



AI isn't the future — it's the present. AI Disha is your direction.

• A STRUCTURED CAREER TRANSITION SYSTEM INTO AI

WALK IN CURIOUS. WALK OUT AI-READY HANDS.

Generative AI + Agentic AI + Python — one sequenced path from a Python interpreter to a production agent a hiring manager can inspect.

20	05	60	90	06
MODULES	PHASES	DAYS	GUIDED HOURS	FLAGSHIP PROJECTS

Don't just learn AI. Get hired in AI.

+91 9133 232300

BEFORE WE TALK CURRICULUM

Curiosity is the only prerequisite.

Everything else — the Python, the APIs, the agents — is taught
from zero, in order, with someone checking your work.

WHY THIS PROGRAM EXISTS

Most AI training stops at the demo.

SYMPTOM

Teams apply an LLM to a problem that needed a classifier — or expect a model to “know” private documents without retrieval.

ROOT CAUSE

Hiring titles are noisy. GenAI Engineer, AI Engineer, Prompt Engineer, ML Engineer — the difference saves months of misdirected study, and almost nobody teaches it.

THE FIX

One sequenced curriculum: concept → worked example → your implementation → connection to the next system. Frameworks arrive only after you can do the job in plain Python.

“Conceptual clarity is a professional skill, **not trivia.**”

THE FORMAT

Trained like Day One at a top company.

Corporates run structured induction bootcamps before a fresher ever touches real work — full days, mentor-led, zero guesswork. AI Disha runs the same model, compressed into 60 days. The only difference: every single hour is AI.

FULL-TIME STRUCTURE

A fixed induction calendar, not a self-paced trickle you can quietly abandon.

MENTOR-LED

Practitioners run the room, the way a company's own engineers run onboarding.

100% AI CURRICULUM

No generic soft-skills filler, no unrelated tech stacks — only AI, start to finish.

PROGRAM THESIS

A complete path from Python to production agents.

This curriculum is a single, sequenced program. It begins with the Python you need to ship, moves through LLM APIs and prompt design, then builds the two architectures that define modern applied AI: retrieval-augmented generation and tool-using agents. Later modules add orchestration, multi-agent collaboration, domain context, production operations, and a portfolio hiring managers can inspect.

WHO IT'S FOR

Students, career switchers, and software developers who want applied GenAI and agentic systems skills — not a research-theory degree, not a weekend demo.

WHAT YOU NEED FIRST

Comfort using a computer and installing software. No prior machine learning required — Module 01 brings motivated beginners to a working standard.

HOW YOU LEARN

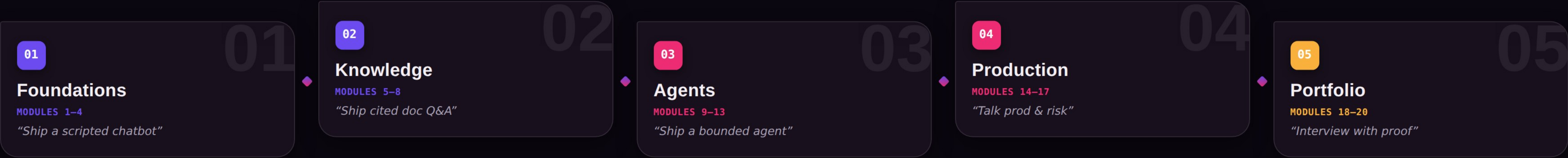
Concept → worked example → you implement a slice → you connect it to the next system. Abstractions stay optional and debuggable.

WHAT SUCCESS LOOKS LIKE

You design, build, evaluate, and explain a RAG application and a bounded tool-using agent — with a portfolio that shows traces, citations, and trade-offs.

PROGRAM STRUCTURE

Five phases. One sequence.



60 days, start to finish — roughly 90 guided hours across five phases, with portfolio work built in along the way, not bolted on at the end.

CURRICULUM AT A GLANCE

Twenty modules. One map.

● FOUNDATIONS & KNOWLEDGE (UNDERSTAND) ● AGENTS & PRODUCTION (ACT) ● PORTFOLIO & CAREER (PROVE)

● 01 Python Foundations for AI	● 02 AI & GenAI Fundamentals	● 03 APIs & LLM Integration	● 04 Prompt Engineering	● 05 LLM Application Development
● 06 Embeddings & Vector Databases	● 07 Retrieval-Augmented Generation	● 08 Advanced RAG Systems	● 09 LangChain (Framework)	● 10 AI Agents Fundamentals
● 11 Agents with Tools	● 12 LangGraph (Advanced Workflows)	● 13 Multi-Agent Systems	● 14 AI in Cybersecurity	● 15 AI in Robotics
● 16 AI Trends & Career Paths	● 17 Production AI Systems	● 18 Projects (Portfolio)	● 19 Assessments	● 20 Internship & Job Prep

BEFORE MODULE 1

Four rules that make the rest of this work.

FOLLOW THE ORDER

Early modules are sequential — skipping Python or APIs stalls RAG. Agent and multi-agent work assume a working single-agent loop first.

