iSKKUBot

RAG-based Multilingual Chatbot for Exchange Students at SKKU

EQUIPO SASHIMI

- GeonWoo Bang
- JeongYoon Hwang
- Jiae Choi
- Mikel Larrarte Rodriguez

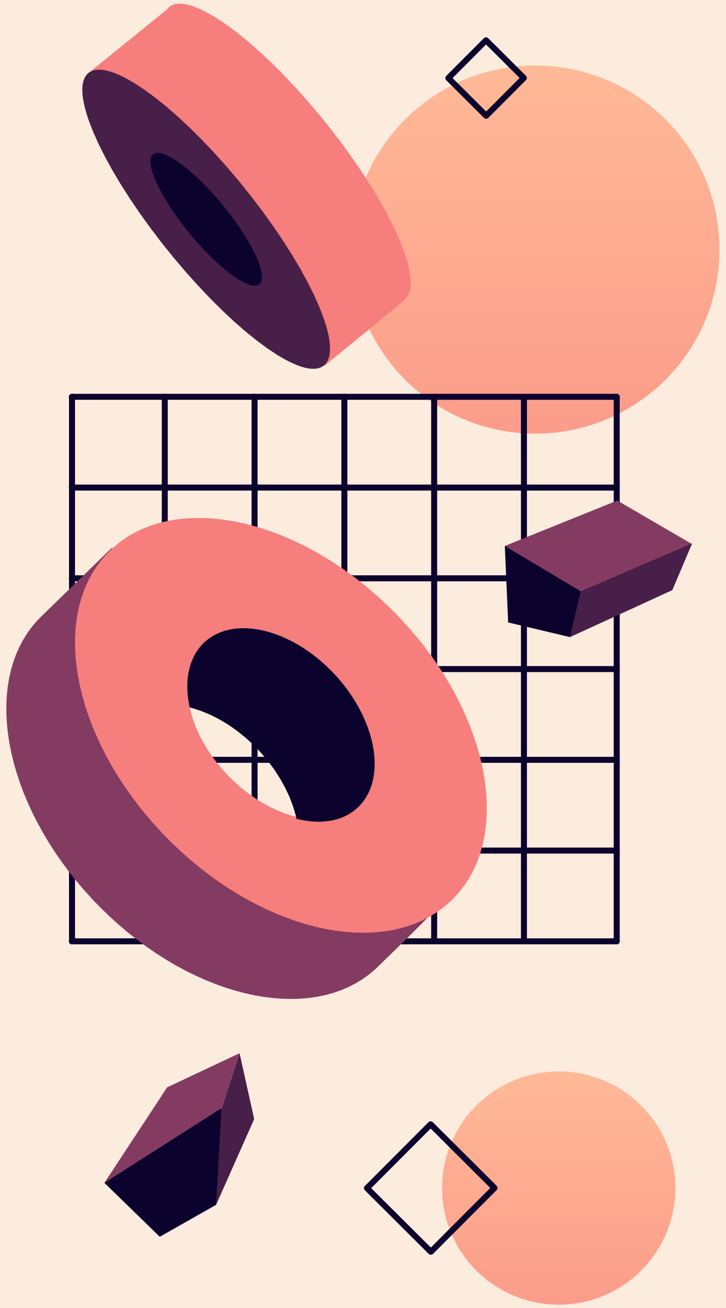


Table of contents.

01

Motivation

Is there a real need?

