

# 한양대학교 ERICA 아트테라피 홈 페이지 백엔드 개발 프로젝트

Spring Boot 기반 전시회 관리 시스템 구축

MIRI COMPANY



# 목차

## 01 프로젝트 개요

- 프로젝트 배경
- 개발 목표
- 담당 역할
- 주요 기술

## 04 API 설계

- RESTful API
- 예외 처리
- 인증/인가
- Swagger 문서화

## 02 주요 기능

- 전시회 관리
- 작품 관리
- 교수진 관리
- 마이페이지

## 05 개발 과정

- ERD 설계
- 기능 구현
- 테스트
- 배포 및 CI/CD

## 03 기술 스택

- Spring Boot
- JPA, MySQL
- Spring Security
- AWS, Docker

# 프로젝트 개요 및 역할

한양대학교 ERICA 아트테라피 학과의 전시회 관리 시스템 구축을 위한 백엔드 개발 프로젝트입니다. 실제 운영 환경에 적합한 기능 중심의 API를 구현하여 사용자 편의성과 관리 효율성을 높이는 것이 주요 목표입니다.



## 프로젝트 배경 및 목표

- 개발 목표: 전시/작품/교수진 관리 기능이 포함된 학과 백엔드 시스템 구현
- 개발 기간: 2025.05.03 ~ 2025.06.18
- 서버 주소 :
- 주제: 미술치료학과 전시회 사이트 백엔드 구축

## 담당 역할 및 주요 기술

- 담당 파트: 백엔드 전반 (전시회/작품/교수진/마이페이지)
- 주요 기술: Spring Boot, JPA, Spring Security, MySQL
- 배포 환경: AWS EC2, Docker
- 추가 도구: GitHub Actions, Postman

# 주요 기능 개요



## 전시회 및 작품 관리 기능

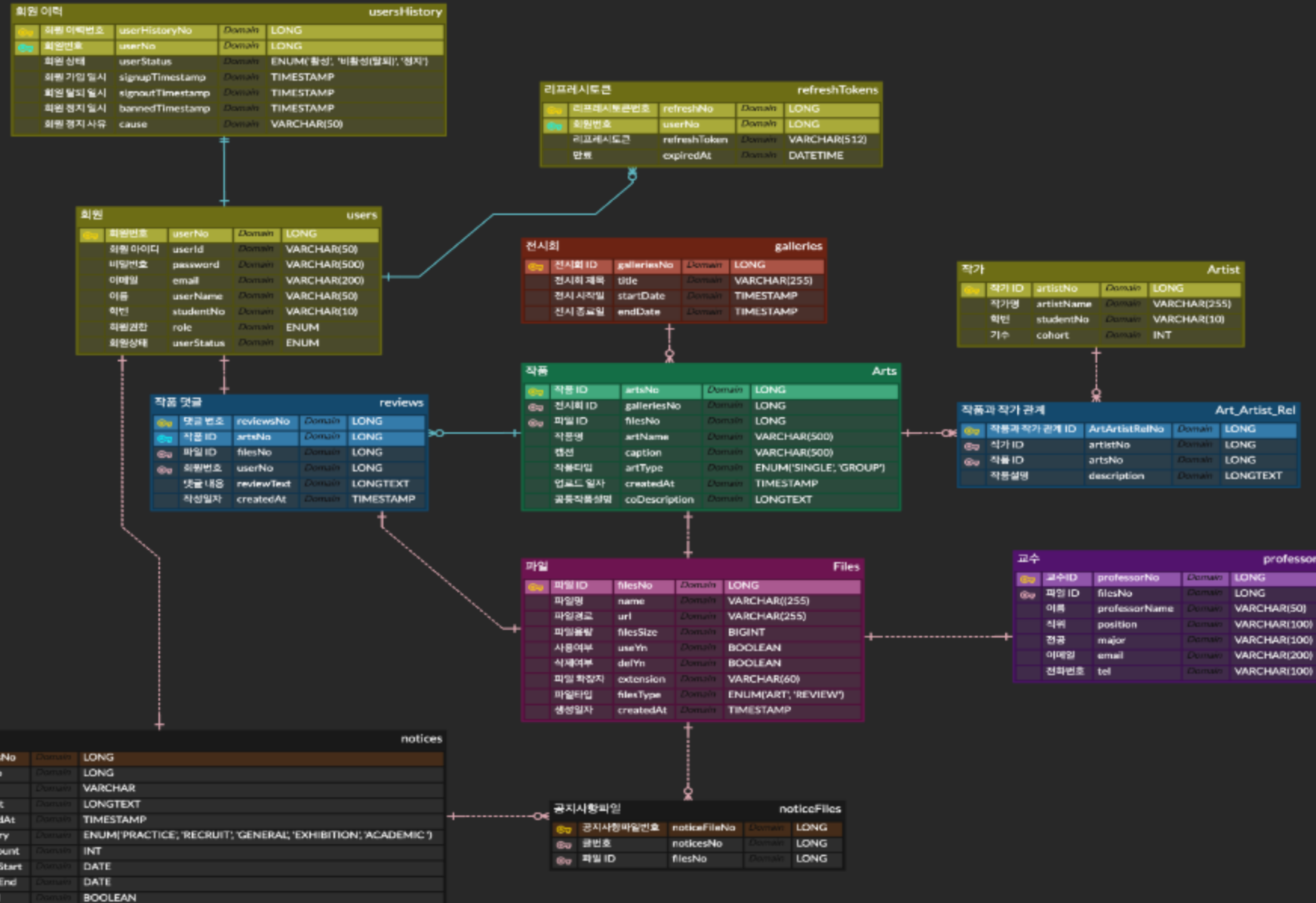
- 관리자 전용 전시회 CRUD API 구현
- 전시 제목, 기간, 장소, 이미지 파일 처리
- 작품 등록(개인/단체), 수정, 삭제, 조회 API
- 다대다 관계 처리 (작품-작가 연결)
- 예외 처리: 날짜 검증, 리소스 미존재 등



