Boston University
Electrical & Computer Engineering

EC 463 Senior Design Project

First Semester Report

by
Team 26

Dilhara DeSilva dilharad@bu.edu
Arthur Hua ahua102@bu.edu
Alex Muntean munteana@bu.edu
Jared Solis jared11@bu.edu
Sourav Shib shib0826@bu.edu

Submitted: 12/8/2024

Table of Contents

Executive Summary - Alex

EC463 First Semester Report

26 - Simple Sprouts

The challenges of urban food production include limited space, labor-intensive maintenance, and dependency on external energy sources, which restrict traditional and modern farming solutions. Simple Sprouts addresses these issues by providing an autonomous, scalable, and sustainable vertical farming system designed for urban environments, schools, and restaurants.

Our final deliverable is a fully integrated farming solution that includes a modular planter system, renewable energy-powered operations, automated plant care, and a user-friendly mobile application. The system enables users to grow plants efficiently with minimal effort, making sustainable farming accessible in diverse settings.

The project employs a technical approach that combines hardware, software, and renewable energy solutions. Key features include a solar-powered 12V battery system, an ESP32 microcontroller managing automated plant care through real-time sensor data, and an advanced image recognition module for monitoring plant health and ripeness. These components are seamlessly integrated with a mobile application that provides real-time control and monitoring.

Simple Sprouts stands out with its innovative integration of renewable energy, autonomous functionality, and advanced monitoring. The solar-powered system reduces reliance on external energy sources, while automation eliminates the need for constant user intervention. Image recognition and real-time data visualization enhance the user experience, making it a practical solution for individuals and institutions.

This project demonstrates a forward-thinking approach to modern urban farming, blending sustainability, technology, and user convenience to address the growing demand for efficient food production.

1.0 Introduction - Dilhara

Efficient food production has become an increasingly significant challenge, particularly in urban environments, institutions, and indoor settings where space is limited, and access to natural sunlight is restricted. Traditional farming methods are labor-intensive, resource-demanding, and unsuitable for environments such as school campuses, restaurants, and urban communities. These limitations highlight the urgent need for a sustainable, autonomous farming solution that optimizes space and resource usage while minimizing user effort.

Our project addresses this problem by developing Simple Sprouts, an autonomous vertical farming system that combines smart technologies with sustainable design principles. The system integrates automated grow lights, image recognition for plant health monitoring, and a web-based application for real-time data visualization. By offering convenience and flexibility, it provides an efficient, low-maintenance solution tailored for individuals and institutions seeking innovative agricultural options.

The challenges faced by institutions such as schools and restaurants further underline the importance of such a system. Limited space, no access to direct sunlight, and the labor-intensive maintenance of plant growth create barriers that traditional farming methods cannot overcome. Even existing vertical farming systems are often costly or require significant user intervention, making them impractical for widespread use. Simple Sprouts offers a smarter, automated alternative to address these shortcomings.

To bridge the gap between sustainable farming and user convenience, Simple Sprouts integrates automation and real-time monitoring. The system maximizes the core benefits of vertical farming—efficient space usage, reduced water consumption, and high crop yields—while minimizing user interaction. Its goal is to make farming accessible even in the most challenging environments, empowering users to grow plants sustainably without requiring specialized expertise.

The design of Simple Sprouts incorporates an ESP32 microcontroller, which manages grow lights, water pumps, and soil moisture sensors. This system ensures plants receive the optimal amount of water by automating irrigation processes, while a camera integrated with an image recognition model identifies plant types, assesses ripeness, and monitors overall health. The result is a farming system that can make automated decisions for plant care with minimal human oversight.

To further enhance usability, the system includes a user-friendly web application that visualizes real-time data, such as soil moisture levels, light hours, and water usage. Optional solar panels make the system even more sustainable by reducing reliance on external energy sources, enhancing its applicability in off-grid scenarios.

By automating farming processes and integrating renewable energy, Simple Sprouts eliminates the barriers traditionally associated with agriculture. Automation significantly reduces labor requirements, while real-time monitoring ensures users can track and adjust plant care remotely. The incorporation of solar panels not only reduces the environmental footprint but also makes the system cost-efficient and adaptable to diverse use cases, from urban apartments to institutional kitchens.

The system boasts several innovative features that highlight its potential. Autonomous grow light control dynamically replicates sunlight cycles, ensuring optimal growth conditions even in shaded or indoor environments. A camera-driven image recognition model identifies plant types, ripeness, and health, automating decisions like harvesting and maintenance. Real-time data visualization provides insights for optimization, while the modular design ensures scalability and flexibility, making the system adaptable to a variety of applications.

Through its seamless integration of advanced technologies like image recognition and IoT-based automation with sustainable farming practices, Simple Sprouts meets the diverse needs of urban and indoor farming while minimizing environmental impact. Its innovative features and user-friendly design represent a scalable solution that aligns with the growing demand for sustainable and accessible food production methods. Simple Sprouts demonstrates the potential to revolutionize small-scale farming by making it more efficient, sustainable, and practical for users in modern society.

2.0 Concept Development - Sourav

The goal of the Simple Sprouts project is to address the challenges of growing plants in limited spaces while minimizing the expertise and manual labor required for traditional farming. This system must serve both small-scale users, such as individuals in urban homes, and larger-scale users, such as restaurants and school campuses. Customers need a solution that is scalable, user-friendly, and capable of taking care of various plants with minimal intervention.

