

훈련기관 - 더조은컴퓨터아카데미

BTOF

경매에 미칠 수 밖에 없는 NFT 플랫폼

Feat. 분할판매

TEAM 3조 Blockchain Team On Fire

김선진, 박광범, 박재현, 이선우, 허진혁, 홍지연(ㄱ-ㅎ)

CONTENTS

BTFO 가즈아!!

01

프로젝트 개요

02

프로젝트 팀
구성 및 역할

03

프로젝트 수행
절차 및 방법

04

프로젝트 수행
결과

05

자체 평가 의견

프로젝트 주제 및 선정배경

주제 - NFT (Non-fungible token)

- NFT란 : 블록체인 기술을 이용하여 디지털 자산의 소유주를 증명하는 가상의 토큰. 디지털 콘텐츠는 물론 현실의 미술품 등에도 활용 중
- NFT 시장은 2020년도 1000억에서 2021년도 19조로 21,350% 상승했다.
- 미국 CES2022 미래 주목해야할 트렌드 기술 5개 중 하나로 지정
- 카카오, 네이버, LG 등 국내 굴지의 대기업과 주요 은행들도 NFT 시장에 뛰어 들고 있음

주제 선정배경

- 미래 유망 기술이라는 것을 판단함. 또한 본 과정이 ‘블록체인기반 핀테크 및 응용SW 개발자’인만큼 NFT에 대한 조원들의 관심도가 높았음
- 위의 이유들로 본 과정의 마지막 단계가 블록체인 관련 과정임을 알고 있으나 미리 관련된 지식들을 조사, 습득하는 것이 조원들에게 유익할 것으로 판단되어 NFT 판매 사이트를 만드는 것으로 주제선정을 함

01

모두가 올릴 수 있는 오픈마켓 플랫폼

누구든 콘텐츠크리에이터가 될 수 있는 플랫폼



02

경매 및 분할 판매

한정판 NFT 및 고가 예술품 실시간 경매 기능



03

카카오페이 결제 및 전자지갑 충전

NFT 카카오페이로 결제 가능

전자지갑 충전

판매자 전자지갑 출금

주요 기술 : 블록체인 없는 블록체인 플랫폼





우리만의 차별 점: 실시간 경매



프로젝트 팀 구성 및 역할 은 프로젝트를 기본 단위로 작성하며 팀원의 수에 따라 칸을 추가/삭제할 수 있다.

<div>훈련생</div> <div>박광범</div>	팀장	전반적인 프로젝트 관리 로그인 회원가입 입찰작품등록, 작가등록, 메인페이지, 지정자구매 입찰가 구매, 카카오페이 결제구현, 공지사항, 회원간 거래내역 계좌 등록 개발
<div>훈련생</div> <div>박재현</div>	팀원	와이어프레임, WebSocket 기반 실시간 경매 시스템 개발
<div>훈련생</div> <div>허진혁</div>	팀원	지정가판매, 입찰판매, 마켓플레이스, 카테고리, 마이NFT 콜렉션 개발,
<div>훈련생</div> <div>홍지연</div>	팀원	아티스트 등록페이지, DB테이블 관리 및 구축, 메인페이지, 개발 및 전반적인 개발 연동
<div>훈련생</div> <div>이선우</div>	팀원	관리자 페이지 작가/작품 승인처리, 메인페이지, 마이페이지 개발
<div>훈련생</div> <div>김선진</div>	팀원	ERD 테이블, 화면기능설계서, 전자지갑 입금 및 출금, 좋아요, 아티스트 페이지, 랭킹페이지

구분	기간	활동
사전 기획	4월 7일 ~ 11일	주요 시장 조사와 경쟁사 조사 및 웹사이트 컨셉 및 주요 기능 기획
화면 프로세스 정의서	4월 11일 ~ 12일	주요 기능과 경쟁사 벤치마킹으로 기본 화면 프로세스 정의
ERD 설립	4월 11일 ~ 12일	필요 데이터 선정 및 테이블 구축과 Entity-relationship Diagram 도식화
WireFrame 설계	4월 11일 ~ 14일	웹사이트 와이어프레임 구상 및 작성
개발	4월 14일 ~ 26일	담당 개발 착수
검토	4월 26 ~ 28일	전반적인 검토

