

Vibe Coding Take-Home: VC Intelligence Interface + Live Enrichment

Build a functional VC discovery interface (Harmonic-style workflows) and integrate a real live data pull using an AI scraping/enrichment tool. Creativity is encouraged. Timebox: **7-8 hours**. Use Lovable / Google AI Studio / Cursor / v0 / Replit or any vibe-coding platform to ship fast.

Deliverable	Deployed app URL + GitHub repo + README (env vars + setup).
Core test	Can you ship a usable interface AND wire real enrichment safely (server-side) into the UI?
Deploy	Vercel (preferred) or Netlify (acceptable).
Data	Seed with mock companies JSON; enrich from real public pages on demand.

Creative brief (aim for "real product" quality)

- Think in workflows: discover -> open profile -> enrich -> take action (save, note, follow, export).
- Make it feel premium: clean typography, spacing, responsive layout, fast interactions.
- Be opinionated: pick a design system and keep it consistent.

Minimum app scope (must-have)

- **App shell**: sidebar navigation + global search.
- **/companies**: search + filters + results table (sortable + pagination).
- **/companies/[id]**: company profile with overview, signals timeline, notes, save-to-list.
- **/lists**: create lists, add/remove companies, export list (CSV/JSON). Persist in localStorage.
- **/saved**: save and re-run searches (persist in localStorage).
- **Live enrichment**: on the profile, click "Enrich" to fetch real public website content via AI scrape and display extracted fields.

Live enrichment spec (what to show)

- **Summary** (1-2 sentences).
- **What they do** (3-6 bullets).
- **Keywords** (5-10).
- **Derived signals** (2-4 signals inferred from pages, e.g., careers page exists, recent blog post, changelog present).
- **Sources**: list the exact URLs scraped, with timestamp.

Important note (keys + production)

Implement enrichment through a **server-side** endpoint (e.g., Next.js `/api/enrich`) so API keys are never exposed in the browser. Use environment variables and document setup in README.

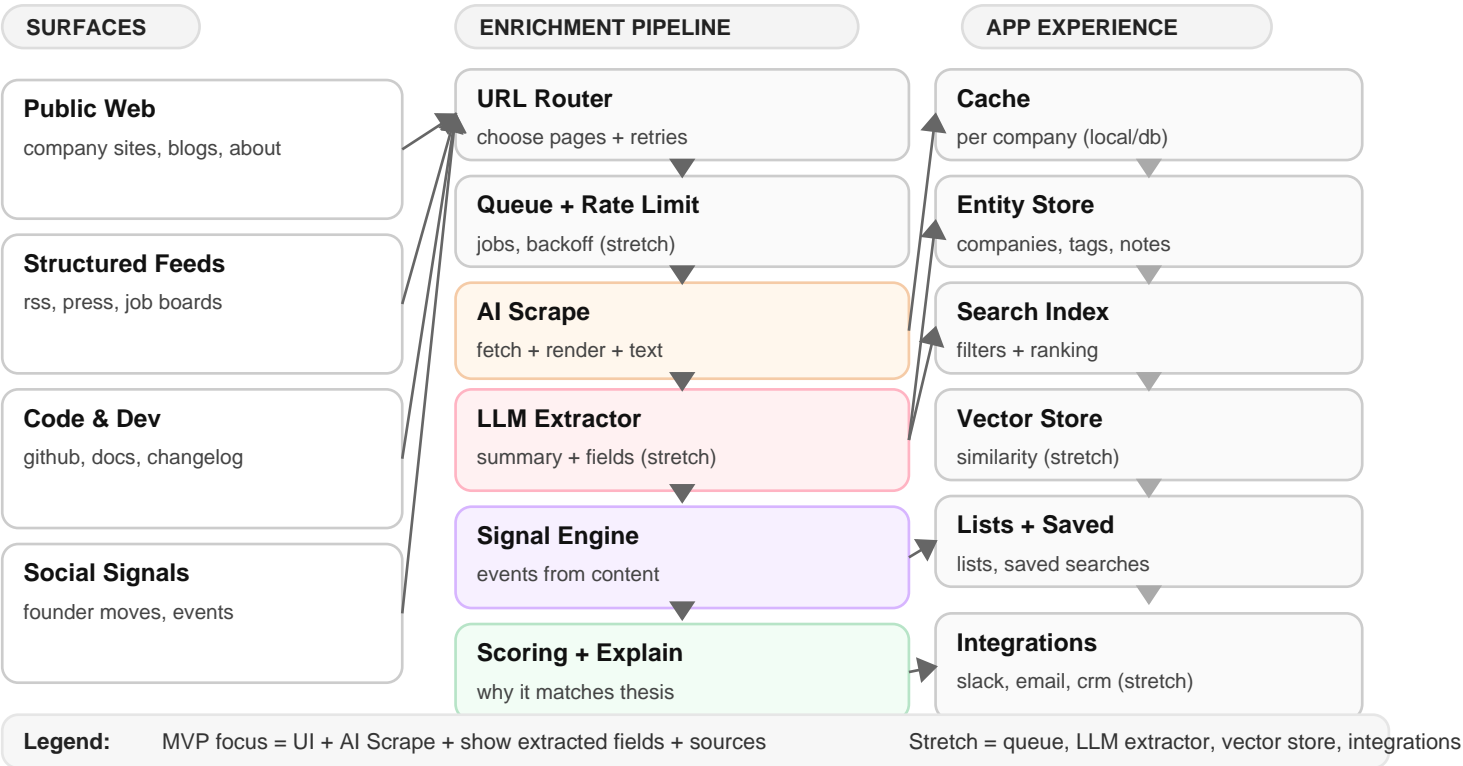
Use public pages only. Do not attempt to evade access controls.

North Star diagram (aspirational)

This diagram is intentionally more sophisticated than what is achievable in 8 hours. Use it as a target. Your MVP only needs one working enrichment path end-to-end.

North Star Architecture + Product Flow (aspirational)

MVP: implement the UI + one live enrichment path end-to-end. Everything else is a stretch target.



Suggested time plan (fits 8 hours)

0-1h	Scaffold app + layout + routing. Add a small mock dataset.
1-3h	Companies page: search + filters + table + pagination.
3-5h	Company profile: overview, signals, notes, save-to-list, saved searches.
5-7h	Live enrichment: server API route -> AI scrape -> parse fields -> cache + UI states.
7-8h	Polish + deploy + README + quick QA.

Evaluation focus

- **Interface quality:** feels like a real tool; fast, clean, usable.
- **Live enrichment:** works reliably in production; good loading/error states; shows sources; caches results.
- **Engineering:** keys are safe; code is readable; state is handled well; no obvious footguns.
- **Creativity:** thoughtful UX choices, power-user touches (bulk actions, keyboard shortcuts, etc.) without breaking scope.