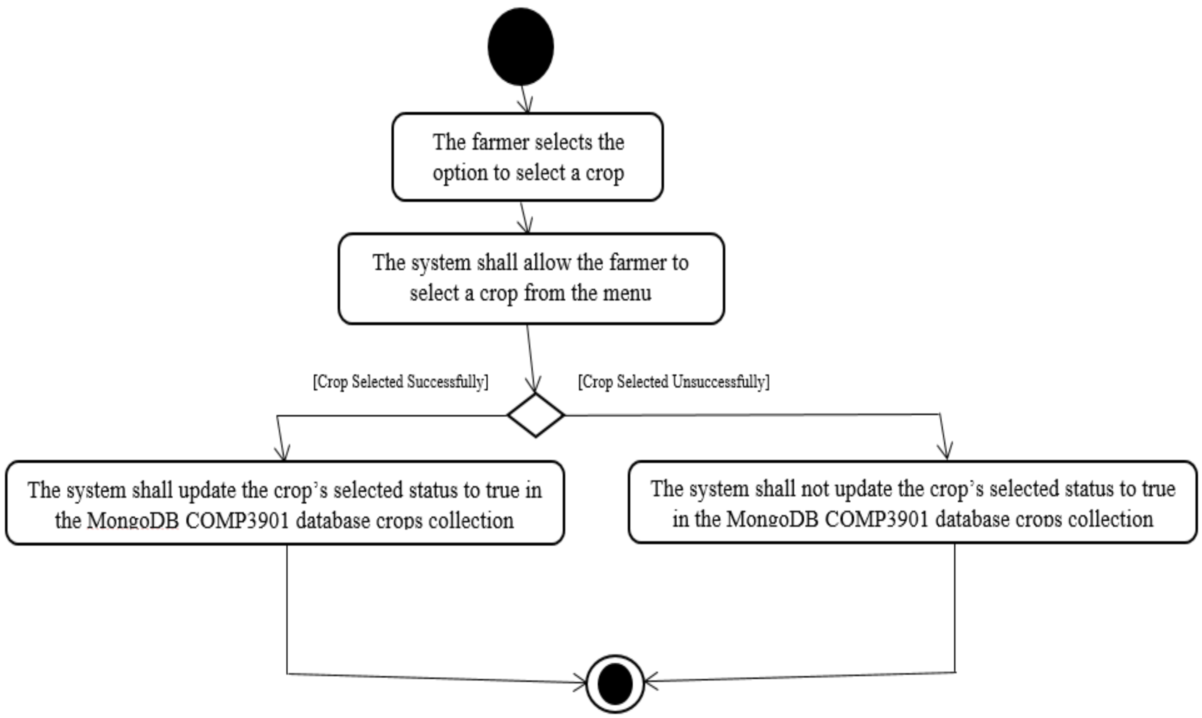


Figure 2 shows the activity diagram for the "Select A Crop" use case.

Requirement 2: Alter RGB LEDs

Table 5 showing the use case specification for requirement 2

Use case name	**Alter RGB LEDs**
Related Requirements	Requirement 2
Goal in Context	Allows the farmer to alter the brightness or color of the RGB LEDs
Preconditions	The farmer has fully implemented the system
Successful End Condition	The RGB LEDs reflect the brightness or color input from the farmer

Failed End Condition	The RGB LEDs do not reflect the brightness or color input from the farmer	
Primary Actors	The farmer	
Trigger	The farmer enters the brightness or color desired	
Main Flow	Step	Action
	1	The farmer enters the brightness or color desired
	2	The system shall alter the brightness or color of the RGB LEDs based on the farmer's input
Extensions	Steps	Branching Action
	2.1	The system shall not alter the brightness or color of the RGB LEDs based on the farmer's input

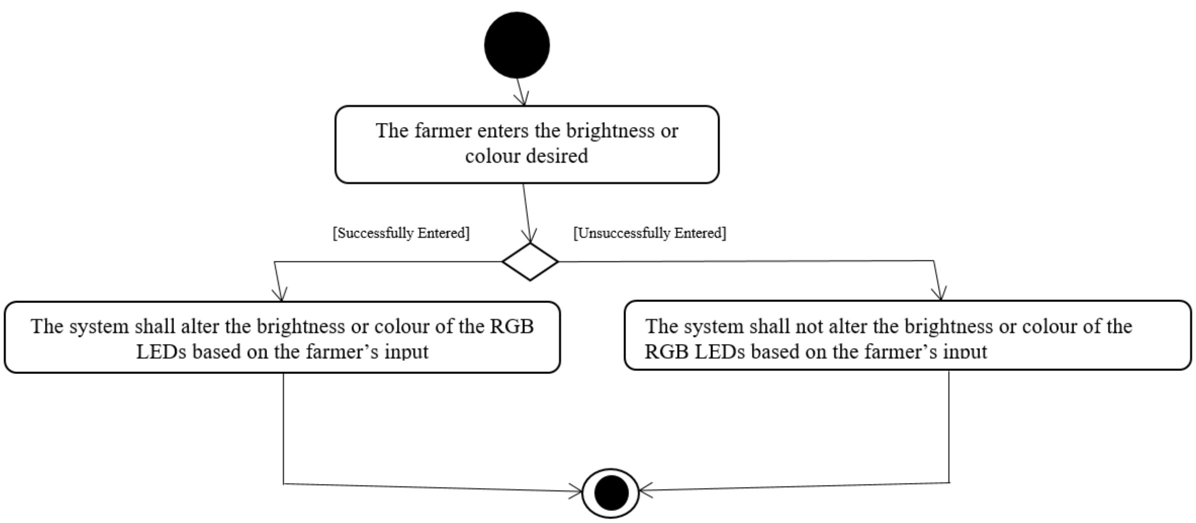


Figure 3 shows the activity diagram for the "Alter RGB LEDs" use case.

