POCKET GUIDE · MINI

₹49 EDITION

Become an AI Engineer — No CS Degree Needed

The honest 2026–27 roadmap in 20 minutes: the exact 7 stages, the math myth, and what the job really pays. Proof of work, not hype.

WEKNOWWHATYOURELOOKINGFOR.COM

Pocket Insight Guide

WHAT'S INSIDE

Zero to hired, mapped in pointers

AI Engineer is the fastest-growing job title in tech. This mini gives you the whole path — what the role is, the 7-stage skill map, the skills you use daily, the portfolio that gets you hired, and real salary data. Read it once and the fog clears.

THE ONE IDEA TO HOLD THE WHOLE WAY

You don't invent the models — you make them useful. AI Engineering is a **software engineering** job, not a research job. That's why you don't need a PhD or heavy math to start.

CHAPTER 01

What an AI Engineer actually does

- **The role:** builds software products powered by large language models. OpenAI/Anthropic/Google train the models; you do the last mile — connecting them to real data, tools and users.
- **A typical week:** wiring LLM APIs into features, building RAG pipelines, writing evals, cutting token cost/latency, and normal shipping — code review, deploys, monitoring.
- **Not a research job:** ~90% of AI roles *apply* existing models rather than train them.
- **Pick your lane:** AI Engineer (build apps on LLMs) ≠ ML Engineer (train/serve models) ≠ Data Scientist (analysis). Aim for the first.
- **The honest catch:** "fastest-growing" also means crowded at the entry gate — proof of work is how you stand out (Chapter 9).

DEFINITION

AI Engineer — a software engineer who designs, builds, evaluates and ships applications on top of foundation models, using prompting, RAG, agents, evals and cost/latency optimization.

Why 2026–27 is the moment

#1	~90%	3
Fastest-growing job title in tech	Of AI roles apply models, not train them	Deployed projects can beat a CS degree at screening

- **Demand outpaces supply** at the applied layer — companies need people who ship reliable LLM features, and few can.
- **The barrier is skills, not credentials.** Hiring managers repeatedly say deployed GenAI projects beat a degree at the first screen.
- **The window matters:** tools have standardised (APIs, RAG, MCP) so the path is learnable in months — before the field matures and gatekeeps.

KEEP THIS IN PROPORTION

Fast-growing ≠ easy money. The people who get hired are the ones who build and ship, not the ones who only watch tutorials.

CHAPTER 02

The roadmap — 7 stages at a glance

The whole journey, in the exact order the industry agrees on. Build something real at each stage — proof, not theory.

0	**Foundations**	Python · Git · APIs · light math
1	**LLM Core**	APIs · prompting · structured output
2	**Embeddings & RAG**	vector search · chunking · citations
3	**Evals**	golden sets · LLM-as-judge · tracing
4	**Agents & MCP**	tool use · workflows · MCP servers
5	**Fine-tuning**	LoRA/QLoRA · open models · when NOT to
6	**Portfolio & Hired**	ship 3 deployed projects · job hunt

THE FIVE RULES OF THE ROAD

Build in public · ship deployed demos, not notebooks · escape tutorial hell (40% learn / 60% build) · master fundamentals before frameworks · one domain beats ten features.

How long it really takes

STARTING POINT	REALISTIC TIMELINE TO JOB-READY
Complete beginner	**6–12 months**
Working developer	**3–6 months**

- **Consistency beats intensity.** Steady weekly building compounds far faster than weekend cramming.
- **Proof at every stage:** a CLI tool, then a RAG app, then an agent — each deployed and demoable.
- **Bookmark the map.** You'll return to it — every skill later is just composition of these stages.

THE 40/60 RULE

Spend 40% of your time learning and 60% building. Tutorial hell is where motivated beginners stall — the cure is shipping something imperfect.

CHAPTER 03

Foundations & the math myth

More people are scared off by imaginary math requirements than by any real obstacle. Here's the real list.

PEOPLE THINK THEY NEED	WHAT THE JOB NEEDS
Calculus, backprop	Nothing — you never differentiate at work
Linear algebra proofs	One idea: a dot product = similarity
Probability theory	Intuition: models pick next token by odds
A stats degree	Reading eval metrics (precision, recall)

- **Python checklist:** 4–6 weeks from zero — variables, functions, APIs, virtual envs, Git. That's enough to start.
- **Libraries do the math.** NumPy and the model API handle computation. Your job is knowing what the numbers mean.
- **First milestone:** ship an AI-powered CLI tool. Small, real, deployed.

CHAPTER 04

Core LLM skills — used every day

- **The only theory you need:** an LLM does one thing — given text so far, predict the next *token*. Chat, code, agents = that trick in a loop.
- **Tokens are the unit:** ~1,000 tokens ≈ 750 words. Everything is priced, limited and timed in tokens.
- **Context window & temperature:** how much it can "see" at once, and how random its choices are. Learn these without the myths.
- **Prompt engineering:** techniques that survive production — clear instructions, few-shot examples, format constraints.
- **Structured outputs & tool calling:** make the model return clean JSON and call functions. This is the foundation of agents.
- **The cost playbook:** caching, routing, smaller models — pay 10× less for the same result.

MASTER THIS CHAPTER FIRST

Everything after — RAG, agents, evals — is just composition of these core skills.

CHAPTER 05 · THE JOB-WINNER

RAG — chat with your data

- **Why it exists:** the model doesn't know your data — your wiki, docs, tickets were never in training.
- **What it is:** at question time, retrieve the most relevant passages from your own docs, ground the answer in them, and *cite* them.
- **Three wins:** fresh knowledge, fewer hallucinations, verifiable sources.
- **Don't fine-tune for knowledge — retrieve it.** Fine-tuning changes behaviour, not facts.
- **Chunking is the highest-leverage decision** you'll make. Then: vector DB, hybrid search, reranking.

