

Forward Deployed Engineer (FDE)

AI Implementation, Agent Engineering and Enterprise Automation

Day-wise Training TOC and Curriculum | 10 Days | 80 Hours

DELIVERY

Instructor-led, implementation-first

PRACTICAL EMPHASIS

Approximately 70% hands-on

CORE FOCUS

Agents, automation, integration and deployment

FINAL EVIDENCE

Production-oriented customer capstone

Prepared for practical Forward Deployed Engineer capability development
July 2026

Programme Overview

PROGRAMME PURPOSE Develop Forward Deployed Engineers who can discover customer problems, design AI-enabled workflows, build and integrate agents, implement practical automations, validate quality and security, deploy to production, and drive measurable adoption.

Programme Format

Duration	10 training days / 80 total hours
Practical allocation	Approximately 70% guided builds, independent labs, testing and capstone work
Delivery mode	Instructor-led training with demonstrations, customer simulations and project-based assessment
Longitudinal approach	One enterprise agent solution is built progressively from discovery to production rollout

End-to-End FDE Learning Journey

1. Discover	2. Architect	3. Build	4. Validate	5. Deploy
6. Automate	7. Secure	8. Observe	9. Adopt	10. Scale

Side-by-Side Learning Method

1. Learn Understand the implementation pattern, constraints and business relevance.	2. Observe Review an instructor-built reference implementation and its engineering decisions.
3. Build Implement the same capability in a guided lab.	4. Apply Adapt the capability to the participant's independent capstone scenario.
5. Test Run normal, edge, failure and adversarial test cases.	6. Improve Refactor based on evidence and add failures to regression tests.

10-Day Curriculum at a Glance

Day	Primary Focus	Core Practical Work	Progressive Evidence
1	FDE mindset, discovery and use-case definition	Discovery simulation, workflow mapping and prioritisation	Pilot scope and acceptance criteria
2	AI architecture and model selection	Model benchmark, architecture and cost planning	Architecture decision record
3	Application engineering and prompt-driven backend	Validated API, database state and structured prompts	Working service foundation
4	Single-agent engineering and enterprise tools	Agent with safe read/write tools and approvals	Agent and tool catalogue
5	Knowledge agents and RAG	Document ingestion, retrieval, citations and access filters	Grounded knowledge capability
6	Multi-agent orchestration and automation	Specialist handoffs and event-driven business workflow	End-to-end automation
7	Enterprise integration, MCP and data engineering	Secure connector and governed data pipeline	Enterprise connectivity
8	Evaluation, security and governance	Regression suite, red team and release gate	Quality and security evidence
9	Deployment, CI/CD and observability	Container deployment, pipeline, dashboard and incident drill	Staging-ready solution
10	Integrated capstone and customer rollout	Final build, live incident, demonstration and adoption plan	Production-oriented capstone

DAY 1

FDE Mindset, Customer Discovery and Use-Case Definition

8 Hours | Turning ambiguity into an implementable AI engagement

DAY OUTCOME Participants will convert an unclear customer request into a prioritised, measurable and technically feasible AI implementation opportunity.

Learning Objectives

1. Explain the Forward Deployed Engineer role and its end-to-end ownership model.
2. Conduct business, technical, data, integration and security discovery.
3. Map the current workflow, exceptions, users and decision points.
4. Decide whether a problem requires rules, automation, RAG, an agent or conventional software.
5. Define pilot scope, success measures, acceptance criteria and non-goals.

Day TOC and Time Allocation

Session	Hours	Curriculum Focus	Applied Output
1	0.75	FDE role, operating model and delivery mindset	Engagement ownership map
2	1.25	Customer discovery methods and stakeholder interviewing	Discovery question bank
3	1.25	As-is and to-be workflow mapping	Process and exception map
4	1	AI suitability: software, rules, automation, RAG or agents	Technology decision matrix
5	1	Use-case prioritisation, pilot scope and success metrics	Prioritised use-case backlog
6	2	Guided customer discovery and workflow lab	Customer-ready discovery pack
7	0.75	Independent scenario challenge and peer review	Defensible scope recommendation
TOTAL DAILY LEARNING TIME			8 Hours

Detailed Curriculum Coverage

- FDE versus software engineer, consultant, solution architect and implementation engineer.
- Ownership from discovery through deployment, adoption and measurable business outcome.
- Stakeholder mapping, user personas, current process, bottlenecks, rework and exceptions.
- Discovery of data availability, API dependencies, privacy, security and operational constraints.
- Value-feasibility-risk-adoption scoring and identification of the smallest useful pilot.
- Business KPI baselines, technical acceptance criteria, non-functional requirements and non-goals.
- Statement of work, assumptions register, dependency log and initial implementation backlog.

