

A large yellow graphic resembling a dome or a stylized mountain peak, with a small yellow finial on top. The word "CONTENTS" is centered within the upper triangular part of the dome.

CONTENTS

- 01 프로젝트 개요
- 02 제공 기능
- 03 사용자 인터랙션 시나리오
- 04 데이터베이스 스키마
- 05 데이터베이스 스키마 쿼리
- 06 시스템 구조
- 07 개발환경
- 08 부족한 부분



Keyword

간편

흥미

Chat-Bot을 통해 사용자들이 평소 웹사이트를 들어가 확인해야 했던 정보들을 간편하게 제공한다.

또한, 흥미로울 수 있는 정보들을 인기 검색어와 통계치를 통해 사용자들에게 보여준다.



Chat-Bot

카카오톡 플러스 친구

학사 일정

아주대 학사 일정 제공

동아리

아주대 내 모든 동아리에 관한 정보 제공

배달음식

아주대 근처 배달음식 정보 및 요일별 통계 제공

인기검색어

역대 인기검색어 / 오늘의 인기 검색어 제공

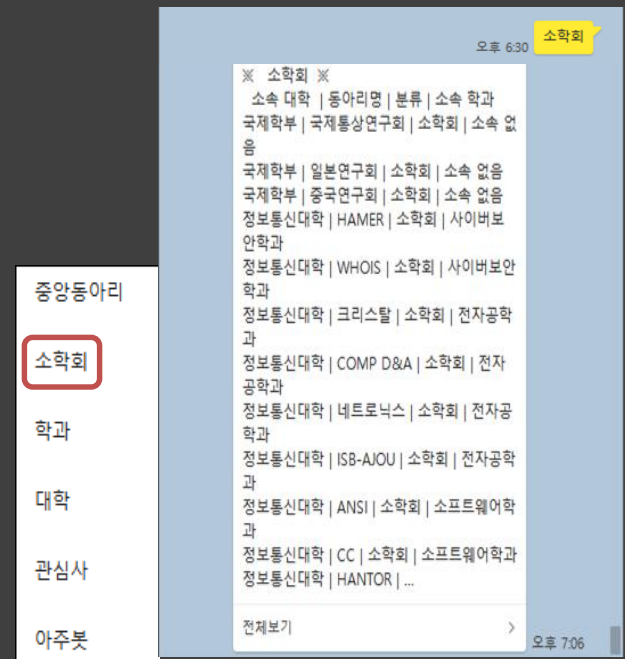
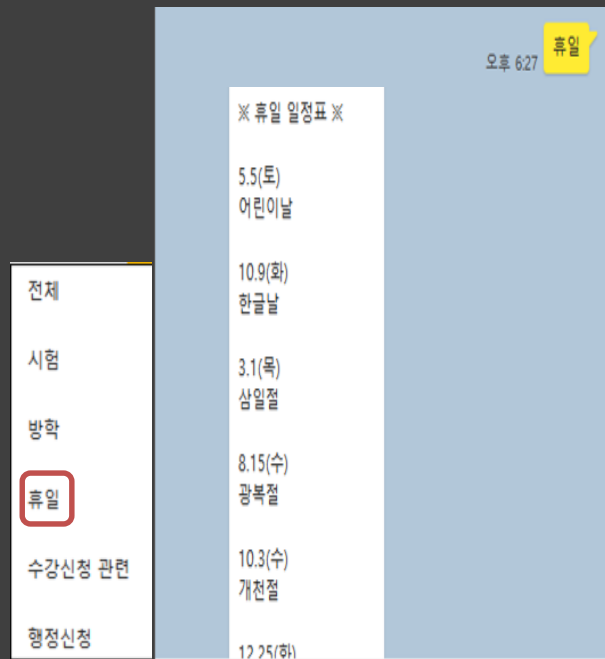
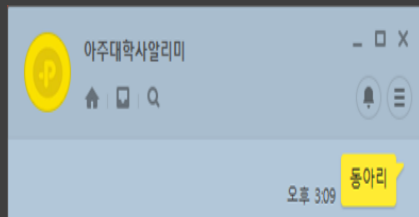
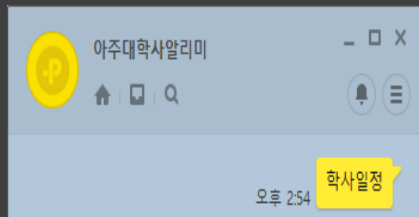
03 사용자 인터랙션 시나리오 💡

분류

학사 일정

동아리

학사일정 / 동아리는 사용자들이
확인 하기 쉽게 미리 분류를 나눠 놓음.

