



2022

PORTFOLIO

정신력이 강한
협업하는 개발자, 장현진



장현진

생년월일

1996.11.27

학교 / 전공

창원대학교
컴퓨터공학 전공

정신력이 강한 협업하는 개발자

프로그래밍 역량

Skills ...

업무에도 활용해요. 6

✓ Java

✓ Spring

✓ MySQL

✓ JPA

✓ Redis

✓ Git

+ 새로 만들기

어느 정도 사용할 줄 알아요. 5

⚙ Docker

✓ C

✓ Jenkins

✓ Kafka

✓ AWS

+ 새로 만들기

보고 이해할 수는 있어요. 3

✓ Python

Vue

JSP

+ 새로 만들기

Notion 포트폴리오 주소 : <https://www.notion.so/88ca40bf4a154c0193ed7037c4180760>

경력 및 프로젝트

1. 주유소 (SSAFY 공통프로젝트 우수상)

담당 : 백엔드 개발 / 참여인원 6명

2022.01 - 2022.02

#JAVA #Spring #JPA #RDBMS #WebSoket #Docker

화상기능 서비스

자유롭게 방을 생성하고 친구 또는 새로운 사람들과 만남을 가질수 있다

게임 및재미 서비스

단순한 화상회의 서비스가 아닌 여러가지 게임을 통해 술자리에 맞는 분위기를 유도할수 있다

커뮤니티 서비스

일대일 채팅을 통해 다양한 사람과 대화하고 만남을 주도한다.

확성기 채팅을 통해 여러 사람들에게 자신의 방을 홍보 하거나 만남을 주도 할 수 있다.

알림 기능으로 친구기능 채팅이 왔을때 실시간으로 확인할 수 있다

마이데이터 서비스

자신의 프로필 사진과 별명등을 변경한다.마이데이터를 통해 출석정보 , 주량기록을

🔗 링크 | <https://github.com/winckey/Juyuso>

2. 모여봐요 우리동네

담당 : 백엔드 개발 / 참여인원 5명

2022.01 - 2022.02

#JAVA #Spring #JPA #RDBMS #MSA #Docker #Kafka

• 이웃 간의 소통과 공동 배달, 공동 구매를 위한 어플리케이션

주요 서비스

- 이웃 간 소통을 위한 커뮤니티 기능
- 이웃과 공동 배달을 위한 게시판 제공
- 이웃과 공동 구매를 위한 게시판 제공
- 구매를 확정하기 위한 채팅 기능

 링크 | <https://github.com/winckey/Modong>

3. MILK - NFT 프로젝트

담당 : 백엔드 개발 / 참여인원 6명

2022.01 - 2022.02

#JAVA #Spring #JPA #RDBMS #NFT #Docker

핵심 기능

NFT 명품 디지털 인증서

- 명확한 소유자와 거래 과정 그리고 최초 인증서 발급 회사를 확인할 수 있는 NFT 디지털 인증서

NFT 거래

- Opensea와 같이 자유로운 NFT 거래
- 최초의 NFT를 구매한 구매자들은 자유롭게 해당 NFT를 플랫폼에서 판매 가능

NFT 실물화 과정

- NFT를 실물화 하게 되면 NFT 거래 정지 및 해당 실물화 내용이 NFT에 기록될 수 있게 구현

라이브 커머스(명품 NFT 경매 시스템)

 링크 | <https://github.com/winckey/MLIK>

교육과정

스마트팩토리 빅데이터교육 (문성고등학교)
스마트 팩토리 빅데이터 활용 교육

2021.06 - 2021.07

삼성 청년 SW 아카데미 (삼성전자)
SW역량향상 교육

2021.07 - 2022.06

자격사항

정보처리기사

2021.06

수상내역

SSAFY 공통프로젝트 우수상(삼성 SW 아카데미)

2022.02

깃허브 <https://github.com/winckey>

블로그 <https://velog.io/@winckey0>



01 주유소 (酒流所)

-비대면 화상 만남 프로젝트

프로젝트 기간 : 2022.1 - 2022. 2

주유소 - 화상 모임 서비스(공통프로젝트 우수상)

기간 : 2022 . 01 - 2022 . 02

담당 직무 : 백엔드 엔지니어(팀장)

참여인원 : 6명

Github : <https://github.com/winckey/Juyuso>

프로젝트 개요

코로나 시대에 술자리와 같은 모임을 가지기 힘든 사람들을 위해 비대면으로도 여러 놀이 요소들을 통해 화상으로 모임을 즐길수 있도록 도와주는 서비스 입니다.

