

DSMGR (DSM Github Rank)

DSM(본교) 재학생들의 동기부여를 위한 Github 기여수 랭킹 시스템

개발 기간

- 2020.12.02 ~ 2020.12.07

Github

- https://github.com/woochanleee/DSM-Github_Rank

View On Service

- <http://dsm-rank.site>

기여 항목

Front-end / 1명

- 유저 관련 및 랭킹 기능 개발을 담당하였습니다.(100%)

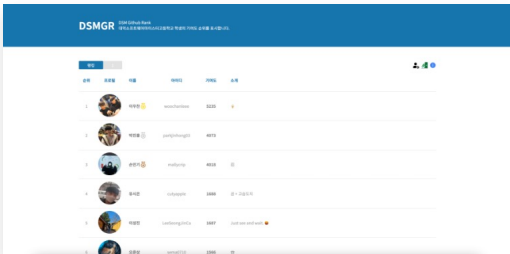
사용 기술

- 개발 언어: Typescript
- 라이브러리: Sapper, PostCSS, Sass(Scss)
- 버전 관리: Git, Github

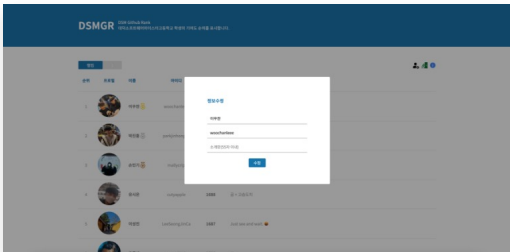
성장경험

- 가상돔을 사용하지 않는 Svelt의 성능과 동작 과정을 이해하며 **항상 가상 돔이 빠르지 않는다**는 것을 깨달았습니다.
- Sapper를 사용하여 검색엔진 최적화를 하며 **서버 사이드 렌더링**의 이해도가 한층 가까워 졌습니다.
- 어떻게 프론트에서 **Clean Architecture**를 적용할수 있을까에 대해 오래 고민하고 코드를 수정하기를 반복하며 Clean Architecture의 이해도가 향상 했습니다.
- 사이트를 개발하고 본교 재학생들에게 홍보하기 위해 SNS에 사이트 주소를 첨부하는데 미리보기 이미지 및 부가 정보가 나타나지 않아 해결방안을 알아보던 중 **Open Graph** 프로토콜에 대해 발견했고 이를 공부해보고 적용해볼 수 있었습니다.

▼ 순위를 볼 수 있는 메인 페이지입니다.



▼ 모달을 이용하여 정보 수정, 로그인, 회원가입을 할 수 있습니다.



▼ 우측 중상단에 아이콘을 이용하여 로그인, 회원가입, 정보 수정, 서비스 소개 모달을 켤 수 있고 로그아웃을 할 수 있습니다.

▼ 비로그인 상태



▼ 로그인 상태





Scarfs 2.0 (Science Assignment Reminder For Students)

기존 과학 과제 제출 시스템의 불편함을 해결하기 위한 온라인 시스템

개발 기간

- 2020.01. ~ 2020.10.

Github

- <https://github.com/Science-assignment-reps-for-student/scarfs-front>

View On Service

- <https://v2.dsm-scarfs.hs.kr>

기여 항목

Front-end / 3명

- 과제 제출 관련 및 게시글 파트를 담당하였습니다.(33.3%)