## 교수진 관리 및 마이페이지 기능

- 교수 정보 등록, 수정, 삭제, 조회 API 구현
- 이미지 업로드 기능 (files 테이블 연동)
- 사용자 정보 조회 및 수정 기능
- 내 작품, 댓글 목록 조회 API

# ERD



## 데이터베이스 구조 (ERD)

- 주요 테이블: Users, Professors, Galleries, Arts, Artists, Files
- 관계 설계: 전시회-작품(1:N), 작품-작가(N:M)
- ArtArtistRel 테이블로 다대다 관계 구현

## 백엔드 기술 스택

- 프레임워크: Spring Boot
- ORM: JPA (Hibernate)
- 보안: Spring Security
- 데이터베이스: MySQL

## 배포 및 테스트 환경

- 클라우드 서비스: AWS EC2
- 컨테이너화: Docker
- 웹 서버: Nginx
- CI/CD: GitHub Actions
- API 테스트: Swagger, Postman

# 전시회 기능 개요



## 관리자 전용 API

- 전시회 등록, 수정, 삭제, 조회 API 제공
  - ROLE\_ADMIN 권한 필요
  - RESTful API 설계 원칙 준수
- 보안 강화를 위한 인증 절차 적용



## 전시 정보 처리

- 전시 제목, 기간, 장소 정보 관리
- 이미지 파일 업로드 및 처리 기능
  - 기간 역전 검증 (1년에 한개)
- 데이터 일관성 유지를 위한 유효성 검사



## 예외 처리

- GALLERY\_NOT\_FOUND: 전시회 정보 없음
- INVALID\_DATE\_RANGE: 잘못된 날짜 범위
- UNAUTHORIZED\_ACCESS: 권한 없는 접근
- 일관된 예외 처리로 클라이언트 대응 용이



