

Evgeniy Poznyak

Product Engineer

518-772-8217 | dev.4.remote@gmail.com | Richfield Springs, NY 13439 | US Citizen

SUMMARY

Product engineer who ships. Jobbi: 1,537 registered users and paying subscribers, all organic, nothing spent on ads. Its auto-apply engine filed 1,183 job applications end to end for my own job search. Softery.io: seven free tools live in production, six of them AI-powered. Three iOS apps live in the App Store. 11 years of engineering: three large tech enterprises in a row (Cox Automotive, Motorola Solutions, Extreme Networks), my own products with paying users, and an AI workflow I run daily on everything. I own the whole widget: architecture, frontend, backend, AI orchestration, App Store, growth. The AI is invisible. If the user can see it, the product failed.

SHIPPED AI PRODUCTS

Jobbi | AI Resume Tailoring Platform | 2025 to Present

Web: <https://jobbi.app> | iOS: <https://jobbi.app/ios>

Upload a resume. Paste a job link. Get a tailored version in seconds. No prompting. No setup. 1,537 registered users and paying subscribers. Nothing spent on ads. Organic growth only.

- Built every layer end to end: iOS (SwiftUI, SwiftData, Combine), Web (Next.js 16, React 19, TypeScript), backend (Supabase Postgres, Auth, Edge Functions, Deno).
- Multi-model LLM orchestration across Google Gemini, Anthropic Claude, and OpenAI GPT behind a unified service layer. Users bring their own keys. The system picks the right provider per task. Users never see the orchestration.
- **Privacy by architecture, not by policy.** Custom placeholder system strips PII before any AI call and restores it client-side. Personal data never touches third-party AI APIs. Users chose Jobbi for this reason.
- Apple Vision for OCR, PDFKit for parsing, WebKit for PDF export. RevenueCat syncs subscriptions across iOS (StoreKit 2) and web (Stripe).
- Full product lifecycle: App Store submission, production monitoring, iterative releases, content marketing, SEO, outreach to job search communities.

Jobbi Auto-Apply | Application Engine | 2026

Job hunting is mostly retyping. The auto-apply engine behind Jobbi does that part for you. You see a job, you press Apply, you see what happened. Underneath, it opens the employer's real application form, fills in the identity and legal fields from what you told it once, answers the written questions using only your own resume, attaches a resume written for that posting, and sends it.

- Pointed it at my own job search first. It filed 1,183 applications for me, end to end, 52 waves over 18 days, and 349 of the 357 that reached a form went through.
- Held the model bill to \$3.20 for a month of running, 6,168 calls and 27 million tokens, by grouping failures by cause instead of by employer and fixing each one with a test that failed on the old code.
- Designed around what the model is not allowed to do: it only ever sees the applicant's own resume, so it cannot invent a degree or a job, and equal opportunity questions never reach a model at all.
- Shipped a readiness check that refuses an application before the browser opens and names the missing field, so a run that cannot succeed answers in seconds and costs the person nothing.
- Node and TypeScript service driving a real browser, Postgres as the work queue with claim locking so two workers can never double file, charge on success billing, Docker on a VPS.

Foreclosure Radar | Forced Sale Intelligence | 2026

Foreclosure Radar tracks every property heading to a forced sale in Pennsylvania and New York. Whoever bids on one of those houses is deciding almost blind, so instead of printing another list the product does the homework on each lot: what it is worth against the price, a written read on its condition, comparable sales, which debts stay with the house, and who still lives there.

- Covers 3,812 properties across 48 auctions in 18 Pennsylvania and 8 New York counties.
- Corrects every Pennsylvania assessment through the state ratio table before anyone sees it, because one county still values houses on a 1961 base year, where the county figure is about 7 percent of market value.
- Built the restraint in on purpose: a court record only attaches to a property when an address matches, so on one sale list the search returned 1,379 name matches and none were kept. Family, support and custody cases are refused outright.
- Took the heaviest page from 2.4 seconds to 0.34 by replacing a full catalog build with a single record query.
- Next.js 16, React 19, Postgres with Drizzle, a headless browser pipeline, and a vision pass that ranks and discards property photos that show no building.

Softery.io | Free Tools Platform for Job Seekers | 2025

<https://softery.io>

Seven free tools live in production, six of them AI-powered: ATS Resume Scanner, Resume Roaster, LinkedIn Roaster with shareable cringe-score results and dynamic OG images, Resignation Letter Generator, Invoice Generator, Markdown to PDF Converter with live preview, and a Word and Character Counter with a live meter for every platform limit, built entirely client side.