TREAT OUTCOMES AS TESTS

If you can't demonstrate an outcome without notes, you're not done. Practice tasks are the minimum — portfolio projects are the proof.

KEEP A LAB NOTEBOOK

Log the model, the parameters, and whether the answer was actually faithful for every experiment. It becomes interview gold and eval data later.

SAFETY FROM DAY ONE

No secrets in git, no unconstrained agents with write access, no offensive security content. Cybersecurity modules stay strictly defensive.

Four rules, one purpose: **make sure competence is provable, not assumed.**

BEFORE MODULE 1 · WHERE YOU END UP

By the end, you'll have shipped — not just watched.



Target roles: GenAI Engineer, AI Engineer, Prompt Engineer, AI Product Developer.

PHASE 01 · FOUNDATIONS

Build the habits every later module depends on.

01 Python Foundations for AI
Control flow, functions, data structures, JSON, virtual environments, pip.

02 AI & Generative AI Fundamentals
AI vs ML vs DL vs GenAI, tokens, context windows, hallucination and bias.

03 APIs & LLM Integration (Python)
REST calls, streaming responses, parameter tuning, a reusable client.

04 Prompt Engineering
Roles, few-shot, chain-of-thought, structured JSON output, templates.

Outcome: ship a scripted LLM chatbot.

Give the model a memory it doesn't hallucinate.

05 LLM Application Development
Chat architecture, message roles, tool-calling concepts, workflows.

06 Embeddings & Vector Databases
Embeddings, cosine/L2 similarity, FAISS and Chroma, metadata filters.

07 Retrieval-Augmented Generation
Ingest → chunk → embed → retrieve → generate, with citations.

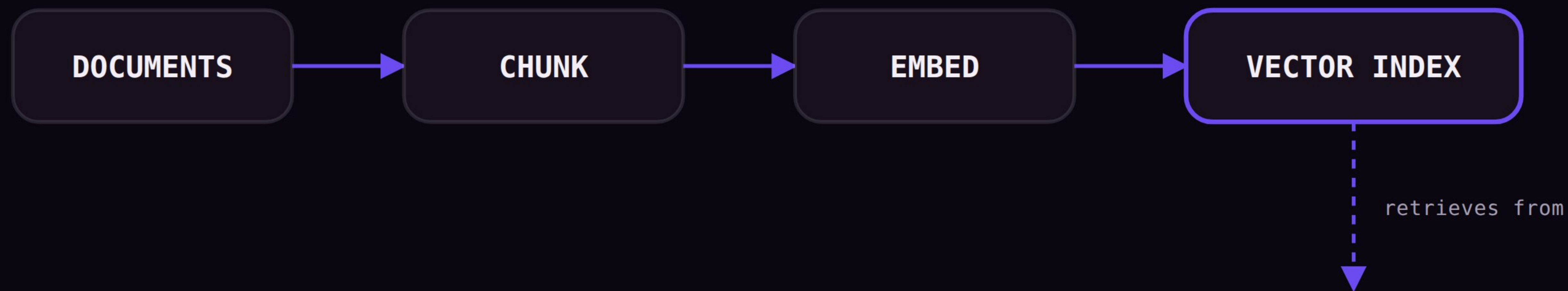
08 Advanced RAG Systems
Hybrid search, query rewriting, re-ranking, faithfulness evaluation.

Outcome: ship cited document Q&A.

SYSTEM SCHEMATIC

Retrieval-Augmented Generation, end to end.

