

Vladyslav Dykyi

AI Engineer | LLMs · AI Agents · LangChain · Backend Systems

Linz, Austria · +43 688 64059508 · vladdikiy17@gmail.com · linkedin.com/in/vladyslav-dykyi-4321a9283 · github.com/dykyivladk1

Profile

AI Engineer working on LLM products: AI agents with LangChain and LangGraph, tool calling, multi-agent setups and the backend APIs around them, in Python and TypeScript (NestJS). I have built this kind of system for a real product, a voice assistant for hotels, and I keep building my own agent and search projects on the side. I like taking an AI feature from a first prototype to something that is deployed with Docker and CI/CD and used every day.

Skills

AI & LLMs	LangChain, LangGraph, AI agents, multi-agent systems, tool calling, structured outputs, OpenAI and Anthropic APIs, multimodal AI, speech recognition
Search & Embeddings	Sentence Transformers, Qdrant, FAISS, semantic search, MMR re-ranking
Backend	Python, TypeScript, NestJS, FastAPI, Flask, REST APIs, SSE streaming, async programming, JWT auth
Databases	PostgreSQL, MongoDB, Elasticsearch, SQL
Cloud & DevOps	Docker, Docker Compose, Kubernetes, GitHub Actions, Azure, Terraform, AWS, pytest
ML & Edge	PyTorch, ONNX Runtime, INT8/INT4 quantization, computer vision, Raspberry Pi

Experience

Feb 2026 – present	Production Engineer Tractive Linz, Austria · Full-time - Write Python automation, internal Flask APIs and diagnostic tools used across our factory sites. - Dig into production incidents and hardware failures, find the root cause and ship the fix. - Query production data in MongoDB and Elasticsearch (Kibana) to spot problems before they grow. - Work with engineering and production teams to improve automated testing and diagnostics.
Sep 2024 – Sep 2025	AI Engineer LORAtch e.U. Linz, Austria · Full-time - Only AI engineer in a small team building LORA Cube, a voice assistant for hotels. Designed the AI architecture, starting from the first MVP and growing it with the product. - Built the agent layer with LangChain and LangGraph: several LLM providers, tool calling and multimodal input. - Wrote the backend REST APIs in TypeScript with NestJS, linking devices, AI services and agents, with data in PostgreSQL. - Built async pipelines that coordinate local inference, cloud AI services and device actions, so answers come back fast. - Built the on-device speech pipeline: wake word, voice activity detection and speech recognition on a Raspberry Pi. - Made the on-device models faster with ONNX Runtime and INT8/INT4 quantization. - Ran the backend and agents on Azure with Terraform and Kubernetes, with Docker images and CI/CD in GitHub Actions.
Mar 2024 – Dec 2025	AI Engineer Johannes Kepler University Linz Linz, Austria · Part-time - Built an AI platform where students chat with assistants made for specific subjects, like biology. - Added LLM features in Python and connected the assistants to existing university services. - Helped design and build other AI applications for the institute using machine learning and LLMs.
2022 – 2024	Freelance Backend Developer Upwork Remote - Built Python backends for clients abroad: REST APIs, data pipelines and third-party integrations. - Worked mostly with FastAPI, Flask, MongoDB and SQL databases. - Took projects from first requirements through architecture, testing and deployment to handover.
2021 – 2022	Python Developer National University “Odesa Law Academy” Odesa, Ukraine · Part-time - Built Python tools and automated data-processing workflows for university projects. - Made a Telegram bot with an AI travel assistant that suggested restaurants and hotels based on what the user liked.

Projects

Universe Python · Flask · LangChain · LangGraph · OpenAI · Anthropic - A small platform for building and running custom AI agents with OpenAI and Anthropic models. - Agent builder with templates, model and tool settings, and a playground chat to test each change right away. - Agent teams run by a supervisor (LangGraph StateGraph) that uses structured output to pick the next agent. - Tokens, tool calls and agent hand-offs stream to the browser over SSE.
Adify AI Python · Flask · Sentence Transformers · Qdrant · Spotify API · Docker - Type a mood or activity (“rainy sunday jazz”) and get two Spotify playlists back: a closest match and a wider mix. - The prompt is parsed into genres, moods and decade, embedded with Sentence Transformers and searched in Qdrant. If there are not enough good hits, the app pulls new tracks from Spotify, embeds them and adds them to the index. - No prebuilt dataset: the catalog grows as people use it. The first version used FAISS and a fixed dataset. - MMR re-ranking keeps the playlists varied, with a limit per artist. Runs with Docker Compose and is tested with pytest.

Education

BSc, Artificial Intelligence Johannes Kepler University Linz
BSc, Economics Odesa National Economic University

Languages

Ukrainian	Native
Russian	Fluent
English	C1
German	A2