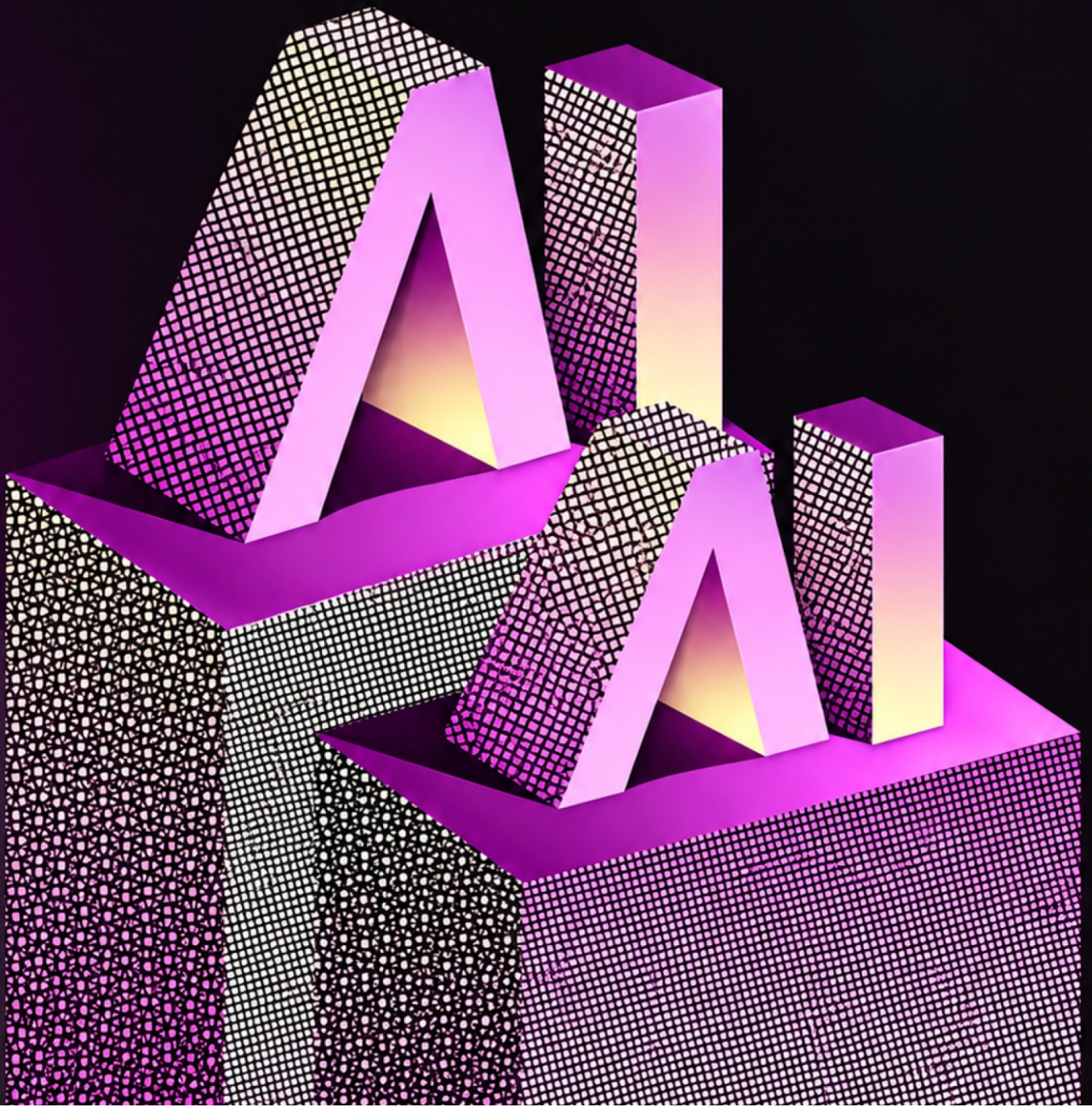


Build, Deploy & Scale Production-Ready AI Systems

AI Engineering & Agentic AI



10-WEEK TECHNICAL PROGRAM

AI Engineering & Agentic AI

Build, Deploy & Scale Production-Ready AI Systems

LLM Engineering / RAG / AI Agents / Multi-Agent Systems / MCP / AI Evaluation / AI Security / Production / AI Architecture