주요기능 및 소개

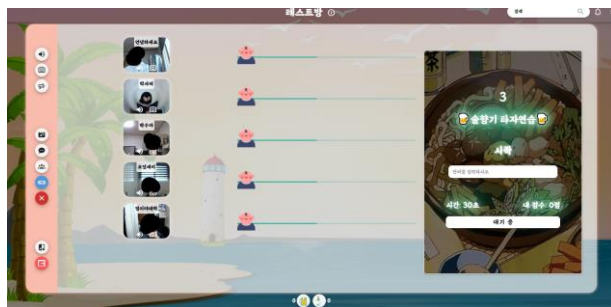
1.화상기능 서비스

- 자유롭게 방을 생성하고 친구 또는 새로운 사람들과 만남을 가질수 있다
- 다른사람들과 화상기능을 이용해 자유로운 만남을 가질수 있다.



2.게임 및 재미 서비스

- 단순한 화상회의의 서비스가 아닌 여러가지 게임을 통해 술자리에 맞는 분위기를 유도할수 있다.
- 사용자는 카메라 필터 음성변조기능을 이용하여 자신의 모습, 목소리를 바꿀수 있다.
- 게임과 재미기능들을 이용해 즐거운 술자리를 유도한다.



3.커뮤니티 서비스

- 일대일 채팅, 확장기 채팅을 을 통해 다양한 사람과 대화하고 만남을 주도한다.
- 알림 기능으로 친구기능 채팅이 왔을때 실시간으로 확인할 수 있다.



주요기능 및 소개

4.마이데이터 서비스

-자신의 프로필 사진과 별명등을 변경한다.

-마이데이터를 통해 출석정보,음주기록을 확인하고 저장할 수 있다



5.로그인 서비스

-웹 로그인과 카카오톡 로그인을 지원한다.



Expeiece

1. JWT

해당 프로젝트는 서버에 로그인 정보를 저장해야 하는 세션 방식보단 확장성을 고려해 토큰 유효성만을 체크하는 JWT 토큰방식을 이용하여 로그인을 구현했습니다.

또한 로그아웃과 재 인증 기능의 구현을 위해 refresh token 를 저장하기위한 추가 공간이 필요했습니다. JWT의 stateless와 토큰만료시 마다 refresh token에 접근해야 하기 때문에 데이터베이스 접속 부하를 줄이기 위하여 여러 DB중에 Spring에서 직접 지원해주는 Redis를 이용하여 인메모리에 저장하기로 결정하였습니다.

이를 이용하여 refresh token토큰을 저장하고 로그인 기능을 구현 하였습니다

2. 채팅기능 구현(Socket)

채팅기능 구현을 위하여 기존의 주기적인 호출을 통한 폴링 방식보다는 TCP 연결을 통한 WebSocket을 이용하기로 하였으며 trade off를 고려하여 스프링에서 지원하는 stomp를 이용하여습니다.

각 채팅 방 별로 방정보를 DB에 저장하고 엔드 포인트를 부여하여 WebSocket을 통해 실시간 채팅이 가능하도록 구현 하였습니다.

3. Error Handling

백엔드 에러 발생시 팀원간 서로다른 에러 처리로 인해 프론트와의 소통이 문제 발생으로 인해 에러 핸들링을 고려 하여 구현 하였습니다.

효율적인 에러관리와 중복코드를 막고 팀원간 통일된 에러처리를 위해 Custom Exception을 만들어 Global로 관리 하였습니다.