8.4 Evaluation and Verification Test Plan

Table 6 shows the system's test cases for system evaluation and verification

Test Case	**Expected Result**	**Actual Result**
The water or fertilizer level is critically low.	User receives the warning “Water Level is Critically Low” or “Fertilizer Level is Critically Low” when the water or the fertilizer level is less than or equal to 10% capacity, respectively	User received the warning “Water Level is Critically Low” or “Fertilizer Level is Critically Low” when the water or the fertilizer level is less than or equal to 10% capacity, respectively
The user wants to select crops and adjusting LED settings	User sees a menu with the available crop options when they select the option to select a crop and user is presented with a visually interactive graphical user interface to adjusting LED settings	User saw a menu with the available crop options when they select the option to select a crop and user is presented with a visually interactive graphical user interface to adjusting LED settings
The user attempts to read and navigate the webpage.	User sees a navigation bar at the top of each webpage	User saw a navigation bar at the top of each webpage

Information presentation.	The user sees a graphical representation of the information to avoid wordiness, ensuring a concise presentation to avoid user frustration or confusion.	The user saw a graphical representation of the information to avoid wordiness, ensuring a concise presentation to avoid user frustration or confusion.
User is able to change the colour to purple and set brightness to max (250)	The user sees the system reflect these changes within 5 seconds	User saw the system reflect these changes within 5 seconds
User attempts to select rice and maize simultaneously	Option menu closes once the first crop is selected	Option menu closed once the first crop was selected
The user attempts to select rice only from the provided list.	The system updates the rice’s selection status to true in the MongoDB COMP3901 database crops collection	The system updated the rice’s selection status to true in the MongoDB COMP3901 database crops collection

The current soil moisture is 100mm, the current temperature is 15°C, tank water level to be at 10% and the fertilizer level to be at 50%.	The system detects changes in soil moisture and temperature parameters as well as the changes in the levels of the tank resources using sensors. The user sees that the reading is taken at a sample rate of at least 0.5Hz	The system detected changes in soil moisture and temperature parameters as well as the changes in the levels of the tank resources using sensors. The user saw that the reading is taken at a sample rate of at least 0.5Hz
The data: name: rice, temperature: 15°C, soil moisture: 100mm, water level: 10%, fertilizer level: 50% has been collected by the sensors.	The system stores the values of the parameters resources in MongoDB COMP3901 database greenhouse collection	The system stored the values of the parameters resources in MongoDB COMP3901 database greenhouse collection
The greenhouse collection soil moisture (100mm) deviated below the minimum value of the allowable range from the crops collection (182.5616319mm to 298.5601175mm), the greenhouse collection temperature (15°C) deviated below the minimum value of the	The system compares the MongoDB COMP3901 database greenhouse collection parameter values with the pre-existing optimal parameter values and allowable deviation ranges that are stored in the MongoDB COMP3901 database crops collection. The user	The system compared the MongoDB COMP3901 database greenhouse collection parameter values with the pre-existing optimal parameter values and allowable deviation ranges that are stored in the MongoDB COMP3901 database crops collection. The user saw that the comparison is executed within an average processing time of ≤500ms

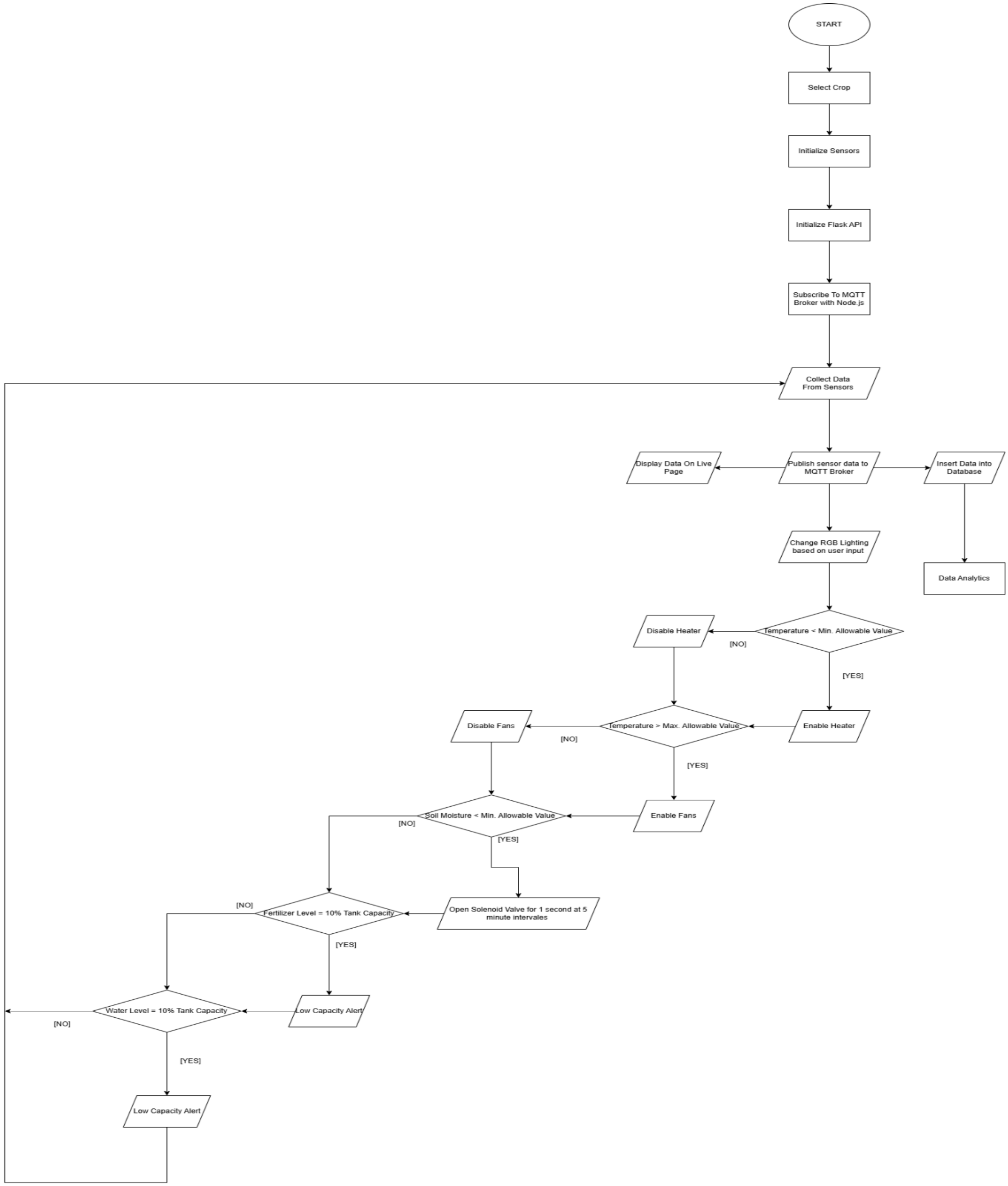
allowable range from the crops collection (20.0454142°C to 26.92995077°C), the water level has reached critical value and the fertilizer level has not reached critical value.	sees that the comparison is executed within an average processing time of ≤500ms	
The levels of the tank resources reach 10% capacity	The system alerts the user "Water Level is Critically Low" within 5 seconds	The system alerted the user "Water Level is Critically Low" within 5 seconds
The temperature and the soil moisture get too low	The system enables the valve and the heaters within 5 seconds. The valve is opened for 2.5 seconds at 5-minute intervals until the deviation is corrected. The heater remains on until the deviation is corrected.	The system enabled the valve and the heaters within 5 seconds. The valve is opened for 2.5 seconds at 5-minute intervals until the deviation is corrected. The heater remains on until the deviation is corrected.

