Use Case: Company Knowledge Assistant

Objective

Build a full-stack RAG-powered knowledge assistant where users log in and query a knowledge base tailored to them. The system should automatically fetch relevant information for the logged-in user without asking them to select a role.

Requirements

Backend

- **User Authentication**
  - **Allow user registration and login.**
  - **Each user should have a workspace or context automatically used in queries.**
- **Document Ingestion**
  - **Users should be able to upload documents.**
  - **The system should process, chunk, and embed these documents and store them for retrieval.**
- **RAG Pipeline**
  - **Build a Retrieval-Augmented Generation flow using LangChain.**
  - **Retrieval must be filtered by logged-in user's data so they only get answers relevant to their workspace.**
  - **Include source citations (document names/sections) in the response.**
- **Conversation Memory**
  - **Maintain chat context per user session so answers can reference previous messages.**

Frontend

- **Login / Register Page**
  - **Simple form for user authentication.**
- **Chat Interface**
  - **Single chat window for asking questions.**
  - **Display answers along with source documents.**
  - **Show previous conversation history in the session.**
- **Document Upload UI**

- **Allow users to upload new documents to their own knowledge base.**
- **(Optional) Let users create or switch between multiple workspaces.**

Optional Features (Bonus)

- **Multiple Workspaces – Allow users to manage and switch between separate knowledge bases.**
- **Document Versioning – Support re-uploading updated documents and re-indexing.**
- **No-Answer Fallback – If no relevant documents are found, respond with a graceful "I don't know" or similar.**
- **Prompt Templates per Workspace – Customize system prompts for different workspaces or domains.**

Evaluation (50 Marks Total)

Area	Description	Marks
Backend Core	Working ingestion, RAG retrieval, and generation	15
User Context Filtering	Answers are restricted to logged-in user's knowledge base	10
Conversation Memory	Context is maintained across user questions	7
Frontend Core	Login, chat interface, document upload	10
Database & Architecture Design	Efficient structure and clean integration	5
Code Quality	Clear documentation, clean structure, error handling	3

Bonus (Up to +5): Multiple workspaces, document versioning, fallback responses, or polished UI.