Hands-On Implementation

GUIDED BUILD

Discovery Simulation: Enterprise Request Resolution

- Interview simulated business, IT and security stakeholders.
- Map the current email-and-spreadsheet request process.
- Identify manual decisions, repeat work, data sources and escalation points.
- Select the first AI-enabled workflow and define measurable success.
- Create acceptance criteria and identify what will remain human-controlled.

INDEPENDENT IMPLEMENTATION

Scope a Different Departmental Use Case

- Choose support, sales, HR, finance, procurement or operations.
- Produce an as-is and to-be workflow.
- Rank at least three use cases using value, feasibility, risk and adoption readiness.
- Present a pilot recommendation and defend why alternative ideas are deferred.

Practical Test Cases and Failure Scenarios

- The customer asks for a tool before defining the business problem.
- Business and security stakeholders provide conflicting requirements.
- No reliable baseline exists for the target KPI.
- The highest-value process has poor or restricted data.
- The requested timeline supports only a prototype, not a safe production rollout.
- Users are likely to resist the redesigned workflow.

Daily Deliverables and Capstone Progress

EVIDENCE PRODUCED

- Customer discovery questionnaire
- Stakeholder and workflow map
- AI suitability decision matrix
- Prioritised use-case backlog
- Pilot scope and acceptance criteria

PROGRESSIVE CAPSTONE MILESTONE

Select the capstone domain, document the current workflow, define the target outcome and establish measurable acceptance criteria.

DAY 2

AI Solution Architecture, Model Selection and Technical Planning

8 Hours | Designing a feasible, secure and cost-aware enterprise AI solution

DAY OUTCOME Participants will design a vendor-neutral AI architecture and justify model, retrieval, automation and integration choices using evidence.

Learning Objectives

1. Explain generative AI behaviour, model limitations and implementation trade-offs.
2. Select model categories based on quality, latency, cost, privacy and tool-use requirements.
3. Choose between prompting, retrieval, fine-tuning, rules and agentic patterns.
4. Design synchronous, asynchronous, batch and real-time execution paths.
5. Create an architecture decision record and initial capacity estimate.

Day TOC and Time Allocation

Session	Hours	Curriculum Focus	Applied Output
1	1	Generative AI implementation fundamentals	AI behaviour and risk notes
2	1	Model categories and model-selection criteria	Model comparison scorecard
3	1	Rules, prompts, RAG, fine-tuning and agent patterns	Pattern-selection decision
4	1	Enterprise architecture and vendor-neutral design	Logical architecture
5	1	Latency, token, cost, privacy and residency planning	Capacity and cost estimate
6	2	Guided architecture and model benchmark lab	Architecture decision record
7	1	Design review and failure-path challenge	Revised production design
TOTAL DAILY LEARNING TIME			8 Hours

Detailed Curriculum Coverage

- Tokens, context windows, structured outputs, embeddings, reasoning and multimodal model categories.
- Probabilistic model behaviour and deterministic validation around AI outputs.
- Hosted, private and self-hosted models; data residency and vendor constraints.
- Model routing, fallback, graceful degradation and abstraction layers.
- Synchronous APIs, queues, background workers, batch processing and real-time sessions.
- Quality-latency-cost-security trade-offs and expected volume calculations.
- Architecture diagrams, trust boundaries, integration contracts and decision records.

Hands-On Implementation

GUIDED BUILD

Model Benchmark and Reference Architecture

- Run the same classification, extraction and summarisation tasks using alternative model configurations.
- Measure accuracy, latency, token usage, structured-output compliance and estimated cost.
- Design the architecture for the Enterprise Request Resolution Agent.
- Add fallback, approval, audit and failure-handling paths.
- Capture decisions and rejected alternatives in an architecture decision record.

INDEPENDENT IMPLEMENTATION

Architecture for the Participant Capstone

- Choose the AI patterns required for the selected business workflow.
- Define model, data, API, storage, automation and user-interaction components.
- Estimate expected transaction volume, latency and model spend.
- Present the design to a mock architecture review board.

Practical Test Cases and Failure Scenarios

- The most accurate model exceeds latency or cost limits.
- Sensitive data cannot leave the customer network.
- The preferred model or vendor is unavailable in the required region.
- A request exceeds the expected context window.
- The fallback model behaves differently when calling tools.
- The design has no safe degradation path during an external service outage.

Daily Deliverables and Capstone Progress

EVIDENCE PRODUCED

- Model benchmark report
- Logical and deployment architecture diagrams
- Architecture decision record
- Cost and latency estimate
- Risk and dependency register

PROGRESSIVE CAPSTONE MILESTONE

Complete the solution architecture, model-selection rationale, trust boundaries, data flows and high-level implementation backlog.