이를 통해 모든 에러처리를 GlobalExceptionHandler에 위임하고

Error Code를 정의하여 예외처리시 code만으로 에러를 요청하며 프론트와의 소통 또한 더욱 편리 해졌습니다.

```
public enum ErrorCode {  
  
    INVALID_REFRESH_TOKEN(BAD_REQUEST, message: "리프레시 토큰이 유효하지 않습니다"),  
    MISMATCH_REFRESH_TOKEN(BAD_REQUEST, message: "리프레시 토큰의 유저 정보가 일치하지 않습니다"),  
    CANNOT_FOLLOW_MYSELF(BAD_REQUEST, message: "자기 자신은 팔로우 할 수 없습니다"),  
    CANNOT_ENTER_MEETING(BAD_REQUEST, message: "방 인원 초과 입니다"),  
  
    INVALID_AUTH_TOKEN(UNAUTHORIZED, message: "권한 정보가 없는 토큰입니다"),  
    UNAUTHORIZED_MEMBER(UNAUTHORIZED, message: "현재 내 계정 정보가 존재하지 않습니다"),  
    USER_PW_INVALID(UNAUTHORIZED, message: "사용자의 PW가 일치하지 않습니다."),  
  
    USER_NOT_FOUND(NOT_FOUND, message: "해당 유저의 정보를 찾을 수 없습니다."),  
    FRIEND_NOT_FOUND(NOT_FOUND, message: "해당 유저와 친구가 아닙니다."),  
    FRIEND_REQUEST_NOT_FOUND(NOT_FOUND, message: "친구 요청 정보를 찾을 수 없습니다."),  
    MEETING_NOT_FOUND(NOT_FOUND, message: "해당 방을 찾을수 없습니다"),  
  
    USER_ID_DUPLICATE(CONFLICT, message: "중복된 사용자 ID입니다."),  
    USER_NICKNAME_DUPLICATE(CONFLICT, message: "중복된 사용자 닉네임입니다."),  
    USER_EMAIL_DUPLICATE(CONFLICT, message: "중복된 사용자 Email입니다."),  
    OAUTH_EMAIL_DUPLICATE(CONFLICT, message: "이미 아이디로 가입된 계정이 존재합니다."),  
    FRIEND_REQUEST_DUPLICATE(CONFLICT, message: "이미 친구 요청하였습니다."),  
    FRIEND_DUPLICATE(CONFLICT, message: "이미 등록된 친구입니다."),  
}
```

4. OpenVidu API

화상기능 구현을 위하여 WebRtc , kurento 개발의 효율성을 위해 OpenVidu가 가장 적합하다고 생각하였고 OpenVidu API를 Docker를 이용하여 별도의 동작하게 하여 프론트에서 서버로 바로 사용할 수 있도록 구현 하여 개발의 생산성을 증진시킬 수 있었습니다.

02 모여봐요 우리동네

-이웃 간의 소통을 위한 어플리케이션

프로젝트 기간 : 2022.5 - 2022. 6

모여봐요 우리 동네

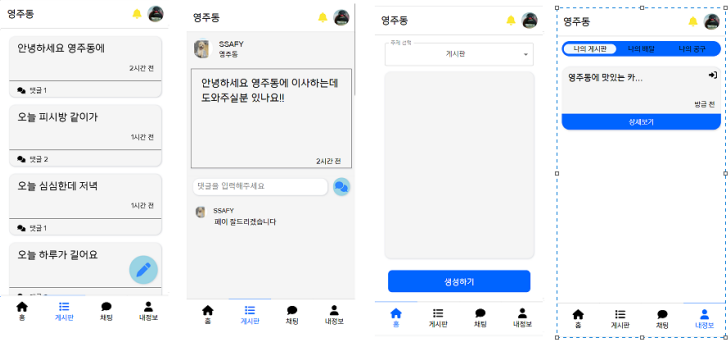
기간 : 2022 . 04 - 2022 . 05
담당 직무 : 백엔드 엔지니어
참여인원 : 5명
Github : <https://github.com/winckey/Modong>

