

REST MUSIC

팀프로젝트
결과보고서

5조
팀장 : 김영웅
김동환 김현우 박성준 장동필 정윤정



프로젝트 소개

01 기획의도

02 팀원소개



개발 환경

01 개발도구환경



주요 기능

01 플로우차트

02 기능상세설명



DB 설계

01 ERD

02 DB설명



시연 및 후기

01 프로젝트 시연

02 문제해결

03 후기

STEP 1

프로젝트 소개

기획 의도

REST # API

01

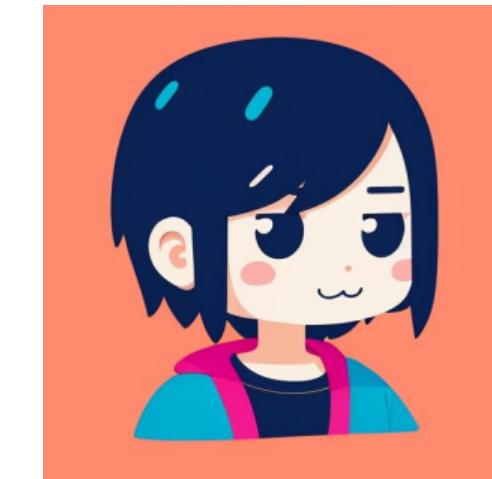
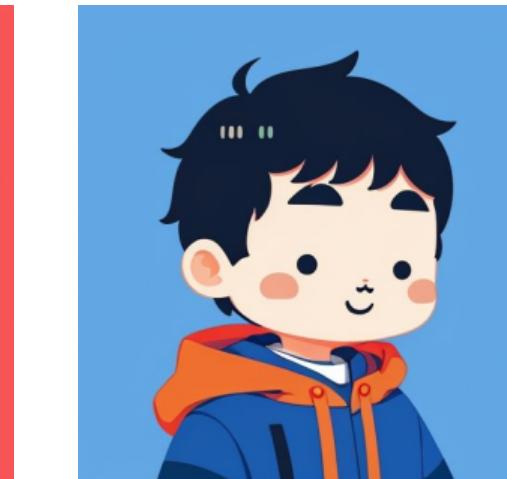
본 프로젝트의 목적은 수업 시간에 배운 내용을 최대한 활용하면서, 추가적으로 배우지 않은 부분도 응용하여 개발 및 구현하는 데 중점을 두었습니다.

02

"Rest"라는 프로젝트 이름은 사용자에게 편안하고 쉬운 음악 감상 경험을 제공함과 동시에, 기술적으로는 안정적이고 효율적인 서비스 구조를 구현하는 것을 목표로 합니다.

팀원 소개

REST # TEAM



정윤정

김영웅

김동환

김현우

장동필

박성준

회원 정보 관리
메인 페이지

음원 재생 페이지
앨범 상세 페이지
데이터베이스 설계

음원 상세 페이지
음원 검색
결제 API 호출

음원 차트 페이지
회원 정보 관리

앨범 페이지
메인 페이지

마이 페이지
플레이리스트 페이지
인물 상세 페이지



정윤정



로그인

아이디를 입력해주세요

비밀번호를 입력해주세요

아이디 저장

[아이디 찾기](#) | [비밀번호 찾기](#)

로그인

[회원가입](#)



會員註冊

아이디를 입력해주세요

비밀번호를 입력해주세요

비밀번호를 한번 더 입력해주세요

이름을 입력해 주세요

예: rest@rest.com **인증번호 받기**

인증번호를 입력해주세요 **인증번호 확인**

닉네임을 입력해주세요

힌트 질문

힌트 답변을 입력해주세요

가입

[로그인](#)