8.5 Architectural Model

Presentation Layer (User Interface) → Handles crop selection, LED control, warning alerts and data display via a graphical user interface (GUI)
Application Layer (Logic Processing) → Implements sensor data and Kaggle data comparison,

automation actuators, and database interactions
Data Layer (Storage & Management) → Stores sensor readings and Kaggle crop parameters in MongoDB COMP3901 database greenhouse and crops collections, respectively

8.6 Code Flowchart



<u>Figure 4 shows the flowchart for the code.</u>

DELIVERABLES

9.1 Hardware Deliverables

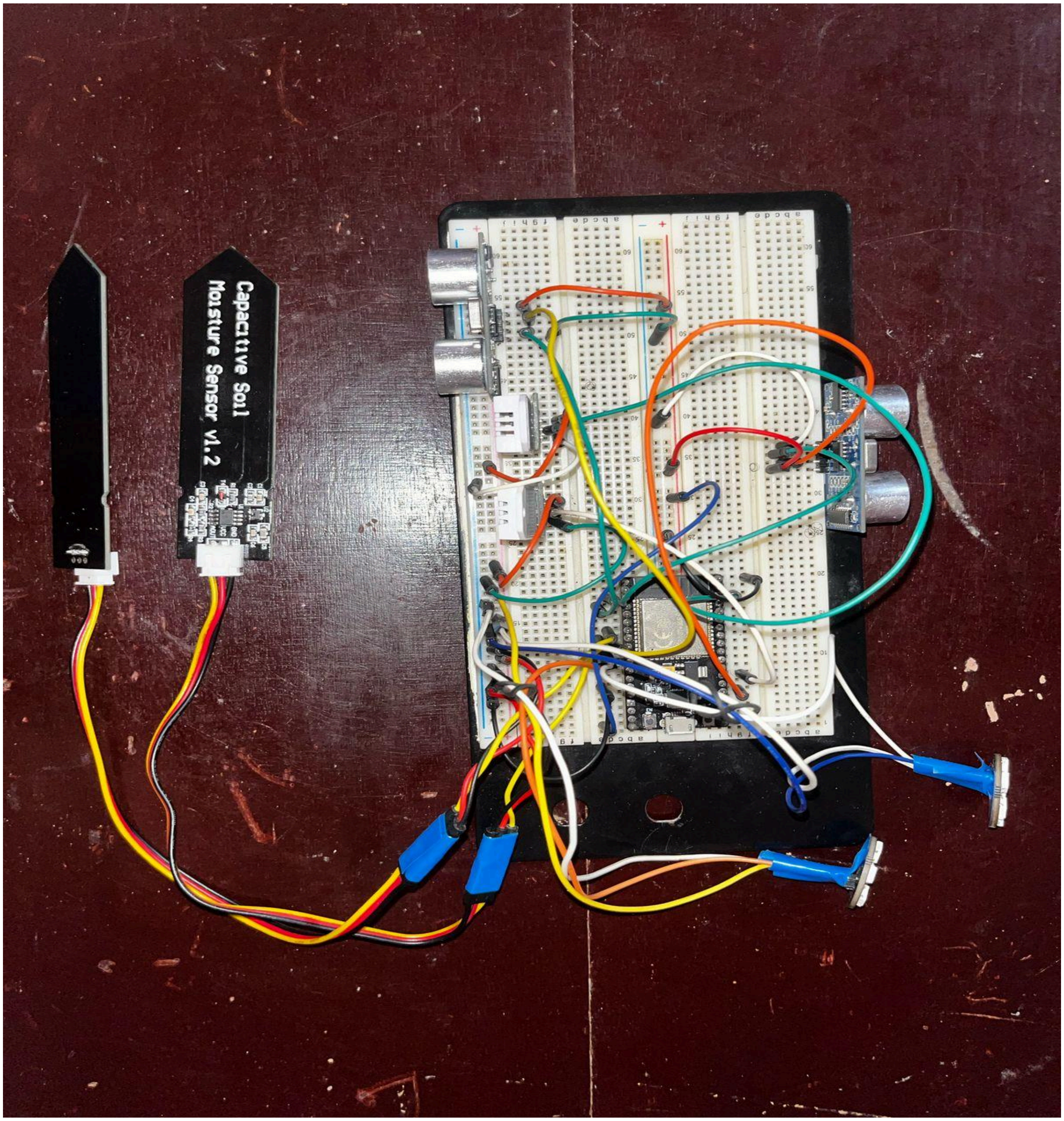


Figure 5 showing hardware prototype of ESP32 connected to sensors and RGB LEDs

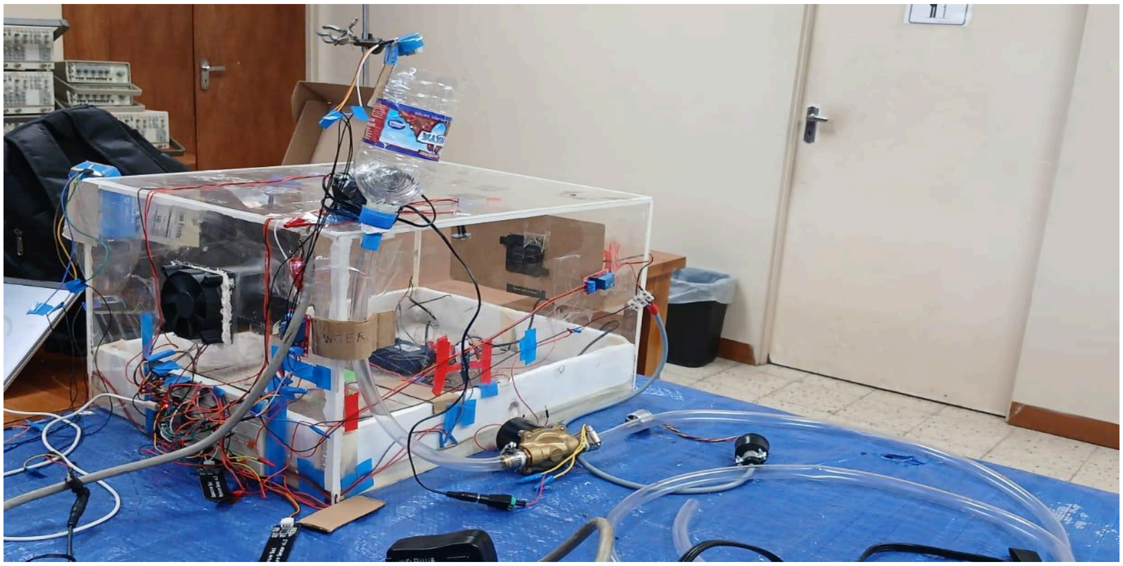


Figure 6 showing front view of hardware prototype of ESP32 connected to sensors and actuators

