Boston University
Electrical & Computer Engineering
EC 464 Senior Design Project

User's Manual

Simple Sprouts

by
Team 26

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

Submitted: April 18, 2025

Simple Sprouts User's Manual

Table of Contents

Executive Summary

Simple Sprouts
Team 26

Simple Sprouts is an autonomous vertical farming system that enables users to grow plants indoors with minimal manual intervention. Designed for compact urban environments like university campuses and restaurants, the system integrates smart sensing, adaptive watering and lighting, and AI-based plant health monitoring. Controlled by a Raspberry Pi 5 and powered via a tiered 24VDC distribution system, Simple Sprouts automates plant care through sensor feedback and a web-based user interface.

Key features include dual-layer irrigation, grow lights for each layer, a water level sensor, temperature control via a Peltier module, and camera-based AI evaluation of plant health. All components are integrated on a fully assembled PCB with MOSFET-based control, allowing reliable and compact operation. The platform supports manual, scheduled, and adaptive modes, with data streamed to Firebase for real-time monitoring.

1 Introduction

Simple Sprouts is a smart indoor vertical farming system designed to automate the care of edible plants using embedded sensing, AI vision, and environmental control technologies. The project addresses a common problem for students, hobbyists, and professionals living in dense urban environments: the lack of space, time, and consistent conditions required to maintain healthy plant growth. Whether for growing herbs in a city apartment or for research in a university lab, Simple Sprouts allows users to grow crops like sweet basil and hot banana peppers in a compact, fully managed setup.

Our solution provides a fully integrated system with a two-tiered enclosure and a custom-built PCB at its core. Unlike typical automated planters, which rely solely on timers or remote control, Simple Sprouts features real-time sensing of soil moisture, temperature, humidity, air quality, and water tank levels. It combines this data with web-based monitoring, manual and scheduled modes, and daily AI-driven plant image assessments. Users can interact with the system via a mobile-friendly web interface to manually trigger irrigation or lighting, view system diagnostics, or receive feedback on plant health based on computer vision analysis.

From a technical standpoint, the system is powered by a 24V power supply and regulated down to 12V and 5V as needed. All key components: a water pump, two solenoid valves, LED grow lights, an ultrasonic water level sensor, and Peltier-based temperature control, are actuated by MOSFETs on a custom PCB. The Raspberry Pi 5 handles all computation, runs Python control scripts, and sends data to Firebase for real-time feedback. Imaging is conducted through two Raspberry Pi NoIR cameras mounted at each growth layer, with processed results stored and interpreted via AI tools.

From the user's perspective, Simple Sprouts offers several highlights:

- The web app displays live data for up to eight sensor channels and allows full system control.
- The soil moisture sensors automatically trigger the water pump when soil moisture falls below the defined threshold of 450 (on a scale of 400 - 2000).
- A daily image is taken of each plant level using infrared cameras and processed with NDVI-based color analysis to identify signs of stress, yellowing, or drying.
- The Peltier module maintains internal growing temperatures in the 70 - 85°F range, adjustable based on ambient conditions.
- The system fits within a 27" x 10" footprint, ideal for colleges, restaurants, or research benches.

Safety features include a physical water wall mounted between the electrical and irrigation sides of the build, as well as onboard fusing and the use of low-voltage DC components only. The software fails gracefully if sensor data is not available and alerts users via the dashboard if thresholds are exceeded.

The remaining sections of this manual describe the layout of the system, how to set it up, how to operate it in its various modes, and how to troubleshoot or respond to failure conditions. Technical and standards background, a cost breakdown, and appendices with specifications and contact information are also provided.

