

이력서 전체보기

회사 6곳 · 프로젝트 40개. 회사를 누르면 그 회사에서 한 일이 펼쳐집니다.

[sbjang123456@gmail.com](#) [Blog](#) [GitHub](#) [이력서](#) [회고](#)

한샘 · Front-end Engineer

2023-08 — 재직 중

ERP 디자인 시스템 개발

ERP 개발팀의 UI 표준을 위한 디자인 시스템 개발

업무

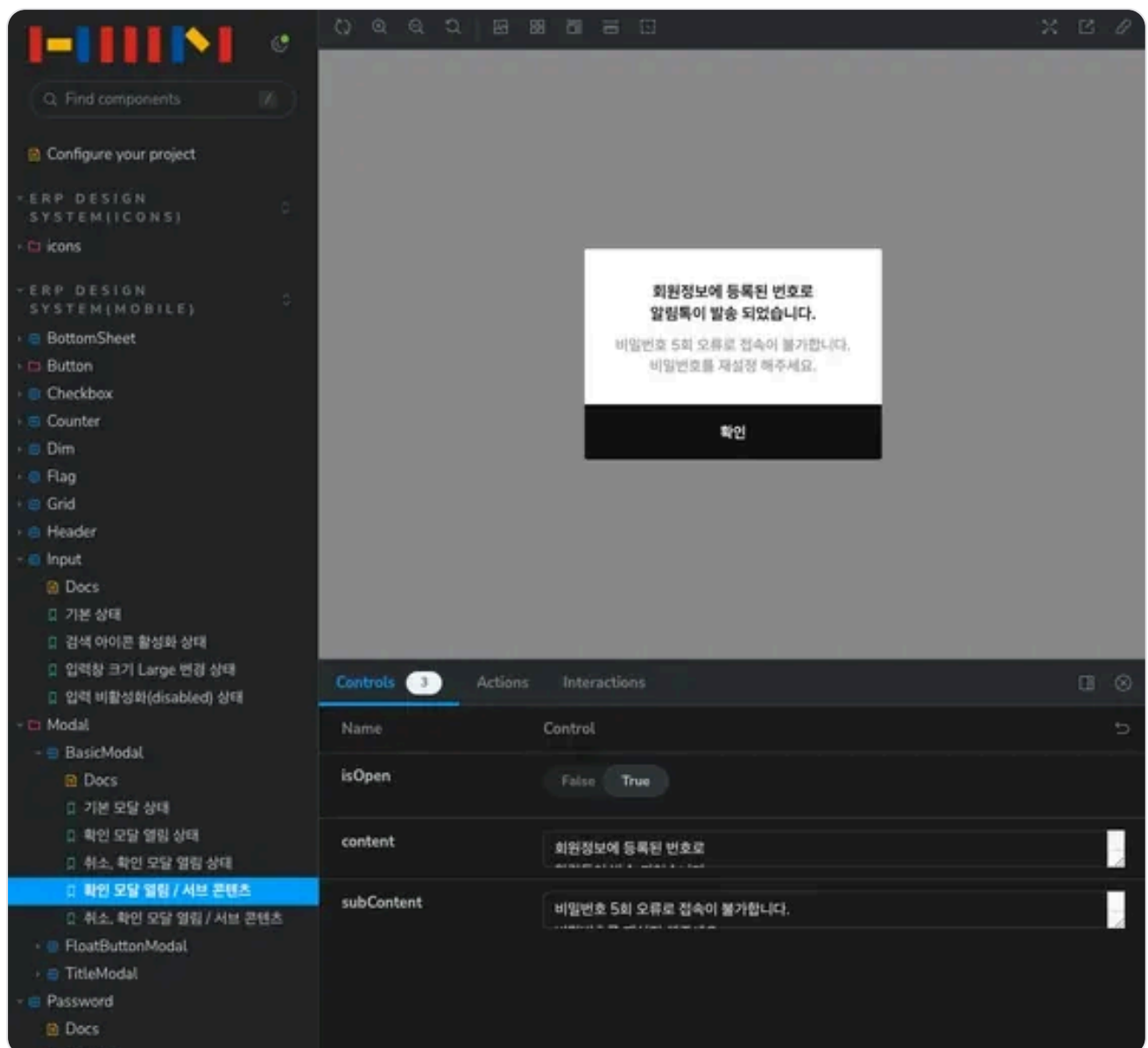
- turborepo, pnpm, rollup, typescript, react, vanilla-extract, storybook, vitest 기술 점검 및 선택
- 디자인시스템을 통해 기획자들이 공통 컴포넌트를 참고 및 사용해서 기획문서를 작성할 수 있도록 가이드하고, 이를 통해 의사소통 비용을 줄이며 상대적으로 완성된 기획서를 얻을 수 있는 효과를 기대.
- 기간제 디자인 시스템 구축 리딩
 - 코딩 컨벤션 정의
 - 컴포넌트 파일명 : PascalCase vs KebabCase → KebabCase
 - 컴포넌트 선언 : 익명함수표현식 vs 기명함수선언식 → 기명함수선언식
 - 내보내기 : named export vs default export → named export
 - barrel index 사용여부 → 사용
- 디자인시스템 패키지 구조를 설계 및 정의
 - eslint 커스텀 정의
 - icons 패키지 정의
 - svg 아이콘을 @svgr/cli 를 활용하여 자동 생성
 - design-system 패키지 정의
 - CSS-in-JS vs vanilla-extract → vanilla-extract
 - React Server Component 에서 원활한 사용을 위해 build 타입에 css를 추출해주는 vanilla-extract 선택
 - tailwind 는 클래스명의 길어짐으로 인한 팀원들의 불호와 props 에 따른 스타일을 좀 더 효율적으로 관리되는 vanilla-extract 로 채택
 - storybook 패키지 정의
 - UI 구성 요소를 독립적으로 개발하고 디자인 시스템 패키지에 종속적으로 관리하지 않기 위해 분리
 - icons 패키지와 design-system 패키지를 참조
- 번들 환경 구성
 - 최초 rollup 으로 구성 → vite 환경으로 변경
- 공통 디자인 요소 토큰화
 - 각 컴포넌트가 공통으로 사용할 스타일 css 토큰화
 - opacity, border, space, neutral color, brand color, accent color 등을 사전 정의
 - vanilla-extract 를 통해 global 에 위에서 생성한 토큰 변수를 생성
 - atomic 한 css 사용을 위한 sprinkles 정의
- 컴포넌트 개발

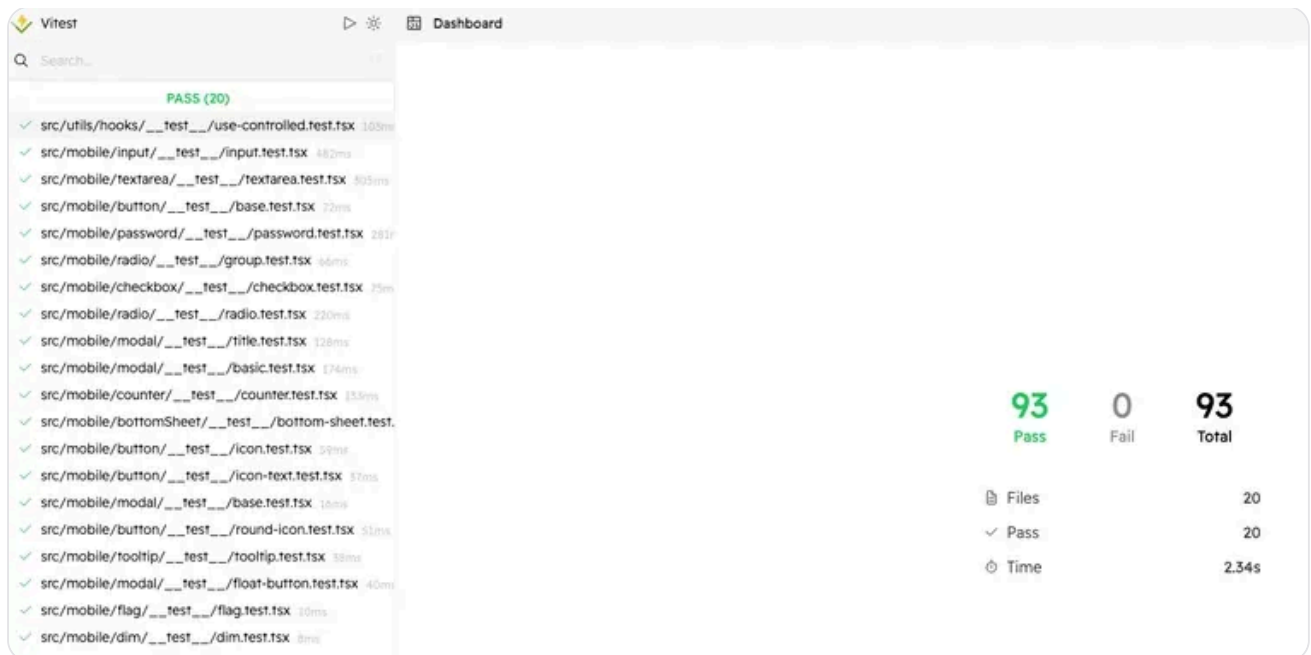
- 탭 메뉴 컴포넌트 개발
- 타이포그래피 컴포넌트 개발

작업 결과물

```

.
├─ package.json
├─ packages (9개의 package)
│   └─ erp-design-system
│   └─ erp-icons
│   └─ erp-storybook
│   └─ eslint-config-hanssem
│   └─ utils
├─ pnpm-lock.yaml
├─ pnpm-workspace.yaml
└─ turbo.json
  
```





vite 환경 변경으로 변경

- vitest 사용으로 환경 통일
- 기존 rollup 번들러에 따로 트랜스파일러를 추가해야했음 -> vite 는 기본적으로 rollup (+esbuild) 를 기본
- 빌드 속도 비교

※ 빌드속도만 놓고보면 최소 30% 차이로 보여짐 (물론 아직 패키지가 적다는 점도 고려바랍니다.)