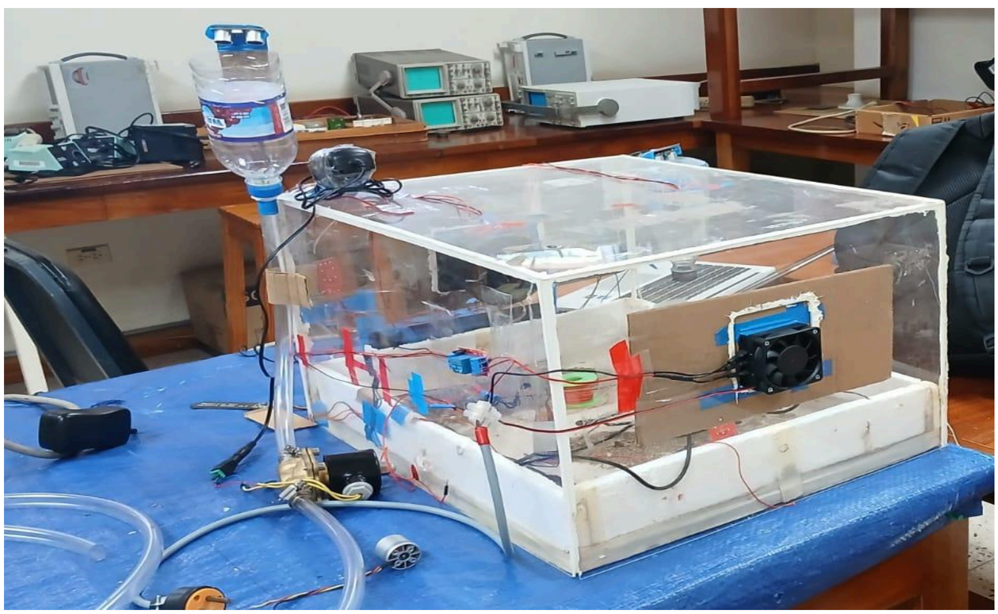


Figure 7 showing back view of hardware prototype of ESP32 connected to sensors and actuators

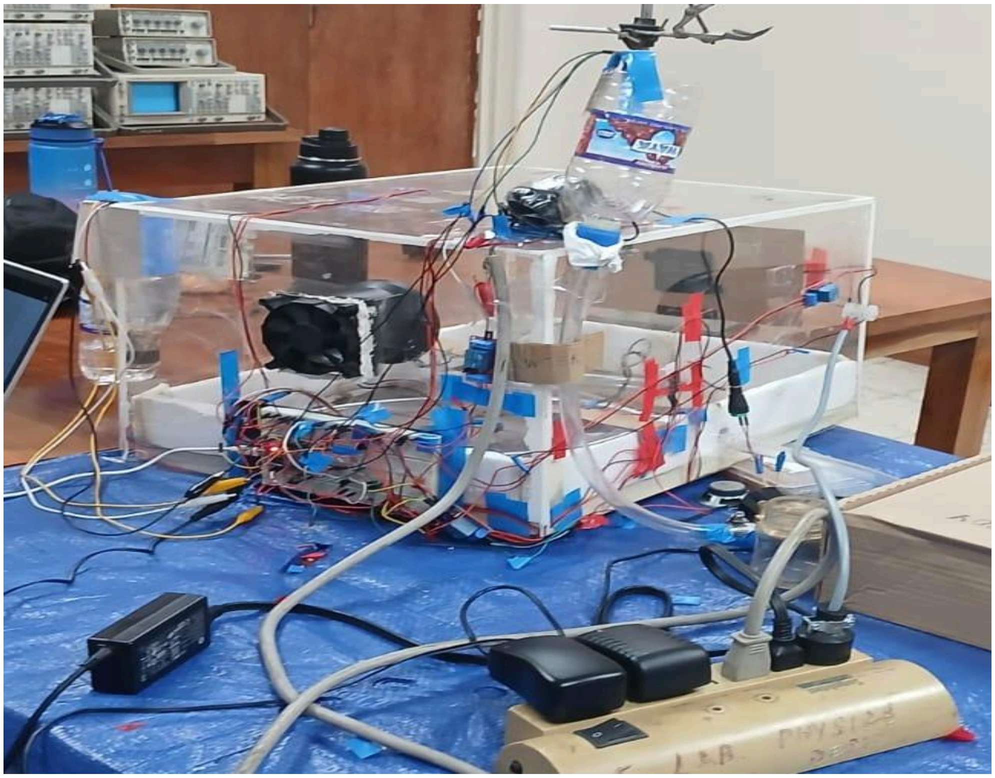


Figure 8 showing final hardware setup of ESP32 connected sensors and actuators

9.2 Software Deliverables:

- ➢ Backend: Flask API (routes), MQTT (broker), Pymongo (database)
  Data is able to be stored in the database and accessed by the frontend

- ➢ Frontend: Vuejs, Vuetify – HTML, CSS and JavaScript, Home Webpage (introduces the system), Live Webpage (displays live data being collected and shows the current status of the actuators such as the fans, heaters, pumps, etc), Analysis Webpage (data collected by sensors can be analyzed over time for patterns and possible predictions), Control Webpage (to control the colour and intensity of the lights, select the crops that the system should regulate the environment for and anything else we may want the farmer to control),

