theatre.ai App Summary (One Page)

Generated from repository evidence on 2026-02-06 - scope includes README, src/, docs/, and config files.

What it is

theatre.ai is a React + TypeScript web app for managing AI-era performance rights in entertainment. It combines CastID-style verification, consent-controlled digital assets, licensing, and collaboration workflows in a single frontend experience.

Who it is for

Primary persona: actors and artists who need to protect and license voice, likeness, and motion rights.
Also represented in product flows: studios, agencies, and legal users via role-based onboarding.

What it does

- Role-based onboarding and sign-in with identity and professional verification states.
- Vault workflow for voice/face/motion uploads, per-asset visibility, consent matrix controls, and usage logs.
- Search and discovery across actors, studios, agencies, and casting calls with filter panels.
- Messaging center with requests, secure attachment presets, referenced assets, and conversation states.
- Contracts workspace with status filters, clause highlights, and mock license certificate generation.
- Payments flow showing studio pay-in, platform fee split, actor payout statuses, and downloadable receipts.
- Feed and network areas for posts, communities, events, and professional connection management.

How it works (repo evidence only)

- Frontend SPA: Vite + React + TypeScript entrypoint at src/main.tsx and route graph in src/App.tsx.
- Routing: react-router-dom with public pages plus dashboard and canonical route aliases.
- UI composition: shared layout components (DashboardLayout, DashboardTopNav) and shadcn-ui/Tailwind primitives.
- State/data layer: src/lib/store.ts uses localStorage helpers and in-repo mock datasets.
- Data flow: user actions update component state, store helpers persist to localStorage, UI re-renders with toast feedback.
- Backend API services, server runtime, and production database: Not found in repo.

How to run (minimal)

1. Install Node.js and npm (README lists this prerequisite).
2. From repo root, install dependencies: npm i
3. Start dev server: npm run dev (vite.config.ts sets port 8080).