Please find the details of the assignment below:

Objective: To evaluate the candidate's ability to design a simple knowledge graph, query it, and utilize an LLM to generate natural language answers based on the retrieved information, along with a frontend.

Tasks:

1. **Focused Knowledge Domain & Schema Design:**
   ○ The candidate should select a very specific and limited domain relevant to potential customer support scenarios. Examples include:
      § Features of a single, specific software product.
      § Troubleshooting steps for one particular type of hardware issue.
      § Details of different tiers of a single subscription service.
   ○ They need to design a concise schema for a knowledge graph in this domain, clearly defining the key entity types and the relationships between them. They should document their chosen domain and schema.

2. **Concise Knowledge Graph Construction:**
   ○ Based on their designed schema, the candidate should populate the knowledge graph with a small but representative set of sample data (aim for 5-10 key entities and the relationships connecting them). They can use an in-memory graph structure (e.g., Python dictionaries) or a lightweight graph database like Neo4j. The focus should be on demonstrating the core concepts rather than building a large dataset.

3. **Knowledge Graph Querying & Information Retrieval:**
   ○ Implement a robust mechanism to answer user questions by querying their constructed knowledge graph. This should involve:
      § (Optional)Using an LLM (e.g., Llama 2, Mistral via Langchain or LlamaIndex) to understand the user's intent, identify key entities and relationships mentioned in the question, and formulate a query or traversal strategy for their knowledge graph.
      § The querying mechanism should be able to handle simple multi-hop relationships within their defined schema.

4. **LLM-based Natural Language Answer Generation:**
   ○ After successfully retrieving relevant information from the knowledge graph, the candidate must use an open-source LLM to generate a coherent and natural-sounding answer to the user's question, grounded in the retrieved knowledge. The LLM should synthesize the information from the graph into a user-friendly response.

5. **Frontend Development:**
   ○ Create a basic web interface using Flask or React or a similar lightweight framework.
   ○ The frontend should allow users to input questions and display the natural language answers generated by the system (powered by the knowledge graph retrieval and the LLM).

6. Evaluation and Presentation:

○ Prepare a brief presentation (5-7 minutes) covering:

§ The chosen knowledge domain and the rationale behind it.

§ The designed knowledge graph schema and why those entities and relationships were chosen.

§ How the LLM is used to understand the user's question and guide the knowledge graph query process.

§ How information retrieved from the knowledge graph is used by the LLM to generate the final answer.

§ Demonstration of the question answering system with relevant example questions.

§ Discussion of any challenges faced, particularly in bridging the gap between natural language questions and structured graph queries, and in ensuring the LLM generates accurate and relevant answers.

Key Instructions:

- Please aim to complete the assignment to the best of your ability within a reasonable timeframe. We understand you may have other commitments.
- Be prepared to present your solution in approximately 10-15 minutes during the interview. Please focus on the architecture, key implementation details, and a demonstration of your application.
- Wherever applicable, you can use DS internal resources you have access to. e.g. for LLMs, you can use FM APIs.