

2025-2

캡스톤디자인과창업프로젝트 그로쓰 2차 보고서

공연의 순간을 말로 그리는 나만의 디지털 티켓북  
**Re:cord**

18팀 코드만두

이세연, 김수민, 조서윤

# I. Team Info

## 1.1 과제명

STT와 GPT-4로 후기 작성을 돕고 AI 티켓 이미지로 기록 공백을 채우는, 공연 마니아들을 위한 공연 경험 아카이빙 서비스 Re:cord

## 1.2 팀 정보 및 구성원

팀 번호·이름: 18팀 코드만두

팀 지도교수: 심재형 교수님

팀 구성원

이름	학번	역할
이세연	2176266	리더, 기획자, BE & AI (이미지생성, 후기질문-GPT), DB 설계 및 구축
김수민	2176071	팀원, 기획자, BE & AI (OCR-STT-GPT Image), DB 설계 및 구축
조서윤	2276304	팀원, 기획자, FE, 디자이너

## 1.3 과제 요약

### 1.3.1 과제 키워드

생성형 AI, 음성인식, 티켓북

### 1.3.2 과제 내용 요약

Re:cord는 STT와 GPT-4로 빠른 후기 작성을 돕고 AI 티켓 이미지를 생성하여 기록 공백을 채우는 공연 마니아들을 위한 공연 경험 아카이빙 어플리케이션이다. 다음의 과정을 통해 하나의 디지털 티켓이 완성된다.

- 1) 공연 정보 작성: 사용자가 관람한 공연 정보를 작성한다.
- 2) 공연 후기 작성: 관람한 공연 후기를 작성한다. 관람한 공연 장르 맞춤 질문을 제공하고, 사용자가 음성으로 남긴 후기를 STT로 텍스트 변환 후, GPT-4로 정리리하여 제안한다.
- 3) AI 티켓 이미지 생성: 공연 정보 및 사용자의 후기를 분석하여 추출한 요소를 활용하여 티켓 이미지를 생성한다.

완성된 티켓의 앞면은 AI 티켓 이미지, 뒷면에는 공연 후기로 구성된다. 이를 통해 기록 공백 없이 모든 공연 후기를 쉽고 빠르게 작성할 수 있다.

## II. Project Summary

### 2.1 문제 정의

한국 공연 시장은 2022년 코로나19의 팬데믹 시기를 지나 티켓 판매액이 꾸준히 증가하여 2024년에는 1.4조 원으로 역대 최고치를 기록하는 등 지속적인 성장을 이어가고 있다.<sup>1</sup> 시장이 성장함에 따라 공연 관람은 점점 대중적인 여가문화로 자리 잡았고, 관람객들 사이에는 공연 후기를 작성하는 문화가 확산되었다. 특히, 다수의 공연을 즐기는 2040 여성들 사이에서는 관람한 티켓을 파일이나 다이어리에 모아 추억을 기록하는 ‘티켓북’ 방식이 널리 활용되고 있다. 그러나 이들을 대상으로 한 자체 설문조사 결과, 관람 후 기록 과정에서 여러 불편을 겪고 있는 것으로 나타났다. 응답자들은 후기를 작성할 적절한 시점을 놓쳐 기억이 흐려지거나, 공연마다 다른 티켓 포맷으로 인한 정리·보관이 어렵고, 티켓이 없는 공연에서는 기록 공백이 생기는 등의 불편을 주로 경험한다고 답했다.

문제 유형	설문 결과
후기 작성 관련	후기를 작성할 적절한 시점을 놓쳐 기억이 흐려짐(75%)
	무엇을 써야할지 떠올리기 어려워 막막함을 느낌(45%)
	글의 완성도에 대한 부담감(45%)
	작성할 공연 수가 많음 (37%)
티켓 보관 관련	티켓 포맷이 다양해서 깔끔하게 정리하기 어려움 (60%)
	보관 규격과 다른 티켓이 있음 (53%)
	예매내역만 있어 시각적으로 남기기 어렵다 (20%)
기록 공백 관련	티켓이 없을 때 특별한 방법이 없어 기록하지 못함 (42%)

\*\* 중복 답변 가능

<sup>1</sup> 양인성(2024.01.16), 「작년 공연 시장 1조2696억원... 영화 처음 넘었다」, 조선일보. <https://www.chosun.com/culture-life/performance-arts/2024/01/16/KWSSXKNOEVEHPILXI35TAHPWSA/>  
서울신문, “국내 공연예술시장 규모 약 1조 원...전체 매출액 증가”, 2024.11.29 ([seoul.co.kr](http://seoul.co.kr))

## 2.2 기존 연구와의 비교

공연 관람 문화의 확산과 함께 관람객들은 여러 방식으로 공연 후기를 기록하고 있다. 이들의 주요 후기 기록 방식을 비교 분석하고 Re:cord가 가진 차별적인 요소를 제시한다.

### 2.2.1 공연후기 기록방식 비교

기존 후기 기록 방식	장점	단점
SNS (Instagram, X(구 트위터) 등)	<ul style="list-style-type: none"><li>- 사진이나 영상과 함께 시각적으로 기록 가능.</li><li>- 다른 이용자와 공유하기 용이함.</li></ul>	<ul style="list-style-type: none"><li>- 다양한 주제의 게시물이 섞여있어 공연 후기만 분리하여 관리하기 어려움.</li><li>- 피드 구조상 시간이 지나면 재열람이 불편함.</li></ul>
네이버 블로그	<ul style="list-style-type: none"><li>- 장문 작성에 적합한 플랫폼으로 감정이나 경험을 충분히 기록할 수 있음.</li><li>- 분량 제한 없이 자유로운 기록 가능.</li></ul>	<ul style="list-style-type: none"><li>- 하나의 포스트 작성에 시간이 많이 소요됨.</li><li>- 무엇부터 작성해야 하는지 막막함을 느낄 수 있음.</li></ul>
오프라인 티켓북	<ul style="list-style-type: none"><li>- 실물 티켓 보관을 통해 관람 경험 물리적 기록 가능.</li></ul>	<ul style="list-style-type: none"><li>- 발권 되지 않은 공연은 기록 불가.</li><li>- 실물 티켓 훼손 및 분실 위험 존재.</li><li>- 보관하는 파일 혹은 다이어리와 규격이 맞지 않는 경우가 있음.</li></ul>

### 2.2.2 Re:cord의 차별성

#### (1) 통합된 기록 환경 제공

티켓과 후기를 하나의 플랫폼에서 관리할 수 있어 관람 경험을 돌아보기 용이하다.

#### (2) 음성 인식 기반 후기 작성 지원

STT(Speech-to-Text)와 GPT-4 기반의 음성 인식 및 문장 정리·요약 기능을 통해 사용자는 별도의 타이핑 없이 빠르고 간편하게 후기를 작성할 수 있다.

#### (3) 디지털 티켓북 형태의 안전한 보관

디지털 기반으로 제공되어 티켓 훼손 및 분실 우려 없이 공연 후기 기록을 안전하게 관리할 수 있다.

