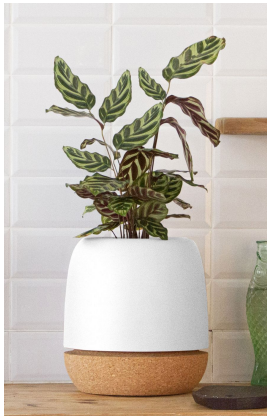
Team B3: Sproutly Proposal Presentation

Jana Armouti (Presenter),
Zara Mansoor, Yuna Shin

Problem Statement

- New plant owners struggle with proper care
- Frequent travellers can’t monitor plants consistently
- Plant owners need to monitor multiple factors
  - Watering schedules
  - Light levels
  - Temperature
  - Humidity
  - Nutrient levels
- These factors vary for each plant

Existing Solutions



Click & Grow, Koru, AC Infinity, PlantHive

Click & Grow ($99-$299):
- Automatic Watering & Light Control

Koru ($400):
- Soil Moisture, Light, and Temperature Sensors
- Automatic Watering

AC Infinity ($399-$1499):
- Full Automation & Advanced Sensors

PlantHive ($984):
- Full Automation & Advanced Sensors
- AI-Driven Controls

Our Solution

- All-in-one automated greenhouse system
- Sensors track key factors
- Provide real-time data, live footage, and recommendations
- Remote-controlled system to manage plant health
- Option for manual or automatic control
- ML for plant health analysis
- Scalable design

Areas covered: Software, Hardware

Requirements

Condition Control	Maintain temperature: ±2°C range Maintain soil moisture, light intensity, nutrients, pH, humidity: ±10% range
Response Time	Actuator response to changes: <1 minute Conditions should reach target: <1 hour
Secure Live Streaming	Latency: <5 seconds 24/7 streaming: night and day vision
Water Capacity	Up to 2 weeks without refill: 3.5L (~250mL per day)
Health Classification	False positive rate: <10% False negative rate: <5%
Scalability	Should provide actionable recommendations based on plant type
Real-Time Data	Sensor to Web communication: <500 ms

Technical Challenges

- Reliable connections and quick response between software and hardware
  - Live streaming/real-time data should reflect current plant conditions
  - Plant condition should be precisely controlled in a timely manner
  - Possible challenges: network latency, slow response from sensors/actuators
- Integration of multiple sensors
  - Conflicting sensor data
- Precise control of water amount
  - Possible challenges: pump inaccuracies, water leakage
- Accurate health detection
  - Possible challenges: inconsistent data quality, different light level affecting results
- Different conditions for different plants
  - Possible challenges: incorrect classification of plants

