Roger Richard Demello

AI Engineer | GenAI, Agentic Systems and Backend Engineering | India | Immediate joiner

+91 94217 78898 | rogerdemello289@gmail.com | linkedin.com/in/rogerdemello | github.com/rogerdemello | rogerdemello.vercel.app

Professional Summary

AI engineer building **LLM applications** and **human-approved agentic workflows** on Python backends, with **RAG**, **evaluation**, **security** and **embedded edge AI** in scope. Three internships, 3 production apps kept at **99.5% uptime**, and a record of measuring whether the systems actually work. AWS Certified.

Technical Skills

Programming Languages: Python, TypeScript, JavaScript, SQL, Java, C++, C, Bash
Generative AI: LLMs, RAG, hybrid retrieval, AI agents, MCP, prompt engineering, LangChain, LangGraph
Machine Learning: scikit-learn, PyTorch, LightGBM, two-tower embeddings, ANN retrieval
Evals and Governance: CI rubrics, benchmark harness, schema validation, RBAC, OIDC, audit logging
Backend and APIs: FastAPI, Express, Node.js, REST APIs, OAuth, PostgreSQL, pgvector, Redis
Frontend: React, Next.js, Preact, TypeScript, Tailwind CSS, HTML, CSS
Cloud, DevOps and Testing: AWS (Cloud Practitioner), Docker, CI/CD, GitHub Actions, pytest, Git, Linux
Embedded and Edge AI: ESP32, Arduino C++, RS485/Modbus RTU, OpenCV, TensorFlow Lite, on-device inference
AI Tools: Claude Code, Cursor, Claude, Copilot, Antigravity, Gemini, ChatGPT, OpenCode

Work Experience

Innovun Global August 2026 - November 2026
AI Engineer Intern
- Built Auri's **Preact** widget, a Spanish-English **RAG** assistant for an International NGO client over 67 courses
- Implemented **hybrid retrieval**, combining ranked **pgvector** and keyword search results with reciprocal rank fusion (**RRF**)
- Prototyped a purchase-order-to-sales-order flow with **OCR** and **Gemini** extraction, human approval and a REST API for SAP
- Tested the prototype on 13 sample PO PDFs, including scans with no text layer and multi-column pages

AI LifeBOT January 2026 - June 2026
AI Engineer Intern
- Built **RAG** pipelines with document chunking for **3 production applications** serving **500+ users** across 3+ companies
- Reduced query and response **latency by 35%** by optimising retrieval and embedding paths in an inherited codebase
- Maintained **99.5% uptime** across the 3 apps with telemetry, **audit logging** and incident response
- Wrote **pytest** suites that run in **CI/CD** pipelines

CFM, RCOEM May 2025 - July 2025
Machine Learning Research Intern
- Engineered features with **Python** and **pandas** from **1,000+** raw health records, cleaning and validating the data
- Trained a **clinical classification** model with **scikit-learn**, achieving **87% accuracy** on held-out records
- Raised validation reliability **25%** through **cross-validation** across multiple splits
- Standardized data pipelines into tested, reusable components, cutting development time **30%**

Projects

Autonomous Executive Email Copilot | *Python, FastAPI, SQLAlchemy, PostgreSQL, Azure OpenAI, Docker* | Live | GitHub
- Built an inbox copilot (Gmail, Microsoft 365 via **OAuth**): the LLM drafts, deterministic code routes, a human approves
- Secured tenants with **OIDC** single sign-on, **RBAC** with 3 roles, encrypted mailbox tokens and a per-organization **audit log**
- Fact-checked each draft against the source message, so email triage still works when the LLM is unavailable
- Evaluated it with a **CI rubric** that caught the model inventing a deadline and a seeded benchmark of 3 agents, 27 episodes each
- Exported and hard-deleted tenant data on request, with a per-organization audit log of every action
- Published benchmark results that include where the LLM and multi-agent agents underperformed the heuristic

ML Guardian | *Python, FastAPI, MCP, DataHub, Google Gemini, pytest, GitHub Actions* | GitHub
- Built an **agent** on DataHub that detects **3 failure types** (stale upstreams, null-rate creep, renamed columns)
- Implemented lineage tracing to identify downstream models and dashboards at risk, then generate and run remediation code
- Integrated mcp-server-datahub over the Model Context Protocol (**MCP**) and packaged it as a reusable DataHub Skill
- Added an **offline-first** fixture mode tested in **GitHub Actions** CI
- Wrote findings back into DataHub as tags and glossary terms, with risk gauges, lineage impact radius and an agent-activity timeline on a dashboard
- Ran on bundled fixtures with no Docker or API keys, switching to a live DataHub with one environment variable

music-recsys | *Python, PyTorch, LightGBM, FastAPI, MLflow, Redis, Prometheus, Grafana, Docker* | GitHub
- Developed a two-stage recommender: **two-tower embeddings** feed **ANN retrieval** into a **LightGBM** ranker served by FastAPI
- Lifted **Recall@10** from 0.094 to 0.105 on a held-out temporal split of synthetic data, tracked with **MLflow**
- Measured HNSW ANN recall@20 of 1.0 against exact nearest-neighbour search on the same synthetic data
- Decoupled **5 components** (cache, event bus, store, ANN index, registry) behind **Protocol** interfaces with swappable backends
- Added an A/B simulator, a model registry and retrain and embedding-refresh jobs around the ranker
- Built it to run CPU-only on Windows with no external services, designed to scale to Docker, Kafka and Kubernetes

NPK Crop Recommendation System | *Python, scikit-learn, pandas, micromlgen, TensorFlow Lite, Arduino/C++, ESP32, RS485/Modbus RTU* | GitHub
- Built a system that recommends the best crop from soil **Nitrogen, Phosphorus and Potassium** levels read from an RS485 sensor over **Modbus RTU** through a MAX485 converter on an **ESP32**
- Trained a **Decision Tree** classifier in Python with scikit-learn and deployed it to the ESP32 as a C++ header for **fully on-device inference** with no internet
- Also ran **int8 TensorFlow Lite** variants on the ESP32 alongside the Decision Tree
- Exported the trained model to C++ with micromlgen so the recommendation happens in the field in seconds with no network

Automatic Solar Panel Cleaning System | *Python, FastAPI, OpenCV, NumPy, Arduino/C++, ESP32, ESP32-CAM, TB6612 motor driver* | GitHub
- Built and physically tested a prototype that detects dust on solar panels with **ESP32-CAM** and **OpenCV** and triggers cleaning through motor control
- Connected **two ESP32** microcontrollers (camera and motor) over WiFi to a **FastAPI** server that handles image processing
- Added configurable dust and contrast thresholds and IR sensors for panel-edge detection so cleaning cycles stay safe
- Programmed the camera and motor ESP32 boards in Arduino C++ and drove the motors through a TB6612 driver

Education

Shri Ramdeobaba College of Engineering and Management (RCOEM) 2022 - 2026
B.Tech, Electronics & Communication Engineering **8.9/10**
Minor: **Artificial Intelligence & Machine Learning** (2022 - 2026) **Grade AA**

Certifications and Awards

AWS Certified Cloud Practitioner - Amazon Web Services (October 2025)
Hackathon Finalist - Paytm x Sarvam x Logitech AI National Hackathon
Second Place - ByteSize Sage AI National Hackathon