블록체인 3조 BLOCKCHAIN TEAM ON FIRE

Level1	Level2	Level3	Level4	Level5	Level6	Level7	화면 ID	화면명	설명	화면유형	비고								
로그인							BTF_WLO01	로그인	아이디, 비번 로그인 / 소셜 로그인 (네이버, 카카오, 구글)	page	LOGIN								
	아이디 찾기						BTF_WFI01	아이디찾기	아이디찾기 및 결과	page	FIND ID								
	비번찾기						BTF_WFP01	비밀번호찾기	비번찾기 및 변경하기	page	FIND PW								
							N/A	비번 변경 안내	90일 경과 - 비번 변경 안내 팝업	popup	N/A								
							N/A	로그인 완료	로그인 완료 - 토스트 메시지	toast	로그인 시 빨간색으로 표시됨								
							N/A	아이디/비번 확인	아이디/비번 확인 - 토스트 메시지	toast	N/A								
							N/A	비번 불일치	비밀번호 불일치 - 토스트 메시지	toast	N/A								
							N/A	비번 변경 완료	비번 변경 완료 - 토스트 메시지	toast	N/A								
							N/A	비번 조합 안내	비번 조합 안내 - 토스트 메시지	toast	N/A								
							N/A	가입정보 불일치	가입정보 불일치 - 토스트 메시지	toast	N/A								
							N/A	이메일 확인 요청	이메일 확인 요청 - 토스트 메시지	toast	N/A								
	회원가입						BTF_WJO01	회원가입	회원가입 정보 - 이메일, 비번, 약관, 안내, 본인인증 / 소셜가입 (네이버, 카카오, 구글) 이력	page	JOIN								
		약관동의					BTF_WJO03	약관동의	약관동의	page	AGREEMENT								
		중복안내					BTF_WJO04	중복안내	가입된 계정 중복 안내	page	DUPLICATION INFO								
		소셜					N/A	소셜가입	소셜 가입정보 동의하기	N/A	API - 네이버, 카카오, 지메일	JOIN SOCIAL							
							BTF_WJO05	휴대폰 번호 등록	소셜 추가정보 - 휴대폰 번호 등록 -> 문자인증 (회원 가입 시 1회 문자인증으로 주류 구입 전까지 결제하기에서 정회원과 동일함)	page	소셜회원								
							BTF_WJO06	구글 OTP 등록	소셜 추가정보 - OTP 번호 등록 -> OTP 인증 (회원 가입 시 1회 인증으로 사용)										
							BTF_WJO07	취향 선택	개인적인 취향들										
							BTF_WJO02	닉네임 설정	소셜 추가정보 - 닉네임 설정	page	소셜회원								
							N/A			N/A	N/A								
							N/A		일반 회원가입과 동일	N/A	N/A								
		약관동의					N/A			N/A	N/A								
		중복안내					N/A			N/A	N/A								
		MyPage							BTF_WLO02	회원정보	로그인 된 회원정보	page	MEMBERSHIP INFO						
						N/A	닉네임 입력 요청	닉네임 입력 요청 - 토스트 메시지	toast	N/A									
						N/A	정보 수정완료	정보 수정완료 - 토스트 메시지	toast	N/A									
휴대폰번호변경						BTF_WLO05	휴대폰 번호 변경	휴대폰 번호 변경 -> 문자인증	page	정회원, 소셜회원									
						N/A	휴대폰 번호 변경 완료	휴대폰 번호 변경 완료 - 토스트 메시지	toast	N/A									
비밀번호 변경						BTF_WLO03	비밀번호 변경	비밀번호 변경	page	PW MODIFY MEMBERSHIP									
						BTF_WLO04	회원탈퇴	회원탈퇴	page	WITHDRAW									
탈퇴						N/A	회원정보 불일치	회원정보 불일치 - 토스트 메시지	toast	N/A									
						N/A	탈퇴 확인	탈퇴 확인 팝업	popup	N/A									
						N/A	탈퇴 완료	탈퇴 완료 팝업	popup	N/A									
로그아웃						N/A	로그아웃 완료	로그아웃 완료 - 토스트 메시지	toast	N/A									
marketplace	제품리스트		제품상세페이지	즉시구매	결제	결과	결과	BTF_WM01	메인 결과	배너, Browe by category 등 결과	page	추천순, 최신순, 가격순, 가나다							
		BTF_WPR01						작품 상세	작가, 작품설명, *가격평균선 등 결과	page	PRODUCT								
		BTF_WPA01						구매 확인창	구매 상품, 가격 결과	page									
		BTF_WPA02						원화 결제	원화로 결제	page									
		BTF_WPA03						가상화폐 결제	가상화폐로 결제	page									
		N/A						로그인 창	로그인 창 보여주기	popup	N/A								
		N/A						구매약관	구매약관 보여주기	popup	N/A								
		N/A						약관 확인	구매약관 체크 실패	popup	N/A								
								결제 완료	카카오페이/가상화폐 결제완료	page									
		BTF_WPARE01							결제완료 창 -> 제품리스트 페이지로 이동	popup									
		BTF_WPARE02							결제실패 창 -> 제품상세페이지로 이동	popup									
		N/A						OTP 인증	OTP 인증코드 입력	popup	N/A								
		N/A						OTP 인증 실패	OTP 인증코드 입력 실패	popup	N/A								
		BKT_WSN1						작가 상품 목록	작가가 만든 작품들 결과	page									
		myNFT						collection	판매중 미판매							출력선 확인	소유한 NFT 출력선	page	
																판매상품 목록	소유한 판매하는 출력선	page	
																보유상품 목록	소유한 판매하지 않는 출력선	page	

테이블명	테이블ID	개요				
관심목록	wish	테이블 따로만들지?				
칼럼명	칼럼ID	타입	PK키	FK키	NULL	특 성
아이디	wish_id	varchar(50)		O		
상품번호	wish_prod	varchar(50)		O		

테이블명	테이블ID	개요				
출석체크	attend					
칼럼명	칼럼ID	타입	PK키	FK키	NULL	특 성
아이디	att_id	varchar(50)		O		
출석일	att_check	date				sysdate

테이블명	테이블ID	개요				
관심분야	attention					
칼럼명	칼럼ID	타입	PK키	FK키	NULL	특 성
닉네임	atte_nick	varchar(50)		O		
음악	music	varchar(10)				01
영상	video	varchar(10)				02
미술	art	varchar(10)				03
사진	photo	varchar(10)				04
스포츠	sport	varchar(10)				05
무용	dancing	varchar(10)				06
패션	fashion	varchar(10)				07
동물	animal	varchar(10)				08
자연	nature	varchar(10)				09
게임	game	varchar(10)				10
가상현실	virtual	varchar(10)				11
과학	science	varchar(10)				12
여행	trip	varchar(10)				13
종교	religion	varchar(10)				14
공예	crafts	varchar(10)				15
건축	construct	varchar(10)				16

테이블명	테이블ID	개요				
은행	hbank					
칼럼명	칼럼ID	타입	PK키	FK키	NULL	특 성
계좌번호			이름			

테이블명	테이블ID	개요				
회원목록	user	주민번호 받음지, 회원등급 기준, 판매등급 기준				
칼럼명	칼럼ID	타입	PK키	FK키	NULL	특 성
아이디	user_id	varchar				이메일주소
비밀번호	user_pwd	varchar2(100)				암호화
전화번호	user_tel	varchar2(50)				비번찾기.
닉네임	user_nick	varchar2(50)				
생일(년)	user_bir1	varchar2(10)				
생일(월)	user_bir2	varchar2(10)				
생일(일)	user_bir3	varchar2(10)				
성별	user_gend	varchar2(10)				M,F
추천인ID	user_rec	varchar2(50)			○	리워드
포인트	user_wallet	number			○	지갑
관심분야	user_inter	varchar2(100)			○	테이블따로
구매순위	user_grade	varchar2(10)				
판매순위	user_sales	varchar2(10)				판매왕
파워여부	user_artist	varchar2(10)				U,A,M

