

오픈소스 소프트웨어 프로젝트 최종보고서



2023.12.10

2019112127 박세호

2019112162 윤창섭

서론

1. 개요

본 보고서는 오픈소스 소프트웨어 프로젝트 수업의 팀 프로젝트 최종보고서이다. 팀원 협의 결과로 선정한 프로젝트인 '동국대학교 통합 사이트 Portal 재구성'에 대한 요구사항 분석 및 프로젝트 설계, 수행계획 수립을 진행하였다. 요구사항 분석에는 현행 시스템의 분석, 사례 조사를 수행하여 요구사항 명세서를 작성하였다. 프로젝트 설계에서는 구현에 필요한 오픈소스 선정, 사용자 인터페이스 설계, DB 설계를 진행하였다. 수행계획 수립에서는 역할, 개발환경, 협업 규칙을 선정하고, 개발 일정을 수립하였다.

2. 참여자 명단

분야	학번	전공	학년	이름	역할
Frontend	2019112127	정보통신공학	3	박세호	팀장
Backend	2019112162	정보통신공학	3	윤창섭	팀원

목차

1. 프로젝트 개요
 - a. 프로젝트 주제
 - b. 기존 프로젝트 분석
 - c. 선정 오픈소스 분석
 - d. 요구사항 분석
2. 프로젝트 목표
 - a. 개발환경 및 라이선스
 - b. 프로젝트 개선사항 및 개발사항
3. 프로젝트 설계
 - a. 사용자 인터페이스 설계
 - b. 데이터베이스 설계
 - c. API 명세
 - d. 기대효과
4. 프로젝트 일정 및 규칙
 - a. 팀 규칙
 - b. 역할 분담
 - c. 협업 규칙
 - d. 개발 일정
5. 오픈소스 활용
 - a. 개발 과정 오픈소스 개선
 - b. 배포 과정 오픈소스 활용
6. 최종 결과 보고
 - a. 로그인
 - b. 공지사항 조회
 - c. 강의 목록 조회
 - d. 강의 목록 상세 조회
 - e. 강의 수강 신청
 - f. 본인 학적 조회
 - g. 성적증명서 조회
 - h. 학사 정보 조회
7. 결론
 - a. 배포 결과
 - b. 마무리하며

1 장 프로젝트 개요

1.1 프로젝트 주제

해당 프로젝트의 목표는 오픈소스를 활용하여 흩어진 동국대학교 웹 사이트의 기능들을 동국 포탈 사이트에 통합하는 것이다. 수강신청, 학사정보 확인 등 불필요하게 분할된 서비스들을 이용하며 느꼈던 불편함을 통합 포탈 내 하위 서비스로 제공할 수 있도록 해 사용자들의 편의 제공을 위해 해당 프로젝트를 주제로 선정하였다.

궁극적으로 해당 프로젝트에서 본 조는 광범위하게 흩어진 다양한 서비스들을 통합 포탈의 통일성 있는 UI를 통해 사용자 경험을 증진시키고, 빠른 렌더링을 추구하는 것을 목표로 한다.

1.2 기존 프로젝트 분석

통합하고자 하는 시스템인 mdrims, 수강신청, E-Class 사이트에서 통합 기능과 불필요 기능들을 나눠 필요 요소들을 통합 포탈 사이트에 포함시킬 예정이다.

1.2.1 개선 필요사항

1. 기존 DGU 포탈에서 관련 탭을 클릭할 시 해당 사이트로 이동하는 점
2. mdrims의 여러 기능 중 자주 사용하는 기능을 DGU 포탈로 통합
3. 학사관리 페이지에서 해당 수업이 언제 어느 교실에서 진행되는지 메인 화면에 존재하지 않는 점

1.2.2 유지 사항

1. mdrims 내의 전반적인 기능 유지
2. DGU 포탈에서 공지사항 컴포넌트 유지
3. DGU 포탈의 UI 유지

1.2.3 프로젝트 진행 방향

위에서 분류한 분석을 바탕으로 동국 통합 포탈 프로젝트를 진행하고자 한다.

먼저, 기존의 문제는 수강신청 사이트만 하더라도, 정규 학기 수강신청과 계절학기 수강신청이 각각 sugang.dongguk.edu 와 mdrims.dongguk.edu 로 분할되어 있는 상황이다.

추가적으로 기능별로 불필요하게 분할된 웹 페이지들은 각기 다른 UI 를 제공한다. 이렇게 UI/UX 관점에서 개선이 필요함을 인지하여 통합 포탈에서 통일된 UI 와 더 나은 UX 를 제공하려고 한다.

Mdrims 내 실제 사용하는 서비스가 제한적이라는 점을 바탕으로 사용자들이 자주 사용하는 서비스들만을 가져옴으로써 완성도 높은 통합 포탈 서비스를 제공하고자 한다.

1.3 선정 오픈소스 분석

1.3.1 선정 오픈소스 정보

공통

동국대학교 통합형 관리 웹 서비스 (Dongguk Click)

- Github URL: <https://github.com/CSID-DGU/2023-1-OSSProj-M.C.theMax-7>
- 오픈소스 소개: 해당 오픈소스는 동국대학교 메인 페이지, 유드림스, 이클래스 등 학교의 여러 행정 시스템들을 통합한 웹 페이지를 개발하여 학교 구성원들의 편의성을 높인 프로젝트이다.

프론트엔드

1 Axios

- Github URL: <https://github.com/axios/axios>
- 오픈소스 소개: 서버와의 HTTP 통신을 도와주는 자바스크립트 라이브러리로 클라이언트와 서버 간의 데이터를 전송 기능 제공한다.

2 React

- Github URL : <https://github.com/facebook/react>
- 사용자 인터페이스 구축을 위한 자바스크립트 라이브러리로 컴포넌트 기반 개발 환경 제공한다.

3 Next.js

- Github URL : <https://github.com/vercel/next.js>
- Next.js 는 웹 애플리케이션을 구축하기 위한 React 프레임워크로 번들링, 컴파일링 등 리액트에 필요한 툴링을 추상화하고 자동으로 구성한다. 이를 통해 구성에 시간을 보내는 대신 애플리케이션 구축에 집중할 수 있도록 돕는다.

4 Sweetalert2

- Github URL : <https://github.com/sweetalert2/sweetalert2>
- 다양한 디자인과 색감으로 디자인이 된 alert 창을 제공해주는 라이브러리. alert, confirm, prompt 를 지원하며, 사용자에게 따라 커스터마이징이 가능하다는 장점이 있다.

5 Zustand

- Github URL : <https://github.com/pmndrs/zustand>
- Flux 원리를 사용하는 작고 빠르며 확장 가능한 상태 관리 솔루션이다. Hooks 에 기반해 편리한 API 를 통해 상태관리를 쉽게 할 수 있다.

백엔드

1 Spring boot

- Github URL: <https://github.com/spring-projects/spring-boot>
- 오픈소스 소개: Spring Boot 는 초소한의 노력으로 Spring 기반의 생산성 높은 어플리케이션과 서비스를 생성할 수 있도록 도와주는 도구이다. Spring Boot 를 사용하면 java 를 사용하여 시작할 수 있는 독립적인 Java 어플리케이션 배포를 실행할 수 있다.

2 Spring Security

- Github URL: <https://github.com/spring-projects/spring-security>
- 오픈소스 소개: 해당 오픈소스는 보안 프레임워크로, Java 기반의 어플리케이션에서 보안 관련 작업을 수행할 수 있도록 도와주는 프로젝트이다. Spring security 는 Spring 프레임워크와 통합되어 있으며, 사용자 인증, 권한 부여, 세션 관리 및 기타 보안 관련 작업을 지원한다.
-

3 Mockito

- Github URL: <https://github.com/mockito/mockito>
- 오픈소스 소개: Java 언어를 기반으로 한 오픈 소스 mocking 및 테스트 프레임워크이다. Java 기반의 소프트웨어 개발에서 유닛 테스트를 작성하고 테스트 중에 객체를 mocking 하는 데 사용된다. Mocking 은 객체나 메서드의 동작을 가짜 객체 또는 메서드를 대체하여 테스트 중에 원하는 동작을 시뮬레이션 하는 것을 의미한다.

1.3.2 SWOT 분석

Strength	Weakness
1. 하나의 포털로 동국대학교의 모든 서비스를 이용할 수 있음 2. 팀원이 각자 관심있는 분야가 뚜렷하여 프론트엔드 및 백엔드로 역할을 구분할 수 있음.	1. 신규 추가기능 없이, 기존 기능을 활용하여 사이트 재구성.
Opportunity	Threat
1. 통합된 기능으로 더 나은 UX 를 제공할 수 있음 2. 공통된 컴포넌트를 활용해 유지 보수가 원활함 3. 컴포넌트 구조를 통해 지속적으로 다른 사람들이 contribute 할 수 있음	1. 이미 기존 시스템에 익숙한 사람들이 신규 통합 포털 이용에 관심이 없을 수 있음

1.3.3 선정 이유

1. 기본적으로 mdriums 및 eclass 의 통합 페이지로 기능적인 구분에 대한 분리가 되어있어 디자인 외 코드의 수정 및 추가를 통해 구현하고자 하는 서비스 제작이 가능함.
2. 팀원 모두에게 익숙한 JavaScript 를 기반으로 생성된 프로젝트이며, 해당 코드를 TypeScript 로 재구성하기 용이함.
3. Backend 의 경우 Java Springboot 을 기반으로 작성되어있어 팀원이 코드를 이해하기 쉬우며, 해당 코드를 기반으로 더 체계적인 프로젝트로 발전시킬 수 있음.

