

Alireza Ghasemi

AI Automation Engineer · LLM coding agents in production · workflow automation · API integrations · Python, PHP, SQL

Tabriz, Iran · me@alire7a.com · alire7a.com · +98 914 612 6970 · linkedin.com/in/alireza-ghasemi-712651320

Needs visa sponsorship (EU Blue Card or a national skilled-worker permit) · Open to relocation · Time zone UTC+3:30

SUMMARY

AI automation engineer with over two years of professional experience (since May 2024) at a web hosting and e-commerce company, where I own the technical side of its WordPress site-builder service and also work on a tech news site. I turn manual operations into automation that runs unattended: I write the specification, direct LLM coding agents to build it, review every change and ship it behind guardrails (self-tests, static-analysis deploy gates, conflict checks between parallel agent sessions, human approval for risky actions, git rollback). I have built:

- my own 4,500-product catalog: a third-party plugin imports the products, and my code classifies them (uncertain ones wait for human review), matches them against two other stores and records each store's price daily (in production);
- a one-way failover mirror that keeps a 40,000-post news archive in step on a second server (employer project, in production);
- a messenger bot for marketplace vendors where every write is previewed, confirmed, deduplicated and logged (my own project, in staging).

CORE STRENGTHS

- **Running AI coding agents safely in production:**
 - several agent sessions work on one live system in parallel, sharing one guide file of rules, decisions and lessons;
 - a deploy tool I specified (Python) refuses an upload if the live file's sha256 changed since the session read it, if the server-side lint fails, if the new file drops a function the live file defines or hooks a callback that does not exist, or if PHPStan reports a new finding;
 - every change is read back from the server and committed to git.
- **Tool access with human-in-the-loop control:** connected agents to the CMS and the CDN through existing MCP servers and CLIs, and wrote the rules they work under. Agents write, test and deploy each change and record its undo steps; settings, spending, deletions and anything visitors see wait for my explicit approval.
- **Automation that recovers on its own:** 20 scheduled jobs; the API-heavy ones throttle themselves, lock against double runs and stop on the first error. Dated measurement tasks re-check a change's effect later and report the numbers.
- **Evaluation and cost control:** dry runs over all live data before every rule change, a held-out test set for the classifier, and before-and-after metrics from logs and analytics. I measure agents' token cost per session and set context and model limits from those numbers.

EXPERIENCE

Owner and Developer, bartarpart.ir (car audio price-comparison catalog)

Jan 2024 – present · own project, in production, earns affiliate commission

- Built an automation pipeline of 14 modules (about 12,600 lines, written by AI coding agents to my specifications and reviewed by me), checked by 19 self-test functions on the live site (the changed file's own test runs after each deploy).
- Ran the build as parallel agent workstreams: one plan split into 11 streams (7 shipped in 5 days), a separate reviewer agent that audited the 10 busiest pages at phone size before a redesign, and scheduled agent tasks that re-measure each change on set dates and write the numbers back to the project guide.
- Human-in-the-loop product classifier for 6 taxonomies and 37 categories: multinomial Naive Bayes over title words and word pairs with rule overrides, trained only on human labels and retrained weekly. It holds a product for review when its top two categories are close: 99.3% correct category on 834 held-out products when built and 98.3% in 5-fold cross-validation on 4,191 labels; vehicle categories were the weak spot.
- Evaluated a local LLM second opinion (Qwen3 4B and 8B through Ollama, structured JSON output, retrieved few-shot examples, abstention) for the product classifier's uncertain band, with 5-fold cross-validation on 4,191 labels, a control set and an out-of-sample check. The 8B model cut wrong labels in the band from 34 to 25 at the same human review load. A spec-based rule that looked twice as good in-sample failed the out-of-sample check (13 to 22 errors), so it was rejected.
- Rule-based cross-store product matcher that prefers no link to a wrong one: every rule was added with the real pair that broke it (a 25-pair regression test) and dry-run over the live links before deploy; a full re-check of all 960 live links removed the 9 known wrong links and nothing else.
- Daily price-history table with a 30-day-low detector, an automatic daily price-drop post to Telegram and Bale channels (images drawn on the server), and back-in-stock alerts by bot.

