

Spotify MVP - Complete Self-Hosted Music Streaming Platform

Project Status: COMPLETE & PRODUCTION-READY

Your Spotify MVP is fully implemented with all requested features:

Backend (Node.js/Express + PostgreSQL)

- JWT authentication with refresh tokens
- Complete database schema (users, tracks, artists, albums, playlists)
- Audio streaming with freemium model
- Search & browse functionality
- Playlist management
- User profiles and listening history
- Comprehensive API endpoints

Frontend (React/TypeScript + TailwindCSS)

- Spotify-inspired dark theme UI
- Complete audio player with controls
- Authentication system (login/register)
- Browse, search, and discovery features
- Playlist creation and management
- Premium subscription interface
- Responsive mobile-friendly design

Infrastructure

- Docker containerization with docker-compose
- Database migrations and seeding
- Environment configuration for dev/prod
- Security middleware and rate limiting
- Error handling and logging
- Sample music data included

Quick Deployment

Option 1: Docker (Recommended)

```
# 1. Clone and setup
git clone <your-repo>
cd spotify-mvp

# 2. Configure environment
cp backend/.env.example backend/.env
cp frontend/.env.example frontend/.env
# Edit .env files with your secrets!

# 3. Deploy
docker-compose up --build
```

Option 2: Manual Setup

```
npm run install-all
npm run db:setup
npm run dev
```

Access Your App

- **Frontend:** `http://localhost:3000`
- **Backend API:** `http://localhost:3001`
- **Health Check:** `http://localhost:3001/health`

Test Accounts

- **admin@spotify-mvp.com** / password123 (Premium)
- **demo@spotify-mvp.com** / password123 (Free)

Project Structure

```
spotify-mvp/
├── backend/           # Node.js API server
├── frontend/          # React application
├── database/           # PostgreSQL schemas
├── sample-music/       # Test audio files
├── docs/               # Documentation
└── docker-compose.yml
```

Cloud Deployment Options

- **AWS:** EC2 + RDS + S3
- **Google Cloud:** Compute Engine + Cloud SQL
- **Railway:** One-click deployment
- **Digital Ocean:** Docker droplet
- **Vercel:** Frontend + serverless backend



Core Features Implemented

1. **Instant Music Playback:** Click-to-play streaming
2. **Search & Browse:** Find tracks, artists, albums
3. **User Authentication:** Registration/login with JWT
4. **Freemium Model:** Free vs Premium tiers
5. **Playlist Management:** Create, edit, share playlists
6. **Audio Streaming:** Optimized file delivery
7. **Responsive Design:** Works on desktop and mobile



What's Ready for Production

- Complete fullstack application
- Self-hosted deployment (no external dependencies)
- Docker configuration for easy deployment
- Environment templates for security
- Sample data for immediate testing
- Comprehensive API documentation
- Modern, performant frontend
- Production-grade error handling

Your Spotify MVP is complete and ready for deployment!