

2021-1학기

캡스톤디자인종합프로젝트2

프로젝트 최종 보고서

팀번호 (팀명)	ARTARY
프로젝트명	ARTA : 사용자 맞춤형 온라인 전시회 플랫폼
프로젝트 수행기간	2021. 3. ~ 2021. 6.
팀장 및 팀원	송무송, 문석희, 조은진, 주소영
프로젝트 지도교수	이수원
작성일	2021. 06. 16.

〈목 차〉

1. 프로젝트 개요	4
1.1 프로젝트 배경	4
1.2 프로젝트 최종목표	4
1.3 키워드	4
2. 국내·외 기술과 시장 현황	5
2.1 관련 서비스 현황	5
2.2 관련 연구 현황	6
2.3 관련 지적 재산권 현황	8
2.4 본 프로젝트의 차별성	8
3. 프로젝트 설계	9
3.1 서비스 개념도	9
3.2 주요기능	10
3.3 서비스 구조도	11
3.4 데이터베이스 설계	12
3.5 사용자 인터페이스 디자인	17
3.6 사용자 인터페이스 설계서	19
3.7 REST API 설계서	27
3.8 모듈별 기능 명세서	34

4. 프로젝트 수행 결과	42
4.1 개발환경	42
4.2 Open Source & Library	42
4.3 구현 내용	43
4.4 결과 화면	50
4.5 평가 항목 및 방법	52
4.6 평가 결과	52
4.7 구현 시 문제 상황 및 해결 방법	53
5. 기대 효과 및 활용 방안	55
5.1 기술적 측면	55
5.2 경제적 산업적 측면	55
5.3 사회 문화적 측면	55
6. 프로젝트 수행 체계	56
6.1 팀원 인적 사항	56
6.2 프로젝트 역할 분담	56
6.3 프로젝트 수행 일정	57
6.4 프로젝트 수행 방법	57
7. 참고 문헌	58

1. 프로젝트 개요

1.1 프로젝트 배경

현재 코로나 바이러스에 의하여 다수의 사람들이 밀접하는 공간에 대한 방문이 급감하고 이에 전시회 또한 정상적인 운영을 진행하지 못하고 있다. 이로 인한 비대면 산업의 증가는 예술 분야가 대중에게 쉽게 다가갈 수 있는 전환점이 될 수 있다. 비대면은 단순히 스마트폰 단말기만 있으면 연령, 지위, 지역에 관계없이 누구나 접근 가능함을 뜻하며, 전시회를 온라인으로 제공함으로써 접근성의 경계를 없애 보편적 문화로 자리 잡을 수 있는 기회가 될 것이다. 그러나 온라인 전시회는 검색한 범위에 한해서 관람이 가능하며 직접적인 경험이 불가능하다는 단점이 있다. 즉, 정보 전달에 부족함이 있으며 커뮤니케이션에 제약이 있음을 뜻한다.

이러한 배경에 따라 온라인 전시회와 오프라인 전시회의 단점이 상호 보완된 온라인 전시회의 필요성이 증대되고 있다. 본 프로젝트는 온라인 전시회 개최가 가능한 공간을 제공하며 현재 운영 중인 온라인 전시회를 통합하고 온/오프라인 전시회에 대한 사용자 맞춤형 안내 및 추천 서비스 제공을 목표로 한다. 전시회 안내는 해당 플랫폼을 이용하는 관람객들의 이용 정보를 분석하여 인공지능이 도출한 결과에 따라 운영 중인 온/오프라인 전시회로 연결해주는 서비스이다.

본 프로젝트는 집단 간의 문화 예술적 격차를 줄일 수 있을 것이며, 코로나 이후의 상황이 도래하여도 꾸준히 이용 가능한 서비스로 자리잡을 것이다.

1.2 프로젝트 최종 목표

본 프로젝트는 온라인 전시회 개최가 가능한 공간을 제공하며, 현재 운영 중인 온라인 전시회를 통합하고 온/오프라인 전시회에 대한 사용자 맞춤형 안내 및 추천 플랫폼 제공을 목표로 한다.

1단계 (2021년 3월 ~ 6월) : 관람객 간 커뮤니케이션이 가능한 온라인 전시회 플랫폼 구현 및 기본적인 서비스 데이터 수집

2단계 (2021년 6월 이후 데이터가 구축된 시점부터 ~ 11월) : AI 기반 사용자 맞춤형 온/오프라인 전시회 추천 서비스 구현

1.3 키워드

한글	사용자 맞춤	전시회	인공지능	비대면	추천
영문	custom	exhibition	AI	non-contact	recommendation

2. 국내·외 기술과 서비스 현황

2.1 관련 서비스 현황

1) 오픈부스(Open Booth) [1]

오픈부스(<https://www.openbooth.net/exhibition-list>)는 미팅 테크놀로지 기반의 온라인 전시, ¹⁾웨비나 복합 개최 플랫폼이다. 크게 소개관, 전시관, 박람회관으로 구분되며 1:1 상담이 가능하다. 또한 부스 제작 시 작성된 정보를 바이어에게 전하고 바이어가 참가 기업에게 의사를 표현하는 두 차례에 걸친 정보교류로 박람회 참가자들간의 커뮤니케이션을 가능하게 한다.

2) Gallery360 [2]

Gallery360(<https://www.gallery360.co.kr>)은 예술 및 IT 부문의 혁신기업이 운영하는 VR 기술 기반의 가상 갤러리이다. 작가를 선택하면 작가의 작품들을 확인할 수 있으며 VR갤러리 형식으로도 관람이 가능하다. 추천 기능은 각 테마별 작품 추천과 특징별 작품 추천으로 구분된다. 사용자의 취향에 따라 관람을 할 수 있다. 또한 AI 페인터 기능이 있어 사용자가 소유하고 있는 사진을 마치 그림처럼 표현하는 서비스를 제공하고 있다.

3) 국립현대미술관 온라인 미술관 [3]

국립현대미술관온라인 미술관(<https://www.mmca.go.kr/main.do>)은 유튜브, SNS 등으로 제공하던 영상과 음성 콘텐츠를 모아 놓은 온라인 전시 서비스를 제공한다. 검색을 통해 자신이 원하는 키워드를 입력하면 관련된 콘텐츠를 관람할 수 있다. ‘오늘, 이 작품’ 배너에서 추천 작품 확인이 가능하다는 점에서 본 프로젝트와 기능이 유사하다고 할 수 있으나 인공지능이 추천해주는 것이 아닌 특정 날짜에 한정된 작품을 추천한다는 점에서 차이가 있다.

4) Vavafair [4]

Vavafair(<https://www.vavafair.com/home/>)은 4일간 전시 행사 후 참가한 기업 정보를 모아 보여주는 온라인 박람회 플랫폼을 제공한다. 개최예정인 전시 참가 기업 제품의 미리보기 서비스와 지난 전시회 나중에 보기 서비스를 제공해 참석하지 못한 관람객에게도 전시회를 관람할 수 있는 기회를 제공한다.

1) 웨비나: 웹(Web)과 세미나(Seminar)의 합성어로, 온라인으로 진행되는 세미나를 뜻한다.

2.2 관련 연구 현황

2.2.1 온라인 전시회 관련 연구 현황

1) 온라인 전시 행사에서 개인 맞춤형 추천 시스템 적용 사례 연구 : 산학협력 EXPO를 중심으로 [5]

온라인 전시 및 행사에 대한 몰입감을 높일 수 있는 전략으로 개인 맞춤형 인공지능 추천 알고리즘을 제안하였다. 이 논문에서는 개인화 맞춤 추천과 유사 전시물 추천 시스템을 적용하여 연구를 진행하였다. 개인화 맞춤 추천은 전시 로그를 기반으로 구현하였으며, 전시물의 텍스트를 벡터로 변환 후 코사인 유사도를 적용해 판단하여 유사 전시물 추천 시스템을 제공하였다.

2) 발전 가능성이 많은 작가들을 위한 온라인 개인 전시 플랫폼 어플리케이션 개발 연구 [6]

공감각적 감상이 어려운 온라인 전시회의 단점을 보완하기 위해 ‘몸 서비스’를 구현하였다. ‘몸 서비스’는 온라인으로 작품 전시, 홍보 등의 기능을 제공하는 서비스로 본 프로젝트와 유사하나 3D 온라인 갤러리라는 점에서 차이가 있다. 또한 사용자의 3D 캐릭터가 존재하여 1인칭, 3인칭 시점으로 변경이 가능하고 캐릭터를 컨트롤하여 전시를 감상하거나 캐릭터끼리 상호작용이 가능하도록 구현했다.

2.2.2 추천 관련 연구 현황 [7],[8]

1) 협업 필터링

협업필터링이란 기존 사용자 행동 정보를 분석하여 해당 사용자와 비슷한 성향의 사용자들이 선호했던 항목들을 추천하는 기술로, 메모리 기반 협업 필터링이라 부른다. 결과를 직관적으로 보여주며 항목의 구체적 분석이 필요없다는 장점이 있다. 비슷한 패턴의 사용자나 항목추출하는 기술이 핵심적이다. 그러나 협업 필터링은 기존 데이터를 필요로 하기 때문에 새로운 항목의 추가 혹은 평가되지 않은 항목에 대해서는 예측이 어렵다는 단점이 있다. 이를 ‘콜드 스타트(Cold Start)’라고 일컫는다. 또한, 행렬분해(Matrix Factorization), k-최근접 이웃 알고리즘(k-Nearest Neighbor algorithm) 등이 사용되어 비교적 계산량이 많다. 따라서 사용자 수가 많은 경우 효율적인 추천이 불가능하다. 마지막으로, 사용자들은 소수의 인기있는 항목에 집중하기 때문에 관심이 적은 항목들은 추천을 위한 정보를 제공하지 못해 비대칭적 쏠림 현상이 발생한다. 즉, 시스템이 관리하는 항목들이 많을수록 협업 필터링에는 한계가 발생하게 된다.

2) 콘텐츠 기반 필터링

콘텐츠기반 필터링은 항목 자체를 분석하여 추천 서비스를 제공하는 기법이다. 항목을 분석한 프로파일과 사용자의 프로파일을 매치해 선호도를 예측하는 것이다. 콘텐츠 내용 분석을 위한 아이템 분석 알고리즘이 중요한데 이를 위해 군집분석(Clustering analysis), 인공신경망(Artificialneural network), tf-idf(term frequency inverse document frequency) 등의 기술이 사용된다. 콘텐츠 기반 필터링은 항목 자체를 분석하기 때문에 새로운 상품의 정보가 추가되어도 문제가 발생하지 않는다. 즉, 협업 필터링의 콜드 스타트 문제가 해결되는 것이다. 그러나 사용자의 입력에 기반을 두기 때문에 사용자의 참여가 부족한 경우에는 적절한 추천이 불가능할 가능성이 있다. 또한, 항목의 형식에 따라 얻을 수 있는 정보가 다르기 때문에 다양한 형식의 항목 추천이 어렵다는 단점이 있다.

3) 하이브리드 추천시스템

하이브리드추천 시스템은 협업 필터링과 콘텐츠 기반의 필터링을 혼합한 방법이다. 사용자 프로필을 콘텐츠 기반 필터링으로 관리하고 항목을 분석하여 유사한 사용자를 찾는 방법은 협업 필터링으로 이용하였다. 하이브리드 추천 시스템은 협업 필터링 방법과 콘텐츠 기반 추천방법을 통합하여 두 기법의 단점을 보완할 수 있으나, 다양한 프로필과 콘텐츠를 분석하기 위해서는 많은 데이터가 요구되며, 많은 시간과 비용이 소요될 수 있다.

4) 모델 기반 협력 필터링

모델기반 협력 필터링은 ‘넷플릭스 프라이즈(Netfilx Prize)’에서 수상한 알고리즘으로, 협력 필터링에 기반하여 이를 고도화한 것이다. 이는 기존 항목 간 유사성 비교에서 벗어나 자료 안의 패턴을 이용하는 기법이다. 이 때문에 내재된 패턴 분석 기술이 핵심이며, LDA(Latent Dirichlet Allocation), 베이지안 네트워크(Bayesian Network)등의 알고리즘이 사용된다. 잠재 모델에 기반을 둔 방법이 주목받고 있는데 잠재 모델은 특정 항목을 선호하는 이유를 알고리즘적으로 알아내는 기법이다. 이러한 점 때문에 직관적인 이유의 추천이 가능하여 추천의 신뢰성 또한 높일 수 있다.

2.3 관련 지적재산권 현황

<온라인 전시 관리>

특허명	온라인 전시회 관리 방법 및 그 서버 Method and Server for managing online exhibition		
	Int CL	G06Q 50/10(2012.01) G06Q 30/06(2012.01) H04N 13/00(2016.01)	
	출원번호	10-2016-0130004(20161007)	
	출원인	고려대학교 산학협력단	
	발명자	도락주, 백인식	
	대리인	심경식, 홍성욱	

주요 검색어는 ‘온라인 전시회’로 온라인 전시회 관리 방법 및 그 서버에 관한 특허이다. 가상 공간에 아바타가 특정 전시 부스로 진입할 경우 전시 부스에 대한 이미지 등을 관람객의 아바타에 대응되는 디바이스에게 제공하고 동시에 해당 관람객의 정보를 수집하여 전시 부스에 대응되는 제 2 사용자에게 정보를 제공한다.

2.4 본 프로젝트의 차별성

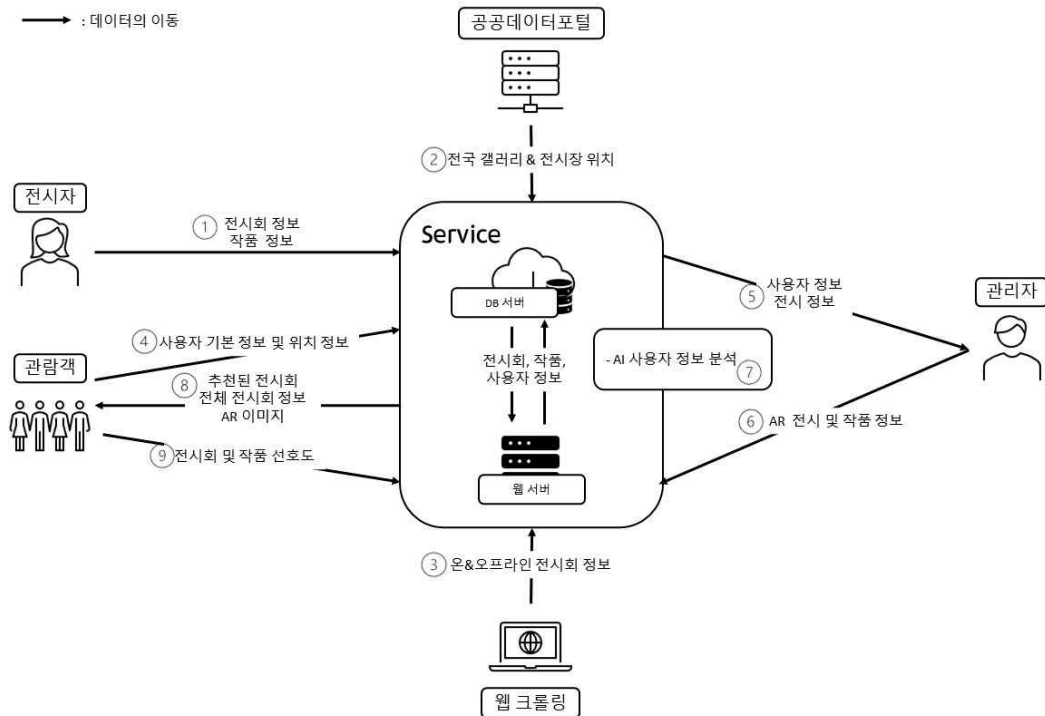
비교항목	ARTA	오픈부스	Gallery 360	국립현대 미술관 온라인 미술관	Vavafair
인공지능을 이용한 사용자 맞춤형 추천 서비스를 제공하는가?	O	X	O	X	X
커뮤니케이션 기능(댓글, 쪽지 등)을 제공하는가?	O	X	X	X	X

[표 1]. 온라인 전시회 플랫폼 특징 비교

본 프로젝트는 딥택트 시대를 반영한 사용자 맞춤형 온라인 전시회 플랫폼을 구현하며 그 결과물을 ‘ARTA’라 칭한다. 타 플랫폼에서 제공하는 추천 기능과 ARTA의 추천 기능은 유사하지만, 추천 방식에서 차이가 발생한다. 타 플랫폼에서는 사용자의 선호도와 상관없이 추천 작품들 속에서 사용자가 원하는 전시회를 관람하는 방식이라면, ARTA는 사용자의 선호도를 분석한 결과를 토대로 사용자 맞춤형 전시회와 연결해주는 서비스를 제공한다는 점에서 차이점이 있다.

3. 프로젝트 설계

3.1 서비스 개념도



해당 서비스에 전시 권한을 가진 전시자가 전시회 정보와 작품의 정보를 등록함으로써 전시회를 개최할 수 있다. 공공데이터 포털에서 획득한 전시회 정보와 타 웹서비스에서 추출한 온&오프라인 전시회 정보, 그리고 전시자가 제공한 정보들을 통합하여 웹 전시회를 제공한다. 제공되는 전시회의 목록은 관람객들의 사용자 기본 정보와 위치 정보를 분석하여 사용자에게 맞춘 전시회를 1차적으로 분석하여 제공된다. 이후 전시회 관람을 통해 사용자가 어느 작품, 전시회를 더 선호하는지를 분석하여 2차 AI 추천 결과를 제공한다.

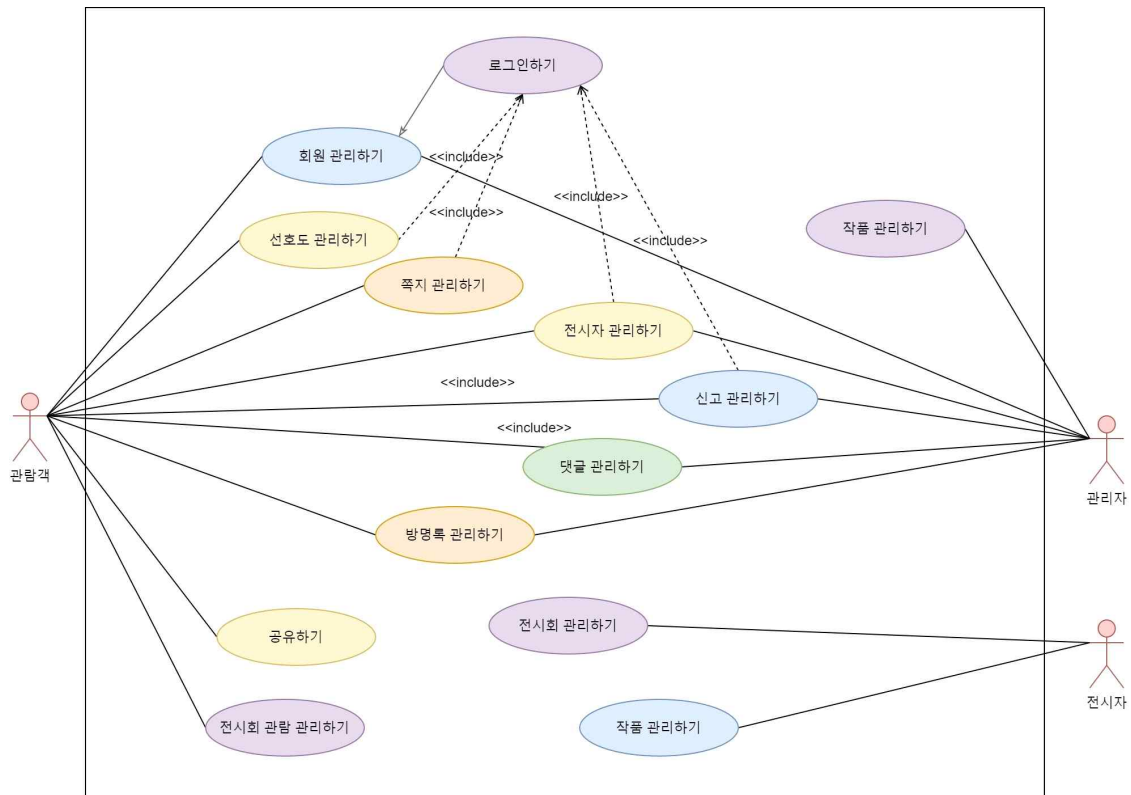
3.2 주요기능

3.2.1 주요 기능 목록 (회색 음영은 구현 후순위이다.)