01) Motivation

Challenges Faced by Exchange Students:

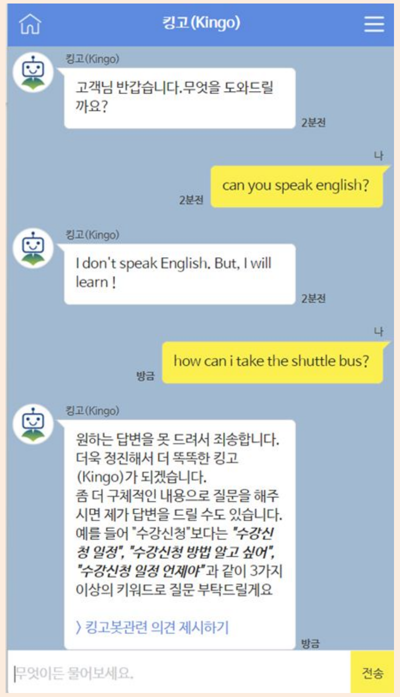
1. **Language Barriers**: Most official information is only available in Korean, which makes navigation difficult.
2. **Scattered Resources**: Information on academics, campus services, and social activities is spread across multiple platforms.

01) Motivation

Limited Chatbots:

Existing chatbots are limited to monolingual capabilities or simple keyword matching → complex queries ✖

- ❖ **Kingobot (SKKU's Chatbot):** Primarily in Korean + strict rules for querying + minimal information for exchange students.
- ❖ **UK International Student Chatbot:** Keyword-based matching + lacks flexibility to handle complex questions.


킹고(Kingo)

킹고(Kingo)
고객님 반갑습니다.무엇을 도와드릴까요?
2분전

나
can you speak english?
2분전

킹고(Kingo)
I don't speak English. But, I will learn !
2분전

나
how can i take the shuttle bus?
방금

킹고(Kingo)
원하는 답변을 못 드려서 죄송합니다.
더욱 정진해서 더 똑똑한 킹고(Kingo)가 되겠습니다.
좀 더 구체적인 내용으로 질문을 해주시면 제가 답변을 드릴 수도 있습니다.
예를 들어 "수강신청"보다는 ***"수강신청 일정", "수강신청 방법 알고 싶어", "수강신청 일정 언제야"***과 같이 3가지 이상의 키워드로 질문 부탁드릴게요

〉킹고봇관련 의견 제시하기
방금

02

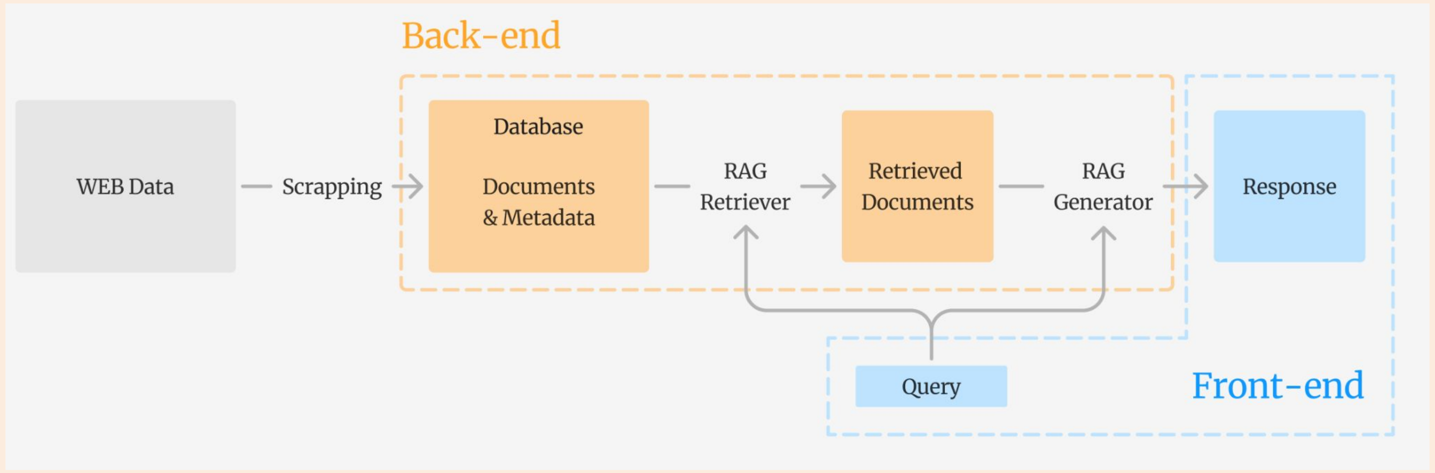
Method Overview

General workflow of the project.

