High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **PropTrack - Real Estate Listing & Analytics**. PropTrack is a web platform for real estate listings, advanced search, interactive maps, media uploads, and analytics, built with Next.js, TypeScript, and Tailwind CSS. The document provides an overview suitable for developers and stakeholders, focusing on essential design elements.

---

1. System Architecture Overview

Architecture Description:
PropTrack follows a modular, client-server architecture. The frontend (Next.js) communicates with backend APIs for data operations. State is managed via Redux Toolkit. Media storage and map services are integrated via external APIs.

Module	Role/Responsibility
Frontend (Next.js)	UI rendering, routing, user interaction
State Management	Global state via Redux Toolkit
Backend API	Listing CRUD, user management, analytics, auth
Database	Persistent storage for listings, users, analytics
Media Storage	Stores uploaded images/videos (e.g., cloud storage)
Map Service	Provides interactive map data (e.g., Mapbox/Google)
Authentication	User login, registration, session management

---

2. Component Interactions

Interaction Flow
1. User interacts with UI (search, upload, view listings)
2. Frontend dispatches Redux actions; triggers API calls
3. Backend API processes requests, interacts with DB/media/map services
4. API returns data/results; Redux updates state; UI re-renders
5. Authentication handled via secure API endpoints and session tokens

---

3. Data Flow Overview

Data Flow	Source	Destination	Description

Listing Search/Fetch	Frontend	Backend API	User queries listings; API returns results
Listing Creation/Update	Frontend	Backend API	User submits listing; API stores in DB
Media Upload	Frontend	Media Storage	Images/videos uploaded; URLs stored in DB
Map Data	Frontend	Map Service	Map tiles/geo-data fetched for UI display
User Authentication	Frontend	Backend API	Login/register; session/token management
Analytics Data	Backend API	Database	Usage/events stored for reporting

---

4. Technology Stack

Layer/Function	Technology/Framework
Frontend	Next.js, TypeScript, Tailwind CSS
State Management	Redux Toolkit
Backend API	Node.js/Express (assumed)
Database	PostgreSQL/MongoDB (assumed)
Media Storage	AWS S3/Cloudinary (assumed)
Map Service	Mapbox/Google Maps API
Authentication	JWT/Session-based Auth
Styling	Tailwind CSS

---

5. Scalability & Reliability

- **Scalability:**
  - Stateless frontend and backend enable horizontal scaling.
  - Media and map services leverage scalable cloud APIs.
  - Database indexing and caching for high-volume search.
- **Reliability & Security:**
  - Secure authentication and authorization for all endpoints.
  - Input validation and error handling throughout.
  - Regular backups for persistent data.
  - HTTPS enforced for all client-server communication.

End of Document