

PORTFOLIO

끊임없이 성장하는 개발자

손 현 석

Tel: 010-9144-2948

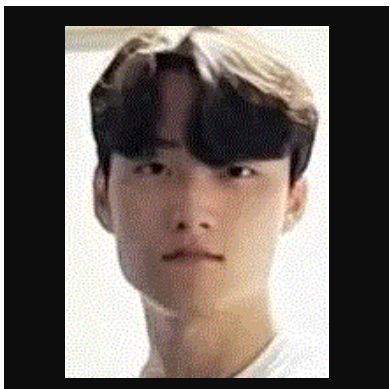
Email: snrnakat@naver.com

Github: hyeonseok0912

Blog: <https://blog.naver.com/snrnakat>

CONTENTS

1. 자기소개
2. 경험기술서
3. 기술 보유
4. 참여 프로젝트 목록
5. 주요 포트폴리오
 1. 학원 실무 프로젝트
 2. 영수증을 부탁해



Son Hyeon Seok

안녕하세요. 끊임없이 성장하는 개발자 손현석 입니다.

다양한 인턴 및 프로젝트 경험

제조 및 항공 분야 회사에서의 2회의 인턴 경험을 통해 사회성과 효율성의 중요성에 대해 느끼고 체득했습니다. 또한 앱 개발 프로젝트를 시작으로 위성 통신 프로그램을 거쳐 웹 개발 및 API 연동 프로젝트까지 이어지는 다양한 경험을 통해 다양한 파트에 대한 관심과 그를 기반으로 한 역량을 지니고 있습니다.

교육 기관 이수를 통한 스텝업

대학 전공때 했던 프로젝트 경험만으로는 다양한 기술 스택을 활용해보지 못해서 아쉽다 판단했습니다. 그래서 개인 역량도 함양시키고 프로젝트도 더 해보고자 싶어서 독학과 동시에 교육기관을 알아보았습니다. 마침내 국비지원 JAVA 기반 풀스택 개발자 과정을 통해 각종 언어와 프레임워크, 기술 스택 등에 대해 익히면서 한층 더 개발자가 되기 위해 나아가는 중입니다.

항상 배우고 성장하려는 의지

처음부터 코딩, 나아가 프로그래밍에 대해 자신이 있었던 것은 아니었습니다. 오히려 생소한 언어들과 프로그래밍은 그를 주저하게 만들었습니다. 하지만 효율성을 추구하고 새로운 것에 흥미를 많이 느끼는 저였기에 프로젝트를 하면서 마주했던 문제들은 되려 저를 끊임없이 노력하게 만들었고, 그 결과 눈부신 성장을 이룰 수 있었습니다.

경험



손 현 석

EDUCATION

- 2016 경상대학교 항공우주소프트웨어공학과 전공
- 2023 국비지원 자바(JAVA)기반 풀스택 개발자 취업과정 이수

CAREER

- 2017.04 ~ 2019.03 트리엔(주) 근무
- 2021.01 ~ 2021.01 한국항공우주연구원 인턴
- 2021.06 ~ 2021.08 한국표면처리(주) 인턴

LICENSE

- 정보처리기사 (2023.06 취득)

EXPERIENCE

- 교내 프로젝트 참여(앱 개발)
- 한국항공우주연구원 위성 통신 프로그램 참여
- 교내 프로젝트 종합설계 참여
- 학원 실무 프로젝트1 참여
- 학원 실무 프로젝트2 참여(진행중)

Skill Set

- **Language** JAVA, JavaScript, Kotlin
- **OS** Window, Linux(Ubuntu 22.04 LTS)
- **Framework + Library** JQuery, Ajax, Spring, SpringBoot, MyBatis, Bootstrap, Node.js, Vue.js
- **Database** Oracle, MariaDB, PostgreSQL, MySQL
- **Development Tool** Eclipse, HeidiSQL, eGovFrameDev, Visual Studio, Android Studio, Docker
- **ETC** HTML5, CSS3, JSP, Thymeleaf, Github, Apache Tomcat, AWS(Amazon Web Services), OpenLayers,
QGIS, GeoServer

참여 프로젝트 목록

프로젝트명	프로젝트 기간	설명	담당역할
영수증을 부탁해	2021.09 ~ 2021.12	소비 기한을 사용하여 식품을 관리하는 서비스를 제공, 불필요한 지출 및 식품 폐기물 감소	품목 추출 및 OCR 전처리 담당
SDR 통신 실습	2022.01 ~ 2022.01	위성신호를 수신하여 해당 데이터 값을 이미지 파일로 변환	YAGI 안테나 제작 및 신호해석 툴인 GNURadio를 이용하여 GNURadio components 로직 제작
버드스트라이크 방지 드론 제작	2022.03 ~ 2022.06	비행체의 비행 시의 위험 감지 및 후속 상황에 대한 예방 기술 개발	드론 제작 참여 및 조류 인식 프로그램 구현
지리 정보 시스템 프로젝트 (학원 실무 프로젝트)	2024.03 ~ 2024.04	도시 및 지역의 공간 단위에서 전력 사용량을 파악함으로써 탄력적인 에너지 효율화를 실현하기 위함이다	선택 지역의 전력 사용량 표출 및 범례 구현, 사용자의 데이터 DB에 삽입, 시/도 별 전력 사용량 차트화

