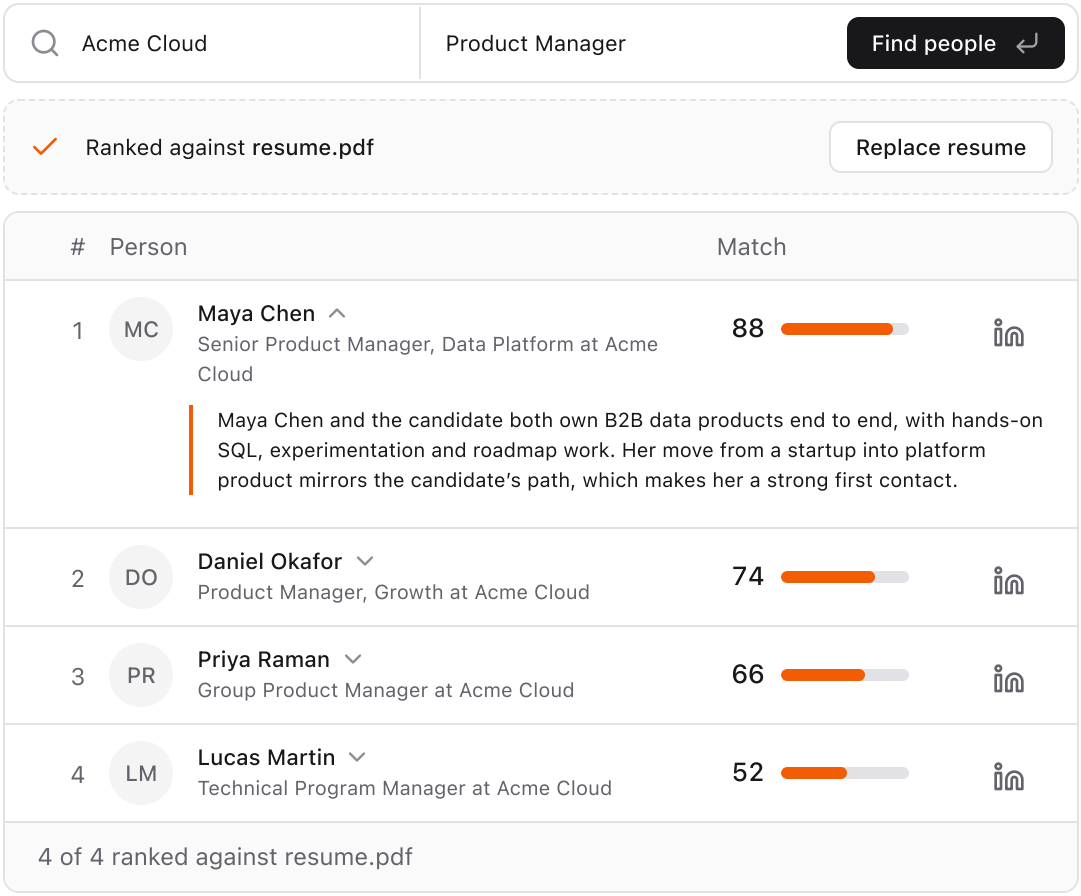
LinkedIn Scout

A tool I built to find who to reach out to at a company, ranked by how well their background matches my resume.

Rishav Dutta, linkedin.com/in/duttarishav
Live app: rishav-linkedin-scout.vercel.app
Code: github.com/rishav-dutta/linkedin-scout
Write-up: rishav-dutta.github.io/linkedin-scout



People at a company, ranked against a resume, with the reason for each score. Sample data.

Role	Timeline	Built with
Solo: product, design and engineering	February 2026 to now	React, n8n, Gemini, Supabase

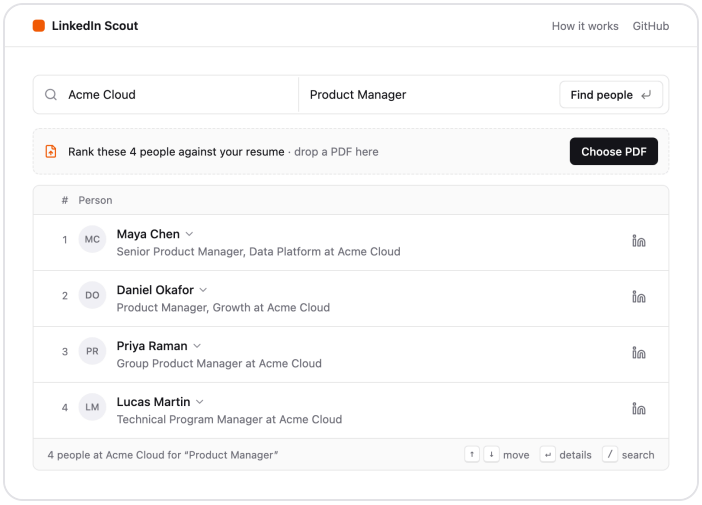
Why I built it

During my MBA job search, networking took most of my time. For every company on my list I searched LinkedIn for people in the team I wanted to join, opened profile after profile, and tried to work out whose background was close enough to mine that a message would get a reply. It was slow, and I was mostly guessing.

