Providing Support for Exchange Students at SKKU: SKKUEXS

Haeun Lee, Huigeun Han, Hyunwon Seo, Juwon Kim

Capstone Design Project[SWE3028], Sungkyunkwan University

Abstract. SKKUEXS is a project focused on enhancing the support system for exchange students at Sungkyunkwan University. The project aims to address challenges faced by these students, including cultural differences, language barriers and social isolation. By providing a community platform, organizing learning materials, connecting students with mentors and alumni and offering reliable information, SKKUEXS strives to create a comprehensive service. This project aims to improve the exchange student experience, promote a sense of belonging and facilitate a smoother transition into a new culture.

Keywords: Exchange student · Information support · Community building

1 Introduction

Exchange programs provide students with valuable opportunities to explore different cultures, gain new experiences and broaden their knowledge. However, exchange students often encounter challenges such as cultural differences, language barriers and social isolation. “SKKUEXS” aims to address these challenges and create a nurturing environment for exchange students at SKKU[1]. By offering services like access to learning materials, a community platform and reliable information, we strive to enhance the exchange student experience and facilitate a smoother transition into their new academic and cultural surroundings. Our techniques involve utilizing Django[1] and AWS Lightsail[2] for web development, SQLite for database management, Figma[3] for web design, KoT5[4], GPT-4[5] for text summarization and WordCloud[6] for visual representation of university-related keywords. Through these approaches, we aim to provide a user-friendly and comprehensive support platform. The logo and slogan of our project is shown in Figure 1.

In the process of implementing our project, we faced challenges in code functionality, collaboration and code sharing using GitHub. Familiarity with GitHub was initially limited and communication barriers hindered the generation of creative ideas and technical solutions during collaborative meetings. However, we

[1] Sungkyunkwan University

overcame these challenges through research, problem-solving and open discussions. We have described the recent activity and the corresponding code in Notion[2], along with relevant photos and organized the tasks to be done. As a result, we achieved significant progress in improving our coding skills, mastering GitHub and enhancing communication and collaboration. Empirical evaluations confirmed improvements in code functionality, efficient code sharing and more productive teamwork. Despite the initial difficulties, our perseverance and growth have ultimately strengthened the outcomes of our project.

Fig. 1. The logo of SKKUEXS

2 Background and Related Work

2.1 Background

The exchange student program is a system where students are dispatched to foreign partner universities with which the SKKU has a student exchange agreement, for a semester or a year of study. However, for students preparing for their overseas experience, there is a significant burden, as a considerable number of students reported independently preparing for the foreign environment.[9]

We conducted a survey using Google Forms to investigate further inconveniences. Figure 2 shows survey results conducted among Sungkyunkwan University students revealed that downloading individual study reports was considered cumbersome by 47.6% of the respondents, while 9.5% found it difficult to have a comprehensive overview of the information. The result says the process of accessing individual study reports is cumbersome and time-consuming, as students need to review and compare information from separate files.

In line with this, as shown Figure 3, 79.5% of the students expressed their willingness to use a website that offers summarized and organized study reports. This demonstrates the significant value of a service that summarizes and organizes the reports.

[2] multi-purpose productivity and collaboration software

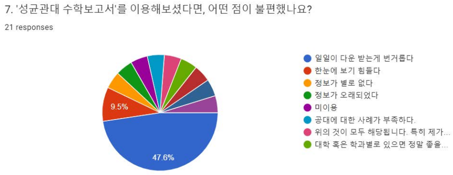


Fig. 2. Inconveniences of study reports

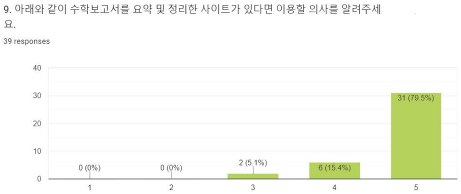


Fig. 3. Intention to utilize a website that offers summarized and organized study reports

2.2 Related work

Several previous works have addressed the challenges faced by exchange students and proposed various techniques and approaches to support their study abroad. Two notable examples include the Yonsei Exchange and the Everytime.

Yonsei Exchange classifies menus by country and provides detailed information about each university within the country. It offers insights into student satisfaction, the number of students who have participated in the program and the majors of seniors who have studied there. This approach primarily assists students in making informed decisions about their exchange program choices by providing comprehensive evaluations and relevant infomations.

