Innovatiespotter - PHP Developer

The goal of this test is to evaluate your coding skills, problem-solving abilities, and approach to writing clean, maintainable code. We do not expect perfection—rather, we want to see how you think through problems and structure your solutions.

Instructions

- Review the tasks below and complete them to the best of your ability.
- Focus on code quality, readability, and efficiency.
- You are free to use any online resources, but the work should be your own.
- Once completed, **push your solutions to a public GitHub or GitLab repository** and send us the link.

Task 1: refactoring

This class has several errors and the readability can be improved.
Tip: the code must be clean and maintainable.

```php
<?php

class CompanyClass
{
    public function normalizeCompanyData(array $data): ?array
    {
        $c = [];

        if (!$this->isCompanyDataValid($data)) {
            return;
        }

        $c['name'] = strtolower(trim($data['name']));

        (preg_match('/http?:\/\//i', $cleanWebsite))
            ? $c['website'] = parse_url($data['website'], PHP_URL_HOST)
            : $c['website'] = $data['website'];


        if ($c['website'] == null) {
            unset($c['website']);
        }

        if (isset($data['address']))
            $c['address'] = trim($data['address']);
```

```
        if (empty($c['address'])) {
            $c['address'] = null;
        }

        return $c;
    }

    private function isCompanyDataValid(array $data): bool
    {

        return isset($data[0]) && isset($data['address']);
    }
}

// Test Data
$input = [
    'name' => '  OpenAI ',
    'website' => 'https://openai.com  ',
    'address' => '   '
];

$input2 = [
    'name' => 'Innovatiespotter',
    'address' => 'Groningen'
];

$input3 = [
    'name' => '  Apple ',
    'website' => '<HIDDEN INPUT>  ',
];

$company = new CompanyClass();
$result = $company->normalizeCompanyData($input);

var_dump($result);

$result2 = $company->normalizeCompanyData($input2);

var_dump($result2);

$result3 = $company->normalizeCompanyData($input3);

var_dump($result3);
```

Task 2: PostgreSQL

The company collects data about companies from different sources (API, scraping, manual). The data may contain duplicates and inconsistencies. The main tables are:
Optional: dockerize the project pre-loading the model and your test data.

```sql
CREATE TABLE companies (
    id SERIAL PRIMARY KEY,
    name VARCHAR(255),
    website VARCHAR(255),
    address TEXT,
    source VARCHAR(50), -- Es: 'API_1', 'SCRAPER_2', 'MANUAL'
    inserted_at TIMESTAMP DEFAULT NOW()
);

CREATE TABLE normalized_companies (
    id SERIAL PRIMARY KEY,
    name VARCHAR(255) UNIQUE,
    canonical_website VARCHAR(255),
    address TEXT
);
```

1. **Identify potential duplicates**
   Identify companies with potential duplicates across different data sources.
   - Find entries with identical or very similar names (case-insensitive)
   - Show occurrence count
   - List the sources linked to the company
2. **Normalize the data**
   Write a query that inserts only unique companies into the `normalized_companies` table, choosing the most reliable name and website from the available sources (MANUAL has priority over API, which has priority over SCRAPER).
3. **Get statistics on sources**
   Write a query that returns the number of companies collected for each source, sorting from the source that provided the most companies to the one that provided the fewest.

Optional task: JavaScript

Write a Node function that takes a list of API URLs and downloads the JSON data, combining them into a single array.
If an API fails (responses with HTTP error or timeout), the function must log the error and continue with the others.

Optional task: PHP & PostgreSQL

Connect Task 1 and Task 2 by writing PHP functions that will fetch companies from the database, it will de-duplicate them based on company name, it will keep the highest-

priority version of a company, it will normalize the website and it will manipulate the database to keep only the required records. Then it should be able to export these companies in a csv. This task requires the candidate to generate a testing set.