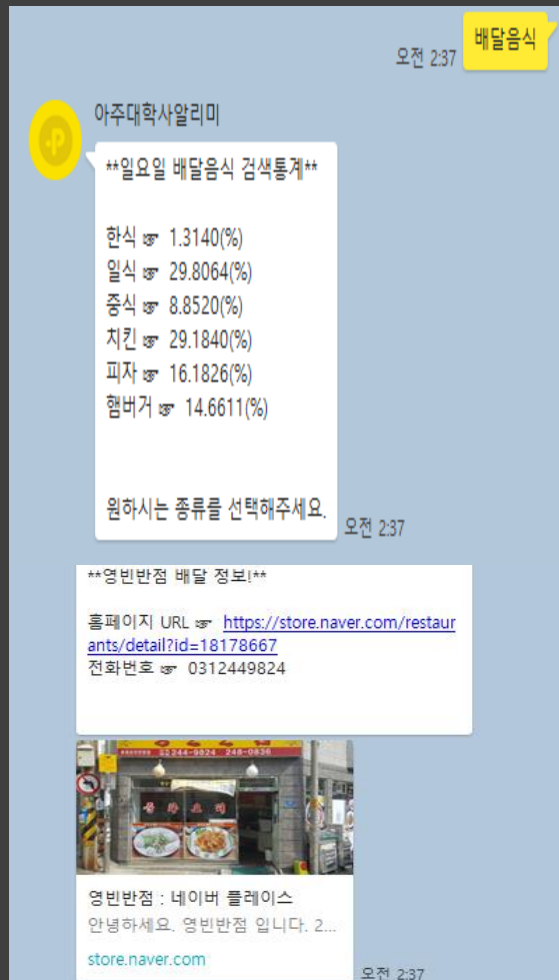


03 사용자 인터랙션 시나리오

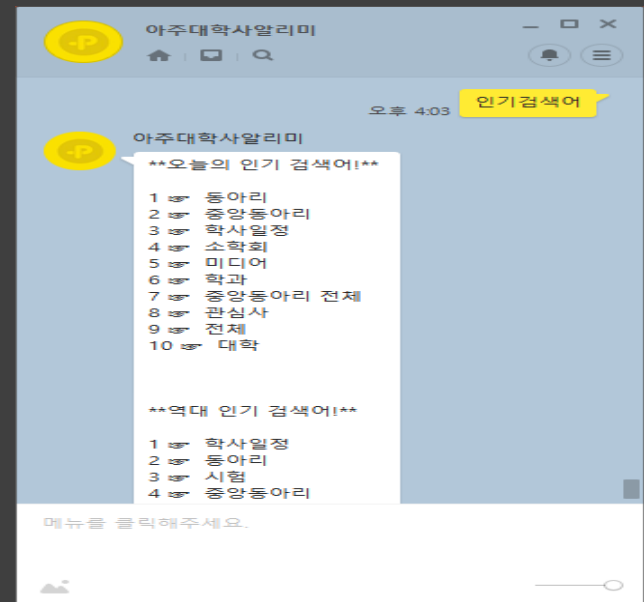
분류



배달 음식



인기 검색어



1. 배달 음식
요일별 통계
제공

2. 배달 음식
정보 제공

1. 사용자들의 선호
키워드를 알려줌

2. 아주봇 이용 시 참고.