- Built on Next.js 16, React 19, TypeScript, Tailwind CSS, and 30+ Radix UI components. Supabase for auth and storage. Deno Edge Functions for AI processing. Dynamic OG image generation on the Edge runtime.
- Blog with admin panel, rich text editing, image cropping. Resend email integration. TypeScript across client and server.

Prompt Crafter | AI Prompt Marketplace | 2025

<https://prompt.softery.io>

AI prompt marketplace with Stripe checkout and subscriptions. Supabase backend with Postgres and RLS. Semantic search powered by OpenAI embeddings (text-embedding-3-small) stored in pgvector columns. Dual-provider content moderation across OpenAI Moderation and DeepSeek Chat, toggleable by environment.

iOS Apps

Three live in the App Store today, all published under Poznyak & Co. LLC.

- **Jobbi: AI Resume** (2026): <https://apps.apple.com/us/app/jobbi-ai-resume/id6757321088> | SwiftUI, SwiftData and Combine. The iOS side of the Jobbi platform above.
- **FaxGo** (2026): <https://apps.apple.com/us/app/faxgo/id6796798069> | Send a fax from an iPhone and pay per fax instead of by the month.
- **Timesseek** (2024): <https://apps.apple.com/us/app/timesseek/id6478344509> | React Native time tracker with multi-page PDF reports and invoices.

Two earlier apps shipped and are no longer listed:

- **Bloopsik** (2021 to 2022): SwiftUI Bluetooth device finder and reminder.
- **DL Appt Seeker** (2020 to 2021): Swift iOS app with a NestJS backend for Texas DPS appointment notifications.

PROFESSIONAL EXPERIENCE

Founder and Product Engineer | Softery.io | 2025 to Present

Full-time, Remote

Shipped four products end to end. Jobbi reached 1,537 registered users and paying subscribers with nothing spent on ads.

- Filed 1,183 job applications end to end with an engine I built and pointed at my own search first, 52 waves in 18 days, and 349 of the 357 applications that reached a form went through.

- Shipped Jobbi, the product that runs on it, on web and iOS: 1,537 registered users and paying subscribers, all from organic traffic with nothing spent on ads.
- Kept the model bill at \$3.20 for a month of running, 6,168 calls and 27 million tokens, by grouping failures by cause instead of by employer and fixing each one with a test that failed on the old code.
- Shipped a check that refuses an application before the browser opens and names the missing field, so a run that cannot succeed answers in seconds and costs the person nothing.
- Built Foreclosure Radar, which tells a buyer what a house going to a forced sale is actually worth before they bid, across 3,812 properties in 18 Pennsylvania and 8 New York counties.
- Shipped FaxGo to the App Store, an iPhone app that sends a fax and charges you for the fax instead of the month.
- Built every layer of Jobbi: iOS (SwiftUI), Web (Next.js 16, React 19), backend (Supabase Edge Functions, Deno), AI orchestration across Gemini, Claude, and GPT, App Store submission, production monitoring, growth marketing.
- Owned the full product lifecycle: architecture, customer feedback loops, content marketing, SEO, outreach to job search communities.

Staff Software Engineer | Extreme Networks | 2022 to 2026

Full-time, Remote

Owned customer-facing, revenue-carrying applications at a publicly traded cloud networking company. Angular and React in parallel, four years of sprint-by-sprint improvements built on direct feedback from the company's most senior product stakeholder.

