🚀 Project 1. AI Engineer Candidate Project Proposal

Company: ScaleX Innovation
Project Title: **Multi-Source RAG Chatbot with LangGraph, ChromaDB, and Web Search**
Duration: 5 Days - **18 April 2025**
Evaluation Tool: LangSmith
Candidate Level: Intermediate to Advanced AI Engineer

🧠 Project Summary

The objective of this project is to design and implement a **Retrieval-Augmented Generation (RAG)** chatbot capable of answering user queries using three retrieval modes:

1. **Native LLM responses** (direct from model)
2. **Vectorstore-based RAG** using ChromaDB and LangGraph
3. **Web search-based** augmentation using a search API/tool (e.g., Tavily or Serper)

The chatbot should dynamically **choose the best retrieval mode** based on the nature of the user's query, demonstrating **context awareness and retrieval strategy switching**.

🎯 Project Objectives

- Build a **LangGraph-based pipeline** with decision nodes for mode selection.
- Integrate **ChromaDB** for vectorstore RAG over a custom document set.
- Incorporate a **web search tool** for real-time data retrieval.
- Implement a chatbot frontend (CLI or web) to interact with users.
- Evaluate chatbot performance on **50 benchmark questions** using **LangSmith**.

🔧 Technical Requirements

- **LLM**: OpenAI GPT-4-turbo (or DeepSeek if available) or FREE LLMs.
- **Framework**: LangChain + LangGraph
- **Vector DB**: ChromaDB
- **Web Search**: Tavily API, Serper, or Bing API (free tier acceptable)
- **Evaluation**: LangSmith for tracing and comparison
- **Frontend**: CLI or Streamlit Web UI (optional but preferred)
- **Deployment**: Local (Docker setup optional)

📂 Dataset & Evaluation

- **Corpus for RAG**: Provide 10–20 documents (PDFs or text files) in a predefined domain (e.g., AI, cloud, or fintech)

- **Evaluation Set**: 50 structured benchmark questions covering:
  - Fact recall from documents
  - Web-dependent queries
  - Generic conversational prompts

- Use **LangSmith traces and metrics** (e.g., latency, source usage, accuracy) for evaluation

📤 Submission Guidelines

Candidates must submit:

- **GitHub Repo Link** with well-structured code, including a `run.sh` script for automatic execution.
  A hosted deployment link (optional, for bonus points).
- **Documentation** explaining the project, setup instructions, API usage, and LangSmith integration (including how it was used for evaluation). Also, mention which coding AI tool (e.g., GitHub Copilot, ChatGPT) you have used to accomplish this project.

- **LangSmith Report**: Include a summary of evaluation results using LangSmith, with key screenshots or links to traces if applicable.
- **Video**: A 2–3 minute video explaining your code and showing a sample execution. Post the video on **YouTube as an unlisted video** and only share the link.

Submission Form

https://docs.google.com/forms/d/1-btsLJyb_Wo12DVfgPxgaI_vqsb-gCyCwlS2qrN3Ar4/preview