- 빌드속도는 같은 모듈번들러를 사용하기 때문에 차이가 없어야 하는것이 맞지만, 내부적인 설정차이가 있는것으로 판단됨.(상대적으로 적은 설정으로 번들러 최적화설정을 쉽게 할 수 있을 것으로 판단)

홈퍼니싱 시공 PC/모바일 운영

시스템 운영 및 개선

업무

- 상용 Dhtmlx 그리드 라이브러리 메모리 누수 분석 및 개선 요청
 - 크롬 개발자모드의 performance 탭에서 그리드 스크롤링 시 리스너가 증가했다가 다시 감소되지 않는 현상 발견
 - 이 후, 라이브러리 소스코드내용 점검 결과 컬럼의 필터 이벤트가 이미 있어도 계속 add 되는 소스 발견
 - 상용 업체에 개선 요청
- FE 파트 리딩(정기배포를 위한 일정 플래닝)
 - 기획 회의 참여 및 기술적 점검
 - 각 기획에 대한 일정 산정 및 담당자 할당
 - QA 배포 및 운영 배포 항목 점검
 - git 브랜치 관리 및 히스토리 관리
 - merge 전략 재검토 및 재정의
 - 1 commit + rebase and merge 방식으로 정의
- 홈퍼니싱 시공 시스템 개선 및 운영 건 처리
 - 기획 팀 선정 회의를 통한 개선 건 추가 개발
 - 운영 버그 건 수정 개발

- ## 작업 결과물

한샘물 디자인 시스템 운영 및 개발

업무

- 4 / 34

작업 결과물

PC Storybook

@storybook/cli - Storybook

NewFlag

Flag(=깃발) 미리 정의된 케이스를 노출할때 사용하는 컴포넌트

- color와 textColor를 지정해서 사용하고, 그에 따른 개발의 편의성을 위해 NewFlagColorSet 값을 제공합니다((Inspect)D_variants color와 매칭)
- fontStyle: regular(기본값) 또는 bold 중에 선택하여 폰트 굵기를 조정할 수 있습니다

Search:

label

Show code

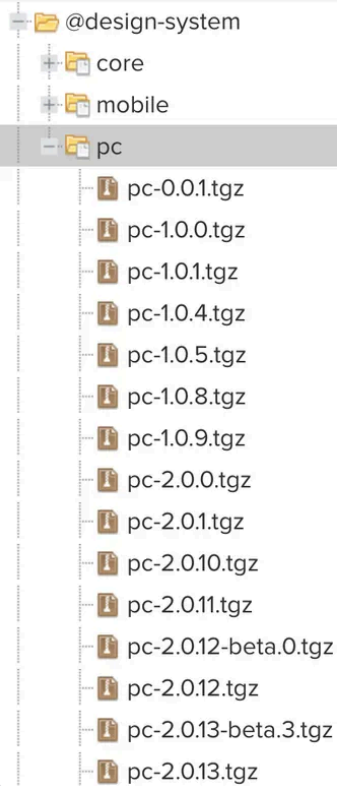
Name	Description	Default	Control
variant*	"solid" "line"	-	☒ solid ☐ line
size	"lg" "sm"	"sm"	☐ lg ☒ sm
fontStyle	"bold" "regular"	"regular"	☐ bold ☒ regular
content*	string	-	label
color*	ResponsiveValue<Color ColorKey, Omit<Required<Theme<TLengthStyledSystem>>, "colors"> & { colors: Object<Color ColorKey, "colors">; }> undefined	-	color_chip.blue.500
textColor*	ResponsiveValue<Color ColorKey, Omit<Required<Theme<TLengthStyledSystem>>, "colors"> & { colors: Object<Color ColorKey, "colors">; }> undefined	-	color_chip.primary
icon	{ variant: "ic_apple_solid" "ic_band_solid" "ic_facebook_solid" "ic_instagram_solid" "ic_kakao_solid" "ic_kakao_story_solid" "ic_kakao_txt_solid" "ic_line_solid" ... 234 more ... "ic_notification_bold"; align: "right" "left"; }	-	Set object

STORIES

Mobile Storybook

@storybook/cli - Storybook

라이브러리 Publish



외주 시공프로 관리 서비스 구축

외주 시공프로 기사들의 출석 및 현장 출입, 시공 종류를 관리하기 위한 신규 서비스 구축

업무

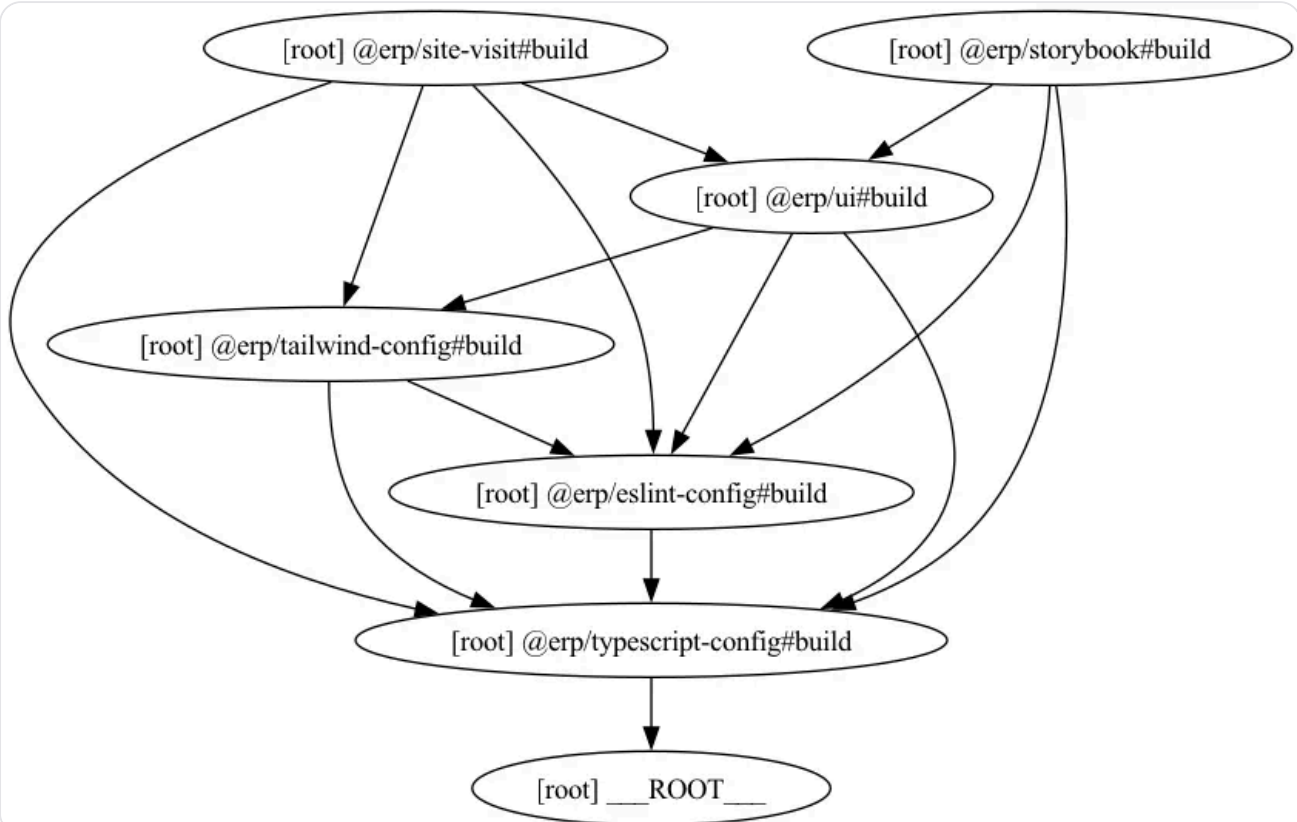
- turborepo, pnpm, vite, typescript, react, @tanstack/query, tailwindcss, storybook, vitest 기술 점검 및 선택
- FE 프로젝트 구축 리딩
- 신규 ERP(기간제) 모노리포 구조 정의
 - eslint 패키지 정의
 - Flat Config 구성
 - ui 패키지 정의
 - 공통화한 아토믹한 컴포넌트를 점진적으로 확대하여 ERP의 디자인 시스템으로 활용하기 위해 UI 패키지 구성
 - 디자인 가이드의 기초 구성 (토큰화)
 - tailwind의 커스텀 클래스 확장
 - 컴포넌트 개발
 - 버튼, 체크박스, 필터칩, 다이얼로그, 타이포그래피
 - 외부 사용 시 class 명 순서를 보장받기 위한 tailwind-merge 도입
 - storybook 모듈 추가
 - CDD 개발을 위한 컴포넌트 개발 툴 추가
 - ui 구성요소와는 별개로 stories 를 용이하게 관리하기 위해 별도 분리
- 화면 개발
 - 모바일 웹앱(웹)
 - toss/slash 의 funnel 을 리액트용으로 개선 후 추가

작업 결과물

프로젝트 구조

```
├─ README.md
├─ apps
│   └─ site-visit
│       └─ storybook
├─ package.json
├─ packages
│   └─ eslint-config
│   └─ tailwind-config
│   └─ typescript-config
│   └─ ui
├─ pnpm-lock.yaml
├─ pnpm-workspace.yaml
├─ .npmrc
├─ .nvmrc
├─ .prettierrc
├─ lint-staged.config.mjs
├─ bitbucket-pipelines.yml
└─ turbo.json
```

의존성 관계



외주 시공 출입 관리 시스템

AS모바일 Hybrid 전환

레거시 환경의 Native AOS 개발 환경에서 하이브리드 웹뷰 환경으로 전환하기 위한 프로젝트

- 전체 화면을 웹뷰로 전환함으로인한 유지보수성 증가

업무

- 기술 검토 및 선택
- 기본 템플릿 구성
- 기술 문서 작성 (저장소 내 README.md 및 팀 컨플루언스)
 - 전체 모노리포 구조 및 실행 방법
 - 하위 패키지들에 대한 사용방법
- 전체 프로젝트 구조 설계(FSD)
- 공통 디자인가이드 적용
- 공통 컴포넌트 개발
- 화면 개발
 - 메인, 메뉴, as게시판, 제품교육자료, 동영상교육센터, 현장지식DB, 상시자재관리, AS 및 오더 조회

작업 결과물

프로젝트 구조

```
.
├── CHANGELOG.md
├── eslint.config.js
├── index.html
├── lint-staged.config.js
├── package.json
├── public
│   ├── favicon.ico
│   ├── index2.html
│   └── robots.txt
├── README.md
├── src
│   ├── app
│   │   ├── app-providers.tsx
│   │   ├── index.css
│   │   ├── main.tsx
│   │   └── providers
│   ├── entities
│   │   ├── always-on-material
│   │   ├── board
│   │   ├── construction-guide
│   │   ├── education
│   │   ├── home
│   │   ├── inquiry
│   │   ├── schedule
│   │   └── site-knowledge-db
│   ├── features
│   │   ├── always-on-material
│   │   ├── as-board
│   │   ├── construction-guide
│   │   ├── education
│   │   ├── home
│   │   └── inquiry
```



```
| | |─ schedule
| | |─ site-knowledge-db
| |─ pages
| | |─ always-on-material
| | |─ as-board
| | |─ construction-guide
| | |─ education
| | |─ error
| | |─ home
| | |─ inquiry
| | |─ layout
| | |─ schedule
| | |─ site-knowledge-db
| |─ shared
| | |─ apis
| | |─ env
| | |─ lib
| | |─ session
| | |─ ui
| |─ vite-env.d.ts
| |─ widgets
| | |─ education
| | |─ home
| | |─ layouts
| | |─ schedule
| | |─ schedule-pkg
|─ tsconfig.json
|─ vite.config.ts
```


