A DIY BUILD PLAYBOOK · V1.0 · FREE

The Local Comp Kit

Build your own neighborhood-comparable-sales tool with public data, an AI assistant, and a weekend. No MLS license, no $500/mo data subscription, no permission needed.

> *Every property sale in the United States is recorded as public information. The hard part isn't getting the data — it's stitching it together so it answers one question fast: what did houses like this one nearby actually sell for?*

BY	Alex Jahn
EDITION	v1.0 — April 2026
AUDIENCE	Investors, flippers, contractors
BUILD TIME	~1 weekend with an AI assistant
COST	$0 if you have an LLM subscription

What this is, and who it's for

The problem. You drive up to a house. Maybe you're meeting a seller, walking a flip candidate, or eyeballing what your buddy's place might be worth. You want a number — not a Zestimate (those are off by 7–10% on off-market homes), not the asking price, but a real *this is what comparable houses just sold for* answer. Today that answer is locked behind one of three doors: an MLS license you don't have, a $500/month data API, or a friend who's a realtor and owes you a favor.

The opportunity. The real sale prices are already public. Every U.S. state requires sales to be recorded; most counties publish parcel data online; the federal Census provides geocoding. The pieces exist. They just don't fit together unless someone glues them.

This kit is the glue. It walks you through building a small tool that:

- Looks up any address in your county and returns its assessed value, square footage, year built, and last sale price
- Finds nearby properties that sold recently with similar specs
- Spits out a one-page comp report you can read before you walk the house
- Costs $0 to run after the build, forever

Who this is for. Investors, flippers, contractors, and small-time landlords who want better numbers than Zillow and aren't going to pay $500/mo for ATTOM. You don't need to know how to code. You need to know how to drive an AI assistant (Claude, ChatGPT, Cursor — pick one) and copy-paste the prompts in this kit.

Who this is not for. If you're trying to build the next Zillow, this isn't that. If you need lender-grade appraisals, this isn't that either. This is a flipper's tool. Same accuracy as the appraiser's first pass, in 5 seconds, on your phone, on the curb.

> **HI, I'M ALEX.** Working carpenter and house-flipper in Fond du Lac, Wisconsin. I built this tool for my own deals first — flips, walk-throughs, "is this worth the drive" calls — before turning the playbook into something I could give away. I run Agent Logic, a small consultancy that helps trade businesses run their operations through AI agents. If this kit is useful, the rest of what I do probably is too. (More at the end.)

> **ONE SENTENCE:**
> By the end of this kit, you'll have an address-queryable comp tool running on your laptop, fed by your county's public data, that answers "what did houses like this one sell for nearby?" in under 5 seconds.

How comps actually work (5-minute primer)

If you skip this and start building, you'll build the wrong thing. Comp accuracy is 80% in the comp-selection rules and 20% in the math.

What's a "good" comp

- **Recent.** Last 6 months ideal. 12 months acceptable in slow markets. Older sales need a price-appreciation adjustment, which is squishy.
- **Nearby.** 1 mile is the standard suburban default. Rural areas need 2–5 miles. Cross-school-district = different sub-market = bad comp.
- **Similar.** Sqft within 20%. Same bed/bath count when possible. Similar lot, similar age, similar condition. Two-story comp for a two-story subject; ranch for a ranch.

The "Three D's" when comps are scarce

Appraiser shorthand for what to do when you can't find clean matches: **Dated** (older sales, same neighborhood) · **Distant** (similar properties, farther out) · **Dissimilar** (less-similar properties, same neighborhood). Each forces an explicit adjustment. Use the one that requires the smallest defensible adjustment.

The adjustments that actually matter

Difference	Typical impact
Square footage	Most important. Compute $/sqft from comps; apply to subject's sqft.
Bed / bath count	$5–15K each in most markets. Highly local — check with two paired sales in your area.
Lot size	Varies wildly. Urban: small. Rural: huge. Use neighborhood comps only.
Garage stalls	$5–15K per stall in most U.S. markets.
Condition	The squishiest — this is where you walk the house. Algorithm can't see roof age or smell mold.
Time / appreciation	Adjust older comps up if the market rose. Use county-wide median trend.

> **Practical truth:** for a flipper-grade tool, *flagging* the differences between subject and comps is more useful than computing exact dollar adjustments. You're going to walk the property anyway. Your eye is better than any algorithm at condition. The tool's job is to surface the candidates and the deltas; your job is to call the number.

What a useful comp report looks like

Three to seven nearby recent sales, ranked by similarity, with the deltas surfaced (sqft difference, bed/bath difference, time-since-sale, distance). One-line $/sqft summary at the bottom. That's it. Don't over-engineer the output — you're reading it on your phone in a driveway.

How to find your county's public data

Every U.S. county publishes some property data online. The exact format varies wildly; the *categories* of source are universal. Find one of each:

The three categories every U.S. county has (somewhere)

Source	What it gives you	Search keywords
1. Assessor portal (county-level)	Parcel ID, address, assessed value, sqft (sometimes), beds/baths (sometimes), owner, year built, lot size	"*[county] county assessor*" "*[county] real estate tax search*" "*[county] property records*"
2. Register of Deeds / Recorder (county-level)	Recorded deeds with sale prices, mortgages, liens, ownership history	"*[county] register of deeds*" "*[county] recorder*" "*[county] land records*"
3. State sales database (state-level, sometimes called "transfer return" or "real estate transfer tax")	Bulk sales data, often searchable, often free, often more complete than the county-by-county view	"*[state] real estate transfer return*" "*[state] property sales data*" "*[state] department of revenue property*"

The discovery script

Spend 30 minutes here. Save URLs as you go.

1. **Find your county assessor portal.** Search "*[your county] county assessor property search*" and find the official .gov site. Try a real address you know — can you search by address? What fields come back?
2. **Find your register of deeds.** Most counties have an online deed search. Many use third-party platforms with names like *LandShark*, *Beacon*, *qPublic*, *Ascent*. Note the platform — that tells you the data shape.
3. **Find your state-level sales database.** Search "*[your state] real estate transfer return*" or "*[your state] property sales data*". Some states (Wisconsin, North Carolina, Colorado) make this easy. Others (Texas, Louisiana) make it harder.
4. **Check for bulk downloads.** Search "*[your state] statewide parcel data download*" or "*[state cartographer's office]*". Many states (Wisconsin, Massachusetts, Minnesota, Vermont) offer free statewide bulk parcel files that include last sale price — this is a goldmine if it exists.
5. **Identify the platform stack.** If the assessor portal is built on *ArcGIS*, there's almost certainly a queryable REST endpoint underneath. If it's *Ascent*, it has a JSON API you can probe with browser DevTools. If it's a custom homegrown site, you may need to scrape with a real browser (Playwright). Note this — it determines your build path.

> **STUCK ON THIS STEP?** County data discovery is the single most painful part of building this tool. Some counties have firewalled portals (Akamai, Cloudflare); some have JS-only interfaces with hidden APIs; some are stuck in 2003 and don't have searchable databases at all. I do this discovery for clients in under a day — reverse-engineering the portal, identifying the cleanest data path, and writing the adapter so they don't have to. If your county is fighting you, see the back page.

Worked example: Fond du Lac County, Wisconsin

Concrete is more useful than abstract. Here's how the discovery script played out for one Midwestern county. Yours will look different, but the shape is the same.