- Google Search Console API integration without an SDK (service-account JWT): every sitemap URL inspected weekly within the quota, and a health dashboard that ranks pages for manual action.
- Turned a 3-minute outage, caused by a deploy that removed a function still in use, into a permanent guard in the deploy tool. Fixed a WP-Cron lost-update race between concurrent processes, with a self-test that simulates both.

Programmer (WordPress and WooCommerce), AzarOnline, Tabriz, Iran

May 2024 – present · web hosting and e-commerce company

- Own the technical side of the company's WordPress site-builder service and build every client site delivered through it.
- Build and customise WordPress and WooCommerce sites and stores: themes, plugins and custom features, payment-gateway setup and checkout improvements.
- Handle the company's expert-level technical support for customers' WordPress sites.
- **In parallel, Website Technical Specialist for Technoc (technoc.ir), a tech news site:**
 - Built, with AI coding agents to my specifications, a one-way failover mirror plugin (about 9,500 lines, in production) that keeps a 40,000-post archive, with its media, terms, authors and SEO data, in step on a second server in another region. Requests are HMAC-SHA256 signed with a nonce, and writes keep the original IDs. After a clock drift between the servers silently dropped posts, I added retry rules by error code, with backoff, and a test that replays the episode.
 - Built an at-most-once submitter for a paid per-URL API, so no URL is paid for twice: a UNIQUE key as the duplicate check, atomic row claims, no automatic retry after an ambiguous timeout, a cost preview before bulk runs, and a wait until each URL answers 200 on every host (before that check, 268 of 428 submissions went out too early).

Owner and Developer, TabrizAbzar (tabrizabzar.com), multi-vendor marketplace, Tabriz, Iran

Jul 2026 – present · own project

- Built, with AI coding agents to my specifications, a rule-based Bale messenger bot (now in staging) that lets marketplace vendors check and change stock and prices from chat. Incoming webhooks are deduplicated by update ID and handled by a worker that claims rows with leases. Every write is previewed, confirmed, guarded by a sha256 idempotency key, re-checked on confirm, verified after it runs and audit-logged. Tests prove that one vendor can never reach another vendor's data. It is the same safe-action pattern an AI agent needs before it can act on a live system.
- About 230 PHPUnit tests; a GitHub Actions workflow runs lint, PHPCS, PHPStan (level 5) and PHPUnit on PHP 7.4, 8.1 and 8.3.

SKILLS

- **AI and agents:** LLM coding agents (spec-driven development, parallel sessions, subagents), agentic workflows, MCP connectors and agent permissions, context engineering (instruction files, compaction), human-in-the-loop approvals, guardrails, evaluation (cross-validation, held-out and control sets, abstention, regression dry runs), local open-weight LLMs (Ollama, structured JSON output), token cost tracking and model routing, text classification (Naive Bayes written from scratch, no ML library)
- **Automation and integration:** REST APIs, webhooks and bots (Telegram Bot API), Google APIs with service accounts, scheduled jobs, queues with leases, idempotency and retries, web data extraction and product matching
- **Languages and data:** PHP 8 and Python (I write the specifications and review every diff; agents write most of the code), SQL (MySQL), basic HTML, CSS and JavaScript
- **Quality and delivery:** PHPUnit, PHPStan, PHPCS, GitHub Actions CI, Git, scripted deploys with read-back and rollback
- **Platforms and ops:** WordPress, WooCommerce, Linux basics, cPanel, DirectAdmin, LiteSpeed, Cloudflare, DNS, Google Search Console

EDUCATION

MSc Information Technology (E-commerce), Shiraz University, Iran (institution anabin H+) · 2024 – Mar 2027 (expected) · online programme

BSc Civil Engineering, University of Tabriz, Iran · 2018 – 2022

LANGUAGES

Persian (native) · Azerbaijani (native) · Turkish (intermediate) · English (intermediate, self-assessed; IELTS Academic planned for 2027) · Spanish (beginner)