업무현황

[상세보기](#)

총	접수	임시	미완료	완료	자재취소
6	1	1	1	3	2



일괄 문자



25063000638

부엌

가구

건재

기본

조치미완료(미)



일 1좌석 / 안락2동



25070900002

부엌

가구

건재

기본

조치완료(완)(온라인)
자재취소(3)



일 5좌석 / 안락2동



25072500706

부엌

가구

건재

기본

조치완료(완)



일 3좌석 / 안락2동



25072500709

부엌

가구

건재

기본

조치완료(완)



일 4좌석 / 안락2동

25072901110

부업

가구

건재

기본



조치미완료(접)



일 1좌석 / 안락2동

자재취소(1)

25073000929

부업

가구

건재

기본



조치미완료(임)



일 6좌석 / 안락2동

퇴근기록

메인화면

MDS(기준정보시스템) 구축

레거시 MDM 솔루션의 제작 업체 폐업으로 시스템 운영 및 관리가 어려워짐에 따라 기준정보시스템(MDM) 구축을 위한 프로젝트

업무

- 기술 검토 및 선택
- 기본 템플릿 구성
- 기술 문서 작성 (저장소 내 README.md 및 팀 컨플루언스)
 - 전체 모노리포 구조 및 실행 방법
 - 하위 패키지들에 대한 사용방법
- 현재 프로젝트 분석 단계로 프론트 프로젝트에 대한 기술 검토 및 선택, 프로젝트 초기 구성 진행
- 공통 요소 개발
 - 좌측 메뉴 및 그에 따른 탭 메뉴 레이아웃 구성
 - 많은 양의 데이터로 인한 성능 문제로 인해 최초에는 라우트 탭 형식으로 설계 및 개발되었으나, 탭 이동 시 작업내용 초기화로 인해 마운트 유지 방식으로 변경
 - 메인 탭 내의 화면상에서 api를 통해 공통으로 관리되는 탭, 그에따른 컬럼, 데이터 MUI DataGrid(Premium) 에서 사용할 수 있는 데이터로 가공하여 공통 그리드(탭+컬럼+데이터) 영역 개발
- 작업 등록 및 그에 따른 6대 마스터 기준정보 작업화면 개발(작업관리)
 - 상품, BOM, 가격, 인포소스 작업화면 공통영역 개발
 - 작업등록 화면 개발
 - 배포대기열 화면 개발

작업 결과물

프로젝트 구조

```
.
├─ CHANGELOG.md
├─ coverage
│   └─ base.css
│   └─ block-navigation.js
│   └─ clover.xml
│   └─ coverage-final.json
│   └─ favicon.png
│   └─ index.html
│   └─ mds
│       └─ index.html
│       └─ lint-staged.config.js.html
│       └─ src
│   └─ prettify.css
│   └─ prettify.js
│   └─ sort-arrow-sprite.png
│   └─ sorter.js
├─ eslint.config.js
├─ index.html
├─ lint-staged.config.js
├─ nginx.conf
├─ package.json
├─ public
│   └─ favicon.ico
│   └─ robots.txt
├─ README.md
├─ src
│   └─ app
│       └─ app-providers.tsx
│       └─ main.tsx
│       └─ providers
│       └─ theme
│   └─ entities
│       └─ apply-target
│       └─ bom-ecn
│       └─ bom-grid
│       └─ bom-manufacture
│       └─ bom-sales
│       └─ bom-task
│       └─ code
│       └─ deploy-queue
│       └─ download-task
│       └─ field
│       └─ file
│       └─ generate
│       └─ goods
│       └─ goods-grid
│       └─ goods-task
```

- | | | └─ goods-task-deletion
- | | | └─ grid-preset
- | | | └─ history
- | | | └─ info-source
- | | | └─ info-source-grid
- | | | └─ info-source-task
- | | | └─ menu
- | | | └─ price
- | | | └─ price-grid
- | | | └─ price-task
- | | | └─ reference
- | | | └─ sales-office
- | | | └─ sales-office-grid
- | | | └─ task
- | | | └─ task-target
- | | | └─ vendor
- | | | └─ vendor-grid
- | | └─ features
 - | | | └─ auth
 - | | | └─ bom-ecn-list
 - | | | └─ bom-manufacture-list
 - | | | └─ bom-sales-list
 - | | | └─ bom-task
 - | | | └─ code-list
 - | | | └─ download-list
 - | | | └─ field
 - | | | └─ generate
 - | | | └─ goods-list
 - | | | └─ goods-task
 - | | | └─ grid-preset
 - | | | └─ history-list
 - | | | └─ info-source-list
 - | | | └─ info-source-task
 - | | | └─ price-list
 - | | | └─ price-task
 - | | | └─ sales-office-list
 - | | | └─ task
 - | | | └─ vendor-list
- | | └─ pages
 - | | | └─ bom-ecn-list
 - | | | └─ bom-manufacture-list
 - | | | └─ bom-sales-list
 - | | | └─ code-list
 - | | | └─ deploy-queue
 - | | | └─ download-list
 - | | | └─ error
 - | | | └─ generate-list
 - | | | └─ generate-register
 - | | | └─ goods-list
 - | | | └─ history-list
 - | | | └─ home
 - | | | └─ info-source-list

- └─ layout
- └─ price-list
- └─ sales-office-list
- └─ task
- └─ task-list
- └─ task-register
- └─ vendor-list
- └─ shared
- └─ apis
- └─ env
- └─ lib
- └─ session
- └─ ui
- └─ vite-env.d.ts
- └─ widgets
- └─ layout
- └─ task
- └─ tsconfig.json
- └─ vite.config.ts

작업 화면

HANSSEPI 통합기중정보시스템(MDS) DEV

...

작업 목록

도메인

전체

작업구분

전체

등록일 기간

2025.11.01

2026.01.15

전환일자

전체

검색

작업명

작업구분

작업구분

작업명

작업 목록

작업 No.

작업명

도메인

작업구분

작업구분

요청종류

진행상태

예약제로 등록(일제,시)

무선호의 번호

작업자

등록일

완료일

상태메세지

☐	355	BOM_20260106.3	BOM	변경	변경완료	MDS(화면)	임시저장		보통			2026. 1. 6. 오전 10:44...
☐	361	상용_20260105.1	상용	삭제호시		MDS(화면)	임시저장		보통			2026. 1. 5. 오후 4:28:13
☐	362	BOM_20260105.1	BOM	변경	변경완료	MDS(화면)	비트 완료		보통			2026. 1. 5. 오후 4:19:31 2026. 1. 6. 오전 9:40:49
☐	332	상용_20251112.3	상용	삭제호시		MDS(화면)	임시저장		보통			2025. 12. 12. 오후 6:3...
☐	238	상용_20251112.3	상용	생성		MDS(화면)	임시저장		보통			2025. 11. 27. 오전 9:0...
☐	236	상용_20251022.1,사본(2)	상용	변경		MDS(화면)	임시저장		보통			2025. 11. 26. 오후 12...
☐	282	가계_20251112.3	가계	변경	기분	MDS(화면)	임시저장		보통			2025. 11. 21. 오전 10:0...
☐	281	인력(소스_20251121.1)	인력(소스)	변경	인력제출	MDS(화면)	임시저장		보통			2025. 11. 21. 오전 9:02...
☐	255	가계_20251110.1	가계	생성	기분	MDS(화면)	임시저장		보통			2025. 11. 20. 오전 7:13...
☐	252	가계_20251110.1	가계	생성	기분	MDS(화면)	임시저장		보통			2025. 11. 19. 오후 3:07...
☐	247	임시(상용)등록구입(임시)등록(상용)서본	상용	변경		MDS(화면)	임시저장		보통			2025. 11. 17. 오후 4:36...
☐	245	상용_20251111.4	상용	삭제호시		MDS(화면)	임시저장		보통			2025. 11. 17. 오후 4:08...
☐	244	상용_16894474.87.2	상용	생성		MDS(화면)	예약문의 상태	2025. 11. 22. 오전 1:05:00	보통			작업할 대상이 없습니다. 작업화일을 확인해주세요.
☐	243	상용_16894474.87.2	상용	생성		MDS(화면)	예약문의 상태		보통			작업할 대상이 없습니다. 작업화일을 확인해주세요.
☐	233	상용_20251114.1	상용	생성		MDS(화면)	임시저장		보통			2025. 11. 14. 오전 10:2...
☐	226	BOM_20251113.2	BOM	생성	생성	MDS(화면)	임시저장		보통			2025. 11. 13. 오후 2:21...
☐	202	가계_20251110.3	가계	생성	작성	MDS(화면)	데이터입력 완료		보통			1 건의 데이터 등록상태에 실패하였습니다.
☐	178	상용_20251110.13	상용	변경		MDS(화면)	임시저장		보통			2025. 11. 10. 오후 1:14...
☐	177	상용_20251022.1,사본(2)	상용	변경		MDS(화면)	임시저장		보통			2025. 11. 10. 오후 1:13...
☐	176	상용_20251022.1,사본	상용	변경		MDS(화면)	비트 완료		보통			2025. 11. 10. 오후 12:5...
☐	171	상용_16894474.87.2	상용	생성		MDS(화면)	예약문의 상태		보통			2025. 11. 10. 오전 11:0...
☐	170	상용_16894474.87.2	상용	생성		MDS(화면)	예약문의 상태		보통			작업할 대상이 없습니다. 작업화일을 확인해주세요.
☐	169	상용_20251110.18	상용	생성		MDS(화면)	임시저장		보통			2025. 11. 10. 오전 11:0...

