

2024 PORTFOLIO



Junseong Bae

지속 가능한 코드를 만드는 개발자



배준성

Bae Junseong

• • •

안녕하세요 지속 가능한 코드를
만드는 개발자 배준성입니다.
적정 기술을 사용하는 것을 지향
하고 믿음직한 팀원이 되는 것을
중요하게 생각합니다.

개인정보

생년월일 1996. 06. 23
전화번호 010-2302-6123
이메일 gentlmanjun@naver.com
학교/전공 한국외국어대학교/전자공학과 졸업
희망 근무지 서울

경력 및 프로젝트

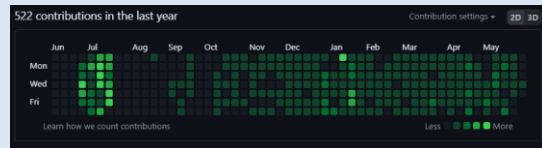
2022 한국외국대학교 (전자, 컴퓨터) 졸업
2022 WE-TNESS 프로젝트 참여
2022 Marsterpiece 프로젝트 참여
2022 SSAFY 프로젝트 우수상 수상
2022 InTagral 프로젝트 참여
2022 SSAFY 7기 수료
2023 42 서울 입과

Programing skill

React	4/5
TypeScript	4/5
HTML/CSS	3/5
Spring	3/5

Github

<https://github.com/wet6123>



오픈소스 프로젝트 기여

2024

08.03 ~

CodePair (코드페어)

오픈소스 Markdown 실시간 동시 편집 도구

#React #Typescript #NestJS #Yorkie #CRDT

<https://codepair.yorkie.dev/>

Add notifications when there is an error in the service #264

2024

08.14 ~

MariaDB

오픈 소스 RDBMS

#C++ #RDBMS

Fix typo at mysql-test README #3457

프로젝트 요약

- 오픈소스 실시간 동시 편집 기술인 Yorkie의 MD 웹 편집기
- CRDT를 기반으로 실시간 동시 편집 구현

나의 역할

- 커스텀 에러 메시지 정의 및 출력에 기여
- CRDT 논문 분석 및 발표

프로젝트 요약

- 오픈 소스 관계형 데이터베이스 관리 시스템

나의 역할

- MaraiDB 오타 수정에 기여

오픈소스 프로젝트 기여

2023

11.03 ~

ko.react.dev

리액트 공식문서 번역

#React #Translate

fix: state 업데이트 큐 페이지 중복된 문장 제거 #885

Translate/experimental_taintUniqueValue #836

Translate: renderToStaticNodeStream #802

2023

10.22 ~

42seoul-translation

42 서울 서브젝트 한글 번역

#C #C++ #pipe #Translate

private

프로젝트 요약

- 리액트 공식 한글화 번역

나의 역할

- State 업데이트 큐 번역 수정
- taintUniqueValue 번역
- renderToStaticNodeStream 번역

프로젝트 요약

- 42 서울의 서브젝트 문서 번역본 모음

나의 역할

- 42 서울의 pipex, cpp_02 서브젝트 문서 번역

경력 및 프로젝트

2024

07.16 ~

Cabi (까비)

42 서울 사물함 관리 시스템

#React #Typescript #Spring Boot #Docker Compose

<https://github.com/innovationacademy-kr/Cabi>

2024

06.23~08.18

Webserv (웹서브)

C++을 기반으로 HTTP 1.1을 만족하는 웹 서버 구현

#C++ #Kqueue #I/O Multiplexing #HTTP 1.1

<https://github.com/wet6123/webserv>

프로젝트 요약

- 1400명의 42 서울의 구성원들이 사물함을 대여할 수 있는 웹 서비스
- 400개 사물함의 이용 현황을 확인하고 관리할 수 있는 관리자 페이지 운영

나의 역할

- 웹 프론트엔드 고도화 및 개선
- 사물함 관리 및 운영

프로젝트 요약

- RFC 7230~4 (HTTP 1.1 명세) 를 만족하는 웹 서버를 구현
- I/O Multiplexing으로 다중 요청에 대한 응답을 동시에 처리 가능

나의 역할

- Kqueue를 이용하여 이벤트 단위로 소켓을 관리하는 I/O Multiplexing 구현
- 파싱과 Req/Res를 관리하는 코드가 HTTP 1.1을 잘 만족하는지 테스트

경력 및 프로젝트

2022

09.19~11.21

InTagraJ (인테그랄)

AI로 사진을 분석해 태그 달기를 도와주는 SNS 앱
삼성 SDS 연계 프로젝트

#Spring Boot #Kotlin #Image처리 #AWS #MySQL #YOLOv5

<https://github.com/wet6123/Intagral>

2022

08.22~09.16

Masterpiece (마스터피스)

외국인이 한글의 과학성과 예술성을 경험할 수 있는 NFT 플랫폼
SSAFY 우수 프로젝트 선정

#React #Typescript #Spring Boot #MySQL #Web3.js #Meta Mask #AWS

<https://github.com/wet6123/rnasterpiece>

프로젝트 요약

- 안드로이드 기반 SNS 앱
- 사진을 업로드하면 태그를 자동으로 생성

나의 역할

- ERD 작성, DB 구성
- 이미지 업로드를 위한 AWS S3 구성
- Post 업로드, 삭제, 조회를 위한 API 구축

프로젝트 요약

- 한글 NFT를 디자인하고 생성하여 거래할 수 있음.
- 한글 NFT를 제작하며 한글의 원리와 디자인을 경험

나의 역할

- 로그인, 마이페이지 구현
- NFT 목록, 상세, 거래 페이지 구현
- 한글 게임 페이지 구현
- 한글 NFT 에디터 구현 참여

경력 및 프로젝트

2022

07.05~08.19

WE-TNESS (위트니스)

WebRTC와 모션인식을 활용한 운동 게임 서비스

#React #Redux-toolkit #Spring Boot #AWS #Web RTC #Teachable Machine

<https://github.com/wet6123/WE-tness>

2022

05.16~05.26

모두의 PT

회원 기반 유튜브 운동 영상 웹 서비스

#Vue.js #Spring Boot #MySQL

<https://github.com/wet6123/Everyones-PT>

프로젝트 요약

- 웹캠을 활용한 웹 기반 운동 게임
모션 인식을 통해 운동 횟수 카운트
- 다른 유저와 화상을 공유하여
비대면으로 운동할 수 있는 서비스

나의 역할

- 메인 페이지, 튜토리얼 페이지 구현
- 게임방, 유저 검색, 유저 페이지 구현
- 알림 기능 구현

프로젝트 요약

- 운동 관련 영상을 시청 가능
- 영상을 운동 부위별 분류
- 플레이 리스트, 시청 기록 확인 가능

나의 역할

- Vue의 Intersection Observer를
활용한 무한 스크롤 구현
- 창 크기 변경에 따른 반응형 설계
- 영상 검색 및 정렬 기능 구현