The primary engineering challenges include designing a modular system that can accommodate different user situations, automating plant care tasks such as watering and lighting based on real-time sensor data, and providing accurate metrics such as plant health, soil quality, and environmental conditions. Additionally, the system must integrate with a cloud-based platform to allow for remote monitoring and control through an intuitive interface.

The engineering requirements for Simple Sprouts reflect the customer's needs for automation, data integration, flexibility, and user accessibility. The system must be modular to allow users to customize the size of the setup, making it suitable for both small and large applications. Automation to reduce user input by using sensors to track environmental conditions and control actuators for watering and lighting. Data integration via Firebase ensures that real-time sensor readings and plant metrics are accessible remotely. The system must also be able to adapt to different plant types based on user input, and the interface should be simple enough for anyone to use, regardless of their gardening experience.

The Simple Sprouts system combines hardware, software, and cloud technologies to create an automated vertical farming solution. It integrates environmental sensors, microcontrollers, and cloud communication to monitor and manage plant health. The current system uses soil sensors to measure moisture levels, temperature, and humidity sensors to maintain optimal growing conditions and an air quality sensor to track CO2 levels and volatile organic compounds. In the future, additional sensors such as a water level sensor for the tank and a nitrogen sensor for soil health will enhance functionality. All sensors are sourced from Adafruit, ensuring reliability and compatibility.

The microcontroller platform is currently an ESP32, which handles all sensor inputs and controls actuators for watering and lighting. A Raspberry Pi is used to run a local Llama 3 model that analyzes images of the plants, providing metrics such as plant health and ripeness. The system has plans to transition to a single microcontroller for all functions and to place stationary electronics on a custom PCB. This approach simplifies the architecture.

Firebase serves as the system's cloud integration platform. It is a communication hub where sensor readings, plant health stats, and user inputs are stored and exchanged. The ESP32 sends real-time environmental data to Firebase, while the Raspberry Pi adds analysis from the image. Users interact with the system via a React Native-based web app, where they can view sensor readings, schedule watering and lighting, and receive alerts. For instance, when the planned water level sensor is added, data will flow from the ESP32 to Firebase and then to the web app, notifying users when the tank requires refilling.

The web app, currently under development, is designed to make the system highly user-friendly. It displays essential metrics such as plant health, harvest schedules, water usage, power consumption, and environmental conditions. By reducing the need for user input, the system allows even inexperienced users to start gardening.

Other approaches were evaluated during the concept development phase. Hydroponics-only systems, while not fully rejected, were not chosen as standalone soil options because of potential high cost. Instead, a hybrid approach incorporates hydroponic mixtures in the soil to enhance plant growth. A fully manual vertical farming system was also considered but discarded because it would require significant user expertise and time, which conflicts with the project's goal of automation. The current system balances automation and ease of use.

The chosen design for Simple Sprouts addresses customer needs and overcomes key engineering challenges. By using widely available sensors from Adafruit, the system ensures reliability and scalability. Cloud integration via Firebase allows for a communication network, enabling interactions between the hardware, software, and user interface. The modular design allows for scalability for diverse user needs, from individual households to restaurants. With plans for future enhancements, such as additional sensors and a PCB, Simple Sprouts offers a comprehensive solution for efficient and accessible vertical farming. By automating tasks and providing detailed metrics, the system reduces user effort while enhancing plant growth, making it an innovative approach to modern urban farming.

3.0 System Description - Jared

Our solution that we will provide is two-fold: a physical planter box that can be stacked vertically and holds our sensors and actuators as well as a web-app in React Native Expo that will allow the user to control the various control modes the user has access to. The default mode is a Scheduling Mode in which the user is able to schedule the amount of time in hours between turning on and off the watering and lighting as well as the amount of water in milliliters and duration of light in hours. The other mode currently being used is a Manual Mode in which the user is able to click a button to actuate the lighting on/off as well as a field to write the amount of water in milliliters to be added to the planter box.

The final mode to be added is an Adaptive Mode in which the system uses the soil moisture sensor in order to determine how much water is needed based on the optimal soil moisture range per plant. This can change per plant due to the different types but for vegetables the acceptable range is 41-80%. Because the soil moisture sensors measure in values of 200-2000 of capacitance, we can map this to the following percent and keep it within that range. These modes allow users to control the care of these plants exactly how they want it, whether it be through preset scheduling, real-time actuating, or purely through sensor feedback.