DAY 3

AI Application Engineering and Prompt-Driven Backend Development

8 Hours | Building the reliable service layer underneath enterprise AI

DAY OUTCOME Participants will build a validated backend service that persists workflow state, invokes a model, returns structured outputs and survives common failures.

Learning Objectives

1. Design clean APIs, JSON schemas and service contracts.
2. Implement authentication, validation, database persistence and audit records.
3. Handle asynchronous jobs, webhooks, retries, timeouts and duplicate events.
4. Write robust prompt templates with structured outputs and validation.
5. Create unit, integration and prompt regression tests.

Day TOC and Time Allocation

Session	Hours	Curriculum Focus	Applied Output
1	1	Backend service architecture and REST API design	API contract
2	1	Schemas, validation, authentication and structured errors	Validated endpoints
3	1	Database state, status, audit and migrations	Operational data model
4	1	Async jobs, webhooks, idempotency, retries and timeouts	Resilience patterns
5	1	Prompt and context design for production tasks	Prompt template
6	1	Structured outputs, schema validation and prompt versioning	Validated AI output
7	1.5	Guided backend and prompt implementation lab	Working service
8	0.5	Failure-test challenge and code review	Test evidence
TOTAL DAILY LEARNING TIME			8 Hours

Detailed Curriculum Coverage

- Python/FastAPI or TypeScript/Node.js service structure and separation of concerns.
- API versioning, OpenAPI documentation, JSON schema and consistent error responses.
- Relational tables for requests, status, tool events, approvals and audit history.
- Environment configuration, service identities and secure secret handling.
- Idempotency keys, duplicate webhook protection, retry policies and partial completion.
- System, developer and user instructions; context, constraints and success criteria.
- Few-shot examples, structured JSON, schema enforcement, prompt versioning and change records.

Hands-On Implementation

GUIDED BUILD

Build the AI Request Processing API

- Create endpoints to submit, read and monitor requests.
- Validate incoming payloads and persist request state.
- Invoke the model to extract intent, urgency, missing information and recommended action.
- Validate the model response against a strict schema.
- Record processing time, prompt version and errors in the audit log.

INDEPENDENT IMPLEMENTATION

Extend the Service for a New Business Workflow

- Add domain-specific input and output schemas.
- Implement one asynchronous or webhook-driven step.
- Add unit and integration tests for normal and failure paths.
- Create a prompt test dataset and compare two prompt versions.

Practical Test Cases and Failure Scenarios

- Missing mandatory fields or malformed JSON.
 - Duplicate webhook event or repeated client submission.
 - Database unavailable during processing.
- Model timeout or premature connection close.
 - Model returns readable text but invalid structured output.
 - Application restarts after partially completing the request.

Daily Deliverables and Capstone Progress

EVIDENCE PRODUCED

- Working backend API
- Database schema and migration
- Versioned prompt template
- API and prompt test suite
- Technical API documentation

PROGRESSIVE CAPSTONE MILESTONE

Implement the capstone service skeleton, request database, validated AI task and audit-ready processing status.

DAY 4

Single-Agent Engineering and Enterprise Tool Use

8 Hours | Building an agent that can safely reason, use tools and complete work

DAY OUTCOME Participants will deliver a controlled single-agent application with enterprise tools, state, approvals, stopping conditions and recovery behaviour.

Learning Objectives

1. Differentiate conversational assistants, deterministic workflows and agents.
2. Design agent instructions, execution loops and completion criteria.
3. Create secure read and write tools with validated schemas.
4. Maintain session and task state without uncontrolled context growth.
5. Implement human approval, escalation and loop prevention.

Day TOC and Time Allocation

Session	Hours	Curriculum Focus	Applied Output
1	0.75	Chatbot, workflow and agent decision criteria	Agent suitability decision
2	1	Agent anatomy: model, instructions, tools, state and loop	Agent execution design
3	1.25	Tool and function schema engineering	Tool catalogue
4	1	State, memory, completion and stopping conditions	State model
5	0.75	Approvals, side effects and least privilege	Permission matrix
6	0.75	Retries, timeouts, escalation and recovery	Failure-handling policy
7	2	Guided single-agent build	Working agent and tools
8	0.5	Tool-use test challenge	Agent trace review
TOTAL DAILY LEARNING TIME			8 Hours

Detailed Curriculum Coverage

- Agent objective, constraints, decision loop, state and completion boundaries.
- Clear tool names, descriptions, inputs, outputs and validation rules.
- Read-only versus write-enabled tools and side-effect classification.
- Session memory, task memory, checkpoints and maximum-step controls.
- Human approval for updates, financial actions, notifications and privileged operations.
- Tool result sanitation, error translation, retry limits and human escalation.
- Trace-based review of model decisions, tool calls and completion status.