PDF CURRICULUM

EDUCITY

AI ENGINEERING & AGENTIC AI

Build, Deploy & Scale Production-Ready AI Systems

10-Week Technical Program

LLM Engineering → RAG → AI Agents → Multi-Agent Systems → MCP → AI Evaluation → AI Security → Production → AI Architecture

PROGRAM OVERVIEW

AI ENGINEERING & AGENTIC AI / 10-WEEK TECHNICAL PROGRAM

PROGRAM OVERVIEW

Don't Just Use AI. Build It.

AI is moving beyond chatbots.

Modern AI applications can retrieve information, reason through complex tasks, use external tools, interact with software, execute workflows and operate with varying levels of autonomy.

AI Engineering is the discipline of turning these capabilities into reliable software systems.

The Educity **AI Engineering & Agentic AI** program is a hands-on 10-week technical program designed to take learners from AI application fundamentals to building, evaluating, securing and deploying production-ready AI systems.

Throughout the program, learners work with:

- Large Language Models (LLMs)
- LLM APIs
- Prompt & Context Engineering
- Retrieval-Augmented Generation (RAG)
- Vector Databases
- AI Agents
- Multi-Agent Systems
- Tool Calling
- Model Context Protocol (MCP)
- AI Evaluation
- AI Security & Guardrails
- Observability
- Cloud Deployment
- AI System Architecture

The objective is simple:

Learn the engineering behind modern AI — and build systems that actually work.

WHO IS THIS PROGRAM FOR?

This program is designed for technical professionals who want to build production AI applications.

Software Developers

Build AI-powered features and applications using modern LLM technologies.

Backend Developers

Extend backend engineering skills into AI APIs, agents, tools and production systems.

Full-Stack Developers

Learn how to integrate AI into real-world software products.

Data Scientists & ML Professionals

Expand from models and experimentation into modern LLM application engineering.

Technical Founders

Learn how to prototype, architect and deploy AI-powered products.

Developers Transitioning Into AI

Build practical AI engineering skills and a portfolio of production-oriented projects.

PREREQUISITES

Recommended prior knowledge:

- Basic programming experience
- Familiarity with software development concepts
- Basic understanding of APIs
- Basic Git/GitHub knowledge
- Familiarity with JavaScript or TypeScript is recommended

The program uses **Node.js + TypeScript** as its primary development stack.

Learners will also gain exposure to Python interoperability for AI/ML workflows where relevant.

YOUR 10-WEEK AI ENGINEERING JOURNEY

AI ENGINEERING & AGENTIC AI / 10-WEEK TECHNICAL PROGRAM

YOUR 10-WEEK AI ENGINEERING JOURNEY

WEEK 1

AI Engineering Foundations

WEEK 2

LLM Engineering

WEEK 3

RAG & Knowledge Systems

WEEK 4

AI Agents

WEEK 5

Multi-Agent Systems

WEEK 6

MCP, Tools & AI Integrations

WEEK 7

AI Evaluation, Security & Observability

WEEK 8

Production AI & Deployment

WEEK 9

AI System Design & Architecture

WEEK 10

Production AI Capstone

WEEK 1

AI ENGINEERING & AGENTIC AI / 10-WEEK TECHNICAL PROGRAM

WEEK 1

AI ENGINEERING FOUNDATIONS

From Software Developer to AI Builder

Before building sophisticated AI agents, you need to understand how AI applications fit into modern software systems.

This module establishes the software engineering foundation required to build reliable AI-powered applications using **Node.js, TypeScript and NestJS**.