## 2.3 제안내용

[2.1 문제 정의]에서 제시한 공연 기억의 휘발성과 후기 완성도에서 발생하는 부담감, 그리고 티켓 보관문제를 해결하기 위해 Re:cord는 다음과 같은 목표를 가진다.

- 1) 모든 티켓을 확보하여 동일한 규격으로 표준화
- 2) 후기 작성 소요시간을 단축
- 3) 작성과정에서의 심리적 부담 완화

따라서, 아래에 제시하는 두 가지 주요 기능으로 기록공백 없이 모든 공연 후기를 빠르고 간편하게 남길 수 있는 디지털 티켓북을 구현하고자 한다.

### 2.3.1 후기 작성 지원

후기 작성 지원 기능은 (1) 장르 맞춤 질문 제공 과 (2) STT + GPT-4 기반 문장 요약·정리 의 두 가지로 구성된다. 공연 후기를 작성하는 과정에서 막막함을 느끼는 사용자를 위해 Re:cord는 관람한 공연 장르에 적합한 맞춤형 질문을 제시함으로써 후기 작성의 가이드 역할을 수행한다. 이때, 사용자가 음성으로 질문에 답변하면 STT(Speech-to-Text)와 GPT-4가 응답 내용을 자동으로 정리·요약하여 자연스러운 문장을 구성한다.

### 2.3.2 AI 티켓 이미지 생성

AI 티켓 이미지 생성 기능은 다음과 같은 사용자를 대상으로 한다.

- 1) 처음부터 발권된 티켓이 없는 공연을 관람한 경우
- 2) 티켓을 훼손 및 분실한 경우
- 3) 나만의 티켓을 소유하고자 하는 경우

이러한 사용자는 AI 티켓 이미지 생성 기능을 통해 티켓을 생성할 수 있다. 공연 정보와 사용자가 작성한 후기에서 추출한 시각적 요소를 반영하여 공연의 순간을 표현한 개인화된 티켓 이미지가 구현된다.

## 2.4 기대효과 및 의의

Re:cord는 사용자가 관람한 모든 공연의 순간을 빠르고 간편하게 기록하고, 생생하게 기억하는 새로운 공연 기록 문화를 만들고자 한다. 이를 통해 공연 관람 경험의 기록 / 보관 / 회상 과정을 디지털화 함으로써 지속 가능한 후기 기록 문화가 형성될 것을 기대한다.

- (1) 기록 공백 해소: 발권 티켓이 없는 공연도 AI 티켓 이미지 생성 기능을 통해 디지털 티켓을 생성·보관할 수 있어 물리적 한계로 인한 기록 공백을 해소할 수 있다.
- (2) 표준화된 티켓 관리: 4:5 비율의 티켓 규격 표준화로 모든 공연 티켓을 일관된 형식으로 저장할 수 있으며 시각적으로 깔끔하고 통일감있는 아카이빙이 가능하다.
- (3) 후기 작성 효율 개선: 장르별 맞춤 질문 제공과 STT+GPT-4 기반 문장 정리·요약 기능을 통해 후기 작성 시간을 단축하고 작성 과정에서의 심리적 부담을 완화한다.
- (4) 관람 데이터 통합 관리: 티켓과 후기가 하나의 플랫폼 내에서 통합 관리되어 과거의 관람 경험을 체계적으로 저장할 수 있다. 이를 바탕으로 사용자는 개인의 공연 관람 패턴과 선호도를 확인하고, 이를 이후 공연 선택에 반영하여 관람 만족도를 높일 수 있다.

## 2.5 주요 기능 리스트

### 2.5.1. OCR 공연 정보 추출

#### (1) Google Vision API를 활용한 텍스트 추출

티켓 이미지를 업로드하고 Google Vision API의 DOCUMENT\_TEXT\_DETECTION 기능을 사용하여 이미지 내 모든 텍스트를 추출한다.

#### (2) GPT-4를 통한 구조화된 데이터 자동 파싱

추출된 OCR 텍스트를 GPT-4에 전송하여 JSON 형식의 구조화된 데이터로 변환하고 자연어 처리를 통해 다양한 날짜/시간 표기를 정규화한다.

#### (3) 지능형 오류 보정 및 누락 데이터 복구

GPT-4가 파싱에 실패한 경우, 정규표현식 기반 로컬 파서가 2차 시도를 수행한다. OCR 오인식 자동 보정 기능을 통해 "14일 5번"을 "14월 5번" 좌석 정보로 수정하고 특정 공연(사의 찬미, 위키드 등) 키워드 감지 시 자동으로 관련 정보를 보완한다.

### 2.5.2. STT + GPT-4 문장 정리 및 요약

#### (1) OpenAI Whisper를 활용한 음성 인식

녹음한 후기 음성 파일을 Whisper API로 변환하여 텍스트로 추출한다. 25MB 이상의 대용량 파일은 ffmpeg를 통해 자동으로 압축(mono, 16kHz, AAC 64k)하여 처리하고 주 사용 언어인 한국어 최적화 설정으로 공연 관련 전문 용어도 정확하게 인식하도록 한다.

#### (2) GPT-4를 통한 자동 문장 정리

구어체의 산만한 음성 기록을 정돈된 문어체로 자동 변환해 핵심 정리본을 생성한다.

원본: "아 진짜... 오늘 그... 레베카 봤는데... 와... 대박... 특히 그 2막에서..."

정리본: "레베카 공연을 관람했습니다. 2막 무도회 장면의 웅장한 무대 연출과 주연 배우의 감정 표현이 특히 인상 깊었습니다. 음악과 조명이 완벽하게 어우러진 작품이었습니다."

### 2.5.3. DB와 GPT-4를 활용한 맞춤형 질문 제공

#### (1) 신규 사용자용 질문 템플릿 제공

사용자의 후기 작성 개수가 3개 미만인 경우, 질문 템플릿 DB(questions\_templates)에서 장르별로 사전 정의된 질문을 랜덤으로 선택하여 제공한다. 5개 카테고리(PERFORMANCE: 연기/공연, MUSIC: 음악, STAGE: 무대/연출, STORY: 스토리/기획의도, OVERALL:전체 평가)별로 적합한 질문을 사전에 DB에 저장해두고, 이후 사용자가 선택한 공연 장르에 따라 자동으로 제공한다.

#### (2) GPT-4를 통한 사용자 맞춤형 질문 생성

사용자의 후기 작성 개수가 3개 이상인 경우, GPT-4를 활용하여 해당 사용자의 기존 후기 데이터를 분석하고, 분석 결과를 바탕으로 맞춤형 질문을 생성한다. 사용자의 후기가 3개 단위로 증가할 때마다 이를 카운팅하여 자동으로 사용자의 후기 패턴을 재분석하여 선호 요소를 업데이트하고, 이를 바탕으로 사용자가 자주 언급하는 요소에 대한 질문을 더 많이 생성한다.