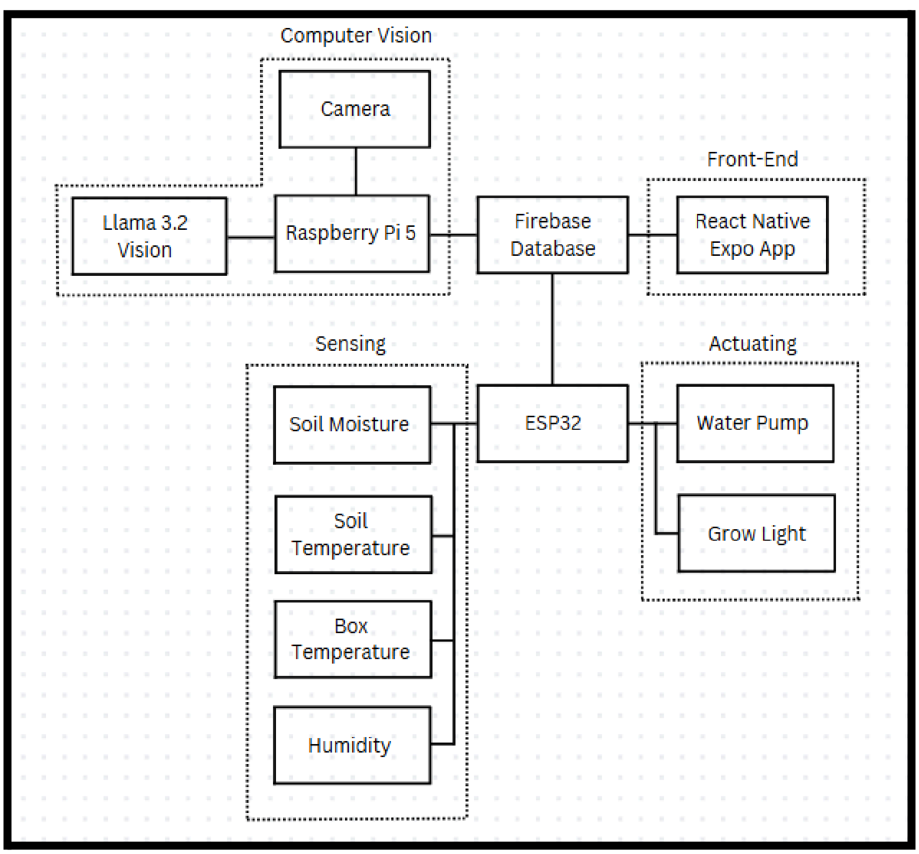


Figure 1: High Level Block Diagram of System

As seen in Figure 1, these modes are controlled by several modules, which we will call Sensing, Computer Vision, Front-End, and Actuating. The sensing and actuating are done on the physical planter box, in which an ESP32 controls our water sensor, box temperature, soil moisture sensor, humidity sensor as well as our water pump, grow light, and dehumidifier/fans. These sensors allow us to use the adaptive mode that will be added for the ease of the user. The front-end is purely in software, running through React Native Expo, and the computer vision is within both realms, using the physical camera and interfacing with a Raspberry Pi 5 which will run a computer vision model (i.e. Microsoft Phi 3.5) to identify if the plant is ripe to be picked as well as aspects of plant health including discoloration, wilting of leaves, or stunted/delayed growth.

These modules will communicate using React Native Firebase, in which the separate controllers (ESP32, RPi 5, and React Native Expo) will read and write to several access points. As seen in Figure 6 of the Appendix, an example means of communication is where the system is in Manual Mode, the web-app will write True to the cmd_to_esp flag, and because the ESP32 is constantly querying it, it will then control the actuators as necessary by starting the water pump or grow lights.

For the web-app that will provide the user almost hands-free interaction with the system, the user will first be at the home page in which they can select the plant that they are caring for. Each plant will be marked with an ID on Firebase in order to identify the different scheduling. Initially, the mode being used will be preset scheduling. On the next page, the user will be able to switch between modes to fully control the actuation of the system as well as see the current scheduling and time till next activation of watering/lighting. When out of the app, the Raspberry Pi 5 will run the computer vision model on a daily image to let the user know feedback on the plant health qualities mentioned earlier. The user can also navigate to a page in which they can see this daily image and current sensor information that is queried every minute to update with current information. These will allow the user to make accurate decisions on which modes are working best for their plant.

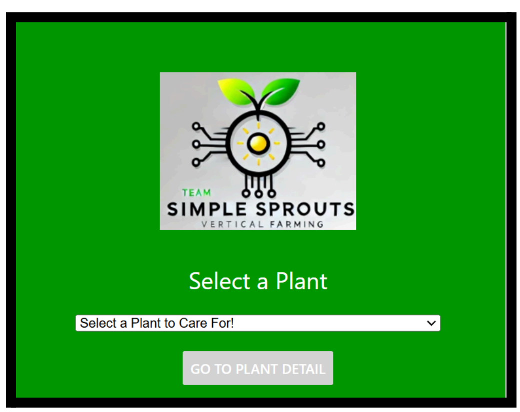


Figure 2: Initial Sketch of Home Page

4.0 First Semester Progress - Dilhara

This semester, the team made substantial advancements in the development and testing of the Simple Sprouts prototype, focusing on integrating core functionalities and validating system components through rigorous testing. These efforts culminated in several key accomplishments that highlight the viability of our design.

The ESP32 microcontroller successfully controlled the grow lights and water pump, demonstrating reliable on/off operation during manual testing. However, the lack of scheduling functionality revealed a critical limitation, underscoring the need for automation to reduce user intervention and enhance long-term reliability. This insight has given us the next steps for improving the system's autonomy.

Data integration across the system was seamless. The soil moisture sensors, temperature and humidity sensors, and the MOX air quality sensor provided real-time readings that were accurately displayed on the web application. These sensors enabled users to monitor key environmental metrics remotely, showcasing the system's capability for comprehensive monitoring. While certain cumulative statistics were not yet functional, the successful integration and display of sensor data validate the system's potential as a reliable platform for autonomous operation and environmental data tracking.

