

Assignment Instructions

As part of the assessment process, please complete the following assignment and share the GitHub repository URL upon completion. Your submission must include the following:

1. A README.md file detailing the project setup, architecture, and usage instructions.
2. A demo video URL showcasing the functionality of your solution.

Please upload your completed project to GitHub and send us the repository link at your earliest convenience.

Background

John, a customer support executive, encounters difficulties in managing and retrieving critical customer information dispersed across multiple systems and policy documents. To enhance efficiency and improve response quality, your task is to build a Generative AI-powered Multi-Agent System that enables natural language interaction with both structured and unstructured data.

Assessment Objective

Design and develop a Generative AI Multi-Agent System that can:

- Query structured customer-related data using natural language.
- Process and organize unstructured documents into a searchable knowledge base.
- Deliver accurate, context-aware, and user-friendly responses to customer queries.

Feature Examples

- John uploads company policy PDF documents to the system.
- John asks the chatbot: For Ex. "What is the current refund policy?" The assistant responds with a context-aware summary from the uploaded document.
- John asks the chatbot: For Ex. "Give me a quick overview of customer Ema's profile and past support ticket details." The assistant retrieves and summarizes structured customer data from the SQL database.

Data

- Utilize any synthetic or dummy customer dataset (e.g., customer profile, past support ticket history, etc.) and store structured data in any SQL database.
- For policy PDFs, use any publicly available company policy documents.

Technology Stack

- Core Libraries/Frameworks: LangChain, LangGraph, etc.
- Models: Any LLM and embedding model.
- Databases: Any SQL DB (for structured data) + Vector DB (for unstructured data).
- Server: MCP Server.
- UI: Streamlit, Gradio, or any equivalent.