페이지 당 열: 30

1-30 of 37

HANSSEPI 통합기준정보시스템(MDS)

메뉴검색

검색어 입력

마스터 관리

작업 관리

작업 목록

작업 등록

배포 대기열

코드 채번

조회

상품

BOM 생산

BOM 영업

BOM 변경번호

가격

원료소스

구매처

판매처

다운로드 관리

이력 관리

공통 관리

작업 목록

작업 등록

상품

BOM 영업

배포 대기열

상품(변경)

No.178 상품_20251110.13

선택추가

작업취소

작업완료

기본(목록)

기본

기본(단위)

기본(재고)

영업설정

영업

영업일련.구매

MRPI2

MRP3

작업설정

회계/평가

현재 레이아웃

호드레거

공통관리

+

영 추가

-

영 삭제

S

□

영 추가

C

□

영 삭제

* 상품코드

□

* 상품명

□

* 대역 단위

□

* 분포

□

* 분자

□

입력(추가)

□

입력(추가)

□

농업(추가)

□

자수단위(추가)

□

CM2

2

선택안함

CM2

EA

EAA

KG

M

M3

MM

PC

ROL

YD

페이지당 행 50

1-50 of 2025

HANSSEPI 통합기준정보시스템(MDS) DEV

메뉴검색

검색어 입력

마스터 관리

작업 관리

작업 목록

작업 등록

배포 대기열

코드 채번

조회

다운로드 관리

이력 관리

공통 관리

작업 목록

상품(변경)

작업 등록

작업등록

도메인 *

상품

작업유형 *

생성

작업구분 *

선택

작업명

작업명 입력 (최대 20자)

작업정의

내용입력

작업선택

○

화면작업

○

엑셀작업

엑셀업로드

파일찾기

배포예약

YYYY.MM.DD hh:mm

우선순위

보통

작업시작

16 / 34

통합기준정보 시스템구축(2단계-MDC통합)

1단계 MDS에 이어 레거시 MDC(마스터 데이터 통합) 업무를 기준정보시스템으로 흡수하는 2단계 프로젝트

유통·제조 기준정보가 1단계에서 만든 기준정보시스템과 레거시 MDC로 나뉘어 있어, 같은 마스터를 두 시스템에서 따로 관리해야 했다. 2단계는 MDC가 맡던 Non-SAP 기준정보 업무까지 기준정보시스템 한 곳으로 모아, 작업 등록부터 검증·배포까지 한 흐름으로 잇는 것을 목표로 한다.

업무

- FE 파트 리딩 — 프론트엔드 아키텍처와 개발 범위 정의
 - 1단계에서 정한 스택(React + TypeScript + Vite, MUI X DataGrid, TanStack Query, Zustand, react-hook-form + zod)을 그대로 이어받고, FSD 레이어 규칙을 2단계 화면에 맞게 확장
 - 화면 단위로 개발 범위를 쪼개 일정과 담당을 배분하고, 공통으로 빼야 할 요소를 먼저 식별해 선개발
- API 계약 정의 — 응답 스키마(zod)와 타입, api client 메서드를 화면 개발보다 먼저 확정
 - Non-SAP 작업·작업단위·템플릿·필드그룹
 - 메타데이터, 참조코드, 권한 조회
 - 그리드 통합조회와 엑셀 다운로드
 - 장 전개·장 가격, 가격안·비규격 가격
 - SAP 로그 조회와 SSE 구독
 - 1단계에서 빠져 있던 엔드포인트 보완
- Non-SAP 작업·템플릿 화면 개발
 - 작업 목록·등록 화면
 - 템플릿 관리 목록과 등록·수정 다이얼로그
 - 작업 상세 화면
 - 작업단위 탭 전환 — 저장하지 않은 변경이 있으면 이동 전에 확인
 - 대상 그리드 인라인 편집과 행 추가·삭제, 작업 복사, 선택 행 일괄 변경
 - 버전 검증(낙관적 잠금) 기반 임시저장 — 다른 사람이 먼저 저장한 경우를 충돌로 잡아낸다
 - 엑셀 업로드와 대상 추가 팝업
 - 검증 실패 건만 골라내는 재작업
 - 작업단위 개별 배포와 전체 배포, 시각을 지정하는 배포 예약
- 기타 작업 화면 개발
 - OMS 카탈로그, 카탈로그 상품구분 맵핑 — 그리드 조회·편집·저장, 엑셀 업로드와 비동기 전체 다운로드
 - 상품이미지 조회·작업 — 이미지 업로드와 원본 크기로 보는 뷰어 다이얼로그
 - 공통 파일 업로드 다이얼로그·업로드 버튼과 엑셀 양식 다운로드 버튼을 공통 컴포넌트로 개발해 화면들이 나눠 쓰도록 정리
- SSE 실시간 반영 공통 인프라 구축
 - 사내 공통 패키지에 SSE 클라이언트와 훅을 추가하고, 토큰 재발급과 연결 중단 처리를 여러 앱이 함께 쓰도록 공유화
 - 도메인별로 흩어져 있던 SSE 훅을 공통 base 훅 경유로 정리
 - 엑셀 파싱 진행률과 작업단위 상태를 화면에 실시간 반영 — 새로고침으로 결과를 확인하던 흐름을 없앴
 - 알림이 연달아 와도 묻히지 않도록 스택형 스낵바 도입
- MUI v9 마이그레이션 — system props를 sx로, 컴포넌트 커스터마이징을 슬롯 API로 전환. 앱과 사내 공통 UI 패키지에 함께 적용

- 1단계(MDS) 본체 수정
 - 작업 API 경로를 SAP 작업 기준으로 정리
 - 일괄 변경 요청 필드명과 작업 상태값을 공통 코드셋에 맞춰 정렬
 - BOM 외주생산 조회·작업 화면 추가

작업 결과물

- 두 시스템에 나뉘어 있던 기준정보 업무를 한 화면 흐름으로 모아, 작업 등록부터 검증·배포까지 기준 정보시스템 안에서 끝낼 수 있게 함
- 엑셀 업로드·배포처럼 오래 걸리는 작업의 진행 상태를 SSE로 실시간 노출해, 결과를 기다리며 화면을 새로 고치던 대기 시간을 없앴

프로젝트 구조

```

.
├─ CHANGELOG.md
├─ eslint.config.js
├─ index.html
├─ lint-staged.config.js
├─ nginx.conf
├─ package.json
├─ public
│   └─ favicon.ico
│   └─ robots.txt
├─ README.md
├─ src
│   └─ app                # 4
│   └─ entities           # 47
│   └─ features           # 43
│   └─ pages              # 49
│   └─ shared             # 5
│   └─ vite-env.d.ts
│   └─ widgets            # 15
├─ tsconfig.json
└─ vite.config.ts
  
```

숫자는 레이어별 슬라이스 개수다.

AS포탈(업무시스템) 넥사크로 → React 전환

넥사크로 레거시 A/S 업무시스템 전체를 React로 전환. Claude Code 에이전트 스킴으로 분석-PRD-구현-테스트-리뷰-커밋 파이프라인을 만들어 50여 개 화면을 병렬 마이그레이션

10년 넘게 운영된 넥사크로 기반 A/S 업무시스템을 React SPA로 전면 재구축하는 프로젝트. 화면 수가 많고 레거시 동작을 그대로 보존해야 하는 전환 작업이라, 사람이 화면을 하나씩 옮기는 대신 AI 에이전트가 레거시를 분석해 명세를 만들고 구현·검증까지 수행하는 파이프라인을 설계하는 데 집중했다. 사람은 명세 검토와 머지 같은 게이트에서 판단만 맡는다.

업무

- FE 전환 설계·개발 전담 — 아키텍처 정의와 전 화면 마이그레이션
 - React + TypeScript + Vite SPA, FSD 아키텍처. TanStack Query·Table·Virtual, react-hook-form + zod, zustand, Tailwind CSS + shadcn/ui 조합

- 기준정보·A/S 접수·A/S 요청·조치관리·고객관리·평가분석·정산(유상수금, 현금영수증·카드결제)·게시판·교육·관리자 등 업무 도메인 전반의 화면 50여 개 전환
- 공통 기능·인프라
 - 메인·영업조직 선택, GNB 메뉴, 탭 워크스페이스(탭 상태 유지·스크롤), breadcrumb, SSO 로그인 리다이렉트, nginx 배포 구성
 - 검색 다이얼로그를 합성 컴포넌트로 공통화하고 중복 도메인 서비스를 정리 — 화면들이 같은 조회·검색 패턴을 나눠 쓰도록 수렴
 - 엑셀 다운로드를 서버사이드 EXPORT로 전환(컬럼 그룹화·다중 시트)하고 암호화 팝업을 전 화면에 적용
- UI 현대화 — shadcn/ui 기반 디자인 토큰·다크모드·Pretendard 적용, Radix에서 Base UI로 전환
- 테스트 체계 — Vitest + Testing Library + MSW로 레거시 동작 보존을 검증하는 테스트 374파일 3,867건 구축, 실행 속도 개선 병행
- 두 차례 QA 라운드 대응 — 수정필요 판정 207건을 이슈당 1커밋·1PR 원칙으로 처리

AI 에이전트 활용

이 프로젝트의 방법론 자체가 성과다. Claude Code 에이전트 스킬을 직접 제작해 "레거시 분석 → PRD 생성 → 구현 → 테스트 → 리뷰 → 커밋·PR"을 잇는 마이그레이션 파이프라인을 만들었고, 화면 단위로 워크트리를 쪼개 여러 에이전트가 동시에 화면을 옮기게 했다.

- PRD 생성 스킬(nexacro-react-prd) — 레거시 화면을 분석해 마이그레이션 명세를 자동 작성
 - 메뉴 스코프·화면 분석·API 추적·PRD 조립·통합테스트 시나리오 5개 에이전트가 병렬로 분석
 - 같은 분석을 반복해 누락(갭)이 0이 될 때까지 도는 자동 보완 루프(최대 10회) — 기능 누락 문제를 스킬 수준에서 해소
 - 전체 이관용 full 모드와 누락 기능 탐지용 gap 모드 분리
- 구현 스킬(nexacro-react-migration) — PRD를 입력받아 코드·테스트·리뷰까지 수행
 - 코딩 컨벤션 선로딩, 기존 API 스캔(재사용 맵 작성으로 중복 엔드포인트 생성 방지) 두 게이트를 통과해야 코드 편집 가능
 - UI·API/상태·테스트 등 역할별 에이전트 4~5개 병렬 구현, 이후 컨벤션 리뷰어와 자동 코드 리뷰 에이전트가 순차로 후처리
 - 분석과 구현을 한 세션에서 하던 초기 접근이 컨텍스트 오염으로 품질이 흔들리는 문제를 겪고, PRD와 구현을 별도 스킬로 분리한 것이 가장 큰 전환점
- 컨벤션 스킬(basis-web-conventions) — FSD·React·테스트 규칙을 라우터 + 주제별 레퍼런스 구조(Progressive Disclosure)로 설계해, 모든 에이전트가 필요한 규칙만 로드하면서도 같은 규칙으로 코드를 쓰게 함
- 워크트리 병렬화(git-ops) — 화면(이슈) 단위로 git 워크트리·브랜치·에이전트 세션을 만들어 다수 화면을 동시에 진행
 - 워크트리 일괄 생성 시 서브에이전트가 install·build·세션 구성을 독립 컨텍스트에서 수행, 터미널 멀티플렉서(tmux·cmux·Orca) 자동 감지
 - rebase → 커밋 → 명세 문서 정리 → PR 생성까지 커밋 파이프라인을 스킬로 자동화, PRD에서 워크트리 생성과 구현 지시까지 한 번에 이어지는 원스텝 트리거 완성
- 멀티에이전트 오케스트레이션(Orca) — 코디네이터 에이전트가 작업 DAG를 만들어 워커 에이전트들에게 분배
 - PostToolUse 훅으로 오케스트레이션 시작 순간 프로젝트 규칙을 자동 주입 — 키워드 트리거에 기대지 않고 규칙 적용을 구조적으로 보장
 - 워커는 plan mode로 기동해 사람 승인 전 코드 수정을 차단, 모노레포 테스트 동시 실행은 2개로 제한(순번 세마포어), 머지는 코디네이터만 수행

