SYLVR
RISKS to RESOLVE to REWARDS

SYLVR | Assignment | NLP & Agentic AI Intern

Challenge: Build a **Conversational Database Agent** that can have conversation with user about data stored in a database by

1 Mapping natural language text into various query types (definitions, filters, aggregation, trends, comparisons, etc.)

2 Executing appropriate database queries on relevant collections to obtain relevant information, or else, flag gaps in data

3 Providing quality responses and handling follow up text to maintain conversation context across multiple exchanges

4 Extracting objective, actionable insights on database gaps and user's emotional tone, query intent for World Model

Key Requirements

1 **Database integration** – Use a real MongoDB database; no PDF/document querying

1.1 sample_analytics database (finance domain) from MongoDB's free Atlas cluster is sufficient for base requirements

1.2 For extra credit go beyond default database by using a more complex financial database meeting these requirements:

1.2.1 More collections, Financial numeric data fields, Nested fields, One or more high volume collections (>500 documents)

2 **Query execution** – Handle various question types and schema, metadata, vector embeddings/ similarity for relevance

3 **Conversation context** – Handle follow-up text referencing prior conversation parts and 'memorize' multiple exchanges

4 **Deliverables** – GitHub repo with working code, README.md with setup and brief architecture explanation, dataset used (upload and give link in README file) and short demo video demonstrating various user questions

4.1 **NOTE: If you can't complete Secondary and Bonus features fully in given time then** include a brief writeup in your README explaining how you would implement the remaining, and what further you would implement given more time. A thoughtful writeup about features you didn't have time to build will be duly considered.

What we are evaluating

PRIMARY (Must Have)	**SECONDARY (Nice to Have)**	**BONUS (Optional Extra Credit)**
Database connectivity and query execution on atleast 3 collections to simulate real-world query	Agent's Context use for conversation management with simple in-memory data structures or LangChain's memory modules	World Model insights dashboard
Natural language understanding and query translation	Error handling for 'ambiguous/ impossible' queries	Advanced agentic reasoning
Basic conversation memory and context handling	Multiple database operations in one conversation	Query optimization
Working solution in Colab/ Jupyter with end-to-end demo	UI using any simple framework like Gradio, Streamlit, React, etc.	Creative solutions to complex problems
Database - MongoDB	Clean, organized code architecture	Voice interface (STT/ TTS)
Programming language – Python (backend, logic & DB interactions)		Docker file to containerize your application for easy deployment

NOTE: We are primarily evaluating AI/backend functionality of Query, Response, Context quality with clean architecture & code, a working implementation and problem-solving approach with technical planning. Focus on making the Agent smart.