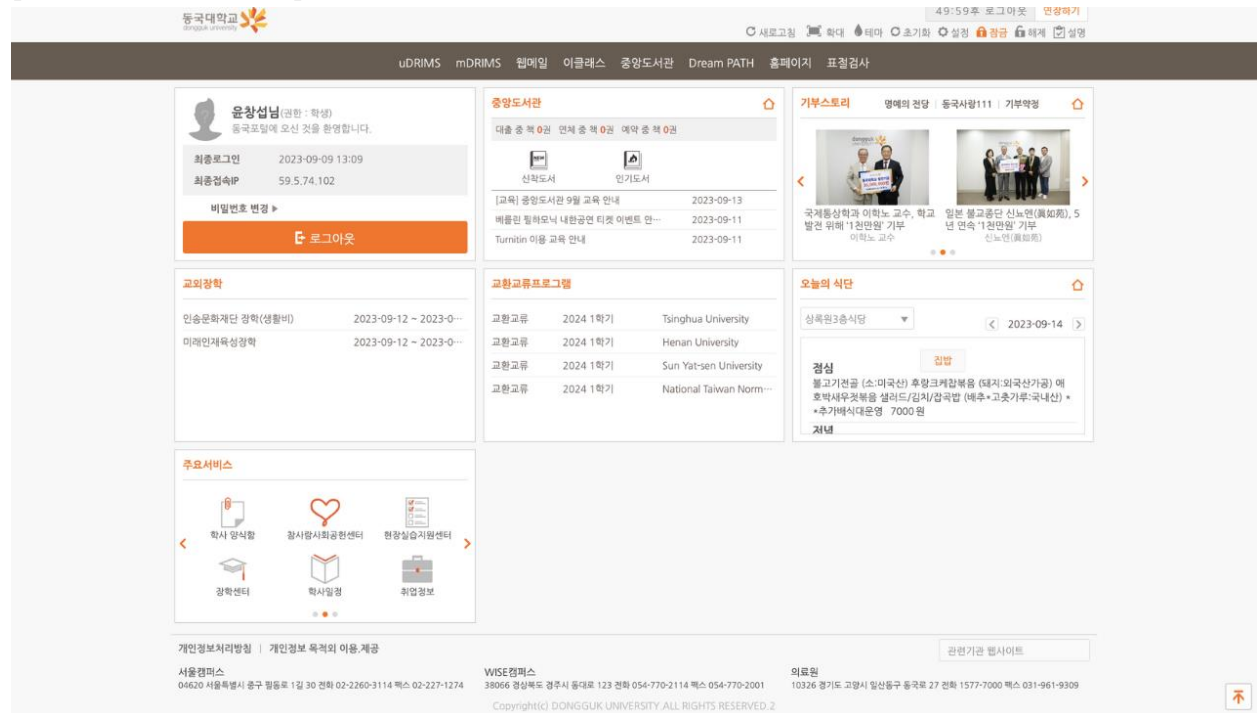
1.4 요구사항 분석

소프트웨어의 기능과 제약조건의 명확화를 위해 현행 시스템 분석과 기존 사례 분석을 통해 요구사항을 수집하였다. 이를 통해 서비스 기능의 통합의 필요성을 느끼고 그에 따라 요구사항 명세서를 작성하였다.

1.4.1 요구사항 수집

개발할 소프트웨어의 기능과 제약조건 등을 명확히 하기 위해 현행 시스템 분석과 기존 사례 분석을 통한 요구사항 수집을 시행하였다.

[현재 DGU 포탈 분석]



[그림 1] 동국대학교 포탈 DGU Portal 메인 페이지 UI

동국대학교 현 DGU 포탈 상의 전반적인 기능을 분석하여, 기능과 개선점을 파악하였다. 기존 포탈의 메인 페이지는 [그림 1]과 같다. 기존 DGU 포탈의 기능은 여러 공지사항들을 확인하기 편리하게 관리하고, 상세 탭에 있는 여러 사이트들로 연결시켜주는 기능이 구성되어있다. 해당 탭에서 다른 사이트로 이동하는 기능 대신, 동일한 사이트 내에서도 해당 사이트의 기능을 사용할 수 있다면 해당 포탈 사이트의 활용도를 높일 수 있을 것이다.

[수강신청 페이지]

현재 수강신청 페이지는 수강신청 기간에만 오픈되며, 별도의 수강신청 페이지가 존재한다. 이는 일정한 기간에 트래픽이 몰리는 수강신청 페이지의 특성상 페이지 및 도메인을 구분한 것으로 보인다. 그러나 api 의 분리와 함께 서비스를 기존 서비스와 분리한다면, 수강신청 서비스만 별도의 서버에서 사용 가능할 것이다. 또한 현재 계절학기 수강신청은 udrims, 희망강의 신청은 udrims 와 mdrims, 본 학기 수강신청은 별도의 수강신청 페이지이므로, 통일성이 없다. 따라서 하나의 페이지에서 관리할 수 있도록 통합된 서비스를 제공한다면 통일성을 증진시킬 수 있을 것이다.

[mdrims 페이지]

The screenshot shows the Dongguk University mdrims main page. The header includes the university logo and navigation links. The main content area displays the user's current status as a student (학부) with ID 2019112162. Below this, there is a section for account management with a table showing account details. The table has columns for account type, ID, password, and user name. The user's account is listed as a student (학부) with ID 2019112162. Below the table, there are links for various services, including a link to the user's account page.

계좌번호	계좌명	계좌종류	계좌비밀번호	계좌비밀번호
2019112162	학생(학부)	학생	2019112162	

[그림 2] 동국대학교 mdrims 메인 페이지

기존 mdrims 페이지에서 학생들은 보통 학사관리 페이지와 성적관련 페이지를 주로 사용하고, 그 외 기능에 대해서는 별도로 많이 사용하지 않는다. 따라서 mdrims 에서 자주 사용하는 기능들에 대해 DGU Portal 에서 바로 사용할 수 있도록 한다면 학생들의 편의성을 증진시킬 수 있다. 또한, DGU 포털에서 학적관리와 학사관리를 동시에 진행하여 새로운 페이지로의 접속과 같은 번거로움을 줄일 수 있다.

2023-2 Open Source Software Project

[E-Class 페이지 분석]



[그림 3] E-Class 내 마이 페이지



[그림 4] E-Class 내 수강 과목 페이지

기존 E-Class 에서 과제 관리 및 공지사항을 관리할 수 있는 페이지는 마이페이지에 존재한다. 해당 마이페이지는 기존 E-Class 의 메인 페이지에서 바로 진입할 수 있는 방법이 부재하고, 수강 과목 페이지로 이동 후 마이페이지로 접근할 수 있다. 학생들이 과제 관리 및 공지사항 관리를 편리하게 사용하기 위하여 메인 페이지에 과제 및 일정 등을 명시할 필요성을 느꼈다. 또한 수강 과목 페이지에서 학생들은 대다수 출석, 과제, 공지사항, 학습자료실 외에는 사용하지 않는 기능들이 많다. 따라서 해당 기능들 중 출석, 과제, 공지사항과 관련된 기능을 DGU Portal 에 통합하여 학생들의 편리성을 증진 시킬 것이다.

1.4.2 요구사항 명세서

현행 시스템 분석과 기존 사례 분석을 통해 분석한 요구사항은 다음과 같다.

사용자	메뉴	필요기능	기능설명
학생	로그인페이지	로그인	학번, 비밀번호로 접속 기능.
	메인페이지	로그아웃	로그아웃 기능.
		공지사항 탐색	공지사항 모아보기 및 링크 이동 기능
		학사일정 탐색	월별 학사일정을 제공 기능
		수업일정 탐색	해당일 수강과목과 시간을 표시 기능
		오늘의 할일	Todo list 로 오늘의 할일 작성 기능
	학적관리 페이지	학적부 메인	학적 기본 상태 정보 열람 기능
		성적 확인	학기별 전체 성적 조회 기능
	수강과목 관리 페이지	현재 수강과목 관리	현재 수강과목 수강취소 기능
		계절학기 과목 관리	계절학기 수강신청 및 취소 기능
		희망강의 신청	다음학기 희망강의 신청 기능
	학습관리 페이지	공지 확인	수강과목 공지 확인 기능
		출결 확인	수강과목 출결 확인 기능
		과제 확인	수강과목 과제 확인 기능

2 장 프로젝트 목표

2.1 개발환경 및 라이선스

2.1.1 개발환경

OS: MacOS, Ubuntu

Code Editor: Visual Studio Code, IntelliJ, PyCharm

Language: Html, CSS, JavaScript, TypeScript, Python

Collaboration Tool: Notion, Github, Slack

2.1.2 라이선스

Spring boot: Apache 2.0 License

Next.js: MIT License

Dongguk Click: Apache 2.0 License

두 라이선스는 혼용 가능하므로, 특허보호를 우선시할 경우 Apache 2.0 License 를, 단순성을 우선시 할 경우 MIT License 를 사용한다. 따라서 본 프로젝트의 라이선스는 Apache 2.0 License 를 채택한다.

