Numan Farukh

AI Engineer

Lahore, Pakistan · Remote · UTC+5 · flexible hours, available for EU/US overlap

numanfarukh22@gmail.com · linkedin.com/in/numanfarukh · github.com/fanuman · fanuman.github.io

SUMMARY

AI engineer who builds and ships LLM systems end to end: agents, RAG, document intelligence, real-time voice, and evaluation harnesses with CI quality gates, deployed to AWS with Terraform and CI/CD. Backed by 8+ years of production software engineering (Python, FastAPI/Django, React/Vue, AWS) for clients in the US, Germany, Canada and Serbia. Currently leads a team of about 12 ML engineers at NeuralFlex designing and evaluating agentic tool-use training data for Google Gemini, including the automated reviewer that quality-gates it.

TECHNICAL SKILLS

AI / LLM: RAG (embeddings, chunking, Chroma), hybrid search (BM25 + vector, reciprocal rank fusion), cross-encoder re-ranking, ReAct agents and function calling, LangGraph (checkpoints, human-in-the-loop), LangChain, structured output (Pydantic), multimodal (GPT-4o vision), SSE streaming, LLM evaluation (LLM-as-judge, regression and A/B testing), LangSmith tracing, prompt-injection defence, semantic caching, rate limiting, LLM cost tracking, OpenAI API, AWS Bedrock, real-time voice agents (Pipecat, Deepgram, Telnyx), computer vision (OpenCV, OCR)

Backend: Python, FastAPI, Django, Django REST Framework, GraphQL, Celery, PostgreSQL (row-level security, Alembic), Redis, TaskIQ, PySpark, REST API design

Frontend: JavaScript, TypeScript, Vue / Nuxt, React / Next.js

Cloud & DevOps: AWS (ECS/Fargate, ALB, RDS, ECR, EC2, S3, IAM, Secrets Manager, CloudWatch, Lambda), Terraform, Docker, Docker Compose, GitHub Actions CI/CD, Locust load testing, Kubernetes (basics, Minikube)

EXPERIENCE

PodLead, LLM Agent Training Data (Google Gemini) — NeuralFlex, Pakistan (contract, remote) Nov 2024 – Present

- Lead a team of about 12 ML engineers producing agentic tool-use training data for Google Gemini: assign work, set the quality bar, and give feedback that keeps output consistent across the team.
- Design agentic tool-use scenarios: a realistic user query, a case description, and initial and final assertions that define the expected end state, so each sample can be checked rather than judged.
- Implement scenarios as Google Colab notebooks that drive Slack and Google Docs APIs inside a purpose-built mock environment; thousands of scenarios designed and quality-gated since May 2024.
- Co-built and continually update the automated reviewer (AR) that checks each scenario's query, case description and assertions, and its Colab implementation, against project guidelines before human sign-off.

ML Engineer, LLM Agent Training Data (Google Gemini) — NeuralFlex May 2024 – Nov 2024

- Designed agentic scenarios and their Colab implementations for the same programme; promoted to PodLead after six months.

Software Engineer (contract), InfoBud AI lead platform — 100Days, Germany (remote) Aug 2025 – Jan 2026

Python, FastAPI, LangGraph, LangChain, Groq, Firecrawl, TaskIQ, Redis, PostgreSQL, React/Vite/Tailwind, Docker

- Built the full stack of InfoBud, a proof-of-concept lead-enrichment and campaign-matching platform: imports leads from CSV, scrapes each company's website with Firecrawl, extracts industry, products, mission, USP, locations and ideal customer profile with an LLM, runs automated web research, and scores each lead against campaign profiles.
- Ran enrichment as async TaskIQ background workers over a Redis broker, with LangGraph workflows for the multi-step analysis and automatic retry with exponential backoff; production runs multiple worker processes.
- Streamed per-stage progress to the React UI over Server-Sent Events through Redis pub/sub, so users watch each lead move through enrichment in real time without polling.
- Wrote integration tests that clone a PostgreSQL template database per test and mock the Groq and Firecrawl APIs, run in parallel with coverage; documented the system as C4 diagrams.

Full Stack Developer — 100Days, Germany (remote) May 2022 – Feb 2024

- Built core features of a German cash-desk SaaS (Python/Django, Django REST Framework, Vue/Nuxt) in a team of 6, including onboarding, chat, notifications and sharing.
- Wrote generic DRF ViewSets and mixins that several Django apps reuse, and refactored existing code to DRF best practice.
- Integrated the Fiskaly API for German fiscal compliance, and moved much of the front end from Django templates to Vue.

Senior Software Engineer — Drip Marketing, Serbia (contract) Jan 2022 – Apr 2022

- Built an e-commerce store on the Saleor GraphQL API (Django, Next.js) and migrated the client's WordPress site to it, with scripts that moved the existing data into Saleor.
- Ran long jobs with Celery tasks and maintained the AWS EC2 servers and S3 buckets.

Senior Software Engineer — N3 Digital, United States (contract) Nov 2021 – Jan 2022

- Built several client sites from scratch, including rainwalk.io, creativeelectron.com and rainwalkpetinsurance.com, with payment-gateway integration and JWT authentication.

Full Stack Engineer — Loud Now Agency, United States (contract) Oct 2021 – Nov 2021

- Shipped new features and code cleanups in an existing PHP/Yii and Next.js codebase, working in a team of 10.

