

AI Engineer

Career Launch Programme

Six months to build and deploy a serious AI application.

Programme at a glance

6

MONTHS

180

CONTACT HOURS

Hybrid

FORMAT

25

MAX BATCH

Evenings and weekends. A project-led programme for students and recent graduates building modern AI applications with language models, retrieval, agents, MCP, deployment, evaluation and production engineering.

Target roles

- Junior AI Engineer
- AI Application Developer
- GenAI Developer
- RAG Developer / RAG Engineer
- AI Automation Developer
- AI Engineering Intern
- Junior Software Engineer working on AI applications

Job titles vary between companies. The programme focuses on engineering capabilities rather than one job title.

Programme philosophy

Work an employer can inspect, run and question you about. No two students complete the same capstone. Realistic commitment: approximately 11–13 hours per week including project time.

What students leave with

A live AI product

A deployed application that can be opened and tested through a real URL.

A six-month GitHub history

A repository that shows how the system developed week by week.

A technical design document

A clear explanation of the architecture, decisions, alternatives considered and engineering trade-offs.

A technical walkthrough

A recorded ten-minute explanation of your system, architecture, important decisions and results.

Mini-builds + individual capstone

Short practical builds during modules (LLM APIs, RAG, agents, MCP, Dockerised APIs, evaluation harnesses) teach individual skills. Each student develops one substantial AI application over six months that combines application logic, retrieval, tools, agents, databases, evaluation, monitoring, deployment and documentation.

The mini-builds teach the engineering skills. The capstone shows you can combine them into a working system.

MONTHS 1–2 · FOUNDATIONS

Module	Hours	Topics
Python for AI engineering	12	object-oriented programming · typing · async programming · environments · packaging · APIs · JSON · file handling · error handling · debugging basics
Modern AI landscape	10	transformers · tokens · embeddings · attention · context · model behaviour · model limitations · cost · latency · selecting models based on application requirements
Prompt engineering	14	structured prompting · structured output · tool calling · prompt templates · prompt versioning · reasoning workflows · systematic prompt evaluation · guardrails
LLM engineering	16	OpenAI SDK · Anthropic SDK · Gemini API · streaming · function calling · structured outputs · guardrails · tokens · cost control · rate limits · local models · Ollama
Embeddings and vector search	8	embeddings · similarity search · indexing · metadata search · hybrid search · ChromaDB · FAISS · Pinecone

MONTHS 3–4 · AI SYSTEMS

Module	Hours	Topics
Retrieval Augmented Generation Build: LangChain · LlamaIndex	20	document ingestion · document parsing · chunking · embeddings · retrieval · reranking · context construction · memory · hallucination reduction · citation · evaluation
AI Agents Build: LangGraph · CrewAI	20	agent architecture · planning · tool use · state · memory · reflection · multi-step workflows · human-in-the-loop design · reliability · failure handling
Multi-agent systems	8	specialised agents · delegation · collaboration · orchestration · state management · event-driven workflows · communication between agents · understanding when a single-agent or conventional software architecture is preferable
Model Context Protocol Build: MCP Python SDK · MCP Inspector	12	MCP architecture · servers · clients · resources · tools · transport · authentication · security · file access · database access · workflow integration
Deployment and software delivery	12	FastAPI · APIs and HTTP · application structure · environment variables · deployment · Docker · Git · GitHub · GitHub Actions · CI/CD · cloud deployment · basic authentication and access-control patterns

MONTHS 5–6 · PRODUCTION ENGINEERING

Module	Hours	Topics
Evaluation and testing	12	evaluation datasets · qualitative evaluation · quantitative evaluation · regression testing · testing retrieval · testing prompts · measuring system changes · comparing versions · unit and integration testing where appropriate
Observability, reliability and cost	10	tracing · logging · monitoring · latency · failures · model usage · token economics · cost analysis · debugging production behaviour
Security and responsible AI	10	prompt injection · jailbreaks · data leakage · unsafe tool access · personal information · access control · OWASP guidance for LLM applications · India's Digital Personal Data Protection framework · human oversight · safe failure
Capstone engineering	8	code review · architecture review · evaluation · reliability testing · security review · deployment review · documentation · technical design document
Interview and portfolio	8	technical resume · GitHub presentation · portfolio presentation · system design discussion · explaining engineering decisions · debugging questions · mock technical interviews · project walkthrough practice