프로젝트 개요

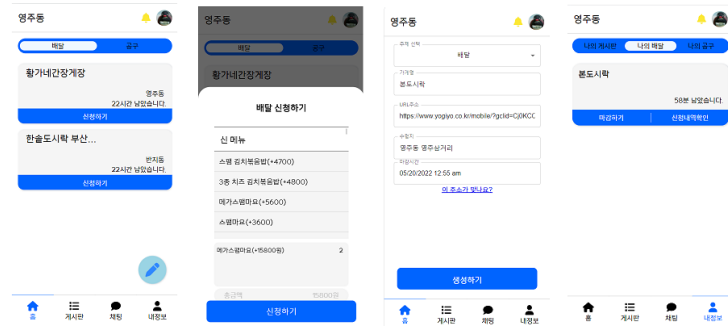
이웃 간의 소통과 공동 배달, 공동 구매를 위한 어플리케이션

주요기능 및 소개

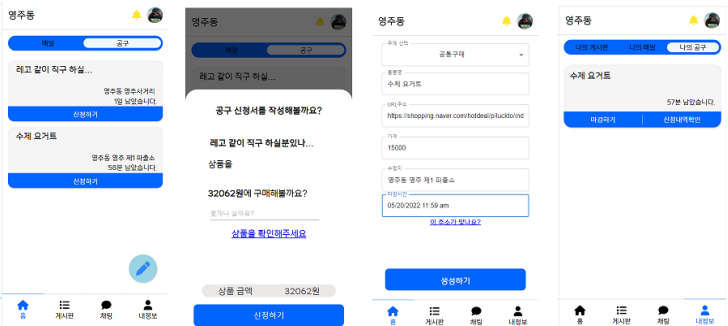
1. 이웃간 소통을 위한 커뮤니티 기능



2.. 이웃간 공동배달을 위한 게시판 기능



3. 이웃과 공동 구매를 위한 게시판 제공



1 MSA 프로젝트 구현

이 프로젝트 이전에는 mono프로젝트로 구현해 왔고 협업 중 코드수정에 따라 서비스가 다름에도 수정후 배포로 인해 서비스의 중단과 프론트와의 소통에서 불편함을 느꼈고 이를 해결하고 추가로 원활한 scale out 과 기능추가 유연성을 위해 MSA 구조로 설계하였으며 서비스 별로 개별Docker 와 데이터베이스를 가지도록 구현되었습니다.

```
> alarm-service
> apigateway
> board-service
> chatting-service
> config-service
> crawling-Service
```

```
> order-service
> order-service-re
> service-discovery
> user-service
> yml
```

각서비스 정보를 EurekaServer에 저장하고 Api-gateway를 통해 client 요청을 알맞은 서비스로 보낼수 있도록 구현하였습니다.

이를 통해 서비스간 더 분명한 책임 분리를 느낄 수 있었고 팀원간 백 엔드 통신에서도 서로 다른 서비스의 API를 호출 함으로 기능분리를 잘 느낄 수 있었습니다.

2 API 문서화

MSA구조로 인해 서비스가 많아지면서 api문서화에도 서비스 별로 문서화를 작성해주어야 하는 문제가 있었습니다. 이로인해 프론트에서 작업속도에 문제가 생겨 이를 해결하고자 하였습니다.

우선 spring의 swagger 통한 Api문서화를 시도하여 서비스별로 기능들을 확인 할수 있도록 하였습니다. 하지만 swagger 통합하지 못해 서비스별로 Api를 확인 하기 위해서는 Eureka를 통해 해당 서비스 포트와 주소를 확인 하고 접속하여 매 서비스 별로 다른 주소로 접속하는 해야 하였습니다.

이를 위해 api gateway에서 요청되는 서비스문서들을 추가로 필터에 등록후 'SwaggerResourcesProvider' 를 빈으로 등록하여 문서들을 리스트화 하여 출력후 문제를 해결하였습니다. 이를 통해 소통과 개발 속도 향상에 많은 도움을 주었습니다.

3 비동기 통신(Kafka)

주문 , 채팅 서비스는 주문마감이후 채팅방 생성으로 동작합니다. 서비스 분리로인해 서비스간 통신이 필요했고 이를 Fegin Clinet를 이용하여 동기로 처리하려 하였으나 주문마감이후 채팅방생성시 주문서비스에서 응답을 기다릴 필요성이 없어 이를 해결하고자 하였습니다.

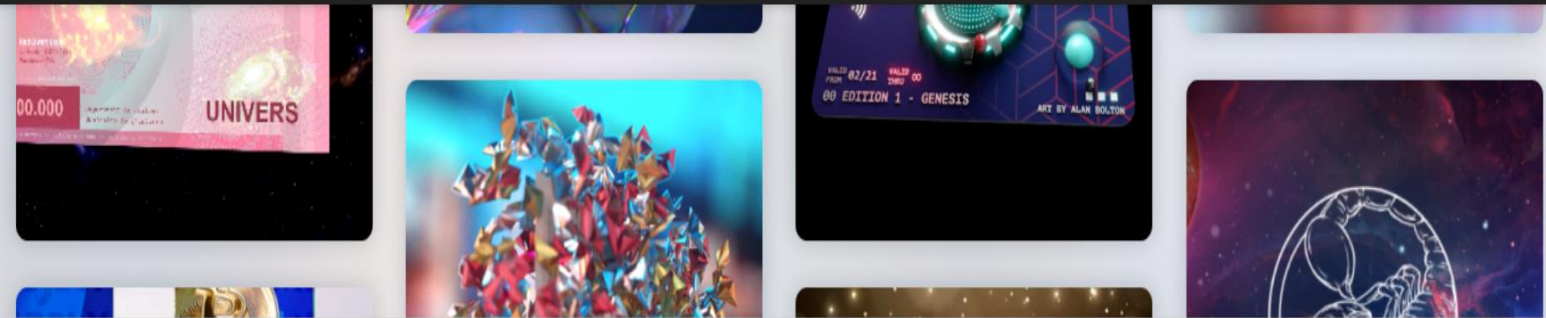
이를 위해 비동기 통신 방법중 하나인 Mseeage Queue방식을 사용하고자 하였고 여러 솔루션중 다수인원에게 메세지 전달을 위해 Kafka를 이용하기로 하였습니다. Kafka를 개별 Docker로 등록하여 서비스별 통신에 이용하였으며 이를 통해 주문마감후 채팅방 생성을 응답을 기다리지 않고 진행하는 방식으로 구현 하였습니다.

