

음악퀴즈
설계 보고서

같이 듣고, 채팅으로 먼저 맞히는 실시간 노래 맞추기

단계	0단계 · 설계 (개발 시작 전)
작성일	2026-10-18
주소	https://mquiz.tkrmagid.kr (포트 2974)
리포	git.tkrmagid.kr/tkrmagid/music_quiz
마지막 장	질문 · 요청사항 — 답변 부탁드립니다

1. 한눈에 보기

- **무엇**: 친구들이 방에 모여 노래를 듣고(또는 가사를 보고) 채팅으로 제목·가수를 먼저 맞히는 사이트. 스타크래프트 노래 맞추기 맵 감성 그대로
- **어떻게**: joke-app(같이 놀자 보드게임)의 로그인·관리자·채팅·실시간 방 구조를 가져오고, 컴퓨터 대결은 빼고, 인원 제한을 없앰
- **모드 3가지**: 듣고맞추기(1초~끝까지, 처음/하이라이트) · 가수노래 맞추기(가수 여러 명 선택) · 가사보고 맞추기(30초~무제한)
- **방 설정**: 장르(복수), 연도 슬라이더, 난이도 별 10개(색 4단계), 정답방식(제목/가수/제목+가수 순서무관), 정답인식 토글, 힌트, 스킵인원
- **채팅**: 화면 가운데 아래 가장 크게, 입력 중 표시, 지연 없음, 정답 아래 5줄 공백 사이에 "노래제목 - 가수", Ctrl+I 힌트 · Ctrl+K 스킵
- **노래 DB**: 스타맵 803곡 + 노라보세 8,572곡 + 멜론·지니·벅스 연도별/실시간 + 유튜브·스포티파이 플레이리스트 → 멜론 정보로 연도·장르, 여러 지표로 난이도 10단계, 유튜브 Topic 주소(`https://youtu.be/ID`)
- **관리자**: joke-app과 같은 관리자 화면 + 노래 관리(검색·필터·추가·삭제·수정), 슈퍼어드민 2명 동일

미리 확인한 것

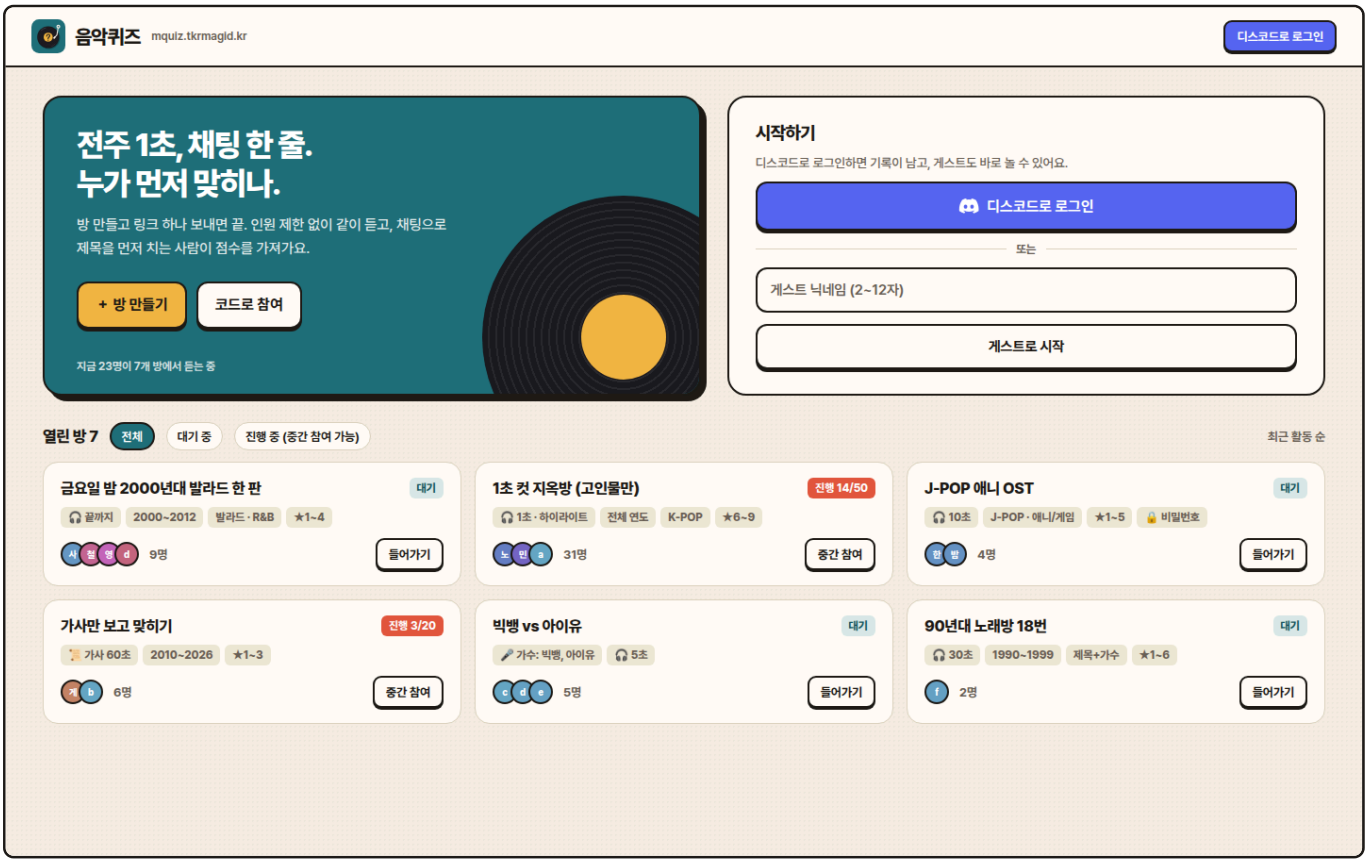
확인	결과
디스코드 앱 (1561284577564037140)	클라이언트 ID·시크릿 정상(토큰 발급 성공), 앱 이름 "음악퀴즈", **아이콘 비어 있음 → 업로드 필요**
도메인 mquiz.tkrmagid.kr	DNS 등록됨(CNAME → tkrmagid.kr). 프록시·인증서는 1~2단계에서 설정
유튜브 Topic 찾기	일반 유튜브 검색 + "- Topic" 필터로 공식 음원 확인(예: 광(狂) → SG Wannabe - Topic). 유튜브 뮤직 "노래" 검색은 로그인 쿠키 필요, Lavalink `ytmsearch:` 는 현재 빈 결과
하이라이트	유튜브 "가장 많이 다시 본 구간" 데이터를 받을 수 있음(광(狂) → 75.8초 = 후렴)
노라보세 데이터	8,572곡(1984~2026 주간 TOP50), 멜론 곡 ID 기준, 곡별 유튜브 ID·가사 포함
멜론·지니·벅스	연도별·실시간 차트 대부분 그대로 받아짐(벅스 연도별만 별도 처리)
sta_songs.json	803곡, 연도는 275곡만 있음 → 멜론 발매일로 다시 맞춤

진행 순서

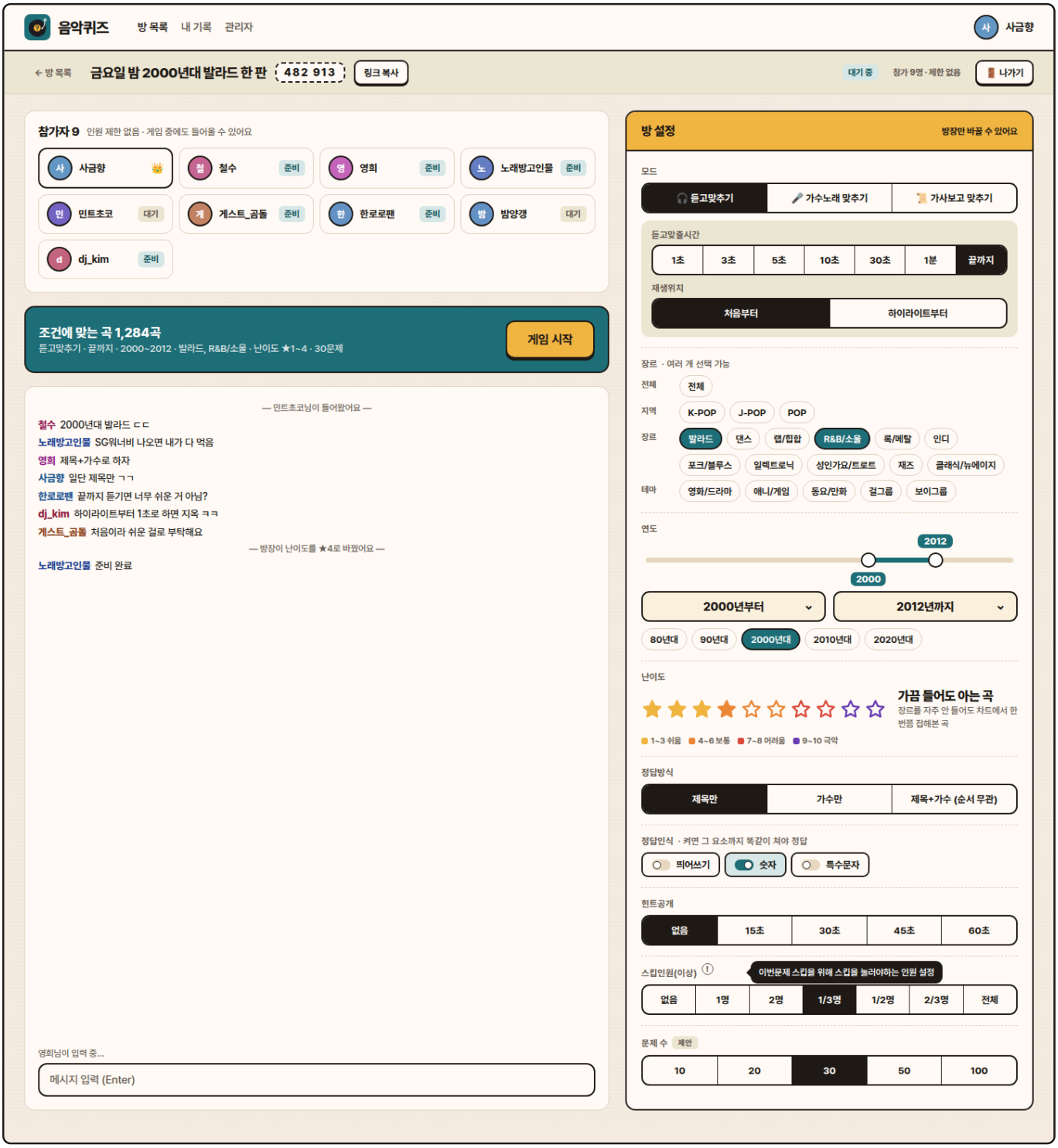
1. **설계(지금)** → 답변 받으면
2. 뼈대 + **배포 먼저**: 실제 주소에서 디스코드 로그인(아이콘 보이는 동의 화면) 확인
3. 노래 DB 1차(약 9천 곡) · 방/로비 · 게임 핵심 · 채팅을 차례로, 단계마다 gitea에 커밋
4. 힌트·스킵·모드 3종 → 관리자 완성 → 수집 확장(1만 곡+) → 휴대폰·다크 테마·부하 테스트

2. 화면 예시

실제 앱이 아닌 디자인 확인용 예시입니다. 곡·점수·통계 숫자는 모두 예시 데이터입니다.