03 사용자 인터랙션 시나리오💡

분류

아주봇

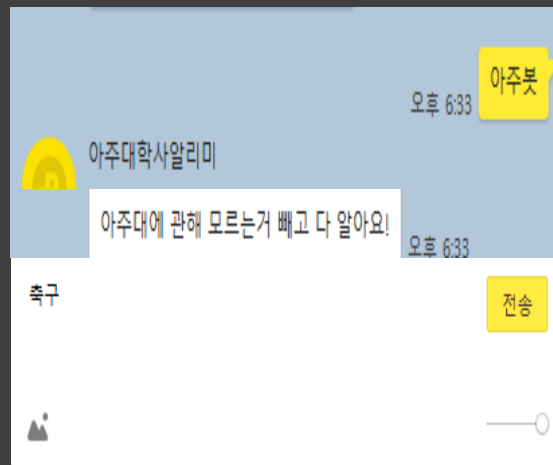
오늘의 인기검색어

학사일정

배달음식

동아리

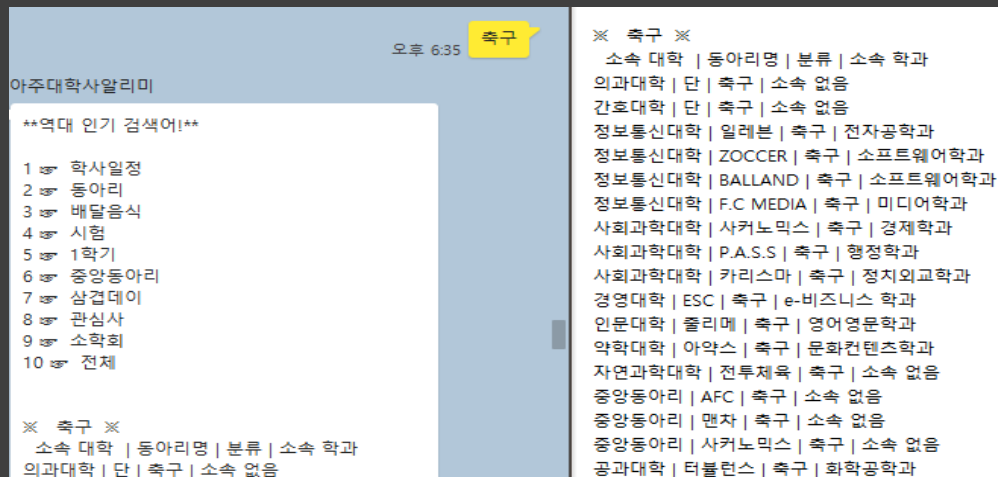
아주봇



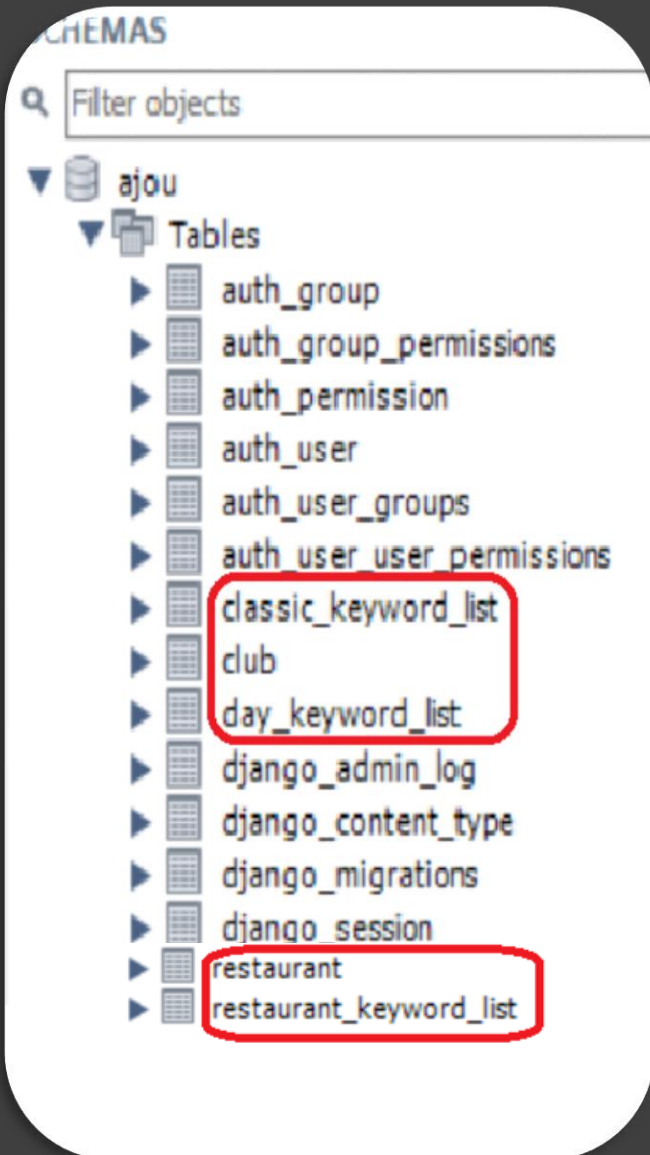
1. 임의로 분류해 놓은 키워드로 찾아가기
힘듦
2. 원하는 키워드가
없음.

아주봇에 원하는
키워드를 검색

해당 키워드에
해당하는 정보들을
확인할 수 있다.



