



As we know, the world is polluted because people like us and companies around the globe emit CO<sub>2</sub> on a daily basis (directly or indirectly), so while the population isn't asked to offset its carbon emission, the companies have to keep up with the government regulations to curb their carbon emissions, otherwise they might face hefty penalties for being inconsistent with the norms.

With the rise in pollution concerns, something called "carbon credits" was created as a way to help clean up our air. Think of carbon credits like special certificates - each one represents removing or preventing 1 ton of harmful gases (like CO<sub>2</sub>) from going into the atmosphere through good projects like planting trees, building solar panels, or protecting forests.

Here's how it works: imagine there's a company that creates 10 tons of pollution. To make up for this harm, they can buy 10 carbon credits from people who run clean-up projects. It's like saying, "I made a mess, so I'll pay someone else to clean up the mess to balance things out." This way, the total amount of pollution in the world stays the same or gets better, even though the company is still creating some pollution.

Offset was founded with the purpose of making sure every carbon credit has a clear life story: created → sold → retired. Nothing is ever deleted to enhance transparency within the market and the companies (that'll buy carbon credits)

Now that you've gained a surface-level understanding of what a carbon credit is, we'd like you to complete a short exercise to help us better understand your skills in relation to the company's requirements.

## BACKEND EXERCISE

At Offset, we must make sure every carbon credit has a definite lifespan: created → sold → retired. This is to make sure that nothing is ever deleted, to prevent double-selling of carbon credits. So, we'd like to ask you to build a tiny prototype of our credit ledger.

- Build a small REST API (use Python + FastAPI, or another backend framework you like).
- The API must support: POST /records → Create a record with fields {project\_name, registry, vintage, quantity}. Must generate a unique, deterministic ID (same input → same ID).
- POST /records/{id}/retire → Mark a record as "retired" by adding a new event (not updating the original record).
- GET /records/{id} → Return full record details + all events.
- Store data in a database (Postgres preferred, but SQLite is fine for the test).

Click here to download the mock JSON for you to complete the exercise: [Sample Data](#).

**Reflection Questions (write in README):**

- How did you design the ID so it's always the same for the same input?
- Why did you use an event log instead of updating the record directly?
- If two people tried to retire the same credit at the same time, what would break?  
How would you fix it?

**Note:** We know AI tools are tempting (and sometimes smarter than us before coffee), but for this exercise, please rely on your own brainpower. We're not testing how well ChatGPT or Copilot can code; we're testing your thought process. It's okay if your code isn't perfect (bugs are just shy features); what matters is that it's genuinely yours.