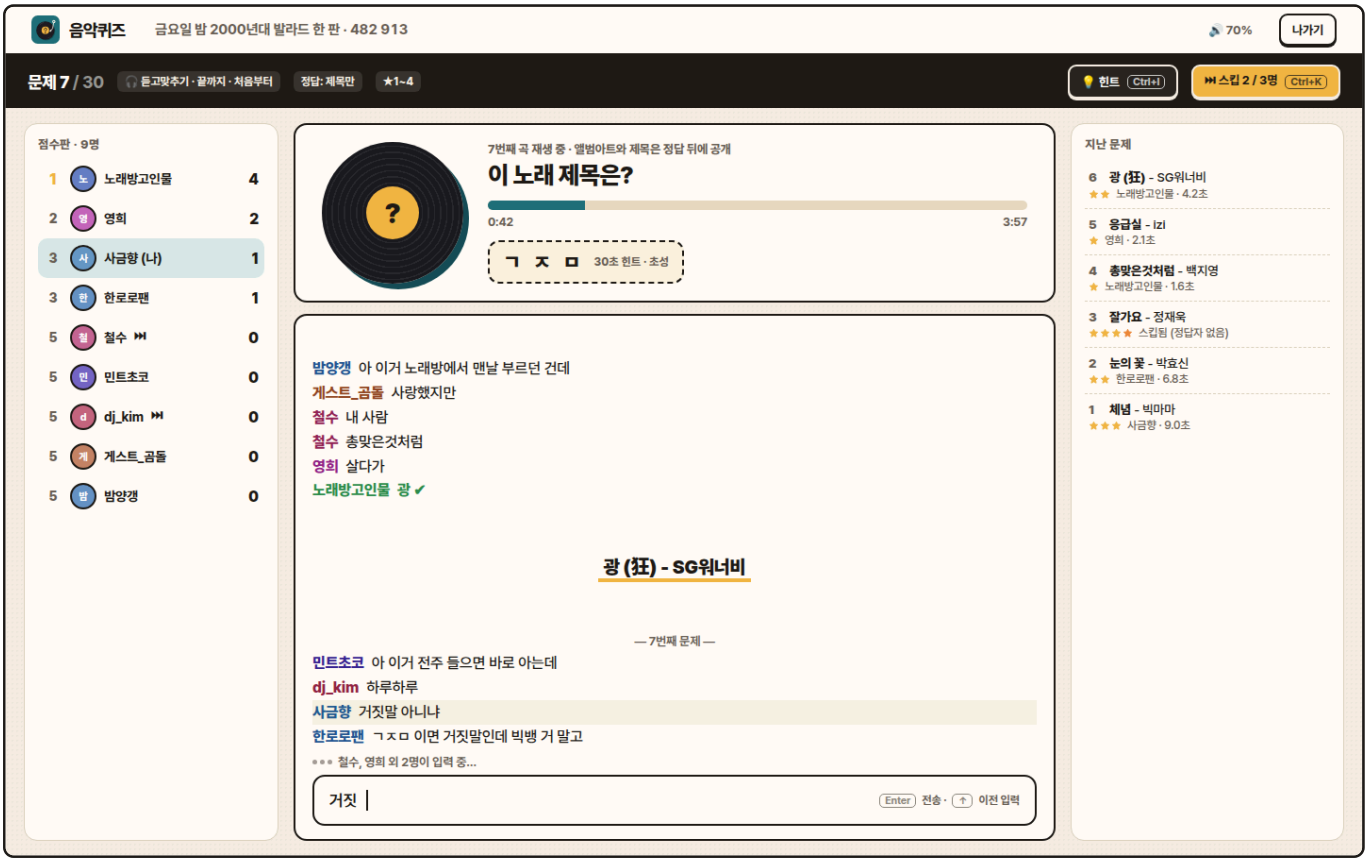
첫 화면 디스코드 로그인/게스트 시작, 열린 방 목록(대기·진행 중, 중간 참여).



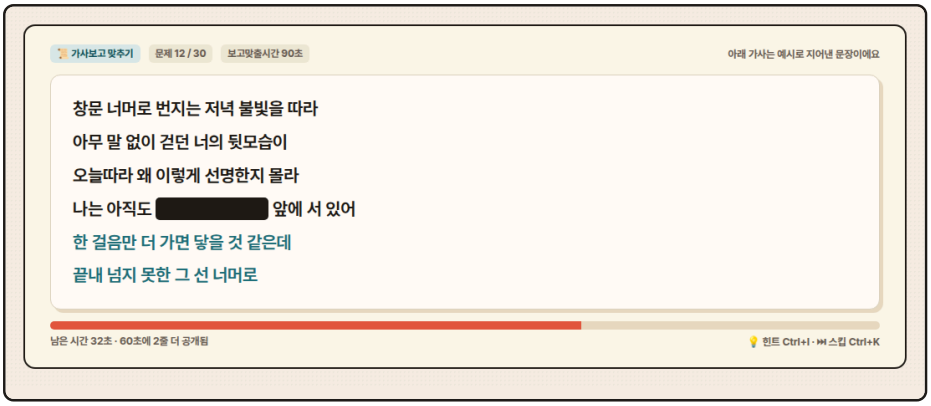
대기실 + 방 설정 인원 제한 없음. 모드·세부모드·장르·연도 슬라이더·난이도 별 10개·정답방식·정답인식 토글·힌트·스킵인원(느낌표 말풍선)·문제 수.



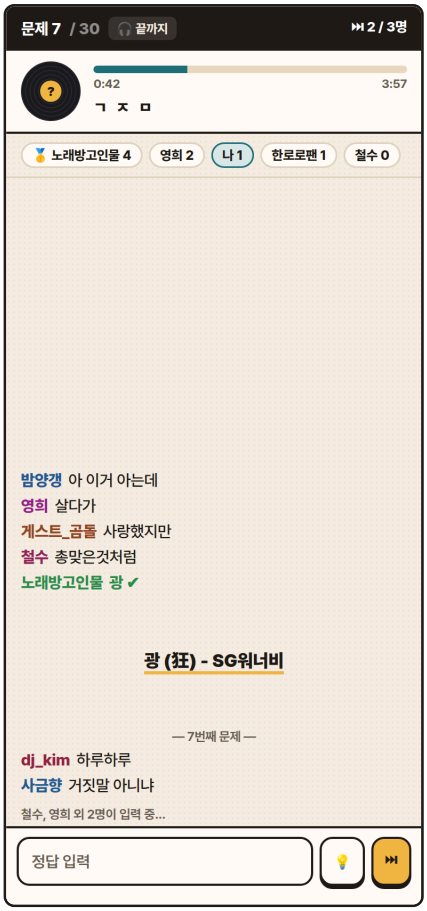
모드별 세부 설정 듣고맞추기(듣는 시간·재생위치) / 가수노래 맞추기(검색해서 추가, 누르면 빠짐) / 가사보고 맞추기(보는 시간).



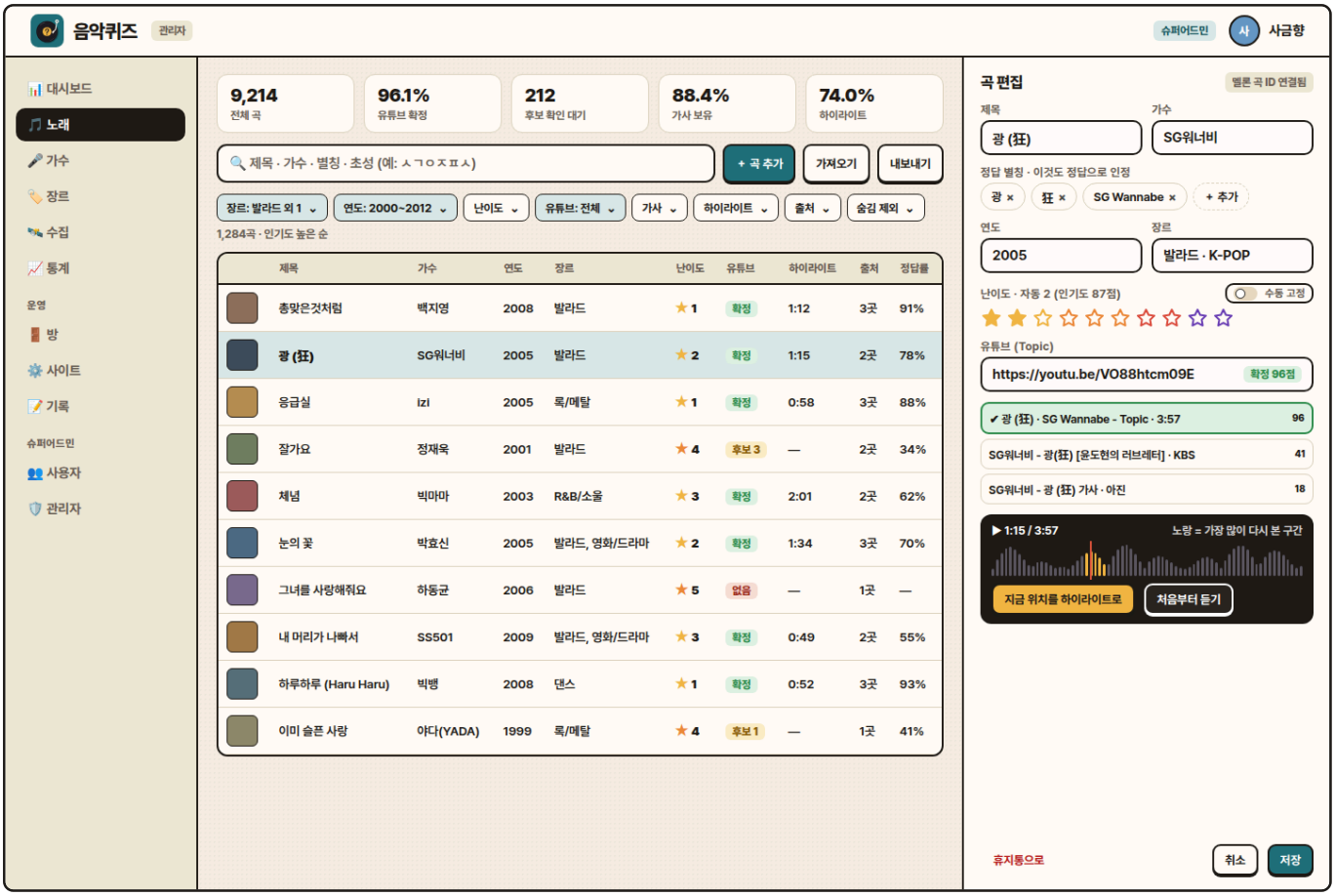
게임 화면 — 듣고맞추기 채팅을 가운데 아래에 가장 크게. 정답 줄 아래 공백 5줄 가운데에 "노래제목 - 가수". 입력 중 표시, 힌트(Ctrl+I)·스킵(Ctrl+K) 버튼과 투표 수.



가사보고 맞추기 무대 가사 4줄 → 시간이 지나면 2줄씩 추가, 제목 단어는 가림. (가사는 예시로 지어낸 문장)



휴대폰 무대는 위 띠, 점수는 가로 칩, 채팅이 화면 대부분, 💡⏭ 버튼.



관리자 — 노래 관리 검색(초성 포함)·필터·정렬 표 + 오른쪽 편집 패널: 정답 별칭, 난이도 자동/수동, 유튜브 Topic 후보와 점수, 미리듣기에서 하이라이트 지정.

