Test 1/2 - DB Script

Summary:

Create a CRUD script that will accept input data and process it into a database and send back a response.

Details:

- The script must be written in PHP.
- The possible input data is: id, name, description, age.
- You do not have to build the form HTML.
- You do not need to create the database structure either, it is purely theoretical.
- The script is to respond to the request with the status of the database operation, for example, if it was successful.
-

Bonus:

- Attention to code format, e.g. syntax, naming conventions, spacing, etc.
- Best practices for database performance and script performance.

Time Estimation:

Between 30 minutes to 1 hour.

Test 2/2 - Performance Script

Summary:

Create a simple script to fetch/merge data. This script has to focus on performance.

Request:

We have 3 endpoints for the data and we have to merge it and create 1 single view.

You need to use “x-token” as a key and “removify” as value inside your header request.

Source 1 (List of Data):

GET “https://service-dev.rmvfy.com/interview?source=db”

Source 2: (List of Companies)

GET “https://service-dev.rmvfy.com/interview?source=companies”

Source 3: (Name of the Company by Id)

GET “https://service-dev.rmvfy.com/interview?source=companies&company_id=1”

Result:

Retrieve 5 records from Source 1 and try to display this result below:

“id”, “name”, “title”, “company_name”

Example:

“1”, “John”, “Administrator”, “Google”

Details:

- The script must be written in PHP.
- The script should run via command line.
- You do not need to create the database structure either, or use a real endpoint, this is purely theoretical.
- You can write this script in pure PHP or you can use any kind of framework.

Bonus:

Attention to code format, e.g. syntax, naming conventions, spacing, etc.

Best practices for script performance.

Time Estimation:

Between 30 minutes to 1 hour.