School of Artificial Intelligence & Software Engineering

GRADUATE PROFESSIONAL PROGRAM

PROFESSIONAL GRADUATE CERTIFICATE

Full-Stack AI Engineering

An end-to-end program that takes engineers from software fundamentals to designing, building, deploying, and operating production-grade AI systems.

STRUCTURE	PARTS	CREDITS	APPLIED WORK
12 courses	5 phases	45	6+ projects

LEVEL	OUTCOME
Graduate	3+ yrs capability

Program Overview

The AI engineering field rewards builders who can carry an idea from a blank editor to a monitored, secure product in the hands of users. This program is engineered around that arc. Learners first master the fundamentals of software, data, and systems, then layer machine learning, large language models, and autonomous agents on top of a genuinely production-ready foundation.

Every course pairs concepts with a guided build and a hackathon or project. From the second course onward, all work is version-controlled, tested, containerized, and deployed to a live URL — the standard expected of professional engineers. The program favors depth over breadth: graduates finish able to build, deploy, and operate real systems, not merely name the tools.

At a Glance

DELIVERY

Concept → guided build → hackathon, cohort-based

PREREQUISITE

Basic programming comfort & logical problem-solving; all else built from Course 501

ASSESSMENT

Continuous projects, four graded hackathons & a capstone

CREDENTIAL

Professional Graduate Certificate in Full-Stack AI Engineering

Program Learning Outcomes

1. Build full-stack web applications with React / TypeScript and FastAPI
2. Model and query data across SQL, NoSQL, and vector stores
3. Architect systems at scale (HLD & LLD) and defend them in interviews
4. Train and evaluate ML models across the modern deep-learning stack
5. Engineer LLM applications: prompting, RAG, fine-tuning, and evaluation
6. Design agentic AI systems with tools, memory, and orchestration
7. Deploy, monitor, secure, and cost-optimize AI in production (DevOps & LLMOps)
8. Ship a portfolio-grade capstone and demonstrate interview readiness

CURRICULUM STRUCTURE

Twelve Courses, Five Parts

A deliberate progression from engineering foundations through full-stack delivery, system design at scale, applied AI, and finally production operations and career practice.

PART I Foundations

Courses 501 – 510 · 8 credits

AIE 501 — Software Engineering Foundations

3 CR

Build the mental model and core skills every engineer needs before specializing.

CORE CONCEPTS

- **Fundamentals of software engineering** — the SDLC, requirements, trade-offs, technical debt
- **Programming** — Python as the primary language (data structures, OOP, idioms); JavaScript basics for the web
- How software systems layer together — client, server, API, database, cache, cloud
- The map of the field — problem solving, backend, frontend, data, cloud, DevOps, AI/ML, and how they connect

ENGINEERING CRAFT

- Problem solving & data structures — arrays, strings, hash maps, recursion, complexity (Big-O)
- Clean code, readability, and code-review basics

STUDIO Environment setup; small Python programs and a command-line tool.

AIE 502 — Engineering Toolkit & Delivery

2 CR

Learn to ship — every later project is versioned, tested, containerized, and deployed from day one.

- **Git & GitHub** — branching, pull requests, the code-review workflow
- Command line & Linux basics
- **Testing foundations** — unit / integration / e2e, pytest, mocking, the TDD mindset (revisited throughout)
- **Containers 101** — Docker images, containers, Dockerfiles, docker-compose
- **Your first deploy** — ship a simple app to a live URL
- Debugging, logging, and reading stack traces

> *Delivery skills are placed early so no project is ever stuck on localhost — every hackathon reaches a live URL.*

AIE 510
3 CR

Data Systems & Modeling

Store and query data well across relational, non-relational, and vector stores.