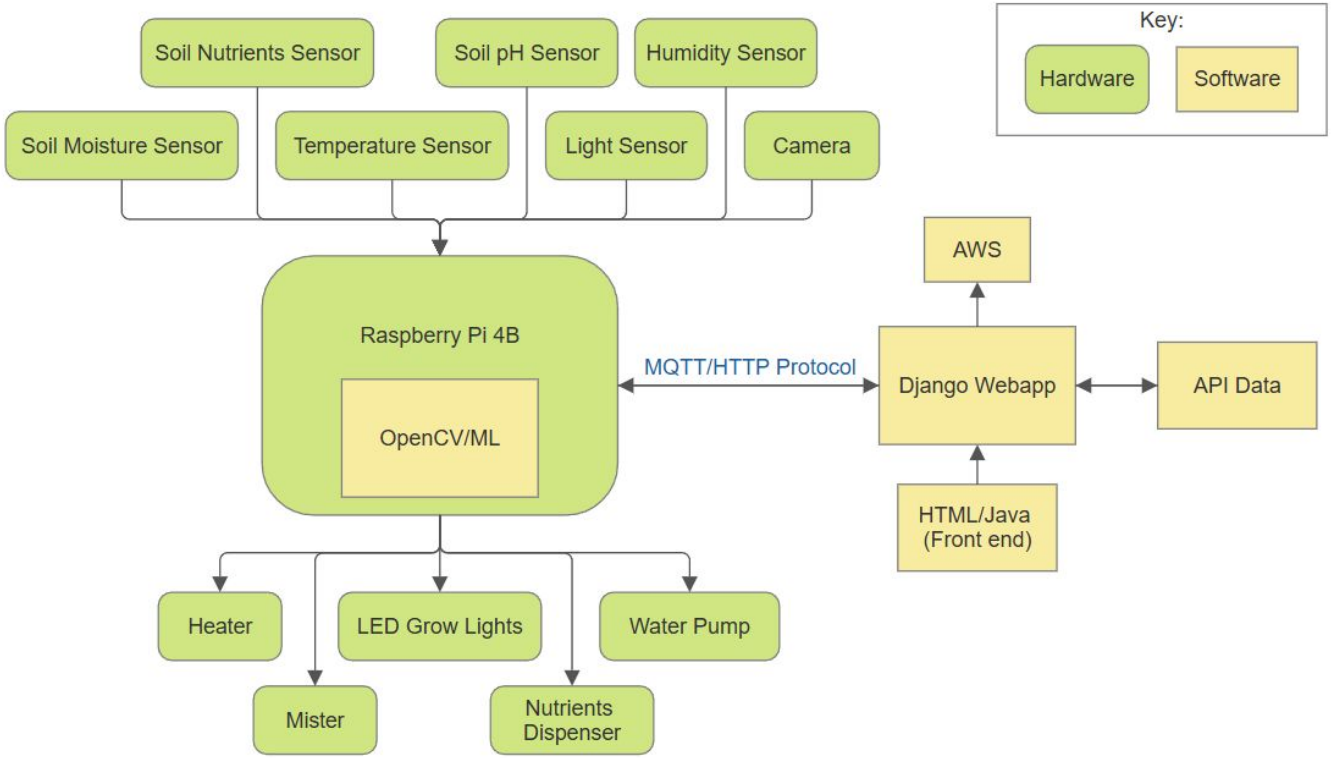
Solution Approaches - Hardware

- Raspberry Pi: Core system for processing and continuity
- Sensors:
  - Soil Moisture Sensor: Monitors soil hydration
  - Temperature Sensor: Tracks temperature levels
  - Light Sensor: Measures light intensity
  - Optional Sensors: Nutrient, pH, and humidity sensors
- Actuators:
  - Water Pump, Heater, LED Grow Lights: Automatically adjust environmental conditions
  - Mister, Nutrient Dispenser: Control humidity, nutrients, and pH levels
- Camera: Provides live footage for real-time plant monitoring

Solution Approaches - Software

- Web Application (Django, HTML, Java):
  - Displays real-time data and live camera feed.
  - Allows users to manually or automatically adjust conditions.
  - Integration with plant database API for classification and care recommendations
- Data Analysis:
  - Machine Learning & Computer Vision: Analyzes plant health and growth stages.
  - OpenCV: Processes visual data for plant monitoring
- Cloud Hosting (AWS): Manages server-side data and application hosting.
- Communication Protocols: Transmit data between hardware and web application.
  - HTTP Protocol to stream live video feed
  - MQTT Protocol to send sensor data

Solution Approaches



Testing, Verification and Metrics

	Testing Method	Metrics
Sensor Accuracy	Use calibrated reference instruments Compare sensor readings to reference values Repeat 3-5 times to ensure consistency	Accuracy of within ±10% of true values
Condition Control	Monitor readings every 10 minutes until the condition stabilizes Once stabilized, compare reading to target environmental condition	Maintain temperature: ±2°C range Maintain soil moisture, light intensity, nutrients, pH, humidity: ±10% range
Response Time	Measure time taken between setting target value and actuator response Monitor readings every 10 minutes until the condition reaches target value	Actuator response: <1 minute Conditions should reach target: <1 hour
Live Streaming	Latency: use software tool to measure video latency Day and night vision: monitor the quality of the video feed in daylight and nighttime conditions	Latency: <5 seconds Video quality: 1080p at 30fps

