# 📘 Phase 3.5 – RAG History Enhancements & Prompt Engineering

This document provides the complete **functional, technical, and testing documentation** for Phase 3.5 of the Generative AI project.

## 🔹 1. Overview

Phase 3.5 extends the RAG (Retrieval-Augmented Generation) comparison system with: - **Prompt Engineering support** (Zero-Shot, Few-Shot, Role, RAG, Hybrid). - **PromptStyle persistence** in history for analytics. - **RAG History Page Enhancements**: - Advanced filtering (SLA, Date, Query, PromptStyle). - Detailed modal view with expandable retrieved chunks. - Compare Mode: side-by-side comparison of multiple runs. - Export options (CSV/Excel, with Chunks included). - **UI fixes for Bootstrap 5** (Run Comparison button, spinners, toggling).

## 🔹 2. Features Implemented

### 2.1 Prompt Engineering Styles

Supported prompt styles: - **Zero-Shot** → Direct query answering. - **Few-Shot** → Uses examples to guide the model. - **Role Prompting** → Model answers as a domain expert. - **RAG-Augmented** → Uses retrieved chunks for context. - **Hybrid (Role + RAG)** → Combines role expertise with retrieved context.

### 2.2 Database Changes

* New column added in RagComparisonHistories:
* ALTER TABLE RagComparisonHistories  
  ADD PromptStyle nvarchar(50) NOT NULL DEFAULT 'rag';
* Logs the PromptStyle used in each comparison.

### 2.3 RAG Comparison Page

* Query input + PromptStyle dropdown.
* Bootstrap 5 compliant button (type="button").
* Results section:
  + Baseline vs RAG answers.
  + Retrieved Chunks (expand/collapse previews).
  + SLA metrics (Retrieval, LLM, Total).
* SLA badge (✅ OK / ⚠️ Slow).

### 2.4 RAG History Page

* Filters: SLA, date range, query search, PromptStyle.
* Details Modal:
  + Query, PromptStyle, Baseline, RAG answers.
  + Retrieved Chunks with expand/collapse.
  + Latency breakdown.
* Compare Mode:
  + Toggle ON → select 2+ runs.
  + Side-by-side modal comparison.
  + Each column shows query, answers, chunks, latencies.
* Export Options:
  + **Comparison Modal:** Export selected runs (CSV/Excel).
  + **History Table:** Export all runs with filters applied (CSV/Excel).
  + Includes Chunks JSON for debugging.

## 🔹 3. Technical Implementation

### 3.1 Backend

* **RagService.cs** updated:
  + Handles promptStyle parameter.
  + Builds prompts dynamically per style.
  + Persists PromptStyle in RagComparisonHistories.
* **RagComparisonController.cs** updated:
  + Accepts promptStyle and saveAsNew.
  + Returns enriched RagResultDto.

### 3.2 Frontend

* **RagComparison/Index.cshtml**:
  + Bootstrap 5 inputs and buttons.
  + Results section with Baseline, RAG answers, SLA metrics.
* **rag-comparison.js**:
  + AJAX call to /RagComparison/Run.
  + Spinner toggling with d-none.
  + Expand/Collapse retrieved chunks.
* **RagHistory/Index.cshtml**:
  + Filter UI.
  + Compare Mode toggle.
  + History DataTable.
  + Single Detail modal & Compare Modal.
* **rag-history.js**:
  + Filtering logic for DataTable.
  + Row click behavior (single vs compare mode).
  + Comparison renderer.
  + Export functions (CSV/Excel).

## 🔹 4. Testing Checklist

### 4.1 Query Execution

* Run queries with each PromptStyle.
* Verify answers differ accordingly.
* Check DB RagComparisonHistories → PromptStyle saved correctly.

### 4.2 History Page

* Apply SLA filter (OK/Slow).
* Filter by PromptStyle.
* Filter by date range and query search.
* Verify DataTable updates.

### 4.3 Details Modal

* Check query, answers, PromptStyle.
* Verify retrieved chunks expandable (View Full / Show Less).
* Latency values shown.

### 4.4 Compare Mode

* Enable Compare Mode.
* Select multiple runs with different PromptStyles.
* Side-by-side modal shows correct results.

### 4.5 Export Functions

* Export selected comparisons (CSV/Excel).
* Export all filtered history (CSV/Excel).
* Verify Chunks JSON included in exports.

### 4.6 Edge Cases

* Run query with no chunks retrieved → history saved.
* Run long queries → answers and exports work.
* Apply filters with no results → DataTable shows empty state.

## 🔹 5. Known Issues / Next Steps

* UI currently shows chunks as raw JSON in exports; may format better later.
* Future Phase (3.6) will add **analytics dashboards**:
  + SLA trends.
  + PromptStyle effectiveness (accuracy vs latency).
  + Visualization of retrieval quality.

# ✅ Phase 3.5 Completed

* Prompt Engineering integrated into RAG pipeline.
* History persistence extended with PromptStyle.
* Admin UI supports filtering, comparison, and exports.
* Database schema updated with PromptStyle.

👉 Ready for **Phase 3.6 – Analytics Kickoff**.