

Amir Hossein Armaniya

PRODUCT PORTFOLIO — How I think, test, build, and learn

Open to relocation across Europe

✉ amirarmaniya@gmail.com in [linkedin.com/in/amir-hossein-armaniya](https://www.linkedin.com/in/amir-hossein-armaniya) ● github.com/amir-armaniya 🌐 amir-armaniya.pages.dev

AI Product Builder focused on turning ambiguous problems into working prototypes and testable MVPs through Product Discovery, experimentation, and rapid prototyping.

THE PRODUCT APPROACH

01 — Frame the problem. Understand the user, constraints, business context, and the decision the product must support.

03 — Prototype quickly. AI tools, APIs, lightweight builds — make an idea testable before full commitment.

02 — Define the smallest useful MVP. Separate the core hypothesis from what can wait; test the riskiest assumption.

04 — Learn and iterate. Experiments are decision tools; change direction when evidence contradicts the assumption.

WHAT I BRING TO A PRODUCT TEAM

Comfort with ambiguity: unclear idea → problem statement, target user, hypothesis, MVP.

Hands-on experimentation: building enough to test real behavior instead of debating features.

Shipped-product record: 3 live mobile games with 1.6M+ combined installs, plus published AI tools.

Technical fluency without making engineering the goal.

Creative range: games, AI tools, AdTech, retail analytics, resilience-oriented products.

CASE STUDY 01 — ADQUEST: VERIFIED ATTENTION / ADTECH

THE QUESTION

Can a rewarded-ad interaction create more valuable attention by making the user an active participant rather than forcing a timer?

THE MVP

Lightweight web prototype (HTML5/JS, Supabase, Cloudflare Pages): pre-ad primer, video flow, short attention challenge before the reward. Randomized UI buttons as anti-cheat.

THE EXPERIMENT

A/B test, forced-wait control vs AdQuest flow, 93 real users, developer traffic excluded.

THE RESULT

Average engagement 13.7s vs 6.9s — roughly 2× longer

18.4% of sessions completed the full video + challenge; every session that started the video finished it

THE PRODUCT LESSON

More engagement ≠ a better business. The next milestone became a three-sided incentive problem (users / developers / advertisers) — a business-model question, not a feature.

Live demo: adquest.tech

CASE STUDY 02 — PROMPT PACKER: AI DEVELOPER TOOL

PROBLEM

Prompt quality affects AI coding workflows, but optimization was treated as a one-off trick, not a reusable capability.

TARGET USER

Developers using Claude Code, Cursor, Cline and other MCP-compatible tools.

MVP & DISTRIBUTION

MCP server with 7 templates + cloud prompt library; published on npm with a free tier and Telegram-bot subscription flow; launched on Product Hunt.

LEARNING

A single optimization gave no reason to return — direction moved to a reusable library and batch workflow.

github.com/amir-armaniya/Prompt-Packer-mcp

CASE STUDY 03 — RETAIL ANALYTICS MVP: CPU-ONLY FOOTFALL COUNTING

PROBLEM

Small retailers can't justify recurring GPU/cloud costs for basic footfall measurement.

DISCOVERY → ITERATION

Real-video testing exposed double-counting when a person disappeared behind a shelf and returned. Added custom track-stitching logic that reconnects disappeared paths; exposed calibration controls instead of a black box.

PRODUCT DIRECTION

Local processing, privacy, installability, low-cost edge deployment.

CASE STUDY 04 — SMART DEFENSE RADAR: CRISIS MANAGEMENT

PROBLEM

Crisis dashboards fail exactly when needed most — when connectivity and centralized infrastructure go down.

DESIGN CHOICES

<500KB total · client-side rendering · Google Sheets as zero-cost real-time database · no-code automation (Activepieces) for ingestion · runs on 2G.

PRODUCT LESSON

Infrastructure constraints can be treated as product requirements rather than engineering problems.

smart-defense-radar.optemos67.workers.dev · github.com/amir-armaniya/Smart-Defense-Radar

MORE BUILDS

- Guardian** — risk-aware monitoring concept for AI-agent operations.
github.com/amir-armaniya/Guardian
- VisaFlow** — LLM document processing & multi-step procedure guidance.
github.com/amir-armaniya/visaflow-public

EARLIER EXPERIENCE — BUZZY BEE GAME STUDIO (2021–2024)

Started as a C# programming intern, moved into generalist production. Unity UI, gameplay support, particles, shaders, animation/bone systems; specialized in 2D Spine animation.

Game	Installs	Rating
Online Ludo (Manchbazi)	1.1M+	4.5★
Sara's House	450K+	4.7★
Traffic Puzzle — Amir's Story	66K+	4.6★

EDUCATION

B.A., Illustration— Pars University of Art and Architecture · 2023–2025 · Research focus: generative AI tools in creative workflows

Associate Degree, Computer Software Engineering— Shahid Chamran University of Ahvaz · 2020–2022 · GPA 17.95/20

LANGUAGES

English — Professional working proficiency · German — Beginner (actively studying) · Persian — Native

CONTACT

If you're building an AI or early-stage product and need someone who can turn an ambiguous idea into a working prototype, validate it with real users, and iterate toward an MVP: amirarmaniya@gmail.com