Hands-On Implementation

GUIDED BUILD

Build an IT Service-Desk Agent

- Create tools to search knowledge, check a user or asset, create a ticket and send a notification.
- Define validated tool schemas and permission levels.
- Persist task state and display live execution status.
- Require approval before a ticket update or notification is sent.
- Add maximum-step, timeout, retry and human-escalation controls.

INDEPENDENT IMPLEMENTATION

Build Domain-Specific Agent Tools

- Implement at least three tools for the participant capstone.
- Include at least one read tool and one controlled write tool.
- Define completion criteria and a safe refusal path.
- Create trace-based tests for correct and incorrect tool selection.

Practical Test Cases and Failure Scenarios

- The agent selects the wrong tool.
 - Tool arguments fail schema validation.
 - The external API is unavailable or returns unexpected data.
- The user requests an unauthorised action.
 - The agent repeatedly calls the same tool.
 - Approval is denied, withdrawn or times out.

Daily Deliverables and Capstone Progress

EVIDENCE PRODUCED

- Single-agent application
- Enterprise tool catalogue
- Tool permission matrix
- Agent state and flow diagram
- Tool-use test report

PROGRESSIVE CAPSTONE MILESTONE

Add the primary agent, validated enterprise tools, persisted task state, approval controls and clear completion rules.

DAY 5

Enterprise Knowledge Agents and Retrieval-Augmented Generation

8 Hours | Grounding agent responses in approved customer knowledge

DAY OUTCOME Participants will build an access-aware knowledge agent that retrieves evidence, cites sources and declines unsupported requests.

Learning Objectives

1. Design a document ingestion and retrieval pipeline.
2. Choose chunking, embedding, metadata, search and reranking strategies.
3. Apply tenant, department and role-based retrieval controls.
4. Generate evidence-backed answers with citations and refusal thresholds.
5. Evaluate retrieval quality and grounded-answer accuracy.

Day TOC and Time Allocation

Session	Hours	Curriculum Focus	Applied Output
1	1	RAG architecture and enterprise knowledge patterns	Retrieval design
2	1	Document ingestion, extraction, cleaning and versioning	Ingestion pipeline
3	1	Chunking, embeddings and vector storage	Indexed knowledge base
4	1	Semantic, keyword, hybrid search and reranking	Retrieval strategy
5	1	Metadata, tenant isolation and access filters	Access-aware retrieval
6	1	Grounded generation, citations and refusal	Evidence policy
7	1.5	Guided knowledge-agent lab	Working RAG agent
8	0.5	Retrieval evaluation challenge	Retrieval test evidence
TOTAL DAILY LEARNING TIME			8 Hours

Detailed Curriculum Coverage

- PDF, DOCX, text, CSV and web content ingestion with document identity and version control.
- Chunk size, overlap, semantic boundaries and metadata enrichment.
- Embeddings, vector stores and hybrid retrieval across structured and unstructured knowledge.
- Query rewriting, filtering, reranking and context assembly.
- Role-based and tenant-aware access controls before retrieval and generation.
- Source citation, evidence thresholds, conflict handling and refusal when support is insufficient.
- Retrieval precision, relevance, groundedness and answer-completeness evaluation.

Hands-On Implementation

GUIDED BUILD

Build a Policy and Procedure Knowledge Agent

- Ingest approved business documents and preserve source metadata.
- Create document chunks and a searchable index.
- Apply department and access-level filters.
- Retrieve evidence and answer with source references.
- Decline answers when no approved evidence is available.

INDEPENDENT IMPLEMENTATION

Add Knowledge Retrieval to the Capstone

- Ingest domain-specific documents and define access metadata.
- Compare at least two chunking or retrieval strategies.
- Create a retrieval test set containing direct, multi-document and unsupported questions.
- Measure and improve retrieval relevance and groundedness.

Practical Test Cases and Failure Scenarios

- The answer is spread across multiple documents.
- Two policy versions conflict.
- The user lacks access to the relevant document.
- No evidence supports the requested answer.
- A document contains malicious instructions.
- The retrieval index contains stale or duplicate content.

Daily Deliverables and Capstone Progress

EVIDENCE PRODUCED

- Document ingestion pipeline
- Vector or hybrid index
- Access-aware knowledge agent
- Retrieval test dataset
- Grounded-answer evaluation report

PROGRESSIVE CAPSTONE MILESTONE

Add approved knowledge sources, access filters, citations and refusal behaviour to the capstone agent.

DAY 6

Multi-Agent Orchestration and AI-Powered Automation

8 Hours | Coordinating specialists and automating work across business systems

DAY OUTCOME Participants will orchestrate specialist agents and build an event-driven automation with approvals, retries, business updates and auditability.