#01

2024

CodePair

오픈소스 Markdown 실시간 동시 편집 도구

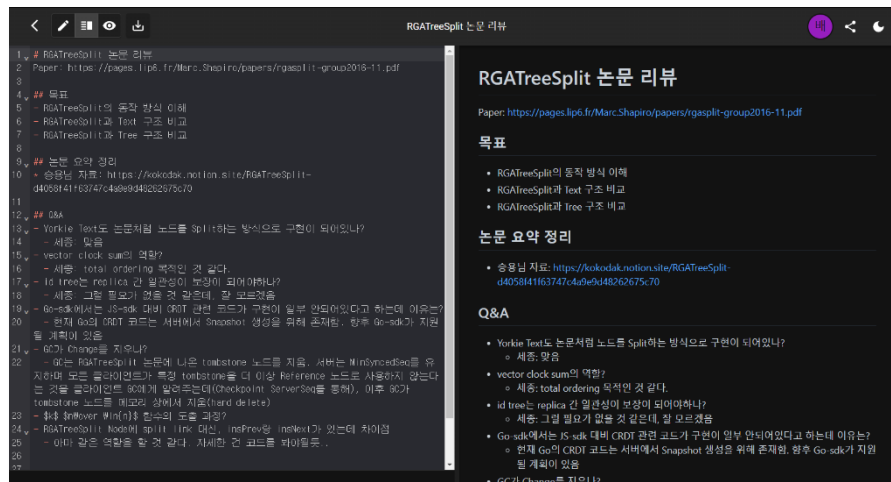


코드페어 - 오픈소스 Markdown 실시간 동시 편집 도구

코드페어는 AI 인텔리전스를 갖춘 오픈소스 실시간 협업 마크다운 편집기입니다.
동시 편집을 위해서 CRDT 기반의 Yorkie를 활용했습니다.

서비스 링크 : <https://codepair.yorkie.dev/>

깃허브 링크 : <https://github.com/yorkie-team/codepair>



Project

- 실시간으로 여러 사람이 동시에 편집할 수 있는 MD 에디터

참여 기간

- 참여 기간 : 2024.08.03 ~

적용 기술

- React
- NestJS
- LangChain
- CRDT

기여 부분

- 커스텀 에러 메시지 정의 및 출력에 기여
- CRDT 논문 분석 및 발표

커스텀 에러 메시지 구현

1

기존에 구현된 에러에 대한 예외에는 에러에 대한 자세한 이유가 없기 때문에 사용자가 에러가 발생한 이유를 파악하기 어려웠습니다.

이 문제를 해결하기 위해 각 에러에 대한 예외와 JWT AuthGuard에 에러가 발생한 자세한 이유를 추가했습니다.

이렇게 추가된 에러 메시지는 프론트엔드의 스낵바 메뉴를 사용하여 출력했습니다.

에러 예외처리 분석

1. 현재 시스템 분석

현재 서비스에서는 NestJS의 기본 에러 객체를 활용했기 때문에 에러 메시지를 출력해도 기본적인 에러 메시지만 표시되었습니다.

이는 불친절한 기술적 메시지였으며 사용자는 에러 상황별 구체적인 대응이 불가능했습니다. 따라서 각 상황에서 에러의 이유를 분석하고 적절한 에러 메시지 작성과 분기 처리를 했습니다.

```
if (conflict) {
  throw new ConflictException();
}

if (conflict) {
  throw new ConflictException("Workspace title is already in use.");
}
```

2

커스텀 예외 출력

1. 토스트 메뉴 수정

이렇게 추가된 에러 메시지를 유저에게 전달하기 위해서 기존에 사용하고 있던 토스트 메뉴를 활용했습니다. 이후 작업에서 또 다른 기본 에러 객체가 추가될 것에 대비하여 커스텀 에러 메시지와 기본 에러 메시지를 모두 처리 가능 하도록 구현했습니다.

커스텀 에러 메시지 구현

3

기존에 구현된 에러에 대한 예외에는 에러에 대한 자세한 이유가 없기 때문에 사용자가 에러가 발생한 이유를 파악하기 어려웠습니다.

이 문제를 해결하기 위해 각 에러에 대한 예외와 JWT AuthGuard에 에러가 발생한 자세한 이유를 추가했습니다.

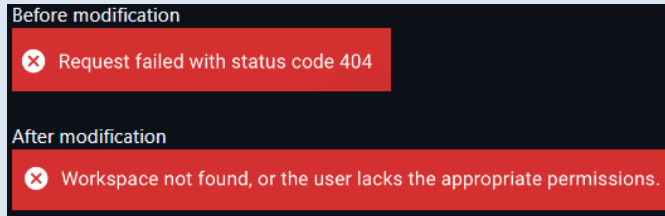
이렇게 추가된 에러 메시지는 프론트엔드의 스낵바 메뉴를 사용하여 출력했습니다.

결과 및 고민했던 부분

1. 에러 메시지 개선

불친절한 에러 메시지를 개선하여 에러가 발생한 이유를 쉽게 파악할 수 있게 되었습니다.

이를 통해서 사용자 경험 개선과 더불어 디버깅 시간을 단축할 수 있었습니다.



하지만 위 예시와 같이 분기 처리가 완벽하게 처리되지 않은 부분에 대해 더 세분화된 에러 분류가 필요합니다.

2. NestJS HTTP 에러 객체

이슈를 해결하며 한 번도 사용해 보지 않은 NestJs 서버를 분석하고 이슈를 해결하여 3일 만에 머지에 성공했습니다. NestJS HTTP 에러 객체 또한 처음 다뤄봤기 때문에 LLM과 코드 리뷰의 내용을 적극적으로 활용했습니다.