7. 주요 포트폴리오

- (1) 지리 정보 시스템 프로젝트
- (2) 영수증을 부탁해

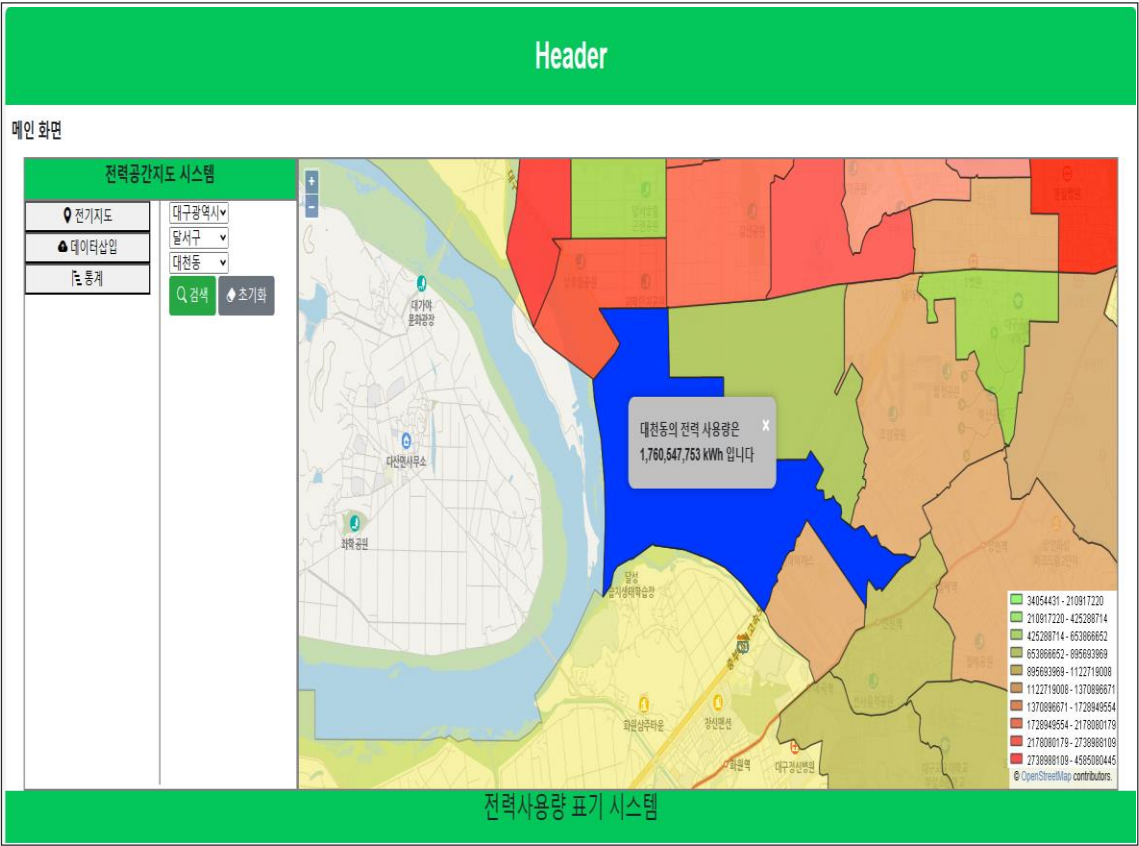
(1) 지리 정보 시스템 프로젝트

프로젝트 기본정보

프로젝트명	학원 실무 프로젝트I
프로젝트기간	2024.03 ~ 2024.04
프로젝트인원	5명
설명	배경 지도 표출, 시도/시군구/법례 선택 시 지도 확대 및 레이어 표출, 레이어 선택 시 해당 레이어의 전력 사용량 조회 팝업, 파일 업로드 기능, 업로드 상태에 따른 스낵바, 시도/시군구 별 통계 그래프 표출
담당역할	전기지도(레이어, 팝업, 필터링) 및통계 부분 담당

개발 주요사항

- MyBatis 활용 및 SQL Query를 실행하여 DB로부터 필요한 값을 Map 형식으로 받아 Json 타입으로 jsp에 넘겨줌
- Vworld API를 활용한 배경지도 표출
- GeoServer와 QGIS를 사용하여 지도 위에 레이어 표출
- PostgreSQL DBMS로 프로젝트 서비스 구현을 위한 데이터베이스 구축
- Map 객체를 통해 데이터 전송
- JSP에서 HTML, CSS, JavaScript, JQuery, Bootstrap 등을 활용해 화면 구축
- Modal, Swal, SweetAlert 등 화면에 필요한 디자인적 요소를 사용해 더욱 편리하고 깔끔한 화면 구성



[그림]. 레이어 및 팝업 표시 화면]

(1) 지리 정보 시스템 프로젝트

기술 스택

OS	Windows 11
Develop Tool	GovFrameDev (3.10.0)
Data Base	PostgreSQL(4.1)
Language	Java, JavaScript
Java Version	JavaSE-1.8
FrameWork & Library	GovFrameDev (3.10.0) / Mybatis (3.5.6) / json / Lombok (1.18.39) / jquery
Web Application Server	Apache Tomcat 9.0.
Collaborative Software	Github, Google Sheets(Excel)