Learning Objectives

1. Decide when multi-agent design is justified and when it adds unnecessary complexity.
2. Implement manager-worker, triage-specialist and agents-as-tools patterns.
3. Design controlled handoffs, shared state and failure recovery.
4. Create trigger-action automations using APIs, webhooks, schedules and low-code platforms.
5. Implement human approval, duplicate prevention, retries and error workflows.

Day TOC and Time Allocation

Session	Hours	Curriculum Focus	Applied Output
1	0.75	When to use multi-agent orchestration	Complexity decision
2	1	Manager-worker, triage and specialist patterns	Agent responsibility map
3	0.75	Handoffs, shared state, memory and loop control	Orchestration flow
4	0.75	Workflow engines, triggers and automation platforms	Automation design
5	1	Webhooks, schedules, queues and conditional routing	Event flow
6	1	Approvals, retries, idempotency and error workflows	Resilience controls
7	2	Guided multi-agent automation lab	End-to-end workflow
8	0.75	Independent automation challenge	Exportable automation
TOTAL DAILY LEARNING TIME			8 Hours

Detailed Curriculum Coverage

- Role boundaries, specialist prompts, shared versus isolated context and one source of truth.
- Agent handoff contracts, loop detection, conflict resolution and escalation.
- Email, form, chat, file, webhook, schedule and event triggers.
- Low-code tools such as n8n or Power Automate combined with coded services.
- Human approval, timeout, rejection and rework paths.
- Idempotency keys, retry strategies, dead-letter handling and execution history.
- Long-running workflows, status visibility, notifications and audit logs.

Hands-On Implementation

GUIDED BUILD

Automate Enterprise Request Resolution

- Route a new request through triage, knowledge and action specialists.
- Trigger the flow from email, form or webhook.
- Retrieve evidence, classify risk and prepare a recommended action.
- Pause for human approval before a system update.
- Update the business system, notify the requester and write an audit record.

INDEPENDENT IMPLEMENTATION

Build a Departmental Automation

- Create an invoice, recruitment, CRM, support or document-review automation.
- Include an AI decision, at least one business-system update and one approval.
- Implement duplicate prevention and an error workflow.
- Export the workflow and create an operations runbook.

Practical Test Cases and Failure Scenarios

- Duplicate trigger or repeated delivery.
- The approver rejects the recommended action.
- The approver does not respond before timeout.
- The business API fails after partial completion.
- Two specialist agents provide conflicting recommendations.
- The orchestration enters a handoff or retry loop.

Daily Deliverables and Capstone Progress

EVIDENCE PRODUCED

- Multi-agent orchestration flow
- Automation workflow
- Approval and escalation matrix
- Error-handling workflow
- Automation operations runbook

PROGRESSIVE CAPSTONE MILESTONE

Add specialist orchestration and one complete event-driven automation from trigger through approval, business update, notification and audit.

DAY 7

Enterprise Integration, MCP and Data Engineering for AI

8 Hours | Connecting agents to enterprise capabilities and trustworthy data

DAY OUTCOME Participants will expose approved business operations through a secure connector or MCP server and create a quality-controlled data pipeline.

Learning Objectives

1. Integrate AI solutions with APIs, databases, files and enterprise applications.
2. Explain MCP hosts, clients, servers, tools, resources and prompts.
3. Build a custom connector or MCP server with secure authorisation.
4. Profile, clean, transform and validate operational data.
5. Define data contracts, lineage, PII handling and schema-change controls.

Day TOC and Time Allocation

Session	Hours	Curriculum Focus	Applied Output
1	1	Enterprise API and integration patterns	Integration contract
2	1	MCP concepts, architecture and security boundaries	MCP design
3	1	Custom tools, connectors, OAuth and service identities	Secure capability exposure
4	1	Data-source discovery and profiling	Data inventory
5	1	Batch, event and database ingestion pipelines	Data pipeline design
6	1	Data quality, PII, lineage and schema contracts	Data governance controls
7	1.5	Guided connector and data-pipeline lab	Working integration
8	0.5	Integration failure challenge	Contract test evidence
TOTAL DAILY LEARNING TIME			8 Hours

Detailed Curriculum Coverage

- REST, GraphQL, webhooks, relational databases, file storage and collaboration systems.
- MCP host, client and server responsibilities; tools, resources and prompts.
- OAuth, token scopes, service identities, least privilege and credential rotation.
- Read and write capability boundaries, tool approvals and sensitive-field filtering.
- Data profiling, nulls, duplicates, invalid formats and master-data alignment.
- ETL/ELT, batch and event-driven ingestion, change data capture and schema evolution.
- Data lineage, retention, PII classification, data contracts and quality thresholds.