Testing, Verification and Metrics - Continued

	Testing Method	Metrics
Plant Classification	Verify that the system correctly identifies plants by comparing predicted value to ground truth label	Correctly identifies plants: >90% accuracy rate
Health Classification	Verify that the system correctly identifies plant health status by comparing predicted value to ground truth label	False positive rate: <10% False negative rate: <5%
Web App User Experience	End users will interact with the application and rate it on a scale of 1 to 5 on the following categories: ease of use, visual design, functionality, performance	Each category: >4 average
Real-Time Data	Measure time for web app to receive input data from raspberry pi	Sensor-Web communication: <500 ms

Tasks and Division of Labor

Zara

Core Sensors:
- Temperature Sensor
- Soil Moisture Sensor
- Light Sensor

Core Controls:
- Heater
- Water Pump
- LED Grow Lights

Automatic Actuator Control Logic

Yuna

Web Application Development:
- User Interface
- Plant Data API

Manual Actuator Control via web application

Soil pH and Nutrients Sensors

Nutrient Dispenser

Jana

Machine Learning:
- Sensor & Image Data Collection
- OpenCV for Leaf Detection
- Health Predictions

Camera Setup

Humidity Sensor

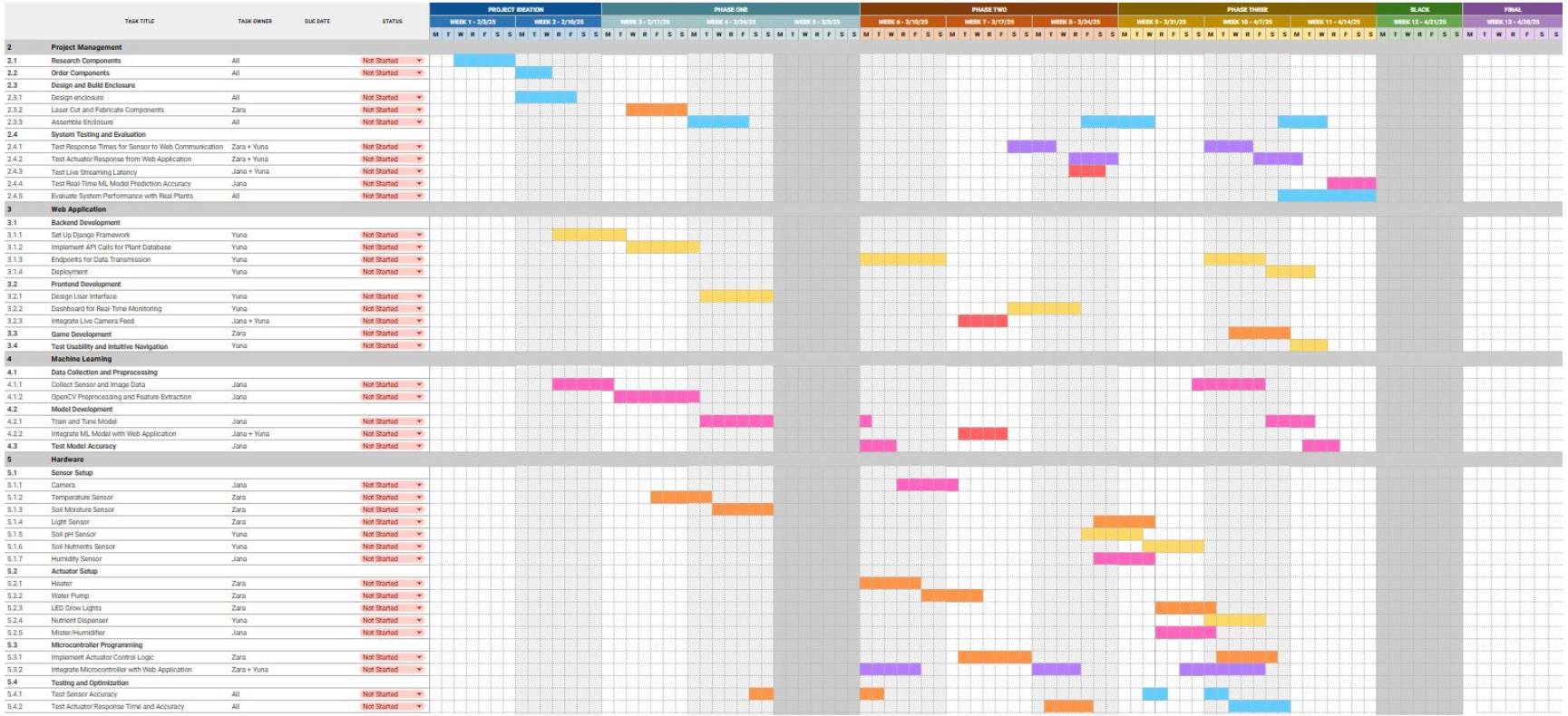
Humidifier/Mister

Sensor & Microcontroller Integration for real-time data

Live Streaming Integration

Schedule

Jana
Yuna
Zara
Zara + Yuna
Jana + Yuna
All



	TASK TITLE	TASK OWNER	DUE DATE	STATUS
2	**Project Management**			
2.1	**Research Components**	All		Not Started
2.2	**Order Components**	All		Not Started
2.3	**Design and Build Enclosure**			
2.3.1	Design enclosure	All		Not Started
2.3.2	Laser Cut and Fabricate Components	Zara		Not Started
2.3.3	Assemble Enclosure	All		Not Started
2.4	**System Testing and Evaluation**			
2.4.1	Test Response Times for Sensor to Web Communication	Zara + Yuna		Not Started
2.4.2	Test Actuator Response from Web Application	Zara + Yuna		Not Started
