

여기어째...

2020.08.14 ~ 2020.09.03

융복합 프로젝트형 클라우드(MSA) 서비스 개발
인터페이스 개발 프로젝트

목차

01 서론

- 1-1 주제 선정 배경
- 1-2 팀 구성 및 역할
- 1-3 시스템 아키텍처
- 1-4 개발에 사용된 기술
- 1-5 세부수행 간트표

02 Front-End

- 2-1 기능 - 시각화
- 2-2 UI/UX 개선 (1)
- 2-3 UI/UX 개선 (2)

03 Back-End

- 3-1 데이터 수집
- 3-2 DB 구성도
- 3-3 Django Rest Framework
- 3-4 네이버 뉴스 웹크롤링
- 3-5 Heroku에 Django 앱을 배포

04 Conclusion

- 4-1 개선점
- 4-2 느낀점

1-1 주제 선정 배경

집값이 높을 수록 안전한 동네일까?

[2014국감]서울 강남 3구 체감안전도 높아

최종수정 2014.10.20 08:40 기사입력 2014.10.20 08:40

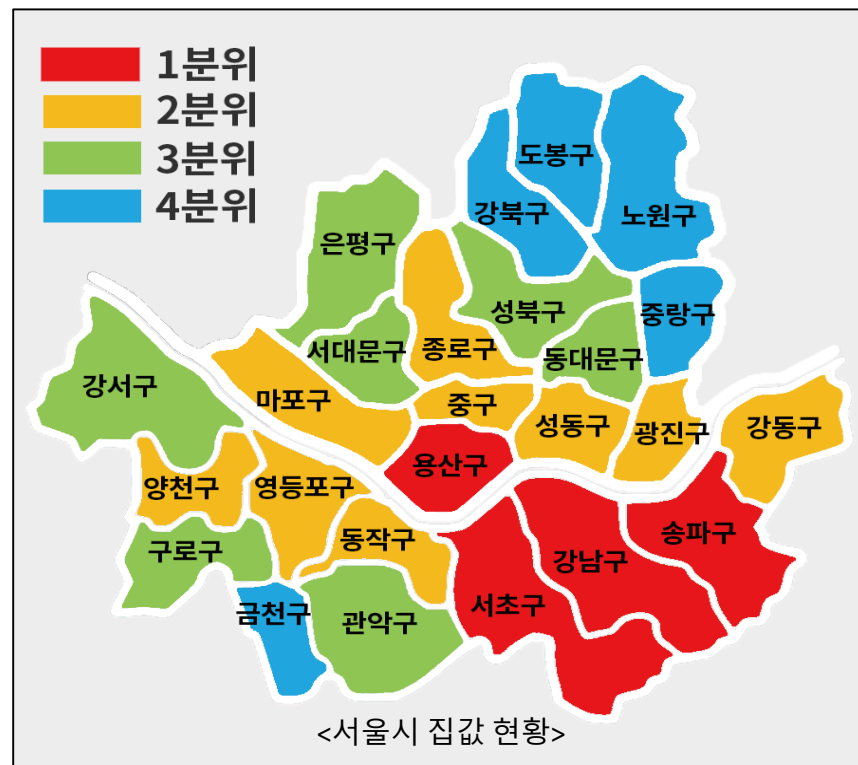


가+

[아시아경제 김재연 기자] 상대적으로 부유층이 많이 사는 서울 강남 3구(강남·서초·송파구)의 체감 안전도가 높은 것으로 나타났다.

강기윤 의원이 서울경찰청으로 받은 자료에 따르면 2014년 상반기 경찰서별 체감안전도 조사 결과 체감안전도 상위 10위에 강남·서초·용산·서초·송파·방배 경찰서가 이름을 올렸다.

반면, 체감안전도가 가장 낮은 곳은 관악서로 60.0점이었으며, 금천서와 성북서가 62.6점으로 뒤를 이었다.



재난/범죄/토지 기반 등의 정보를 고려하여
안전도를 확인할 수 있는 **웹사이트**를 구축하자!