4 JWT

해당 프로젝트는 서버에 로그인 정보를 저장해야 하는 세션 방식보단 확장성을 고려해 토큰 유효성만을 체크하는 JWT 토큰방식을 이용하여 로그인을 구현했습니다.

또한 로그아웃과 재 인증 기능의 구현을 위해 refresh token 를 저장하기위한 추가 공간이 필요했습니다. JWT의 stateless와 토큰만료시 마다 refresh token에 접근해야 하기 때문에 데이터베이스 접속 부하를 줄이기 위하여 여러 DB중에 Spring에서 직접 지원해주는 Redis를 이용하여 인메모리에 저장하기로 결정하였습니다.

이를 이용하여 refresh token토큰을 저장하고 로그인 기능을 구현 하였습니다



Explore Luxury goods

고유성에 가치를 두는 명품 패션 시장도 NFT 시장에 눈을 돌리기 시작했습니다.
 실물 소재와 제작 과정, 장인 정신과 브랜드 헤리티지에 따라 고가의 가치가 매겨졌는데,
 NFT가 대상 하나하나에 고유의 객체라는 존재감을 나타낼 수 있어서 실물 명품보다 희소성은 우위에 있다고 볼 수 있습니다.
 MILC는 세계적인 명품 브랜드와 협업하여 NFT 뿐만 아니라 실물까지 받을 수 있도록 제공하고 있습니다.
 여기서 구매한 NFT를 자유롭게 거래하십시오.

Trending collections

02 MLIK -NFT 거래 플랫폼

포트폴리오

MILK – nft거래 서비스

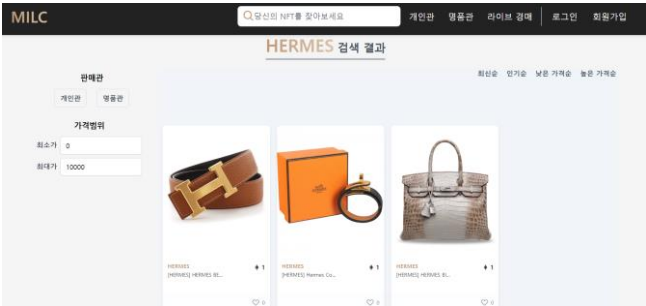
기간 : 2022 . 03 - 2022 . 04
담당 직무 : 백엔드 엔지니어
참여인원 : 6명
Github : <https://github.com/winkey/MILK>

프로젝트 개요

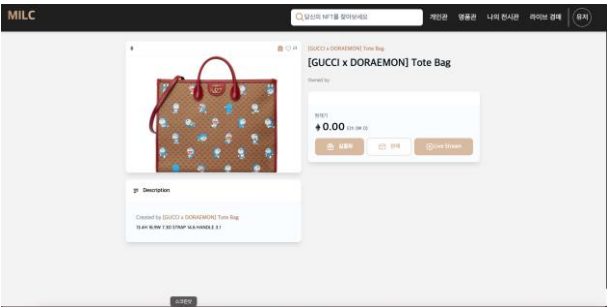
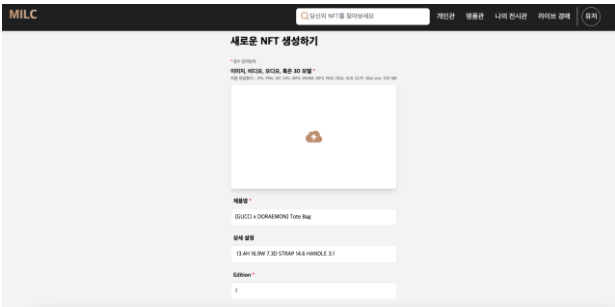
명품 디지털 인증서 발급과 자유로운 NFT 거래 플랫폼과 소유의 패러다임을 바꿔줄 나만의 전시장