아티스트 순위 테이블	작품 순위
칼럼명	칼럼명
작가 닉네임, 작가명	작가 (FP)
신청일	heart table
하트수	h_number bigint auto_increment
작가작품들	prod_no (fk)
상품고유번호	user_id (fk)
상품명	heart int
상품파일	constraint primary key(h_number)

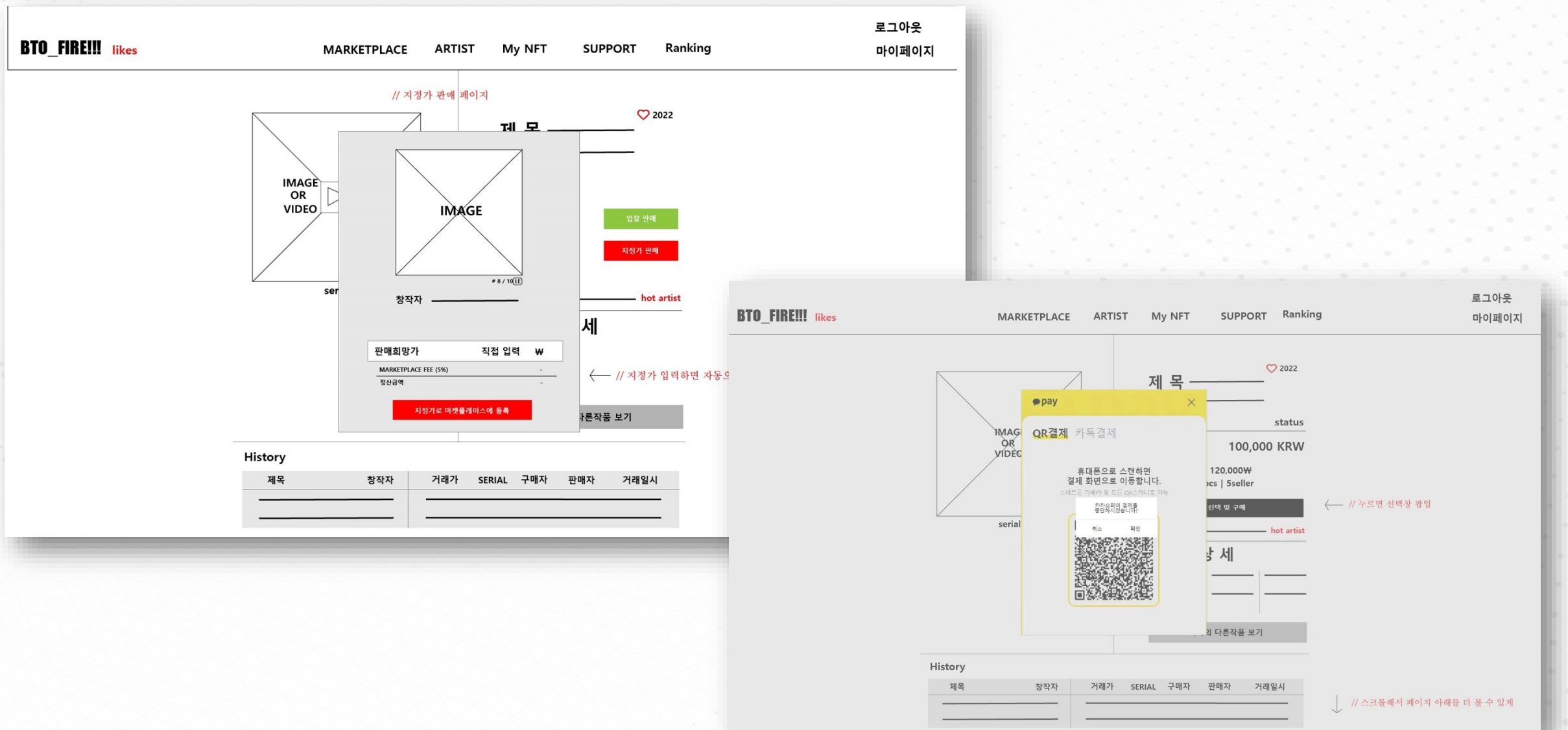
테이블명	테이블ID	개요					
관심분야	attention						
칼럼명	칼럼ID	타입	PK키	FK키	NULL	특	성
닉네임	atte_nick	varchar(50)		O			
음악	music	varchar(10)					01
영상	video	varchar(10)					02
미술	art	varchar(10)					03
사진	photo	varchar(10)					04
스포츠	sport	varchar(10)					05
무용	dancing	varchar(10)					06
패션	fashion	varchar(10)					07
동물	animal	varchar(10)					08
자연	nature	varchar(10)					09
게임	game	varchar(10)					10
가상현실	virtual	varchar(10)					11
과학	science	varchar(10)					12
여행	trip	varchar(10)					13
종교	religion	varchar(10)					14
공예	crafts	varchar(10)					15
건축	construct	varchar(10)					16

테이블명	테이블ID	개요					
회원목록	user	주민번호 받음지, 회원등급 기준, 판매등급 기준					
칼럼명	칼럼ID	타입	PK키	FK키	NULL	특	성
아이디	user_id	varchar(50)	O				이메일주소
비밀번호	user_pwd	varchar(100)					암호화
전화번호	user_tel	varchar(50)					비번찾기.
닉네임	user_nick	varchar(50)					
생일(년)	user_bir1	varchar(10)					
생일(월)	user_bir2	varchar(10)					
생일(일)	user_bir3	varchar(10)					
성별	user_gend	varchar(10)					
추천인ID	user_rec	varchar(50)			O		리워드
포인트	user_wallet	number(100)			O		지갑
관심분야	user_inter	varchar(100)			O		테이블따로
회원등급	user_grade	varchar(10)					
판매등급	user_sales	varchar(10)					판매왕
작가여부	user_artist	varchar(10)					T,F