Everytime focuses on managing evaluations of courses taken by exchange students. It allows students to rate their experiences and leave comments, providing comprehensive evaluations of each university. By summarizing evaluations and providing numerical data on satisfaction, this system assists exchange students in making wise choices when selecting a university for their exchange program.

3 Service design

By incorporating several design choices, we aim to create a robust and user-friendly system that addresses the challenges faced by exchange students at SKKU, providing them with comprehensive support and facilitating a seamless study abroad experience.

3.1 Overall Architecture

The proposed system follows a client-server architecture where exchange students can access learning materials, connect with the community and find relevant information through a user-friendly web application interface. Figure 4 shows the overall architecture of SKKUEXS. The frontend development involves designing web pages using Figma and implementing them in HTML. On the server side, a backend server hosted on AWS Lightsail, powered by Django, handles user requests, processes data and interacts with the database. The server communicates with external APIs for data retrieval and integrates NLP models for text summarization and analysis. A relational database (SQLite) is used for efficient data storage and retrieval. This system architecture enables seamless interaction, efficient data management and enhanced functionality for exchange students studying abroad.

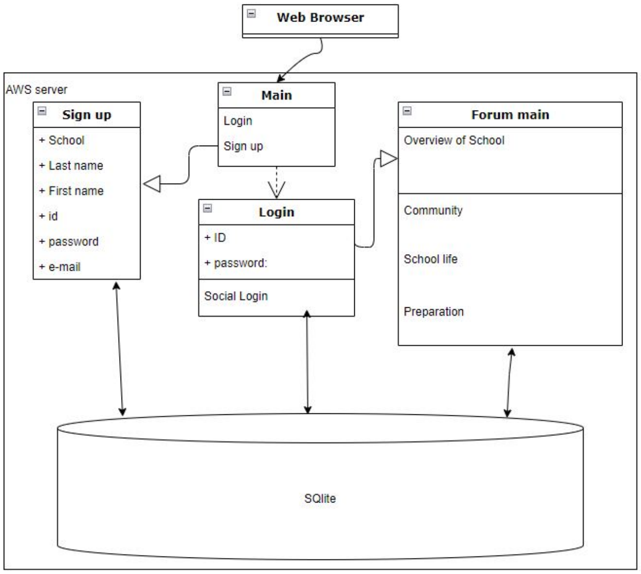
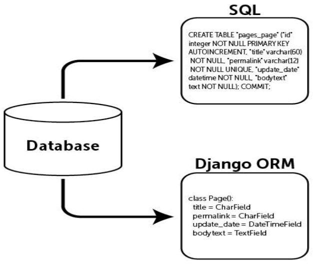


Fig. 4. Overall Architecture

3.2 Core Skill/Technique/Model

One of the key skills in this project involves preprocessing the crawled documents, which are downloaded using Selenium in various file formats such as docx, doc, pdf and hwp. Using Python, the desired data is extracted and organized item by item based on the specific file format and then transformed into a structured format for conversion to Excel. This process requires integrating the extracted information and storing it in a database, which involves designing a suitable database schema, creating tables and establishing connections between the application and the database.

Second core skill utilized in this project is web development using Django framework. Django enables efficient server-side processing, handling user authentication, data storage and routing. It simplifies the development process by providing pre-built functionalities, such as user management and database integration. Django incorporates an object-relational mapping (ORM) system, which is a programming technique that facilitates the interaction between a relational database and an object-oriented programming language. Figure 5 shows the benefits of using ORM (Object-Relational Mapping) over SQL. ORM provides a simpler and more readable way to create and manage database schemas.



```
CREATE TABLE "pages_page" ("id"
integer NOT NULL PRIMARY KEY
AUTOINCREMENT, "title" varchar(60)
 NOT NULL, "permalink" varchar(12)
 NOT NULL UNIQUE, "update_date"
datetime NOT NULL, "bodytext"
text NOT NULL); COMMIT;
```

```
class Page():
  title = CharField
  permalink = CharField
  update_date = DateTimeField
  bodytext = TextField
```

Fig. 5. The benefit of using Django ORM over SQL

3.3 General Challenges and Approach

During the process of preprocessing mathematical reports using Python, the libraries for handling PDF, HWP and DOC files lacked functionality or faced parsing issues. As a result, the documents were converted to DOCX format for preprocessing. However, even in the DOCX conversion process, there were parsing problems, requiring manual intervention in data cleaning. Recently, after preprocessing 2,096 reports, it was observed that there were differences in the format of more 1,139 reports from the years 2009-2010. To address this, new code had to be developed to handle the format changes. Accordingly, Outdated or irrelevant data had to be removed from older report formats. Furthermore, the inconsistent report writing by students posed challenges in data extraction. Some responses were incomplete or had excessive information, causing misalignment with the expected format.