아이디 찾기

이름

이메일

조회

비밀번호 찾기



비밀번호 찾기

아이디

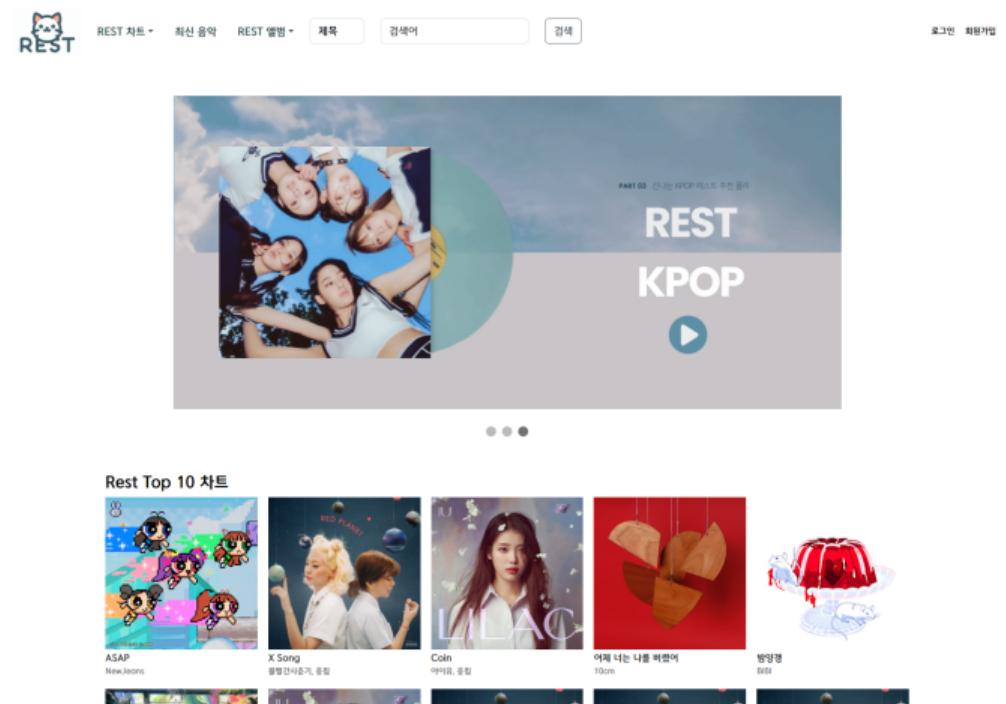
이메일

힌트 질문

힌트 답변

조회

아이디 찾기

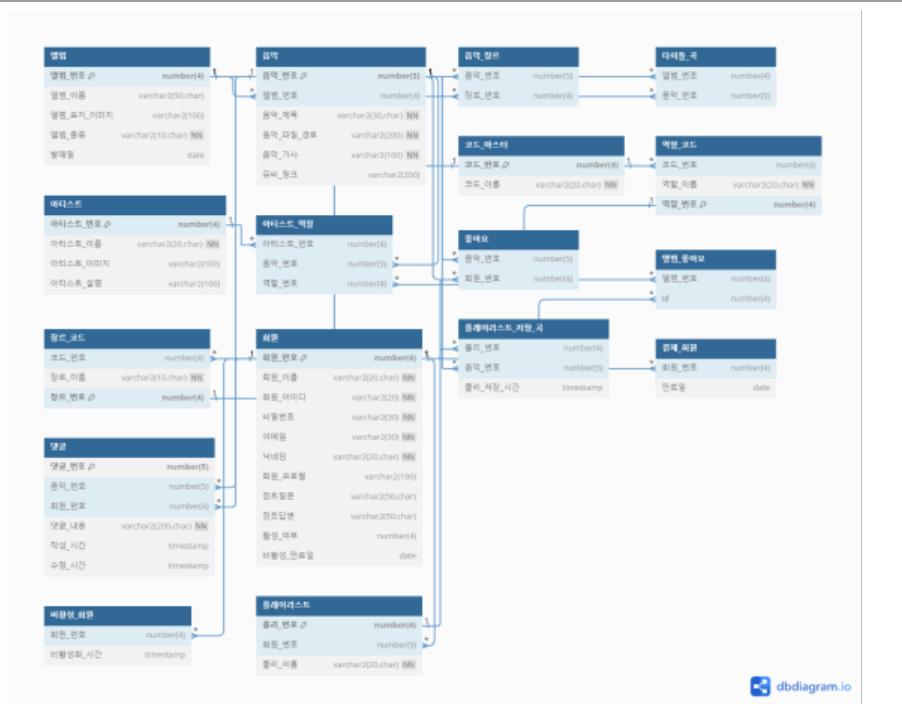


로그인, 회원가입

메인 페이지

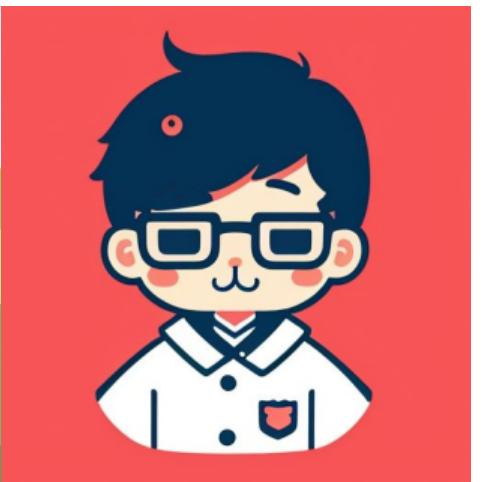


김영웅



앨범 상세 페이지, 음원 재생 바

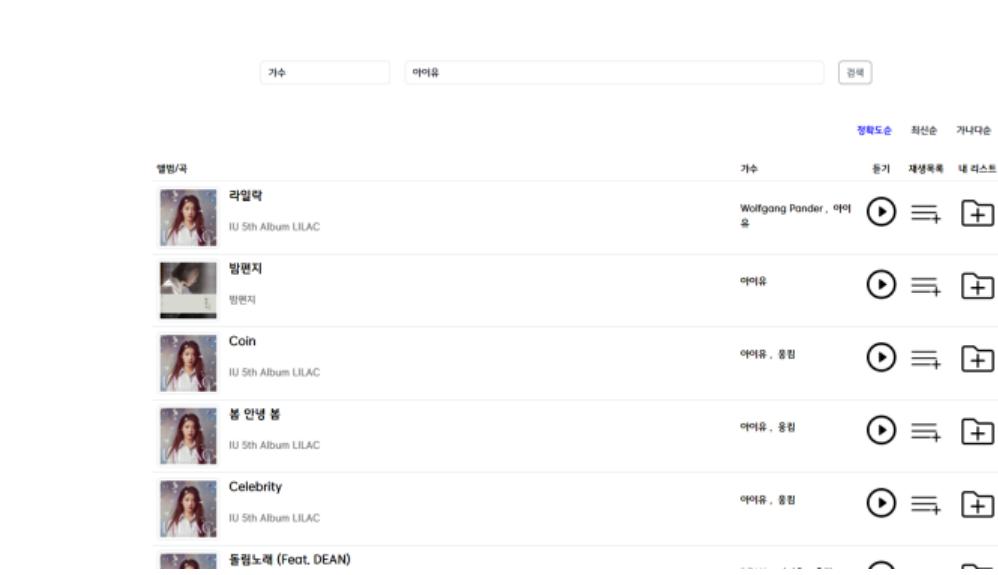
테이블 설계



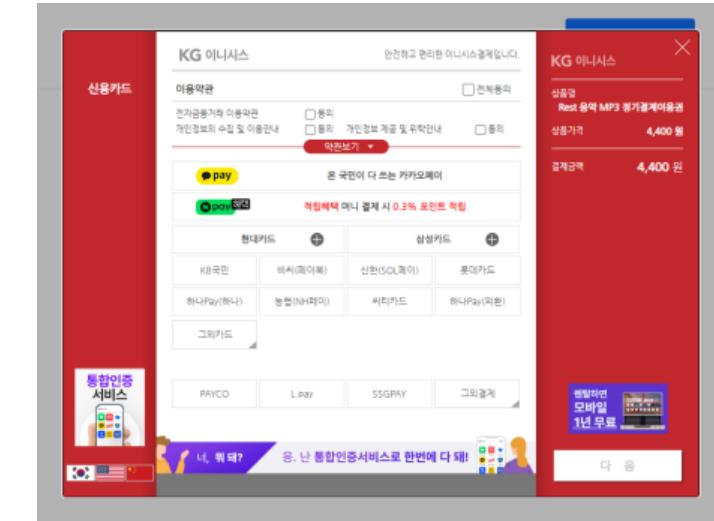
김동환



음원 상세 페이지



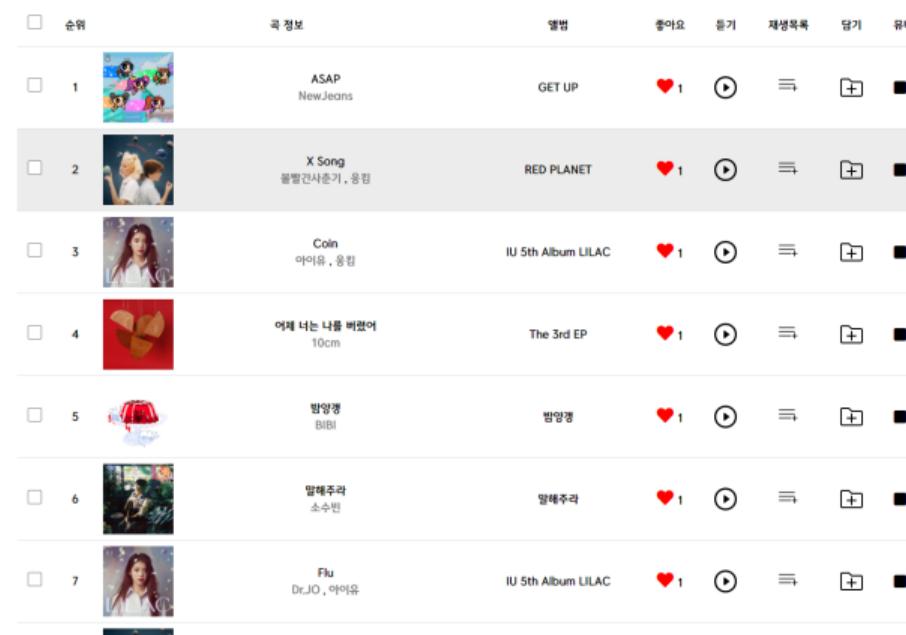
음원 검색 기능



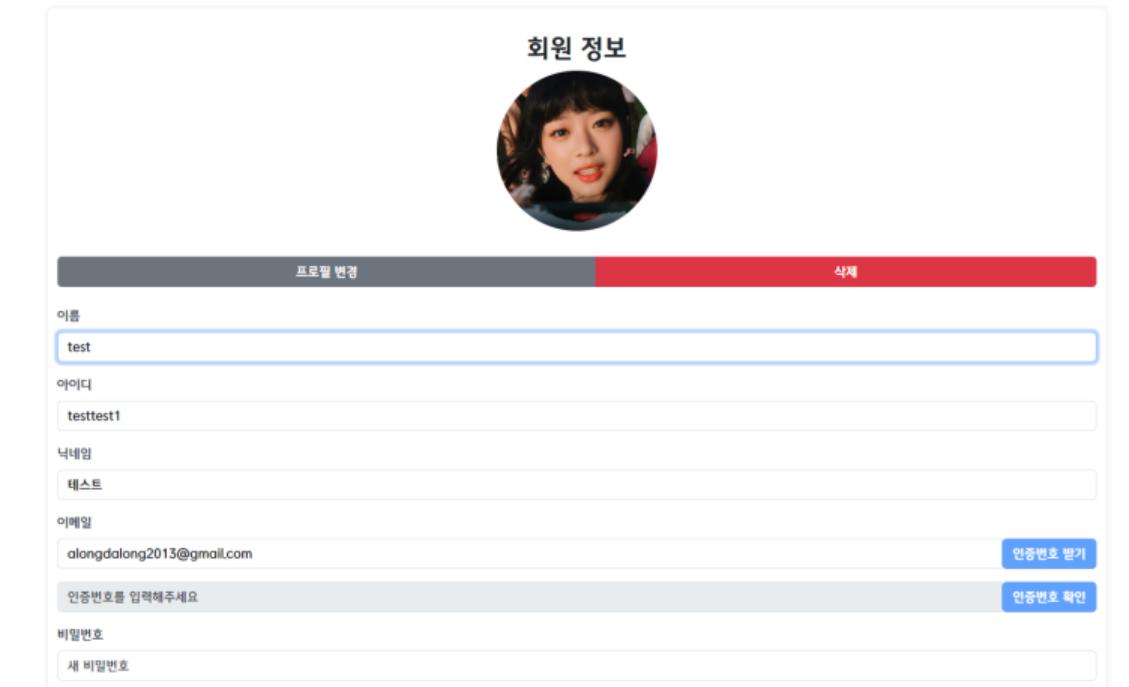
결제 페이지



김현우



음원 차트



회원 정보 변경 및 탈퇴



장동필

최신앨범 인기앨범

Rest 추천 뮤직 비디오

따끈 신상 앨범

장동필 페이지

최신앨범 인기앨범

What Time!

Who You Are

Red Planet

Love poem

방현지

사춘기집!

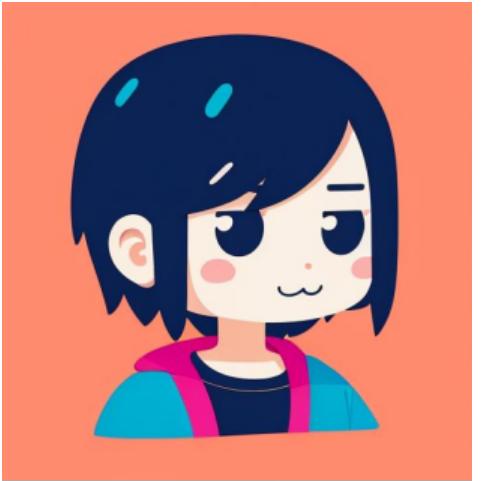
Love Lee

방현경

갈해주라

메인 페이지

앨범 페이지



박성준

マイページ

マイページ

播放アーティスト

플레이리스트

ABBA ARRIVAL

청량한 여름의 맛

播放アーティスト

播放アーティスト

상세 정보

NewJeans

상세 정보

播放アーティスト

마이 페이지

플레이리스트

아티스트 페이지

STEP 2

개발 환경

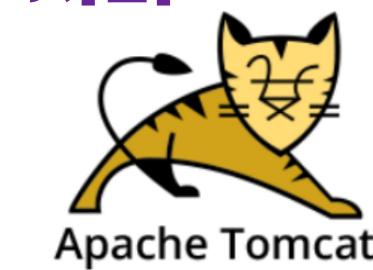
개발 환경

REST # TEAM

개발 언어



서버



데이터베이스



라이브러리



협업도구

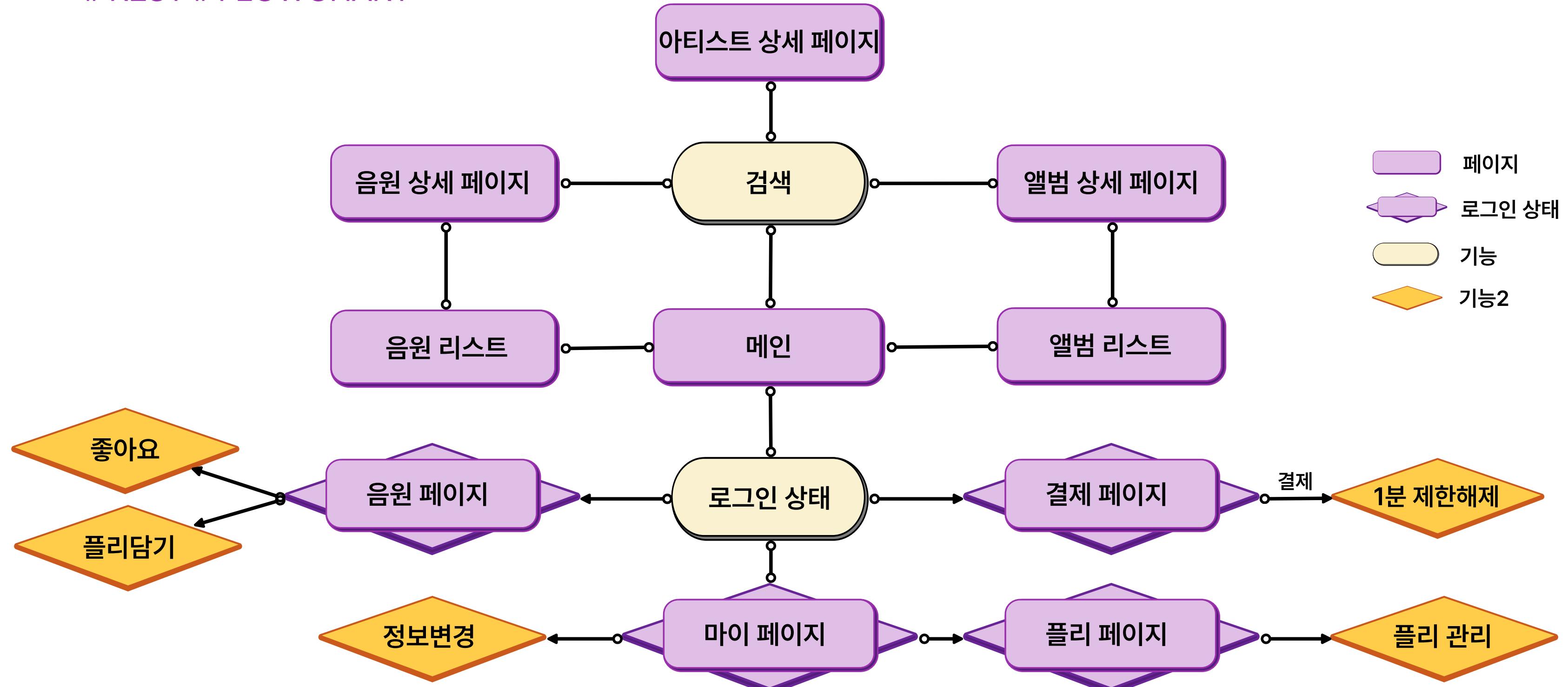


STEP 3

주요 기능

플로우 차트

REST # FLOWCHART



상세 기능

REST # DETAIL

플리에 곡 추가

여러 곡을 선택해서 두 개 이상
플레이리스트에 노래 추가 가능

The screenshot illustrates the process of adding songs to playlists in a music application. It consists of three main panels:

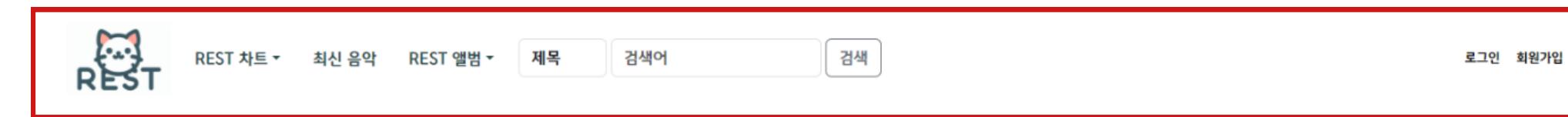
- Top Panel:** A modal window titled "플레이리스트 선택" (Playlist Selection) lists three playlists: "플리4" (selected), "플리5", and "플리6". Below the list are buttons for "닫기" (Close) and "곡 추가" (Add Song). To the right of the modal are buttons for "재생목록" (Playlist), "담기" (Save), and "뮤비" (Music Video).
- Middle Panel:** A list of songs being selected for addition. Two songs are checked: "5" by "Kenzi" and "6" by "babymonster". A blue confirmation overlay says "2개 선택됨" (2 songs selected) with buttons for "듣기" (Listen), "재생목록" (Playlist), and "담기" (Save). Other songs listed include "1" by "LILAC", "2" by "LILAC", "3" by "ASAP", "4" by "What Time!", "5" by "armageddon", "6" by "BIBI", "7" by "supernova", "8" by "RED PLANET", and "9" by "Love poem".
- Right Panel:** A list of songs in a playlist. A message box at the top right says "localhost:8080 내용: 선택한 플레이리스트에 곡이 추가되었습니다." (Content of localhost:8080: The selected playlist has been added to a song). A red box highlights this message and the "확인" (Confirm) button. The list includes songs like "ASAP" (1), "Kenzi" (2), "What Time!", "armageddon" (3), "BIBI" (4), "supernova" (5), "RED PLANET" (6), "RED PLANET" (7), "RED PLANET" (8), and "Love poem" (9).