#### 2.5.4. 후기를 활용한 프롬프트로 GPT-image-1 티켓 이미지 생성

##### (1) 후기 기반 시각 요소 자동 추출

GPT-4가 사용자의 공연 후기를 분석하여 주요 시각적 요소를 추출한다.

추출되는 요소	<ul style="list-style-type: none"><li>- emotion (감정): 후기의 정서적 톤</li><li>- theme (주제): 작품의 핵심 메시지</li><li>- setting (배경): 공간적/시간적 설정</li><li>- characters (등장인물): 주요 인물들</li><li>- actions (동작): 핵심 장면의 움직임</li><li>- lighting (조명): 분위기를 결정하는 조명 효과</li></ul>
---------	---

##### (2) 장르별 특화 프롬프트 생성

뮤지컬	예시: "A melancholic musical theater scene about betrayal, set in 19th century London, featuring a noble lady and mysterious gentleman..."
밴드/콘서트	예시: "A moody alternative rock live performance scene, featuring powerful set with emotional lyrics, deep blue and purple lighting..."

##### (3) gpt-image-1을 통해 사용자의 후기 기반 이미지 생성

생성된 프롬프트를 gpt-image-1 API에 전송하여 1080x1350 크기의 4:5 비율 이미지를 생성한다.

##### (4) 이미지 재생성

만약 만들어진 이미지가 사용자의 요구사항을 충분히 반영하지 못한 경우, 사용자가 작성한 요구사항에 따라 이미지를 재생성한다.

## 2.6 서브 기능

### (1) 캘린더

티켓을 생성하면 입력된 공연 정보를 바탕으로 캘린더에 해당 티켓이 자동으로 등록된다.

공연 관람 전에는 후기와 이미지 없이 티켓을 생성하여 공연 일정 관리가 가능하다. 공연 관람 후에는 후기를 작성하고 티켓 이미지를 생성하여 티켓을 완성하고, 공연의 순간을 기록할 수 있다.

### (2) 친구들과 상호작용

사용자가 친구와 팔로잉/팔로워 관계를 맺으면 서로의 티켓과 공개된 후기, 그리고 캘린더를 확인할 수 있다. 일정과 후기를 공유하며 공연 관람 경험을 나누고 소통할 수 있으며 티켓 단위로 좋아요 기능이 제공되어 상호 피드백이 가능하다.

### (3) 히스토리

사용자가 생성한 티켓 데이터를 기반으로 월간·연간 공연 정산을 제공한다. 자주 관람한 공연, 아티스트 등의 통계 정보를 제공하여 관람 패턴과 취향을 확인할 수 있다.

## III. Project Design

### 3.1 요구사항 정의

#### 3.1.1 기능별 상세 요구사항 (Functional Requirements)

기능명	기능 설명	사용 기술 / API / DB	목표 수치
OCR 공연 정보 추출	사용자가 업로드한 티켓 이미지에서 공연 메타 데이터를 추출하여 필드에 자동으로 채움	- Google OCR Vision API - GPT-4 API	- 아티스트 외의 모든 필드 자동 채움 - 5초 이내 완료
장르별 맞춤 질문 제공	공연 장르(뮤지컬/밴드)를 기반으로 questions_templates DB 및 사용자의 기존 후기 데이터를 참조하여 GPT-4가 사용자별 맞춤 질문 생성 및 제공	- GPT-4 API - (AWS RDS) PostgreSQL - Spring Boot 백엔드	- 질문 생성 시간 : 5초 이내 - 질문 개수 : 2-3개
음성 후기 작성 및 AI 정리·요약	사용자의 음성 후기를 STT로 변환 후, GPT-4가 핵심 내용을 정리하여 저장	- OpenAI Whisper API - GPT-4 API - Spring Boot 백엔드 처리	- STT 변환 정확도 : 90% 이상 - 정리·요약본 생성 시간 : 9초 이내 - 적합도 : 사용자 만족도 80% 이상
AI 티켓 이미지 생성	후기 내용을 GPT-4로 분석하여 프롬프트를 생성하고 기존 정보(musical_db, band_db, musical_characters)를 참고하여 gpt-image-1 를 통해 개인화된 티켓 이미지 생성 및 AWS S3 저장	- GPT-4 API - gpt-image-1 - AWS S3 - (AWS RDS) PostgreSQL	- 이미지 생성 성공률 : 85% 이상 - 생성 시간 : 60초 이내 - 프롬프트 키워드 반영률 : 70% 이상 - 이미지 퀄리티 : 7/10 이상



### 3.1.3 UI 분석 / 설계 모델

#### Re:cord 티켓 후기 작성

[Re:cord 티켓 생성 과정]



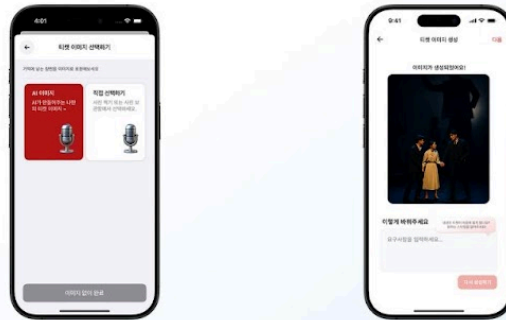
##### 1단계 - 공연 정보 입력하기(OCR)

- 하단의 '티켓 추가' 버튼을 클릭한 뒤, 공연 정보 입력 방식을 선택하세요.
- OCR로 인식된 결과는 직접 수정할 수 있습니다.

##### 2단계 - 공연 후기 작성

- 우측 하단의 버튼을 누르면 음성 녹음이 시작됩니다.
- 화면 상단의 질문을 보고 후기를 작성하세요.
- 질문이 필요 없다면 카드를 삭제할 수 있습니다.
- 후기 작성 후 '후기 요약하기' 버튼을 누르면 작성한 내용이 요약되어 모달에 표시됩니다.
- 요약된 문장은 복사하여 자유롭게 활용할 수 있습니다.

#### Re:cord 티켓 이미지 등록



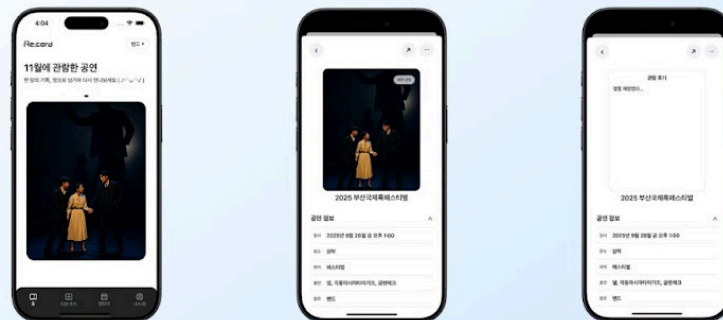
##### 3단계 - 티켓 이미지 생성 방법 선택

- 사용자는 생성형 AI 또는 직접 앨범에서 티켓 이미지를 선택할 수 있습니다.