#02

2024

Cabi

42 서울의 캐비닛 대여 서비스 & 발표 예약 서비스

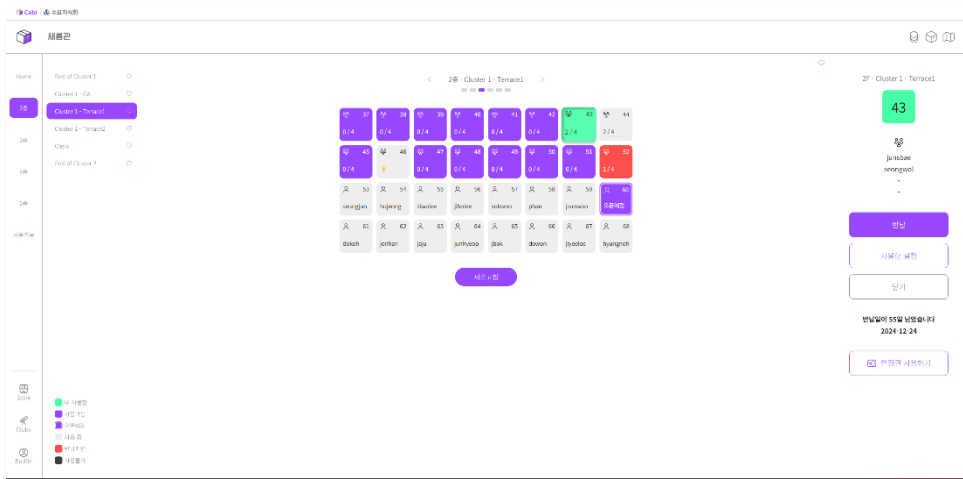


까비 - 42서울의 캐비닛 대여 서비스

42서울의 캐비닛 400여 개를 편리하게 대여 및 반납할 수 있는 서비스입니다.
1400명의 교육생들이 원활하게 캐비닛을 사용할 수 있도록 돕는 것을 목표로 합니다.
관리자 플랫폼을 통해서 캐비닛 대여 현황 및 상태를 도식화해서 확인할 수 있으며 관리할 수 있습니다.

데모 링크 : <https://api-dev.cabi.42seoul.io/demo>

깃허브 링크 : <https://github.com/innovationacademy-kr/Cabi>



Project

- 캐비닛을 대여하는 42서울 교육생을 위한 서비스
- 캐비닛을 관리하는 42서울 관리자를 위한 서비스

참여 기간

- 참여 기간 : 2024.07.16 ~ 운영중

적용 기술

- React
- Spring Boot
- Docker
- Recoil
- MariaDB

기여 부분

- 프론트엔드 개발
- 캐비닛 시스템 버그 수정
- 발표 신청 서비스 고도화
- 사물함 관리

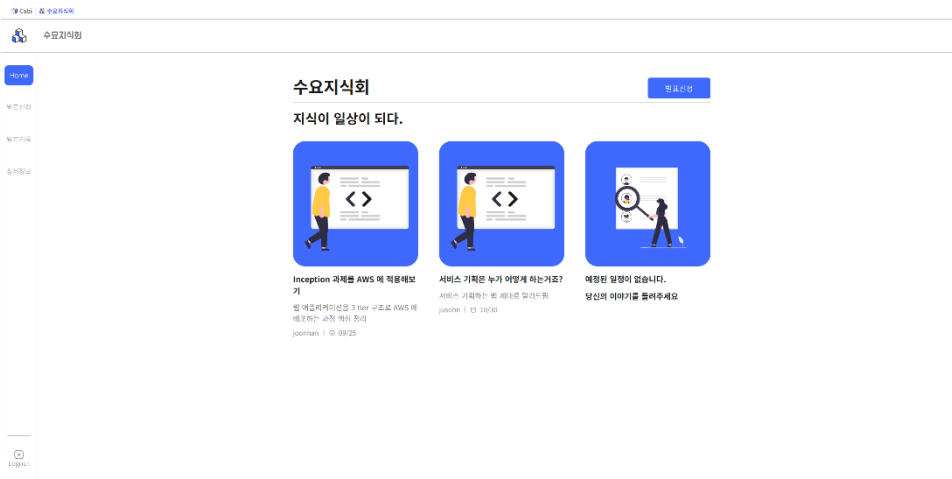


수요지식회 - 42서울의 발표 예약 서비스

42서울에서 1, 3 주 수요일마다 진행되는 발표 행사인 [수요지식회] 예약 서비스입니다.
Cabi 웹 사이트에서 수요지식회 탭을 통해서 발표 신청을 접수하고 있습니다.

