【Day 9】 Demo: RAG 個人助手

距離完成個人助手 RAG 的 demo，我們要結合前幾天學習過的內容，再加上以下幾處細節。

1. 代碼

1.1 Chat History + RAG

我們的 prompt 現在包含 `chat_history`，`input` 和 `context` 三個 key。`retrieval chain` 會幫我們加入檢索到的文件 `context`，它回傳一個字典，LLM 回覆會放在 `answer` 這個 key 所以將 `output_messages_key` 設為 `answer`。

```
prompt_template_with_rag = ChatPromptTemplate.from_messages(
    [
        ("system", system_prompt),
        MessagesPlaceholder(variable_name="chat_history"),
        ("human", "**Human Message:**{input}\n\n**Content:**{context}")
    ]
)

... # retrieval_chain 就是 day7 用兩個函數定義的那個

chat_with_rag = RunnableWithMessageHistory(
    retrieval_chain,
    get_chat_history,
    input_messages_key="input",
    history_messages_key="chat_history",
    output_messages_key="answer"
)
```

1.2 緩存資料庫

FAISS 向量資料庫，使用 `@st.cache_resource` 將他緩存起來，不同的使用者，跨 session 都會讀取到同一個資料庫。

```
@st.cache_resource
def get_vector_store():
    return FAISS.load_local(
        "vector_store",
        GoogleGenerativeAIEmbeddings(
            model="models/gemini-embedding-001",
            google_api_key=st.secrets["GOOGLE_API_KEY"]),
        allow_dangerous_deserialization=True
    )
```

1.3 RAG 啟用按鈕

使用 `st.segmented_control` 選擇是否啓用 **RAG** 功能。

```
option_map = {
    0: ":material/database: RAG",
}
selection: list = st.segmented_control(
    "**Tool**",
    options=option_map.keys(),
    format_func=lambda option: option_map[option],
    selection_mode="single",
)
```

1.4 Retrieved Context

在 AI 回覆後使用 `st.expander` 來展示檢索的內容，這個功能並非必須，此處是為了檢查回覆的正確性。

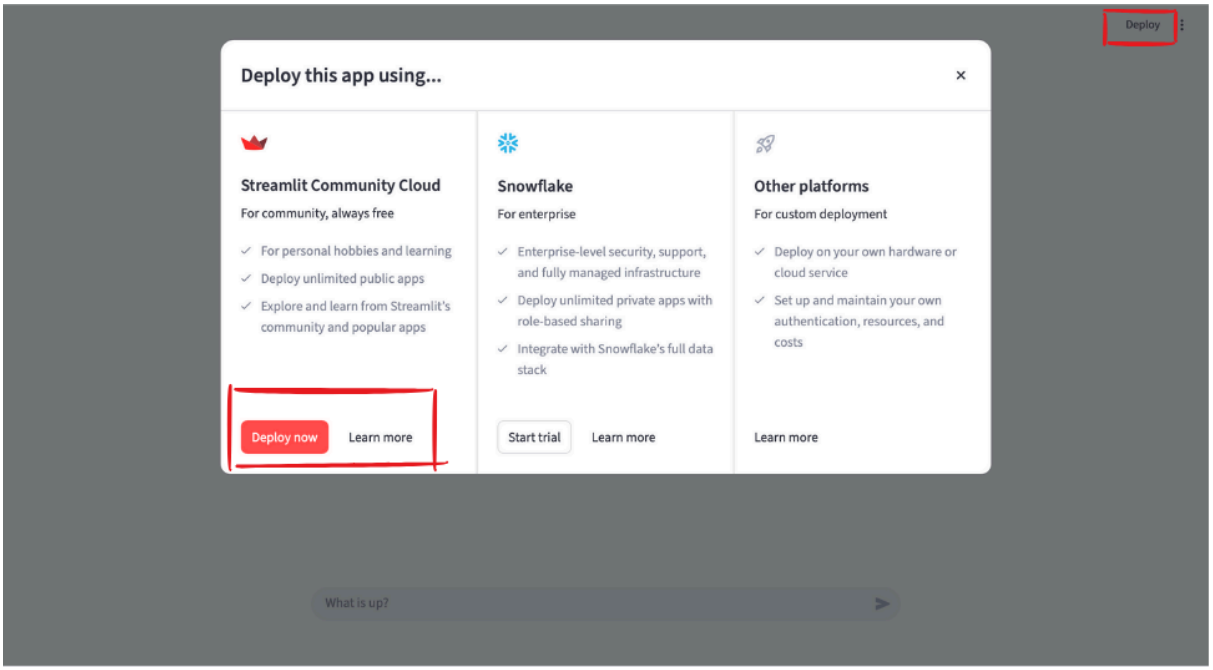
```
if got_context:
    with st.expander("🔎 Retrieved Context", expanded=False):
        for doc in context:
            st.markdown(f"**Source:** {doc.metadata['source']} - Chunk {doc.
metadata['chunk_index']}")
            st.markdown(doc.page_content)
            st.markdown("---")
```

