# Phase 3 – Retrieval-Augmented Generation (RAG) Enhancements

## ✅ Overview

Phase 3 introduced a full RAG pipeline with ingestion, embeddings, retrieval, prompt engineering, history tracking, and analytics. Each sub-phase incrementally added features to make the system more robust, debuggable, and measurable.

## 🔹 Phase 3.1 – Ingestion Layer

* Upload + chunking of documents.
* SQL storage of documents + metadata.

## 🔹 Phase 3.2 – Embedding + Vector DB

* Added embeddings using OpenAI.
* Stored vectors in Postgres with pgvector.
* Query-time semantic search.

## 🔹 Phase 3.3 – Retrieval Service

* Added semantic, keyword, and hybrid retrieval.
* Debug UI for retrieval comparisons.

## 🔹 Phase 3.4 – Prompt Engineering

* Support for Zero-Shot, Few-Shot, Role, RAG, and Hybrid prompts.
* Configurable prompt templates in Admin UI.

## 🔹 Phase 3.5 – RAG History + Docs Module

* Save all RAG runs into history (latencies, prompts, answers).
* History filters (date, SLA, provider, prompt style).
* Compare runs side-by-side.
* Export comparison/history.
* Added Docs module for documentation storage.

## 🔹 Phase 3.6 – Analytics Dashboard

* SLA compliance and latency tracking.
* Analytics Dashboard with charts:
  + SLA compliance by style.
  + Avg latency trends.
  + Usage stats.
* Drill-down from analytics → history.

## 🔹 Phase 3.7 – Advanced Analytics ✅

* **Multi-provider support** (OpenAI, Gemini, Claude).
* **SLA compliance tracking** (per provider/model + per prompt style).
* **Token tracking**: QueryTokens, ContextTokens, TotalTokens.
* Enhanced Analytics Service with provider/model aggregation.
* UI Updates:
  + Extended Analytics Dashboard with 6+ charts.
  + KPI widgets (Total Runs, Avg Latency, SLA Compliance).
  + Provider/model breakdowns.
* Database migration for token fields.
* Phase 3.7 completed.

## 📂 Project Structure (till Phase 3.7)

/AirNir  
├── Library  
│ ├── ArNir.Core → Entities, DTOs, Config, Validations, Utils (Tokenizer)  
│ │ ├── DTOs/Analytics → AvgLatencyDto, SlaComplianceDto, ProviderAnalyticsDto, TrendDto, PromptStyleUsageDto  
│ │ └── Utils → TokenizerUtil.cs  
│ ├── ArNir.Data → DbContexts (SQL Server + Postgres), EF Core migrations  
│ └── ArNir.Services → Business logic (EmbeddingService, RetrievalService, RagService, RagHistoryService, Analytics)  
│  
├── Presentation  
│ ├── ArNir.Admin → AdminLTE UI (ASP.NET Core MVC)  
│ │ ├── Controllers → Docs, Retrieval, RAG Comparison, RAG History, Analytics  
│ │ ├── Views → Razor views for RAG, History, Analytics Dashboard  
│ │ ├── wwwroot/js → rag-comparison.js, rag-history.js, analytics.js  
│ │ └── Layout → Sidebar includes Analytics module  
│ └── ArNir.Frontend → Planned end-user chat/search UI (Phase 3.8+)  
│  
├── sql  
│ ├── create\_tables.sql  
│ ├── update\_documents\_chunks.sql  
│ ├── update\_embeddings.sql  
│ ├── update\_rag\_history.sql (adds token counts)  
│ └── migrations for Phase 3.7 token tracking  
│  
└── docs  
 ├── Phase3\_RAG\_Architecture.png  
 ├── Phase3.3\_Architecture.png  
 ├── Phase3.4\_Architecture.png  
 ├── Phase3.5\_Architecture.png  
 ├── Phase3.6\_Architecture.png  
 ├── Phase3.7\_Architecture.png  
 ├── Phase3.7\_Documentation.md  
 └── GenerativeAI\_KnowledgeBase.md