License: Apache 2.0 License

2.2 프로젝트 개선 및 개발 사항

2.2.1 개선사항

	개선할 점	개선 방향
1	서비스별로 웹 페이지가 분산되어 있음	분산되어있는 시스템을 한 번의 로그인으로 한 눈에 확인할 수 있도록 개선
2	기존 존재하는 동국 포탈은 공지사항이 부족함	학생들이 관심있어 하는 학사/일반/장학 공지들을 추가로 제공
3	동국 포탈은 자체 페이지 내에서 기능을 제공하지 않음	통합 포탈에서는 자체 페이지 내에서 주요 서비스를 제공
4	React-beautiful-dnd, react-table, jsPDF 의 UI 가 DGU 포탈 UI 와 다름	통일된 UI 로 사용자들에게 서비스를 제공

2.2.2 개발 사항

	추가 기능	개발 내용
1	서버 구축	AWS 를 이용하여 서버 구축
2	데이터베이스 구축	MySQL 을 사용하여 데이터베이스 구축 <ul style="list-style-type: none"> - 유저 테이블 - 공지사항 테이블 - 수강과목 테이블 - 학사일정 테이블
3	주요 서비스 통합	DGU portal 에 mdrims, eclass 를 통합
4	오늘의 할일 / 과목 추가	학습관리 기능에서 오늘의 할일과 오늘의 과목 구현
5	동국대학교 공지 카테고리 추가	학생들이 관심있어하는 공지 사항 알림 추가

3 장 프로젝트 설계

3.1 사용자 인터페이스 설계

로그인 페이지



2023-2 Open Source Software Project

메인 페이지



학적관리

수강관리

학습관리

Dream Path

000님 반갑습니다

Logout

일반공지

[LINC 3.0] 2023학년도 2학기 캠프톤디자인 과제신청 및...

2023.09.23.

학사공지

2024학년도 봄 조기졸업 신청 안내

2023.09.22.

장학공지

2023년도 준제장학회 키움나무 및 귀감나무 장학생 모집 안내

2023.09.21.

학사일정

2023.09.25. ~ 2023.09.27. 2024-1학기 학교현장실습 협약학교 신청 (주강부서 : 사범대학 교직원)

2023.09.25. ~ 2023.09.27. 조기졸업 신청 (주강부서 : 교무팀)

2023.09.25. ~ 2023.09.27. 나노-마이크로디그리 포기 신청 (주강부서 : 교무팀)


오늘의 수업

10:30 데이터사이언스개론

13:00 오픈소스소프트웨어프로젝트

15:00 딥러닝

학적관리 페이지



학적관리

수강관리

학습관리

Dream Path

000님 반갑습니다

Logout

학적 조회

성적 조회

등록 조회

장학 조회

졸업 관리

학적기본

캠퍼스		성명(영문)	여권과 동일하게	
생년월일		성별	<input type="radio"/> 남자 <input type="radio"/> 여자	승적번호

학적변동 및 등록정보

교과과정 이수년도		전과제외여부	<input type="checkbox"/>	전과신청횟수	
최종학적변동		최종변동일자		휴학학기수	

입학 및 재입학정보

입학일자		정원내외여부	<input type="radio"/> 정원내 <input type="radio"/> 정원외	입학구분	
전형구분				공학인증심화대상	
재입학일자		재입학학년		글로벌인재트랙여부	

졸업정보

조기졸업여부	<input type="checkbox"/> 조기졸업	졸업일자		전체이수학기수	
졸업증서번호		학위번호(주전공)		학위명(주전공)	

15

2023-2 Open Source Software Project

학습관리 페이지



[학적관리](#)
[수강관리](#)
[학습관리](#)
[Dream Path](#)

000님 반갑습니다

Logout

오픈소스소프트웨어프로젝트 - 과제정보

과제
제안서 및 발표자료를 팀장님이 pdf로 제출해주시면 됩니다.
정해진 양식은 따로 없습니다.

파일명
T팀번호_발표자료.pdf
T팀번호_제안서.pdf
*팀 번호는 팀 구성 공지를 통해 확인하시면 됩니다.

오늘의 할 일

머신러닝 수업 내용 요약
한이음 프로젝트 2차영상 제출

데이터사이언스개론 - 공지사항

안녕하세요

파이썬을 처음하시는 분들을 위한 기초강의입니다.
혹시 설치를 못하시는 분들이나 프로그래밍이 처음이신분들은
이 강의를 시청해주시면 감사하겠습니다.

공학선형대수학 - 출결정보

수업일지(시간)[보강일자]	출석	지각	결석	취업 결석	유고 결석	미처리
2023-09-05 (1200 ~ 1330)	1	0	0	0	0	0
2023-09-07 (1200 ~ 1330)	1	0	0	0	0	0
2023-09-12 (1200 ~ 1330)	1	0	0	0	0	0
2023-09-14 (1200 ~ 1330)	1	0	0	0	0	0
2023-09-19 (1200 ~ 1330)	1	0	0	0	0	0

딥러닝 - 과제정보

Lecture Note 2 Homework and Lab Assignment

Submit three files: 1 Assignment and 2 Python Code

오늘의 수업

10:30 데이터사이언스개론

13:00 오픈소스소프트웨어프로젝트

15:00 딥러닝

수강관리 페이지



[학적관리](#)
[수강관리](#)
[학습관리](#)
[Dream Path](#)

000님 반갑습니다

Logout

수강과목관리

희망강의신청

계절학기관리

학적상태

학번

대학

학과

전공

강의년도

강의학기

거물학기

교과과정

교과명

교과명

주요구분

대학

학부(과)/전공

교과목

교과목명

수업형태

수업형태

조회

확인초기화

개설강의목록 총 0건

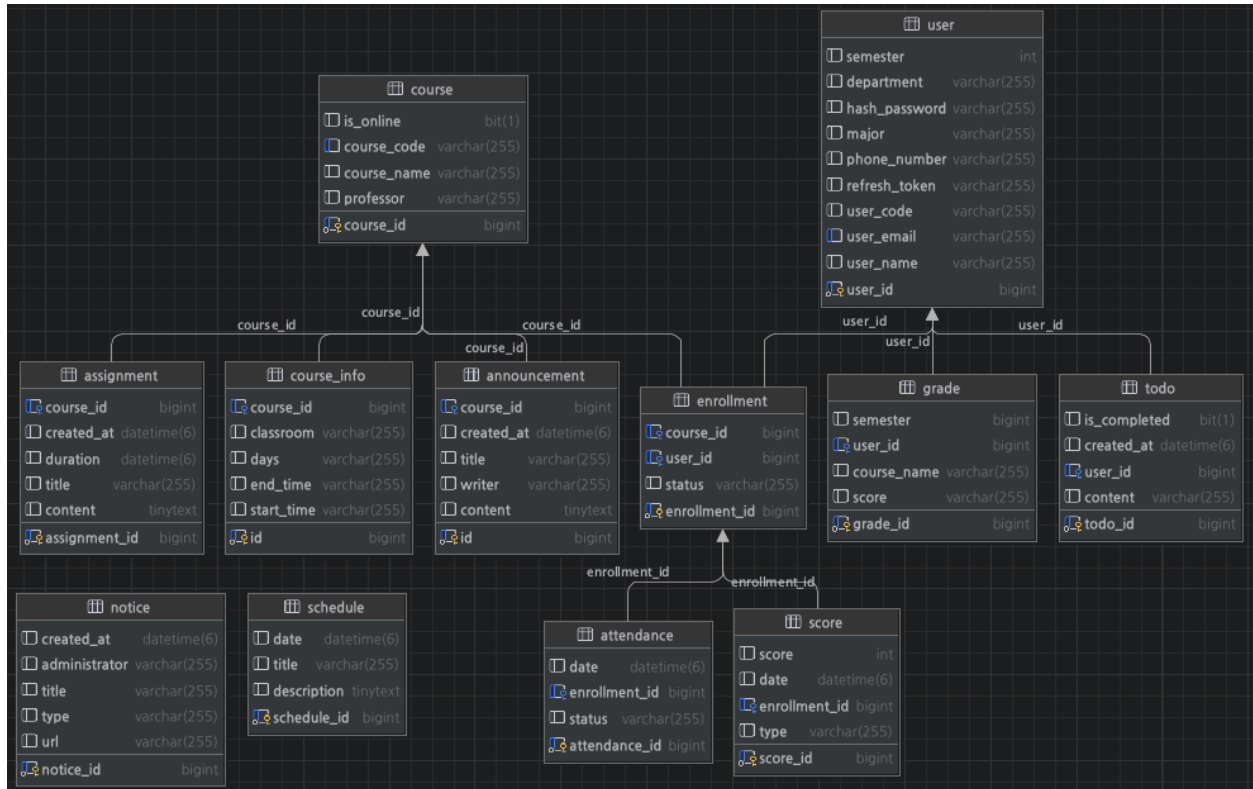
신청	이수	학수강좌번호	교과목명	학점	정원	여분	수강료	수업형태	요일/교시/시간	강의실	담당교원	비고
조회된 데이터가 없습니다.												