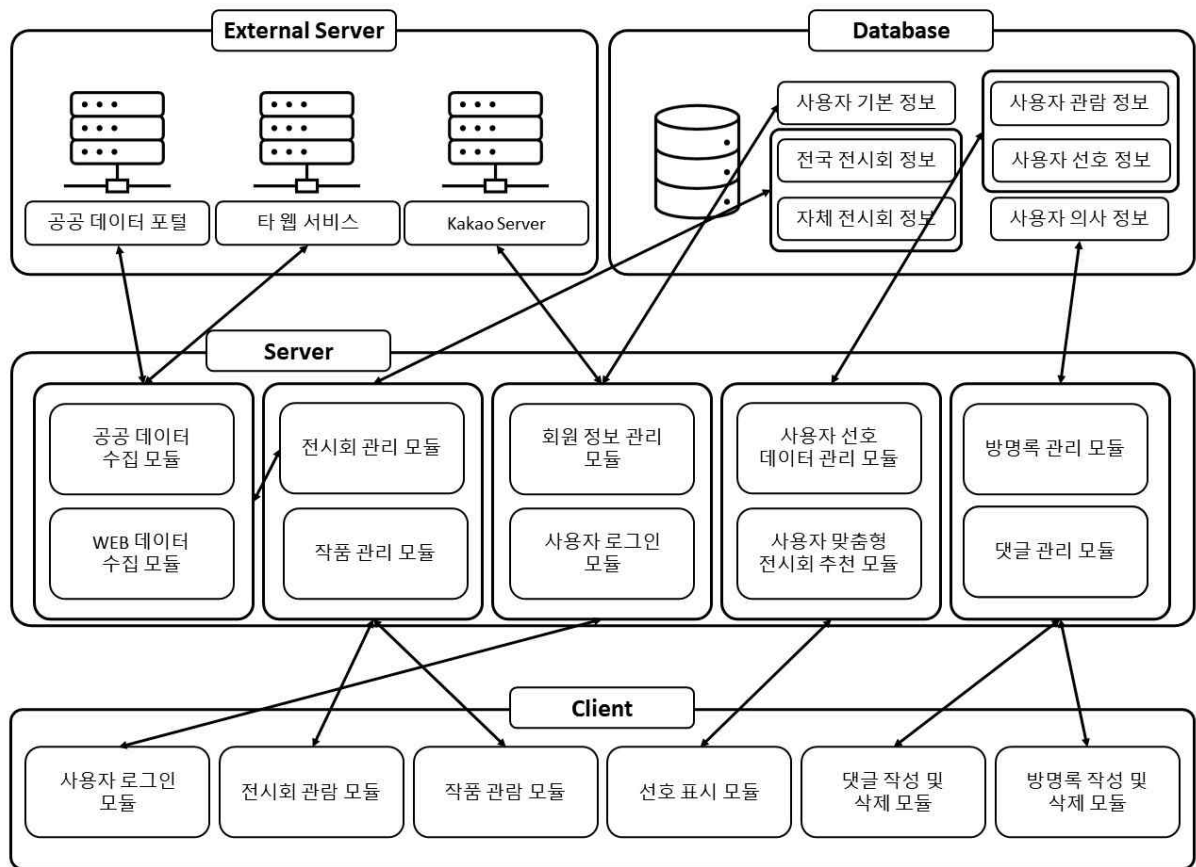
1단계 기능	일반 사용자 (관람객)	1) 회원 관리 (회원 등록/조회/수정/삭제, 아이디/비밀번호 찾기, 로그인/로그아웃) 2) 사용자 선호도 관리 (전시회, 작품 선호 등록/조회/삭제) 3) 전시회 관람 관리 (전시회 조회/검색, 작품 조회) 4) 방명록/댓글 관리 (방명록 등록/조회/삭제, 댓글 등록/조회/검색/삭제) 5) 공유하기 6) 쪽지 관리 (쪽지 등록/조회/삭제) 7) 신고관리 (신고 등록/조회)
	전시자	1) 전시회 관리 (전시회 등록/조회/수정/삭제) 2) 작품 관리 (작품 등록/조회/수정/삭제)
	관리자	1) 전시자 관리 (전시자 승인/조회/거부) 2) 작품 관리 (작품 수정/삭제) 3) 방명록/댓글 관리 (방명록/댓글 삭제) 4) 신고 관리 (신고 조회/처리) 5) 회원관리 (회원 조회/수정/삭제)
2단계 기능	공공데이터 포털의 전시회 정보를 이용한 사용자의 위치 분석 사용자의 선호도, 댓글, 후기, (가입 정보) 등의 이용 데이터 이용한 분석 사용자 정보기반 (이용 내역, 위치정보) 온/오프라인 전시회 추천	

- 해당 기능에 대한 추가 설명은 첨부 B. 프로젝트 산출물의 요구사항 목록 확인

3.2.2. Use-Case Diagram



3.3 서비스 구조도



본 서비스는 ‘Client’ 단과 ‘Sever’ 단으로 구분되며 ‘Server’ 는 ‘External Server’ 인 ‘공공 데이터 포털’ 과 ‘Kakao Server’ 그리고 ‘타 웹 서비스’ 에서 필요한 정보를 획득한다. 해당 서비스를 운영하면서 발생한 데이터와 필요한 데이터는 ‘Database’ 에 ‘사용자 기본 정보’, ‘전국 전시회 정보’, ‘자체 전시회 정보’, ‘사용자 관람 정보’, ‘사용자 선호 정보’, ‘사용자 의사 정보’로 분류하여 관리된다.

Server의 ‘공공 데이터 수집 모듈’ 과 ‘WEB 데이터 수집 모듈’ 로 외부 전시회 정보를 획득하고 해당 정보들을 DATABASE의 ‘전국 전시회 정보’에 저장한다. 그리고 ‘전시회 관리 모듈’ 과 ‘작품 관리 모듈’ 은 본 서비스에서 자체적으로 운영하는 온라인 전시회를 관리하며, ‘자체 전시회 정보’에 저장된다. ‘회원 정보 관리 모듈’ 과 ‘사용자 로그인 모듈’ 은 Database의 ‘사용자 기본 정보’를 관리하며, 사용자의 로그인을 담당한다.

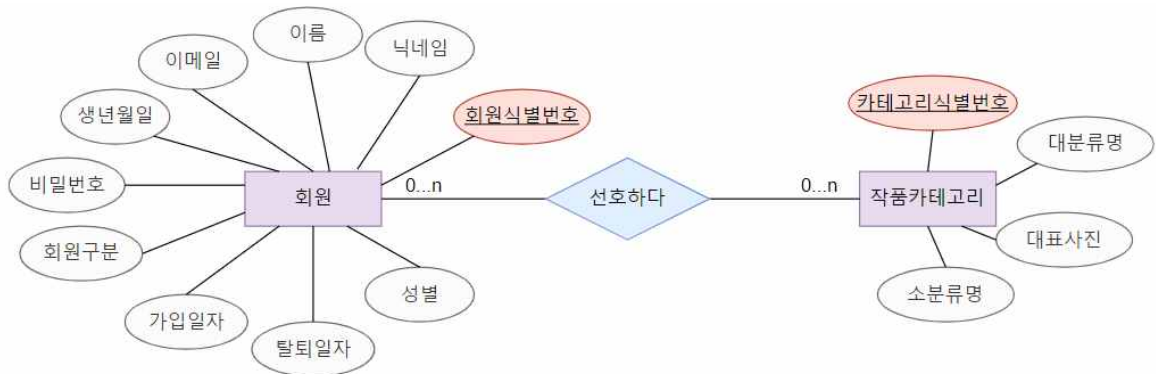
Server의 ‘방명록 관리 모듈’ 과 ‘댓글 관리 모듈’ 은 본 서비스를 이용하면서 관람객들이 등록한 댓글과 방명록을 관리하며 이는 사용자의 선호도와 밀접한 관련이 있다. 그리고 ‘사용자 선호 데이터 관리 모듈’ 과 ‘사용자 맞춤형 전시회 추천 모듈’ 은 방명록과 댓글 이외의 사용자가 선호를 표현한 개체에 대한 선호를 관리하는 모듈이다.

사용자의 선호와 관련되어 있는 정보인 ‘사용자 관람 정보’, ‘사용자 선호 정보’, ‘사용자 의사 정보’를 정제하여 최종적으로 사용자의 취향과 유사한 전시회를 Client 단의 ‘전시회 관람 모듈’, ‘작품 관람 모듈’을 통해 제공한다.

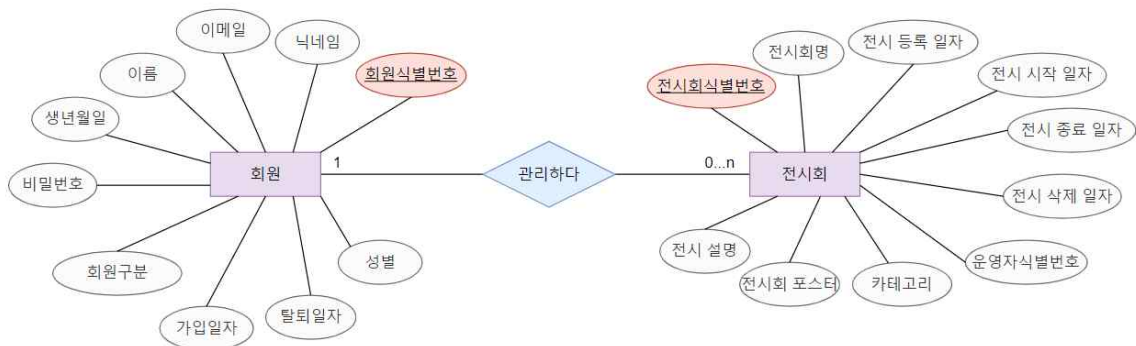
3.4 데이터베이스 설계

3.4.1) 개념 스키마

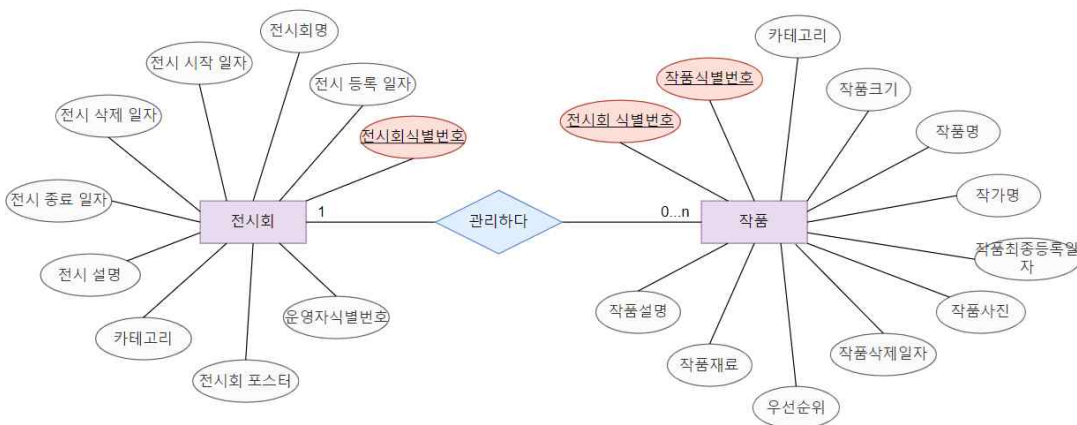
Relation 1) 회원이 작품의 카테고리를 선호하다.



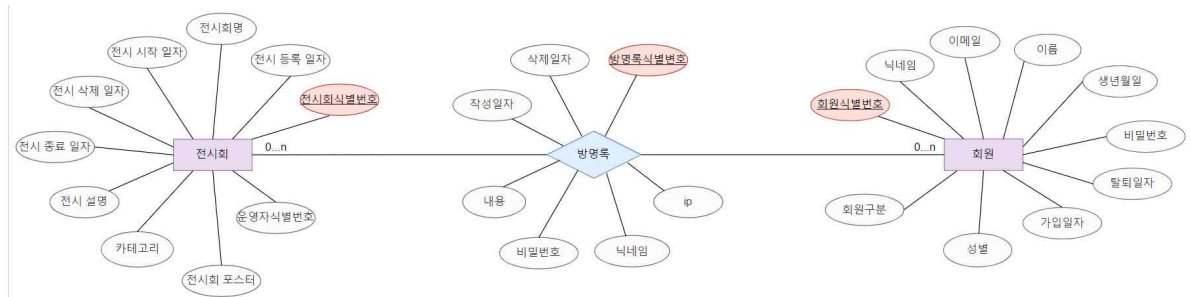
Relation 2) 회원이 전시회를 관리하다.



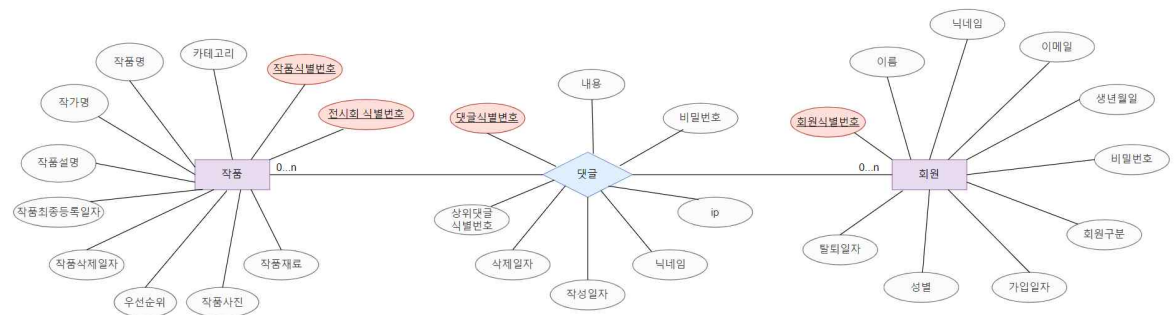
Relation 3) 전시회에 작품을 관리하다



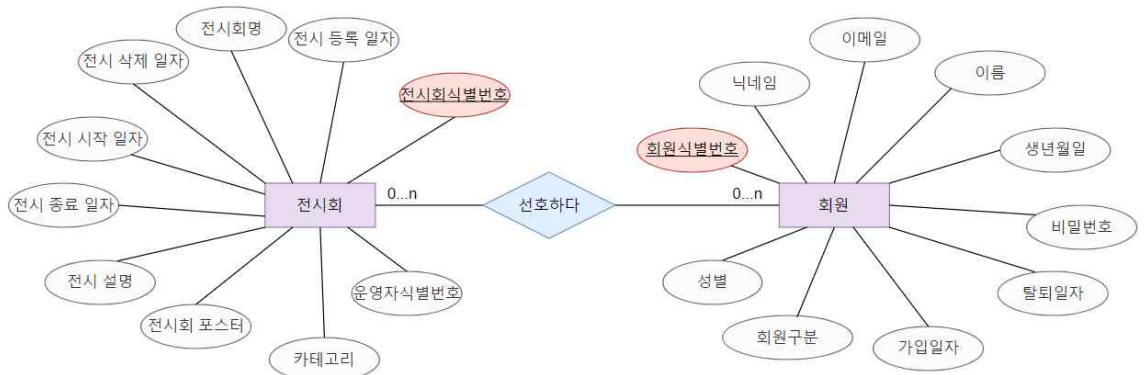
Relation 4) 회원이 전시회에 방명록을 작성하다.



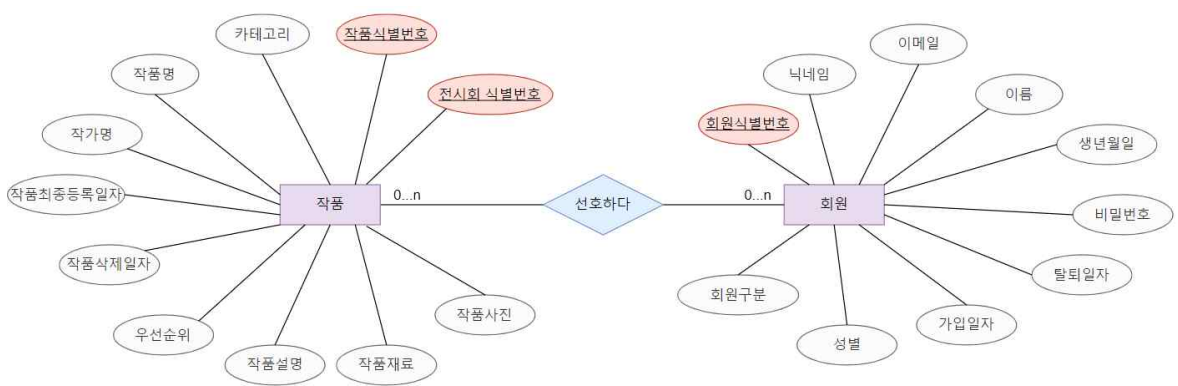
Relation 5) 회원이 작품에 댓글을 작성하다.



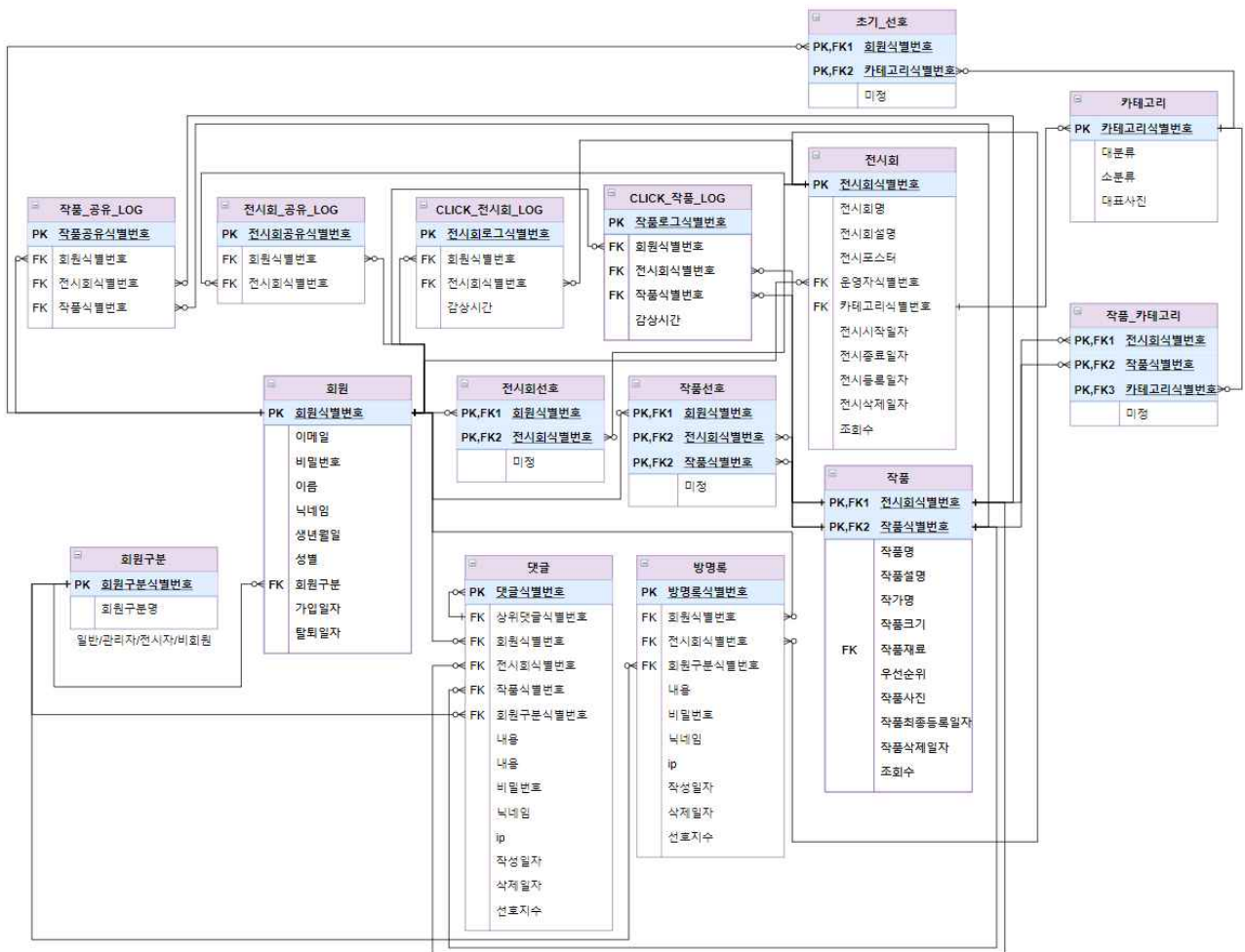
Relation 6) 회원이 전시회를 선호하다.



Relation 7) 회원이 작품을 선호하다.



