Mohammed Almeseri

Junior AI Engineer: LLM Agents, Tool Calling (MCP), RAG and Evaluation · Python

Ankara, Turkey | +90 501 614 20 10 | rastodevv@gmail.com

Portfolio: mohammed-almeseri.vercel.app | **Live product:** planovya.me

LinkedIn: linkedin.com/in/mohammed-almeseri-479323402 | **GitHub:** github.com/MohammedAlmeseri

Open to: remote roles, and Ankara-based internships or part-time work

Availability: 36+ hours a week (6 hours a day, 6 days a week; more when needed) · Available immediately

SUMMARY

Software Engineering student (B.Sc. expected 2028) who builds, tests and measures LLM applications. Built **Ken**, a desktop AI agent that reads WhatsApp, Gmail and Google Calendar through three MCP servers, using tool calling, embedding-based retrieval and safety rules enforced in code. Its evaluation harness measured **96% of factual claims grounded** in retrieved data, **0% of prompt-injection emails obeyed**, and blind-judge preference for retrieval-based drafts rising from **14% to 32%**. Also built and runs **Planovya**, a live SaaS whose AI assistant answers from each user's own data (TypeScript, Express, PostgreSQL, ~430 tests). Certified **AI Engineer for Developers Associate** by DataCamp.

PROJECTS

Ken: AI Agent for WhatsApp, Gmail and Google Calendar

Personal project · Oct 2026

github.com/MohammedAlmeseri/ken-assistant

Python (asyncio), OpenAI API, Model Context Protocol, embeddings, SQLite, NumPy, Ollama, PySide6, pytest, Git

- **Agent and tool calling:** built an agent loop that calls tools on 3 MCP servers (WhatsApp, Gmail, and a Google Calendar server I wrote), with OpenAI *gpt-6-luna* as the main model and a local Ollama model as a private fallback behind the same interface.
- **Safety enforced in code, not only in the prompt:** tool allowlist that hides delete and archive tools; every message and email wrapped as untrusted data; send-intent check, recipient guard and confirmation gate before any send; rate limits (3 WhatsApp / 5 email sends a minute) and an audit log.
- **RAG:** embedded 985 "message → reply" pairs with *text-embedding-3-small* in SQLite; each draft retrieves the 3 most similar exchanges plus the last 24 h of the chat. Also indexed the project docs so Ken answers questions about how he works.
- **Evaluation harness** with blind LLM-as-judge comparisons on 50 real replies: **96%** of factual claims (26 of 27) backed by retrieved data; **0%** prompt-injection emails obeyed; **0%** unrequested sends; judge preference **14% → 32%** (no RAG: 24%); length overshoot **+45% → +8%**; right chat found from a misspelled name in **90%** of cases, across Arabic and Latin spellings.
- **Measured model migration** (gpt-4.1-mini → gpt-6-luna), each model also judging blind: picked as closer to the real reply **60–68%** vs 18–22%; meaning similarity **51.6% → 58.5%**; evaluation cost **−76%**; accepted trade-off of median latency 1.0 s → 1.6 s.
- **Prompt iteration and debugging:** tuned one prompt over 4 measured tries (multi-line replies 0% → 78% → 4% → 52% against a 22% target) and documented that an exact rate needs code, not wording; traced a startup timeout to *npx* taking ~18 s per launch and fixed it with a pinned local server copy.
- **Testing and documentation:** 151 offline pytest tests with a guard that fails any real API call; architecture, design-brief and evaluation-results documents; total evaluation spend ≈ $0.64 over 2,187 model calls.

Planovya: AI-Powered Productivity Workspace (SaaS)

Personal product, live · Mar 2026 – present

planovya.me

TypeScript, React, Node.js/Express on Vercel, Supabase (PostgreSQL, Auth, Storage), OpenAI Responses API, Lemon Squeezy, Jest

- Built and run a full-stack SaaS for students, people with ADHD and small teams: goals, tasks, kanban, focus timers, notes, calendar, stats, study groups with messaging, leaderboards, paid Pro and Team plans, and an admin panel.
- **AI assistant grounded in the user's own data:** the server builds a per-user context (open tasks, the 10 nearest deadlines, goals, notes, 7 days of focus time) capped at 16,000 characters and cut at a line boundary with an explicit "lines omitted" notice; responses stream over SSE; the prompt forbids inventing facts missing from the data and treats injected instructions as data.

- **Security and reliability:** users reach only their own rows through Supabase row-level security and server-side JWT checks; the OpenAI key stays server-side, with a build guard that blocks secret-looking variables from the browser bundle; free-tier AI usage is capped daily and weekly on the server and recorded only after OpenAI accepts the call.
- **Testing and maintenance:** ~430 Jest test cases (chat-context builder against a fake Supabase client, AI usage caps, webhook signatures, bundle secret leaks); 236 Git commits and 22 SQL migrations; architecture, API and schema docs, plus an audit that traced high-severity feature-gating bugs to their root cause and fixed them.

NLP Practice Work

Personal / academic

- Built chatbot, question-answering and summarization prototypes in Python with Hugging Face Transformers (BERT), the OpenAI API, NLTK, spaCy and scikit-learn.

SKILLS

Languages: Python, TypeScript, JavaScript, SQL, HTML/CSS

LLM engineering: OpenAI API, tool/function calling, agent loops, Model Context Protocol (MCP), RAG, embeddings, vector similarity search, prompt engineering, LLM-as-judge evaluation, Ollama (local models), Hugging Face Transformers

Engineering practice: pytest, Jest, offline test doubles, evaluation harnesses, Git/GitHub, technical documentation, async Python (asyncio)

Data and storage: PostgreSQL (Supabase, row-level security, SQL migrations), SQLite, NumPy, pandas, scikit-learn, NLTK, spaCy

Frameworks and tools: React, Node.js/Express, PySide6, Vercel, Claude Code

AI-assisted development: review every Claude Code diff for algorithm design and integration; run unit and adversarial tests as each case needs; reject, fix or improve generated code until it meets PEP 8, DRY and project rules.

Familiar with: Pinecone, LangChain, FastAPI, LLMOps

EDUCATION

Bachelor of Science (B.Sc.), Software Engineering

Expected 2028

OSTIM Teknik University, Ankara, Turkey

CERTIFICATIONS

- **AI Engineer for Developers Associate** (certification exam) · DataCamp · Oct 2026 · ID AIEDA0011794083394
- **Associate AI Engineer for Developers** (29-hour track) · DataCamp · Sep 2026
- Introduction to Model Context Protocol (MCP) · Developing LLM Applications with LangChain · Building AI Applications with Pinecone · Software Engineering Principles in Python · DataCamp · Sep 2026
- Introduction to Embeddings with the OpenAI API · Working with the OpenAI Responses API · DataCamp · Aug 2026
- LLMOps Concepts (Jul 2026) · Prompt Engineering with the OpenAI API · Working with the OpenAI API · Working with Hugging Face (Jun 2026) · DataCamp
- The AI Engineer Course 2026: Complete AI Engineer Bootcamp · Udemy / 365 Careers · Jun 2026
- Python Essentials 1 · Data Science Essentials with Python · Cisco Networking Academy / OpenEDG · Jan 2026
- Introduction to FastAPI · Introduction to APIs in Python · Intermediate Python for Developers · DataCamp · Jan 2025

LANGUAGES

Arabic (native) · English (upper-intermediate, B2) · Turkish (beginner) · Russian (beginner)