Tools and technologies

Build with

Language and model SDKs

Python · OpenAI SDK · Anthropic SDK · Gemini API

Frameworks

LangChain · LangGraph · CrewAI

Retrieval

ChromaDB

Protocol

MCP Python SDK · MCP Inspector

Application and deployment

FastAPI · Docker · Git · GitHub · GitHub Actions

Data

PostgreSQL

Work with

Models

Claude · GPT · Gemini · Llama · Mistral · Qwen

Local and open models

Ollama · Hugging Face

Frameworks

LlamaIndex · Pydantic AI

Vector stores

FAISS · Pinecone

Interfaces

Streamlit · Gradio

Cloud

Render · Railway · Azure AI

Data

MongoDB

Observability

tracing tools for LLM applications · evaluation tools for LLM applications

Know and compare

Frameworks and agent platforms

AutoGen · Haystack · OpenAI Agents SDK · Google ADK

Vector databases

Qdrant · Weaviate

Cloud AI platforms

AWS Bedrock · Google Vertex AI

Models

Grok · relevant frontier and open-model releases available during the programme

AI engineering changes quickly. Tools may be updated during a batch while preserving the engineering concept being taught.

Capstone domains and assessment

Example capstone domains

- **Healthcare:** A medical knowledge assistant that works from approved reference material, retrieves and cites evidence, identifies unsupported questions, records feedback and measures retrieval quality. It is not designed to diagnose patients independently.
- **Education:** A learning system that records student performance, identifies weak concepts, retrieves relevant learning material, adapts subsequent questions and provides teachers with evidence of student progress.
- **Human Resources:** A recruitment support system that extracts candidate evidence, compares it against defined job requirements, identifies missing information and provides explainable recommendations for human review. Discriminatory automated hiring logic is not part of the programme.
- **Legal:** A contract intelligence system that identifies obligations, dates, unusual clauses, defined terms and risks while linking every extracted point to its source text.
- **Finance:** A financial research system that reads financial statements, filings and approved source documents, extracts relevant information, identifies changes and provides evidence-backed analysis with appropriate controls. It is not marketed as autonomous financial advice.
- **Retail:** A catalogue-aware shopping system connected to structured product information that understands constraints, compares appropriate products, explains recommendations and handles changing inventory.

Students are not restricted to these domains.

Assessment checkpoints

Month 1 — Foundations check — Python, APIs, data handling and core software skills.

Month 2 — Independent build — A small working AI application built with limited assistance.

Month 4 — Engineering assessment — Students build a small working system from an empty project and complete a debugging task involving unfamiliar code. Part of this assessment is completed without AI assistance so that students and instructors can identify whether the underlying engineering skills are strong enough. Students who need additional support enter a structured remediation period before moving forward.

Month 6 — Capstone review — The final application is evaluated on whether it works, architecture, code quality, evaluation, reliability, deployment, documentation and technical explanation.

Placement support

Resume and GitHub preparation, portfolio presentation, mock technical interviews, system-design practice, employer demo day, and introductions where relevant opportunities exist.

Placement support is provided. Employment cannot be guaranteed.

Programme design

Siddharth T. Janakiraman, Co-founder, Intraneurals. Techno-legal background across intellectual property, technology commercialisation and regulatory practice. Works with founders, innovators and university-linked programmes on intellectual property, innovation and protection of technology products.

Selvaganesh J, Co-founder and Chief Executive, Intraneurals. Software delivery background including TCS and Capgemini, with enterprise project experience across large engineering teams.

Teaching team: Core technical modules are delivered by experienced faculty members and AI practitioners who currently work with AI and software systems in industry.

Industry mentors: Working engineers and industry practitioners contribute through selected technical sessions, project reviews, capstone feedback and mock interviews.

Fees and admission

Founding Batch — ₹49,500

First cohort. Includes programme access and reasonable programme-related API and cloud usage during the six-month course. Usage is provided for normal course, lab and capstone requirements subject to the programme's reasonable-use policy.

Standard Fee — ₹65,000

From subsequent batches.

Instalments

Instalment payment available. Contact admissions for the current schedule.

Admission process

1. • **Enquire or apply** — Apply through the website or contact Intraneurals through WhatsApp.
2. • **Basic entry check** — A short assessment of basic Python, reasoning and readiness. Approximate duration: 40 minutes.
It is used to identify whether the applicant is ready for the programme or would benefit from the Python bridge first.
3. • **Short programme conversation** — A brief discussion about the student's background, goals, available time and potential capstone interests.
4. • **Confirm admission** — Eligible applicants confirm their seat and receive onboarding instructions.

Batch size capped at 25. Next batch: [month to be announced].

Enquire about the AI Engineer Career Launch Programme

WhatsApp: +91 98409 41910

Email: admissions@intraneurals.com

Web: intraneurals.com/ai-engineer/