1-2 팀 구성 및 역할



팀장 황동윤

React를 이용한
Front-End 개발

Github pages를 통해 배포



강륜화

DB 구축 및 정규화

Django Model 설계

PostgreSQL 연동 세팅



김유진

DRF API 서버 구축

Heroku를 통해 배포

News 크롤링 및 업데이트

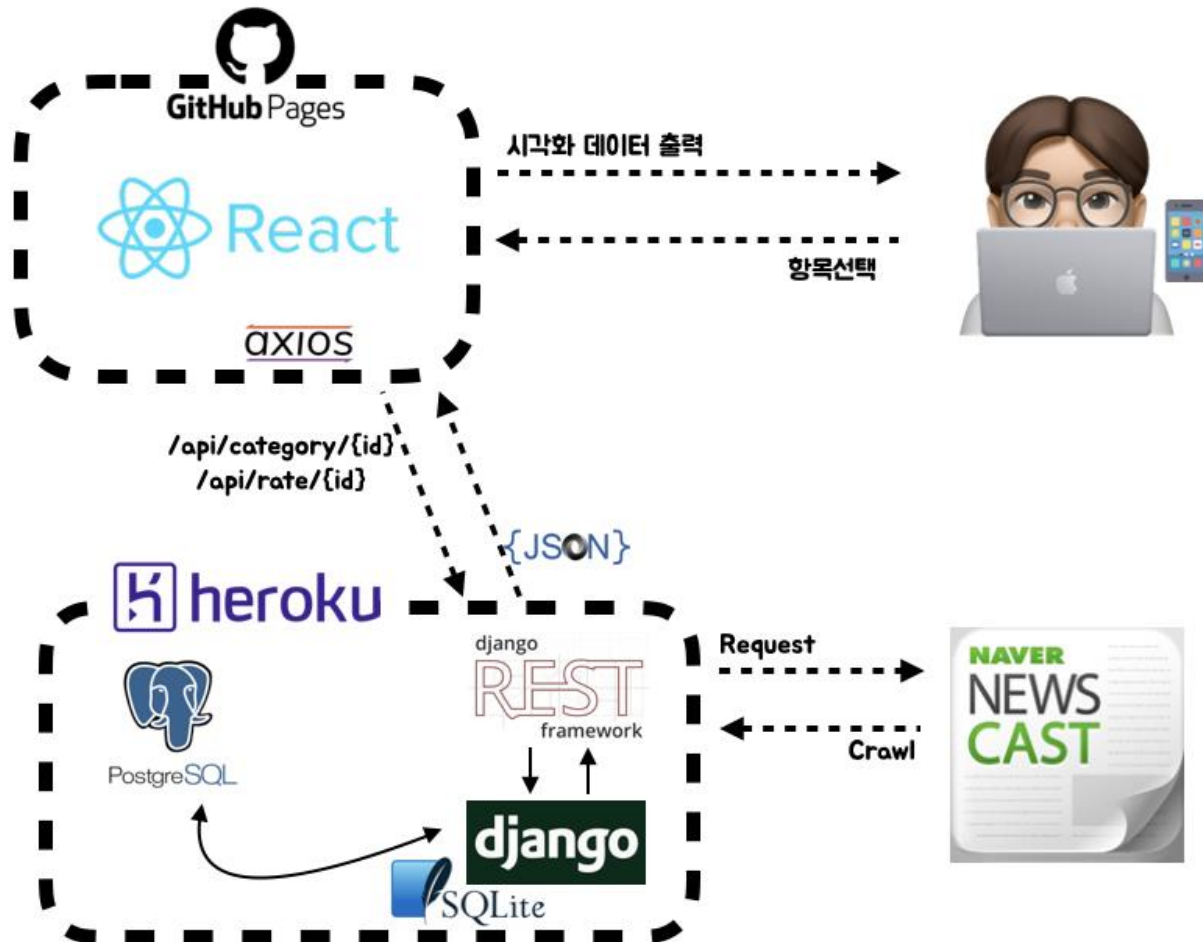


박진우

공공데이터 수집

