

Concert Review Platform — Developer Spec

Generated: 2025-11-11 00:13 UTC

1. Overview

A web platform where users log concerts they've attended, leave reviews (rating + text + optional media), and connect with friends to see each other's reviews. The goal for the MVP is to provide a simple, fast, mobile-first experience for posting and discovering concert reviews.

2. Recommended Tech Stack

Frontend: Next.js (React) + TailwindCSS. Hosting: Vercel. Backend: Node.js with Express or NestJS. Database: PostgreSQL with Prisma ORM. Auth: Clerk or Supabase Auth. Storage for media: Cloudinary or S3. Optional: Songkick or Bandsintown APIs for concert data prefill.

3. Key User Stories (MVP)

As a user I can:

- Sign up and log in via email or social providers.
- Create and edit my profile (avatar, bio).
- Add a concert I've attended (artist, date, venue, city).
- Leave a review with rating (1-5 stars), text, and optional photo/video.
- Send and accept friend requests to view friends' reviews.
- View a feed of reviews from friends and explore concert pages with all reviews.

4. Database Schema (MVP)

Suggested tables and key fields:

USERS: id (uuid), username (unique), name, email, password_hash (if applicable), avatar_url, bio, created_at

CONCERTS: id (uuid), artist_name, date (date), venue_name, city, source_id (optional), created_by (user id), created_at

REVIEWS: id (uuid), user_id (fk), concert_id (fk), rating (int 1-5), text, media_url (nullable), created_at

FRIENDS: id (uuid), user_id (fk), friend_id (fk), status (pending/accepted/rejected), created_at

LIKES (optional): id, user_id, review_id, created_at

5. API Endpoints (examples)

Auth:

POST /api/auth/signup — body: {email, password, name}

POST /api/auth/login — body: {email, password}

GET /api/auth/me — returns current user

Users / Profiles:

GET /api/users/:id — public profile with recent reviews

PUT /api/users/:id — update profile (avatar, bio)

GET /api/users/:id/reviews — user's reviews

Concerts & Reviews:

GET /api/concerts?artist=&city=&date= — search concerts

POST /api/concerts — create concert (artist_name, date, venue_name, city)

GET /api/concerts/:id — concert detail with reviews

POST /api/concerts/:id/reviews — add review (rating, text, media)

GET /api/reviews/:id — single review

Social:

POST /api/friends/request — body {to_user_id}

POST /api/friends/:id/accept — accept friend request

GET /api/feed — returns recent reviews from friends (paginated)