Project Name:

Intelligent Enterprise Knowledge Assistant with GraphRAG

Problem Statement:

Modern enterprises have **huge volumes of unstructured data** scattered across:

- PDFs, Word documents, reports
- Slack/Teams chat messages
- Internal wikis, spreadsheets
- Emails and meeting notes

Employees often struggle to:

- Quickly find relevant information
- Connect information across multiple documents
- Extract actionable insights or KPIs
- Answer complex questions like **"Which projects managed by Alice in Q4 exceeded revenue targets?"**

Traditional RAG (Retrieval-Augmented Generation) can retrieve answers from documents using vector similarity but **struggles with multi-hop reasoning—**connecting data points across different sources.

GraphRAG solves this by adding a **graph knowledge layer**, where relationships between entities can be explicitly represented and queried.

Project Goal:

Build an **AI assistant** that:

1. Ingests all enterprise documents and chats
2. Extracts **entities** (projects, KPIs, people, dates) and relationships
3. Builds a **knowledge graph** of the enterprise information

4. Supports **hybrid retrieval**:
    - Graph queries for relationships
    - Vector embeddings for unstructured document search
5. Feeds retrieved data into a **Large Language Model** (LLM)
6. Generates **structured, accurate answers** to complex, multi-hop questions
7. Provides summaries, insights, and action items via a **chat interface**

How It Works (Pipeline):

1. **Document Collection & Preprocessing**
    - Collect PDFs, Word docs, Slack/Teams messages, wiki pages
    - Extract text, clean it, and chunk it into smaller pieces
2. **Entity Extraction & Graph Construction**
    - Use NLP (NER, rule-based extraction) to find entities like projects, KPIs, people, and dates
    - Define relationships (edges) between entities, e.g., "Project → has KPI → Revenue"
    - Build the graph in Neo4j or TigerGraph
3. **Embeddings & Vector Store**
    - Generate embeddings for document chunks for semantic search
    - Store embeddings in a vector DB (e.g., Chroma)
4. **Hybrid Retrieval (Graph + Vector)**
    - Graph queries handle **multi-hop relationships**
    - Vector search handles **unstructured content retrieval**
    - Combine results to feed the LLM
5. **LLM Integration (GraphRAG)**
    - LLM uses graph + retrieved documents to generate answers
    - Answers can be **structured** (tables, summaries) or **natural language**

6. **Multi-hop Query Support**
    - Example: “Which projects managed by Alice exceeded revenue target in Q4 and had positive client feedback?”
    - LLM reasons over graph and document data
7. **Frontend / Chat Interface**
    - Users ask questions via a simple UI (Streamlit or React)
    - Receive **structured answers, summaries, and insights**

Why This Project is Strong:

1. **Enterprise Relevance:** Solves a real problem of scattered knowledge and slow information retrieval
2. **Technical Complexity:**
    - Combines **graph databases, vector embeddings, NLP, and LLMs**
    - Supports **multi-hop reasoning** and **hybrid RAG**
3. **Scalable & Extensible:**
    - Can add new document sources or multi-modal data (images, charts)
4. **Portfolio Impact:**
    - Shows end-to-end AI system design with **data engineering + knowledge reasoning**
5. **Demo Ready:**
    - Users can ask real questions and get **contextually accurate, structured answers**

Example Query & Output:

User Query:
"Show me all Q4 projects managed by Alice where revenue exceeded $50K and client satisfaction was >90%"

GraphRAG Answer:

Project	Revenue	Client Satisfaction	Status	Notes
Project X	$75K	95%	Completed	Delivered early
Project Y	$60K	92%	Completed	Positive feedback

The LLM can also generate a **summary paragraph**:
"Alice managed 2 projects in Q4 that exceeded revenue targets and had high client satisfaction. Both projects were completed successfully and received positive client feedback."
