Project Overview

Project Name

OTT, Mobile Plan, and Broadband Subscription Management Platform

Objective

To build a unified web and mobile application that allows users to manage OTT subscriptions, mobile plans, and broadband plans in a single platform. The system focuses on subscription tracking, renewals, payments, notifications, and cost optimization.

Platforms

Web Application

- Technology: React.js
- Target: Desktop and mobile browsers
- Purpose:
- Subscription and plan management
- Dashboard and analytics
- Payment tracking

Mobile Application

- Technology: React Native
- Platform: Android (initial phase)
- Type: Hybrid application
- Purpose:
- Subscription tracking on the go
- Renewal reminders and notifications
- Payment visibility

Backend Architecture

Runtime & Framework

- Node.js
- Framework: NestJS (preferred for modular, scalable architecture)

API Style

- RESTful APIs (API-first approach)
- Shared API contracts for web and mobile

Database Strategy

User Database (Confirmed Decision)

- Database: PostgreSQL
- Reason:
- Strong data integrity
- ACID-compliant transactions
- Security and authentication reliability
- Suitable for free-tier hosting and first deployment

User Domain Includes

- Users
- Roles & permissions
- Authentication & authorization
- Sessions / refresh tokens
- Device registrations

Subscription & Plan Data

- Database: MongoDB (planned)
- Reason:
- Flexible schemas for OTT, mobile, and broadband plans
- Easier evolution of plan attributes

Hybrid Database Approach

- PostgreSQL: User, authentication, and critical relational data
- MongoDB: Plans, subscriptions, providers, usage data

User Database – High-Level Schema

Users Table

- id (UUID, primary key)
- email (unique)
- phone (unique, optional)
- password_hash
- name
- status (ACTIVE, BLOCKED, DELETED)
- email_verified
- phone_verified
- created_at
- updated_at
- last_login_at

Core Functional Modules

1. User Management

- Registration and login
- JWT-based authentication
- Role-based access control

2. Subscription Management

- OTT subscriptions
- Mobile plans
- Broadband plans
- Individual and shared subscriptions
- Subscription lifecycle (active, expired, cancelled)

3. Plan Management

- Provider-wise plan catalog
- Plan pricing and validity
- Bundled plans (OTT + Mobile / Broadband)

4. Payment & Billing

- Track payments per subscription
- Single payer or split payments
- Payment history and due tracking

5. Notifications

- Renewal reminders
- Payment due alerts
- Plan expiry notifications

Background Jobs & Notifications

Job Processing

- Tool: BullMQ with Redis
- Use cases:
- Renewal reminders
- Expiry checks
- Payment follow-ups

Push Notifications

- Android: Firebase Cloud Messaging (FCM)

Development & Deployment Strategy

Development Phase

- Local PostgreSQL for user DB
- MongoDB for subscription data
- Docker-based local setup (recommended)

First Deployment (Free Tier)

- PostgreSQL (Supabase / Neon / Render)
- MongoDB Atlas (free tier)
- Node.js backend hosted on Render / Railway

Technology Stack Summary

Layer	Technology
Web Frontend	React.js + TypeScript
Mobile App	React Native (Android)
Backend	Node.js + NestJS
User DB	PostgreSQL
Subscription DB	MongoDB
Cache / Jobs	Redis
Notifications	Firebase Cloud Messaging

Key Architectural Decisions

- Hybrid web + mobile strategy
- API-first backend
- PostgreSQL for user and auth data
- MongoDB for flexible subscription data
- Free-tier friendly for MVP and first deployment
- Scalable to production with minimal migration

Next Planned Steps

- Finalize detailed database schemas
- Define API contracts (Auth, Users, Subscriptions)
- Design MVP feature list
- Create UX wireframes for web and mobile
- Set up repository and CI/CD pipeline