The camera and image processing components performed well, providing accurate assessments of plant type, ripeness, and health. Coupled with chatbot functionality, this feature offered actionable insights in an interactive format. While the image processing currently relies on a teammate's laptop for computational tasks, this temporary solution successfully demonstrated the potential of this component. Future iterations will focus on transitioning this functionality to an onboard solution for improved scalability.

Overall, the system functioned reliably, with all sensors and actuators meeting expected performance metrics, as shown by the table below reviewing our prototype test results. The prototype represents a critical step towards achieving a fully autonomous vertical farming system. Addressing the scheduling limitations and advancing image processing capabilities will be priorities for future development to ensure enhanced autonomy and scalability in subsequent iterations. These results mark significant progress toward realizing the project's goals.

Test Case	Expected Outcome	Pass/Fail
Light Control	On/off function	✓
Water Pump Activation	On/off function	✓
Image Capture	Successful image capture	✓
Ripeness Detection	Accurate ripeness assessment (≥ 80%)	✓
Sensor Data Display	Real-time, accurate data on web app	✓

Figure 3: Score Sheet used in Prototype Testing

5.0 Technical Plan - Alex

Task 1. Port App for Mobile Use

The current web application shall be adapted to mobile platforms to improve user accessibility and compatibility with either one of Android and iOS devices. The app will allow users to control system functionalities such as scheduling watering and lighting while monitoring real-time sensor data. It shall be tested for responsiveness, performance, and synchronization with the Firebase database to ensure reliability and ease of use.

- **Start:** December 10, 2024 / January 8, 2025 (Research / Development)
- **End:** January 25, 2025 / February 16, 2025 (Research / Development)
- **Lead:** Jared Solis, Arthur Hua
- **Assisting:** Sourav Shib
- **Deliverable:** Mobile application compatible with either Android or iOS.

Task 2. Determine Power Draw

The power requirements of all system components, including sensors, actuators, and microcontrollers, shall be measured under typical and peak operating conditions. The analysis shall ensure components meet energy efficiency standards and align with the system's overall power budget. Testing will involve recording and validating power consumption data to identify potential optimizations.

- **Start:** January 14, 2025
- **End:** January 21, 2025
- **Lead:** Dilhara DeSilva, Alex Muntean
- **Assisting:** Jared Solis
- **Deliverable:** Documented power draw analysis.

Task 3. Build Modular Planter Structure

The physical planter system shall be fabricated and assembled to support vertical farming requirements, prioritizing modularity for scalability and durability to withstand long-term use. The structure will be designed to accommodate sensors, actuators, and water flow while ensuring easy access for maintenance and user customization. Testing will include structural integrity under load and compatibility with all integrated components. The material used shall be determined at a later date.

- **Start:** January 21, 2025
- **End:** January 31, 2025
- **Lead:** Alex Muntean, Dilhara DeSilva
- **Assisting:** Sourav Shib, Arthur Hua, Jared Solis
- **Deliverable:** Completed modular planter structure prototype.

Task 4. Design and Test Housing for Components

The housing for the microcontroller, battery, sensors, and water pump shall be designed, fabricated, and tested to ensure proper protection against environmental factors such as moisture and dust. The housing will also ensure secure mounting and integration of components while maintaining accessibility for maintenance and adjustments. Testing will include evaluating durability, thermal dissipation, and compatibility with system functionality.

- **Start:** January 29, 2025
- **End:** February 2, 2025
- **Lead:** Alex Muntean
- **Assisting:** Dilhara DeSilva, Jared Solis
- **Deliverable:** Functional housing prototypes for system components.

Task 5. PCB/PLC Design and Integration

The system's control circuitry shall be evaluated to determine whether a custom PCB or a PLC is more suitable. If a PCB is selected, it shall be designed, fabricated, and tested to meet the requirements of modularity, reliability, and cost-effectiveness. If a PLC is chosen, it shall be configured and integrated into the system. Testing will ensure functionality, scalability, and seamless communication with other system components.

- **Start:** February 3, 2025
- **End:** February 26, 2025
- **Lead:** Alex Muntean, Dilhara DeSilva
- **Assisting:** Sourav Shib, Jared Solis
- **Deliverable:** Fully functional PCB or integrated PLC, tested and validated for the system.

Task 6. Battery Power Supply

A 12V battery with a minimum capacity of 24 amp hours shall be researched and purchased to meet the system's power requirements. It will include rechargeable capabilities via solar panels and meet specifications for weight, heat dissipation, and battery life. Testing will involve integrating the battery with the system and validating its compatibility with solar charging under typical operational conditions.

- **Start:** February 2, 2025
- **End:** February 16, 2025
- **Lead:** Dilhara DeSilva
- **Assisting:** Alex Muntean
- **Deliverable:** Fully operational and tested battery supply.

Task 7. Integrate Solar Panel

A solar panel system shall be researched, acquired, and integrated to provide renewable energy for recharging the 12V battery and powering the system. The panels shall be selected to ensure sufficient energy generation under various lighting conditions and compatibility with the battery's charging requirements. Testing will involve validating the solar panel's output, charging efficiency, and performance under real-world conditions.

- **Start:** February 2, 2025
- **End:** February 23, 2025
- **Lead:** Dilhara DeSilva, Alex Muntean
- **Assisting:** Jared Solis
- **Deliverable:** Integrated solar panel system capable of supporting system operations.