OFFLINE – INDEXING



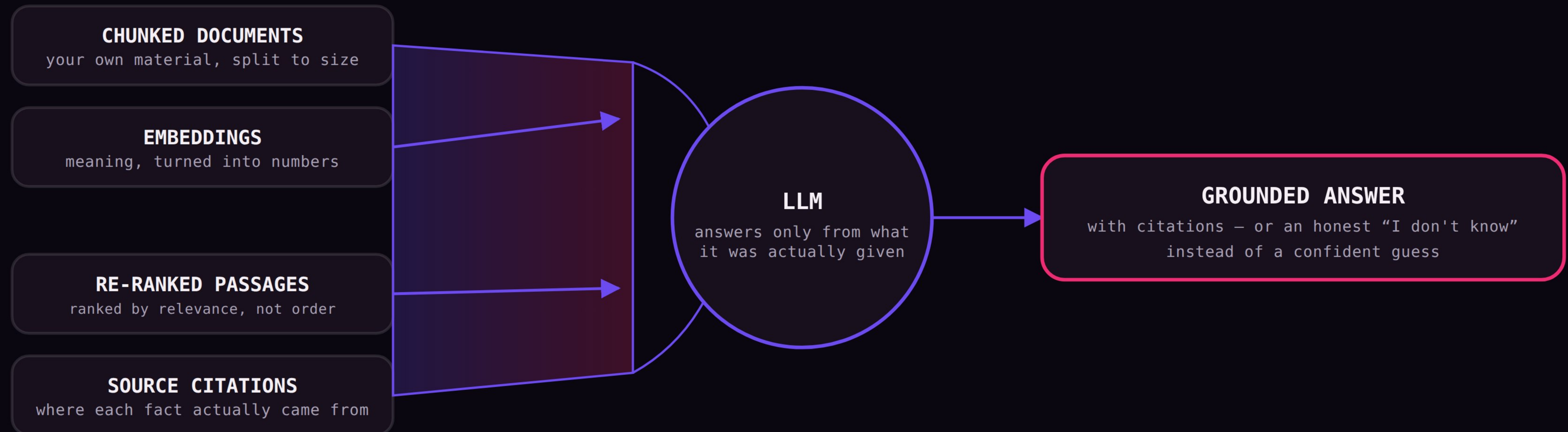
ONLINE – QUERY



Naive RAG hallucinates with confident citations. Hybrid search, re-ranking, and abstention (“I don’t know”) turn a demo into **something you'd trust in review**.

THE IDEA UNDERNEATH THE MODULE

Models don't fail because they're not smart enough.



They fail because they're missing context. **RAG's whole job is feeding them the right context first.**

PHASE 03

From chatbots to real agents.

A chatbot waits to be asked. An agent notices the work, decides what to do, and acts — inside a budget you set.

From answering questions to taking action.

09 LangChain (Framework)
Templates, chains, output parsers, and retrievers as a modular toolkit.

10 AI Agents Fundamentals
The observe → think → act loop, planning, memory, stop conditions.

11 Agents with Tools
Function calling, task execution strategies, confirmation before side effects.

12 LangGraph (Advanced Workflows)
Explicit graphs, shared state, conditional routing, human-in-the-loop.

13 Multi-Agent Systems
Supervisor plus specialists, delegation, bounded critique loops.

Outcome: ship a tooled, bounded agent.

A CLOSER LOOK

Anatomy of one module, not just a title.

LEARNING OBJECTIVE

Give an agent the ability to call real functions safely — typed argument schemas, a hard step limit, and full trace logging on every call.

WHY IT MATTERS

This is where “agent” stops being a buzzword. Tool calls are the only way a model touches a knowledge base, an internal API, or a file — and the only place a bad decision does real damage.

TOPIC · FUNCTION-CALLING SCHEMAS

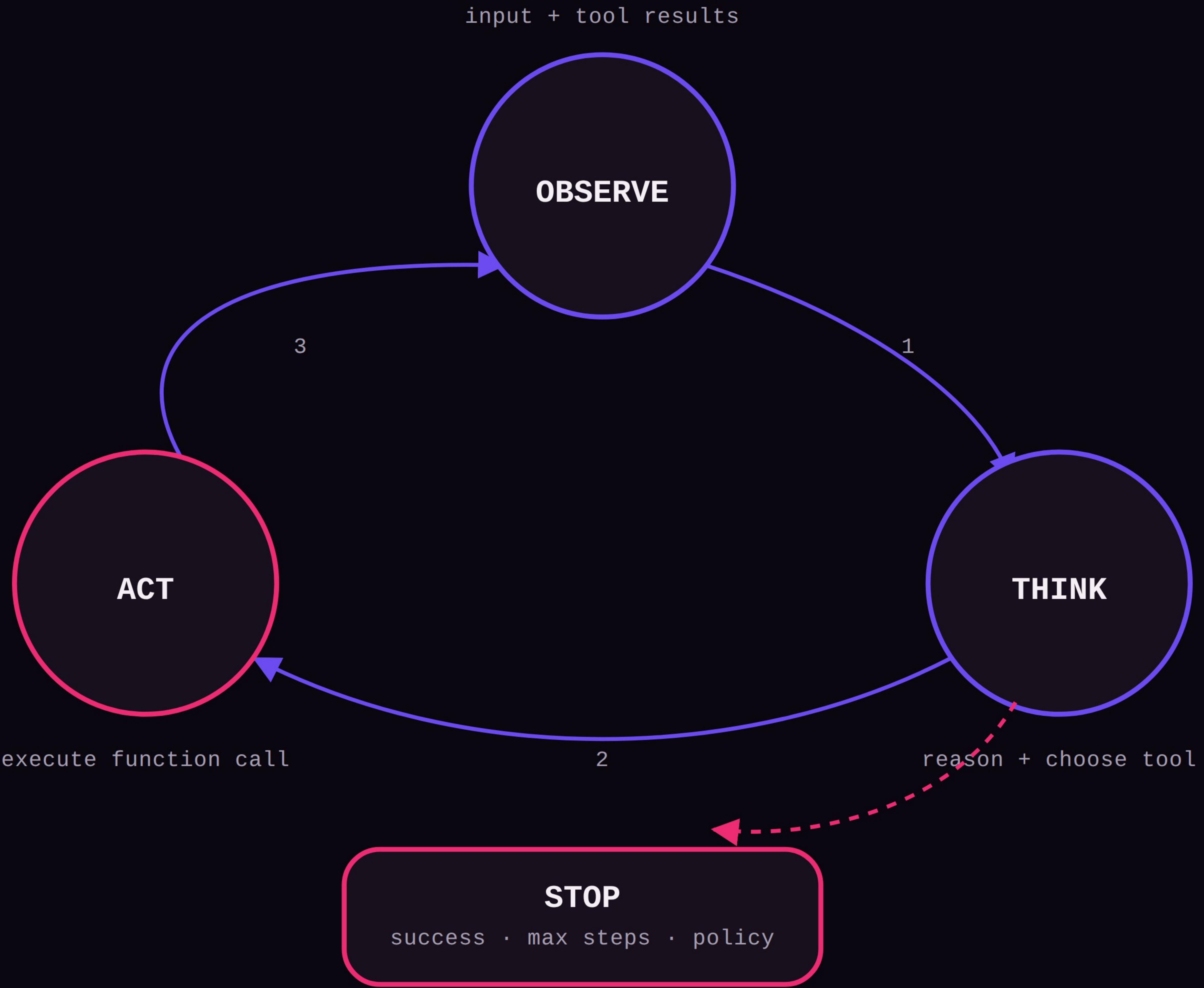
Every tool gets a strict argument schema. The model's job is to fill it correctly, not invent new parameters or skip validation.

