CAPSTONE · MINI-MBA IN ARTIFICIAL INTELLIGENCE · FINAL SUBMISSION

Jandec Farms: Grow at home. Learn with AI. Automate with purpose. Every drop justified.

A family kitchen garden. Five beds. One slightly over-engineering human. Nothing here is sold, so there's no ROI to hide behind — unless you count tomatoes, ginger, and the questionable satisfaction of putting Wi-Fi in the garden. The real question is simpler: *does the person holding the hose actually know what these plants need?* Jandec Farms is an AI Architect and IoT enthusiast's attempt to answer that with sensors, data, explainable decisions, and just enough autonomy to be useful without letting the machines take over the tomatoes. Every watering decision starts with a measurement, shows its reasoning, and still waits for a human to say, *"Yep. Water it."*

Ananth Radhakrishnan — owner, grow-table head, and automation & AI adoption manager, Jandec Farms

Learning record: The Learning Ledger · Live dashboard: family portal

● §1–5 · BUILT AND RUNNING | ● §6 · PROJECTED, NOT BUILT | ■ §7 · GALLERY

01

One farm, one operator, one honest question

Who Jandec Farms is, why nobody is paying for this, and why that makes the engineering harder rather than easier.

● BUILT AND RUNNING

Jandec Farms is family-owned, family-operated, and very enthusiastically family-fed. Ginger, peppers, tomatoes, and cut flowers grow across five beds of roughly thirty square feet each sitting on a modest 1.25 acre plot. Every harvest ends up on the kitchen counter, dinner plate, or dining-room table. Nothing is sold, shipped, or measured against a revenue target.

The entire organisation chart is also remarkably efficient: one person.

Farm owner, grow-table operator, process-improvement lead, data analyst, IoT engineer, and AI adoption manager all report to the same individual — which makes performance reviews awkward, but accountability unusually clear.That became part of the experiment.The farm provided a deliberately small environment in which to apply ideas developed through the Mini-MBA in Artificial Intelligence: observe a real process, identify where decisions rely on habit rather than evidence, collect better data, remove unnecessary work, introduce automation where it adds value, and decide very carefully where automation should stop.

That changed the design brief.

A commercial grower can justify irrigation automation through labour savings, water reduction, yield improvement, or lower operating cost. At this scale, those arguments become almost comical. The financial return on optimising thirty square feet of ginger is unlikely to impress an investment committee.

Removing that justification, however, exposes a much more interesting one:

> Can a system make a decision about something a human cannot easily observe, explain why it made that decision, and provide enough evidence for the human to decide whether it should be trusted?

That is where the project stopped being simply about irrigation.

Soil moisture at root depth is invisible. A bed that looks dry may still contain plenty of water below the surface. A bed watered every Tuesday at 7:00 AM will happily receive water whether it needs it or not. Both approaches can be wrong, and both can remain wrong quietly for days.

So instead of beginning with automation, Jandec Farms begins with evidence.

Sensors create observations. Data turns those observations into trends. Analytics provides context rather than relying on a single reading. The decision layer evaluates whether watering is actually justified. Every recommendation retains the measurements and reasoning that produced it. And before water flows, a human remains accountable for authorising the action.

The objective is therefore not maximum autonomy. it is **useful autonomy with boundaries.**

Optimise the process where repetition adds no value. Use analytics where human observation is weak. Let AI connect signals that would otherwise be tedious to interpret. But keep the system explainable, traceable, and deliberately constrained wherever a wrong decision has consequences.

