LearnTube
by CareerNinja

Objective

Develop an interactive AI-powered chat system that helps users optimize their LinkedIn profiles, analyze job fit, and provide career guidance. The system should intelligently process LinkedIn profile data, suggest improvements, and generate job-specific recommendations. The interaction should be fully chat-based with memory capabilities to retain context across sessions.

Key Functionalities

1. **Interactive Chat Interface**
    - There should be input for linkedin profile url and then a chat interface where the bot communicates the feedback and the improvement point. (you can use apify's linkedin scraper for this comes with free credits)
    - The assistant should guide them through profile optimization, career recommendations, and job fit analysis.

2. **Profile Optimization, Job Fit Analysis & Career Guidance**
    - **Profile Analysis:** Extract and evaluate LinkedIn profile sections (About, Experience, Skills, etc.), identifying gaps and inconsistencies.
    - **Job Fit Analysis:** Users can provide job roles that they want to apply for, and the system should compare an industry standard job description for the role with the user's profile, generate a match score, and suggest improvements.
    - **Content Enhancement:** Generate rewritten versions of profile sections for better alignment with industry best practices and specific job descriptions.
    - **Career Counseling & Skill Gap Analysis:** Identify missing skills needed for target roles and suggest learning resources or career paths.

3. **Memory System for Personalized Experience**
    - The app should maintain session-based and persistent memory to track user interactions, profile state, and career goals.(You can use LangGraph's checkpointers, Memory saver and more)
    - Ensure seamless context retention across multiple user queries.

LearnTube
by CareerNinja

Technical Expectations

- Strong **prompt engineering** skills to refine AI responses.
- Implement a **multi-agent system** (via LangGraph or Langchain) to handle profile analysis, content generation, and job matching.
- An **effective memory system** for user context tracking.

Submission Requirements

1. **Hosted Application:**
    - Can be built using any framework (Prefered Streamlit, FastAPI, LangGraph).
    - Should be accessible via a public URL.
    - Can use any external tool available.
2. **GitHub Repository:**
    - Must include complete source code, setup instructions, and a well-documented README.
    - Include `requirements.txt` or equivalent for dependencies.
3. **Documentation:**
    - Clear code comments and explanation of architecture.
    - A separate document outlining approach, challenges, and solutions.