Task 8. Image Recognition for Plant Health

A camera system shall be developed and implemented to analyze plant health, ripeness, and type using image recognition technology. The system will integrate with the platform's existing hardware and software, providing real-time analysis and actionable insights for automated plant care. Testing will involve validating the accuracy of plant health assessments and ensuring seamless data transfer to the user interface.

- **Start:** February 15, 2025
- **End:** March 1, 2025
- **Lead:** Arthur Hua
- **Assisting:** —
- **Deliverable:** Functional image recognition system integrated into the Simple Sprouts platform.

Task 9. System Integration and Final Testing

All hardware and software components, including sensors, actuators, the battery, solar panel, camera system, and user interface, shall be integrated into a cohesive system. Comprehensive testing will be conducted to validate functionality, ensure seamless communication between components, and identify potential improvements. The testing process will simulate real-world conditions to confirm the system meets performance and reliability requirements.

- **Start:** March 1, 2025
- **End:** March 31, 2025
- **Lead:** Arthur Hua, Jared Solis, Sourav Shib
- **Assisting:** Alex Muntean, Dilhara DeSilva
- **Deliverable:** Fully operational Simple Sprouts system.

Task 10. Prepare and Deliver for Final Testing

The customer site shall be prepared for installation, ensuring compatibility with the system's requirements, including power, space, and environmental conditions. The fully assembled Simple Sprouts system will then be installed and tested to validate its functionality and performance under real-world conditions. Feedback from the customer will be collected and used to implement final adjustments

- **Start:** April 20, 2025
- **End:** May 1, 2025
- **Lead:** TBD at a later date
- **Assisting:** All Team Members
- **Deliverable:** Fully deployed system, ready for customer feedback and adjustments.

6.0 Budget Estimate - Arthur

Item	Description	Cost
1	Physical Plant Box	$24.98
2	Potting Mix and Rocks	$16.75
3	Water Pump	$24
4	Wires and Sensors	$50
5	Light Holder and Sockets	$11
6	Grow Lights	$10
7	Power Inverter	$20
8	SD Card	$13
9	Raspberry Pi 5	$60
	Total Cost	$229.73

Figure 4: Budget Estimate Chart

The overall bill for the physical component of the plant box itself is around $70 and we plan on making another one to put the vertical farming portion of our project into play. That would almost double our expenses due to basically having to remake the system, as well as buying frames so as to have 2 planter boxes on top of each other. We also want to buy an LCD display for our planter box to be connected to the Raspberry Pi 5 as an easy-to-use interface to monitor and also to control the plant in person.

7.0 *Attachments*

7.1 Appendix 1 - Engineering Requirements - Arthur

Team #26 - Simple Sprouts

EC453 First Semester Report

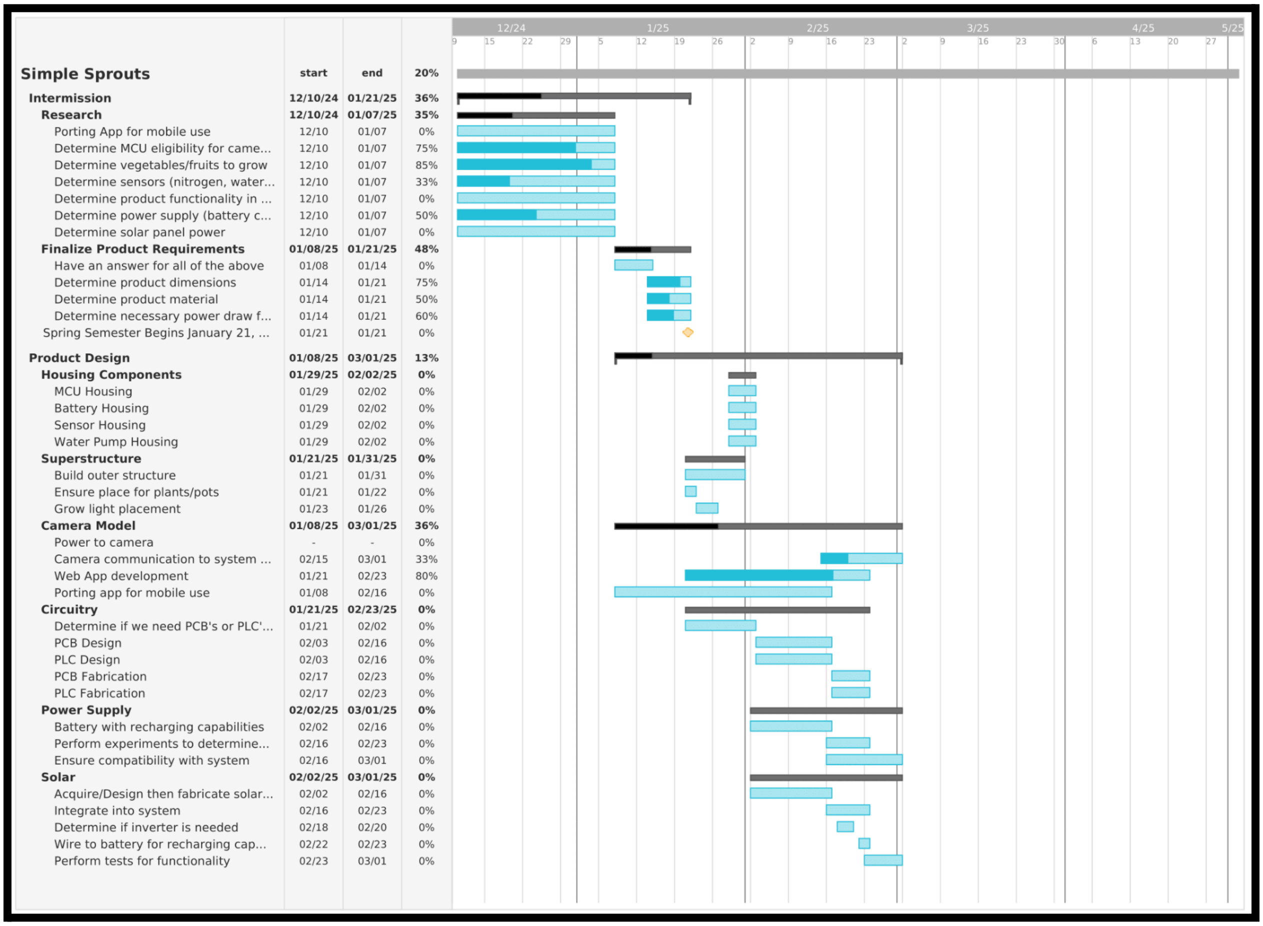
Requirement	Value/Unit/Range
Case Dimensions	1.2 m x 0.4 m x 0.8 m
Plants per Box	3
Plant type	Pepper
Sensor and ESP Electronics	5 V
Model Response Time	5-6 min
Water Tank Size	5 gal