##### 4단계 - 티켓 이미지 생성 및 재생성

- 완성된 이미지를 확인합니다.
- 이미지가 원하는 방향과 다르게 생성 되었다면 사용자 요청사항을 입력하여 이미지를 재생성할 수 있습니다.

#### Re:cord 티켓 완성



##### 홈 화면

- 한 달 동안 관람한 공연 티켓을 모아볼 수 있습니다.

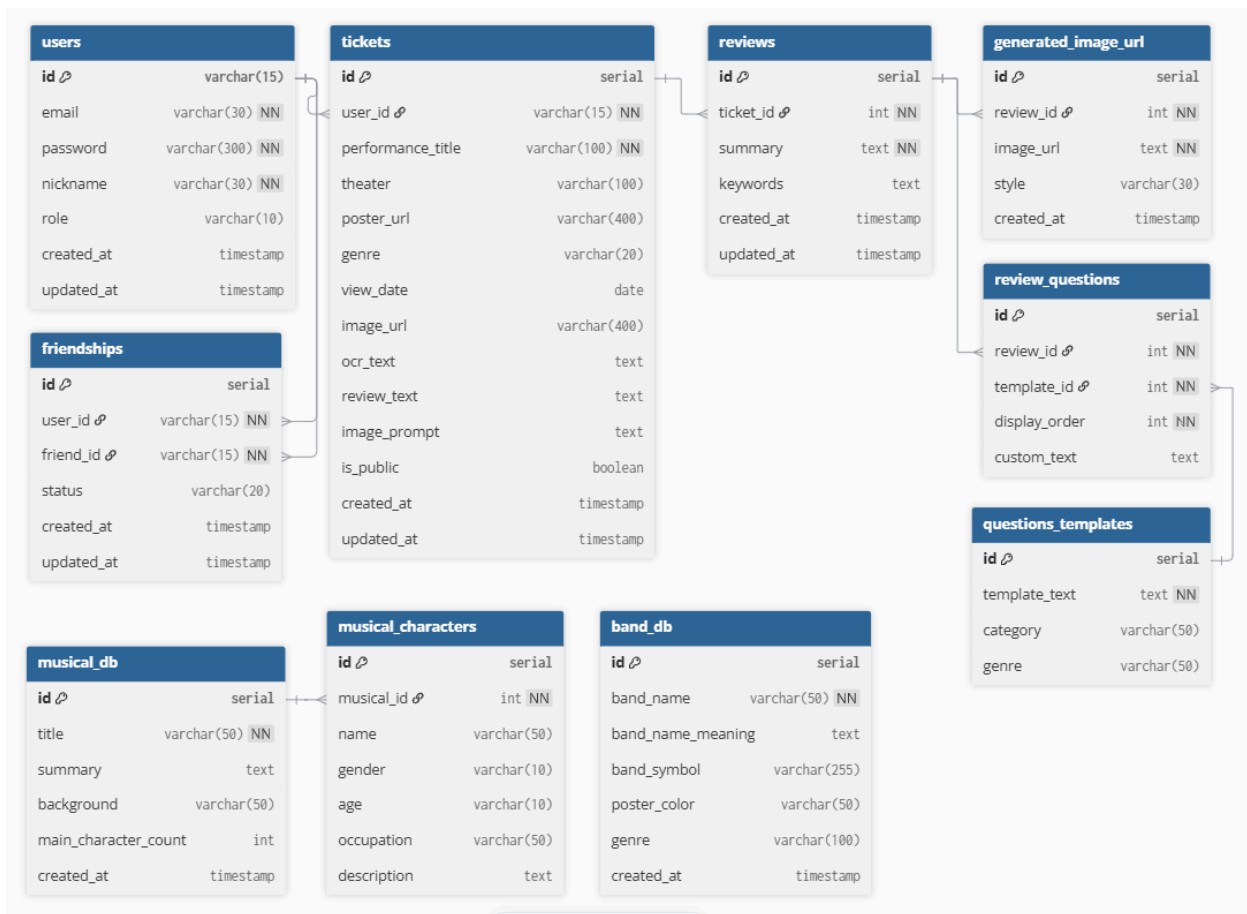
##### 티켓 앞면

- 앞면은 선택한 티켓 이미지로 구성됩니다.
- 티켓 이미지 아래에 공연 정보가 나타납니다.

##### 티켓 뒷면

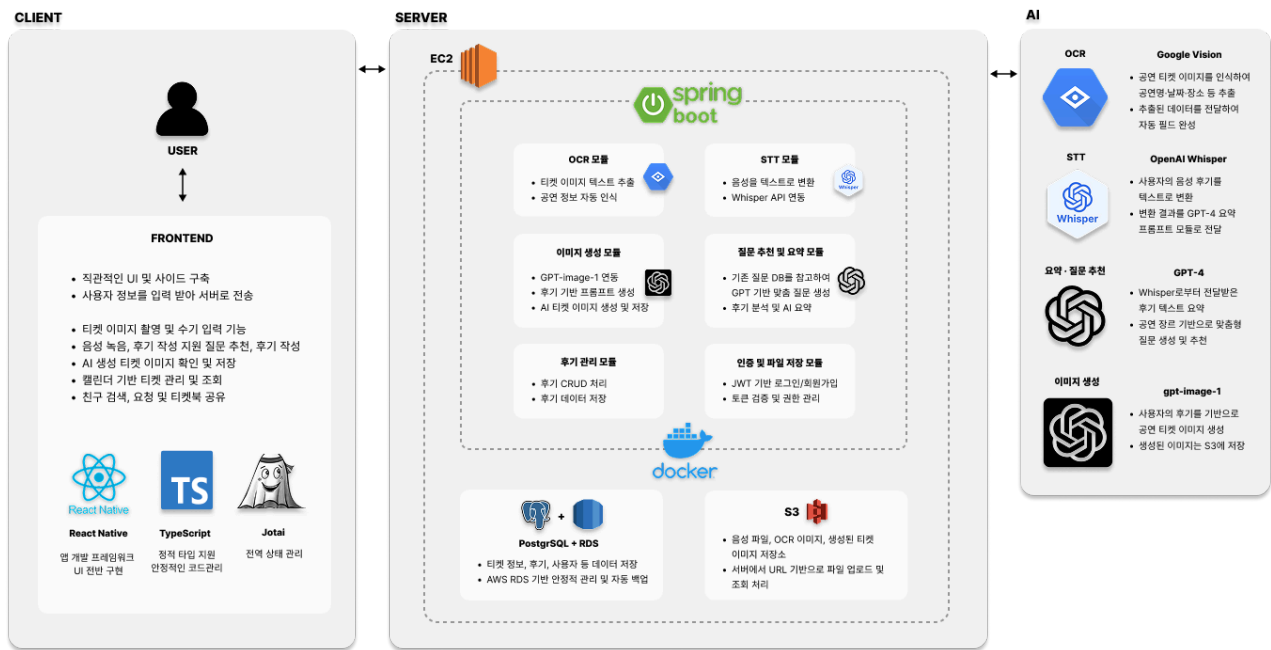
- 뒷면은 후기 텍스트로 구성되어 있습니다.
- 작성한 후기는 우측 상단의 버튼을 통해 수정 가능합니다.

