DAVEWANG.AI

The Hiring Intelligence Prompt

A workflow for turning public job postings into an investor signal layer

> **How to use it.** Paste this into Claude Code and replace [COMPANY] with the company you want to monitor. The workflow is built for public job data, raw-file auditability, requisition-level counting, theme classification, and a tracker you can rerun over time.

You are an equities analyst and data engineer.

I want to build a hiring-intelligence tracker for [COMPANY].

Use only public job-posting data and public source pages.

Step 1. Map the public job source.

- Identify the search endpoint or pages.
- Capture pagination logic.
- Identify the stable posting ID or requisition ID.
- Do not rely on visible page text if structured payloads exist.

Step 2. Collect raw data.

- Save every raw search result page.
- Save every raw job-detail payload.
- Keep source URLs and capture timestamps.
- Stop if terms or access rules make collection inappropriate.

Step 3. Normalize the data.

- Count one unique requisition as one investor-counted role.
- Split locations into a separate table.
- Do not double-count multi-location roles.
- Separate retail, support, evergreen, and pipeline roles from corporate or technical roles.

Step 4. Classify the roles.

- Use the job title as the strongest signal.
- Use team or department as the second strongest signal.
- Use qualifications and description as supporting context.

- Assign investor themes with confidence scores.
- Flag low-confidence strategic labels for manual review.

Step 5. Build the investor readout.

- Show the current role mix by theme, function, geography, and seniority.
- Identify the highest-signal individual postings worth reading.
- Separate facts from inference.
- Do not claim that postings prove strategy, budget, launch timing, or revenue impact.

Step 6. Design the tracker.

- Store first-seen date, last-seen date, active status, temporary absences, closures, and re-openings.
- Treat the first snapshot as composition.
- Treat repeated captures as velocity.

Output a Markdown report, a dashboard, raw data files, processed tables, QA checks, and a watchlist of interesting postings.

Why it is structured this way.

- **Raw-first collection** keeps the research auditable.
- **Requisition-level counting** prevents multi-location roles from inflating demand.
- **Retail separation** prevents store and pipeline jobs from drowning technical hiring signals.
- **Title-weighted classification** keeps generic boilerplate from driving the read.
- **Confidence scoring** keeps weak matches from becoming fake certainty.
- **Watchlist extraction** forces the model to surface individual titles worth manual review.
- **Tracker-ready schema** turns one snapshot into a future velocity monitor.

> **More AI x finance workflows:** davewang.ai

Research and educational analysis only. Not investment advice. Public recruiting signals should be treated as directional evidence, not proof of hiring, strategy, budget, product timing, or revenue impact.

davewang.ai