04 데이터베이스 스키마



ajou schema

1. haksa table

- 학사일정 관련 날짜, 정보등을 포함하는 테이블

2. club table

- 동아리 관련 소속대학, 전공, 카테고리, 동아리명을 포함하는 테이블

3. classic_keyword_list table

- 종합 인기 검색어 관련 검색어, 검색 횟수를 포함하는 테이블

4. day_keyword_list table

- 하루 인기 검색어 관련 검색어, 검색 횟수를 포함하는 테이블

5. restaurant table

- 근처 배달 음식점 관련 정보들을 포함하는 테이블

6. restaurant_keywordd_list table

- 음식점 category에 따라 요일별로 검색한 횟수를 update해 요일별로 rate를 보여 주기 위해 사용되는 테이블

04 데이터베이스 스키마



haksa

club

classic_keyword_list

idhaksa	date	info
1	2017.12.27(수)~2018.1.3(수)	복학생 전과신청
2	2017.12.26(화)~2018.1.18(목)	1학기 1차 복학 신청 기한
3	2018.1.1(월)	신청
4	1.22(월)~1.24(수)	1학기 예비(책가방식) 수강신청
5	1.25(목)~2.5(월)	1학기 최종 복학 신청 기한
6	2.7(수)~2.13(화)	수강신청(재학생,복학생,재입학생)
7	2.15(목)~2.17(토)	설 연휴
8	2.21(수)	평일상 오리엔테이션
9	2.22(목)	전기 학위수여식
10	2.22(목)~2.28(수)	1학기 등록
78	2019.2.4(월)~2.6(수)	설 연휴
79	2019.2.22(금)	전기 학위수여식
80	3.2(금)	1학기 개강
81	9.3(월)	2학기 개강
NULL	NULL	NULL

idClub	club_univ	club_name	club_category	club_major
1	의과대학	SIX Lines	음악	소속 없음
2	의과대학	텃밭	연극	소속 없음
3	의과대학	CMF	종교	소속 없음
4	의과대학	☺, ☹ 사랑	봉사	소속 없음
5	의과대학	Medic Chamber	오케스트라	소속 없음
6	의과대학	ARTISTS	미술	소속 없음
7	의과대학	ORBIT	사진	소속 없음
8	의과대학	메디콤	컴퓨터	소속 없음
9	의과대학	아메바	농구	소속 없음
10	의과대학	Paovrus	독서토론	소속 없음
252	공과대학	E-CUBIC	소학회	건축학과
253	공과대학	FACE	소학회	건축학과
254	공과대학	Studio I.S.M	소학회	건축학과
255	공과대학	UD2	소학회	건축학과
NULL	NULL	NULL	NULL	NULL

idclassic_keyword_list	keyword	num
116	시험	15
117	2학기	1
118	학사일정	35
124	전체	3
133	동아리	28
134	학과	2
135	경영학과	0
137	대학	1
138	공과대학	0
146	행정신청	1

day_keyword_list

restaurant

restaurant_keyword_list

idday_keyword_list	keyword	num
47	시험	3
48	2학기	0
49	학사일정	6
55	전체	0
64	동아리	3
65	학과	0
66	경영학과	0
68	대학	0
69	공과대학	0
77	행정신청	0

idrestaurant	restaurant	category	location	phone_number
1	삼겹데이	한식	https://...	0312156717
2	호세아오리바베큐	한식	https://...	0312145293
3	곱창	한식	https://...	0312540880
4	황금배슬핌닭	한식	https://...	0312142778
5	마스터통삼겹	한식	https://...	0312147714
6	짜장마을	중식	https://...	0312245563
7	영빈반점	중식	https://...	0312449824
8	리할	중식	https://...	0312111003
9	궁자이나	중식	https://...	0312247745
10	짬뽕이좋아	중식	https://...	0312567951
28	롯데리아	햄버거	https://...	0312151294
29	KFC	햄버거	https://...	0312170337
30	맥도날드	햄버거	https://...	07072090922
31	버거킹	햄버거	https://...	0312152256
NULL	NULL	NULL	NULL	NULL

idrestaurant_keyword_list	category	monday	tuesday	wednesday	thursday	friday
1	한식	NULL	NULL	NULL	NULL	NULL
2	중식	NULL	NULL	NULL	NULL	NULL
3	일식	NULL	NULL	NULL	NULL	NULL
4	치킨	NULL	NULL	NULL	NULL	NULL
5	피자	NULL	NULL	NULL	NULL	NULL
6	햄버거	NULL	NULL	NULL	NULL	NULL

05 데이터베이스 스키마 쿼리



스키마(테이블) 생성 Query

DDL for ajou.haksa

```
1 CREATE TABLE `haksa` (  
2   `idhaksa` int(10) unsigned NOT NULL AUTO_INCREMENT,  
3   `date` longtext,  
4   `info` longtext,  
5   PRIMARY KEY (`idhaksa`)  
6 ) ENGINE=InnoDB AUTO_INCREMENT=82 DEFAULT CHARSET=utf8
```

