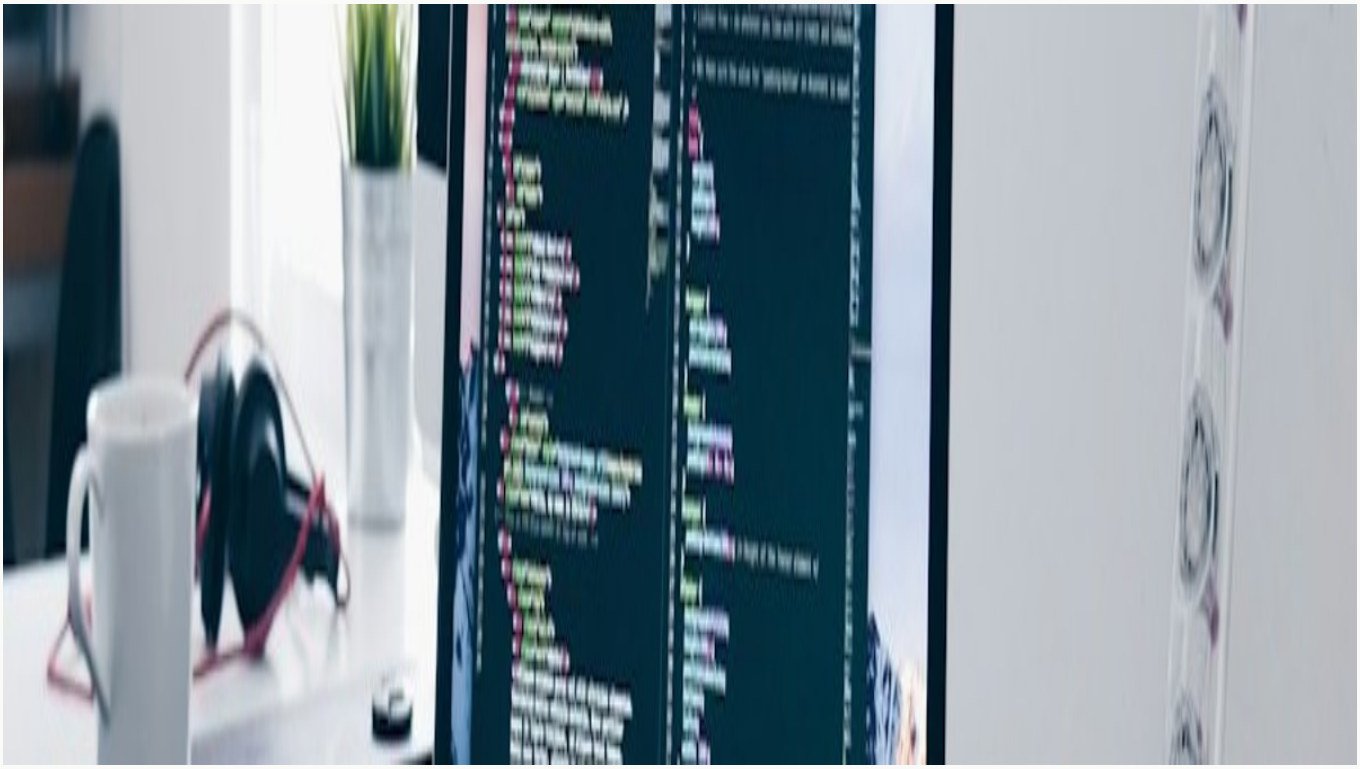
AXIA LEARNING / AIEN

Diploma in AI Engineering

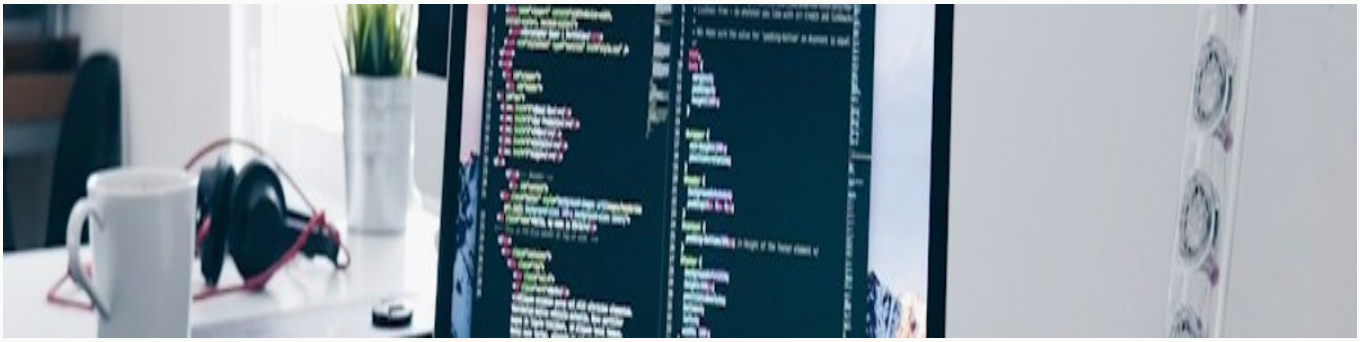
Five months. Practical learning. A clear next step.



20 weeks | **220 hours** | **Live classes**

Developing Talent. Building Intelligent Solutions.

Your pathway



Focus and scope. Applied AI application engineering: build, evaluate and deploy bounded AI services using Python, existing models, retrieval and controlled tools. Training foundation models and advanced ML research are outside the core scope.

Entry requirements. Required: working Python functions, loops, collections, file/JSON handling, basic Git, HTTP concepts and arithmetic/probability basics. Applicants complete a practical entry test before admission.

Pre-course preparation. Applicants who need preparation complete a separate 30–40-hour Python/API/Git bridge and pass a retest. This preparation is outside the five-month programme; complete beginners should not skip it.

Primary tools. Core: Python, Git/GitHub, pandas, scikit-learn, pytest, FastAPI, a model SDK, SQLite/PostgreSQL and one retrieval/vector-search solution. Use a hosted small model or a small open model; GPU work and fine-tuning are optional extensions.

Career direction. Portfolio preparation for junior applied-AI/LLM application developer roles for candidates with the stated prerequisites; not a substitute for broad software-engineering experience.

What you'll be able to do

LO1. Build reproducible Python services with tested interfaces, configuration and basic deployment.

LO2. Prepare datasets and compare a simple predictive baseline using valid train/test separation.

LO3. Integrate model APIs with structured outputs, validation and failure handling.

LO4. Build retrieval-augmented responses with source attribution and document-access boundaries.

LO5. Evaluate quality, latency, cost and security against an explicit test set.

LO6. Deploy, monitor and document a bounded AI application with an operational runbook.