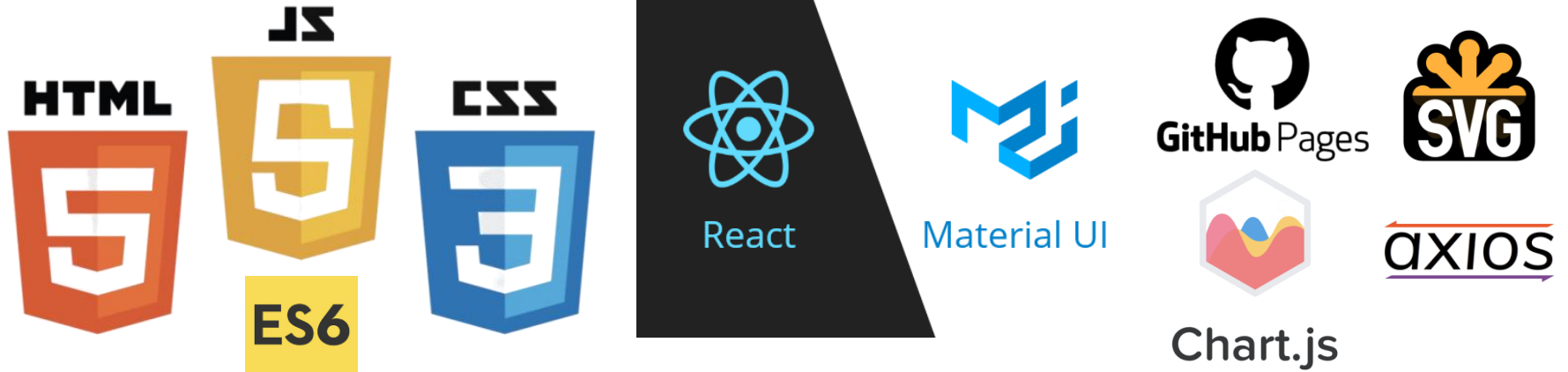
Pandas를 이용한
데이터 전처리

1-3 시스템 아키텍처



1-4 개발에 사용된 기술

Front-End



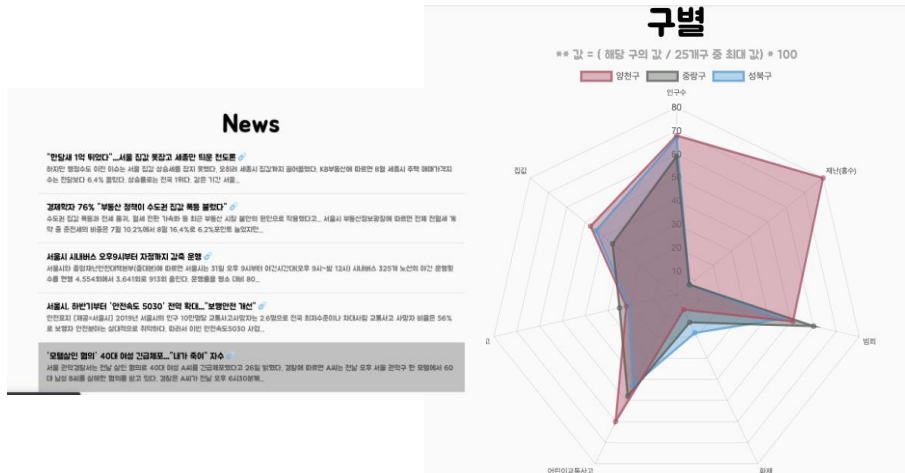
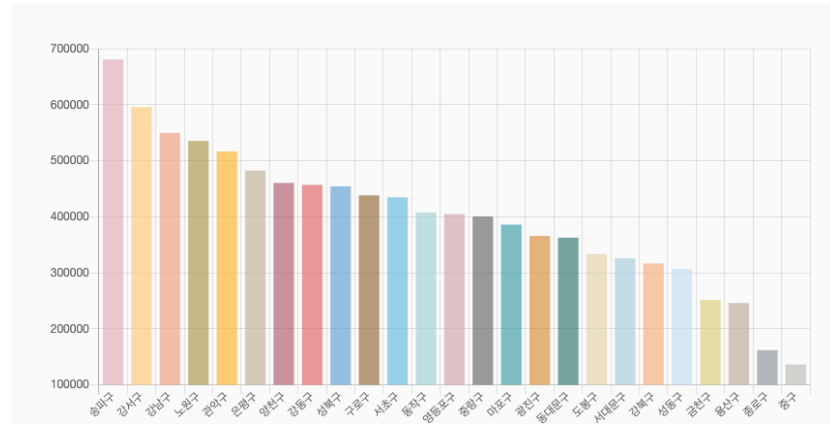
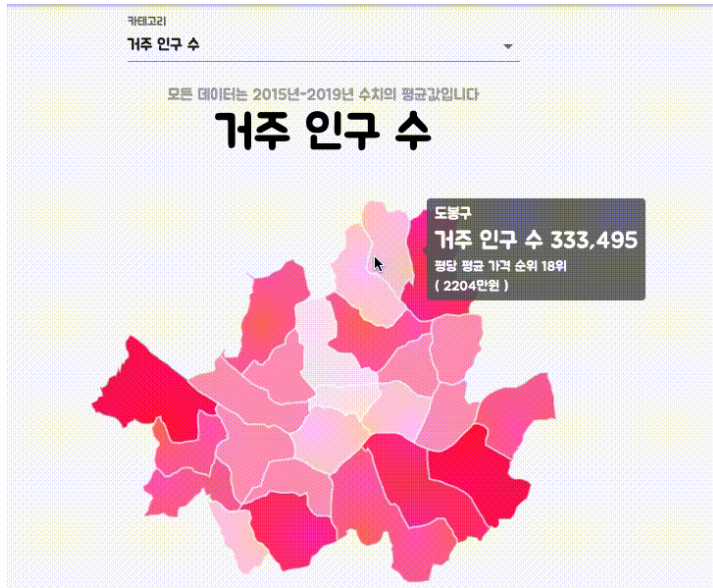
Back-End



