**Project Name: CapstoneTV**

**🔧 PHASE 1: PROJECT SCOPING & TECH STACK**

**🎯 Core Features**

| **Module** | **Admin Panel** | **Client Panel** | **Mobile App (React Native)** |
| --- | --- | --- | --- |
| Auth & Roles | ✅ Super Admin, Mods | ✅ Users | ✅ Users |
| Movie Catalog | ✅ CRUD Movies | ✅ Browse/Play | ✅ Browse/Play |
| Categories | ✅ Manage Genres | ✅ Filter/Search | ✅ Filter/Search |
| Subscriptions | ✅ Plans & Billing | ✅ Subscribe/Upgrade | ✅ Subscribe/Upgrade |
| CMS | ✅ Pages & Banners | ✅ View | ✅ View |
| Analytics | ✅ Views/Reports |  |  |
| Localization | ✅ Add Languages | ✅ Multilingual UI | ✅ Multilingual UI |

**🧱 Tech Stack**

| **Layer** | **Tech** |
| --- | --- |
| Frontend (Web) | Next.js 14+, TypeScript, Tailwind, MUI |
| Mobile App | React Native (Expo) with TypeScript |
| Backend API | Node.js + NestJS OR Express.js with TypeScript |
| Realtime Media | HLS/RTMP + Media Server (e.g., Wowza, Nginx-RTMP, or Mux) |
| Authentication | JWT, Refresh Tokens, OAuth (Google, Apple, etc.) |
| Database | PostgreSQL (Prisma ORM) |
| Media Storage | AWS S3, Cloudflare R2, or Firebase Storage |
| DevOps | Docker, NGINX, PM2, GitHub Actions, AWS/GCP |
| Payments | Stripe (subscriptions), In-App Purchases |
| CMS Integration | Optional: Strapi/Direct DB/CMS Table |
| Analytics | Mixpanel, PostHog, or custom dashboards |

**🧩 PHASE 2: MONOREPO STRUCTURE**

Use Turborepo or Nx for code sharing:

bash

CopyEdit

cine-streamx/

│

├── apps/

│ ├── admin-panel/ # Next.js Web UI

│ ├── client-panel/ # Next.js Public UI

│ ├── mobile-app/ # React Native (Expo)

│ └── backend-api/ # NestJS or Express API

│

├── packages/

│ ├── ui/ # Shared UI Components (e.g. MUI + Tailwind)

│ ├── types/ # Shared TypeScript types

│ └── utils/ # Shared utility functions (e.g. formatTime, parseDuration)

│

├── prisma/ # Prisma schema and migrations

├── docker/ # Docker config files

├── .env # Env variables

└── turbo.json / nx.json # Monorepo config

**🧠 PHASE 3: FEATURE MODULES**

**1. Authentication System**

* Admin & User separation
* Email + OAuth Login
* JWT access & refresh tokens (httpOnly)

**2. Media Ingestion & Encoding**

* Upload raw files (admin panel)
* Convert to HLS (.m3u8) via ffmpeg
* Store in CDN (Cloudflare/AWS S3)
* Serve via video.js or react-native-video

**3. Movie Catalog System**

ts

CopyEdit

// Prisma Schema Example

model Movie {

id String @id @default(cuid())

title String

description String

releaseDate DateTime

genres Genre[]

tags String[]

videoUrl String

thumbnailUrl String

duration Int

language String

createdAt DateTime @default(now())

updatedAt DateTime @updatedAt

}

**4. Subscription & Billing**

* Plans: Free, Basic, Premium
* Stripe integration
* Payment tracking and enforcement

**5. Watch History & User Progress**

* Per-user resume data
* Like/watch later lists

**6. Internationalization**

* Support for multilingual UI (i18n)
* Movies tagged with language + subtitle files

**📱 PHASE 4: MOBILE (React Native)**

* Expo + React Navigation
* Reuse API services (axios + RTK Query style)
* Offline mode with local DB (e.g. MMKV)
* Push notifications (FCM, Expo Notifications)
* In-App Purchases (Stripe + Apple/Google billing)

**⚙️ PHASE 5: CI/CD & DEVOPS**

* Docker for containerization
* NGINX reverse proxy
* GitHub Actions / GitLab CI
* Production hosting on Vercel (web) + AWS/GCP (API & encoding)
* Monitor logs: Sentry, LogRocket, CloudWatch

**🧪 PHASE 6: OPTIONAL EXTRAS**

* **AI Recommendations**: Based on watch history
* **Content DRM**: Prevent piracy with signed URLs or DRM providers
* **Community Features**: Comments, Ratings, Reviews
* **Live Streaming**: Add support via RTMP/WebRTC
* **Progressive Downloads**: For slow connections
* **Accessibility**: Subtitles, high-contrast UI

**✅ NEXT STEP**

Would you like me to scaffold the backend with:

* Prisma schema (PostgreSQL)
* NestJS/Express REST API
* Admin auth + movie CRUD?

Or should we begin with frontend app routing (admin-panel or client-panel)?

Let me know what to prioritize:

* **Backend API**
* **Admin Panel**
* **Client Panel**
* **Mobile App**
* **Database Models**

We’ll proceed step-by-step with full code.