2 System Overview and Installation

2.1. Block Diagram Overview

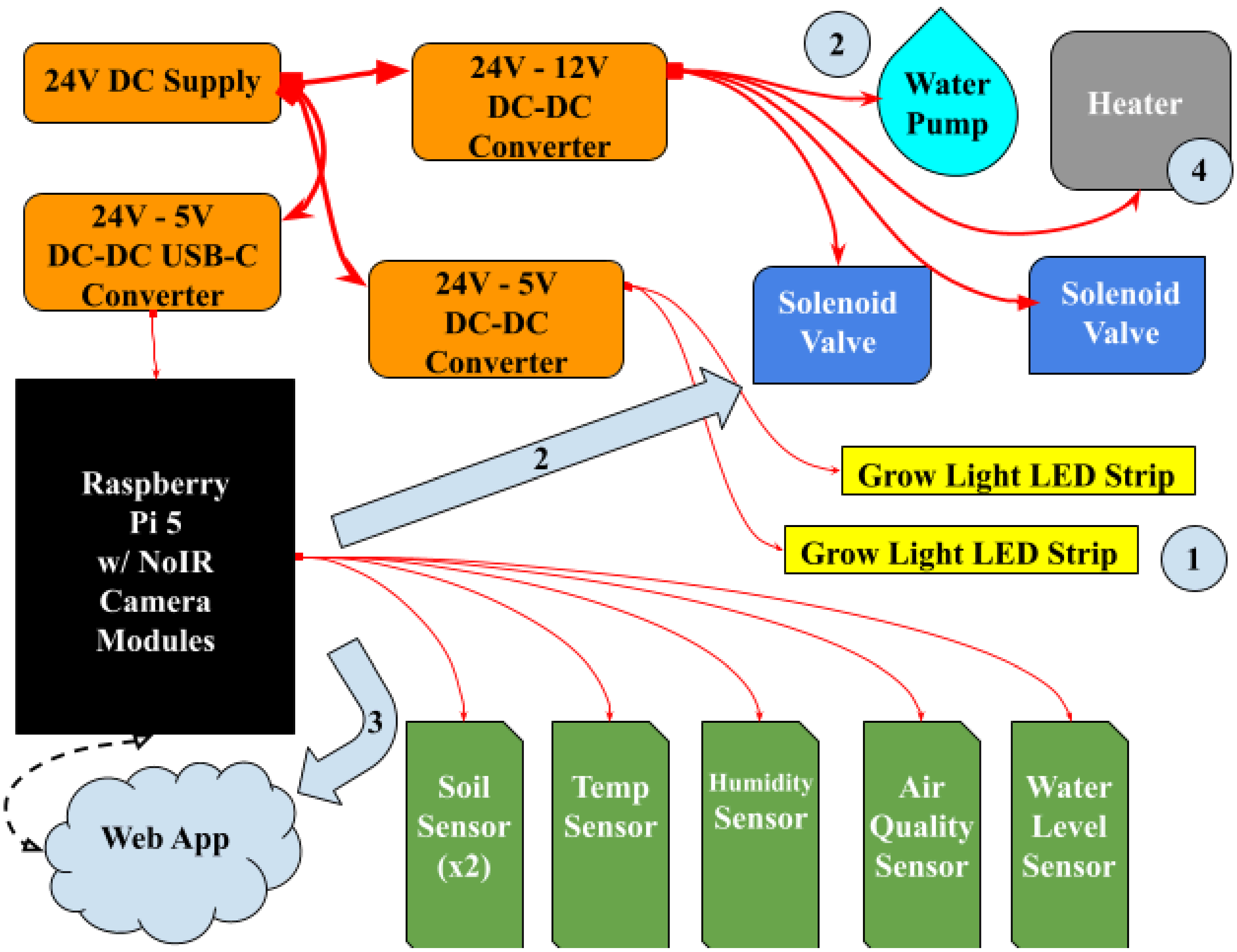

Figure 2.1 The system uses a Raspberry Pi 5 as the central controller, equipped with NoIR camera modules to monitor environmental conditions and control peripheral components. The Pi gathers data from a variety of sensors including soil moisture, temperature, humidity, air quality, and water level, and uploads it to a web application for monitoring and control. The 24V DC power supply is regulated through DC-DC converters to provide appropriate voltages for components such as LED grow lights, solenoid valves, water pump, and a heater. Commands from the web app control the Raspberry Pi, which in turn activates specific hardware to maintain optimal growing conditions in an automated environment. The system is divided into logical zones, with power distribution, sensor inputs, and actuator outputs clearly organized for efficient operation.

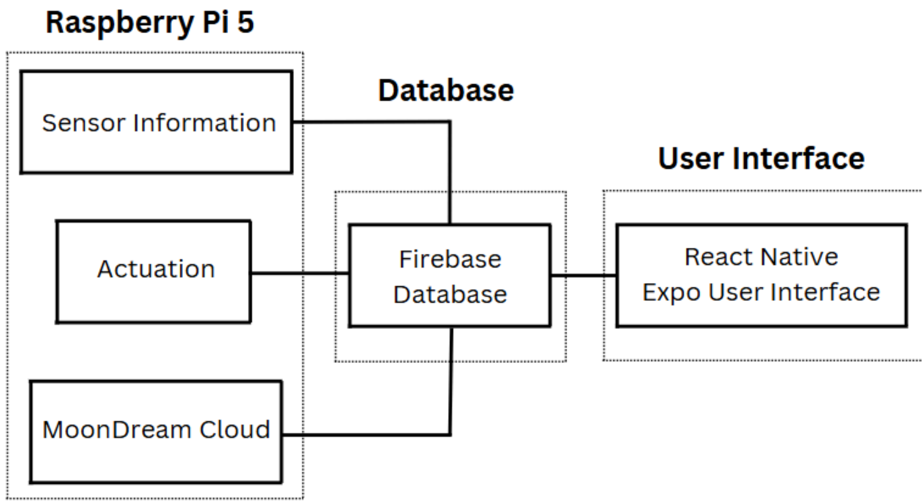

Figure 2.11 The diagram illustrates how the software components of Simple Sprouts interact to enable real-time automation and user control. At the core of the system is the Raspberry Pi 5, which collects live sensor data, controls actuators (such as the water pump, heater, and lights), and interfaces with MoonDream Cloud for AI-based plant health analysis. All sensor readings, actuation commands, and classification outputs are synchronized with the Firebase database. The user interacts with the system through a React Native Expo-based mobile dashboard, which displays system status and allows manual or scheduled control. This architecture enables two-way communication between the user and the hardware through a cloud-hosted, secure, and accessible platform.