아이콘



사이트 / 앱 아이콘



디스코드 (원형으로 잘림)

음악퀴즈



파비콘 64 / 48 / 32 / 16

사이트·디스코드·파비콘 LP 판 + 물음표 레이블 + 톤암. 청록 바탕·겨자 레이블. 디스코드용은 원형으로 잘려도 보이게 여백을 둠. 16px 파비콘은 단순화 버전.

3. 상세 설계

이하 각 문서는 리포 docs/ 폴더의 md 파일과 같은 내용입니다

- 01 구조와 기술 스택
- 02 로그인 · 권한 · 관리자 페이지
- 03 UI · 디자인
- 04 방 설정
- 05 게임 진행
- 06 채팅
- 07 노래 DB
- 08 유튜브 Topic · 재생 · 하이라이트
- 09 배포 · 운영
- 10 작업 순서

01. 구조와 기술 스택

결정

joke-app과 같은 스택을 쓴다. 이미 운영 중이고 로그인·관리자·채팅·실시간 방이 검증돼 있어서, 같은 틀 위에 음악퀴즈만 새로 얹는 게 가장 빠르고 안전하다.

영역	선택	joke-app과 관계
런타임	Bun 1.3 + TypeScript 5.9, Bun workspaces	같음
서버	Hono 4 (HTTP) + Bun.serve WebSocket `/ws`, zod 4 검증	같음
DB	bun:sqlite (WAL), 번호 붙은 SQL 마이그레이션	같음
웹	React 19 + Vite 8 + react-router + zustand, Pretendard	같음
스타일	손으로 쓴 CSS 변수 한 벌(`global.css`), CSS 프레임워크 없음	구조 같음, 색·형태는 새로
테스트	bun test(단위·통합) + playwright-core e2e(시스템 Chrome)	같음
배포	Docker compose(프록시 모드), NPMplus 뒤	같음, 포트만 2974
음원 처리	yt-dlp + ffmpeg (서버 캐시 재생 방식일 때)	새로 추가
곡 수집기	Bun 스크립트(`tools/collector`), 필요시 Python 보조(하이라이트 분석)	새로 추가

폴더 구조 (예정)

```
music_quiz/
├ apps/
│  ├ server/            # Hono + WS + 게임 진행
│  │  ├ src/auth/       # joke-app에서 가져옴 (discord.ts, sessions.ts, users.ts)
│  │  ├ src/admin/      # roles.ts, audit, settings + 노래 관리 API
│  │  ├ src/rooms/      # 방, 라운드 진행(quiz-engine), 투표
│  │  ├ src/songs/      # 곡 조회·출제 선택·정답 판정(matcher)
│  │  ├ src/audio/      # 음원 캐시, 클립 스트리밍
│  │  └ src/db/migrations/
│  └ web/
│     ├ src/pages/      # Home, Room, Game, Admin/*
│     ├ src/components/ # ChatPanel, StarRating, YearRange, ArtistPicker, ...
│     └ src/styles/global.css
├ packages/shared/      # 프로토콜(zod), 정답 정규화 함수(서버·클라 공용), 장르/난이도 상수
├ tools/collector/      # 차트·플레이리스트 수집, 유튜브 매칭, 분류
├ data/seed/            # sta_songs.json 등 초기 데이터
├ deploy/               # Dockerfile, docker-compose.proxy.yml, npmplus 설정
├ assets/brand/         # 아이콘
└ docs/                 # 이 문서들
```

joke-app에서 가져올 것

- `auth/discord.ts`, `auth/sessions.ts`, `auth/users.ts` — 디스코드 OAuth(state 서명 쿠키), `sid` 세션 쿠키, 게스트 로그인/병합
- `admin/roles.ts` — `SUPERADMIN_DISCORD_IDS` + `admins` 표, 감사 기록(`admin_audit`)
- `ws/rate-limit.ts`, `db/index.ts`, 마이그레이션 001~003(사용자·세션·관리자·게스트 정리)
- 웹: `ChatPanel` / `ChatMessage`, `useTypingSignal`(입력 중 신호), `Sheet`, `Toasts`, `SiteHeader`, `NicknameForm`, `SchemaForm`(관리자 자동 폼), 관리자 레이아웃(`.admin-layout`, `.data-table`)
- 프로토콜 방식: `{t: ...}` JSON 메시지 + zod 판별 유니온, 방 상태 브로드캐스트

가져오지 않을 것 (컴퓨터 대결 제거)

- `packages/shared/src/bots.ts`, `packages/games/src/bots/*`, 서버 `rooms/bots.ts` · `bot-profile.ts` · `luck*.ts` · `ai/*`
- 프로토콜 `addBot/removeBot/botLevel`, 설정 `botChat`, "컴퓨터와 연습" 타일, `/api/quick/*`, `docker-compose.ai.yml`
- 보드게임 엔진(`cards.ts`, `grid.ts`, `betting.ts`)과 게임 카탈로그
- `BotActivity.tsx`는 사람 "입력 중" 표시 부분만 남긴다

바뀌는 점

- **인원 무제한**: joke-app은 좌석(seats) 배열 + 최대 32명. 음악퀴즈는 좌석 개념 없이 `players` 목록으로 관리하고 상한을 두지 않는다. 서버 보호용 상한만 관리자 설정에 두되 기본값은 "없음".
- **방 상태 전송**: 인원이 많아도 버티도록 전체 상태 대신 변경분만 보낸다(joke-app `view-patch.ts` 활용). 점수판은 점수 변동분만, 채팅은 개별 메시지로.
- **게임 진행**: 턴제 엔진 대신 시간 기반 라운드 진행기(`quiz-engine`)를 서버에 둔다. 서버 시계가 기준이고 클라이언트는 `serverNow` 보정값으로 카운트다운을 그린다.

실시간 메시지 (초안)

클라이언트 → 서버 | t | 내용 | |---|---| | `join` / `leave` | 방 입장/퇴장 | | `config` | (방장) 방 설정 변경 | | `ready` / `start` / `kick` / `host` | 준비, 시작, 강퇴, 방장 넘기기 | | `chat` | 채팅 = 정답 입력 (서버가 정답 판정) | | `typing` | 입력 중 on/off | | `hint` | 힌트 요청(Ctrl+I) | | `skip` | 스킵 투표 토글(Ctrl+K) | | `audioReady` | 이번 문제 음원 준비 완료 보고 |

서버 → 클라이언트 | t | 내용 | |---|---| | `room` / `patch` | 방 정보, 변경분 | | `round` | 문제 시작(번호, 모드, 재생 시작 시각, 길이, 음원 토큰 또는 가사 일부) | | `hint` | 힌트 공개(초성 등) | | `vote` | 스킵/힌트 투표 현황 | | `answer` | 정답자, 정답(제목-가수), 점수 변화 | | `chat` / `typing` | 채팅, 입력 중 목록 | | `result` | 게임 종료 결과 |

메모리·디스크 예산 (8GB 규칙)

- 앱 컨테이너 상한 `mem_limit: 1g` (예상 사용 200~400MB)
- 수집기·하이라이트 분석은 `systemd-run --user --scope -p MemoryMax=2G`로 따로 실행, 곡 단위로 스트리밍 처리
- 음원 캐시는 디스크 상한(기본 5GB, 관리자 설정)을 두고 오래 안 쓴 것부터 지운다. .9 남은 용량이 이미 140GB라 원본 전체 보관은 하지 않는다 (11-questions.md 참고)

02. 로그인 · 권한 · 관리자 페이지

로그인 (joke-app과 동일)

디스코드 OAuth2

- 시작: `GET /api/auth/discord/start?next=/r/123456`
- `state` 24바이트 난수 → HMAC 서명 쿠키 `oauth` (10분, `path=/api/auth/discord`)
- `https://discord.com/oauth2/authorize?client_id=1561284577564037140&response_type=code&redirect_uri=https://mquiz.tkrmagid.kr/api/auth/discord/callback&scope=identify&state=...&prc`
- 콜백: `GET /api/auth/discord/callback` → state 검증 → 토큰 교환 → `/users/@me` 조회
- 이미 연결된 계정이면 로그인(게스트 기록은 병합), 게스트 상태면 디스코드 연결 후 회원 전환, 처음이면 회원 생성(닉네임 = `global_name ?? username` , 12자 제한)
- 디스코드 토큰은 저장하지 않는다. scope는 `identify` 만
- **디스코드 개발자 포털에 등록해야 할 Redirect URI**: `https://mquiz.tkrmagid.kr/api/auth/discord/callback` (로컬 테스트용으로 `http://localhost:5173/api/auth/discord/callback` 도 추가 권장)
- 클라이언트 ID/시크릿은 `deploy/.env` (권한 600)에만 넣고 리포에는 절대 커밋하지 않는다. 호스트 비밀 저장소(`secrets.json` 의 `discord.com/oauth/music_quiz`)에 보관해 두었다.
- 클라이언트 자격으로 토큰 발급 테스트 완료 → ID·시크릿 정상, 앱 이름 "음악퀴즈", **앱 아이콘은 아직 비어 있음**(포털에서 `assets/brand/icon-discord-1024.png` 업로드 필요)

게스트

- `POST /api/auth/guest {nickname}` (IP당 분당 20회)
- 닉네임 2~12자, 한글/영문/숫자/ `_-.` , 비속어 필터
- 로그아웃하면 게스트 계정 삭제, 하루 미사용 게스트 자동 정리
- 게스트도 게임은 모두 가능, 관리자는 될 수 없음

세션

- `sid` httpOnly 쿠키(SameSite=Lax, https면 Secure), DB에는 `sha256(token)` 만
- 회원 90일 / 게스트 365일, 하루 한 번 연장
- `/api/*` 쓰기 요청은 `Origin` 검사로 CSRF 방지

권한

역할	정하는 방법	할 수 있는 것
superadmin	환경변수 `SUPERADMIN_DISCORD_IDS` (joke-app과 같은 값: `293719842274541579,1352267557213573160`)	전부 + 사용자 관리 + 관리자 추가/삭제
admin	`admins` 표에 디스코드 ID 등록(로그인 전에도 미리 등록 가능)	노래·수집·방·사이트 설정
user	나머지	게임

- 요청마다 다시 계산(디스코드 연결 ID 기준). 환경변수로 정한 슈퍼어드민은 관리자 목록에 "고정"으로 보이고 삭제 불가
- 모든 관리자 변경은 `admin_audit` 에 변경 전/후 JSON으로 기록

관리자 페이지 `/admin`

joke-app 관리자와 같은 레이아웃(왼쪽 사이드 메뉴 + 오른쪽 내용, 표는 `.data-table`)을 쓴다.

