tripleten

AI & Machine Learning

20-week accelerator

Online · Part-time

For tech professionals with production experience

Syllabus

2026

tripleten

How this program works

Ship models, not notebooks.

Five production-grade projects, one continuous storyline: you are hired into real product teams, get the same briefs real engineers get, and ship models, services, fine-tuned LLMs, and agents that end up in one deployed SaaS product. Claude Code runs through the whole program as your engineering assistant.

5 projects → one deployed AI product

~31% theory / ~69% hands-on

~20 hours a week · on your schedule

~2 hours of live sessions weekly

20 weeks total

Project

The unit of the program. You get a realistic brief, a provided production-shaped codebase or dataset, milestones, and a deliverable you can defend.

Theory topics

Short lessons attached to each project — around 15 minutes each. Experienced students move faster, others take more time: project hours are fixed, theory time flexes.

Hours

Every project shows theory hours, project hours, and the total. Totals assume the recommended pace of 20 hours per week.

Who this program is for

A quick self-check before you commit — honest in both directions.

You're a good fit if

- You have hands-on production experience — as an engineer or in an adjacent technical role
- You can find your way around an unfamiliar codebase and pick up new tools as you go
- You want to grow into higher-leverage engineering work without quitting your job — the program is part-time, on your own schedule

This is probably not for you if

- You're starting from zero — no coding experience yet. TripleTen's core bootcamps are the right entry point; this program assumes a working base.
- You're after a certificate rather than a portfolio of defended work.

Program summary

At the recommended pace of 20 hours per week. Project hours are fixed; theory time flexes with your experience, so the total may vary by a week or two.

6.2	13.9	20
weeks of theory ~125 hours	weeks of project work ~279 hours	weeks total

THE 5 PROJECTS · HOURS = THEORY + PROJECT WORK, TOTAL PER PROJECT

1	Pulse: from PRD to working model	~111 h
2	Pulse in production: model as a service	~85.5 h
3	CareScribe: fine-tune a clinical LLM	~66 h
4	Resolve: a production support agent	~78 h
5	Atlas: the capstone AI product	~63.5 h

Each project is detailed on the following pages.

Project 1

Pulse: from PRD to working model

Theory	~45 h
Project work	~66 h
Total	**~111 h**

You join a team shipping Pulse, and turn a product requirements doc into a working model: pull data from live APIs into a database, then train, evaluate, and track it. You finish with a model that solves a stated business problem.

THEORY TOPICS

- Project Brief: PRD to ML Spec
- Claude Code for AI Engineering
- The Web as a Data Source
- APIs to Database: The Ingestion Pipeline
- Regression Fundamentals
- Training Regressors in scikit-learn
- Classification Fundamentals
- Preparing Features for Training
- Encoding & Scaling
- Accuracy & the Confusion Matrix
- Precision, Recall & Class Imbalance
- Thresholds & ROC/PR Curves
- Regression Metrics
- How Neural Networks Learn
- Training Mechanics: Gradients & Validation
- Optimizers, Losses & Model Fit
- PyTorch Fundamentals
- Applied Neural Network Builds
- Training Deep Networks Well
- Experiment Tracking with MLflow

SKILLS YOU'LL ADD

- Tensorflow/Keras/PyTorch
- Model Training
- Data collection
- Model Evaluation
- Claude Code

LEARNING OBJECTIVES

- → Translate a PRD into an ML spec with framing, success metrics, and baselines
- → Use Claude Code as an engineering assistant with a draft → verify discipline (CLAUDE.md, core workflows, decision framework)
- → Build an ingestion pipeline: MBTA V3 API → validation → PostgreSQL (`stop_events`, idempotent upserts)
- → Train and evaluate regression and classification models in scikit-learn.

Project 2

Pulse in production: model as a service

Theory	~27.5 h
Project work	~58 h
Total	**~85.5 h**

Take the Pulse model out of the notebook and put it behind a real service: FastAPI, Docker, Kubernetes, and AWS, with an API-to-queue-to-worker flow for async inference. You finish with a deployed, monitored system that survives traffic and failure.

THEORY TOPICS

FastAPI for ML Services · Model Serialization & Management · API Best Practices · Docker Fundamentals · Building Docker Images · Docker for ML Applications · Kubernetes · Cloud Fundamentals & AWS · Amazon S3 & IAM · Amazon EC2 · Amazon ECR & ECS · Async Inference with SQS, SNS & Lambda · NoSQL on AWS · Production Requirements & System Design · MLOps Foundations & Testing · Observability

SKILLS YOU'LL ADD

Dockerization · **Kubernetes** · **AWS** · **SQL/NoSQL** · **Production Requirements/System Design** · **Claude Code**

LEARNING OBJECTIVES

- Serve models with FastAPI + Pydantic: validation, auth (API keys), rate limiting, auto-generated docs
- Serialize, version, and load models safely (joblib traps, semver, caching, startup loading)
- Containerize with Docker (images, layer caching, env vars, volumes) and compose multi-container systems
- Orchestrate with Kubernetes: Deployments, Services, ConfigMaps/Secrets, probes, HPA, zero-downtime rollouts

Project 3

CareScribe: fine-tune a clinical LLM

Theory	~20 h
Project work	~46 h
Total	**~66 h**

Fine-tune an open LLM to draft clinical patient summaries in the style hospital doctors actually write. You curate and de-identify the training data, run PEFT/QLoRA training on AWS GPUs, and evaluate generated text against a clinician rubric rather than a loss curve.

