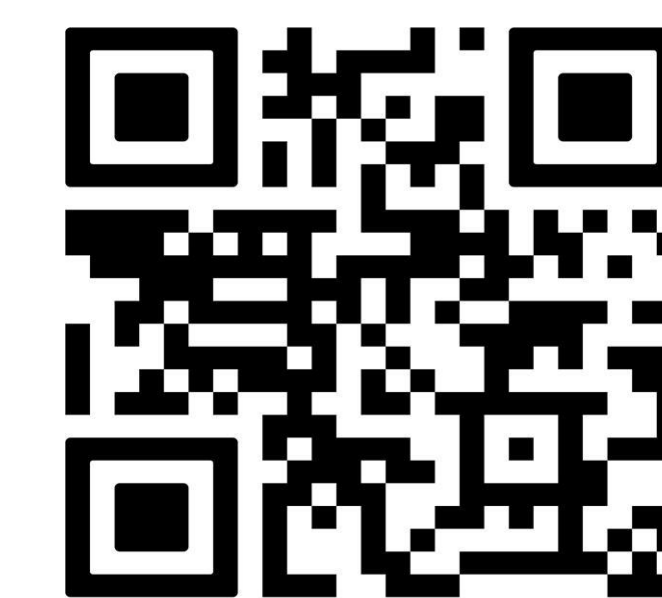
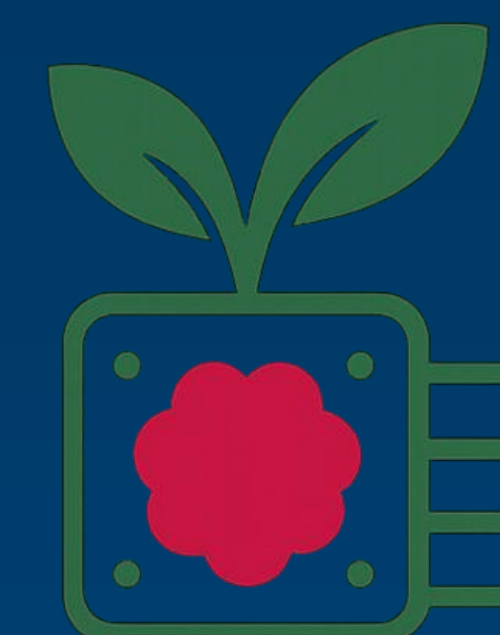




Background

Botanical Pi is a Raspberry Pi based smart plant monitoring and remote care system.

Many plant owners struggle to keep watering and lighting consistent, especially when away from home. Botanical Pi addresses this by collecting live sensor data such as soil moisture, temperature and humidity, and ambient light. The system logs readings for history and troubleshooting and provides a web dashboard for real time monitoring and remote control. Users can log in, view live telemetry from the physical device, and remotely trigger watering and lighting. The goal is to make plant care more reliable, convenient, and observable over time.

Key Requirements

Functional Requirements:

- User authentication for dashboard access
- Live telemetry display for soil moisture, temperature and humidity, and light
- Remote actuator control for Water Now and Light On and Off
- Configurable thresholds and timing values for automation

Non-Functional Requirements:

- Safe remote control that ignores stale commands older than 30 seconds
- Reliable end to end operation from frontend to AWS to Raspberry Pi to hardware
- Readable user interface for demo use

Architectural Design

System Overview:

- Botanical Pi is a Raspberry Pi based smart plant monitoring and remote care system that provides live telemetry, configurable thresholds, and remote watering and lighting control through a web dashboard.

Layered Architecture:

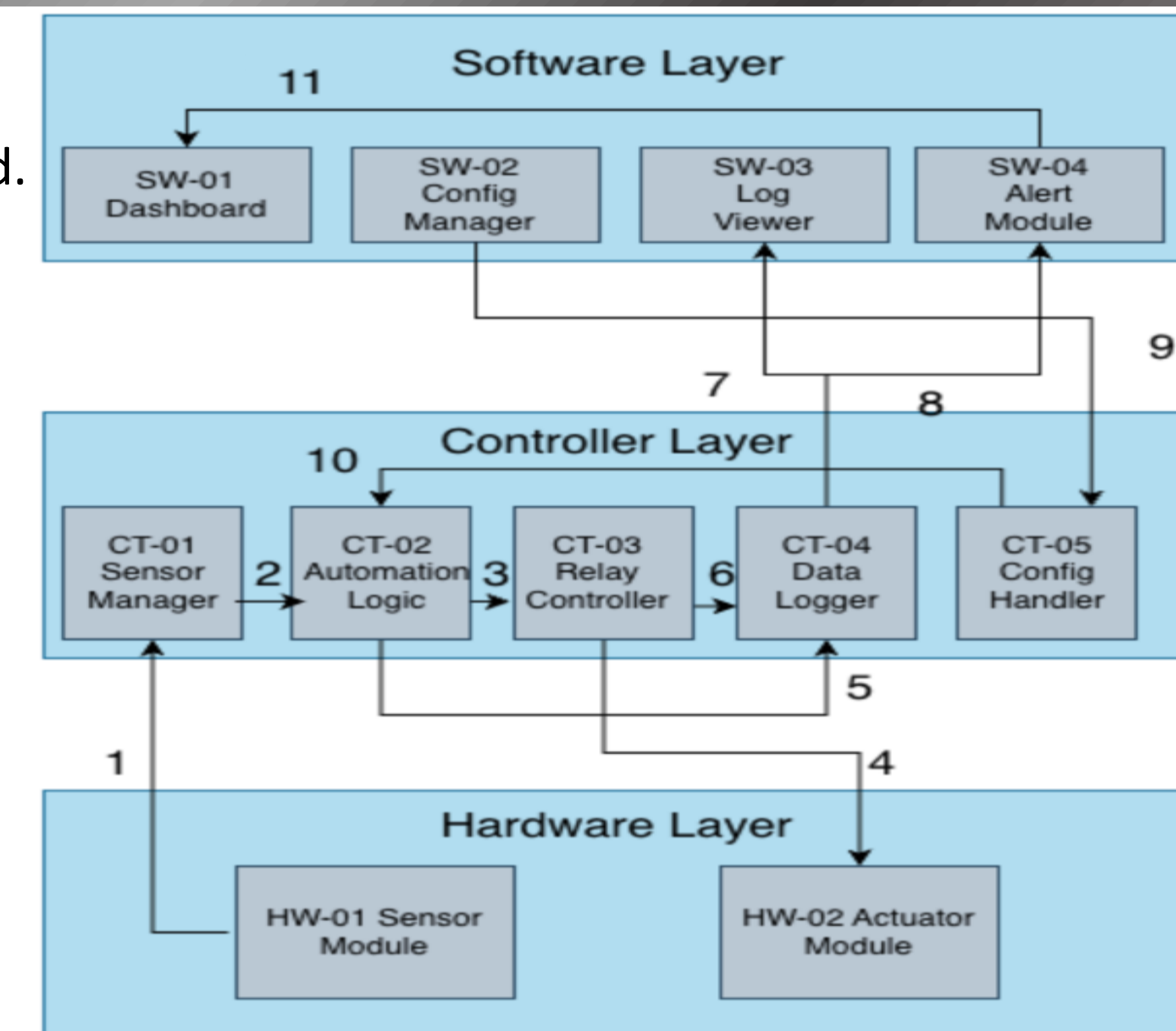
- Hardware Layer: HW-01 Sensor Module, HW-02 Actuator Module (pump and grow light)
- Controller Layer: CT-01 Sensor Manager, CT-02 Automation Logic, CT-03 Relay Controller, CT-04 Data Logger, CT-05 Config Handler
- Software Layer: SW-01 Dashboard, SW-02 Config Manager, SW-03 Log Viewer, SW-04 Alert Module

Cloud Integration:

- Dashboard communicates through the live AWS backend for telemetry and remote commands
- Data persistence supports access from any device on the network

Data and Control Flow:

- Sensors → Raspberry Pi processing and logging → dashboard live display
- Manual control commands flow dashboard → AWS → Raspberry Pi → relays → pump or light
- Safety logic discards stale commands older than 30 seconds