메뉴	내용	권한
대시보드	접속자 수, 진행 중 방, 전체 곡 수, 유튜브 매칭률, 가사 보유율, 최근 수집 작업, 오늘 플레이 수	admin
노래	곡 목록 표 + 상세 편집 (아래)	admin
가수	가수 별칭(아이유 = IU = 이지은), 그룹/유닛 연결, 대표 이미지	admin
장르	장르 목록 추가·이름 변경·순서·숨김	admin
수집	출처별 수집 실행(멜론 연도별 등), 진행률·로그, 자동 실행 주기, 유튜브 매칭/하이라이트 분석 일괄 실행	admin
통계	곡별 출제 수·정답률·평균 정답 시간 → 난이도 조정 제안(승인 버튼)	admin
방	진행 중 방 목록, 참가자, 강제 종료	admin
사이트	사이트 이름, 공지, 점검 모드, 금칙어, 기본 방 설정, 라운드 시간 기본값, 채팅 도배 방지, 음원 캐시 상한	admin
사용자	검색, 닉네임 변경, 정지, 강제 로그아웃, 삭제	superadmin
관리자	디스코드 ID로 추가/삭제	superadmin
기록	감사 로그(누가 언제 무엇을)	admin

노래 관리 화면

- **검색**: 제목·가수·별칭 통합 검색(초성 검색 지원: "ㅅㄱㅇㅈㅍㅅ" → 사건의 지평선)
- **필터**: 장르(복수), 연도 범위, 난이도 범위, 유튜브 상태(매칭됨/후보 있음/없음/재생 불가), 가사 유무, 하이라이트 유무, 출처(멜론·지니·벅스·스타맵·노라보세 등), 활성/숨김

- **정렬**: 최근 추가, 인기도 점수, 난이도, 연도, 정답률
- **표 열**: 앨범아트 · 제목 · 가수 · 연도 · 장르 · 난이도(별) · 유튜브(▶ 미리듣기) · 하이라이트 · 가사 · 출처 수 · 정답률
- **상세 편집(오른쪽 패널)**: 제목, 가수, 정답 별칭(여러 개), 연도, 장르(복수), 난이도(자동값 + 수동 고정), 유튜브 주소(`https://youtu.be/ID`) + 후보 목록에서 고르기 + 미리듣기, 하이라이트 시작점(미리듣기 중 "지금 위치로 지정"), 가사, 메모, 숨김
- **추가**: 제목+가수만 넣으면 자동으로 멜론 정보·유튜브 후보 채움 → 확인 후 저장
- **일괄 작업**: 선택 곡 장르/난이도 변경, 유튜브 재매칭, 숨김, 삭제(휴지통 30일 보관)
- **가져오기/내보내기**: JSON·CSV
- **중복 정리**: 제목·가수 정규화가 같은 곡 묶어서 보여주고 병합

03. UI · 디자인

방향

joke-app의 뼈대(상단 헤더, 방 바, 대기실 2단, 게임 3단, 시트, 토스트, 관리자 레이아웃)는 그대로 쓰고, 색과 형태를 **"동네 레코드 가게"** 느낌으로 바꾼다.

- 종이색 바탕 + 먹색 테두리 + 청록 + 겨자색. LP 판, 레이블 스티커, 점선 티켓 같은 소품을 조금씩만
- "AI가 만든 것 같은" 느낌을 피하기 위해 하지 않는 것: 보라-파랑 그라데이션, 유리(블러) 카드, 네온 글로우, 의미 없는 반짝이 아이콘, 모든 걸 가운데 정렬한 카드 나열, 회색 그림자만 있는 밋밋한 카드
- 대신: 2px 먹색 테두리, 버튼 아래쪽 3px 단색 그림자(눌리는 키캡 느낌), 살짝 어긋난 인쇄 그림자(LP), 아주 옅은 종이 질감 점무늬
- 정보 밀도는 높게. 게임 중엔 채팅이 주인공이라 장식은 최소화

색 (라이트)

이름	값	쓰임
`--bg`	`#F5EFE3`	바탕(종이)
`--surface`	`#FFFDF8`	카드
`--surface-2`	`#EFE6D3`	보조 면, 세부 설정 상자
`--ink`	`#1E1B18`	글자, 테두리, 선택된 버튼
`--text-sub`	`#6B6157`	보조 글자
`--line`	`#DDD2BE`	옅은 구분선
`--primary`	`#1F6F78`	청록 — 주요 버튼, 선택 칩, 진행 막대
`--mustard`	`#F2B544`	겨자 — 게임 시작, 정답 밑줄, 강조
`--coral`	`#E4573D`	진행 중 배지, 가사 모드 시간 막대
`--correct`	`#2F8F4E`	정답 줄

다크 테마도 joke-app처럼 제공(`[data-theme=dark]`): 바탕 `#17150F`, 카드 `#221F1A`, 테두리 `#E9DFC9`, 청록 `#4FA9B2`, 겨자 그대로.

난이도 별 색 (단계가 올라갈수록 뜨거워지는 느낌)

단계	색	느낌
1~3	`#F2B544` 겨자	쉬움
4~6	`#EF8A3A` 주황	보통
7~8	`#DC4B3E` 빨강	어려움
9~10	`#6B3FB5` 보라	극악
빈 별은 같은 색 테두리만, 찬 별은 색으로 채움.		

채팅 이름 색

사용자별 고정 색(이름 해시 → 색상환). 정답 초록과 헷갈리지 않게 연두·초록 구간(60°~160°)은 뺀다.

글꼴 · 크기

- Pretendard 가변(joke-app과 같음). 본문 500, 제목 800, 숫자는 고정폭 숫자(`tabular-nums`)
- 기본 15px, 채팅 16px, 채팅 입력 17~18px, 정답 블록 20px
- 글자 크기 키우기(보통/크게/아주 크게) 개인 설정 유지

형태

- 모서리 14px(카드), 11~12px(버튼·입력), 999px(칩)
- 주요 컨트롤은 2px 먹색 테두리 + 아래 3px 그림자, 누르면 그림자 0 + 2px 내려감
- 터치 영역 최소 44px

아이콘

`assets/brand/` — LP 판 가운데 레이블에 물음표, 오른쪽 위에 톤암. 청록 바탕, 겨자 레이블. - 사이트 헤더 로고, 파비콘(16~64px는 단순화 버전), PWA(192/512), 애플 터치(180) - 디스코드 앱 아이콘: `icon-discord-1024.png` (꽉 찬 사각형, 디스코드가 원으로 자름)

화면별 구성 (예시 이미지: `docs/mockups/out/*.png`)

첫 화면 `/`

- 왼쪽 큰 카드: 한 줄 소개 + `방 만들기` `코드로 참여`
- 오른쪽: 디스코드 로그인 / 게스트 닉네임

- 아래: 열린 방 목록(대기/진행 중 필터, 중간 참여 버튼, 방 설정 요약 배지)

대기실 `/r/:code`

- 상단 방 바: 방 이름, 6자리 코드(점선 티켓), 링크 복사, 인원 "제한 없음", 나가기
- 왼쪽: 참가자 격자(방장 👑, 준비 배지), "조건에 맞는 곡 N곡" + 게임 시작, 채팅
- 오른쪽: 방 설정 카드(겨자색 머리). 방장만 바꿀 수 있고 다른 사람은 읽기 전용

게임 화면

- 맨 위 검은 띠: 문제 번호, 모드 배지, 힌트/스킵 버튼(단축키 표시, 투표 수)
- 가운데 위: 무대(돌아가는 LP + 재생 막대 + 힌트 초성) — 높이 190px로 낮게
- **가운데 아래: 채팅(화면에서 가장 큼)**
- 왼쪽: 점수판 / 오른쪽: 지난 문제(정답, 정답자, 걸린 시간, 난이도)
- 1100px 미만: 양옆 접힘, 휴대폰: 무대는 상단 띠, 점수는 가로 스크롤 칩, 입력창 옆 💡 ⏭ 버튼

관리자 `/admin`

- joke-app 관리자와 같은 사이드 메뉴 + 표 구조, 노래 화면은 표 + 오른쪽 편집 패널(유튜브 후보·미리듣기·하이라이트 지정)

접근성

- 색만으로 구분하지 않기(정답 줄 ✔, 난이도 숫자 병기)
- 대비 4.5:1 이상, 키보드만으로 모든 조작, joke-app의 axe 검사 e2e 그대로
- 소리: 볼륨 개인 설정, 정답/오답 효과음 끄기

04. 방 설정

방장만 바꿀 수 있고, 대기실에서만 바뀐다. 바꾸면 즉시 모두에게 반영되고 준비 상태는 풀린다(joke-app 방식). 설정 아래에 **"조건에 맞는 곡 N곡"**을 실시간으로 보여준다.

한눈에

항목	선택지	기본값
모드	듣고맞추기 / 가수노래 맞추기 / 가사보고 맞추기	듣고맞추기
└ 듣고맞출시간 (듣고맞추기·가수노래)	1초, 3초, 5초, 10초, 30초, 1분, 끝까지	끝까지
└ 재생위치 (듣고맞추기·가수노래)	처음부터, 하이라이트부분부터	처음부터 (Q6)
└ 가수 선택 (가수노래)	검색해서 여러 명 추가, 누르면 빠짐	없음(1명 이상 필수)
└ 보고맞출시간 (가사보고)	30초, 60초, 90초, 120초, 150초, 무제한	무제한
장르 (복수)	아래 목록	전체
연도	범위 슬라이더 + 시작/끝 선택 상자	전체(가장 오래된 곡 ~ 올해)
난이도	별 10개, 왼쪽부터 채움	3 (Q7)
정답방식	제목만 / 가수만 / 제목+가수(순서 무관)	제목만
정답인식	띄어쓰기 · 숫자 · 특수문자 각각 켜기/끄기	모두 끔 (Q2)
힌트공개	없음, 15초, 30초, 45초, 60초	없음
스킵인원(이상)	없음, 1명, 2명, 1/3명, 1/2명, 2/3명, 전체	1/3명
문제 수 (제안)	10, 20, 30, 50, 100	30 (Q8)
방 공개 (제안)	공개 / 비공개(코드·링크로만) / 비밀번호	공개

모드

듣고맞추기

- 노래를 듣고 맞힌다. 기본 모드
- **듣고맞출시간**: 재생 길이. "끝까지"는 곡이 끝날 때까지
- **재생위치**: 처음부터 / 하이라이트부분부터 (하이라이트 구하는 법은 08 문서)

가수노래 맞추기

- 고른 가수들의 노래만 출제. 듣는 방식은 듣고맞추기와 같음(듣고맞출시간·재생위치 함께 보임)
- **가수 선택 UI**: 검색창에 이름 입력 → 아래로 후보(이름, 곡 수, 주 장르) → 누르면 위쪽 "선택한 가수" 줄에 칩으로 추가 → 칩을 다시 누르면 빠짐
- 별칭 검색 지원(IU → 아이유, SG Wannabe → SG워너비), 초성 검색(ㅂㅂ → 빅뱅)
- 그룹과 솔로 연결(태연 선택 시 "소녀시대 곡도 포함" 체크는 추후)
- 이 모드에서도 장르·연도·난이도 필터는 그대로 적용(원하면 "전체"로)

가사보고 맞추기

- 음악 없이 가사 일부를 보고 맞힌다
- **보고맞출시간**: 문제당 제한 시간. "무제한"이면 정답·스킵까지
- 가사 4줄 → 시간이 갈수록 2줄씩 추가, 제목 단어는 ○로 가림

장르 (여러 개 선택)

멜론 장르 분류와 노라보세 분류를 합쳐 정리. 곡 하나가 여러 장르를 가질 수 있다.