WHY THIS LANDS THE JOB

Over 40% of AI-engineering interview take-homes are some flavour of "build a RAG system." Get dangerous at it and you're interview-ready.

CHAPTER 06

Agents & MCP

- **An agent is a loop:** the model decides → calls a tool → observes the result → repeats until done. ~50 lines of code, no framework needed.
- **Learn the workflow patterns first** (Anthropic's five) before reaching for multi-agent complexity.
- **Cap the loop:** multi-step reliability collapses fast — always bound the number of steps and add checks.
- **Context engineering** is the skill that replaced prompting — managing what the model sees at each step.
- **MCP = the "USB-C of AI":** a now-standard way to connect models to tools and data sources.

WHERE BEGINNERS BURN THEMSELVES

Agents are the fastest-growing area — and the easiest to over-engineer. Discipline (the loop, patterns, caps) beats hype every time.

CHAPTER 07 · JUNIOR VS SENIOR

Evals & production discipline

- **Prompt-by-vibes is the #1 beginner mistake.** Without evals, every prompt change is a coin flip — fix one case, silently break three.
- **The 3-layer eval stack:** golden datasets (20–50 real failures) → LLM-as-judge (cheap, at scale) → human review (calibrates the judge).
- **Observability:** trace everything; gate every release on the evals passing.
- **Cost & latency discipline:** caching, routing and smaller models routinely save 40–80%.
- **Security:** know the OWASP LLM Top 10 — prompt injection, data leakage, and friends.

EVALS ARE THE MOAT

Anyone can build a demo in a weekend. Evals + production discipline are what separate juniors from seniors — and what interviewers probe for.

CHAPTER 08

Fine-tuning — the last resort

- **The decision order:** prompt → RAG → fine-tune, in that order. Knowing when *not* to fine-tune is the senior signal.
- **Always prompt first:** few-shot + format constraints + a better model tier solve most problems for near-free.
- **RAG for knowledge:** fine-tuning does not reliably inject facts — retrieval does.
- **Fine-tune only for behaviour:** consistent style/format, structured-output reliability, or cutting cost/latency on one narrow high-volume task.
- **It's absurdly cheap now:** LoRA/QLoRA can fine-tune on one consumer GPU — a small run can cost around $10.

THE OPEN-WEIGHTS WORLD

Llama, Qwen, DeepSeek, Mistral — capable open models you can run and adapt yourself. Great for the right narrow job.

CHAPTER 09 · PROOF OF WORK

The portfolio that gets you hired

78%	~10×	<2min
Of portfolios are generic tutorials — skipped	A live demo beats a bare GitHub link	Recruiter time per repo, first screen

- **The 30-second README test:** what it does, why it matters, how to run it — in the first three sentences, demo link on line one.
- **Add an architecture diagram:** data in → retrieval/agent loop → model → output. Proves you designed a system.
- **A "Technical Decisions" section:** why this chunk size, model, vector store — the part interviewers quote back at you.
- **Delete the anti-portfolio:** MNIST, Titanic, iris. Domain specificity beats feature count.

CHAPTER 10

Salaries & the job hunt

US LEVEL (2025–26)	TYPICAL TOTAL COMP
Entry (0–2 yrs)	**$145–170K**
Mid (2–5 yrs)	**$150–210K**
Senior (5–10 yrs)	**$200–310K**
Staff / Principal	**$300–500K+**

- **The interview loop** (decoded from 1,765 job descriptions): a take-home RAG/agent build, a system-design round, and behavioural.
- **The canonical take-home** is a RAG system — which you've already built by following the roadmap.
- **Resume mechanics:** ATS-friendly, lead with shipped projects and metrics, not buzzwords.

THE THROUGH-LINE

Build → deploy → show. Three real projects, made legible, beat a wall of certificates.

YOU'VE GOT THE MAP. NOW WALK IT.

This was the 20-minute version. The full roadmap is the whole journey.

You now know the path most people never see clearly. The complete **AI Engineer Roadmap 2026–27** gives you every stage with the tools, the learning stack, the build briefs and the code-level detail.

- **All 10 chapters** — every stage, illustrated and expanded.
- **The full 7-stage skill map** with what to build and the free learning stack per stage.
- **The 6 portfolio projects** ranked by hiring signal per hour.
- **Real interview questions** with strong answers, and the full salary data across 3 continents.

UPGRADE TO THE FULL EDITION

~~₹199~~ **₹99**

Complete illustrated roadmap · instant download

SCAN / TAP
to upgrade

GO FURTHER, PAY LESS

Two ways to keep going

THE AI & TECH STACK BUNDLE

AI Engineer + Prompt Engineering + Python + No-Code

Everything to break into tech with AI, in one download.

~~₹396~~ **₹199**

THE COMPLETE LIBRARY

All 30 Skill Series books

Tech, money, career, spirituality, health — the entire collection, one price.

₹499

GET EVERY NEW BOOK FREE

Follow **@weknowwhatyourelookingfor** and join our WhatsApp list — buyers get first access and reader-only pricing on everything we publish next.

weknowwhatyourelookingfor.com

NOTICE

This guide is for **general educational purposes only**. Individual results vary with effort and circumstances, and no specific outcome is warranted.

Published by weknowwhatyourelookingfor.com · For queries, corrections, or licensing, visit weknowwhatyourelookingfor.com.