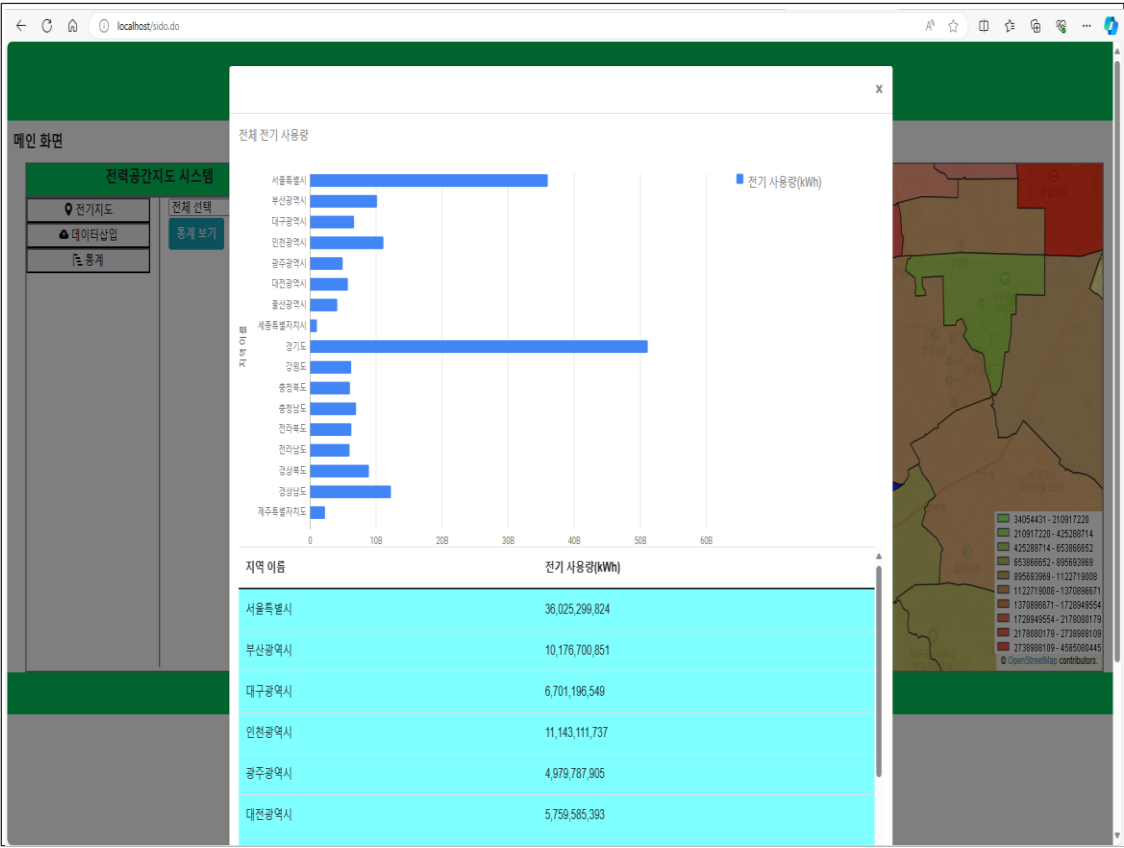
설명

서비스의 메인 화면에서는 크게 세가지 기능으로 나뉘어져 있습니다.

첫 번째는 지도 영역을 선택하여 해당 영역의 범위와 범례, 영역 및 팝업을 통한 전력사용량을 확인할 수 있는 지역 선택 기능입니다.

두 번째는 txt 파일을 업로드할 수 있는 파일 업로드 기능이며, txt 파일만 업로드 할 수 있습니다. Txt파일이 아닌 경우는 예외처리를 통해 업로드 기능이 실행되지 않도록 하였습니다.

마지막으로 각 구역이 사용하는 전력량의 값을 차트 및 테이블로 표시하여 나타내주는 그래프 기능이 있습니다.



[그림2.통계 기능 그래프 및 테이블]

(1) 지리 정보 시스템 프로젝트

7. 주요 포트폴리오

• 설명

현재 보시는 것은 실무 프로젝트의 화면 흐름도 입니다.

첫번째 기능으로는, 사용자가 선택한 지역의 전력 사용량을 표출해주고, 시각적인 표현을 위해 범례 또한 구현하는 것입니다.

두번째 기능으로는, 이미 저장되어 있는 정보 외에도 사용자가 새로운 데이터 파일에 대해서도 볼 수 있도록 데이터베이스에 정보를 저장할 수 있도록 파일 업로드 기능을 구현합니다.

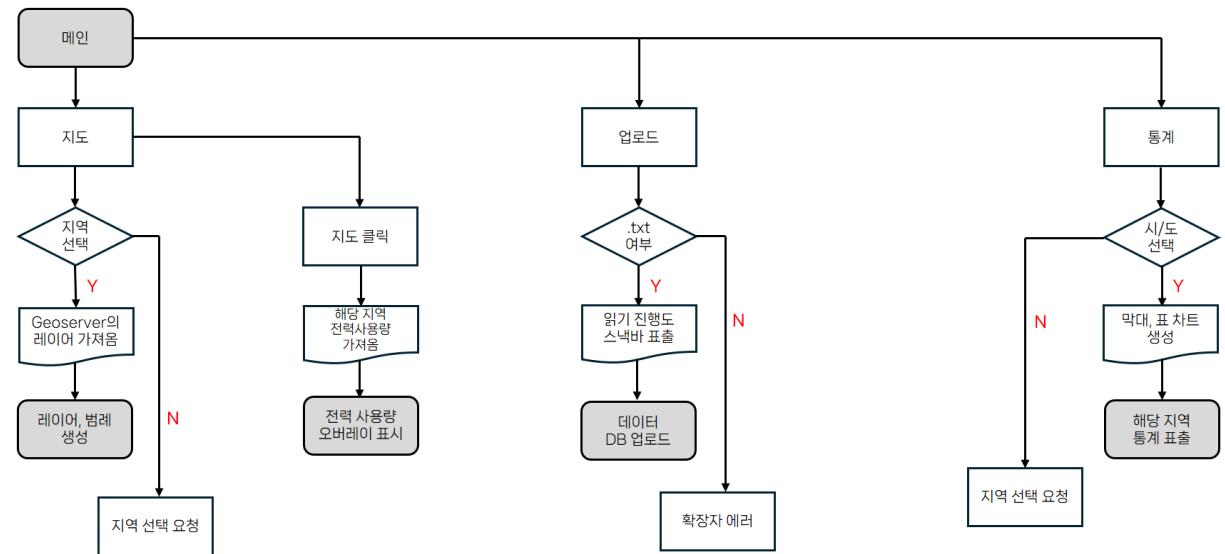
마지막으로 세번째 기능은, 시/도별 전력 사용량을 차트화 및 테이블화 시켜서 단순한 숫자의 나열일 때 떨어질 수 있는 가독성을 높여 사용자들에게 편의성을 제공해 주는 것입니다.