묶음	장르	멜론 코드
전체	전체	—
지역	K-POP	한국대중음악 GN0100~0800
	J-POP	GN1900
	POP	GN0900 해외POP
장르	발라드	GN0100
	댄스	GN0200
	랩/힙합	GN0300
	R&B/소울	GN0400
	록/메탈	GN0600
	인디	GN0500

묶음	장르	멜론 코드
	포크/블루스	GN0800
	일렉트로닉	(세부 장르 일렉트로니카)
	성인가요/트로트	GN0700
	재즈	GN1700
	클래식/뉴에이지	GN1600, GN1800
테마	영화/드라마	GN1500 OST(국내외 영화·드라마)
	애니/게임	GN1506 애니메이션/게임 (+ 게임 BGM)
	동요/만화	GN2200 어린이 + 만화 주제가
	걸그룹	가수 정보(그룹·여성)로 자동 분류
	보이그룹	가수 정보(그룹·남성)로 자동 분류

- 관리자 페이지 "장르"에서 추가·이름 변경·숨김 가능(예: 뮤지컬, CCM, 국악은 곡이 쌓이면 추가)
- **고르는 규칙**: 같은 묶음 안에서는 "또는", 다른 묶음끼리는 "그리고"
- 예) [K-POP] + [발라드, 댄스] → 국내 곡 중 발라드 또는 댄스
- 예) [발라드] 만 → 국내외 발라드 전부
- "전체"를 누르면 나머지 선택이 풀림 (Q9)

연도

- 참고 이미지 1.png 방식: 막대 위 두 손잡이(시작은 아래 말풍선, 끝은 위 말풍선) + 아래에 "OOOO년부터", "OOOO년까지" 선택 상자 2개
- 손잡이를 끌거나 상자에서 고르면 서로 맞춰서 움직임
- 빠른 선택 칩: 80년대, 90년대, 2000년대, 2010년대, 2020년대 (누르면 범위가 그 10년으로)
- 범위: DB에서 가장 오래된 곡 연도 ~ 올해
- 연도는 **발매일 기준**(리메이크는 리메이크 발매 연도). sta_songs.json은 803곡 중 528곡이 연도가 없고, 적힌 연도도 맵 제작자가 넣은 값이라(2026년으로 적힌 곡만 26곡) 전부 멜론 발매일로 다시 맞춘다

난이도 (별 10개)

참고 이미지 2.png 방식: 빈 별 10개를 왼쪽부터 채우고, 오른쪽에 단계 이름과 설명.

단계	이름	설명
1	누구나 아는 대표곡	장르를 전혀 안 듣는 사람도 후렴을 따라 부를 수 있는 곡
2	길거리 단골 노래	방송, 매장, 길거리에서 워낙 자주 나와서 귀에 익은 곡
3	한 번쯤 들어본 히트곡	차트 상위권에 올라서 제목은 몰라도 대부분 들어본 곡
4	가끔 들어도 아는 곡	장르를 자주 안 들어도 차트에서 한 번쯤 접해본 곡
5	장르 팬 필수곡	크게 뜨진 않았지만 장르 팬이라면 다 아는 곡
6	뒤늦게 알려진 곡	입소문이나 역주행으로 뒤늦게 조금 알려진 곡
7	고인물이 아끼는 숨은 명곡	차트 성적은 없지만 고인물들 사이에서 명곡으로 꼽히는 곡
8	찾아 들어야 아는 곡	어디서도 들을 일이 없어서 직접 찾아 들은 사람만 아는 곡
9	골수 리스너 인증곡	고인물 중에서도 극소수만 들어본 곡
10	인간 샤잠만 가능	들어본 사람 자체가 거의 없는 곡

- 별 색: 1~3 겨자 / 4~6 주황 / 7~8 빨강 / 9~10 보라 (03 문서)
- 마우스를 올리면 그 단계 이름·설명 미리보기, 누르면 확정
- **선택 의미(제안)**: 별 N개 = "N단계까지(1~N)" 출제. 옆에 "이 단계만" 토글을 두면 N단계 곡만 (Q7)

정답방식 · 정답인식

- 정답방식: 제목만 / 가수만 / 제목+가수 — 제목+가수는 **순서 상관없이** 한 줄에 둘 다 (05 문서 판정 규칙)
- 정답인식: 버튼 3개, 누를 때마다 켜짐/꺼짐
- **켜짐 = 그 요소까지 정확히 쳐야 정답**, 꺼짐 = 무시하고 비교 (해석 확인: Q2)
- 대소문자는 항상 무시

힌트공개

- 문제 시작 후 그 시간이 지나면 초성 힌트 자동 공개. 없음 = 자동 공개 안 함
- Ctrl+I 힌트 요청 규칙은 05 문서·Q3

스킵인원(이상)

- 글자 오른쪽 위에 동그라미 속 느낌표 아이콘 ⓘ(!) — 마우스를 올리면 말풍선: **"이번문제 스킵을 위해 스킵을 눌러야하는 인원 설정"**
- 기준 인원은 현재 접속 참가자 수 기준 올림 (05 문서)

설정 저장

- 방 설정은 방이 살아있는 동안 유지, `다시 하기` 도 같은 설정
- 개인 "최근 설정 불러오기"(로그인 사용자): 방 만들 때 내가 마지막에 쓴 설정으로 시작
- 관리자는 사이트 기본 설정(새 방의 기본값)과 잠금(특정 항목 고정)을 바꿀 수 있음

05. 게임 진행

큰 흐름

```
대기실 ─(방장 시작)→ 출제 목록 뽑기 → [문제 준비 → 문제 진행 → 정답 공개] × N → 결과
```

1) 출제 목록 뽑기

- 방 설정(모드·장르·연도·난이도·가수)에 맞는 곡을 DB에서 고른다. 유튜브 주소가 확인된 곡만(가사 모드는 가사가 있는 곡만)
- 같은 게임 안에서 중복 없음, 같은 방에서 최근 3게임에 나온 곡은 뒤로 미룬다
- 같은 가수가 연달아 나오지 않게 섞는다(가수노래 맞추기 모드 제외)
- 조건에 맞는 곡이 문제 수보다 적으면 시작 전에 "조건에 맞는 곡이 N곡뿐이에요"라고 알려주고 그대로 시작/설정 변경 선택

2) 문제 준비 (최대 5초)

- 서버가 다음 곡 음원을 미리 준비(현재 문제 진행 중에 다음 2곡을 미리 받아 둠)
- 클라이언트는 음원을 받아 재생 준비 후 `audioReady` 보고
- 전원 준비 또는 5초 경과 시 시작(늦은 사람은 늦게 들어온 만큼 건너뛰고 재생)
- 화면: "3번째 문제" 카드 + 3·2·1 카운트

3) 문제 진행

모드	화면	끝나는 조건
듣고맞추기	LP가 돌아가고 재생 막대(남은 시간), 앨범아트·제목은 가림	정답 / 스킵 통과 / 제한 시간
가수노래 맞추기	위와 같음 + 상단에 "선택한 가수" 표시	정답 / 스킵 / 제한 시간
가사보고 맞추기	음악 없이 가사 카드(제목 단어는 ○로 가림), 남은 시간 막대	정답 / 스킵 / 보고맞출시간

- **듣고맞출시간**이 1초~1분이면 그만큼만 재생하고 멈춘다. 재생이 끝나도 문제는 계속(채팅으로 계속 맞힐 수 있음). `다시 듣기` 버튼으로 같은 구간을 본인만 다시 들을 수 있다(최대 3회, 11-questions.md Q4)
- **끝까지**면 곡이 끝날 때까지 재생, 곡이 끝나면 문제 종료
- 문제 하나의 최대 시간(제한 시간): 짧은 클립 모드 기본 60초, 끝까지 모드는 곡 길이, 관리자 설정에서 변경

4) 정답 공개 (5초)

- 무대에 앨범아트 + `노래제목 - 가수` + 연도·장르·난이도 별 표시, 정답자 이름과 걸린 시간
- 듣고맞추기면 음악은 그대로 이어서 5초 더(하이라이트 구간이면 그 뒤로)
- 채팅에 5줄 정답 블록(06-chat.md)
- 점수판 애니메이션(순위 바뀌면 자리 이동)

5) 결과

- 순위, 점수, 맞힌 곡 수, 가장 빨리 맞힌 곡
- 이번 게임 곡 목록(유튜브 링크 `https://youtu.be/ID` 포함) — 몰랐던 노래 다시 듣기
- `다시 하기` (같은 설정으로 대기실 복귀), `나가기`

정답 판정

정규화 (서버·클라이언트 공용, `packages/shared/answer.ts`)

1. 유니코드 NFKC(전각→반각), 영문 소문자
2. 괄호 안 내용 분리: `사랑을 했다(LOVE SCENARIO)` → 정답 후보 `사랑을 했다`, `LOVE SCENARIO`, `사랑을 했다 LOVE SCENARIO`
3. `feat.`, `(Prod. …)`, `[… OST]`, `(원곡: …)` 같은 꼬리 제거한 후보도 추가
4. 관리자가 넣은 별칭 추가(예: `좋은 날` ↔ `Good Day`, `IU` ↔ `아이유`)
5. 정답인식 설정에 따라 비교 전에 제거 - **띄어쓰기** 끔 → 공백 무시 (`사건의지평선` = `사건의 지평선`) - **숫자** 끔 → 숫자 무시, 켬 → 숫자까지 정확히 (`24시간이 모자라`) - **특수문자** 끔 → `!?.,'"()-~·…♡` 등 무시 (`Bubble Pop!` = `bubble pop`) - (해석은 11-questions.md Q2에서 확인)
6. **빈 비교 키 방지** — 숫자·특수문자를 지우면 정답이 사라지는 곡이 실제로 있다(`0+0`, `160`, `365`, `3! 4!`, `11:11`, `24시간이 모자라`의 `24` 등) - 정답 후보마다 비교 키를 만들 때, 지운 뒤 **키가 비거나 한글·영문이 하나도 남지 않으면** 그 후보는 숫자를 보존하고, 그래도 비면 특수문자까지 보존한 키를 쓴다 (`0+0` → 숫자 보존 `00` → 그래도 모호하면 원문 `0+0`) - 채팅 쪽도 같은 규칙을 후보별로 똑같이 적용해서 비교한다(후보가 숫자를 보존했으면 채팅도 숫자를 보존) - **빈 문자열끼리는 절대 일치로 보지 않는다**: 채팅 키나 후보 키 중 하나라도 비면 그 비교는 오답 - 최소 길이: 후보 원문이 2글자 이상인데 키가 1글자만 남는 경우도 숫자·특수문자 보존 키로 비교 - 단위 테스트에 sta_songs.json의 숫자·기호 제목(`0+0`, `160`, `365`, `365일`, `3! 4!`, `3!4!`, `11:11`, `2 L0VE`, `A`, `Y`)을 고정 사례로 넣는다

정답방식별 판정

방식	정답 조건
제목만	채팅(정규화) = 제목 후보 중 하나
가수만	채팅 = 가수 후보 중 하나. 여러 명이 부른 곡(`허각 / 정은지`)은 그중 한 명만 맞혀도 정답

방식	정답 조건
제목 +가수	한 줄에 둘 다, **순서 상관없이**. `광 sg워너비`, `sg워너비 광`, `sg워너비-광`, `광/sg워너비` 모두 정답. 판정: 채팅을 정규화한 뒤 `제목+가수` 또는 `가수+제목` 형태로 정확히 나뉘는지 검사(사이 구분자 `- / ,` 공백 허용)

- 대소문자 무시, 앞뒤 공백 무시
- 정답이 아닌 채팅은 그냥 채팅으로 보인다
- 오타 허용(유사도 판정)은 하지 않는다 — 유즈맵처럼 정확히 쳐야 맞음. 단, 관리자 별칭으로 흔한 표기 차이를 흡수

점수

- 기본: 먼저 맞힌 1명 +1점 (유즈맵 방식)
- 동점이면 맞힌 문제의 평균 정답 시간이 짧은 사람이 위
- (선택) 추가 규칙은 11-questions.md Q5

힌트

- **자동 공개**: 방 설정 "힌트공개" 시간(15/30/45/60초)이 지나면 공개. "없음"이면 자동 공개 안 함
- **Ctrl+I 요청**: 누르면 힌트 투표 1표. 스킵과 같은 인원 기준을 넘으면 즉시 공개 (11-questions.md Q3)
- 힌트 내용(단계별) 1. 초성: `ㄱ (ㄱ) - ㅅㄱㅇㄴㅂ` (정답방식에 해당하는 부분만, 띄어쓰기 유지) 2. (가사 모드) 가사 2줄 더 공개
- 힌트가 공개된 뒤 맞혀도 점수는 같음(옵션으로 0.5점 검토)

스킵

- Ctrl+K 또는 `스킵` 버튼 = 내 스킵 표 토글
- 기준: 없음(스킵 불가) / 1명 / 2명 / 1/3 / 1/2 / 2/3 / 전체 — **현재 접속 중인 참가자 수 기준, 올림**
- 예) 7명, 1/3 → 3명 이상 누르면 스킵
- 진행 중 나간 사람은 표와 분모에서 빠짐
- 화면에 `스킵 2/3` 같이 현재 표 수 표시, 누른 사람 이름 옆에 ⏭
- 통과하면 정답 공개(정답자 없음) 후 다음 문제

중간 입장 · 퇴장

- 게임 중에도 입장 가능(0점으로 바로 참가), 다음 문제부터 음원 재생
- 나갔다 60초 안에 돌아오면 점수 유지
- 방장이 나가면 가장 먼저 들어온 접속자가 방장

방장 권한

- 설정 변경(대기실에서만), 시작, 강퇴, 방장 넘기기, 게임 중 `일시정지` / `게임 끝내기`

06. 채팅

채팅은 이 게임의 입력 장치다. 정답도 채팅으로 치고, 분위기도 채팅에서 나온다. 그래서 화면에서 가장 크고 잘 보이는 자리에 둔다.