# 전시회 API 명세

**[POST]**  
**/api/admin/galleries**

- 새로운 전시회 등록
- Request: 제목, 시작일, 종료일, 장소
  - Response: 생성된 전시회 정보
  - 관리자 권한 필요

**[GET]**  
**/api/admin/galleries**

- 전체 전시회 목록 조회
- Params: 페이지, 크기, 정렬
  - Response: 전시회 목록, 페이지 정보
  - 모든 사용자 접근 가능

**[GET]**  
**/api/admin/galleries  
/{id}**

- 전시회 상세조회
- PathVariable: galleriesNo (전시회 ID)
  - Response: 제목, 기간, 장소, 이미지 포함 전시 상세 정보
  - 모든 사용자 접근 가능

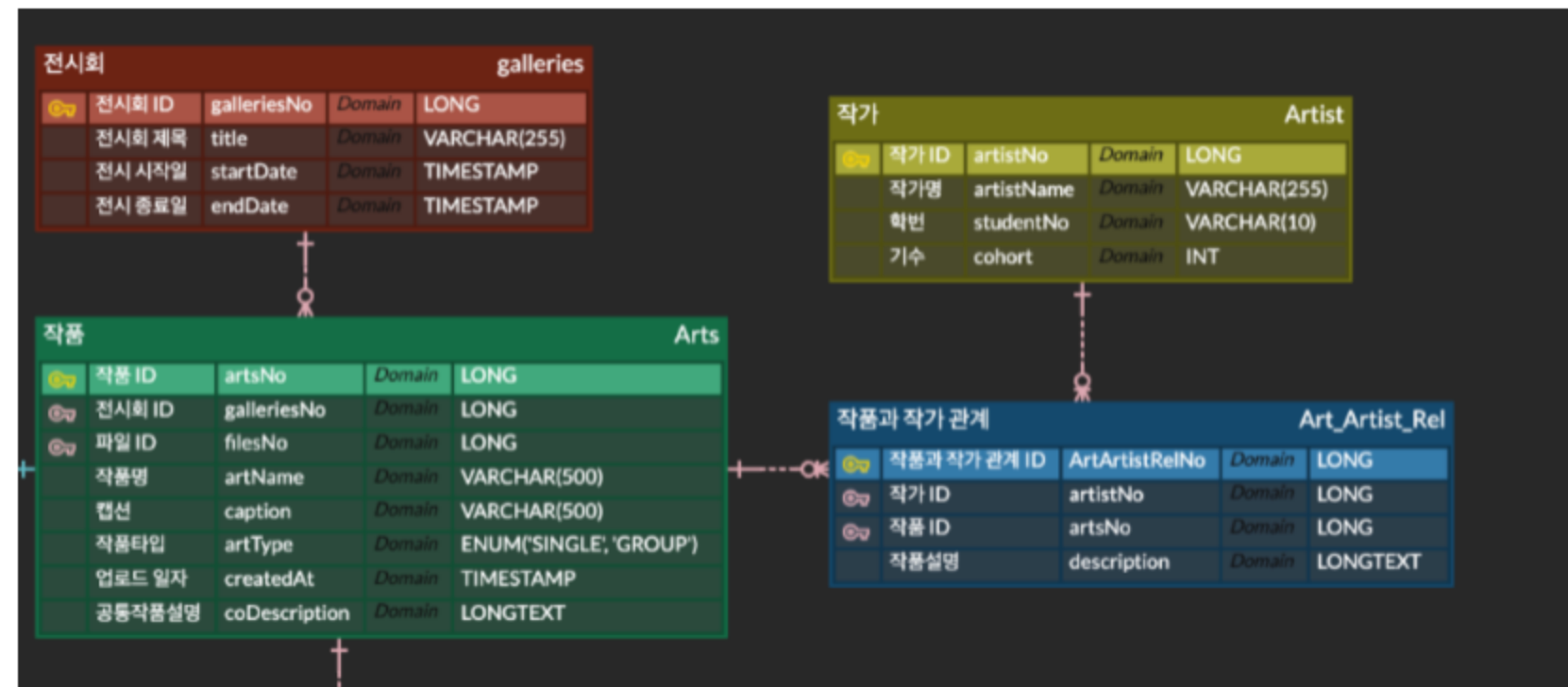
**[PATCH]**  
**/api/admin/galleries  
/{id}**