상세 기능

REST # DETAIL

● 메인 페이지

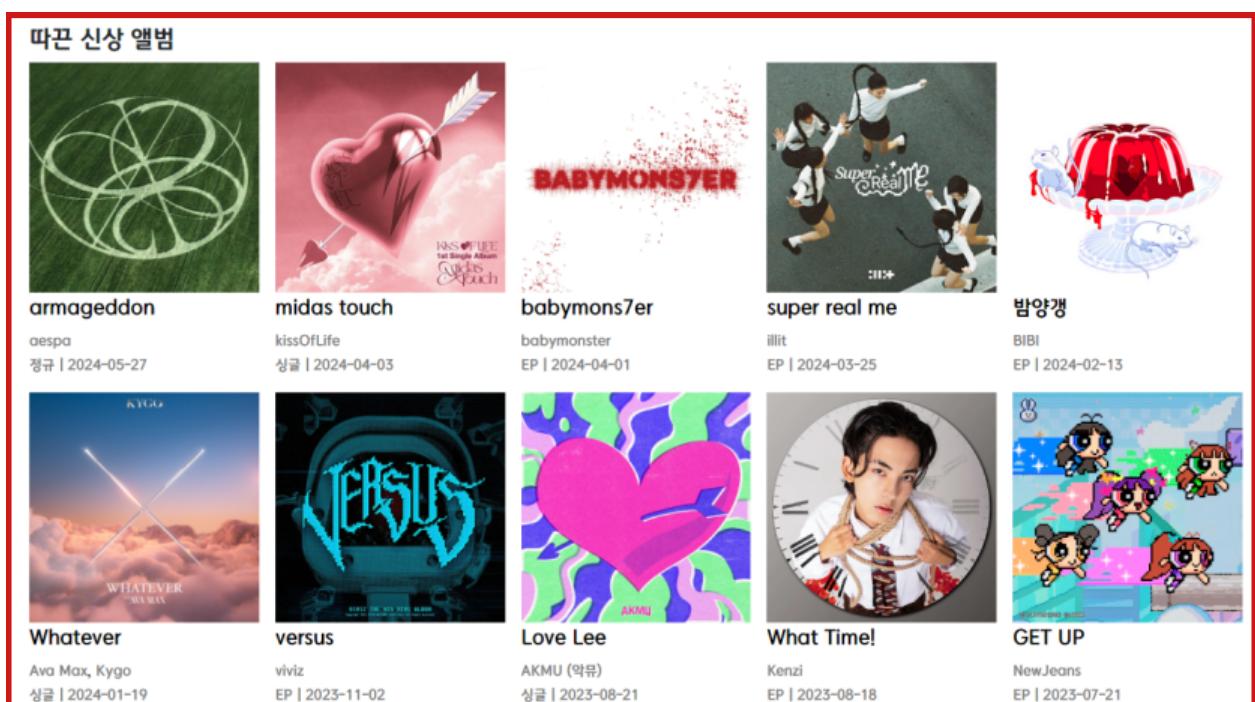
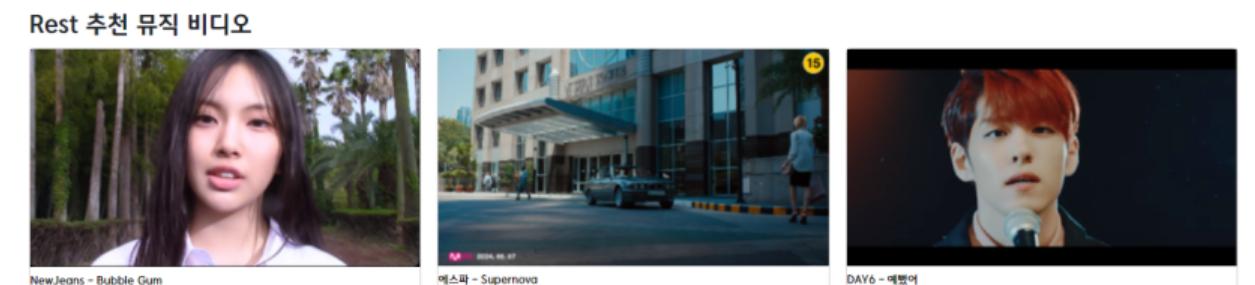
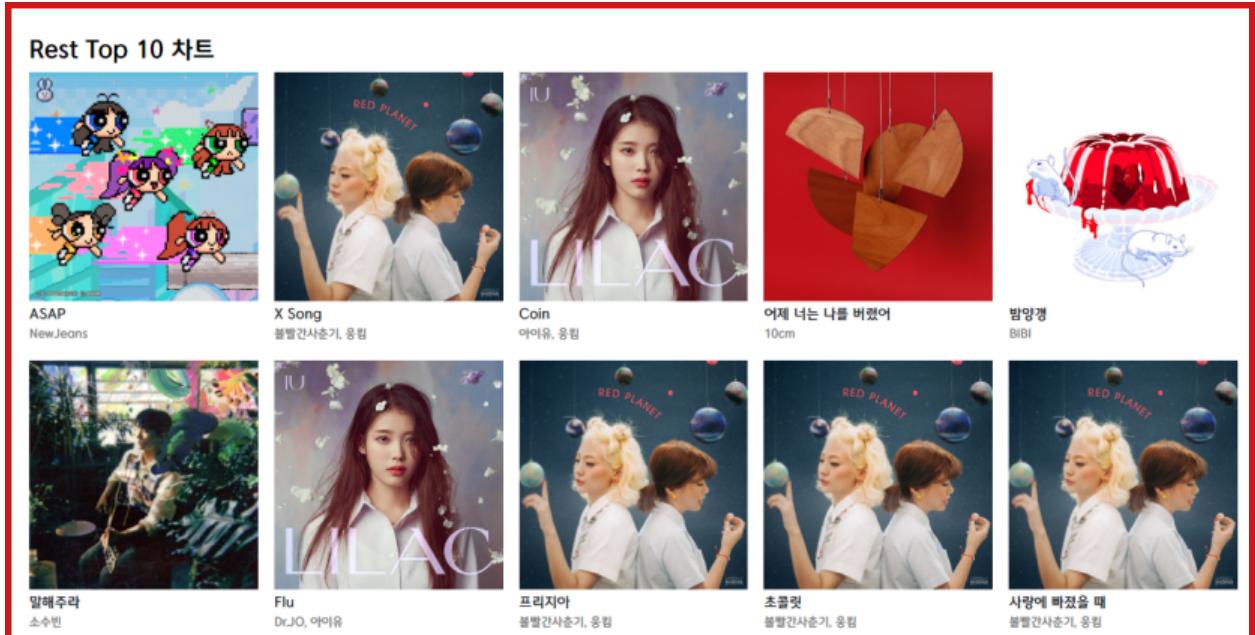
페이지가 이동이 되도 아래 재생바가 고정이 되게
페이지를 두 개로 나눈 뒤 프레임셋을 사용



A screenshot of the REST homepage featuring a music player interface. At the top, it says 'Rest Top 10 차트'. Below that is a horizontal scrollable list of album covers for the top 10 tracks. The first track in the list is 'Coin' by IU and Sungkyu. The music player interface includes a play bar at the bottom with a progress indicator showing '0:1 / 1:00', a volume slider, and standard media control buttons (back, forward, stop, repeat). A red rectangular box highlights the play bar area.

상세 기능

REST # DETAIL



● 메인 페이지

차트 페이지와 앨범페이지에서 정보를
불러와 10개씩 메인화면에 출력

상세 기능

REST # DETAIL

차트 페이지

<input type="checkbox"/> 순위	곡 정보	앨범	좋아요	듣기	재생목록	담기	유비
<input type="checkbox"/> 1	ASAP NewJeans	GET UP	♥ 1	▶	☰	[+]	◀
<input type="checkbox"/> 2	밤양갱 BIBI	밤양갱	♥ 1	▶	☰	[+]	◀
<input type="checkbox"/> 3	프리지아 불빨간사춘기, 웅킴	RED PLANET	♥ 1	▶	☰	[+]	◀
<input type="checkbox"/> 4	좋겠어 (feat. 현서) Kenzi	What Time!	♥ 1	▶	☰	[+]	◀
<input type="checkbox"/> 5	초콜릿 불빨간사춘기, 웅킴	RED PLANET	♥ 0	▶	☰	[+]	◀
<input type="checkbox"/> 6	사랑에 빠졌을 때 불빨간사춘기, 웅킴	RED PLANET	♥ 0	▶	☰	[+]	◀
<input type="checkbox"/> 7	자장가 아이유	Love poem	♥ 0	▶	☰	[+]	◀
<input type="checkbox"/> 8	나만, 봄 불빨간사춘기	사춘기집 I 꽃기운	♥ 0	▶	☰	[+]	◀
<input type="checkbox"/> 7	long cho aespa	3개 선택됨	▶ 듣기 ☰ 재생목록 [+] 담기	♥ 0	▶	☰	◀
<input type="checkbox"/> 8	prologue aespa	선택 해제	▶ 듣기 ☰ 재생목록 [+] 담기	♥ 0	▶	☰	◀
<input type="checkbox"/> 9	armageddon aespa		▶ 듣기 ☰ 재생목록 [+] 담기	♥ 0	▶	☰	◀

최신음악 페이지



체크박스를 체크하면 선택된 곡 개수에 따라
플로팅버튼에 정보가 출력되고, 듣기, 재생목록추가,
플리담기 등의 기능을 이용 할 수 있음

상세 기능

REST # DETAIL

인기앨범 최신앨범



What Time!
Kenzi

EP | 23.08.18
♥ 0
▶ ⌂



Who You Are
Jessie J

정규 | 11.11.14
♥ 0
▶ ⌂



천천히가
Kenzi, 스카이민혁, 웅김

싱글 | 21.05.01
♥ 0
▶ ⌂



RED PLANET
불빨간사춘기, 웅김

정규 | 16.08.29
♥ 0
▶ ⌂



Love poem
아이유

미니 | 19.11.18
♥ 0
▶ ⌂



삐삐
아이유

싱글 | 18.10.10
♥ 0
▶ ⌂



bampanji
아이유

싱글 | 17.03.24
♥ 0
▶ ⌂



사춘기집 | 꽃기운
불빨간사춘기

미니 | 19.04.02
♥ 0
▶ ⌂



SUMMER EPISODE
AKMU (악뮤)

싱글 | 17.07.20
♥ 0
▶ ⌂



Love Lee
AKMU (악뮤)

싱글 | 23.08.21
♥ 0
▶ ⌂



bamyanggang
BIBI

EP | 24.02.13
♥ 0
▶ ⌂



Mallehjura
소수빈

Single | 23.05.10
♥ 0
▶ ⌂

• 앨범리스트 페이지

카테고리 별로 음원의 리스트를 좋아요,
발매일 순서로 정렬해서 보여줌.

상세 기능

REST # DETAIL

● 검색 페이지

검색 구현 원리:

‘CONTAINS’ 함수를 사용해서 텍스트 데이터를 검색.
스코어 함수로 검색 결과에 점수를 부여하여 관련성을 평가.
관련성이 높은 결과부터 순서대로 보여줌.

앨범/곡	가수	듣기	재생목록	내 리스트
라일락 IU 5th Album LILAC	Wolfgang Pander, 아이유			
밤편지	아이유			
Coin IU 5th Album LILAC	아이유, 웅킴			
봄 안녕 봄 IU 5th Album LILAC	아이유, 웅킴			
Celebrity IU 5th Album LILAC	아이유, 웅킴			
돌림노래 (Feat. DEAN) IU 5th Album LILAC	DEAN, 아이유, 웅킴			
빈 컵 (Empty Cup) IU 5th Album LILAC	아이유, 웅킴			

상세 기능

REST # DETAIL

localhost:8080 내용:
로그인이 필요합니다. 로그인 하시겠습니까?