3.4.2) 논리 스키마



3.4.3) 테이블 명세서

테이블이름	열이름	데이터 형식	Null 유무	기본키	외래키	FK 테이블 이름	FK 열이름
user	user_id	INT	not null	O			
	user_name	VARCHAR	not null				
	user_password	VARCHAR	not null				
	user_email	VARCHAR	not null				
	is_superuser	BOOL	not null				
	is_staff	BOOL	not null				
	is_active	BOOL	not null				

user table : 사용자 정보를 저장하는 테이블

테이블이름	열이름	데이터 형식	Null 유무	기본키	외래키	FK 테이블 이름	FK 열이름
category	category_id	INT	not null	O			
	category_BigName	VARCHAR	not null				
	category_SmallName	VARCHAR	null				
	category_Image	VARCHAR	null				
	category_slug	VARCHAR	not null				

category table : 전시회와 작품의 카테고리 정보를 저장하는 테이블

테이블이름	열이름	데이터 형식	Null 유무	기본키	외래키	FK 테이블 이름	FK 열이름
material	material_id	INT	not null	O			
	material_name	VARCHAR	not null				
	material_slug	VARCHAR	not null				

material table : 작품의 재료 정보를 저장하는 테이블

테이블이름	열이름	데이터 형식	Null 유무	기본키	외래키	FK 테이블 이름	FK 열이름
exhibition	exhibition_id	INT	not null	O			
	exhibition_name	VARCHAR	not null				
	exhibition_explain	TEXT	not null				
	exhibition_poster	VARCHAR	not null				
	exhibition_create_at	DATETIME	not null				
	exhibition_remove_at	DATETIME	null				
	exhibition_start_at	DATETIME	not null				
	exhibition_end_at	DATETIME	null				
	exhibition_click_count	INT	null				
	user_id	INT	not null		O	user	user_id
	category_id	INT	not null		O	category	category_id

exhibition table : 전시회 정보를 저장하는 테이블

테이블이름	열이름	데이터 형식	Null 유무	기본키	외래키	FK 테이블 이름	FK 열이름
piece	piece_id	INT	not null	O			
	piece_name	VARCHAR	not null				
	piece_explain	TEXT	not null				
	piece_artist	VARCHAR	null				
	piece_size	VARCHAR	null				
	piece_image	VARCHAR	not null				
	piece_order	INT	not null				
	piece_create_at	DATETIME	not null				
	piece_remove_at	DATETIME	null				
	piece_click_count	INT	null				
	material_id	INT	not null		O	material	material_id
	category_id	INT	not null		O	category	category_id

piece table : 작품 정보를 저장하는 테이블

테이블이름	열이름	데이터 형식	Null 유무	기본키	외래키	FK 테이블 이름	FK 열이름
guestbook	guestbook_id	INT	not null	O			
	user_id	INT	not null		O	user	user_id
	exhibition_id	INT	not null		O	exhibition	exhibition_id
	guestbook_content	TEXT	not null				
	guestbook_create_at	DATETIME	not null				
	guestbook_remove_at	DATETIME	null				

guestBook table : 방명록 정보를 저장하는 테이블

테이블이름	열이름	데이터 형식	Null 유무	기본키	외래키	FK 테이블 이름	FK 열이름
-------	-----	--------	---------	-----	-----	-----------	--------

comment	comment_id	INT	not null	O			
	user_id	INT	not null		O	user	user_id
	piece_id	INT	not null		O	piece	piece_id
	comment_content	TEXT	not null				
	comment_create_at	DATETIME	not null				
	comment_remove_at	DATETIME	null				

comment table : 댓글 정보를 저장하는 테이블

테이블이름	열이름	데이터 형식	Null 유무	기본키	외래키	FK 테이블 이름	FK 열이름
InitialLike	initial_id	INT	not null	O			
	category_id	INT	not null		O	category	category_id
	user_id	INT	not null		O	user	user_id

InitialLike table : 초기 선호 정보를 저장하는 테이블

테이블이름	열이름	데이터 형식	Null 유무	기본키	외래키	FK 테이블 이름	FK 열이름
exhibitionLike	exhibitionlike_id	INT	not null	O			
	exhibition_id	INT	not null		O	exhibition	exhibition_id
	user_id	INT	not null		O	user	user_id

exhibitionLike table : 전시회 선호 정보를 저장하는 테이블

테이블이름	열이름	데이터 형식	Null 유무	기본키	외래키	FK 테이블 이름	FK 열이름
pieceLike	piecelike_id	INT	not null	O			
	piece_id	INT	not null		O	piece	piece_id
	user_id	INT	not null		O	user	user_id

pieceLike table : 작품 정보를 저장하는 테이블

테이블이름	열이름	데이터 형식	Null 유무	기본키	외래키	FK 테이블 이름	FK 열이름
exhibitionClick	exhibitionclick_id	INT	not null	O			
	exhibition_id	INT	not null		O	exhibition	exhibition_id
	user_id	INT	not null		O	user	user_id

exhibitionClick table : 전시회를 클릭한 정보를 저장하는 테이블

테이블이름	열이름	데이터 형식	Null 유무	기본키	외래키	FK 테이블 이름	FK 열이름
pieceClick	piececlick_id	INT	not null	O			
	piece_id	INT	not null		O	piece	piece_id
	user_id	INT	not null		O	user	user_id

pieceClick table : 작품을 클릭한 정보를 저장하는 테이블

3.5 사용자 인터페이스 디자인

3.5.1) 모바일 전용 사용자 인터페이스

1. 메인페이지	2. 네비게이션 바	3. 회원가입	4. 사용자 초기 선호 선택
			
5. 서비스 소개	6. 로그인	7. 마이페이지	8. 검색
			

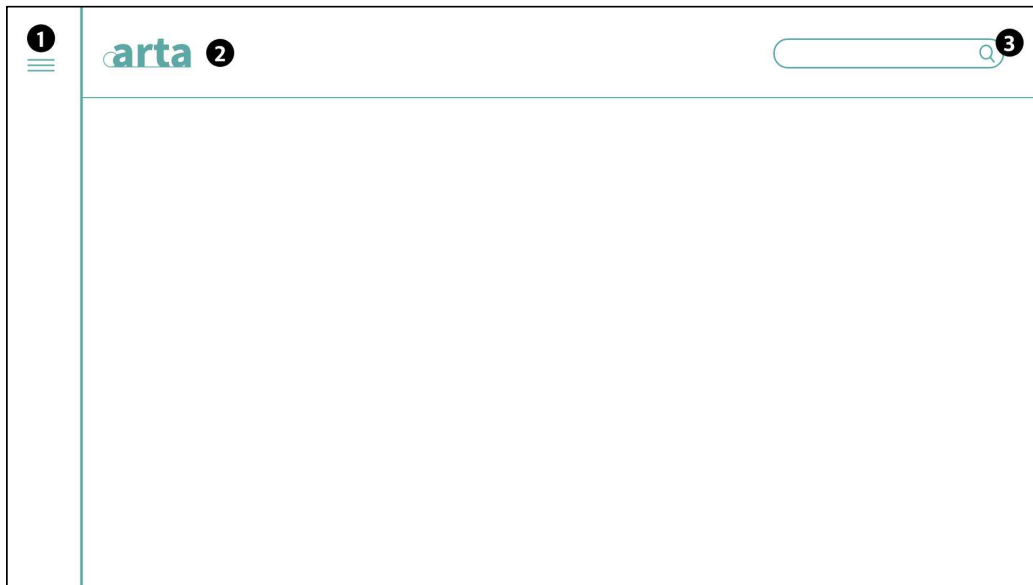
<p>9. 전시회 선호 리스트</p> 	<p>10. 작품 선호 리스트</p> 	<p>11. 전시회 목록</p> 	<p>12. 전시회 보기</p> 
<p>13. 작품 보기</p> 	<p>14. 전시회 등록</p> 	<p>15. 전시회 수정</p> 	<p>16. 작품등록</p> 

3.5.2) 웹 전용 사용자 인터페이스

3.6 UX/UI 설계에서 확인.

3.6 UX/UI 설계서

3.6.1) 메인페이지



#	기능	
1	네비게이션바	- 네비게이션 버튼을 누르면 메뉴바가 펼쳐진다
2	로고	- 로고를 누르면 메인 페이지로 이동한다
3	검색	- 돋보기 버튼을 누르면 검색 페이지로 이동한다

3.6.2) 메인페이지_네비게이션바



#	기능	
1	메뉴 닫힘 버튼	- 버튼을 누르면 메뉴가 닫힌다
2	소개 메뉴	- 아르타 소개 페이지로 이동한다
3	전시회 보기 메뉴	- 전시회 목록 페이지로 이동한다
4	로그인 메뉴	- 로그인 페이지로 이동한다
5	회원가입 메뉴	- 회원가입 페이지로 이동한다

3.6.3) 로그인 페이지

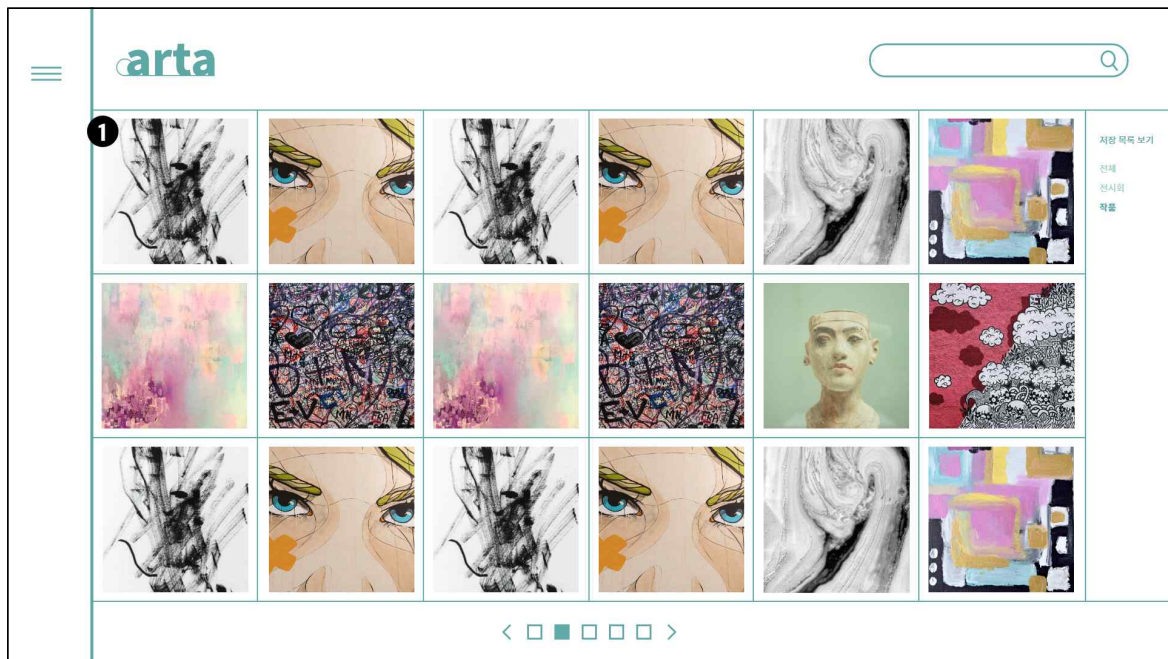
#	기능	
1	본 페이지가 로그인 페이지임을 알린다	
2	이메일	- 일치하는 아이디를 가진 회원이 없거나 해당 아이디에 맞는 비밀번호가 아닐 때 “아이디 또는 비밀번호가 잘못되었습니다.” 라는 문구 출력, 로그인 화면 (아이디, 비밀번호 입력화면)으로 돌아감.
3	비밀번호	- 존재하는 아이디에 올바른 비밀번호를 입력 받았을 때, 메인 페이지(or 로그인 전 페이지)로 돌아감.
4	- 로그인 버튼을 누르면 로그인이 완료된다	
5	- 비밀번호 찾기 버튼을 누르면 비밀번호 찾기 페이지로 이동한다	

+ 카카오 로그인 API 연동 예정

3.6.4) 전시회 선호 리스트 페이지

#	기능
1	한 페이지에 사용자가 좋아요를 누른 전시회가 3개씩 보여진다
2	전시회 운영 단체와 전시회 이름이 순서대로 나열된다

3.6.5) 작품 선호 리스트 페이지



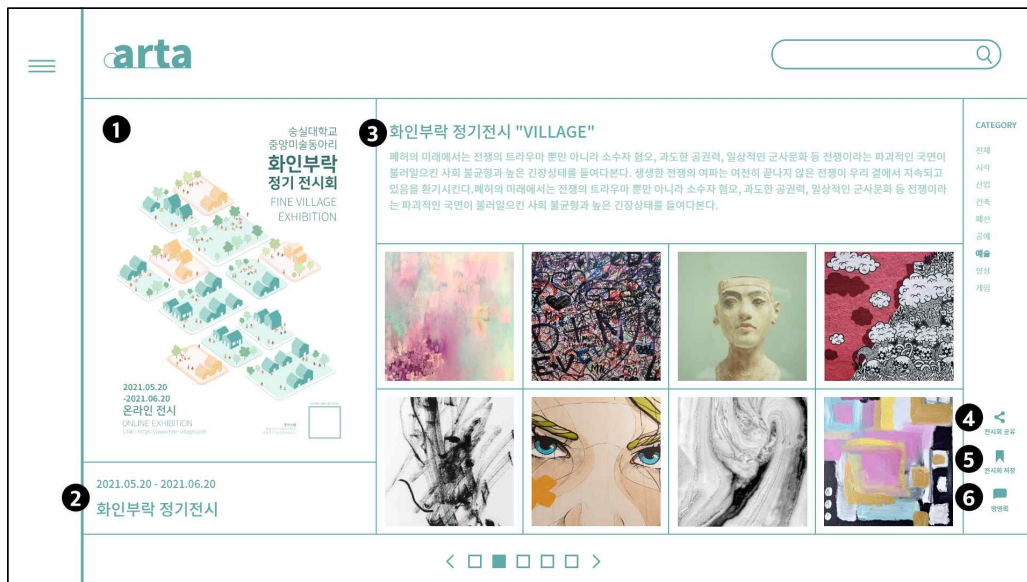
#	기능
1	사용자가 좋아요를 누른 작품이 한 페이지 당 18개씩 보여진다

3.6.6) 전시회 목록 페이지



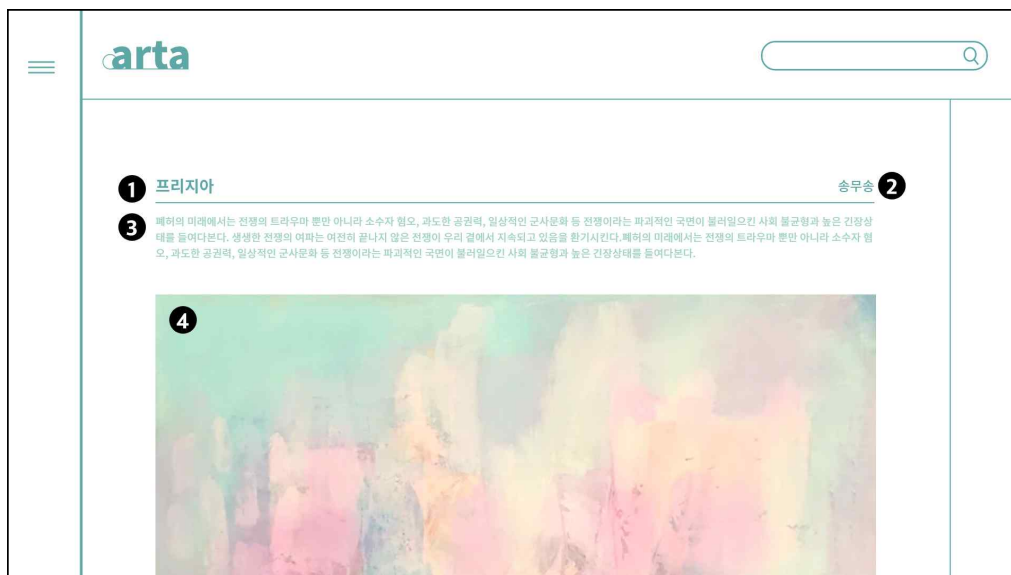
#	기능
1	한 페이지 당 8개의 전시회가 보여진다.
2	전시회 운영 단체와 전시회 이름이 순서대로 나열된다
3	카테고리 <ul style="list-style-type: none"> - 회화, 조소, 디자인, 건축, 패션, 공예, 사진, 기타로 구성된 카테고리가 있다. - 이름 오름차순으로 정렬, 다중 선택 가능

3.6.7) 전시회 보기 페이지



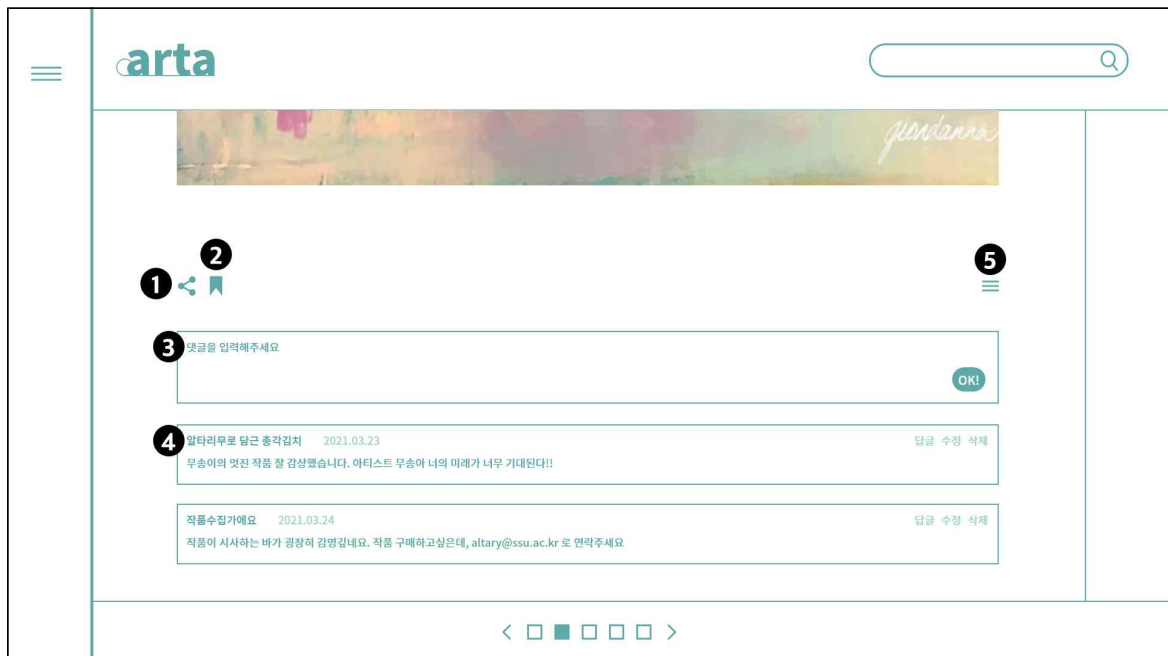
#	기능
1	전시회 메인 포스터
2	전시회 운영기간과 전시회 운영 단체 이름
3	전시회 이름과 함께 간략한 소개가 있다
4	전시회 페이지를 공유할 수 있는 버튼으로, 버튼을 누르면 페이지 URL이 복사된다
5	전시회를 저장할 수 있는 버튼
6	방명록 버튼으로, 누르면 방명록 페이지로 이동한다

3.6.8.1) 작품 보기 페이지



#	기능
1	작품 이름
2	작가 이름
3	작품에 대한 간략한 소개글
4	작품

3.6.8.2) 작품 보기 페이지



#	기능	
1	작품 공유 버튼	작품 공유 버튼을 누르면 해당 작품 페이지 URL이 복사된다
2	작품 저장하기 버튼	
3	댓글 입력 박스	
4	댓글 보기	<ul style="list-style-type: none"> - 목록에 보여지는 항목의 개수: 10개 - 페이지 처리: 맨 앞으로, 5페이지 단위, 맨 뒤로 - 항목의 요소: 닉네임, 작성 일자, 작성 내용 - 항목 나열 순서: 작성 일자 내림차순

3.6.9) 전시회 관리 페이지



#	기능	
1	edit 버튼	<ul style="list-style-type: none"> - 버튼을 누르면 전시회를 등록, 수정할 수 있는 페이지로 이동한다 - 해당 버튼은 일반 회원이 아닌 전시자 회원에게만 보여진다

3.6.10) 전시회 등록 페이지

1 전시회 등록

2 전시회명 필수입력입니다

3 전시회 소개 필수입력입니다

4 운영기간 2021.03.12 - 2021.03.15

5 전시회 포스터 OK!