TOPIC · CONFIRM BEFORE SIDE EFFECTS

Any action that writes, sends, or spends pauses for human confirmation — one of the guardrails that separates a demo from a deployable feature.

Suggested practice: build an agent that calls two tools with logged traces for a success run, a tool error, and a max-steps stop.

Observe. Think. Act. Stop.



1 THINK PICKS A TOOL · 2 ACT RETURNS A RESULT TO OBSERVE · 3 OBSERVE FEEDS THINK AGAIN — UNTIL THINK ROUTES TO STOP

“Agent” is an overloaded word. The engineering definition is precise: a loop with a goal, a policy, optional memory, tools — and a budget.

WHY THIS PHASE MATTERS

Most AI training stops at a chatbot.

OLD WAY

- Watch someone else build a chatbot on screen
- Copy the finished code into your own project
- Collect a certificate for attending



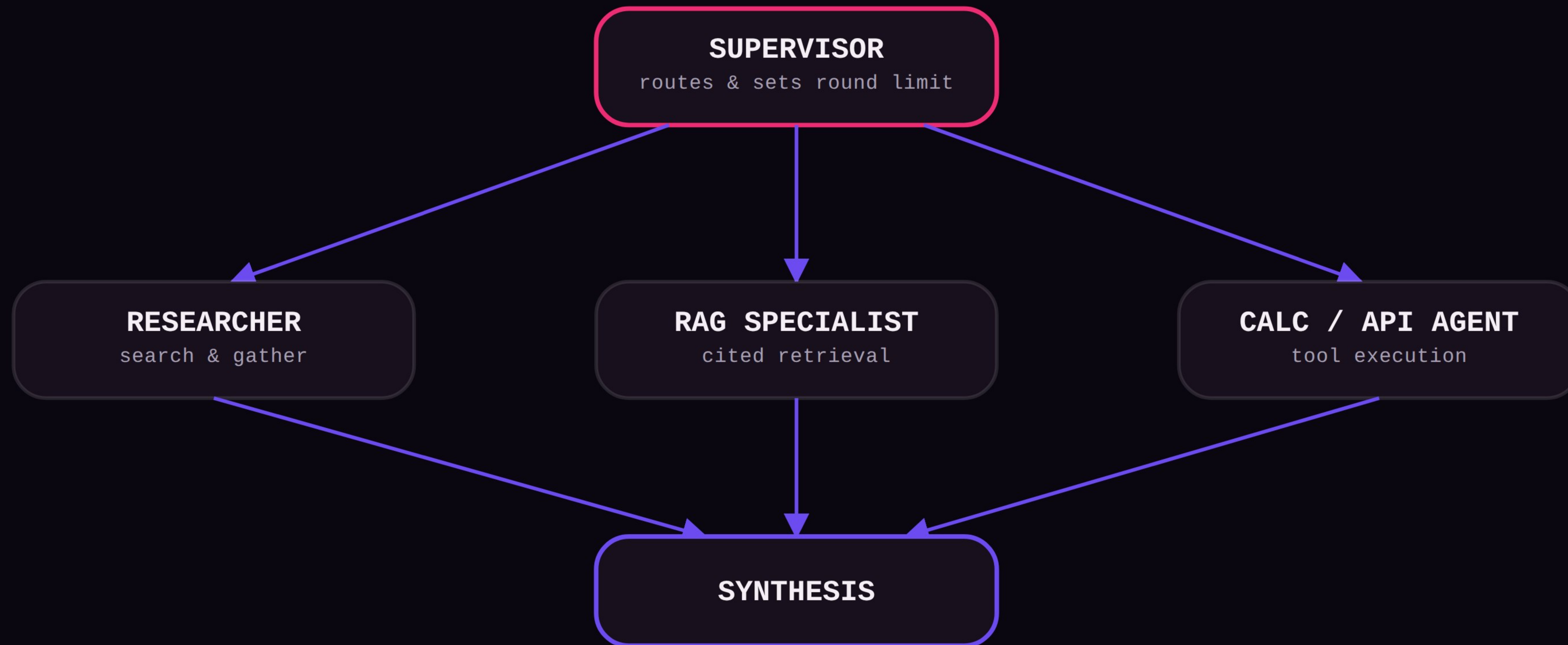
AI DISHA WAY

- Build the agent yourself, live, with a mentor watching
- Get evaluated at four real assessment gates
- Walk away with a portfolio — not a PDF