주요기능 및 소개

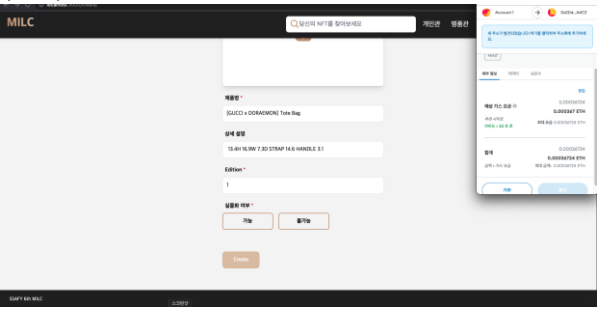
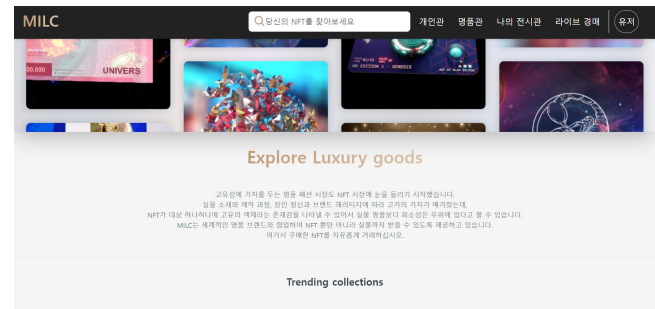
- 1.NFT 명품 디지털 인증서
-명확한 소유자와 거래 과정 그리고 최초 인증서 발급 회사를 확인할 수 있는 NFT 디지털 인증서
- 2.NFT 거래
-최초의 NFT를 구매한 구매자들은 자유롭게 해당 NFT를 플랫폼에서 판매 가능



- 3.NFT 실물화
-NFT를 실물화 하게 되면 NFT 거래 정지 및 해당 실물화 내용이 NFT에 기록될 수 있게 구현



- 4. NFT 갤러리
-NFT로 구매한 명품을 디스플레이에 전시할 수 있게끔 이미지 제공(360°)



MILK – NFT거래 서비스

기간 : 2022 . 03 - 2022 . 04
담당 직무 : 백엔드 엔지니어
참여인원 : 6명
Github : <https://github.com/winkey/MILK>