Throughout the project, Collaboration and code sharing using GitHub posed difficulties due to the team's limited familiarity with the platform. To tackle this, we invested time in learning GitHub's version control system, branching, merging and issue tracking. Communication challenges were overcome through regular meetings, open discussions and active participation with Notion.

3.4 Reasoning about Design Choice

The choice of Django as the web framework was driven by its robustness, scalability and extensive community support. It provides a structured development environment, adhering to the Model-View-Controller (MVC) pattern shown as figure 6, facilitating code organization and separation of concerns. Additionally, AWS Lightsail was selected for its reliable hosting capabilities, scalability and ease of deployment, ensuring optimal performance for the web application. Moreover, ORM offers several benefits, including increased productivity, as it eliminates the need to write low-level SQL queries. It also provides database independence, as the ORM framework can generate appropriate SQL statements for different database systems. Additionally, ORM helps to mitigate common security risks, such as SQL injection attacks, by automatically sanitizing input and output.

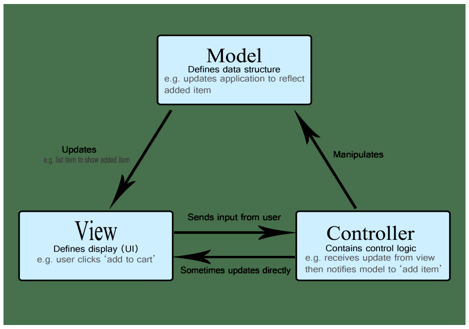

Model
Defines data structure
e.g. updates application to reflect added item
Updates
e.g. list item to show added item
Manipulates
View
Defines display (UI)
e.g. user clicks 'add to cart'
Sends input from user
Sometimes updates directly
Controller
Contains control logic
e.g. receives update from view then notifies model to 'add item'

Fig. 6. MVC architecture: the model, the view, the controller

3.5 ML Model/Algorithm

NLP, or Natural Language Processing, is an AI field that focuses on the interaction between computers and human language. GPT (Generative Pre-trained Transformer) and KoT5 (Korean Text-to-Text Transfer Transformer) are advanced language models that have been pre-trained on large text datasets. These models can generate coherent and contextually relevant text by leveraging the patterns and structures learned during pre-training. By combining NLP techniques with models GPT and KoT5, we summarized the visa application process and dormitory features. We compared the outputs of both models and selected the one that provided more accurate and informative summaries. The selected model's results were presented to showcase the key points of the visa application process and the distinctive features of the dormitories.

4 Implementation

The implementation combines frontend and backend technologies to deliver a user-friendly and interactive web application. The frontend focuses on providing an intuitive and visually appealing interface, while the backend handles data processing, storage and integration with external services. The overall implementaion of project is shown as Figure 7.

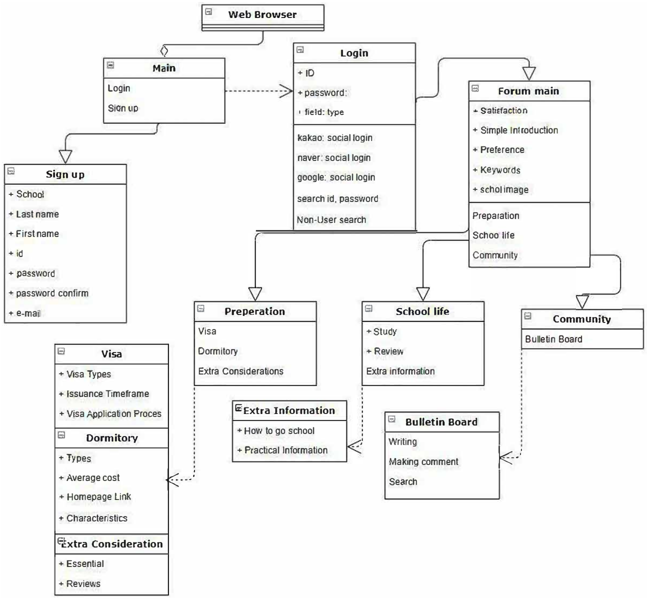


Fig. 7. Overall implementation of SKKUEXS

4.1 UX/UI Viewpoints

The user experience (UX) and user interface (UI) aspects of the system were prioritized to ensure a seamless and intuitive user interaction. The frontend was designed with a user-centric approach, focusing on usability and visual appeal. User research and feedback were considered to create an intuitive navigation flow, clear information hierarchy and visually pleasing design elements. The UI elements were designed to be responsive and accessible across different devices and screen sizes, enhancing the overall user experience.