확인 취소

좋아요 듣기 재생목록 담기 뮤비

순위	곡 정보	작곡자	앨범	좋아요	듣기	재생목록	담기	뮤비
1	ASAP NewJeans	GET UP	ASAP	♡ 1	▶	☰	[+]	[CAM]
2	좋겠어 (feat. 현서) Kenzi	What Time!	좋겠어	♡ 1	▶	☰	[+]	[CAM]
3	armageddon aespa	armageddon	armageddon	♡ 1	▶	☰	[+]	[CAM]
4	밤양갱 BIBI	밤양갱	밤양갱	♡ 1	▶	☰	[+]	[CAM]
5	supernova aespa	supernova	armageddon	♡ 1	▶	☰	[+]	[CAM]
6	프리지아 불빨간사춘기, 응킴	RED PLANET	프리지아	♡ 1	▶	☰	[+]	[CAM]
7	초콜릿	RFD PLANET	초콜릿	♡ 0	▶	☰	[+]	[CAM]

● 로그인 권고

로그아웃 상태로
좋아요나 플리담기 버튼을 눌렀을
때 로그인 해달라는 알림이 뜨고,
확인시 해당 페이지로 이동

상세 기능

REST # DETAIL

● 회원가입 페이지

아이디 체크 :

REST API를 활용하여 아이디 중복체크

비밀번호 체크:

비밀번호와 비밀번호 확인 필드의 유효성 검사

이메일 체크:

- 정규식 활용하여 이메일 형식 검사
- 이메일 인증번호 기능 구현

```
const emailPattern = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/i;
```

사용할 수 없다는 알림 메시지가 나오면

가입 버튼이 비활성화

상세 기능

REST # DETAIL

마이페이지



닉네임: 테스트
ID/PW 변경 (내정보 관리)
이용권 구매

● 마이 페이지

ID/PW 변경 클릭시 회원정보 페이지로 이동

- 비밀 번호 변경
- 힌트 질문 & 답변 변경
- 프로필 이미지 변경

회원 정보



프로필 변경

이름	test
아이디	testtest1
닉네임	테스트
이메일	alongdalong2013@gmail.com
비밀번호	새 비밀번호
비밀번호 확인	비밀번호 확인
힌트 질문	힌트 질문 선택
힌트 답변	

회원 정보 변경

회원 탈퇴

프로필

localhost:8080 내용:
프로필 이미지가 성공적으로 변경되었습니다!

새 프로필 이미지 선택

파일 선택 아롱다롱.jpg

변경 **취소**



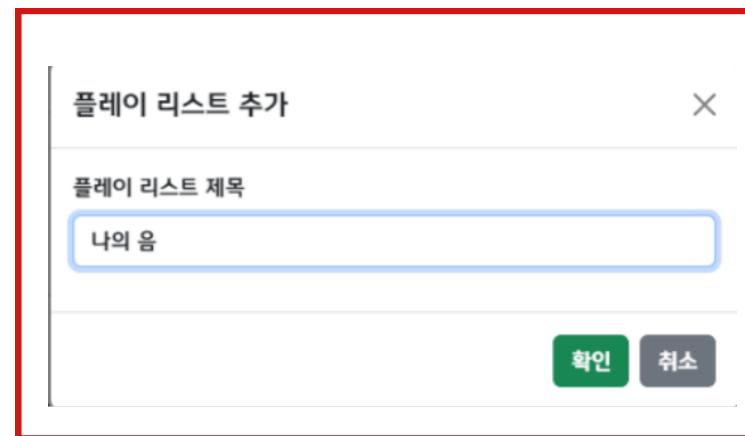
프로필 변경 버튼을 눌러 이미지 변경한 모습

상세 기능

REST # DETAIL

● 마이 페이지

- 유저가 저장한 플레이리스트를 보는 기능
- 플레이리스트 추가 버튼을 누르면 새로 생성 됨
- 좋아요를 누른 곡을 이곳에서 확인할 수 있음



마이페이지

마이페이지

닉네임: 테스트
ID/PW 변경 (내정보 관리)
이용권 구매

플레이리스트

- 청량한 여름의 맛 😊🍒🍊
- 새벽 감성을 가득 담아.. * ☀️
- 더위를 날려버릴 K-POP 모음 🎵
- 풀리

좋아요 누른 곡

커버	노래 제목	아티스트	듣기	재생목록
	ASAP GET UP	NewJeans	▶	≡+
	armaaeddon		↶	▬

상세 기능

REST # DETAIL

● 플레이리스트 페이지

플레이리스트에 저장된 음원리스트를 보여줌



더위를 날려버릴 K-POP 모음 🎵 ↗

총 10곡

플레이리스트 재생

커버	노래 제목	아티스트	듣기	재생목록
	좋겠어 (feat. 현서) What Time!	Kenzi	▶	☰+
	삐삐 삐삐	IU	▶	☰+
	Get Up GET UP	NewJeans	▶	☰+
	New Jeans GET UP	NewJeans	▶	☰+

상세 기능

REST # DETAIL

플리에 곡 추가

여러 곡을 선택해서 두 개 이상
플레이리스트에 노래 추가 가능

The screenshot illustrates the process of adding songs to playlists in a music application. It consists of three main panels:

- Top Panel:** A modal window titled "플레이리스트 선택" (Playlist Selection) lists three playlists: "플리4" (selected), "플리5", and "플리6". Below the list are buttons for "닫기" (Close) and "곡 추가" (Add Song). To the right of the modal are buttons for "재생목록" (Playlist), "담기" (Save), and "뮤비" (Music Video).
- Middle Panel:** A list of songs being selected for addition. Two songs are checked: "5" by "좋겠어 (feat. 현서)" (Kenzi) and "6" by "batter" (babymonster). A blue confirmation overlay says "2개 선택됨" (2 songs selected) with options to "듣기" (Listen), "재생목록" (Playlist), or "담기" (Save). Other songs in the list include "ASAP" by "NCT DREAM", "What Time!" by "Kenzi", "armageddon" by "despa", "방양경" by "BIBI", "supernova" by "despa", "프리지아" by "볼빨간사춘기, 웽킴", "초콜릿" by "볼빨간사춘기, 웽킴", "사랑에 빠졌을 때" by "볼빨간사춘기, 웽킴", and "자장가" by "아이유".
- Right Panel:** A log entry from "localhost:8080" stating "선택한 플레이리스트에 곡이 추가되었습니다." (Song added to selected playlist). This panel also includes buttons for "확인" (Confirm), "재생목록" (Playlist), "담기" (Save), and "뮤비" (Music Video).

상세 기능

REST # DETAIL

● 앨범상세 페이지

앨범의 타입(싱글, 정규, 미니 등),
아티스트,
발매일,
노래 장르,
타이틀 곡 (여러개일 경우 모두 표시)

"GET UP" 앨범정보



[EP]
 앨범 명 GET UP
 아티스트 NewJeans
 발매일 23.07.21
 장르 댄스/밀렉
 타이틀 곡 Cool With You, ETA, Super Shy

♡ ▶ ≡+

수록곡 (6)

#	음원	아티스트	듣기	재생목록	리스트
1	Title Super Shy	NewJeans	▶	≡+	[+]
2	Title ETA	NewJeans	▶	≡+	[+]
3	Title Cool With You	NewJeans	▶	≡+	[+]
4	New Jeans	NewJeans	▶	≡+	[+]
5	Get Up	NewJeans	▶	≡+	[+]
6	ASAP	NewJeans	▶	≡+	[+]

상세 기능

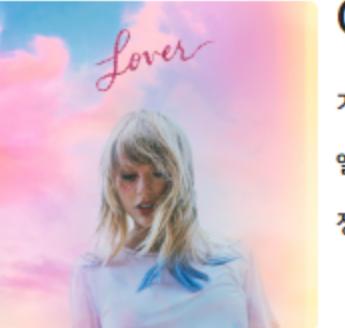
REST # DETAIL

● 음원상세 페이지

기본 정보:

아티스트 이름, 앨범 이름, 음원의 장르를

상세 정보: 각 곡의 곡명, 작사가, 작곡가, 편곡가, 가사



Cruel Summer

가수 Taylor Swift
앨범 Lover
장르 팝

Heart icon, play icon, list icon, folder icon

상세 정보

곡명 Cruel Summer

작사 Annie Clark, Jack Antonoff, Taylor Swift

작곡 Annie Clark, Jack Antonoff, Taylor Swift

편곡

Fever dream high in the quiet of the night

You know that I caught it

Bad, bad boy

Shiny toy with a price

You know that I bought it

Killing me slow, out the window

I'm always waiting for you to be waiting below

Devils roll the dice, angels roll their eyes

What doesn't kill me makes me want you more

And it's new, the shape of your body

It's blue, the feeling I've got

And it's ooh, whoa, oh

It's a cruel summer

It's cool, that's what I tell 'em

No rules in breakable heaven

But ooh, whoa oh

It's a cruel summer

With you

Hang your head low

In the glow of the vending machine

I'm not dying

You say that we'll just screw it up in these trying times

We're not trying

상세 기능

REST # DETAIL



10cm

곡

앨범

상세 정보

10CM는 포크음악을 주로 하는 대한민국의 인디밴드이다. 2010년 4월 [10cm The First EP]로 데뷔했으며, 2010년 발매한 싱글 '아메리카노'가 미디어에서 화제가 되며 주목을 받았다. 2011년 2월 첫 정규 앨범 [1.0]과 싱글 '안아줘요'로 "제8회 한국 대중음악상"에서 최우수 팝 부문을 수상했다. 2016년 4월에 발매된 싱글 '봄이 좋나'는 멜론 월간 차트 1위, 가온 차트 월간 1위를 달성했고, 2016년 "멜론 뮤직 어워드"에서 포크 부문 뮤직 스타일상을 수상한 바 있다.

이듬해 기타를 담당하던 윤철중이 탈퇴하면서 권경열 1인 체제로 변경. 그 해 9월 네 번째 정규앨범 [4.0]을 공개했고, 11월에는 EXO 첸과의 컬래버레이션 'Bye Babe'로 새로운 조화를 보여주었다. 또한 드라마 "호텔 일루니", "사랑의 불시착", "스타트업" 등의 OST에 참여해 국의 분위기를 더했으며 2021년에는 [The 3rd EP]를 출시해 특유의 꾸밈없이 솔직한 가사와 진심을 담은 어쿠스틱 사운드로 리스너들의 귀를 사로잡았다.

2023년 2월, 곡 선정부터 편곡까지 약 2년간 준비해온 오리지널 리메이크 프로젝트 앨범 [Remake 1.0]을 공개했다.



10cm

곡

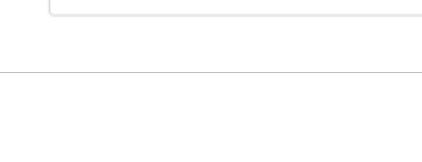
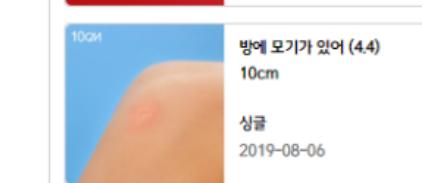
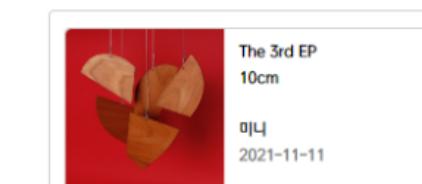
앨범

상세 정보

10CM는 포크음악을 주로 하는 대한민국의 인디밴드이다. 2010년 4월 [10cm The First EP]로 데뷔했으며, 2010년 발매한 싱글 '아메리카노'가 미디어에서 화제가 되며 주목을 받았다. 2011년 2월 첫 정규 앨범 [1.0]과 싱글 '안아줘요'로 "제8회 한국 대중음악상"에서 최우수 팝 부문을 수상했다. 2016년 4월에 발매된 싱글 '봄이 좋나'는 멜론 월간 차트 1위, 가온 차트 월간 1위를 달성했고, 2016년 "멜론 뮤직 어워드"에서 포크 부문 뮤직 스타일상을 수상한 바 있다.

