AimNxt
School of Tech Leaders

Gen AI + Agentic AI with Python

Accredited By:

NASSCOM®
Certified Member

MSME
MICRO, SMALL & MEDIUM ENTERPRISES
सूक्ष्म, लघु एवं मध्यम उद्यम
OUR STRENGTH • हमारी शक्ति
सत्यमेव जयते
Ministry of MSME, Govt. of India

DSCI
PROMOTING DATA PROTECTION

Table of Contents

Phase 0 — Setup & Toolchain (1 week)

Outcomes: Productive Python dev environment, CI-ready repo.

- Environment: conda, venv, VS Code, Jupyter.
- Packaging: pip.
- Quality: black, ruff, mypy, pytest, coverage.
- Git essentials; GitHub/Azure DevOps repo + simple CI pipeline.

Lab: Initialize repo, enforce formatting in CI, publish a small wheel.

Phase 1 — Python Foundations (2–3 weeks)

Outcomes: Solid Python fluency for data + services.

- Core: data types, control flow, functions, iterators/generators.
- OOP + data classes + pydantic models.
- Files, JSON/YAML, pathlib, logging.
- Typing with typing & mypy.
- Concurrency: asyncio, aiohttp, task groups.

Labs:

- CLI log parser (sync vs async).
- Typed config loader with pydantic + env overrides.

Phase2—DataStructures,Algorithms&SystemDesign(2–3weeks)

Outcomes: Problem solving + design for scalable AI apps.

- DS/Algo refresh (lists, dicts, heaps, graphs; Big-O).
- API design: REST (FastAPI), GraphQL (Strawberry optional).
- Caching: Redis patterns, idempotency keys.
- Storage: PostgreSQL basics (SQLModel/SQLAlchemy), Cosmos DB intro.
- Design patterns for Python (Strategy, Factory, Adapter).
- Cloud basics on Azure: resource groups, VNet, identity.

Labs:

- FastAPI service with rate-limit + Redis cache.
- Design a URL shortener (read/write paths, cache, DB).

Phase 3 — ML/AI Foundations in Python (3 weeks)

Outcomes: ML literacy and workflow discipline.

- Data wrangling: pandas, polars (optional), EDA with notebooks.
- ML: scikit-learn pipelines, metrics, model validation.
- DL intro: torch or tensorflow (choose one), training loop anatomy.
- Experiment tracking: mlflow basics.
- Packaging + reproducibility for ML code.

Labs:

- Regression + classification pipeline with proper CV & metrics.
- Torch MLP baseline on tabular data with MLflow tracking.

Phase 4 — Generative AI with Python + Azure AI Foundry (4–5 weeks)

Outcomes: Build real GenAI apps, safely and reliably.

- GenAI concepts: transformers, tokenization, prompting patterns.
- Azure AI Foundry (formerly AI Studio):
  - Create hub/project, connect Azure OpenAI deployments.
  - Evaluate prompts in Prompt Flow.
  - Data connections, environments, managed endpoints.
- Python SDKs & APIs:
  - openai (Azure OpenAI), azure-identity, azure-core.
  - Embeddings + vector stores: Azure AI Search (azure-search-documents) or FAISS.
- **RAG end-to-end:**
  - Chunking, embeddings, metadata.
  - Query rewriting, reranking options.
  - Evaluation (faithfulness, groundedness).
- Content Safety & Guardrails: **Azure AI** Content Safety **SDK**; red teaming basics.

Labs:

- Prompt engineering notebook with eval set in Foundry.
- Build a RAG API: FastAPI + Azure OpenAI + Azure AI Search + Blob Storage.
- Add Content Safety checks and structured output via JSON mode.

Phase 5 — Fine-Tuning & Optimization (2 weeks)

Outcomes: Tailor models and control costs/latency.

- Fine-tuning small LLMs via Azure AI Foundry pipelines (where available) or **LoRA/PEFT (Hugging Face)** hosted on Azure.
- Prompt caching, response streaming, batching.
- Latency/throughput tuning; cost observability.

