

Saim Malik

Senior Software Engineer & AI Engineer | Python • Django/FastAPI • LLMs • NextJS

+923444411987 saimmalik8397@gmail.com [LinkedIn](#) [Github](#)

Lahore, Pakistan (GMT+5) • Available full-time as a remote contractor • Overlaps US/EU hours

SUMMARY

Senior Software Engineer with 6+ years of experience building scalable SaaS platforms, AI-powered applications, and multi-tenant backend systems using Python, Django/FastAPI, React, Next.js, PostgreSQL, AWS, and GCP. Experienced in LLM-powered features, RAG assistants, AI agents, LangChain/LangGraph workflows, semantic search, and production-grade API systems. Built an AI-driven auditing platform that uncovered \$40M+ in missed commissions and developed venture products from zero to launch for a European venture builder, including Kilow, a health app that raised \$2.5M+. Led 10+ engineers across architecture, delivery, code reviews, CI/CD, and production reliability.

EXPERIENCE

Senior Software Engineer → Associate Team Lead

Lahore, Pakistan

[Devsinc](#)

2021 - Present

Software and product engineering company delivering enterprise web, SaaS, and AI-enabled solutions for global clients.

- Built and scaled multi-tenant SaaS platforms in Django + PostgreSQL on AWS, designing the API layer and async processing backbone.
- Shipped LLM-powered features RAG assistants, intelligent search, and workflow automation from prototype to production.
- Built AI agent workflows and LLM-driven pipelines using LangChain/LangGraph, powering an AI career platform for job discovery, scoring, resume optimization, application tailoring, and interview-preparation generation across multiple LLM providers with schema-constrained outputs.
- Cut production defects 75% by tightening CI/CD, review standards, and test discipline.
- Led a team of 10+ engineers across architecture, delivery, code review, and production support.
- **Client engagement via Stryber (European venture builder) — Remote • (2023-2025)**
- Built two venture products from zero to launch as senior engineer: Kilow (health app) and Miza (fintech app). Kilow went on to raise over \$2.5M in investment for the product.
- Developed cross-platform mobile apps in React Native backed by Python APIs (FastAPI, Flask), with CI/CD and deployments on Google Cloud Platform.
- Shipped a companion web application end-to-end, owning architecture and delivery from first commit through production.
- **Client engagement via letsremotify (a Devsinc company) — Remote • (2022-2023)**
- Delivered 25+ production features across multiple SaaS applications using Django, React, PostgreSQL, and cloud services.
- Built full-stack workflows spanning REST APIs, frontend interfaces, background jobs, and integrations.
- Improved code maintainability by 50% through structured reviews and modular, reusable architecture.
- Automated CI/CD pipelines, cutting deployment cycles by 30%.

Software Engineer

Lahore, Pakistan

[CCJK Technologies](#)

2020 - 2021

Technology company providing software development and localization/translation technology services.

- Developed backend APIs and services using Python and Ruby on Rails for enterprise and translation platforms.
- Designed and implemented RESTful APIs supporting scalable business workflows.
- Built scalable systems for enterprise and translation platforms.
- Contributed to full-stack development including API design and database architecture.
- Participated in Agile development and code review processes.

EDUCATION

Bachelor of Computer Sciences

Lahore

University of Central Punjab

2016 - 2020

SKILLS

Programming Languages: Python, JavaScript, TypeScript, Ruby

Backend Frameworks: Django, FastAPI, Flask, Ruby on Rails, Django REST Framework, GraphQL API

Frontend Frameworks: ReactJS, NextJS, React Native, Redux, VueJS, HTML5, Tailwind CSS

AI & Machine Learning:

Large Language Models (LLMs), OpenAI, Recommendation System, LLM Integration, Retrieval-Augmented Generation (RAG), LangChain, AI Agents, Prompt Engineering, Token/Context Optimization, Vector Search, Embeddings, Semantic Search

Cloud & DevOps:

AWS, S3, GCP, EC2, RDS, SQS, IAM, Docker, Kubernetes, CI/CD Pipelines, Jenkins, CircleCI, GitHub Actions, Cloud Deployment

SKILLS

Architecture:

Microservices Architecture, Event-Driven Systems, Distributed Systems, Asynchronous Processing, Message Queues, High-Concurrency Systems, Multi-Tenant Architecture, Scalable Backend Systems

