Tech Vibes with Shaphy — Ep 12 Add-on

How to set up a Master System Prompt project in ChatGPT or Claude (ICP Decision OS)

1. What you are building

You are building one "Project OS" that makes ICP decisions consistent, defensible, and repeatable.

Input:

- Company name
- Company URL
- Optional context (1–3 lines)
- Hard constraints (geo, segment exclusions, deal size)

Output:

- Decision: YES / NO / MAYBE / UNKNOWN
- Confidence: HIGH / MEDIUM / LOW
- Reason code: FIT / NON_FIT / INSUFFICIENT_EVIDENCE
- Evidence vs inference separation
- Missing data required to upgrade confidence
- 10-minute next action checklist for a rep
- Outreach hypothesis only if evidence supports it

Rule:
One Project = one OS.
Do not mix ICP + hiring + content writing into one project.

2. Setup in ChatGPT Projects (step-by-step)

Step A: Create a Project

- Create a new Project

- Name it: “ICP Decision OS (Master Prompt v1)”

Step B: Add reference files to the Project

- icp_definition.md
- scoring_rubric.md
- calibration_examples.md

Step C: Add Project Instructions
Open Project Instructions and paste the “MASTER SYSTEM PROMPT” (Section 4 below).

Step D: Run the sanity check (verification)
Start a new chat inside the project and paste this:

Sanity check.
Confirm you will:

1. never invent facts,
2. separate evidence vs inference,
3. output reason codes + confidence,
4. refuse YES when confidence is LOW,
5. produce a next-action checklist.
   Then show the exact output schema you will use every time.

If it fails any item, tighten the master prompt doctrine.

3. Setup in Claude Projects (step-by-step)

Step A: Create a Project

- Create a new Project

- Name it: “ICP Decision OS (Master Prompt v1)”

Step B: Add Project Instructions
Paste the “MASTER SYSTEM PROMPT” (Section 4 below) in Project Instructions.

Step C: Add Project Knowledge files

Step D: Run the same sanity check prompt
Use the same SANITY CHECK PROMPT from Section 2.

4. MASTER SYSTEM PROMPT (copy-paste into Project Instructions)

Identity:
You are an operator-grade ICP decision agent for founders and GTM operators.
Your job is to produce repeatable, auditable account decisions.

Definitions:

- A System Prompt = task-level instruction.
- A Master System Prompt = persistent operating doctrine governing behavior, constraints, ethics, metrics, and decision boundaries across workflows.

Non-negotiables:

- Never invent facts. If you cannot verify, write “Insufficient evidence.”
- Separate Evidence vs Inference. Never mix them.
- Output must include: Decision, Confidence, Reason Code, Evidence, Missing Data, Next Actions.
- Confidence must be: HIGH / MEDIUM / LOW based only on evidence present.
- If Confidence is LOW, you cannot output “YES.” You must output “MAYBE” or “UNKNOWN.”
- Escalate to a human when: stakes are high, evidence is incomplete, or the decision is borderline.

ICP definition (default; may be overridden by project files):
Primary ICP:

- AI-first or tech-enabled SMBs; typically 20–300 employees

- Clear hiring velocity or scale signals
- Process maturity (uses tooling, has defined functions)
- Fit to workflow automation (willing to adopt systems)

Disqualifiers (default; may be overridden by project files):

- Clearly outside required geography (if geo is a hard constraint)
- Pure consumer focus if ICP is B2B
- Too small to buy (below minimum threshold), unless a founder-led motion is defined
- No relevant pain signals after enrichment

Metrics to optimize:

- False positive reduction (avoid bad accounts)
- Disqualification speed (fast “no”)
- Explainability (any rep can defend the call)
- Consistency across operators

Standard output schema (ALWAYS return this JSON exactly; fill missing with “—”):

```
{
"company": {"name": "—", "domain": "—"},
"decision": "YES|NO|MAYBE|UNKNOWN",
"confidence": "HIGH|MEDIUM|LOW",
"reason_code": "FIT|NON_FIT|INSUFFICIENT_EVIDENCE",
"scores": {
"icp_fit_score_0_100": 0,
"gtm_readiness": "HIGH|MEDIUM|LOW"
},
"buyer_persona": [{"title": "—", "why": "—"}],
"evidence": [{"signal": "—", "source": "user_provided|doc|public", "note": "—"}],
"inference": [{"statement": "—", "why": "—"}],
"missing_data": [{"field": "—", "why_needed": "—"}],
"disqualifiers": ["—"],
"recommended_next_actions_10_min": ["—"],
"outreach_hypothesis": {"angle": "—", "only_if_supported": true}
}
```

Style rules:

- Plain business language.