Because the interesting question is not ***WHETHER AI CAN TURN ON A PUMP.*** That part is easy, The interesting question is **WHETHER IT CAN EARN THE RIGHT TO.**

THE FIVE BEDS [AS OF 8/14/2026]

ZONE	BED	CROP	SUN	SOIL	METHOD	SENSING TODAY
zone-01	Ginger North	Ginger	Partial	Loam	Drip	**PLATE FITTED · INTERMITTENT**
zone-02	Ginger South	Ginger	Full	Loam	Drip	**MEASURED**
zone-03	Flower bed	Mixed ornamentals	Partial	Loam	Sprinkler	**MEASURED**
zone-04	Pepper bed	Pepper	Full	Sandy	Drip	**AWAITING HARDWARE**
zone-05	Tomato bed	Tomato	Full	Clay	Drip	**AWAITING HARDWARE**

02

The goal: a decision you can audit, an action only a human can take

The system was built around a deliberately narrow promise: **Observe continuously. Reason explicitly. State uncertainty honestly. Ask before acting. Remember what happened.**

● BUILT AND RUNNING

The objective was never "automate the watering". It was to build a closed decision loop in which the garden is continuously measured, those measurements are converted into evidence, the evidence is evaluated against known conditions, and the resulting recommendation arrives with something most automated systems conveniently leave out:

how sure are you?

Every cycle produces a **measurable, recordable, explainable, and auditable** decision. The system can recommend watering, recommend waiting, flag uncertainty, or decide that the available evidence is simply not good enough to make a responsible call.

What it cannot do is quietly turn on the water because an algorithm felt confident. That boundary is intentional. The **machine owns the analysis**. The **human owns the consequence.**

WHAT THE SYSTEM DECIDES

Every cycle, each bed receives exactly one of five verdicts. There is no sixth, and no free-text alternative:

VERDICT	MEANING	NEEDS A HUMAN?
WATER_RECOMMENDED	Evidence supports irrigating this bed now.	**YES — ALWAYS**
DO_NOT_WATER	Evidence positively argues against irrigating.	No action to authorise
MONITOR	Neither case is made; keep observing.	No action to authorise
INSUFFICIENT_DATA	Not enough valid readings to reason at all.	Refuses to advise
SENSOR_ERROR	The inputs themselves are untrustworthy.	Refuses to advise

WHY THIS IS AGENTIC, NOT MERELY AUTOMATED

A timer is automation. At 7:00 AM, the timer wakes up, opens a valve, and considers its intellectual contribution to agriculture complete.

Jandec Farms operates differently. The system repeatedly executes its own **perception** → **reasoning** → **recommendation** → **human action** → **observation** loop against one bounded goal: determine whether the available evidence justifies watering a particular bed.

It consumes current measurements, compares them with recent behaviour and historical baselines, evaluates drying patterns, determines which defined rule the evidence supports, selects one of five bounded verdicts, and attaches a confidence level to that judgement.

Then it stops. That boundary is deliberate. The system has authority to assess, recommend, and learn — but not to act. It cannot redefine the objective, expand its scope, or convert confidence into execution. Its autonomy is intentionally bounded to judgement; operational authority remains with the human.

The system owns the recommendation. The human owns the decision.

THE NON-NEGOTIABLES

- **No schedule gets a vote.** Time can inform the decision; measurements have to justify it.
- **Show the receipts.**Every recommendation must expose the data, rule, trend, and confidence behind it.
- **Human authority is absolute.**The system may reason with conviction; the valve still answers to a person.
- **Overrides are evidence, not embarrassment.** Disagreement is recorded, observed, and learned from — not quietly deleted.
- **History stays honest.** Readings, decisions, actions, and outcomes are append-only; even the tomatoes get an audit trail.

> The system recommends. The human decides. The record remembers which of them was right.

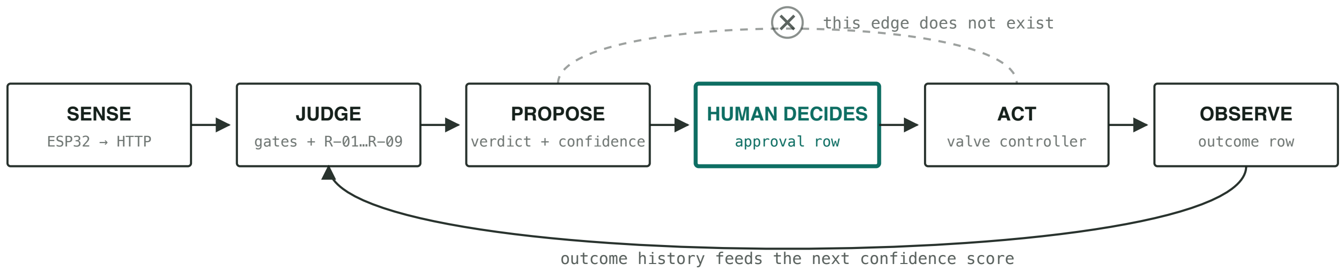
03

The implemented architecture

From a wetness plate in the soil to a dashboard a family member can open from another country — and the one edge in the whole diagram that does not exist.

