Rental Property Discovery App (React Native or Web)

Target a city or region and make property search simple.

Features:

Filter by location, rent, amenities

Map + list view

In-app chat with landlords

Verified badges, saved listings

Monetization:

Paid listing boost for landlords

Google AdMob or Interstitial Ads

Affiliate links for movers/furniture/rent insurance

BONUS TIP: Combine Monetization

You can combine these in your app:

Ads (AdMob/AdSense)

Subscriptions (Stripe or IAP)

In-app purchases (for AI features, documents, themes, etc.)

Referral bonuses (Grow userbase)

Affiliate links (Amazon, Udemy, Health products, etc.)

Project: Rental Property Discovery App

A mobile/web app that lets users search, view, and contact landlords or agents for rental properties. Think of a mini NoBroker or Zillow clone with potential for monetization.

✅ Tech Stack

🧠 Core Tech

Purpose Tech

Frontend (Web) Next.js (with Tailwind)

Frontend (Mobile) React Native + Expo

Backend Supabase (Postgres + Auth)

Images Supabase Storage or Cloudinary

Payments Stripe or Razorpay

Maps Google Maps or Mapbox API

Push Notifications Expo Notifications (mobile only)

Analytics PostHog or Firebase

Ads (Mobile) AdMob (via Expo)

📱 Key Features

👨‍💻 For Users:

🔍 Search by location, price, type, availability

🗺️ Map view and list view of properties

🏠 View detailed property info with photos

❤️ Save favorites

📞 Contact owner/agent

🔔 Notifications for new listings

📄 Rental agreement help (optional)

🏠 For Property Owners:

📝 Post new rental listing

📊 See number of views, contacts

🖊️ Edit/delete listings

💬 Reply to inquiries

Monetization Ideas

Method Notes

AdMob / Google Ads Show ads to free users in app

Premium Listings Owners pay to "highlight" listings

Subscription Plans Users pay for early access or verified listings

Affiliate Services Rental agreements, movers, home services

Referral Bonuses Invite friends, earn credits

🧭 App Flow (UX/UI Structure)

🧑 User App Flow

Splash + Onboarding

Login / Signup

Home

Featured listings

Filters (price, area, BHK, etc.)

Map View / List View

Property Details

Photos, amenities, location, owner info

Contact / Book Visit

Saved Properties

User Profile

Edit info, logout, subscriptions

🧑 Owner Dashboard Flow

Login / Signup

Post Property

Title, rent, address (with map), images

View Listings

Edit / delete options

Upgrade to Premium

Razorpay or Stripe checkout

Messages / Inquiries

Extra Features (Advanced)

Feature

Notes

🔍 AI Chatbot

Ask "Find me 1BHK near Bandra under ₹15k"

🎥 Virtual Tour

Upload walkthrough videos

📃 Auto Lease Gen.

Auto-generate rental agreements

🤖 Smart Alerts

Notify based on saved search

📍 Nearby Facilities

Show schools, gyms, hospitals on map

✅ First Milestone Plan (MVP)

✅ Auth with Supabase (login/signup)

✅ Add listing (title, image, address)

✅ Browse properties (filter, list, details)

✅ View and contact owner

✅ Monetize (ads or pay-to-post)

✅ 1. 📦 Tech Stack

For Mobile (React Native with Expo):

React Native (Expo)

Supabase for Auth + Database + File storage

NativeWind + Gluestack UI for UI

React Navigation with file-based routing

Lucide Icons

Redux Toolkit (optional) or Zustand

For Web (Next.js):

Next.js (App Router)

Tailwind CSS

Supabase client SDK

Shadcn/ui for components

NextAuth or Supabase Auth

✅ 2. 🔢 Features List
🧑 For Renters:

Browse properties by location, price, type, etc.

Search + Filters + Map View

Save favorites

Book a visit or contact owner

Reviews & ratings

🏡 For Owners:

Login / Register

Add/edit/delete property listings

Upload images & documents

Track booking requests

Chat with renters

🔒 Auth:

Email + OTP (Supabase)

Forgot/reset password (deep linking on mobile)

Product Requirements Document: Rental Property Discovery App

✨ TL;DR

Build a mobile-first rental discovery platform targeting urban renters and property owners. The platform enables users to search rentals by location, price, and features; explore listings on a map or list view; contact verified landlords via in-app chat; and optionally handle leasing paperwork. Owners can post listings and upgrade to premium for visibility. Monetized via ads, listing boosts, subscriptions, and affiliate services.

🔍 Problem Statement

Renting a home, especially in densely populated cities, is painful — cluttered listings, poor communication with landlords, and no reliable way to verify property details. Landlords, meanwhile, struggle to get quality leads without paying hefty fees to brokers or classified portals.