수강신청가능학점

수강신청목록 총 0건

식제	이수구분	학수강좌번호	교과목명	학점	수강료	제강여부	재수강여부	요일/교시/시간	비고
조회된 데이터가 없습니다.									

3.2 데이터베이스 설계



사용자의 데이터를 저장하기 위해 관계형 데이터베이스인 MySQL 을 사용하였다. 이는 본교 공지사항 및 학사일정에 대한 데이터를 크롤링하여 살펴본 결과 정형데이터의 특징을 가진 데이터가 대부분이었기 때문에 해당 관계형 데이터베이스인 MySQL 을 기준으로 데이터베이스를 설계하였다.

3.3 API 명세

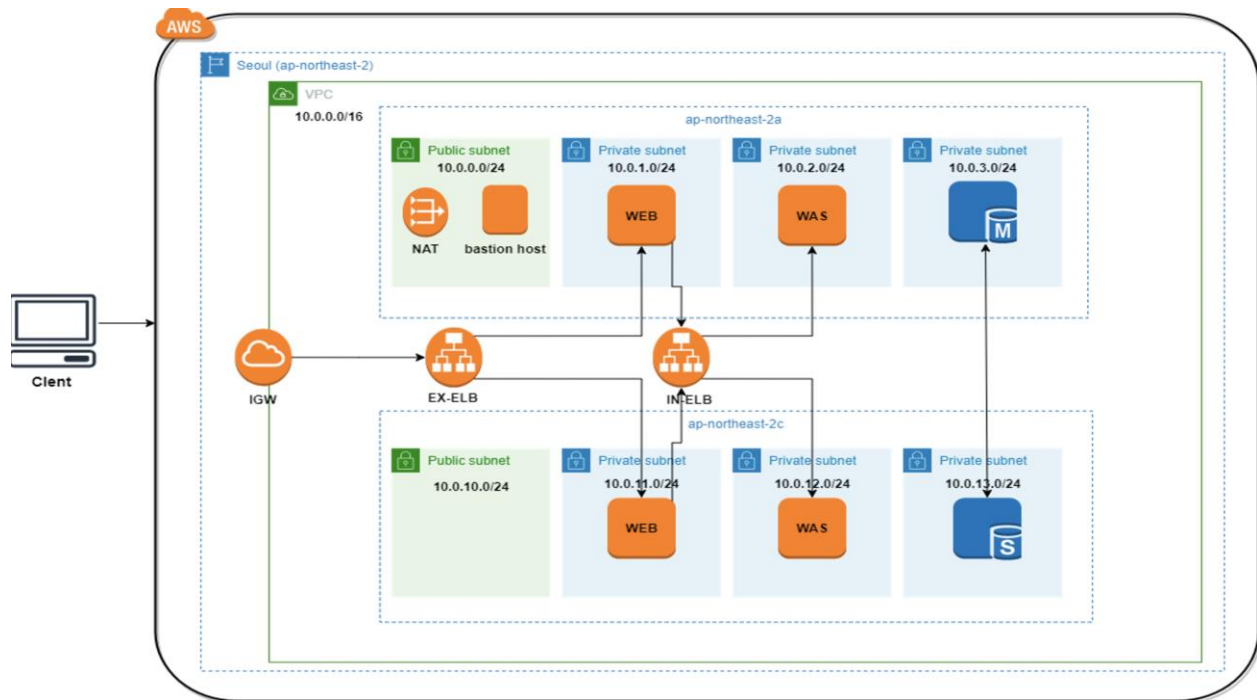
API 명세서				
Aa 기능	HTTP method	API Endpoint	Test	Status
로그인 페이지			<input type="checkbox"/>	
로그인	POST	api/v1/user/login	<input checked="" type="checkbox"/>	Done
공통			<input type="checkbox"/>	
로그인 연장	POST	/api/v1/user/refresh	<input checked="" type="checkbox"/>	Done
로그아웃	POST	/api/v1/user/logout/{user_code}	<input checked="" type="checkbox"/>	Done
메인 페이지			<input type="checkbox"/>	
컴포넌트 불러오기	GET	/api/v1/main/{user_code}	<input checked="" type="checkbox"/>	Done
학적관리 페이지			<input type="checkbox"/>	
학적 기본 정보 불러오기	GET	/api/v1/info/{user_code}	<input checked="" type="checkbox"/>	Done
전체 성적 불러오기	GET	/api/v1/grade/{user_code}	<input checked="" type="checkbox"/>	Done
직전학기 성적 불러오기	GET	/api/v1/before/{user_code}	<input checked="" type="checkbox"/>	Done
수강관리 페이지			<input type="checkbox"/>	
모든 강의 불러오기	GET	/api/v1/course	<input checked="" type="checkbox"/>	Done
교과목으로 검색하기	POST	/api/v1/course	<input checked="" type="checkbox"/>	Done
교수명으로 검색하기	POST	/api/v1/course	<input checked="" type="checkbox"/>	Done
강좌번호로 검색하기	POST	/api/v1/course	<input checked="" type="checkbox"/>	Done
내 수강목록 불러오기	GET	/api/v1/register/{user_code}	<input checked="" type="checkbox"/>	
수강 신청하기	POST	/api/v1/register	<input checked="" type="checkbox"/>	
수강 취소하기	DELETE	/api/v1/register	<input checked="" type="checkbox"/>	
내 수강목록 불러오기 (내부 api)	GET	/api/v1/enroll/{user_code}	<input checked="" type="checkbox"/>	Done
수강 신청하기 (내부 api)	POST	/api/v1/enroll	<input checked="" type="checkbox"/>	Done
수강 취소하기 (내부 api)	DELETE	/api/v1/enroll	<input checked="" type="checkbox"/>	Done
학습관리 페이지			<input type="checkbox"/>	
컴포넌트 불러오기	GET	/api/v1/lms/{user_code}	<input checked="" type="checkbox"/>	Done
학습관리 페이지 디테일	GET	/api/v1/lms/{user_code}/course/{course_code}	<input checked="" type="checkbox"/>	Done
오늘의 할일 등록하기	POST	/api/v1/todos	<input checked="" type="checkbox"/>	Done
오늘의 할일 삭제하기	DELETE	/api/v1/todos/{todo_id}	<input checked="" type="checkbox"/>	Done
오늘의 할일 수정하기	PATCH	/api/v1/todos/{todo_id}	<input checked="" type="checkbox"/>	Done
오늘의 할일 체크	POST	/api/v1/todos/{todo_id}	<input checked="" type="checkbox"/>	Done

Notion 의 Database 기능을 활용하여 API 명세서를 자세하게 작성하였다.

페이지 별로 API 를 구분하여 설계하였고, 엔드포인트를 최대한 줄여 간단하게 API 를 구성하는 것을 주 목적으로 설계하였다.

해당 API 들은 Status 를 구분하며, Done, In progress, Not started 해당 3 가지 status 를 통해 진행상황을 팀원들과 공유한다.

3.4 인프라 아키텍처 설계



3 Tier Architecture 로 AWS 인프라 환경구성을 설정하였다.

이는 소프트웨어 시스템을 세 가지 주요 계층으로 나누는 아키텍처 패턴이다.

해당 계층은 사용자의 인터페이스를 처리하고 클라이언트와 상호작용하는 계층, 비즈니스 로직을 처리하고 애플리케이션의 핵심 기능을 실행하는 계층, 데이터베이스 및 데이터 저장소와 상호 작용하여 데이터를 저장하고 검색하는 계층으로 나뉜다.

이를 통해 각 계층을 독립적으로 개발 및 유지보수할 수 있으므로 코드의 모듈화가 쉽고, 비즈니스 로직은 비즈니스 계층 중점적으로 구현되므로, 해당 로직을 다른 클라이언트나 애플리케이션에서 재사용할 수 있다.

3.4 기대효과

1. 페이지 하나에 학생들이 사용하는 대부분의 기능이 통합 되어있어 사용자의 편의성 증진
2. PC, 모바일 환경을 모두 고려한 개선을 통해 사용자의 편의성 증진
3. 프로젝트 수행을 통한 팀원들의 웹 시스템 구조 이해 상승
4. 협업 툴 (Github, Notion, Slack)을 활용하여 팀원들의 커뮤니케이션 능력 향상
5. 오픈소스를 통해 개발하는 과정에 대한 이해 상승

4 장 프로젝트 일정 및 규칙

4.1 팀 규칙

1. 수업 시간 (월, 수 13:00 ~ 15:00) 외 매주 금요일 17:00 대면으로 회의를 진행한다.
2. 이전 회의의 Agenda 및 각자 진행상황에 대해 논의한다.
3. 추가 회의가 필요할 경우 추가적인 대면 회의를 진행한다.
4. 모든 팀원이 비대면 회의를 희망할 경우 비대면 회의로 진행한다.
5. 회의 내용은 Notion 의 팀 스페이스 중 회의록 페이지에 작성한다.

