

# Gemini RAG Pipeline

*A Focused, Readable & Reproducible Retrieval-Augmented  
Generation System*

*Team: SWAT\_Genies*

*Members:*

- *Manish Chatla*
- *Siddharth Jain*
- *Pradyut Parida*
- *Pardheev Krishna Tammineni*

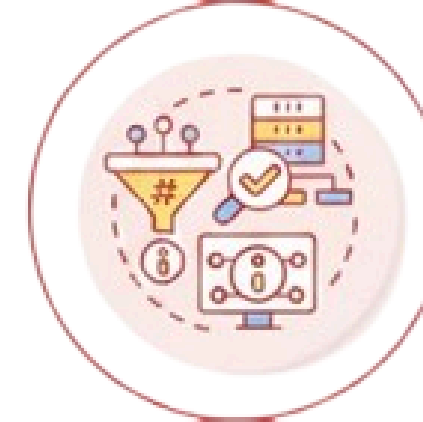


# What is RAG?

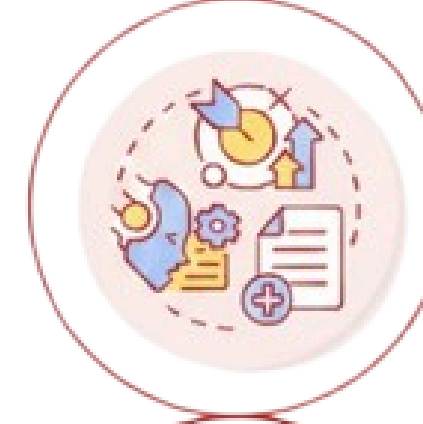
- Transform raw content into a structured catalog. Segment information into coherent units, capture descriptive signals (topic, entities, timestamps), and persist them in a form optimized for fast, high-quality lookup.
- Interpret the request and locate the most pertinent units from the catalog. Apply relevance scoring, lightweight filtering (e.g., scope, recency), and ordering so only the strongest evidence is returned.
- Curate the retrieved material for consumption: deduplicate, sanitize, and compress while preserving attribution. Assemble a concise, well-framed context that aligns with the task and stays within operational limits.
- Produce the final output using the curated context and explicit instructions. The response prioritizes factual grounding, clarity, and required structure (e.g., citations or schema), minimizing speculation and drift.



Indexing



Retrieval



Augmentation



Generation

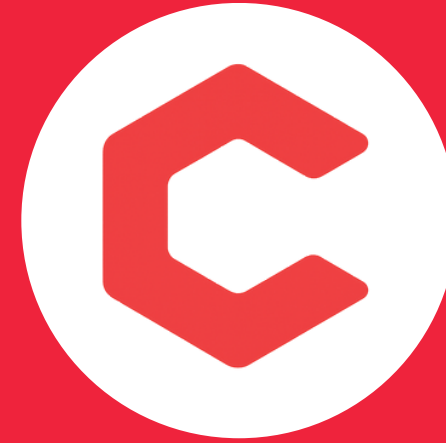
# Project Overview

*A Modular and Production grade RAG System Powered by:*



## Google Gemini

Google Gemini produces dense embeddings and grounded answers, with query expansion, batching, caching, and deterministic runs to reduce hallucinations.



## ChromaDB

ChromaDB is a fast, persistent vector store with HNSW retrieval, storing vectors and metadata for hybrid, filterable search.



## Streamlit

Streamlit UI lets you query, tweak settings, inspect retrieved context and scores, and export reports with metrics.