2.2. User Interface

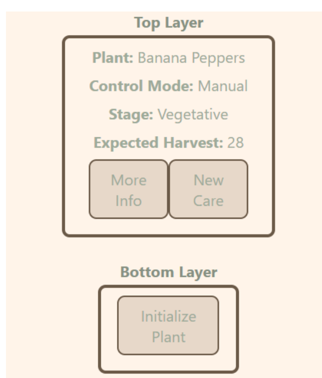

Figure 2.2 Basic Status Page

In Figure 2.2, the Basic Status page is the initial user interaction point. Each layer is shown as either initialized or uninitialized. For uninitialized layers, i.e., Bottom Layer, users can tap “Initialize Plant” to begin setup via the Initialization page. For initialized layers (i.e., Top Layer), users can tap “More Info” to view control mode, plant health, and sensor data, or select “New Care” to reinitialize the layer with a new plant.

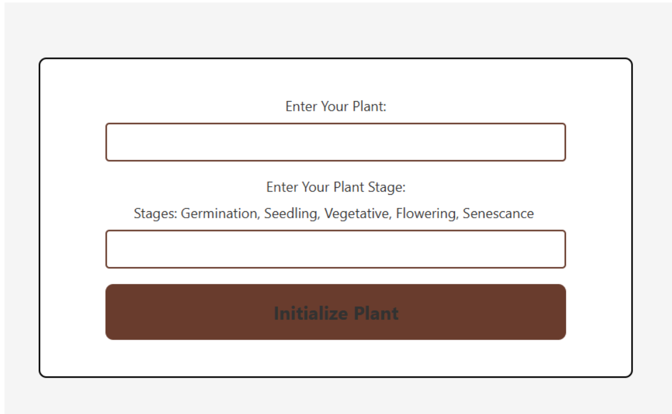

Figure 2.21 Initialization Page

Figure 2.21 captures the Initialization page, which validates plant type and growth stage inputs. These parameters are validated against system constraints (i.e. size and space requirements) via an LLM. If validation fails, navigation is blocked. Upon successful input, users proceed to the Control Mode page. Selecting Scheduling Mode routes to a configuration screen where users must define numeric values for irrigation and lighting schedules; these inputs are mandatory for proper system operation. For all other modes, we then navigate to the More Info page.

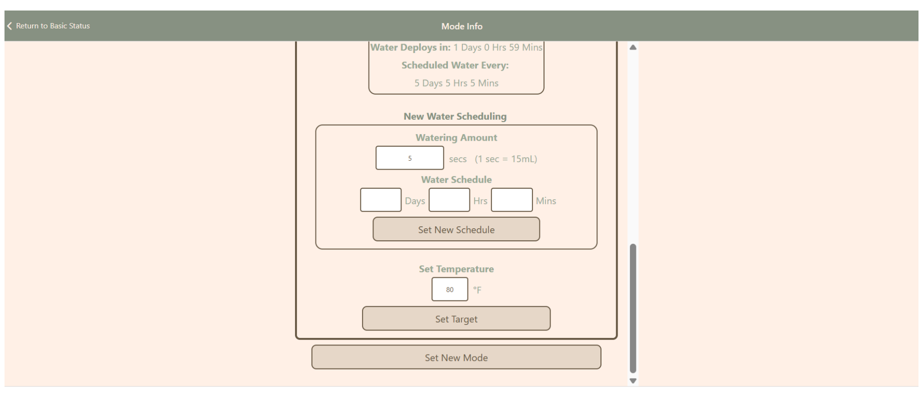

Figure 2.22 More Info Page (Scheduling Mode)

After initialization, the user is directed to the More Info screen, which displays mode-specific configuration and status. In Scheduling Mode, users can view the active schedule and define a new one. Navigation controls include a Back button (top left) to return to the Basic Status page, and a Set New Mode button to reconfigure the control mode for the current layer. A bottom navigation bar provides access to Plant Info and Health Info screens for additional data views.

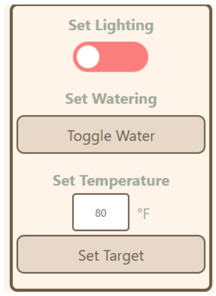

Figure 2.23 More Info Page (Manual Control)

If in Manual Mode, the user will be able to control lighting and watering on the Mode Info page with these toggle buttons.

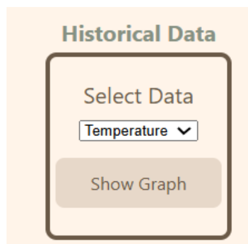

Figure 2.24 Plant Info Page (Historical Data Container)

In the Plant Info tab, the GUI will display sensor information, and the user can select a sensor to see a graph of the historical data over the last week.

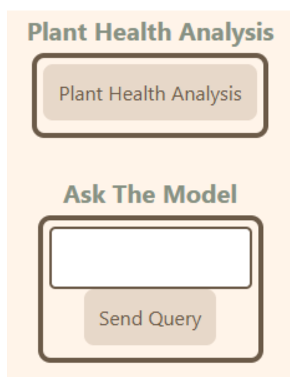

Figure 2.25 Health Status Page (Plant Health/Model Query Containers)

The Health Status tab allows users to initiate a plant health assessment via the Plant Health Analysis button. After a brief processing delay, the system displays a plant image along with diagnostic data on potential issues such as pests or blight. Users can toggle between NDVI and standard imagery using the button below the image. The Ask The Model section enables natural language queries to the MoonDream Cloud API for care recommendations (e.g., “How can I better take care of my plants?”).

