

# CNaghsh Web App Summary

## What it is

Next.js web application in the monorepo under apps/web that delivers public listings (jobs, profiles, movies) and internal admin tooling for billing and moderation. It uses Prisma with PostgreSQL and NextAuth-backed sessions for server-side access control.

## Who it is for

Primary persona: internal admin/operator who moderates jobs and manages billing in the web app.

## What it does

- Public jobs listing with filters plus SEO sitemaps and JSON-LD metadata.
- Admin jobs moderation workflow (approve, reject, suspend, feature, close) with audit events and rate limits.
- Billing sandbox for products, plans, prices, checkout sessions, invoices, and entitlements.
- Payment webhooks with idempotent processing and provider normalization.
- Notification fan-out with dedupe and optional email channel.
- Movies catalog pages and admin management endpoints.

## How it works

- UI and routing: Next.js App Router in apps/web/app with server components for public and admin pages; API route handlers live under app/api for billing, webhooks, auth, and data endpoints.
- Data layer: Prisma client (apps/web/lib/prisma.ts, lib/db) backed by PostgreSQL; migrations and seed scripts in apps/web/prisma.
- Auth and access: NextAuth configured in app/api/auth/[...nextauth]/route.ts with getServerAuthSession guards on admin routes.
- Domain services: lib modules for jobs, billing, entitlements, notifications, search; critical transitions run in Prisma transactions.
- Caching and revalidation: unstable\_cache for public job queries and revalidateTag/revalidatePath after moderation updates.
- Background sync: subscription entitlement reconciliation via /api/internal/cron/subscriptions and scripts/check-subscriptions.ts.

## How to run

- pnpm install
- cp apps/web/.env.example apps/web/.env.local
- pnpm --filter @app/web prisma migrate dev
- pnpm --filter @app/web db:seed
- pnpm --filter @app/web dev