03

프로젝트 수행 절차 및 방법

와이어프레임



구분	상세설명
개발 언어	Java jdk 11, Javascript, Ajax, jquery 3.6.0, json, html5, CSS
개발 툴	Eclipse
Server/DB	oracle DB 19C, Tomcat Server9, Eclipse
사용 API	SpringFramework 5.15 , Google Login, Kakao Login, KakaoPay, Websocket API


```

88
89
90 @Override
91 public int createUUID(String prod_dtl_name, String prod_dtl_artist, long price, int pieces) {
92     String[] uuid_val = new String[pieces];
93     for(int i = 0; i<pieces; i++) {
94         UUID uuid = UUID.randomUUID();
95         String str = uuid+"_";
96         uuid_val[i]=str;
97     }
98     int result = -1;
99     System.out.println("그냥 piece---"+pieces);
100     for(int i=0 ; i<pieces;i++){
101         System.out.println("for문 입니다");
102         result =memberMapper.insUUID(prod_dtl_name,prod_dtl_artist,price,uuid_val[i]);
103     }
104
105
106     return result;
107
108
109 }

```

작품을 분할해서 등록할때, 작가에게 희망하는 분할 갯수를 받아서 그 갯수만큼 uuid를 만들어서 조각당 시리얼 갯수 값을 넣어줘서 분할 판매를 구현함

PROD_DTL_NO	PROD_DTL_NAME	PROD_DTL_ARTIST	PROD_DTL_UUID	PROD_DTL_PRICE
1	53 ccArt1	artist3	b2d59b5d-179a-44ba-8d92-d3b559e81ec0	10000 T
2	54 ccArt1	artist3	e4d4206e8-832d-4801-8d33-9e86afcbcbef	10000 F
3	55 ccArt1	artist3	d2377286-3179-4451-9a13-61b0cad593ee	8000 T
4	56 ccArt1	artist3	19e91d50-d571-4c33-a801-6d4e86db72f2	90000 F
5	57 ccArt1	artist3	738099d7-9d99-41b5-8b0a-b2e29c0136e4	90000 F
6	58 ccArt1	artist3	c032697d-7bc8-4d6a-b4d1-fdbee9a6eeef	90000 F
7	59 ccArt1	artist3	0b971eb3-1e14-4e31-a035-19cbb940cd63	10000 F
8	60 ccArt1	artist3	702f20fd-5741-4911-9713-38543597c6d4	10000 T
9	61 ccArt1	artist3	eb020e8b-5f46-404a-ae63-34b0d0e9954a	10000 F
10	62 ccArt1	artist3	534581f6-29d4-4e98-9d8a-4d536735ceee	10000 F
11	63 nnnnn	pkkk	08e71dc0-d644-439c-950c-3bd50b1fcfef	100000 F
12	64 nnnnn	pkkk	e9112b13-4047-4851-90dd-ce694d1d445f	100000 F
13	65 nnnnn	pkkk	f36ffec3-c82c-405d-8be4-79513432d48a	100000 F
14	66 nnnnn	pkkk	c410fd4c-3240-464a-9469-9b07930e1ab6	100000 F
15	67 nnnnn	pkkk	e58ca05e-5680-40bf-89ca-5d75a18e2530	100000 F
16	68 nnnnn	pkkk	692c1fd1-8204-41b6-ad29-3d6f1bbe8ca2	100000 F
17	69 nnnnn	pkkk	d7b52291-3fd5-4be8-8f08-c939ed62d5e6	100000 F
18	70 nnnnn	pkkk	4b7f8fba-5e9b-4df4-8700-7807fa84b85c	100000 F
19	71 nnnnn	pkkk	d08474cb-1a0a-4bb4-b296-9059ed2e33e9	100000 F
20	72 nnnnn	pkkk	4beb244c-be09-440b-b069-a1b0794eeefc	100000 F
21	141 null	(null)	7638446c-b33c-4e60-8e7b-c5f10802d8a7	555 F
22	142 null	(null)	0b8e263b-fe2c-42d3-b165-d6d6b08e544a	555 F

실제로 판매도 이름은 같은 제품이지만 각각 다른 uuid로 인해서 nft같이 조각으로 분할되었기 때문에 다른 제품처럼 구매 와 판매가 되는 모습

작가: artist3

제목: ccArt1

창작자: artist3

분할갯수: 1

SERIAL 가격 판매자

1	입찰	120000	aa@aa.com	입찰
2	지정가	10000	kimkim	구매
3	입찰	54000	aa@aa.com	입찰
4	지정가	90000	kimkim	구매
5	지정가	90000	aa@aa.com	구매
6	지정가	90000	aa@aa.com	구매
7	입찰	51000	aa@aa.com	입찰
8	지정가	10000		구매
9	지정가	10000	jj	구매
10	지정가	10000		구매