Five-month overview

Month	Focus	Main assessment
1	Engineering and machine-learning foundations	Tested predictive baseline service with sound data splitting and error analysis.
2	LLM application development	LLM application with structured outputs, input validation and measured runtime behaviour.
3	Retrieval and bounded tool use	Retrieval/tool service with citations, no-evidence handling and access-isolation tests.
4	Evaluation, security and deployment	Separate staging-deployment and optimisation exercise with regression and security evidence.
5	Capstone build and engineering defence	Final capstone and individual practical

MONTH 1 / 05

Engineering and machine-learning foundations



1. Reproducible Python engineering

Skills & concepts

Use environments, packages, Git branches, typing, logging and testable project structure.

Practical evidence

Package an existing Python script with tests and reproducible setup instructions.

2. Data preparation and statistical foundations

Skills & concepts

Profile datasets, handle missing values and understand sampling, leakage and feature/target separation.

Practical evidence

Build a cleaned dataset and document a valid evaluation split.

3. Supervised-learning baseline

Skills & concepts

Train a simple regression or classification model and choose a suitable metric.

Practical evidence

Compare a trained baseline with a naive baseline on held-out data.

4. Model service and engineering review

Skills & concepts

Serve a small prediction endpoint, validate inputs and report uncertainty and limitations.

Practical evidence

Submit Month 1 tested baseline service and evaluation report.

MONTH 2 / 05

LLM application development



5. LLM architecture and model APIs

Skills & concepts

Explain tokens, context limits, inference, embeddings and basic transformer concepts without training a model.

Practical evidence

Integrate one model API and measure token use and latency on sample requests.