What I found in 30 minutes

Source	Platform	What I got	Reachable?
FDL County Treasurer real-estate search	Custom (firewalled by Akamai)	Parcel + assessed value + ownership; address-searchable	403 to scrapers
FDL "Ascent" Property Listing at landinfo.fdlco.wi.gov	Ascent (Transcendent Tech)	Same data plus deed history, parcel boundaries	Yes — AngularJS SPA over JSON API
FDL GIS portal at gisweb.fdlco.wi.gov	ArcGIS Online (React + Esri)	Geometry, parcel ID lookup, basic attributes	Yes — ArcGIS REST under the hood
WI DOR RETR sales database at propertyinfo.revenue.wi.gov	State portal (cookie/JS-gated)	5 years of sales statewide with prices — the goldmine	Yes via Playwright
WI Statewide Parcel Map V11 at sco.wisc.edu/parcels	Bulk shapefile / geodatabase download	3.5M parcels statewide with last-sale-price + sqft + year-built + geometry	Yes — free download

The path I picked, and why

For v1: **WI Statewide Parcel V11 bulk download.** The data is one year old, but it includes *last sale price + last sale date per parcel* already — which covers most flipper use cases. One file, downloaded once, parsed once, never scraped. Zero anti-bot fights, zero vendor dependency.

For "fresher than 12 months": Phase 2 falls back to either a Playwright-driven RETR sync (weekly cron) or, if I get desperate, paying for ATTOM's API ($95–500/mo). I'm betting I won't need to.

> **Lesson for your county:** always check for *statewide bulk parcel data* first. Many states publish it free as part of a state cartographer's office or department of natural resources. If yours does and it includes sale price + date, you can skip 80% of the engineering pain. Try the search "*[your state] statewide parcel data*" before anything else.

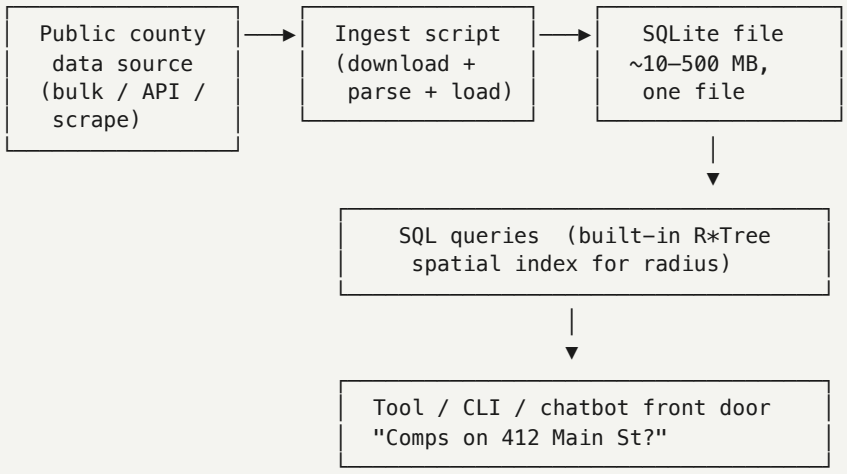
The fallback ladder

1. **Free statewide bulk file** if available. (Best.)
2. **Free county API** via ArcGIS REST or Ascent reverse-engineering.
3. **Free public portal** via Playwright-driven scrape (slower, fragile, but works).
4. **Paid API (ATTOM, etc.)** as a last resort — $95–500/mo. Only worth it if free sources fail systematically.

Most U.S. counties land on tier 1 or 2. Almost none require tier 4 unless you're trying to build something commercial.

The architecture you'll build

Five moving pieces. None of them require a server, a cloud account, or a paid service.



Why SQLite

- One file. No server. No login. Backs up by copying.
- Built into Python's standard library — nothing to install.
- R*Tree spatial index ships with it — "find parcels within 1 mile" runs in milliseconds, no PostGIS needed.
- Handles your county trivially. Handles your *state* trivially. SQLite is fine up to multi-GB, which is well past your needs.

Three tables

- **properties** — one row per parcel: address, lat/lon, sqft, year built, last sale, assessed value, owner, source.
- **sales** — one row per sale (multiple per parcel possible): date, price, transfer type, source.
- **address_aliases** — "412 Main St" / "412 N Main" / "412 Main Street" all map to the same parcel. Built during ingestion.

Plus an R*Tree virtual table for fast geographic radius queries, and a small migration table so future schema changes are painless.

What it doesn't do

- No web UI. The tool returns text. Your AI assistant or your chat app is the interface.
- No real-time feed. Data refreshes weekly (or whenever you re-run the ingest).
- No automated valuation model (AVM). You see comps; you call the number.
- No multi-state by default. Build for your county; expand only when you actually need it.

> **ARCHITECTURAL RULE:** Build the smallest thing that answers "show me comps near here." Resist adding features until you've used the smallest version on five real walk-throughs.

The seven build phases

One phase per session if you're working in evenings. Whole thing in a weekend if you're focused. Each phase ends with something you can verify before moving on.

#	Phase	What you'll have at the end	Effort
0	**Scaffold**	Folder structure, README, schema file, empty Python files. Your AI assistant set up to drive the build.	~1 hr
1	**Ingest**	Your county's data loaded into SQLite. Verify with a SQL query: `SELECT COUNT(*) FROM properties`.	2–6 hrs
2	**Geocode**	Every parcel has a lat/lon. Fuzzy address matching works. R*Tree populated.	1–2 hrs
3	**Tools**	`lookup_property(addr)` and `find_comps(addr, radius, months)` functions you can call from a Python REPL or hand to an AI assistant.	2–3 hrs
4	**Report**	Markdown comp reports auto-generated. Saved to a folder. Ready to read on phone.	1–2 hrs
5	**Front door**	Wired into your AI/chat app of choice (Claude, ChatGPT, Telegram bot, etc.). Address in — report out.	1–2 hrs
6	**Refresh**	Scheduled weekly re-ingest. Data stays current automatically.	1–2 hrs
7	**Sanity**	5–10 known properties verified. You know the tool's accuracy bounds. You decide when to trust it for real money.	1–2 hrs

How to use this kit

Each of the next pages contains a **ready-to-paste prompt** for your AI assistant. The flow:

1. Open Claude / ChatGPT / Cursor / your AI of choice.
2. Copy the phase's prompt block. Replace anything in `[brackets]` with your specifics.
3. Paste. Let the AI do the work. Read what it produces; ask follow-ups.
4. Run the verification step at the end of the phase. If it passes, move on. If not, paste the failure into the AI and ask it to fix.
5. Don't move to the next phase until the current one verifies. Each phase depends on the one before.

> **Skill check:** If you've never asked an AI assistant to write Python code before, do a 10-minute warm-up. Ask it to "write a Python script that prints 'hello' from a SQLite database." Run it. If it works, you're ready. If you can't get it running, the pre-build skill gap is bigger than this kit fixes.

Build prompts — Phases 0 & 1

Pick your AI: Claude (claude.ai or Claude Code), ChatGPT (chatgpt.com), Cursor (cursor.sh), Cody, Copilot Chat, anything with code-writing tool use. Paste these into the chat.