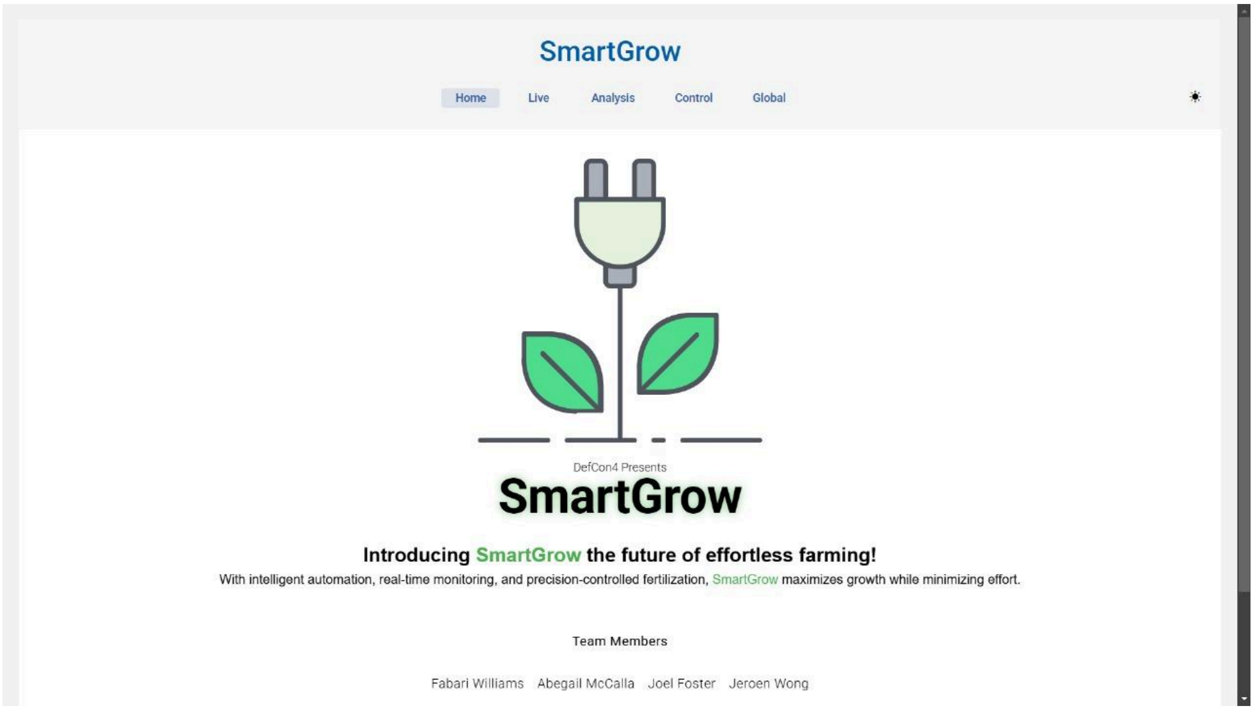


Figure 9 showing home webpage

Figure 10 showing live webpage

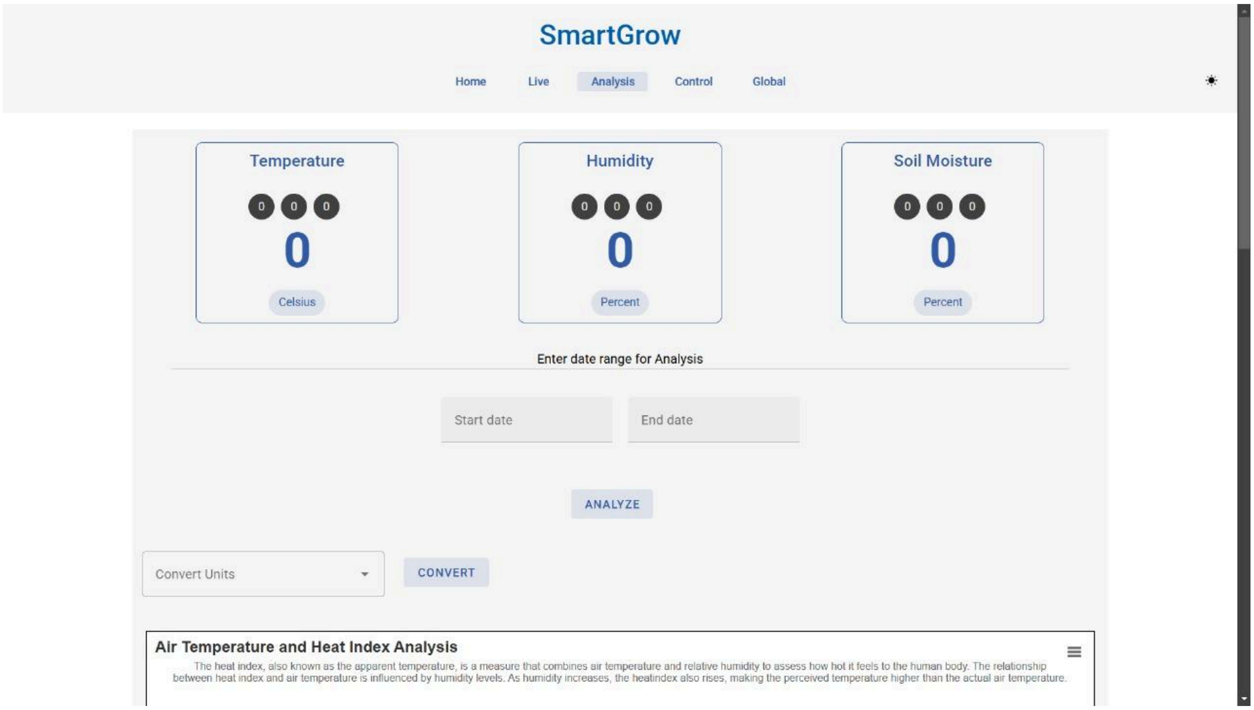


Figure 11 showing analysis webpage

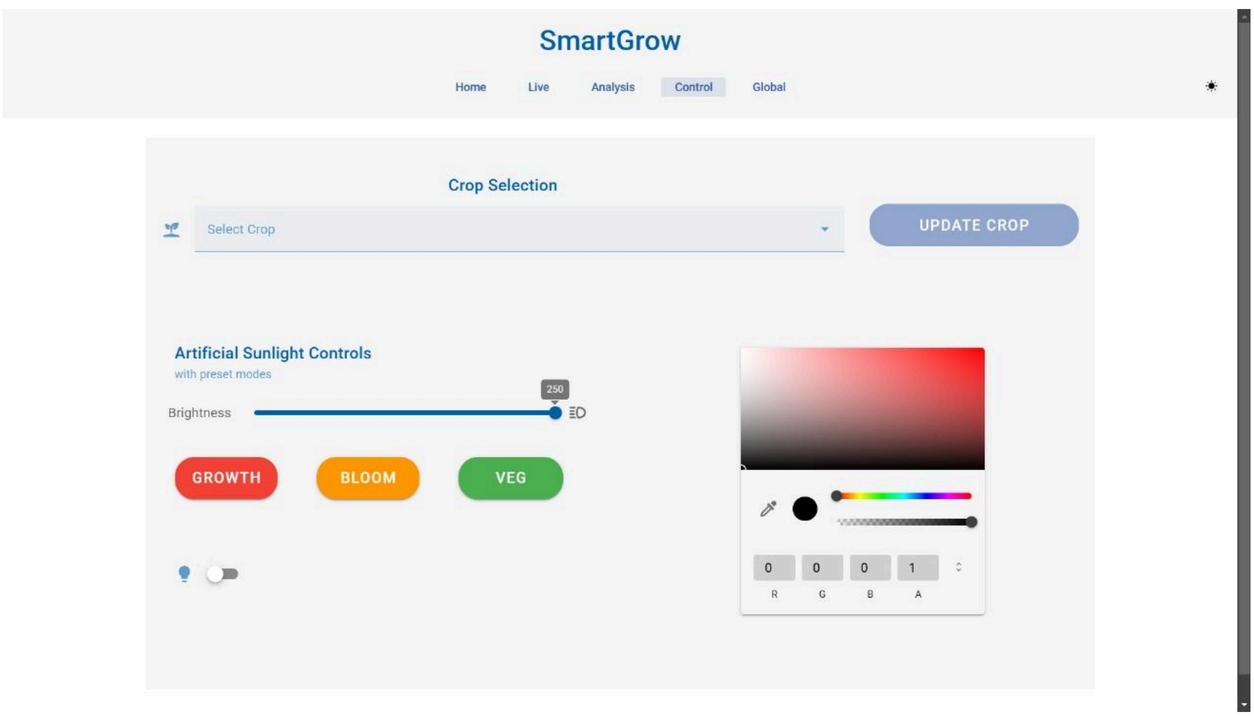


Figure 12 showing control webpage

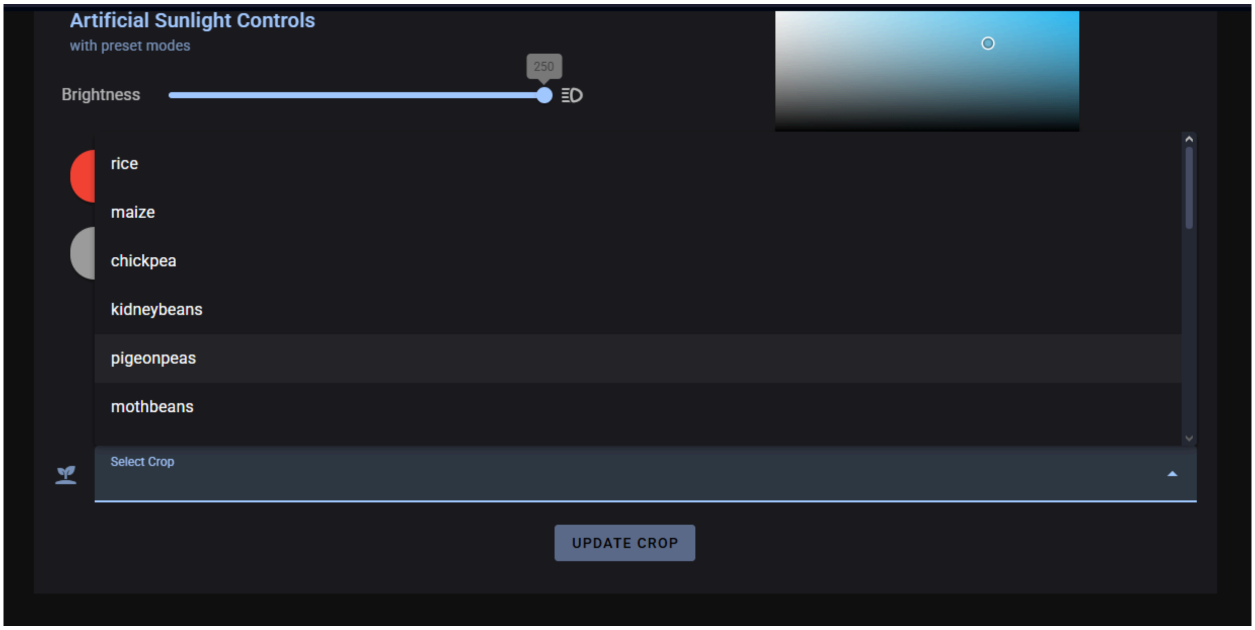


Figure 13 showing control webpage with available crops menu

Figure 9 introduces the IoT-powered smart greenhouse, SmartGrow, and its developers. Figure 10 shows a live webpage, which is used to display the live data obtained by the sensors as well as the current status of the actuators. The live webpage also displays graphs such as a real-time temperature and heat index analysis spline graph, which visually represents live temperature fluctuations and their impact on perceived heat levels, and a humidity graph, which visually represents the live humidity fluctuations read by the sensor. Figure 11 shows an analysis webpage, which is used to display the minimum, range difference between the maximum and minimum values, maximum and average values for the temperature, humidity and soil moisture being monitored for the selected crop over a specified period of time.

The analysis webpage also displays graphs comparing the relationship between certain parameters, such as an air temperature and heat index line graph, which shows how humidity influences the relationship between heat index and air temperature, a frequency distribution bar chart, which reveals how often different values occurred within the specified timeframe, highlighting seasonal trends, extreme conditions, and correlations that impact climate patterns, agricultural decisions, and environmental monitoring, and a temperature and heat index correlation scatter graph, which visualizes the relationship between temperature and heat index as well as revealing patterns or trends in the data. The farmer can use this analysis to make future predictions, such as the amount of water and fertilizer a crop will need over its lifetime so that it can be stored.

Figure 12 displays a control webpage that the farmer can use to adjust the light brightness, change the color of the light and select their desired crop for which the system will optimize the greenhouse environment. Figure 13 displays the available crops that the farmer can select from on the control webpage.