이듬해 기타를 담당하던 윤철중이 탈퇴하면서 권경열 1인 체제로 변경. 그 해 9월 네 번째 정규앨범 [4.0]을 공개했고, 11월에는 EXO 첸과의 컬래버레이션 'Bye Babe'로 새로운 조화를 보여주었다. 또한 드라마 "호텔 일루니", "사랑의 불시착", "스타트업" 등의 OST에 참여해 국의 분위기를 더했으며 2021년에는 [The 3rd EP]를 출시해 특유의 꾸밈없이 솔직한 가사와 진심을 담은 어쿠스틱 사운드로 리스너들의 귀를 사로잡았다.

2023년 2월, 곡 선정부터 편곡까지 약 2년간 준비해온 오리지널 리메이크 프로젝트 앨범 [Remake 1.0]을 공개했다.



• 아티스트 페이지

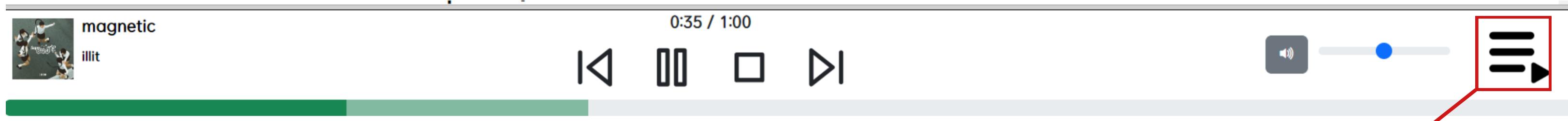
아티스트의 전 곡과, 앨범 정보를 나타냄

상세 기능

REST # DETAIL

● 음원재생 페이지

전반적인 음원 플레이어 기능을 구현함

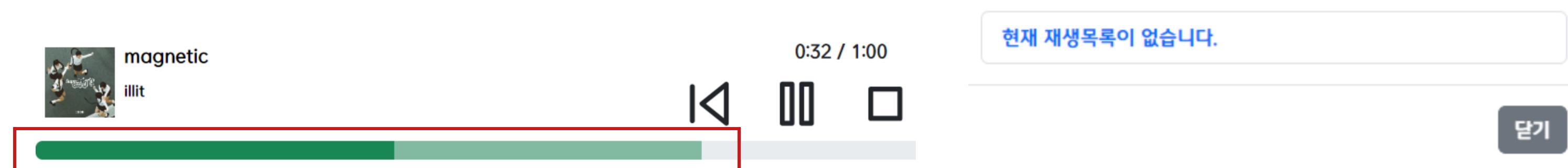


왼쪽부터
이전곡, 재생, 멈춤, 다음곡 버튼

재생목록 아이콘
클릭시 현재 재생 목록을 보여줌

현재 재생 목록

×



미결제 회원은 음원 듣기가 1분제한이 걸림.
음원 재생 시간을 세밀하게 조정 가능.

상세 기능

REST # DETAIL

The image shows a music player interface with several key features highlighted:

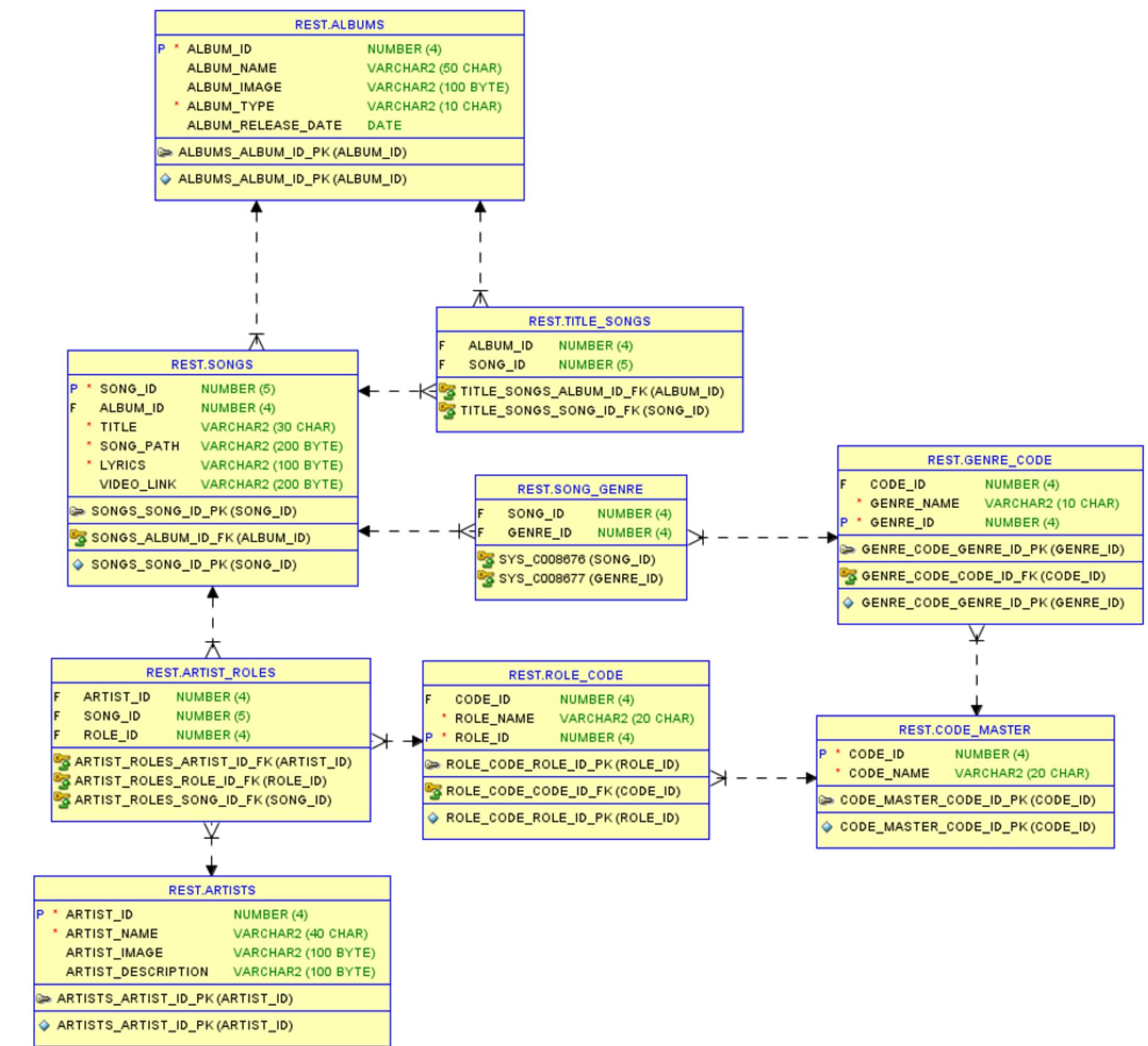
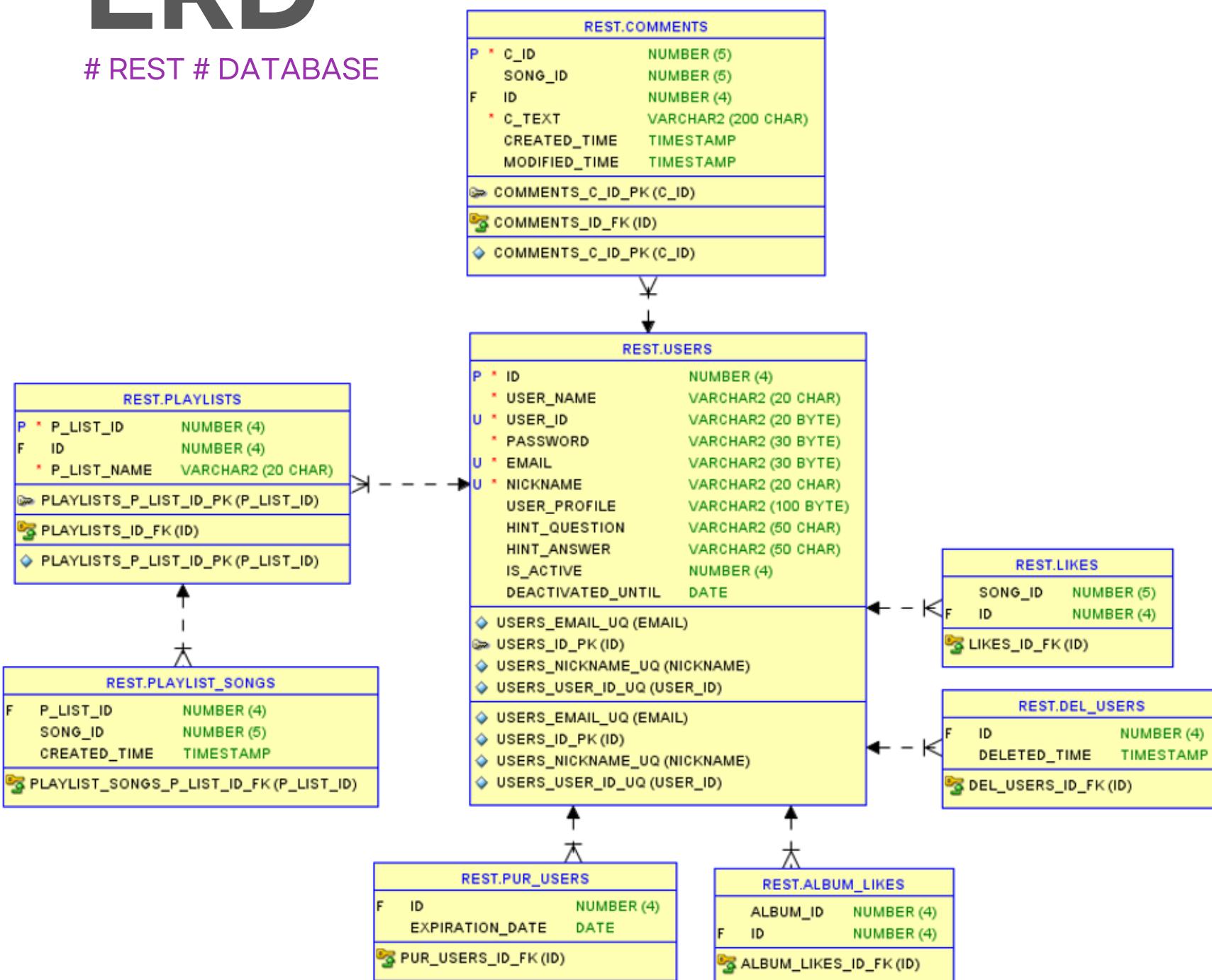
- Top Bar:** Shows album art for "magnetic" by illit, playback time (0:35 / 1:00), and standard media controls (back, pause, forward).
- Volume Control:** A volume slider with a blue dot.
- Current Play History List:** A floating window titled "현재 재생 목록" (Current Play Queue) containing a list of songs: 우주를 줄게, 프리지아, X Song, 사랑에 빠졌을 때, 반지. It includes a close button "X".
- Artist Detail Page:** Shows "What time! (Feat.DumbAss)" by Kenzi, including song details and a list of tracks: 우주를 줄게, 프리지아, X Song, 사랑에 빠졌을 때, 반지. Buttons for liking, playing, adding to queue, and adding to library are shown, with the queue button highlighted with a red box and arrow.
- Album Detail Page:** Shows "What Time!" EP by Kenzi, including track details and a list of tracks: 우주를 줄게, 프리지아, X Song, 사랑에 빠졌을 때, 반지. Buttons for liking, playing, adding to queue, and adding to library are shown, with the queue button highlighted with a red box and arrow.
- Bottom Right:** A summary of the buttons: 좋아요 (Like), 재생 (Play), 재생목록담기 (Add to Queue), and 플리담기 (Add to Library). A purple box highlights the "재생목록담기" button.
- Bottom Left:** A callout box with the text: 각 페이지마다 존재하는 현재재생목록 버튼을 클릭하면 현재재생목록 창을 띠움. 현재재생목록 창에서 다른 곡 선택시 해당 곡이 재생 됨. (Clicking the 'Current Play Queue' button on each page will open the current play queue window. When selecting a different song in the current play queue window, that song will play.)