4.2 Frontend

The frontend development involved translating the UI designs into HTML and CSS code. The frontend team utilized modern frontend frameworks to enhance interactivity and manage dynamic content. To enhance the visual representation of frequent words, we incorporated a WordCloud feature in the UI/UX design, allowing users to easily visualize and explore the most common terms. JavaScript was used for client-side scripting and implementing interactive features. The frontend communicates with the backend server through API calls, retrieving and displaying data to the users.

Login On the root page, there are options for login and registration. When accessing the login page, users can choose to log in using social login options like

Google, Kakao, or Naver, or they can proceed with a regular login using their username and password. Figure 8 shows the login page.

If users forget their username, they can enter their email address, and if it matches the database, the username will be sent to the corresponding email. Similarly, if users forget their password, they can enter their email and username, and a similar process will be followed to send a password reset email and allow them to enter a new password.

If users choose to register using social login, the integration with the respective site's API is performed, and if the user is not already registered, additional information like school name and major will be collected before completing the registration process. If the user is already registered, they will be redirected to their school page.

If user prefers not to log in and use the site for school search purposes, user can search for schools as a guest user. Figure 9 shows the guest users' school search page. Guest users have access to the school main page and other subdomains, but they cannot utilize the bulletin board feature.

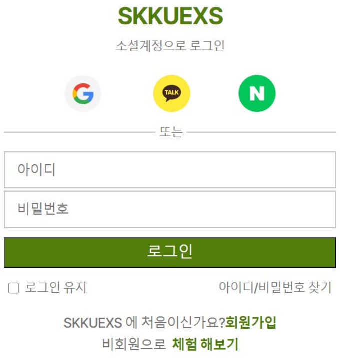


Fig. 8. The login page

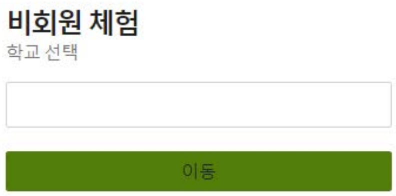
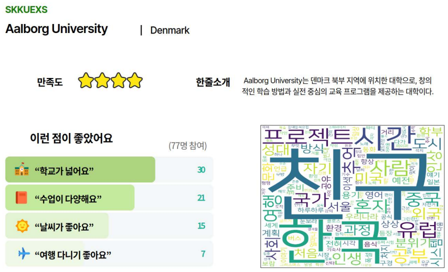


Fig. 9. Guest Experience Page

Registration On the registration page, users are prompted to enter the desired school they want to apply for.shows the registration page. The school name field offers auto-completion suggestions based on the existing school names in the database, making it easier for users to search and select their desired school. Users also provide their first and last name, username, password, password confirmation, and email to create their account.

School Main The school main page displays the name of the school, the country, and a visually intuitive satisfaction rating using a 5-star system. Figure 10 shows the main page of specific school with WordCloud. A brief one-line introduction provides summarized information about the school, and a word cloud visually represents frequent keywords found in the school's academic reports. Additionally, three landmark photos of the school from a Google custom search engine [7] offer a glimpse of the surroundings for users to preview.


SKKUEXS

Aalborg University | Denmark

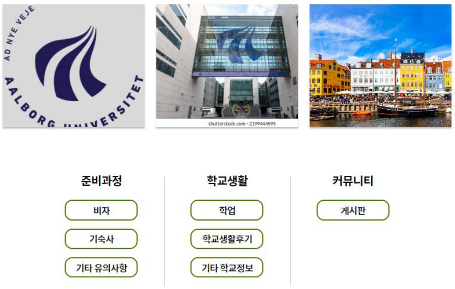
만족도 ★★★★

한줄소개 Aalborg University는 덴마크 북부 지역에 위치한 대학으로, 창의적인 학습 방법과 실전 중심의 교육 프로그램을 제공하는 대학이다.

