LotAnalyzer - Address-Based Lookup Implementation Spec

Version 0.9 (2025-06-10)

1. Project Goal

Convert LotAnalyzer from a bulk-CSV workflow to an address (or Redfin/Zillow URL) search experience that instantly tells us whether a given Austin property can be split and, if so, estimates the maximum buildable square footage for the new lot(s).

2. High-Level Architecture

React Front (Next.js) <-> GET /api/lookup?addr=? <-> Edge Function (Node 18 with KV cache)

Edge Function fans out to:

? Google Maps - geocode

? Travis CAD - parcel data

? Austin GIS - zoning & overlays

? Listing API (e.g. Redfin via ZenRows)

3. External Services and Keys

- Google Places API: Autocomplete & geocode -> lat/lng. 40 000 requests/month free. ENV GOOGLE_MAPS_KEY.
- Travis CAD JSON: Parcel data (lot area, legal lot, year built). Unofficial, no key.
- Austin AGOL GIS: Zoning layer and overlays. 50 000 requests/day free. No key.
- Redfin Scraper API (ZenRows): Current listing metadata. 100 requests/day free. ENV ZENROWS_KEY.
- KV / Redis: 24-hour caching of API responses (1 GB free). ENV KV_URL.

TODO - replace ZenRows with licensed MLS feed.

4. Backend (Edge Function) - Step by Step

1) Input validation and normalization - accept addr or url, trim whitespace, lowercase host.
2) If URL, extract address via regex for redfin.com or zillow.com.
3) Geocode - call Google Maps. Reject if precision < ROOFTOP.
4) Parcel lookup - determine PROP_ID through Austin GIS PropertyProfile, then call Travis CAD endpoint to get lot_sqft, lot_width_ft, year_built.
5) Zoning lookup - query Austin AGOL zoning layer to obtain ZONING_CLASS (e.g. SF-3-NP).
6) Split-eligibility logic - compare lot size and width to twice the minimums from zoning rules.
7) Envelope and FAR - compute max_buildable_sqft = FAR * lot_sqft for each half-lot.
8) Listing metadata (optional) - fetch price, days-on-market via ZenRows if URL/zpid present.
9) Cache JSON result with key addr:{sha256(address)} for 24 hours.
10) Respond with JSON example shown in spec.

5. Front-End Tasks

- Search bar with Google Places Autocomplete -> state.
- Fetch hook for /api/lookup - manage loading, error, cache.
- Mapbox GL overlay - draw parcel polygon.
- Result panel - show verdict chips and collapsible FAR / setback table.
- "Save candidate" button - write to Supabase favorites.

6. Local Development Setup

```
pnpm i
cp .env.sample .env.local      (fill keys)
npx supabase start             (local Postgres)
pnpm dev                       (Next.js + edge functions)
```

7. Acceptance Criteria (MVP)

- User enters any Austin address and gets a split verdict in under 4 s.
- Zoning class matches Austin GIS for the same point.
- FAR and setback values align with official zoning table.
- Second call for same address is served from cache in < 300 ms.
- Parcel outline and envelope numbers render correctly in UI.
- CI runs eslint and vitest on every PR.

8. Future Enhancements

- Batch mode - upload CSV of addresses and stream results.
- Comps module - pull $/sq ft comps to price new builds.
- Subdivision fee calculator - estimate platting and utility costs.
- National support - swap GIS layer and zoning rules per city.

9. Reference Links

- Austin GIS zoning FeatureServer: https://services7.arcgis.com/.../FeatureServer/0
- Travis CAD property API (unofficial): https://propertyapi.traviscad.org/property/123456
- Austin zoning code PDF Table 3-1 -> internal zoning_rules.json build script.
- ZenRows Redfin API docs: https://www.zenrows.com/documentation/redfin-api

Prepared by ChatGPT - 2025-06-10