- Took over a legacy AngularJS application customers paid for: stabilized it, fixed production issues, earned consistently strong customer satisfaction scores, then drove the migration to modern Angular (signals) and React micro front-ends over GraphQL.
- Every sprint the team shipped improvements in both stacks, then demoed the finished work live. Four years of that loop produced consistently outstanding user feedback.
- Acted as the key engineer on the team: led two senior engineers day to day (code reviews, task assignment, onboarding, performance reviews), ran Scrum twice a week, mentored distributed teams across time zones. Applied The Goal (Eliyahu Goldratt) to identify and resolve delivery constraints.
- Rebuilt the real-time tracking map with Azure Maps: intelligent clustering that breaks into smaller clusters on zoom, individual points at full zoom, live status updates over SignalR.
- Built an operations dashboard with AG-Grid Enterprise: custom renderers, custom actions, infinite scroll with virtualization, persistent per-user settings that load and apply automatically each session. TDD with Jest and React Testing Library throughout.
- Led the migration from Material Design (MUI) to Tailwind CSS across the frontend. Smaller bundles, consistent design tokens.
- Built a Shimmer Loader component adopted by product teams across the organization. Contributed to the company-wide Angular UI component library built on signals.
- Developed Node.js BFF services integrating multiple API systems. Splunk logging for full observability. JWT based authentication for secure data access.
- Set up feature management with Harness, including a custom feature flag system prior to adoption. Established the quality foundation: Husky pre-commit hooks and GitHub Actions CI/CD with mandatory unit tests, so every commit had to pass before merging.
- Shipped with GraphQL APIs, AWS, Kubernetes, Docker, TypeScript. Delivered features using BDD so QA could reuse the same behavioral patterns for Jira ticket testing. Applied Clean Code (Robert C. Martin) and Refactoring (Martin Fowler) across code reviews.
- Ran an AI native workflow: Claude Code with custom engineering skills, Cursor AI, and GitHub Copilot integrated into the team code review process.

Senior Fullstack Engineer | Motorola Solutions | 2020 to 2022

Full-time, Remote

Shipped AI-powered public safety tools used daily by first responders in schools, police stations, and highways.

- Built React micro frontends and Node.js services powering license plate recognition, person tracking, and people counting. The Next.js wrapper held the micro frontends together; I owned the wrapper and multiple micro frontends.
- Led the Remote Site Upgrade project end to end: React frontend, Node.js API, cross-functional coordination across C, Node.js, and ML engineers, Swagger documentation. Shipped it as an MVP when no spec or alignment existed; once teams saw it working, adoption accelerated. Became the most valuable feature during COVID when field upgrades could not happen in person.
- Created a TCP connection health checker with auto-reconnection, similar to Gmail's offline mode, for field reliability in the hands of first responders.
- Integrated SignalR for real-time device synchronization. All deployments to Azure Cloud.
- Applied Domain Driven Design principles to structure service boundaries for the public safety domain. GitHub Copilot used as an assistive coding tool during this role.

Fullstack Engineer | Cox Automotive (originally F&I Express) | 2017 to 2020

Full-time, Remote

Startup-to-acquisition experience. Shipped features on Express Recoveries, the flagship product that drove 80% of company profits and led to F&I Express being acquired by Cox Automotive in 2018.

- Helped transition the stack from AngularJS and custom PHP to a React microservices architecture under a new CTO.
- Built custom data transfer systems for banking partners. Every bank had different security and formatting requirements. Sensitive financial data, zero room for error. Applied Design Patterns (Gang of Four) to keep integration modules maintainable and extensible.
- One Angular 6 application was showcased at the NADA conference in Las Vegas, attended by car dealers from across the U.S.
- 80%+ Angular unit test coverage with Karma. Integrated Okta and Swagger.

Full-Stack Engineer | Pacific Northern | 2015 to 2017

Full-time, Remote

Built multi-site platforms and internal tools with Angular and PHP. Delivered a 35% increase in functionality and performance across the platform.

- Migrated WordPress legacy code to Laravel and Angular 4, including a custom video platform for internal training with progress saving.
- Published three WordPress plugins to the official WordPress store, including a Microsoft Active Directory integration with a cron-driven sync.
- Integrated Facebook, Google, and Active Directory APIs across multiple production sites.

AI TOOLS AND PRODUCTIVITY ENHANCEMENT

AI is the spring technology of this era. I ride it daily, not as a novelty, but as the core of how I build and ship products. Three shipped AI products prove it: Jobbi (multi-provider orchestration across Gemini, Claude, and GPT), Prompt Crafter (pgvector semantic search with OpenAI embeddings), Softery.io (seven free tools, six of them AI-powered). Production systems serving real users, not experiments.

Claude Code with custom-built skills is my primary AI development environment. I build custom skills for polishing software, delivering the product lifecycle, designing architecture, validating ideas, researching user behavior, and discovering information on the web and in documentation, distilled from canonical engineering books (Clean Code, Refactoring, Design Patterns, Domain Driven Design). The quality of AI output depends on how well you constrain its reasoning space. Generic AI produces plausible garbage. AI constrained by proven frameworks produces the right answer.