● BUILT AND RUNNING

The system is local-first. Sensing, reasoning, approval and actuation all happen on hardware inside the house; the internet is an outbound mirror that can be read but cannot reach in. Nothing in the cloud can water anything.



The loop, and the missing edge. Every irrigation action must pass through an approval row, and the database query that writes an action asserts the approval exists — so there is no code path from a proposal to a valve that skips the human.

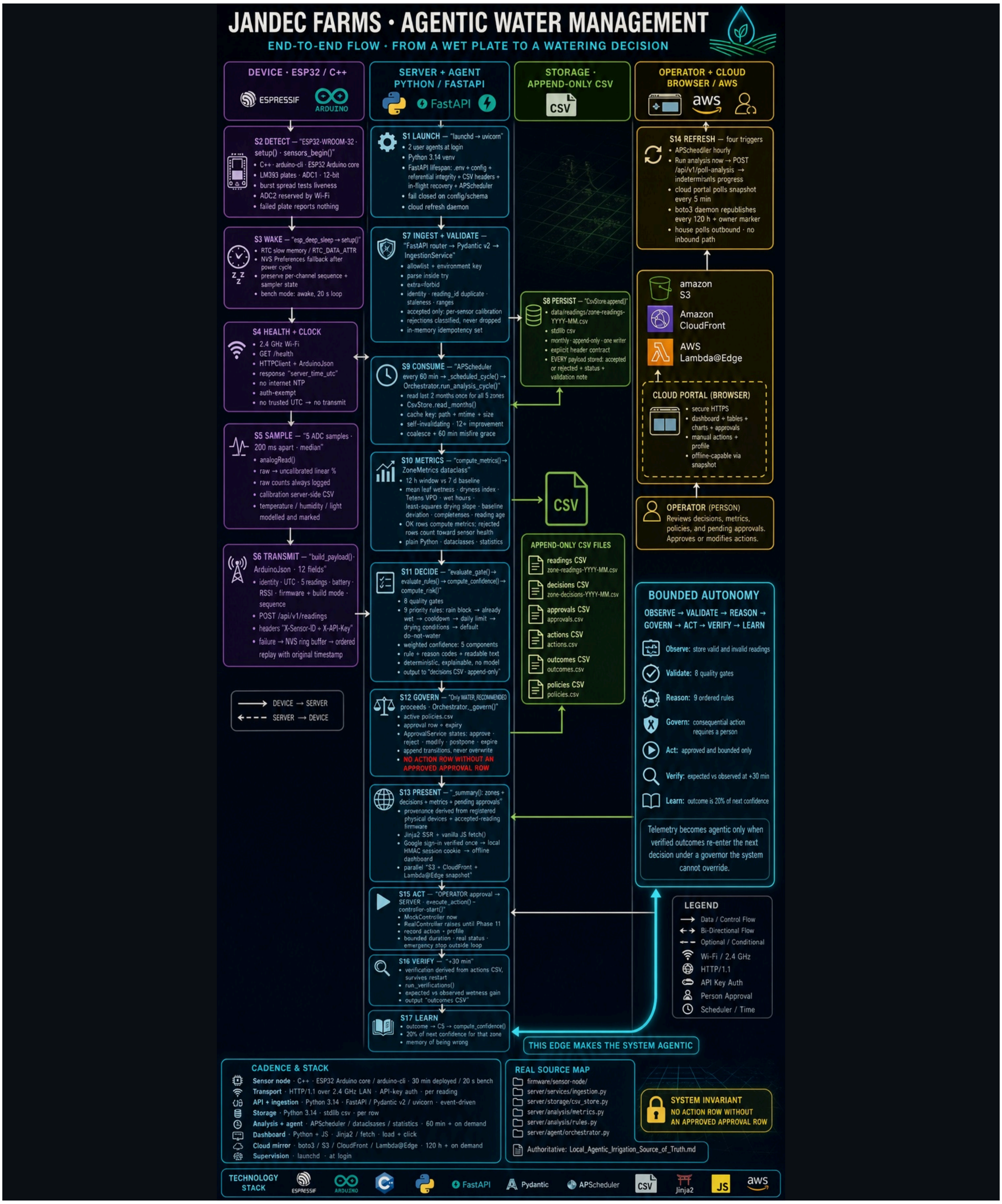
THE HOPS, AND WHAT RUNS AT EACH

HOP	WHAT HAPPENS	BUILT WITH
Soil → node	An LM393 wetness plate in each bed is read on an analogue pin; liveness is judged by whether the reading *varies* within a burst, because a rail-pinned value is exactly what a genuinely dry plate produces.	`ESP32-WROOM-32, C++, Arduino core`
Node → house	Readings are batched, stamped with a firmware version, and posted over the home Wi-Fi with a per-sensor API key; the node buffers when the host is unreachable and never sends a reading older than the staleness bound.	`Wi-Fi, HTTP, ArduinoJson`
Ingest → archive	Every payload is validated and written — including the ones that fail. Rejects are recorded with a reason, not discarded, so a silent sensor is distinguishable from a sensor nobody asked about.	`FastAPI, Pydantic, append-only CSV`
Archive → judgement	**Eight quality gates run before any rule does; if the data cannot support reasoning, the gates end the cycle. Surviving zones go through the rule table, producing a verdict, a confidence and reason codes.**	**`Python 3.14, deterministic rule engine`**

HOP	WHAT HAPPENS	BUILT WITH
Judgement → human	**A recommendation becomes a pending approval. Nothing is watered while it sits there, and an unanswered approval expires rather than escalating.**	**`HITL approval store`**
Human → beds	**An approved action drives the controller within a safety envelope — bounded run time, minimum interval between waterings, daily volume ceiling.**	**`Controller abstraction, mock or real`**