Tools & Collaboration: Git, Bitbucket, JIRA, Agile/Scrum, Code Reviews, Technical Documentation, SDLC

PROJECTS

Neighborhoods.com – Real-Estate Data Platform

Technology Stack: Python, Django, FastAPI, Snowflake, PostgreSQL, ReactJS, AWS (S3, EC2, RDS), Redis, Celery, Docker, Microservices Architecture

- Architected distributed microservices-based data platform ingesting 100+ MLS feeds and public datasets into Snowflake data warehouse.
- Designed large-scale ETL pipelines in Python processing millions of structured and semi-structured records using Snowflake's native JSON capabilities.
- Built a property recommendation engine using collaborative filtering (Neural Collaborative Filtering and item/user-based k-NN) over implicit user signals, views, saves, clicks, and inquiries to surface listings matching each user's interests.
- Handled cold-start via onboarding interest profiles and content-based fallback, transitioning to collaborative filtering as interaction history accumulated.
- Built an agent auditing pipeline analyzing a decade of transaction data, using record-matching logic (MLS-to-parcel linkage via ID, date-proximity, and price-similarity matching) to detect unreported deal closures and recover \$40M+ in unpaid commissions.
- Developed Django & FastAPI-based REST APIs exposing high-volume listing and transaction datasets.
- Implemented asynchronous processing with Celery + Redis for background ingestion, enrichment, and reconciliation tasks.
- Designed multi-tenant RBAC-secured API layer for enterprise analysts and legal teams.
- Built ReactJS dashboards enabling real-time data exploration and analytics.
- Deployed services on AWS with Dockerized containers and scalable infrastructure.
- Implemented monitoring and alerting to ensure high reliability of ETL jobs.

AI Career Ops – LLM-Powered Job Discovery & Application System

Technology Stack: TypeScript, Node/Express, React, LLM integration (multi-provider), Docker; Python/LangChain components

- Building a multi-tenant platform that discovers, scores, and tailors job applications through an LLM-driven pipeline (discover → score → select → tailor → PDF), with per-tenant data isolation.
- Designed an LLM abstraction layer across multiple providers with retry/fallback, credential validation, and config-based model swapping; constrained outputs with JSON schemas for deterministic parsing.
- Built a pluggable, auto-discovered job-extractor system and an AI CV-tailoring/optimization service with streaming (SSE) output.
- Implemented deep role evaluation (CV-to-requirement gap analysis, level strategy, interview-prep story generation) via structured LLM calls, with token/context optimization to control cost.
- Backed by an extensive automated test suite (API contracts, tenant isolation, scoring, tailoring); deployed via Docker Compose with health checks.
- Evolving orchestration to LangGraph (checkpointed, human-in-the-loop) with Postgres + pgvector, powering an embedding-based recommendation engine that matches candidates to jobs by semantic similarity between profile/CV and job descriptions.
- Engineered and evaluated LLM prompting for CV-optimization, tuning outputs for accuracy and reducing hallucination through structured prompts and validation.

ParkGuard – Enterprise Parking Protection & Claims Platform

Technology Stack: Django, PostgreSQL, Celery, Redis, Elasticsearch, React, AWS (ECS/EKS, S3), Docker, GitHub Actions

- Architected an enterprise multi-tenant claims platform serving parking partners, subsidiaries, and admins with hierarchical partner relationships and strict RBAC data isolation (MFA, API-key auth, IP whitelisting, audit logging).
- Built the full claim lifecycle engine (Created → Review → Approved/Rejected → Closed) with evidence uploads to S3, controlled status transitions, automated validation, and PDF/ZIP claim-record generation.
- Designed automated BI pipelines: daily/monthly KPI aggregation with Redis caching, scheduled Celery Beat reporting, and partner-level financial-exposure dashboards delivered via email.
- Implemented Elasticsearch-powered full-text search and filtering across high-volume customer/claim data, with async indexing via Celery workers.
- Led infrastructure migration from Amazon EKS to ECS, cutting operational complexity and cost while keeping zero-downtime CI/CD (GitHub Actions → ECR → rolling deploys).
- Shipped partner-facing React/Tailwind dashboard with CSV bulk import/export, dynamic claim forms, and role-driven views; integrated third-party claims-update API for an external processor.