- 보조 스킬 — 테스트 작성 워크플로우(write-test), 대용량 OpenAPI 명세 부분 조회(swagger), Jira 일감 일괄 생성·Bitbucket PR 자동화까지 개발 프로세스 전반을 스킬로 도구화
- QA 대응도 에이전트 루프로 처리 — 이슈마다 레거시 원본과 대조 조사 후 이슈당 1커밋·1PR로 수정하고, 코드 결함이 아닌 건은 근거 코멘트로 판별해 종결 요청
 - 1차 라운드: 수정필요 152건 전량 처리 — 커밋 131건, PR 31건, 전체 테스트 3,867건 통과 상태 유지
 - 2차 라운드: 55건 중 49건 PR 머지, 나머지 6건은 기해결·재현정보 필요·범위 밖 판정 근거를 달아 종결 — 지적 범위 밖 잠재 결함 14건과 QA 시트 오기재 5건을 역으로 발견해 보고

작업 결과물

- 넥사크로 레거시 업무시스템의 화면 50여 개를 React로 전환 — 작업 이슈 350여 건을 에이전트 파이프라인으로 처리
- 한 화면씩 수작업으로 옮기던 방식을 여러 화면 동시 진행으로 바꾸고, 자동 보완 루프로 기능 누락을, 공유 컨벤션 스킬로 코드 스타일 불일치를 구조적으로 제거
- 레거시 동작 보존을 검증하는 테스트 374파일 3,867건이 매 수정의 회귀 안전망으로 동작
- 파이프라인 스킬 제작기와 운영 규칙을 사내 문서로 공유해 다른 프로젝트가 같은 방식을 재사용할 수 있게 함

프로젝트 구조

```

.
├─ CHANGELOG.md
├─ components.json
├─ docs
├─ eslint.config.js
├─ index.html
├─ lint-staged.config.js
├─ nexacro-legacy
├─ nginx.conf
├─ package.json
├─ public
├─   └─ legacy
├─ src
├─   └─ app           # 4
├─   └─ entities      # 119
├─   └─ features      # 64
├─   └─ pages         # 57
├─   └─ shared        # 6
├─   └─ vite-env.d.ts
├─   └─ widgets       # 31
├─ tsconfig.json
└─ vite.config.ts
  
```

숫자는 레이어별 슬라이스 개수다.

기존 vuejs + firebase 형태의 시스템을 nextjs + spring web flux 로 재구축

스토어링크의 메인 프로젝트로 FE 파트에서 전체적인 파트를 맡으며 리딩
산발적으로 흩어져있는 Github의 FE 저장소를 보다 관리에 용이하게 하기 위해 monorepo 를 적극 제안

업무

- 프론트 엔드 시스템들의 패키지 라이브러리 공유를 위해 모노리포 도입(yarn workspace)
 - eslint + prettier 를 모든 패키지 및 앱에서 같은 룰로 사용함으로써 중복 작업 방지
 - 여러 시스템에서 공통으로 사용할 자바스크립트(or 타입스크립트) 유틸성 라이브러리를 공유
 - 디자인 시스템 공통 패키지 구성으로 중복 작업 최소화
- ui 시각적 공유를 위한 스토리북 + 크로마틱 도입
 - CDD 개발을 위한 스토리북 구성
 - 컴포넌트의 시각적인 테스트를 위한 chromatic 구성
- 컴포넌트 단위 테스트를 위한 jest 및 react-testing-library 도입
 - 최초 도입은 jest 테스트 프레임워크를 도입하였으나, 체감되는 테스트 속도 개선이 된 vitest 로 전환
- github action 을 이용해 ci 수립
 - 각 패키지 및 앱의 ci 구성
 - lint, typecheck, unit test, build test
 - chromatic 연동으로 ui 패키지 변경 시 visual test 가능하도록 구성
 - changeset 도구를 연동하여 각 패키지, 앱의 changelog 관리
- 스토어링크 5.0 화면 구현
 - 사용자 서비스 개발
 - 백오피스 개발

작업 결과물

[스토어링크 | 이커머스 마케팅의 새로운 기준](#) 

애드링크 시스템 신규 구축

광고업체를 관리하고 마케팅 시 사기사례를 서로 공유하여 피해를 최소화하고, 향후 광고 성과 분석 기능의 확장을 통해 마케팅의 효율성을 가시화하는 시스템의 프론트엔드를 담당

사이드하게 진행된 시스템 개발 건

업무

- 사용자 및 백오피스 로그인 기능 구현
- 백오피스 시스템 개발

작업 결과물

[메인페이지](#)  [로그인](#) 

쿠팡 모니터링 시스템 신규 개발

쿠팡상품 목록을 크롤링 이 후 현재 등록된 상품의 가격의 인상 및 인하를 모니터링 하는 시스템의 프론트엔드 영역의 개발을 담당

사내에서 긴급하게 나온 신규 시스템 개발건으로 고객들에게 사이드하게 서비스로 제공을 위해서 제작
추 후 개발될 스토어링크 5.0 과의 로그인 공유를 위한 쿠키 토큰 저장 방식 선택

업무

- 로그인 기능 개발
- 쿠팡 기준 현재 스토어링크에 등록된 상품의 가격 인상 및 인하를 표시하는 목록을 보여주는 메뉴 개발

작업 결과물

pam.storelink.io 

유니비 시스템 유지보수

베트남 공동구매 이커머스 플랫폼

베트남에서 사용하는 마케팅 캠페인 시스템으로 신규구축 때는 참여하지 않았지만 사이트맵과 seo 메타태그를 적용해보며, og 공유를 통해 특정 url에서 공유 시 보여지는 미리보기를 확인할 수 있었습니다. 또한 다국어처리를 삽입 시 next export 기능을 사용하지 못한점을 해결하지 못해서 아쉬웠습니다.

업무

- next-sitemap 을 통한 사이트맵 생성
- next-seo 를 통한 메트태그 추가

관련 뉴스

- https://mobile.newsis.com/view.html?ar_id=NISX20220427_0001850193

스토어링크 4.0 백오피스 기능개선 및 유지보수

기존 레거시 스토어링크 4.0 기능개선

업무

- vuejs + firebase 로 구성되어있는 기구축 시스템의 추가 기능개발 및 인입되는 QA 처리

위메프오 · Front-end Engineer

2021-10 — 2022-04

위메프오 플러스 웹뷰 및 어드민 시스템 기능 개선 및 유지보수

프랜차이즈의 자체 앱 솔루션을 제공하는 D2C 솔루션 플랫폼 개발

업무

- 앱의 메인 홈의 구성 및 변경하는 관리 페이지 백오피스 메뉴 개발
 - 추가 및 삭제, 순서 변경 기능 추가
- 쿠폰 조건 및 적용매장을 설정하는 관리 페이지 백오피스 메뉴 개발
 - 운영관리자들이 쿠폰을 발급할 수 있는 조건 선택 기능추가
 - 지점 선택 기능 추가
- 알림 및 푸시 발송을 작성하는 관리 페이지 백오피스 메뉴 개발
- 앱 로그인 화면(웹뷰) 캡차 기능 추가
- 앱 후기 등록 화면(웹뷰) 개발
- 앱 매장 상세의 스크롤에 따른 카테고리 메뉴 활성화 기능 추가
 - intersectionObserver 를 통해 사용자가 스크롤을 통해 메뉴의 카테고리가 변경되면 상단 카테고리 메뉴를 활성화

위메프로 POS 웹뷰 기능 개선 및 유지보수

위메프로 결제 시 가맹점에서 주문을 처리하는 POS 시스템의 웹(하이브리드) 화면 기능개선 및 유지보수

업무

- 라이더 요청 사항 추가
- 배달 대행사 외부 연동 기능 개발
 - 기존 등록되어있지않은 대행사를 선택할 수 있도록 개선
- Windows 버전 업에 따른 POS 네이티브 함수 변경
- 시스템 운영 QA 건 처리

작업 결과물

[위메프로 파트너스](#) 

위메프로 앱 내 일부 웹뷰 시스템 기능 개선 및 유지보수

위메프로 앱 내의 웹뷰화면에 대한 유지보수

업무

- 배달주문상세, 티켓구매상세 UI 개선 및 유지보수
 - 배달과 티켓이 구분됨에 따른 ui 분리
- 서울페이 결제 기능 추가
- 후기 등록 화면 개선
 - 다중 이미지 업로드 기능 추가
- 시스템 운영 QA 건 처리

작업 결과물

[위메프로](#) 

위메프로 파트너스 서비스 기능 개선 및 유지보수

위메프로 입점 사장님들을 위한 파트너스 시스템(하이브리드) 기능 개선 및 유지보수

업무

- 단골쿠폰관리 메뉴 UI 개선 및 유지보수
 - 사용자(사장님)가 직접 단골사용자를 관리할 수 있도록 기능 변경
- 단골알림발송 메뉴 UI 개선 및 유지보수
 - 알림 메시지 내용 입력 폼 추가
 - 제목 입력 폼 추가
- 시스템 운영 QA 건 처리

작업 결과물

[위메프로 파트너스](#) 

위메프로 어드민 시스템 기능 개선 및 유지보수

위메프로 앱 내의 홈 구성 및 매장, 메뉴, 쿠폰 등 위메프로 사내에서 사용하는 관리자 시스템(백오피스) 기능 개선 및 유지보수

업무

- 매장 상세 및 메뉴관리 페이지 UI 개선
- 배달관리와 티켓관리가 구분됨에 따른 개선

- 기존 배달 및 티켓이 함께 관리되었는데, 이를 화면 상 메뉴로 분리
- 픽업 메뉴도 분리될 것을 고려하여 개선
- 매장 상세 내 카카오 지도 api 를 네이버 지도 api 로 마이그레이션
 - 배달지역설정에 대한 그리기 기능 추가
- 위메프오 앱의 메인 홈을 구성 및 관리하는 메뉴 기능 개선
 - 배너 순서 변경 기능 추가
 - 메인배너, 중간띠배너, 하단배너에 대한 다중업로드 지원
- 매장에 대한 지역 선택 및 프랜차이즈 선택 공통 모달 UI 개선
 - 주소검색 기능 추가
 - 프랜차이즈 선택 기능 신규 추가
- 시스템 운영 QA 처리
- 현재 위메프오 어드민 시스템은 레거시(앵귤러)와 차세대(리액트)로 구성되어 있으며, 기획팀에서 일정 여유에 따라 리액트로 일부 화면 리팩토링

