Calculation Engine Take-home Task

Minimum — Technical Interview Stage

Introduction

Your task is to create a primitive Calculation Engine for calculating emissions based on structured Activity data.

This is indicative of the type of work you'd do at Minimum, so we'll kick off by defining some of the terms used in the exercise:

- *Activity Data* represents real-world activities undertaken by a business or individual, resulting in the emission of carbon dioxide and other greenhouse gases.

- An *Emission Factor* is a coefficient that relates activity data to the amount of carbon dioxide equivalent released into the atmosphere, where carbon dioxide equivalent (CO_2e) is a value representing the combined effect of various greenhouse gases in terms of their contribution to global warming.

Task

We'd like you to build a calculator for emissions, where an emission can be represented as an object with the following fields:

```
{
  "co2e": 1000,
  "scope": 3,
  "category": 1,
  "activity": "Air Travel",
}
```

The calculator should be able to accept emission factors and activity data and calculate emissions that can then be represented in a simple UI. You may also choose to build an API that can return the calculated emissions.

Input Data

You will find 4 CSV files attached to this test definition:

- One for a normalised set of emission factors

- Three which represent activity data for three activities:

    1. Business Travel

    2. Purchased Goods & Services

    3. Electricity

Each Activity Data CSV has a slightly different format based on the lookup for the emission factor. E.g. for Business travel you will need to look up the emission factor based on two columns and for Purchased Goods & Services and Electricity the lookup can be done based on one column

Files

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8f67516b-1121-42d3-b9e5-02c0db18b41c/Technical_interview_-_Emission_Factors_and_Activity_Data_-_Emission_Factors.csv

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ddd6931b-c68f-4f31-8020-472d8fd7b918/Technical_interview_-_Emission_Factors_and_Activity_Data_-_Air_Travel.csv

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9e1504d2-981e-4066-8a66-8957db6bff9b/Technical_interview_-_Emission_Factors_and_Activity_Data_-_Purchased_Goods__Services.csv

Outcomes

We would expect the following outcomes for this test:

1. Ability to calculate emissions using the calculator described above

2. Ability to display the calculated results

Calculate Emissions

Calculate emissions for the activity data attached, using the following calculation:

`co2e value = activity data * emission factor co2e value`

You will need to look up the appropriate emission factor based on the type of activity and its lookup identifier — e.g. if a company submits the following data for business flights, we'd look up the relevant emission factor based on the passenger class and flight range.

Date	Distance travelled	Distance units	Flight range	Passenger class
01/01/2023	3800	kilometres	Long-haul	Business class
15/01/2023	3800	kilometres	Long-haul	Business class
12/02/2023	4780	kilometres	International	Premium Economy class
23/03/2023	3800	kilometres	Short-haul	Business Class

E.g. for the first row of activity data here, we'd look up the emission factor with lookup identifier `Long-haul, Business class` to retrieve the value 0.04696 and calculate the emissions by multiplying distance travelled by this value i.e. `co2e = 0.04696 * 3800`.

Render the results

- Render the emissions in simple table format by building a basic UI, enabling the emissions to be:

  - sorted by CO2e

  - grouped by activity (i.e. aggregating CO2e emissions per activity type)

  - filtered by scope and category (two separate filters)

- Render the total emissions sum

Notes

- We'd like to test your Python experience in this exercise in doing the calculations, but feel free to use any web framework you're comfortable with for rendering any UI elements.

- A good rule of thumb is to spend a maximum of 2 hrs on the task. When you're happy with the work, send it over — preferably as a git repository with instructions on how to run it. We will then review the solution and book a 45 minute follow-up to discuss your approach.

- Please keep in mind that you may choose to build an API that can return the calculated emissions.