6. Structured outputs and prompt versioning

Skills & concepts

Design input contracts, structured outputs, schema validation and prompt/model version records.

Practical evidence

Build a structured extraction endpoint with invalid-output handling.

7. Application state and failure handling

Skills & concepts

Manage sessions, rate limits, timeouts, retries and configuration without exposing secrets.

Practical evidence

Implement robust request handling and an auditable conversation or task state.

8. AI service integration

Skills & concepts

Build a small UI/client around a tested AI API and compare a non-AI baseline.

Practical evidence

Submit Month 2 application, test cases and a cost/latency report.

MONTH 3 / 05

Retrieval and bounded tool use



9. Retrieval foundations

Skills & concepts

Ingest permitted documents, chunk text, create embeddings and compare retrieval approaches.

Practical evidence

Build a document index and label relevant passages for sample questions.

10. Grounded response generation

Skills & concepts

Retrieve context, cite supporting passages and abstain when evidence is missing.

Practical evidence

Create a question-answering service with source links and a no-evidence fallback.

11. Access boundaries and retrieval evaluation

Skills & concepts

Filter document access, test leakage and separate retrieval quality from answer quality.

Practical evidence

Evaluate retrieval with labelled cases and test cross-user document isolation.

12. Bounded tools and agent workflows

Skills & concepts

Use tool schemas, allowlists, approval gates and deterministic orchestration.

Practical evidence

Submit Month 3 retrieval/tool application with a documented threat model.

MONTH 4 / 05

Evaluation, security and deployment



13. Evaluation harnesses

Skills & concepts

Build representative datasets, compare versions and combine automatic checks with human review.

Practical evidence

Run a regression evaluation set and investigate failures rather than only reporting averages.

14. AI security and reliability

Skills & concepts

Test prompt injection, sensitive-data leakage, untrusted tool output and resource exhaustion in a sandbox.

Practical evidence

Implement mitigations and document residual risks with adversarial test evidence.

15. Deployment and observability

Skills & concepts

Containerise or package the service, automate checks and log quality, cost and latency signals.

Practical evidence

Deploy a staging service with authentication/access controls and a rollback procedure.

16. Optimisation and engineering trade-offs

Skills & concepts

Compare caching, smaller models and retrieval/prompt changes without leaking private data.

Practical evidence

Submit Month 4 reliability/optimisation report; approve a separately scoped capstone.

MONTH 5 / 05

Capstone build and engineering defence



17. Capstone architecture and baseline

Skills & concepts

Define the task, permitted data, baseline, acceptance gates and evaluation set.

Practical evidence

Submit a technical design, data contract and an initial runnable baseline.

18. Capstone implementation

Skills & concepts

Build retrieval or controlled tool use, interfaces and reproducible deployment.

Practical evidence

Demonstrate a complete staging application with automated component tests.

19. Capstone evaluation and hardening

Skills & concepts

Test representative and adversarial cases, measure cost/latency and fix critical failures.

Practical evidence

Deliver an evaluation report, access tests and release-readiness review.

20. Engineering defence and handover

Skills & concepts

Explain implementation decisions and debug an unseen change independently.

Practical evidence

Submit repository, runnable service, runbook, model/data notes and individual practical defence.

Your assessment checkpoints

Checkpoint	Assessed project	Outcome coverage
Week 4	Tested predictive baseline service with sound data splitting and error analysis.	LO1, LO2
Week 8	LLM application with structured outputs, input validation and measured runtime behaviour.	LO1, LO3
Week 12	Retrieval/tool service with citations, no-evidence handling and access-isolation tests.	LO4
Week 16	Separate staging-deployment and optimisation exercise with regression and security evidence.	LO5, LO6

The capstone and final individual practical together verify all six outcomes. Instructors use the weekly evidence to identify gaps early; the mapping above does not limit remediation to those weeks.

Your final capstone

Build and deploy a bounded document-support, internal-knowledge or service-assistance application. Include retrieval or constrained tools, source-aware responses, evaluation and operational controls.