2.3. Physical Description

Figure 2.3 The Simple Sprouts system is housed in a custom-built, multi-level wooden enclosure with clear acrylic panels for visibility. It contains two planter boxes and neatly integrates all essential hardware: sensors, cameras, lights, and control electronics. The setup is compact and organized, with all wiring routed cleanly through a custom PCB, making it both functional and visually polished.

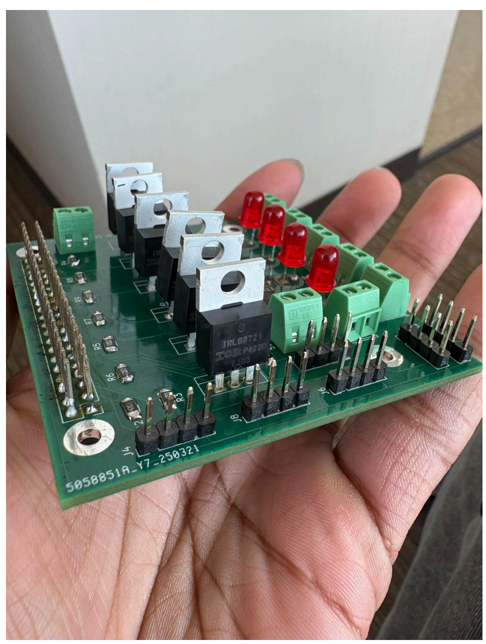

Figure 2.31 The custom PCB is a compact, professionally manufactured board designed for clean and efficient wiring of the Simple Sprouts system. It features multiple terminal blocks for actuator connections, pin headers for sensor connections, and a row of MOSFETs with corresponding red indicator LEDs for high-current device control. The layout is organized and modular, allowing for easy integration with the Raspberry Pi and peripheral components. Its sturdy build and labeled connections make it well-suited for a reliable and scalable plant automation setup.

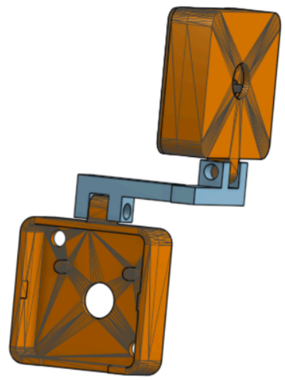

Figure 2.32 Above is a custom-made 3D-printed adjustable camera mount for the NoIR camera. It supports movement in all axial directions.

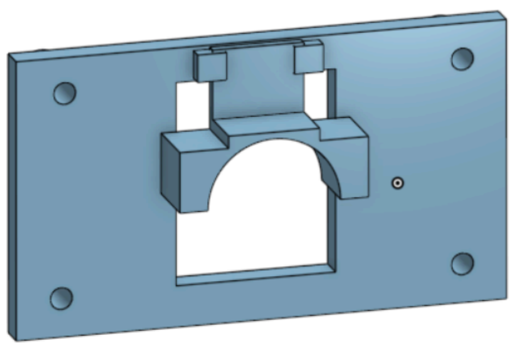

Figure 2.33 Above is a custom-made 3D-printed mount that we used to fasten the solenoid valves to and attach on the component wall. The solenoid valve slots in and the little rectangular/U-shaped section slides down to lock the valve into place.

2.4. Installation, Setup, and Support

1. Place the enclosure on a stable surface with access to a power outlet.
2. Fill the water tank with clean water.
3. Connect the 24V power adapter.
4. Ensure Wi-Fi connection for Firebase access.
5. Open the web app on a browser or mobile device (See: To Open Web App).
6. Configure plant settings (plant type, mode).
7. Begin monitoring and allow the system to handle the rest.

To Open the App:

1. Install Node.js from nodejs.org.
2. Pull the GitHub repository Simple-Sprouts and navigate to the app folder in a terminal.
3. In your terminal, run npm install expo.
4. Once in the app folder, navigate to assets/config/firebaseConfig.js and input the API Key from the Firebase Database.
5. Navigate back to the root app folder and run npx expo start –tunnel.
6. To Open On Mobile
    a. Install React Native Expo Go on the App Store.
    b. In the terminal, after Step 4, click and scan the QR code on your mobile device.
    c. Click Expo Go.
7. To Open Web App
    a. Go to the locally hosted site where it says: *Web is waiting on <website>*

3 Operation of the Project

This section describes how the user operates the Simple Sprouts system through its three core modes: Manual, Scheduled, and Adaptive. Each mode is accessible via the web-based user interface and offers different levels of automation and control. Clear instructions are provided to help users navigate each mode, understand the outcomes of their actions, and recover from unexpected behavior.

3.1 Operating Mode: Manual Operation

Manual mode provides the user with direct control over all system components. This mode is useful for initial setup, testing, troubleshooting, and precision tuning of environmental conditions.