DDL for ajou.club

```
1 CREATE TABLE `club` (  
2   `idClub` int(11) unsigned NOT NULL AUTO_INCREMENT,  
3   `club_univ` varchar(45) DEFAULT NULL,  
4   `club_name` varchar(45) DEFAULT NULL,  
5   `club_category` varchar(45) DEFAULT NULL,  
6   `club_major` varchar(45) DEFAULT NULL,  
7   PRIMARY KEY (`idClub`)  
8 ) ENGINE=InnoDB AUTO_INCREMENT=256 DEFAULT CHARSET=utf8
```

DDL for ajou.classic_keyword_list

```
1 CREATE TABLE `classic_keyword_list` (  
2   `idclassic_keyword_list` int(11) NOT NULL AUTO_INCREMENT,  
3   `keyword` varchar(45) NOT NULL,  
4   `num` int(10) NOT NULL DEFAULT '0',  
5   PRIMARY KEY (`idclassic_keyword_list`),  
6   UNIQUE KEY `keyword_UNIQUE` (`keyword`)  
7 ) ENGINE=InnoDB AUTO_INCREMENT=300 DEFAULT CHARSET=utf8
```

DDL for ajou.day_keyword_list

```
1 CREATE TABLE `day_keyword_list` (  
2   `idday_keyword_list` int(11) NOT NULL AUTO_INCREMENT,  
3   `keyword` varchar(45) COLLATE utf8_bin NOT NULL,  
4   `num` int(10) NOT NULL DEFAULT '0',  
5   PRIMARY KEY (`idday_keyword_list`),  
6   UNIQUE KEY `keyword_UNIQUE` (`keyword`)  
7 ) ENGINE=InnoDB AUTO_INCREMENT=231 DEFAULT CHARSET=utf8 COLLATE=utf8_bin
```

DDL for ajou.restaurant

```
1 CREATE TABLE `restaurant` (  
2   `idrestaurant` int(10) unsigned NOT NULL AUTO_INCREMENT,  
3   `restaurant` varchar(45) DEFAULT NULL,  
4   `category` varchar(44) DEFAULT NULL,  
5   `location` longtext,  
6   `phone_number` varchar(45) DEFAULT NULL,  
7   PRIMARY KEY (`idrestaurant`)  
8 ) ENGINE=InnoDB AUTO_INCREMENT=117 DEFAULT CHARSET=utf8
```

DDL for ajou.restaurant_keyword_list

```
1 CREATE TABLE `restaurant_keyword_list` (  
2   `idrestaurant_keyword_list` int(10) unsigned NOT NULL AUTO_INCREMENT,  
3   `category` varchar(45) DEFAULT NULL,  
4   `monday` int(11) DEFAULT NULL,  
5   `tuesday` int(11) DEFAULT NULL,  
6   `wednesday` int(11) DEFAULT NULL,  
7   `thursday` int(11) DEFAULT NULL,  
8   `friday` int(11) DEFAULT NULL,  
9   PRIMARY KEY (`idrestaurant_keyword_list`)  
10 ) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8
```

크롤링을 이용한 tuple 생성 Query

```
7 conn = pymysql.connect(  
8   host="localhost",  
9   user = "root",  
10  passwd = "1q2w3e4r",  
11  db = "ajou",  
12  charset = "utf8",  
13  use_unicode=True  
14 )  
15 cur = conn.cursor()  
16  
17 url = "http://www.ajou.ac.kr/new/life/schedule_haksa.jsp"  
18 html = requests.get(url).text  
19  
20 soup = BeautifulSoup(html, "html.parser")  
21 crol = soup.select('td[class=left_pd]')  
22  
23 i=0  
24  
25 for content in crol:  
26     check = i % 2  
27     if check == 0 :  
28         data = content.text  
29     else :  
30         data1 = content.text  
31     sql = "INSERT INTO haksa(date, info) VALUES ('"+data+"','"+data1+"')"  
32     cur.execute(sql)  
33     i += 1  
34  
35 conn.commit()  
36  
37 conn.close()
```