Capstone acceptance criteria

Maintain at least 50 evaluation cases, including at least ten failure/adversarial cases and ten access-boundary checks; document overlap. Compare with a defined baseline and report task success, failure categories, latency and estimated cost per successful task. Enforce an explicit access model; demonstrate that users cannot retrieve another user's restricted documents. Provide schema/input checks, secret management, automated tests, failure handling and a rollback/runbook. A reviewer must reproduce the application from the repository; defend and debug an unseen change individually.

Evidence pack. Include a short brief, source/permissions record, versions, practical artefact, testing or user-research evidence, limitations, handover instructions and a record of material AI assistance. Include a 5–8-minute demonstration and an individual 15-minute defence inside the final week's facilitated schedule.

How the programme works

Programme model and delivery

Programme structure. Six separate pathways, each lasting 20 teaching weeks, organised into five four-week learning blocks. A student enrols in one diploma. The five-month calendar should be confirmed against holidays and local delivery dates. This is a proposed institutional curriculum blueprint, not an accreditation decision.

Audience and admission. Mixed cohort: beginners and career changers can enter the non-coding tracks after an orientation. Technical tracks use practical entry checks and pre-course bridging where required. AI Engineering requires Python/API/Git competence before Week 1. Assess applicants by demonstrated skills rather than degree title alone.

Weekly workload: 11 hours. 3 hours of interactive teaching; 2 hours of supervised practical work; 1 hour of critique, review or mentoring; 5 hours of independent reading, building and assessment preparation. Total: 220 learning hours per diploma, including 120 facilitated hours and 100 independent hours. Projects and assessment preparation are included, not added on top.

How each week runs. Begin with a concrete work problem and a short retrieval exercise; teach the relevant concepts; demonstrate a worked example; complete a supervised lab; finish an individual deliverable and receive feedback. Use a different dataset or brief for independent work so students cannot pass by copying the demonstration.

Shared foundation without losing track time. Week 1 in each course includes AI limitations, data privacy, responsible tool use and evidence checking through a track-specific exercise. Communication, version history, documentation and critical thinking recur throughout. Do not add a separate generic month that displaces the core discipline.

Progression gates. Week 4: demonstrate foundational work independently. Week 8: complete a usable core project. Week 12: show role-specific AI application and evaluation. Week 16: demonstrate integrated work and obtain approval for a separate capstone. Weeks 17–20: deliver, test and defend the capstone. Learners who miss a gate receive targeted remediation and a reassessment plan.

AI use in assessment. Allow AI in normal labs if students disclose material use and verify outputs. Keep a brief record of prompts, generated material and corrections. Do not accept fabricated interviews, invented datasets, unrun tests or unsupported impact claims. The final practical uses a fresh task, allows reference documentation and excludes AI-generated solutions; the oral defence verifies individual understanding.

Data and lab practice. Use public, licensed, synthetic or appropriately consented data. Do not place confidential organisational or personal data in external AI tools. Cybersecurity practice takes place only in written-scope isolated labs; automation uses sandbox accounts and approval before external side effects. Students keep a source and permissions register.

Capstone format. Prefer individual work for a five-month part-time diploma. If a team of up to three is approved, each learner keeps a contribution log and submits an individually assessed component and defence. A cross-track showcase is optional; a

learner's grade must not depend on another track delivering its component.

Portfolio and employability. Career preparation is embedded in project write-ups and mentor review. Each student finishes with four graded monthly project artefacts, one capstone case study, a concise CV/profile and a rehearsed demonstration. Describe evidence of competence honestly; the curriculum does not promise employment or advanced professional expertise.

Assessment and progression

Assessment	Weight	Evidence
Weekly practical work, Weeks 1–16	20%	Sixteen individual practical submissions; mark correctness, reasoning and verification. Each contributes 1.25%.
Four monthly projects, Weeks 4, 8, 12 and 16	30%	Each contributes 7.5%. Grade the integrated project against its outcome map; do not award the same points twice for a weekly lab.
Capstone, Weeks 17–20	35%	A separate brief with implementation/prototype, evaluation evidence, documentation and individual defence. Reusing earlier techniques is allowed; copied project submissions are not a new capstone.
Final individual practical, Week 20	15%	An unseen role-specific task, normally 90 minutes, followed by a short explanation of decisions. Scheduled inside the weekly facilitated hours.