User Interface Elements:
- Toggle switches for water pump and a solenoid valve (top or bottom tiers separated with both devices operating at once), grow lights (top or bottom), and a heater

Typical User Actions and System Responses:
- Toggle watering → Water pump and top/bottom solenoid valve (depending on user input) toggles on or off
- Toggle grow lights → LED strips turn on or off on both levels
- Capture image → NoIR camera takes a snapshot and sends it to the AI module for health classification

Abnormal Results:
- Pump runs without water flow: verify water tank level and inspect tubing for blockages
- Missing dashboard data: check for sensor disconnection or failure
- No image feedback: ensure the camera is connected and Firebase is syncing

Exiting Manual Mode:
- Clicking Set New Mode in the More Info tab and choosing a new mode
- All manual control toggles automatically reset to OFF when exiting this mode

3.2 Operating Mode: Scheduled Operation

Scheduled mode allows the user to automate plant care through fixed daily routines. This mode is ideal for routine, low-maintenance plant management.

User Interface Elements:
- Configuration panel with dropdowns for:
  - Lighting schedule (start time and duration)
  - Watering schedule (start time and duration)
- View-only panel displaying upcoming events
- Override button to skip the next scheduled action

Typical Operation:
- User sets the lighting to run from 8:00 AM to 10:00 PM
- User schedules watering at 9:00 AM and 4:00 PM for ~10 seconds per cycle
- System executes actions automatically at set times and confirms them via the dashboard

Abnormal Results:
- Missed schedule: Reset the schedule on the app and make sure Firebase is syncing
- Lights do not activate: inspect the LED wiring and MOSFET output
- Overwatering: Check watering amount and connections to valves and pump

Exiting Scheduled Mode:
- Clicking Set New Mode in the More Info tab and choosing a new mode
- Schedules remain saved but become inactive until re-enabled

3.3 Operating Mode: Adaptive Operation

Adaptive mode is now fully implemented and is intended to make real-time decisions based on live sensor data, dynamically adjusting plant care to maintain optimal growing conditions.

Features:

- Watering triggers automatically when soil moisture falls below 450
- AI image analysis helps detect plant stress and informs adjustments

User Interface Elements:

- Toggle to enable or disable Adaptive mode
- Action log showing recent system responses

Expected Behavior:

- Dry soil detected → pump and valves activate for ~10 seconds
- Lighting schedule → Determined from LLM prompt

Abnormal Results:

- Overwatering/Underwatering: Check soil sensor readings on the app, check wiring of the pump and valves
- Incorrect Lighting schedule: Check LLM responses, check the grow light wiring

How to Exit Adaptive Mode:

- Clicking Set New Mode in the More Info tab and choosing a new mode
- Select Manual or Scheduled mode to resume standard control

3.4 Additional Operations

- Live readouts of sensor data (soil moisture, temperature, humidity, air quality, and water level)
- Button to capture plant health images with NoIR cameras
- Automatic Heating Control using the Peltier module and target temperature set by the user. The heater system activates based on air temperature thresholds (set to 70℉ minimum)

3.5 Safety Concerns

The Simple Sprouts system is designed for safe indoor use and built with low-voltage components, enclosed electronics, and physically separated subsystems. However, users must remain aware of key safety concerns related to electrical, water, thermal, and power risks during setup and continued operation

Electrical Safety:

- The system uses a 24V DC supply, stepped down to 12V and 5V. These voltages are not lethal but can still damage components or short out if wires are improperly connected.
- All connections are routed through a custom through-hole PCB with onboard MOSFETs. Users should verify secure both screw terminal and pin header connections and avoid loose, exposed wires.
- No electrical components should be touched while the system is powered.

Water and Moisture Protection:

- A 12V water pump, solenoid valves, and a tubing system operate across both layers. Users must inspect for leaks and ensure the tubing is firmly connected.
- A 3D-printed divider is permanently mounted to isolate electrical components from the irrigation area, reducing splash risk or shorts.
- In case of a spill, power down the unit immediately and inspect for contact with the PCB or Raspberry Pi.

Thermal Hazards:

- The Peltier module and heater system may reach surface temperatures above 90°F. Users should avoid physical contact and ensure the cooling fan is not obstructed.
- The enclosure is ventilated and open to ambient air, but placement in a well-ventilated area is still recommended.

Power Source and Fire Risk:

- The system must be powered using a regulated 24V adapter rated for at least 3A.
- It is recommended that users include an inline fuse to guard against electrical faults.
- Never attempt to connect the system to unregulated or modified power sources.

Data Security:

- Sensor data is streamed to Firebase and does not contain personal information.
- Still, users should ensure their Wi-Fi network is secured to prevent unauthorized access to the dashboard or device.

Operator and Environmental Safety:

- Place the unit on a flat, dry surface indoors, away from potential spills, traffic, or impact.

- Do not install or use the system outdoors, in high-humidity environments, or near flammable materials.

When properly installed and routinely inspected, Simple Sprouts is safe for long-term, unattended operation. These precautions help ensure both plant and user safety under all expected conditions.

