Grow A Garden

Project Description:

Grow A Garden is an IoT-integrated automated sprinkler and monitoring system designed to support sustainable agriculture through real-time soil condition tracking and smart irrigation. Developed by the Grow A Garden Team, this system leverages NodeMCU-based technology combined with soil moisture, temperature, and humidity sensors to create a responsive and efficient watering solution.

This system was developed to aid both urban and rural gardeners-especially those who lack time or technical experience-by automating plant care, minimizing water waste, and ensuring healthier crops. The mobile application connected to the device allows users to manually water or enable an automatic mode based on soil moisture thresholds.

System Requirements:

Compatibility:	Must be Android 7.0 and higher (Not compatible with IOS)
Required storage:	100 mb and higher
Memory (RAM):	2gb or higher
Internet:	Requires internet to access the software
Bluetooth:	Does not support bluetooth connection

Prototype Description:

The prototype was developed by coding the GUI directly using React, Tailwind CSS, with the help of HTML, and Visual Studio Code. This approach allowed for a more realistic simulation of the user experience and better alignment with the final product's intended behavior. Major interface components include:

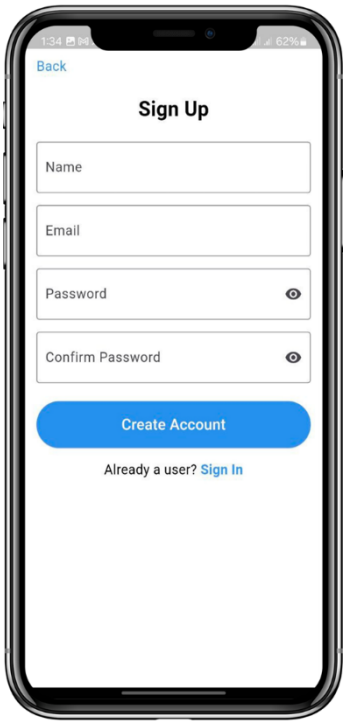
- Create Account feature for better user compatibility
- Sign Up old account for User old presets
- User-Friendly and Easy to Navigate UI
- Soil Moisture, Temperature, Humidity tracker
- On and Off button for the Sprinkler and Automatic Mode.

Yancy, a struggling Local farmer because he has low workers,. She uses Grow A garden App to better monitor his Plants and Crops for better gains and the app’s monitoring algorithm finds an equivalent match for his liking. Yancy then schedules a session at the Farm with his few coworkers about the new app that he discovered for better workload distribution.

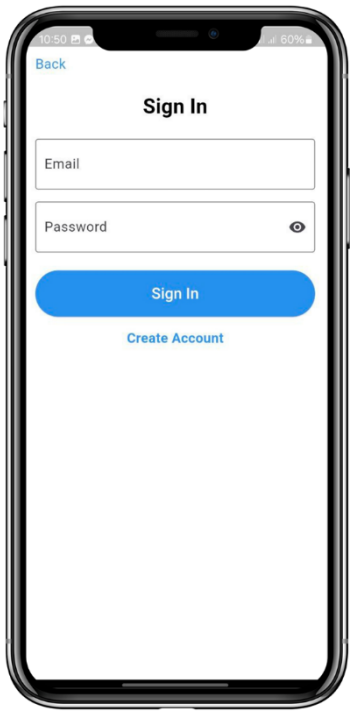
Grow A Garden Prototype:

LOG IN SCREEN:

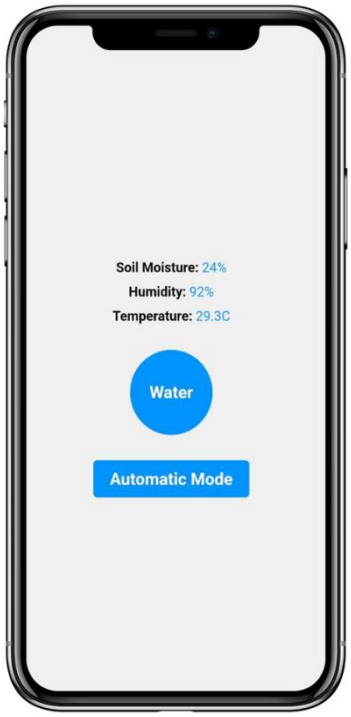
SIGN UP ACCOUNT:



SIGN IN ACCOUNT:



SOIL MONITOR UI:



Rationale:

We decided to code the system interface and controls manually to ensure full customization over

the automation logic, sensor integration, and real-time feedback. This approach allowed us to precisely implement environment-based triggers and flexible controls that prebuilt platforms may not support. The final system design integrates key features from earlier concepts:

- Dashboard-centric for real-time monitoring
- Sensor-priority for accuracy and responsiveness
- Automation-first for water efficiency

Changes to the Requirements:

- **Added:** Automated trigger logic based on moisture thresholds and adjustable settings for temperature and humidity ranges
- **Removed:** Manual override via mobile app (replaced with physical control panel for reliability)
- **Refined:** Sensor calibration process, water scheduling interface, and alert notification system

Initial Evaluation Plan:

The evaluation will be done on-site with controlled testing conditions. Testers will observe system behavior by manipulating environmental factors and noting responses in real time.

The assessment will follow three main components: Usability Specifications, Heuristic Evaluation, and Participant Survey and Feedback.