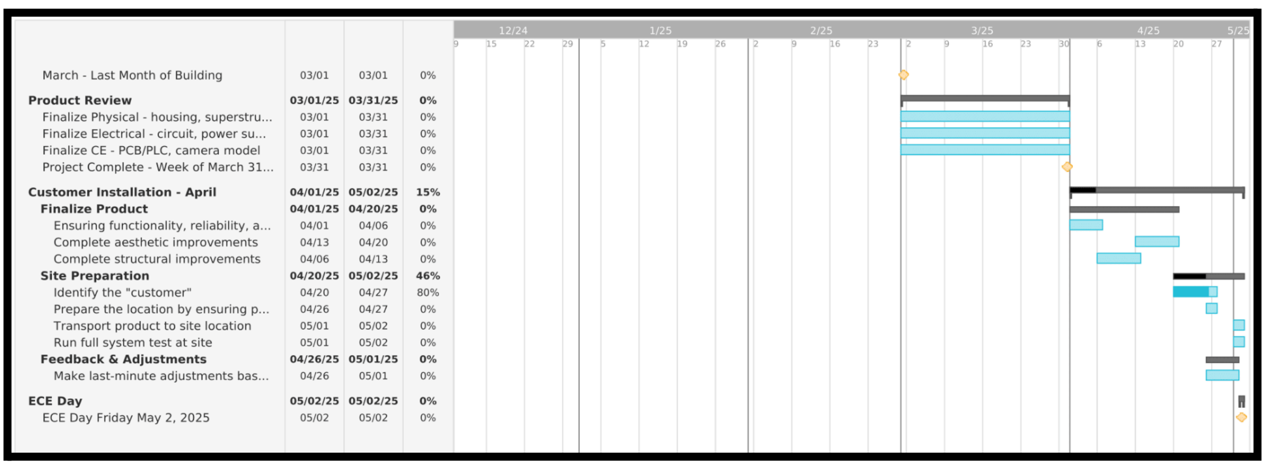
Figure 5: Engineering Requirements

Device/Sensor	Operating Voltage (V)	Current Draw (mA)	Power Draw (W)	Hours
Pump	12	300	3.6	.5
Light Sensor	3.3	0.06	0.0002	12
Soil Sensor (2x)	5	5	0.025	.5
Raspberry Pi	5	500	2.5	1
ESP-32	3.3	240 mA	0.792	12
Logi Webcam	5	700 mA	3.5	1
Grow Lights (2x)	120	92	11	12