Core Engineering

- Node.js fundamentals
- TypeScript fundamentals
- Project structure and architecture
- Package management
- Environment configuration
- Configuration management
- Git & GitHub
- Branching and version control
- Debugging
- Logging
- Error handling
- Async/await
- Promises
- Concurrency fundamentals
- Middleware
- Dependency injection

Backend Development

- REST API fundamentals
- HTTP methods and status codes
- REST API design
- NestJS architecture
- Controllers
- Services
- Modules
- Dependency injection
- DTOs

- Request validation
- API documentation
- Authentication fundamentals

Data & Validation

- JSON
- Schema design
- Zod
- Type safety
- Input validation
- Output validation
- Environment variables
- Secrets management

Working With APIs

- Consuming external APIs
- API authentication
- API keys
- Request/response handling
- Rate limits
- Retries
- Timeouts
- Error recovery

AI Application Foundations

- What is an AI application?
- Traditional software vs AI-powered software
- Model layer
- Application layer
- Data layer
- Tool layer
- Orchestration layer
- User interface layer
- AI application lifecycle

HANDS-ON PROJECT

Build Your First AI-Enabled API

Build a backend application that connects to an external AI API and exposes AI functionality through a production-style REST API.

Outcome

By the end of Week 1, learners understand how AI capabilities fit into modern application architecture and can build a functional AI-enabled backend.

WEEK 2

LLM ENGINEERING

Stop Treating LLMs Like Chatbots

Large Language Models are powerful components—but building reliable applications around them requires more than knowing how to write prompts.

This module covers the fundamentals of LLM application engineering.

LLM Fundamentals

- What are Large Language Models?
- Transformer architecture — developer level
- Tokens
- Tokenisation
- Context windows
- Attention mechanisms
- Parameters
- Inference
- Temperature
- Top-p
- Model capabilities
- Model limitations
- Hallucinations
- Model knowledge vs external knowledge

Model Selection

- OpenAI models
- Anthropic models
- Google Gemini
- Open-source models
- Model capability comparison
- Quality vs latency
- Cost vs performance
- Model routing
- Selecting the right model for a task

Prompt Engineering

- System prompts

- User prompts
- Instruction hierarchy
- Zero-shot prompting
- Few-shot prompting
- Prompt templates
- Role prompting
- Prompt versioning
- Prompt injection awareness

Context Engineering

- Context windows
- Context selection
- Context prioritisation
- Context compression
- Dynamic context
- Conversation history
- Memory vs context
- Designing effective AI context

Structured AI Outputs

- Structured outputs
- JSON responses
- Schema enforcement
- Function calling
- Tool calling
- Typed AI responses
- Validation
- Error handling

LLM APIs

- OpenAI API
- Anthropic API
- Gemini API
- Streaming responses
- API error handling
- Retries
- Rate limits
- Token tracking

AI Application Frameworks

- Introduction to LangChain.js
- Models
- Prompts
- Tools
- Chains
- Structured outputs
- Runnable workflows

AI Economics

- Token economics
- Input vs output costs
- Latency
- Throughput
- Caching
- Model routing
- Cost optimisation

HANDS-ON PROJECT

Build a Production-Style LLM Application

Build an application using LLM APIs, structured outputs and tool/function calling.

Outcome

Learners understand how to integrate and engineer around modern LLMs rather than simply interact with them through a chat interface.

WEEK 3

RAG & KNOWLEDGE SYSTEMS

Give AI Access to Knowledge It Didn't Learn During Training

Real businesses need AI to work with their own documents, databases, knowledge bases and continuously changing information.

Retrieval-Augmented Generation enables AI applications to retrieve relevant information before generating an answer.