1-5 세부 수행 간트표

개발 내용		기간																											
		8월																								9월			
		10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3			
계획	기획안 작성																												
Front end	설계																												
	구현																												
	보완																												
	검증																												
	배포																												
Back end	데이터 수집																												
	데이터 전처리																												
	DB 구축 및 정규화																												
	DRF 서버 구축																												
	Heroku 배포																												
	postgresql 연동																												
	뉴스 크롤러 제작																												
최종	완성 및 시뮬레이션																												
	포트폴리오 작성																												

2-1 기능 - 시각화



순 위	지 역 구	가 구 수	남(전 체)	여(전 체)	거 주 인 구 수	남(외국 인)	여(외국 인)
1	송파구	279,301	329,058	351,825	680,883	3,199	3,589
2	강서구	263,645	288,134	307,569	595,703	3,049	3,352
3	강남구	233,624	263,163	286,735	549,898	2,509	2,585
4	노원구	217,148	258,696	276,799	535,495	1,970	2,488
5	관악구	270,760	258,982	257,680	516,662	8,239	9,049
6	은평구	208,209	231,953	250,556	482,509	1,806	2,684
7	양천구	177,436	226,109	234,423	460,532	1,868	2,325
8	강동구	190,019	225,226	231,816	457,042	2,037	2,359

간격 축소

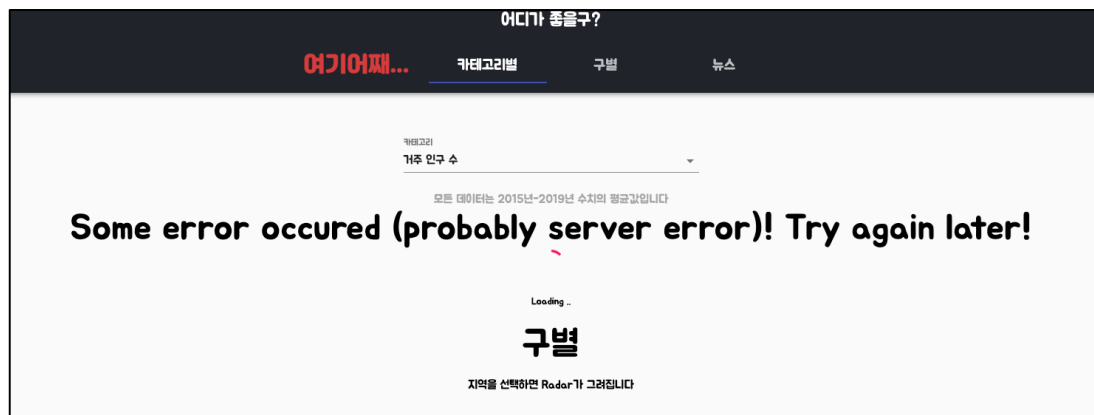
2-2 UI/UX 개선 (1)