각 기능에 대한 시연 영상입니다.

전기 지도 : <https://youtu.be/ga0GdQ-Vmm4>

파일 업로드: <https://youtu.be/F4KwH3iJoLs>

통계 : <https://youtu.be/uRxiRkiKlhw>



[그림3. 실무 프로젝트 화면 흐름도]

(1) 지리 정보 시스템 프로젝트

• 개발 시 어려웠던 부분(1)

JSON data type에서의 문제

우측 코드는 옵션바에서 시도를 선택하면, 시도값을 AJAX를 통해 Controller로 보내 주고 해당 시도 이름값을 가진 시군구 데이터 값을 map 형식으로 jsp에 보내주고, jsp에서는 그 시군구 값을 json 형식으로 받아 사용하는 것입니다.

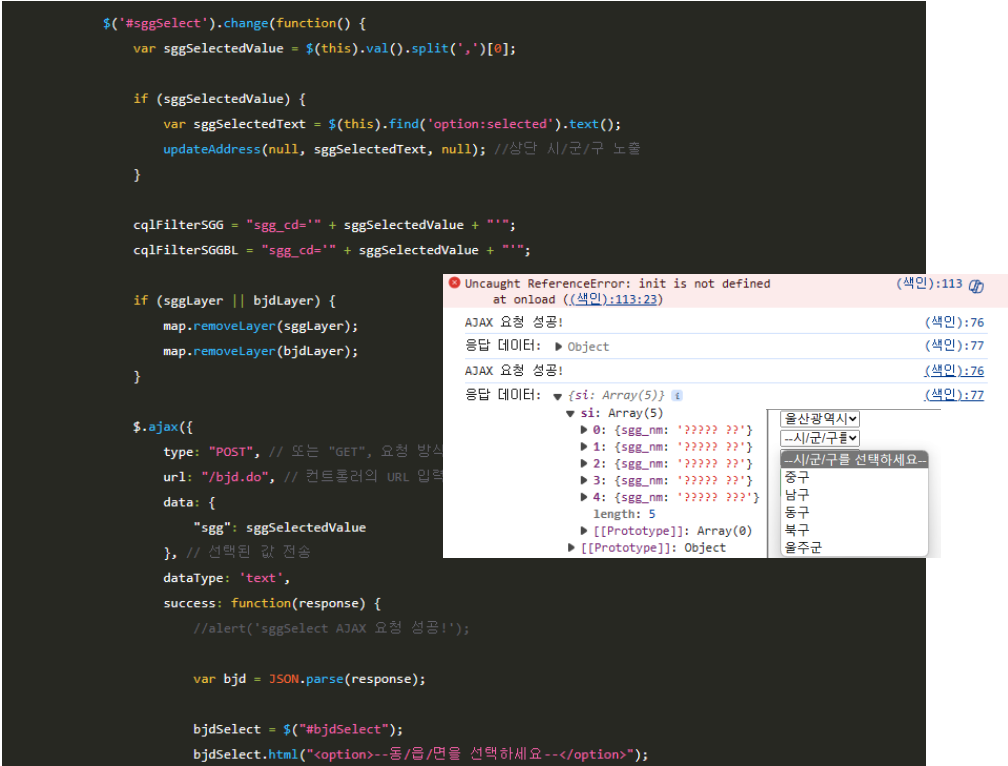
하지만 해당 부분을 개발할 때, 시군구 데이터 값을 map 형식으로 받아내는 것까지는 System.print문을 통해 확인했으나, 계속해서 웹 상에서 우측 이미지와 같이 시군구의 이름이 ???로 뜨는 것이 문제였습니다.

이는 jsp에서 json으로 받을 때 문제가 발생한 것이었고, 이 부분을 해결하기 위해 처음에는 언어가 맞지 않아서 그런가 싶어 jsp, controller 등 맞춰줄 수 있는 모든 부분에 대해 charset을 모두 UTF_8로 맞춰주었습니다. 하지만 문제는 해결되지 않았습니다.

이번에는 controlle에서 map 형식을 다시 json 타입으로 변환해서 jsp로 보내주는 Jackson 및 Gson librar도 활용했습니다. 하지만 여전히 해당 문제는 해결되지 않았습니다.

이 문제를 해결하기 위해 구글링을 계속 했고, Spring MVC가 @Controller에 요청을 보내기 위해 필요한 HandlerMapping과 HandlerAdapter를 bean으로 등록하고, 이렇게 등록된 bean에 의해 요청 url과 컨트롤러를 매칭할 수 있다는 사실을 발견했습니다. 그래서 pom.xml 파일에 아래 문구를 추가해주고 나니 정상적으로 json 데이터를 ???가 아닌 정상적인 값으로 받을 수 있었습니다.