PHASE 0 — SCAFFOLD (PASTE THIS FIRST)

```
You are helping me build a property comp lookup tool. I am a real estate investor in [YOUR COUNTY, STATE]. I am not a programmer but I can read code and run commands. We will build this in phases over a weekend. Today is Phase 0: scaffolding.

Create this folder structure on my machine at [PATH WHERE YOU WANT THE PROJECT, e.g. ~/projects/property-comps]:

property-comps/
├── README.md         (populated)
├── data/             (gitignored — for the SQLite file)
├── schema/
│   └── 001_initial.sql   (populated — see below)
├── fixtures/         (empty, for test data later)
├── docs/             (empty, we add notes as we learn)
└── scripts/
    ├── ingest.py     (placeholder — one comment line saying "Phase 1")
    ├── geocode.py    (placeholder)
    ├── comp_finder.py    (placeholder)
    └── report_builder.py (placeholder)

Plus a top-level .gitignore that excludes data/*.sqlite and data/*.db.

The README explains: what this tool will do (look up address → return comp report), what database it uses (SQLite + R*Tree spatial index), and that I will build it phase by phase.

The schema/001_initial.sql defines three tables: properties (parcel_id PRIMARY KEY, county, full_address, lat, lon, building_sqft, year_built, property_class, assessed_value, last_sale_price, last_sale_date TEXT, owner_name, source TEXT, ingested_at TEXT), sales (id INTEGER PRIMARY KEY, parcel_id, sale_date, sale_price, transfer_type, source, ingested_at), address_aliases (alias TEXT PRIMARY KEY, parcel_id, confidence REAL DEFAULT 1.0). Plus an R*Tree virtual table properties_geo with parcel_id_int + min_lat/max_lat/min_lon/max_lon. Plus a parcel_id_map (parcel_id_int INTEGER PRIMARY KEY AUTOINCREMENT, parcel_id TEXT UNIQUE). Plus a schema_migrations table for version tracking.

After you're done, list every file you created with line counts and confirm the structure matches.
```

VERIFICATION

Run `find property-comps/ -type f`. You should see exactly the files described above. Read the README out loud — it should explain what we're building in plain English.

PHASE 1 — INGEST (PASTE THIS ONCE PHASE 0 VERIFIES)

```
Continuing the property comp build. Phase 1: ingest [YOUR COUNTY] parcel data into SQLite.

My data source is: [URL OR DESCRIPTION OF YOUR DATA SOURCE — e.g., "Wisconsin Statewide Parcel V11
shapefile from sco.wisc.edu/parcels/data-county/, downloaded as Fond_du_Lac.zip (.shp format)"].

The fields in the source data are: [PASTE FIELD LIST IF YOU HAVE IT, OR SAY "discover them by inspecting
the file"].

Build scripts/ingest.py. It should:
1. Apply the schema from schema/001_initial.sql to data/property_cache.sqlite (creating it if needed).
Track applied migrations in schema_migrations.
2. Read the source data (use pyogrio for shapefiles, requests for HTTP, json/csv for those formats). Use
only standard library + pyogrio + requests.
3. For each parcel: insert into properties table; if there's a last sale price, also insert into sales
table; populate address_aliases with normalized address forms (e.g., "412 N Main St" plus "412 Main",
"412 Main Street", "412 N Main").
4. Populate the R*Tree spatial index with each parcel's bounding box (or a tiny box around its lat/lon).
5. Print a summary: how many parcels ingested, how many had valid lat/lon, how many had a last sale
price, how long it took, any parse errors.

Be defensive about source data — fields may be missing or malformed. Skip rows you can't parse and log
the count, don't crash.

Run the script. Show me the summary. Then run these verification SQL queries and show output:
- SELECT COUNT(*) FROM properties;
- SELECT COUNT(*) FROM sales;
- SELECT COUNT(*) FROM properties WHERE lat IS NOT NULL;
- SELECT * FROM properties LIMIT 3;
- SELECT MAX(last_sale_date), MIN(last_sale_date) FROM properties WHERE last_sale_date IS NOT NULL;
```

VERIFICATION

Properties count should be in the realistic range for your county (5K–500K). The lat-not-null count should be at least 80% of total. Last sale dates should range across multiple recent years. If any of these look wrong, paste the failure into the AI and ask it to fix the parser.

Build prompts — Phases 2 & 3

PHASE 2 — GEOCODE & ADDRESS MATCHING

```
Continuing the property comp build. Phase 2: address normalization and geocoding fallback.

Most of my parcels already have lat/lon from the bulk source. But when I look up an address that doesn't match exactly, I need fuzzy matching. And for parcels that came in without coordinates, I need a geocoder.

Build scripts/geocode.py. It should expose two functions:

1. resolve_address(query: str) -> dict
   Given an address string like "412 N Main St" or "412 main", normalize it (uppercase, strip punctuation, expand "Main St" / "Main Street" / "Main"), then look up against the address_aliases table.
   - If exact match: return parcel_id and confidence 1.0.
   - If no match: try fuzzy match (Levenshtein distance ≤ 3, or shared tokens) against the full_address column.
   - Return: {parcel_id, full_address, lat, lon, confidence}. None if nothing within reason.

2. fill_missing_coordinates() -> int
   For every property in the DB with NULL lat/lon, call Nominatim (https://nominatim.openstreetmap.org/search) with a 1-second rate limit and a descriptive User-Agent ("comp-kit/1.0 contact: [your email]"). Cache results in a local SQLite cache (data/geocode_cache.sqlite) keyed by the normalized query string. Update the property's lat/lon. Return how many were filled.

Use only standard library + requests. Respect Nominatim's terms — 1 req/sec max, descriptive user agent.

After it's built: run resolve_address("[a real address you know in your county]") and show me the result. Then run fill_missing_coordinates() and tell me how many it filled.
```

PHASE 3 — LOOKUP & COMP FINDER

```
Continuing the property comp build. Phase 3: the two main tools.

Build scripts/comp_finder.py. Two functions:

1. lookup_property(address: str) -> dict
   Calls resolve_address from geocode.py. If found, returns a dict with the parcel's full record from properties table plus any sales from sales table. If not found, returns None.

2. find_comps(address: str, radius_miles: float = 1.0, months: int = 12, sqft_tolerance: float = 0.20) -> list[dict]
   Steps:
   a. Look up the subject's lat/lon and sqft.
   b. Compute a bounding box around subject_lat/subject_lon for radius_miles. (1 degree latitude ≈ 69 miles; 1 degree longitude ≈ 69 * cos(lat) miles.)
   c. Query properties_geo (R*Tree) for parcels within the bounding box.
   d. Filter those down: same property_class, sqft within ±sqft_tolerance of subject, last_sale_date within last `months` months, last_sale_price > 0, parcel_id ≠ subject's parcel_id.
   e. Compute exact distance with haversine for each remaining candidate. Filter to radius_miles.
   f. Compute $/sqft for each comp. Compute deltas (sqft delta, distance, days since sale).
   g. Sort by similarity score (closer + more similar = higher). Return top 10.

Each comp dict should include: parcel_id, full_address, last_sale_price, last_sale_date, building_sqft, sqft_delta, distance_miles, price_per_sqft, days_since_sale.

Show me: lookup_property("[your test address]") and find_comps("[your test address]") output side by side. Sanity check the comps — do they look like reasonable houses near your subject?
```

VERIFICATION (BOTH PHASES)

Phase 2: pick three address variants of one real property ("412 N Main St", "412 Main", "412 main street") — all three should resolve to the same parcel. Phase 3: `find_comps` on a known address should return at least 2–5 results in suburban areas. If you get zero, your radius/months/tolerance are too tight, or the source data has gaps. Loosen and re-run.

> **HITTING FRICTION HERE?** Phase 2 (address fuzzy matching) is where most builds get stuck — addresses are dirty and Nominatim has rate limits. Phase 3 (radius math) is where bugs hide; off-by-degree errors are common. If you'd rather skip this part, Agent Logic ships pre-built `geocode.py` and `comp_finder.py` as part of the DFY install, configured for your specific county. Details on the back page.

