PROJECT TITLE

SmartSprout: Intelligent, Data-Driven Plant Care

STUDENT/TEAM INFORMATION

Team Name if any: **Team # on Canvas you have self-signed-up for:**	**The Green Thumbs**
Team member 1 (Team Lead)	Gedle Gedleh (ggedleh1275@sdsu.edu)

ABSTRACT (15 points)

(Summarize your project (motivation, goals, system design and results). Max 300 words).

SmartSprout is an open-source IoT system that continuously measures soil moisture, air temperature, and humidity to optimize plant watering. A TTGO-LoRa32 T-Display (ESP32) reads a capacitive soil probe and an AHT20 sensor, shows moisture percentage on its OLED, and triggers an LED strobe and buzzer when soil falls below a configurable threshold. Every five seconds, it sends JSON data over Wi-Fi to a Flask API on AWS EC2, which logs readings and issues a Telegram alert if moisture dips under 40 percent. In testing, SmartSprout accurately mapped raw ADC values to 0–100 percent moisture, reliably fired local and remote alerts, and maintained stable operation over extended intervals. Future enhancements include a web dashboard, predictive water scheduling via machine learning, and solar-powered autonomy.

INTRODUCTION (15 pts)

Motivation/Background (3 pts)

(Describe the problem you want to solve and why it is important. Max 300 words).

Motivation / Background:

- **Plant health struggles:** Many plant owners either underwater when busy or overwater by sticking to fixed schedules, leading to root rot or stress.
- **Existing solutions:** Manual moisture checks are inconsistent, and generic timers don’t adapt to real-time soil conditions. Commercial"smart pots" are costly and non-customizable.
- **Data-driven advantage:** By leveraging soil-moisture and ambient sensors, we can make precise, evidence-based watering decisions to save water and improve plant vitality.

Project Goals (6 pts)

(Describe the project general goals. Max 200 words).

1. Real-time monitoring of soil moisture (0–100 percent).
2. Local alert via LED and buzzer when soil is too dry.
3. Remote notifications through Telegram.
4. Cloud logging for long-term trend analysis.

Assumptions (3 pts)

(Describe the assumptions (if any) you are making to solve the problem. Max 180 words).

Assumptions:

- **Connectivity**: The sensor node remains within Wi-Fi range (≤10m from the home router).
- **Power**: TTGO board is powered via a stable USB or battery source; future solar integration assumes sufficient sunlight.
- **API reliability**: Telegram and AWS services are available with minimal downtime.
- **Single-plant focus**: Each node monitors one container or plant to simplify calibration and thresholds.

SYSTEM ARCHITECTURE (20 pts)

(Describe the final architecture you have implemented listing sensors, communication protocols (Wi-Fi, BLE, ...), cloud services and user interfaces. Include a block diagram of the system. Max 300 words).

SmartSprout consists of three main layers:

1. Sensor Node (ESP32 TTGO-LoRa32):

- Capacitive soil moisture sensor (GPIO 36)
- AHT20 I^2C temperature/humidity sensor (GPIO 21/22)
- OLED display (ST7789, TFT_eSPI)
- LED + buzzer for dry alerts
- Firmware: PlatformIO/Arduino, non-blocking millis() loops

2. Cloud Backend (AWS EC2 + Flask):

- Flask API listens on /data for JSON payloads
- Parses fields: moisture, raw, tempF, hum
- When moisture < threshold, sends Telegram DM

3. User Interfaces:

- Local: OLED shows moisture %; red "DRY!" message below threshold
- Remote: Telegram direct messages; planned web dashboard

```
TTGO Sensor Node → Wi-Fi → Flask API on EC2 → {Database, Telegram Bot, Dashboard}
 ↓                               ↑
LED & buzzer alerts          POSTs every 5 s
```

FINAL LIST OF HARDWARE COMPONENTS (5 pts)

(Write the final list and quantity of the components you have included in your system)

Component/part	Quantity
TTGO-LoRa32 T-Display (ESP32 + OLED)	1

Capacitive soil moisture sensor V1.2	1
Adafruit AHT20 temperature/humidity	1
LED (5 mm) + 220 Ω resistor	1
Active buzzer	1
Breadboard & jumper wires	several
USB power supply cable	1

PROJECT IMPLEMENTATION (30 PTS)

Tasks/Milestones Completed (15 pts)

(Describe the main tasks that you have completed in this project. Max 250 words).

Task Completed	**Team Member**
• **Week 1–2:** Integrated soil and AHT20 sensors; ADC and I²C readings verified. • **Week 3:** OLED display of moisture %; threshold logic for LED and buzzer. • **Week 4:** Wi-Fi connectivity; HTTP POST to Flask backend. • **Week 5:** Deployed Flask API on AWS EC2; implemented Telegram bot alerts. • **Week 6:** End-to-end testing; stable alerts and logging over 24 h.	Gedle

Challenges/Roadblocks (5 pts)

(Describe the challenges that you have faced and how you solved them if that is the case. Max 300 words).

Flask dependency errors: pip install blocked by Debian-managed environment.
Solution: created Python venv and installed flask and requests inside it.

Tasks Not Completed (5 pts)

(Describe the tasks that you originally planned to complete but were not completed. If all tasks were completed, state so. Max 250 words).

Task	**Reason**
Web dashboard	Postponed to focus on core sensing and alerts

WEAK POINTS / FUTURE WORK (15 pts)

(Mention at least two points of your project that have room for improvement. These points can be additions to the existing project setup (components) or improvement of the current implementation. Max 200 words).

1. **Web Dashboard:** Develop a React/Flask front-end for trend visualization and threshold tuning.
2. **Predictive Scheduling:** Use logged data to train a simple ML model for watering forecasts.
3. **Solar Power:** Integrate LiPo battery and solar charger for outdoor, off-grid operation.
4. **Additional Sensing:** Add light, soil temperature, and pH sensors for comprehensive plant health.

SOURCE CODE (25 pts)

Please include a link to the source code of your project. A link to a repository (like GitHub) is preferred.

Your answer here