`<mvc:annotation-driven/>`



[그림4. 실무 프로젝트 AJAX 통신]

(1) 지리 정보 시스템 프로젝트

7. 주요 포트폴리오

• 개발 시 어려웠던 부분(2)

CORS 충돌 문제

다음은 지도의 좌표 클릭 시 해당 부분의 좌표값을 통한 지역 이름과 해당 지역의 전력사용량을 나타내는 팝업 기능을 개발할 때 발생한 문제입니다.

여기서 발생한 문제는 CORS 충돌이었는데, 보안을 강화하기 위해 다른 출처의 정보는 막기 위해 존재하는 스템이 바로 CORS 였습니다.

하지만 저는 기존에 이러한 개념을 알지 못했던 상태였습니다. Geoserver의 port번호는 8080이고, 제 프로젝트의 port번호는 80이었기 때문에 80 포트에서 8080포트의 정보를 가져오려고 할 때 CORS 충돌이 났었습니다.

이를 해결하기 위해 서버측 에서 허용할 출처를 헤더의 Access - Control - Allow - Origin 에 기입해주었습니다. 하지만 이 방법으로 해결되지 않았고, 다른 방법을 시도해 보기로 했습니다.

이번에는 proxy 서버를 이용해 모든 출처를 허용해주는 것으로 해결해보려고 했으나, 이 역시 geoserver에서 값을 가져오는데 실패했습니다.

제가 CORS 문제를 해결한 방법은 다음과 같습니다.

C:\apache-tomcat-9.0.84\webapps\geoserver\WEB-INF 내에 있는 web.xml 파일에서 CORS에 대해 허용해 주게끔 코드를 변경해 주었고, 이후 CORS 충돌을 해결할 수 있었습니다.

```
//클릭 이벤트 리스너 설정
map.on('singleclick', function(evt) {
    // 클릭한 지점의 좌표를 가져옴
    var coordinate = evt.coordinate;

    // 해당 좌표에서의 지리적 정보를 가져오는 요청을 서버에 보냄
    var featureRequest = new ol.format.WFS().writeGetFeature({
        srsName: 'EPSG:3857',
        featureNS: 'http://localhost:8080/geoserver/cite',
        featurePrefix: 'cite',
        featureTypes: ['shinjinview23'],
        outputFormat: 'application/json',
        geometryName: 'geom',
        filter: new ol.format.filter.Intersects('geom', new ol.geom.Point(coordinate))
    });

    // 서버에 요청 보내기
    fetch('http://localhost:8080/geoserver/cite/wms', {
        method: 'POST',
        body: new XMLSerializer().serializeToString(featureRequest)
    })
    .then(function(response) {
        return response.json();
    })
    .then(function(json) {
        // 가져온 정보에서 단계 구분 값을 추출하여 팝업에 표시
        if (json.features.length > 0) {
            var properties = json.features[0].properties;
            var sgg_pu = properties['usage'];
            var sgg_cd = properties['sgg_cd'];
            var sgg_nm = properties['sgg_nm'];

            // 서버에 요청 보내기
            fetch('http://localhost:8080/geoserver/cite/wms', {
                method: 'POST',
                body: new XMLSerializer().serializeToString(featureRequest)
            })
            .then(function(response) {
                return response.json();
            })
            .then(function(json) {
                // 가져온 정보에서 단계 구분 값을 추출하여 팝업에 표시
                if (json.features.length > 0) {
                    var properties = json.features[0].properties;
                    var sgg_pu = properties['usage'];
                    var sgg_cd = properties['sgg_cd'];
                    var sgg_nm = properties['sgg_nm'];
                }
            });
        }
    });
});
```

Access to fetch at 'http://localhost:8080/geoserver/cite/wms' from origin 'http://localhost' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource. If an opaque response serves your needs, set the request's mode to 'no-cors' to fetch the resource with CORS disabled.

localhost:8080/geoserver/web/?2

```
<!-- Uncomment following filter to enable CORS in Tomcat. Do not forget
<filter>
<filter-name>cross-origin</filter-name>
<filter-class>org.apache.catalina.filters.CorsFilter</filter-class>
<init-param>
<param-name>cors.allowed.origins</param-name>
<param-value>*</param-value>
</init-param>
<init-param>
<param-name>cors.allowed.methods</param-name>
<param-value>GET,POST,PUT,DELETE,HEAD,OPTIONS</param-value>
</init-param>
<init-param>
<param-name>cors.allowed.headers</param-name>
<param-value>*</param-value>
</init-param>
</filter>
```

[그림6. web.xml 파일 코드]

≠

Port Name	Port Number
Tomcat admin port	8006
HTTP/1.1	80

[그림5. 포트 번호 불일치]

(1) 지리 정보 시스템 프로젝트

7. 주요 포트폴리오

• 느낀 점(회고)

이번 프로젝트는 게시판을 만들면서 배웠던 것 외에 처음 접하는 라이브러리나 툴, 프로그램들로 진행되면서 다양한 경험을 쌓을 수 있었는데, 배운 점이 많아 만족스러웠습니다.

