Setup

We've been collecting a bunch of *prompts and completions* that our models have run for different brands — e.g. marketing copy generation, tone of voice drafts, slogan ideas, etc.

E.g. On behalf of Nike

Prompt	What are the best running shoes?
Completion	Nike's latest running shoes, like the Air Zoom Pegasus 40, are designed for durability and comfort. They're great for daily training and available in multiple colorways.

Now we want to build a simple RAG system that, given a query like:

> show me prompts in the "sport" category

Returns the most relevant previous prompts or ideas that match that context.

Context

- Each prompt averages ~20 tokens (~20 words)
- Each completion averages ~100 tokens (~1 paragraph)
- Around 10 million prompt–completion pairs per day across all brands
- Dataset grows by ~100 000 new entries daily
- Data is multilingual (~15 % non-English)
- Must support low-latency semantic search and metadata filtering (e.g., brand, category, tone, model, language)

Questions:

1) How would you architect this system?

Walk through:

- The **data ingestion** and **indexing pipeline**
- **Embedding strategies**
- **Storage and retrieval design** (vector DB + metadata store or hybrid)
- How you'd **scale** to tens of millions of records with new data arriving continuously

2) How would you evaluate your system?

- How would you **evaluate** the quality of retrieval and generation?
- How would you detect when relevance or grounding quality drops?
- What metrics or dashboards would you build to monitor cost, latency, and freshness?