

DARLINGTON GOSPEL

Senior AI Engineer | RAG, Agents, MCP & Production LLM Systems

Nigeria | Remote • darlingtongospel@gmail.com • +234 810 605 6545 • linkedin.com/in/darlington-gospel • github.com/Daltonic • portfolio.dappmentors.org

PROFESSIONAL SUMMARY

AI engineer with 10+ years of software experience and 6+ years building production AI systems. Builds RAG pipelines, agentic workflows, and LLM applications across web, Discord, and Telegram, covering the full path from ingestion to response generation. Also builds Model Context Protocol (MCP) servers in Python and Next.js that let chat-based LLMs read live blockchain network data and query structured databases directly, rather than wiring a one-off integration for every use case. Works with OpenAI, Claude, Gemini, DeepSeek, and Ollama, and designs for cost and latency from the start instead of retrofitting them later. Founded Dapp Mentors, where AI features are built for 5,800+ real learners rather than a demo audience.

CORE STRENGTHS

- Production-first mindset. Builds AI systems for reliability and cost efficiency at scale, not just accuracy in a notebook.
- MCP server development. Builds Model Context Protocol servers that connect chat-based LLMs directly to blockchain networks and databases, so a conversational interface can read account balances, network state, or query records as a callable tool instead of a hardcoded integration.
- Retrieval quality as the main lever. Treats RAG pipeline quality, not just model choice, as what actually moves answer accuracy.
- Predictable agent behavior. Designs agentic systems that degrade gracefully under uncertainty instead of failing silently.
- Multi-channel delivery. Ships one backend across web, Discord, and Telegram with a consistent experience.
- Cost and latency awareness. Balances model capability against operating cost as a standing design constraint, not an afterthought.

PROFESSIONAL EXPERIENCE

Founder & Lead AI Engineer | Dapp Mentors / Daltonic LLC (US-based) 2022 – Present

- Architected and deployed AI systems, including RAG, multi-agent workflows, and LLM automation, for thousands of learners.
- Built MCP (Model Context Protocol) servers in Python and Next.js that expose blockchain network information, account balances, and database queries as callable tools for chat-based LLMs.
- Built FastAPI backends with Chroma vector storage and LangChain orchestration across multiple channel interfaces.
- Designed ingestion pipelines for crawling, chunking, and indexing web content.
- Implemented context-aware responses with prompt optimization and LLM routing across providers.
- Used n8n for no-code workflow automation alongside custom-built agent and MCP tooling.
- Owned the system lifecycle: design, Docker deployment, monitoring, and metrics-based iteration.
- Published 45+ articles and 140+ videos on LangChain, RAG, and agent design.

AI Data Evaluator (Contract) | AfterQuery AI Lab 2026 – Present

- Curates and evaluates AI training datasets, benchmarking LLM outputs against defined quality thresholds.
- Tests for data integrity and structural consistency across datasets.
- Crafts realistic, high-difficulty tasks designed to challenge frontier AI models.
- Evaluates task solvability and calibrated difficulty so training data meets production-grade standards.

Senior Software Engineer | Independent / Project-Based 2019 – 2022

- Designed backend services supporting AI workflows, data-heavy applications, and real-time processing.
- Built production APIs and backend architectures using Python (FastAPI, Flask) and Node.js for ML-driven applications.
- Delivered systems handling real-time data ingestion, asynchronous processing, and decision logic.
- Deployed and maintained applications using Docker, CI/CD pipelines, and VPS infrastructure.

- Contributed to developer-facing integrations and tooling for a real-time communication platform.
- Built practical implementations alongside technical documentation for complex backend systems.
- Translated technical concepts into usable examples and integration guides for product teams.

- Led development of full-stack solutions with backend APIs and data processing systems, and trained junior developers on backend and early AI/ML fundamentals.

- Built web and mobile applications with React and Node.js, with early exposure to AI concepts.

KEY PROJECTS & ACHIEVEMENTS

Blockchain & Database MCP Servers — A set of Model Context Protocol servers built in Python and Next.js that expose live blockchain network information, account balances, and structured database queries as tools callable directly from chat-based LLM interfaces, removing the need for a custom one-off integration per use case.

RumbleCourt — Multi-agent debate system using OpenRouter LLMs for advocates and a judge model, with real-time reasoning and streaming output. An MCP server sits between the LLM chat layer and the blockchain, exposing on-chain record writes as a callable tool rather than hardcoding contract calls into the agent logic. Integrated Opik SDK for tracing and evaluation. Featured in the Commit To Change AI Agents Hackathon.

OpenBench — AI dataset curation and evaluation platform with a nine-stage automated pipeline. Runs CrewAI multi-agent probes across five attempts per task with early-exit logic to cut cost, scores output with an LLM-as-judge across eleven quality criteria, and routes across DeepSeek, Claude, GPT, Gemini, and Llama through OpenRouter. Every approved training example is stored permanently on Sia Network through content-addressed storage.

Rebit — Privacy-first video processing engine that runs entirely on local AI. Whisper produces timestamped transcripts locally, an Ollama-powered model ranks segments by virality and hook strength, and FFmpeg cuts, reformats, and captions the final clips. No proprietary cloud AI API is called at any point.

AI Content Production Engines — Suite of nine internal AI-powered content tools, each built for a specific format, from blog generation to thumbnail prompting. Handles real daily production workload for an active content operation.

AI Scraper Bot & Multi-Channel AI Assistant — Full RAG pipeline covering crawling, chunking, embedding, semantic search, and LLM response synthesis, deployed across Discord, Telegram, and a Gradio web interface for real-time conversational access.

TECHNICAL SKILLS

AI & Applied Machine Learning: LLM integration and orchestration (OpenAI, Claude, Gemini, DeepSeek, Ollama), retrieval-augmented generation, vector databases (Chroma), agentic systems and multi-agent workflows, Model Context Protocol (MCP) server development, LangChain, prompt engineering, cost and latency optimization

Backend & Infrastructure: Python (FastAPI, Flask), Node.js (Express), MongoDB, PostgreSQL, SQLite, Docker, GitHub Actions CI/CD, Linux/VPS deployment, RESTful APIs, WebSocket streaming

Frontend & Interfaces: React, Next.js, TypeScript, Tailwind CSS, Gradio, Discord and Telegram bot development

Automation: n8n (no-code workflow automation)

EDUCATION

B.Sc. Computer Science, Ignatius Ajuru University of Education, Port Harcourt — 2014 – 2018 | CGPA: 4.09 / 5.0

OPEN SOURCE & COMMUNITY

- Maintains 90+ public repositories on GitHub, followed by 350+ developers, focused on AI systems, RAG pipelines, MCP servers, and LangChain implementations.
- Leads the Dapp Mentors community of 5,800+ learners, providing free tutorials and open-source AI codebases.
- Published 45+ technical articles and produced 140+ YouTube videos on LangChain, agent workflows, and production AI deployment.