Track A – Research + Prototype (Agent/LLM Use Case)

Objective:

Design and prototype a lightweight AI agent that performs a *marketing-relevant* research task.

🧠 **Suggested Agent Ideas:**

- An agent that reviews Meta/Google ad performance CSVs and outputs insights and creative improvement suggestions
- A multi-step agent that uses a vector database (like Chroma or Pinecone) to search a set of marketing blogs and answer a user's query (e.g. "Best ad copy for summer sale campaigns")
- An agent that rewrites user-uploaded ad text using a particular tone (e.g., fun, professional) and optimizes it for different platforms

In addition to completing the prototype, please address the following in your submission write-up:

1. **Use of Graph RAG / Agentic RAG** (if applicable):
    - Highlight if your agent leverages **Graph-based Retrieval Augmented Generation** or **Agentic RAG**.
    - Explain how this helps with complex, multi-step reasoning or improves recall and precision.
2. **Knowledge Graph Integration**:
    - Specify how your solution could make use of a **Knowledge Graph** to represent structured domain knowledge (e.g., ad platforms, user intent, creative types).
    - You may include examples of how relationships between entities improve response relevance.
3. **Evaluation Strategy**:
    - Clearly define how you would **evaluate your agent's performance**.

    - Include sample metrics (e.g., relevance, hallucination rate, F1 score for extraction, ROUGE for summaries) and whether it involves manual or automated testing.

4. **Pattern Recognition and Improvement Loop**:
    - Briefly describe how your agent can **learn or adapt over time**.
    - For example: using memory modules (like LangGraph's memory nodes), feedback loops, or prompt refinement based on prior errors.

💼 **Deliverables:**

- A working prototype (Colab, GitHub, or demo link)

- The final solution **must be served using a FastAPI backend** — this is a mandatory requirement. The API should expose your agent's functionality through at least one working route (e.g., POST `/run-agent`)

- **Note:** While making any additions or improvements to the code or workflow, please ensure that the existing functionality and context remain intact. Build *on top of* the current structure, not by replacing or removing core logic.
- Technical write-up (400–500 words) with:
    - Architecture and tools used (LangGraph, RAG, LangChain, LLMs, etc.)
    - Challenges faced and how you solved them
    - Potential improvements and next steps