작품 이름은 같지만 서로 다른 uuid로 인해서 DB에서도 서로 다른 제품으로 인식하는 모습

04

프로젝트 결과

입찰 소스, 입찰 화면, 잔액 거절, 입찰성공

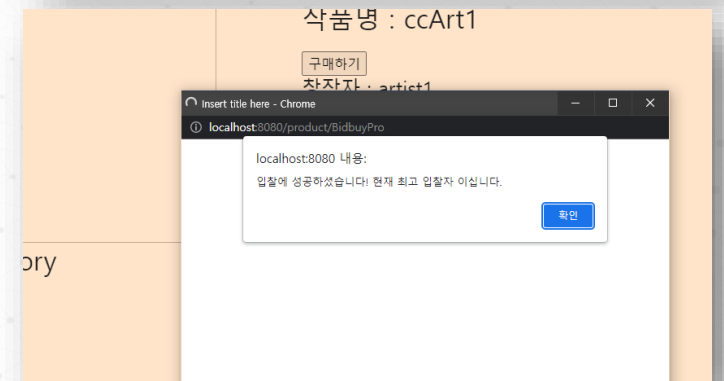
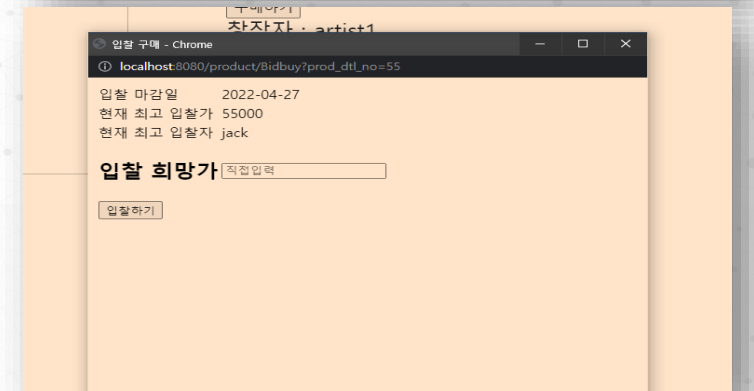
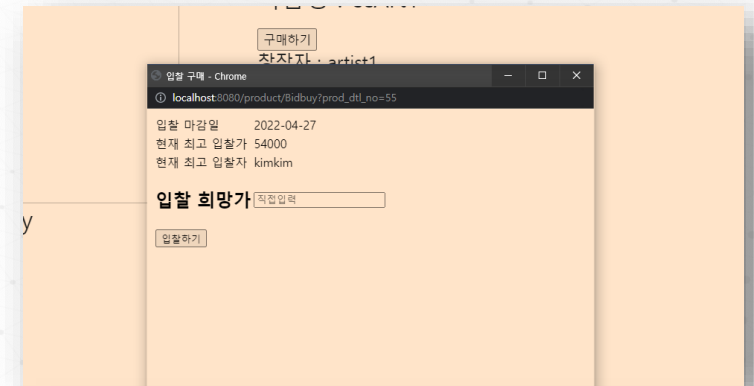
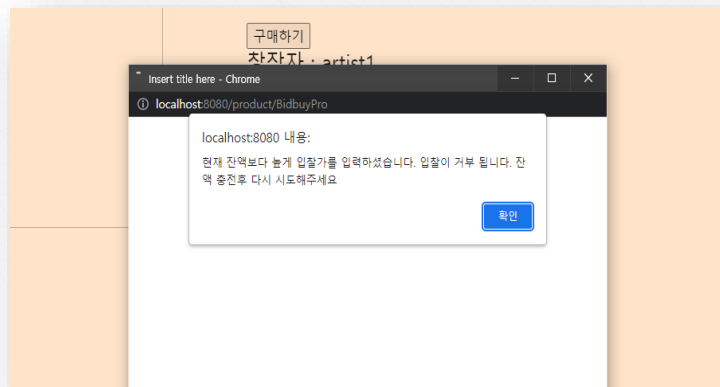
```

model.addAttribute("list",vo);
}
@PostMapping("BidbuyPro")
public String BidbuyPro(SellStatusVO vo,HttpSession session,Model model) {

    System.out.println("BidbuyPro입니다"+vo.getProd_dtl_bid_price());
    System.out.println("BidbuyPro입니다"+vo.getProd_dtl_artist());
    System.out.println("BidbuyPro입니다"+vo.getProd_dtl_name());
    System.out.println("BidbuyPro입니다"+vo.getProd_dtl_no());
    String id = (String)session.getAttribute("memId");
    vo.setId(id);
    System.out.println("id입니다-----"+id);
    int result = -1;
    int balance = productService.getBalance(vo); // 잔액 가져오는 쿼리문
    int price = vo.getProd_dtl_bid_price();
    if(price>balance) { // 입찰 희망 가격이 현재 잔액보다 높다면
        System.out.println("입찰 희망 가격보다 잔액이 작아서 입찰 실패");
        result=0;
    }
    else if (balance>price) { //잔액이 입찰 희망 가격보다 높다면
        System.out.println("잔액이 입찰 희망가보다 높아서 정상적으로 입찰 처리");
        result = productService.changeBid(vo);
    }
    System.out.println("입찰하는 쿼리문 실행후-----"+result);
    model.addAttribute("result",result);

    // update 쿼리문 날리기
    // update test_dtl2 set prod_dtl_bid_price = #{prod_dtl_bid_price}, prod_dtl_owner=id when
    return "/product/BidbuyPro";
}

```



```

//낙중
@GetMapping("enrollProduct")
public void enrollProduct() {
    System.out.println("작품 등록 페이지 요청");
}

@PostMapping("enrollProductPro")
public String enrollProductPro(EnrollProductVO enroll, MultipartHttpServletRequest request, HttpSession session,
    MultipartFile mf = request.getFile("file"));
    System.out.println("getname 메소드 실행될 때 출력문" + mf.getName());
    System.out.println("getOriginalName 실행될 때 출력문" + mf.getOriginalFilename());
    String fimeName = mf.getOriginalFilename();
    enroll.setProd_file(fimeName);
    String id = (String) session.getAttribute("memId");
    String artist = memberService.getArtist(id);
    enroll.setProd_artist(artist);
    int result = memberService.insProd(enroll);

    System.out.println("파일이름 : " + enroll.getProd_file());
    System.out.println("prod_name : " + enroll.getProd_name());
    System.out.println("cate : " + enroll.getProd_tag());
    System.out.println("pieces : " + enroll.getProd_div());
    System.out.println("price : " + enroll.getProd_price());
    System.out.println("작품소개 : " + enroll.getProd_descr());
    model.addAttribute("result", result);

    //int uuid_result = memberService.createUUID(enroll.getProd_name(), enroll.getArtist(), enroll.getProd_price());
    //System.out.println("uuid-result -----: " + uuid_result);
    // UUID는 서비스에서 만들기

```

작품 등록 신청 페이지

localhost:8080/member/enrollProduct

네이버 다음 구글 저가 항공권 출발지... Gmail YouTube 지도 Bex 배송대행지

등록할 작품명 작품종류

작품파일 업로드

※ 승인심사 후에 작품이 marketplace에 등록이 가능합니다

선택된 파일 없음

분할 갯수

수수료 발생 : 분할 갯수당 10000원이 부과됩니다.