Message: a certificate proves you attended. **A portfolio proves you can build.**

SYSTEM SCHEMATIC

Scale by splitting roles, not by looping harder.



Multi-agent setups improve quality through specialization — or multiply cost and failure modes. *Start with one well-tooled agent; split only when roles or permissions truly differ.*

PHASE 04 · DOMAIN & PRODUCTION

Where the portfolio meets the real world.

14

AI in Cybersecurity
Defensive triage, alert summarization — no autonomous remediation.

15

AI in Robotics
Sensing → decision → actuation; why “the LLM said so” is never a safety case.

16

AI Trends & Career Paths
GenAI Engineer vs. AI Engineer vs. Prompt Engineer — a realistic market map.

17

Production AI Systems
Architecture, monitoring, guardrails, prompt-injection defense, cost control.

Outcome: talk production and risk like an engineer, not a demo builder.

Portfolio demos die from three causes.

UNBOUNDED COST

No token budget, no caching, no smaller-model routing. **Fix:** per-request budgets, batched embeddings, cheap paths for FAQ hits.

SILENT REGRESSION

Prompt changes ship without evals. **Fix:** tracing across retriever and model, golden question sets, versioned prompts.

UNSAFE PERMISSIONS

Agents with unchecked write access. **Fix:** confirmation gates, least privilege, no unconstrained agents on production controls.

✔ SECRETS OUT OF GIT

✔ SANDBOXED TOOLS

✔ RATE LIMITS

✔ PROMPT-INJECTION ISOLATION

✔ HUMAN REVIEW

✔ ROLLBACK ON REGRESSION

SUGGESTED PRACTICE, SAMPLED

Every module ends with your hands on the keyboard.

03 API Client Build
A reusable client with retries, streaming, and token-usage logging — reused for the rest of the course.

06 Notebook Semantic Index
Index 200 personal notes; query in plain English; return ranked passages with source and date.

08 RAG Faithfulness Check
Re-run a RAG answer set and flag every response the retrieved context doesn't actually support.

12 State-Graph Workflow
Model a 3-step approval process as an explicit graph with a human-confirmation node.

17 Production Sketch
Diagram secrets handling, logging, and rate limits for one real AI feature, end to end.

A lab notebook of five of these, done honestly, reads better in an interview than **any certificate.**

PHASE 05

From practice tasks to proof.

Every phase left you with an outcome. Now they combine into a portfolio — the evidence an interviewer actually asks for.

Six builds. Pick three to five.

01	LLM Chatbot (Python + API)	FOUNDATION
02	Semantic Search System	INTERMEDIATE
03	RAG-based PDF Q&A	INTERMEDIATE
04	AI Agent with Tools	ADVANCED
05	Automation Assistant	ADVANCED
06	Multi-Agent System	CAPSTONE

A RAG system with an evaluation table and honest failure analysis **outperforms six unmaintained demos.**

FLAGSHIP PROJECTS

Three builds. Real deliverables, not toy demos.

LLM CHATBOT

A streaming chat with system prompt, conversation history, temperature control, and token logging — clean API integration and message-role discipline.

Deliverables: Repo with README, .env.example, pinned requirements, and a 2-minute demo.

SEMANTIC SEARCH SYSTEM

Index a corpus of notes or articles with embeddings. Query by natural language, return ranked passages with scores and metadata — source, date.

Deliverables: Reproducible index-build script, query CLI, and a note on the similarity metric chosen.

RAG-BASED PDF Q&A

Ingest PDFs, chunk, retrieve, and answer with citations. Include an abstain path when context is weak.

Deliverables: Sample PDFs, a 15-question eval set, and a table of faithfulness notes.

FLAGSHIP PROJECTS, CONTINUED

Three more builds, each one harder to fake.

AI AGENT WITH TOOLS

An agent that calls at least two tools via function calling, with a maximum step count and logged traces.

Deliverables: Trace logs for three tasks — success, tool error, max-steps stop.

AUTOMATION ASSISTANT

Turns a user request into a bounded sequence of actions — draft email, create a task, summarize a file — with human confirmation before side effects.

Deliverables: Graph/state diagram, confirmation UX, and a safety note.