### 3.1.4 E-R 다이어그램 / DB 설계 모델



- 사용자 & 친구 관계 : users, freindships
- 공연 후기, 티켓 기록 및 AI 기반 리뷰, 이미지 생성 : tickets, reviews, generated\_image\_url, review\_questions, questions\_templates
- 이미지 생성 참고용 공연 데이터 : musical\_db, musical\_characters, band\_db

### 3.2 전체 시스템 구성



Re:cord는 사용자가 공연 관람 후기를 작성하고 관리하는 과정에서 AI 기술을 적극적으로 활용하여 음성인식, 이미지 생성, 후기 정리, 맞춤형 질문 추천 등의 기능을 제공하는 다계층 아키텍처로 구성되어 있다.

#### (1) 사용자 및 프론트엔드 계층

시스템의 최종 사용자는 프론트엔드 애플리케이션을 통해 서비스에 접근한다. 프론트엔드는 사용자에게 직관적인 UI를 제공하며, 사용자의 정보를 입력받아 서버로 전송하는 역할을 수행한다. 주요 기능으로는 티켓 이미지 촬영 후 OCR 기반 필드 자동 채움, 음성 녹음, 후기 작성 지원 질문 추천 및 후기 요약/정리, 후기 기반 AI 이미지 생성, 티켓 저장, 캘린더 기반 티켓 관리 및 조회, 팔로잉/팔로워 기반 상호작용(친구 관계 맺기, 티켓북/캘린더 공유) 등이 포함된다.

기술 스택으로는 React Native를 앱 개발 프레임워크 및 UI 구현에 사용하고, TypeScript를 통해 정적 타입을 지원하여 안정적인 코드 관리를 수행하며, Jotai를 전역 상태 관리에 활용한다.

## (2) 백엔드 계층

백엔드는 AWS EC2 인스턴스에서 Spring Boot 프레임워크를 기반으로 동작하며, Docker를 통해 컨테이너화되어 배포 및 관리된다. 백엔드는 여러 모듈로 구성되어 다양한 기능을 수행한다.

OCR 모듈은 티켓 이미지에서 텍스트를 추출하고 공연 정보를 자동으로 인식하는 기능을 담당한다. Google Vision API와 연동하여 공연 티켓 이미지에서 공연명, 날짜, 장소 등을 인식하고, 추출된 데이터를 전달하여 자동 필드를 완성한다.

STT 모듈은 음성 후기를 텍스트로 변환하는 기능을 담당한다. OpenAI Whisper API와 연동하여 사용자의 음성 후기를 텍스트로 변환하고, 변환된 결과를 GPT-4 요약/정리 프롬프트 모듈로 전달한다.

이미지 생성 모듈은 gpt-image-1(OpenAI의 이미지 생성 모델)과 연동하여 후기 기반 프롬프트를 생성하고 AI 티켓 이미지를 생성 및 저장한다. 사용자의 후기를 기반으로 공연 티켓 이미지를 생성하며, 생성된 이미지는 S3에 저장된다.

질문 추천 및 후기 요약/정리 모듈은 기존 질문 DB와 사용자의 기존 후기들을 참고하여 맞춤형 질문을 생성하고, 후기 분석 및 AI 요약을 수행한다. GPT-4와 연동하여 공연 장르 및 사용자 기반 맞춤형 질문을 생성 및 추천하고, Whisper로부터 전달받은 후기 텍스트를 정리한다.

후기 관리 모듈은 후기 데이터에 대한 CRUD(Create, Read, Update, Delete) 작업을 처리하고 후기 데이터를 저장한다.

인증 및 파일 저장 모듈은 JWT(JSON Web Token) 기반의 로그인 및 회원가입 기능을 제공하며, 토큰 검증 및 권한 관리를 담당한다.

## (3) 데이터베이스 및 스토리지 계층

PostgreSQL과 AWS RDS를 기반으로 구성된 관계형 데이터베이스는 티켓 정보, 후기 사용자 등 핵심 데이터를 저장한다. AWS RDS를 통해 안정적인 관리와 자동 백업을 제공하여 데이터의 무결성과 가용성을 보장한다.

AWS S3는 음성 파일, OCR 처리 이미지, 생성된 티켓 이미지 등 다양한 파일을 저장하는 객체 스토리지로 활용된다. 서버에서 URL 기반으로 파일 업로드 및 조회 처리를 수행하여 효율적인 파일 관리를 제공한다.



### 3.3 주요 엔진 및 기능 설계

#### 3.3.1 OCR 공연 정보 추출

##### 1. 개요

공연 티켓이나 포스터 이미지를 업로드하면, 공연 제목, 날짜, 시간, 공연장, 출연자, 좌석정보를 자동으로 추출하기 위해 Google Cloud Vision OCR과 GPT 모델을 결합한 기능을 구현하였다. 이를 위해 OCR 처리, 텍스트 정제, GPT 분석, API 응답을 담당하는 네 가지 주요 모듈을 개발하였다.

##### 2. 모듈별 구현 내용

###### a. OCR 서비스 (OcrService)

- i. 기능: Google Cloud Vision API를 사용해 이미지 속 텍스트를 인식.
- ii. 세부 구현
  1. 업로드된 이미지를 Vision API가 읽을 수 있는 형태로 변환
  2. 텍스트 인식 결과 중 전체 문장을 추출해 문자열로 반환
  3. 오류나 빈 응답 시 기본값(빈 문자열)을 반환하도록 예외 처리

###### b. 텍스트 정규화 및 룰 파싱 (OcrController 내부 유틸)

- i. 기능: OCR에서 얻은 텍스트를 일정한 형식으로 정리
- ii. 세부 구현
  1. 날짜·시간 표기를 통일(예: 2025.11.07 → 2025-11-07, 3:30 PM → 15:30)
  2. 불필요한 개행, 공백 제거
  3. 공연장 이름 후보("센터", "홀", "씨어터", "극장" 등)를 찾아 우선순위 적용
  4. 라인 단위로 분석해 가장 신뢰도 높은 날짜·시간·장소를 선택

###### c. GPT 기반 정보 추출 (GptClient)

- i. 기능: OCR 결과에서 필요한 공연 정보를 AI가 자동 정리
- ii. 세부 구현
  1. GPT에게 "공연 정보만 JSON 형태로 출력하라"는 지시문을 보냄

2. 모델이 분석한 결과를 받아 title, date, time, venue, artist 5개 항목으로 변환
3. 불완전한 응답이나 비정상 JSON은 안전하게 처리하고 기본 구조 유지

iii. 사용된 프롬프트 예시

"다음 공연 티켓/포스터에서 제목(title), 날짜(date), 시간(time), 장소(venue), 출연자(artist)를 찾아 JSON 형식으로만 출력하시오. 설명 없이 순수 JSON만 출력하세요."

