Software Engineer Exercise (PHP & Laravel):

Description

A stakeholder has requested the development of an application to aid users in learning capital cities around the world. The application should be built using React on the frontend and PHP Laravel on the backend.

The application will randomly select a country and present the user with three options for the correct capital city. The user will be able to select one of the options, and the application will provide feedback on whether the selection is correct. Additionally, the application should include a button to repeat the process with a different country.

Acceptance Criteria

1. The application should have a user interface that allows users to interact with the quiz.
2. Upon launching the application, a random country should be selected.
3. The application should present the user with three options for the correct capital city.
4. The user should be able to select one of the options as their guess.
5. After submitting the guess, the application should provide feedback indicating whether the guess is correct.
6. The application should display the correct capital city if the user's guess is incorrect.
7. The application should include a button to repeat the quiz process with a different random country.
8. The quiz should continue until the user chooses to exit the application.
9. The application should be responsive and user-friendly on different devices and screen sizes.
10. Instructions on how to install and run the application should be provided in the project documentation.

Architectural Notes

To fetch the capital cities data, use the following GET endpoint:

Endpoint: https://countriesnow.space/api/v0.1/countries/capital

This endpoint provides the necessary information about countries and their corresponding capital cities.

Note: The provided endpoint is not protected in this scenario. However, it's important to keep in mind that API endpoints might be protected with authentication and authorization mechanisms in real-world scenarios. If necessary, create your own API endpoint within your application to call the external endpoint securely.

Ensure that you handle any potential errors or exceptions that may occur during the API call. Implement appropriate error handling and consider caching strategies if frequent requests to the external endpoint are anticipated.

Remember to document your implementation decisions, including any additional security measures, error handling strategies, and libraries used, to provide a clear understanding of your application's architecture and data flow.

Bonus Points:

- Ensure your errors are properly managed, and the users are notified, e.g. no loading spinners waiting for an API call which has already failed.
- Add basic unit/integration testing
- Using a Frontend CSS / Component framework

Please Note:

The implementation of Laravel is the primary focus for this exercise. Ensure that you follow the Laravel & PHP standards as closely as possible.