파일에서 사용한 tuple 생성 Query

```
14 cur = conn.cursor()  
15  
16 cur.execute("INSERT INTO club(idclub, club_univ, club_name, club_category) VALUES (1, '외국대학', 'SIX Lines', '음악')")  
17  
18 cur.execute("INSERT INTO club(idclub, club_univ, club_name, club_category) VALUES (2, '외국대학', '뽕뽕', '연극')")  
19  
20 cur.execute("INSERT INTO club(idclub, club_univ, club_name, club_category) VALUES (3, '외국대학', 'OMP', '종교')")  
21  
22 cur.execute("INSERT INTO club(idclub, club_univ, club_name, club_category) VALUES (4, '외국대학', 's.o 사랑', '종교')")  
23  
24 cur.execute("INSERT INTO club(idclub, club_univ, club_name, club_category) VALUES (5, '외국대학', 'Medic Chamber', '오케스트라')")  
25  
26 cur.execute("INSERT INTO club(idclub, club_univ, club_name, club_category) VALUES (6, '외국대학', 'ARTISTS', '미술')")  
27  
28 cur.execute("INSERT INTO club(idclub, club_univ, club_name, club_category) VALUES (7, '외국대학', 'ORBIT', '사진')")  
29  
30 cur.execute("INSERT INTO club(idclub, club_univ, club_name, club_category) VALUES (8, '외국대학', '헤디콜', '음류')")  
31  
32 cur.execute("INSERT INTO club(idclub, club_univ, club_name, club_category) VALUES (9, '외국대학', '아미바', '농구')")  
33  
34 cur.execute("INSERT INTO club(idclub, club_univ, club_name, club_category) VALUES (10, '외국대학', 'Papyrus', '독서토론')")  
35  
36 cur.execute("INSERT INTO club(idclub, club_univ, club_name, club_category) VALUES (11, '외국대학', '광치기', '사회과학')")  
37  
38 cur.execute("INSERT INTO club(idclub, club_univ, club_name, club_category) VALUES (12, '외국대학', '단', '축구')")  
39  
40 cur.execute("INSERT INTO club(idclub, club_univ, club_name, club_category) VALUES (13, '외국대학', 'COON', '문화생활')")  
41  
42 cur.execute("INSERT INTO club(idclub, club_univ, club_name, club_category) VALUES (14, '외국대학', '아카리', '종교')")  
43  
44 cur.execute("INSERT INTO club(idclub, club_univ, club_name, club_category) VALUES (15, '진보대학', 'SIX Lines', '음악')")
```

```
sql = \  
"select ((select sum(" + day + ") from restaurant_keyword_list where category = '한식') / " \  
"(select sum(" + day + ") from restaurant_keyword_list) *100)as korea," \  
\  
"((select sum(" + day + ") from restaurant_keyword_list where category = '일식') / " \  
"(select sum(" + day + ") from restaurant_keyword_list) *100)as japan," \  
\  
"((select sum(" + day + ") from restaurant_keyword_list where category = '중식') / " \  
"(select sum(" + day + ") from restaurant_keyword_list) *100)as china," \  
\  
"((select sum(" + day + ") from restaurant_keyword_list where category = '치킨') / " \  
"(select sum(" + day + ") from restaurant_keyword_list) *100)as chinken," \  
\  
"((select sum(" + day + ") from restaurant_keyword_list where category = '피자') / " \  
"(select sum(" + day + ") from restaurant_keyword_list) *100)as pizza," \  
\  
"((select sum(" + day + ") from restaurant_keyword_list where category = '햄버거') / " \  
"(select sum(" + day + ") from restaurant_keyword_list) *100)as hamburger from restaurant_keyword
```

요일 별 통계를 내기 위한 쿼리

05 데이터베이스 스키마 쿼리

데이터 베이스 서비스 기능 관련 연계 Query

```
199 #휴일관련 일정
200 elif info in ('휴일', '공휴일', '쉬는날'):
201     result = '※' + info + ' 일정표 ※'
202     querys = \
203         "SELECT " + sentence1 + ", " + sentence2 + \
204         " from " + sentence3 + \
205         " where " + sentence2 + " LIKE " + "%" + "날" + "%" + \
206         " UNION ALL " + \
207         "SELECT " + sentence1 + ", " + sentence2 + \
208         " from " + sentence3 + \
209         " where " + sentence2 + " LIKE " + "%" + "절" + "%" + \
210         " and " + "not info like " + "%" + "계절" + "%" + \
211         " UNION ALL " + \
212         "SELECT " + sentence1 + ", " + sentence2 + \
213         " from " + sentence3 + \
214         " where " + sentence2 + " LIKE " + "%" + "일" + "%" + \
215         " and " + "not info like " + "%" + "수업" + "%" + \
216         " and " + "not info like " + "%" + "일절" + "%" + \
217         " and " + "not info like " + "%" + "보강" + "%" + \
218         " and " + "not info like " + "%" + "학위" + "%" + \
219         " UNION ALL " + \
220         "SELECT " + sentence1 + ", " + sentence2 + \
221         " from " + sentence3 + \
222         " where " + sentence2 + " LIKE " + "%" + "연휴" + "%" + \
223         " UNION ALL " + \
224         "SELECT " + sentence1 + ", " + sentence2 + \
225         " from " + sentence3 + \
226         " where " + sentence2 + " LIKE " + "%" + "선거" + "%"
```