- 기존 전시회 정보 수정
- PathVariable: 전시회 ID
  - Request: 수정할 필드와 값
  - Response: 수정된 전시회 정보

# 작품 기능 개요

## 작품 관리 API 및 데이터 모델 설계

작품 관리 API는 개인 및 단체 작품의 등록, 수정, 삭제, 조회 기능을 제공합니다. 각 API는 RESTful 설계 원칙을 따르며, 관리자 권한 인증을 통해 보안을 강화했습니다. 데이터 모델은 작품의 다양한 속성을 효율적으로 저장하고 관리할 수 있도록 설계되었습니다.



### 작가-작품 연결 구조

- ArtArtistRel 테이블 사용
- 작품과 작가의 다대다 관계 구현
- 단체 작품의 공동 설명 처리
- 유연한 작가 정보 연결 가능



# 작품 API 명세

**[POST]**  
**/api/admin/arts**

새 작품 등록 (관리자 전용)  
- Request: 작품명, 설명, 유형, 파일번호  
갤러리번호, 작가목록, 공동설명  
- Response: 생성된 작품 정보

**[GET]**  
**/api/admin/arts**

작품 목록 조회  
- Params: 페이지, 크기, 정렬 옵션  
- Response: 작품 목록, 페이지 정보  
- 필터링, 서치 옵션 지원 (작가, 전시회 등)

**[GET]**  
**/api/admin/arts/{id}**

작품 상세 정보 조회  
- PathVariable: 작품 ID  
- Response: 작품 상세 정보, 연관 작가  
갤러리 정보, 파일 정보 포함

**[PATCH]**  
**/api/admin/arts/{id}**

작품 정보 수정 (관리자 전용)  
- PathVariable: 작품 ID  
- Request: 수정할 필드와 값  
- Response: 수정된 작품 정보

# 작품 API 요청 및 인증

## Request Body 구조

- JSON 형식으로 전송
- 필수 필드: artName, artType

## artName (작품명)

- String 타입, 최대 100자
- 고유한 작품명 권장

## caption (설명)

- String 타입, 최대 500자
- 작품에 대한 간단한 설명

## artType (작품 유형)

- Enum 타입 (INDIVIDUAL, GROUP)
- 개인 또는 단체 작품 구분

## ROLE\_ADMIN 인증

- JWT 토큰 기반 인증
- Authorization 헤더에 포함

# 교수진 기능 개요

교수진 관리를 위한 종합적인 API 시스템을 구현했습니다. 이 시스템은 교수 정보의 등록, 수정, 삭제, 조회 기능을 제공하며, 관리자 권한을 가진 사용자만 접근할 수 있도록 설계되었습니다.



## 교수 정보 관리

- 교수 정보: 이름, 직위, 전공, 이메일, 전화번호 포함
- 이미지 파일 업로드 기능: files 테이블과 연동
- 데이터 무결성: 필수 정보 검증 로직 구현
- API 엔드포인트: RESTful 설계 원칙 준수

## 예외 처리 및 보안

- PROFESSOR\_NAME\_REQUIRED: 이름 필드 누락 시 발생
- DUPLICATE\_PROFESSOR: 중복 교수 등록 방지
- 관리자 인증(ROLE\_ADMIN): Spring Security 활용
- 데이터 접근 제어: API 레벨에서 권한 확인

# 교수진 API 명세

**[POST]**  
**/api/admin/professors**

교수진 등록

- Request: 이름, 직위, 전공, 이메일, 전화번호, 이미지 파일 번호 (filesNo)
- Response: 생성된 교수 정보
- 인증: ROLE\_ADMIN 필요