02) Method Overview

Project Workflow:

1. **Scraping**: Gather data from official SKKU webpages, PDFs & other online sources + construct a database for the RAG model to retrieve relevant documents from.
2. **RAG**: With the constructed database, the model returns a response based on retrieved documents & user's query.
3. **Front-End**: Web application that outputs RAG's response given user's query. When requested, the front-end also receives full documents that RAG retrieved, and the link of the documents.



2.1) Scraping

We will gather information from:

➔ SKKU Office of International Student Services webpage.

➔ Today's School restaurant menu

➔ 2024 SKKU Club Guidebook

🦜🔗 LangChain

LangChain → *AsyncHtmlLoader, BeautifulSoup* & other tools to scrape the information.

Pinecone → Storage of embeddings in an external vector database.

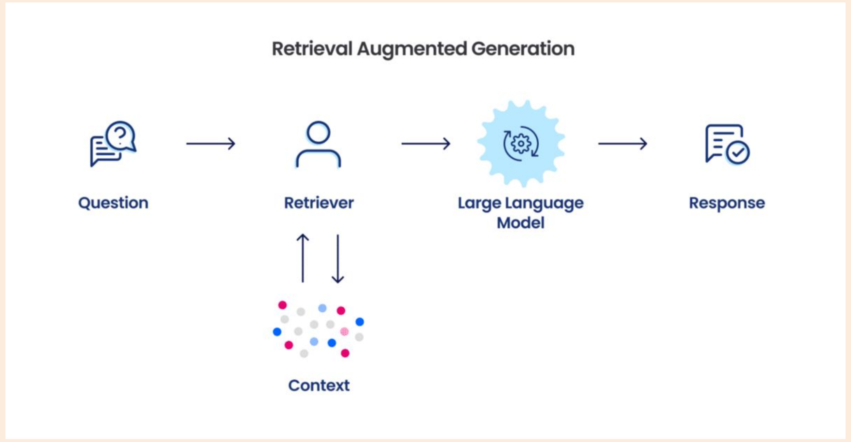
Pinecone Beautiful Soup

2.2) RAG Implementation

Retrieval-Augmented Generation combines a document retriever with a text generator to create accurate responses based on external documents.

Retriever: Uses FAISS for fast, similarity-based document retrieval.

Generator: Takes retrieved documents as input and generates a natural language response using a model like Llama or similar Korean-English bilingual models. **Pretrained + No Fine-Tuning**



2.3) Front-end

We opted for a responsive web design using **React.js** to ensure accessibility on all devices.

Advantages of Responsive Web:

1. Accessible on all devices and browsers.
2. Easy updates without manual intervention.
3. Can be integrated into existing SKKU systems.

React JS

Tools:

❖ **React.js** for building the UI.

❖ **TypeScript** for type safety and error handling during development.

TS

2.4) Back-end

We use **Python** due to its vast library support for NLP, document retrieval, and API development.

Key Libraries:

- **FastAPI:** For building a REST API that handles multiple requests.
- **LangChain:** To retrieve relevant documents and integrate RAG models.
- **Hugging Face:** To obtain the generative model.

FastAPI will handle incoming requests, route them to LangChain for document retrieval, and send the result back to the user.

