Couchbase Solution Engineer Tech Challenge

Building a Semantic Cache with LangChain & Couchbase Capella

Objective

Build a Python-based command-line semantic caching application using Couchbase Capella's built-in **Travel Sample** dataset and LangChain. Your application should retrieve relevant information based on user queries, combining keyword and semantic (vector) search.

✅ Skills Evaluated

- JSON querying with SQL++
- Semantic retrieval using LangChain
- Couchbase Capella's Full-Text Search (FTS) & Vector Indexing
- Clear technical communication

📌 Step-by-Step Challenge Guide

1️⃣ Set Up Couchbase Capella

- Sign up for the Couchbase Capella Free Tier.
- Deploy the built-in **Travel Sample** dataset from your Capella dashboard.

2️⃣ SQL++ Queries

- Explore and query the Travel Sample dataset using Capella's Query Editor.

3️⃣ Generate Embeddings with LangChain

- Install LangChain and select an embedding model (e.g., SentenceTransformers).
- Generate embeddings for relevant fields (e.g., hotel descriptions).
- Store these embeddings back into Couchbase documents.

4️⃣ Create Vector Search Index (FTS) in Capella

- In Couchbase Capella, set up a Full-Text Search (FTS) index for your embeddings.

5️⃣ Semantic Retrieval with LangChain

- Implement retrieval logic combining SQL++ keyword searches and semantic (vector) searches.

- Format responses clearly for users.

6️⃣ Build a Simple CLI

- Create a Python command-line interface allowing interactive queries and responses.

7️⃣ Documentation & Submission

Prepare a brief document clearly explaining:

- Your embedding and schema choices
- How SQL++ and vector search complement each other
- Challenges encountered and areas for improvement

Include a short code walkthrough (5-10 min).

Submit your Python application via GitHub or ZIP archive.

📌 Evaluation Criteria

- **Couchbase Proficiency** – Data querying, indexing
- **LangChain Integration** – Effective semantic retrieval
- **Vector Search** – Capella FTS implementation
- **Code Quality** – Structured, readable Python
- **Technical Communication** – Clear explanations

🚀 Bonus Points

- Optimize retrieval performance
- Experiment with different embedding models

🛠️ Recommended Tools

- Couchbase Capella Free Tier
- LangChain
- SentenceTransformers (for embeddings)

🎯 **We're excited to see your semantic cache in action!**