6 카테고리 ☐ 시각 ☐ 산업 ☐ 건축 ☐ 패션 ☐ 공예 ☐ 예술 ☐ 영상 ☐ 게임 ☐ 기타

7 취소 8 등록

전시회 등록
작품 등록
전시회 관리
작품 관리

#	기능
1	본 페이지가 전시회 등록 페이지임을 알린다
2	전시회명 -1글자 이상 30자 이내 텍스트
3	전시회 소개 -1자 이상 300자 이내 텍스트
4	운영 기간 -달력(datepicker 이용) 구현
5	전시회 포스터 -전시회 당 이미지 1개만 첨부 가능
6	카테고리 - 전시회에 해당하는 카테고리 선택하도록
7	취소 버튼을 누르면 전시회 등록이 취소된다
8	등록 버튼을 누르면 전시회 등록이 완료된다

3.6.11) 전시회 수정 페이지

1 전시회 수정

2 화인부락 정가전시

3 전시회명 필수입력입니다

4 전시회 소개 필수입력입니다

5 운영기간 2021.03.12 - 2021.03.15

6 카테고리 ☐ 시각 ☐ 산업 ☐ 건축 ☐ 패션 ☐ 공예 ☐ 예술 ☐ 영상 ☐ 게임 ☐ 기타

7 취소 8 전시회 삭제 9 수정완료

전시회 등록
작품 등록
전시회 관리
작품 관리

#	기능
1	전시회 포스터
2	전시회 단체명과 전시회 명
3	본 페이지가 전시회 수정 페이지임을 알린다
4	전시회 소개 수정
5	전시회 운영기간 수정
6	전시회 카테고리 수정
7	취소 버튼을 누르면 전시회 등록이 취소된다
8	전시회 삭제 버튼을 누르면 전시회가 삭제된다
9	수정완료 버튼을 누르면 전시회 수정이 완료된다

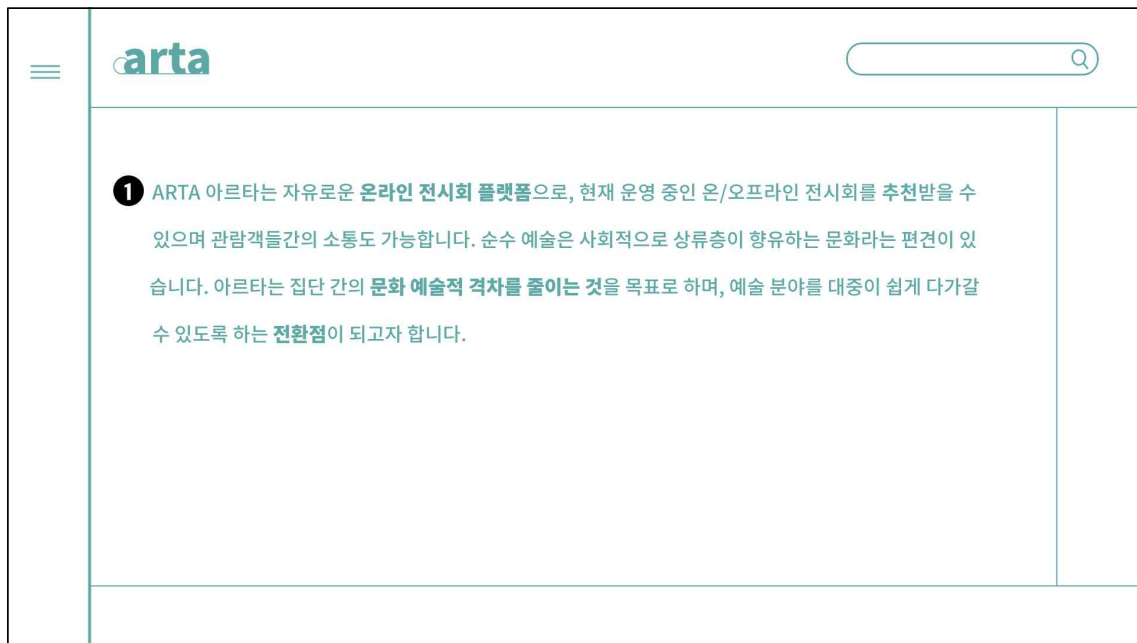
3.6.12) 작품 등록 페이지

#	기능
1	본 페이지가 작품 등록 페이지임을 알린다
2	작품명 -1글자 이상 30자 이내 텍스트
3	작품 소개 -1자 이상 300자 이내 텍스트
4	작가명 -1자 이상 300자 이내 텍스트
5	우선순위 -우선순위를 활용하여 작품 순서 지정
6	화풍 카테고리 - 체크박스
7	재료 카테고리 - 체크박스
8	취소 버튼을 누르면 작품 등록이 취소된다
9	등록 버튼을 누르면 작품 등록이 완료된다

3.6.13) 작품 수정 페이지

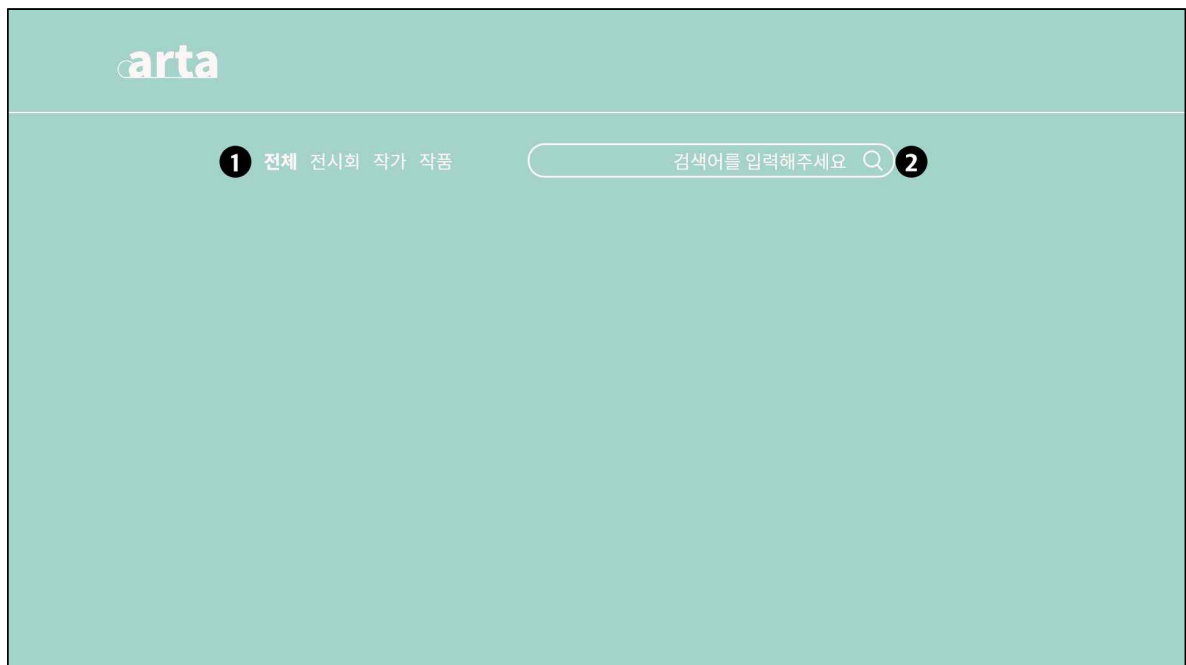
#	기능
1	작품 이미지
2	작가이름과 작품명
3	본 페이지가 작품 수정 페이지임을 알린다
4	전시회 소개 수정
5	우선순위 - 우선순위를 활용하여 작품 순서 지정
6	취소 버튼을 누르면 작품 수정이 취소된다
7	수정완료 버튼을 누르면 작품 수정이 완료된다

3.6.14) 소개 페이지



#	기능
1	아르타 소개 글

3.6.15) 검색 페이지



#	기능
1	검색 카테고리를 선택하여 결과물을 볼 수 있도록 한다
2	검색 글자 수 제한은 없이 검색의 결과를 조회하는 버튼.

3.7 REST API 설계서 (구현 순위가 낮은 경우 추후 기재 예정)

End Point	/	Method	HTTP GET	대상	일반 사용자
Usage	해당 서비스에 처음 방문했을 때 보이는 화면 출력한다.				
Request Body					
Response Body					
<pre>{ "message" : "메인 페이지 출력 성공" }</pre>					

End Point	/about	Method	HTTP GET	대상	일반 사용자
Usage	해당 서비스에 대한 소개 페이지 출력한다.				
Request Body					
Response Body					
<pre>{ "message" : "소개 페이지 출력 성공", }</pre>					

End Point	/exhibition	Method	HTTP GET	대상	일반 사용자
Usage	해당 서비스에서 제공하고 있는 전시회 목록을 출력한다.				
Request Body					
<pre>{ "page" : "1", }</pre>					
Response Body					
<pre>{ "message" : "전시회 목록 출력 성공", "exhibitions" : [{ "poster": "image1.jpg", "title": "전시회 이름" }, { "poster": "image2.jpg", "title": "전시회 이름" }, ...] }</pre>					

End Point	/exhibition/<int:pk>	Method	HTTP GET	대상	일반 사용자
Usage	해당 전시회에 대한 전시회 페이지 출력				
Request Body					
{ “page” : “1”, }					
Response Body					
{ “message” : “전시회 관람 정보 출력 성공”, “exhibition” : { “poster” : “전시회포스터.jpg”, “name” : “전시회 이름”, “explain” : “전시회 설명”, “start_at” : “전시회 시작 날짜”, “end_at” : “전시회 종료 날짜”, “count” : “조회수”, “category” : “카테고리”, }, “pieces” : [{“image”:“picture.jpg”, “title”:“작품 명”, “artist”:“작가명”}, {“image”:“picture.jpg”, “title”:“작품 명”, “artist”:“작가명”}, ...] }					

End Point	/exhibition/piece/<int:pk>	Method	HTTP GET	대상	일반 사용자
Usage	해당 작품에 대한 상세 정보를 출력한다.				
Request Body					
Response Body					
<pre>{ "message" : "작품 정보 출력 성공", "pieces" : { "image": "picture.jpg", "name": "작품명", "artist": "작가명", "explain": "작품 내용", "size" : "작품 크기", "create_at" : "게시 날짜", "count" : "조회수", "category" : "카테고리명", "material" : "작품 재료", ...} }</pre>					

End Point	/exhibition/guestbook/<int:pk>	Method	HTTP POST	대상	일반 사용자
Usage	해당 전시회에 대한 방명록을 등록한다.				
Request Body					
<pre>{ "content" : "방명록 내용", }</pre>					
Response Body					
<pre>{ "message" : "방명록 등록 성공", }</pre>					

End Point	/exhibition/guestbook/<int:pk>	Method	HTTP DELETE	대상	일반 사용자
Usage	해당 전시회에 대한 방명록을 삭제한다.				
Request Body					
Response Body					
<pre>{ "message" : "방명록 삭제 성공", }</pre>					

End Point	/exhibition/piece/comment/<int:pk>	Method	HTTP POST	대상	일반 사용자
Usage	해당 작품에 대한 댓글을 등록한다.				
Request Body					
<pre>{ "content" : "댓글 내용", }</pre>					
Response Body					
<pre>{ "message" : "댓글 등록 성공", }</pre>					

End Point	/exhibition/piece/comment/<int:pk>	Method	HTTP DELETE	대상	일반 사용자
Usage	해당 작품에 대한 댓글을 삭제한다.				
Request Body					
Response Body					
<pre>{ "message" : "댓글 삭제 성공", }</pre>					

End Point	/exhibition/like/<int:pk>	Method	HTTP POST	대상	일반 사용자
Usage	해당 전시회에 대한 선호를 표시한다.				
Request Body					
Response Body					
<pre>{ "message" : "전시회 좋아요 성공", }</pre>					

End Point	/exhibition/like/<int:pk>	Method	HTTP DELETE	대상	일반 사용자
Usage	해당 전시회에 대한 선호를 취소한다.				
Request Body					
Response Body					
{ “message” : “전시회 좋아요 취소 성공”, }					

End Point	/exhibition/piece/like/<int:pk>	Method	HTTP POST	대상	일반 사용자
Usage	해당 작품에 대한 선호를 표시한다.				
Request Body					
Response Body					
<pre>{ "message" : "작품 좋아요 성공", }</pre>					

End Point	/exhibition/piece/like/<int:pk>	Method	HTTP DELETE	대상	일반 사용자
Usage	해당 작품에 대한 선호를 취소한다.				
Request Body					
Response Body					
<pre>{ "message" : "작품 좋아요 취소 성공", }</pre>					

End Point	/manage	Method	HTTP GET	대상	전시자
Usage	전시 권한을 부여받은 전시자들이 전시회를 관리하기 위해 확인하는 메인 페이지 출력하기				
Request Body					
Response Body					
<pre>{ "message" : "전시자 메인 페이지 출력 성공", }</pre>					

End Point	/manage/exhibition	Method	HTTP GET	대상	전시자
Usage	운영중인 전시회의 목록을 출력하기				
Request Body					
<pre>{ "page" : "1", }</pre>					
Response Body					
<pre>{ "message" : "전시회 목록 출력 성공", "exhibitions" : [{"poster":"image1.jpg", "title":"전시회 이름"}, {"poster":"image2.jpg", "title":"전시회 이름"}, ...] }</pre>					

End Point	/manage/exhibition/<int:pk>	Method	HTTP GET	대상	전시자
Usage	해당 전시회에 대한 전시회 페이지 출력				
Request Body					
{ “page” : “1”, }					
Response Body					
{ “message” : “전시회 관람 정보 출력 성공”, “exhibition” : { “poster” : “전시회포스터.jpg”, “name” : “전시회 이름”, “explain” : “전시회 설명”, “start_at” : “전시회 시작 날짜”, “end_at” : “전시회 종료 날짜”, “count” : “조회수”, “category” : “카테고리”, }, “pieces” : [{“image”:“picture.jpg”, “title”:“작품 명”, “artist”:“작가명”}, {“image”:“picture.jpg”, “title”:“작품 명”, “artist”:“작가명”}, ...] }					

End Point	/manage/exhibition/create	Method	HTTP GET	대상	전시자
Usage	해당 전시회에 대한 전시회 등록 페이지 출력하기				
Request Body					
Response Body					
<pre>{ "message" : "전시회 등록 페이지 출력 성공" }</pre>					

End Point	/manage/exhibition/create	Method	HTTP POST	대상	전시자
Usage	해당 전시회에 대한 전시회 등록하기				
Request Body					
{ “poster” : “전시회포스터.jpg”, “name” : “전시회 이름”, “explain” : “전시회 설명”, “start_at” : “전시회 시작 날짜”, “end_at” : “전시회 종료 날짜”, “count” : “조회수”, “category” : “카테고리”, }					
Response Body					
{ “message” : “전시회 등록 성공” }					

End Point	/manage/exhibition/update/<int:pk>	Method	HTTP GET	대상	전시자
Usage	해당 전시회에 대한 전시회 수정 페이지 출력				
Request Body					
Response Body					
<pre>{ "message" : "전시회 정보 출력 성공", "poster" : "전시회포스터.jpg", "name" : "전시회 이름", "explain" : "전시회 설명", "start_at" : "전시회 시작 날짜", "end_at" : "전시회 종료 날짜", "count" : "조회수", "category" : "카테고리", }</pre>					

End Point	/manage/exhibition/update/<int:pk>	Method	HTTP POST	대상	전시자
Usage	해당 전시회에 대한 전시회 정보 수정				
Request Body					
{ “poster” : “전시회포스터.jpg”, “name” : “전시회 이름”, “explain” : “전시회 설명”, “start_at” : “전시회 시작 날짜”, “end_at” : “전시회 종료 날짜”, “count” : “조회수”, “category” : “카테고리”, }					
Response Body					
{ “message” : “전시회 수정 성공” }					

End Point	/manage/exhibition/update/<int:pk>	Method	HTTP DELETE	대상	전시자
Usage	해당 전시회에 대한 전시회 정보 삭제				
Request Body					
Response Body					
<pre>{ "message" : "전시회 삭제 성공" }</pre>					

End Point	/manage/exhibition/piece/<int:pk>	Method	HTTP GET	대상	전시자
Usage	해당 작품에 대한 상세 정보 출력				
Request Body					
Response Body					
<pre>{ "message" : "작품 정보 출력 성공", "pieces" : { "image": "picture.jpg", "name": "작품명", "artist": "작가명", "explain": "작품 내용", "size" : "작품 크기", "create_at" : "게시 날짜", "count" : "조회수", "category" : "카테고리명", "material" : "작품 재료", ...} }</pre>					

End Point	/manage/exhibition/piece/create/<int:pk>	Method	HTTP GET	대상	전시자
Usage	해당 전시회에 대하여 작품을 등록하기 위한 기본 페이지 출력.				
Request Body					
Response Body					
<pre>{ "message" : "작품 등록페이지 출력 성공", }</pre>					

End Point	/manage/exhibition/piece/create/<int:pk>	Method	HTTP POST	대상	전시자
Usage	해당 전시회에 대하여 작품을 등록하기				
Request Body					
{ "image": "picture.jpg", "name": "작품명", "artist": "작가명", "explain": "작품 내용", "size" : "작품 크기", "create_at" : "게시 날짜", "count" : "조회수", "category" : "카테고리명", "material" : "작품 재료", }					
Response Body					
{ "message" : "작품 등록 성공", }					

End Point	/manage/exhibition/piece/update/<int:pk>	Method	HTTP GET	대상	전시자
Usage	해당 작품을 수정하기 위한 페이지 출력				
Request Body					
Response Body					
<pre>{ "message" : "작품 정보 출력 성공", "pieces" : { "image": "picture.jpg", "name": "작품명", "artist": "작가명", "explain": "작품 내용", "size" : "작품 크기", "create_at" : "게시 날짜", "count" : "조회수", "category" : "카테고리명", "material" : "작품 재료", ...} }</pre>					

End Point	/manage/exhibition/piece/update/<int:pk>	Method	HTTP POST	대상	전시자
Usage	해당 작품을 수정하기				
Request Body					
<pre>{ "image": "picture.jpg", "name": "작품명", "artist": "작가명", "explain": "작품 내용", "size" : "작품 크기", "create_at" : "게시 날짜", "count" : "조회수", "category": "카테고리명", "material" : "작품 재료", }</pre>					
Response Body					
<pre>{ "message" : "작품 정보 출력 성공", }</pre>					

End Point	/manage/exhibition/piece/delete/<int:pk>	Method	HTTP DELETE	대상	전시자
Usage	해당 작품을 삭제				
Request Body					
Response Body					
<pre>{ "message" : "작품 삭제 성공" }</pre>					

3.7 모듈별 기능 명세서

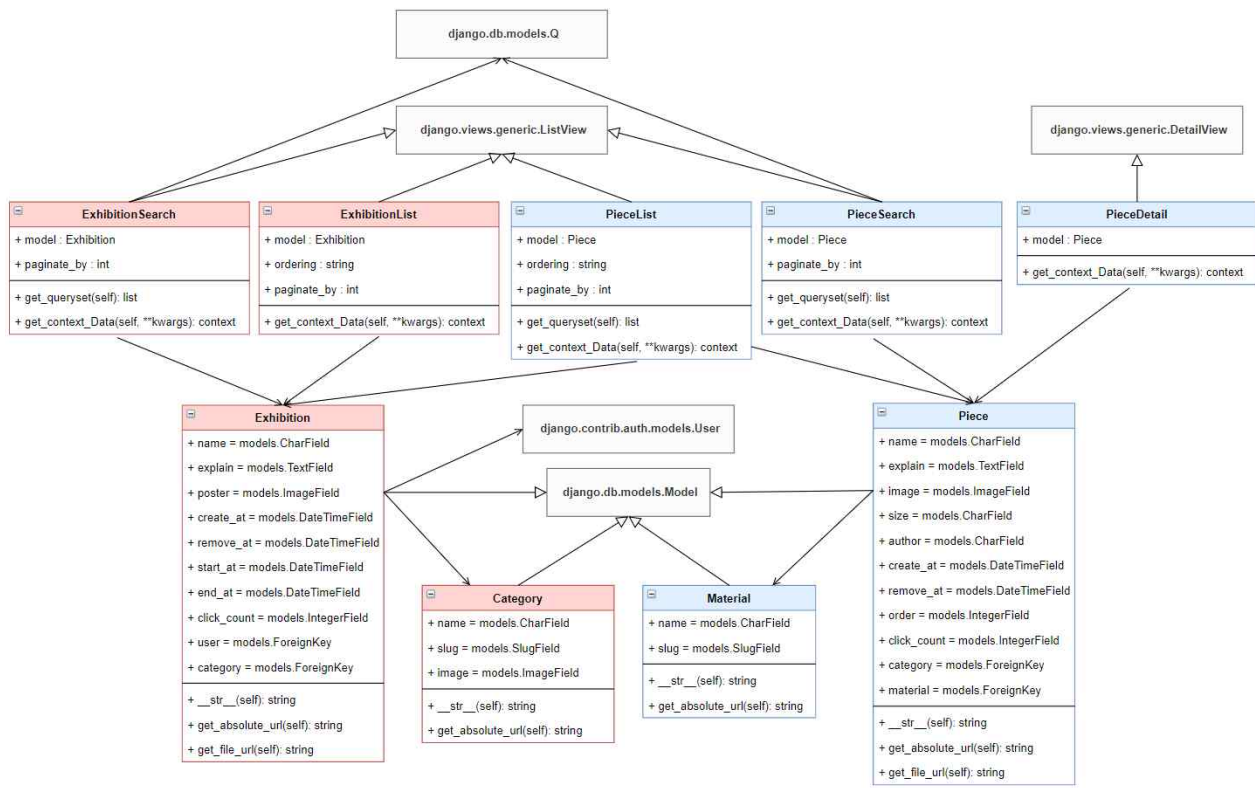
3.6.1 전시회 관람 모듈

모듈명		전시회 관람 모듈		모듈 ID		UM-001	
1. 개요							
사용자가 전시회의 목록을 확인하고, 전시회의 작품들을 감상할 수 있다.							
2. 주 사용 테이블							
테이블 명		테이블 설명			연관 테이블		
exhibition		전시회 정보 저장을 위한 테이블			user, category		
piece		작품 정보 저장을 위한 테이블			meterial, category		
3. 기능 명세							
기능명		전시회 목록보기		기능 ID			
개요		사용자는 존재하는 모든 전시회 목록을 볼 수 있다.					
상세설명		<ul style="list-style-type: none">- 사용자는 해당 플랫폼에 존재하는 모든 전시회 목록을 볼 수 있다.- 전시회 대표 이미지, 전시회 제목, 전시회 운영자, 온/오프라인 정보로 구성되어 나열된다.- 분야 안에서도 회화, 조소, 디자인, 건축, 패션, 공예, 사진, 기타로 구성된 카테고리가 있다.- 카테고리: 이름 오름차순으로 정렬, 다중 선택 가능- 사용자 정보를 기반으로 선별한 추천 전시회 우선적으로 보여준다.- 사용자 정보: 사용자 위치 정보, 좋아요 기록, 자주 본 전시회 기록 등- 비로그인인 경우나 추천 전시회가 없는 경우는 전시회 마감 일자가 빠른 순으로 정렬.- 오픈 예정 전시회는 오픈 일자가 빠른 순으로 정렬해서 보여준다.					
Web Server 담당자		송무송		Web Client 담당자		조은진	
서비스 대상		일반 사용자		중요도		상	
기능명		전시회 보기		기능 ID			
개요		사용자는 전시회를 선택하여 전시회를 관람할 수 있다.					
상세설명		<ul style="list-style-type: none">- 현재 운영 중인 온라인 전시회 혹은 운영 종료된 온라인 전시회를 선택해야 서비스가 제공.- 오프라인 전시회, 오픈 예정 전시회를 선택할 경우 전시회에 대한 정보를 제공.- 첫 화면에는 전시회 메인 포스터와 간략한 소개가 있다.- 이후에는 전시자가 구성한 대로 작품을 볼 수 있는 페이지 제공.					
Web Server 담당자		송무송		Web Client 담당자		조은진	
서비스 대상		일반 사용자		중요도		상	
기능명		작품 목록보기 (전시회 보기에 포함)		기능 ID			
개요		사용자는 해당 전시회의 그림 목록을 볼 수 있다.					
상세설명		<ul style="list-style-type: none">- 작품 목록은 카테고리(회화, 조소, 디자인, 건축, 패션, 공예, 사진, 기타)별로 나누어져 있어 원하는 카테고리에 해당하는 작품을 볼 수 있다.- 카테고리는 이름의 오름차순으로 정렬되어 있다.- 카테고리는 다중 선택이 가능하다.- 각 카테고리에는 재료에 따른 세부 카테고리가 있다.- 정렬 최우선 순위는 사용자 정보를 기반으로 한 추천이다.- 정렬 후 순위는 작품이 업데이트된 시간 순으로 정렬된다.					
Web Server 담당자		송무송		Web Client 담당자		조은진	
서비스 대상		일반 사용자		중요도		상	

