# README: RAG Evaluation Project

## Purpose

Evaluate and compare the accuracy of responses generated from enterprise data using three approaches: 1. Power BI Inbuilt Q&A 2. Power BI Copilot (Fabric Enabled) 3. Agentic RAG Solution built using Azure OpenAI and LangGraph

## Architecture Overview

### Data Layer

* **Azure SQL Database** hosts fact and dimension tables.
* Synthetic dynamic data generated for two years.
* SQL Views created for analytics and Power BI semantic modeling.

### Power BI Layer

* Build semantic models with relationships.
* Create Power BI dashboards.
* Test with both **Q&A** and **Copilot**.

### RAG Layer

* **Frameworks:** LangChain / LangGraph
* **Models:** OpenAI LLM, Embedding, Moderation
* **Vector Store:** Azure AI Search (Standard SKU)
* **Backend:** FastAPI
* **UI:** Chainlit
* **Hosting:** Azure Container Apps
* **CI/CD:** GitHub Actions
* **Agent Hosting:** LangGraph Cloud

## Implementation Stages

### Stage 1: Power BI Q&A Evaluation

* Use inbuilt Q&A for natural language queries.
* Evaluate accuracy of returned results.

### Stage 2: Power BI Copilot Evaluation

* Enable Copilot (requires Microsoft Fabric).
* Compare Copilot results with Q&A.

### Stage 3: Agentic RAG Solution

* Build RAG pipeline with Azure SQL + Azure AI Search.
* Compare RAG responses against Power BI outputs.

## Functional Requirements

| ID | Requirement | Description |
| --- | --- | --- |
| FR-1 | Data Preparation | Generate synthetic data and create SQL views. |
| FR-2 | Semantic Modeling | Build Power BI semantic models. |
| FR-3 | Q&A Testing | Evaluate Q&A accuracy. |
| FR-4 | Copilot Testing | Compare with Q&A results. |
| FR-5 | RAG Query Handling | Retrieve data through Azure AI Search. |
| FR-6 | Comparison Analysis | Compare all three systems for accuracy and context. |
| FR-7 | Auto Deployment | CI/CD through GitHub Actions. |
| FR-8 | Session Memory | Short-term session memory only. |

## Non-Functional Requirements

* **Scalability:** Auto-scale via Azure Container Apps.
* **Performance:** <3s for cached responses; <8s for new queries.
* **Availability:** 99% uptime (test environment).
* **Security:** No authentication (restricted test setup).
* **Maintainability:** Source control via GitHub.
* **Reproducibility:** Consistent test results.

## Constraints

* Only short-term memory.
* No authentication or profiles.
* No long-term memory.
* Read-only query access.

## Evaluation Metrics

| Metric | Description |
| --- | --- |
| Accuracy | Query intent match rate. |
| Relevance | Context alignment. |
| Latency | Response generation time. |
| Interpretability | Transparency of reasoning. |
| Consistency | Result stability. |

## Technology Stack

| Component | Technology |
| --- | --- |
| Data | Azure SQL Database |
| Modeling | Power BI |
| LLM Framework | LangChain / LangGraph |
| Model | OpenAI GPT |
| Embeddings | OpenAI Embeddings |
| Vector Store | Azure AI Search |
| Backend | FastAPI |
| UI | Chainlit |
| Hosting | Azure Container Apps |
| CI/CD | GitHub Actions |
| Agent Hosting | LangGraph Cloud |

## Deliverables

1. SQL views and synthetic data generation scripts.
2. Power BI semantic model and dashboards.
3. Comparative accuracy analysis (Q&A, Copilot, RAG).
4. Fully deployed RAG system (FastAPI + Chainlit).
5. Evaluation summary report.

## Future Enhancements

* Add authentication and user profiles.
* Introduce long-term memory.
* Add observability with Azure Monitor.
* Extend to additional LLM models.