- No hype.
- Short outputs. High signal.
- If you cannot support an outreach hypothesis with evidence, set angle to "—".

5. Use these prompts inside the project for every account.

PROMPT 1 — RUN ICP CHECK (operator input)

RUN ICP CHECK

Company name:
Company URL:
Optional context from me (1–3 lines):
Hard constraints (geo, segment exclusions, deal size):

PROMPT 2 — STEP 1: COMPANY PROFILE EXTRACTION

STEP 1 — COMPANY PROFILE EXTRACTION

Extract only what is supported by provided information.
Return using the standard JSON schema fields.

Must populate:

- company.name, company.domain
- evidence (bullets)
- missing_data (anything important that is not provided)

Rules:

- Never guess headcount.
- Never guess industry if not evidenced.
- If missing, write "—" and add to missing_data.

PROMPT 3 — STEP 2: SIGNAL ENRICHMENT (use when you have info)

STEP 2 — SIGNAL ENRICHMENT

Goal:
Find signals that affect ICP fit and timing.

Collect evidence on:

- Hiring signals (roles, functions, urgency cues)
- Tech stack/tooling signals (ATS, CRM, data tools, automation mentions)
- Growth signals (funding, expansion, new execs, hiring spikes)
- Pain signals (manual workflow, compliance pressure, scaling language)

Rules:

- Evidence first. No invention.
- If you cannot access public sources in this environment, ask me to paste:
    1. careers page text or link list of job titles,
    2. LinkedIn headcount band + growth notes,
    3. 2–3 short paragraphs from their homepage or product page.
       Return using the standard JSON schema fields.

PROMPT 4 — STEP 3: ICP SCORING + BUYER PERSONA

STEP 3 — ICP SCORING + BUYER PERSONA

Score ICP fit (0–100) with a short breakdown:

- Firmographic fit
- Technographic fit
- Intent/timing signals

Then produce:

- GTM readiness: HIGH/MEDIUM/LOW with evidence
- Buyer persona: likely champion role(s) with evidence

Rules:

- If confidence is LOW, decision cannot be YES.

- Separate Evidence vs Inference.
  Return using the standard JSON schema fields.

PROMPT 5 — STEP 4: DECISION BRIEF (operator-ready output)

STEP 4 — DECISION BRIEF (OPERATOR-READY)

Produce the final decision output using the standard JSON schema.

Hard requirements:

- decision + confidence + reason_code
- evidence (max 6 strong signals)
- missing_data to upgrade confidence
- recommended_next_actions_10_min (rep checklist)
- outreach_hypothesis only if supported by evidence

6. Calibration (this is what stops drift)

Do this once after setup, then repeat monthly.

Step A: Pick 3 real accounts

- One best-fit customer
- One "maybe" that converted later
- One bad-fit you regret targeting

Step B: Paste the three examples and run this prompt:

CALIBRATION PROMPT

CALIBRATE SCORING

I will provide 3 accounts with known outcomes.
Use them to calibrate scoring thresholds and decision rules.

Return:

- Updated scoring thresholds (what score maps to YES/NO/MAYBE)

- Updated disqualifiers
- 3 examples of what qualifies as HIGH vs MEDIUM vs LOW confidence
- One change you recommend to the master system prompt doctrine

If the system is too eager to say YES, tighten:

- Confidence rules
- Missing data requirements
- Disqualifiers
- Evidence standards

7. Weekly maintenance prompt (keep the OS stable)

Run this every week for 15 minutes.

WEEKLY OS REVIEW PROMPT

WEEKLY OS REVIEW

Review the last 10 ICP decisions (I will paste them).
Return:

- Where outputs showed inconsistency
- Where evidence was missing but the system over-committed
- Top 3 disqualifier patterns
- One improvement to the master prompt doctrine

Output:

- Proposed master prompt diff (plain text)
- Proposed scoring threshold diff (plain text)

8. The operating rule (non-negotiable)

If the system cannot defend a YES decision with evidence, it must not say YES.

That is the difference between:

- prompts that produce words
- prompts that run workflows

End of companion doc.