Build prompts — Phases 4 through 7

PHASE 4 — REPORT BUILDER

```
Continuing the property comp build. Phase 4: human-readable comp reports.

Build scripts/report_builder.py. One function:

build_comp_report(address: str, radius_miles: float = 1.0, months: int = 12) -> str

It calls lookup_property and find_comps from comp_finder.py, then formats a markdown report with:

# Comps for [address]

## Subject
- Address, parcel ID, assessed value, sqft, year built, last sale (if any)

## Recent Comps (within X miles, last Y months)
A markdown table with: address, sale price, sale date, sqft, $/sqft, distance, sqft delta.

## Summary
- Comp $/sqft: median and range
- Implied subject value (median $/sqft × subject sqft)
- Number of comps used

## Caveats
- Static reminder: "These are raw comps. Walk the property. Adjust for condition. This is a starting
point, not an appraisal."

Save the report to a folder of your choice (default: ~/comp-reports/). Filename:
Comps_[normalized_address]_[YYYY-MM-DD].md.

Build the function, save the file, then print the markdown to terminal so I can read it.
```

PHASE 5 — FRONT DOOR

```
Continuing the property comp build. Phase 5: how I actually use this from my phone or chat app.

Pick ONE based on what I use:
- Telegram bot (most common for solo investors)
- Discord bot
- Slack bot
- Just a CLI I run on my laptop
- A function I expose to Claude/ChatGPT via tool use

I use [PICK ONE].

Wire it up so that when I send a message like "comps on 412 Main", the bot/CLI calls build_comp_report
and replies with either the full markdown or a one-line summary plus a link/path to the saved report
file.

For Telegram: use the python-telegram-bot library, add a single command handler for /comp.
For CLI: a thin click-based wrapper.
For Claude/ChatGPT tool use: register comp_finder + report_builder as tools.

Show me the wiring and a real end-to-end test: send the message, get the report back, paste both.
```

PHASE 6 — WEEKLY REFRESH

```
Continuing the property comp build. Phase 6: keep data fresh.

Set up a weekly refresh:
1. Modify scripts/ingest.py to be idempotent — re-running it should update existing parcels (UPSERT) and add new ones, not crash on duplicates.
2. Add a cron job (or launchd plist on Mac) that runs ingest.py weekly at 3 AM Sunday.
3. Log the run to data/refresh.log with timestamp + row counts.

Show me the cron line / plist file you created and confirm the next run time.
```

PHASE 7 — SANITY CHECK

```
Final phase: verify the tool is trustworthy enough to act on.

Pick 5 properties in [YOUR COUNTY] where you know what they actually sold for (recent sales of friends' houses, your own deals, public records). Add them to fixtures/known_properties.csv with columns: address, actual_sale_price, actual_sale_date.

Write scripts/sanity_check.py that loops over the fixture file, runs find_comps on each, computes the median comp $/sqft, multiplies by the subject's sqft to get an implied price, compares to the actual sale price, and reports the % error per property.

Run it. Tell me the median error and worst error. If median error is under 10%, the tool is decision-grade for first-pass screening. Over 15% means there's a problem in the comp-selection logic — debug it.
```

VERIFICATION

After Phase 7, you have a tool you can trust within ±10% on first pass. Walk the property to close the rest of the gap. **Do not skip Phase 7.** A tool that's wrong by 30% is more dangerous than no tool at all — it gives you false confidence.

Known limits & sanity rules

Every tool lies a little. Knowing where this one lies keeps you out of trouble.

Where the tool gets it wrong

- **Condition is invisible to data.** Two houses with identical sqft / beds / baths / age can be $50K apart in real value. The algorithm can't see the kitchen, the roof, or the smell. Walk the property.
- **Rural parcels are sparse.** Few comps, often non-arms-length transfers (family, $1, estate). Filter on transfer_type if your data has it. Default to wider radius (5+ miles) and longer time window (24 months).
- **Recent sales lag the data.** Bulk parcel files are 6–12 months behind. RETR / state databases are usually 30–60 days behind. Last week's sale won't appear yet. Check the local MLS for active/pending listings if you need real-time.
- **Non-arms-length transfers** (family, divorce, foreclosure) skew the data low. Many bulk sources flag these; some don't. Filter aggressively when you have the flag, eyeball when you don't.
- **"Recent renovations" don't show up.** If a comp was renovated last year and the subject wasn't, the comp's price is misleadingly high. No data source captures this reliably.

Sanity rules you should run before you trust the tool

1. **Test it on properties you already know the answer for.** Five of them. If the tool's median number is off by more than 10% from the truth, debug before relying on it.
2. **Compare to the public Zestimate as a sanity check, not as the truth.** Same ballpark = healthy. Tool is wildly off vs. Zestimate = something's broken in your filters.
3. **Watch for radius degeneracy.** If your tool returns 0 comps too often, you're filtering too tight. If it returns 30+, too loose.
4. **Re-run sanity check after every schema change.** Things break silently when you add a feature. Have a "did the median error get worse" check.
5. **Don't act on a single comp.** One sale is an anecdote. Three is a pattern. Five is a number.

How to evolve this tool over time

- **Months 1–2:** use it on every walk-through. Note where it's right, where it's wrong. Log misses in a markdown file.
- **Month 3:** add the adjustments that actually matter for your market — the squishiest one for flippers is usually condition, but you can't algorithm that. Lot size and garage count are the next two.
- **Month 6:** if you've expanded into another county, build a small adapter pattern so each county has its own loader but shares the same schema and tools.
- **Year 1:** consider whether to publish your tool as a service. If your county was hard to find data for, others will have the same problem.

> **The biggest sanity rule:** a tool that's wrong by 30% is worse than no tool, because it gives you false confidence. *Always* run Phase 7's sanity check before you make a real-money decision based on the tool. Always.

What's next

You finished. You have a working comp tool. Two things to think about now.

Use it before you improve it

The next 5–10 walk-throughs are the most important part of this whole project. Use the tool. Notice where it nails the number. Notice where it misses. Note the misses in a file. **Resist the urge to add features.** Most "improvements" are premature. The tool's job is to surface candidates and deltas; the rest is your judgment in the driveway.

If you want help going further

I built this kit because the discovery work for your county is genuinely painful, and because most flippers shouldn't have to learn ArcGIS REST APIs to comp a house. If you got stuck or want to go deeper, here are the three ways I work with people:

Tier	What it is	When it's right for you
DIY	This kit. Free. Forever.	You have a weekend and an AI subscription. You're comfortable copy-pasting prompts and running scripts. Your county isn't unusually hostile.
DWY (Do With You)	Workshop / cohort. Small group, monthly office hours. *Coming late 2026.*	You want company through the build. You're stuck on county data and want a second pair of eyes. You want to share notes with other investors building the same thing.
DFY (Done For You)	Custom install. I build the tool for your specific county, wire it into your chat app of choice, hand it back working. Setup + retainer.	You'd rather not. Your time is worth more than a weekend. Your county's data is firewalled or fragmented and you don't want to fight it. You want it running by Friday.

