LEARN AI · FREE GUIDE 03

The RAG Blueprint

How to make AI answer from YOUR documents — SOPs, contracts, policies — instead of confidently making things up. Architecture, pitfalls, and a 2-week rollout plan.

RAG in one page

- Ingest — split documents into chunks (300–800 tokens works for SOPs)
- Embed — convert chunks to vectors so 'meaning' becomes searchable
- Retrieve — for each question, fetch the top 3–8 relevant chunks
- Generate — the LLM answers USING those chunks, citing its sources
- Result: answers grounded in your documents, with receipts

Where RAG pays for itself first

- Warehouse SOP assistant — 'What's the putaway rule for cold chain?'
- HR/policy bot — leave, reimbursement, shift-allowance questions
- Contract intelligence — 'Which transporter contracts have fuel clauses?'
- Customer support co-pilot — grounded macros instead of guesswork
- Onboarding buddy — new hires stop interrupting your best people

Industry example — A dark-store network put 214 SOP pages behind a RAG bot; new-picker ramp-up questions to supervisors dropped visibly within a month.

The 7 classic RAG failures

- Chunks split mid-table, so answers cite half a rule
- Stale index — the SOP changed, the bot didn't
- No metadata filters — Chennai rules answered with Delhi documents
- Retrieval top-k too small for multi-part questions
- No 'I don't know' path — the bot improvises instead of admitting gaps
- PDFs that are scans — OCR first or retrieval sees nothing
- No feedback button — you never learn which answers were wrong

A sane 2-week pilot

- Days 1–2: pick 30–60 high-value pages, clean them, add owners
- Days 3–5: build ingest + retrieval, wire up citations
- Days 6–8: 20 test questions from real users, score groundedness
- Days 9–12: fix chunking/filters, add the 'not in my documents' reply
- Days 13–14: soft launch to one team with a feedback channel

Want this implemented, not just downloaded?
Book a free 30-min discovery call → consult@hariadvisory.com · +91 98801 27460