RAG Fundamentals

- What is RAG?
- Why RAG?
- RAG vs fine-tuning
- RAG architecture
- Retrieval pipeline
- Generation pipeline

Document Processing

- PDF processing
- Document parsing
- Text extraction
- HTML/web content
- Document cleaning
- Metadata extraction
- Document versioning

Chunking

- Fixed-size chunking
- Recursive chunking
- Semantic chunking
- Document-aware chunking
- Chunk overlap
- Chunk size optimisation
- Parent-child retrieval

Embeddings

- What are embeddings?
- Embedding models

- Semantic similarity
- Vector representations
- Embedding generation
- Embedding storage

Vector Databases

- Vector database fundamentals
- PostgreSQL + pgvector
- Vector indexing
- Similarity search
- Metadata filtering
- Hybrid retrieval

Retrieval

- Semantic search
- Keyword search
- Hybrid search
- Query expansion
- Query transformation
- Reranking
- Top-k retrieval
- Retrieval filtering

Advanced RAG

- Multi-query RAG
- Contextual retrieval
- Parent-document retrieval
- Query routing
- Multi-source retrieval
- Agentic RAG
- Graph-based retrieval concepts

Grounding & Hallucination

- Grounded generation
- Source attribution
- Citations
- Confidence
- Retrieval failures
- Hallucination reduction

- "I don't know" behaviour

HANDS-ON PROJECT

Build an Enterprise Knowledge Assistant

Documents → ingestion → chunking → embeddings → retrieval → reranking → grounded response → citations

Outcome

Learners can build AI systems that work with private and external knowledge rather than relying solely on a model's training data.

WEEK 4

AI AGENTS

From AI That Answers → AI That Acts

AI agents extend LLM applications by giving them goals, tools, memory, state and the ability to execute multi-step workflows.

Agent Fundamentals

- What is an AI agent?
- LLM vs workflow vs agent
- Agent architecture
- Agent loop
- Goal setting
- Planning
- Reasoning
- Action
- Observation
- Completion

Agent Memory

- Short-term memory
- Long-term memory
- Conversation memory
- Working memory
- State
- Memory persistence
- Context management

Tools & Actions

- Function calling
- Tool definitions
- Tool schemas
- Tool selection
- Tool execution
- Tool validation
- Tool errors
- Tool retries

Agent Workflows

- Sequential workflows
- Conditional workflows
- Routing
- Parallel execution
- Loops
- Branching
- Human approval
- Interruptions

Agent Reliability

- Guardrails
- Retry strategies
- Timeouts
- Maximum iterations
- Failure handling
- Fallbacks
- Termination conditions
- Human-in-the-loop

Agent Framework

LangGraph.js

- Nodes
- Edges
- State
- Graph workflows
- Checkpoints
- Persistence
- Interrupts

HANDS-ON PROJECT

Build an Autonomous AI Agent

The agent receives a goal → plans the task → selects tools → executes actions → evaluates results → completes the task.

Outcome

Learners understand how to design AI systems that can perform multi-step tasks rather than simply generate responses.

WEEK 5

AI ENGINEERING & AGENTIC AI / 10-WEEK TECHNICAL PROGRAM

WEEK 5

MULTI-AGENT SYSTEMS

Build a Team of AI Workers

Complex tasks can sometimes be decomposed into specialised roles handled by multiple agents.

Multi-Agent Architecture

- Single-agent vs multi-agent systems
- When to use multiple agents
- Agent specialisation
- Agent roles
- Agent boundaries
- Agent coordination

Architecture Patterns

- Supervisor pattern
- Worker pattern
- Hierarchical agents
- Sequential agents
- Parallel agents
- Peer-to-peer agents
- Router architecture
- Review and critique patterns

Agent Communication

- Agent-to-agent communication
- Shared state
- Shared memory
- Messages
- Task delegation
- Result passing
- Context management

Reliability

- Agent failure
- Conflicting outputs
- Retry strategies
- Recovery

- Human approval
- Escalation
- Agent termination

Advanced Workflows

- Long-running agents
- Stateful workflows
- Parallel execution
- Background tasks
- Scheduled agents
- Event-driven agents

HANDS-ON PROJECT

Build an AI Team

Example:

Researcher → Analyst → Writer → Reviewer

Each agent has a defined responsibility and contributes to the final result.