4 Technical Background

4.1. Hardware Component

The Simple Sprouts system is built upon a robust hardware architecture designed for fully automated plant monitoring and care. At its core is the Raspberry Pi 5, which functions as the main controller, performing GPIO-based switching of high-power peripherals via a custom-designed, fully assembled PCB equipped with MOSFETs and flyback diodes for efficient and safe actuation. The system is powered by a centralized 24V DC power supply, which is stepped down through a combination of DC-DC converters to supply 12V to high-power components such as the water-cooled diaphragm pump, dual solenoid valves, Peltier heating module with heatsink and fan, while 5V is supplied to the Raspberry Pi, the Grow LED strips, and sensitive sensors through both terminal block connections and USB-C outputs.

Environmental sensing is handled through multiple modules, including dual soil moisture sensors, a digital temperature sensor, a humidity sensor, and an air quality (MOX) sensor, all interfaced with the Raspberry Pi and communicating data to a Firebase cloud database via Python scripts. A key addition is the ultrasonic distance sensor, which is mounted above the water tank to track water level changes in real-time and trigger refill alerts.

For plant health monitoring, the system integrates two Raspberry Pi NoIR cameras capable of capturing infrared images, which are processed through OpenCV and TensorFlow pipelines to generate NDVI representations for AI-based plant health classification. All components are housed within a custom wooden frame enclosure featuring acrylic front and back panels for crop visibility and organized internal wiring through terminal blocks and 4-pin header connectors on the PCB. This cohesive and modular hardware setup provides a scalable, maintainable foundation for long-term smart farming deployments.

4.2. Software Component

Sensor Code

- Sensing.py - This Python script reads environmental and soil data from five different sensors connected to a Raspberry Pi over I2C and sends the readings to a Firebase Realtime Database. It initializes three environment sensors: AHT20 for temperature and humidity, PCT2075 for temperature, and SGP30 for CO_2 and TVOC air quality. Two Adafruit Seesaw soil sensors, each with its own I2C address, provide soil moisture and temperature data. After gathering readings from all sensors, the data is formatted into a structured dictionary, rounded where appropriate, and timestamped. This dictionary is printed to the console and then uploaded to the /sensor_readings/latest node in the Firebase database, overwriting the previous entry. The loop runs indefinitely, updating the Firebase database every 2 seconds. This setup enables real-time monitoring of environmental and soil conditions for IoT-based agricultural or horticultural applications.

Automation Code

- This Python script manages automated control of grow lights, water valves, a shared water pump, and the heater using GPIO pins, based on scheduling and adaptive logic. It reads scheduling and mode flags from a Firebase reference called flags_test and supports three operation modes for each component: manual, scheduling, and adaptive. In scheduling mode, components toggle on and off based on user-defined durations, with countdowns for time-to-switch and time-to-water calculated and updated in Firebase. Adaptive mode adjusts watering behavior using soil moisture readings from external sensors. The heater operates independently, turning on when the sensed temperature is more than 5 degrees below a target value and turning off when the temperature reaches the target. An ultrasonic-distance sensor is sampled every five seconds and its readings are uploaded to Firebase under the sensor_readings reference. The program uses the gpiod library for GPIO control and gpiozero for the ultrasonic sensor interface. Each component's state and timing is tracked using a structured dictionary, and the main loop ensures all components follow their respective logic and remain synchronized with Firebase. This setup enables a fully automated and remotely configurable environment control system.

Camera Code

- This Python script uses the NoIR camera module to capture plant images and generate NDVI images, making it easier to assess plant health. It then encodes the images into base64 format and uploads them to Firebase for display in the app. This setup allows for the photos to not take up too much space on the Firebase as well as processing the images in a timely manner. This code uses the picam library and a predefined .py file that was provided by the official Pi website.

Model Code

- This script acts as the interface between the user and the AI model. It automatically handles predefined queries to keep the system running smoothly, while also allowing user-initiated questions. Other scripts send query flags and command IDs to Firebase, which this script reads and responds to. It uses the MoonDream cloud API to analyze plant images provided by the camera script as well as assess plant health and provide general advice to the user. It then sends the responses back to Firebase to be accessed by the web app. This code does routine health reports that will tell the user if there is anything wrong.

App Software

- The mobile and web application was developed using React Native Expo, enabling streamlined cross-platform development and deployment. The app functions as the primary user interface for interacting with the system's physical infrastructure, including actuators and environmental sensors.
- System control is facilitated through Firebase, which acts as a real-time communication layer between the frontend and backend services. Based on the selected user mode, the app sets specific flags in Firebase to instruct backend processes, such as triggering automation routines or adjusting environmental controls.
- To ensure the user is always presented with the most current system state, the application retrieves sensor and plant data from Firebase whenever a new screen is accessed (changes for particular screens). This pull-based data strategy ensures up-to-date visibility of the system's real-world status.
- The app also integrates with a Large Language Model (LLM). This interaction is mediated through Firebase flags and operates in a call-and-response pattern. Users can request plant care insights or health analyses, and the LLM provides tailored, natural-language responses directly in the interface, enhancing the system's intelligence and usability.