Loading Animation

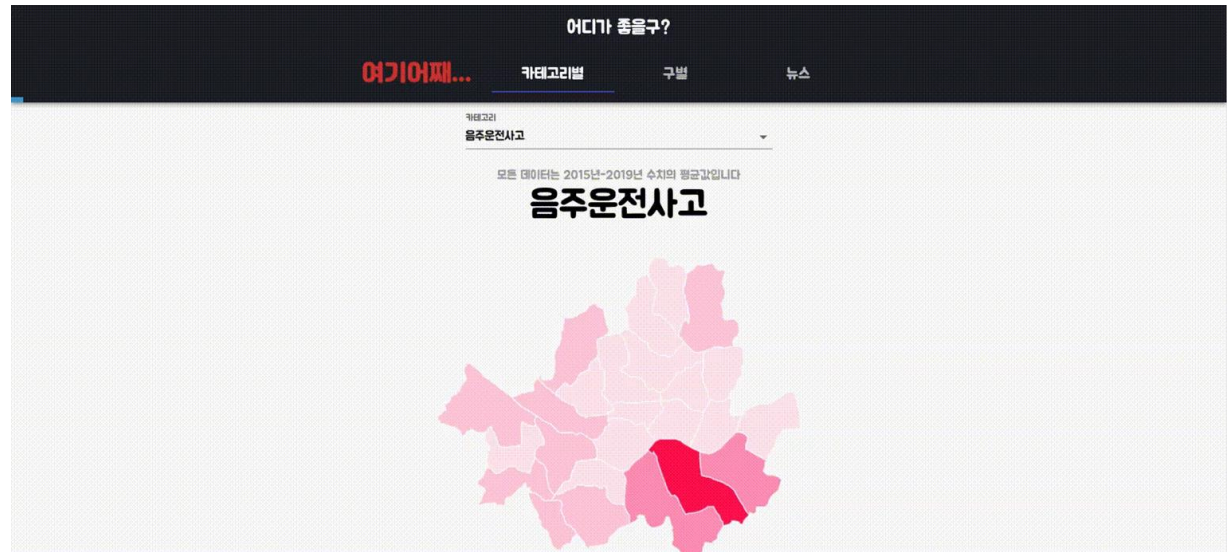


ToolTip



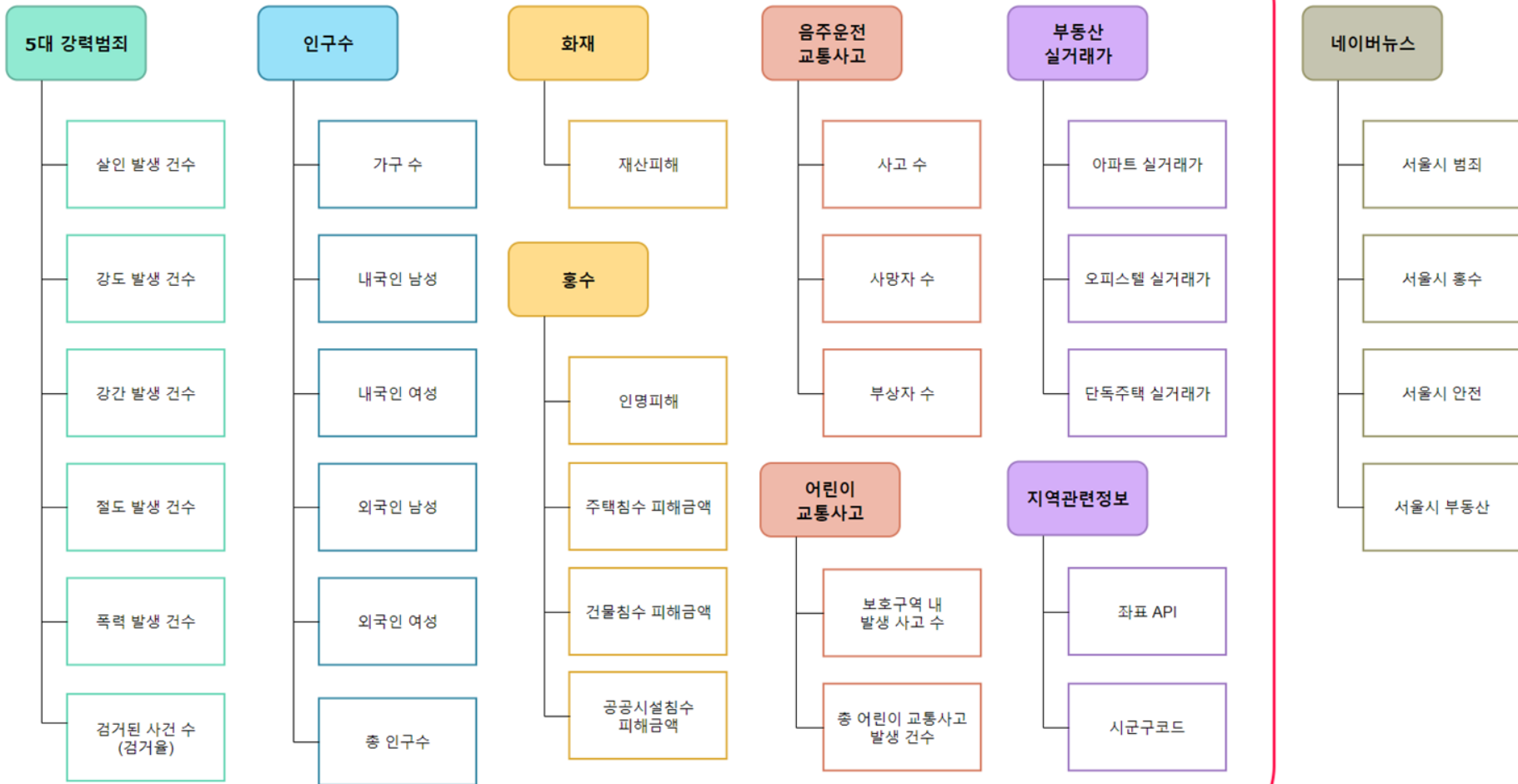
Error handling

2-3 UI/UX 개선 (2)