Outcome

Learners can design multi-agent workflows and understand when multi-agent architecture is appropriate versus using a simpler single-agent or deterministic workflow.

WEEK 6

MCP, TOOLS & AI INTEGRATIONS

Make AI Useful Outside the Chat Window

An AI system becomes significantly more useful when it can interact with external data, software and business systems.

Model Context Protocol

- What is MCP?
- MCP architecture
- MCP clients
- MCP servers
- Tools
- Resources
- Prompts
- Tool discovery
- MCP lifecycle
- MCP security

Building Tools

- Custom tools
- Tool schemas
- Tool validation
- Tool permissions
- Tool execution
- Tool error handling
- Tool responses

External Integrations

Connect AI systems to:

- REST APIs
- Databases
- CRM systems
- Email
- Calendars
- File systems
- SaaS platforms

- Search systems
- Web services
- External applications

Data Access

- SQL tools
- Database querying
- File retrieval
- Knowledge retrieval
- API-based data access

Security

- Authentication
- Authorisation
- Permission boundaries
- Tool-level access control
- Secrets
- User identity
- Data access policies

HANDS-ON PROJECT

Build an AI Worker

Build an AI system that can:

Understand → Decide → Use Tools → Execute Actions → Return Results

Outcome

Learners understand how to connect AI agents with real-world software and business systems.

WEEK 7

AI EVALUATION, SECURITY & OBSERVABILITY

A Demo Is Not a Production System

AI systems are probabilistic. A successful demonstration does not prove that an AI application is reliable.

This module teaches learners how to evaluate, secure and observe AI systems.

AI EVALUATION

Evaluation Fundamentals

- Why AI evaluation is different
- Deterministic vs non-deterministic systems
- Evaluation datasets
- Golden datasets
- Test cases
- Expected behaviour
- Regression testing

LLM Evaluation

- Answer quality
- Relevance
- Correctness
- Faithfulness
- Groundedness
- Safety
- Toxicity
- Bias testing

RAG Evaluation

- Retrieval quality
- Precision
- Recall
- Context relevance
- Context completeness
- Groundedness
- Answer faithfulness

Agent Evaluation

- Task completion
- Tool selection
- Tool execution
- Planning quality
- Failure recovery
- Agent trajectory
- Cost per task

LLM-as-a-Judge

- Evaluation models
- Evaluation rubrics
- Automated scoring
- Human evaluation
- Evaluation consistency

AI SECURITY

Common AI Threats

- Prompt injection
- Indirect prompt injection
- Jailbreaking
- Data leakage
- Sensitive information exposure
- Malicious tool calls
- Excessive agent permissions

Security Controls

- Input validation
- Output validation
- Guardrails
- Permission boundaries
- PII detection
- Data filtering
- Tool restrictions
- Human approval
- Secure secrets management

AI OBSERVABILITY

- Logging
- Tracing
- Token usage
- Latency
- Error rates
- Tool usage
- Agent trajectories
- Cost monitoring
- Performance monitoring
- Debugging AI systems

HANDS-ON PROJECT

Build an AI Evaluation & Observability Layer

Create tests, evaluation metrics, traces and monitoring for an AI application.

Outcome

Learners can measure AI performance and diagnose failures rather than relying solely on subjective testing.

WEEK 8

PRODUCTION AI & DEPLOYMENT

Take Your AI Out of the Playground

A prototype running on a laptop is not a production system.

This module focuses on deployment, reliability, security, scalability and operational requirements.