Proposed capstone rubric (100 points within the capstone component):

Criterion	Points
Problem understanding and scope	10
Discipline-specific execution	35
Testing, evaluation and evidence	20
User/business relevance	10
Responsible practice and risk controls	10
Documentation and reproducibility/handoff	10
Presentation and defence	5

The discipline-specific acceptance criteria in each course are additional gates. A polished presentation cannot compensate for missing core functionality, unsupported findings or lack of individual understanding.

Delivery & learning support

Delivery preparation and quality assurance

Before enrolment. Publish entry requirements, assessment rules, actual tool/API charges and device requirements. Run the diagnostic and bridge. Check access to every compulsory tool on the learner's device. No learner should discover a mandatory paid subscription halfway through the course.

Teaching assets required. For every week prepare one instructor plan, a worked example, a supervised lab brief, an independent task, starter files or data, an answer/review guide and a marking checklist. For each project prepare a brief, rubric, sample acceptable evidence and an individual-defence prompt.

Instructor and mentor support. Use a named practitioner lead per track and approximately one practical mentor per 20–25 learners; reduce the ratio for engineering or complex lab sessions. Offer a recorded session or equivalent guided task for

learners with connectivity constraints.

Device and access baseline. Recommend a laptop with at least 8 GB RAM; provide browser labs or a higher-memory option for virtualisation-intensive work. Cybersecurity learners need isolated lab access; Power BI learners need compatible Windows access or a preselected cohort-wide alternative. AI Engineering does not require students to purchase a GPU.

Tool-cost control. Choose one main tool per capability. Provide sandbox datasets, shared lab infrastructure, bounded API quotas and a mock/offline fallback where feasible. Approve spending limits before live integrations. Verify current pricing and licences before each cohort rather than printing unverified prices in the syllabus.

Feedback and quality assurance. Return weekly feedback within seven days. Moderate a sample of marked work across instructors, verify final individual competence and gather feedback at Weeks 4, 8, 12, 16 and 20. Update tool-specific instructions before the next cohort while keeping learning outcomes stable.

Completion rules proposed for adoption. Require at least 60% overall, 60% in the capstone and 60% in the final practical, plus satisfactory evidence for every core learning outcome. Offer targeted resubmission for unmet outcomes. Require participation in 80% of facilitated activities or approved equivalent tasks; publish any local policy changes before enrolment.

Award and recognition. The provider must approve the final award title, hours and assessment policy. These documents do not establish equivalence to a regulated qualification or

endorsement by the reference organisations. Any accreditation or external-certification claims require a separate, verified approval process.

Reference guidance and curriculum maintenance

These are reference resources used to inform the proposed scope, not course endorsements or a claim of formal certification alignment. Reviewed 17 November 2026. Tool versions, access and pricing must be rechecked before delivery. Human–AI interaction design. Microsoft Research: Guidelines for Human-AI Interaction — https://www.microsoft.com/en-us/research/project/guidelines-for-human-ai-interaction/ AI product risk and evaluation. NIST AI Risk Management Framework and GenAI profile resources — https://www.nist.gov/itl/ai-risk-management-framework Analytics data preparation. Microsoft Learn: Prepare data for analysis with Power BI — https://learn.microsoft.com/en-us/training/paths/prepare-data-power-bi/ Cybersecurity work and skills. NIST NICE Framework Resource Center — https://www.nist.gov/itl/applied-cybersecurity/nice/nice-framework-resource-center AI automation components and evaluation. n8n: Integrate AI — https://docs.n8n.io/build/integrate-ai.md LLM foundations and technical prerequisites. Hugging Face LLM Course — https://huggingface.co/learn/llm-course/chapter1/1 Predictive-model evaluation. scikit-learn: Model selection and evaluation — https://scikit-learn.org/stable/model_selection.html AI application security. OWASP GenAI/LLM application security project — https://owasp.org/projects/top-10-for-large-language-model-applications Human–AI interaction guidance informs expectations, correction and recovery in Design. NIST AI risk guidance informs product evaluation and governance. NICE informs task-based cybersecurity planning. Microsoft data-preparation guidance supports analytics sequencing. n8n and Hugging Face documentation inform automation and engineering capability boundaries. scikit-learn and OWASP inform evaluation and security practice.

Questions? Contact info@axia.africa.