가격

작품소개


```
//낙중
@GetMapping("enrollProduct")
public void enrollProduct() {
    System.out.println("작품 등록 페이지 요청");
}

@PostMapping("enrollProductPro")
public String enrollProductPro(EnrollProductVO enroll, MultipartHttpServletRequest request, HttpSession session,
    MultipartFile mf) {
    System.out.println("getname 메소드 실행될 때 출력문" + mf.getName());
    System.out.println("getOriginalName 실행될 때 출력문" + mf.getOriginalFilename());
    String fimeName = mf.getOriginalFilename();
    enroll.setProd_file(fimeName);
    String id = (String) session.getAttribute("memId");
    String artist = memberService.getArtist(id);
    enroll.setProd_artist(artist);
    int result = memberService.insProd(enroll);

    System.out.println("파일이름 : " + enroll.getProd_file());
    System.out.println("prod_name : " + enroll.getProd_name());
    System.out.println("cate : " + enroll.getProd_tag());
    System.out.println("pieces : " + enroll.getProd_div());
    System.out.println("price : " + enroll.getProd_price());
    System.out.println("작품소개 : " + enroll.getProd_descr());
    model.addAttribute("result", result);

    //int uuid_result = memberService.createUUID(enroll.getProd_name(), enroll.getArtist(), enroll.getProd_price());
    //System.out.println("uuid-result -----: " + uuid_result);
    // UUID는 서비스에서 만들기
```


작품등록 소스 코드 작가 등록 구현 화면

```

}
@PostMapping("enrollArtistPro")
public String enrollArtistPro(ArtistVO vo, MultipartHttpServletRequest request, Model model, HttpSession ses

    String id =(String) session.getAttribute("memId");
    String nick=memberService.getNick(id); // 아이디 주고 닉네임 받아오기
    vo.setArtist_nick(nick);

    System.out.println("작가등록프로페이지요청");
    MultipartFile mf = request.getFile("img");
    System.out.println("작가 이름 :"+vo.getArtist_name());
    System.out.println("자기소개 :"+vo.getArtist_introd());
    System.out.println("파일 크기 : "+mf.getSize());
    System.out.println("파일 확장자 :"+mf.getName());
    System.out.println("파일 이름"+mf.getOriginalFilename());
    System.out.println("enrollArtist vo 입니다-----"+vo);
    String savePath=request.getRealPath("resources\\save");
    System.out.println("savePath :"+savePath);
    String orgName= mf.getOriginalFilename();
    UUID uuid =UUID.randomUUID();
    String newName = uuid+orgName;
    vo.setArtist_id(newName);
    String imgPath = savePath +"\\\\"+newName;
    File f= new File(imgPath);
    try {
        mf.transferTo(f);
    }catch(IOException e) {
        e.printStackTrace();
    }

    int result = memberService.insArtist(vo);
    System.out.println("insArtist 후 result 입니다~~~"+result);
    model.addAttribute("result",result);
    return "/member/enrollArtistPro";
}

```



Insert title here x +

localhost:8080/member/enrollArtist

네이버 다음 구글 저가 항공권 출발지... Gmail YouTube 지도 백송대행지 컴퓨터

등록할 작가명

본인 인증 첨부서류

※ 주민등록증, 운전면허증, 여권 등 법적 효력이 있는 신분증만 유효합니다.

파일 선택 선택된 파일 없음

포트폴리오 URL

※ 심사 받을 포트폴리오의 링크 추가

자기소개

신청하기 reset 취소

계좌등록 소스 코드 계좌 불일치 , 계좌 일치시 등록 성공

```

        model.addAttribute("result",result);
        return "/member/product_detail";
    }
}
@GetMapping("enroll_ba")
public void enroll_ba() {
    System.out.println("계좌등록 폼 요청");
}

@PostMapping("enroll_baPro")
public String enroll_baPro(String pwch,Model model) {
    int result = memberService.checkPw(pwch);
    System.out.println("controller Result : "+result);
    model.addAttribute("result",result);
    // 쿼리문 작성 select count(*) from hbank where acOwner=#{acOwner} and ac_num=#{ac_num}
    return "/member/enroll_baPro";
}

@RequestMapping("ajaxBankCheck")
public ResponseEntity<String> ajaxBankCheck(String Owner , String ac_num){
    System.out.println("acOwenr :"+Owner);
    System.out.println("ac_num :"+ac_num);
    String res = "";
    int result = memberService.checkAccount(Owner , ac_num);
    if(result == 1) {
        res = "사용가능한 계좌입니다";
    }else {
        res ="계좌 정보가 불일치 합니다. 다시 입력해주세요";
    }
    HttpHeaders responseHeader = new HttpHeaders();
    responseHeader.add("Content-Type","text/html;charset=utf-8");
    return new ResponseEntity<String>(res,responseHeader,HttpStatus.OK);
}

```



계좌 등록 페이지

localhost:8080/member/enroll_ba

예금주명
박광범

계좌번호
1234

계좌확인

계좌 정보가 불일치 합니다.