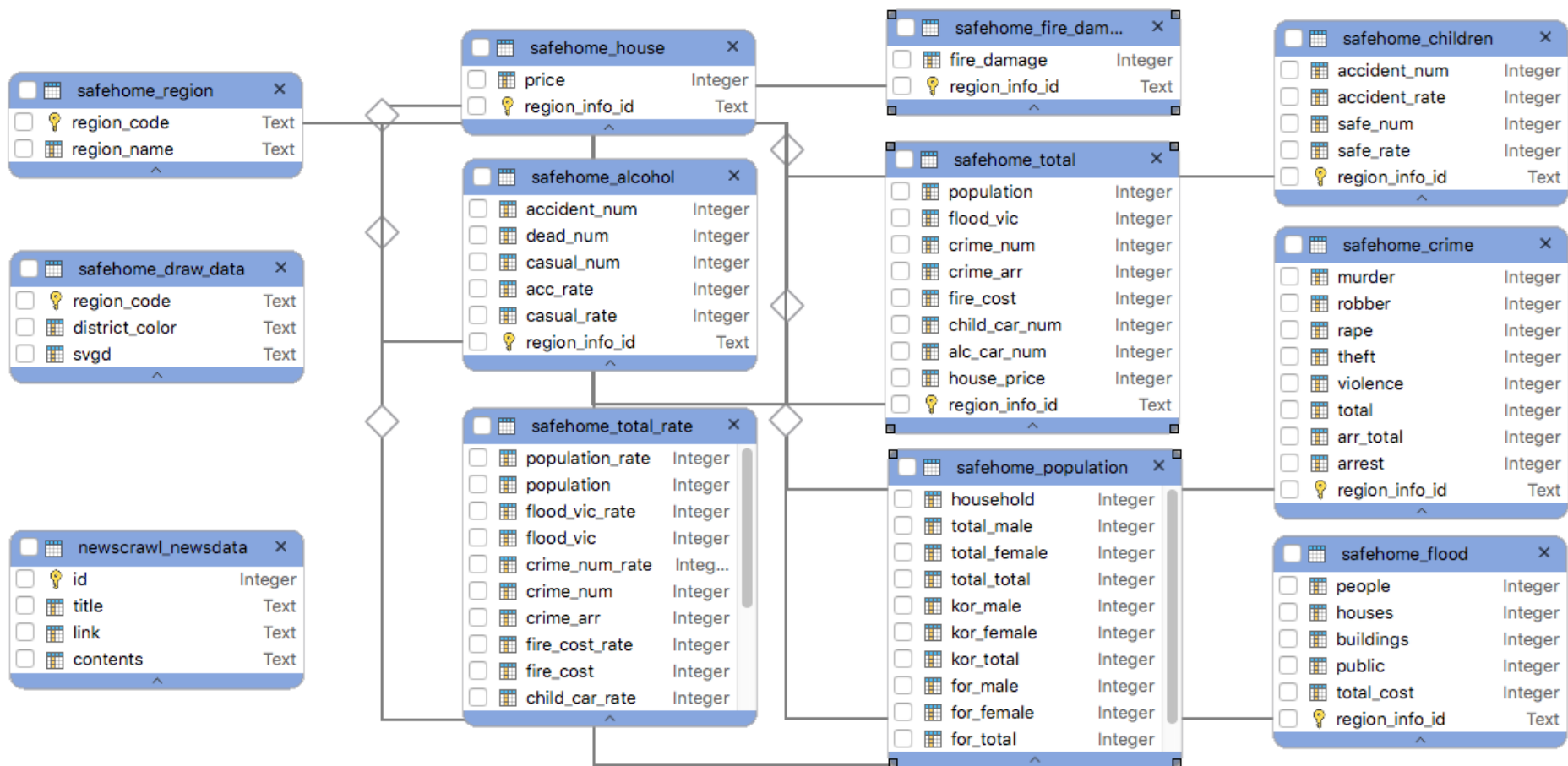


3-1 데이터 수집

서울시 공공데이터



3-2 DB 구성도



3-3 Django Rest Framework

```
from django.db import models

class Region(models.Model):
    # 시군구코드
    region_code = models.CharField(primary_key=True, max_length=30, default='')
    # 자치구 이름
    region_name = models.CharField(max_length=30, default='')

    def __str__(self):
        return self.region_code

class Alcohol(models.Model):
    # 자치구 정보
    region_info = models.ForeignKey(Region, on_delete=models.CASCADE, default='', primary_key=True)
    # 음주운전 교통사고 발생건수
    accident_num = models.IntegerField(default=0)
    # 사망자수
    dead_num = models.IntegerField(default=0)
    # 부상자수
    casual_num = models.IntegerField(default=0)
    # 음주운전 사고 발생 비율
    acc_rate = models.IntegerField(default=0)
    # 부상자 비율
    casual_rate = models.IntegerField(default=0)

    def __str__(self):
        return self.region_info