우선 DB 사용에 있어 많은 어려움을 겪었는데, 지리 정보를 가지고 있는 칼럼에서 좌표를 뽑아내기 위해 SQL 함수 Geom을 다루면서 ST_함수에 대해서 공부해볼 수 있는 계기가 되었으며, DB의 데이터 값들을 받아오는 과정에서 테이블 간 join 기능을 썼을때 오래 걸렸기에, 이를 해결하기 위해 materialized view를 쓰면서 Select문의 처리 시간을 단축시키기 위해 고민 하는 데 많은 시간을 할애해보면서 프로젝트의 보여지는 기능 뿐 아니라 처리 속도와 데이터의 효율화 또한 중요하다는 것을 느끼게 되었습니다.

그리고 파일 업로드 과정에서 pagesize를 지정해주면서 대용량 데이터를 처리하는데 있어 속도와 안정성을 올리기 위한 고민도 해볼 수 있었습니다.

마지막으로 직접 개발 계획서, 보고서를 만들어보면서 프로그래밍을 하는데에는 문서 작업에 대한 경험을 해볼 수 있었습니다.

IT 프로젝트 개발 계획서

(「전력 사용량 지도 구축」)

2024. 04. 16.

훈련과정명 (소속)	프로젝트 기반 자바(JAVA) 응용 SW 개발자 취업과정 (중앙정보기술인재개발원)
팀 명	TEAM E
팀장 성명	손현석
팀원 성명	김지훈
팀원 성명	노재희
팀원 성명	이지은
팀원 성명	정진수
팀원 성명	채영선
지도교사	윤승현

3. 프로젝트 개발일정

작업명	담당자	개발 일정							비고
		1 W	2 W	3 W	4 W	5 W	6 W	7 W	
1. 프로젝트 기획	공통								03.19 ~ 03.20
- 주제 선정	-								
- 기존 앱 분석	-								
- 구현기능 구상	-								
- 업무 분장	-								03.20 ~ 03.27
2. 기획 및 설계	-								
- DB 설계	-								
- 개발환경 설정	-								
- 와이어프레임 제작	-								
- 기획서 작성	-								
- 흐름도 제작	-								03.25 ~ 04.05
- ERD 제작	-								
3. 개발	-								
- 기능구현	-								
- UI 구현/디자인	-								04.01 ~ 04.11
- 추가 기능 개발	-								
4. 테스트/오류수정	-								
- 테스트	-								04.01 ~ 04.11
- 오류수정	-								

[그림6. 실무 프로젝트 보고서 중]

(2) 영수증을 부탁해



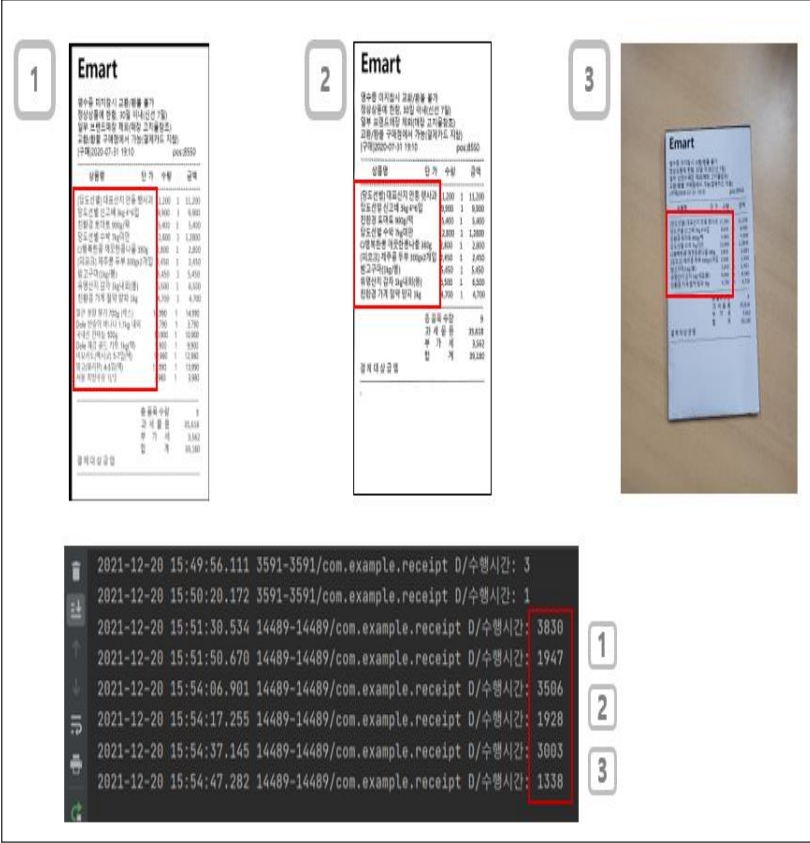
7. 주요 포트폴리오

프로젝트 기본정보

프로젝트명	영수증을 부탁해
프로젝트기간	2021.09~2021.12
프로젝트인원	4명
설명	소비 기한을 사용하여 식품을 관리하는 서비스를 제공, 불필요한 지출 및 식품 폐기물 감소
담당역할	품목 추출 및 OCR 전처리 담당

개발 주요사항

- JSON 형식의 파일을 사용하여 DataBase 구축
- 보관 방법에 따라 실온 DB와 냉장DB로 나눔
- 영수증에서 텍스트를 추출하기 위해 광학 문자 인식(OCR) 프로그램 사용
- OCR 기능의 정확도를 높이기 위해 Grayscale, 이진화, 기울기 보정, Jtessboxeditor 사용
- 시스템의 작업 속도 향상을 위해 imagecrop 기능을 추가



[그림7. 시스템 속도 테스트 화면]