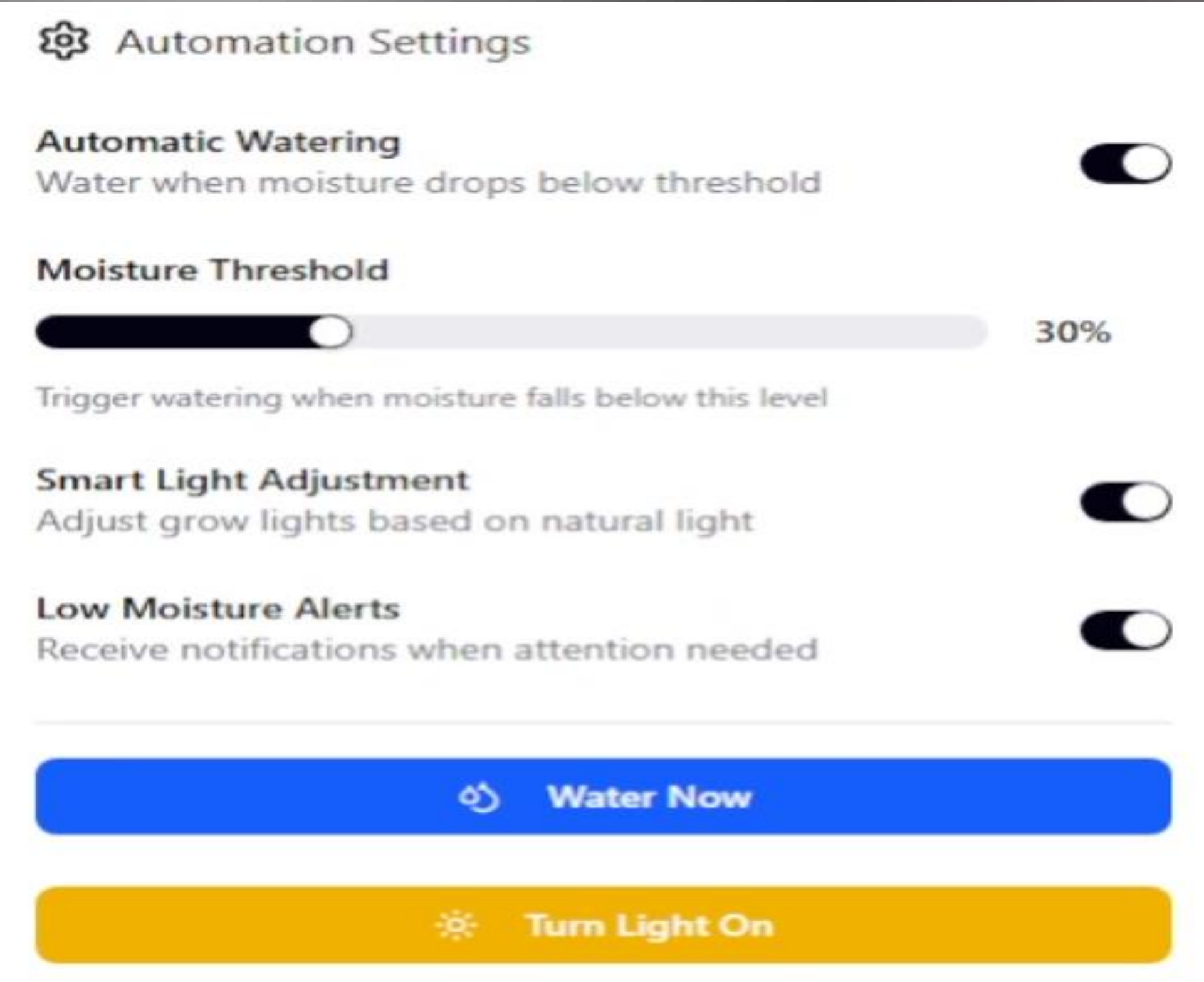
Implementation Details and Test Plan

Implementation Details

- Raspberry Pi collects sensor readings and publishes data to the live AWS backend
- Web dashboard connects to live API data for real telemetry with no hardcoded values
- Manual control commands follow the full path dashboard to AWS to Raspberry Pi to relays to pump and light
- Safety feature discards commands older than 30 seconds

Test Plan

- Sensor validation: confirms readings update live and respond to real condition changes
- End to end control: confirm Water Now and Light On and Off trigger real hardware reliably
- Automation checks: confirm thresholds trigger actions only when limits are exceeded
- Logging verification: confirm readings and events are saved and viewable historically
- Failure handling: test disconnect and reconnect sensors, invalid inputs, and network interruption



Conclusions and Future Work

Conclusions

- Achieved a working end to end demo where a user can log in, view live sensor data, and control the pump and grow light remotely
- Layered architecture improves maintainability and separation between sensing, control, and the dashboard
- Logging and configuration improves reliability, debugging, and usability

Future Work

- Add alerts for out-of-range conditions and failure states
- Improve historical analytics and visualization such as trends and summaries
- Expand support for multiple devices per account