🎯 Goals

📈 Business Goals

Generate revenue through ad monetization, subscriptions, and listing boosts

Drive daily/weekly active users (DAU/WAU) with push notifications and engagement loops

Build a verified listings marketplace with high trust

Reduce churn by providing end-to-end rental value chain (search → chat → lease)

🙋 User Goals

Easily search and filter verified rental properties

Communicate directly with landlords without intermediaries

Save favorite listings, receive alerts, and optionally get rental agreement support

🚫 Non-Goals

Supporting property purchases or commercial listings

Managing physical site visits or tenant background checks (for now)

Handling rent payments (not in MVP)

👤 User Stories

Renter

As a renter, I want to search rentals by location and price so I can find relevant properties

As a renter, I want to view properties on a map and a list so I can explore by neighborhood

As a renter, I want to save listings and get alerts for new properties

As a renter, I want to message owners directly inside the app

As a renter, I want to see verified badges to trust listings

Owner

As an owner, I want to list my property with photos and price

As an owner, I want to receive and reply to inquiries

As an owner, I want to boost my listing to get more visibility

As an owner, I want to track listing performance (views, messages)

🧭 User Experience (UX Flow)

📱 Renter App Flow

Splash → Onboarding

Login / Signup via Email OTP

Home Screen

Featured Listings

Search Bar

Filters (Location, Price, BHK, Furnishing)

Toggle: Map ↔ List

Property Details

Photos, Amenities, Verified Badge

Contact Owner (in-app chat or call)

Save to Favorites

Saved Properties

Notifications (New Listings, Chats)

Profile Settings

Edit Info, Logout, Subscription Status

👨‍💼 Owner Dashboard Flow

Login / Signup

Post Property

Title, Rent, Address (Map Picker), Photos

Manage Listings

Edit/Delete, Track Views

Promote Listing (Upgrade via Razorpay/Stripe)

Messages / Inquiries

📖 Narrative

Imagine Priya, a recent graduate relocating to Mumbai. She's tired of shady brokers and spammy Facebook groups. She downloads this app, enters her budget, and gets instant access to verified 1BHKs near Bandra. The clean map view helps her spot listings near her office. She saves a few favorites and gets notified when a new listing matching her search appears.

Meanwhile, Raj, a landlord, uploads his 2BHK flat in Andheri. Within hours, he sees views ticking up, chats with 3 interested renters, and decides to upgrade to a premium listing to get even more traction. The app makes it simple for both sides — no agents, no spam, just rentals.

📊 Success Metrics

Daily Active Users (DAU) & WAU

Conversion Rate: Listing View → Inquiry

Time to First Contact

Owner Retention Rate (weekly return to app)

Ad Revenue (AdMob)

Premium Listing Upgrades (% of total listings)

Referral Conversions (% of invited users)

⚙️ Technical Considerations

Mobile App

React Native + Expo

NativeWind or Gluestack for styling

Expo Router for navigation

Supabase Auth + Postgres for backend

Supabase Storage for images

AdMob SDK (Expo integration)

Push Notifications via Expo

Web App

Next.js (App Router)

Tailwind + Shadcn UI

Supabase SDK for client-side operations

Shared

Maps: Google Maps or Mapbox

Payments: Stripe (Global) or Razorpay (India)

Chat: Supabase Realtime or external service (e.g. Stream, Pusher)

Analytics: PostHog (self-hosted) or Firebase

🧱 Milestones & Sequencing

⚒️ MVP – "Find & Message"

Timeline: XX weeks

✅ Supabase Auth + Email OTP

✅ List & filter properties

✅ Add new listing (owners)

✅ Property detail view

✅ In-app chat with owner

✅ Map view

✅ AdMob integration

✅ Notifications for new listings

🔮 Future Scope

Feature

Description

AI Search Bot

Natural language property search: "Find me 1BHK under ₹20k near Koramangala"

Smart Alerts

Push alerts for saved searches

Verified Owners

KYC or business document upload for landlord badge

Virtual Tour

Upload videos or 360° images

Auto Rental Agreement

Document generator + e-sign integration

Subscription Tier

Early access to hot listings, see who viewed your listing

Nearby Facilities Map

Show commute distance to offices, schools, gyms

Ratings/Reviews

From previous tenants (moderated)

Referral Program

Invite users → earn credits or premium access

💰 Monetization Strategy

Method

Notes

AdMob

Show ads to renters (list screen, property screen)

Premium Listings

Paid boost via Stripe/Razorpay

Subscriptions

Early access, verified search, AI search bot

Affiliate

Movers, rental agreements, insurance, furniture

Referral

Earn credits or Amazon vouchers for invites

🧩 Open Questions

Should listings be auto-expired after a time window?

Do we offer call/contact details or in-app chat only?

How much friction to add for verified listings (KYC)?

Should renters rate owners?