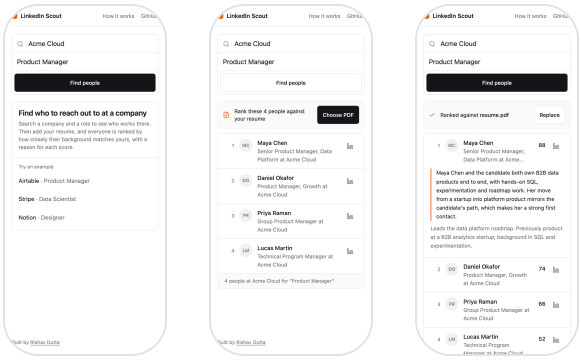
I wanted to type in a company and a role and get back a short list of people, in order, with a reason for each one. So I built that.

What it does

1. Enter a company and a role, for example “Airtable” and “Product Manager”.
2. Matching LinkedIn profiles show up in about a second, and their photos and full profiles fill in over the next few seconds.
3. Upload your resume as a PDF. Gemini scores each person from 0 to 100 against it, writes a sentence or two on why, and the list re-sorts by score.



Right after a search: the people it found, before a resume is added. Sample data.



The same steps on a phone. Sample data.

How it works

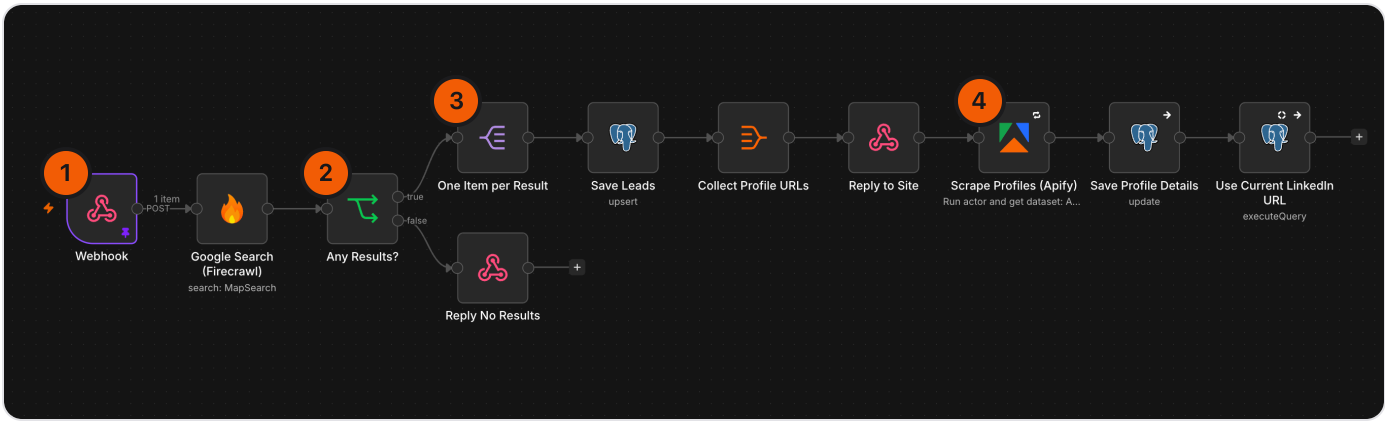
There is no custom backend server. The site is a React app on Vercel, and the work happens in two n8n workflows that I self-host. The browser never talks to n8n directly; every request goes through a small Supabase function first.

Browser	Gatekeeper	Workflows	Services
React app on Vercel	Supabase Edge Function: checks input, rate limits	n8n, self-hosted on Oracle Cloud	Firecrawl to search, Apify for profiles, Gemini to score

Results are saved to Supabase Postgres, and the site reads them back from there.

Finding people

This workflow runs when someone presses Find people.



The search workflow in n8n.

1. Firecrawl searches Google for LinkedIn profile pages that match the company, the role and “United States”.

   Webhook, Google Search (Firecrawl)

2. If the search finds nobody, the workflow replies “no results” and stops, so the site can suggest checking the spelling or trying a broader role.

   Any Results?, Reply No Results

3. Otherwise it saves each person and replies to the site before fetching any profiles, so the first people appear in about a second.