이노팜 · R&D Engineer

2019-07 — 2021-10

드론 영상 AI 분석 시스템 고도화 사업

드론 영상을 AI 분석하여 항행장애물, 재성충, 작물에 대한 검출결과를 웹 지도 상에 렌더링하는 시스템 개발(파일럿)

업무

- Openlayers 를 활용한 지도화면 및 컨트롤 구성
- AI 검출결과 지도 렌더링
- 프로젝트 별 작물 통계 화면 개발
- 많은 라우터가 확장되고, 지도 화면 이외의 사용자 경험 향상을 위해 SSR 프레임워크인 Nextjs 를 도입

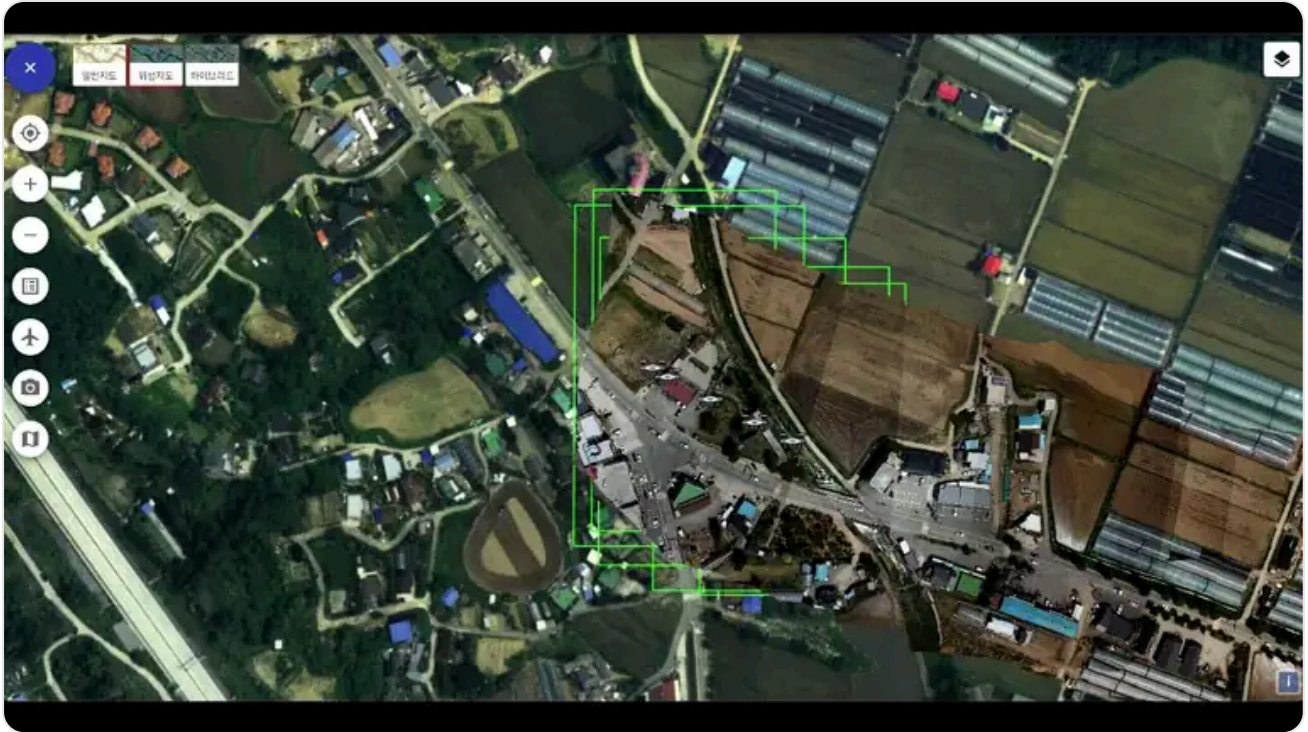
2020년도 창업성장기술개발사업 디딤돌 창업과제

드론 영상 매핑 및 AI 분석 결과 웹 지도 상의 렌더링하는 뷰어 시스템 개발

업무

- 실시간 드론 영상 지도 렌더링
- 실시간 드론 영상 검출 결과 지도 렌더링
- 현재 비행중인 드론 좌표 마커 렌더링
- 확대된 드론이미지 및 검출결과 뷰어 컴포넌트 구현
- docker-compose 를 통해 배포할 때 필요한 nginx 와 react 프로젝트를 도커라이징(Dockerfile) 을 통해 보다 손쉽게 호스팅할 수 있도록 처리

작업 결과물



서울산업진흥원 2020년도 테스트베드 서울 실증지원 사업
서울시 내 자치구별 불법건축물 탐지를 위한 시스템

업무

- 프로젝트 검색 및 목록 조회 페이지 구현
 - 사이드바 컴포넌트 구현
 - 검색 기능 구현
 - 프로젝트 선택 이벤트에 대한 지도상의 항공사진 이미지 노출, 저장되어있는 검출결과 노출 구현
- 사진 업로드 및 프로젝트 생성 수정 삭제 페이지 구현
 - 항공사진을 업로드하는 프로젝트 생성 UI 및 api 호출 구현
- 항공사진으로 불법건축물을 자동 탐지한 결과 데이터 웹 지도 렌더링
 - 항공사진을 업로드한 후 처리된 검출결과와 좌표가 정의된 항공사진을 지도상의 렌더링 구현
- 지도 스와이프 및 분할맵 지도 기능 구현
 - 슬라이드바 지도 컨트롤 컴포넌트 구현
 - 스와이프 시 설정한 두가지의 연도가 지도상에서 스와핑 될 수 있도록 구현
 - 분할맵 지도 컨트롤 컴포넌트 구현
 - 지도화면 분할 구현
 - 각 분할된 지도화면의 이동 및 확대 싱크 설정 구현
 - 분할된 지도영역의 저장된 항공사진을 각자 불러올 수 있도록 Layer on/off 기능 구현

인공지능(AI) 학습용 데이터 구축 2차

드론에서 촬영한 이미지를 통한 월동작물에 대한 분류 및 재배을 면적 측정하는 시스템 개발

업무

- 프로젝트 생성 및 수정, 사진 업로드 페이지 개발
 - React를 활용한 프로젝트관리 UI 컴포넌트 개발
 - 프로젝트 등록, 수정 api 호출 및 응답에 대한 결과 표출 구현

- 다중 이미지업로드 구현
- 분석결과 지도 화면 개발
 - AI 분석 검출결과 GeoJson 형태 → Wfs 형식으로 파싱 후 Openlayers 를 통한 지도 렌더링
- 다이렉트 지오 레퍼런싱 결과 지도상의 렌더링 개발
 - 여러 드론 이미지들을 업로드하여 정사영상으로 반환된 이미지를 지도상의 렌더링
- AI 분석 결과에 대한 범주별 통계 페이지 개발

국립공원공단 드론 영상 보관 및 처리 시스템 구축

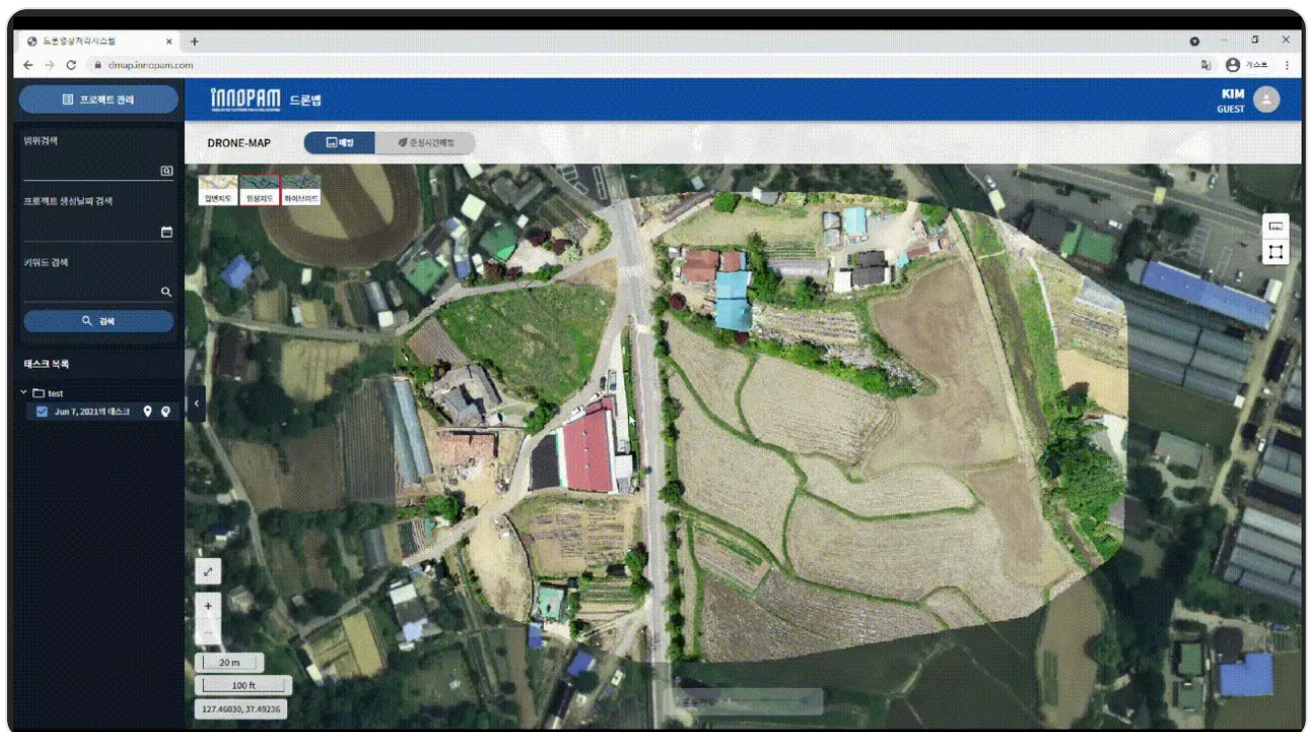
국립공원공단에서 관리하는 국립공원을 드론으로 촬영 한 뒤, 전체 사진을 지오레퍼런싱하여, 그 결과를 배경 위성사진과 비교할 수 있도록 하는 시스템 구현