서비스 링크 : <https://cabi.42seoul.io/>

깃허브 링크 : <https://github.com/innovationacademy-kr/Cabi>



Project

- 발표를 희망하는 42서울 교육생을 위한 서비스
- 발표를 관리하는 42서울 관리자를 위한 서비스

참여 기간

- 참여 기간 : 2024.07.16 ~ 운영중

적용 기술

- React
- Recoil
- Spring Boot
- MariaDB
- Docker

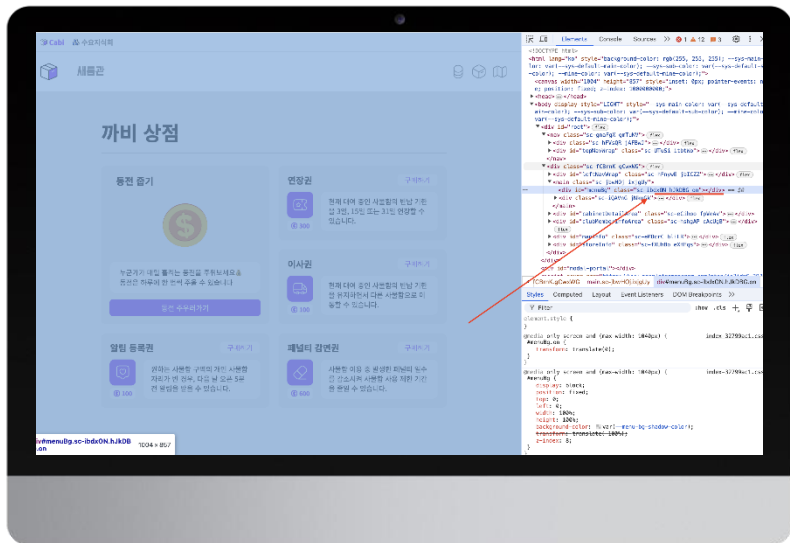
기여 부분

- 프론트엔드 개발
- 캐비닛 시스템 버그 수정
- 발표 신청 서비스 고도화
- 사물함 관리

캐비닛 시스템 인터페이스 버그 수정

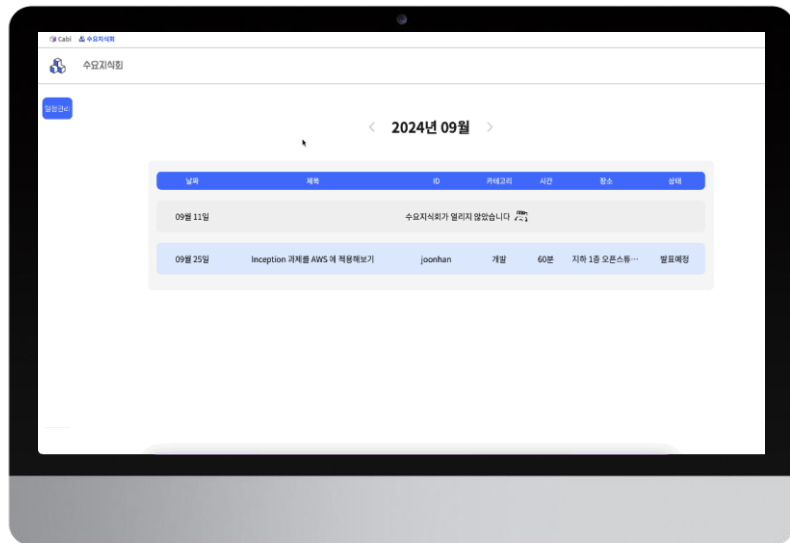
모달 종료 시 배경이 사라지지 않는 버그를 수정했습니다.

이 과정에서 모달 혹은 수정하여 모달을 통해서 새로운 페이지로 이동하는 경우 모달과 관련된 모든 창을 닫도록 수정하여 추후에 같은 문제가 발생하지 않도록 방지했습니다.



메뉴 선택 시 원치 않는 곳으로 라우팅 되는 문제를 해결했습니다.

Navigate에 지정된 상대 경로가 의도치 않은 페이지를 가리키고 있어서 발생하는 문제였습니다. 대부분의 경로가 절대 경로로 작성된 것을 확인하고 통일성을 위해서 절대 경로로 수정했습니다.



발표 신청 서비스 고도화

프론트엔드에 존재했던 발표 날짜
계산 로직을 제거하였습니다.
또한 관리자의 피드백을 수용하여
발표 신청 서비스를 개선했습니다.

프론트엔드의 비즈니스 로직을
백엔드로 이전하고, 발표 일정 출력,
발표 정보 조회 등 기존의 기능을 동
일하게 구현했습니다.

사용자 피드백을 분석하여
원하는 날짜로 일정을 수정하는 기능
을 추가했습니다.

1

기존 시스템 분석

1. 현재 시스템 분석

프론트엔드에서 날짜 계산 비즈니스 로직을 발견하였습니다.
관심사 분리를 위해 비즈니스 로직을 백엔드로 이전하기 위해서
로직이 사용되는 컴포넌트들을 분석하고 도식화 하여 정리하고
발표 서비스에서 사용되는 API 를 분석하고 수정할 내용을 정리했습니다.

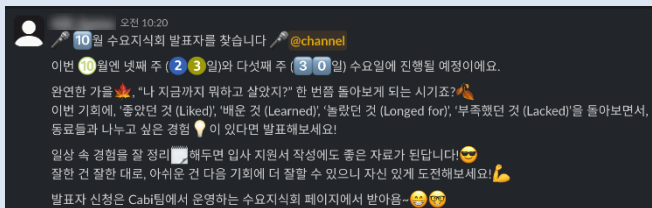
링크 : <https://lrl.kr/pZzn>

2

사용자 피드백 분석

1. 날짜 수정 요구 발생

휴일/행사 등으로 인해서 기존에 정해진 1, 3 주 수요일이 아닌 다른 날짜로 일정 변경이
필요한 경우가 발생했습니다. 미리 정해진 수요지식회 날짜 이외의 다른 날짜로 바꿀 수
없었기 때문에 기능 추가/수정이 필요했습니다.



발표 신청 서비스 고도화

프론트엔드에 존재했던 발표 날짜 계산 로직을 제거하였습니다.
또한 관리자의 피드백을 수용하여 발표 신청 서비스를 개선했습니다.

프론트엔드의 비즈니스 로직을 백엔드로 이전하고, 발표 일정 출력, 발표 정보 조회 등 기존의 기능을 동일하게 구현했습니다.

사용자 피드백을 분석하여 원하는 날짜로 일정을 수정하는 기능을 추가했습니다.

3

목표 설정 및 구현

1. 변경 예정 사항 및 예상 문제점 정리

시스템 분석을 통해서 찾아낸 문제점의 해결 방법을 구상했습니다.
구상한 해결 방법을 적용하는 과정에서 발생 할 것이라고 예상되는 문제점을 정리하였습니다.
개발 기한이 얼마 남지 않았기 때문에 최소한의 수정으로 빠르게 필요한 기능을 구현하고자 했습니다.

2. 비즈니스 로직 이전 및 페이지 기능 수정

프론트엔드에서 발표 날짜를 계산하는 `calculateAvailableDaysInWeeks`를 백엔드로 이전하고
수요지식회가 열리는 날짜를 리턴하는 `able-date` 엔드 포인트를 추가했습니다.
기존 API를 수정하여 기존 기능과 추가 기능을 구현했습니다.

RegisterPage

- 선택 가능한 발표 일자를 반환
- `able-date`에서 이미 발표가 있는 날짜를 `invalid-date`로 호출해 제외한다.

EditStatusModal

- 이미 발표가 있는 날짜는 선택할 수 없도록 처리하고 해당 월의 전체 날짜를 표기
- 피드백을 통해서 해당 월의 날짜를 전부 선택할 수 있도록 변경한다.

DetailContent

- 발표 일정과 아직 발표 가능한 날짜를 전부 표기
- 수요지식회 날짜를 계산해서 표기하는 부분을 제거하고 `schedule` 엔드 포인트로 모든 일정을 불러온다.

발표 신청 서비스 고도화

프론트엔드에 존재했던 발표 날짜 계산 로직을 제거하였습니다.
또한 관리자의 피드백을 수용하여 발표 신청 서비스를 개선했습니다.

프론트엔드의 비즈니스 로직을 백엔드로 이전하고, 발표 일정 출력, 발표 정보 조회 등 기존의 기능을 동일하게 구현했습니다.

사용자 피드백을 분석하여 원하는 날짜로 일정을 수정하는 기능을 추가했습니다.

4

결과 및 고민했던 부분

1. 프론트엔드에서 발표 날짜 계산 로직 제거

기존에는 정해진 날짜만 출력할 수 있었지만 이번 작업을 통해서 날짜 계산 로직을 프론트엔드에서 제거하고 이를 통해서 발표 날짜를 자유롭게 변경할 수 있게 되어 추후에 어떤 일정이 들어와도 이를 처리할 수 있게 되었습니다.