Labs:

- Fine-tune a classification or style adapter; compare vs prompt-only.
- Add streaming + server-sent events to your **FastAPI** endpoint

Phase 6 — Agentic AI in Python (4 weeks)

Outcomes: Design, build, and evaluate autonomous agents.

- Agent concepts: planning, tools, memory, reflection, multi-step control.
- Frameworks (Python):
  - **LangChain** Agents (Tools, Toolkits, AgentExecutors).
  - **LlamaIndex** Agents (QueryEngines, Tools, workflows).
  - AutoGen (multi-agent conversations & tool use).
  - (Optional) Semantic Kernel Python.

- Tooling & Actions:
  - o Web search, DB queries, internal APIs, function-calling.
  - o Structured outputs (pydantic) and schema validation.
- Long-term memory: vector stores (AI Search/FAISS), relational memory, episodic vs semantic.
- Safety for agents: tool authorization, rate limits, sandboxing, human-in-the-loop review.
- Orchestration in Azure AI Foundry Prompt Flow for multi-step pipelines.

Labs: Research assistant agent (web + PDF tools + citations).

- Multi-agent workflow: planner, executor, reviewer; gated by human approval.
- Add memory + capabilities registry; log & replay traces.

Phase7—MLOps/LLMOps&ProductiononAzure(3weeks)-optional

Outcomes: Ship/operate GenAI systems reliably.

- Packaging & deployment:
  - o FastAPI app → Azure Container Apps or AKS.
  - o IaC: Bicep or Terraform (RG, Key Vault, AI Search, Storage).
- Secrets & identity: Managed Identity, Key Vault, role assignments.
- Data: Azure Blob Storage (docs), Event Hubs/Service Bus for async.
- CI/CD: GitHub Actions or Azure Pipelines (build, test, scan, deploy).
- Observability:
  - o App logging & tracing.
  - o Prompt/response telemetry, token & cost meters.
  - o Safety events (blocked/toxicity) dashboards.

Labs:

- One-click infra deploy for the RAG service.
- Canary rollout + auto-rollback using health probes.

Phase 8 — Capstone (2–3 weeks)

Outcomes: A production-grade, demo-ready system.

Pick one and scope it to your org:

1. Enterprise RAG Assistant
   - o Ingest SharePoint/Blob docs; Azure AI Search; citations; Content Safety; audit logs.
2. Agentic Ops Copilot
   - o Multi-agent: planner (LLM), executor (tools: Jira/GitHub/Azure), reviewer (policy).

3. Domain QA with Fine-Tuned Policies

- Mixture of prompting + light fine-tune; eval harness; safety guardrails.

Deliverables: Architecture doc, IaC, CI/CD, runbook, cost model, red-team report.

WeeklyTimeGrid(suggested,24–26 weeks total)

- 0: Setup (1)
- 1–3: Python foundations (3)
- 4–6: DS/Algo + system design (3)
- 7–9: ML foundations (3)
- 10–14: GenAI + Azure AI Foundry (5)
- 15–16: Fine-tuning & perf (2)
- 17–20: Agentic AI (4)
- 21–23: MLOps/LLMOps on Azure (3)
- 24–26: Capstone (2–3)

Tech Stack (Python)

- Core: Python 3.11+, FastAPI, pydantic, httpx/aiohttp, uvicorn.
- ML/GenAI: openai (Azure), langchain, llama-index, faiss-cpu, mlflow, torch (optional).
- Azure SDKs: azure-identity, azure-search-documents, azure-storage-blob, azure-monitor-query, azure-ai-contentsafety (or REST), azure-core.
- Quality/Build: black, ruff, mypy, pytest, coverage, pre-commit.
- Infra: Docker.

Assessments & Milestones

- Checkpoints: Python quiz + coding kata; API design review; ML evaluation report; RAG evaluation (faithfulness/groundedness); agent safety test; deployment DR drill.
- Rubric: Code quality, tests, observability, safety, cost discipline, and user experience.

Certifications

- AI-900

AimNxt
School of Tech Leaders

Ready to turn your dreams into reality?

☎ Speak with our Career Counselor

Connect with your career counselor here:
+91 91 5239 5239 | +91 9059 16 9059

www.aimnxt.org