이런 점이 좋았어요 (77명 참여)

"학교가 넓어요"	30
"수업이 다양해요"	21
"날씨가 좋아요"	15
"여행 다니기 좋아요"	7

Fig. 10. The school main page


준비과정

비자

기숙사

기타 유의사항

학교생활

학업

학교생활후기

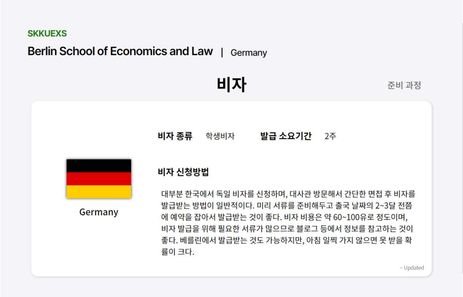
기타 학교정보

커뮤니티

게시판

Fig. 11. The school index page

School Subpages The school subpages are categorized into three sections: *Preparation*, *Student Life*, and *Community*. Figure 11 shows the indexing subpages of school.


SKKUEXS

Berlin School of Economics and Law | Germany

비자

준비 과정

Germany

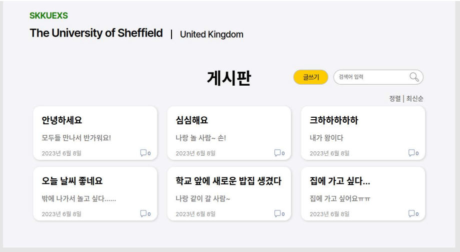
비자 종류 학생비자 발급 소요기간 2주

비자 신청방법

대부분 한국에서 독일 비자를 신청하며, 대사관 방문해서 간단한 면접 후 비자를 발급받는 방법이 일반적이다. 미리 서류를 준비해두고 출국 날짜의 2~3달 전쯤에 예약을 잡아서 발급받는 것이 좋다. 비자 비용은 약 60~100유로 정도이며, 비자 발급을 위해 필요한 서류가 많으므로 블로그 등에서 정보를 참고하는 것이 좋다. 베를린에서 발급받는 것도 가능하지만, 아침 일찍 가지 않으면 못 받을 확률이 크다.

- Updated

Fig. 12. The visa page


SKKUEXS

The University of Sheffield | United Kingdom

게시판

안녕하세요
모두들 만나서 반가워요!
2023년 6월 8일 0

심심해요
나랑 놀 사람~ 손!
2023년 6월 8일 0

크하하하하하
내가 왕이다
2023년 6월 8일 0

오늘 날씨 좋네요
밖에 나가서 놀고 싶다......
2023년 6월 8일 0

학교 앞에 새로운 밥집 생겼다
나랑 같이 갈 사람~
2023년 6월 8일 0

집에 가고 싶다...
집에 가고 싶어요ㅠㅠ
2023년 6월 8일 0

Fig. 13. The forum page

1. *Preparation* Under the Preparation section, users can access information related to visas, dormitories, and other important considerations. Figure 12 shows the specification of visa page. The visa page provides details on visa types, processing time, and application procedures. The dormitory page offers information about different types of dormitories, average costs, links to

dormitory websites, and notable features. The other considerations page provides essential preparation items and valuable tips based on the experiences of senior students.

2. *Student Life* In the Student Life section, the academic page presents reviews and evaluations of courses by senior students, with satisfaction ratings presented in the form of star ratings. The student life review page showcases testimonials and insights from senior students who have attended the school. The Miscellaneous section offers interactive map image on how to get to the school from nearby international airports using MapQuest API[8] and provides useful information for students' school life.
3. *the Community* The Community section includes a bulletin board feature that is unique to each school, allowing users to write posts and search for existing posts. Figure 13 shows that the forum page is intuitive and simplistic display of posts. Users can write a post by providing a title and content within 200 characters, with an option to enable comments. Once a post is created, the title and the first sentence of the post are displayed, along with the date it was posted. Users can also leave comments on each post. school life.

4.3 Backend

Django, a Python web framework, was employed for server-side development. It handles user authentication, routing and business logic implementation. The backend interacts with the database, utilizing SQLite for data storage and retrieval. External APIs are integrated to fetch relevant information for the exchange program and university details. NLP libraries of KoT5 and GPT-4 are utilized for text summarization and analysis. AWS Lightsail provides the infrastructure for hosting the backend server, ensuring reliable and scalable performance. We chose to use AWS server with 1GB RAM, 1 vCPU, and 40GB SSD, as it offers a lightweight and suitable solution for our initial deployment, despite its limitations.