2. 일정의 타입 문제

발표 일정의 category와 status를 enum으로 관리하고 이러한 enum의 값을 key로 사용하는 map을 만들어 화면 인터페이스에 활용하도록 되어있어 확장이 어려웠습니다. 현재는 발표 유저의 유무로 발표 가능 여부를 구분하고 있지만 타입 시스템을 우회하고 있다고 생각합니다.

따라서 추가 작업을 진행하여 이를 구분할 수 있는 필드를 추가하거나 메타데이터 객체를 활용해 개선할 필요가 있습니다.

3. 발표 일정 변경 화면 개선

발표 일정의 경우 드롭다운으로 변경 날짜를 선택하고 있습니다.

이를 달력 형태로 변경하면 사용자 경험을 개선할 수 있을 뿐만 아니라

발표 일정, 발표 가능 날짜 추가 & 삭제 기능을 도입하여 앞의 타입 문제를 함께 해결할 수 있겠다고 생각합니다.

#03

2022

Masterpiece

외국인을 위한 한글 NFT 서비스



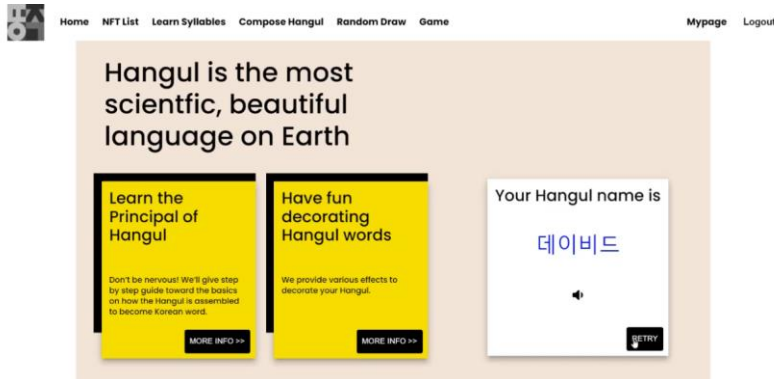
Masterpiece

한글 NFT 서비스

마스터피스 - 외국인을 위한 한글 NFT 서비스

외국인이 한글의 원리와 디자인을 경험하고 한글 NFT를 만들 수 있는 서비스입니다.
마우스로 한글 문구를 조합할 수 있는 GUI와 한글 NFT를 디자인할 수 있는 에디터를 제공합니다.
한글 NFT 거래할 수 있는 NFT 거래소 기능을 제공합니다.

깃허브 링크 : <https://github.com/wet6123/rnasterpiece>



Project

- 한글에 관심있는 외국인을 위한 서비스
- NFT생성과 거래에 관심있는 사람을 위한 서비스

개발 인원 및 기간

- 개발 기간 : 2022.08.29 ~ 2022.10.07 (6주)
- 개발 인원 : 5명 (프론트엔드 3, 백엔드 2)

적용 기술

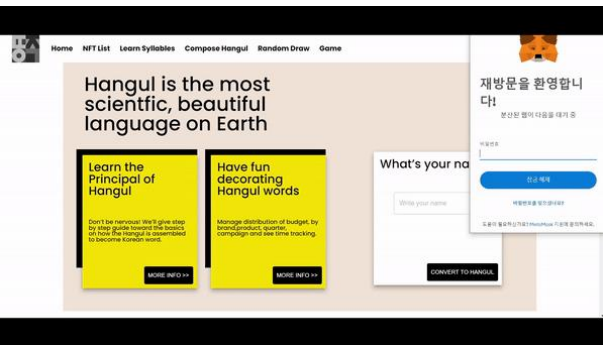
- React
- Spring Boot
- Redux
- Web3.js
- MetaMask

기여도

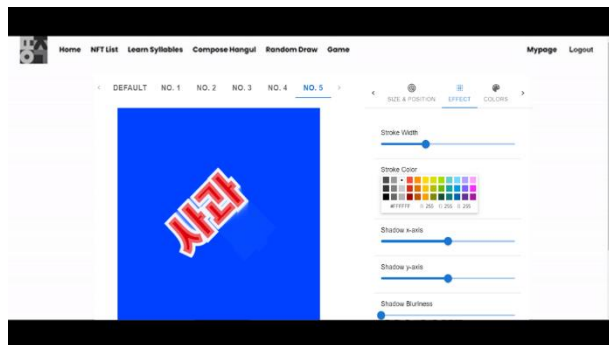
- | | |
|------|-----|
| • 기획 | 40% |
| • 개발 | 40% |

기여 부분

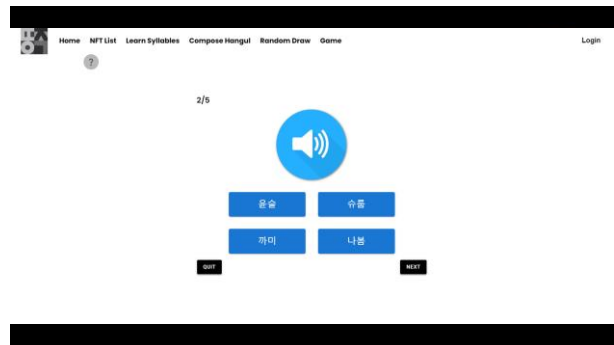
- 팀장, 프론트엔드 개발
- 로그인, 마이페이지 구현
- NFT 목록 보기, 상세 보기, 거래 페이지 구현
- 한글 게임, 한글 자모음 뽑기 구현
- 한글 디자인 에디터 구현 참여
- Notion 협업 공간 관리