THEORY TOPICS

Transformers & BERT · LLM Fundamentals · Generation Parameters · PEFT & LoRA · QLoRA · Hugging Face Fine-Tuning Workflow · Clinical Data Curation · GPU Training on AWS & SageMaker · LLM Evaluation

SKILLS YOU'LL ADD

LLMs · PEFT · qlora · data curation · PyTorch · GPU training on AWS · Claude Code

LEARNING OBJECTIVES

- Explain transformer architecture, attention, and the BERT-to-LLM lineage
- Control generation: temperature, top-p, penalties, and structured decoding
- Apply parameter-efficient fine-tuning: adapters, LoRA, and QLoRA (quantization, bitsandbytes)
- Run the Hugging Face fine-tuning workflow end to end on the course PyTorch foundations

Project 4

Resolve: a production support agent

Theory	~20 h
Project work	~58 h
Total	**~78 h**

Build and productionize a customer-support agent on Bedrock AgentCore that reads incoming tickets, screens them for prompt injection, searches the company knowledge base with RAG, and proposes a resolution behind output guardrails. You finish with an agent that is observable, evaluated, and hardened against the OWASP LLM risks.

THEORY TOPICS

Prompt Engineering · Retrieval-Augmented Generation (RAG) · OpenSearch for RAG · Agent Fundamentals · LangChain & LangGraph · Model Context Protocol (MCP) · Jira & Confluence Tool Integration · Amazon Bedrock & AgentCore · LangSmith Observability · LLM-as-a-Judge Evaluation · Hallucination Metrics · OWASP Threat Modeling for LLMs · AI Defenses & Red Teaming

SKILLS YOU'LL ADD

Langchain · Langgraph · LangSmith (observability) · MCP (Jira, Confluence) · RAG (company knowledgebase) · Tool calling · skills · orchestration · OWASP (framework) · Input and output guardrails · security · prompt injection · LLM-as-a-judge · hallucination metrics

LEARNING OBJECTIVES

- Engineer prompts systematically, including prompting-for-agents (tool calling, skills, orchestration)
- Build RAG pipelines with OpenSearch as the vector store
- Design agent workflows with LangChain & LangGraph (support-ticket graph)
- Connect agents to real tools via MCP: Jira and Confluence sandboxes

Project 5

Atlas: the capstone AI product

Theory	~12.5 h
Project work	~51 h
Total	**~63.5 h**

Bring everything together in Atlas, a SaaS dashboard where the model, the deployed service, the fine-tuned LLM, and the agent all ship as features of one product. You decide feature by feature where generative AI earns its cost and where deterministic code wins, then deploy the whole thing on AWS.

THEORY TOPICS

- Dashboard Frontend Development
- Deterministic vs. Generative AI
- Integrating AI into SaaS Products
- SaaS Architecture & Product Concerns
- Deploying AI Applications on AWS
- End-to-End AI Product Development
- Portfolio Presentation & Delivery

SKILLS YOU'LL ADD

- Everything from the previous projects

LEARNING OBJECTIVES

- Scope AI feature ideas against an existing product: feasibility, value, and the deterministic-vs-generative call
- Clone and run a provided containerized SaaS app locally and integrate new services with it
- Build and train predictive models on provided patient data and expose them as product features
- Optionally build LLM-powered features: fine-tuned, RAG-based, or agentic (patient record → insights)

How every project works

How each project gets reviewed — and what your week looks like along the way.

Reviews on every project

Readiness Review. When your project feels done, you book a short 15–30 min check with your instructor to confirm you're ready.

Final Review. You present and defend the project live on the nearest review day — they run every two weeks. The panel grows with the project, up to a Tier-1 engineer on the capstone.

Your weekly rhythm

A 30-minute 1:1 with an instructor every week — optional, bring anything

Your project group chat with tutors — always on

You move in project cohorts — grouped by the project you're on, not by start week

Review days run every two weeks — you book the nearest one when your project is ready

Group formats run evenings and weekends

Schedule example

At the guideline pace of 20 hours per week. The career track runs alongside your projects — its milestones are the highlighted rows.