> **ABOUT ME.** I'm Alex Jahn. I'm a working carpenter and house-flipper in Fond du Lac, Wisconsin. I've been building agent-based tooling for my own trade business since the beginning of 2026 — from day one of autonomous AI agents being practical inside real business workflows — and the comp tool is one piece of a larger stack that runs proposals, invoicing, scheduling, and field notes. I run Agent Logic, where I help other tradespeople and small business operators build the same kind of stack for their own work. The unfair part of the pitch: no AI consultancy actually runs a real trade business. No carpenter actually runs a multi-agent stack. I'm the only person dumb enough to do both, which means I've already solved the problems you're about to hit.

How to reach me

If this kit was useful, even just to read, I'd love to hear it. If you got stuck, I'd love to hear that too — the failures are how I improve the kit. Three ways:

Email	alexanderjahn79@icloud.com
Text	920-539-8814 — my actual cell, real human, no bot
Schedule a call	920-679-6207 — this number rings my AI assistant. Tell it you want time with me; it'll check my calendar and book the slot. Yes, you're talking to the same kind of system this kit teaches you to build. That's the demo.

Build it. Use it on your next deal. Tell me how it went.
Good luck out there.

Appendix A: A real test list

Below are 100 real, recently-recorded property sales from Fond du Lac County, Wisconsin — pulled directly from the Wisconsin Department of Revenue's RETR (Real Estate Transfer Return) database on April 29, 2026 to prove out the data acquisition path described in this kit. The list was the first page of a 1,000-record result set across 4 years; pages 2–10 hold another 900 records on the same query. Buyer and seller names are redacted because this is a public-facing kit; everything else — document numbers, parcel IDs, addresses, municipalities — is exactly as the state portal returned it.

Notice what's here and what isn't. The list view exposes the structural data (parcel ID, address, who-to-whom). Sale prices and recording dates live one click deeper, in each record's detail view — that's the next loop your build needs to write. But the test list below already proves the entry path: **real public data, queryable by county, returns 100s of records per page, free, no MLS license required.** If you're building a comp tool for a different county, this is the shape you're trying to reproduce.

COUNTY DOC #	MUNICIPALITY	PARCEL NUMBER	PROPERTY ADDRESS	GRANTOR	GRANTEE
1194472	Fond Du Lac, City of	FDL-15-17-13-44-821-00	524 Sweetflag Avenue	— redacted —	— redacted —
1221894	Fond Du Lac, City of	FDL-15-17-12-34-808-00	17 CONCORD DIRVE	— redacted —	— redacted —
1219571	Campbellsport, Village of	V02-13-18-99-TI-331-00	253 BAUMANN ST	— redacted —	— redacted —
1221943	Fond Du Lac, City of	FDL-15-17-16-11-143-00	464 RUGGLES STREET	— redacted —	— redacted —
1221285	Fond Du Lac, City of	FDL-15-17-09-43-593-00	641 FOREST AVE	— redacted —	— redacted —
1218414	Auburn, Town of	T03-13-19-20-12-007-00	W1911 HAVEN DRIVE	— redacted —	— redacted —
1220727	Lamartine, Town of	T12-15-16-21-02-008-00	VACANT LAND - CTH T	— redacted —	— redacted —
1174192	Fond Du Lac, City of	FDL-15-17-15-41-193-00	104 E 11th Street	— redacted —	— redacted —
1205417	Taycheedah, Town of	T20-16-18-14-13-001-00	W2951 Silica Rd	— redacted —	— redacted —
1219833	Fond Du Lac, City of	FDL-15-17-04-44-826-00	510 VERMONT STREET	— redacted —	— redacted —
1222119	Fond Du Lac, Town of	T09-15-17-99-DU-200-00	W5014 MARIEARL LANE	— redacted —	— redacted —
1219083	Empire, Town of	T08-15-18-99-LW-275-00	LOT 7 MARYS AVENUE	— redacted —	— redacted —
1203785	Fond Du Lac, City of	FDL-15-17-23-14-753-00	760 County Road V	— redacted —	— redacted —
1221822	Empire, Town of	T08-15-18-31-12-003-00	W4766 MILL ROAD	— redacted —	— redacted —
1220290	Taycheedah, Town of	T20-16-18-99-SK-020-00	W4550 LAKEPARK DRIVE, UNIT 2	— redacted —	— redacted —
1220906	Fond Du Lac, City of	FDL-15-17-15-14-877-00	33 7TH STREET	— redacted —	— redacted —
1222406	Fond Du Lac, City of	FDL-15-17-14-11-021-00	96 S ROYAL AVE	— redacted —	— redacted —
1222295	Fond Du Lac, Town of	T09-15-17-99-LB-395-00	N7140 WINNEBAGO DRIVE	— redacted —	— redacted —
1190272	Friendship, Town of	T11-16-17-99-BW-110-00	Land along USH 45	— redacted —	— redacted —
1222158	Fond Du Lac, City of	FDL-15-17-11-44-867-00	20 N. RESERVE AVENUE	— redacted —	— redacted —
1219674	Eldorado, Town of	T07-16-16-18-09-001-00	W9577 OLDEN ROAD	— redacted —	— redacted —
1217038	Metomen, Town of	T14-15-14-15-05-002-00	Carter Road	— redacted —	— redacted —
1177958	Fond Du Lac, City of	FDL-15-17-10-32-271-00	185 N. Lincoln Ave.	— redacted —	— redacted —
1218515	Lamartine, Town of	T12-15-16-04-09-013-02	STATE ROAD 23	— redacted —	— redacted —
1218376	Springvale, Town of	T19-15-15-01-11-001-00	W10069 TRIPLE KAY RD.	— redacted —	— redacted —
1220913	Taycheedah, Town of	T20-16-18-22-10-009-00	W3632 SCHUSTER LANE	— redacted —	— redacted —
1210308	Campbellsport, Village of	V02-13-18-99-PE-025-00	121 South Poplar Street	— redacted —	— redacted —
1220914	North Fond Du Lac, Village of	V05-16-17-99-BE-230-00	110 POLK STREET	— redacted —	— redacted —
1219760	Fond Du Lac, City of	FDL-15-17-11-24-820-00	277 ROOSEVELT ST.	— redacted —	— redacted —
1221972	Eldorado, Town of	T07-16-16-04-13-003-00	JAHN ROAD	— redacted —	— redacted —
1220095	Fond Du Lac, City of	FDL-15-17-10-32-273-00	197 N. LINCOLN AVENUE	— redacted —	— redacted —
1183738	Campbellsport, Village of	V02-13-18-99-VB-090-00	657 Valley View Drive	— redacted —	— redacted —
1221836	Forest, Town of	T10-15-19-15-05-001-00	STATE HWY 23	— redacted —	— redacted —
1221949	Byron, Town of	T04-14-17-14-01-004-00	W5374 COUNTY ROAD B	— redacted —	— redacted —
1222228	Marshfield, Town of	T13-16-19-15-07-001-00	CTH WH	— redacted —	— redacted —
1218591	Fond Du Lac, City of	FDL-15-17-14-41-028-00	619 E. 9TH STREET	— redacted —	— redacted —
1219976	Alto, Town of	T01-14-14-26-10-001-00	TOWN OF ALTO VACANT LAND	— redacted —	— redacted —
1220385	Ashford, Town of	T02-13-18-32-16-009-00	W4121 ST KILLIAN DR	— redacted —	— redacted —
1211483	North Fond Du Lac, Village of	V05-16-17-99-FO-724-00	432 Prospect Ave North Fond du Lac WI 54937	— redacted —	— redacted —
1173455	Calumet, Town of	T05-17-18-99-ab-290-00	N10814 ARTESIA BEACH RD	— redacted —	— redacted —
1215300	Empire, Town of	T08-15-18-26-06-003-00	Town of Empire	— redacted —	— redacted —