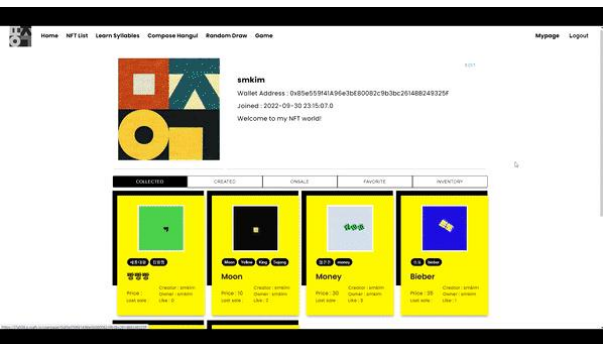
로그인



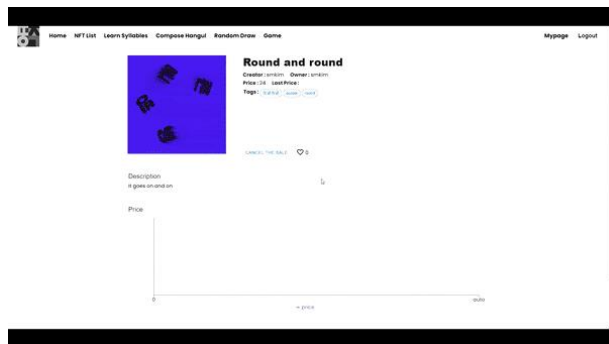
한글 디자인 에디터



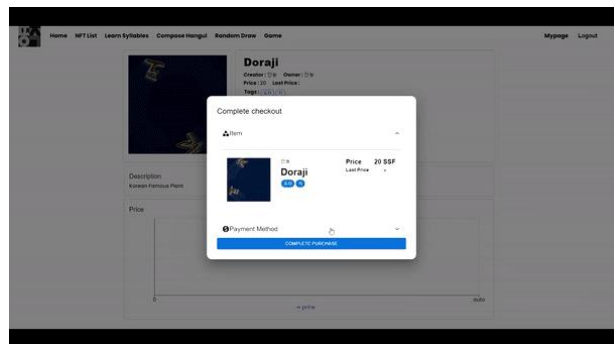
한글 게임



NFT 목록 보기



NFT 상세 보기



NFT 거래

NFT 에디터 구현

1

한글의 원리와 아름다움을 자연스럽게 전달하기 위해서 한글 문장을 아름답게 꾸밀 수 있는 기능을 구현하고자 했습니다.

CSS 만으로 동작하는 디자인 에디터를 구현하는 것이 아니라 한글 문장에 애니메이션을 부여하는 기능을 제공하였습니다.

애니메이션 기능 구상

1. 다양한 타이포그래피 효과 조사

영상에서 사용되는 애프터 이펙트로 구현된 타이포그래피 효과나 정적으로 디자인된

템플릿 등을 조사하며 [Google Font + 한국어](#) 페이지를 알게 되었습니다.

클릭에 따라 변경되는 다양한 타이포그래피 효과를 보면서 에디터의 애니메이션 기능을 구상하게 되었습니다.

2

글자 정보 분석

1. 폰트 정보 분석

브라우저에서 글자를 어떤 방법으로 화면에 그리는지 알아보면서 폰트 파일을 파싱하고 분석해볼 수 있는 OpenType.js 라이브러리를 찾았습니다.

라이브러리를 이용하여 글리프(glyph, 실제 그려질 문자 모양)의 벡터 좌표를 추출하였습니다. 이정보를 SVG로 그려보면 화면에 글자 모양과 똑같은 형태의 벡터 이미지가 출력되는 것을 확인할 수 있었습니다.

NFT 에디터 구현

3

한글의 원리와 아름다움을 자연스럽게 전달하기 위해서 한글 문장을 아름답게 꾸밀 수 있는 기능을 구현하고자 했습니다.

CSS 만으로 동작하는 디자인 에디터를 구현하는 것이 아니라 한글 문장에 애니메이션을 부여하는 기능을 제공하였습니다.

이미지에 효과 부여

1. 이미지 왜곡과 애니메이션

이렇게 만들어진 이미지의 경로를 왜곡 시키면 글자 이미지의 모습을 바꾸는게 가능했습니다. 이후 이러한 경로 왜곡을 지속적으로 변경하면서 글자에 애니메이션을 부여할 수 있었습니다.

4

화면 렌더링 최적화

1. RequestAnimationFrame 사용

RequestAnimationFrame을 사용하여 브라우저 리페인트 주기와 동기화 시켰습니다. 리페인트 주기 동기화를 통해서 프레임 시작 시 JS 실행을 보장하여 밀림 현상을 해결할 수 있었습니다.

이를 통해 보다 부드러운 애니메이션을 제공하고 불필요한 렌더링을 방지할 수 있었습니다.

프로젝트 관리

마스터피스 프로젝트에 팀장으로
지원하여 활동했습니다.
팀장으로서 기능 개발과 동시에
프로젝트를 관리했습니다.
프로젝트 기획, 일정 관리, 산출물 관리,
현황 보고 등 다양한 경험을
해볼 수 있었습니다.

1

프로젝트 일정 관리

1. Jira를 이용한 일정 관리

Jira를 이용하여 태스크를 관리했습니다. 이를 통해서 팀원들과 회의 없이도
프로젝트의 진행 현황을 파악할 수 있었습니다. 매주 프로젝트 현황 보고에서도 이러한 일정
관리 덕분에 팀원들은 방해 없이 개발에 집중하면서도 보고를 잘 마무리 할 수 있었습니다.

2. 일일 스크럼

매일 프로젝트 시작 전 3분 동안 일일 스크럼을 진행하여
팀원의 요구 사항과 변경점을 빠르게 공유할 수 있도록 했습니다.

2

개발 문서 관리

1. API 문서 관리

API 문서를 Notion으로 관리하고 업데이트 규칙을 수립했습니다.
개발 현황과 수정 여부를 표기하는 칼럼을 추가하여 한눈에 개발 현황과 수정 사항을
확인할 수 있도록 했습니다.

구현할 기능에 우선 순위를 부여하여 순위가 높은 태스크 위주로 개발을 진행했습니다.
이를 통해 기간 내에 프로젝트 필수 기능을 전부 구현할 수 있었습니다.

#04

2024

Webserve

HTTP/1.1 호환 고성능 웹서버

Webserv

웹서브 - HTTP/1.1 호환 고성능 웹서버

HTTP 1.1을 만족하며 초당 약 2100개의 트랜잭션을 처리할 수 있는 웹 서버입니다.
I/O Multiplexing을 위해서 Kqueue를 이용하여 이벤트 단위로 소켓을 관리하는 로직을 구현했습니다.
Siege 부하 테스트에서 100개의 클라이언트가 연속적으로 보내는 요청을 100%의 응답률로 커버합니다.

