Day 1 — RAG Mental Model (From First Principles)

This chapter contains the original README.md content (verbatim), followed by commentary.

RAG exists because LLMs do not know your private or proprietary documents.
RAG is a pipeline of transformations: Text → Meaning → Retrieval → Context → Answer.

An embedding is a numeric representation of meaning in a high-dimensional space.
Similarity is geometric, not logical.
Data quality dominates system performance.
Fine-tuning is a polishing step, not a foundation.

RAG failures are upstream failures:
poor chunking, boilerplate, and mixed semantics.

Mental Lock-In

If someone asked: “Why does RAG fail?”
The correct answer is:
Because meaning was distorted before retrieval ever happened.

Day 2 — From Document to Chunks

Today focuses on transforming raw documents into clean, meaningful chunks.

A chunk is the smallest unit of text that still represents a complete idea.

We intentionally avoid embeddings and retrieval here.
If chunking is wrong, everything downstream fails.

Key Insight

Chunking is not a preprocessing step — it is a semantic decision.
Chunks define what meaning even exists to retrieve.

Day 3 — Chunks → Embeddings → Retrieval

The goal of Day 3 is to build and fully understand the retrieval layer.

Retrieval is geometry, not reasoning.
Embeddings encode meaning as vectors.
Similarity search retrieves relevant context — not answers.

Core Pipeline

```
Chunk text
  ↓
Embedding (numeric meaning)
  ↓
Vector store
  ↓
Query embedding
  ↓
Similarity search
  ↓
Ranked chunks
```

Canonical Code Snippets

```
model = SentenceTransformer("all-MiniLM-L6-v2")
query_embedding = model.encode(query)
results = collection.query(query_embeddings=[query_embedding])
```

Day 4 — Retrieval → Context (Judgment Layer)

Day 4 introduces the judgment layer.

Raw retrieval output is noisy and unsafe.
Day 4 decides admissible evidence, enforces budgets, and produces ContextPack.

The ContextPack is the ONLY object allowed to reach an LLM.

ContextPack Boundary

```
@dataclass
class ContextPack:
    query: str
    policy: ContextPolicy
    approved_chunks: list
    dropped_chunks: list
    is_valid: bool
    invalid_reason: Optional[str]
```

One Sentence to Remember

Most RAG hallucinations are not model failures — they are context judgment failures.

Day 5 — Context → Answer (Constrained LLM)

The LLM is treated as stateless, untrusted, and purely generative.

All correctness guarantees are enforced outside the model.
Invalid answers are discarded and replaced with refusals.

Answer Contract

```
@dataclass
class Answer:
    text: str
    citations: list[str]
    refusal_reason: Optional[str]
```

Invariant

If an answer is not cited, it is not trusted.

Day 6 — Semantic Validation (Claim ↔ Context Entailment)

Every factual claim in an answer must be fully supported by approved context.

The LLM is used only as a semantic verifier.
The system fails closed if any claim is not entailed.

Verification Flow

```
extract_claims(answer)
for each claim:
    if not entailed(claim, context):
        FAIL
PASS only if all claims are ENTAILED
```

Day 7 — Claim ↔ Citation Alignment (Derived)

This chapter is derived from implementation and tests.

Even entailed claims must align with citations.
Claims and citations must agree positionally and semantically.

Alignment Status

```
class AlignmentStatus(Enum):
    ALIGNED
    MISSING_CITATION
    INVALID_CITATION
    EXTRANEOUS_CITATION
```

Day 8 — Presentation & Exposure Control

Day 8 decides whether an answer is shown and how it is shown,
without modifying content or citations.

Refusal (Day 5) and presentation suppression (Day 8) are mutually exclusive.

Final Response Model

```
FinalAnswerResponse(
    allowed: bool,
    mode: NORMAL | WARNING | SUPPRESSED,
    answer_text,
    citations,
    refusal_reason,
    presentation_reason
)
```