

# Flippit – Find Real Estate Property Deals

**Note – Your Creativity First....**

The details below are **ideas from our side, not strict requirements**.

Feel free to adapt, simplify, or take a completely different approach.

Use any tools, formulas, or models you prefer—we value **clear reasoning and clean engineering** over following a specific recipe. *[Hint: Feel free to consider any coding copilot tools like [Manus.ai](#), [Cursor](#) etc.]*

---

## Objective

Flippit is a website where a real-estate investor can find old houses to buy that will have high resell value after renovation.

Create a service that scores each **for-sale** home on how profitable it might be to buy, renovate, and resell (“flip”).

Use the **sold homes dataset** to learn market patterns and apply them to the **for-sale dataset**.

---

## Data You’ll Get

- **sold\_properties.csv** – (homes sold in the last 6 months in the city called Warren)
- **for\_sale\_properties.csv** – (currently listed homes in the city called Warren)
- Each row includes:
  - address, city, state, ZIP
  - interior size in square feet (**sqft**)
  - bedrooms/bathrooms
  - list price, sold price (sold file only)
  - **estimated\_value** (what is the estimated\_value of this property according to [realtor.com](#)’s prediction algorithm)
  - year built, property taxes, HOA fee (property taxes, HOA fees are fees you have to pay if you have a property in north america)
  - latitude/longitude
  - free-text **description** of the home (**text**)

- photos and other basic info
- 

## Our Ideas (Optional Starting Points)

- **Market Learning** – From the sold data, learn typical home values (e.g., median price per square foot by ZIP and size range).
  - **Property Scoring** – For each for-sale home, estimate:
    - **Resale price after renovation** – baseline from local price per sq ft. *[Hint: This is your estimation of Resale Price means how high you think you can sell after buying and doing some small fixes..]*
    - **Renovation cost** – simple cost per sq ft, optionally adjusted if the description text suggests heavy repairs or recent upgrades. **Or to simplify the problem just identify low/high/medium renovation costs.** *[Hint: Think outside of the box here. Does description tell how bad the property is or do the photos tell any story? 🤖]*
    - **Selling and carrying costs** – e.g., percentage of resale price plus monthly taxes/HOA for several months.
    - **Expected profit** – resale price minus (list price + all costs).
    - **Return on Investment (ROI)** –  $\text{profit} \div \text{total cash spent} \times 100$ .
    - **Risk score** – combine signals like house age, days on market, and unusual pricing. *[Hint: Do these signals tell any story? Maybe more days on the market means you have more power for negotiation & a better price but on the other hand maybe the condition is too bad?! 🤖]*
    - **Overall grade (A–F)** – summarize profitability and risk.
  - **NLP / ML Enhancements** – Use the `text` description however you like—keywords, embeddings, or a language model (LLM)—to refine resale price or renovation cost.
  - **Image Analysis** - (If necessary) There's all the property photos available too under the `primary_photo` & `alt_photos` column.
- 

## API Output

Provide a REST endpoint (for example, `POST /score`) that returns JSON with all calculated fields and a short explanation of the main factors.

---

## Bonus (Optional)

Create a simple web page like <https://www.realtor.com/>

- Displays properties on a map (using `latitude/longitude`).
- Shows a list of cards with address, list price, estimated resale price, ROI, and grade.
- Includes basic filters (price range, bedrooms, minimum ROI).
- Keep the styling minimal—just enough to demonstrate the data.

---

## Deliverables

- Source code for the API (and optional UI).
- README explaining:
  - how you used the sold data,
  - any formulas, ML models, or LLM methods you chose,
  - how to run the project.
- ***Bonus Point: A short video by screen sharing or recording where you present how you tackled this challenge!***