Containerisation

- Docker fundamentals
- Dockerfiles
- Docker Compose
- Environment configuration
- Container networking

Cloud

- Cloud fundamentals
- AWS
- Azure
- Compute
- Storage
- Databases
- Secrets
- AI services

CI/CD

- Continuous integration
- Continuous deployment
- GitHub Actions
- Automated testing
- Deployment pipelines
- Environment management

Production APIs

- Authentication
- Authorisation
- Rate limiting
- API keys

- Request validation
- Error handling
- Retries
- Timeouts

Scaling

- Horizontal scaling
- Caching
- Queues
- Background jobs
- Async processing
- Load balancing
- Database optimisation

AI Cost Optimisation

- Token optimisation
- Prompt optimisation
- Caching
- Model routing
- Smaller models
- Batch processing
- Inference optimisation
- Cost-per-task analysis

Production Monitoring

- Logs
- Metrics
- Alerts
- Traces
- AI observability
- Incident management

HANDS-ON PROJECT

Deploy an AI Application to the Cloud

Take an AI application from local development to a publicly accessible production environment.

Outcome

Learners understand the practical requirements of taking AI applications from prototype to production.

WEEK 9

AI SYSTEM DESIGN & ARCHITECTURE

Think Like an AI Architect

Building an AI feature is different from designing an AI system that can serve real users at scale.

This module brings together the engineering principles covered throughout the program.

AI Architecture

- AI system components
- Model layer
- Application layer
- Data layer
- Retrieval layer
- Agent layer
- Tool layer
- Observability layer
- Security layer

LLM Architecture

- Model gateways
- Model routing
- Multiple providers
- Fallback models
- Cost-aware architecture
- Latency-aware architecture

RAG Architecture

- Ingestion pipelines
- Retrieval architecture
- Vector databases
- Hybrid search
- Reranking
- Knowledge updates

Agent Architecture

- Agent orchestration
- State management

- Memory
- Tools
- Multi-agent systems
- Human-in-the-loop

Data Architecture

- SQL
- PostgreSQL
- Vector databases
- Data pipelines
- ETL/ELT concepts
- Data quality
- Data governance

Distributed Systems

- Queues
- Event-driven architecture
- Background workers
- Caching
- Asynchronous processing
- Scalability
- Fault tolerance

Enterprise AI

- Security
- Governance
- Permissions
- Compliance considerations
- Data isolation
- Multi-tenancy

Architecture Economics

- Infrastructure cost
- Model cost
- Token economics
- Build vs buy
- Open-source vs proprietary
- Performance vs cost

ARCHITECTURE CHALLENGE

Design an AI system for a real-world enterprise problem.

Students define:

Requirements → Architecture → Data → Models → Agents → Tools → Security → Evaluation → Deployment → Cost

Outcome

Learners develop the ability to think beyond individual AI features and design complete AI systems.

WEEK 10

AI ENGINEERING & AGENTIC AI / 10-WEEK TECHNICAL PROGRAM

WEEK 10

CAPSTONE — BUILD A PRODUCTION AI SYSTEM

Stop Following Tutorials. Solve a Real Problem.

The final week brings the program together.

Students design and build an end-to-end AI system around a real-world use case.

CAPSTONE OPTION 1

AI SALES AGENT

Lead research → Company enrichment → Prospect analysis → Personalised outreach → CRM update → Follow-up

Potential capabilities:

- Research prospects
- Retrieve company information
- Analyse leads
- Generate personalised communication
- Interact with CRM systems
- Maintain state
- Escalate to humans

CAPSTONE OPTION 2

AI HR AGENT

CV ingestion → Candidate screening → Evaluation → Scorecard → Interview scheduling

Potential capabilities:

- Extract candidate information
- Compare requirements
- Generate structured evaluations
- Retrieve relevant policies
- Schedule interviews
- Maintain candidate state

CAPSTONE OPTION 3

AI FINANCE AGENT

Invoice ingestion → Extraction → Validation → Reconciliation → Anomaly detection → Reporting

Potential capabilities:

- Process financial documents
- Extract structured information
- Validate data
- Identify anomalies
- Retrieve financial policies
- Generate reports

CAPSTONE OPTION 4

AI RESEARCH AGENT

Research → Retrieval → Analysis → Verification → Citation → Report

Potential capabilities:

- Search information
- Retrieve sources
- Analyse documents
- Cross-check information
- Generate cited outputs
- Produce structured reports

CAPSTONE OPTION 5

AI CUSTOMER SUPPORT AGENT

Knowledge retrieval → Reasoning → Tool use → Response → Escalation

Potential capabilities:

- Retrieve knowledge
- Answer customer questions
- Access customer information
- Execute permitted actions

-
- Escalate complex cases
 - Log interactions

BUILD YOUR OWN

Students may also propose a custom AI system based on a real-world business or technical problem.