STEP 4

DB 설계

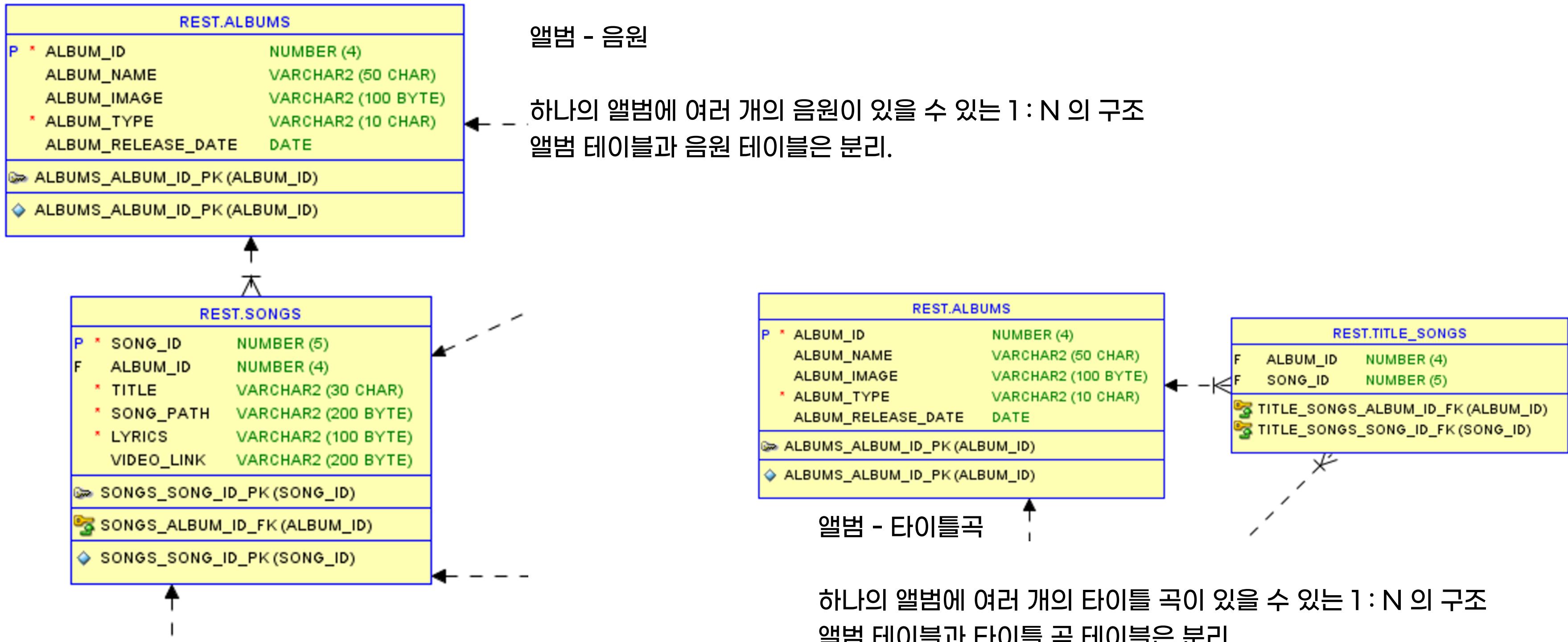
ERD

REST # DATABASE



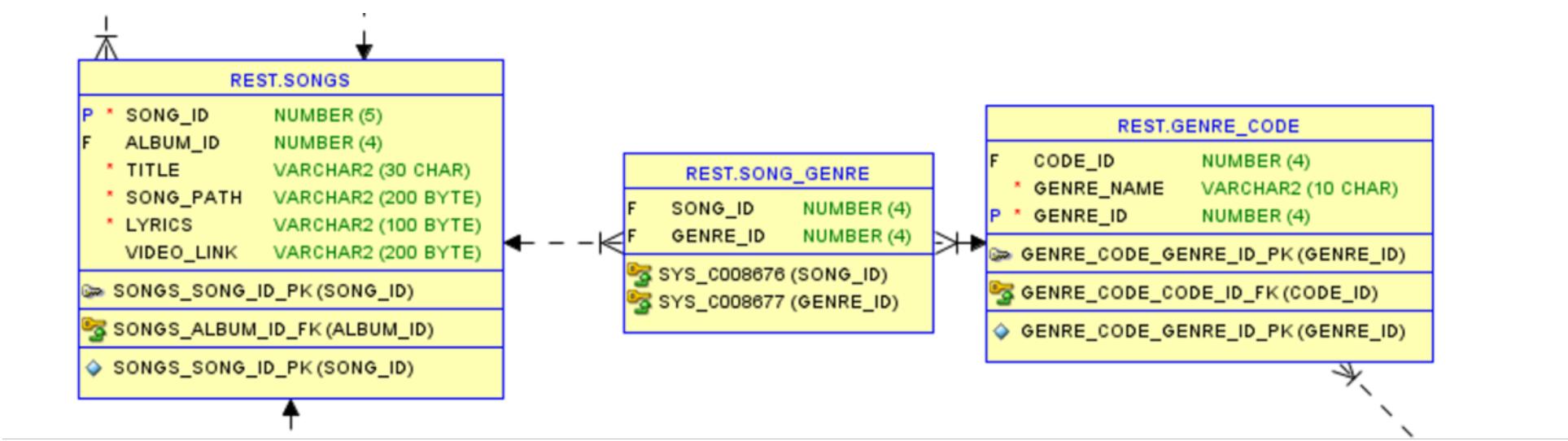
DB 설명

REST # DATABASE



DB 설명

REST # DATABASE



음원 - 장르

한 개의 음원은 여러 개의 장르를 가질 수 있음.

음원과 장르는 1:N의 구조.

음원 테이블과 장르 테이블은 분리.

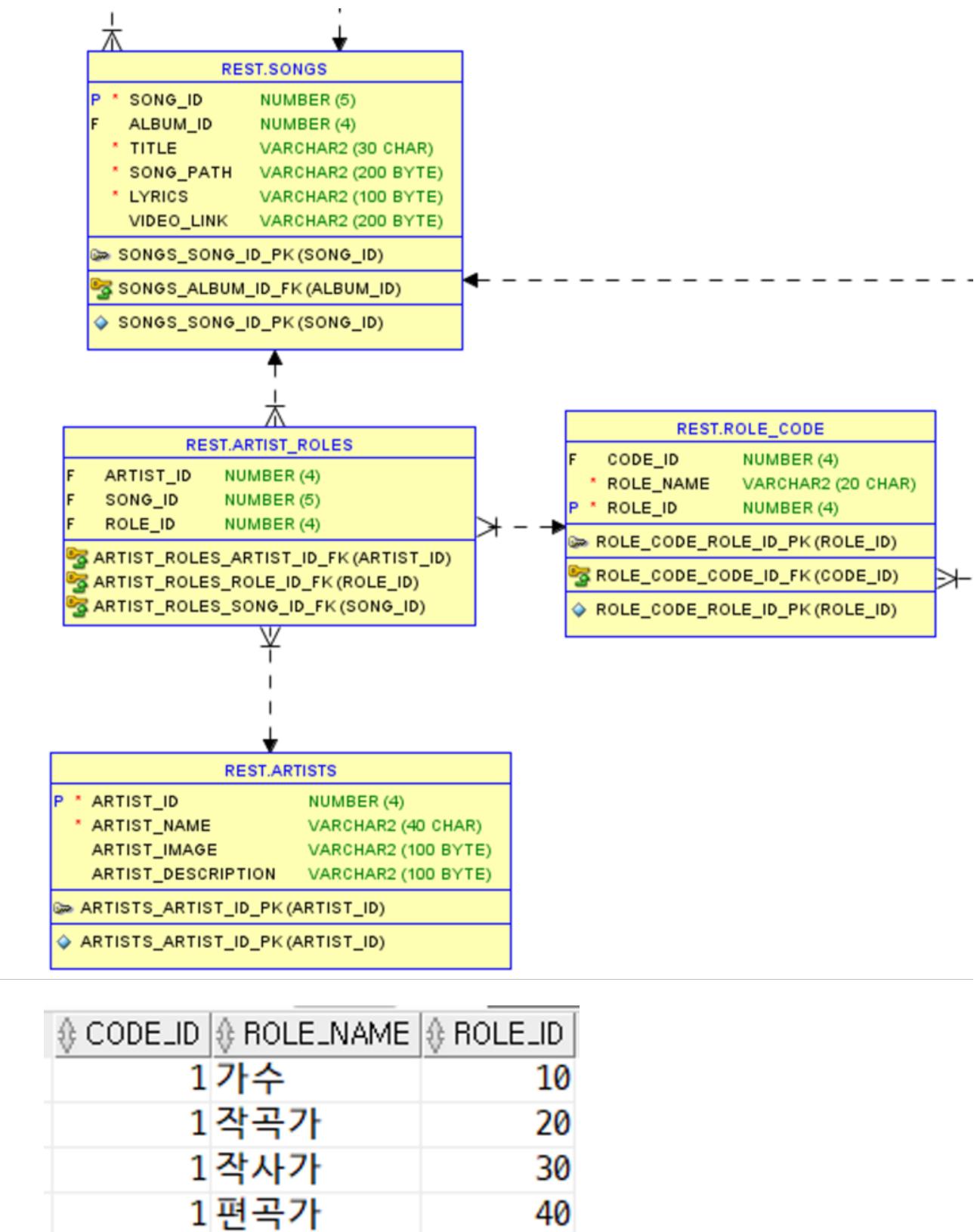
이 때 장르는 팝, 발라드, 힙합, 알앤비 등으로 한정되어 있으므로 코드 테이블로 작성.

장르 코드 테이블:

CODE_ID	GENRE_NAME	GENRE_ID
2	발라드	10
2	팝	20
2	댄스/일렉	30
2	알앤비	40
2	힙합	50
2	트로트	60
2	OST	70
2	인디	80
2	포크/블루스	90
2	록/메탈	100

DB 설명

REST # DATABASE



음원 - 아티스트 - (가수, 작곡, 작사, 편곡)

- 하나의 음원에 여러 명의 아티스트가 참여할 수 있음.
- 음원 테이블과 아티스트 테이블을 분리.
- 또한 한 명의 아티스트가 여러 음원에 참여할 수 있음. N : N 의 구조
- 음원 테이블과 아티스트 테이블을 이어주는 중간 다리 역할의 테이블이 필요함.
- 아티스트 번호와 음원 번호를 참조하는 테이블을 작성.

아티스트 참여 역할 테이블

- 한 음원에 대해 한 아티스트가 가수, 작곡, 작사, 편곡에 참여할 수 있음.
- 아티스트는 한 음원에 대해 여러 역할로도 참여할 수 있고 여러 명이 참여할 수 있다.
- 그리고 한 아티스트가 한 음원에 여러 역할로도 참여할 수 있다.
- 아티스트 - 아티스트가 참여한 음원 테이블 - 참여 역할 - 음원
- 이 때 참여 역할은 가수, 작곡, 작사, 편곡으로 한정되어 코드 테이블로 작성했다.
- 음원 테이블 - 아티스트가 참여한 음원 테이블 - 아티스트 테이블

아티스트 역할 테이블은 아티스트가 어떤 음원에 어떤 역할로 참여한 지에 대한 정보를 담고 있다.