깃허브 링크 : <https://github.com/wet6123/webserv>

```
[WARNING] common/config/Server.cpp:85 - PATH : /
[WARNING] common/config/Server.cpp:94 - PATH : /
[WARNING] common/config/Server.cpp:99 - PATH : /
[INFO] common/config/Server.cpp:51 - FILE EXTENSION : png$
[INFO] common/config/Server.cpp:51 - FILE EXTENSION : py$
[WARNING] common/config/Server.cpp:85 - PATH : /
[WARNING] common/config/Server.cpp:94 - PATH : /
[WARNING] common/config/Server.cpp:99 - PATH : /
[INFO] common/config/Server.cpp:51 - FILE EXTENSION : png$
[INFO] common/config/Server.cpp:51 - FILE EXTENSION : py$
[INFO] client/src/ClientManager.cpp:63 - Send response
[INFO] server/src/WebServer.cpp:156 - Close
[INFO] client/src/ClientManager.cpp:63 - Send response
[INFO] server/src/WebServer.cpp:156 - Close

Lifting the server siege...
Transactions:      19551 hits
Availability:      100.00 %
Elapsed time:      8.92 secs
Data transferred:  9.49 MB
Response time:     0.05 secs
Transaction rate:  2191.82 trans/sec
Throughput:        1.06 MB/sec
Concurrency:       98.94
Successful transactions: 19551
Failed transactions: 0
Longest transaction: 0.23
Shortest transaction: 0.01
```

Project

- Kqueue를 이용한 HTTP/1.1 호환 고성능 웹서버

개발 인원 및 기간

- 개발 기간 : 2024.06.30 ~ 2022.08.18 (7주)
- 개발 인원 : 4명

적용 기술

- C++
- Kqueue
- I/O Multiplexing

기여도

- 개발  30%

기여 부분

- Kqueue를 이용한 I/O Multiplexing 구현
- CGI 구현
- HTTP 1.1 호환성 테스트

Kqueue를 이용한 I/O Multiplexing

입출력이 동시에 일어나는 웹서버의 특성상 많은 소켓을 관리하는 효율적인 로직을 구현해야 했습니다.

이벤트 기반의 소켓 관리를 위해서 Kqueue를 이용하였습니다.

1

서버 메인 로직 구현

1. Kqueue를 이용한 소켓 관리

입출력이 동시에 일어나는 웹서버의 특성상 많은 소켓을 관리하는 효율적인 로직을 구현해야 했습니다. 이를 위해서 I/O Multiplexing을 구현했으며 이벤트 기반의 소켓 관리를 위해서 Kqueue를 이용하였습니다. 소켓의 상태 변화를 이벤트로 관리하여 이에 따른 동작을 구현했습니다

2. 커넥션 풀링

HTML1.1에서 지원하는 Keep-alive를 구현하여 TCP handshake에 의한 오버헤드를 줄일 수 있습니다.

2

구현된 웹서버 테스트

1. Siege 경량 테스트

Siege 경량 테스트를 이용하여 구현된 웹서버의 성능을 측정했습니다.

순수 HTML: 2100TPS, 이미지 포함: 300TPS 로 Nginx의 기본 성능에 근접하는 수준의 성능을 확인할 수 있었습니다.

#05

2022

InTagraJ

홈 인테리어 SNS 플랫폼

InTagraJ

인테그랄 - 홈 인테리어 SNS 플랫폼

자체 SNS 플랫폼을 활용하여 사진을 업로드할 수 있습니다.
AI 이미지 프로세싱을 활용해 업로드한 이미지와 관련된 해시태그를 자동으로 생성해줍니다.
유저는 자신이 선호하는 태그 프리셋을 미리 등록할 수 있고 이것을 기반으로 태그를 생성합니다.

깃허브 링크 : <https://github.com/wet6123/Intagral>



Project

- 인테리어 제품 홍보 기업/판매자를 위한 서비스
- 인테리어에 관심 있는 일반 사용자를 위한 서비스

개발 인원 및 기간

- 개발 기간 : 2022.09.19 ~ 2022.11.21 (6주)
- 개발 인원 : 6명 (백엔드 2, 안드로이드 2, AI 2)

적용 기술

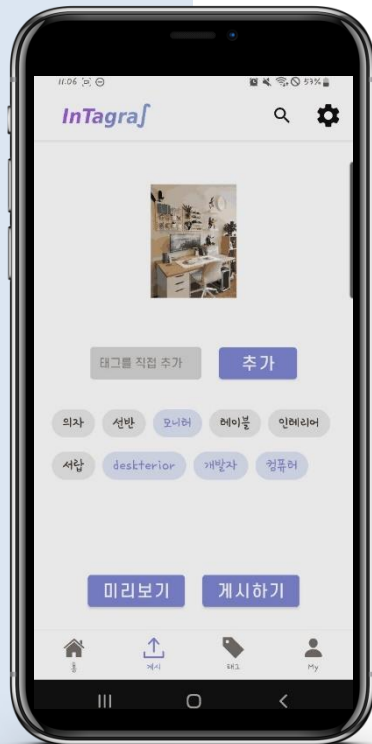
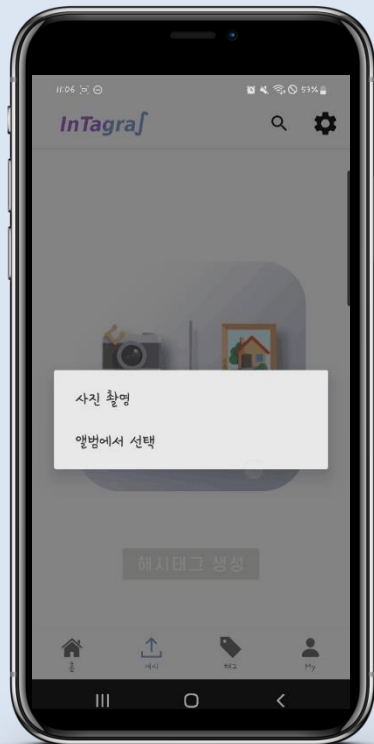
- Spring Boot
- Kotlin
- MySQL
- YOLOv5
- JPA

기여도

- | | | |
|------|-------------|-----|
| • 기획 | <div></div> | 30% |
| • 개발 | <div></div> | 40% |

기여 부분

- ERD 작성, DB 구성
- 이미지 업로드를 위한 AWS S3 Bucket 구성
- 게시물 업로드, 삭제, 조회, 좋아요를 위한 API 구현
- 앱 UI / UX 개선
- 와이어프레임 제작



게시물 기능

유저는 직접 카메라를 켜 사진을 촬영하거나, 앨범에서 사진을 업로드 할 수 있습니다.

사진을 선택한 뒤에 사용할 해시태그를 선택하여 업로드 할 수 있습니다.

업로드 된 게시물은 앱의 상단 검색 기능을 통해서 최신순이나 해시태그, 팔로우한 유저, 작성자를 기반으로 찾아볼 수 있습니다.

게시물의 이미지와 해시태그 확인 뿐만 아니라 좋아요 및 삭제할 수 있습니다.

게시물 기능

SNS 서버에 게시글을 업로드, 조회, 삭제, 좋아요 하기 위한 API를 구현했습니다.

게시물 업로드를 구현하는 과정에서 게시글의 이미지를 저장하기 위한 AWS S3 버킷을 구축했습니다.