4.2 역할 분담

이름	역할	특이사항
박세호	Frontend Developer	전반적인 UI 구현
윤창섭	Backend Developer	전반적인 API 및 DB 처리

4.3 협업 규칙

4.3.1 Github 협업 규칙

1. 전체적인 협업 flow 는 Github flow 를 따른다.
2. Fork 한 저장소를 각자 local 로 가져와 수정한다.
3. 수정한 코드는 add -> commit -> push 후, upstream 에 Pull Request 를 수행한다.
4. main branch 로부터 dev branch, prod branch 를 구성한다.
5. 추가되는 기능에 대해서는 feature branch 를 생성하여 각 기능별 branch 를 구성한다.
6. Pull Request 시 Code Review 이후 Merge 한다.
7. Commit 규칙은 다음과 같다.
 - Feat: 새로운 기능 추가
 - Fix: 버그 수정
 - Docs: 문서 수정
 - Style: 코드 formatting, 세미콜론 누락 등 코드 자체의 변경이 없는 경우
 - Chore: 패키지 매니저 수정 및 기타 수정
 - Design: CSS 등 사용자 UI 변경

4.3.2 Issue 활용

<input type="checkbox"/> 5 Open ✓ 1 Closed	Author ▾	Label ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<input type="checkbox"/> 프론트-백엔드 로그인 연결 #27 opened 2 hours ago by sayyyho						
<input type="checkbox"/> 크롤링 데이터 데이터베이스 적재 #25 opened yesterday by kurtyoon 3 tasks						
<input type="checkbox"/> 로그인 페이지 구조 설계 #22 opened 2 days ago by sayyyho 2 tasks						
<input type="checkbox"/> 하위 컴포넌트 세분화 #20 opened 2 days ago by sayyyho 2 tasks						
<input type="checkbox"/> 메인화면 컴포넌트 반환 API #19 opened 2 days ago by kurtyoon 2 tasks done						

Github 레포지토리의 Issue 탭에 Todo 인 상황 혹은 In progress 에 대한 상황을 작성하고 공유한다. 해당 Issue 번호로 각자의 로컬 레포지토리에 브랜치를 생성하여 Pull Request 시에 해당 Issue 를 언급하여 공유한다. 해당 전략을 사용하여 Merge Conflict 의 발생 가능성을 줄인다.

4.3.3 Pull Request 활용

Feat/18 #23 Edit <> Code

Merged kurtyoon merged 21 commits into `dev` from `feat/18` yesterday

Conversation 0 Commits 21 Checks 0 Files changed 93 +829 -44

kurtyoon commented yesterday · edited Member

TO-DO

- 로그인 관련 JWT 인증 로직 작성 #18
- ☒ JWT Service 작성
- ☒ 테스트를 위한 SignUp, SignIn 로직 작성
- ☒ AccessToken을 통한 인증 로직 작성
- ☐ RefreshToken을 통해 로그인 연장 로직 작성
- ☒ 유효하지 않은 AccessToken으로 접근 시 예외 처리

논의 사항

기타 사항

kurtyoon added 21 commits yesterday

- Chore: Spring Security 관련 패키지 설치
- Docs: JWT 관련 변수 설정
- Docs: 수정된 엔티티 설명을 sql 파일에 반영
- Feat: 로그인 유무에 따라 User 엔티티에 토글값 설정
- Feat: client 환경에 맞게 Cors Domain 변경
- Feat: JWT 인증 관련 필요 함수 Bean 등록
- Feat: JWT 인증 관련 Security 설정 등록
- Feat: UserRepository에서 refreshToken 값을 사용해서 User를 찾을 수 있도록 함수 추가
- Feat: 사용자에 대한 권한이 있는 권한이 시 사용하는 권한 작성

Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Notifications Customize
Unsubscribe
You're receiving notifications because you modified the open/close state.

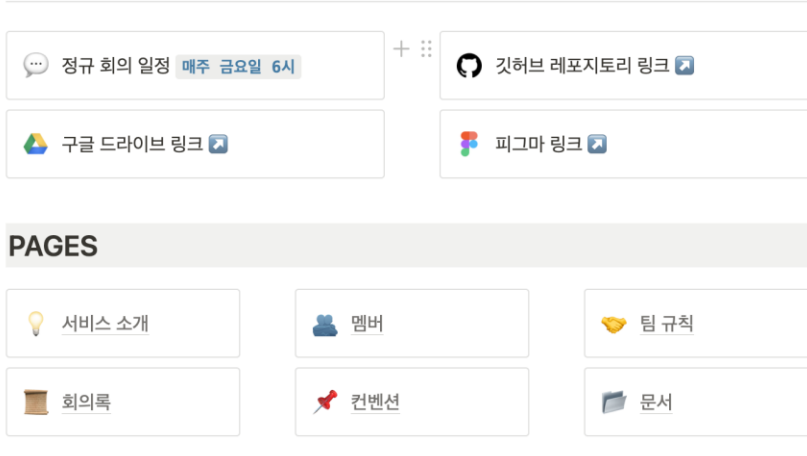
1 participant
277bafa

Lock conversation

다음과 같이 개발 이후 특정 프로젝트에 대한 변경사항을 제안하고, 다른 개발자들과 이를 검토 및 논의한 후, 최종적으로 해당 변경사항을 반영할 수 있도록 하였다. 다른 개발자들은 해당 Pull Request 를 검토하고, 필요한 경우 피드백을 제공할 수 있다. 검토 후, Pull Request 가 승인되면 변경 사항이 메인 프로젝트로 병합된다. 반면, 추가적인 수정이 필요한 경우 개발자는 피드백을 반영하여 수정하고, 수정된 변경사항을 다시 푸시한다.

4.3.4 Notion 협업 규칙

Lonely - 외롭조



1. 문서 작업과 관련된 내용은 문서 페이지에 정리한다.
2. 팀 회의와 관련된 내용은 회의록 페이지에 정리한다.

4.4 개발 일정

분류	내용	담당	9월 4주차	10월 1주차	10월 2주차	10월 3주차	10월 4주차	10월 5주차	11월 1주차	11월 2주차	11월 3주차	11월 4주차	12월 1주차	12월 2주차
오픈소스 분석	로그인 기능 분석	윤창섭												
	성적증명서 기능 분석	박세호												
프로젝트 설계 구체화	화면설계서 구체화	박세호												
	DB 명세서 구체화	윤창섭												
	API 명세서 구체화	윤창섭												
학사정보 수집	공지사항 수집	윤창섭												
	강의목록 수집	박세호												
인프라 설계	AWS 설계	윤창섭												
	Terraform 작성	윤창섭												
메인 페이지	메인페이지 기능 구현	윤창섭												
	메인페이지 화면 구현	박세호												
수강관리 페이지	현재 수강과목 관리 기능 구현	윤창섭												
	화면 구현	박세호												
	희망강의 및 개설학기 기능 구현	윤창섭												
	화면 구현	박세호												
	수강신청 기능 구현	윤창섭												
	화면 구현	박세호												
학습관리 페이지	현재 수강과목 학사관리 기능 구현	윤창섭												
	화면 구현	박세호												
학사관리 페이지	학적 요약 기능 구현	윤창섭												
	화면 구현	박세호												
	전체성적 조회 기능 구현	윤창섭												
	화면 구현	박세호												
	성적증명서 출력 기능 구현	윤창섭												
	화면 구현	박세호												
중간 발표 준비		공통												
도메인 테스트 및 단위 테스트		윤창섭												
최종 발표 준비		공통												

5 장 오픈소스 활용

5.1 개발 과정 오픈소스 활용 및 개선사항

Dongguk Click

Apache-2.0 license 로 공개된 오픈소스인 DonggukClick 의 전반적인 디렉토리 구조를 채택했다. 특히, 성적조회 페이지의 출력은 해당 오픈소스 코드를 활용했다.

이외에, UI 부분은 기존 학생들에게 익숙한 <https://portal.dongguk.edu/> 기반으로 구현했다..

SweetAlert2

로그인, 로그아웃, 수강신청 시 사용자에게 현재 상태를 명확하게 알리기 위해 sweetalert2 를 사용하며 사용자 친화적인 UI/UX 를 개선했다.

5.2 배포 과정 오픈소스 활용

완성된 애플리케이션의 배포에 있어서 우분투, 도커, 테라폼을 사용하여 Ubuntu 가상 서버를 생성할 수 있었다.

테라폼은 하시코프에서 개발한 오픈소스 코드형 인프라스트럭처 도구이다. IaC 는 인프라를 코드로 정의하고, 이 코드를 사용하여 인프라를 프로비저닝, 구성, 관리하는 것을 의미한다.