From Stack Overflow to AI: the tool changed, the rule did not. I learned to code back in 2015 and 2016 the Stack Overflow way, with one rule: never ship a line you cannot explain. AI is the same discipline at a different scale. I direct the model, and I own the understanding of every line it produces. Everything I ship is covered by unit tests and written test-first. TDD is non-negotiable; it is what keeps software reliable instead of fragile.

MCP servers extend the workflow: Notion as the single source of truth, Supabase, Mixpanel, and Gmail connected through MCP, plus custom MCP servers I built to wire my workflows directly into the models (Claude Fable 5, Claude Opus 4.8).

Cursor AI is the secondary development environment. **GitHub Copilot** was used at Motorola Solutions (2020 to 2022) and in the team code review process at Extreme Networks.

Model selection is a product decision, not a technology choice. Claude Fable 5 for deep reasoning and architecture. Claude Opus 4.8 for rapid coding and refactoring. Gemini for analytical workloads. GPT for specific tasks. Every output is reviewed. Tests, contracts, and maintainability stay non-negotiable.

TECHNICAL SKILLS

- **Frontend:** React, Next.js 16, Angular (Signals), TypeScript, JavaScript, Tailwind CSS, Radix UI / shadcn/ui, Vite, React Query (TanStack), React Hook Form, Zod, Redux, Zustand, AG-Grid Enterprise, GraphQL, Framer Motion
- **Mobile:** React Native, SwiftUI, SwiftData, Combine, StoreKit 2, RevenueCat, CoreBluetooth, Apple Vision (OCR), PDFKit
- **Backend:** Node.js, Nest.js, Express.js, Python, Flask, Deno, PHP (Symfony, Laravel), REST APIs, GraphQL, WebSockets, Socket.IO, SignalR, BFF Architecture, Microservices
- **AI/ML:** Google Gemini API, OpenAI API (GPT, Embeddings, Moderation), Anthropic Claude API, pgvector (semantic search), Multi-model orchestration, Content moderation, PII anonymization pipelines
- **Cloud Platforms:** Microsoft Azure (App Services, Entra ID, Azure SQL, Azure Maps, Azure DevOps), AWS (EC2, S3, Lambda, API Gateway), Google Cloud (Gemini API, Cloud Functions, Firebase)
- **Databases and BaaS:** PostgreSQL, SQL Server, Azure SQL, Supabase (Postgres, Auth, Edge Functions, Storage, Realtime), MongoDB Atlas, pgvector
- **DevOps and Observability:** Docker, Kubernetes, GitHub Actions, Harness CI/CD, Splunk, Mixpanel, RevenueCat
- **Payments:** Stripe (Checkout, Connect, Webhooks), RevenueCat (iOS and Web subscription sync, StoreKit 2)
- **AI Development Environment:** Claude Code with custom-built skills for software polishing, product lifecycle delivery, architecture design, idea validation, user behavior research, and web/documentation discovery (primary), MCP servers (Notion as single source of truth, Gmail, Google Calendar, RevenueCat, Supabase, Mixpanel, Stripe, plus custom-built MCP servers), Models (Claude Fable 5, Claude Opus 4.8, Gemini, GPT), Phased Coding Protocol (DDD, Design Patterns, Clean Code + TDD, Refactoring, Polish), Cursor AI (secondary), GitHub Copilot (Motorola Solutions; code review process at Extreme Networks)

EDUCATION

- **Bachelor of Science, Computer Science** | Institute of Technology, Tambov, Russia
- **Bachelor of Theology** | Holy Trinity Seminary, Jordanville, NY

CERTIFICATIONS

- Angular 21 Complete Guide (2026 Edition) | Udemy
- React Complete Guide 2026 (incl. Next.js, Redux) | Udemy
- Complete React Developer (w/ Redux, Hooks, GraphQL) | Udemy
- Next JS: The Complete Developer's Guide | Udemy
- Microservices with Node JS and React | Udemy
- Microfrontends with React | Udemy
- Clean Code | Udemy
- Docker Mastery with Kubernetes and Swarm | Udemy

ADDITIONAL INFORMATION

- **US Citizen:** Yes

- **Location:** Richfield Springs, NY (Eastern Time). 11 years of 100% remote work, distributed team leadership, and async collaboration across time zones.
- **Availability:** Ready to contribute immediately.