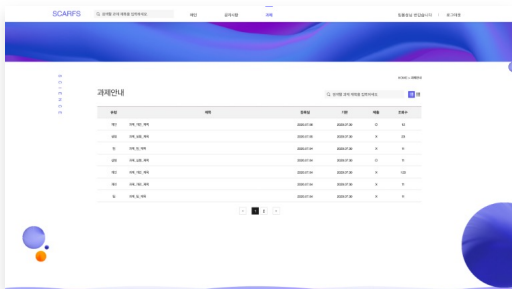
사용 기술

- 개발 언어: Typescript
- 라이브러리: React.js, Redux, Redux-Thunk
- 버전 관리: Git, Github

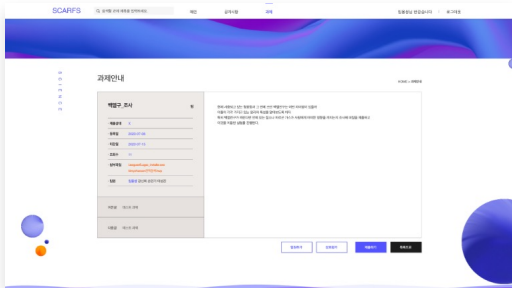
성장경험

- React.js, Redux 와 TypeScript를 함께 사용하여 **강력한 타입 체크**를 지원할 수 있게 되어 코드의 안정성이 향상되었고 작업 속도가 향상되었습니다.
- 프론트 3명이 같은 프로젝트 개발을 진행하였기 때문에 conflict의 위험성이 있다고 판단하였고 **Git-flow 전략**을 사용하여 코드 병합 과정에서의 충돌을 해결할 수 있었습니다.
- Git-flow 전략과 더불어 **Code Review 시스템**을 도입하여 2명 이상의 approve를 받아야 develop 브랜치에 merge시킬수 있도록 하였고 다양한 시각으로 코드를 보며 팀원이 함께 성장할 수 있었습니다.
- 게시글 담당 백엔드가 커스텀 Exception으로 처리하는 반면, 파일 서버 담당자는 기본 예외처리를 사용하여 api 요청 공용 함수에서 status 코드를 어떻게 가져와야 할지에 고민이 있었는데 axios response에 status라는 프로퍼티를 이용하여 **axios 라이브러리의 작동방식을 이해**하고 해결할 수 있었습니다.

▼ 과제를 볼 수 있는 과제 페이지 입니다.(Table View)



▼ 과제 정보를 확인하고 모달을 통해 파일 제출, 팀 생성, 팀원 추가를 할수 있는 페이지입니다.



▼ 팀 생성 모달



▼ 팀원 추가 모달



▼ 과제 제출 모달



▼ 과제 안내(Card View)





FRAME

짧은 글(시, 글귀, 명언)을 공유하는 커뮤니티 시스템

개발 기간

- 2020.10. ~ 2020.12.

Github

- <https://github.com/bookbook-javaproject/Frame-Frontend-Admin>
- <https://github.com/bookbook-javaproject/Frame-Frontend-Main>
- <https://github.com/bookbook-javaproject/Frame-Backend-Core>
- <https://github.com/bookbook-javaproject/Frame-Backend-Admin>
- <https://github.com/bookbook-javaproject/Frame-Frontend-QnA>

View On Service

- <http://s3-2-frame.s3-website.ap-northeast-2.amazonaws.com>
- <http://52.79.253.30:8001>

기여 항목

Front-end / 3명

- 관리자 페이지에서 공지 생성, 신고 게시글, 작가 신청 및 모든 기능을, 또한, 메인 페이지에서 랜딩페이지를 개발하고 모든 프론트 배포와 유지보수를 담당하였습니다.(70%)

Back-end / 2명

- 게시글과 관련된 CRUD기능 개발 및 유지보수를 담당하였습니다.(30%)

사용 기술

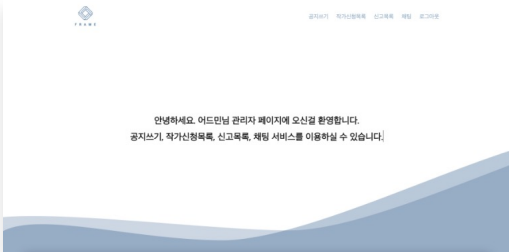
- 개발 언어: JavaScript, Java
- 라이브러리: Vue.js, Nuxt.js, Sass(Scss), Spring Boot, Docker
- 버전 관리: Git, Github