해당 query문 예시1

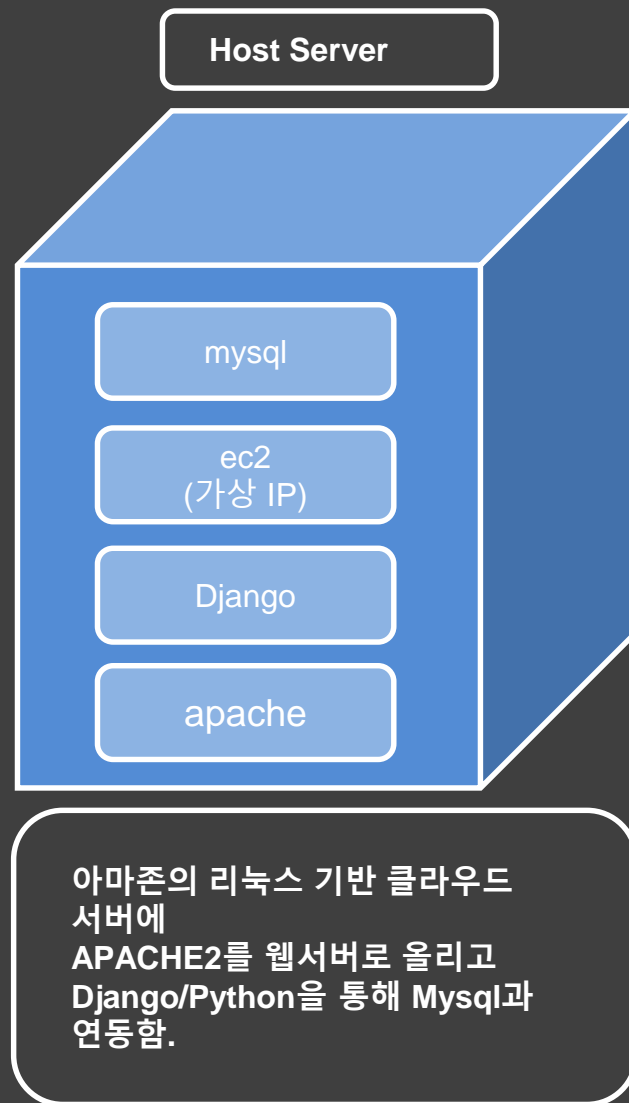
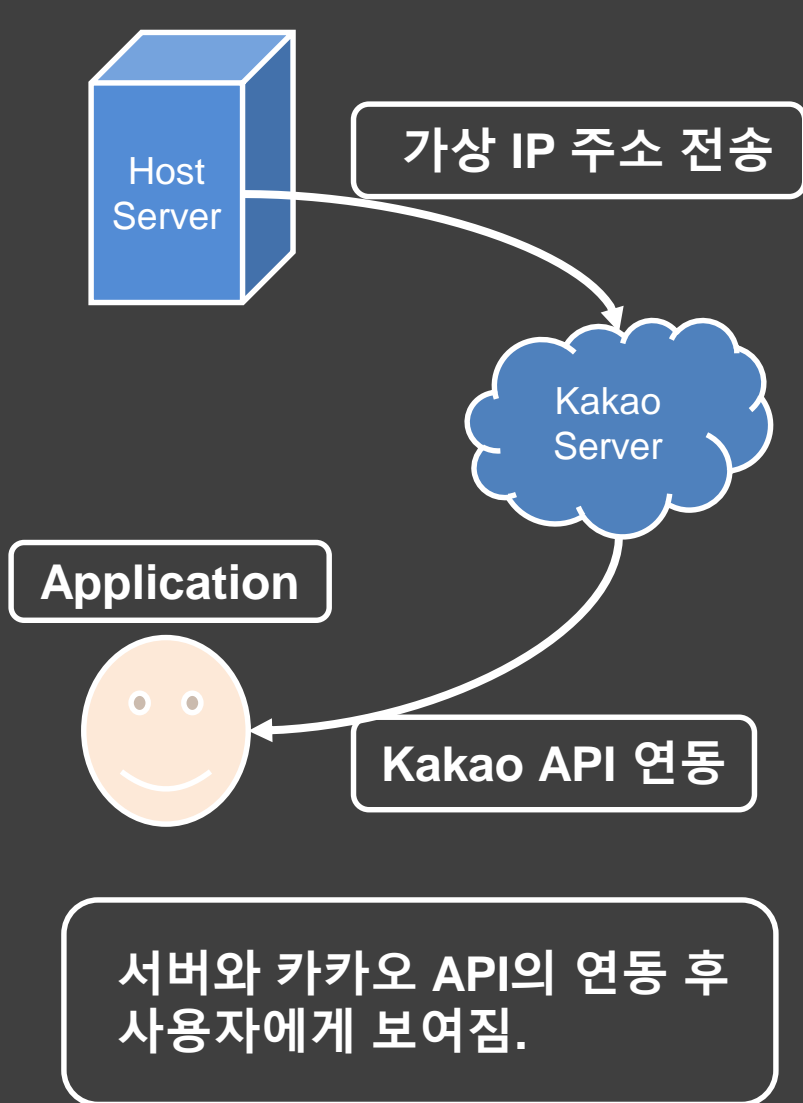
기본적으로 데이터베이스에서 사용자가 원하는 정보를 select하는 Query문

```
39 #사의 10개 키워드 가져오
40 sql = 'select keyword from day_keyword_list order by num desc limit 10'
41 cur.execute(sql)
42 total = cur.fetchall()
43 conn.close()
44
45 result = '**오늘의 인기 검색어!** \n\n'
46 num = 1
47 for row in range(0, len(total)):
48     result += str(num) + ' 🐼 ' + total[row][0] + '\n|'
49     num += 1
50
51 return result
52
53 def init() :
54     conn = pymysql.connect(
55         host="localhost",
56         user = "root",
57         passwd = "1q2w3e4r",
58         db = "ajou",
59         charset = "utf8",
60         use_unicode=True
61     )
62     nowHour = datetime.today().hour
63
64     cur = conn.cursor()
65     if nowHour == 0 :
66         sql = 'SET SQL_SAFE_UPDATES =0'
67         cur.execute(sql)
68         sql = 'UPDATE day_keyword_list SET num = default'
69         cur.execute(sql)
70         conn.commit()
```

