

Mukuru / Remitix Practical Test

for PHP Developer Candidates

Overview

Candidates who apply for a PHP Developer position at Mukuru / Remitix 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.

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.
- 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.
- 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 enter the amount of foreign currency they wish to purchase or the amount of ZAR currency they wish to pay.
- 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 ZAR.
- The user can 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 will be South African Rands (ZAR).
- The currencies that can be purchased are:
 - US Dollars (USD)
 - British Pound (GBP)
 - Euro (EUR)
 - Kenyan Shilling (KES)
- Use the following exchange rates:
 - ZAR to USD: 0.0808279
 - ZAR to GBP: 0.0527032
 - ZAR to EUR: 0.0718710
 - ZAR to KES: 7.81498
- A surcharge must be added to orders and differs for the foreign currencies:
 - USD: 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 ZAR.
 - Amount of surcharge.
 - Date created.
- When an order is saved the following extra actions need to be taken for the different foreign currencies:
 - USD: 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](https://jsonrates.com/) to retrieve the rates. Register for a free API key.