Mukuru Africa Practical Test

for PHP Developer Candidates

Overview

Candidates who apply for a PHP Developer position at Mukuru are required to complete a practical test. This practical test will allow the candidate to demonstrate that he / she fully understand the concepts needed to write coherent software.

Your primary abilities we want to assess is how versed you are in writing ***maintainable*** code. Although you'll use PHP frameworks and packages that have pre-defined paradigms that define how and where to write logic, please do endeavour to demonstrate your skills and knowledge by structuring (business logic) classes as if you were not using a framework.

The Objective

- Using MySQL create a database that will hold all data required to be stored.
- Write a web service in PHP that will do any data retrieval and processing. AJAX must be used from the browser to call the web services.
- Create a web application that will consume the web service and present the user with a workable user experience.

Tools

Use the PHP framework of your choice or a custom one that you have already written (please do not write one for this test!).

We condone the use of 3rd party PHP frameworks and libraries and while we like to see good usage of them, this test is to assess the candidate's abilities in writing bespoke software.

What will be assessed

These are the *main* things that will be assessed.

- Database structure.
- Understanding of OOP.
- Usage of application layering.
- Knowledge of design patterns and best practices that inspires maintainable code.
- Web service (API) design.
- Installation instructions for application (README). We will be deploying your code locally to see that it runs.

What is not required

- A "pretty" UI, though it must be user friendly. Thus colors, fonts, images etc are not needed, but feel free to use JS and CSS frameworks if that's what you are comfortable with.

- User registration and authentication.

Acceptance Criteria

A web application must be created that will allow a user to purchase foreign currencies.

- The page should display the available currencies for selection by the user.
- The page should have inputs where the user can get a quote by entering the amount of currency they wish to purchase **OR** the amount of USD currency they wish to pay. Implement both methods of calculation allowing the user to choose their preferred method of calculation.
- Once the user has entered either amount and selected the foreign currency the necessary calculation needs to be done that will display the amount they need to pay in USD.
- With the calculated amount displayed, the user can then select to “purchase” the foreign currency. An “order” for the currency must be saved to the database and the user must be shown a confirmation.

Details

- The currency used for payment with be US Dollars (USD).
- The currencies that can be purchased are:
    - South African Rands (ZAR)
    - British Pound (GBP)
    - Euro (EUR)
    - Kenyan Shilling (KES)
- Use the following exchange rates:
    - USD to ZAR: 13.3054
    - USD to GBP: 0.651178
    - USD to EUR: 0.884872
    - USD to KES: 103.860
- A surcharge must be added to orders and differs for the currencies:
    - ZAR: 7.5%
    - GBP: 5%
    - EUR: 5%
    - KES: 2.5%
- The following information must be saved with an order:
    - Foreign currency purchased.
    - Exchange rate for foreign currency.
    - Surcharge percentage.
    - Amount of foreign currency purchased.
    - Amount to be paid in USD.
    - Amount of surcharge.
    - Date created.
- When an order is saved the following extra actions need to be taken for the different currencies:
    - ZAR: No action.
    - GBP: Send an email with order details. This can be a basic text or html email to any configurable email address.

- EUR: Apply a 2% discount on the total order amount, this needs to be configurable for the currency and be saved separately on an order. This must not be included in the initial currency calculation.
- KES: No action.

Bonus

Use a foreign exchange API to get exchange rates. Results from the API should be “cached” by updating the foreign exchange rates in the database for each currency. The updating of exchange rates only needs to happen periodically in the real world, thus the update should be triggered via a URL or the command line.

You can use jsonrates to retrieve the rates. Register for a free API key.