5 Relevant Engineering Standards

The Simple Sprouts system complies with a range of engineering standards that ensure its reliability, safety, and interoperability in a home, research, or educational environment. These standards inform the system's electrical design, wireless connectivity, information handling, and component safety.

IEEE 802.11 Wi-Fi Communication Standard
The system connects to the cloud via Wi-Fi using the Raspberry Pi 5, which conforms to the IEEE 802.11 standard. This ensures reliable wireless communication with Firebase and secure web-based interaction.

UL and CE Power Supply Certification
The 24V DC power adapter used in Simple Sprouts meets UL and CE certification standards. This ensures compliance with international safety requirements for electrical power systems in consumer environments.

IPC-2221 (Generic Standard for Printed Board Design)
The custom PCB was designed according to IPC-2221 principles for trace spacing, thermal relief, and electrical clearance. This ensures robust circuit behavior and long-term reliability in embedded applications.

IEC 60529 – Ingress Protection Design Principles
While not formally rated, the system enclosure implements protections aligned with the IPX1 standard by incorporating a vertical acrylic barrier between the irrigation and electrical subsystems to prevent splash-related hazards.

ISO/IEC 27001 – Information Security Best Practices
Sensor data transmitted to Firebase is non-personal and follows best practices outlined by ISO/IEC 27001 by ensuring that no sensitive user information is stored, shared, or transmitted during operation.

RoHS Compliance (EU Directive 2011/65/EU)
All electrical components used in Simple Sprouts are RoHS compliant, meaning they contain no restricted hazardous substances such as lead, cadmium, or mercury. This supports both environmental safety and user health.

Adafruit and Raspberry Pi GPIO Voltage Level Specifications
All GPIO-based interactions with sensors and actuators conform to the Raspberry Pi 5's 3.3V

logic level standard, ensuring compatibility and safe operation of all I2C and digital signal interfaces within the system.

Together, these standards ensure that Simple Sprouts delivers safe, consistent, and high-quality performance to its users across all operational modes.

6 Cost Breakdown

Project Costs for Final version				
#	Quantity	Description	Unit Cost	Extended Cost
1	8	White Spray Paint	$5.98	$47.84
2	1	Plywood	$24.80	$24.80
3	20	Steel L brackets 1-½”	$0.90	$18.00
4	18	Steel L brackets 3”	$1.51	$27.18
5	1	PCB	$21.34	$21.34
6	1	Raspberry Pi 5	$85	$85
7	2	NoIR Camera Module 3	$32.99	$65.98
8	2	Camera Cable	$7.97	$15.94
9	1	PCB components	$36.08	$36.08
10	1	Acrylic	$32.29	$32.29
11	10	Woodscrews	$1.38	$13.80
12	1	Water pump	$10	$10
13	2	Grow Light Strips	$13.99	$27.98
14	1	Water tubes	$8	$8
15	1	Mosfets	$18.80	$18.80
16	6	Whitewood Planks	$3.85	$23.10

17	1	2 Black plastic planter boxes	$31.30	$31.30
18	1	Sensors	$92.55	$92.55
19	2	Solenoid Valves	$10.68	$21.36
20	1	Potting mix and gardening soil	$14.54	$14.54
21	1	USBC step-down converter	$9.39	$9.39
22	1	120W step-down converter	$9.99	$9.99
23	8	Schottky rectifier diodes	$11.80	$11.80
24	1	ADC converter	$12.80	$12.80
25	1	DC 12V-24V step down	$11	$11
26	1	Micro SD cards	$13.99	$13.99
Final Deliverable - Total Cost				**$704.85**

Table 1. Final Deliverable Cost

Above is a table with the cost breakdown of all the components that made it into the final design of our product. Generally speaking, this product would be cheaper if it were developed a bit more. One potential cost-saving measure would be to remove the Raspberry Pi 5, as the model inference is currently being handled in the cloud. This makes the computational power of the Pi 5 unnecessary. Additionally, redesigning the frame using more affordable materials could help reduce manufacturing costs. The addition of the white paint also increases the cost by ~$50. If we wanted a cheaper version, we would have left the wood unpainted.

We used a free version of MoonDream API. However, if this product were to be mass-produced, we would want to use a specially trained model and host it on our own servers, which would incur extra costs.

Overall, the development price on this product was slightly higher than expected, but there is clear potential to lower the overall price with targeted design and production optimizations.

7 Appendices

7.1 Appendix A - Specifications

Specification	Delivered Performance
Power Supply	24V DC input with internal 12V and 5V conversion
Lighting Output	12V (15W) LED strip per layer, ~400 lumens per layer
Water Pump Activation	12V diaphragm pump, !2V Solenoid Valves
Soil Moisture Sensing Range	400(very dry) - 2000(extremely wet). Ideal range 600 - 900
Temperature Range Measured	-55°C to 125°C, ±1°C accuracy, 0.125°C resolution
Humidity Range Measured	0% to 100% RH, ±2% accuracy, 0.024% resolution
Air Quality Sensing	TVOC - 0 to 60,000 ppb CO2 - 400 to 60,000 ppm
Camera	NoIR 12MP Pi Camera Module 3 for IR image analysis
Water Tank Monitoring	2 to 400 cm range with ±0.2 cm resolution
Heater Output	Peltier module, 12V operation, maintains ~22-28°C range
System Control	Raspberry Pi 5 with custom PCB interfacing all components
Control Interface	Web dashboard accessible via Wi-Fi and Firebase backend
Manual Mode	Direct toggles for each subsystem
Scheduled Mode	Lighting and watering are scheduled via dropdown inputs
Adaptive Mode	Real-time sensor-based automation (fully functional)
PCB Design	Custom 4-layer through-hole & surface-mounted board, MOSFET-based switching, terminal blocks for actuators, pin headers for sensors
Physical Enclosure Dimensions	27” x 10” base footprint, 4 feet tall (2 grow layers)
Sensor Update Rate	0.20 Hz (5 seconds) for dashboard display and data logging