배치

- 게임 화면 **가운데 아래**에 넓게(PC 최대 폭 880px, 화면 높이의 약 55%). 위쪽 무대(LP/가사 카드/힌트)는 높이를 줄여 채팅에 자리를 내준다.
- 입력창은 채팅 영역 맨 아래 가운데, 높이 56px, 글자 18px. 게임 중에는 **항상 포커스**(아무 키나 누르면 입력창으로 이동, joke-app의 "Enter로 채팅 포커스" 확장).
- 왼쪽: 점수판, 오른쪽: 지난 문제 목록. 1100px 미만에서는 양옆이 접히고 채팅이 전체 폭을 쓴다. 휴대폰에서는 무대를 상단 띠로 줄이고 채팅이 화면 대부분.
- 대기실에서도 같은 채팅 패널을 쓴다(위치만 오른쪽 아래).

말풍선 대신 "한 줄 로그"

joke-app은 말풍선이지만, 빠르게 많이 오가는 정답 채팅에는 유즈맵처럼 **한 줄 로그**가 훨씬 잘 읽힌다.

```
사금향 : 광
철수   : sg워너비 광
영희   : 아 이거 뭐였지
```

- 이름 색은 사용자별 고정 색상(joke-app `--name-h` 방식), 본인 줄은 옅은 배경
- 시스템 줄(입장/퇴장/스킵/힌트)은 회색 작은 글씨로 가운데
- 최근 200줄 유지, 위로 스크롤하면 자동 스크롤 멈춤 + "새 메시지 ↓" 버튼

입력 중 표시

- joke-app `useTypingSignal` 그대로: 입력 시작 시 `{t:'typing', on:true}`, 2초마다 갱신, 비우거나 보내면 off
- 서버는 1초에 한 번으로 묶어서 중계
- 입력창 바로 위에 표시: `철수님이 입력 중…`, 2명이면 `철수, 영희님이 입력 중…`, 3명 이상이면 `철수, 영희 외 3명이 입력 중…` (점 세 개 애니메이션)

지연 없는 전송

- 보내는 즉시 내 화면에 먼저 표시(낙관적 표시), 서버 확인이 오면 확정 — 체감 지연 0
- 서버는 받은 채팅을 **정답 판정과 동시에** 바로 방 전체에 중계(별도 큐·지연 없음)
- 도배 방지는 지연 대신 토큰 버킷(초당 6회, 순간 12회)으로만 막고, 넘치면 그 줄만 거절하고 안내
- WebSocket 한 연결로 처리, 메시지 크기 200자 제한(joke-app `CHAT_MAX`)
- 정답 판정 순서는 **서버 도착 순서**가 기준(동시에 맞히면 먼저 도착한 사람)

정답 표시 (스타크래프트 방식)

누군가 정답을 치면 채팅에 그 줄을 정답 색(초록 굵게)으로 표시하고, 바로 아래에 **공백 5줄**을 넣고 그 가운데(3번째 줄)에 정답을 크게 보여준다.

```
철수   : sg워너비 광          ← 정답 줄(초록 + ✔)
                                ← 공백 1
                                ← 공백 2
        광 (狂) - SG워너비      ← 공백 3 자리에 정답 (가운데, 크게, 금색 밑줄)
                                ← 공백 4
                                ← 공백 5
영희   : 아 이거였네
```

- 정답 문구 형식은 항상 `노래제목 - 가수`
- 스킵으로 넘어간 문제도 같은 5줄 블록으로 정답을 보여주되 회색(정답자 없음)
- 5줄 블록은 채팅 기록에 그대로 남아서 스크롤로 다시 볼 수 있다

단축키

키	동작
Ctrl + I	힌트 요청
Ctrl + K	스킵 투표 (다시 누르면 취소)
Enter	전송
Esc	입력 지우기
↑	내가 마지막에 보낸 말 다시 불러오기

- 브라우저 기본 동작(Ctrl+K = 주소창 검색, Ctrl+I = 기울임 등)은 게임 화면에서만 막는다(`preventDefault`)
- 화면에 버튼도 같이 둔다: `힌트 (Ctrl+I)` `스킵 (Ctrl+K)` — 휴대폰은 버튼으로

채팅 내용 보호

- 비속어 필터(사이트 설정 금칙어), 방장 채팅 끄기 옵션은 두지 않음(채팅이 곧 입력이라)
- 정답 판정 전에 정답과 너무 비슷한 오답(한 글자 차이)은 그대로 공개된다 — 유즈맵 감성 유지. 필요하면 "정답에 가까운 오답 가리기" 옵션을 추후 추가

07. 노래 DB — 수집 · 분류 · 저장

목표

- 1차 목표 **약 1만 곡** (국내 1984~2026 히트곡 중심 + 해외·J-POP·OST·애니/게임 일부)
- 모든 곡에 멜론 곡 ID(기준 키), 발매일, 장르, 길이, 유튜브 Topic 주소, 하이라이트 시작점, 난이도, (가능하면) 가사
- 관리자 페이지에서 모두 확인·수정

수집 출처

출처	가져오는 것	방법	확인 결과 (2026-10-18)
sta_songs.json (스타 노래맞추기 맵 7종)	803곡 제목·가수(275곡은 연도)	파일 그대로	리포 `data/seed/sta_songs.json`
노라보세 norabose.com	8,572곡 멜론 ID·연도·주간 최고순위·장르·난이도(5단계), 곡별 유튜브 ID·가사·작사작곡·길이·지니/벅스 ID	`data/quiz-index.json` + `/api/norabose/songs/{ids}`	받기 성공, 멜론 ID인 것 확인
마추기 machugi.io 음악 퀴즈	퀴즈별 곡의 유튜브 ID, **정답 별칭 목록**, 구간 시작/끝 초	`POST quiz-question-api.machugi.io/api/play/{quizId}`	응답 구조 확인
멜론 연도별 차트	연도별 TOP100 (KPOP/POP)	`chart/age/list.htm?chartType=YE&chartGenre=KPOP&chartDate=YYYY`	100줄 정상
멜론 실시간 차트	TOP100	`chart/index.htm`	100줄 정상
멜론 곡 상세	발매일, 장르, 가사(로그인 없이), 작사·작곡·편곡	`song/detail.htm?songId=`	정상
지니 연도별 / 실시간	연도별 TOP100, 실시간 TOP200	`chart/musicHistory?year=`, `chart/top200`	정상(페이지당 50)
벅스 실시간 / 연도별	실시간 TOP100, 연도별	`music.bugs.co.kr/chart`, `/years/YYYY`	실시간 정상, 연도별은 자바스크립트 화면이라 별도 처리 필요
구글(유튜브) 차트	주간 인기곡(한국), 연말 결산	charts.youtube.com	수집 시 확인
유튜브 연도별 인기 플레이리스트	"YYYY 인기곡/히트곡" 공식·대형 재생목록	yt-dlp 재생목록 추출	수집 시 확인
스포티파이 연도별 플레이리스트	연도별 Top 플레이리스트, 곡 인기도(0~100)	Spotify Web API (앱 키 필요, Q12)	2024년 말부터 스포티파이 공식 플레이리스트는 신규 앱에서 막혀 있어 사용자 플레이리스트로 대체 가능성
노래방 인기차트 (TJ·금영) — 추가 제안	연도별·월별 노래방 인기곡	공개 차트 페이지	"누구나 따라 부르는 곡" 판단에 좋은 신호
써클차트(가온) 연간 — 추가 제안	연간 디지털 차트	공개 페이지	멜론과 교차 확인용

- 요청 속도: 사이트당 2~3초 간격, 하루 상한, 차단 신호(429·로그인 요구·봇 확인) 보이면 그 출처만 중단
- 원본 응답은 `chart_entries`에 그대로 남겨서 다시 분류할 때 재수집하지 않음

처리 순서

```
1. 수집       출처별 수집기 → chart_entries(원본)
2. 곡 맞추기  제목·가수 정규화 → 멜론 검색으로 멜론 곡 ID 확정 → songs에 합치기(중복 병합)
3. 정보 채우기 멜론 상세(발매일·장르·가사·크레딧), 가수 정보(솔로/그룹·성별)
4. 유튜브      Topic 주소 찾기 → https://youtu.be/ID (08 문서)
5. 하이라이트  유튜브 heatmap → 음원 분석 → 30% 규칙
6. 지표        조회수, 차트 성적, 노래방 순위, 스포티파이 인기도
7. 분류        인기도 점수 → 난이도 1~10, 연도, 장르, 테마
8. 검토        자동 확정 못 한 곡은 관리자 "후보 확인 대기"
```

- 수집기는 `tools/collector`의 Bun 스크립트, 관리자 "수집" 메뉴에서 실행(서버가 별도 프로세스로 띄움, 메모리 상한 2GB)
- 단계마다 다시 돌려도 안전(같은 곡 중복 생성 안 함)

곡 맞추기(중복 정리)

- 기준 키: 멜론 곡 ID. 멜론에 없는 곡(게임 BGM, 일부 해외곡)은 내부 ID
- 정규화: NFKC, 소문자, 괄호·feat·Prod·OST 꼬리 분리, 공백·특수문자 제거한 비교 키
- 가수 별칭 사전: `아이유=IU=이지은`, `SG워너비=SG Wannabe`, `에스파=aespa`, `빅뱅=BIGBANG` … (영문/한글 병기 `RESCENE (리센느)` 자동 분리)
- 같은 제목 다른 가수(리메이크·동명곡)는 다른 곡. sta_songs의 원칙과 같음
- 애매하면 자동 병합하지 않고 관리자 "중복 정리" 목록으로

난이도 분류

1) 인기도 점수 (0~100)

묶음	비중	지표
차트	40%	연간 차트 최고 순위(멜론·지니·벅스·써클), 주간 최고 순위·TOP10 머문 기간(노라보세 데이터), 여러 해 차트인, 실시간 차트 재진입
노출	25%	유튜브 조회수(MV + Topic, 로그 스케일, 발매 연도 보정 — 오래된 곡은 조회수가 적게 나오는 만큼 보정)
노래방	15%	TJ·금영 연간/월간 순위
가수 인지도	10%	가수의 차트 히트곡 수, 멜론 팬 수
퀴즈 출현	10%	스타맵·노라보세·마추기 등 다른 퀴즈에 나온 횟수

