Technical Task: The "Bad Deed" Validator

Goal: We aim to assess how you navigate the intersection of "fuzzy" AI and rigorous financial logic.

The Context: At Propy, we can't just trust an LLM to be right. If an AI hallucinates a number on a deed, we could accidentally record a fraudulent transaction on the blockchain. We need "paranoid" engineering.

The Challenge: Develop a small Python script (or API) that takes a messy, OCR-scanned deed, cleans it up using an LLM, and then rigorously validates it with code before accepting it.

1. The Input (The Mess)

Here is the raw text you get from our OCR engine. It's ugly on purpose. *(Please use this **exact** string in your code)*

```
None
*** RECORDING REQ ***
Doc: DEED-TRUST-0042
County: S. Clara  |  State: CA
Date Signed: 2024-01-15
Date Recorded: 2024-01-10
Grantor:  T.E.S.L.A. Holdings LLC
Grantee:  John  &  Sarah  Connor
Amount: $1,250,000.00 (One Million Two Hundred Thousand Dollars)
APN: 992-001-XA
Status: PRELIMINARY
*** END ***
```

2. The Reference Data

You also have a local file (counties.json - you need to create it) with tax info. You'll need this to calculate closing costs later.

JSON:

```JSON
[
 { "name": "Santa Clara", "tax_rate": 0.012 },
 { "name": "San Mateo", "tax_rate": 0.011 },
 { "name": "Santa Cruz", "tax_rate": 0.010 }
]
```

3. What we need you to build:

We want to see a Python script that orchestrates this workflow:

1. Extract the Data: Use an LLM (OpenAI/Anthropic/etc) to parse that text into a clean object.
2. Enrich the Data: The text says "S. Clara", but our database knows it as "Santa Clara". Implement a smart way to link those two so you can pull the correct tax rate (0.012).
3. The "Sanity Check" (Crucial):
    - Logic Check: The document was *Recorded* (Jan 10) before it was *Signed* (Jan 15). That's impossible. Your code needs to catch this and throw a specific error.
    - money Check: The text lists $1,250,000 in digits but writes out "One Million Two Hundred Thousand" in words. There is a $50k discrepancy. Your system needs to flag this, not just silently pick one.

4. How we review this:

We aren't looking for a perfect prompt that solves everything magically. We are looking for an Engineering Hygiene.

- What did you use to catch the date error, or did you rely on AI if the dates looked okay?
- How did you handle the "S. Clara" lookup?
- Is your code structured well?

Deliverable: A link to a GitHub repo. Please include a short README explaining your approach.