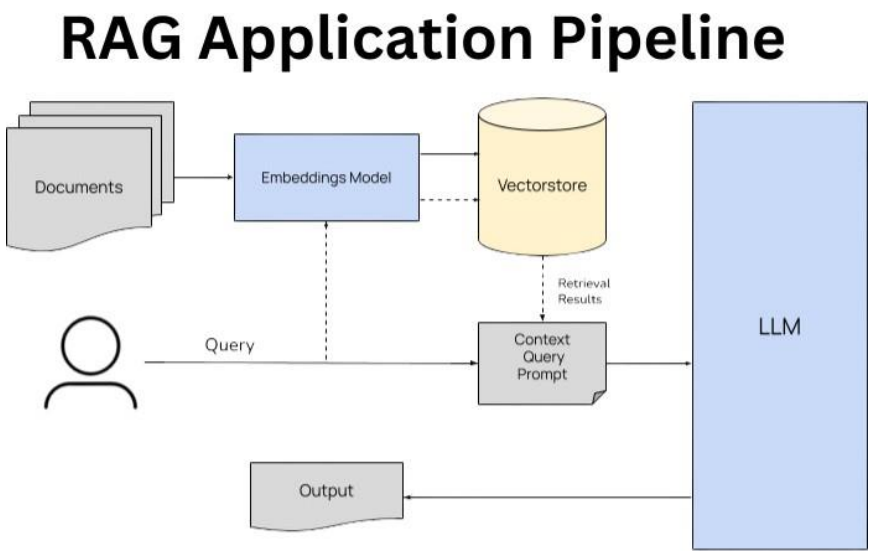
Implementing a Retrieval Augmented Generation (RAG) System



Objective:

Design and develop a prototype RAG system (with UI) that can provide personalized skill recommendations and career advice to users based on their queries about specific job roles.

Requirements:

1. **Data Preparation:**
   - Utilize the provided jobs data containing fields such as job title, job description, job requirements and career level.
   - Implement document embedding using Sentence Transformers to encode job details into dense vectors.
2. **Retrieval Component:**
   - Retrieve relevant job postings based on user queries using a vector database like Elasticsearch with ElastiKNN vector indexing, Faiss or Pinecone.
   - Implement a search mechanism that utilizes vector similarity search to find the most relevant job postings for a given query.
   - PS: Feel free to utilize LangChain in this task.
3. **Generation Component:**
   - Utilize a generative model such as Phi-3 - Mini 4k Instruct, Llama 3 - 8B Instruct, Gemma - 2B Instruct, or Mistral - 7B Instruct for crafting personalized responses and recommendations.

4. **Authentication:**
    - Authenticate users and let them register new accounts.
    - Ensure the security and privacy of users' personal information
5. **Chat & Messages memory**
    - Use SQL or NoSQL database to store chats and messages between users and LLM
    - Allow user to create, read, delete chats (whole conversation, no need to delete a single message)
    - Build a very simple UI to let the user interact with the web app
6. **Parameter Estimation**
    - LLM is required to estimate parameters dynamically based on some learnable insights.
    - The user will ask LLM to suggest a salary range for a specific job given years of experience, the LLM should estimate the salary based on specific parameters and return the result, and justify the estimation.
7. **Integration:**
    - Integrate the retrieval and generation components into a cohesive RAG system.
    - Develop a query processing module to interpret user queries, trigger the retrieval process, and generate personalized recommendations accordingly.
    - Integrate Chat messages with LLM memory.
    - Allow user to upload PDF files and Vectorize the text content for the LLM to use it to answer user's queries.
8. **(Bonus) Documentation and Presentation:**
    - Provide clear documentation detailing the system architecture, implementation details, and usage instructions.
    - Prepare a presentation or demonstration of the RAG system, highlighting its functionality and performance.

Example Input/Output:

- After login & uploading relevant document
- The LLM should be capable of providing the job seeker with personalized career advice, salary range, based on their targeted job title.

Targeted Job Title: Machine Learning Engineer
Personalized Career Advice: Based on your interest in the role of Machine Learning Engineer, here are some personalized career advice:

1. Strengthen your foundation in mathematics and statistics, as they form the backbone of machine learning algorithms. Focus on concepts like linear algebra, calculus, probability, and optimization techniques.
2. Enhance your programming skills in languages commonly used in machine learning, such as Python and R. Familiarize yourself with libraries like TensorFlow, PyTorch, and scikit-learn for implementing machine learning models.
3. Build a strong portfolio of projects showcasing your expertise in machine learning. Work on real-world datasets, develop and deploy machine learning models, and document your process and results on platforms like GitHub or Kaggle.
4. Stay updated with the latest trends and advancements in the field of machine learning. Follow research publications, attend conferences, and participate in online courses or workshops to expand your knowledge and skills.
5. Network with professionals in the machine learning community. Join relevant online forums,

participate in meetups or conferences, and connect with mentors who can provide guidance and support in your career journey.
Remember that continuous learning and practical experience are key to advancing your career as a Machine Learning Engineer. Keep exploring new technologies, solving challenging problems, and seeking opportunities for growth and development

Deliverables:

- Python code implementing the RAG system, including data preparation, retrieval, generation, and integration components.
- Web app (use any technology of your choice)
- Documentation including system architecture, implementation details, evaluation results and demonstration showcasing the RAG system's capabilities, put this in the README file for simplicity.

Notes:

- Utilize Sentence Transformers for document embedding, Elasticsearch with ElastiKNN or Faiss or Pinecone vector indexing for retrieval, and the specified generative models for response generation.
- Ensure scalability and efficiency of the system, considering the large volume of user queries and job postings.
- Explore optimizations and techniques to enhance the performance and relevance of the generated recommendations.
- If you encounter challenges running large language models (LLMs) like Llama 3 - 8B Instruct or Gemma - 2B Instruct on your local machine due to resource constraints, consider using workarounds such as model quantization or opting for smaller models that your system can handle.

Submission:

Please submit your code, documentation, and any relevant materials via email or a shared private repository link within the specified timeframe **(1 Week)**.