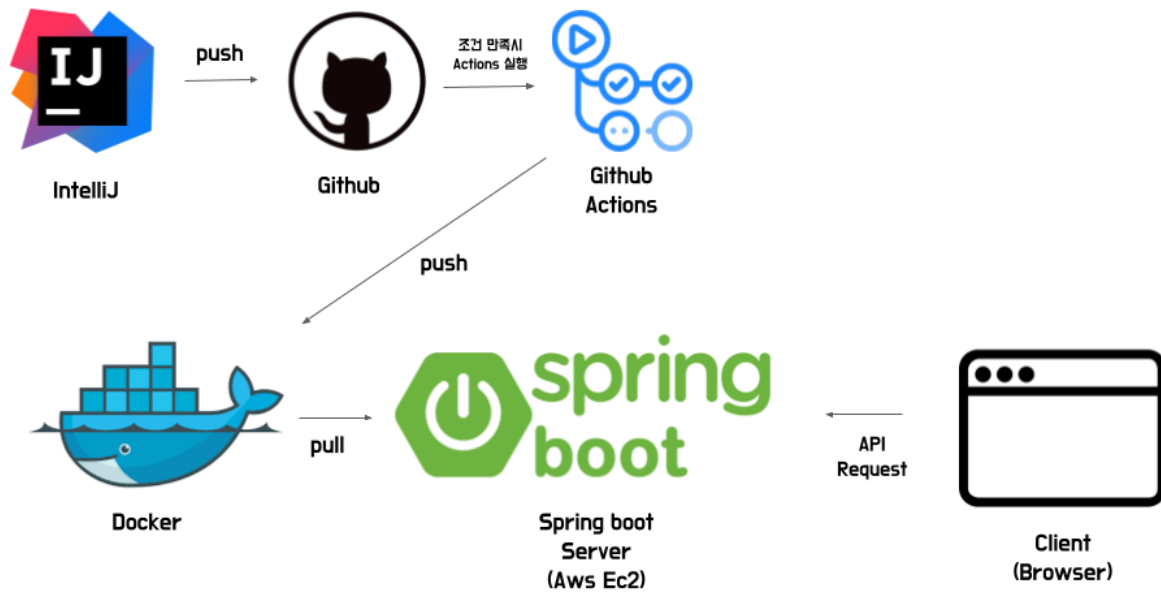
테라폼은 다양한 클라우드 플랫폼과 온프레미스 환경을 지원한다. 이를 통해 코드만으로 다양한 환경에서 인프라를 일관되게 구축하고 관리할 수 있었다.

본 프로젝트에서 테라폼을 사용하여 aws 에 3-tier 아키텍처를 구성하였고, 구성된 서버는 각각 ubuntu 를 사용하여 세팅했다.

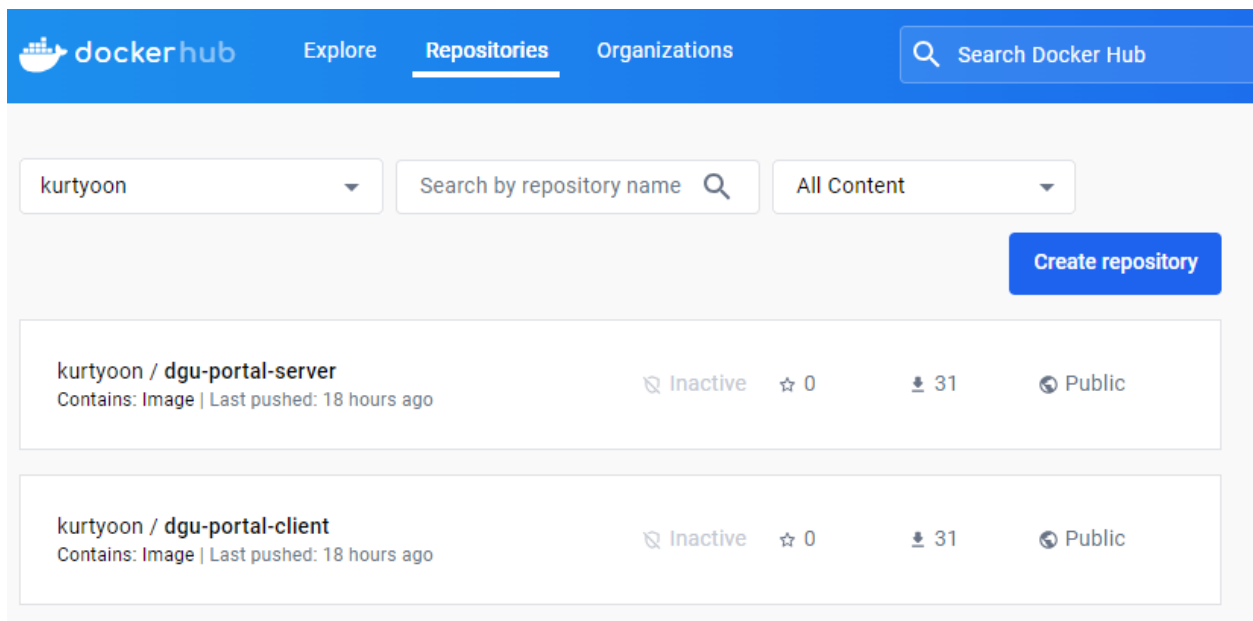


2023-2 Open Source Software Project

Github 의 actions 기능을 활용하여 개발자가 release 브랜치로 feature 프로젝트를 merge 하는 순간 트리거가 발동되어 프로젝트의 버전을 업데이트 할 수 있도록 구성하였다.



Github actions 에서 build 한 docker image 들은 docker hub 로 push 된다. 이후 actions 에서 서버 측으로 진입을 하게 되고, 해당 서버에서 docker hub 에 있는 이미지들을 pull 받아 현재 실행중인 어플리케이션을 다운하고 새로 pull 받은 이미지들을 어플리케이션으로 up 한다.