COUNTY DOC #	MUNICIPALITY	PARCEL NUMBER	PROPERTY ADDRESS	GRANTOR	GRANTEE
1182324	Calumet, Town of	T05-17-19-32-07-009-00	N9913 & N9919 Townhall Road	— redacted —	— redacted —
1220572	Fond Du Lac, City of	FDL-15-17-24-12-256-00	674 MARY LEE DR	— redacted —	— redacted —
1221940	Fond Du Lac, City of	FDL-15-17-11-33-577-00	171 SHEBOYGAN ST	— redacted —	— redacted —
1221304	Rosendale, Town of	T18-16-15-26-08-003-00	N7904 STATE ROAD 26	— redacted —	— redacted —
1178837	Fond Du Lac, City of	FDL-15-17-10-11-009-00	133 Bischoff Street	— redacted —	— redacted —
1179736	Friendship, Town of	T11-16-17-10-10-003-00	N9087 Lakeshore Drive Van Dyne	— redacted —	— redacted —
1194172	Metomen, Town of	T14-15-14-36-12-005-00	N4944 State Road 49	— redacted —	— redacted —
1220448	Ashford, Town of	T02-13-18-13-11-001-00	TOWN OF ASHFORD	— redacted —	— redacted —
1193683	Fond Du Lac, City of	FDL-15-17-10-41-045-00	132 E. Johnson Street	— redacted —	— redacted —
1173366	Taycheedah, Town of	T20-16-18-32-06-012-00	N7634 COUNTY RD WH	— redacted —	— redacted —
1221191	Fond Du Lac, City of	FDL-15-17-11-32-288-00	180 N. PARK AVENUE	— redacted —	— redacted —
1221627	Calumet, Town of	T05-16-18-02-14-001-00	JOHNSBURG RD	— redacted —	— redacted —
1220203	Forest, Town of	T10-15-19-17-03-002-00	N6339 LOEHR ROAD	— redacted —	— redacted —
1219094	Eden, Town of	T06-14-18-12-15-003-00	W2606 COUNTY ROAD B, EDEN, WI 53019	— redacted —	— redacted —
1219630	Waupun, City of	WPN-14-15-99-EA-146-00	26 N. MILL STREET	— redacted —	— redacted —
1221358	Fond Du Lac, City of	FDL-17-17-15-31-091-00	425 MORRIS STREET	— redacted —	— redacted —
1210644	Ripon, City of	RIP-16-14-28-07-330-02	479 W. Griswald Street	— redacted —	— redacted —
1219812	Marshfield, Town of	T13-16-19-35-13-001-00	W495 COUNTY ROAD CCC	— redacted —	— redacted —
1218415	Ashford, Town of	T02-13-18-20-01-014-00	N1206 KATZENBURG DR	— redacted —	— redacted —
1181285	Marshfield, Town of	T13-16-19-36-13-004-00	W166 County Road C	— redacted —	— redacted —
1197384	Fond Du Lac, City of	FDL-15-17-22-33-504-00	UNP Land Sec 22 RR-R/W	— redacted —	— redacted —
1194661	Auburn, Town of	T03-13-19-99-FI-100-00	N744 U.S. HIGHWAY 45	— redacted —	— redacted —
1218929	Oakfield, Village of	V06-14-16-14-13-014-00	233 SOUTH MAIN STREET	— redacted —	— redacted —
1221492	Fond Du Lac, Town of	T09-15-17-99-ST-220-00	N6020 BACKWATER DR	— redacted —	— redacted —
1205902	Fond Du Lac, City of	FDL-15-17-23-14-828-00	838 County Road V	— redacted —	— redacted —
1222288	Marshfield, Town of	T13-16-19-11-11-001-00	CTH G	— redacted —	— redacted —
1194300	Fond Du Lac, City of	FDL-15-17-15-11-051-00	11 E. 2nd Street	— redacted —	— redacted —
1193841	Fond Du Lac, City of	FDL-15-17-09-41-046-00	440 Portage Street	— redacted —	— redacted —
1220854	Taycheedah, Town of	T20-16-18-35-13-008-00	N7470 KONEN ROAD	— redacted —	— redacted —
1192443	Fond Du Lac, City of	FDL-15-17-14-21-077-00	297 East 2nd St	— redacted —	— redacted —
1220863	Friendship, Town of	T11-16-17-17-07-001-00	VAN DYNE ROAD	— redacted —	— redacted —
1180910	Fond Du Lac, City of	FDL-15-17-14-23-544-00	257 5th. Street	— redacted —	— redacted —
1220219	Calumet, Town of	T05-17-18-99-VB-130-11	N10560 VELVET BEACH ROAD	— redacted —	— redacted —
1222109	Fond Du Lac, City of	FDL-15-17-27-11-017-00	180 KNIGHTS WAY	— redacted —	— redacted —
1218794	Fond Du Lac, City of	FDL-15-17-15-34-779-01	251 GRIFFITH ST	— redacted —	— redacted —
1219517	Fond Du Lac, City of	FDL-15-17-15-11-033-00	111 EAST 1ST STREET	— redacted —	— redacted —
1218883	Oakfield, Village of	V06-14-16-99-OA-270-00	364 OAKVIEW CIRCLE	— redacted —	— redacted —
1218594	Forest, Town of	T10-15-19-29-16-002-00	W1725 MUSHROOM RD	— redacted —	— redacted —
1221195	Friendship, Town of	T11-16-17-31-15-018-00	VACANT LAND, FOND DU LAC, WI 54935	— redacted —	— redacted —
1218776	North Fond Du Lac, Village of	V05-15-17-05-02-014-00	545 N PIONEER RD	— redacted —	— redacted —
1220642	Ripon, City of	RIP-16-14-99-GU-280-00	122 STONEY RIDGE ROAD	— redacted —	— redacted —
1219012	Marshfield, Town of	T13-16-19-11-11-001-00	COUNTY ROAD G	— redacted —	— redacted —
1218600	Fond Du Lac, City of	FDL-15-18-18-24-814-00	EASTWIND LANE	— redacted —	— redacted —
1221774	Ripon, City of	RIP-16-14-21-03-460-12	315 E JACKSON ST	— redacted —	— redacted —
1219226	Lamartine, Town of	T12-15-16-14-04-004-00	N6374 NELSON ROAD	— redacted —	— redacted —
1185001	Eden, Village of	V03-14-18-99-AS-302-00	104 Southgate Dr.	— redacted —	— redacted —
1198222	Fond Du Lac, City of	FDL-15-17-11-34-797-00	323 E. Division St.	— redacted —	— redacted —
1219066	Auburn, Town of	T03-13-19-32-04-002-00	N394 US HIGHWAY 45	— redacted —	— redacted —
1197761	Fond Du Lac, City of	FDL-15-17-11-14-836-00	288 Lawrence Ave, Fond du Lac WI 54935	— redacted —	— redacted —
1203124	North Fond Du Lac, Village of	V05-16-17-99-KS-340-00	1513 Fairlawn Avenue	— redacted —	— redacted —
1220211	Fond Du Lac, City of	FDL-15-17-23-21-097-00	288 19TH ST	— redacted —	— redacted —
1221092	Fond Du Lac, City of	FDL-15-17-12-32-269-00	16 BRYN MAWR CIR	— redacted —	— redacted —
1221978	Ripon, Town of	T17-16-14-27-05-009-00	W13140 STATE ROAD 23	— redacted —	— redacted —
1221153	Friendship, Town of	T11-16-17-06-02-003-00	NONE ON TAX RECORDS	— redacted —	— redacted —