- **Relational databases & SQL** — modeling, joins, transactions, normalization
- **NoSQL** — document, key-value, and column stores, and when to use each
- Schema & data modeling; indexing and query optimization
- Caching with Redis; migrations and schema versioning
- **Vector databases** — embeddings preview, Qdrant, and pgvector on Postgres; the wider landscape (Pinecone, Weaviate, Chroma, Milvus)

PROJECT Design tables and schema for a given use case — including indexes and a caching layer.

PART II Building Full-Stack

Courses 520 – 530 · 10 credits

AIE 520
4 CR

Backend Systems Engineering

Build robust, secure, production-shaped APIs.

ARCHITECTURE & APIS

- Backend architecture & the MVC pattern
- API design; the evolution of APIs — SOAP, REST, GraphQL; SaaS API patterns
- **FastAPI with Python** — routing, validation (Pydantic), dependency injection

PRODUCTION ESSENTIALS

- Async programming (asyncio) — essential for slow LLM and network calls
- Authentication & authorization — JWT, OAuth2, sessions
- API security — rate limiting, CORS, input validation, secrets handling
- Streaming responses (SSE / WebSockets) — for token-by-token LLM output
- Background jobs & task queues (Celery + Redis); ORMs (SQLAlchemy), migrations, webhooks; API testing

HACKATHON Build a backend application — with authentication, tests, and a Docker image.

AIE 525
3 CR

Frontend & Interface Engineering

Build modern, responsive UIs — including AI chat interfaces.

- **HTML, CSS, JavaScript** fundamentals; TypeScript — the default for serious React work
- **React** — component model, hooks, routing; Angular overview for context
- State management (Zustand / Redux / Context) and data fetching (React Query / SWR)
- Styling with Tailwind CSS / a component library
- **Building a streaming chat UI** — the signature frontend skill for AI apps

PROJECT A React + TypeScript web application, including a streaming chat interface.

AIE 530
3 CR

Full-Stack Systems Integration

Wire front end, back end, and data into one deployed product.

- **End-to-end full-stack application** — frontend + FastAPI + database + auth
- **Enterprise application patterns** — configuration, environments, error handling, observability basics
- Connecting the pieces and deploying with Docker

HACKATHON I Ship a complete, deployed full-stack web application.

PART III Designing at Scale

Course 540 · 4 credits

AIE 540
4 CR

System Design — HLD & LLD

Design systems that scale — and explain them in interviews.

HIGH-LEVEL DESIGN

- Architecture design, distributed systems, DNS, load balancing, consistent hashing
- CAP theorem, caching, CDNs, transactions, SQL vs NoSQL, scalability
- ZooKeeper + Kafka (distributed messaging), microservices, security, case studies

LOW-LEVEL DESIGN

- Object-oriented programming, SOLID principles, design patterns
- UML diagrams, schema design, API design

HACKATHON II Design and partially implement a scalable system for a real-world scenario.

PART IV Applied AI Engineering

Courses 550 – 570 · 13 credits

AIE 550
4 CR

Machine Learning Foundations

Understand and build ML / deep-learning models before layering LLMs on top.

- **ML overview** and the end-to-end ML workflow; supervised & unsupervised learning
- **Data engineering for ML** — cleaning, preprocessing, feature engineering, splits, data pipelines (ML is ~80% data work)
- Neural networks; CNNs; RNNs; Transformers — the architecture behind modern AI
- Model evaluation & metrics; overfitting and regularization
- Experiment tracking with MLflow / Weights & Biases

PROJECT + PRACTICUM Train and evaluate a model end-to-end.

AIE 560
5 CR

Large Language Models & Generative AI

Engineer LLM applications the way industry actually does.