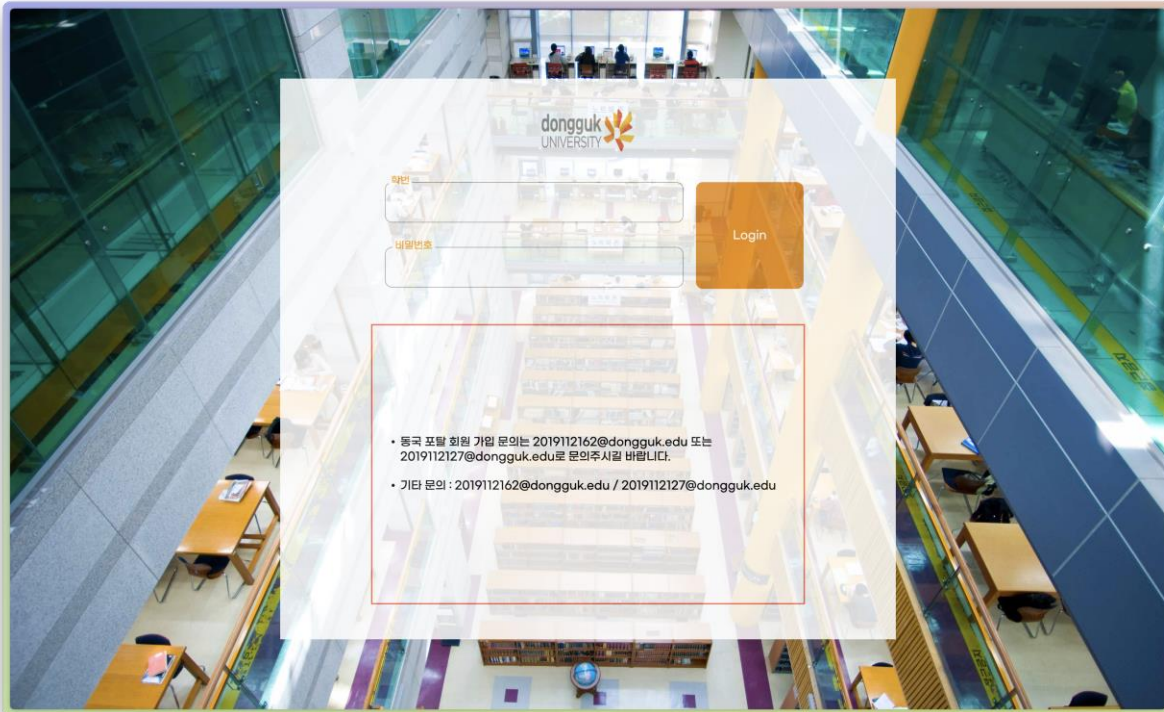




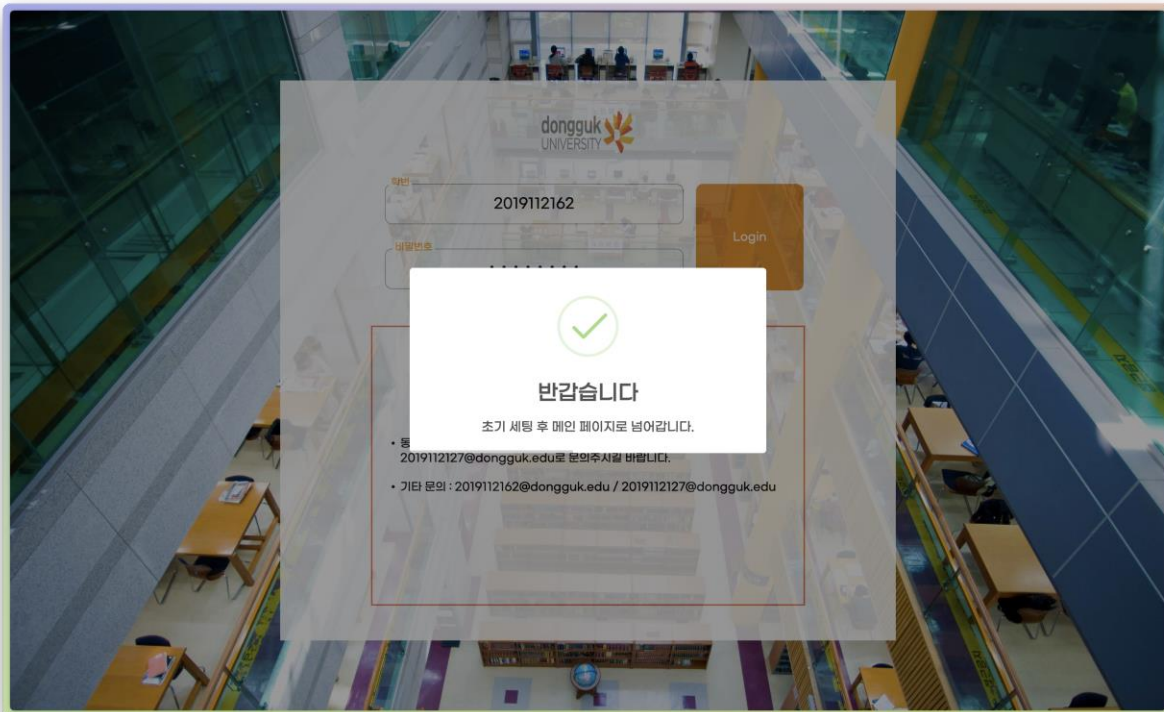
서버에 해당하는 ip 에 https 를 적용하기 위해 소유하고 있던 도메인을 활용하였다.
이후 certbot 을 활용하여 https 를 적용하였다. Certbot 은 Let's encrypt 를 이용해서
ssl 인증서를 발급, 갱신할 수 있게 도와주는 무료, 오픈소스 도구이다. Nginx 와 certbot 을
조합하여 무료로 ssl 인증서를 발급받고, 이를 토대로 https 로 사용할 수 있도록 하여 보안에
대비하였다.

6 장 최종 개발 결과

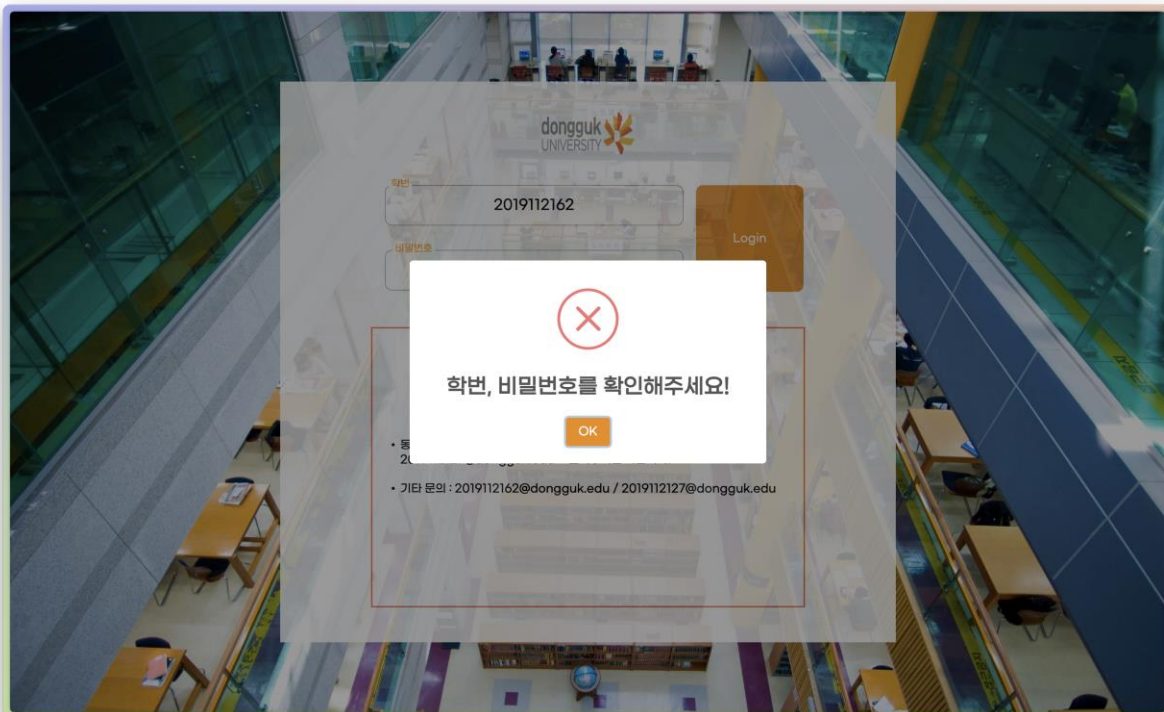
[login]



동국대학교의 대부분의 홈페이지는 학생이 입학 혹은 편입 시에 학번과 그에 해당하는 비밀번호가 주어진다. 따라서 회원가입 기능을 외부로 나타내는 것이 아닌 내부에서 처리하는 것으로 정했다. 이렇게 부여 받은 학번과 비밀번호를 통해서 본 동국포탈에 진입할 수 있다.

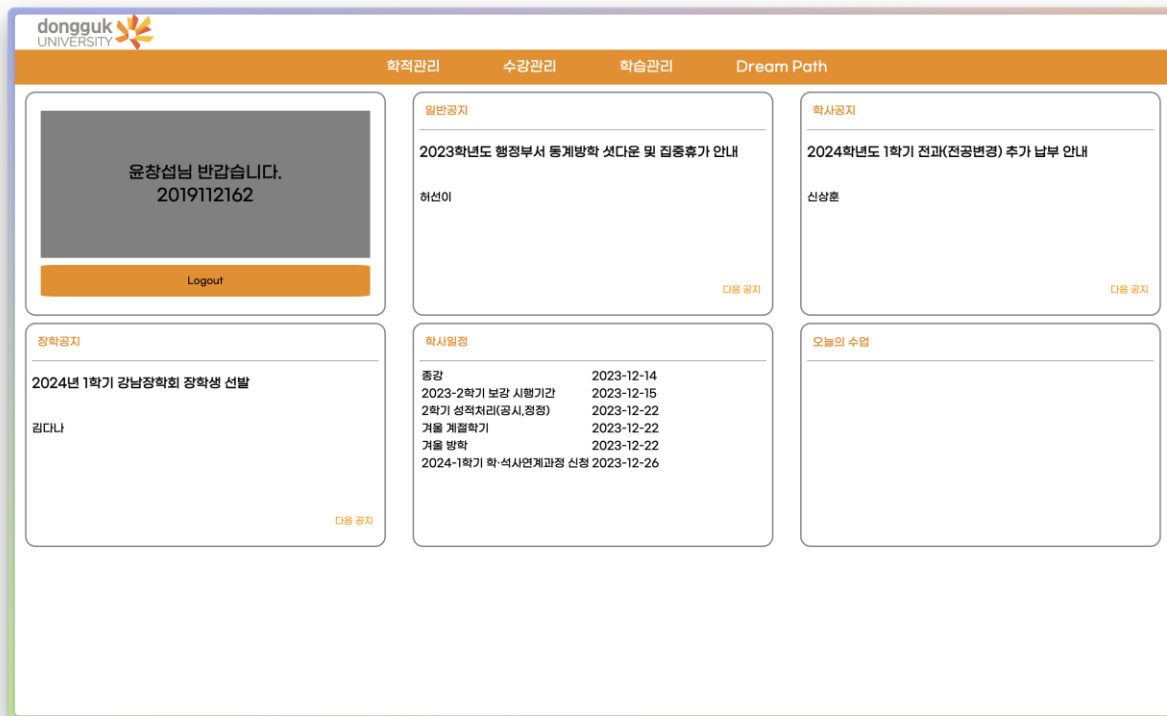


로그인이 정상적으로 처리 되는 경우 반응 모달이 띄워지며 메인 페이지로 이동하게 된다.



올바르지 않은 정보인 경우 다음과 같은 반응 모달이 띄워지게 된다.

[main]



메인 화면에서는 일반공지, 학사공지, 장학공지 등 교내의 공지사항과, 해당 월에 대한 학사일정 정보가 담겨있다. 또한 오늘의 수업 컨테이너에 해당 일에 존재하는 수업과, 그 수업의 강의실, 시간 등의 정보를 제공한다. 각 공지사항은 pagination 으로 이루어져 있으며, 각각의 공지사항을 클릭하는 경우 해당 공지사항의 원문으로 이동한다.