**[GET]**  
**/api/admin/professors**

전체 교수 목록 조회

- Query Params: 없음 (또는 페이징/정렬 옵션)
- Response: 교수 리스트 배열
- 인증: ROLE\_ADMIN 필요

**[GET]**  
**/api/admin/professors**  
**/ {id}**

교수진 상세 조회

- PathVariable: professorNo
- Response: 해당 교수 정보
- 인증: ROLE\_ADMIN 필요

**[PATCH]**  
**/api/admin/professors**  
**/ {id}**

교수진 정보 수정

- PathVariable: 교수 ID
- Request: 수정할 필드와 값
- Response: 수정된 교수 정보

# 마이페이지 기능 개요



## 사용자 정보 관리

- 로그인 사용자 전용 API
- 개인정보 조회 기능
- 정보 수정 기능 제공
- 이메일, 이름 등 수정
- 비밀번호 변경 지원
- 데이터 유효성 검증



## 콘텐츠 관리

- 등록된 작품 목록 조회
- 작성한 댓글 목록 제공
- 콘텐츠 페이지네이션



## 보안 및 인증

- JWT 기반 사용자 인증
- 토큰 유효성 검증
- INACTIVE 상태 변경
- 개인정보 보호 준수
- 접근 권한 관리

# 마이페이지 API 명세

**[GET]**  
**/api/my-page/profile**

나의 정보 조회

- JWT 기반 사용자 인증 후 내 정보 반환
- Response: 이름, 학번, 이메일 등 기본 정보

**[GET]**  
**/api/my-page/profile**

나의 정보수정

- 이름, 학번 이메일수정
- 학번, 이메일 중복 검사
- 이메일 인증

**[PATCH]**  
**/api/my-page/  
my-posts**

나의 작품 조회

- Response: 작품 제목, 전시회명, 썸네일 포함 리스트
- JWT 인증된 사용자 전용

**[GET]**  
**/api/my-page  
/my-reviews**

내가 작성한 댓글 조회

- Query: 키워드 검색 (작품명, 내용), 페이징 지원
- Response: 댓글 내용, 작성일, 연결된 작품명



# 삭제 API 명세

## 전시회

[DELETE]  
/api/admin/galleries/{galleriesNo}

## 작품

[DELETE]  
/api/admin/arts/{artsNo}

## 교수진

[DELETE]  
/api/admin/professors/{professor\_no}

# 추가 기능



## 이미지 파일 업로드

- filesNo를 통한 이미지 파일 참조
- 응답 시 파일 객체 포함
- 다양한 이미지 포맷 지원 (JPEG, PNG 등)



## 관리자 인증

- ROLE\_ADMIN 권한 필요
- JWT 토큰 기반 인증
- API 접근 전 권한 검증



## 예외 처리

- 입력값 유효성 검사
- 중복 데이터 방지
- 존재하지 않는 리소스 요청 처리

# 예외 처리 구조

## 도메인별 ExceptionType(enum )

- Gallery, Arts, Professors, Users 등
- 각 도메인별 고유 예외 정의
- 예: GALLERY\_NOT\_FOUND, INVALID\_DATE\_RANGE
- 일관된 예외 코드 관리 가능

## CustomException 클 래스

- 모든 커스텀 예외의 기본 클래스
- 예외 타입과 메시지 포함
- 예외 발생 시 로그 기록 기능
- 재사용성과 확장성 고려한 설계

## GlobalExceptionHandler

- @ControllerAdvice 사용
- 모든 컨트롤러의 예외를 중앙에서 처리
- CustomException 및 기타 예외 핸들링
- 일관된 예외 응답 형식 제공

## 예외 응답 형식

- 상태 코드: HTTP 상태 코드
- 에러 코드: 커스텀 에러 코드
- 메시지: 사용자 친화적 에러 메시지
- 타임스탬프: 예외 발생 시간

# 개발 과정 및 역할 요약

- 기획: 프로젝트 주제 선정 및 백엔드 기능 분류
- 설계: ERD 설계, API 설계, Swagger 문서화
- 개발: 전시회, 작품, 교수진, 마이페이지 기능 구현
- 배포: AWS EC2, Docker 기반 환경 구성 및 CI/CD 구축