

Ahmet Kazankaya

AI-ASSISTED DEVELOPER / AUTOMATION AND ENABLEMENT

Istanbul, Turkey (remote)

One human, several agents. I decide, review and debug; they type.

ahmetkazankaya1@gmail.com · +90 536 214 8114 · [linkedin.com/in/ahmetkazankaya](https://www.linkedin.com/in/ahmetkazankaya) · github.com/kasparovabi

Summary

I came to software through production work rather than a computer science degree; my focus is systems and automation. Since early 2026, with Claude Code as the primary tool, I have shipped a task system in production since March for a foundation I consult for, an automation line that prices and lists real products on Etsy, an ESP32 robot with voice, and two open source skill sets other people install. A model writes the code and the failures land on me, so I know where these tools mislead. I also run the sessions that put these tools in other people's hands.

How I work with AI

Working model. Architecture and constraints go into per repository steering files; each task routes to the cheapest model that can do it; I read every diff before it lands. The same loop covers analysis, documentation and design, not only coding. Testing runs headless: Unity batchmode builds, xcodebuild with simulator screenshots, pytest and vitest suites the agent must pass. The automation carries the repetitive load, so I run several lines at once.

Risks and limits. A model is confident in exactly the places it is weakest, so my process assumes the output is wrong until something independent says otherwise. I refuse blacklist style fixes, which patch where the symptom shows rather than what broke. One research loop rejected 87 hypotheses over 170 iterations and returned the answer I did not want. I keep a searchable knowledge layer over what I read, with weekly lint runs for contradictions and stale claims, and verify current practice before choosing a stack, because a model's training data is older than the problem in front of me.

Handing it over. Building the automation is half the work; the other half is getting people to use it. At the foundation I ran the enablement for a team of fifteen: mostly one to one at their desks, sometimes a few people at once around a screen. That covered both the AI tools that fit each person's own job and the automations I built for them, and I am now preparing a seminar on AI use for the whole foundation. Most of what I teach is not which button to press but where the tool is confident and wrong, because that is what decides whether someone keeps using it after the first bad answer. The same instinct produced guard-20 and audit-20: my own working rules, written down clearly enough that other developers install and run them unchanged.

Selected projects

01 Maarif Task Management

TÜRKİYE MAARİF VAKFI · CONSULTING ENGAGEMENT · ISTANBUL · MAR 2026 TO PRESENT · IN PRODUCTION

The director used to read every incoming email, decide who should do each job, then track every delivery himself. The system reads the mailbox, turns each message into a task definition with an LLM, assigns it, and puts every job on an office TV board. I also trained the team that uses it, so the system runs without me in the loop.

Obstacle. generated documents drifted from the institution's templates and a text diff showed nothing, because the drift was visual; comparing delivered files against fixed templates visually made template stability measurable.

02 Rifki, an ESP32-S3 desktop robot

PERSONAL PROJECT · ISTANBUL · APR 2026 TO PRESENT · WORKING PROTOTYPE

Microphone to speech recognition to an LLM to speech synthesis to head and arm servos, with wake word detection and over the air updates.

Obstacle. on battery the board browned out and boot looped for a month. My first guess, that the servos stay passive during boot, was wrong: watching the hardware showed them twitching the instant the battery goes in. Floating pins at reset drew enough inrush to take the rail under its brownout threshold, and holding them low in the first line of the constructor fixed it without touching the hardware.

03 **Etsy Factory and Demand Engine**

OWN PRODUCTION BUSINESS · REMOTE · JUN 2026 TO PRESENT · RUNNING DAILY

Public domain museum works and locally generated art become Printify products and Etsy listings through a Telegram approval gate, on a schedule, driven from my phone. A pricing engine pulls Printify's shipping table per variant and country and disables combinations that cannot be profitable.

Obstacle. the agents kept producing near duplicate ideas. The model was the obvious suspect and changing it changed nothing; the cause was the shape of the data handed to them, which is the clearest case I have of an AI failure that is not an AI problem.

04 **Cebimde Claude**

PERSONAL PROJECT · REMOTE · JUN 2026 · BUILDING IOS APPS FROM THE PHONE, WITHOUT SITTING AT A MAC

The Mac becomes a build server you never look at: you describe the app from your phone, the agent scaffolds a SwiftUI project, compiles it, and a second command signs and installs it on the device in your hand. Failed builds go back to the agent, which fixes and rebuilds unprompted.

Obstacle. dropped connections left ghost agents writing to dead sockets; a connection epoch and a per project stop rule closed it.

05 **guard-20 and audit-20**

OPEN SOURCE · REMOTE · JUL 2026 TO PRESENT · GITHUB.COM/KASPAROVABI · INSTALLED BY OTHER DEVELOPERS

Twenty recurring failure classes in AI generated code, turned into process that runs by itself: rules that apply while the model writes, and a twenty pass verification gate whose last pass exists only to falsify the previous nineteen.

Obstacle. the gate made output worse on the newest model generation, which contradicted the whole premise. An evaluation harness running the same task with and without it showed the effect was real and model dependent, so the scaffold is now gated by model rather than always on.

Stack

AI TOOLING	Claude Code across a Windows and a macOS machine joined over Tailscale, Claude Agent SDK in my own applications, Codex CLI, Cursor, Antigravity, Gemini CLI, Ollama and OpenRouter for model routing, MCP servers for Blender and NotebookLM
CODE	Python, TypeScript, JavaScript, Swift, C++ (ESP32/Arduino), C#, SQL, Shell
BACKEND AND DATA	FastAPI, SQLAlchemy, SQLite, PostgreSQL, Prisma, Supabase
FRONTEND AND APPS	Next.js, React, Tailwind, SwiftUI, Capacitor, Tauri, PySide6
EMBEDDED AND INFRA	ESP-IDF, PlatformIO, Arduino, Git, Docker, Vercel, Netlify, Cloudflare Workers, Tailscale, launchd and cron
INTEGRATIONS	REST APIs, Microsoft Graph and Outlook 365, Etsy, Printify, Telegram, WhatsApp, Deepgram, ElevenLabs, Groq, KIE.AI
TESTING	pytest, vitest, Jest, XCTest, Playwright, headless batchmode CI

Experience and education

AI Solutions Designer	Türkiye Maarif Vakfı · Istanbul, consulting engagement	Feb 2026 to present
Generative AI Engineer	Pepper Root, Creative AI Studio · remote	Nov 2025 to Jan 2026
AI Artist	Joy Game · Istanbul	Aug 2025 to Oct 2025
Chief GenAI Art Director	GPT Journey · Istanbul	Mar 2024 to Aug 2025
Generative AI Creator	Fiverr · remote	Oct 2022 to Jun 2025

EDUCATION	Mechanical Engineering, Namık Kemal University from 2012, left before finishing. Ten years in visual production before software, running Stable Diffusion and ComfyUI on my own machines from 2023.
------------------	---

LANGUAGES	Turkish (native), English (professional)
------------------	--