5 Limitations and Discussions

5.1 Limitations

Insufficient Information One limitation of the system is that if a student's desired university is not available, it may fail to provide sufficient information to the student. Another challenge arises from universities with a limited number of exchange students who have shared their reports, resulting in insufficient information for those specific universities. This limitation highlights the need for active participation from students who have already participated in exchange programs to contribute their reports and enhance the available information.

Quality of Sentence The difficulty in providing concise and meaningful summaries or information using existing Korean NLP models. The summarized sentences often sound awkward and fail to capture sufficient important content. We have chosen KoT5 as our preferred model for abstractive summarization, considering its notable performance among existing models. To overcome this limitation and enhance the quality of the generated output, the integration of KoT5 and GPT, two powerful language models, is crucial.

User Engagement The system assumes that exchange students will actively engage with the platform, participate in the community and provide feedback. The effectiveness of the system depends on user involvement and contributions to foster a supportive environment for all exchange students.

5.2 Discussions

Data Availability The system's effectiveness relies on the availability and accuracy of data, including university information, student reports and mentorship details. Limitations in data availability or data quality may impact the system's ability to provide comprehensive and up-to-date information.

Subjectivity and Bias The system's algorithms, such as sentiment analysis, may be subject to inherent subjectivity and bias. Machine learning models are trained on existing data, which may contain inherent biases or limitations. It is essential to continuously monitor and address any biases that may arise in the system to ensure fair and unbiased support for exchange students.

Cultural Context The system may not fully capture the nuanced cultural context of exchange programs and the host country. While efforts are made to provide accurate and reliable information, cultural differences and local practices may not be fully captured, requiring users to exercise judgment and seek additional information as needed.

User Adoption The system's success depends on user adoption and active engagement. If exchange students do not utilize the platform or participate in the community, the system's effectiveness in creating a supportive environment may be limited.

Technical Constraints The system's functionality and performance may be constrained by technical limitations such as server capacity, response times, or external API dependencies. These limitations may impact the system's ability to provide real-time information or handle a large volume of users simultaneously. Regular maintenance and updates are required to address any technical constraints and ensure smooth system operation.

It is crucial to consider these assumptions, constraints and limitations when implementing and utilizing the system to provide realistic expectations and to continuously work towards improving the system's effectiveness and addressing any inherent limitations.

6 Conclusion

The SKKUEXS project addresses the challenges faced by exchange students studying at Sungkyunkwan University by providing a comprehensive support system. SKKUEXS offers access to learning materials, a community platform for connection, mentorship and reliable information. It differentiates itself from previous works by taking a holistic approach and focusing on the broader needs of exchange students.

Through the utilization of Django, KoT5 and other relevant technologies, we have developed a user-friendly web application that enhances the exchange student experience. SKKUEXS enables students to easily navigate through university information, connect with local alumni and access essential resources.

Although challenges were encountered during the implementation process, we overcame them through collaboration, research and problem-solving. The empirical evaluations of our system demonstrate its effectiveness in providing support and enhancing the overall exchange student experience.

In conclusion, the SKKUEXS project aims to create a nurturing and inclusive environment for exchange students, addressing their unique challenges and fostering a sense of community. We believe that our comprehensive support system will greatly contribute to the success and well-being of exchange students at Sungkyunkwan University.

References

1. Django - `https://www.djangoproject.com`
   The web framework used for the backend development.
2. AWS Lightsail - `https://aws.amazon.com/lightsail`
   The cloud hosting service utilized for hosting the backend server.
3. Figma - `https://www.figma.com`
   The design tool utilized for creating web page prototypes and UI designs.
4. KoT5 - `https://huggingface.co/psyche/KoT5-summarization`
   The Korean language model employed for text summarization.
5. GPT-4 - `https://openai.com/gpt-4`
   Generative Pre-trained Transformer to generated the quality summary.
6. WordCloud - `https://github.com/amueller/word_cloud`
   The library used for generating visually appealing ficture to display keyword lists.
7. Google Custom Search Engine - `https://programmablesearchengine.google.com`
   The customized website search engine within a specific domain.
8. MapQuest - `https://developer.mapquest.com`
   MapQuest API provides a range of mapping and geolocation services.
9. Author, Kang, Miri: Exchange Student Program Participant's Experiences: A Case Study at a Korean University(2019). KOREA University The graduate school of education.