# RAG-based Intelligent Q&A; System

A Complete Full-Stack Guide (Hugging Face + FAISS + LangChain)

## 1. Project Overview

This project is a Retrieval-Augmented Generation (RAG) based Intelligent Question & Answering system. Users upload a PDF document and can ask natural-language questions. The system retrieves relevant content from the document using vector search (FAISS) and generates accurate answers using a Hugging Face language model.

## 2. Why RAG?

- Avoids hallucinations by grounding answers in document content.
- Works with private/custom documents (PDFs).
- Scales well using vector databases.
- Industry-standard approach for enterprise Q&A.;

## 3. Technologies & Tools Used

| Layer | Technology |
|---|---|
| Frontend | React.js, Vite, Tailwind CSS |
| Backend API | FastAPI, Uvicorn |
| LLM | Hugging Face (FLAN-T5 Base) |
| Embeddings | Sentence-Transformers (all-MiniLM-L6-v2) |
| Vector DB | FAISS |
| RAG Framework | LangChain |
| Document Parsing | PyPDF |
| Language | Python 3.11 |

## 4. System Architecture

### Flowchart (Textual Representation):

```
User → Frontend → Backend API → PDF Loader → Text Splitter → Embeddings → FAISS Vector Store → Retriever → Prompt Builder → Hugging Face LLM → Answer → Frontend
```

## 5. Working Model (Step-by-Step)

1. User uploads a PDF file from the frontend UI.
2. Backend saves and loads the PDF using PyPDFLoader.
3. Text is split into overlapping chunks.
4. Chunks are embedded using Sentence-Transformers.
5. Embeddings are stored in FAISS.
6. User asks a question.
7. Relevant chunks are retrieved using similarity search.
8. Prompt is constructed with retrieved context.
9. LLM generates an answer.
10. Answer is displayed in the UI.

## 6. Backend API Endpoints

- POST /upload – Upload and index a PDF.
- GET /ask?question=... – Ask a question.

## 7. Setup Instructions

```
Backend: 1. Create & activate virtual environment 2. pip install -r requirements.txt 3. python -m uvicorn main:app --reload
```

```
Frontend: 1. npm install 2. npm run dev 3. Open http://localhost:5173
```

## 8. Conclusion

This project demonstrates a production-ready RAG system suitable for portfolios and real-world use.