- 이 때 아티스트 역할은 가수, 작곡, 작사, 편곡으로 한정되어 있으므로 코드 테이블로 작성했다.
- 역할 코드 테이블:

CODE_ID	ROLE_NAME	ROLE_ID
1	가수	10
1	작곡가	20
1	작사가	30
1	편곡가	40

DB 설명

REST # DATABASE

REST.USERS	
P • ID	NUMBER (4)
• USER_NAME	VARCHAR2 (20 CHAR)
U • USER_ID	VARCHAR2 (20 BYTE)
• PASSWORD	VARCHAR2 (30 BYTE)
U • EMAIL	VARCHAR2 (30 BYTE)
U • NICKNAME	VARCHAR2 (20 CHAR)
USER_PROFILE	VARCHAR2 (100 BYTE)
HINT_QUESTION	VARCHAR2 (50 CHAR)
HINT_ANSWER	VARCHAR2 (50 CHAR)
IS_ACTIVE	NUMBER (4)
DEACTIVATED_UNTIL	DATE
◆ USERS_EMAIL_UQ (EMAIL)	
◆ USERS_ID_PK (ID)	
◆ USERS_NICKNAME_UQ (NICKNAME)	
◆ USERS_USER_ID_UQ (USER_ID)	
◆ USERS_EMAIL_UQ (EMAIL)	
◆ USERS_ID_PK (ID)	
◆ USERS_NICKNAME_UQ (NICKNAME)	
◆ USERS_USER_ID_UQ (USER_ID)	

탈퇴 회원

탈퇴한 아이디를 일정 기간 동안
보관하고 그 동안 동일한 아이디로
가입하지 못하게 하기 위한 테이블

REST.DEL_USERS	
F ID	NUMBER (4)
DELETED_TIME	TIMESTAMP
◆ DEL_USERS_ID_FK (ID)	

코드 마스터 테이블

코드 테이블이 늘어남에 따라 코드 테이블을
참조하기 쉽게 하기 위해 만든 테이블

◆ CODE_ID	◆ CODE_NAME
	2 장르
	1 아티스트

회원

회원 정보를 저장하는 테이블
비밀번호 찾기에 이용되는 질문과
답변 컬럼도 포함되어있음.

REST.PUR_USERS	
F ID	NUMBER (4)
EXPIRATION_DATE	DATE
◆ PUR_USERS_ID_FK (ID)	

결제회원

결제한 회원을 관리하기 위한 테이블
결제 페이지에서 결제한 회원은 음원을 1분이 아닌 전부 들을 수 있음.

REST.COMMENTS	
P • C_ID	NUMBER (5)
SONG_ID	NUMBER (5)
F ID	NUMBER (4)
• C_TEXT	VARCHAR2 (200 CHAR)
CREATED_TIME	TIMESTAMP
MODIFIED_TIME	TIMESTAMP
◆ COMMENTS_C_ID_PK (C_ID)	
◆ COMMENTS_ID_FK (ID)	
◆ COMMENTS_C_ID_PK (C_ID)	

댓글

음원 상세페이지에 있는 댓글을 위한 테이블

REST.LIKES	
SONG_ID	NUMBER (5)
F ID	NUMBER (4)
◆ LIKES_ID_FK (ID)	

앨범 좋아요

앨범 번호와 회원 번호를 참조하여 count함수
를 이용해 좋아요 개수를 얻기 위한 테이블

음원 좋아요

앨범 좋아요와 동일

REST.ALBUM_LIKES	
F ALBUM_ID	NUMBER (4)
F ID	NUMBER (4)
◆ ALBUM_LIKES_ALBUM_ID_FK (ALBUM_ID)	
◆ ALBUM_LIKES_ID_FK (ID)	

DB 설명

REST # DATABASE

회원 - 플레이 리스트 - 음원

한 명의 회원이 여러 개의 플레이 리스트를 가질 수 있으므로
 회원 테이블과 플레이 리스트 테이블을 분리.
 또한 플레이 리스트는 여러 음원을 가질 수 있음.
 플레이 리스트 테이블과 음원 테이블을 분리

플레이 리스트를 저장하기 위한 테이블

REST.SONGS	
P *	SONG_ID NUMBER (5)
	ALBUM_ID NUMBER (4)
*	TITLE VARCHAR2 (30 CHAR)
*	SONG_PATH VARCHAR2 (200 BYTE)
*	LYRICS VARCHAR2 (100 BYTE)
	VIDEO_LINK VARCHAR2 (200 BYTE)
☞	SONGS_SONG_ID_PK (SONG_ID)
◆	SONGS_SONG_ID_PK (SONG_ID)

REST.PLAYLIST_SONGS	
F	P_LIST_ID NUMBER (4)
F	SONG_ID NUMBER (5)
	CREATED_TIME TIMESTAMP
☞	PLAYLIST_SONGS_P_LIST_ID_FK (P_LIST_ID)
☞	PLAYLIST_SONGS_SONG_ID_FK (SONG_ID)

REST.PLAYLISTS	
P *	P_LIST_ID NUMBER (4)
F	ID NUMBER (4)
	P_LIST_NAME VARCHAR2 (20 CHAR)
☞	PLAYLISTS_P_LIST_ID_PK (P_LIST_ID)
☞	PLAYLISTS_ID_FK (ID)
◆	PLAYLISTS_P_LIST_ID_PK (P_LIST_ID)

REST.USERS	
P *	ID NUMBER (4)
	USER_NAME VARCHAR2 (20 CHAR)
U *	USER_ID VARCHAR2 (20 BYTE)
	PASSWORD VARCHAR2 (30 BYTE)
U *	EMAIL VARCHAR2 (30 BYTE)
	NICKNAME VARCHAR2 (20 CHAR)
	USER_PROFILE VARCHAR2 (100 BYTE)
	HINT_QUESTION VARCHAR2 (50 CHAR)
	HINT_ANSWER VARCHAR2 (50 CHAR)
	IS_ACTIVE NUMBER (4)
	DEACTIVATED_UNTIL DATE
◆	USERS_EMAIL_UQ (EMAIL)
☞	USERS_ID_PK (ID)
◆	USERS_NICKNAME_UQ (NICKNAME)
◆	USERS_USER_ID_UQ (USER_ID)
◆	USERS_EMAIL_UQ (EMAIL)
◆	USERS_ID_PK (ID)
◆	USERS_NICKNAME_UQ (NICKNAME)
◆	USERS_USER_ID_UQ (USER_ID)

쿼리 설명

REST # QUERY

2번 앨범 출력 예시:

ALBUM_ID	ALBUM_TYPE	ALBUM_NAME	ALBUM_RELEASE_DATE	ALBUM_IMAGE	SONG_ID	TITLE	GENRE_NAME	ARTIST_NAME	ARTIST_ID
2정규	IU 5th Album	LILAC	2021/03/25/00:00:00	lilac_albumcover.png	5, 7	Coin, 라일락	댄스/일렉, 알앤비	아이유	49

장르 관련 테이블이 Left Join인 이유: 모든 수록곡의 장르를 나열하기 위함.

모든 수록곡의 장르가 중복을 제거하고 나열되어 출력하기 위함.

앨범 아이디로 검색 시 음원 제목 컬럼(title)에는 앨범의 타이틀 곡이 여러 개 일 경우 LISTAGG 문법으로 인해 (,)로 나열되어 출력됨.

앨범 상세보기 페이지에 사용되는 쿼리

```
<!-- 앨범 상세보기에서 사용하는 데이터를 가져오는 쿼리 -->
SELECT
    a.album_id,
    a.album_type,
    a.album_name,
    a.album_release_date,
    a.album_image,
    LISTAGG(DISTINCT s.song_id, ', ') WITHIN GROUP (ORDER BY s.song_id) AS song_id,
    LISTAGG(DISTINCT s.title, ', ') WITHIN GROUP (ORDER BY s.title) AS title,
    LISTAGG(DISTINCT gc.genre_name, ', ') WITHIN GROUP (ORDER BY gc.genre_name) AS genre_name,
    LISTAGG(DISTINCT art.artist_name, ', ') WITHIN GROUP (ORDER BY art.artist_name) AS artist_name,
    LISTAGG(DISTINCT art.artist_id, ', ') WITHIN GROUP (ORDER BY art.artist_id) AS artist_id
FROM
    albums a
    JOIN songs s ON a.album_id = s.album_id
    JOIN title_songs ts ON ts.song_id = s.song_id
    LEFT JOIN song_genre sg ON s.song_id = sg.song_id
    LEFT JOIN genre_code gc ON sg.genre_id = gc.genre_id
    JOIN artist_roles artr ON s.song_id = artr.song_id
    JOIN artists art ON artr.artist_id = art.artist_id
    JOIN role_code rc ON artr.role_id = rc.role_id
WHERE
    a.album_id = #{albumId} AND rc.role_id = 10
GROUP BY
    a.album_id,
    a.album_type,
    a.album_name,
    a.album_release_date,
    a.album_image
```

사용된 페이지: 앨범 상세 페이지

"IU 5th Album LILAC" 앨범정보

[정규]
 앨범 명 IU 5th Album LILAC
 아티스트 아이유
 발매일 21.03.25
 장르 댄스/일렉, 알앤비
 타이틀 곡 Coin, 라일락

장르와 타이틀 곡이 여러 개인 경우 (,)로 구분되어 나열됨.

수록곡 (10)

#	음원	아티스트	듣기	재생목록	리스트
1	Title	라일락	▶	☰	+
2	Title	Coin	▶	☰	+
3	Flu	아이유	▶	☰	+
4	봄 안녕 봄	아이유	▶	☰	+

이 때 앨범 정보에 표시된 장르에는 해당 장르 차트 페이지의 링크를 걸고,
 타이틀 곡에는 해당 음원 상세 페이지의 링크를 걸고 싶어 사용된 코드

DB 설명

REST # DATABASE

```
<script>
  const singerName = '${ album.artistName }';
  const singerId = '${ album.artistId }';
</script>
```

컨트롤러에서 model 객체로 전달받은 데이터를 자바스크립트에 전달.

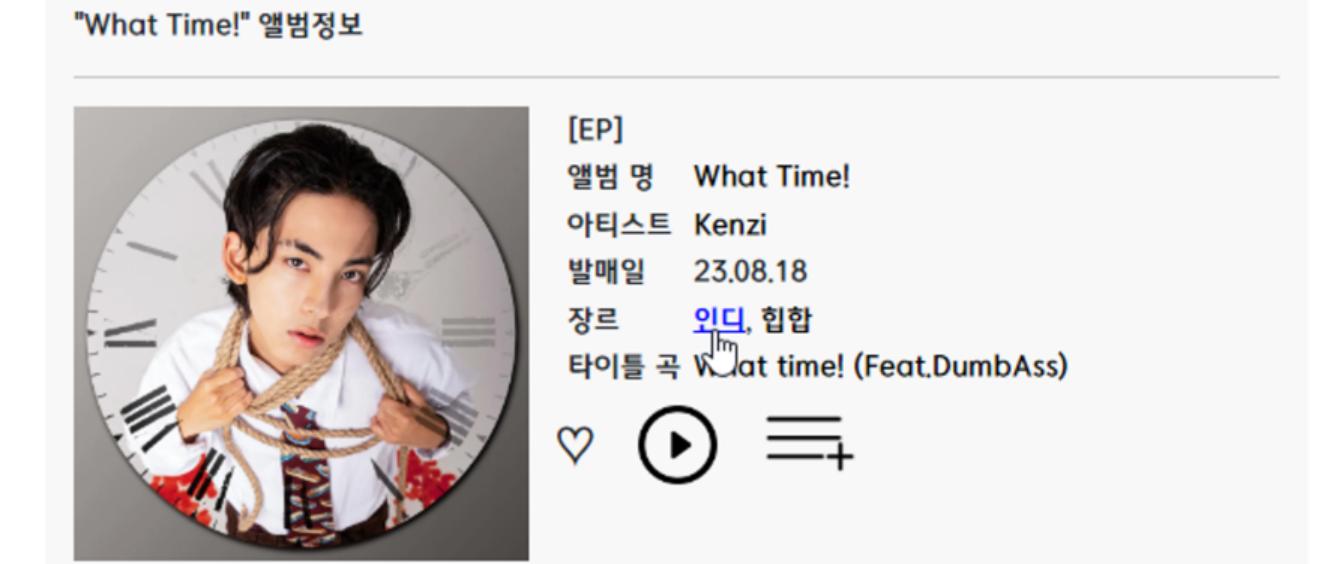
```
const singerNameSpace = document.querySelector('#singerNames');
const splitsingerName = singerName.split(',');
const splitsingerId = singerId.split(',');
artistPageLinked(singerNameSpace, splitsingerName, splitsingerId);

// 장르 작성 & 링크
const splitgenre = genreName.split(',');
const genreSpace = document.querySelector('#genre');
let genreHtml = genreSpace.innerHTML; // 기존 내용 유지
for (let i = 0; i < splitgenre.length; i++) {
  const trimmedName = splitgenre[i].trim();
  const genrePage = `../song/genreChart?genreName=${trimmedName}`;

  if (i === 0) {
    genreHtml += `<span class='text-center' style='cursor: pointer;' onclick="location.href='${genrePage}'"
      onmouseover="this.style.fontWeight='bold'; this.style.textDecoration='underline'; this.style.color='blue'";
      onmouseout="this.style.fontWeight='normal'; this.style.textDecoration='none'; this.style.color='black';">${trimmedName}</span>`;
  } else {
    // 이후 링크들은 첨부와 함께 추가
    genreHtml += `,<span class='text-center' style='cursor: pointer;' onclick="location.href='${genrePage}'"
      onmouseover="this.style.fontWeight='bold'; this.style.textDecoration='underline'; this.style.color='blue'";
      onmouseout="this.style.fontWeight='normal'; this.style.textDecoration='none'; this.style.color='black';">${trimmedName}</span>`;
  }
}
genreSpace.innerHTML = genreHtml;
```