Figure 6: Specific Electronic Requirement Breakdown

7.2 Appendix 2 - Gantt Chart - Alex





7.3 Appendix 3 - Other Appendices - Jared, Dil, Alex

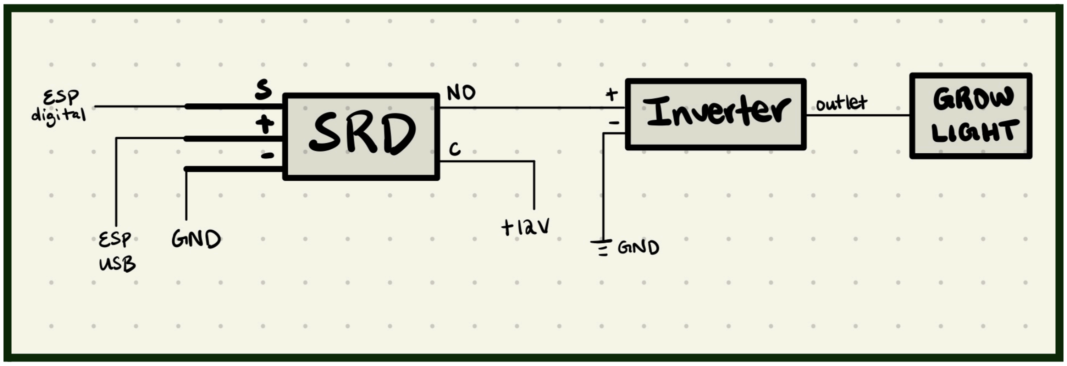


Figure 7: Complete Grow Light Circuit with Inverter

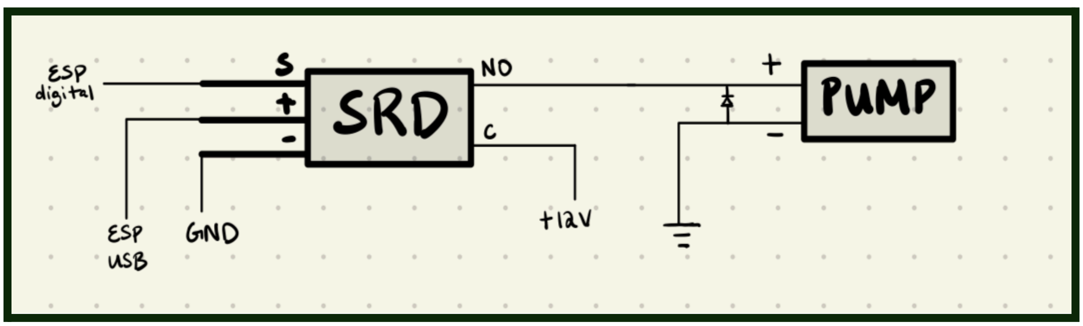


Figure 8: Complete Water Pump Circuit with SRD Control

