

Property-1 Technical Documentation

1. Architecture Overview

Property-1 is a full-stack Single Page Application (SPA) designed for the luxury real estate market. It utilizes a modern serverless architecture, delegating database management, file storage, and backend logic to **Convex**, while authentication is handled by **Clerk** and payments by **Stripe**.

- **Frontend:** React application built with Vite. It handles the UI, routing, and client-side logic.
- **Backend (BaaS):** Convex. It acts as the real-time database, API server, and file storage provider. It replaces a traditional Node.js/Express server.
- **Authentication:** Clerk. Handles user sessions, sign-ups, and identity management, syncing user data to Convex via webhooks/client-side listeners.
- **Payments:** Stripe. Handles subscription billing and one-time payments via Checkout Sessions.

2. Tech Stack

Frontend Core

- **Framework:** React 19
- **Build Tool:** Vite
- **Language:** TypeScript
- **Routing:** React Router DOM (v6)
- **Styling:** Tailwind CSS (v3.4)
- **Animations:** Framer Motion
- **Icons:** Lucide React

Backend & Infrastructure

- **Database:** Convex (Reactive, document-oriented)
- **API:** Convex Query/Mutation Functions
- **Storage:** Convex File Storage (Images/Assets)

Integrations

- **Auth:** @clerk/clerk-react
- **Payments:** Stripe API (Node.js SDK within Convex)

3. Project Structure

```
FLIPPA-Property-1/
├── components/           # React UI Components
│   ├── Admin/           # Admin Panel logic
│   ├── Dashboard/       # Buyer & Agency specific dashboards
│   ├── Home/            # Landing page sections
│   ├── Marketplace/     # Property listing & detail views
│   ├── Layout/          # Global Header/Footer
│   ├── Pages/           # Static & Auth pages (About, Pricing, etc.)
│   └── UI/              # Reusable UI elements (Reveal, etc.)
├── convex/              # Backend Logic
│   ├── _generated/      # Auto-generated Convex types
│   ├── schema.ts        # Database Schema definition
│   ├── users.ts         # User management functions
│   ├── listings.ts      # Property CRUD logic
│   ├── stripe.ts        # Payment processing actions
│   └── ...              # Other business logic handlers
├── data/               # Static mock data interfaces
├── public/             # Static assets & .htaccess
├── App.tsx             # Main Application Routing & Context wrappers
└── index.tsx           # Entry point & Provider configuration
```

4. Database Schema

The database is defined in ``convex/schema.ts``.

Table	Description	Key Fields
users	Application users	<code>`tokenIdIdentifier`</code> (Clerk ID), <code>`role`</code> (buyer/agency/admin), <code>`agencyId`</code> , <code>`hasPaidMembership`</code>
agencies	Real estate agencies	<code>`name`</code> , <code>`logo`</code> , <code>`ownerId`</code> , <code>`sales`</code> , <code>`location`</code>
properties	Real estate listings	<code>`title`</code> , <code>`price`</code> , <code>`agencyId`</code> , <code>`image`</code> (Storage ID), <code>`status`</code> , <code>`amenities`</code>
messages	Inquiries/Chats	<code>`agencyId`</code> , <code>`senderId`</code> , <code>`propertyTitle`</code> , <code>`message`</code> , <code>`read`</code>
plans	Pricing configuration	<code>`name`</code> , <code>`price`</code> , <code>`stripePriceId`</code> , <code>`features`</code>
wishlist	User favorites	<code>`userId`</code> , <code>`propertyId`</code>

5. Setup & Installation

Prerequisites

- Node.js (v18+)
- NPM or Yarn
- A Convex account
- A Clerk account
- A Stripe account

Installation Steps

1. Install Dependencies:

```
npm install
```

- ### 2. Environment Variables:
- Create a `.env.local` file in the root directory. You will need keys from your dashboard:

```
# Deployment  
VITE_CONVEX_URL=https://your-convex-instance.convex.cloud
```