The project should demonstrate the engineering principles covered during the program.

CAPSTONE DELIVERABLES

Every final project should include:

Application

- Working AI application
- Backend implementation
- API integration
- Database integration where applicable

AI

- LLM integration
- Context engineering
- Structured outputs
- RAG where applicable
- Agent workflow
- Tool calling

Advanced AI

- Multi-agent workflow where appropriate
- MCP integration where appropriate
- Memory/state
- Human-in-the-loop

Production

- Authentication
- Security
- Guardrails
- Evaluation
- Observability
- Logging
- Cost analysis
- Cloud deployment

Documentation

- System architecture
- Architecture diagram
- Technical documentation
- API documentation
- Evaluation report

-
- GitHub repository
 - Deployment
 - Final project presentation

TECHNOLOGY STACK

AI ENGINEERING & AGENTIC AI / 10-WEEK TECHNICAL PROGRAM

TECHNOLOGY STACK

Languages & Backend

Node.js • TypeScript • NestJS • REST APIs • Git • GitHub

LLMs

OpenAI • Anthropic • Gemini • Open-source LLMs

AI Engineering

LLMs • Context Engineering • Prompt Engineering • Structured Outputs • Function Calling • Tool Calling

RAG

Embeddings • Vector Databases • PostgreSQL/pgvector • Semantic Search • Hybrid Search • Reranking

Agents

LangChain.js • LangGraph.js • Agent Orchestration • Multi-Agent Systems

Integrations

MCP • APIs • Databases • SaaS • External Tools

Data

PostgreSQL • SQL • Vector Databases • Embeddings • Data Pipelines

Production

Docker • AWS • Azure • CI/CD • Monitoring • Observability

AI Quality

Evals • LLM-as-Judge • RAG Evaluation • Agent Evaluation • Regression Testing

AI Security

Guardrails • Prompt Injection Defence • PII Protection • Permissions • Secure Tool Use

AI/ML Interoperability

Python fundamentals • AI/ML APIs • Model Integration

WHAT YOU WILL BUILD

By the end of the program, your learning journey looks like this:

WEEK 2

LLM Application

WEEK 3

Enterprise Knowledge Assistant

WEEK 4

Autonomous AI Agent

WEEK 5

Multi-Agent AI Team

WEEK 6

Tool-Using AI Worker

WEEK 7

AI Evaluation & Security Layer

WEEK 8

Cloud-Deployed AI Application

WEEK 9

Enterprise AI Architecture

WEEK 10

Production AI System

SKILLS YOU WILL DEVELOP

By completing the program, learners will develop practical skills in:

Software Engineering

Node.js, TypeScript, NestJS, APIs, Git, validation and backend architecture.

LLM Engineering

Model APIs, structured outputs, context engineering, tool calling and model selection.

Knowledge Engineering

RAG, embeddings, vector databases, retrieval and reranking.

Agent Engineering

AI agents, memory, state, planning, tools and autonomous workflows.

Multi-Agent Engineering

Agent orchestration, delegation, communication and coordination.

AI Integration

MCP, APIs, databases and external software systems.

AI Quality

Evaluation datasets, LLM evaluation, RAG evaluation and agent evaluation.

AI Security

Guardrails, prompt injection defence, PII protection and permission boundaries.

Production Engineering

Docker, cloud deployment, CI/CD, monitoring, scaling and cost optimisation.

AI Architecture

System design, scalability, reliability, security and cost-aware AI architecture.

