Egor Trefilov

Python Backend & AI Automation Developer

Nizhny Novgorod (UTC+3) / trefilovegor84@gmail.com / Telegram @cooffeek1337
github.com/coffee1337 / linkedin.com/in/coffee1337 / coffee1337.github.io

PROFILE

Backend developer building Python services that hold up outside a demo: FastAPI APIs, PostgreSQL schemas with versioned migrations, cookie-auth sessions, background workers and Docker deployment. I integrate LLM/RAG into products properly — retrieval over the platform's own content, per-user usage accounting, streamed answers — rather than wrapping a prompt. In an ML / Computer Vision R&D internship I built a physics-based synthetic radar PPI generator and trained a YOLO11s detector on the resulting registered dataset (Rospatent #2026624059). Comfortable beyond the backend too: React/Next.js, Flutter and .NET.

EXPERIENCE

Python Backend & AI Automation Developer — Independent / Freelance — Aug 2025 — Present

- Ship FastAPI services on PostgreSQL with SQLAlchemy and Alembic migrations, cookie sessions, role checks, background workers and Docker-based delivery.
- Built RAG pipelines (pgvector retrieval, streaming) and usage-aware AI features with per-user quotas and token/cost accounting; wired Telegram and email providers behind env-driven adapters.
- Automate repetitive work: Telegram bots, parsers, external API integrations and data-processing scripts with retries and idempotent writes.

ML / Computer Vision R&D Intern — Radio-Engineering Research Institute — 2026

RADARPPI-MTD · DATASET REGISTERED WITH ROSPATENT #2026624059 AT NGIEU

- Built **RadarPPI-MTD**, a physics-based synthetic radar PPI generator with zero-error auto-labelling: 10,000 scenes, 619,153 frames (1120×855), ~11.0M boxes, 11.8 GB.
- Trained YOLO11s on PyTorch: mAP@0.5 = 0.924, precision 0.988, recall 0.908; custom radar metrics 2,000 scenes — Pd = 0.916, F1 = 0.955, FPPI = 0.05, localisation error 0.56 px (≈27 m).
- Validated sim-to-real transfer on live radar frames and added ByteTrack multi-frame association for persistent tracks; ran the whole pipeline offline in an air-gapped environment.

SELECTED PROJECTS

AI Python Mentor

case study · source

Python · FastAPI · PostgreSQL · SQLAlchemy · Alembic · Next.js · Docker · RAG

350 API tests · 10 skipped · 15 sandbox tests · 90 exercises / 288 cases

Full-stack learning platform: 90 skills and exercises, an AI tutor with pgvector retrieval over immutable lesson snapshots, spaced repetition, career-path analysis, and a separate hardened runner that executes user Python outside the API (350 API tests, 15 sandbox tests).

RadarPPI-MTD

case study · Rospatent

Python · PyTorch · YOLO11s · OpenCV · NumPy · ByteTrack

Dataset registered in Rospatent registry · mAP@0.5 0.924 · Pd 0.916 · FPPI 0.05

Synthetic radar PPI dataset generator registered as a database (No. 2026624059) plus a moving-target detector: physical polar-to-pixel geometry, Poisson clutter and missed-scan simulation, balanced easy/medium/hard scenes, radar-native Pd/FPPI evaluation.

portfolio-analytics

case study · source

Python · FastAPI · SQLAlchemy 2.0 · PostgreSQL · React 19 · TypeScript · Tailwind · Docker

Running live on coffee1337.github.io · no cookies, no raw IPs

First-party analytics running live on this site: beacon collector with batch intake and rate limits, salted HMAC instead of raw IPs, async geo/ASN enrichment in background tasks, Core Web Vitals monitoring and a SPA dashboard with a 24×7 activity matrix.

NGIE University App

case study · source

Flutter · Dart · Riverpod · Drift · SQLite · Kotlin · Swift · WidgetKit

Offline-first: schedule opens from cache · widgets rendered natively

Cross-platform university app with an offline-first Drift/SQLite cache, network and cache layers, schedule notifications and native home-screen and lock-screen widgets for Android and iOS.

TransactionMonitor

case study · source

C# · .NET 8 · WinUI 3 · SQL Server

Rule-based scoring stored per transaction · CSV export for analysts

Desktop transaction monitoring and risk analysis: explainable rule-based scoring stored with each transaction, clients/accounts/counterparties, analyst review for high-risk operations and CSV export.

Neural A* Heuristic

case study · source

Python · PyTorch · NumPy · A* · Docker

One architecture, three state spaces · trained from solved states

A value network V(s) trained from backward walks from solved states becomes the A* heuristic (f = g + V); a single architecture covers three state spaces — 15 Puzzle, Toggle Lights and Cylinder — behind one shared environment API that every puzzle implements.

EDUCATION

B.Sc. in Informatics and Computer Engineering (09.03.01) — NGIEU, Nizhny Novgorod State Engineering and Economic University

2024 — 2028

ARTIFICIAL INTELLIGENCE SYSTEMS · 3RD YEAR · CLASS OF 2028

CERTIFICATIONS & REGISTERED IP

> **State Registration Certificate of Database No. 2026624059 — RadarPPI-MTD**
>
> Rospatent (FIPS) · Aug 2026 · Rightholder: NGIEU · 11.8 GB · Radar PPI Generator v1.0.0
>
> FIPS registry · Certificate PDF · Abstract PDF

Generative AI Professional Certificate by Snowflake	Snowflake · Nov 2026	verify
Docker Foundations Professional Certificate	Docker, Inc. · Oct 2026	verify
Microsoft Azure Essentials Professional Certificate	Microsoft · Oct 2026	verify
Career Essentials in GitHub Professional Certificate	GitHub · Oct 2026	verify
Building AI Agents with Snowflake	LinkedIn Learning · Snowflake · Nov 2026	verify
AWS Certified AI Practitioner (AIF-C01) Cert Prep	LinkedIn Learning · Oct 2026	verify

TECHNICAL SKILLS

Backend — Python 3.12, FastAPI, PostgreSQL, SQLAlchemy 2.0, Alembic, Redis, AsyncIO, REST APIs, cookie sessions, background jobs, Argon2

AI, ML & Data — LLM APIs, RAG / pgvector, PyTorch, YOLO11s, OpenCV, NumPy, Pandas, ByteTrack, Snowflake, prompt and token-cost accounting

Cloud & DevOps — Docker, Docker Compose, GitHub Actions, Microsoft Azure, AWS, Linux, Git, Caddy, Prometheus/Grafana, k6 load testing

Frontend, Mobile & Desktop — React 19, Next.js 15, TypeScript, Tailwind CSS, Vite, Flutter/Dart, C#/.NET 8 + WinUI 3, C++17, Swift/WidgetKit, Kotlin

Engineering practice — pytest, Playwright, Playwright visual audits, CI pipelines, Alembic migrations, backup/restore drills, runbooks and threat modelling