Hands-On Implementation

GUIDED BUILD

Build a Secure Enterprise Connector

- Expose approved operations such as search record, create ticket or check invoice status.
- Validate tool inputs and restrict scopes by user role.
- Log tool requests, responses and side effects.
- Ingest API, CSV and database data into validated operational tables.
- Create contract tests and detect a simulated schema change.

INDEPENDENT IMPLEMENTATION

Connect the Capstone to Realistic Enterprise Data

- Implement one custom connector or MCP server.
- Add one relational or file-based data pipeline.
- Document field mapping, quality rules and access restrictions.
- Test expired tokens, insufficient permissions and schema changes.

Practical Test Cases and Failure Scenarios

- Expired or revoked access token.
 - User or tool requests excessive permissions.
 - External API schema changes unexpectedly.
- Duplicate or inconsistent business records.
 - Restricted data appears in a tool response.
 - A write operation is attempted without approval.

Daily Deliverables and Capstone Progress

EVIDENCE PRODUCED

- Custom connector or MCP server
- Tool and API contract
- Authorisation and permission design
- Validated data pipeline
- Data quality and lineage report

PROGRESSIVE CAPSTONE MILESTONE

Connect the capstone to at least one enterprise system and one governed data source through documented, least-privilege contracts.

DAY 8

Evaluation, Security, Guardrails and AI Governance

8 Hours | Proving quality and controlling AI risk before release

DAY OUTCOME Participants will build an automated evaluation suite, red-team the solution, implement guardrails and make an evidence-based release decision.

Learning Objectives

1. Define measurable success for nondeterministic AI workflows.
2. Build representative golden datasets and automated evaluators.
3. Test retrieval, tool use, task completion, latency, cost and safety.
4. Threat-model prompt injection, data leakage, tool abuse and cross-user access.
5. Implement input, output and tool guardrails, approvals, RBAC and audit controls.

Day TOC and Time Allocation

Session	Hours	Curriculum Focus	Applied Output
1	1	Evaluation strategy, metrics and release thresholds	Evaluation plan
2	1	Golden datasets, edge cases and failure taxonomy	Test dataset
3	1	Validators, graders and regression testing	Evaluation harness
4	1	AI threat modelling and attack surfaces	Threat model
5	1	Prompt injection, data leakage and tool abuse	Adversarial test plan
6	1	Guardrails, approvals, RBAC, audit and governance	Security controls
7	1.5	Guided evaluation and red-team lab	Quality and security results
8	0.5	Production release-gate review	Go/no-go decision
TOTAL DAILY LEARNING TIME			8 Hours

Detailed Curriculum Coverage

- Task-completion rate, extraction accuracy, groundedness, retrieval relevance and tool success.
- Deterministic checks, human review and model-based graders.
- Offline tests, online monitoring, regression suites and production feedback loops.
- Prompt injection, indirect injection, system-prompt exposure and data exfiltration.
- Tenant isolation, role-based access, least privilege and human approval for side effects.
- PII detection, output filtering, tool restrictions, audit logging and data retention.
- Release thresholds, risk acceptance, exception handling and governance sign-off.

Hands-On Implementation

GUIDED BUILD

Evaluate and Red-Team the Reference Agent

- Create normal, edge, tool-use, retrieval, security and recovery cases.
- Automate schema, citation, tool and task-completion checks.
- Compare two prompt or model versions.
- Attempt prompt injection, unauthorised access and approval bypass.
- Implement controls and rerun the regression suite.

INDEPENDENT IMPLEMENTATION

Create the Capstone Release Gate

- Build at least 30 representative test cases.
- Define quality, latency, cost and safety thresholds.
- Produce a threat model and security control matrix.
- Make a documented release decision using test evidence.

Practical Test Cases and Failure Scenarios

- A prompt change improves answer quality but worsens tool accuracy.
 - A malicious document attempts to redirect the agent.
 - A user requests another tenant’s information.
- The model provides an unsupported but plausible answer.
 - A sensitive action is attempted without human approval.
 - The solution meets accuracy targets but exceeds cost or latency limits.

Daily Deliverables and Capstone Progress

EVIDENCE PRODUCED

- Evaluation strategy and golden dataset
- Automated evaluation harness
- AI threat model
- Security control matrix
- Production release-gate report

PROGRESSIVE CAPSTONE MILESTONE

Complete the capstone regression suite, threat model, guardrails, access controls and evidence-based release criteria.

DAY 9

Production Deployment, CI/CD, Observability and Reliability

8 Hours | Operating AI solutions safely, reliably and cost-effectively

DAY OUTCOME Participants will containerise, deploy and monitor the solution through a tested pipeline with health checks, rollback, traces, alerts and cost controls.