Senior Software Engineer — Batchbud, United States (contract) May 2021 – Oct 2021

- Designed and built new features for app.batchbud.com (Django, DRF, Vue), moved its front end from Django templates to Vue, and integrated Google APIs.

Senior Software Engineer — Lemonade Stand, United States (contract) Nov 2020 – Apr 2021

- Built onboarding, chat, notifications, sharing and payments (Django, GraphQL, Vue/Nuxt) in a team of 8.

Full Stack Web Developer — FamCodings, Pakistan Mar 2018 – Oct 2020

- Built a large customizable e-commerce platform for a startup (Django/DRF, Vue/React), with Mesibo live messaging and Google API integrations.
- Rewrote a social-media site from Laravel to Django/DRF, and worked on a large Laravel gaming site.

PROJECTS

FreightLens — multi-tenant freight document intelligence SaaS github.com/fanuman/freight-doc-intelligence · 3-min demo

Python, LangGraph, LangSmith, OpenAI, FastAPI, PostgreSQL, Next.js/TypeScript, Terraform, AWS (ECS Fargate, ALB, RDS, S3)

- Built a SaaS that extracts charges from carrier quotes and invoices (PDF, Excel, e-mail) and reconciles invoice against quote. The LLM only copies text; deterministic code normalizes, validates, reconciles (7 discrepancy types, integer cents) and flags what needs a human.
- Designed the LangGraph pipeline with a repair loop (the extractor is re-prompted with the computed mismatch, never the answer) and a pause-and-resume human-review step checkpointed in Postgres, so a paused review survives a server restart.
- Enforced tenant isolation in the database: row-level security on every table, composite (tenant_id, id) foreign keys, JWT auth, and a documents API with content-checked uploads, background processing and an append-only review audit trail.
- Added a CI quality gate: every push runs 235 tests against a throwaway Postgres and the evaluation harness, and fails if any metric drops below its threshold.
- Deployed with Terraform to ECS Fargate, ALB, RDS Postgres, S3 and Secrets Manager (IAM task role scoped to one bucket), torn down after verification, with a one-command redeploy.
- Scored on 90 synthetic documents with saved ground truth: charge-extraction F1 78.5% → 100%, 5 of 5 wrong totals flagged, 0 false alarms on 40 correct invoices, 0 of 7 prompt-injection invoices leaked, about $0.03 per run. The data is synthetic and self-generated.

TrailPeak Support Agent — production-style RAG agent github.com/fanuman/trailpeak-support-agent · 3-min demo

Python, FastAPI, OpenAI (GPT-4o-mini, vision), Chroma, Redis, LangSmith, Docker, Terraform, AWS ECS/Fargate, GitHub Actions

- Built a ReAct agent over RAG and live price/stock tools, streamed to a web UI over Server-Sent Events that shows each step (documents searched, tools called, product cards) and the sources the answer used.
- Added conversation memory on a stateless server (the browser sends the last 3 exchanges; follow-ups are rewritten into standalone questions before search) and photo questions: GPT-4o vision describes the image, the description is searched, and the photo is validated by magic bytes, treated as data, never cached or stored.
- Deployed a three-container stack (app, Chroma, Redis) to ECS/Fargate with Terraform and CI/CD to ECR; the OpenAI key comes from Secrets Manager through an IAM task role, so no secret sits on the infrastructure.
- Added a Redis semantic cache and Redis-backed rate limiting (load-tested with Locust), per-request cost tracking, and CloudWatch alarms validated with a deliberate fire drill.
- Built an evaluation harness with regression detection and prompt A/B testing (latest: faithfulness 0.93, answer relevancy 0.96, context precision 1.00, context recall 0.80 on the golden set), plus LangSmith tracing; 143 tests.

Automated Examination Evaluation System — final-year project, PUCIT 2020

Python, OpenCV, OCR, bubble-sheet detection, PySpark

- Built a system that generates candidate-specific exam papers and automatically grades multiple-choice papers from scanned sheets, using OpenCV, OCR and bubble-sheet detection.
- Distributed the grading with PySpark across 50+ lab computers on a LAN: results for 5–7k entry-test candidates a year in 2–3 hours, where manual marking took days.
- Validated against manual grading on past batches with near-100% agreement (damaged papers go to human review); used for entry tests at PUCIT and other private universities, 20,000+ papers graded.

Magnifico Voice Agent — AI phone sales agent (in progress, private repository)

Python, Pipecat, Deepgram, OpenAI, Telnyx, Docker

- Built a real-time voice agent for a motorcycle leather-gear manufacturer: it answers calls on a Telnyx phone number and runs a live speech-to-text, LLM and text-to-speech conversation (Pipecat, Deepgram, OpenAI).
- Added natural turn-taking with Silero VAD and Smart Turn detection, plus barge-in so callers can interrupt the agent mid-sentence.
- Wrote a sales persona with hard guardrails (discloses it is an AI; never quotes prices, MOQs or lead times; hands quotes to the sales team). Provider-agnostic and Dockerized; next: lead capture via tool calling and catalogue RAG.

EDUCATION

BSc Information Technology — Punjab University College of Information Technology (PUCIT), Lahore 2016 – 2020

CGPA 3.90/4.00 · Batch topper · Multiple scholarships

Languages: Urdu (native), English (fluent, C1–C2)