RAG Design and Implementation Manual

Real Estate Investment Analyzer

1. Introduction

This document serves as the official technical manual for implementing Retrieval-Augmented Generation (RAG) in the Real Estate Investment Analyzer project.

It is intended for developers working on backend systems, data pipelines, and AI integration. The focus is on **correct system design**, **financial safety**, and **clear separation of responsibilities** between deterministic computation and language-based explanation.

This manual should be treated as a reference specification during development and review.

2. Fundamental Design Principle

All financial calculations in this project are deterministic and must be performed by the backend system.

These include, but are not limited to: - EMI and amortization schedules - Tax benefit calculations - Rental yield, cap rate, and cash flow - Rent versus buy simulation - Final BUY or RENT decision

The RAG layer must not perform or alter any financial computation.

The sole purpose of RAG is to retrieve verified backend outputs and present them in a structured, human-readable form.

3. Role of RAG in This System

RAG acts as an **interpretation and communication layer**.

It enables the system to: - Retrieve relevant analyzed properties - Explain why a decision was made - Compare multiple properties - Answer user questions using verified data

RAG does not introduce new data, assumptions, or calculations.

4. Explicit Non-Responsibilities of RAG

The following actions are not permitted within the RAG layer:

- Computing EMI, ROI, tax benefits, or appreciation
- Deciding whether a property is BUY or RENT
- Performing numeric comparisons between properties
- Operating directly on raw CSV or database tables

All such logic must reside in the backend analysis engine.

---

5. System Architecture Overview

The system follows a strict SQL-first, explanation-second architecture.

User Query → Intent Classification → Structured Database Query → Retrieved Property Records → Property Explanation Generation → Language Model Response

This ordering is mandatory to ensure correctness and reproducibility.

---

6. Preparing Data for RAG Consumption

The analytical dataset contains numeric and categorical fields that are not suitable for direct embedding.

Each analyzed property must therefore be converted into a descriptive text document known as a **Property Explanation Record**.

Property Explanation Record Structure

Each record must contain: - Property type and location - Price, rent, and area - Monthly EMI and effective monthly cost - Tax savings summary - Final wealth comparison (buy vs rent) - Final decision - A concise rationale derived from backend logic

Only these explanation records are eligible for embedding.

---

7. Embedding Strategy

The embedding layer is used strictly for semantic retrieval of explanation content.

Permitted embedding sources: - Property explanation records - Tax rule documentation - Decision logic documentation - Simulation assumption notes

Prohibited embedding sources: - Raw CSV files - Numeric tables - Amortization schedules - Intermediate calculation outputs

8. Intent Classification Requirement

Every user query must be classified before data retrieval.

The system must support at least the following intent categories:

- FILTER: requests for listings or subsets of properties
- EXPLAIN: requests for reasoning behind a decision
- COMPARE: requests to evaluate two or more properties
- EDUCATIONAL: requests for explanations of tax or financial logic

Intent classification determines the execution path and prevents unnecessary or unsafe LLM usage.

9. Structured Retrieval via SQL

All property selection and ranking must be performed using SQL or equivalent deterministic queries.

The language model must never be responsible for filtering or ranking properties.

Example operations include: - Filtering by location, budget, or bedroom count - Sorting by wealth difference or yield - Limiting result size

Only the final selected rows may be passed forward for explanation.

10. RAG Response Generation

After retrieval, property records are transformed into explanation text and supplied to the language model.

The language model is permitted to: - Summarize - Rephrase - Compare - Present rationale

The model must not introduce new numeric values or modify existing ones.

11. Example Interaction Flow

User Request: Show undervalued 3 BHK properties in New Town

System Execution: 1. Intent classified as FILTER 2. SQL query executed 3. Top results retrieved 4. Explanation records generated 5. Language model produces a structured summary

12. Technology Stack

The following stack is recommended for consistency and maintainability:

- Database: PostgreSQL
- Backend API: FastAPI
- Embeddings: OpenAI, Gemini, or equivalent
- Vector Store: FAISS, pgvector, or Chroma
- Visualization: matplotlib

13. Common Implementation Errors

The following errors must be avoided:

- Allowing the language model to compute financial values
- Embedding raw analytical tables
- Skipping intent classification
- Using the language model for filtering or ranking
- Mixing explanation logic with computation logic

14. Developer Verification Checklist

Before integration, ensure that:

- All calculations are completed before RAG invocation
- BUY or RENT decisions are stored in the database
- Explanation records are generated from verified outputs
- Intent classification is enforced
- SQL-first retrieval is implemented

15. Final Design Statement

The Real Estate Investment Analyzer is a deterministic financial system augmented by a language interface.

The backend determines outcomes. The RAG layer communicates those outcomes.

This separation must be preserved throughout development.

16. Implementation Expectations

All contributors are expected to follow this manual strictly.

Any deviation that allows the language model to influence financial outcomes is considered a design violation and must be corrected during review.