

Maxim Chumakov

Backend / Full-Stack Developer (Python · Django · FastAPI)

Moscow, Russia (UTC+3) · Remote · Open to relocation · chumakov.mr@gmail.com · Telegram [@k1y0mi](https://t.me/k1y0mi)
GitHub: github.com/k1y0miiii · Portfolio: k1y0miiii.github.io · Hugging Face: huggingface.co/k1y0mi

Summary

Backend / full-stack developer (Python / Django / FastAPI) who has designed and shipped four production multi-tenant platforms largely solo — an admission-management SaaS, an online examination platform, a messaging-integration gateway, and a distributed DNS-propagation service — with self-hosted LLM integration and sandboxed code execution. Comfortable across the whole stack and into infrastructure (PostgreSQL, Redis, Docker, nginx, a ~30-server Linux fleet). Track record of shipping fast under real constraints, owning products end to end, and introducing engineering discipline (version control, load testing) where none existed. Flexible overlap with EU/US business hours.

Technical Skills

Languages: Python, JavaScript, SQL, Bash; Rust & C++ (side projects / in progress)

Backend: Django 5.x (+ DRF, Unfold admin), FastAPI, aiogram 3, SQLAlchemy (async), REST APIs, webhooks, custom task/retry queues on Redis

Databases & caching: PostgreSQL, Redis (caching, queues, shared state)

Infra / DevOps: Docker & docker-compose, Linux (Ubuntu/Debian) administration, nginx, gunicorn, systemd/supervisor, Git, CI basics, VPS provisioning, DNS / SMTP

Performance: k6 load testing (smoke / ramp / stress / soak / spike, xk6-sql), query profiling, custom Django performance tooling

AI / LLM systems: Anthropic Claude API, self-hosted LLMs (OpenWebUI / Ollama), transformer fine-tuning (ruBERT), LLM-as-judge pipelines, sandboxed code execution (Judge0), MCP tooling

Frontend: Tailwind CSS, Bootstrap, Alpine.js, vanilla JS/AJAX, Chart.js, d3-geo

Security (learning track): OWASP Top 10, nmap, Burp Suite, Metasploit, Kali Linux (TryHackMe)

Experience

Full-Stack Developer (Contractor) — Nik-A Tech (ex-iQnix Tech), Remote May 2025 – Present

- Sole or lead developer on four production platforms, plus the company CMS and official website; took over the entire workload of a departing developer.
- Introduced Git version control to a codebase previously edited directly on production servers; established load testing (k6) as a release practice.
- Administer a fleet of ~30 Linux servers across client deployments — provisioning, deploys, DNS/SMTP, ops.

COP-UPK — *multi-tenant admission-management SaaS · Django, PostgreSQL, Redis*

- Designed and built solo in ~2.5 months ahead of the July admission campaign; 200+ routes, now used by 4 colleges/universities and 20–30 staff, 500+ applications per campaign, plus public applicant users. Still solely maintained by me.
- Full admission cycle automation with 5-role RBAC, conditional multi-step application forms, per-institution isolated deployments.
- Configurable DOCX document-generation engine: institutions upload their own templates and map placeholders through a settings UI — zero developer involvement per new template.
- Online exam module with Judge0 sandboxed code execution and LLM-assisted grading via self-hosted OpenWebUI (model selection, evaluation, LLM-as-judge pipeline).

MyDNS — *distributed DNS-propagation service · Django, PostgreSQL*

- Master-master replication across 4 geo-distributed nodes without a message broker: signal-based event journal, UUID idempotency, last-write-wins conflict resolution, cursor-based sync, HMAC peer authentication.
- Parallel resolution across 25 global resolvers with a d3-geo world-map visualization.

MAX — *integration gateway for the MAX messenger*

- Single integration point with the MAX messenger for all company projects. Self-service onboarding: create a bot, receive an API endpoint with instructions; message routing, delivery, and centralized tracking.
- Multi-tenant: SHA-256-hashed client credentials, HMAC-signed callbacks, retry queue for delivery guarantees.

COP-SAO — *exam & test platform (spin-off of the COP-UPK online-exam module)*

- Standalone system, optimized purely for running exams and tests. Built custom `django_perf` profiling package; solved data-migration edge cases (NULL-distinct constraints in PostgreSQL) and multi-worker gunicorn state-sync issues.

Open-Source Projects

- [ru-ai-text-detector](#) — detects human vs AI-written Russian text: fine-tuned ruBERT, calibrated to FPR $\leq 3\%$; [live demo on Hugging Face](#). PyTorch, transformers.
- [token-diet](#) — live playground for what your JSON costs in tokens and dollars across models; backed by a reproducible JTF benchmark (~40% lossless). JavaScript, Python.
- [json-token-format \(JTF\)](#) — lossless compact JSON encoding for LLMs: ~33% fewer tokens than raw JSON, fully round-trippable. Python.
- [glyph-mcp](#) — MCP server giving LLM agents web search & fetch as clean text, ~8× fewer tokens per lookup. JavaScript.
- [glyph](#) — a web browser that runs in the terminal: HTTP(S), html5ever HTML parsing, CSS subset, tabs, forms. Rust.
- [ids-nsl-kdd](#) — network intrusion detection on the NSL-KDD benchmark: RandomForest / GradientBoosting / LogReg, binary & multiclass classification. Python, scikit-learn.

Education

BSc, Applied Informatics (in Economics) — Institute of Business Careers, Moscow — graduated June 2026.

Thesis project: **SCBP** — a web application for software project cost estimation (Django): parametric estimation models, formula-based cost calculation, automated quote generation.

Languages & Logistics

Russian — native · **English** — B2: comfortable in technical discussions; fluent technical reading/writing.

Remote from Moscow (UTC+3) with flexible overlap with EU/US hours · Open to relocation · Independent-contractor setup; flexible on international payment arrangements.