d. API 컨트롤러 & DTO (OcrController, PerformanceInfo)

- i. 기능: 전체 과정을 하나로 연결하는 웹 엔드포인트 구현
- ii. 세부 구현
  1. 업로드된 이미지를 받아 OCR → 정규화 → GPT 분석을 순서대로 수행
  2. 결과를 PerformanceInfo DTO에 담아 JSON으로 반환
  3. 테스트용으로 OCR 원문만 반환하는 간단한 엔드포인트도 추가

3. 출력 예시

```
{  
  "title": "뮤지컬 <디어 에반 헨슨>",  
  "date": "2025-11-22",  
  "time": "19:30",  
  "venue": "OO아트센터 대극장",  
  "artist": "홍길동, 김가영"  
}
```

### 3.3.2 이미지 생성

#### 1. 개요

사용자가 작성한 공연 후기를 바탕으로 공연의 분위기와 장면을 시각화한 이미지를 자동으로 생성하기 위해 OpenAI의 이미지 생성 모델(gpt-image-1)과 GPT-4를 결합한 기능을 구현하였다. 이를 위해 프롬프트 생성, 후기 분석, 이미지 생성, API 응답을 담당하는 네 가지 주요 모듈을 개발하였다.

#### 2. 모듈별 구현 내용

##### a. 프롬프트 생성 서비스(PromptService)

- i. 기능: 공연 정보와 후기 내용을 분석하여 이미지 생성에 적합한 영어 프롬프트를 생성
- ii. 세부 구현
  1. 장르 분기 처리 : 뮤지컬과 밴드 장르에 따라 서로 다른 프롬프트 템플릿 적용
  2. 공연 DB 연계 : MusicalDbRepository와 BandDbRepository에서 작품 정보(요약, 배경, 캐릭터, 색상 등) 조회
  3. 후기 내용 분석 : ReviewAnalysisService를 통해 후기에서 감정, 주제, 배경, 조명, 행동, 캐릭터 정보 추출
  4. 한국어 영문화 : 감정, 시대적 배경, 장소, 관계, 액션 등 한국어 키워드를 영어로 매핑하여 변환
  5. 프롬프트 압축 : OpenAIChatService를 사용하여 긴 프롬프트를 자연스러운 영어 2~3문장으로 압축
  6. 길이 제한 처리 : 문장 단위로 최대 900자 제한을 넘지 않도록 안전하게 절단

##### b. 후기 분석 서비스(ReviewAnalysisService)

- i. 기능: 공연 후기 텍스트에서 이미지 생성에 필요한 구조화된 정보를 추출
- ii. 세부 구현
  1. GPT 기반 분석 : OpenAI Chat API를 호출하여 후기에서 감정(emotion), 주제(theme), 배경(setting), 관계(relationship), 행동(actions), 캐릭터(character1~5), 조명(lightning) 정보 추출

2. JSON 파싱 : GPT 응답에서 순수 JSON 부분만 추출하여 Map 형태로 변환
3. 오류 처리 : JSON 파싱 실패 시 기본 구조를 유지하며 오류 정보 포함

c. 이미지 생성 서비스 (Gpt1PicService)

- i. 기능: OpenAI 이미지 생성 API를 호출하여 실제 이미지 생성
- ii. 세부 구현
  1. 단일 이미지 생성 : 항상 1장의 이미지만 생성하며, 인스타그램 4:5 비율(1080x1350)로 고정됨
  2. 프롬프트 길이 제한 : 최대 900자 까지만 사용하도록 자동 절단
  3. WebClient 통신 : 비동기 방식으로 OpenAI API 호출, 60초 타임아웃 설정
  4. 재시도 로직 : 실패 시 백오프 전략으로 최대 2회 재시도
  5. 응답 처리 : data[0].url을 우선 사용, 없으면 b64\_json 처리

d. API 컨트롤러 & DTO (ImageController, ImageResponse)

- i. 기능: 전체 이미지 생성 과정을 하나로 연결하는 웹 엔드포인트 구현
- ii. 세부 구현
  1. 기본 모드 : PromptRequest(장르, 제목, 후기, 추가 스타일 요청 등)를 받아 프롬프트 생성 → 이미지 생성 순서로 수행
  2. 재생성 모드 : basePrompt가 제공되면 프롬프트 재생성 없이 기존 프롬프트에 추가 요청만 합쳐서 즉시 이미지 생성
  3. 장르 검증 : 뮤지컬/밴드가 아닌 경우 400 에러 반환(재생성 모드는 제외)
  4. 텍스트 제거 규칙 : 모든 프롬프트에 'No caption, no letters, no words, no logos, no watermarks' 규칙 자동 추가
  5. 결과 반환 : ImageResponse에 생성된 프롬프트와 이미지 URL을 담아 JSON으로 반환

### 3. 출력 예시

```
{  
  "prompt": "An emotional musical theater scene about love and sacrifice, set in  
  1940s Korea and depicting the relationship between a poet and a gisaeng,  
  featuring exactly 2 characters only: Yoon Hee (a mid-20s female gisaeng) and Seo  
  Jin (a mid-20s male poet). With singing and acting, under dramatic stage lighting.  
  No captions, no letters, no words, no logos, no watermarks.",  
  "imageUrl": "https://oaidalleapoprodscus.blob.core.windows.net/..."  
}
```

#### 3.3.3 후기 작성 지원 질문 생성

##### 1. 개요

사용자가 후기 작성 화면에 진입했을 때, 화면 상단에 후기 작성에 도움이 될 수 있는 질문들을 미리 표시하여 사용자가 어떤 내용을 작성하면 좋을지 가이드를 제공하기 위해 구현한 기능이다. 사용자의 후기 작성 이력에 따라 두 가지 방식으로 질문을 제공한다.

후기 데이터가 3개 미만인 신규 사용자는 질문 템플릿 DB에서 랜덤으로 질문을 선택하여 제공하고, 후기 데이터가 3개 이상인 사용자는 GPT 모델을 활용하여 해당 사용자의 기존 후기 데이터를 분석하여 맞춤형 질문을 생성한다. 이를 위해 질문 템플릿 관리, 사용자 후기 분석, 맞춤형 질문 생성, API 응답을 담당하는 네 가지 주요 모듈을 개발하였다.