WEEKS	STAGE	FINAL REVIEW
Before start	We get on a call and build your personal study plan together	
Weeks 1–6	Project 1 · Pulse: from PRD to working model	~week 6
~Week 6	**Career** · Your Career Coach joins — resume audit and target roles	
Weeks 6–10	Project 2 · Pulse in production: model as a service	~week 10
Weeks 10–13	Project 3 · CareScribe: fine-tune a clinical LLM	~week 13
~Week 13	**Career** · Resume repositioned, reviewed projects become portfolio cases	
Weeks 14–17	Project 4 · Resolve: a production support agent	~week 17
Weeks 14–20	**Career** · Mock interviews — unlimited AI practice, live technical and HR	
Weeks 18–20	Project 5 · Atlas: the capstone AI product	~week 20 · Tier-1 engineer
Graduation	**Career** · Final package and a structured search pipeline — support continues	

Theory time flexes with your experience, and pace is individual: faster and slower paths exist — treat week numbers as guides, not deadlines.

How we support you

You study on your own schedule, but you're never on your own: three layers of support run through the whole program.

Teaching

- An optional 30-minute 1:1 with an instructor every week — bring your questions, your code, or your architecture
- Fast answers in your project chat whenever you get stuck
- Every project reviewed live by a working engineer

Community

- A project group chat — everyone in it is building the same thing you are
- Senior-level engineers from different companies around you, not a beginners room
- Weekly live group sessions and events with instructors
- Networking and random coffee pairings to meet peers one-on-one
- A student directory — find peers by background and reach out directly
- On graduation you join TripleTen's alumni network — 7,500+ graduates worldwide

Onboarding

- A personal manager who has your back on anything organizational
- Keeps you on track with your deadlines and reviews
- Helps you build and adjust your individual study plan

Career support

Built for engineers with existing experience: we reposition the career capital you already have and get you fully ready for the interview room. Your Career Coach joins after your first Final Review and stays through the program.

1

Positioning & materials

- → Career capital audit: resume and LinkedIn reviewed against your real experience
- → Target-role hypothesis and a personal transition plan
- → Gap analysis against live vacancies for your target role

2

Portfolio from defended work

- → Every defended project becomes a portfolio case study
- → Resume and LinkedIn repositioned for the target role — not rebuilt from scratch

3

Mock interviews & debriefs

- → Unlimited AI interview practice
- → Live technical and HR mocks with clearly stated limits
- → Debriefs of real interview rounds as you complete them

✓

To a signed offer

- → Final package: resume, LinkedIn, portfolio, target company list
- → A structured search pipeline instead of mass applications
- → Support continues past graduation: debriefs and strategy iteration on your search data

Why this beats self-paced courses

Tutorials and course libraries teach pieces. This program makes you prove the whole thing — under review, on a system that behaves like production.

Every project is defended live to a working engineer. Free courses never make you prove it — here the review is the unit of progress.

Weekly 1:1s with an instructor and an always-on project chat. Real feedback when you are stuck, not a comment section under a video.

One production-shaped system built across the whole program. Architecture decisions and trade-offs, not isolated exercises and snippets.

A career track with real people runs alongside your projects. Coach, mock interviews, real-round debriefs — tutorials end where this starts.

Frequently asked questions

The questions engineers actually ask us before enrolling — answered straight.

Is this live classes or self-paced?

Neither, and that's the point. You study on your own schedule — no mandatory class times — but this is not a self-paced course library: every week there's an optional 30-minute 1:1 with an instructor and live group sessions, and every project ends with a live review in front of a working engineer, every two weeks.

Are real humans teaching, or is it videos plus AI?

It's not a prerecorded video library. Theory is short lessons attached to each project; the teaching layer is human — instructors in 1:1s and your project chat, and every project is defended live to a working engineer.

How much instructor time do I actually get?

Every week you can book a 30-minute 1:1 with an instructor — bring your questions, code, or architecture. Your project chat with tutors is always on, and each project ends with a Readiness Review (15–30 min) plus a live Final Review.

How do project reviews work — live or async?

Live. When your project feels done, you book a Readiness Review; then you present and defend the project on the nearest review day — they run every two weeks. The panel grows with the project, up to a Tier-1 engineer on the capstone.

Will I study alone, or is there a real cohort?

You move in project cohorts, grouped by the project you're on. Each has a group chat where everyone is building the same thing, weekly live group sessions and events with instructors, a student directory to find peers, and random coffee pairings — working engineers around you, not a beginners room.

I already know parts of this stack. What's the point?

The program isn't a content library — it's a review bar. You make architecture decisions on production-shaped briefs and defend them live to engineers who do this work in production. Familiar tools mean you move faster: project hours are fixed, theory time flexes.

What if I can't give it 20 hours a week?

20 hours is the guideline pace, not a rule. Fewer hours means a longer timeline — week numbers in this syllabus are guides, not deadlines.

Do you guarantee me a job?

No — and no honest program can. What we commit to is the career track that runs alongside your projects: a personal coach, a portfolio built from defended work, mock interviews and real-round debriefs, and a structured search pipeline that continues past graduation.