➢ Database: MongoDB (Cloud platform for data storage and analysis)

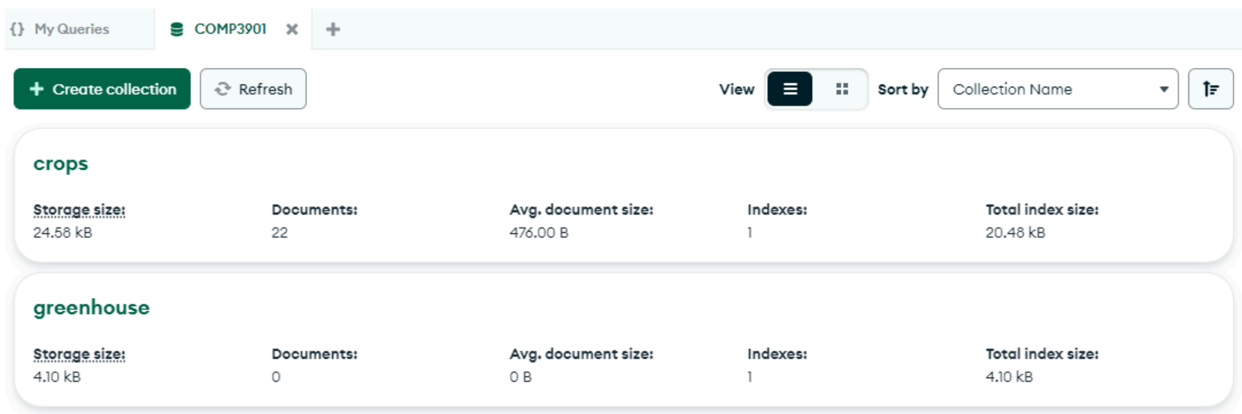


Figure 14 showing the COMP3901 database on MongoDB

The crops collection is used for storing the optimal environmental conditions for growing the crops mentioned in the problem definition and the greenhouse collection is used for storing data obtained by the sensors.

9.3 Additional capabilities

➢ Global Webpage (Allows user to enter a location then displays the current weather condition of that location)

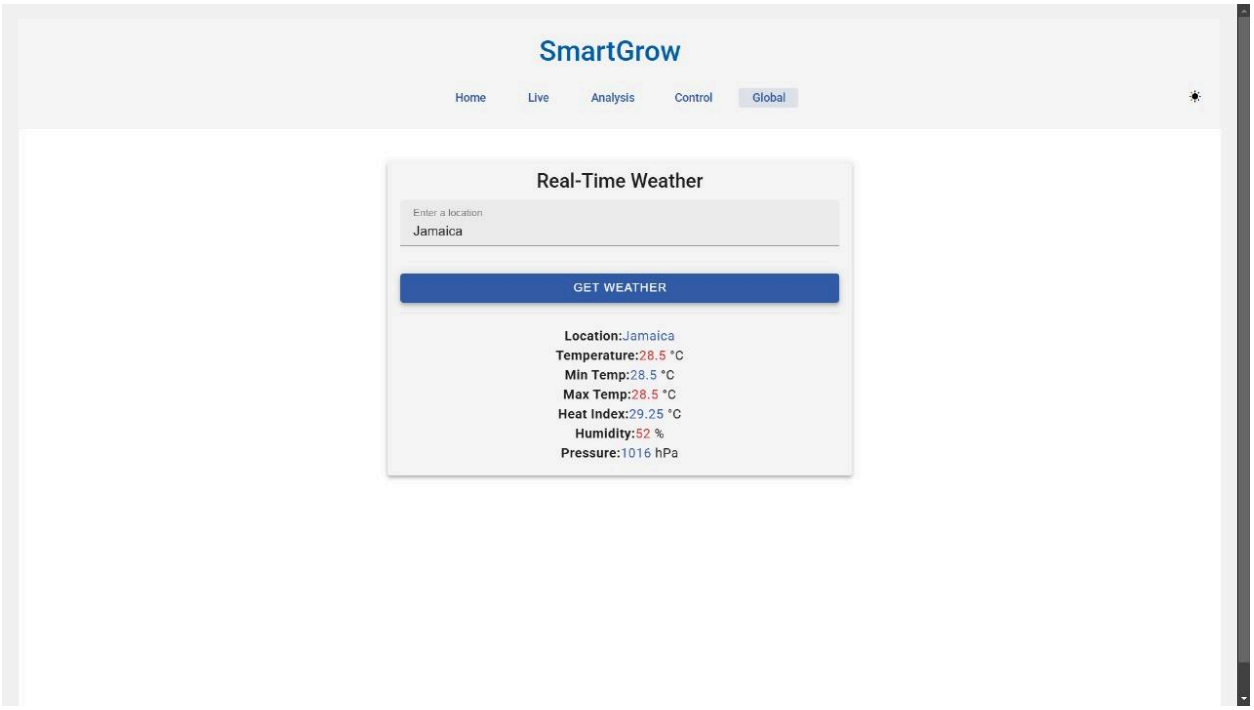


Figure 15 showing global webpage

The global webpage has the added feature of allowing the user to get the atmospheric condition of any location they enter.

RESULTS AND BENEFITS

All planned objectives, features, and deliverables outlined in the project's scope were successfully completed. The system meets its intended goals without major gaps or unresolved functionalities. The smart greenhouse system fully integrates drip irrigation, temperature monitoring and regulation, optimized LED lighting, and real-time water/fertilizer level tracking as intended and initially defined in the project scope. The implementation of IoT sensors, automated actuators, database storage, and user-controlled lighting ensured precise environmental regulation, improving agricultural efficiency and sustainability. Every core functionality, such as LED state setting, crop selection, sensor data collection, automation of actuators, soil moisture adjustment, temperature adjustment, resource monitoring, real-time alerts, and user interface controls, was implemented and operates according to the project requirements and predefined specifications. Essentially, no major planned feature was left incomplete, making the project a full realization of its original design. The completion of this project demonstrates the feasibility of using data-driven automation in modern greenhouse management, providing valuable insights for future expansions and enhancements.

As previously stated, all major planned features were successfully implemented, and the project's objectives were fully achieved without deviations from the proposed methodology. The identified agricultural challenges were effectively addressed, which means that all of the proposed problems were solved, and the system functions as intended, demonstrating its ability to optimize crop growth and resource management. The only minor setback encountered was the failure of three LEDs on one of the RGB LED strips. However, since two LED strips were incorporated into the design, this issue did not impact the proof of concept, and the system continues to operate effectively within its intended scope.

The implemented smart greenhouse system effectively addresses the challenges outlined in the problem definition by integrating optimized lighting, automated sensing, automated actuators, resource management, and alert systems. Each key benefit proposed in the solution was realized through the system's functionalities:

- ➢ Optimized Irrigation: The drip irrigation mechanism, activated based on data input from the soil moisture sensors, ensured precise watering, preventing underwatering, which can damage roots and waste resources.
- ➢ Temperature Regulation: Automated heaters and fans, triggered by real-time temperature monitoring, maintained optimal climate conditions, preventing stress-related plant damage.
- ➢ Optimized Lighting for Photosynthesis: RGB LED spectral tuning allowed farmers to select appropriate wavelengths, enhancing photosynthesis efficiency when natural light was insufficient.
- ➢ Automated Adjustments & Decision-Support: The system compared live sensor data with predefined optimal crop conditions, triggering actuators to correct deviations in environmental parameters dynamically.
- ➢ Efficient Resource Monitoring: The system continuously tracked tank water and fertilizer levels, issuing alerts at critical thresholds (10% capacity) to prevent shortages or excessive application.