업무

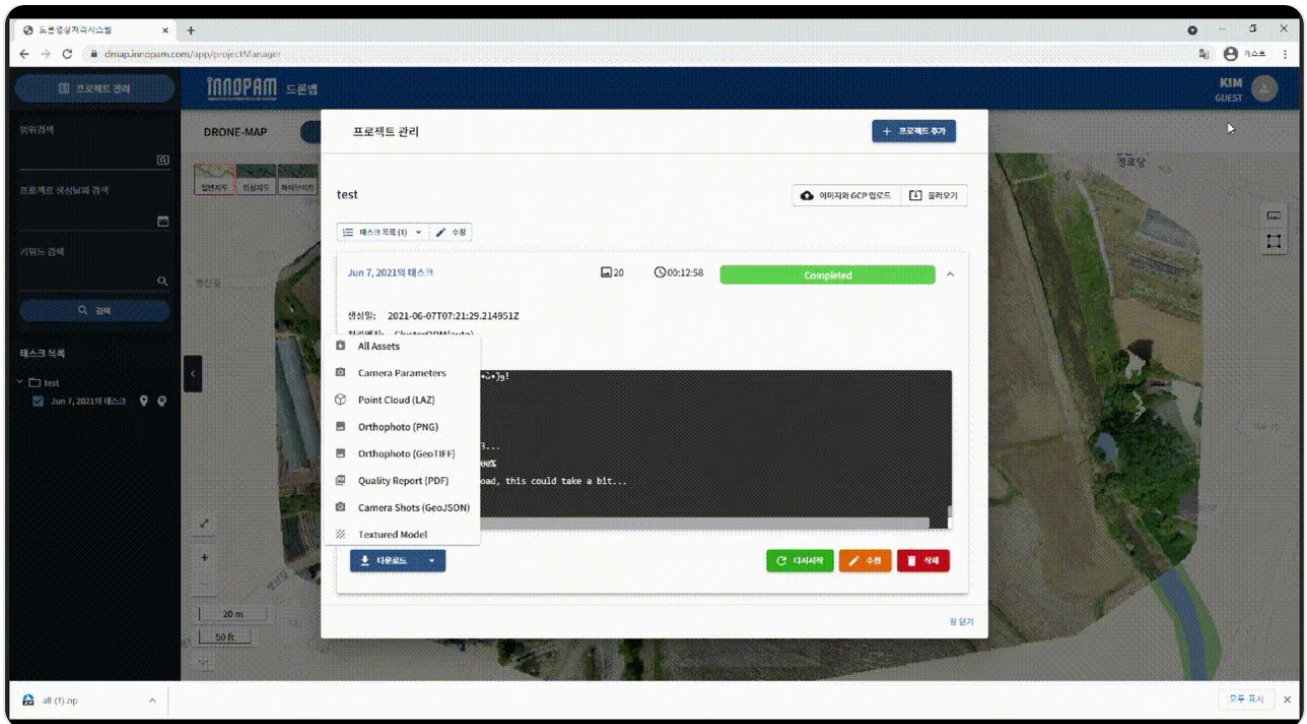
- sso 연계
 - 토큰 기반 개발환경(jwt) 설정
 - 운영환경 빌드 시 인증 요청 URL 만 변경 가능하도록 환경변수 설정
- 지도 UI 개발
 - Openlayers 를 활용한 지도영역 및 컨트롤 , 이벤트 개발
- 프로젝트 및 태스크 관리 기능 개발
 - 프로젝트 관리 모달창과 지도영역과의 상태관리 공유를 위해 Recoil 도입
 - 보다 이쁜 UI를 위한 컴포넌트 라이브러리 Mui 도입
- 관리자 페이지 (사용자, 그룹, 로그, 통계) 기능 개발
 - Nodejs (Express) 를 활용한 API 개발
 - ORM (Sequelize) 도입
 - React 와 Mui 를 활용한 UI 개발

작업 결과물

- 메인 화면



- 태스크 관리 화면



드론영상처리시스템 [🔗](#)

항공안전기술원 2020년 드론 실증도시 구축 사업

드론 기체로 부터 오는 실시간 이미지 및 AI 결과(차, 사람 등) 를 확인할 수 있는 시스템

업무

- 드론으로부터 오는 맵핑 이미지 및 AI 분석결과 (segmentation) 실시간 지도 렌더링
 - 앞 전 산불 검출과 비슷하나 기존에 산불만 검출하던 AI 모델이 차,사람 등을 구분할 수 있도록 변경
 - 지도상에도 실시간으로 오는 검출결과에 대한 구분하여 렌더링(색상)
 - 이미지 뷰어 창의 원본영역과 지오프로세싱 이미지영역, 지도영역의 상태 공유를 위한 redux 도입
- 실시간으로 데이터를 받아서 클라이언트에 보내주는 소켓 서버 구현

장기 체공형 태양광 드론과 인공지능을 이용한 산불 모니터링 플랫폼 개발

장시간 체공하는 드론을 통해 산불 감지 시 즉각 대응할 수 있는 플랫폼(파일럿)

- 해당 솔루션은 건조한 봄, 가을 날씨에 장시간 체공하는 드론을 상시 비행시켜 드론이 보내온 이미지에서 AI를 통한 산불을 검출해 브라우저 지도상에 실시간으로 노출시켜줌으로써 즉각 대응할 수 있는 플랫폼을 구현하기 위함

업무

- 산불감지를 위한 실시간 뷰어 시스템 개발
 - Openlayers 를 통해 지도화면을 구현
 - React 컴포넌트를 활용해 지도컨트롤 UI 및 이벤트 구현
- 지도화면 및 실시간 수신 데이터(사진 및 AI결과) 렌더링
 - 드론에서 실시간으로 보내오는 이미지를 Nodejs 백엔드 서버가 받아 소켓으로 연결되어있는 클라이언트 측에 보내주도록 구현
 - 이 때, 드론이미지는 이미지처리를 담당하는 서버에서 저장하고 실제로 클라이언트측에 보내는 데이터는 이미지 자체가 아닌 s3에 저장된 image url 만 보내도록 구현

국립공원공단 드론관리시스템 구축

국립공원공단에서 관리하는 드론을 보다 효율적으로 관리하기 위한 통합 시스템 구현

업무

- sso 로그인 연계
 - 토큰 기반 개발환경(jwt) 설정
 - 운영환경 빌드 시 인증 요청 URL 만 변경 가능하도록 환경변수 설정
- 드론 신청, 물량, 배차 관리 api 기능개발
 - ORM(JPA)을 활용한 MVC 개발
 - 드론의 종류, 신청과 물량, 배차 관리가 ORM 의 관계 설정만으로는 데이터를 응답받는데에 JPA 의 한계가 있어서, queryDsl 을 도입
 - mybatis, JPQL 과 같은 문자열 형태의 쿼리를 작성하지 않고 자바코드로 SQL문을 작성할 수 있어 컴파일 시 오류를 발견할 수 있음
 - API 명세 및 FE 가 확인할 수 있는 Swagger 문서 작성
 - JUnit 을 통한 단위 테스트 코드 구현

클라우드 기반의 드론 매핑 서비스 플랫폼 개발

AWS 기반의 드론 매핑 플랫폼 신규 개발

업무

- 관리자 (사용자관리, 그룹관리, 권한관리, 지불관리, 통계) 기능 API 개발
 - ORM (JPA) 활용하여 MVC 구현
 - API 명세 및 테스트 가능한 Swagger 도입
 - JUnit 을 통한 단위 테스트 코드 구현
- 권한 별 api 접근 처리
 - Spring Security 를 활용하여 특정 메소드 및 api 경로에 해당하는 권한 설정

신한항업 · Software Engineer

2018-04 — 2019-06

KT 5G NMS 구축 사업

5G 망 확장으로 인한 기지국 및 중계기 위치 기반 시스템 플랫폼 개발

업무

- visual c++ 기반 레거시 시스템을 Spring 기반 웹 시스템으로 변경
- 맵 컨트롤 기능 개발 (POI, 측정기능, 주소검색)

사내 공간정보 포털시스템 구축

각 통계 데이터(지자체별 인구 밀도, 나이대별 인구 분포)를 통한 영업용 통계 지도

업무

- 지도 컨트롤 개발
 - Openlayers 에서 제공하는 히트맵 레이어 구현
 - Openlayers 에서 제공하는 클러스터 레이어 구현

2018 항공사진 관리시스템 구축

항공사진을 통해 해당 지역 내 건축물들을 확인할 수 있는 사내 영업용 시스템

업무

- 항공사진 지도 렌더링
 - 항공사진을 지오서버 사전 업로드 작업
 - 지오서버를 통해 좌표가 정의된 이미지를 Openlayers 를 통해 지도상에 렌더링 작업
 - 연도별 레이어 트리 형태의 UI 구현으로 체크박스를 통해 레이어 On/Off 기능 구현
- 지도 스와이프 및 분할맵 기능 구현
 - 지도 컨트롤 중 스와이프 버튼을 클릭했을 때, 마우스의 이벤트 조합으로 특정 영역에 항공사진을 스와이핑 처리 구현
 - 2분할맵, 4분할맵 구현
 - 연도별 시계열 항공사진을 확인할 수 있도록 지도화면을 비율에 맞게 분할하고, 분할된 맵에서 특정연도를 선택할 수 있도록 구현
- 주소검색 기능 구현
 - 지번주소, 도로명 주소 검색을 통해 지도를 이동할 수 있는 기능 구현
- 나만의 지도 제작 기능 구현
 - Marker, Polygon, Line 과 같은 도형을 그리거나, 거리 및 면적측정을 시도한 지도상의 표시들을 사용자별로 저장할 수 있도록 구현

공간정보기술 · Software Engineer

2014-09 — 2018-04

2018년 교통안전시설물관리시스템(T-GIS) 유지관리 및 기능개선 용역

서울시 교통안전시설물관리시스템 재구축

- 기존 실버라이트 기반 지도에서 Openlayers 로 마이그레이션

업무

- 규제,공사,설계관리 기능 페이지 UI 변경
 - 기존 Asp.net 으로 개발되어있는 백엔드 api 를 spring으로 마이그레이션 하고 디자인에 맞게 화면 UI 수정
- 지도 UI 변경
 - Openlayers 를 적용하여 배경지도(Open api) 및 컨트롤 영역 개발

작업 결과물

- 메인화면
- 지도화면

tgis.e Seoul.go.kr  tgis.e Seoul.go.kr 

서울시 2017년 교통안전시설물 관리시스템(T-GIS) 도로점용공사장 관리 기능

서울시 도로점용 공사 관리를 위한 시스템 구축

- 서울시내 도로에서 특정공사를 시행할 때 공사담당업체가 도로를 얼마만큼 차지할지 시각적으로 지도 상에서 확인할 수 있는 시스템

업무

- 도로점용 영역을 지도상의 렌더링하는 기능 개발
 - 공간DB 를 Polygon 형태로 지도가 로딩될 때 응답받아 지도상에 노출
- 도로점용에 대한 관리(등록/수정/삭제) 기능 개발
 - 클라이언트 측에서 지도상에서 앞으로 점용할 영역을 그려 api를 통해 저장하거나, 선택하여 수정/삭제를 수행할 수 있는 기능 개발