MULTI-AGENT SYSTEM

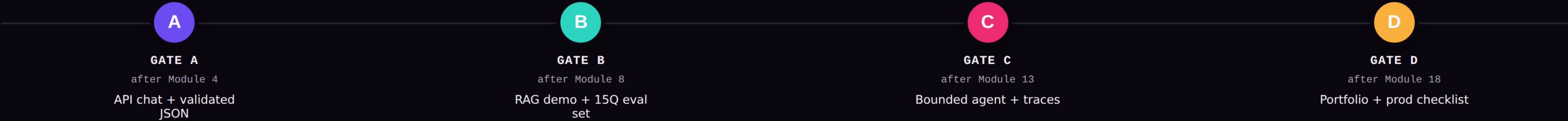
Supervisor plus two specialists, round limits, shared state, and a final synthesis — with tracing and a cost cap.

Deliverables: Architecture diagram, demo recording, and a one-page failure-mode analysis.

Portfolio outcome: three to five real projects — chat, search, RAG, tools, and multi-agent — each with a README that explains design choices, limits, and next steps.

ASSESSMENT GATES

Verify skill, not attendance.



Assessments mix practical assignments, project reviews, and time-bounded exercises that mirror how engineers are actually screened.

HOW YOU'RE ACTUALLY GRADED

Three ways to prove it, not one.

PRACTICAL ASSIGNMENTS

Short, module-scoped tasks scored on correctness, code clarity, error handling, and whether secrets and dependencies are handled professionally. Late work is fine; copied notebooks without understanding are not.

PROJECT EVALUATIONS

Each portfolio project is reviewed against a rubric — problem framing, architecture, implementation, evaluation, safety, communication. Reviewers run the happy path and expect one documented failure case.

SKILL-BASED TESTING

Live or take-home drills: a robust JSON-parsing client, a prompt that always returns a schema, diagnosing a RAG miss, adding a tool with an argument schema, sketching a production diagram.

You can prove competence with artifacts and live explanation — **the same evidence used in internships and junior GenAI interviews.**

THE FINAL STRETCH

From learner to hired.

The gates, the projects, the resume rewrite — all of it points at one moment: the interview where you have real answers.

FINAL OUTCOME

Map yourself to one primary role.

GenAI Engineer

Owns LLM-powered features end to end: prompts, retrieval, evaluation, integration with product backends.

AI Engineer

Broader applied AI — classical ML plus GenAI, data pipelines, APIs, production sense.

Prompt Engineer

Designs, tests, and maintains prompts and evaluation sets, often with structured-output constraints.

AI Product Developer

Shapes user journeys and MVP scope for AI features; implements or closely partners with engineering.

Completing this curriculum does not confer a job by itself. It gives you the stack, the vocabulary, and the artifacts those jobs use every week.

FINAL OUTCOME — ROLE TO MODULE MAPPING

Each role traces back to specific modules.

GenAI Engineer

Owns LLM-powered features end to end. Strong overlap with Modules 3-9, 17, and the RAG/agent projects.

AI Engineer

Broader applied AI — classical ML plus GenAI. Pair this curriculum's GenAI half with extra ML if a job listing asks for model training.

Prompt Engineer

Designs and maintains prompts and eval sets. Most sustainable paired with light engineering — Modules 4, 5, and 17.

AI Product Developer

Shapes user journeys and MVP scope. Modules 5, 10-13, 16, and 20, plus a polished demo, are the core.