프로젝트 개요

명품 디지털 인증서 발급과 자유로운 NFT 거래 플랫폼과 소유의 패러다임을 바꿔줄 나만의 전시장

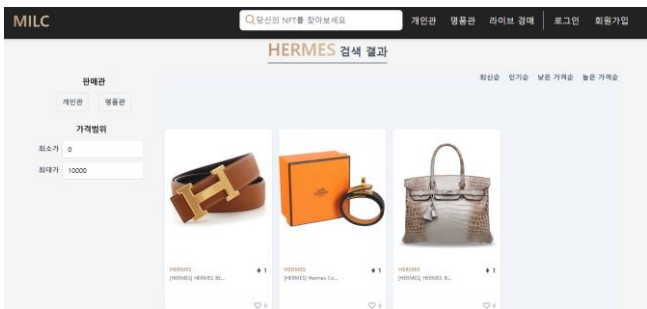
주요기능 및 소개

1.NFT 명품 디지털 인증서

-명확한 소유자와 거래 과정 그리고 최초 인증서 발급 회사를 확인할 수 있는 NFT 디지털 인증서

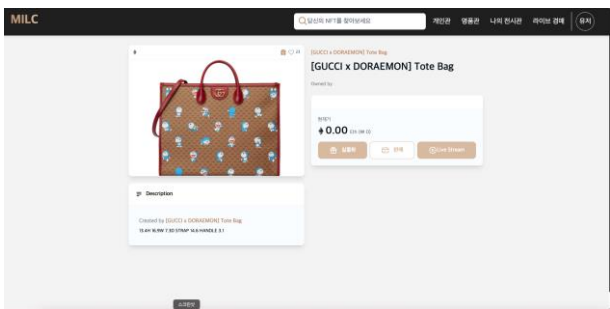
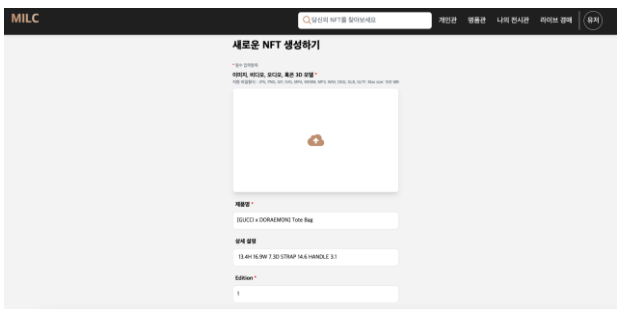
2NFT 거래

-최초의 NFT를 구매한 구매자들은 자유롭게 해당 NFT를 플랫폼에서 판매 가능



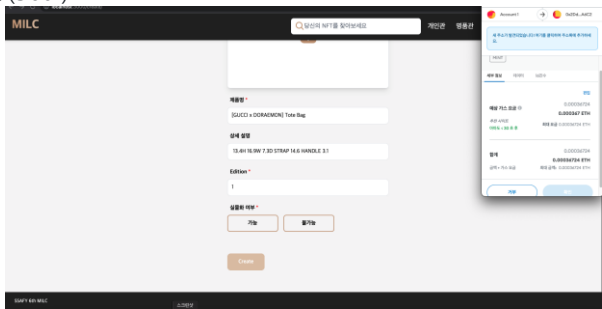
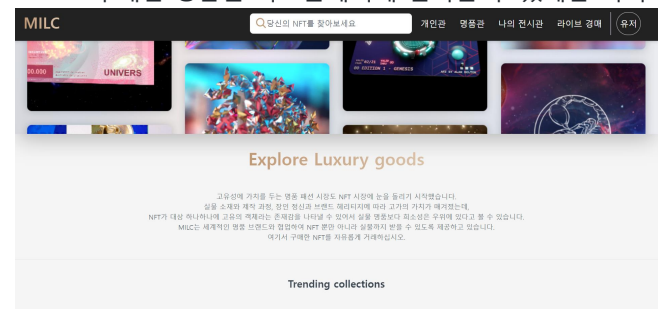
3.NFT 실물화

-NFT를 실물화 하게 되면 NFT 거래 정지 및 해당 실물화 내용이 NFT에 기록될 수 있게 구현



4. NFT 갤러리

-NFT로 구매한 명품을 디스플레이에 전시할 수 있게끔 이미지 제공(360°)



1 NFT 검색 속도 저하 문제

NFT를 민팅후 저장된 NFT를 조회하는데 있어 속도가 매우 저하되는 문제가 있었습니다. 이를 해결하기 위해 실무를 하고 있는 전문가분들에게 조언을 구해 실제 운용중인 OpenSea와 비슷한 방식으로 조회용 DB를 만들어 저장하였습니다.

이를 이용하여 플랫폼 이용시 DB의 데이터들을 보여주고 실제 거래시 스마트컨트랙트에 저장되어있는 데이터를 불러와 기능이 동작하도록 구현 하였습니다.

2 Spring security 예외 처리

로그인시 JWT의 위조, 만료, 형식오류 등의 예외 처리를 위하여 security에서 Exception을 발생시켰으나 동작하지 않는 문제가 있었습니다.

이는 Spring security필터에서의 예외는 @ExceptionHandler로 잡히지 않고 Spring Security는 Spring 이전에 Filter 처리가 되므로 DispatcherServlet 까지 가지 않기 때문입니다.

이를 해결하기위하여 Exception이 아닌 상속받은 Filter에서 직접 에러를 잡고 http응답을 프론트에 전달해 주었습니다.

3 채팅기능 구현(Socket)

채팅기능 구현을 위하여 기존의 주기적인 호출을 통한 폴링 방식보다는 TCP 연결을 통한 WebSocket을 이용하기로 하였으며 trade off를 고려하여 스프링에서 지원하는 stomp를 이용하여습니다.

각 채팅 방 별로 방정보를 DB에 저장하고 엔드 포인트를 부여하여 WebSocket을 통해 실시간 채팅이 가능하도록 구현 하였습니다.