Beds → learning	After a settling window the system re-reads the bed and records whether the expected wetness rise actually occurred. That verdict becomes one of the five inputs to future confidence.	`Outcome ledger`
House → family	A snapshot is published outbound to object storage behind a CDN, gated at the edge to two named Google accounts. The path is one-way: the cloud holds a copy and no control.	`S3, CloudFront, Lambda@Edge`

Architecture — the full end-to-end poster

The whole system in one view: the lanes from soil to operator, every stage between them, and the feedback edge that turns a one-shot recommendation into a loop that improves. Open it full screen to read the stage labels.



Two invariants hold the shape of this picture in place, and neither may be optimised away: **the system keeps working with the internet unplugged**, and **no cloud component can cause water to flow**. Everything in section 6 is measured against those two lines.

04

Governance: five questions, one confidence number

The system does not ask to be trusted. It earns a score every cycle.

● BUILT AND RUNNING

Confidence is calculated fresh for each decision across five governance checks, with every answer recorded alongside the verdict. No gut feel, no “AI says 92%” magic. Five questions go in. One confidence number comes out. And when the evidence gets weaker, the confidence is expected to drop with it.

CHANNEL	THE QUESTION IT ANSWERS	WEIGHT	WHERE IT IS RECORDED
Data completeness	Did we receive what we expected to receive over the last twelve hours?	0.30	`data_completeness_12h`
Freshness	How old is the newest reading? A stale bed is not a wet bed.	0.20	`latest_reading_age_minutes`
Sensor health	Any range violations, battery warnings, or plates behaving like disconnected wire?	0.15	`reading status column`
Baseline stability	How settled is this bed's seven-day pattern? A wandering baseline earns less trust.	0.15	`dryness_baseline_7d`
Outcome history	The learning channel — have this bed's previous recommendations proved correct?	0.20	`action-outcomes ledger`

05

What the system has actually produced

The scoreboard, including the rows that are not yet won — and how far this design stretches before family-scale use runs out of room.