2. 部署 Streamlit Cloud

完成代碼後，我們可以使用 Streamlit Cloud 來部署這個 app。

2.1 部署

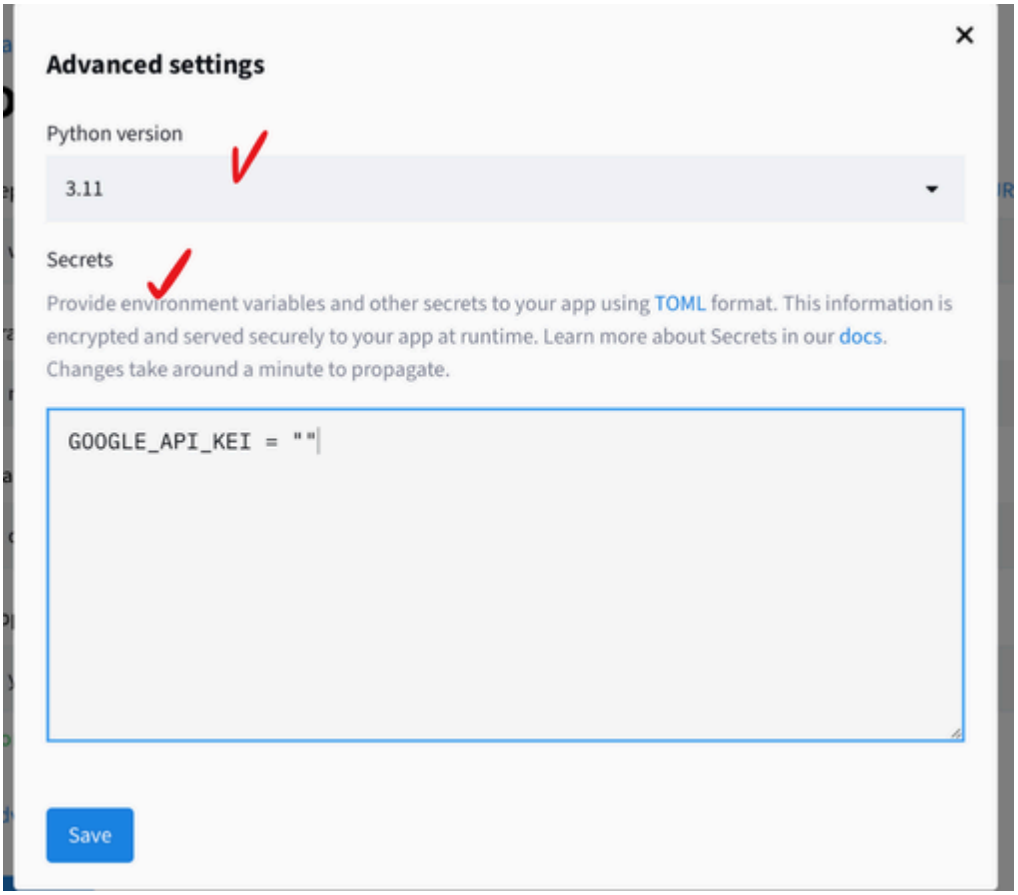
- 在運行這個 app 時點擊**右上角 “Deploy”**
- 接著，選擇 **“Deploy now”**
- 填寫 repository, branch, file 和 url 之後點擊 **“advanced setting”**



2.2 Advanced Setting

進入到進階設定，我們要設置 python 版本以及環境變數：

- python version
  - 在此設為 **3.11**
- streamlit secrets
  - local 運行時 會自動取用 .streamlit/secrets.toml
  - 在 streamlit cloud 部署時，要手動填入環境變數



2.3 Dependencies

streamlit 會自動取用 GitHub repo 的 requirements.txt （或是 conda uv 等套件管理的相關文件），要記得加進去。

2.4 embed

部署好之後，我們可以把 app 嵌入在網頁裡面，步驟如下

- 點擊右上角 **“share”**
- 選擇 **“embed”**
- 設置參數後，點擊 **“<> get embed link”** 獲取連結
- 使用 **<iframe>** 加入網頁

```
<iframe
  src="your_embed_link"
></iframe>
```

3. Demo Time!

[Streamlit Demo 連結](https://personal-database-rag.streamlit.app)