- SNS(X·인스타·틱톡)는 공식 무료 API가 없어서 1차 버전은 유튜브 지표(쇼츠 포함 조회수)로 대신한다 (Q13)

2) 점수 → 10단계

- 점수 순위 비율로 나눈 뒤 규칙으로 보정 | 단계 | 기준(대략) | |---|---| | 1 | 상위 3% + 연간 차트 TOP10 + 노래방 상위 | | 2 | 다음 5% (방송·매장 노출 많은 곡: 조회수 상위) | | 3 | 다음 10% (연간 차트 TOP30급) | | 4 | 다음 12% (연간 TOP100 진입) | | 5 | 다음 15% (차트 중하위, 장르 내 대표곡) | | 6 | 다음 15% + **역주행 신호**(발매 1년 이상 뒤 차트 진입) | | 7 | 차트 성적 없음 + 조회수 대비 좋아요·댓글 비율 높음(아는 사람은 좋아하는 곡) | | 8 | 차트 없음, 조회수 하위 | | 9 | 조회수 매우 낮음, 퀴즈 출현 없음 | | 10 | 최하위 3~5% |
- **실제 플레이로 보정**: 출제 20회 이상인 곡은 정답률·평균 정답 시간을 보고 단계가 안 맞으면 관리자에게 조정 제안(통계 메뉴에서 승인)
- 관리자가 손으로 고정한 난이도는 자동 계산이 덮어쓰지 않음

연도·장르 분류

- 연도: 멜론 발매일(없으면 지니·벅스, 그래도 없으면 첫 차트 진입 연도)
- 장르: 멜론 장르 문자열 → 사이트 장르로 변환(04 문서 표). 여러 장르 허용
- 지역: 멜론 대분류(국내/해외/J-POP)
- 테마: OST 여부, 애니/게임, 동요, 걸그룹/보이그룹(가수 정보)

DB 구조 (SQLite)

```
songs(id PK, melon_id UNIQUE, title, artist_display, release_date, year, duration_ms,
      youtube_id, youtube_status /*matched|candidate|none|dead*/, youtube_score,
      mv_youtube_id, highlight_ms, highlight_source /*admin|heatmap|analysis|rule*/,
      popularity, difficulty_auto, difficulty_manual, hidden, note,
      created_at, updated_at, deleted_at)
artists(id PK, melon_artist_id, name, name_en, kind /*solo|group|unit*/, gender, debut_year)
song_artists(song_id, artist_id, role /*main|feat*/, PK(song_id, artist_id))
artist_aliases(artist_id, alias, PK(artist_id, alias))
song_aliases(song_id, alias, part /*title|artist*/, source /*auto|admin|machugi*/)
genres(id PK, name, grp /*region|style|theme*/, sort, hidden)
song_genres(song_id, genre_id, source)
lyrics(song_id PK, text, source, fetched_at)
chart_entries(id PK, source, chart, period, rank, raw_title, raw_artist, raw_ref, song_id, fetched_at)
song_sources(song_id, source, ref)            -- 예: sta 맵 이름, 노라보세, 마추기 퀴즈 ID
signals(song_id, kind, value, fetched_at)      -- yt_views, karaoke_rank, spotify_pop ...
youtube_candidates(song_id, video_id, title, channel, duration_ms, score, picked)
play_stats(song_id PK, plays, correct, skipped, avg_answer_ms, updated_at)
collect_jobs(id PK, kind, params_json, status, progress, log, started_at, finished_at)
```

- 사용자·세션·관리자·감사 기록 표는 joke-app 마이그레이션 그대로(02 문서)
- 출제 조회 속도를 위해 `(year)`, `(difficulty)`, `song_genres(genre_id)`, `(youtube_status)` 색인

저작권·운영 메모

- 음원 파일은 재생용 캐시로만 두고(상한 있음), 가사는 퀴즈 표시용 일부만 화면에 노출
- 지인용 비공개 성격의 사이트로 운영(검색엔진 차단 `noindex`), 외부 사이트 데이터는 요청 속도 제한을 지키고 원본을 재배포하지 않음

08. 유튜브 Topic 주소 찾기 · 재생 · 하이라이트

1. 유튜브 Topic 음원 주소 찾기

"Topic" 채널(`가수명 - Topic`)에 올라오는 영상은 음원사가 유튜브에 직접 제공하는 공식 음원(Art Track)이다. 뮤직비디오와 달리 인트로·대사·광고성 장면이 없어서 퀴즈용으로 가장 깔끔하다.

사용하는 것

- **music_bot_v2** (`git.tkrmagid.kr/tkrmagid/music_bot_v2`)의 유튜브 뮤직 검색 코드 `bot/src/utils/api/YoutubeMusic.ts`
- `music.youtube.com/youtubei/v1/search` 를 `WEB_REMIX` 클라이언트로 호출, 쿠키 + `SAPISIDHASH` 로 인증, 커버/MR/노래방 결과를 뒤로 미는 `JUNK_RE` 필터 포함
- **Lavalink** (192.168.10.6:2333, v4.2.2, youtube-plugin 1.18.2 + LavaSrc 4.8.3)
- 후보 검색 보조(`ytsearch:`), 재생 가능 여부·길이 확인(`/v4/loadtracks?identifier=https://youtu.be/ID`)

직접 확인한 사실 (2026-10-18)

시험	결과
Lavalink `ytmsearch:아이유 Love wins all`	HTTP 200, `empty` (music_bot_v2 코드 주석대로 ytmsearch는 비어서 못 씀)
로그인 없이 YT Music 검색 API	아티스트/재생목록만 나오고 "노래" 탭이 아예 없음 → 쿠키 인증이 필요
일반 유튜브 검색 `SG워너비 광(狂)`	1위 `광 (狂)` / 채널 `SG Wannabe - Topic` ✔
일반 유튜브 검색 `이문세 깊은 밤을 날아서`	아티스트 공식 채널의 Art Track(제목이 곡명 그대로) ✔
일반 유튜브 검색 `아이유 Love wins all`	상위 6개에 Topic 없음(MV·라이브만) → YT Music 노래 검색이 꼭 필요한 경우
yt-dlp `https://youtu.be/VO88htcm09E` (광)	채널 `SG Wannabe - Topic` , 설명 `Provided to YouTube by Genie Music` , 길이 237초, **가장 많이 다시 본 구간 75.8초**(후렴)
yt-dlp `JleoAppaxi0` (Love wins all MV)	가장 많이 다시 본 구간 161.5초

매칭 순서 (`tools/collector/youtube-match.ts`)

1. **YT Music 노래 검색**(music_bot_v2 방식, 쿠키 인증) `"{가수} {제목}"` → 노래 결과의 `videoId` - `music.youtube.com/watch?v=ID` 주소를 `www.youtube.com/watch?v=ID` 로 바꾸고 최종 저장은 `https://youtu.be/ID`
2. 못 찾으면 **일반 유튜브 검색** `"{가수} {제목}"` , `"{가수} - Topic {제목}"` → 채널명이 `- Topic` 으로 끝나는 결과 우선
3. 그래도 없으면 **Lavalink** `ytsearch:` 결과 중 `author` 가 `- Topic` 으로 끝나는 것
4. 후보 점수 매기기 - 채널 `- Topic` +50, 영상 설명에 `Provided to YouTube by` +30, 공식 아티스트 채널 +15 - 제목 일치(정규화 후) +30, 가수 일치 +20 - 길이가 멜론 곡 길이와 ±5초 +20, ±15초 초과 −30 - `MV` , `Live` , `커버` , `cover` , `MR` , `inst` , `노래방` , `1시간` , `가사` , `lyrics` , `sped up` , `remix` (원곡이 리믹스가 아닐 때) −40
5. 최고 점수 ≥ 80 → 자동 확정(`matched`), 50~79 → `candidate` (관리자 확인 대기), 그 아래 → `none`
6. 확정된 주소는 Lavalink `loadtracks` 로 한 번 더 재생 가능 확인(지역 제한·삭제 영상 걸러냄), 주 1회 재확인

요청 속도

- 곡당 검색 1~3회, 2초 간격, 하루 최대 3,000곡. 차단 신호(HTTP 429, 봇 확인 페이지) 보이면 자동 중단 후 다음 날 재개

2. 재생 방식

두 가지 중 하나를 고른다 (11-questions.md Q1). **A안 추천**.

A안: 서버 음원 캐시 + 구간 스트리밍 (추천)

- 서버가 `yt-dlp` 로 해당 곡 음원만 받아(opus/m4a, 곡당 3~5MB) 디스크 캐시에 두고, 문제마다 필요한 구간만 `ffmpeg` 로 잘라 일회용 주소(`/api/audio/{토큰}`)로 보낸다
- 장점
- **정답이 새지 않음**: 브라우저에 유튜브 ID나 제목이 전혀 안 감(유튜브 플레이어를 쓰면 개발자도구로 영상 ID가 보여서 검색 한 번이면 정답)
- **광고 없음**: 공식 음원은 수익화 영상이라 유튜브 플레이어로 틀면 문제 도중 광고가 나올 수 있음
- **정확한 길이**: 1초/3초도 정확히 그 길이만 재생, 하이라이트 시작점도 정확
- 모두 같은 순간에 시작(서버 기준 시각에 맞춰 재생)
- 단점: 서버가 음원을 받는 시간(곡당 2~5초, 다음 곡을 미리 받아서 감춤), 디스크 사용(캐시 상한 5GB ≈ 1,200곡, 오래 안 쓴 것부터 삭제), 유튜브 차단 대응 필요(music_bot_v2의 쿠키·poToken 갱신 방식 재사용)

B안: 유튜브 플레이어(IFrame)를 화면 뒤에 숨겨서 재생

- 장점: 서버 부담 없음, 구현 간단
- 단점: 위 A안 장점이 전부 반대(영상 ID 노출, 광고, 버퍼링으로 1초 모드 부정확, 일부 영상은 퍼가기 금지)

3. 재생 위치 "처음부터 / 하이라이트부터"

하이라이트 = 그 곡에서 가장 유명한 부분(대개 후렴). 곡마다 `highlight_ms` 를 저장한다.