##### 2. 모듈별 구현 내용

###### a. 질문 템플릿 관리 서비스 (QuestionTemplateService)

- i. 기능: 신규 사용자(후기 3개 미만)를 위한 질문 템플릿 DB에서 랜덤 질문 선택
- ii. 세부 구현

1. 사용자 후기 개수 확인 : ReviewRepository를 통해 해당 사용자의 작성한 후기 개수를 조회

2. 3개 미만 분기 처리 : 후기 개수가 3개 미만인 경우 질문 템플릿 DB(questions\_templates)에서 질문 선택
3. 장르별 질문 필터링 : 사용자가 작성하려는 티켓의 장르(뮤지컬/밴드)에 맞는 질문만 조회
4. 랜덤 선택 : 조회된 질문 템플릿 중 랜덤으로 선택하여 반환
5. 카테고리 다양화 : PERFORMANCE(연기/공연), MUSIC(음악), STAGE(무대/연출), STORY(스토리/내용/밴드의 경우 기획), OVERALL(전체 평가) 등 다양한 카테고리에서 균형있게 선택

b. 사용자 후기 분석 서비스 (UserReviewAnalysisService)

- i. 기능: 사용자의 기존 후기 데이터를 분석하여 사용자가 주로 어떤 요소에 관심이 있는지 파악
- ii. 세부 구현
  1. 사용자 후기 개수 확인 : 사용자의 후기가 3개 단위(3개, 6개, 9개...)로 증가할 때마다 업데이트
  2. 기존 후기 수집 : ReviewRepository를 통해 해당 사용자가 작성한 모든 후기 텍스트를 조회
  3. GPT 기반 분석 : OpenAI Chat API를 호출하여 사용자의 후기 패턴을 분석
  4. 선호 요소 추출 : 사용자가 주로 언급하는 요소를 파악(예 : 뮤지컬 장르의 경우, 배우의 연기에 집중하는지 / 줄거리 분석을 즐기는지 / 연출 요소에 관심이 많은지 등)
  5. JSON 형식 저장 : 분석 결과를 JSON 형태로 변환하여 User 엔티티의 선호 요소 속성에 저장
  6. 주기적 업데이트 : 사용자 후기가 3개 단위로 증가할 때마다 자동으로 재분석하여 최신 정보로 업데이트
  7. 사용한 프롬프트 예시 (뮤지컬)

다음은 한 사용자가 작성한 여러 공연 후기들입니다. 이 후기들을 분석하여 해당 사용자가 후기를 작성할 때 주로 어떤 요소에 관심을 두는지 파악하세요.

분석할 요소 :

- 배우/출연자의 연기, 가창, 호흡에 대한 관심도
- 작품의 줄거리, 스토리, 주제에 대한 관심도
- 연출, 무대, 조명, 의상 등 시각적 요소에 대한 관심도
- 음악, 사운드 등 청각적 요소에 대한 관심도
- 전체적인 감상과 평가에 대한 관심도

출력 형식은 반드시 JSON만으로 주세요. 각 요소별로 관심도 수준(높음/중간/낮음)과 주로 언급하는 키워드를 포함하세요.

후기 목록 : [사용자가 작성한 모든 후기 텍스트]

c. 맞춤형 질문 생성 서비스 (PersonalizedQuestionService)

- 기능: 사용자의 선호 요소 정보를 바탕으로 GPT 모델을 활용하여 맞춤형 질문 생성
- 세부 구현
  - 사용자 선호 요소 조회 : User 엔티티에서 해당 사용자의 선호 요소 정보를 조회
  - GPT 기반 질문 생성 : OpenAI Chat API를 호출하여 사용자의 선호 요소를 반영한 맞춤형 질문 생성
  - 선호 요소 반영 : 사용자가 자주 언급하는 요소에 대한 질문을 더 많이, 자세히 생성
  - 질문 범주 균형 : 선호 요소를 중심으로 하되, 다양한 범주의 질문도 포함하여 균형 유지
  - JSON 형식 강제 : 시스템 프롬프트를 통해 순수 JSON 배열만 출력하도록 유도
  - 파싱 안전성 : JSON 파싱 실패 시 응답에서 배열 부분만 추출하여 재시도, 최종 실패 시 템플릿 질문으로 풀백
  - 사용한 프롬프트 예시

[User 엔티티에 저장된 선호 요소 JSON 데이터]

출력 형식은 반드시 JSON 배열(List<String>)만으로 주세요.

설명/서론/번호/코드블록 금지.

질문 톤 : 대화형(예 : "~은 어땠나요?", "기억나는 ~ 가  
있나요?").

질문 생성 가이드 :

- 사용자가 높은 관심도를 보이는 요소에 대한 질문을 더 많이 생성
- 사용자가 자주 언급하는 키워드를 자연스럽게 질문에 포함
- 선호 요소 외에도 다른 범주의 질문도 일부 포함하여 균형 유지

d. API 컨트롤러 & DTO (QuestionController, QuestionResponse)

- i. 기능: 후기 작성 화면 진입 시 질문을 조회하여 반환하는 웹 엔드포인트 구현
- ii. 세부 구현
  1. 사용자 확인
  2. 후기 개수 확인 : 사용자의 작성한 후기 개수를 확인하여 분기 처리
  3. 3개 미만 처리 : QuestionTemplateService를 호출하여 템플릿 DB에서 랜덤 질문 선택
  4. 3개 이상 처리 : PersonalizedQuestionService를 호출하여 맞춤형 질문 생성
  5. 비동기 분석 트리거 : 사용자 후기가 3개 단위로 증가할 때마다(매 티켓 생성 및 저장 시마다 카운팅) 백그라운드에서 UserReviewAnalysisService를 호출하여 선호 요소 업데이트
  6. 결과 반환 : QuestionResponse에 생성된 질문 목록을 담아 JSON으로 반환



### 3. 출력 예시

#### a. 신규 사용자(후기 3개 미만) - 템플릿 질문(뮤지컬 장르)

```
{
  "questions": [
    "이 공연에서 가장 인상 깊었던 장면은 무엇인가요?",
    "배우들의 연기력은 어떠했나요?",
    "음악과 무대 연출은 만족스러웠나요?",
    "주인공의 성장 과정이 잘 드러났나요?",
    "무대 조명이나 배우의 동선, 무대장치 이용 등 연출적 요소 중 특별히 기억에 남는 것이 있나요?",
    "이 작품의 주제가 개인적으로 어떤 의미로 다가왔나요?"
  ]
}
```

#### b. 기존 사용자(후기 3개 이상) - 맞춤형 질문

```
{
  "questions": ["주인공 배우의 가창력과 연기력에서 인상깊은 부분이 있었나요?",
    "배우들의 호흡과 앙상블이 작품의 몰입도에 어떤 영향을 주었나요?",
    "주요 배우들의 캐릭터 해석이 작품의 주제 전달에 어떻게 기여했나요?",
    "이번 공연에서 배우들의 연기 스타일이 이전 관람과 비교해서 어떤 차이점이 있었나요?",
  ]
}
```

### 3.3.4 STT(Speech-to-Text) 기능 구현 진행 현황

#### 1. 진행 상황 개요

사용자가 직접 녹음한 공연 후기를 텍스트로 변환하기 위해 STT(Speech-to-Text) 기능을 개발 중이다.

초기에는 Google Cloud Speech-to-Text API를 사용하여 음성 파일을 문자로 변환하고, 이후 해당 결과를 요약 모듈로 연동하는 구조를 계획하였다. 현재는 구글 STT 대신 OpenAI Whisper API를 도입하기로 결정하였으며, 관련 검증 및 적용 단계에 있다.