Next Steps

Finalize MVP features + split into sprints

Lock tech stack & choose AdMob vs Interstitial Ads

Build user + owner flows in parallel

Launch in 1 city or neighborhood

Collect feedback → prioritize AI, subscriptions, ratings

This is an awesome foundation. If you'd like, I can also:

Build sprint breakdowns or JIRA-ready stories

Draft onboarding copy, chat UX, or upsell screens

Help you pitch this to investors or stakeholders

Awesome — you're going full throttle 💪

Let's break this down clearly and cleanly:

We'll cover:

🚀 Roadmap (Phased Plan)

🧭 UI Flow (with clear user journeys)

🧱 MVP Build (Feature-by-feature breakdown)

💸 Monetization Strategy (Prioritized & Sequenced)

🚀 1. Roadmap: Phase-by-Phase Plan

Phase

Focus

Key Deliverables

Phase 1: MVP (Week 0-8)

Core discovery and messaging

Auth, Search + Filter, List + Map view, Property detail, In-app chat, Owner listing post, AdMob

Phase 2: Engagement (Week 9–14)

User retention + verified listings

Push notifications, Save favorites, Smart Alerts, Verified badges, Listing views

Phase 3: Monetization (Week 15–18)

Revenue engine

Premium Listings (pay to boost), Stripe/Razorpay integration, Affiliate links (rental docs, movers)

Phase 4: Expansion (Week 19–24)

Delight & Growth

AI chatbot search, Auto lease generator, Virtual Tours, Referral program, Subscription plans

Phase 5: Scale (Week 25+)

City expansion, analytics, ops

Admin dashboard, Owner verification (KYC), Reporting tools, Data dashboards, Growth campaigns

💮 2. UI Flow: User-Centric Journey Maps

🧑 Renter App Flow

Splash → Onboarding

Quick value prop, option to skip login

Login / Signup

Email OTP (Supabase), minimal friction

Home

Featured Listings

Filter bar (Rent, BHK, Area)

Toggle: List / Map view

Map View

Pins with price

Tap to preview → tap to open detail

Property Detail

Photos (carousel)

Amenities, location

Owner info + "Verified Badge"

Message owner CTA

Save button

Favorites Tab

Saved listings

Push alerts toggle per saved search

Profile

Edit info

Logout

Subscription Status (once added)

🧑‍💼 Owner Dashboard Flow

Login / Signup

Same Supabase flow

Post Property

Title, Address (Google/Mapbox autocomplete)

Price, Type (1BHK etc)

Upload Images (Supabase Storage)

Submit

My Listings

Edit / Delete buttons

View metrics: views, messages

Promote Listing (Premium button → Razorpay/Stripe flow)

Messages

Chat UI with interested renters

Inbox view → thread view

🧱 3. MVP Build Plan

🛠️ Target: 6-8 weeks for core functionality with 1 engineer + 1 designer

✅ Week 1–2: Setup

Expo project setup (mobile) or Next.js (web)

Supabase setup: DB schema for users, listings, messages

Email OTP auth (Supabase)

✅ Week 3–4: Listings

Add Listing (form with image upload)

Property browse screen (list + filters)

Map view with markers

Property detail screen

✅ Week 5: Chat & Saved Properties

In-app chat (Supabase Realtime or 3rd party)

Saved/favorite listing support

Profile screen

✅ Week 6: Ads + Notifications

Integrate AdMob banners/interstitials

Push Notifications (Expo for mobile)

Setup PostHog or Firebase analytics

✅ Week 7–8: Polish + Launch

UI/UX polish

Error handling, loading states

MVP testing and internal dogfooding

Optional Buffer:

Web app parity (Next.js + Tailwind)

Web listing-only portal

💸 4. Monetization Strategy

💰 Immediate (MVP-Ready)

Method

Implementation

Notes

AdMob

Banner or interstitial ads in list/detail screens

Easy to integrate with Expo

Pay-to-Post

Razorpay/Stripe checkout for premium listing

Simple flag in DB to "highlight"

💎 Phase 2: Add-On Monetization

Method

Implementation

Notes

Affiliate Links

Embed links in property detail or success screen (e.g. movers, insurance, furniture)

Use Bitly or tracking to measure

Referral Program

Invite code tracking via Supabase

Offer credit or premium upgrade

Subscription

Add plan with early access to listings, AI chatbot, auto rental docs

Stripe integration + Supabase roles

🧠 Longer-Term Revenue Streams

Method

Description

Smart AI bot as an upsell

Example: "Find me a 2BHK under ₹20k near metro"

Verified badge upsell

Pay to get verified (or via KYC upload)

Featured banner on homepage

Rotating property banner for premium owners

Monetization Stack Recommendations

Tool

Use

AdMob (Expo)

For all free-tier renters

Razorpay or Stripe

For one-time payments (listing boost)

Stripe Subscriptions

For recurring subscription plans (Premium User)

Affiliate Partners

Use Indian brands like NoBroker Docs, UrbanClap, Amazon, or global partners

🧠 Tips for Prioritizing Monetization:

Start with Premium Listing Boost — fastest to implement and test.

Use AdMob as baseline monetization — unobtrusive but reliable.

Add Subscription Plans only when you see repeat renters returning.

Keep affiliate links contextual — don't spam.