구하는 순서 1. **유튜브 "가장 많이 다시 본 구간"** — 조회수가 충분한 영상은 `yt-dlp` 가 재생 구간별 인기도(heatmap)를 준다. 가장 높은 구간 시작점을 쓴다(MV와 Topic 영상 길이가 같으면 MV 쪽 heatmap도 활용) 2. **음원 분석** — 캐시된 음원으로 반복 구간(후렴) 찾기(크로마 자기유사도 방식, Python `librosa` , 곡당 1~2초, 메모리 300MB 이하) 3. **관리자 지정** — 관리자 노래 편집 화면에서 미리듣기 중 "지금 위치를 하이라이트로" 버튼 4. 셋 다 없으면 곡 길이의 30% 지점(대부분 첫 후렴 근처)

- 우선순위: 관리자 지정 > heatmap > 음원 분석 > 30% 규칙

- "듣고맞출시간"이 하이라이트 이후 남은 길이보다 길면 남은 곳까지만

4. 가사 (가사보고 맞추기)

- 멜론 곡 상세 페이지의 가사를 수집해 DB에 저장(가사 없는 곡은 이 모드에서 제외)
- 출제할 때 보여줄 부분: 후렴 아닌 1절 가운데 4줄 → 시간이 지나거나 힌트를 쓰면 2줄씩 추가
- 가사 안에 **제목 단어가 있으면 `○○`로 가림** (예: "사건의 지평선" → "○○○ ○○○")
- 영어 가사만 있는 곡은 그대로, 일본어 곡은 원문(번역 없음)

09. 배포 · 운영

구성

```
인터넷 → NPMplus (192.168.10.3:443, mquiz.tkrmagid.kr, Let's Encrypt)
       → 192.168.10.9:2974 → 컨테이너 music_quiz-app:3000 (Bun)
                                ├─ /data/app.db      (SQLite, WAL)
                                └─ /data/audio-cache (음원 캐시, 상한 5GB)
       ↘ Lavalink 192.168.10.6:2333 (유튜브 검색·재생 가능 확인, 읽기 전용 사용)
```

- 위치: .9 `/home/claude/projects/music_quiz` (joke-app과 같은 방식)
- compose 프로젝트 `music_quiz`, 볼륨 `music_quiz_app-data`, 포트 `2974:3000`
- 갱신: `git pull && cd deploy && docker --context default compose -f docker-compose.proxy.yml up -d --build` (`--context default` 필수 — 기본 컨텍스트가 docker5라서)
- 컨테이너 메모리 상한 1GB, 수집기는 별도 프로세스(상한 2GB)

환경변수 (`deploy/.env`, 권한 600, 커밋 금지)

이름	값
`PUBLIC_ORIGIN`	`https://mquiz.tkrmagid.kr`
`APP_PORT`	`2974`
`SESSION_SECRET`	새로 생성(32자 이상)
`DISCORD_CLIENT_ID`	`1561284577564037140`
`DISCORD_CLIENT_SECRET`	비밀 저장소에서 주입
`SUPERADMIN_DISCORD_IDS`	`293719842274541579,1352267557213573160`
`LAVALINK_URL` / `LAVALINK_PW`	music_bot_v2 설정 재사용
`YT_COOKIE_PATH`	(A안일 때) 유튜브 쿠키 파일 경로
`AUDIO_CACHE_MAX_GB`	`5`

프록시 · 인증서

- DNS: `mquiz.tkrmagid.kr` → CNAME `tkrmagid.kr` (이미 등록돼 있음 확인)
- NPMplus에 프록시 호스트 추가: 대상 `192.168.10.9:2974`, WebSocket 켜기, joke-app과 같은 보안 헤더(`npmplus-advanced.conf` 기반, CSP에 음원 주소 허용)
- 인증서: Let's Encrypt, joke-app 때 쓴 인증서 스크립트 방식 그대로

백업

- 매일 새벽 SQLite `VACUUM INTO` 스냅샷 → 7일 보관(.9) + 주 1회 Nextcloud `/claude/music_quiz/` 로 복사
- 음원 캐시는 백업하지 않음(다시 받으면 됨)

모니터링

- `/healthz` (DB·디스크·Lavalink 연결 상태), 컨테이너 healthcheck
- 관리자 대시보드에 수집 작업 실패, 유튜브 차단 신호, 캐시 사용량 표시
- 주 1회 유튜브 주소 재확인(삭제·지역제한 감지 → "재생 불가"로 표시 후 재매칭)

디스크

- .9 남은 공간이 약 140GB(권장 150GB 이상보다 적음). 음원 캐시는 상한 5GB로 시작, 늘리려면 Q15 결정에 따름

10. 작업 순서

각 단계가 끝날 때마다 gitea `main`에 커밋·푸시하고, 테스트 결과를 남긴다.

단계	내용	끝났다고 보는 기준
0. 설계 (지금)	md 문서, 아이콘, 예시 화면, 보고서.pdf	사용자 확인 + 질문 답변
1. 뼈대	joke-app에서 auth·세션·관리자·DB·UI 키트 가져오기, 봇 코드 제거, 새 색·아이콘 적용	디스코드 로그인 → 아이콘 보이는 동의 화면 → 로그인 성공(e2e), 슈퍼어드민 2명 관리자 접근
2. 배포 먼저	Docker, 2974 포트, NPMplus, 인증서	https://mquiz.tkrmagid.kr 에서 로그인 실제 확인
3. 노래 DB 1차	sta_songs 803곡 + 노라보세 8,572곡 가져오기, 멜론 정보, 유튜브 Topic 매칭	곡 9천+, 유튜브 확정률 90%+, 관리자 노래 목록에서 확인
4. 방·로비	방 목록, 방 만들기, 인원 무제한, 방 설정 전체(04 문서)	설정 변경이 모두에게 즉시 반영, "조건에 맞는 곡 N곡" 정확
5. 게임 핵심	라운드 진행, 음원 재생(Q1 방식), 정답 판정, 점수, 결과	3명 동시 e2e로 30문제 끝까지, 정답 판정 단위 테스트
6. 채팅	가운데 아래 큰 채팅, 입력 중, 5줄 정답 블록, 단축키	체감 지연 없음(로컬 왕복 < 50ms), Ctrl+I/K 동작
7. 힌트·스킵·모드	초성 힌트, 스킵 투표, 가수노래·가사 모드, 하이라이트	모드 3종 e2e
8. 관리자 완성	노래 CRUD·필터·후보 확인·하이라이트 지정, 수집 작업 UI, 통계, 사용자/관리자	관리자 e2e(joke-app admin.e2e 확장)
9. 수집 확장	멜론·지니·벅스 연도별/실시간, 유튜브·스포티파이 플레이리스트, 노래방 차트, 난이도 계산	곡 1만+, 난이도 분포 확인
10. 다듬기	휴대폰 화면, 다크 테마, 접근성 검사, 부하 테스트(100명 방)	axe 위반 0, 100명 방에서 채팅 지연 < 200ms

- 1~2단계를 먼저 해서 **로그인과 아이콘을 실제 주소에서 바로 확인**할 수 있게 한다
- 3단계(노래 DB)는 오래 걸리므로 4~6단계와 나란히 진행

4. 질문 · 요청사항

> 답을 주시지 않은 항목은 **추천안(★)** 으로 진행합니다. 번호만 골라 답해 주셔도 됩니다.
> 예) "Q1 A, Q7 2번, 나머지 추천대로"

해주셔야 하는 일

#	할 일	어디서
R1	앱 아이콘 업로드: `assets/brand/icon-discord-1024.png` (보고서에 같이 첨부)	디스코드 개발자 포털 → 음악퀴즈 앱 → General Information → App Icon
R2	Redirect 주소 추가: `https://mquiz.tkrmagid.kr/api/auth/discord/callback` (개발용 `http://localhost:5173/api/auth/discord/callback` 도 추가 권장)	OAuth2 → Redirects
R3	(선택) 클라이언트 시크릿이 채팅에 올라왔으니, 정식 공개 전에 Reset Secret 후 새 값 전달	OAuth2 → Client Secret
R4	아래 질문 답변	이 채널

게임 규칙

Q1. 노래 재생 방식

- ★ A. 서버가 유튜브 음원을 받아 필요한 구간만 보내기 — 정답(영상 ID) 안 새고, 광고 없고, 1초도 정확
- B. 유튜브 플레이어를 화면 뒤에 숨겨서 틀기 — 간단하지만 개발자도구로 정답이 보이고 광고·버퍼링 문제

Q2. 정답인식(띄어쓰기·숫자·특수문자) 켜짐의 뜻

- ★ 1. 켜짐 = 그 요소까지 똑같이 쳐야 정답, 꺼짐 = 무시 (기본: 모두 꺼짐 = 너그럽게)
- 2. 반대로 켜짐 = 무시해 줌

Q3. Ctrl+I 힌트

- ★ 1. 힌트 투표 — 스킵인원과 같은 기준을 넘으면 모두에게 공개. 힌트공개가 "없음"이어도 투표로는 공개 가능
- 2. 누구든 누르면 바로 모두에게 공개
- 3. 힌트공개가 "없음"이면 Ctrl+I도 막기

Q4. 짧게 듣기(1초~1분)에서 다시 듣기

- ★ 1. 본인만 같은 구간 다시 듣기 3회까지
- 2. 다시 듣기 없음(한 번만)

Q5. 점수

- ★ 1. 먼저 맞힌 1명만 +1 (유즈맵 방식)
- 2. 맞힌 순서대로 3·2·1점(3명까지)
- 3. 제목+가수 모드에서 하나만 맞혀도 0.5점

Q6. 재생위치 기본값

- ★ 1. 처음부터 · 2. 하이라이트부터

Q7. 난이도 별 N개를 골랐을 때

- ★ 1. 1~N단계 모두 출제 (별 4개 = 1~4단계)
- 2. 정확히 N단계만
- 3. 시작·끝 두 번 눌러 범위 선택(예: 4~7)

Q8. "문제 수" 설정 추가해도 될까요? (요청 목록에 없어서 추가 제안)

- ★ 10 / 20 / 30 / 50 / 100, 기본 30

Q9. 장르 여러 개 고를 때 규칙

- ★ 같은 묶음(장르끼리)은 "또는", 다른 묶음(지역 vs 장르)은 "그리고" — 예) K-POP + 발라드 = 국내 발라드
- 테마에 "걸그룹/보이그룹" 추가한 것 괜찮은지

Q10. 가수노래 맞추기의 정답방식

- ★ 방 설정의 정답방식을 그대로 따름(가수를 이미 알고 있으니 보통 "제목만")

데이터

Q11. 다른 퀴즈 사이트 데이터 — 노라보세(8,572곡, 멜론 ID·유튜브 ID·가사 포함)와 마추기 음악 퀴즈(정답 별칭·구간)를 속도 제한 지켜서 가져와도 될까요? ★ 예

Q12. 스포티파이 — Lavalink(LavaSrc)에 이미 들어가 있는 스포티파이 앱 키를 재사용해도 될까요, 새 키를 주실까요? ★ 재사용

Q13. SNS(X·인스타·틱톡) 지표 — 공식 무료 API가 없어서 1차 버전은 유튜브 조회수(쇼츠 포함)로 대신해도 될까요? ★ 예

Q14. 유튜브 인증 — 유튜브 뮤직 "노래" 검색과 음원 받기에는 로그인 쿠키가 필요합니다. music_bot_v2가 쓰는 쿠키·poToken을 같이 써도 될까요? (같은 계정이라 막히면 봇도 같이 영향)
★ 같이 쓰고, 문제 생기면 별도 계정 - 별도 버리는 계정 쿠키를 주시면 더 안전합니다

Q15. 음원 캐시 저장 위치 — .9 남은 공간이 약 140GB입니다.

- ★ 1. .9에 5GB 상한(약 1,200곡, 오래 안 쓴 곡부터 삭제, 필요할 때 다시 받음)
- 2. 상한을 더 늘리기(몇 GB?)
- 3. Nextcloud(.10)에 전곡 보관 + .9엔 작은 캐시

사이트

Q16. 게스트 로그인 — joke-app처럼 게스트(닉네임만) 참여 허용? ★ 예

Q17. 첫 화면 열린 방 목록 — joke-app엔 방 목록 페이지가 없는데, 음악퀴즈는 첫 화면에 열린 방 목록을 보여줄까요? ★ 예 (비공개 방은 숨김)

Q18. 공개 범위 — 친구용으로 검색엔진에는 안 뜨게(noindex) 할까요? ★ 예

Q19. 디자인 — 예시 화면의 색(종이색 + 청록 + 겨자, 먹색 테두리)과 아이콘(LP + 물음표) 방향이 괜찮은지, 바꾸고 싶은 부분