📘 Enterprise AI Q&A System

AI Assistant Powered by Internal Enterprise Knowledge

🧠 Product Overview

This system is an AI-powered enterprise Q&A solution built on the RAG (Retrieval-Augmented Generation) framework and ChatGPT. By uploading internal documents, companies can create a private knowledge base and allow users to ask natural language questions through an intuitive interface to receive accurate answers with source references.
“Turn enterprise knowledge into something searchable, askable, and usable.”

🚀 Core Functions and Features

🔎 Documents as Knowledge

- Users can upload internal corporate documents in formats such as PDF, TXT, and DOCX.
- The system automatically parses the document content, segments it, and converts it into vectors, which are stored in a vector database (FAISS).

💬 Instant Q&A

- Users can freely input natural language questions in the interface.
- The system compares the question with the vector knowledge base and returns the most relevant passages to ChatGPT for further answer generation.

📚 Referenced Sources

- The names of the original documents used as references are displayed below the answers, enhancing information credibility and traceability.

🎨 Intuitive Interface (Streamlit)

- A modern dark-themed interface with a sidebar navigation menu.
- Supports document uploading, question input, answer display, and progress prompts.

🧪 Technical Architecture

Component	Technology/Tool
Vectorization	LangChain, FAISS, OpenAI Embeddings
Language Model	ChatOpenAI (GPT-4o, GPT-3.5...)
Document Parsing	PyMuPDF, docx2txt, TextLoader
Front-end Interface	Streamlit + Custom CSS
Parameter Settings	.env + streamlit secrets.toml

🔁 RAG Architecture Implementation Process:

1. User inputs a question.
2. The system embeds the question into a vector.
3. The vector database retrieves relevant passages.
4. ChatGPT generates a natural language answer incorporating the document content.
5. The response and referenced sources are displayed.

🧩 Key Innovations

- ✅ No fine-tuning required – just plug in documents
- ✅ Supports multiple formats – PDF, Word, plain text
- ✅ Instant knowledge updates – automatic vectorization upon upload
- ✅ Visible source references – combats hallucination by ensuring answer transparency
- ✅ Easy deployment – supports Streamlit Cloud, Docker, or local deployment

---

🎯 Use Cases

Scenario	Example Query
Customer Support	“How do I apply for a return?”
HR Manual Access	“What is the overtime application process?”
Technical Reference	“What authentication does our API use?”
Knowledge Transfer	Onboarding new employees to internal SOPs

🛠️ Quick Start (Deployment Guide)

```
git clone https://github.com/your-org/your-repo.git

cd your-repo

# Install dependencies

pip install -r requirements.txt

# Create .env file and add your OpenAI API key

cp .env.example .env

# Launch the app

streamlit run app.py
```

📂 Project Structure

```
source_documents/     ← Uploaded source documents

vectorstore/          ← Storage for vectorized data

app.py                ← Main Streamlit application

ingest.py             ← Vectorization script

private_gpt.py        ← Core RAG-based Q&A logic
```

🧱 Extended Features (Under Development / Customizable)

- 🔐 Permission Management / Multi-user Login
- 🌐 Multi-language Document Processing and Translation
- 🧠 User-selectable Language Models (Llama, Claude, Mistral)
- ☁️ Cloud Document Integration (Notion, Google Drive)

📞 Contact Us

For customization requests or business cooperation inquiries, please contact:

🧑‍💻 Developer: Wayne Chou