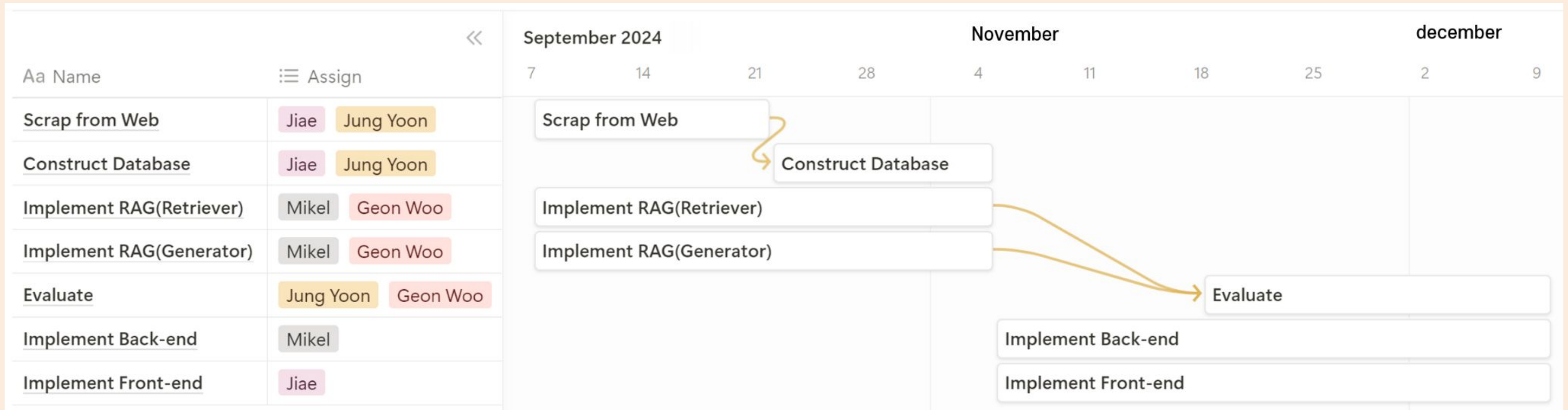
03

Planning & Role Assignment

Workload division and planification.

03) Planification & Role Assignment

- **Scrap from web:** Find web pages with information about SKKU and scrap them.
- **Construct database:** With the scraped data, construct a database for RAG to retrieve relevant documents from.
- **Implement RAG retriever:** Use LangChain to develop the retriever part.
- **Implement RAG generator:** Use a LLM from Hugging Face as the generator.
- **Evaluate:** Construct an evaluation dataset and evaluate the implemented RAG retriever and generator separately.
- **Implement Back-end:** Implement the inference code for RAG to communicate with the front-end.
- **Implement Front-end:** Implement web application that outputs RAG's response given user's query.



Aa Name	Assign
Scrap from Web	Jiae, Jung Yoon
Construct Database	Jiae, Jung Yoon
Implement RAG(Retriever)	Mikel, Geon Woo
Implement RAG(Generator)	Mikel, Geon Woo
Evaluate	Jung Yoon, Geon Woo
Implement Back-end	Mikel
Implement Front-end	Jiae

References :

- 성균관대학교 외국인유학생 지원팀. Available at: https://oiss.skku.edu/oiss/index.do
- *2024-1 arrival guide(en) 성균관대학교 e-Book 라이브러리*. Available at: https://ibook.skku.edu/Viewer/9WLAOPP0LYA9.
- BalaElangovan. *Balaelangovan/International-student_chatbot: Chatbot with the intention to help international students., GitHub*. Available at: https://github.com/BalaElangovan/International-student_Chatbot?tab=readme-ov-file.
- *How to load pdfs 🦜🔗 LangChain*. Available at: https://python.langchain.com/docs/how_to/document_loader_pdf/.
- *Web scraping 🦜🔗 LangChain*. Available at: https://python.langchain.com/v0.1/docs/use_cases/web_scraping/.
- 성균관대학교: 대학생활: 편의 / 복지: 식당 / 메뉴: 인문사회과학캠퍼스. *성균관대학교, SKKU, 성균관대, 성대, Sungkyunkwan University*. Available at: https://www.skku.edu/skku/campus/support/welfare_11.do.

Thank you!

Do you have any questions?