전달받은 데이터로 Html 태그 작성.

split 문자열 함수로 (,)로 구분한 문자열 배열을 만들고 반복문으로 나열
각각의 문자열에 원하는 스타일 작성.



자바스크립트로 작성한 태그 효과가 적용된 모습
해당 링크 클릭 시 인디 장르 페이지로 이동함.

DB 설명

REST # DATABASE

```

SELECT
    s.song_id,
    s.title AS song_title,
    a.album_id,
    a.album_name AS album_name,
    a.album_image AS album_image,
    singer.singer_name,
    singer.singer_ids,
    writer.writers,
    writer.writer_ids,
    composer.composers,
    composer.composer_ids,
    arranger.arrangers,
    arranger.arranger_ids,
    LISTAGG(gc.genre_name, ', ') WITHIN GROUP (ORDER BY gc.genre_name) AS genre,
    s.lyrics
FROM
    songs s
JOIN albums a ON s.album_id = a.album_id
LEFT JOIN (
    SELECT
        ar.song_id,
        LISTAGG(art.artist_name, ', ') WITHIN GROUP (ORDER BY art.artist_name) AS singer_name,
        LISTAGG(art.artist_id, ', ') WITHIN GROUP (ORDER BY art.artist_name) AS singer_ids
    FROM
        artist_roles ar
    JOIN artists art ON ar.artist_id = art.artist_id
    WHERE
        ar.role_id = 10
    GROUP BY
        ar.song_id
) singer ON s.song_id = singer.song_id
LEFT JOIN (
    SELECT
        ar.song_id,
        LISTAGG(art.artist_name, ', ') WITHIN GROUP (ORDER BY art.artist_name) AS writers,
        LISTAGG(art.artist_id, ', ') WITHIN GROUP (ORDER BY art.artist_name) AS writer_ids
    FROM
        artist_roles ar
    JOIN artists art ON ar.artist_id = art.artist_id
    WHERE
        ar.role_id = 30
    GROUP BY
        ar.song_id
) writer ON s.song_id = writer.song_id
LEFT JOIN (
    SELECT
        ar.song_id,
        LISTAGG(art.artist_name, ', ') WITHIN GROUP (ORDER BY art.artist_name) AS composers,
        LISTAGG(art.artist_id, ', ') WITHIN GROUP (ORDER BY art.artist_name) AS composer_ids
    FROM
        artist_roles ar
    JOIN artists art ON ar.artist_id = art.artist_id
    WHERE
        ar.role_id = 20
    GROUP BY
        ar.song_id
) composer ON s.song_id = composer.song_id
LEFT JOIN (
    SELECT
        ar.song_id,
        LISTAGG(art.artist_name, ', ') WITHIN GROUP (ORDER BY art.artist_name) AS arrangers,
        LISTAGG(art.artist_id, ', ') WITHIN GROUP (ORDER BY art.artist_name) AS arranger_ids
    FROM
        artist_roles ar
    JOIN artists art ON ar.artist_id = art.artist_id
    WHERE
        ar.role_id = 40
    GROUP BY
        ar.song_id
) arranger ON s.song_id = arranger.song_id
LEFT JOIN song_genre sg ON s.song_id = sg.song_id
LEFT JOIN genre_code gc ON sg.genre_id = gc.genre_id
WHERE
    s.song_id = #{id}
GROUP BY
    s.song_id, s.title, a.album_id, a.album_name, a.album_image, s.lyrics,
    singer.singer_name, singer.singer_ids,
    writer.writers, writer.writer_ids,
    composer.composers, composer.composer_ids,
    arranger.arrangers, arranger.arranger_ids

```

```

) writer ON s.song_id = writer.song_id|
LEFT JOIN (
    SELECT
        ar.song_id,
        LISTAGG(art.artist_name, ', ') WITHIN GROUP (ORDER BY art.artist_name) AS composers,
        LISTAGG(art.artist_id, ', ') WITHIN GROUP (ORDER BY art.artist_name) AS composer_ids
    FROM
        artist_roles ar
    JOIN artists art ON ar.artist_id = art.artist_id
    WHERE
        ar.role_id = 20
    GROUP BY
        ar.song_id
) composer ON s.song_id = composer.song_id
LEFT JOIN (
    SELECT
        ar.song_id,
        LISTAGG(art.artist_name, ', ') WITHIN GROUP (ORDER BY art.artist_name) AS arrangers,
        LISTAGG(art.artist_id, ', ') WITHIN GROUP (ORDER BY art.artist_name) AS arranger_ids
    FROM
        artist_roles ar
    JOIN artists art ON ar.artist_id = art.artist_id
    WHERE
        ar.role_id = 40
    GROUP BY
        ar.song_id
) arranger ON s.song_id = arranger.song_id
LEFT JOIN song_genre sg ON s.song_id = sg.song_id
LEFT JOIN genre_code gc ON sg.genre_id = gc.genre_id
WHERE
    s.song_id = #{id}
GROUP BY
    s.song_id, s.title, a.album_id, a.album_name, a.album_image, s.lyrics,
    singer.singer_name, singer.singer_ids,
    writer.writers, writer.writer_ids,
    composer.composers, composer.composer_ids,
    arranger.arrangers, arranger.arranger_ids

```

자바스크립트로 작성한 태그 효과가 적용된 모습
해당 링크 클릭 시 인디 장르 페이지로 이동함.

SONG.TITLE	ALBUM_NAME	ALBUM_IMAGE	SINGER_NAME	WRITERS	COMPOSERS	COMPOSER_IDS	ARRANGERS	ARRANGER_IDS	GENRE	LYRICS
5라일락	ZIU 5th Album LILAC	lilac_albumcover.png	아이유	49 아이유	49 Dr.JO, N!ko, 웅킴, 임수호	10, 12, 11, 9	N!ko, 웅킴, 임수호	12, 11, 9	댄스/일렉	아이유라일락가사.txt

키워드 입력

#키워드 #키워드 # 키워드

02

관련된 세부내용을 입력해주세요. 이 폰트는 G마켓 산스 Medium이고, 폰트 크기는 20pt입니다. 행간은 30, 장평은 97%입니다.

01

관련된 세부내용을 입력해주세요. 이 폰트는 G마켓 산스 Medium이고, 폰트 크기는 20pt입니다. 행간은 30, 장평은 97%입니다.

03

관련된 세부내용을 입력해주세요. 이 폰트는 G마켓 산스 Medium이고, 폰트 크기는 20pt입니다. 행간은 30, 장평은 97%입니다.



" 이미지와 관련된 세부내용 또는 강조하고자 하는 핵심 내용을 입력해주세요 "

내용을 입력해주세요

현재 페이지와 관련된 세부내용을 입력해주세요. 페이지의 내용을 요약해서 두문장 정도로 입력해주셔도 좋습니다.

키워드 입력

키워드와 관련된 세부내용을 입력해주세요. 세 줄 정도 입력해주시면 보기 좋습니다.

키워드 입력

핵심 문장을
입력해주세요

키워드 입력

키워드와 관련된 세부내용을 입력해주세요. 세 줄 정도 입력해주시면 보기 좋습니다.

STEP 5

DB 설계

내용을 입력해주세요

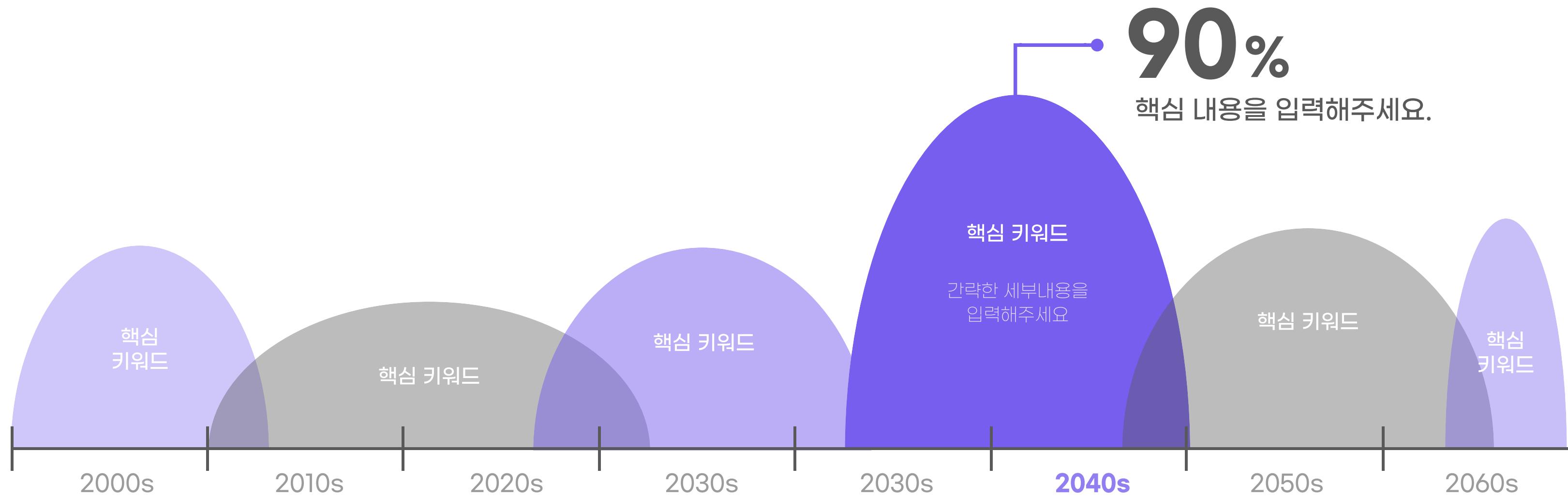
이 페이지는 빅데이터 느낌의 페이지입니다. 원형 도형 안에 키워드를 입력해주세요. 키워드는 필요한 만큼 추가해주세요.



상단 키워드와 세부내용 또는 강조하고자 하는 핵심 문장을 입력해주세요

내용을 입력해주세요

현재 페이지와 관련된 세부내용을 입력해주세요. 페이지의 내용을 요약해서 두문장 정도로 입력해주셔도 좋습니다.



감사합니다,
REST MUSIC이었습니다.