Enclosure Material	Painted plywood frame with acrylic side panels
Electrical Isolation Features	Vertical divider separating water tubing and electronics
System Weight	~50 lbs fully assembled, including soil
Estimated Total Cost	~$700 for full prototype (excluding tools)
Github	

All specifications reflect the final assembly and testing state of the Simple Sprouts system.

7.2 Appendix B - Team Information

The Simple Sprouts system was fully designed and built by Team 26, consisting of the following members:

- **Jared Solis:** Originally from Staten Island, NY, I've always had an interest in creating new things and decided to pursue engineering at BU. In May, I will graduate with a major in Computer Engineering and a minor in Systems Engineering. After graduation, my plans are currently to be determined, but I will continue to use what I've learned here throughout life.
- **Sourav Shib:** I am from Queens, NYC. I grew up in a very multicultural neighborhood. Majored in mechatronics in high school and loved engineering. I came to BU to pursue engineering and find the field of engineering I enjoy the most. In the summer, I will be joining Teradyne as a software test engineer, where I will test components for clients.
- **Arthur Hua:** I am from Gilbert, Arizona. I study Computer Engineering at Boston university in the class of 2025. I will be pursuing a Masters in Computer Engineering (computer systems) at ASU.
- **Dilhara DeSilva:** I'm from Staten Island, NY and came to BU to pursue a degree in electrical engineering, with a concentration in Energy Technologies and Sustainability. Over the past four years, I've had the pleasure of working with different renewable energies including solar and hydropower. Now in my final semester, I'm excited to begin the next chapter of my journey - this summer, I'll be joining Sargent & Lundy's Nuclear Power Group as a Graduate Electrical Engineer, where I'll help shape the future of clean energy infrastructure.
- **Alex Muntean:** Engineering has always been a core part of my life, from playing with LEGOs when I was in middle school to competing on a robotics team in high school to completing my first full-fledged capstone project in college. Born and raised in New York City, I came to Boston University in 2021 where I majored in Electrical Engineering

all four years and am currently about to wrap up my final ever college semester. In the summer, I will be joining the workforce at ARUP here in Boston as a Graduate Electrical Engineer, where I will be designing the systems and circuits of buildings and other large-scale projects!

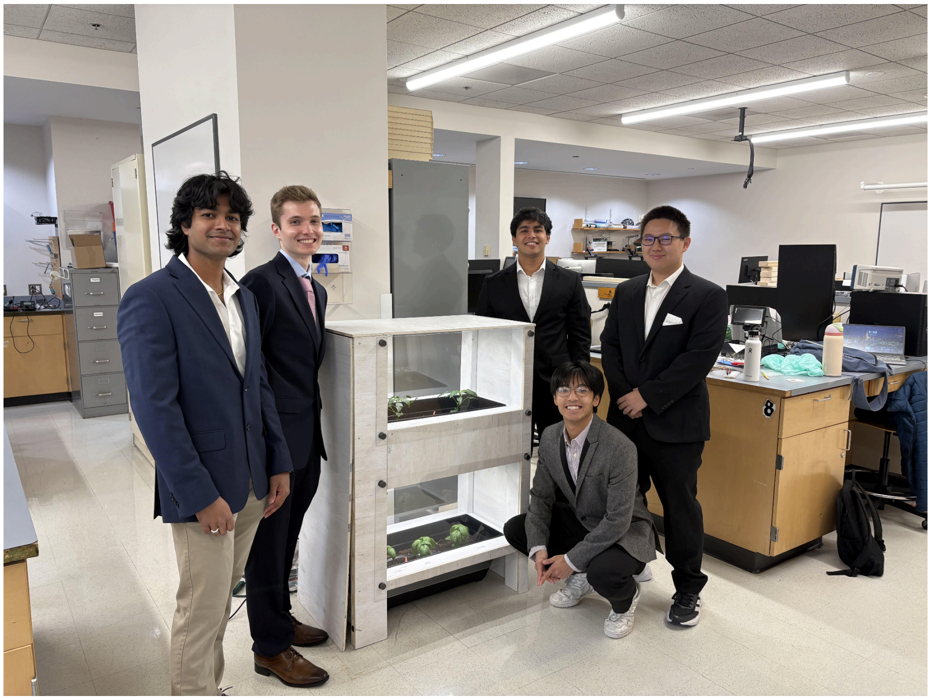

Figure 7.2 Meet the team!