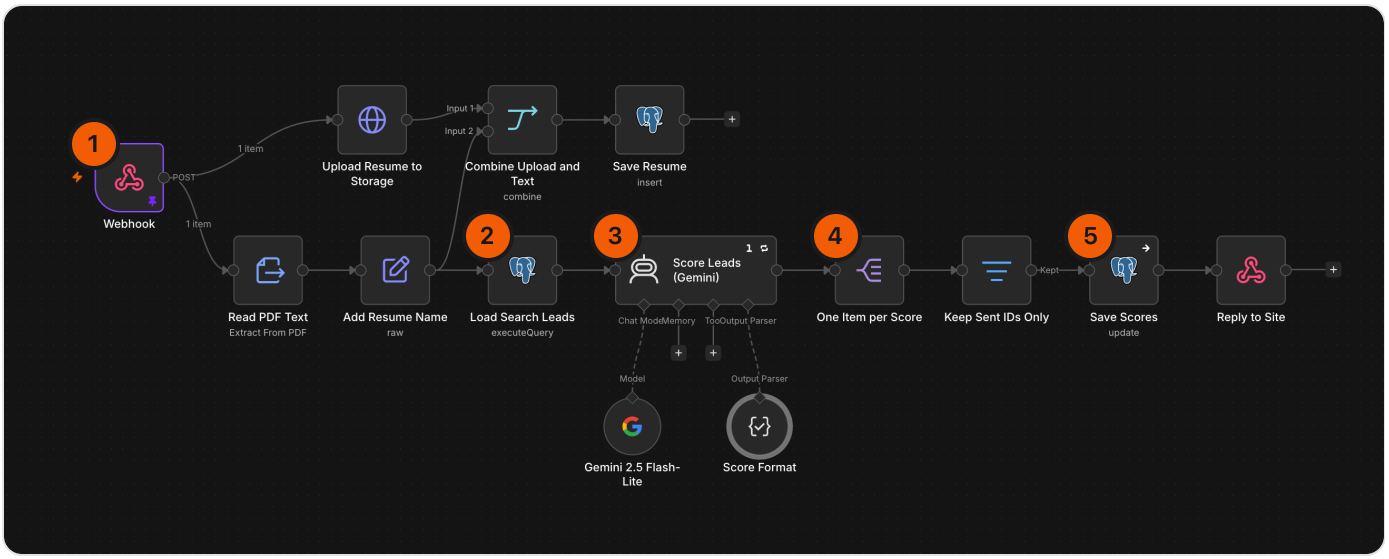
   One Item per Result, Save Leads, Collect Profile URLs, Reply to Site

4. Apify then fetches the full profiles in the background, which takes 7 to 35 seconds, and each one is saved with its person. If a profile’s address has changed since the search, the current address is saved too.

   Scrape Profiles (Apify), Save Profile Details, Use Current LinkedIn URL

Scoring a resume

This one runs when someone uploads a resume.



The scoring workflow in n8n.

1. The resume goes two ways at once. One branch uploads the PDF to private storage, the other pulls out its text, and then both are saved together as a record of the resume.

   Upload Resume to Storage, Read PDF Text, Add Resume Name, Combine Upload and Text, Save Resume

2. It loads everyone from the search whose full profile has arrived. People without one can't be scored.

   Load Search Leads

3. Gemini scores them all in one call. Each profile is labelled with its database ID, and the reply has to follow a fixed format: an ID, a score from 0 to 100 and a reason for each person. Matching on IDs means Gemini never has to copy a URL back. Flash-Lite sometimes returns an empty reply, so this step retries up to five times, two seconds apart.

   Score Leads (Gemini), Gemini 2.5 Flash-Lite, Score Format

4. Any score for an ID that wasn't sent is dropped, so Gemini can't add someone who isn't in the search.

   One Item per Score, Keep Sent IDs Only

5. It saves the scores and replies to the site, and the site re-sorts the list.

   Save Scores, Reply to Site

Numbers

As of October 2026, including my own testing:

95	searches, across 65 companies
363	profiles found
91	resumes scored
133	visitors in four weeks (Vercel Analytics, April 12 to May 10, 2026)
~1 s	until the first people appear, down from 7 to 35 seconds
$0	a month to run

What I left out

There are no accounts and no saved search history. I left both out so I could ship the whole flow end to end first. Searches cover US profiles and return up to four people each, which keeps scraping costs low.

Stack

Frontend	React 18, TypeScript, Vite, Tailwind CSS, on Vercel
Gateway	Supabase Edge Function (Deno)
Workflows	n8n, self-hosted in Docker on Oracle Cloud
AI	Google Gemini 2.5 Flash-Lite, through n8n's AI Agent with structured output
Search and profiles	Firecrawl (web search), Apify (LinkedIn profile scraper)
Data	Supabase Postgres and private storage, with versioned migrations
Checks	GitHub Actions: lint, type check and build on every pull request

The code, database migrations and both n8n workflows are in the GitHub repo, with steps to run your own copy.

About me

I'm Rishav Dutta, AI Product Owner at SquadLocker. I have 6+ years of product management experience across mobility, logistics and e-commerce, at Zoomcar, Blackbuck and IUVA Foods, and an MBA from Simon Business School at the University of Rochester (2026). I built LinkedIn Scout on my own, as product manager, designer and developer.