기능명	작품 보기	기능 ID	
개요	사용자는 작품을 관람할 수 있다.		
상세설명	<ul style="list-style-type: none"> - 작품과 함께 간략한 설명, 댓글을 볼 수 있는 서비스이다. - 전시회 보기 서비스 혹은 작품 목록 보기 서비스에서 작품을 선택하였을 때 나타나는 페이지이다. - 작품들과 간략한 설명, 작가 이름으로 구성되어 있다. 		
Web Server 담당자	송무송	Web Client 담당자	문석희
서비스 대상	일반 사용자	중요도	상

기능명	검색하기	기능 ID	
개요	사용자는 전시회, 그림, 작가를 통합하여 원하는 것을 검색하고, 검색 결과 목록을 볼 수 있는 서비스이다.		
상세설명	<ul style="list-style-type: none"> - 검색 글자 수 제한은 없다. - 모음 또는 자음, 숫자로만 검색하는 것도 허용한다. - 검색할 때 전체, 전시회, 작품, 작가 분야를 선택할 수 있으며 다수 선택도 가능하다. - '입결한 글자와 완전히 일치하는 결과물' 선택창을 선택하면, 검색한 글자와 완전히 일치하는 검색 결과만을 얻을 수 있다. - 전시회 결과 목록은 전체, 전시회, 작품, 작가 총 4 분야로 나뉜다. - 다수 분야 선택시 분야의 우선순위는 전시회, 작품, 작가 순으로 지정된다 - 분야 중 전체를 선택한 경우, 다수 선택이 불가능하다. - 검색한 단어와 완벽히 일치하는 결과물을 우선으로 보여주고, 이후에는 단어를 포함하는 결과물을 이름의 오름차순으로 정렬한다. 		
Web Server 담당자	송무송	Web Client 담당자	문석희
서비스 대상	일반 사용자	중요도	상

4. 클래스 다이어그램



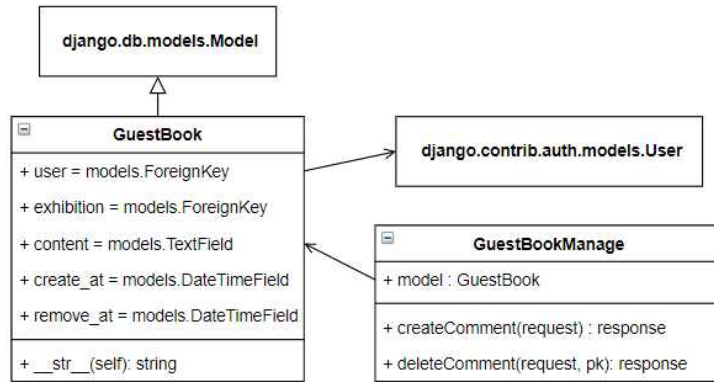
3.6.2 선호도 관리 모듈

모듈명	선호도 관리 모듈	모듈 ID	UM-002
1. 개요			
사용자는 '♡' 표시를 눌러 전시회와 작품에 대한 선호를 표현 할 수 있다. 토글 방식으로 동작하며, 선호를 표시한 전시회와 작품은 모아서 따로 확인할 수 있다. 전시회와 작품에 대한 공유 행위 또한 선호 표현으로 판단한다.			
2. 주 사용 테이블			
테이블 명	테이블 설명	연관 테이블	
exhibitionLike	사용자가 선호하는 전시회 정보 저장을 위한 테이블	user, exhibition	
pieceLike	사용자가 선호하는 작품 정보 저장을 위한 테이블	user, piece	
3. 기능 명세			
기능명	좋아요 목록보기	기능 ID	
개요	사용자가 선호를 표시했던 작품이나 전시회 목록을 볼 수 있다.		
상세설명	<ul style="list-style-type: none">- 마이페이지에 있는 좋아요 목록보기를 선택한다면, 지금까지 사용자가 '좋아요'를 누른 항목들을 확인할 수 있다.- 카테고리는 '전시회 좋아요'와 '작품 좋아요' 2가지로 분류된다.- '좋아요'목록은 사용자가 '좋아요'를 눌렀던 순서의 역순으로 정렬된다. (가장 최근에 호감을 표시한 작품 or 전시회가 우선으로 보여진다.)		
Web Server 담당자	송무송	Web Client 담당자	조은진
서비스 대상	일반 사용자	중요도	상
기능명	전시회 좋아요	기능 ID	
개요	사용자가 마음에 드는 전시회에 선호를 표시할 수 있다.		
상세설명	<ul style="list-style-type: none">- 사용자는 하트(좋아요)를 눌러 해당 전시회에 대한 호감을 표현할 수 있다.- '좋아요'가 설정되었다면, 좋아요 버튼에 표시.- '좋아요'가 설정된 상태에서 다시한번 더 하트를 클릭하면 '좋아요'가 취소된다.		
Web Server 담당자	송무송	Web Client 담당자	조은진
서비스 대상	일반 사용자	중요도	상
4. 클래스 다이어그램			
<pre>classDiagram class Exhibition { + name = models.CharField + explain = models.TextField + poster = models.ImageField + create_at = models.DateTimeField + remove_at = models.DateTimeField + start_at = models.DateTimeField + end_at = models.DateTimeField + click_count = models.IntegerField + user = models.ForeignKey + category = models.ForeignKey + __str__(self): string + get_absolute_url(self): string + get_file_url(self): string } class Piece { + name = models.CharField + explain = models.TextField + image = models.ImageField + size = models.CharField + author = models.CharField + create_at = models.DateTimeField + remove_at = models.DateTimeField + order = models.IntegerField + click_count = models.IntegerField + category = models.ForeignKey + material = models.ForeignKey + __str__(self): string + get_absolute_url(self): string + get_file_url(self): string } class ExhibitionLike { + user = models.ForeignKey + exhibition = models.ForeignKey + __str__(self): string } class PieceLike { + user = models.ForeignKey + piece = models.ForeignKey + __str__(self): string } class LikeManage { + exhibition_like(request, pk): response + exhibition_dislike(request, ak, pk): response + piece_like(request, pk): response + piece_dislike(request, ak, pk): response } class LikeExhibitionList { + model : Exhibition + ordering : string + paginate_by : int + get_context_Data(self, **kwargs): context } class LikePieceList { + model : Piece + ordering : string + paginate_by : int + get_context_Data(self, **kwargs): context } class django_views_generic_List_View_1[django.views.generic.ListView] class django_views_generic_List_View_2[django.views.generic.ListView] Exhibition < -- ExhibitionLike Piece < -- PieceLike ExhibitionLike < -- LikeExhibitionList PieceLike < -- LikePieceList LikeManage < -- LikeExhibitionList LikeManage < -- LikePieceList django_views_generic_List_View_1 < -- LikeExhibitionList django_views_generic_List_View_2 < -- LikePieceList</pre>			

3.6.3 방명록 관리 모듈

모듈명		방명록 관리 모듈		모듈 ID		UM-003	
1. 개요							
사용자는 해당 전시회에 대하여 방명록을 확인할 수 있으며, 또한 전시회에 방명록을 등록하거나 삭제할 수 있다.							
2. 주 사용 테이블							
테이블 명		테이블 설명			연관 테이블		
guestBook		방명록 정보 저장을 위한 테이블			user, exhibition		
3. 기능 명세							
기능명		전시회 방명록 보기		기능 ID			
개요		사용자는 관람중인 전시회에 대한 모든 방명록을 확인할 수 있다.					
상세설명		<ul style="list-style-type: none">- 사용자는 현재 관람중인 전시회에 대한 방명록을 확인할 수 있다. 댓글일 경우 댓글을 단 항목에 대한 작품 그림도 옆에 게재한다.- 목록에 보여지는 항목의 개수: 10개- 페이지 처리: 맨 앞으로, 5페이지 단위, 맨 뒤로- 항목의 요소1(방명록): 닉네임, 작성 일자, 작성 내용- 항목의 요소2(댓글): 닉네임, 작성일자, 작성내용, 작품 이미지- 항목 나열 순서: 작성 일자 내림차순					
Web Server 담당자		송무송		Web Client 담당자		조은진	
서비스 대상		일반 사용자		중요도		상	
기능명		방명록 등록		기능 ID			
개요		사용자는 자신이 작성한 방명록을 등록할 수 있다.					
상세설명		<ul style="list-style-type: none">- 사용자는 로그인 여부 상관없이 방명록을 남길 수 있다.- 방명록은 페이지 위치 상관없이 하단에 방명록을 작성할 수 있다.- 로그인한 사용자 입력 항목: 방명록 내용- 비로그인 사용자의 입력 항목: 방명록 내용, 닉네임, 비밀번호- '등록하기' 버튼을 눌러 방명록의 등록을 마친다.- 비로그인 사용자는 접속한 IP도 함께 등록된다.- 등록이 완료되었음을 알리는 alert메시지를 출력.- 등록이 실패하였을 경우, 등록 실패 알림 메시지를 출력하며 리디렉션을 하지 않고 현재 페이지에 포커스를 맞춘다.					
작성항목 조건		<ul style="list-style-type: none">- 방명록 내용: 5글자 이상의 텍스트- 방명록 닉네임: 2글자 이상의 텍스트- 방명록 비밀번호: 4글자 이상의 텍스트					
Web Server 담당자		송무송		Web Client 담당자		조은진	
서비스 대상		일반 사용자		중요도		상	
기능명		방명록 삭제		기능 ID			
개요		사용자는 자신이 작성한 방명록을 삭제할 수 있다.					
상세설명		<ul style="list-style-type: none">- 로그인한 사용자의 삭제 요건: 본인이 맞는지 사용자 검증을 한 이후 삭제를 진행한다. DB상에는 삭제 날짜만 기입하여, 정보는 1년간 보존하도록 한다.- 비로그인 사용자의 삭제 요건: 삭제를 할 경우, 등록당시 기입했던 사용자의 비밀번호 대조를 통해 본인 확인을 거친 후, 삭제를 진행한다. DB상에 삭제한 날짜와, 삭제한 IP를 기록하고, 정보는 1년간 보존하도록 한다.- 삭제가 완료되었음을 나타내는 alert메시지 출력.- 삭제에 실패하였을 경우, 실패 알림 메시지를 출력하며 리디렉션을 하지 않고 현재 페이지에 포커스를 맞춘다.- 삭제된 항목은 '사용자가 삭제한 방명록입니다.'로 대체된다.					
Web Server 담당자		송무송		Web Client 담당자		조은진	
서비스 대상		일반 사용자		중요도		상	

4. 클래스 다이어그램

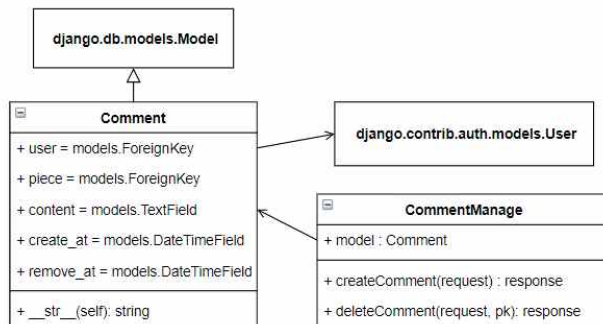


3.6.4 댓글 관리 모듈

모듈명		댓글 관리 모듈		모듈 ID		UM-004	
1. 개요							
사용자는 관람 중인 작품에 대하여 댓글을 확인할 수 있으며, 또한 댓글을 등록하거나 삭제할 수 있다.							
2. 주 사용 테이블							
테이블 명		테이블 설명			연관 테이블		
comment		댓글 정보 저장을 위한 테이블			user, piece		
3. 기능 명세							
기능명		작품 댓글 보기		기능 ID			
개요		사용자는 관람중인 그림에 대한 전체 댓글을 확인할 수 있다.					
상세설명		<ul style="list-style-type: none">- 사용자는 현재 관람중인 그림에 대한 방명록을 확인할 수 있다.- 목록에 보여지는 항목의 개수: 10개- 페이지 처리: 맨 앞으로, 5페이지 단위, 맨 뒤로- 항목의 요소: 닉네임, 작성 일자, 작성 내용- 항목 나열 순서: 작성 일자 내림차순					
Web Server 담당자		송무송		Web Client 담당자		문석희	
서비스 대상		일반 사용자		중요도		상	
기능명		댓글 등록		기능 ID			
개요		사용자는 관람중인 그림에 대하여 댓글을 작성한다.					
상세설명		<ul style="list-style-type: none">- 사용자는 로그인 여부 상관없이 댓글을 남길 수 있다.- 댓글을 페이지 위치 상관없이 하단에 댓글을 작성할 수 있다.- 로그인한 사용자 입력 항목: 댓글의 내용- 비로그인 사용자의 입력 항목: 댓글의 내용, 닉네임, 비밀번호- ‘등록하기’ 버튼을 눌러 댓글의 등록을 마친다.- 비로그인 사용자는 접속한 IP도 함께 등록된다.- 등록이 완료되었음을 알리는 alert메시지를 출력.- 등록이 실패하였을 경우, 등록 실패 알림 메시지를 출력하며 리디렉션을 하지 않고 현재 페이지에 포커스를 맞춘다.					
작성항목 조건		<ul style="list-style-type: none">- 댓글 내용: 5글자 이상의 텍스트- 댓글 닉네임: 2글자 이상의 텍스트- 댓글 비밀번호: 4글자 이상의 텍스트					
Web Server 담당자		송무송		Web Client 담당자		문석희	
서비스 대상		일반 사용자		중요도		상	

기능명	댓글 삭제	기능 ID	
개요	사용자는 자신이 작성한 댓글을 삭제할 수 있다.		
상세설명	<ul style="list-style-type: none"> - 로그인한 사용자의 삭제 요건: 본인이 맞는지 사용자 인증을 한 이후 삭제를 진행한다. DB상에는 삭제 날짜만 기입하여, 정보는 1년간 보존하도록 한다. - 비로그인 사용자의 삭제 요건: 삭제를 할 경우, 등록당시 기입했던 사용자의 비밀번호 대조를 통해 본인 확인을 거친 후, 삭제를 진행한다. DB상에 삭제한 날짜와, 삭제한 IP를 기록하고, 정보는 1년간 보존하도록 한다. - 삭제에 실패하였을 경우, 실패 알림 메시지를 출력하며 리디렉션을 하지 않고 현재 페이지에 포커스를 맞춘다. - 삭제된 항목은 '사용자가 삭제한 댓글입니다.'로 대체된다. 		
Web Server 담당자	송무송	Web Client 담당자	문석희
서비스 대상	일반 사용자	중요도	상

4. 클래스 다이어그램



3.6.5 전시회 관리 모듈

모듈명

전시회 관리 모듈

모듈 ID

AM-001

1. 개요

전시 권한을 가진 사용자는 전시회를 등록, 수정, 삭제를 통한 전시회 관리를 할 수 있다.

2. 주 사용 테이블

테이블 명	테이블 설명	연관 테이블
exhibition	전시회 정보 저장을 위한 테이블	user, category

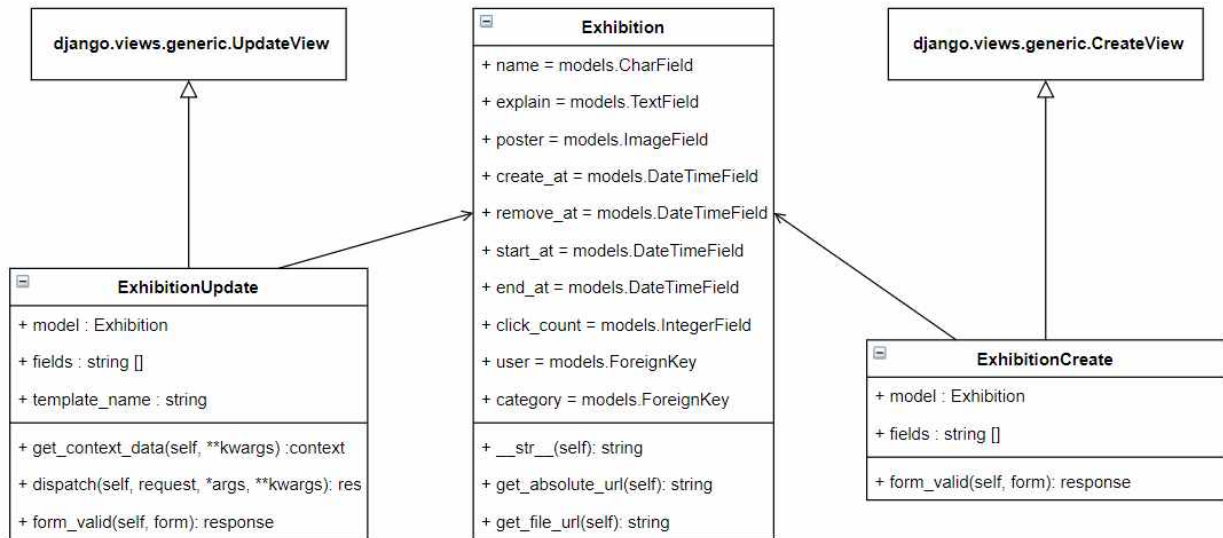
3. 기능 명세

기능명	전시회 등록	기능 ID	
개요	전시자가 전시회를 등록한다.		
상세설명	<div>- 한 전시회 당 한 명만 전시회를 등록할 수 있는 권한을 지님</div> <div>- 전시회를 등록한 사용자가 전시회 관리자가 됨</div> <div>- 전시회 등록 서비스 제공</div>		
Web Server 담당자	송무송	Web Client 담당자	조은진
서비스 대상	전시자 (전시 권한을 받은 사용자)	중요도	상

기능명	전시회 수정	기능 ID	
개요	전시자가 해당 전시회의 내용을 수정할 수 있다.		
상세설명	<div>- 전시 정보 수정 서비스 제공</div> <div>- 수정은 관리자의 승인 하에 이루어진다.</div>		
Web Server 담당자	송무송	Web Client 담당자	조은진
서비스 대상	전시자 (전시 권한을 받은 사용자)	중요도	상

기능명	전시회 삭제	기능 ID	
개요	전시자는 자신의 전시회를 삭제할 수 있다.		
상세설명	-전시회 삭제 서비스 제공 -전시 운영 기간 이후에는 자동으로 전시회 종료(삭제)		
Web Server 담당자	송무송	Web Client 담당자	조은진
서비스 대상	전시자 (전시 권한을 받은 사용자)	중요도	상

4. 클래스 다이어그램



3.6.6 작품 관리 모듈

모듈명	작품 관리 모듈	모듈 ID	AM-002
-----	----------	-------	--------

1. 개요

전시 권한을 가진 사용자는 등록된 전시회에 대한 작품을 등록, 수정, 삭제를 할 수 있다.