Learning Objectives

1. Create repeatable development, staging and production environments.
2. Containerise multi-service AI applications and manage configuration safely.
3. Build CI/CD quality gates including code, integration and AI evaluation tests.
4. Implement logs, metrics, traces, dashboards and alerts.
5. Apply reliability, performance and AI FinOps patterns.

Day TOC and Time Allocation

Session	Hours	Curriculum Focus	Applied Output
1	1	Environments and containerisation strategy	Deployment architecture
2	1	Dockerfiles, Compose and service dependencies	Container stack
3	1	CI/CD pipeline and release quality gates	Automated pipeline
4	1	Secrets, migrations, health checks and rollback	Production controls
5	1	Logs, metrics, traces and operational dashboards	Observability design
6	1	Reliability, load, latency, fallback and AI FinOps	SLO and cost plan
7	1.5	Guided deployment and monitoring lab	Staging deployment
8	0.5	Incident simulation and recovery	Incident evidence
TOTAL DAILY LEARNING TIME			8 Hours

Detailed Curriculum Coverage

- Development, test, staging and production configuration and promotion paths.
- Dockerfiles, Docker Compose, persistent data, dependency order and health checks.
- CI/CD pipeline stages, artifact registries, approvals and rollback procedures.
- Secret management, database migrations and environment-specific configuration.
- Application logs, model calls, tool calls, agent traces, business metrics and correlation IDs.
- Service-level objectives, alerts, retries, backoff, circuit breakers, queues and caching.
- Load testing, token usage, model cost, fallback routing and capacity planning.

Hands-On Implementation

GUIDED BUILD

Deploy and Operate the Reference Solution

- Containerise the API, database, retrieval and automation components.
- Create health and readiness endpoints.
- Build a CI/CD workflow that runs tests and evaluations.
- Deploy to staging and create an operational dashboard.
- Simulate an outage or failed release and execute rollback.

INDEPENDENT IMPLEMENTATION

Productionise the Participant Capstone

- Create the deployment package and environment configuration.
- Add logs, traces, metrics and cost telemetry.
- Define alerts, SLOs and an incident runbook.
- Run a load or reliability test and implement one optimisation.

Practical Test Cases and Failure Scenarios

- Missing secret or incorrect environment configuration.
 - Database starts after the application or migration fails.
 - AI evaluation score falls below the release threshold.
- Model API outage or rate limit.
 - Slow database or business API causes cascading latency.
 - New release fails health checks and requires rollback.

Daily Deliverables and Capstone Progress

EVIDENCE PRODUCED

- Dockerised application stack
- CI/CD workflow
- Staging deployment
- Monitoring dashboard and alert rules
- Incident and rollback runbook

PROGRESSIVE CAPSTONE MILESTONE

Deploy the capstone to staging with automated quality gates, health checks, monitoring, alerts, rollback and basic cost visibility.

DAY 10

Integrated FDE Capstone, Customer Rollout and Value Realisation

8 Hours | Demonstrating a secure production-ready implementation and adoption plan

DAY OUTCOME Participants will deliver, defend and operate an end-to-end AI implementation from customer problem through production rollout and measurable value.

Learning Objectives

1. Complete and integrate the capstone solution against agreed acceptance criteria.
2. Run final functional, AI, security and operational validation.
3. Respond to an unannounced production incident.
4. Demonstrate the solution to business and technical stakeholders.
5. Prepare pilot, UAT, adoption, support, KPI and scale-up plans.

Day TOC and Time Allocation

Session	Hours	Curriculum Focus	Applied Output
1	0.75	Architecture, backlog and acceptance-criteria review	Final build plan
2	3	Capstone implementation and integration sprint	Completed solution
3	1	Evaluation, security and production-readiness validation	Release evidence
4	1	Deployment and live incident challenge	Operational response
5	0.75	Customer demonstration and technical defence	Stakeholder presentation
6	0.75	Pilot, UAT, adoption, KPI and rollout planning	Rollout plan
7	0.75	Final review, feedback and certification assessment	Assessment outcome
TOTAL DAILY LEARNING TIME			8 Hours

Detailed Curriculum Coverage

- Acceptance-criteria traceability and final defect prioritisation.
- Integration of agent, retrieval, tools, automation, data, security and observability.
- Final regression, red-team, latency, cost and reliability validation.
- Go/no-go decision, risk acceptance and production-readiness review.
- Executive demonstration, technical walkthrough and stakeholder objection handling.
- Pilot selection, user-acceptance testing, onboarding and change-management actions.
- Adoption, business KPI, ROI, feedback, support, handover and expansion planning.