(2) 영수증을 부탁해



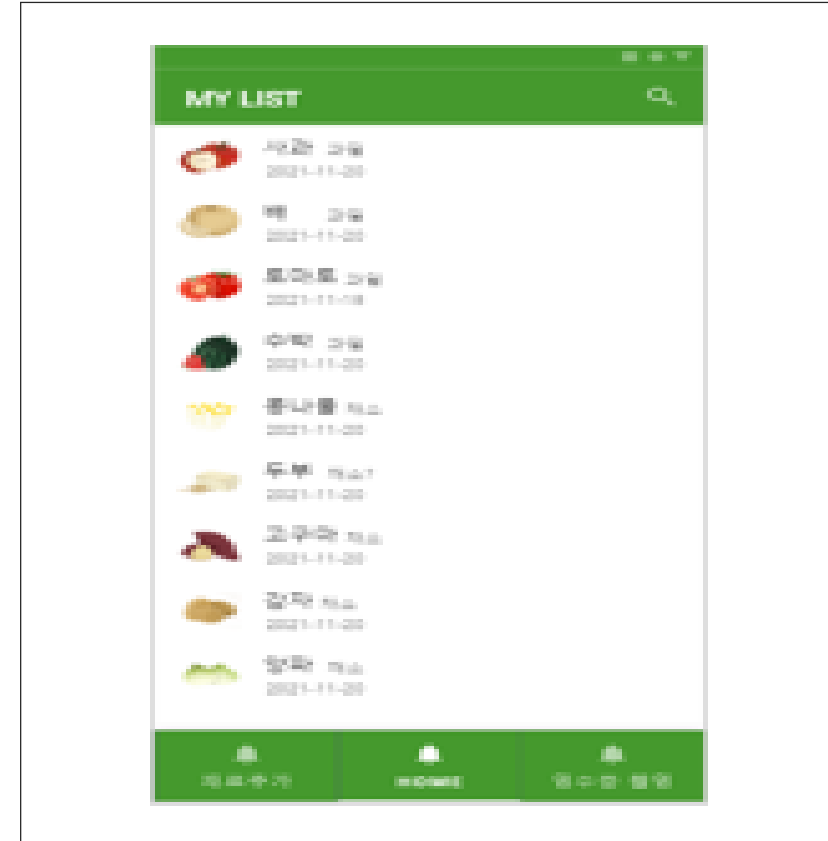
7. 주요 포트폴리오

• 기술 스택

- Kotlin
- Android Studio
- Data Format(Json)
- Open CV
- OCR(Optical Character Recognition)

• 설명

국내에서는 2023년부터 유통기한이 아닌 소비기한을 표시합니다. 소비기한은 보관 조건을 지키면서 식품을 소비하면 안전에 이상이 없는 기한을 의미합니다. 저희는 이런 소비기한을 사용하여 식품을 관리하는 서비스를 제공함으로써 불필요하게 버려지는 식품 폐기물과 음식물 쓰레기를 줄이고, 보관기간이 길어짐에 따라 불필요한 지출 또한 최소화하는 것을 목표로 개발을 진행하였습니다.



[그림8. 영부 앱서비스 III 화면]

(2) 영수증을 부탁해



7. 주요 포트폴리오

- 품질 시나리오(시스템의 작업 속도 3초 이내)

```
private fun launchImageCrop(uri: Uri): Bitmap? {  
    CropImage.activity(uri)  
        .setGuidelines(CropImageView.Guidelines.ON)  
        .setCropShape(CropImageView.CropShape.RECTANGLE)  
        .start(activity: this)  
    return loadBitmap(uri)  
}
```

```
CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE -> {  
    val result = CropImage.getActivityResult(data)  
    if (resultCode == Activity.RESULT_OK) {  
        var start = System.currentTimeMillis()  
        result.uri?.let { uri -> {  
            val bitmap = loadBitmap(uri)  
            binding.imagePreviewOCR.setImageURI(uri)  
  
            if (bitmap != null) {  
                imageExtract(bitmap.copy(Bitmap.Config.ARGB_8888, isMutable: true))  
            }  
        }  
        binding.imagePreviewOCR.setImageURI(result.uri)  
        var end = System.currentTimeMillis()  
        Log.d(tag: "수행시간", (end-start).toString())  
    }  
    else if (resultCode == CropImage.CROP_IMAGE_ACTIVITY_RESULT_ERROR_CODE) {  
        Log.e(TAG, msg: "Crop error: ${result.error}")  
    }  
}
```

[그림]. 영수증을 부탁해 코드]

IMAGECROP 기능 추가

속도 측면에서의 요구사항을 만족시키기 위해 IMAGECROP 기능을 추가하였습니다.

IMAGECROP은 촬영한 이미지를 잘라서 원하는 부분만 OCR의 INPUT으로 사용하는 방법입니다.