2. 주 사용 테이블

테이블 명	테이블 설명	연관 테이블
piece	작품 정보 저장을 위한 테이블	material, category

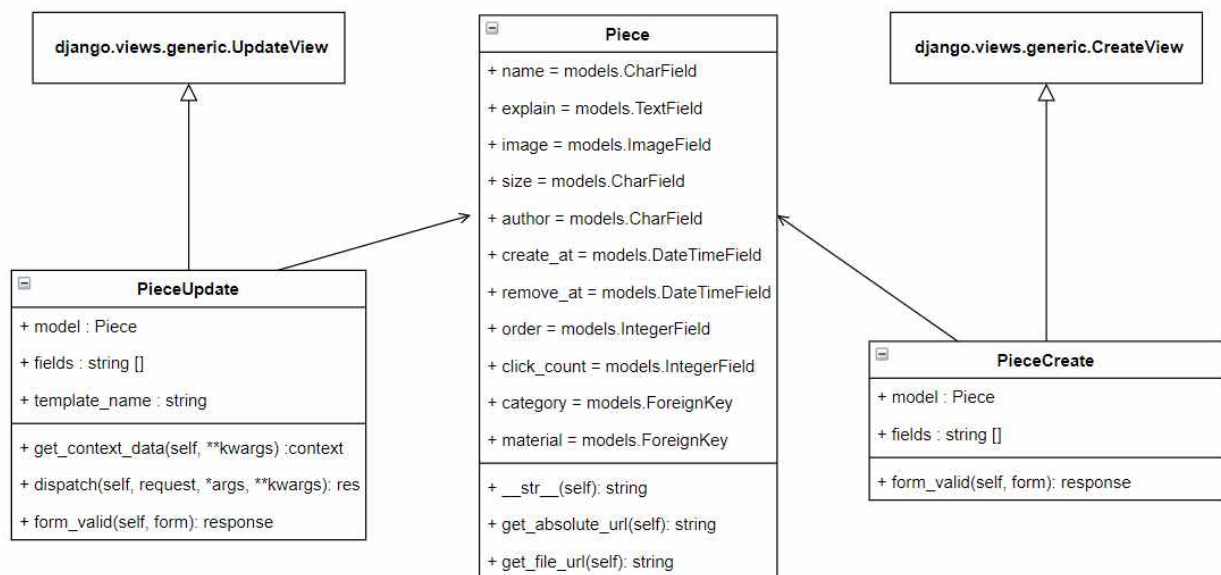
3. 기능 명세

기능명	작품 등록	기능 ID	
개요	전시자(전시회 관리자)가 작품을 개별적으로 등록한다.		
상세설명	-전시회 작품 등록 서비스 제공 -작품 관리는 전시회 관리자가 담당 -작품은 하나씩만 업로드 가능		
Web Server 담당자	송무송	Web Client 담당자	문석희
서비스 대상	전시자 (전시 권한을 받은 사용자)	중요도	상

기능명	작품 수정	기능 ID	
개요	전시자(전시회 관리자)가 등록된 작품 정보를 수정할 수 있다.		
상세설명	-등록 작품 수정 서비스 제공 -'내 작품 목록'에서 수정 버튼 클릭 시 수정 페이지 띄우기 -수정 페이지에서 하단 취소 버튼 클릭 시, '수정된 내용은 저장되지 않습니다.' 메시지 출력 확인: '내 작품 목록' 페이지로 이동 취소: 해당 페이지(수정 페이지)에 머물기 ※작품 이미지는 수정 불가능 확인 : 변경된 내용 수정 취소 : 해당 페이지(수정 페이지)에 머물기		
Web Server 담당자	송무송	Web Client 담당자	문석희
서비스 대상	전시자 (전시 권한을 받은 사용자)	중요도	상

기능명	작품 삭제	기능 ID	
개요	전시자(전시회 관리자)가 작품을 삭제 할 수 있다.		
상세설명	-작품 삭제 서비스 제공 -'내 작품 목록'에서 삭제 버튼 클릭 시 '정말 삭제 하시겠습니까?' 메시지 띄우기 확인 : '작품이 삭제 되었습니다' 메시지 출력 -> 해당 작품 삭제 취소 : 해당 페이지(내 작품 목록 페이지) 머물기		
Web Server 담당자	송무송	Web Client 담당자	문석희
서비스 대상	전시자 (전시 권한을 받은 사용자)	중요도	상

4. 클래스 다이어그램



4. 프로젝트 수행 결과

4.1 개발 환경

개발 PC	CPU	Intel Core i7-1065G7, Intel Core i5-7200U
	RAM	16GB
	OS	Windows 10
	IDE and Tool	Pycharm Professional 21.1.1, WebStrom 21.1.1, ATOM 1.57.0 cmdr, putty, Adobe Xd 40, drawio, ovenio
Server	Cloud Server	AWS-EC2
	OS	Ubuntu 18.04
	Web Server	Nginx
	Framework	Django
	DBMS	SQLite
	Language	python3 (3.6.5)
Client	Language	html, css
	미지원 브라우저	IE

4.2 Open Source, Open data, API, Library 활용

4.2.1. Python packages

package name	version	package name	version
asgiref	3.3.4	oauthlib	3.1.0
certifi	2020.12.5	Pillow	8.2.0
cffi	1.14.5	pycparser	2.20
chardet	4.0.0	PyJWT	2.1.0
cryptography	3.4.7	python3-openid	3.2.0
defusedxml	0.7.1	pytz	2021.1
Django	3.2.2	requests	2.25.1
django-allauth	0.44.0	requests-oauthlib	1.3.0
django-crispy-forms	1.11.2	sqlparse	0.4.1
idna	2.10	urllib3	1.26.4

4.2.2. Open data

기재 예정

4.2.3. Library

기재 예정

4.3 구현 내용

4.3.1. Model & Database

서비스 운영 시 사용되는 테이블			
type	name	tbl_name	
1 table	django_migrations	django_migrations	
2 table	sqlite_sequence	sqlite_sequence	
3 table	auth_group_permissions	auth_group_permissions	
4 table	auth_user_groups	auth_user_groups	
5 table	auth_user_user_permissions	auth_user_user_permissions	
6 table	account_emailaddress	account_emailaddress	
7 table	account_emailconfirmation	account_emailconfirmation	
8 table	django_admin_log	django_admin_log	
9 table	django_content_type	django_content_type	
10 table	auth_permission	auth_permission	
11 table	auth_group	auth_group	
12 table	auth_user	auth_user	
13 table	exhibition_category	exhibition_category	
14 table	exhibition_exhibition	exhibition_exhibition	
15 table	exhibition_material	exhibition_material	
16 table	exhibition_pieceshare	exhibition_pieceshare	
17 table	exhibition_piecelike	exhibition_piecelike	
18 table	exhibition_piececlick	exhibition_piececlick	
19 table	exhibition_initiallike	exhibition_initiallike	
20 table	exhibition_guestbook	exhibition_guestbook	
21 table	exhibition_exhibitionsare	exhibition_exhibitionsare	
22 table	exhibition_exhibitionlike	exhibition_exhibitionlike	
23 table	exhibition_exhibitionclick	exhibition_exhibitionclick	
24 table	exhibition_comment	exhibition_comment	
25 table	django_session	django_session	
26 table	django_site	django_site	
27 table	socialaccount_socialapp_sites	socialaccount_socialapp_sites	
28 table	socialaccount_socialaccount	socialaccount_socialaccount	
29 table	socialaccount_socialapp	socialaccount_socialapp	
30 table	socialaccount_socialtoken	socialaccount_socialtoken	
31 table	exhibition_piece	exhibition_piece	

서비스 운영시 사용되는 주요 테이블

테이블 명	exhibition_exhibition	DBMS	SQLite																																																																																				
테이블 설명	전시회의 정보를 담기 위한 테이블																																																																																						
테이블 구성																																																																																							
SQL	pragma table_info("exhibition_exhibition");																																																																																						
<table><thead><tr><th></th><th>cid</th><th>name</th><th>type</th><th>notnull</th><th>dflt_value</th><th>pk</th></tr></thead><tbody><tr><td>1</td><td>0</td><td>id</td><td>integer</td><td>1</td><td><null></td><td>1</td></tr><tr><td>2</td><td>1</td><td>name</td><td>varchar(50)</td><td>1</td><td><null></td><td>0</td></tr><tr><td>3</td><td>2</td><td>explain</td><td>text</td><td>1</td><td><null></td><td>0</td></tr><tr><td>4</td><td>3</td><td>poster</td><td>varchar(100)</td><td>0</td><td><null></td><td>0</td></tr><tr><td>5</td><td>4</td><td>create_at</td><td>datetime</td><td>1</td><td><null></td><td>0</td></tr><tr><td>6</td><td>5</td><td>click_count</td><td>integer</td><td>1</td><td><null></td><td>0</td></tr><tr><td>7</td><td>6</td><td>category_id</td><td>bigint</td><td>0</td><td><null></td><td>0</td></tr><tr><td>8</td><td>7</td><td>user_id</td><td>integer</td><td>0</td><td><null></td><td>0</td></tr><tr><td>9</td><td>8</td><td>mode_id</td><td>bigint</td><td>0</td><td><null></td><td>0</td></tr><tr><td>10</td><td>9</td><td>end_at</td><td>date</td><td>0</td><td><null></td><td>0</td></tr><tr><td>11</td><td>10</td><td>start_at</td><td>date</td><td>0</td><td><null></td><td>0</td></tr></tbody></table>					cid	name	type	notnull	dflt_value	pk	1	0	id	integer	1	<null>	1	2	1	name	varchar(50)	1	<null>	0	3	2	explain	text	1	<null>	0	4	3	poster	varchar(100)	0	<null>	0	5	4	create_at	datetime	1	<null>	0	6	5	click_count	integer	1	<null>	0	7	6	category_id	bigint	0	<null>	0	8	7	user_id	integer	0	<null>	0	9	8	mode_id	bigint	0	<null>	0	10	9	end_at	date	0	<null>	0	11	10	start_at	date	0	<null>	0
	cid	name	type	notnull	dflt_value	pk																																																																																	
1	0	id	integer	1	<null>	1																																																																																	
2	1	name	varchar(50)	1	<null>	0																																																																																	
3	2	explain	text	1	<null>	0																																																																																	
4	3	poster	varchar(100)	0	<null>	0																																																																																	
5	4	create_at	datetime	1	<null>	0																																																																																	
6	5	click_count	integer	1	<null>	0																																																																																	
7	6	category_id	bigint	0	<null>	0																																																																																	
8	7	user_id	integer	0	<null>	0																																																																																	
9	8	mode_id	bigint	0	<null>	0																																																																																	
10	9	end_at	date	0	<null>	0																																																																																	
11	10	start_at	date	0	<null>	0																																																																																	
실제로 테이블에 담긴 데이터 예시																																																																																							
SQL	Select name, poster, category_id, explain from exhibition_exhibition;																																																																																						

name	poster	category_id	explain
1 화인부락 1회 온라인	exhibition/2021/05/21/전...	1	화인부락은 송실대학교의 유...
2 르누아르	exhibition/2021/05/26/2...	1	프랑스의 대표적인 인상주의
3 Salman Khoshroo	exhibition/2021/05/27/k...	2	사연은 이랬습니다. 코로나로
4 Sehban.ali.akbar	exhibition/2021/05/27/8...	3	그는 브랜드 로고디자인을 전

테이블 명	exhibition_piece	DBMS	SQLite
테이블 설명	작품의 정보를 담기 위한 테이블		

테이블 구성	
SQL	pragma table_info("exhibition_piece");

	cid	name	type	notnull	dflt_value	pk
1	0	id	integer	1	<null>	1
2	1	name	varchar(50)	1	<null>	0
3	2	explain	text	1	<null>	0
4	3	image	varchar(100)	1	<null>	0
5	4	size	varchar(100)	1	<null>	0
6	5	author	varchar(50)	1	<null>	0
7	6	create_at	datetime	1	<null>	0
8	7	click_count	integer	0	<null>	0
9	8	exhibition_id	bigint	0	<null>	0
10	9	material_id	bigint	0	<null>	0
11	10	make_at	date	0	<null>	0
12	11	order	integer	0	<null>	0

실제로 테이블에 담긴 데이터 예시	
SQL	Select name, size, author, click_count, exhibition_id, material_id from exhibit...

	name	size	author	click_count	exhibition_id	material_id
1	겨울의 코스모스	60x50cm	김미나	77	1	2
2	금봉어	50x60cm	김민주	77	1	2
3	공허	128x180mm	김세영	106	1	3
4	곧목길	60x50cm	김수연	74	1	2
5	Tu me manques	21x21cm	김예은	55	1	3
6	평화로운 고슴도치	22x27cm	김지호	53	1	4
7	긴 하루	2732x2048pixel	김혜수	51	1	1
8	그릴 수 있지	3000x3000pixel	김혜수	42	1	1
9	타르트 한 조각	30x20cm	김수연	105	1	2
10	사슴	22x28cm	문관우	32	1	4
11	화관을 쓴 소녀	600x800pixel	문관우	37	1	5
12	청동꽃병에 담긴 왕관초가 있는 정물	60x50cm	문유진	56	1	2
13	black rose	60x50cm	박건희	80	1	2
14	두 송이	25x18cm	성나영	49	1	6
15	술에서 만난	15x15cm	성나영	33	1	2
16	회화나무	130x80cm	심영지	43	1	2
17	머머줄	34x24cm	심영지	31	1	4
18	달기케이크	33x45cm	안유빈	77	1	2

테이블 명	auth_user	DBMS	SQLite
테이블 설명	사용자의 정보를 담기 위한 테이블		

테이블 구성	
SQL	pragma table_info("auth_user");

	cid	name	type	notnull	dflt_value	pk
1	0	id	integer	1	<null>	1
2	1	password	varchar(128)	1	<null>	0
3	2	last_login	datetime	0	<null>	0
4	3	is_superuser	bool	1	<null>	0
5	4	username	varchar(150)	1	<null>	0
6	5	last_name	varchar(150)	1	<null>	0
7	6	email	varchar(254)	1	<null>	0
8	7	is_staff	bool	1	<null>	0
9	8	is_active	bool	1	<null>	0
10	9	date_joined	datetime	1	<null>	0
11	10	first_name	varchar(150)	1	<null>	0

실제로 테이블에 담긴 데이터 예시

SQL SELECT password, is_superuser, leftstr(username, 1) || '**' as username , leftstr(auth_user.email, 5) || '**@****' as email from auth_user join account_emailaddress ae on auth_user.id = ae.user_id;

	password	is_superuser	username	email
1	!VSkw2Vwovj3AdNX4LJyurKHmITznxkrCCJC10aE5	0	송**	real.**@****
2	!9isDgmSmgrSdHn50EPBzHoLBH7LP4091Zq1yEAUE	0	S**	songe**@****
3	!6iu0v3QGxnrj00pLCMV2avrWVIGfP3aVok2zpY0uz	0	최**	wnsdu**@****
4	!330sLY2SNLXBCWca7869nLBvwh41YSF0mDuKJUzN	0	황**	2000h**@****
5	!Fyza0oQZ0Y6LvH0ijD8Jctp91UuQNsJkouxCEuLN	0	윤**	gmin0**@****
6	!AoKtM4cVCLX2D0mZ1PPbr9ro30W66oDA9rf1AQLr	0	p**	wnsdu**@****
7	!V5wFhc9L6YcClTE9Gqfo2pWwhuSPvrgsLK5fvyQq	0	최**	neilm**@****
8	!0LIktZCTZCVW2eHoGhiW7ZhzzVFqcnLRnfWx173e	0	조**	98dms**@****
9	!nB9fm1jJjcpCnUh5i18s6xFeA0ob2i657L2CWxx5	0	진**	jgk08**@****
10	!M6f5utIF8HpeAldp4QFzU6Z7oTXWPdppTYy8ToQV	0	남**	ng006**@****
11	!LKSWyKtRSM1WfS1yCXYjEEFTZvSNxzpXHPzXrWI	0	김**	r1ar1**@****
12	!cBPNxQR4g7u4sHiSnw13ktiLJldcs96sKjdCQhPi	0	자**	jay95**@****
13	!PMxuMM38fKtDJ4xbAd2V0KvyT2bS4m5zEzwN0unj	0	강**	khw42**@****

테이블 명	exhibition_category	DBMS	SQLite
-------	---------------------	------	--------

테이블 설명 카테고리 정보를 담기 위한 테이블

테이블 구성

SQL pragma table_info("exhibition_category");

	cid	name	type	notnull	dflt_value	pk
1	0	id	integer	1	<null>	1
2	1	name	varchar(50)	1	<null>	0
3	2	slug	varchar(50)	1	<null>	0
4	3	image	varchar(100)	0	<null>	0

실제로 테이블에 담긴 데이터 예시

SQL SELECT * from exhibition_category;

	id	name	slug
1	1	회화	회화
2	2	공예	공예
3	3	디자인	디자인
4	4	미디어	미디어
5	5	기타	기타

테이블 명	exhibition_material	DBMS	SQLite																																																
테이블 설명	재료의 정보를 담기 위한 테이블																																																		
테이블 구성																																																			
SQL	pragma table_info("exhibition_material");																																																		
<table><tr><th></th><th>cid</th><th>name</th><th>type</th><th>notnull</th><th>dflt_value</th><th>pk</th></tr><tr><td>1</td><td>0</td><td>id</td><td>integer</td><td>1</td><td><null></td><td>1</td></tr><tr><td>2</td><td>1</td><td>name</td><td>varchar(50)</td><td>1</td><td><null></td><td>0</td></tr><tr><td>3</td><td>2</td><td>slug</td><td>varchar(200)</td><td>1</td><td><null></td><td>0</td></tr><tr><td>4</td><td>3</td><td>image</td><td>varchar(100)</td><td>0</td><td><null></td><td>0</td></tr></table>					cid	name	type	notnull	dflt_value	pk	1	0	id	integer	1	<null>	1	2	1	name	varchar(50)	1	<null>	0	3	2	slug	varchar(200)	1	<null>	0	4	3	image	varchar(100)	0	<null>	0													
	cid	name	type	notnull	dflt_value	pk																																													
1	0	id	integer	1	<null>	1																																													
2	1	name	varchar(50)	1	<null>	0																																													
3	2	slug	varchar(200)	1	<null>	0																																													
4	3	image	varchar(100)	0	<null>	0																																													
실제로 테이블에 담긴 데이터 예시																																																			
SQL	SELECT * from exhibition_material;																																																		
<table><tr><th></th><th>id</th><th>name</th><th>slug</th></tr><tr><td>1</td><td>1</td><td>아이패드</td><td>아이패드</td></tr><tr><td>2</td><td>2</td><td>캔버스에 유채</td><td>캔버스에-유채</td></tr><tr><td>3</td><td>3</td><td>도화지에 오일파스텔</td><td>도화지에-오일파스텔</td></tr><tr><td>4</td><td>4</td><td>캔버스에 아크릴</td><td>캔버스에-아크릴</td></tr><tr><td>5</td><td>5</td><td>노트북</td><td>노트북</td></tr><tr><td>6</td><td>6</td><td>도화지에 색연필</td><td>도화지에-색연필</td></tr><tr><td>7</td><td>7</td><td>도화지에 흑연</td><td>도화지에-흑연</td></tr><tr><td>8</td><td>8</td><td>나무 책장에 스프레이</td><td>나무-책장에-스프레이</td></tr><tr><td>9</td><td>9</td><td>이비스페인트x</td><td>이비스페인트x</td></tr><tr><td>10</td><td>10</td><td>도화지에 색연필 파z</td><td>도화지에-색연필-파z</td></tr><tr><td>11</td><td>11</td><td>캔버스에 섬유</td><td>캔버스에-섬유</td></tr></table>					id	name	slug	1	1	아이패드	아이패드	2	2	캔버스에 유채	캔버스에-유채	3	3	도화지에 오일파스텔	도화지에-오일파스텔	4	4	캔버스에 아크릴	캔버스에-아크릴	5	5	노트북	노트북	6	6	도화지에 색연필	도화지에-색연필	7	7	도화지에 흑연	도화지에-흑연	8	8	나무 책장에 스프레이	나무-책장에-스프레이	9	9	이비스페인트x	이비스페인트x	10	10	도화지에 색연필 파z	도화지에-색연필-파z	11	11	캔버스에 섬유	캔버스에-섬유
	id	name	slug																																																
1	1	아이패드	아이패드																																																
2	2	캔버스에 유채	캔버스에-유채																																																
3	3	도화지에 오일파스텔	도화지에-오일파스텔																																																
4	4	캔버스에 아크릴	캔버스에-아크릴																																																
5	5	노트북	노트북																																																
6	6	도화지에 색연필	도화지에-색연필																																																
7	7	도화지에 흑연	도화지에-흑연																																																
8	8	나무 책장에 스프레이	나무-책장에-스프레이																																																
9	9	이비스페인트x	이비스페인트x																																																
10	10	도화지에 색연필 파z	도화지에-색연필-파z																																																
11	11	캔버스에 섬유	캔버스에-섬유																																																