```

```
from rest_framework import serializers
from .models import Region, Alcohol, Children, Crime, Fire_damage, Flood, House, Population, \
    Total, Total_rate, Draw_data

class RegionSerializer(serializers.ModelSerializer):
    class Meta:
        fields = (
            'region_code',
            'region_name',
        )
        model = Region

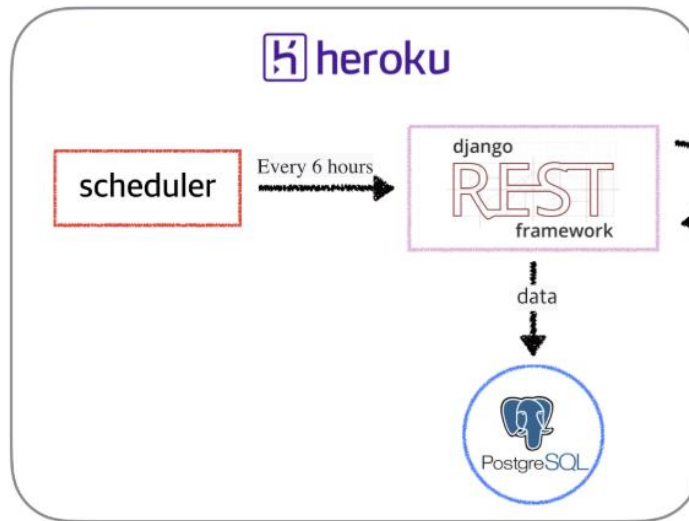
class AlcoholSerializer(serializers.ModelSerializer):
    region_info = RegionSerializer(read_only=True)
    class Meta:
        fields = (
            'region_info',
            'accident_num',
            'dead_num',
            'casual_num',
            'acc_rate',
            'casual_rate',
        )
        model = Alcohol
```

Django Rest Framework와 MariaDB, SQLite3 과의 연동을 위한 setting 작업

- Database 연동 모델 생성
- MariaDB를 local에서 작업할 수 있도록 SQLite3로 migrate하면서 보다 효율적으로 모델 재구성

