castor

Software Engineer (PHP) Coding Challenge

Overview
At Castor it's all about "Faster, Smarter Medical Research". This is what drives us all. We support our customers to accelerate the discovery of evidence that cures diseases. BOOM! Our true purpose!

The problem
Castor EDC is our main product, a data collection and management tool for medical researchers. One of the main problems researchers face when transitioning to our platform is that they have existing datasets which need to be imported. Oftentimes these datasets come from different sources and have different formats. We would like to introduce functionality that allows researchers to easily map existing datasets into Castor EDC.

Your assignment
Create a small PHP command line application that allows a user to transform an existing dataset into a dataset that is usable within Castor EDC.

The command line tool should accept at least one argument, the first of which must always be the input.csv. The output.csv can be printed to the command line or written to a file, whichever you prefer.

For this assignment, assume that the external files arrive in CSV format, where the first row represents the column headers and the next rows the data. The resulting output file should be a CSV as well, but should be able to have differently ordered columns than the input.An example input and output file are provided as input.csv and output.csv.

As the data can be in a different format than Castor expects, and as the column names may be different, the client should be able to provide transformation options that need to be applied on the incoming data. For each incoming column there should be at two options defined:

1. The target column name
2. Any transformation that needs to be applied (i.e. format a date, recode option values, perform a calculation, strip sensitive information, etc).

You are free to choose the implementation of the above transformation specification.

castor

The goals

1. Develop a PHP application that transforms data from the input CSV into an output CSV whilst applying the transformation options. The application must be able to flexibly import source files with different column orders, column names and transformations.

2. Implement three different transformations used in the input.csv → output.csv transformation:

a. Recode values (mapping a finite list of values from one to another, for example textual to numeric values)
b. Calculate (i.e. multiply input value by 10)
c. Transform date

3. Make your code future proof, so it would be able to import data from other sources (Excel, RSS, API endpoints, etc.). Make it possible to easily add new transformations. You don't need to implement these, just explain how the application would handle them.

4. Create automated tests for your app using PHPUnit or any other test suite of your liking. You can use the given CSV's for integration testing.

Things to keep in mind

1. Add a README.md file to explain your architecture and justify your choices.

2. Focus on SOLID and DRY principles into your code. Explain how and why you applied these principles.

3. You are free to choose your preferred way of project setup, application execution, use of existing composer packages, etc, as long you provide us with proper instructions on how to install and execute it.

4. Please do not post your solution publicly online.

Duration and Evaluation of the assignment

We recommend timeboxing your work to 4 hours. It's okay not to finish; it's not a typing speed test! The assessment will be evaluated based on code structure, cleanliness, scalability, readability, documentation, OOP and knowledge/application of general programming principles. After the hand-in the assessment will be reviewed and the results will be communicated as soon as possible.