성장경험

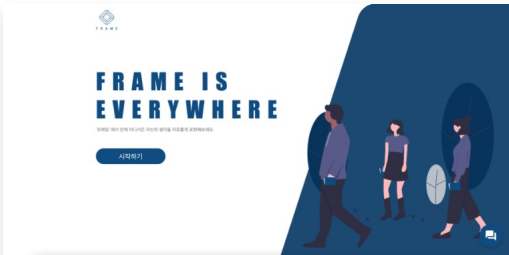
- Nuxt.js를 사용하여 SSR 서비스를 처음 개발하게 되었고, 토큰을 기존에 관리하던 것처럼 브라우저의 내부저장소에 할수 없어서 쿠키를 이용하여 이 문제를 해결하였고 **서버에서 렌더링 한다는 개념**을 이해하는데 도움이 되었습니다.
- Vuex의 action에서 **api 비동기 처리**를 모아 관리할 수 있었고 그 결과 상태관리 관점에서 봤을때 깔끔하게 동작할 수 있었습니다.
- 팀원 중 한명이 본인이 맡은 파트를 다 끝내지 못하여 프로젝트 마감일이 남은 일주일동안 그 부분을 급하게 개발하게 되었다. 이 과정에서 팀 프로젝트에서 **각자가 맡은 역할의 중요성**을 깨닫게 되었고 내가 직접 겪으면서 나는 앞으로도 팀원에게 피해가 가지 않도록 **책임감**을 가져야 겠다고 느꼈습니다.

- rollup.js를 사용하여 메인 페이지에서 라이브러리 형태로 사용할 수 있게 개발을 하였는데 프로젝트 도중 채팅 기능을 빼기로 하여서 UI만 확인할 수 있지만 **GitHub를 통한 라이브러리 제작**을 하며 npm 패키지와 package.json이 어떻게 구성되어있는지 이해할 수 있었습니다.

▼ 어드민 페이지의 기능을 소개하는 메인 페이지입니다.



▼ 스크롤을 통해 서비스 소개를 확인할 수 있는 메인 랜딩 페이지입니다.



Entry System 5.0

원서 접수 및 일정 안내 등 대덕SW마이스터고등학교 신입생 입학 전형을 위한 시스템

개발 기간

- 2020.04. ~ 2020.10.

Github

- <https://github.com/EntryDSM/poodle>

View On Service

- <http://apply.entrydsm.hs.kr>

기여 항목

Front-end / 2명

- 계정 관련 기능과 전형 일정 파트를 담당하였습니다.(50%)

사용 기술

- 개발 언어: TypeScript
- 라이브러리: React.js, Redux, Reudx-saga, Styled-Components
- 버전 관리: Git, Github

성장경험

- Redux-Saga를 이용하여 **비동기 액션 처리**를 경험하며 callback 지옥에서 벗어날 수 있었고 제너레이터 문법에 익숙해질 수 있었습니다.
- 직접 제작한 서비스가 100명이 넘는 **실제 사용자가 이용하게 된다는 점**에서 오류가 없는 서비스를 만들어야 한다는 **책임감**을 갖게 되었습니다.
- 선배들과의 **코드 리뷰**를 통해 더욱 깔끔하고 효율적인 코드를 배울 수 있었고 새로운 시각에 눈을 뜨게된 기회가 되었습니다.
- 인증 번호에 쓰이는 유효기간 타이머를 구현하는 과정에서 다른 곳에서도 쓰일것 같다 판단하여 **useTimer** 라는 **hooks**를 만들어 보며 custom hooks를 잘 만들어 사용하면 효율적으로 코드를 짤 수 있다는 사실을 깨달았습니다.

▼ 입학 전형 일정을 파악할 수 있는 메인 페이지입니다.

▼ 지원자가 계정을 생성할 수 있는 페이지입니다.