2.4.3	Test Live Streaming Latency	Jana + Yuna		Not Started
2.4.4	Test Real-Time ML Model Prediction Accuracy	Jana		Not Started
2.4.5	Evaluate System Performance with Real Plants	All		Not Started
3	**Web Application**			
3.1	**Backend Development**			
3.1.1	Set Up Django Framework	Yuna		Not Started
3.1.2	Implement API Calls for Plant Database	Yuna		Not Started
3.1.3	Endpoints for Data Transmission	Yuna		Not Started
3.1.4	Deployment	Yuna		Not Started
3.2	**Frontend Development**			
3.2.1	Design User Interface	Yuna		Not Started
3.2.2	Dashboard for Real-Time Monitoring	Yuna		Not Started
3.2.3	Integrate Live Camera Feed	Jana + Yuna		Not Started
3.3	**Game Development**	Zara		Not Started
3.4	**Test Usability and Intuitive Navigation**	Yuna		Not Started
4	**Machine Learning**			
4.1	**Data Collection and Preprocessing**			
4.1.1	Collect Sensor and Image Data	Jana		Not Started
4.1.2	OpenCV Preprocessing and Feature Extraction	Jana		Not Started
4.2	**Model Development**			
4.2.1	Train and Tune Model	Jana		Not Started
4.2.2	Integrate ML Model with Web Application	Jana + Yuna		Not Started
4.3	**Test Model Accuracy**	Jana		Not Started
5	**Hardware**			
5.1	**Sensor Setup**			
5.1.1	Camera	Jana		Not Started
5.1.2	Temperature Sensor	Zara		Not Started
5.1.3	Soil Moisture Sensor	Zara		Not Started
5.1.4	Light Sensor	Zara		Not Started
5.1.5	Soil pH Sensor	Yuna		Not Started
5.1.6	Soil Nutrients Sensor	Yuna		Not Started
5.1.7	Humidity Sensor	Jana		Not Started
5.2	**Actuator Setup**			
5.2.1	Heater	Zara		Not Started
5.2.2	Water Pump	Zara		Not Started
5.2.3	LED Grow Lights	Zara		Not Started
5.2.4	Nutrient Dispenser	Yuna		Not Started
5.2.5	Mister/Humidifier	Jana		Not Started
5.3	**Microcontroller Programming**			
5.3.1	Implement Actuator Control Logic	Zara		Not Started
5.3.2	Integrate Microcontroller with Web Application	Zara + Yuna		Not Started
5.4	**Testing and Optimization**			
5.4.1	Test Sensor Accuracy	All		Not Started
5.4.2	Test Actuator Response Time and Accuracy	All		Not Started