# Workflow Overview

- **Ingestion pipeline**

- Input Documents (JSON)
- Chunking & Preprocessing
- Gemini Embeddings
- Store in ChromaDB (Vector + Metadata)

- **Query pipeline**

- User Query
- Query Embedding

- **Retrieval paths**

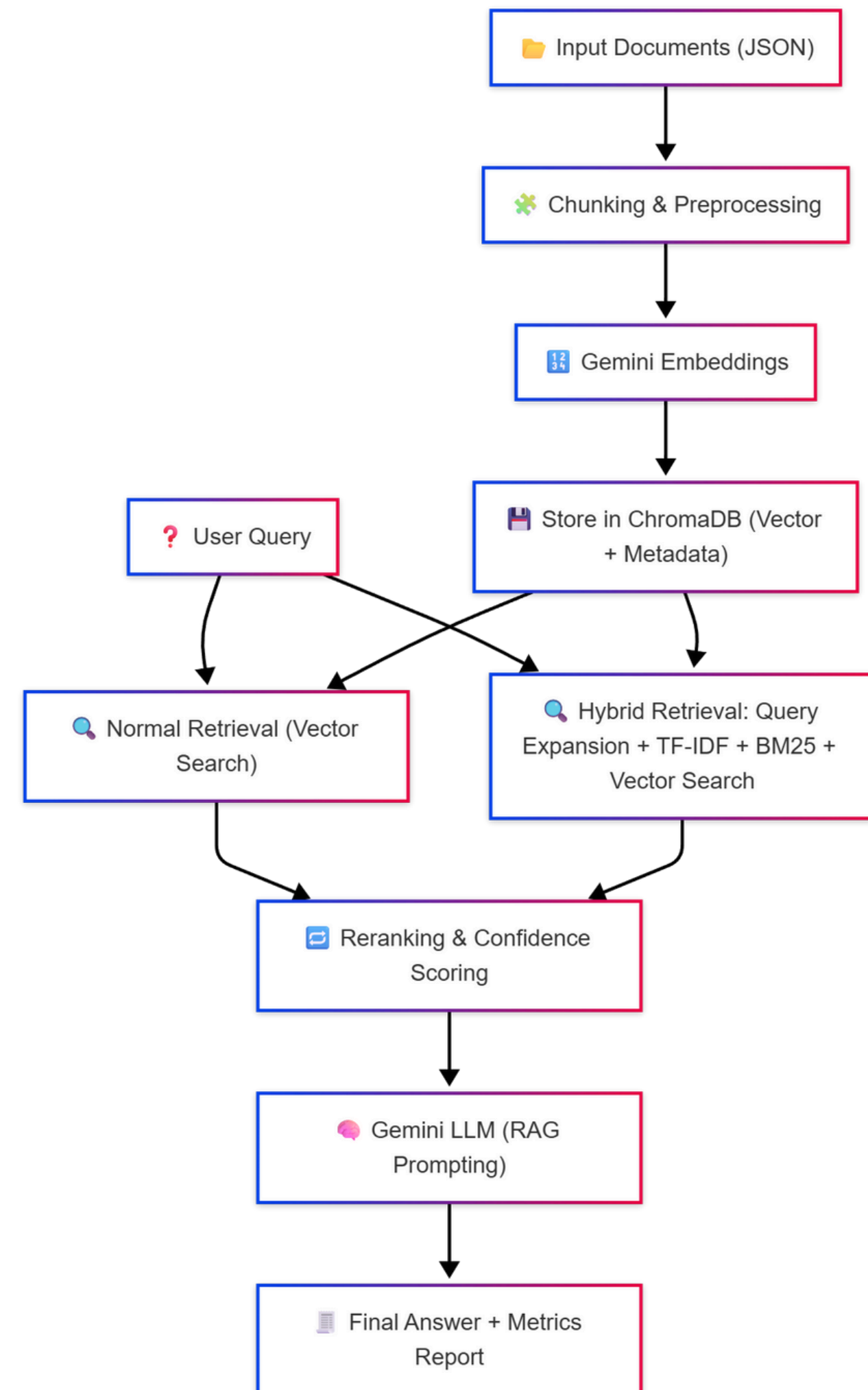
- Normal Retrieval (Vector Search)
- Hybrid Retrieval: Query Expansion + TF-IDF + BM25 + Vector Search

- **Ranking & reasoning**

- Reranking & Confidence Scoring
- Gemini LLM (RAG Prompting)

- **Output**

- Final Answer + Metrics Report





## Step 1 – Chunking & Context Preservation

We used a hybrid fixed-size + overlap strategy for better context preservation.