테이블 명	exhibition_guestbook	DBMS	SQLite																																																																		
테이블 설명	방명록의 정보를 담기 위한 테이블																																																																				
테이블 구성																																																																					
SQL	pragma table_info("exhibition_guestbook");																																																																				
<table><tr><th></th><th>cid</th><th>name</th><th>type</th><th>notnull</th><th>dflt_value</th><th>pk</th></tr><tr><td>1</td><td>0</td><td>id</td><td>integer</td><td>1</td><td><null></td><td>1</td></tr><tr><td>2</td><td>1</td><td>content</td><td>text</td><td>1</td><td><null></td><td>0</td></tr><tr><td>3</td><td>2</td><td>create_at</td><td>datetime</td><td>1</td><td><null></td><td>0</td></tr><tr><td>4</td><td>3</td><td>exhibition_id</td><td>bigint</td><td>0</td><td><null></td><td>0</td></tr><tr><td>5</td><td>4</td><td>user_id</td><td>integer</td><td>0</td><td><null></td><td>0</td></tr></table>					cid	name	type	notnull	dflt_value	pk	1	0	id	integer	1	<null>	1	2	1	content	text	1	<null>	0	3	2	create_at	datetime	1	<null>	0	4	3	exhibition_id	bigint	0	<null>	0	5	4	user_id	integer	0	<null>	0																								
	cid	name	type	notnull	dflt_value	pk																																																															
1	0	id	integer	1	<null>	1																																																															
2	1	content	text	1	<null>	0																																																															
3	2	create_at	datetime	1	<null>	0																																																															
4	3	exhibition_id	bigint	0	<null>	0																																																															
5	4	user_id	integer	0	<null>	0																																																															
실제로 테이블에 담긴 데이터 예시																																																																					
SQL	SELECT * from exhibition_guestbook;																																																																				
<table><tr><th></th><th>id</th><th>content</th><th>create_at</th><th>exhibition_id</th><th>user_id</th></tr><tr><td>1</td><td>3</td><td>전시 참여해주신 분들 너무 고맙네요~ 작품 하나하나 다 너무 잘 나와서 기분이 날아갈 것</td><td>2021-05-21 09:50:31.112159</td><td>1</td><td>4</td></tr><tr><td>2</td><td>4</td><td>화인부락 화이팅! 온라인 전시회 넘 멋져요</td><td>2021-05-21 05:57:56.311697</td><td>1</td><td>11</td></tr><tr><td>3</td><td>5</td><td>날이 초콜 오후입니다. 하나하나 ... 그런 이의 마음이 소중하게 느껴지는군요...참으로 잘</td><td>2021-05-21 06:04:31.899907</td><td>1</td><td>12</td></tr><tr><td>4</td><td>6</td><td>예술은 영혼을 살피우는 음식이라고 했었나요. 화인부락민들의 작품 하나 하나를 보며 저의 영</td><td>2021-05-21 15:11:19.572641</td><td>1</td><td>17</td></tr><tr><td>5</td><td>8</td><td>화인부락 화이팅! 준비하느라 애썼겠다! 이 시국에도 전시회 잘 준비하는 거 멋져요</td><td>2021-05-22 02:06:01.621787</td><td>1</td><td>19</td></tr><tr><td>6</td><td>9</td><td>잘보고 가요~ㅎㅎ 방명록 찾기가 힘들어용ㅠ!</td><td>2021-05-22 02:08:57.143264</td><td>1</td><td>18</td></tr><tr><td>7</td><td>10</td><td>준비하느라 고생 많았어요 온라인으로 전시회를 볼 수 있다니,, 너무 넘 멋진 부락민들</td><td>2021-05-22 02:13:47.255448</td><td>1</td><td>20</td></tr><tr><td>8</td><td>11</td><td>최고다 화부~~~~</td><td>2021-05-22 02:28:57.156664</td><td>1</td><td>21</td></tr><tr><td>9</td><td>12</td><td>잘 보고 갑니다!! 그럼 멋지네요!!</td><td>2021-05-22 02:56:45.537968</td><td>1</td><td>22</td></tr><tr><td>10</td><td>13</td><td>전시회 잘보고 갑니다~ 이렇게 온라인으로 전시회 볼수있다니 영광입니다 화부 화이팅!! ㅎㅎ</td><td>2021-05-23 01:49:44.135778</td><td>1</td><td>24</td></tr></table>					id	content	create_at	exhibition_id	user_id	1	3	전시 참여해주신 분들 너무 고맙네요~ 작품 하나하나 다 너무 잘 나와서 기분이 날아갈 것	2021-05-21 09:50:31.112159	1	4	2	4	화인부락 화이팅! 온라인 전시회 넘 멋져요	2021-05-21 05:57:56.311697	1	11	3	5	날이 초콜 오후입니다. 하나하나 ... 그런 이의 마음이 소중하게 느껴지는군요...참으로 잘	2021-05-21 06:04:31.899907	1	12	4	6	예술은 영혼을 살피우는 음식이라고 했었나요. 화인부락민들의 작품 하나 하나를 보며 저의 영	2021-05-21 15:11:19.572641	1	17	5	8	화인부락 화이팅! 준비하느라 애썼겠다! 이 시국에도 전시회 잘 준비하는 거 멋져요	2021-05-22 02:06:01.621787	1	19	6	9	잘보고 가요~ㅎㅎ 방명록 찾기가 힘들어용ㅠ!	2021-05-22 02:08:57.143264	1	18	7	10	준비하느라 고생 많았어요 온라인으로 전시회를 볼 수 있다니,, 너무 넘 멋진 부락민들	2021-05-22 02:13:47.255448	1	20	8	11	최고다 화부~~~~	2021-05-22 02:28:57.156664	1	21	9	12	잘 보고 갑니다!! 그럼 멋지네요!!	2021-05-22 02:56:45.537968	1	22	10	13	전시회 잘보고 갑니다~ 이렇게 온라인으로 전시회 볼수있다니 영광입니다 화부 화이팅!! ㅎㅎ	2021-05-23 01:49:44.135778	1	24
	id	content	create_at	exhibition_id	user_id																																																																
1	3	전시 참여해주신 분들 너무 고맙네요~ 작품 하나하나 다 너무 잘 나와서 기분이 날아갈 것	2021-05-21 09:50:31.112159	1	4																																																																
2	4	화인부락 화이팅! 온라인 전시회 넘 멋져요	2021-05-21 05:57:56.311697	1	11																																																																
3	5	날이 초콜 오후입니다. 하나하나 ... 그런 이의 마음이 소중하게 느껴지는군요...참으로 잘	2021-05-21 06:04:31.899907	1	12																																																																
4	6	예술은 영혼을 살피우는 음식이라고 했었나요. 화인부락민들의 작품 하나 하나를 보며 저의 영	2021-05-21 15:11:19.572641	1	17																																																																
5	8	화인부락 화이팅! 준비하느라 애썼겠다! 이 시국에도 전시회 잘 준비하는 거 멋져요	2021-05-22 02:06:01.621787	1	19																																																																
6	9	잘보고 가요~ㅎㅎ 방명록 찾기가 힘들어용ㅠ!	2021-05-22 02:08:57.143264	1	18																																																																
7	10	준비하느라 고생 많았어요 온라인으로 전시회를 볼 수 있다니,, 너무 넘 멋진 부락민들	2021-05-22 02:13:47.255448	1	20																																																																
8	11	최고다 화부~~~~	2021-05-22 02:28:57.156664	1	21																																																																
9	12	잘 보고 갑니다!! 그럼 멋지네요!!	2021-05-22 02:56:45.537968	1	22																																																																
10	13	전시회 잘보고 갑니다~ 이렇게 온라인으로 전시회 볼수있다니 영광입니다 화부 화이팅!! ㅎㅎ	2021-05-23 01:49:44.135778	1	24																																																																

테이블 명	exhibition_comment	DBMS	SQLite																																										
테이블 설명	댓글의 정보를 담기 위한 테이블																																												
테이블 구성																																													
SQL	pragma table_info("exhibition_comment");																																												
<table><tr><th></th><th>cid</th><th>name</th><th>type</th><th>notnull</th><th>dflt_value</th><th>pk</th></tr><tr><td>1</td><td>0</td><td>id</td><td>integer</td><td>1</td><td><null></td><td>1</td></tr><tr><td>2</td><td>1</td><td>content</td><td>text</td><td>1</td><td><null></td><td>0</td></tr><tr><td>3</td><td>2</td><td>create_at</td><td>datetime</td><td>1</td><td><null></td><td>0</td></tr><tr><td>4</td><td>3</td><td>piece_id</td><td>bigint</td><td>0</td><td><null></td><td>0</td></tr><tr><td>5</td><td>4</td><td>user_id</td><td>integer</td><td>0</td><td><null></td><td>0</td></tr></table>					cid	name	type	notnull	dflt_value	pk	1	0	id	integer	1	<null>	1	2	1	content	text	1	<null>	0	3	2	create_at	datetime	1	<null>	0	4	3	piece_id	bigint	0	<null>	0	5	4	user_id	integer	0	<null>	0
	cid	name	type	notnull	dflt_value	pk																																							
1	0	id	integer	1	<null>	1																																							
2	1	content	text	1	<null>	0																																							
3	2	create_at	datetime	1	<null>	0																																							
4	3	piece_id	bigint	0	<null>	0																																							
5	4	user_id	integer	0	<null>	0																																							

실제로 테이블에 담긴 데이터 예시				
SQL	SELECT * from exhibition_comment;			
	id	content	create_at	piece_id user_id
1	5	고슴도치 귀엽네요	2021-05-21 04:17:45.014854	6 6
2	6	두번째 작품이라는게 믿기지 않을 정도로 명작이네요!! 역시 숭실대의 빈센트 반 고흐 최	2021-05-21 04:48:45.744370	13 7
3	7	너무 멋진 작품 잘 봤습니다^^ <장미 디테일이 최고네요 ! 빈센트 반 고흐 최고 ~	2021-05-21 05:16:03.314791	13 8
4	8	색조화가 너무 예쁘네요 몽글몽글~ 빈센트 반 고흐님의 작품 잘 보고 갑니다 ^^	2021-05-21 10:46:57.280593	13 16
5	9	스바라시	2021-05-22 02:11:58.186986	17 20
6	10	그림 잘 보고 갑니다..^^	2021-05-22 02:58:36.461090	25 23
7	11	66666666666666	2021-05-23 04:13:15.068263	4 25
8	12	66666666666666	2021-05-23 04:13:27.634987	5 25

테이블 명	exhibition_exhibitionlike	DBMS	SQLite																												
테이블 설명	사용자 선호 전시회 정보를 담기 위한 테이블																														
테이블 구성																															
SQL	pragma table_info("exhibition_exhibitionlike");																														
<table><thead><tr><th></th><th>cid</th><th>name</th><th>type</th><th>notnull</th><th>dflt_value</th><th>pk</th></tr></thead><tbody><tr><td>1</td><td>0</td><td>id</td><td>integer</td><td>1</td><td><null></td><td>1</td></tr><tr><td>2</td><td>1</td><td>exhibition_id</td><td>bigint</td><td>0</td><td><null></td><td>0</td></tr><tr><td>3</td><td>2</td><td>user_id</td><td>integer</td><td>0</td><td><null></td><td>0</td></tr></tbody></table>					cid	name	type	notnull	dflt_value	pk	1	0	id	integer	1	<null>	1	2	1	exhibition_id	bigint	0	<null>	0	3	2	user_id	integer	0	<null>	0
	cid	name	type	notnull	dflt_value	pk																									
1	0	id	integer	1	<null>	1																									
2	1	exhibition_id	bigint	0	<null>	0																									
3	2	user_id	integer	0	<null>	0																									

실제로 테이블에 담긴 데이터 예시			
SQL	SELECT * from exhibition_exhibitionlike;		
	id	exhibition_id	user_id
1	4	1	10
2	5	1	11
3	6	1	12
4	8	1	17
5	9	1	18
6	10	1	21
7	11	1	24
8	12	1	4
9	13	1	1
10	14	2	1

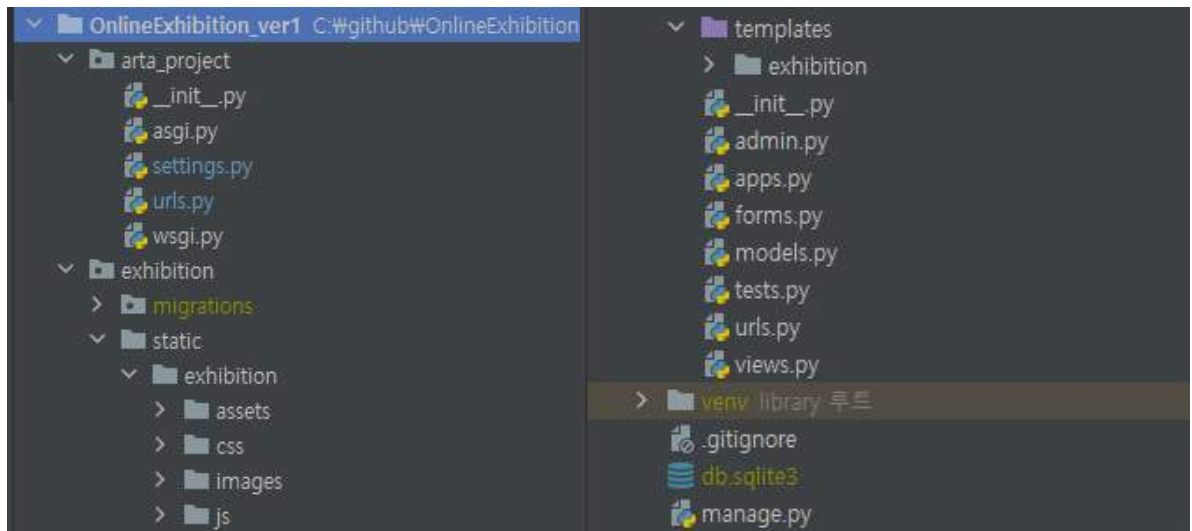
테이블 명	exhibition_piecelike	DBMS	SQLite			
테이블 설명	사용자 선호 작품 정보를 담기 위한 테이블					
테이블 구성						
SQL	pragma table_info("exhibition_piecelike");					
	cid	name	type	notnull	dflt_value	pk
1	0	id	integer	1	<null>	1
2	1	piece_id	bigint	0	<null>	0
3	2	user_id	integer	0	<null>	0

실제로 테이블에 담긴 데이터 예시			
SQL	SELECT * from exhibition_piecelike;		
	id	piece_id	user_id
32	42	30	10
33	44	31	10
34	45	26	12
35	46	30	13
36	47	31	13
37	48	32	13
38	49	3	12
39	50	31	4
40	51	1	4
41	52	30	4
42	53	32	4
43	55	30	10
44	56	1	15
45	57	13	16

테이블 명	exhibition_exhibitionlike	DBMS	SQLite																																				
테이블 설명	사용자 전시회 관람 정보를 담기 위한 테이블																																						
테이블 구성																																							
SQL	pragma table_info("exhibition_exhibitionclick");																																						
<table><tr><th></th><th>cid</th><th>name</th><th>type</th><th>notnull</th><th>dflt_value</th><th>pk</th></tr><tr><td>1</td><td>0</td><td>id</td><td>integer</td><td>1</td><td><null></td><td>1</td></tr><tr><td>2</td><td>1</td><td>exhibition_id</td><td>bigint</td><td>0</td><td><null></td><td>0</td></tr><tr><td>3</td><td>2</td><td>user_id</td><td>integer</td><td>0</td><td><null></td><td>0</td></tr></table>					cid	name	type	notnull	dflt_value	pk	1	0	id	integer	1	<null>	1	2	1	exhibition_id	bigint	0	<null>	0	3	2	user_id	integer	0	<null>	0								
	cid	name	type	notnull	dflt_value	pk																																	
1	0	id	integer	1	<null>	1																																	
2	1	exhibition_id	bigint	0	<null>	0																																	
3	2	user_id	integer	0	<null>	0																																	
실제로 테이블에 담긴 데이터 예시																																							
SQL	SELECT * from exhibition_exhibitionclick;																																						
<table><tr><th></th><th>id</th><th>exhibition_id</th><th>user_id</th></tr><tr><td>373</td><td>416</td><td>1</td><td>20</td></tr><tr><td>374</td><td>417</td><td>1</td><td>17</td></tr><tr><td>375</td><td>418</td><td>1</td><td>20</td></tr><tr><td>376</td><td>419</td><td>1</td><td>20</td></tr><tr><td>377</td><td>420</td><td>1</td><td>21</td></tr><tr><td>378</td><td>421</td><td>1</td><td>21</td></tr><tr><td>379</td><td>422</td><td>1</td><td>18</td></tr><tr><td>380</td><td>423</td><td>1</td><td>21</td></tr></table>					id	exhibition_id	user_id	373	416	1	20	374	417	1	17	375	418	1	20	376	419	1	20	377	420	1	21	378	421	1	21	379	422	1	18	380	423	1	21
	id	exhibition_id	user_id																																				
373	416	1	20																																				
374	417	1	17																																				
375	418	1	20																																				
376	419	1	20																																				
377	420	1	21																																				
378	421	1	21																																				
379	422	1	18																																				
380	423	1	21																																				

테이블 명	exhibition_piececlick	DBMS	SQLite																																																				
테이블 설명	사용자 작품 관람 정보를 담기 위한 테이블																																																						
테이블 구성																																																							
SQL	pragma table_info("exhibition_piececlick");																																																						
<table><tr><th></th><th>cid</th><th>name</th><th>type</th><th>notnull</th><th>dflt_value</th><th>pk</th></tr><tr><td>1</td><td>0</td><td>id</td><td>integer</td><td>1</td><td><null></td><td>1</td></tr><tr><td>2</td><td>1</td><td>piece_id</td><td>bigint</td><td>0</td><td><null></td><td>0</td></tr><tr><td>3</td><td>2</td><td>user_id</td><td>integer</td><td>0</td><td><null></td><td>0</td></tr></table>					cid	name	type	notnull	dflt_value	pk	1	0	id	integer	1	<null>	1	2	1	piece_id	bigint	0	<null>	0	3	2	user_id	integer	0	<null>	0																								
	cid	name	type	notnull	dflt_value	pk																																																	
1	0	id	integer	1	<null>	1																																																	
2	1	piece_id	bigint	0	<null>	0																																																	
3	2	user_id	integer	0	<null>	0																																																	
실제로 테이블에 담긴 데이터 예시																																																							
SQL	SELECT * from exhibition_piececlick;																																																						
<table><tr><th></th><th>id</th><th>piece_id</th><th>user_id</th></tr><tr><td>255</td><td>280</td><td>35</td><td>13</td></tr><tr><td>256</td><td>281</td><td>21</td><td>13</td></tr><tr><td>257</td><td>282</td><td>24</td><td>13</td></tr><tr><td>258</td><td>283</td><td>7</td><td>13</td></tr><tr><td>259</td><td>284</td><td>13</td><td>7</td></tr><tr><td>260</td><td>285</td><td>3</td><td>12</td></tr><tr><td>261</td><td>286</td><td>13</td><td>7</td></tr><tr><td>262</td><td>287</td><td>3</td><td>12</td></tr><tr><td>263</td><td>288</td><td>17</td><td>12</td></tr><tr><td>264</td><td>289</td><td>1</td><td>13</td></tr><tr><td>265</td><td>290</td><td>16</td><td>4</td></tr><tr><td>266</td><td>291</td><td>31</td><td>4</td></tr></table>					id	piece_id	user_id	255	280	35	13	256	281	21	13	257	282	24	13	258	283	7	13	259	284	13	7	260	285	3	12	261	286	13	7	262	287	3	12	263	288	17	12	264	289	1	13	265	290	16	4	266	291	31	4
	id	piece_id	user_id																																																				
255	280	35	13																																																				
256	281	21	13																																																				
257	282	24	13																																																				
258	283	7	13																																																				
259	284	13	7																																																				
260	285	3	12																																																				
261	286	13	7																																																				
262	287	3	12																																																				
263	288	17	12																																																				
264	289	1	13																																																				
265	290	16	4																																																				
266	291	31	4																																																				

4.3.2. 프로젝트 내 파일 구조



4.3.3. views.py

첨부C. Source Code 확인

4.3.4. 추천 알고리즘 구현 방안

개발환경: anaconda 4.5.4 , python 3.6.5 , tensorflow 2.5.0






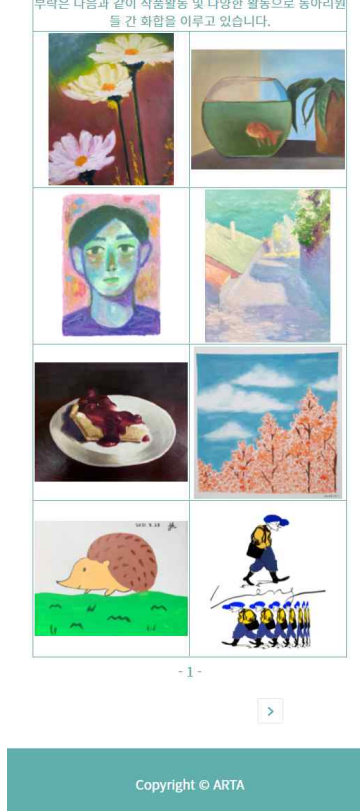
사용 데이터: 좋아요 클릭(작품/전시회), 방문로그(작품/전시회). 공유 클릭(작품/전시회), 방명록, 댓글, 작품의 태깅


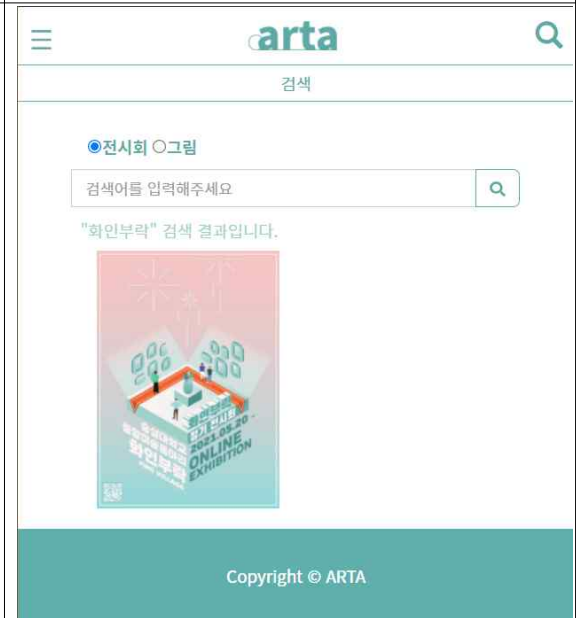
데이터 활용 방법

- 1) 작품 별 태깅을 한다. (ex) 수채화, 풍경화
- 2) 그림에 해당하는 태그별 점수를 보고 상위 3개의 점수를 가진 태그를 포함하는 작품군을 구성한다.
- 3) 작품군에 있는 작품을 가장 많이 포함하고 있는 전시회를 추천한다.