광주광역시 도로 및 상하수도 관리시스템 고도화

투입 인력의 리소스 부족으로 인해 추가 투입

업무

- 관리자 페이지 개발(사용자관리, 그룹관리, 권한관리)
 - 일반 사용자의 개별권한, 비밀번호 초기화, 삭제 및 간단한 정보 수정을 할 수 있는 API 및 화면 개발
 - 각 사용자가 속해있는 그룹에 권한을 할당하거나 추가/수정/삭제 할 수 있는 API 및 화면 개발
 - 롤에 대한 추가/수정/삭제할 수있는 API 및 화면 개발
 - 각 롤이 접근할 수 있는 메뉴를 선택하는 트리구조의 메뉴선택기 개발

대전광역시 교통안전시설물 관리시스템(T-GIS) 구축 용역

대전시 교통안전시설물을 관리하는 시스템 신규 구축

업무

- 공사관리, 규제관리, 설계관리 페이지 및 기능 개발
- 관리자 기능 (사용자, 그룹, 권한, 로그) 개발
- 각 시설물에 대한 지자체별 통계 페이지 개발
- 지도 컨트롤 (시설물 POI, 등록/수정/삭제, 측정기능) 개발
- 주소검색 기능 및 각 레이어 on/off 기능 개발

작업 결과물

- 메인화면
- 지도화면

[대전광역시 교통안전시설물관리시스템\(T-GIS\)](#) 

[대전광역시 교통안전시설물관리시스템\(T-GIS\)](#) 

개발제한구역 항공사진판독시스템 재구축 및 서비스 확대

기 구축된 Visual Basic 프로그램의 ArcObjects 9 버전을 닷넷 기반 ArcObjects 13으로 마이그레이션

업무

- UI 개선 및 버전 업으로 인한 버그 처리
 - Visual Basic의 낡은 디자인을 DotNet의 서드파티라이브러리 DotNetBar 마이그레이션 작업
 - 각 이벤트 액션에 담겨있던 기존 알고리즘이 ArcObjects 버전 업으로 인한 사이드이펙트 수정

서부발전 부동산영상정보 유지관리 용역

상시 시스템 점검 및 유지관리

업무

- 상시로 태안 지역의 서부발전 본사 방문하여 개발된 시스템의 헬스 체크

업무

- 시설물 등록/수정/삭제 기능 개발
 - 울산시에서 관리하는 교통안전시설물에 대해 지도상에서의 이벤트 발생으로인한 등록/수정/삭제 기능 구현
 - 지도상에서 편집모드 발생 시 공간 좌표를 전달받아서 api를 통해 화면상에 노출
- DXF 웹 지도 렌더링
 - DXF 파일을 백엔드 api로 넘겨준 뒤, kabeja 라이브러리를 통해 객체 타입을 비교하여 클라이언트에서 읽을 수 있는 데이터(geojson)로 파싱 후, Openlayers 를 통해 지도상에 도형 렌더링 처리

농지정보시스템 개발 및 정보화사업 발전방향 수립

농어촌공사에서 관리하는 농지정보시스템의 페이지 개발

업무

- DBA 를 통해 작성된 쿼리를 mybatis 에 바인딩 파라미터 영역을 변경하여 mvc 형태로 api 개발 후 화면상에 Extjs 그리드(유료)틀을 이용해 화면상에 노출
 - 검색조건을 통해 노출되는 그리드 리스트가 출력되는 모든 페이지 개발

작업 결과물

부동산 영상 정보 시스템 구축

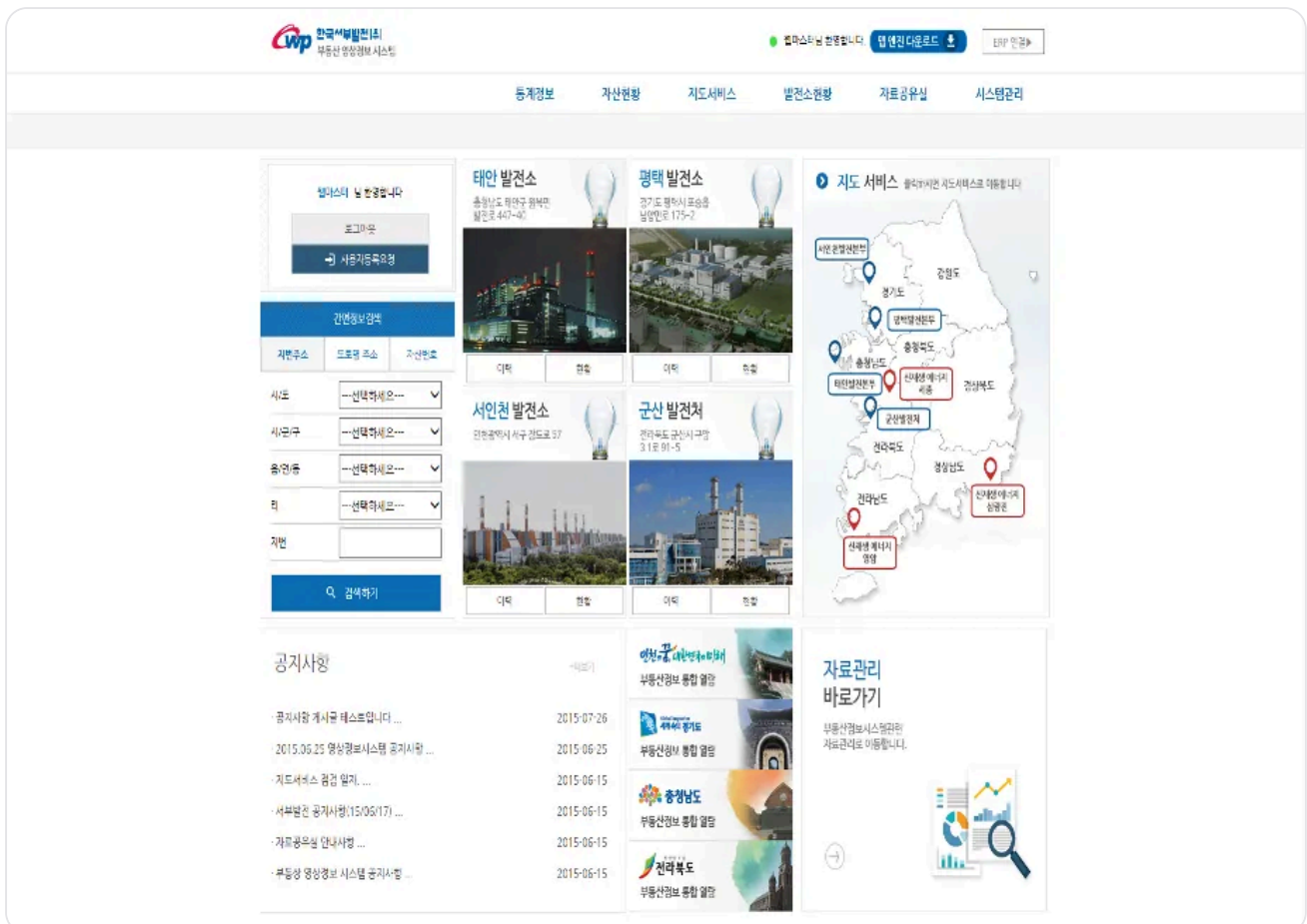
한국서부발전소의 부동산 영상 자산에 대한 관리 시스템 구축

업무

- SSO 로그인 연계
 - 로그인 이 후 저장된 쿠키 토큰 값을 백엔드 요청 시 마다 검사하는 방식으로 구현
 - 자체 권한 관리를 위해 최초 토큰 검사 시 회원정보 키값만 저장
- ERP 연계를 통한 UI 노출
 - SSO 토큰을 활용하여 서부발전소에서 제공한 api 호출에 대한 응답 값을 화면 테이블에 노출
- 부동산 자산 관리 통계 페이지 개발
 - 사업소/사업기능 별 통계 데이터를 화면상에 노출
 - oracle query 를 ibatis 로 DB에서 리턴받은 통계 데이터를 클라이언트로 응답받아 테이블로 노출
- 로그 관리 페이지 개발
 - 클라이언트에서 api를 호출할 때마다 Spring 의 interceptor를 통해 호출되는 로그를 쌓고, 화면상에 서 리스트 형태로 노출

작업 결과물

- 메인페이지



- ERP 연계 데이터 UI 노출

상세정보
자산현황정보
실시간정보연동

소제지	인천광역시 서구 경서동 674 번지			
도로명주소				
순번	자산번호	자산내역	자산구분	면적
1	1010000477	발전소부지	토지	294,909.5
2	2010000005	통합사무실	사무실(본관)	5,999.45
3	2010000014	통합창고	지체창고	3,706.2
4	2010000017	창고	유지류창고	350
5	2010000018	창고	유지류창고	350

- 통계 페이지

- 사업기능
- 자산단위
- 취득가액
- 취득유형
- 건물구조
- 법정동

전체 통계

사업기능	총 계(개수)	총 연면적(m²)	총 필지(개수)
기 력	545	3,559,142.46	136
복 합	332	740,956.634	31
열공급	3	1,066.56	2
신재생	9	52,838.27	5
업 무	37	6,493.094	12

사업소별 통계

사업소	사업기능	총 계(개수)	총 연면적(m²)	총 필지(개수)
태안발전본부	기 력	403	2,811,803.534	111
	업 무	4	124	2
태안건설본부	기 력	4	9,468.46	4
	업 무	4	1,051.14	3
평택발전본부	기 력	120	703,527.81	23
	복 합	55	49,318.12	5
	업 무	25	3,738.594	5
서인천발전본부	기 력	1	1	1
	복 합	165	452,734.394	10
	열공급	3	1,066.56	2
	업 무	3	1,489.23	1

교통지리정보시스템(TGIS) 기능개선 및 고도화 사업

청주시 교통안전시설물에 대한 기구축된 관리시스템(VB)을 웹 시스템으로 재구축

업무

- 청주시 교통시설물에 대한 지역별 / 시설물별 개수에 대한 통계 쿼리 작성 및 화면 노출
 - oracle query 를 특정 조건 값이 바인딩 될 수 있도록 ibatis 형태로 작성 후 api를 개발
 - 특정 조건을 api 로 파라미터를 보낸 뒤 받는 응답 값을 화면에 테이블형태로 지역별/경찰서별 갯수 노출
- 교통시설물 통계 자료 엑셀 다운로드 기능 개발
 - 백엔드에서 Apache POI 를 이용하여 요청한 데이터의 응답결과를 엑셀 시트로 변환 후 클라이언트에 리턴
- 지도상에서 교통시설물 등록/수정/삭제 백엔드 API 구현 및 프론트엔드 지도 액션 이벤트 추가
 - 지도상에서 특정 권한을 갖고 있는 사용자가 그리기 모드를 실행시켰을 때, 특정 시설물을 등록, 수정, 삭제 할 수 있는 기능을 추가

작업 결과물

- 통계 화면
- 지도 화면

교통지리정보시스템 

교통지리정보시스템 