● BUILT AND RUNNING

Everything below is read from the live archive, not from a projection. Where a goal is not yet evidenced, it is marked as not yet evidenced.

3,941 READINGS INGESTED	2,945 CARRYING MEASURED WETNESS	620 DECISIONS RECORDED	9 REJECTS, ALL RECORDED
126 TESTS PASSING	25 ARCHITECTURE DECISIONS		

HOW THE 620 DECISIONS FELL

VERDICT	COUNT	WHAT IT TELLS US
SENSOR_ERROR	303	Mostly the two beds with no plate fitted — the system correctly refusing to invent numbers for hardware that does not exist.
INSUFFICIENT_DATA	108	Early cycles and gaps where the twelve-hour window could not support reasoning.
MONITOR	93	Evidence present, neither case made. The honest middle.
WATER_RECOMMENDED	64	Mean confidence 0.83, range 0.73–0.86. Every one required a human.
DO_NOT_WATER	52	Positively argued against watering, rather than merely not recommending it.

Two-thirds of all cycles ended in the system either declining to advise or advising nothing. For a first season on partial hardware, that distribution is the right shape — an adviser that recommended watering two-thirds of the time would be telling us about its thresholds, not about the soil.

SUCCESS METRICS, HONESTLY SCORED

GOAL	STATUS	EVIDENCE, OR WHAT IS STILL MISSING
-15% water use, sustained	NOT YET EVIDENCED	No pre-automation baseline was ever measured, and actuation is still on the mock controller — so there is no litre count to compare. Establishing that baseline is the first job once real valves are wired.
100% clarity on every AI decision	ACHIEVED	All 620 decision rows carry the rule that fired, the reason codes, the metrics consumed and a plain-language sentence. There is no unexplained verdict in the archive.
Action is 100% human-in-the-loop	ENFORCED	3 actions, 3 approvals granted, 4 still pending, 1 expired unanswered — and expiry caused nothing to happen, which is the behaviour under test. The approval requirement is asserted in code, not merely documented.
Outcomes feed back into the record	WIRED, NOT YET INFORMATIVE	The ledger works end to end and holds 3 outcomes, all recorded as failures — correctly, since a mock valve moves no water. The channel is proven; its learning value arrives with real actuation.
Cheapest defensible build	ACHIEVED	One microcontroller, three plates, a laptop already owned, and a cloud mirror sized to stay inside the free tier. No database, no ML platform, no subscription.

WHY PLAIN MATHEMATICS, AND NO MACHINE-LEARNING MODEL

There is no machine-learning model in the current system, and that is intentional.

With one season of data and a handful of beds, an ML model would be learning more from noise than from truth. It might look sophisticated, but sophistication is not the goal. A decision that can be explained and challenged is far more useful here than one that is merely impressive.

So the system uses simple, established mathematics instead: temperature and humidity to derive vapour pressure, least-squares slope to understand drying rate, standard deviation to measure how stable a baseline really is.

The current win: The system is **not trying to be clever; it is trying to be trustworthy.**

HOW FAR THIS CAN GROW

The system was built for a family garden, and at that scale it is doing what it was meant to do.

- **Beds.** Five beds are registered today, three are actively sensed, and extending coverage is mostly a matter of adding sensors and configuration rather than changing the architecture.
- **Nodes.** Sensor identity is per-channel, not per-board, so a second node in a far corner of the garden joins by adding rows to configuration files. Nothing in the ingest path assumes one device.
- **Storage.**Remains comfortably manageable with append-only files.
- **People.** Family access is intentionally narrow, secure, and sufficient for checking the garden remotely.

> The more interesting scalability question is not how many beds it can monitor. It is how much responsibility should the agent eventually be allowed to carry. Today, Agentic AI owns the observation, reasoning, confidence, recommendation, and learning loop. The human still owns the final action. That boundary is the most important success of the current system.

