

RecoDeck App Summary

One-page, repo-evidence snapshot

What It Is

RecoDeck is a desktop music library app built with Tauri (Rust backend) and a React/TypeScript frontend.

It scans local audio folders, stores metadata in SQLite, and adds playback, analysis, and AI playlist tools.

Who It Is For

Primary persona (inferred from code): digital DJs managing local libraries and playlist/set prep workflows.

What It Does

- Recursive scan/import for mp3, flac, wav, aiff/aif, m4a, and ogg files.
- SQLite-backed tracks, playlists, genres, settings, and analysis data.
- Folder tree + playlist folders with add/remove/rename/delete operations.
- Playback queue with seek, repeat, shuffle, volume, and crossfade.
- BPM + key analysis (Camelot/Open Key) with progress UI and batch runs.
- AI chat + AI playlist generation using cached library context.
- File watcher emits library-changed events for automatic refresh.

How It Works (Architecture)

Components and services

- Frontend: React UI + Zustand stores.
- Bridge: `src/lib/tauri-api.ts` uses `invoke()`.
- Backend: Rust command modules for library, scanner, analysis, playback, playlists, genres, settings, watcher, and AI (`src-tauri/src/commands/*`).
- Data: rusqlite migrations and queries on `recodeck.db` in app data.

Data flow

1. UI action triggers `invoke()` command.
2. Rust services read/write DB and files.
3. Rust emits `library-changed/audio-*` events.
4. Frontend listeners refresh UI state.

How To Run

1. `npm install`
2. `npm run tauri dev`
3. In app, click Scan Folder (or add folders in Settings).

Not Found In Repo

- Explicit persona statement: Not found in repo.
- Node/Rust version requirements: Not found in repo.
- Production deployment guide: Not found in repo.