#### 2. 일정 지연의 원인

STT 기능은 당초 1차 개발 이후 빠르게 검증할 예정이었으나, 실제 검증이 여러 차례 미뤄지면서 일정이 지연되어 Google STT의 제약사항 파악이 늦어지게 되었다.

- 1분 길이 제한(동기식 처리): Google STT의 동기식 API는 최대 1분까지만 인식 가능하며, 공연 후기처럼 긴 길이의 녹음 파일은 처리할 수 없었다.
- 비동기식 처리의 과도한 지연: 비동기 모드로 전환할 경우 대기 시간이 지나치게 길어, 사용자 경험과 실시간 처리 흐름에 적합하지 않았다.

이러한 문제를 예상보다 늦게 확인하게 되면서 원했던 결과를 내기 어려움을 뒤늦게 알았고 Whisper 모델로 전환하였지만 이 검증 또한 일정에 차질이 발생하여 미뤄졌다.

#### 3. 현재 단계

현재는 Google STT에서 OpenAI Whisper로의 전환 작업을 진행 중이다.

백엔드에서는 Whisper API를 WebClient 기반으로 호출하는 로직을 구현하였으며, 이제 실제 음성 파일 업로드 및 텍스트 변환 결과를 검증하는 단계에 있다.

#### 4. 향후 계획

현재는 Whisper를 이용한 음성으로 텍스트 변환 결과가 효과적인지 검증하는 단계에 있으며, 이후에는 변환된 텍스트를 GPT 기반 요약 및 정리 모듈과 연동할 예정이다. 최종적으로는 STT로 음성을 인식하고, GPT로 정리한 뒤, 이를 데이터베이스에 자동 저장하는 전체 통합 흐름을 완성하는 것을 목표로 하고 있다.

### 3.4 주요 기능의 구현

사용자가 새로운 티켓을 생성하면 공연명, 일시, 출연진 등의 기본 정보를 입력하게 되며, 이때 OCR과 GPT-4를 활용하여 티켓 또는 포스터의 공연 정보를 자동으로 인식하고 정리한다. 이후 공연의 장르에 따라 질문 데이터베이스와 GPT-4가 맞춤형 질문을 제시하여 사용자가 보다 쉽게 후기를 작성할 수 있도록 지원한다. 사용자가 음성으로 후기를 남기면 STT를 통해 텍스트로 변환되고, GPT-4가 핵심 내용을 분석하여 정리한다. 마지막으로, 이렇게 정리된 후기 텍스트를 기반으로 프롬프트를 형성하여 gpt image로 커스텀 티켓 이미지를 생성함으로써 하나의 완성된 Re:cord 티켓이 만들어진다. 만약 생성된 이미지가 원하는 방향과 다를 경우, 사용자는 재생성 단계에서 추가 요청사항을 입력할 수 있고 이는 2차 프롬프트에 반영 된다.

#### 3.4.1 OCR 공연 정보 추출

##### 1. 전체 흐름

- a. 사용자가 이미지 파일 업로드
- b. OCR 모듈이 이미지에서 텍스트를 인식
- c. 정규화 모듈이 날짜·시간 등을 표준 형태로 정리
- d. GPT 모듈이 문맥을 분석해 필요한 공연 정보를 추출
- e. API 컨트롤러가 결과를 JSON 형태로 반환

##### 2. 처리 순서 요약

- a. 사용자가 이미지 업로드
- b. OcrService가 Vision API로 텍스트 추출
- c. 텍스트 정규화(날짜/시간/장소 포맷 통일)
- d. GptClient가 필요한 공연 정보만 구조화
- e. OcrController가 JSON 형태로 결과 반환

### 3.4.2 이미지 생성

#### 1. 전체 흐름

- a. 사용자가 공연 정보(장르, 제목, 후기, 추가 스타일 요청 등)를 입력
- b. PromptService가 장르에 따라 DB에서 공연 정보 조회
- c. ReviewAnalysisService가 후기 텍스트를 분석하여 영어 프롬프트 생성
- d. OpenAIChatService가 프롬프트를 압축
- e. Gpt1PicService가 OpenAI 이미지 생성 API를 호출하여 이미지 생성
- f. ImageController가 생성된 프롬프트와 이미지 URL을 JSON 형태로 반환

#### 2. 처리 순서 요약

- a. 사용자가 이미지 생성 요청(장르, 제목, 후기, 추가 스타일 등)
- b. PromptService가 장르별로 공연 DB 정보 조회 및 후기 분석 결과 결합
- c. 프롬프트 영문화 및 키워드 변환
- d. OpenAIChatService가 프롬프트를 자연스러운 문장으로 압축
- e. 프롬프트 길이 제한 처리(최대 900자, 문장 단위 절단)
- f. Gpt1PicService가 OpenAI 이미지 생성 API 호출(4:5 비율, 1장)
- g. ImageController가 프롬프트와 이미지 URL을 JSON으로 반환

### 3.4.3 후기 작성 지원 질문 생성

#### 1. 전체 흐름

- a. 사용자가 후기 작성 화면에 진입
- b. QuestionController가 사용자 확인 후 ReviewRepository를 통해 사용자의 후기 개수 확인
- c. 후기 개수가 3개 미만인 경우 : QuestionTemplateService가 질문 템플릿 DB에서 장르별 랜덤 질문 선택
- d. 후기 개수가 3개 이상인 경우 : PersonalizedQuestionService가 User 엔티티의 선호 요소 정보를 조회하여 맞춤형 질문 생성
- e. 사용자 후기가 증가한 경우(3개 단위로 카운팅) : UserReviewAnalysisService가 백그라운드에서 사용자 후기 재분석 및 선호 요소 업데이트
- f. QuestionController가 생성된 질문 목록을 JSON 형태로 반환하여 화면 상단에 표시

## 2. 처리 순서 요약

### a. 신규 사용자(후기 3개 미만) 흐름

- i. 사용자가 후기 작성 화면 진입
- ii. QuestionController가 사용자 확인
- iii. ReviewRepository를 통해 사용자의 후기 개수 확인(3개 미만)
- iv. QuestionTemplateService가 티켓의 장르에 맞는 질문 템플릿 조회
- v. 랜덤으로 질문 선택(카테고리 다양화)
- vi. QuestionController가 QuestionResponse로 질문 목록 반환

### b. 기존 사용자(후기 3개 이상) 흐름

- i. 사용자가 후기 작성 화면 진입
- ii. QuestionController가 사용자 확인
- iii. ReviewRepository를 통해 사용자의 후기 개수 확인(3개 이상)
- iv. User 엔티티에서 사용자의 선호 요소 정보 조회
- v. PersonalizedQuestionService가 GPT 모델을 호출하여 맞춤형 질문 생성
- vi. 선호 요소를 반영한 질문 생성
- vii. QuestionController가 QuestionResponse로 질문 목록 반환