BY THE END OF THE PROGRAM

You will be able to:

- Build LLM-powered applications
- ✓ Work with multiple LLM providers
- ✓ Engineer prompts and context
- ✓ Build structured AI workflows
- ✓ Build production-grade RAG systems
- ✓ Work with vector databases
- ✓ Design and build AI agents
- ✓ Build autonomous workflows
- ✓ Build multi-agent systems
- ✓ Connect AI to external tools
- ✓ Build MCP integrations
- ✓ Design AI memory and state
- ✓ Implement human-in-the-loop systems
- ✓ Evaluate LLM applications
- ✓ Evaluate RAG systems
- ✓ Evaluate AI agents
- ✓ Secure AI applications
- ✓ Implement AI guardrails
- ✓ Monitor AI systems
- ✓ Trace agent behaviour
- ✓ Optimise AI costs
- ✓ Deploy AI applications to the cloud
- ✓ Design scalable AI architectures
- ✓ Translate business problems into AI systems
- ✓ Build an end-to-end production AI application

CAREER PATHS

The skills developed in this program are relevant to roles such as:

AI Engineer

Build and integrate AI capabilities into software products.

Generative AI Engineer

Develop applications powered by modern generative AI models.

LLM Engineer

Build applications and systems around large language models.

Agentic AI Engineer

Design AI agents capable of using tools and executing multi-step workflows.

AI Application Engineer

Integrate AI capabilities into real-world software applications.

AI Solutions Engineer

Translate business requirements into deployable AI solutions.

AI Solutions Architect

Design scalable architectures for enterprise AI systems.

AI Automation Engineer

Build AI-powered workflows and autonomous business processes.

AI Platform Engineer

Build infrastructure and systems that support AI applications.

CERTIFICATE

EDUCITY CERTIFICATE OF COMPLETION

Learners who successfully complete the program requirements will receive an:

Educity Certificate of Completion in AI Engineering & Agentic AI

The certificate recognises completion of the 10-week technical program and its required coursework, hands-on projects and capstone requirements.

Certificate verification can include:

- Learner name
- Program name
- Completion date
- Credential ID
- Skills covered
- Capstone project
- Verification link

This is an Educity-issued certificate of completion and is not presented as a university degree or externally accredited academic qualification.

THE EDUCITY AI ENGINEERING APPROACH

LEARN → BUILD → EVALUATE → DEPLOY

LEARN

Understand the technology and engineering principles.

BUILD

Apply the concepts through hands-on projects.

EVALUATE

Measure reliability, quality, security and performance.

DEPLOY

Take your systems from local development to production.

ARCHITECT

Learn to design complete AI systems rather than isolated demos.

THE FINAL OUTCOME

AI ENGINEERING & AGENTIC AI / 10-WEEK TECHNICAL PROGRAM

THE FINAL OUTCOME

Don't Just Learn AI.

BUILD AI.

By the end of 10 weeks, you won't simply have watched lectures about LLMs and AI agents.

You'll have worked through the architecture, engineering, evaluation, security and deployment of real AI systems—and completed a production-oriented capstone that demonstrates what you can build.

10 Weeks. Real Systems. One Production AI Capstone.

AI Engineering & Agentic AI

Build → Evaluate → Secure → Deploy → Scale

INDEX

Contents

Program Overview / 3

Who Is This Program For? / 5

Prerequisites / 6

Your 10-Week AI Engineering Journey / 7

WEEK 1 / 9

WEEK 2 / 13

WEEK 3 / 16

WEEK 4 / 19

WEEK 5 / 22

WEEK 6 / 25

WEEK 7 / 27

WEEK 8 / 30

WEEK 9 / 32

WEEK 10 / 35

Capstone — Build A Production AI System / 37

Build Your Own / 40

Capstone Deliverables / 41

Technology Stack / 43

What You Will Build / 45

Skills You Will Develop / 46

By The End Of The Program / 47

Career Paths / 48

Certificate / 49

The Educity Ai Engineering Approach / 50

The Final Outcome / 51