실시간 검색어 순위 관련 Query

1. 검색한 횟수를 내림차순으로 정렬한 뒤 첫 번째 행부터 10번째 행까지 keyword를 select하는 query문
2. Safe Update 모드를 해제하는 query문 과 하루 검색어 순위를 출력하기 위해 검색 횟수를 default(0으로 설정)으로 초기화 시켜준다.

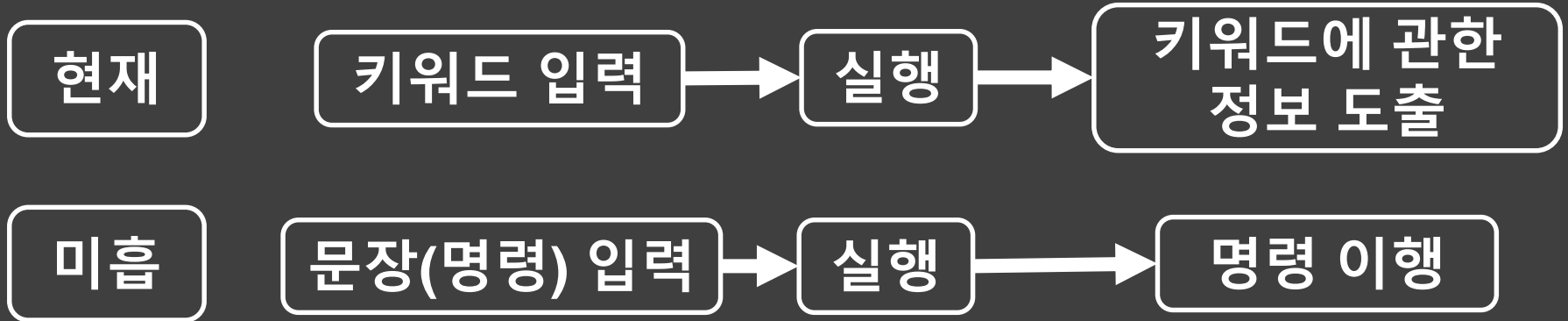
06 시스템 구조 💡



07 개발 환경 💡



필요한 부분



실제로 챗봇에 있는 서비스들이 상용화가 되기 위해서는 예를 들어 배달음식의 경우 정보만을 보여주는 것이 아니라 실제로 챗봇에게 명령을 내리면 주문까지 이루어져야 한다. 현재의 챗봇은 단지 정보를 보여주는 정도의 역할밖에 할 수 없다.



END