태깅 방식과 구체적인 추천 알고리즘은 2학기에 계속 과제로 진행할 예정이다.

4.4 결과 화면

1. 메인페이지	2. 전시회 목록	3. 전시회 관람
		
4. 사용자 선호 페이지		
		

<div data-bbox="256 161 655 224">5. 그림 보기</div> <div data-bbox="256 224 655 969">  </div>	<div data-bbox="655 161 1441 224">6. 방명록</div> <div data-bbox="655 224 1441 969">  </div>
<div data-bbox="256 969 863 1030">7. 검색</div> <div data-bbox="256 1030 863 1646">  </div>	<div data-bbox="863 969 1441 1030">8. 검색 결과</div> <div data-bbox="863 1030 1441 1646">  </div>

실제로 해당 서비스를 운영하는 화면이다.

4.5 평가 항목 및 방법

평가 부문	평가 항목	평가 방법(기준)
기능성	기능 구현 완성도	요구사항 목록의 기능이 모두 구현되어 있는가?
		UX/UI의 디자인 대로 Web Client 단이 구현되었는가?
		로그인한 사용자에게 대한 선호 정보에 따른 사용자 맞춤 전시회 제공이 이루어지는가?
		현재 운영 중인 오프라인 전시회에 대한 정보가 업로드되는가?
실용성	목표 기준 충족성	목표한 사용량을 충족하였는가?
		전시회에 조회가 1000회 이상인가?
		플랫폼에 가입한 사용자가 20명 이상인가?
		좋아요 기능을 사용한 사용자가 5명 이상인가?
		해당 플랫폼에 전시회가 3개 이상인가?

4.6 평가 결과

4.5.1. 기능성 평가

평가 항목	평가 방법(기준)	자체 평가
기능 구현 완성도	요구사항 목록의 기능이 모두 구현되어 있는가?	주요 기능 중 선 순위로 선정한 7가지 기능 중 전시자 부문을 제외한 5가지의 큰 기능을 구현하였다.
	UX/UI의 디자인 대로 Web Client 단이 구현되었는가?	기능이 구현되지 않은 것을 제외했을 때 13개의 UI 중 9개의 UI가 디자인 대로 구현되었다.
	로그인한 사용자에게 대한 선호 정보에 따른 사용자 맞춤 전시회 제공이 이루어지는가?	사용자 맞춤 전시회 제공은 2학기때 계속 진행할 예정이다.
	현재 운영 중인 오프라인 전시회에 대한 정보가 업로드되는가?	다른 웹에서 진행하는 오프라인 전시회 제공은 2학기때 계속 진행할 예정이다.

4.5.2. 실용성 평가

평가 항목	평가 방법(기준)	자체 평가
목표 기준 충족성	6월 14일까지 전시회에 조회가 1000회 이상이 되었는가?	6월 14일자로 전시회 조회는 2300회이다.
	6월 14일까지 플랫폼에 가입한 사용자가 20명 이상인가?	6월 14일자로 플랫폼 가입자는 35명이다.
	6월 14일까지 좋아요 기능을 사용한 사용자가 5명 이상인가?	6월 14일자로 좋아요 기능을 사용한 사용자는 12명이다.
	6월 14일까지 해당 플랫폼에 전시회가 3개 이상인가?	6월 14일자로 현재 플랫폼에 진행 중인 전시회는 총 1개이다.

4.7 구현 시 문제 상황 및 해결 방법

4.7.1 Client 단 개발팁

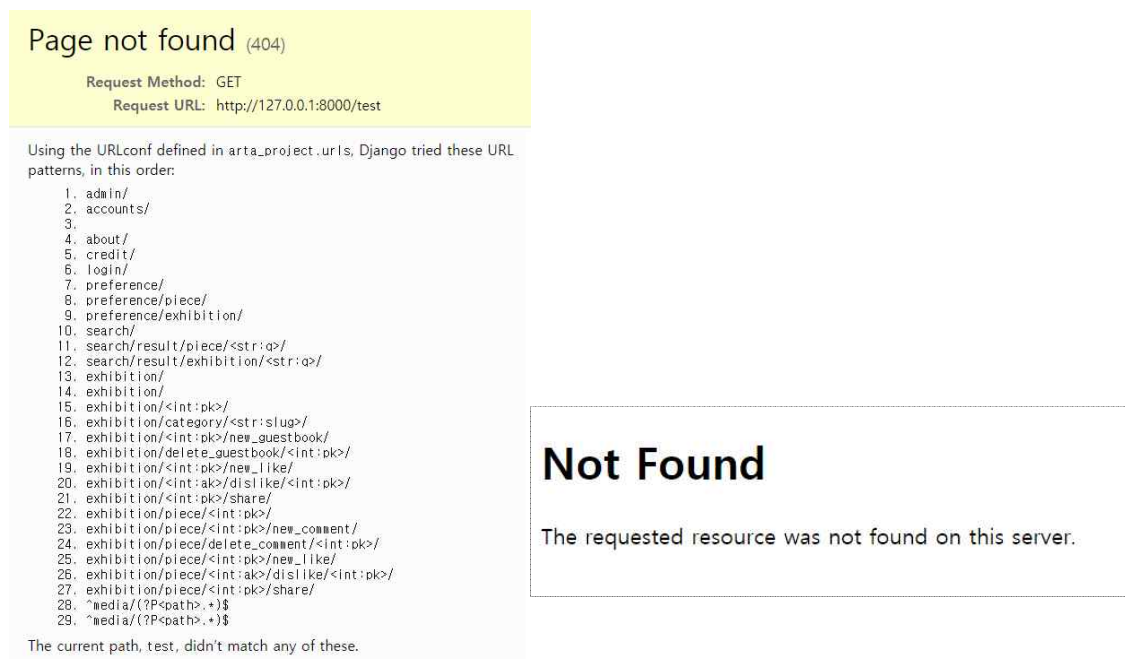
4.7.1.1 카카오 공유 API 사용 시 ErrorCode(4002) 문제

공유 버튼 클릭 시 로그인 은 되지만 공유가 되지 않고 잘못된 요청이라는 팝업이 뜬다. 이는 카카오에서 제공하는 웹 플랫폼에 사용하고 있는 사이트 도메인을 정확히 입력하지 않았기 때문에 발생한 에러로, 웹 플랫폼에 정확한 사이트 도메인과 포트를 추가하여 이 문제를 해결하였다.

4.7.2 Server 단 개발팁

4.7.2.1 서비스 배포시 Debug 모드로 인한 url 소스 코드 노출 문제

원활한 개발 환경을 위해서 장고에서는 디버깅 모드에 대한 제어를 제공하는데, 디버깅 모드가 true일 경우로 서버 배포를 하게 되면, 사용자가 의도적으로 다른 경로로 접근했을 경우 url의 정보가 노출된다. 따라서 Debug mode를 False로 두어 배포하여 해당 문제를 해결하였다.



Debug = True

Debug = False

4.7.2.2 Debug 모드 제한으로 인한 static, media 파일 접근 불가 문제

debug 모드를 false로 설정할 경우, static, media 폴더의 접근이 웹앱 내에서 가능하지 않다. 따라서 웹의 모든 정적파일을 끌고오지 못해 모든 페이지가 깨지게 된다. 이를 해결하기 위해서는 웹서버에서 연결해주는 작업을 진행해야 정적파일에 접근이 가능하다. etc/nginx/site-enabled/default 파일에서 장고가 접근하여 사용할 디렉터리 location을 직접적으로 지정하여 웹서버를 구동했다

```
ubuntu@ip-172-31-42-214: ~
ubuntu@ip-172-31-42-214:~$ ls /etc/nginx/
conf.d          koi-win          nginx.conf       sites-enabled
fastcgi.conf    mime.types       proxy_params     snippets
fastcgi_params  modules-available scgi_params      uwsgi_params
koi-utf          modules-enabled  sites-available  win-utf
```

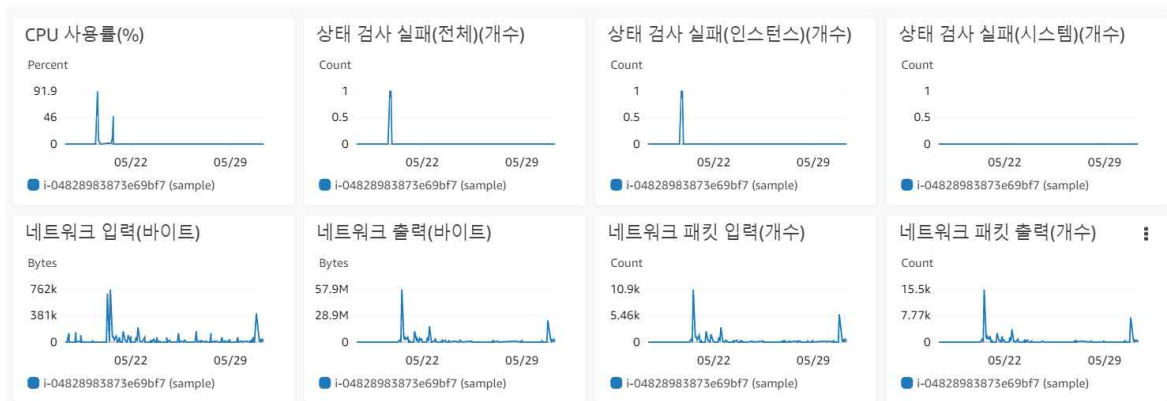
```
ubuntu@ip-172-31-42-214: ~
location / {
    # first attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    # try_files $uri $uri/ =404;
    include /etc/nginx/uwsgi_params;
    uwsgi_pass django;
}

location /static/{
    alias /home/ubuntu/ARTA_SERVER_DEVELOP/exhibition/static/;
}

location /media/{
    alias /home/ubuntu/ARTA_SERVER_DEVELOP/_media/;
}
```

ubuntu 서버의 etc/nginx/site-enabled/default 파일

4.7.2.3 동시접속자의 증가로 인한 서버 다운



5월 21일 서비스를 처음 배포하고 커뮤니티에 홍보글을 올린 이후 사용자의 접속이 증가하여 서버가 다운되었다. 따라서 서버를 수동으로 재가동하였는데, 서버가 다운된 이유는 현재 운영하고 있는 서버가 많은 사용자의 요청을 처리할 수 있는 사양이 아니기 때문이다.

t2.micro	1	i386, x86_64	1	-	-	Low to Moderate
----------	---	--------------	---	---	---	-----------------

t2.micro를 이용하여 서버를 구축한 이유는 동시접속자가 많지 않을 것이며 운영 기간을 길게 둘 것이기에, 추후 또 다시 서버 장애가 발생할 경우 서버를 업그레이드 하기로 하였으나, 이후 서버 다운 현상은 발생하지 않았다.

5. 기대 효과 및 활용 방안

5.1 기술적 측면

- 현재 온라인 전시회를 제공하는 플랫폼은 다양하게 존재하지만 인공지능을 이용한 사용자 맞춤형 추천 서비스를 제공하는 플랫폼은 아직까지 활성화된 것이 없다. 본 프로젝트는 사용자 맞춤형 추천 서비스를 개발하여 온/오프라인 전시회의 추천이 가능하게 한다. 이러한 기술을 개선한다면 전시회만이 아닌 사용자의 선호도가 중요시되는 영화, 연극 등의 예술 분야로 확장될 수 있다.

5.2 경제적·산업적 측면

- 코로나 19로 인한 불안감으로 오프라인 전시회의 수요는 감소하고 있는 추세이다. 오프라인 전시회를 대체할 방안으로 제시된 온라인 전시회는 코로나 19와 같은 환경적인 영향을 받지 않아 예술산업의 안정성을 만들어준다. 또한, 오프라인 전시회는 온라인 전시회에 비해 전시회를 여는 비용이 저렴한 편이다. 이는, 비용적 측면의 허들을 낮추어 이름 없는 작가들에게 자신의 작품을 선보일 수 있는 기회를 제공하여 더 많은 예술가들이 예술 산업에 뛰어들 수 있게 한다.
- 개인의 측면에서는 온라인 전시회를 관람하기 위해 여러 플랫폼에 가입을 해야 하는 번거로움이 있다. 하지만 본 프로젝트는 통합된 온라인 전시회를 지향하므로 하나의 플랫폼에서도 충분히 다양한 전시회를 관람할 수 있다. 또한 사용자 맞춤형 추천 서비스를 제공하기 때문에 자신의 취향에 맞는 전시회를 찾기 위해 소비되는 시간을 절약할 수 있게 된다.

5.3 사회·문화적 측면

- 본 프로젝트는 코로나 19로 인한 시간적, 공간적 제약으로 전시회 관람이 힘들었던 관람객들에게 시간적, 공간적 제약을 해소해줌으로써 예술 분야의 접근성을 높이는 데 도움이 될 것이다. 예술 분야의 접근성이 높아진다면 집단 간 문화 예술적 격차를 줄이는데도 기여할 것이다.

6. 프로젝트 수행 체계

6.1 팀원 인적 사항

성명	소속	경력
송무송 (팀장)	소프트웨어학부 (4학년)	- "스니핑 기술을 이용한 데이터 통신 프로토콜 분석 및 시각화에 관한 연구" (한국정보과학회 춘계학술대회, 2019.05) - "1인의 만찬 : 1인 가구를 위한 마감 음식 패키징 서비스 개발" (2020.07~08) - "Foreign Love : 교환학생 커뮤니티 개발" (2020.04~05)
문석희	소프트웨어학부(4학년)	- 2020 공공데이터 청년 인턴십, (2020.09~12)
조은진	소프트웨어학부 (4학년)	- "스니핑 기술을 이용한 데이터 통신 프로토콜 분석 및 시각화에 관한 연구" (한국정보과학회 춘계학술대회, 2019.05) - "AI personal trainer 개발" (캡스톤디자인종합프로젝트1, 2019.09~12)
주소영	소프트웨어학부 /건축학부(5학년)	- "머신러닝을 이용한 피싱사이트 탐지 기법" (융합보안프로젝트1, 2020.03~06)

6.2 프로젝트 역할 분담

성명	소속	역할	기여도
송무송 (팀장)	소프트웨어학부(4학년)	요구분석	요구사항 목록, 요구사항 정의서, 요구사항 명세서 작성
		시스템 명세	Rest API 설계
		설계	개념 스키마 모델링, 논리 스키마 모델링, 물리 스키마 모델링, Class Diagram 설계,
		구현	Web Server 단 개발, Web Client 단 템플릿 설정 및 모듈화
		배포	웹 앱 배포
		테스트	배포 전 베타 서버 테스트
		기타	프로젝트 발표, 프로젝트 중간 보고서 작성, 프로젝트 최종 보고서 작성, 발표 자료(ppt) 제작
문석희	소프트웨어학부(4학년)	요구분석	요구사항 목록, 요구사항 명세서 작성
		설계	개념 스키마 모델링, 논리 스키마 모델링
		구현	Web Client 단 개발 (랜딩, 작품 관람, 검색, 로그인 페이지)
		기타	프로젝트 계획서 작성, 프로젝트 최종 보고서
조은진	소프트웨어학부(4학년)	요구분석	요구사항 목록, 요구사항 명세서 작성
		설계	개념 스키마 모델링, 논리 스키마 모델링
		구현	Web Client 단 개발, 평가 기준 작성 (전시회 목록, 전시회 관람, 선호 페이지)
		기타	프로젝트 회의록 작성, 프로젝트 최종 보고서
주소영	소프트웨어학부 /건축학부(5학년)	요구분석	요구사항 목록, 요구사항 명세서 작성
		설계	UX/UI 설계, UX/UI Design
		구현	XD 제작
		기타	PPT 제작, 프로젝트 일정 도표 작성

6.3 프로젝트 수행 일정

단계	세부단계	3월										4월										5월									
		2	6	8	13	15	20	22	23	27	29	3	5	10	12	17	19	24	26	1	3	8	10	15	17	18	20	24	29	31	
기획	주제선정																														
	자료조사 및 분석																														
	기획서 작성																														
	스토리보드 작성																														
	컨텐츠 수집																														
분석	요구기능정의 목록작성 (시나리오)																														
	요구기능정의서 작성																														
	요구기능명세서 작성																														
설계	Naming Convension 정의																														
	사용자 인터페이스 설계																														
	웹앱 디자인																														
	데이터베이스 설계																														
구현	SW 설계																														
	Web Client 구현																														
	Web Server 구현																														
	세부 테스트 계획																														
테스트																															
배포																															
발표	프로젝트 ppt 작성																														
	리허설																														
	발표																														

단계	세부단계	6월							7월							8월							9월							10월						
		5	7	12	14	19	21	28	5	12	19	26	2	9	16	23	6	11	13	18	20	25	27	27	27	27	27	27	27	27	27	27	27	27	27	27
인공지능 추천 모듈 개발	사용자 데이터 수집																																			
	데이터 선택 및 가공																																			
	인공지능 기반 전시회 추천 모듈 구현																																			
위치기반 추천 모듈 개발	오프라인 전시회 정보 크롤링																																			
	사용자 위치정보 및 전시회 정보 수집																																			
	오프라인 전시회 거리 기반 추천 모듈 구현																																			
적용	Web Server 구현																																			
테스트	세부 테스트 계획																																			
배포																																				
발표	발표 자료 작성																																			
	리허설																																			
	발표																																			

6.4 프로젝트 수행 방법

- Waterfall (전반후) + 애자일 개발 프로세스(Scrum) (후반후)
- 월요일:off-line 회의, 화~금: on-line 회의 (카카오톡/Zoom 화상회의)
- 협업공간: 코드/github, 문서/google drive, ovenio, drawio

7. 참고 문헌

- [1] 오픈부스, <https://www.openbooth.net/about>
- [2] gallery360, <https://www.gallery360.co.kr>
- [3] 국립현대미술관, <https://www.mmca.go.kr/main.do>
- [4] vavafair, <https://www.vavafair.com/home/>
- [5] 변문경, 이정훈 (2021). “온라인 전시 행사에서 개인 맞춤형 추천 시스템 적용 사례 연구 : 산학협력 EXPO를 중심으로.” 한국디지털콘텐츠학회 논문지, 22(4), 655-661.
- [6] 강혜인, 김휘재, 이민구 (2021). “발전 가능성이 많은 작가들을 위한 온라인 개인 전시 플랫폼 어플리케이션 개발 연구.” 한국HCI학회 학술대회, 746-749.
- [7] 정지윤, 김명준 (2018). “사용자의 음악 선호요인 분석을 통한 개인 맞춤형 음악추천모델 연구.” 한국디지털콘텐츠학회 논문지, 19(11), 2041-2047.
- [8] 서봉원 (2016). “콘텐츠 추천 알고리즘의 진화”, 방송트렌드&인사이트. 5: 19-24.