Sentence boundary detection

Token-based windowing (e.g., 512–1024 tokens)

Overlap context preservation (typically 10–20%)

## Step 2 – Embedding Logic

Gemini's embedding model (`models/embedding-001`) creates high-dimensional vector representations of text.

These embeddings capture the semantic meaning of the content for better retrieval accuracy

Deterministic behavior by seed control.

Batching for API efficiency and Metadata binding for traceability





## Step 3 — Vector Storage (ChromaDB)

ChromaDB provides persistent, local storage with **HNSW** indexing for fast approximate nearest-neighbor search. It co-stores vectors and metadata, enabling explainable retrieval, efficient filtering by attributes, and reproducible runs ideal for experimentation and production hardening.

### Hybrid Retrieval Strategy

- Query Expansion – For reformulating user queries to enhance recall
- TF-IDF (for fast keyword relevance scoring) + BM25 (for improved term weighting and ranking) - Used for lexical matching to capture exact term overlaps and key word relevance
- ChromaDB (Vector-based) – For semantic retrieval using cosine similarity
- This hybrid fusion ensures that both exact keyword matches and semantic meaning contribute to retrieval ranking, leading to richer and accurate context sets

### Reranking Strategy

- LLM-based Reranker – Uses Gemini to analyze contextual match with the query.
- Cohere Reranker – Leverages Cohere's rerank-english-v2.0 model for relevance scoring.
- CrossEncoder Reranker – Uses transformer-based pair scoring (query, document) similarity.



## Step 4 – Confidence Scoring

Retrieval Confidence:  $\text{avg}(\text{top-K cosine scores}) / 1.0$

Answer Confidence: Combines retrieval, coherence, and diversity

Also tracks: Retrieval latency, Tokens used and Answer length

## Step 5 – Explainability & Reporting

Context used for the final answer

Confidence breakdown

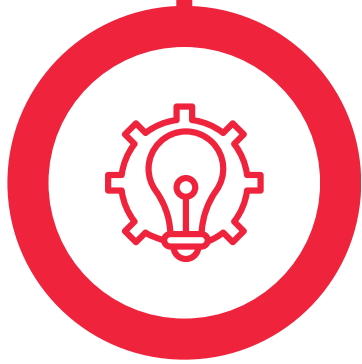
Top retrieved passages

Latency & token stats

JSON report for auditing



# Trade-offs & Optimization



## Accuracy

Improved by multi-stage reranking and overlap chunking.

## Speed

Achieved through batch embedding, caching, and HNSW indexing.



## Cost

Minimized via checkpointing and caching embeddings locally.





















# Experimental Rigger and Impact

COMPONENT	ALTERNATIVES TESTED	CHOSEN STRATEGY	IMPACT
Chunking	Fixed	Overlap 20%	Improved contextual continuity
Embeddings	E5, Gemini	Gemini	Best semantic match for factual text
Vector Store	FAISS	Chroma	Lightweight, persistent, easy integration
Retrieval	Pure vector	Hybrid (TF-IDF + BM25 + Semantic)	Higher recall & precision
Reranking	None	Cohere, CrossEncoder, LLM Based	Boosted contextual accuracy by ~12%

## Testing & Engineering Discipline

-  Unit tests for all major modules (tests/)
-  Structured logging for debugging and observability
-  Checkpointing and error handling for long runs
-  Deterministic results (seeded randomness)
-  Secure API key loading via .env

## Future Enhancements

-  Dynamic chunking based on sentence or semantic boundaries
-  FAISS or HNSW backend for large-scale corpora
-  Automated threshold tuning for confidence calibration
-  Multi-lingual embeddings for cross-language KB search
-  Context-aware query expansion using user interaction history
-  Explainable retrieval with highlighted matched terms or sections
-  Adaptive top-K selection based on query complexity
-  Retrieval analytics dashboard for precision, recall, and coverage
-  Integration with multiple LLM providers for fallback and ensemble reasoning
-  Secure handling of sensitive data in embeddings and metadata
-  Cloud-based scaling for distributed retrieval and embedding pipelines

**Thank  
You**



**SWAT\_  
Genies**