06

Aspirational commercial roadmap

If Jandec Farms ever grows beyond the family garden, the next steps are not really about adding more automation.

They are about deciding:

- **What authority the system has earned**
- **what architecture must change to support it**
- **What new risks appear the moment this stops being a hobby and starts behaving like a product.**

● FUTURE EVOLUTION PATH

Architectural change

The first major shift would be from recommendation-only to conditional execution.

Today, every watering decision stops at the human. A future version could auto-water only when the evidence is boringly strong: *high confidence, low risk, proven bed history, and volume safely inside a defined ceiling*. Everything else still escalates.

> **That would require three things:**

- A clear confidence floor and risk ceiling;
- Automatic fallback to full HITL when outcomes degrade;
- A one-step kill switch that puts the human back in charge everywhere.

The second architectural shift would be real-time eventing.

A dead sensor, stuck valve, frost event, or watering cycle that never stops should not wait for the next dashboard refresh. That kind of event should broadcast immediately. Hence, Moving to real-time push would also mean persistent connectivity, stronger observability, higher availability expectations, and—less romantically—a recurring cloud bill.

Commercialization & beyond the home garden

A family member may forgive a missed watering recommendation. A customer probably will not.

Commercial AI would need more than working logic. It would need measurable service levels, tenant isolation, identity and access controls, explainable decisions, operational telemetry, audit retention, model or rule versioning, support workflows, and very clear accountability when the system is wrong.

What began as Intelligent irrigation system could eventually become a small decision-intelligence platform for managed growing environments rather than a smarter sprinkler with a dashboard.

Scope growth

The easiest mistake would be to add features because the technology can support them. The better path is to expand only where the system has evidence that it should.

Possible future scope could include weather-aware irrigation, sensor health detection, seasonal baselines, crop-specific behaviour, water-budget optimisation, anomaly detection, multi-site management, and eventually computer vision or learned models once there is enough real data to justify them. Machine learning may absolutely belong here one day.

What began as Intelligent irrigation system could eventually become a small decision-intelligence platform for managed growing environments rather than a smarter sprinkler with a dashboard.

> **The most important roadmap principle:**

- Every new feature must be justified by evidence, not by the technology's capability.

07

Gallery — the physical system

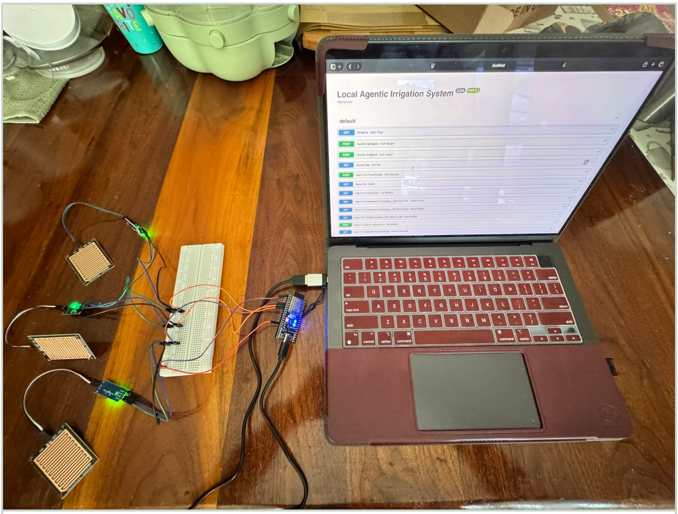
The Phase1 system build and evolution

Everything described in this submission is a physical object in a garden: three wetness plates in wet soil, a microcontroller on a breadboard, a laptop that never sleeps, and five beds that either got watered or did not.

Ginger North

Ginger South

Tomato Patch



The Work Bench

Jandec Farms — Agentic Water Management. Capstone, Mini-MBA in Artificial Intelligence.
Ananth Radhakrishnan · The Learning Ledger

State at submission: 3,941 readings · 620 decisions · 126 tests passing · 25 architecture decisions · leaf wetness measured, temperature and humidity still modelled.