COUNTY DOC #	MUNICIPALITY	PARCEL NUMBER	PROPERTY ADDRESS	GRANTOR	GRANTEE
1220601	Ashford, Town of	T02-13-18-13-08-008-00	N1572 COUNTY ROAD W	— redacted —	— redacted —
1220597	Oakfield, Village of	V06-14-16-99-OR-830-00	395 NORTH MAIN STREET	— redacted —	— redacted —
1207612	Rosendale, Village of	V07-16-15-35-14-002-00	102 E. Rose-Eld Rd.	— redacted —	— redacted —
1222113	Fond Du Lac, City of	FDL-15-17-10-24-832-00	150 WEST FOLLETT STREET	— redacted —	— redacted —
1222343	Fond Du Lac, City of	FDL-15-17-14-33-584-00	235 14TH ST	— redacted —	— redacted —

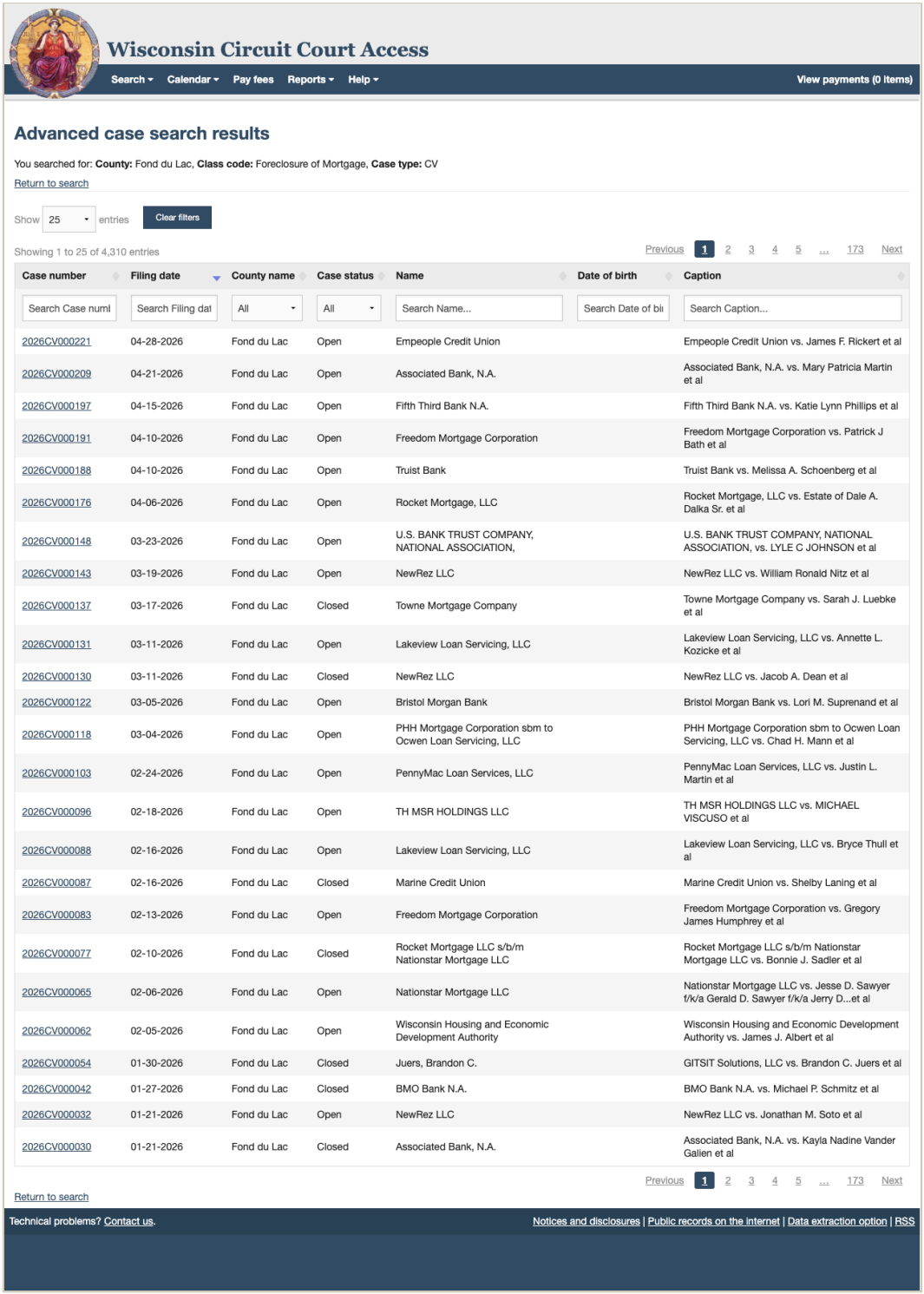
Source: Wisconsin DOR RETR portal (tap.revenue.wi.gov/RETRHistoric) · **Captured:** 2026-04-29 21:34 CDT · **Method:** Playwright-driven session, cookie-gated · **Scope:** Fond du Lac County, recorded 29-Apr-2022 through 29-Apr-2026 · **Total result set:** 1,000 (max cap; more exist beyond) · **This page:** records 1–100 · **Data status:** public records under Wisconsin law · **Redactions:** grantor & grantee names blanked for distribution. The full dataset including names is freely searchable on the source portal.

Appendix B: A variant build — foreclosure tracking

The same 7-phase methodology in this kit also points cleanly at a different problem. The main path is for *closed sales* — what already changed hands, what a fair offer looks like. But the same county-discovery → build-loop steps adapt to *foreclosure cases* — the lender-vs-borrower lawsuits filed weeks to months *before* the property hits sheriff's sale. That's the early-warning signal flippers actually want.

For Wisconsin, the public surface is **Wisconsin Circuit Court Access (CCAP)** at `wcca.wicourts.gov`. Click-through agreement only, no login. The Advanced Case Search lets you filter by county, case type (Civil), and class code — **30404 (Foreclosure of Mortgage)** for the main residential / commercial path; **30403 (Agricultural Foreclosure)** for farm / ag; **31002 (Eviction Due to Foreclosure)** as a secondary signal for already-foreclosed properties heading to vacancy.

The screenshot below was captured live on April 30, 2026. It shows the result of a single CCAP query — *County: Fond du Lac, Case type: Civil, Class code: Foreclosure of Mortgage* — returning **4,310 cases** across CCAP's full record period, with the most recent filing two days prior (Empeople Credit Union vs. Rickert, filed 04-28-2026). The list view exposes case number, filing date, status (Open/Closed), plaintiff (lender or servicer), and full caption. That's the structural shape of a working foreclosure-lead feed.



