LangChain RAG Project: Insurance Domain (Beginner Guide)

■ *Project Goal*

Build a question-answering chatbot that can read and understand insurance documents. It retrieves relevant info from the document and answers user queries using a pre-trained Hugging Face model.

■ *Tools & Libraries Used*

- LangChain: RAG Framework
- Hugging Face Transformers: Pre-trained model (FLAN-T5)
- FAISS: Vector database for semantic search
- SentenceTransformers: To embed chunks
- PyPDFLoader: For reading PDFs
- Gradio: For chatbot UI

■ *Step-by-Step Workflow*

1. **Install Libraries** using pip (LangChain, FAISS, etc)
2. **Mount Google Drive** in Colab to access insurance PDFs
3. **Load Documents** using PyPDFLoader
4. **Split into Chunks** using RecursiveCharacterTextSplitter
5. **Embed Chunks** into vectors using HuggingFaceEmbeddings
6. **Store & Search** chunks in FAISS vector DB
7. **Load FLAN-T5 Model** from Hugging Face
8. **Build RAG Pipeline** using LangChain's RetrievalQA
9. **Ask Questions** about the documents and get responses
10. **Launch UI** with Gradio for easy interaction

■ *Example Query*

- Question: What does the insurance cover?
- The system finds relevant chunks, sends to the model, and answers based on real content.

■ *Deployment Notes*

■ Runs completely in Google Colab.

■ No API key needed (open-source Hugging Face model).

■ Documents stored in Google Drive.

■ Outputs can be shared as PDF or deployed to GitHub.

■ *File Suggestions for GitHub*

- `rag_insurance_colab.ipynb`: Colab notebook with full code
- `README.md`: Project summary and how to run
- `requirements.txt`: Required pip libraries
- `sample_policy.pdf`: Sample insurance doc (if allowed)

■ *Outcome*

You now have a beginner-friendly Retrieval-Augmented Generation system that can answer real questions from custom insurance documents using LangChain + Hugging Face.