비밀번호 확인

등록하기

계좌 등록 페이지

localhost:8080/member/enroll_ba

예금주명
박광범

계좌번호
1111-1111

계좌확인

사용가능한 계좌입니다

비밀번호 확인

등록하기

지정가 판매 팝업 페이지

```
//판매처리
@GetMapping("sellPopup")
public void sellPopup(SellVO vo, Model model, HttpSession session) {
    String id=(String)session.getAttribute("memId");

    vo.setProd_dtl_owner(id);
    List<String> tF = productService.getTF(vo);
    model.addAttribute("tF", tF);
    System.out.println(tF);
    List<Integer> prdNoList = productService.getSellListWithF(vo);
    List<Integer> r = productService.getSN(vo);

    System.out.println("vo 입니다~~"+vo);
    model.addAttribute("vo",vo);
    System.out.println("r값들 모음입니다!!"+r);

    int i=0;
    for(String n:tF) {
        if(n=="T") {
            r.remove(i);
            prdNoList.remove(i);
        }
        i++;
    }
    model.addAttribute("r",r);
    model.addAttribute("prod_dtl_no", prdNoList);
}
```

sellPopup - Chrome

localhost:8080/product/sellPopup?prod_name=ccArt1

작품명 : ccArt1
창작자 : artist3

○ 1 판매 중
○ 2 판매 중

판매희망가 :

marketplace fee(10%)

정산 금액 :

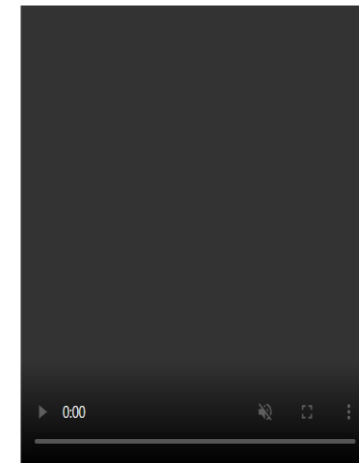
승인 대기중 작품 코드

```

82 @GetMapping("ready")
83 public void ready(ProductVO vo, @ModelAttribute("cri")Criteria cri, Model model,HttpSession session) {
84     vo=adminService.get(vo.getProd_no());
85     //분할갯수
86     model.addAttribute("product", adminService.get(vo.getProd_no()));
87 }
88 //승인처리
89 @PostMapping("readyPro")
90 public String readyPro(ProductVO product, HttpSession session,Model model,RedirectAttributes rttr) {
91     int prod_number = product.getProd_no(); // 상품 번호 저장하고
92     model.addAttribute("product",adminService.updatePass(product)); //업데이트 쿼리문 날려주고
93     // 제품 번호가 넘어 오니까 제품 번호를 통해서 작품명이랑 작가명 가격 분할갯수
94     product = adminService.bringPro(product);
95     String prod_dtl_name = product.getProd_name(); // 작품명
96     String prod_dtl_artist = product.getProd_artist(); //작가명
97     long price = product.getProd_price(); //가격
98     int pieces = product.getProd_div(); // 분할갯수
99     System.out.println("product");
100     int uuid_res = adminService.createUUID(prod_dtl_name,prod_dtl_artist,price,pieces); // 작품명이랑 작가
101     rttr.addAttribute("uuid_res",uuid_res);
102     return "redirect:/admin/Approval";
103 }
104 //거부처리
105 @PostMapping("readyPro2")
106 public String readyPro2(ProductVO product, HttpSession session, Model model) {

```

ready



승인여부 F

승인

거절

아티스트 작가 신청 코드

```

@RequestMapping("ajaxNickAvail")
public ResponseEntity<String> ajaxNickAvail(String nick) {
    System.out.println("ajaxNickAvail 요청 : " + nick);
    int result = commonService.NickAvail(nick);
    String resStr = "";
    if(result == 1) {
        resStr = "중복 not available";
    }else {
        resStr = "가능 available";
    }

    // 헤더 정보 추가 . 응답해더
    HttpHeaders responseHeader = new HttpHeaders();
    responseHeader.add("Content-Type", "text/html; charset=utf-8");

    return new ResponseEntity<String>(resStr, responseHeader, HttpStatus.OK);
}

<h4 style="display: none;"><a href="">작가등록신청폼</a></h4>
<form action="/member/enrollArtistPro" method="post" enctype="multipart/form-data" name="enrollArt
<!-- 어드민으로 보내기 -->
<!--
    비밀번호 확인<br/>
    <input type="password" name="pwCheck"> <br/>
-->

    등록할 작가명<br/>
    <input type="text" name="artist_name" width="250px">
    <!-- <button onclick="window.location='member/artistNCheck'">중복확인</button> -->
    <hr/>
    <div class="formInnerWrap">
        <div class="left item">
            본인 인증 첨부서류
        </div>
        <div class="right item">
            <p>* 주민등록증, 운전면허증, 여권 등 법적 효력이 있는 신분증만 유효합니다.</p>
            <input type="file" name="img" />
        </div>
    </div>
    <hr/>
    <div class="formInnerWrap">
        <div class="left item">
            포트폴리오 URL
        </div>
        <div class="right item">
            <p>* 심사 받을 포트폴리오의 링크 추가</p>
            <input type="text" name="artist_url" /><!-- button으로 하면 form에 연결된요청으로 들어감 -->
        </div>
    </div>
    <hr />
    자기소개 <br />
    <textarea rows="10" cols="45" name="artist_introd"></textarea>

    <div class="right">
        <input type="submit" value="신청하기" />
    </div>

```



BTO_FIRE ♥LIKE

MARKETPLACE | ARTIST | RANKING | SUPPORT

LOGIN | SIGN UP

등록할 작가명

본인 인증 첨부서류

※ 주민등록증, 운전면허증, 여권 등 법적 효력이 있는 신분증만 유효합니다.

Choose File No file chosen

포트폴리오 URL

※ 심사 받을 포트폴리오의 링크 추가

자기소개

신청하기

reset

취소

전자지갑 충전 및 거래 확인

```

@ResponseBody
@PostMapping(value = "cashIn")
public String cashIn(@RequestParam ("amount") int amount, String id, Model model, HttpSession session)
//은행계좌 있는지 확인
id = (String)session.getAttribute("memId");
Integer checkResult = memberService.doYouHaveAccount(id);
if(checkResult > 0) {
    //String id = session.getId();
    int wallet = memberService.getBalance(id);
    int addamount = amount+wallet;
    int result = memberService.updateBalance(addamount,id);
    model.addAttribute("result",result);

    String r_id = id;
    int deposit = amount;
    memberService.recordCashFlowForDeposit(r_id, deposit);

    int currentWallet = memberService.getBalance(id);
    model.addAttribute("currentWallet", currentWallet);
}