실제 테스트를 진행한 결과 이미지를 자르지 않고 텍스트를 추출했을 때는 약 3000~4000MS이 소요되었지만, 필요한 부분의 이미지만 잘라서 텍스트를 추출했을 때는 약 2000MS 줄어든 것을 볼 수 있습니다.

```
2021-12-20 15:49:56.111 3591-3591/com.example.receipt D/수행시간: 3  
2021-12-20 15:50:20.172 3591-3591/com.example.receipt D/수행시간: 1  
2021-12-20 15:51:30.534 14489-14489/com.example.receipt D/수행시간: 3830  
2021-12-20 15:51:50.670 14489-14489/com.example.receipt D/수행시간: 1947  
2021-12-20 15:54:06.901 14489-14489/com.example.receipt D/수행시간: 3506  
2021-12-20 15:54:17.255 14489-14489/com.example.receipt D/수행시간: 1928  
2021-12-20 15:54:37.145 14489-14489/com.example.receipt D/수행시간: 3003  
2021-12-20 15:54:47.282 14489-14489/com.example.receipt D/수행시간: 1338
```


(2) 영수증을 부탁해

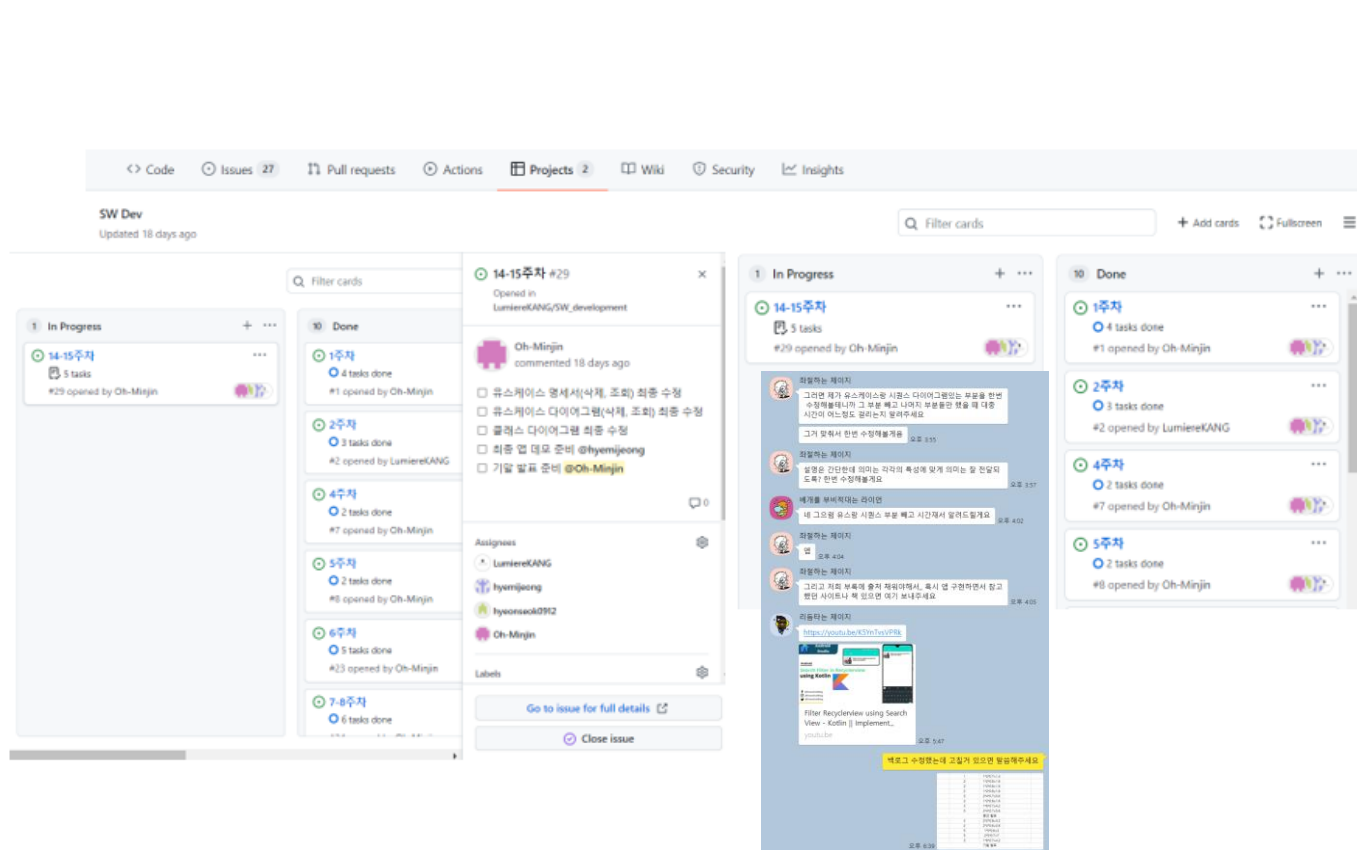


• 느낀 점(회고)

저희 조는 매주 수업이 끝난 이후 비대면 M 대면 회의
의를 통해 발표 당시 받은 피드백을 반영하여 개발
상황을 조정하였고, 해당 주차의 과제를 수행 및 역
할 분담, 각자 맡은 개발 진행 상황 브리핑 등을 수행
하였습니다.

또한, 회의록 작성자를 지정하여 매주 회의했던 내
용을 정리하고 각자의 진행상황을 한눈에 파악할
수 있도록 깃허브의 칸반 보드를 활용하였고, 원활
한 소통을 위해 카카오톡 단체 채팅방을 사용하여
의견 조율을 하였습니다.

칸반 보드를 활용함으로써 서로 맡은 사항이 어디
까지 진행되었는지, 각자가 진행해야 할 과제가 무
엇인지 잘 파악할 수 있었고 계획대로 개발이 큰 어
려움 없이 잘 진행되었습니다.



[그림1. 영수증을 부탁해 프로젝트 단톡방]

7. 주요 포트폴리오

END

손 현 석

Tel: 010-9144-2948

Email: snrnakat@naver.com

Github: hyeonseok0912

Blog: <https://blog.naver.com/snrnakat>