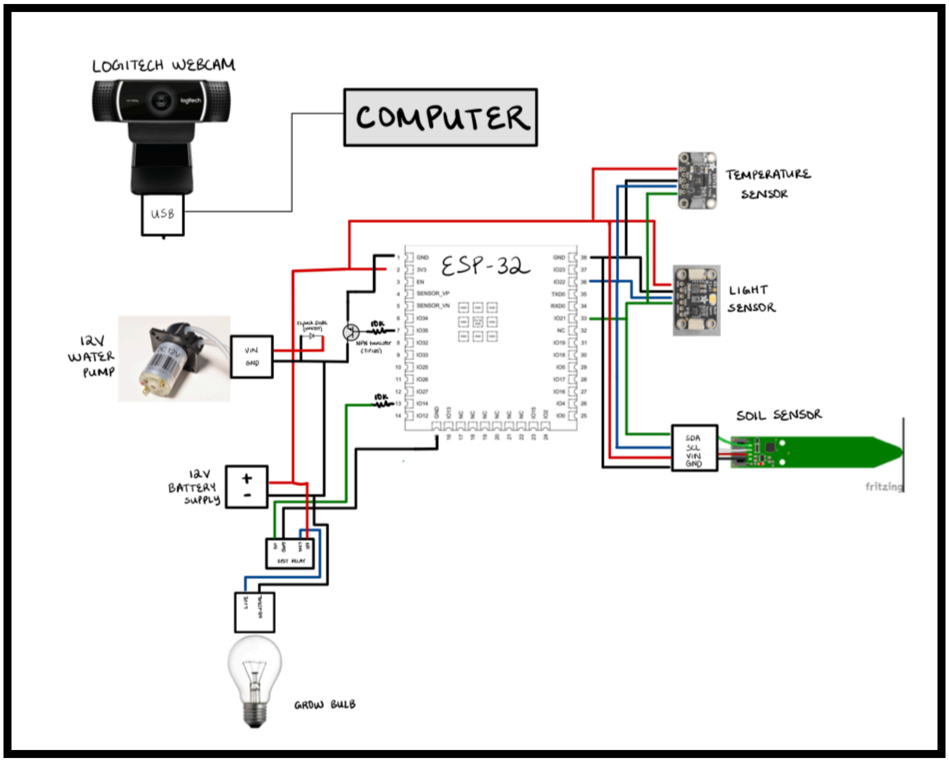


Figure 9: Complete Prototype Circuit

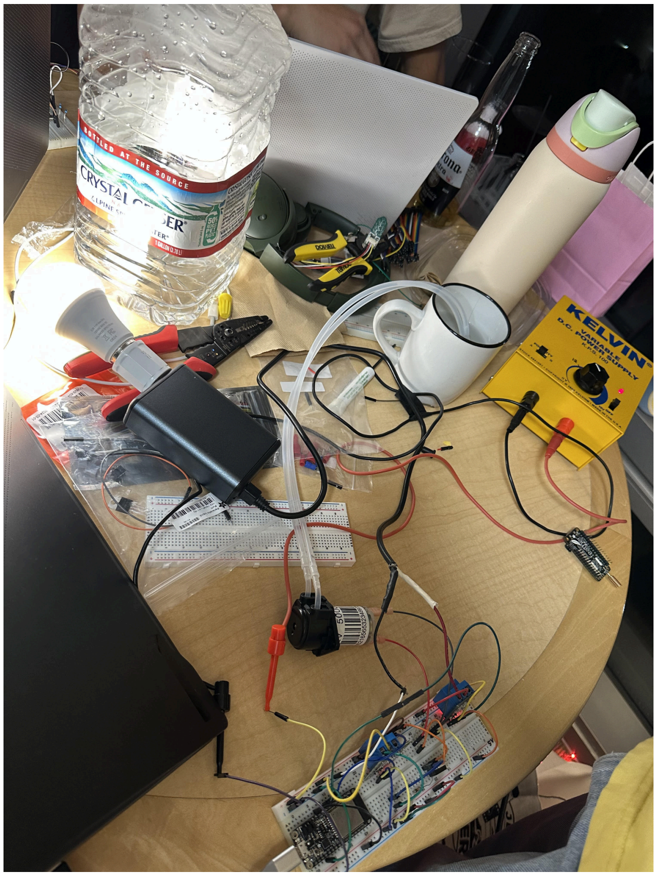


Figure 10: Water Pump and Grow Light Circuit Assembly with Grow Light On

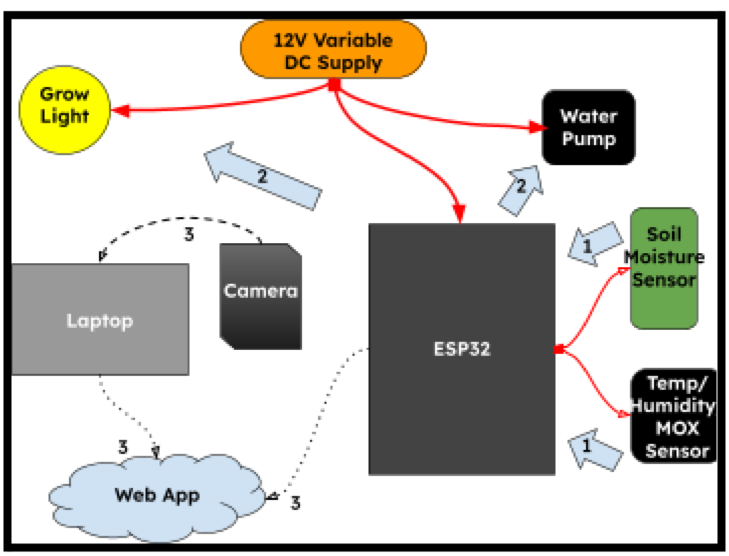


Figure 11: Illustration of Setup and Process Flow

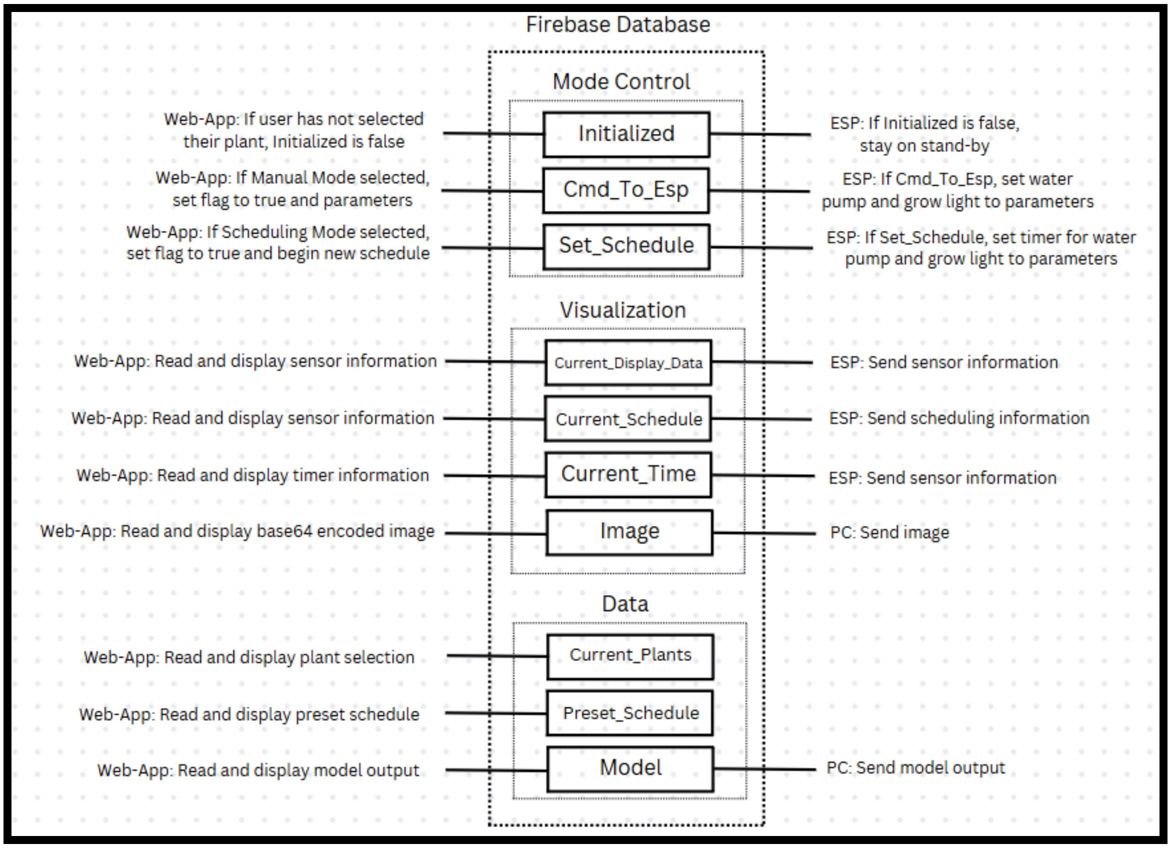


Figure 12: Overview of Firebase Database usage.