- searlizer.py 생성
- queryset과 model instance를 json으로 렌더링하기 위함
- Django 모델과 매핑

3-4 네이버 뉴스 웹크롤링



Scheduler process overview

Function implementation

```
#Procfile
release: python3 manage.py migrate
clock: python3 parser_scheduler.py # clock 기능을 사용
web: gunicorn mytest.wsgi
```

```
# parser_scheduler.py
from apscheduler.schedulers.blocking import BlockingScheduler
import datetime
import pytz

import parser as pr

sched = BlockingScheduler()
local_tz = pytz.timezone('Asia/Seoul')

@sched.scheduled_job('interval', minutes=60*6) # 6시간
def timed_job():
    print('This job is run every six hour.')

    pr.del_save_data() # news update

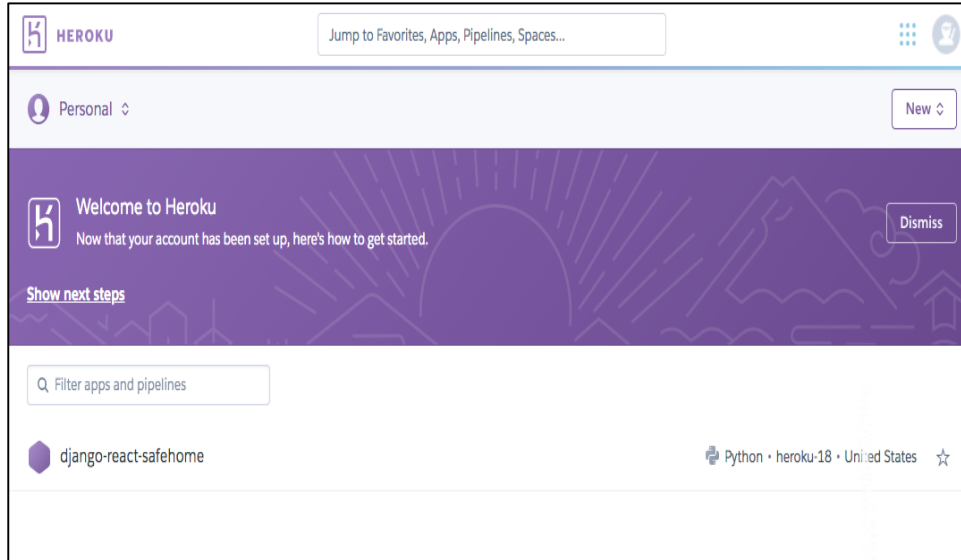
sched.start()
```

Web Crawling

- 5개 검색어 선정 후, BeautifulSoup, Requests를 이용해 카테고리 당 2개 기사 크롤링
- Hero의 Clock기능을 이용해 6시간마다 parser_schedule.py 실행
- python의 APScheduler 모듈 사용

```
(venv) → safehome_django_rest_framework git:(master) heroku run |
Running python3 parser.py on ● django-react-safehome... up, run.1
10
"한달새 1억 뛰었다"...서울 집값 못 잡고 세종만 띄운 천도론
경제학자 76% "부동산 정책이 수도권 집값 폭등 불렀다"
10
서울시 시내버스 오후9시부터 자정까지 감축 운행
서울시, 하반기부터 '안전속도 5030' 전역 확대..."보행안전 개선"
10
'모델살인 혐의' 40대 여성 긴급체포..."내가 죽여" 자수
'수면제 커피'로 내연남 살인 혐의... 40대 여성 구속
10
한강대교 9년 만에 '홍수주의보'..."하류 지역 주민 유의"
집중호우로 서울서 29명 임시대피...한강대교는 홍수주의보(종합)
```

3-5 Heroku에 Django 앱을 배포



HEROKU

- 무료로 사용 가능한 클라우드 서비스
- Heroku에서 제공하는 **CLI**와 **GIT**을 이용해서 Django 프로젝트를 업로드하고 제어

- 1) react-django-safehome 웹앱을 생성하고 Django Project 업로드
- 2) 그 후, **git push heroku master** 명령을 통해 Heroku에 배포
- 3) Heroku 홈페이지(<https://dashboard.heroku.com/apps>) 에서 등록된 웹앱을 확인
- 4) <https://django-react-safehome.herokuapp.com/> 로 접속하여 배포된 Django app을 확인



<https://tinyurl.com/y68xp2v8>

4-1 개선점

01

서비스 지역 확대

전국 단위로 서비스 지역을
확대하기

02

동별 데이터 제공

광범위한 구별 데이터보다는
동별 데이터 제공

03

생활 만족도 데이터 추가

더 나아가 편의/문화 시설 데이터 제공

04

커뮤니티 공간으로 확대

로그인, 댓글 기능을 통한
사용자간 의견 공유

4-2 교훈 및 느낀 점

황동운

React로 SPA를 구현해보면서, 추가적으로 앞으로 필요한 Redux, React router 같은 기술들에 대해 관심이 생겼고, 이번 프로젝트에서는 사용하지 못했지만, 공부해보고 싶다.

강훈화

기록이 굉장히 중요하다는 사실을 알게 되었다. 다음 프로젝트를 할 때에는 설계부터 시작해서 프로젝트 진행상황, 각 조원들 간의 협의과정 등 다양한 것을 세세하게 기록 후 지표화 하면 좋겠다.

김유진

배포 과정에서 수많은 오류를 해결하면서 계획 단계에서의 꼼꼼한 설계가 중요하다는 것을 알았다.

박진우

Front-End와 Back-End간의 소통이 중요하다는 것을 느낄 수 있었습니다. 다음에는 React와 같은 Front-End도 도전해보고 싶습니다.

Thank you..!!!

Thank you for listening.
Tell us if you have any questions