Hands-On Implementation

GUIDED BUILD

Final Integrated Capstone Delivery

- Complete the solution and close critical test failures.
- Deploy the approved build and verify monitoring and audit records.
- Respond to an unannounced outage, injection, data or integration incident.
- Demonstrate the customer workflow from trigger through business outcome.
- Present evidence of quality, security, reliability and expected value.

INDEPENDENT IMPLEMENTATION

Customer Rollout and Scale Plan

- Define pilot users, UAT cases and go-live criteria.
- Create onboarding, training, support and escalation materials.
- Define adoption, accuracy, time-saving and financial KPIs.
- Identify reusable components and the next workflows to scale.

Practical Test Cases and Failure Scenarios

- Model or integration outage during the live demonstration.
- Prompt-injection attempt from a user or document.
- Duplicate automation execution after a retry.
- Retrieval returns an outdated policy.
- Users do not adopt the technically correct solution.
- Executive stakeholders challenge the cost, risk or expected value.

Daily Deliverables and Capstone Progress

EVIDENCE PRODUCED

- Complete deployable capstone solution
- Evaluation and security report
- Production-readiness checklist
- Customer demonstration and technical handover
- Pilot, adoption and KPI plan

PROGRESSIVE CAPSTONE MILESTONE

Final submission must demonstrate discovery, architecture, coded service, agent tools, retrieval, automation, enterprise integration, evaluation, security, deployment, observability and customer rollout.

Assessment and Certification Framework

ASSESSMENT PRINCIPLE Participants are assessed on working evidence: discovery quality, implementation correctness, test coverage, security, operational readiness, customer communication and measurable outcome—not on theory recall alone.

Assessment Component	Weight	Evidence
Daily guided labs and engineering deliverables	25%	Working features, code quality, implementation notes and test evidence
Independent agent and automation challenges	20%	Original adaptation, business integration and failure recovery
Evaluation, security and governance	15%	Golden dataset, regression suite, threat model and controls
Integrated FDE capstone	30%	Secure, evaluated, deployable solution meeting acceptance criteria
Customer demonstration and handover	10%	Stakeholder communication, runbook, adoption and KPI plan

Recommended Passing Criteria

- Minimum overall score of 70%.
- Completion of all mandatory daily labs and capstone milestones.
- No unresolved critical security, access-control or data-exposure issue.
- Capstone successfully deployed to a staging environment.
- Evaluation results meet the release thresholds defined by the participant team.
- Participant demonstrates the ability to diagnose and communicate a live incident.

Mandatory Capstone Components

- Customer discovery summary and as-is/to-be workflow.
- Measurable success criteria and acceptance tests.
- Architecture and model-selection decision record.
- Backend API and operational database.
- AI agent with at least three validated enterprise tools.
- Knowledge retrieval with citations and access controls.
- At least one event-driven automation and human approval step.
- At least one external business-system integration or MCP connector.
- Evaluation dataset, regression harness and release thresholds.
- Guardrails, RBAC, audit logging and threat model.
- Containerised deployment with CI/CD, health checks and rollback.
- Monitoring, tracing, alerts, incident runbook and cost visibility.
- Customer demonstration, pilot, adoption and KPI plan.

Expected Learning Outcomes

After completing the 10-day programme, participants will be able to:

1. Conduct customer discovery and translate ambiguous requirements into a prioritised AI implementation.
2. Decide when to use conventional software, rules, workflow automation, retrieval or agents.
3. Design secure, vendor-neutral AI architectures with realistic latency and cost estimates.
4. Build validated backend services and structured prompt pipelines.
5. Create agents with safe tools, state, approval, recovery and completion controls.
6. Build enterprise knowledge agents that answer with approved evidence and citations.
7. Orchestrate specialist agents and practical event-driven automations.
8. Integrate agents with enterprise systems through APIs, databases and MCP connectors.
9. Prepare governed operational, retrieval and evaluation data pipelines.
10. Build automated AI evaluation and regression-testing suites.
11. Apply threat modelling, guardrails, least privilege, audit and responsible-AI governance.
12. Containerise, deploy and release AI applications through CI/CD.
13. Operate production AI solutions using traces, metrics, alerts, incident response and FinOps.
14. Run pilots, user acceptance, adoption and business-value measurement.
15. Convert field learning into reusable implementation patterns, accelerators and product feedback.

FDE Toolkit Produced During Training

- Discovery questionnaire, stakeholder map and workflow templates.
- Use-case prioritisation matrix and pilot scoping template.
- Architecture decision record and model comparison scorecard.
- API contracts, data schemas and agent tool specifications.
- Prompt library, retrieval test set and evaluation harness.
- Automation workflow, approval matrix and error runbook.
- Threat model, security control matrix and release checklist.
- Docker deployment, CI/CD pipeline and monitoring dashboard.
- Incident response, customer handover, adoption and KPI templates.