adeo web

Description

Create a service, which returns product recommendations depending on current weather.

Technology stack

You are expected to use the following:

- PHP 7.1+ version;
- MySQL or equivalent database engine;
- Symfony/Laravel framework;

Example

You can use the following example as a starting point for your application.

GET /api/products/recommended/:city

Input	Output

Parameters:
:city - Vilnius
	{
	"city": "Vilnius",
	"current_weather": "rain",
	"recommended_products": [
	{
	"sku": "UM-1",
	"name": "Black Umbrella",
	"price": 10.11
	},
	{
	"sku": "HAT-15",
	"name": "Pink Hat",
	"price": 6.07
	}
	]
	}

adeo web

Requirements

- Use GIT properly - just like you will do in production;
- Store your product data in database;
- Service should be realized using REST API principles. Request and response should be handled in JSON format;
- Integrate third-party API to get the current weather information in any Lithuania city. We recommend using the LHMT API: https://api.meteo.lt/. (Note, that this API requires you to inform the user about the source of the data, which is LHMT).

Suggestions

- Create sample product data by using the Faker library;
- Host it somewhere (e.g. Heroku, Google) or provide us with the ability to launch your app (include the command that would start it, bonus if you include docker-compose.yml);
- Fill the README.md file as you would do for a production application. Include challenge description, used technologies, setup guide and usage examples.

How we review

The submitted code will be evaluated for the following aspects:

- **Presentation.** Is GIT used properly? Do you make separate commits for different features? Is the README file present and well-formatted? Does README include setup instructions? Is the application hosted somewhere?
- **Requirements.** Does it take the current weather from the API? How is the sample data generated? Does code follow REST principles? Is the database structured correctly?
- **Architecture.** How well the components of the application are separated?
- **Correctness.** Does service do what is asked? Are data validated? Are results limited? How errors are handled? Does it return appropriate status codes?
- **Code style**. Is the coding style consistent with the language's guidelines? Is it consistent throughout the codebase?
- **Code quality.** Is the code simple, easy to understand, and maintainable? Are there any code smells or other red flags? Does object-oriented code follow principles such as the single responsibility principle? Any automated tests (unit, integration) written?
- **Security**. Are there any obvious vulnerabilities?

adeo web

Code submission

The project should use a free **GitHub private** repository and share it with “joinAdeoWeb”.