FDT (Free Dormitory Teacher)

QR 코드를 이용하여

점호를 무인으로 할수 있고 무인으로 기숙사 서비스를 이용할수 있는 시스템

개발 기간

- 2020.11.25 ~ 2020.11.27

Github

- <https://github.com/F-Dormitory-Teacher/FDT-Admin>
- <https://github.com/F-Dormitory-Teacher/FDT-Server>

기여 항목

- Front-end / 2명
공지사항, 분실물 관련 및 모든 기능 유지보수를 담당하였습니다.(40%)
- Back-end / 2명
유저, 분실물 관련, 출석 리스트 생성을 담당하였습니다.(50%)

사용 기술

- 개발 언어: JavaScript, TypeScript
- 라이브러리: React.js, Mobx, Sass(Scss), Node.js Express.js, TypeORM
- 버전 관리: Git, Github

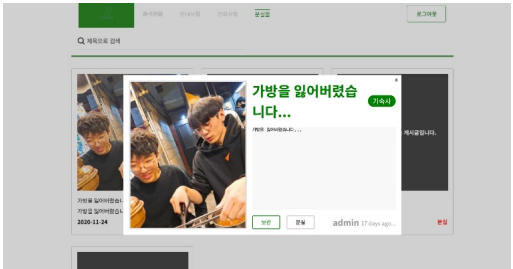
성장경험

- Mobx를 처음 사용해 보고 매우 간단하게 전역 상태를 관리할 수 있다는 점에 놀랐고 **Decorator** 문법에 대해 처음 알게 된 후 **데코레이터에 대해 흥미**를 갖게된 계기가 되었습니다.
- **TypeORM**을 사용하여 객체와 데이터베이스의 변형에 유연하게 대처할 수 있었고 개발을 끝마친뒤 보니 사용한 패턴이 Data Mapper 패턴임을 알고 Active Record 패턴도 있음을 깨달았습니다.
- 매일 아침, 저녁 일정한 시각마다 학생들의 출석 여부 튜플을 넣어 주어야 했는데 이를 해결하기 위해 **node-schedule**을 사용하였고 이 과정에서 **Cron 형식**에 대해 새롭게 알게 되었습니다.

▼ 출석 리스트를 생성하는 스케줄러를 작동시킬수 있고 현재 출석 여부를 확인할 수 있는 페이지입니다.



▼ 분실물의 상태를 수정할 수 있는 페이지입니다.





workout-log

학교 체육수업시간 수행평가인 운동일지 작성을 온라인으로 기록할수 있는 시스템

개발 기간

- 2020.05. ~ 2020.8

Github

- <https://github.com/workout-log/front-end>
- <https://github.com/workout-log/back-end>

View On Service

- <http://workoutlog.club>

기여 항목

Front-end / 1명

- 게시글, 댓글, 마이페이지 등 모든 서비스를 담당하였습니다.(100%)

Back-end / 1명

- 게시글, 댓글, 마이페이지 등 모든 서비스를 담당하였습니다.(100%)

사용 기술

- 개발 언어: TypeScript
- 라이브러리: React.js, Recoil, Node.js Koa.js, Mongoose, Docker
- 버전 관리: Git, Github
- CI/CD: Github Actions

성장경험

- **webpack**을 사용하여 직접 개발환경을 구성함으로써 내부에서 어떻게 동작하는지 이해할 수 있었고 코드를 압축, 최적화 할 수 있었습니다.
- 친구의 추천으로 **Docker**를 이용하여 백엔드 서비스를 배포하였고 이 과정에서 **쉽고 빠른 실행환경 구축**을 할 수 있었습니다.
- 혼자 처음으로 프론트, 백엔드 개발을 모두 진행하다 보니 백엔드에서 api가 어떤식으로 돌아가는지 등 **백엔드 지식**을 쌓을 수 있었습니다.