Wisconsin Circuit Court Access

Search | Calendar | Pay fees | Reports | Help | View payments (0 items)

Advanced case search results

You searched for: **County:** Fond du Lac, **Class code:** Foreclosure of Mortgage, **Case type:** CV

Return to search

Show 25 entries | Clear filters

Showing 1 to 25 of 4,310 entries

Previous 1 2 3 4 5 ... 173 Next

Case number	Filing date	County name	Case status	Name	Date of birth	Caption
Search Case numl	Search Filing dat	All	All	Search Name...	Search Date of bi	Search Caption...
2026CV000221	04-28-2026	Fond du Lac	Open	Empeople Credit Union		Empeople Credit Union vs. James F. Rickert et al
2026CV000209	04-21-2026	Fond du Lac	Open	Associated Bank, N.A.		Associated Bank, N.A. vs. Mary Patricia Martin et al
2026CV000197	04-15-2026	Fond du Lac	Open	Fifth Third Bank N.A.		Fifth Third Bank N.A. vs. Katie Lynn Phillips et al
2026CV000191	04-10-2026	Fond du Lac	Open	Freedom Mortgage Corporation		Freedom Mortgage Corporation vs. Patrick J Bath et al
2026CV000188	04-10-2026	Fond du Lac	Open	Truist Bank		Truist Bank vs. Melissa A. Schoenberg et al
2026CV000176	04-06-2026	Fond du Lac	Open	Rocket Mortgage, LLC		Rocket Mortgage, LLC vs. Estate of Dale A. Dalka Sr. et al
2026CV000148	03-23-2026	Fond du Lac	Open	U.S. BANK TRUST COMPANY, NATIONAL ASSOCIATION,		U.S. BANK TRUST COMPANY, NATIONAL ASSOCIATION, vs. LYLE C JOHNSON et al
2026CV000143	03-19-2026	Fond du Lac	Open	NewRez LLC		NewRez LLC vs. William Ronald Nitz et al
2026CV000137	03-17-2026	Fond du Lac	Closed	Towne Mortgage Company		Towne Mortgage Company vs. Sarah J. Luebke et al
2026CV000131	03-11-2026	Fond du Lac	Open	Lakeview Loan Servicing, LLC		Lakeview Loan Servicing, LLC vs. Annette L. Kozicke et al
2026CV000130	03-11-2026	Fond du Lac	Closed	NewRez LLC		NewRez LLC vs. Jacob A. Dean et al
2026CV000122	03-05-2026	Fond du Lac	Open	Bristol Morgan Bank		Bristol Morgan Bank vs. Lori M. Suprenand et al
2026CV000118	03-04-2026	Fond du Lac	Open	PHH Mortgage Corporation sbm to Ocwen Loan Servicing, LLC		PHH Mortgage Corporation sbm to Ocwen Loan Servicing, LLC vs. Chad H. Mann et al
2026CV000103	02-24-2026	Fond du Lac	Open	PennyMac Loan Services, LLC		PennyMac Loan Services, LLC vs. Justin L. Martin et al
2026CV000096	02-18-2026	Fond du Lac	Open	TH MSR HOLDINGS LLC		TH MSR HOLDINGS LLC vs. MICHAEL VISCUSO et al
2026CV000088	02-16-2026	Fond du Lac	Open	Lakeview Loan Servicing, LLC		Lakeview Loan Servicing, LLC vs. Bryce Thull et al
2026CV000087	02-16-2026	Fond du Lac	Closed	Marine Credit Union		Marine Credit Union vs. Shelby Laning et al
2026CV000083	02-13-2026	Fond du Lac	Open	Freedom Mortgage Corporation		Freedom Mortgage Corporation vs. Gregory James Humphrey et al
2026CV000077	02-10-2026	Fond du Lac	Closed	Rocket Mortgage LLC s/b/m Nationstar Mortgage LLC		Rocket Mortgage LLC s/b/m Nationstar Mortgage LLC vs. Bonnie J. Sadler et al
2026CV000065	02-06-2026	Fond du Lac	Open	Nationstar Mortgage LLC		Nationstar Mortgage LLC vs. Jesse D. Sawyer f/k/a Gerald D. Sawyer f/k/a Jerry D...et al
2026CV000062	02-05-2026	Fond du Lac	Open	Wisconsin Housing and Economic Development Authority		Wisconsin Housing and Economic Development Authority vs. James J. Albert et al
2026CV000054	01-30-2026	Fond du Lac	Closed	Juers, Brandon C.		GITSIT Solutions, LLC vs. Brandon C. Juers et al
2026CV000042	01-27-2026	Fond du Lac	Closed	BMO Bank N.A.		BMO Bank N.A. vs. Michael P. Schmitz et al
2026CV000032	01-21-2026	Fond du Lac	Open	NewRez LLC		NewRez LLC vs. Jonathan M. Soto et al
2026CV000030	01-21-2026	Fond du Lac	Closed	Associated Bank, N.A.		Associated Bank, N.A. vs. Kayla Nadine Vander Galien et al

Previous 1 2 3 4 5 ... 173 Next

Return to search

Technical problems? Contact us. | Notices and disclosures | Public records on the internet | Data extraction option | RSS

CCAP advanced-search results — Fond du Lac County, class code 30404, all dates. Captured 2026-04-30. 25 cases per page, 173 pages, 4,310 total.

Why this is a separate tool, not a comp tool

The output here isn't a comp number — it's a **lead feed**. Different question, different output, different action. Comps tell you what to *offer*; foreclosure leads tell you what to *chase*. If you build this version, what you ship to your phone every morning isn't an estimated value — it's a list of properties whose owner is months away from losing the house, with names and addresses you can mail directly. Same underlying methodology, different wedge into the market.

What changes vs the main kit's 7 phases

Most of the framework carries over without edits. The places it shifts:

- **Phase 1 (county data discovery):** CCAP statewide instead of county-by-county portals. Same gating-question pattern, simpler answer because Wisconsin centralized it.
- **Phase 2 (address resolution):** CCAP returns case captions, not addresses directly. You'll need a second hop — click into each case's detail page or cross-reference the parcel ID via the county GIS portal — to land on a mailable street address. That's where 70% of the build effort lives.
- **Phase 3 (radius math):** mostly skip. Foreclosure leads aren't comp-driven; you want county-wide coverage, not "within 0.5 miles."
- **Phases 4–7 (delivery / sanity):** identical pattern. Daily digest to your phone, sanity rules ("don't mail without a current owner-of-record check"), iteration after real use.

> **Anti-bot reality check:** CCAP runs hCaptcha in *invisible* mode. It didn't challenge the test capture above, but it absolutely can if it detects scraping patterns — rapid sequential requests, missing cookies, headless-browser fingerprints. A production scraper needs realistic per-request timing (2–5s between calls), persistent cookies, and a daily rate cap. The state also publishes an official **paid REST API** (linked from the WCCA footer as "Data extraction option") for legitimate bulk extraction — that's the DFY-tier path if you want to skip the captcha question entirely. Worth pricing if you're doing this seriously.

Tier ladder applies the same way

The DIY / DWY / DFY ladder from the main kit fits this variant cleanly:

- **DIY:** Use this appendix, the 7-phase prompts, and CCAP. Realistic if you're comfortable with Playwright or willing to use the REST API on subscription.
- **DWY:** Same workshop / cohort cadence, foreclosure track instead of comp track.
- **DFY:** Custom install. I build the foreclosure-lead feed for your county, wire it into your chat app, hand it back working — same setup & retainer model.

If you're already doing direct mail to distressed sellers and want this running by next month, contact info is on the previous page. Same channels, same human, same AI assistant on the booking line.

Source: Wisconsin Circuit Court Access (wcca.wicourts.gov/advanced.html) · **Captured:** 2026-04-30 07:42 CDT · **Method:** Playwright-driven session, click-through agreement only · **Query:** County=Fond du Lac, Case type=Civil (CV), Class code=Foreclosure of Mortgage (30404) · **Total result set:** 4,310 cases across CCAP's record period · **Most recent filing:** 2026-04-28 · **Anti-bot status:** hCaptcha invisible mode active, did not challenge this session · **Data status:** public records under Wisconsin open records law.