Usability Specifications

The system is expected to meet the following criteria during testing:

- **Learnability:** First-time users can operate the system and interpret sensor data within 5 minutes
- **Efficiency:** Automated response (sprinkler activation) happens within 10 seconds of threshold detection
- **Safety:** System shuts off automatically in extreme conditions or when not needed
- **Flexibility:** Users can toggle between manual and automatic modes
- **Feedback:** Real-time display of soil moisture, humidity, and temperature with clear alerts

Population

Targeting 10–15 individuals (e.g., agriculture students or garden caretakers). They will complete the following test tasks:

- Set up the system and calibrate sensors
- Monitor environmental data from the dashboard
- Test automatic sprinkler activation through simulated dry soil
- Adjust system thresholds and review alert responses

• **Developer / UI Designer Member**	• **Task(s)**
• Karl Francis R. Sumampong	• Will be recording time users interact with a task section, taking notes of the user's experience, and relay the task that the participant will do.
• Earl Yancy L. Cuyos	• Will be recording time users interact with a task section, taking notes of the user's experience, and relay the task that the participant will do.
• John Benedict A. Landa	• Will observe user navigation flow, document usability issues or confusion, and help summarize participant feedback post-session.

•

Profile Setup	Within 3 minutes or Below	Highly Acceptable	Successful

	Above 3 minutes	Not Acceptable	Unsuccessful
Matching	Within 2 minutes or Below	Highly Acceptable	Successful
	Above 2 minutes	Not Acceptable	Unsuccessful
Booking a Session	Within 5 minutes or Below	Highly Acceptable	Successful
	Above 5 minutes	Not Acceptable	Unsuccessful
Feedback Submission	Within 2 minutes or Below	Highly Acceptable	Successful
	Above 2 minutes	Not Acceptable	Unsuccessful

Heuristic Evaluation

The evaluation of our Automated Sprinkler System will follow the 10 Usability Heuristics for User Interface Design.

Visibility of System Status

The system provides clear and immediate feedback through the dashboard, showing current soil moisture, temperature, and humidity levels, as well as sprinkler status (on/off).

Match Between System and Real World

The interface uses simple, real-world terms such as “Dry Soil,” “Humidity Level,” and “Temperature Alert,” avoiding technical language to make it understandable for everyday users like farmers or gardeners.

User Control and Freedom

Users can switch between manual and automatic modes at any time, stop watering, or adjust sensor thresholds. Controls are reversible, giving users full control without being locked into any action.

Consistency and Standards

All displays, buttons, and icons follow consistent labeling and layout standards. Sensor readings and system terms such as “Activate,” “Threshold,” and “Auto Mode” are used uniformly.

Error Prevention

Sensor input settings are validated with limits to prevent misconfiguration. Only acceptable ranges can be set, reducing the risk of overwatering or system failure.

Recognition Rather Than Recall

Key information such as current readings, status indicators, and control options are always visible on the dashboard, minimizing the need for users to remember settings.

Flexibility and Efficiency of Use

The system supports both beginners and advanced users with a simple control panel for first-time use and quick-access functions for frequently adjusted settings.

Aesthetic and Minimalist Design

The interface maintains a clean layout, focusing on essential data and controls. Visual cues and organized sections guide the user without overwhelming them.

Help Users Recognize, Diagnose, and Recover from Errors

When a sensor fails or input is invalid, plain-language alerts explain the issue and suggest a solution. For example, “Sensor Disconnected – Check Wiring” helps users quickly resolve problems.

Help and Documentation

Guided tooltips and short instructions are built into the interface, especially near sensor settings and system mode toggles. Users can operate the system confidently with minimal external help.

Participant Survey and Feedback

After conducting the on-site system testing,

DATA GATHERING METHOD

DESCRIPTION

Survey (Quantitative)

Following the system test, participants will be asked to complete a survey to evaluate their experience with the automated sprinkler system. The team will analyze the results using a 5-point Likert scale.

Feedback (Qualitative)

The survey also includes a section for open-ended feedback, allowing participants to express any concerns, suggestions, or issues encountered during their interaction with the system.

Question	Method of Answer
Section 1	
Participant Number	Short Answer
On a scale of 1 to 5, how would you rate your experience with Stubby?	5-Point Scale
On a scale of 1 to 5, how was the UI design of the prototype?	
How easily were you able to follow the tasks provided?	
Section 2: Features of the Prototype	
Navigation and Screen Transitions	5-Point Scale
Viewing Match Suggestions	
Scheduling a Study Session	
Selecting a Campus Venue	
Submitting Feedback After Sessions	
Using Dashboard to View Sessions	

Accessing and Editing Profile Information	
Section 3: Feedback Section	
Your Feedback	Short Answer

Interpretation Scale:

Score	Rating	Outcome
5	Highly Acceptable	Excellent
4	Acceptable	Meets expectations
3	Neutral	Needs improvement
2	Fairly Acceptable	Needs redesign
1	Not Acceptable	Failed usability

Implementation Challenges and Justification:

While manually developing the interface and logic for the automated sprinkler system allowed us full control, it also came with its share of difficulties. For example, real-time sensor simulation was limited to test data, as full integration with hardware components (e.g., actual sensors and water pumps) could not be completed due to time limitations. Similarly, mobile connectivity and remote control features had to be excluded from the initial version.

Despite these limitations, we prioritized building key screens and control flows that would allow for effective usability testing. The system was developed to reflect its core purpose: smart, efficient, and responsive irrigation based on live environmental data.

Early feedback from classmates and testers indicated a positive response to the dashboard layout and the ability to switch between manual and automatic modes. This reinforced our decision to

custom-code the interface and logic to better simulate the real-world behavior and usability of the system.