```

```
return checkResult.toString();
```

```

@GetMapping("cashFlowRecord")
public void showRecord(Model model, String id, HttpSession session) {
    id = (String)session.getAttribute("memId");
    int num= 1;
    List<CashFlowRecordVO> result = memberService.showRecordForCashFlow(id, num);
    model.addAttribute("recordList", result);
}

```

```

@GetMapping(value="cashFlowRecord2", produces = {MediaType.APPLICATION_JSON_UTF8_VALUE})
public ResponseEntity<List<CashFlowRecordVO>> showRecordForCashFlow(String id, Integer num, Model model,
    id = (String)session.getAttribute("memId");
    System.out.println("*****" + id + num);
    List<CashFlowRecordVO> recordList = memberService.showRecordForCashFlow(id, num);
    System.out.println("*****" + recordList);
    System.out.println("*****" + recordList + "가져오기 성공!!");
    model.addAttribute("recordList", recordList);

```

```

HttpHeaders responseHeader = new HttpHeaders();
responseHeader.add("Content-Type", "text/html;charset=utf-8");
return new ResponseEntity<>(recordList, HttpStatus.OK);
}

```

cash-in

현재 아이디 : "kimkim"

현재금액 : "9890000"

계좌 예금주 : "kimkim"

계좌번호 : "123123"

cashFlowRecord

최근 일간의 거래내역 조회

아이디 출금액 입금액 거래기록

kimkim 0	100000	2022-04-29	금요일	11:29:29
kimkim 0	5000	2022-04-29	금요일	11:29:34
kimkim 0	70000	2022-04-29	금요일	11:29:44
kimkim 0	150000	2022-04-29	금요일	11:29:47

고객은 먼저 계좌등록을 하여야 충전과 출금을 할 수 있음
충전과 출금 이후 원하는 날짜를 적어 거래기록을 확인할 수 있음

최신 및 HOT 아티스트 랭킹

```

@GetMapping("artistRanking")
public void getArtistRanking(Model model, HttpSession session) {
    System.out.println("*****artistRanking");
    String id = (String)session.getAttribute("memId");
    List<ArtistVO> artistRankList = artistService.getRankList();
    List<ArtistVO> artistListByStartDay = artistService.getListByStartDay();

    List<Integer> prod_no_list1 = artistRankList.stream().map(vo -> vo.getProd_no()).collect(Collectors.toList());
    List<Integer> artistRankListLikes = new ArrayList<Integer>();
    for(int no : prod_no_list1) {
        artistRankListLikes.add(memberService.getLikes(id, no));
    }

    List<Integer> prod_no_list3 = artistListByStartDay.stream().map(vo -> vo.getProd_no()).collect(Collectors.toList());
    List<Integer> artistListByStartDayLikes = new ArrayList<Integer>();
    for(int no : prod_no_list3) {
        artistListByStartDayLikes.add(memberService.getLikes(id, no));
    }
    System.out.println("*****artistListByStartDayLikes");
    System.out.println(artistListByStartDay);
    System.out.println(prod_no_list3);
    System.out.println(artistListByStartDayLikes);

    model.addAttribute("artistRankList", artistRankList);
    model.addAttribute("artistRankListLikes", artistRankListLikes);

    model.addAttribute("artistListByStartDay", artistListByStartDay);
    model.addAttribute("artistListByStartDayLikes", artistListByStartDayLikes);
    // 리스트

    List<ArtistVO> prod_no2 = artistService.getRankList();

























    //좋아요 별 리스트 **별도 좋아요용 테이블 필요
    //전체 작가 리스트
    Long artist_name_total = artistService.getRankList().stream().
        map(ArtistVO::getArtist_name).count();

    System.out.println("*****" + artist_name_total);
    model.addAttribute("RankingNum", artist_name_total);

    Long product_total = artistService.getRankListByLikes().stream().count();
    model.addAttribute("RankingNumByLikes", product_total);
}

```



HOT ARTIST					Ranking				
1					2				
artist1	작가페이지로이동	구매하러가기			artist1	작가페이지로이동	구매하러가기		
작가 URL	작가소개	대표작품	작품소개		작가명	작가이미지	작품명	대표작	작품소개
instagram.com/aa		 	초코파이는정		artist1	초코파이	 	 	초코파이는정
2					3				
artist1	작가페이지로이동	구매하러가기			artist1	작가페이지로이동	구매하러가기		
작가 URL	작가소개	대표작품	작품소개		작가명	작가이미지	작품명	대표작	작품소개
instagram.com/aa		 	im aa		artist1	again	 	 	hi hi hi hi
3					4				
artist1	작가페이지로이동	구매하러가기			artist1	작가페이지로이동	구매하러가기		
작가 URL	작가소개	대표작품	작품소개		작가명	작가이미지	작품명	대표작	작품소개
instagram.com/aa		 	im bb		artist1	iii	 	 	iiiiiiii
4									
artist1	작가페이지로이동	구매하러가기			artist1	작가페이지로이동	구매하러가기		
작가 URL	작가소개	대표작품	작품소개		작가명	작가이미지	작품명	대표작	작품소개
instagram.com/aa		 	im cc		artist1	hiiiaqqq	 	 	welshaaaaa

소감

우선 국가전략산업직종훈련으로 지원을 받아 좋은 수업을 받을 수 있었던 것에 감사한다. 강사님과 학원도 최고였다

블록체인기반 핀테크 및 응용SW개발자 과정의 첫 프로젝트로 웹개발을 하면서 모든 조원들은 자바언어 기반으로 개발을 할 수 있는 기본적인 역량을 갖추게 되었다.

또한 팀 프로젝트로 진행되는 과정을 거치며 모든 조원들은 서로 다른 배경지식과 수준을 이해하며 공동작업을 하였다. 이 경험은 앞으로 새로운 환경에서 업무를 하는데 있어 많은 도움이 될 것으로 판단된다.
대한민국 짱짱.

THANK YOU

(주)BTFO

서울시 강남구 테헤란로 270길 8, SR빌딩 3층 | 고객센터 02-2222-2222