✓ BUILD WITH PYTHON + LLM APIS

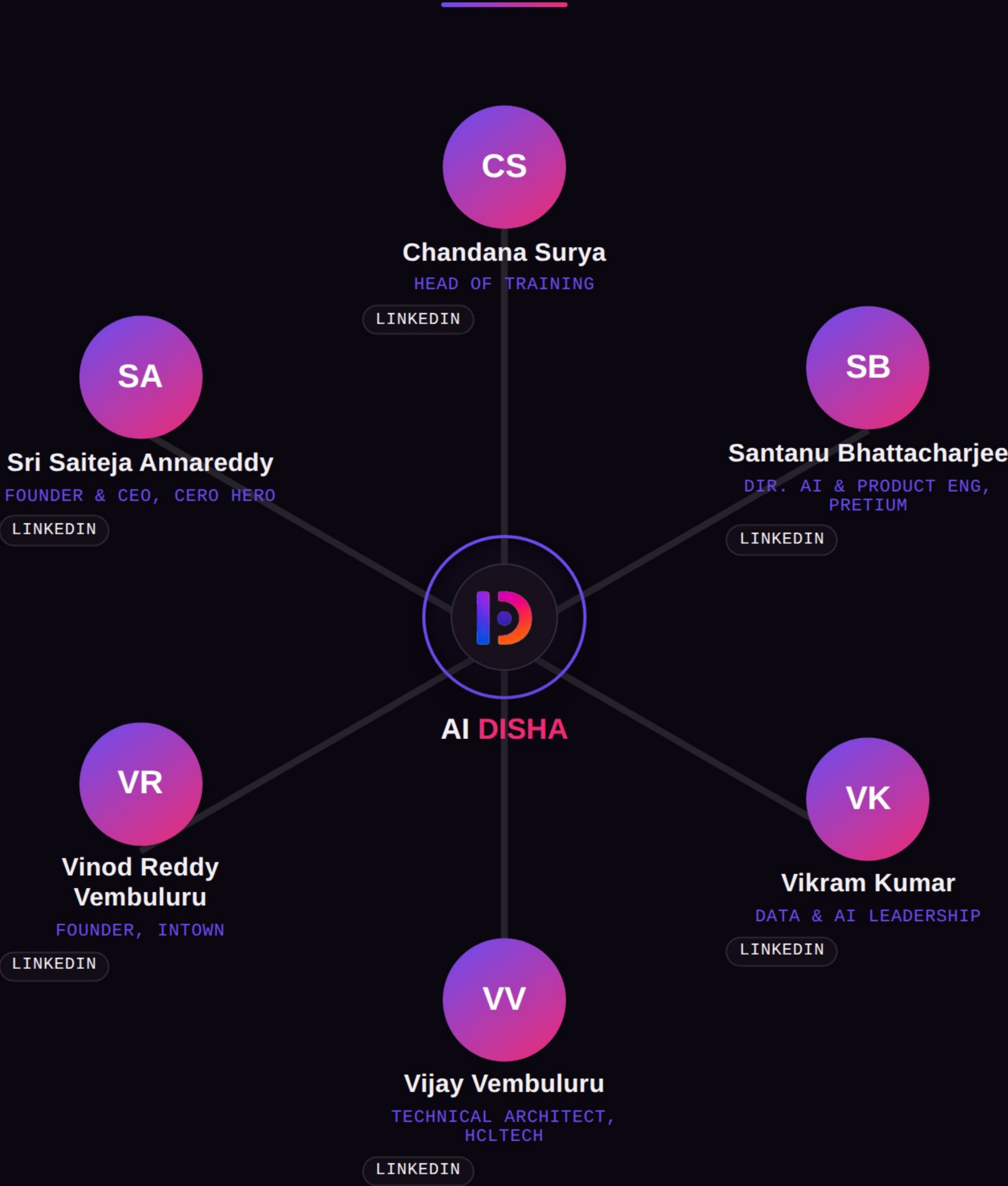
✓ DESIGN RAG, AGENTS & WORKFLOWS

✓ REASON ABOUT COST & GUARDRAILS

✓ PRESENT PROOF, NOT JUST SCREENSHOTS

TRAINERS & SPEAKERS

One hub. Six people who build the real thing.



WHERE OUR TRAINERS ARE BASED

One classroom. Trainers from four countries.

-  India
-  Australia
-  Finland
-  United States

Every trainer is a practitioner currently building AI, cloud, and product systems — based across four countries, not just instructors reading slides.

OUR TRAINERS COME FROM

Faculty who've shipped inside these companies.

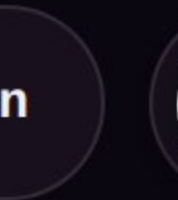
 Google

 Microsoft Azure

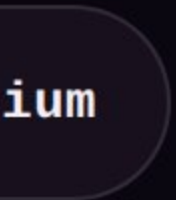
 Infosys

 Samsung

 Logiq Gen

 HCLTech

 Pretium

 GSPANN Technologies

Trainers average **16+ years** of industry experience — the classroom runs on production war stories, not just theory.

BEYOND THE CORE FACULTY

Five guest lectures from people already shipping AI.

Your regular trainers carry the curriculum end to end. Five sessions bring in outside voices — leaders and builders from different corners of the AI industry — so you hear more than one perspective on the same field.

INDUSTRY CAREER PATHS

What GenAI Engineer, AI Product Developer, and Prompt Engineer roles actually look like day to day — and how people got hired into them.

BUILDING AI PRODUCTS

How a real AI feature goes from prototype to production — the decisions that don't show up in a tutorial.

AI IN THE ENTERPRISE

How large organizations adopt AI safely — governance, cost control, and change management at scale.

STARTUPS & FOUNDERS

Building an AI-native company from zero — what changes when you own the product, not just a feature.

LIVE Q&A, OPEN FLOOR

Unfiltered questions about careers, interviews, and the realities of working in AI — answered directly, in the room.

Five extra rooms to sit in **that most training programs never open.**

COMPANIES & ECOSYSTEM

AI Disha is a group company.

AI Disha is a group company of AI Nestham, INtown, Logiq Gen, and VKnowTech AI — a working AI ecosystem, not a standalone course.



Logiq Gen

Product engineering venture behind AI Disha's training roadmap — building JBLUE, a white-label AI-powered LMS.

LOGIQGEN.COM →



VKnowTech AI

Workday consulting & AI training arm — delivery bench and enterprise training experience feeding back into the curriculum.

VKNOWTECH.AI →



INtown

Hyperlocal commerce venture — merchant loyalty and consumer savings, founded by AI Disha guest faculty.

INTOWNLOCAL.COM →



AI Nestham

AI-powered learning platform bringing gamified AI literacy to schools — part of the AI Disha ecosystem.