1

게시글 업로드, 조회, 삭제, 좋아요 API

1. 게시글 조회 페이지네이션 처리

게시글 목록의 출력 속도를 높이기 위해서 앱 내부에서 무한 스크롤로 처리했습니다.

이를 위해 페이지네이션 처리를 하여 게시글 목록과 페이지 번호, isNext를 리턴 했습니다.

2. 유저 팔로우, 게시글 해시태그, 좋아요 처리

팔로우, 해시태그, 좋아요 등 복잡한 관계를 관리하기 위해서 DB 설계 시 정규화를 통해서 데이터 무결성을 확보 했습니다. 이를 통해서 소셜 기능을 쉽게 확장할 수 있었습니다.

2

이미지 저장을 위한 AWS S3 버킷 구축

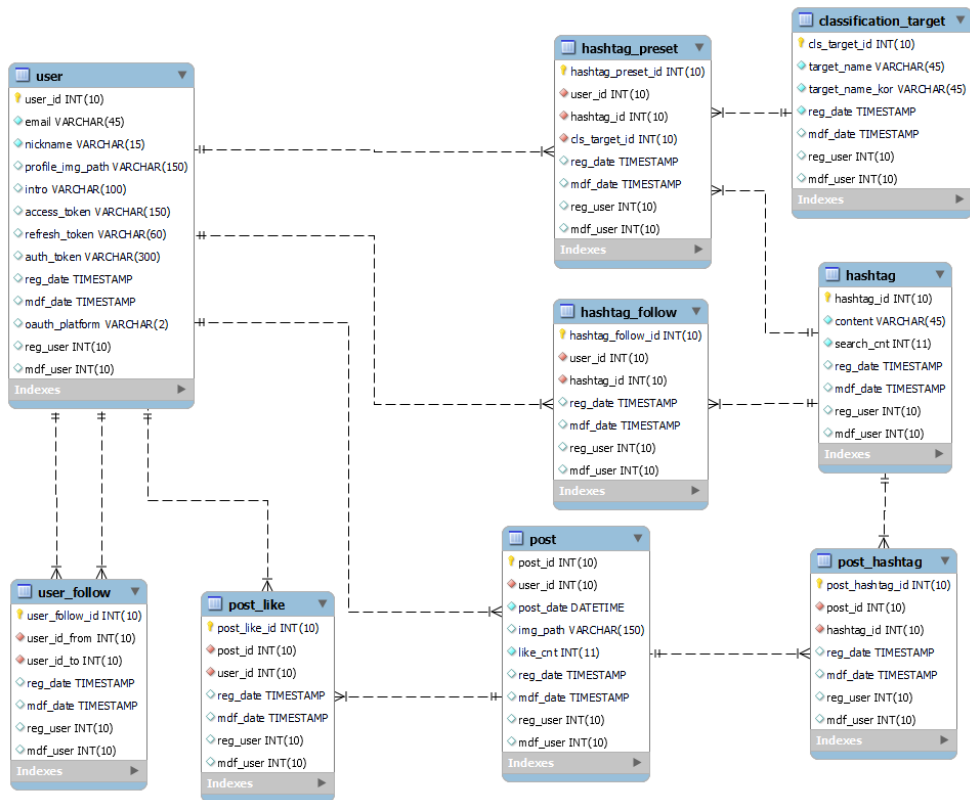
1. 대용량 파일 처리

업로드 하는 이미지의 크기가 너무 큰 경우 이미지를 제대로 업로드 할 수 없었습니다.

이를 해결하기 위해서 MultipartFile을 이용하여 처리했습니다.

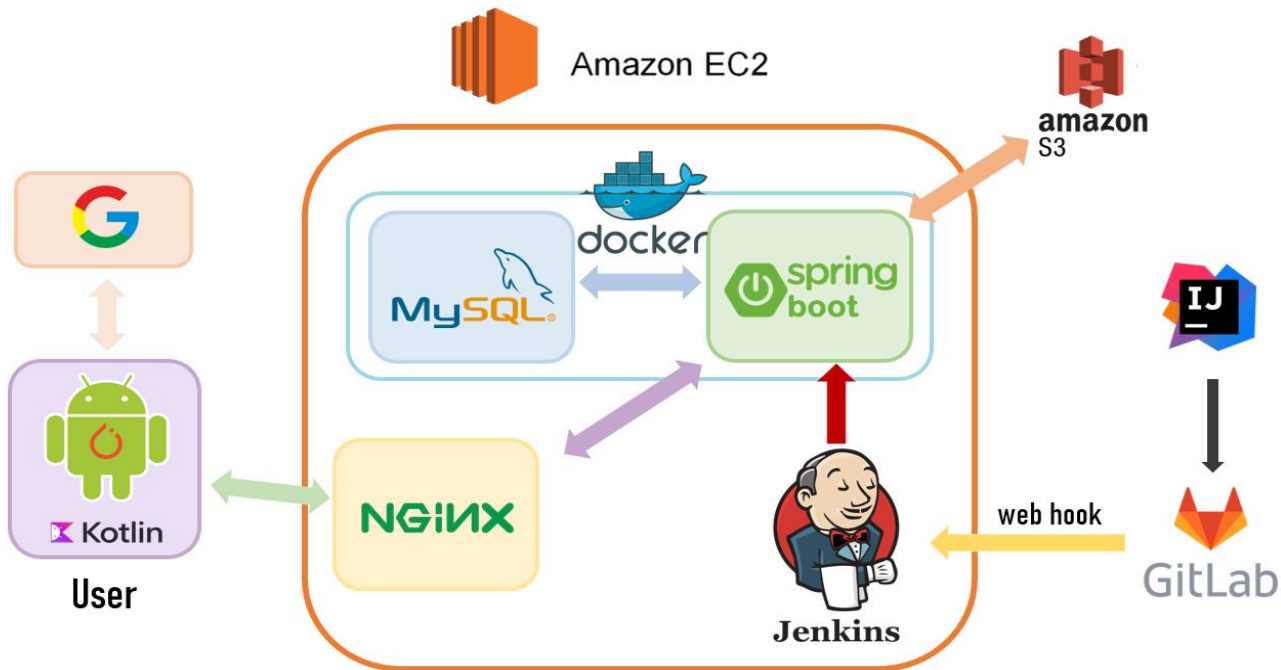
2. 파일 저장 경로 문제

이미지 파일을 EC2 웹서버에 저장하는 것은 비용과 관리 측면에서 효율적이지 못했습니다. 따라서 S3 버킷을 구축하여 이를 해결하고 보안과 인프라 관리 측면에서도 이점을 가질 수 있었습니다.



시스템 구성도

#5 인테그랄



감사합니다