

Low-Code AI Automation

Three days for people who do not write code. Everyone finishes having built AI agents that read the organisation's own documents through RAG, make decisions, take real actions in real systems, and run unattended — using n8n and NotebookLM, with no programming at all.

DURATION	HANDS - ON	CODING	TOOLS
3 days · 1,170 min	~75% of contact time	None required	n8n · NotebookLM

TECHNOLOGIES AND TOPICS COVERED

RAG (Retrieval-Augmented Generation)

AI agents

Agentic AI

n8n

NotebookLM

No-code automation

Low-code AI

LLM (Large Language Model)

GenAI (Generative AI)

Prompt engineering

Context engineering

Grounded answers

Citations

Hallucination control

Structured output

Agent orchestration

Multi-agent workflows

Vector store

Document search

Conditional logic and branching

Webhooks

Scheduled triggers

Human-in-the-loop approval

Error handling

API credentials

Google Sheets integration

Workflow monitoring

Usage quotas and cost

Data safety and governance

What each participant can do afterwards

✓ Tell AI, generative AI, AI agents and agentic AI apart, using an example from their own work.

✓ Write instructions that keep an AI agent inside its reference data instead of guessing.

✓ Make an automation **write to a real system of record**, not just print an answer.

✓ Trigger a workflow on a schedule, a form, an email or a webhook — with nobody watching.

✓ Use **RAG to get grounded, cited answers** from their own documents.

✓ Build and chain several AI agents that hand work to one another.

✓ Add branching so a workflow handles the exceptional case, not only the happy path.

✓ Put a **human approval step** in front of anything consequential.

EACH PARTICIPANT LEAVES WITH

– A **grounded RAG workspace** built on their own team's documents.

– A **multi-agent workflow** running end to end that writes a real record.

– One automation for **their own recurring task**, on a trigger, with approval and error handling.

– A written manual with every step and instruction, so it can all be rebuilt independently.

THE ORGANISATION GAINS

– **Automation capability outside the engineering queue** — the people closest to a process can improve it.

– Working automations on day three, not a report recommending them.

– A shared understanding of **what is safe to put into an AI tool** and what is not.

– A clear boundary for escalating to engineering, and a handover they will accept.

WHO THIS IS FOR

People who do not write code — operations, business analysis, finance, HR, customer support, quality, project management and team leadership. Comfort with a browser, email and spreadsheets is all that is assumed. It suits organisations that want automation capability distributed across the business rather than concentrated in engineering.

Day by day

Timed to an eight-hour day with 90 minutes of breaks — 390 minutes of instruction. One realistic example is grown across the three days, set to the client's own domain during scoping, before Day 3 moves onto each participant's real work.

DAY 1 Grounded answers — GenAI and RAG, hands-on

60 + 90 + 105 + 135 = 390 min

- 60
min

Orientation, and the confident wrong answer CONCEPT + LAB
AI, generative AI (GenAI), AI agents and agentic AI told apart on an example from your own operation, then why a fluent answer is not a correct one. **Build:** everyone provokes a confident, wrong answer about their own department.
- 90
min

Prompt and context engineering LAB
Role, context, constraints, output shape and examples — turning a vague request into one that returns the same usable result every time. **Build:** write an instruction for a real task, then swap and try to break a neighbour's.
- 105
min

RAG — answers grounded in your own documents LAB
Retrieval-Augmented Generation with no code, using NotebookLM: upload a document set, read the citations, see the honest “not in the sources” refusal, then add the missing document and watch the answer change. Where document AI struggles — scans, spreadsheets, tables. **Build:** a grounded workspace on your own team's documents.
- 135
min

n8n foundations, and your first AI agent LAB
Workflows, nodes, triggers and saved credentials. The AI agent node and the LLM behind it. Writing instructions that pin an agent to its reference data. **Build:** a working single-purpose AI agent that correctly refuses out-of-scope questions.

DAY 2 AI agents that work together

90 + 120 + 105 + 75 = 390 min

- 90
min

A second agent, and structured output LAB
The pattern repeats faster the second time. Then why fixed fields matter the moment anything downstream consumes the answer, and how free text corrupts a spreadsheet or ticket queue. **Build:** a third agent forced to return a defined set of fields every time.
- 120
min

Agent orchestration — handing work along a chain LAB
Chaining AI agents on one canvas and passing data between them, including the referencing details that catch everyone first time. **Build:** a three-agent workflow running end to end, tested on a routine case and an urgent one.
- 105
min

Conditional logic and branching LAB
Conditions, filters and routing — urgent items down one path, routine down another. The single biggest difference between a demo and something a team relies on. **Build:** add a decision point and test both paths.
- 75
min

Taking a real action LAB
Connecting to a system of record and writing to it — access, field mapping, and why a durable log beats a notification email. **Build:** the workflow appends a real, correctly-typed record every run.

- 90min

Triggers — starting without a human

LAB

Running on a schedule, on a submitted form, on an incoming email or message, or on a webhook from another system. The step that turns an assistant into an automation. **Build:** convert your workflow to run unattended, twice over.
- 90min

Your own RAG pipeline inside the workflow

LAB

The grounded answering from Day 1, rebuilt inside n8n with a vector store so it runs on your documents, on your schedule, inside your own automation. **Build:** ingest a document set and have an agent answer from it, with sources, in-workflow.
- 105min

Approval, failure, cost and data safety

LAB

Human-in-the-loop approval before anything commits. Partial runs, duplicates and silent failures when the AI is confidently wrong. Usage quotas and running cost. What must never be put into a hosted tool. **Build:** add an approval step, then break the workflow and make the failure visible.
- 105min

Build your own, and know where to stop

BUILD + REVIEW

Supported build time on a real task from each participant's role. Closing on the honest boundary — volume, auditability, latency and data sensitivity are the four lines where a workflow belongs with engineering. **Close:** each participant demonstrates an automation built for their own work.

How it is delivered

GROUP SIZE Hands-on work in groups of five to six, with a second facilitator for larger cohorts, so every participant gets their own build working.	VIRTUAL DELIVERY Six half-days across two to three weeks — same content and labs, with better attendance and retention than consecutive full days.	NOTHING TO INSTALL Everything runs in the browser. A laptop and an internet connection is all that is needed — no software, no servers, no developer environment.
GETTING READY A short readiness call and a simple setup checklist are provided ahead of delivery, so the first morning starts with teaching rather than configuration.		

Arjun Thakur

Principal AI Engineer · AI Architect · Corporate AI Trainer

- **15+ years of production software engineering** across payments, travel, fintech and education technology — senior engineering, engineering management and director-level roles at global product companies.
- **Builds and operates a production AI product** — AI agent orchestration, RAG, evaluation and guardrails running live with paying customers. This material is taught by someone who runs it.
- **Teaches non-technical audiences as readily as engineers.** A low-code programme is judged by whether the least confident person in the room finishes with something working, and every day is designed around that.