AINESTHAM.COM →

IN ASSOCIATION WITH **T-HUB**, TELANGANA'S STARTUP INNOVATION HUB.

INTERNSHIP DESIGN

A guaranteed internship — for every graduate.

Every student who clears the program is guaranteed a placement in the internship phase. What you work on, and with whom, is decided by performance — not chance.

GUARANTEED

Every graduate gets an internship placement — no exceptions, no waitlist.

PROJECTS

Industry-aligned, matched to real company needs.

EXPOSURE

Direct exposure to real engineering workflows.

MENTORSHIP

Guided by working professionals, not teaching assistants.

The internship is guaranteed — the project and team you land depend on your Gate D performance.

GETTING HIRE-READY

Your resume and portfolio, engineered like a project.

RESUME BUILDING

Lead with a tight professional summary — role target, stack, one measurable result. Bullets follow “built X using Y that achieved Z” — for example, retrieved from N documents with citation rate R on an eval set. Tools are listed once; impact is listed often.

PORTFOLIO CREATION

Three pinned repositories with architecture diagrams, eval snippets, and a 90-second demo. A short profile README stating the role you want, linking your best two projects. Pin the RAG and agent projects; keep the chatbot only if it shows streaming, tests, or unusual polish.

GETTING HIRE-READY, CONTINUED

Practiced answers. Targeted applications.

INTERVIEW PREPARATION

Practice explaining tokens and cost, RAG failure modes, tool-calling safety, and the differences between chatbots, RAG, and agents — plus a production sketch: secrets, logging, rate limits. Coding screens stay in Python. Behavioral stories use your own projects as the plot.

INTERNSHIP GUIDANCE

Target teams that already ship LLM features — search, support, internal tools — rather than “AI” as a slogan. Apply with a tailored note referencing their product and one relevant repo. Keep a tracker of applications, referrals, and follow-ups.

Ready to pursue internships and junior roles in AI / Generative AI with a coherent story, evidence, and interview fluency.

BILL OF MATERIALS

The stack, named.

CORE CURRICULUM STACK

 Python 3

 pip + venv

 OpenAI-Compatible LLM API

 FAISS

 Chroma

 LangChain

 LangGraph

 JSON

 Git

ALSO IN THE TOOLKIT

 Hugging Face

 Docker

 FastAPI

 Jupyter

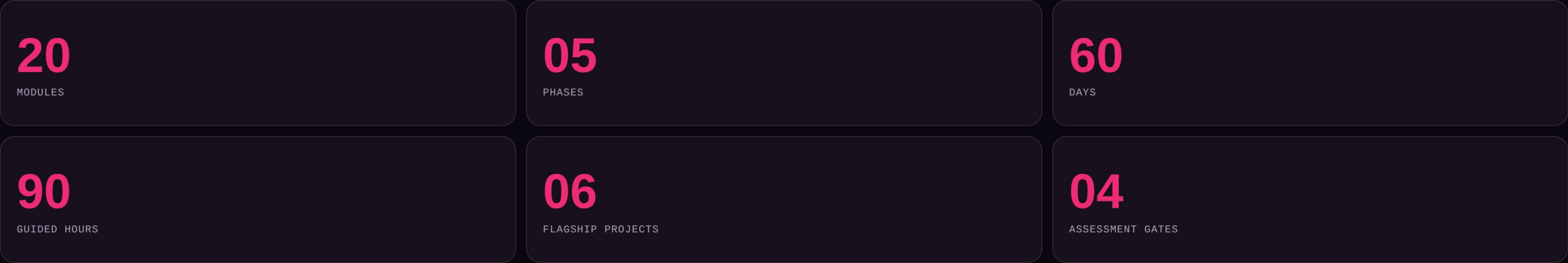
 Streamlit

 Anthropic

You may substitute equivalent libraries — the ideas are the product: messages, embeddings, retrieval, tools, graphs, evals.

THE PROGRAM, BY THE NUMBERS

Twenty modules. One outcome.



- ✓ NO PRIOR AI BACKGROUND NEEDED
- ✓ LIVE, MENTOR-LED – NOT PRE-RECORDED
- ✓ PORTFOLIO, NOT JUST A CERTIFICATE
- ✓ GUARANTEED INTERNSHIP

COHORTS ARE SMALL BY DESIGN

Your seat is waiting.

Sixty days from now, you could be explaining your own RAG evaluation table to an interviewer. Let's talk about starting.

GET IN TOUCH

Seats are limited to keep every cohort mentor-led.

Reserve a free demo slot, or talk to us about bringing this program to your campus or team — we'll walk you through the syllabus, the projects, and how the assessment gates work.

PHONE

+91 9133 232300

WHATSAPP

+91 9133 232300

EMAIL

info@aidishlearning.com

WEBSITE

www.aidishlearning.com



CLOSING

STUDY IN ORDER. MEASURE YOUR ANSWERS. BOUND YOUR AGENTS. SHOW YOUR WORK.

That is the professional standard this curriculum is written to produce.

Don't just learn AI. Get hired in AI.