**ReTax360 Backend Intern Take-Home Assessment**

**Due:** Monday, May 12th @ 4:20PM  
**Purpose:** Build a containerized microservice that, given a property’s PIN, computes an estimated tax-refund using our comparables API and a provided CSV of interest rates.

# Assignment Overview

You are going to be building a miniature containerized server from scratch using any Python framework you deem appropriate (recommended to use Flask). You will be provided with two resources for this task. The first is interest\_rates.csv, a table containing the yearly interest rates from the last four years. These will be important for calculations, which will be detailed later. The second key resource is an existing API endpoint which you can use to get information about a given property and properties that are similar to it, called comparables. Your service should expose a REST endpoint (e.g. POST/refund) that:

1. Accepts a JSON payload (PIN stripped of any non-numeric characters):  
   {“pin”: “26062070090000”}
2. Queries our “comparables” endpoint at http://34.28.139.127:8082/comp?pin={{pin}} to retrieve assessed values for the given property and its set of comparables. The returned data will contain a JSON with nested JSONs within it, each mapped to the PIN of the property it represents. The first PIN-JSON pair will always be the property information of the PIN from the url query [see example](http://34.28.139.127:8082/comp?pin=26062070090000).
3. Loads the provided interest\_rates.csv
4. Applies the following refund logic:
   1. *Compare Values*
      1. Compute the average assessed value among comparables.
      2. If your property’s assessed value <= average, eligibility = zero.
   2. *Determine Full-Year Occupancy*
      1. Extract the property’s sale date from the comparables data.
      2. Count only **full calendar years** between that sale date and today.
      3. Use **at most 4 years** if occupancy is much longer based on the sales date.
   3. *Yearly Refund Amount*
      1. For each full year, refund = (your\_property\_value - average\_comparable\_value).
      2. Use the interest rates to adjust each year’s refund to its present value [see formula](#_6hn9oamshypo).
   4. *Total*
      1. Sum all the present valued refunds. If there are no eligible years, no comparables, or property is worth less than the average, then refund is worth 0.
5. Returns JSON:  
    {  
    “pin”: “26062070090000”,  
    “yearsEligible”: 4,  
    “totalRefund”: 7043.40  
    }

# Minimum Requirements

* /refund endpoint implementing the logic above
* At least one automated test validating a case of your choice
* Dockerfile that builds & runs your service on a configurable port
* README.md with clear build/run/test instructions

# Submission Instructions

1. Create a public GitHub repo (or share a private link) with your code.
2. Include Dockerfile, README.md, tests, and interest\_rates.csv at project root.
3. Point out any environment variables or config needed.
4. Share your repo by **Monday @ 4:20PM**. We will use your latest commit on main to build the docker image for evaluation.
5. Be prepared to discuss your design/implementation choices in a final round interview.

# Present Value Calculation Using Interest Rates

where:

* the present value of the refund from (e.g. 2024, 2023, 2022, 2021)
* the calculated base amount for refund (difference between assessed value and average)
* the geometric product of the interest rates from the years between the start year and now
* = the interest rate for a given year *n*

*Note: We’ve intentionally kept the core requirements minimal (and given you 72 hours) so you have plenty of time and freedom to explore any features or improvements that spark your interest.You can go to https://www.cookcountyassessor.com/pin/{{pin}} to learn more about an individual property* [*see example*](https://www.cookcountyassessor.com/pin/26062070090000)*.*