# Leep Audio MVP Backend Summary

🎯 MVP Flow  
Artist uploads → Fan or Producer remixes → Artist gains exposure → Both engage again

## 🧩 Core Backend Essentials

### 1. Foundation

• Auth & Roles: Register/Login with JWT; roles = fan, artist, producer, admin  
• Ownership Rules: Users can edit or delete their own content  
• Database: Postgres for users, songs, projects, stems, comments, reviews, tips  
• Storage: DigitalOcean Spaces (or S3) for audio/artwork uploads  
• Validation & Logging: Basic input checks and error logs

### 2. Artist Uploads

• Songs: Create, update, publish songs (toggle visibility)  
• Uploads: Upload and store audio/artwork URLs  
Endpoints: POST /songs, POST /songs/:id/publish, GET /songs/:id, POST /uploads

### 3. Fan or Producer Remixes

• Projects: Artists create projects, invite producers  
• Stems: Producers upload stems to projects they’re invited to  
Endpoints: POST /projects, POST /projects/:id/invite, POST /projects/:id/stems

### 4. Artist Gains Exposure

• Discovery: Public feed of published songs  
• Analytics (Lite): Log plays/views, show artist totals + simple trends  
Endpoints: GET /songs?public=1, POST /events, GET /analytics/artist/:id/dashboard

### 5. Both Engage Again

• Fan Interactions: Reviews, comments, tips (recorded only)  
Endpoints: POST /songs/:id/comments, POST /songs/:id/reviews, POST /songs/:id/tips

### 6. Admin Tools

• Moderation: Hide/delete comments, take down songs  
Endpoints: POST /admin/takedown/song/:id, POST /admin/moderate/comment/:id

## 🕒 What You Can Defer (not MVP-critical)

Playlists, full analytics dashboards, search, email invites, chat, payments, Kafka, Redis caching

## ✅ Why These Are Enough

• Artists can upload & publish songs  
• Producers can collaborate & remix  
• Fans can listen, comment, and tip  
• Artists see basic engagement data  
  
This achieves the MVP feedback loop fully.