[enrollment]

dongguk UNIVERSITY

학적관리 수강관리 학습관리 Dream Path

수강 신청 페이지

과목 검색 선택 검색어를 입력하세요

검색

미적분학및연습1	문환표	PRI4001-01	수강 신청
미적분학및연습1	박장희	PRI4001-02	수강 신청
일반물리학및실험1	이동환	PRI4002-01	수강 신청
미적분학및연습2	염건애	PRI4012-01	수강 신청
미적분학및연습2	최광호	PRI4012-02	수강 신청
미적분학및연습2	임영빈	PRI4012-03	수강 신청
미적분학및연습2	심호진	PRI4012-04	수강 신청
미적분학및연습2	김동호	PRI4012-05	수강 신청
미적분학및연습2	염건애	PRI4012-06	수강 신청
미적분학및연습2	임영빈	PRI4012-07	수강 신청
미적분학및연습2	박장희	PRI4012-08	수강 신청
미적분학및연습2	박정준	PRI4012-09	수강 신청
미적분학및연습2	박상용	PRI4012-10	수강 신청
미적분학및연습2	임영빈	PRI4012-11	수강 신청
미적분학및연습2	권기운	PRI4012-12	수강 신청
미적분학및연습2	박호	PRI4012-13	수강 신청
미적분학및연습2	박정준	PRI4012-14	수강 신청
미적분학및연습2	임영빈	PRI4012-15	수강 신청
미적분학및연습2	조영덕	PRI4012-16	수강 신청
미적분학및연습2	박호	PRI4012-17	수강 신청
미적분학및연습2	우승익	PRI4012-18	수강 신청

Logout

수강신청

수강신청 결과조회

최초 수강신청 페이지로 이동하는 경우 수강신청 대상 학기의 강의가 무한스크롤로 주어진다. 모든 강의에 대하여 수강신청이 가능하며, 해당 수강신청의 결과는 수강신청 결과조회 탭에서 확인할 수 있다.

dongguk UNIVERSITY

학적관리 수강관리 학습관리 Dream Path

수강 신청 페이지

교과목 ▼ 오픈소스소프트웨어프로젝트 검색

오픈소스소프트웨어프로젝트 김동호 SCS4045-01 수강 신청

Logout

수강신청
수강신청 결과조회

dongguk UNIVERSITY

학적관리 수강관리 학습관리 Dream Path

수강 신청 페이지

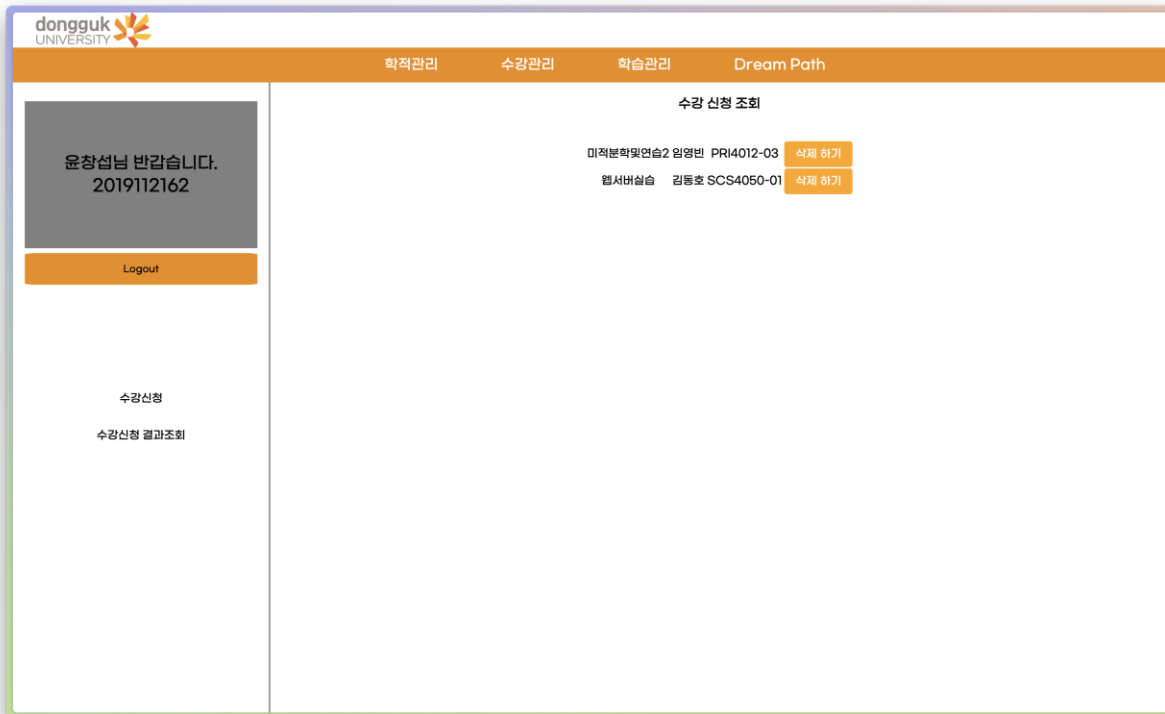
교수명 ▼ 김동호 검색

미적분학및연습2	김동호	PRI4012-05	수강 신청
공학수학1	김동호	PRI4025-03	수강 신청
인사관리	김동호	MGT4053-03	수강 신청
오픈소스소프트웨어프로젝트	김동호	SCS4045-01	수강 신청
웹서버실습	김동호	SCS4050-01	수강 신청

Logout

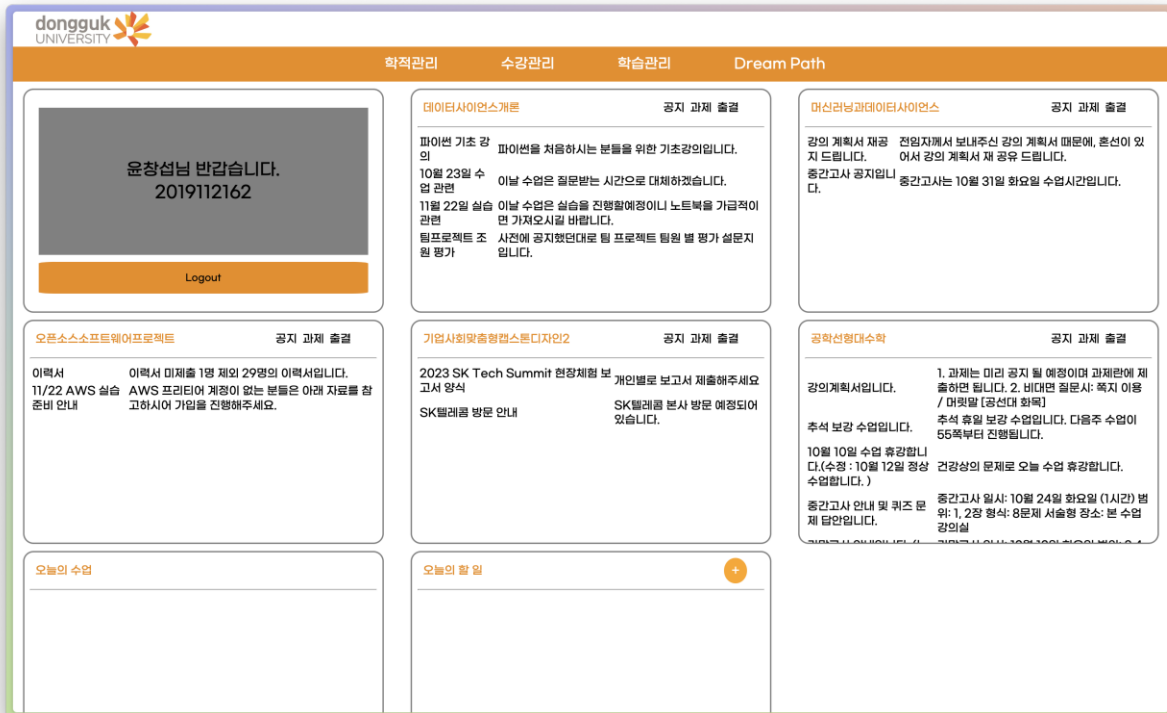
수강신청
수강신청 결과조회

강의에 대해서 세부적인 검색이 가능하다. 검색의 종류는 교과목 검색, 교수명 검색, 학수번호 검색이 존재한다.



수강신청 결과조회 탭에서는 수강신청 탭에서 이루어진 수강신청에 대한 결과를 조회할 수 있다. 또한, 수강 신청된 강의에 대해서 삭제를 진행할 수 있다.

[lms]



각 강의별 공지사항, 과제, 출결 정보를 확인할 수 있다. 또한 오늘의 할 일 컴포넌트를 통해 각 과제에 대해서 수행해야 하는 항목들을 관리하고 작성할 수 있다.

메인 화면과 같이 오늘의 수업 컴포넌트에서는 해당 일에 존재하는 수업에 대해 각 강의명, 강의 교실, 강의 시간을 확인할 수 있다.

기존 e-class 에서 학생들의 편의 기능을 추가하여 구현하였다.

7 장 결론

7.1 배포 결과

<https://dev-changseop.site>

7.2 마무리하며

기존 수강신청 사이트의 문제에 대해 수강신청 페이지가 분할되어있는 점에 대해서 동국 포탈로 통합을 통해 일관성을 부여하였다. 또한 lms 인 E-class 에서 학생들의 편의 기능의 부재한다는 점과, 그로 인해 편의 기능의 경우 타 프로그램을 사용하여 관리한다는 점에 대해 편의기능을 추가하여 본 사이트 내에서 사용할 수 있도록 하였다. 이와 같이 여러 기능별로 불필요하게 분할된 웹 페이지들에 대해서 각기 다른 UI 를 통합하여 통일성 있는 UI 와 더 나은 UX 를 부여할 수 있었다.