LLM FUNDAMENTALS

- LLMs, SLMs, and BERT-family models — how they differ
- Tokenization, context windows, temperature & sampling parameters
- Calling LLM APIs (OpenAI / Anthropic / open models) — messages, streaming, cost & latency

CORE AI-ENGINEERING SKILLS

- **Prompt engineering** — zero/few-shot, chain-of-thought, structured prompting, templates & versioning
- Function / tool calling & structured (JSON) outputs — the foundation agents build on
- Embeddings & semantic search; vector databases in practice

RETRIEVAL & ADAPTATION

- **RAG in depth** — chunking, retrieval strategies, reranking, hybrid RAG; evaluation with RAGAS
- **Fine-tuning in depth** — PEFT, LoRA / QLoRA, quantization; the prompt vs RAG vs fine-tune decision
- Serving your own models — vLLM, Ollama, TGI; GPU basics; multimodal (vision / audio)

HACKATHON III Build and evaluate a production-style RAG application.

AIE 570
4 CR

Agentic AI Systems

Build autonomous, tool-using AI systems.

- **Agents vs. agentic systems** — and when (and when not) to use them
- Planning & reasoning patterns — ReAct, tool use; agent memory (short-term, long-term, vector)
- **Model Context Protocol (MCP)** — connecting tools and data to models
- Frameworks — LangChain / LlamaIndex, LangGraph, CrewAI, AutoGen
- Multi-agent orchestration & human-in-the-loop; observability & tracing with LangSmith
- Guardrails and agent evaluation

HACKATHON IV Build a multi-step agent that uses tools, memory, and external data.

PART V

Production & Professional Practice

Courses 580 – 590 · 10 credits

AIE 580

4 CR

Cloud, DevOps & LLMOps

Deploy, automate, monitor, secure, and cost-optimize AI in production.

CLOUD & DEVOPS

- Cloud fundamentals; AWS core services; serverless
- Containers in production — Docker deep-dive + Kubernetes basics
- CI/CD pipelines — Git, Jenkins, and GitHub Actions; Infrastructure as Code (Terraform)
- Monitoring & observability (Prometheus / Grafana); cloud AI services (AWS Bedrock / SageMaker)

LLMOPS & AI SECURITY

- Model deployment & versioning; offline + online evaluation
- Cost & latency monitoring, semantic caching, fallbacks/retries, drift; guardrails & content moderation
- AI security — prompt injection, OWASP Top 10 for LLMs, PII / data privacy, secrets management

AIE 590

6 CR

Capstone & Professional Practice

Prove it — and get hired.

- **Capstone project** — a portfolio-grade, end-to-end AI application (full-stack + RAG or agents, deployed and monitored)
- System design interview preparation; DSA / coding interview preparation
- Resume, GitHub, and portfolio polish; mock interviews and feedback

CAPSTONE A shippable capstone in your portfolio and demonstrated readiness for AI engineering roles.

Assessment & Capstone

- Continuous, project-based assessment in every course
- Four graded hackathons of rising complexity across Parts II–IV
- Deploy-to-URL requirement from Course 502 onward
- A cumulative capstone reviewed as a portfolio-grade deliverable
- Interview-style defense of system-design and capstone work

Career Outcomes

Graduates target roles such as AI Engineer, Full-Stack Engineer (AI), LLM / Applied ML Engineer, and AI Platform / MLOps Engineer. The program is built to develop capability equivalent to 3+ years of industry experience, backed by a deployed portfolio, system-design fluency, and structured interview preparation.

Reference Technology Stack

LANGUAGES

Python · TypeScript / JavaScript · SQL

BACKEND

FastAPI · SQLAlchemy · Celery · Redis

FRONTEND

React · TypeScript · Tailwind · React Query

DATA

PostgreSQL (+ pgvector) · a NoSQL store · Qdrant · Redis

ML / AI

PyTorch · Hugging Face · scikit-learn · MLflow / W&B

LLM / AGENTS

LLM APIs + open models · LangChain / LlamaIndex · LangGraph · CrewAI · LangSmith · vLLM / Ollama · MCP

DEVOPS / CLOUD

Git · Docker · Kubernetes · GitHub Actions / Jenkins · Terraform · Prometheus / Grafana · AWS (Bedrock / SageMaker)

GUIDING PRINCIPLE

Depth over breadth — graduate able to build, deploy, and operate real systems.