The solution fulfilled the proposed benefits, solving the identified problems without deviation from the intended methodology. While minor hardware issues (such as a few damaged LEDs) were encountered, they did not compromise the proof of concept. The system successfully optimized irrigation, climate regulation, and resource efficiency, demonstrating its value for modern smart greenhouse farming.

CHALLENGES AND TECHNICAL ISSUES

- ➢ Acquiring the sensors and actuators: Save the money to purchase the sensors and actuators on Amazon
- ➢ Correctly connecting the sensors and actuators to the ESP32 microcontroller: Review the datasheets and watching youtube videos on how to connect the sensors and actuators to the microcontroller
- ➢ Writing code for the ESP32 to correctly interface with each electronic component: watch youtube videos and do independent research
- ➢ Writing code to send data to the database and display it on the webpages: watch youtube videos and do independent research

FUTURE IMPROVEMENTS

For future improvements, the system could be optimized to use the stored data in the MongoDB COMP3901 database greenhouse collection to make predictions on the amount of water and fertilizer the farmer might need to store to grow certain crops and allow them to thrive over their lifetimes. The system could be enhanced by interfacing it with soil pH and npk sensors as well as suitable actuators to ensure that these parameters remain within their respective allowable deviation ranges. The system could also be enhanced by implementing an automated water and fertilizer distribution and mixing system to regulate soil pH along with improved error handling to alert farmers to specific problems as they occur, such as deviations beyond allowable ranges. The system can also be updated to achieve an optimization balance for soil npk, soil pH and soil moisture by ensuring that each factor remains within its allowable range, even though these parameters are interconnected.

A drip irrigation pump and emitter can also be added to the system to improve efficacy. A TFT SPI display could be added inside the greenhouse to display the live environmental conditions inside the greenhouse. Some sensors may have minor deviations, requiring better calibration or alternative models for improved precision. The heaters, fans, and LED systems could be further optimized to reduce power consumption while maintaining effectiveness. The comparison between real-time data and stored crop requirements may be improved by implementing indexed queries or optimized data structures. Currently, the system supports monitoring only one crop at a time, but future iterations could explore simultaneous multi-crop tracking with advanced logic. Enhancing webpage readability, responsiveness, and ease of navigation to improve user experience. For scalability, a larger, real-world greenhouse would require industrial-grade sensors and actuators to ensure precision, reliability, and automation at scale.

CONCLUSION AND PROJECT REPOSITORIES

The proposed IoT-based technological approach to farming (SmartGrow) offers a powerful, efficient solution for maintaining controlled environments. Utilizing real-time sensor data, suitable actuators and alerts will ensure optimal conditions, improve resource efficiency, and reduce costs, leading to increased productivity and prolonged asset viability. This solution will be indispensable for the agricultural industry, which is reliant on tightly controlled atmospheric conditions.

The SmartGrow system was successfully developed and implemented, effectively addressing key challenges in traditional greenhouse management through automated sensing and real-time data processing. It eliminated underwatering issues by optimizing irrigation cycles based on soil moisture levels, maintained stable temperature control through sensor-driven climate adjustments, and fine-tuned RGB LED lighting for enhanced photosynthesis efficiency. Continuous monitoring of water and fertilizer tank levels prevented shortages, ensuring uninterrupted resource availability. Through this process, the team discovered that AI integration could further enhance decision-making, improving predictive automation and adaptive system responses. Additionally, challenges like crop rotation inefficiencies were recognized, with drip irrigation identified as a solution to optimize water distribution, minimize waste, and improve nutrient absorption for sustainable agricultural cycles.

Below is a link to the github repository
SmartGrow’s Code Repository: https://github.com/Fabari1/smartfarming.git
SmartGrow’s Marketing Repository: https://github.com/fizzygad/COMP3901Website.git

REFERENCES

Akpenpuun, T. D., Ogunlowo, Q. O., Na, W.-H., Rabia, A., Adesanya, M. A., Dutta, P., Zakir, E., Ogundele, O. M., Man, H. T., & Lee, H.-W. (2023). Review of temperature management strategies and techniques in the greenhouse microenvironment. Adeleke University Journal of Engineering and Technology, 6(2), 126-147. https://www.aujet.adelekeuniversity.edu.ng

Boyalite, Manufacturer of High Quality LED Grow Lights. (2025, January 9). What is the optimal light spectrum for plant growth. https://boyagrowlight.com/the-optimal-light-spectrum-for-plant-growth/

Liu, J., & van Iersel, M. W. (2021). Photosynthetic physiology of blue, green, and red light: Light intensity effects and underlying mechanisms. Frontiers in Plant Science, 12, Article 619987. https://doi.org/10.3389/fpls.2021.619987

Moulton, A. A., Popke, J., Curtis, S., Gamble, D. W., & Poore, S. (2015). Water management strategies and climate adaptation: Lessons from the 2014 drought in Jamaica. Caribbean Geography, 20.

Olle, M., & Alsina, I. (2019). Influence of wavelength of light on growth, yield and nutritional quality of greenhouse vegetables. Proceedings of the Latvian Academy of Sciences. Section B, 73(1), 1-9. https://doi.org/10.2478/prolas-2019-0001

Qwabe, N. (2024). Good and bad effects of climate change on agricultural production. Farmer's Weekly. https://www.farmersweekly.co.za/agri-technology/farming-for-tomorrow/good-and-bad-effects-of-climate-change-on-agricultural-production/

Sil, P., Chhetri, P., Majumder, S., & Santosh, D. T. (2024). Modern agronomy. Centurion University of Technology and Management, Odisha.

St. Martin, C. C. G., Bedasie, S., Ganpat, W. G., Orrigio, S., Isaac, W. A. P., & Brathwaite, R. A. I. (2009). Greenhouse technology is once again washing the Caribbean—Can we ride the wave this time around? The University of the West Indies, St. Augustine, Trinidad and Tobago.

Suresh, N., Hashiyana, V., Kulula, V. P., & Thotappa, S. (2019). Smart water level monitoring system for farmers. In Title of the Book (Chapter 14). University of Namibia; JSS Science and Technology University, India.