**SDET/Automation Technical Test - Web Applications**

**Purpose**

As part of our hiring process for technical roles, we require all candidates to complete a small technical exercise. This is an integral part of our process to assess an individual’s technical ability when completing an activity in their own time.

Given the importance of this exercise to our hiring process, we highly recommend you give this adequate consideration and address this task as you would do any other professional assignment in your current/previous workplace.

**Instructions**

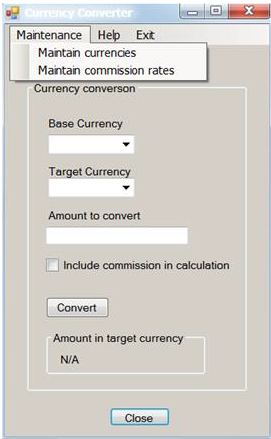
* There are two parts to this technical assessment
  + A written assessment of your ability to tackle a typical testing problem you might face as a test analyst
  + A code-based assessment, testing your ability to automate a test scenario
* Written answers for part 1 should be submitted in a suitable format - e.g. a word document
* Please work on a GitHub, create the New branch within this repository and when you are happy to submit, create a pull request back into the main branch
* Please use the pull request to comment on any aspects of your solution that you didn’t have time to complete, are not complete to your satisfaction or are not working
* Would you please add a markdown file with instructions on how to set up and run your solution to question 2

**Question 1**

**Introduction**

The following is the main form of an application that converts values from one currency to another. The application has the following characteristics:

* The collection of currencies, exchange rates, and rules for applying commission are held in a shared database, accessed by multiple users of this application on a network.
* The user of this form selects a base currency, a target currency, specifies an amount (expressed in the base currency), can choose to include or exclude commission, then hits the Convert button to generate the equivalent value in the target currency. That value is shown where “N/A” is displayed in the screenshot below.
* The separate “Maintain currencies” form (not shown) provides operations for adding, deleting and editing currencies.
* A separate process constantly updates the database with exchange rates for any recognised currencies (you do not need to worry about how this works, but you do have control over stopping and starting it).
* The separate “Maintain commission rates” form (not shown) provides operations for specifying a minimum commission value and a collection of commission rates for different ranges of values. The minimum commission value and ranges are defined in Sterling. For example, for “amount to convert” values in the range 0.01 to 100.00 pounds Sterling, the commission rate is 5%. An example minimum commission value is £10.



Given the information provided, please answer the following questions:

1. **Using the database and the GUI, how would you test the functionality of this form?**

**Answer 1.** First, we will perform Manual test cases to verify the basic functionality of the application, by entering the values in the `Base Currency`, `Target Currency` and `Amount to Convert` and then click on the Convert button and verify the amount appears in `Amount in target currency` appears correct or not.

* We can try entering the different values like integer, float, etc., and see how to applications will perform.
* We can verify the database connection by checking the `Base Currency`, `Target Currency` dropdowns if the values are showing in the dropdown correctly, then it means we are having good Database connections and we are good to go.
* And we can also write automation test cases for the same and provide various input data in the different test method and put some assertions also.

1. **What tests would you perform to test the operations supported by the “Maintain currencies” form?**

**Answer 2:-** We can test the form “Maintain currencies” by performing various actions like Adding, Updating, and Deleting the currencies.

* We can write the test case to add the currency and then verify in the database that added currency is visible or not.
* We can update the existing currency and then verify in the database that the edited currency is updated or not in the database.
* We can also write test cases for deleting the existing currency and then verify in the database like that currency is removed from the database or not.
* We can write manual as well as automation test cases for the above scenarios.

1. **What tests would you perform to test the operations supported by the “Maintain commission rates” form?**

**Answer 3:-**  We can test the “Maintain commission rates” by performing the below-mentioned steps-

* We can write test cases for verifying the commission rates for different ranges and verify that they are appearing correctly as mentioned in the database or requirements.
* We can check that, the minimum commission rate is at least 5% of the values

1. **What tests would you perform on the form above in order to test accessibility?**

**Answer 4:-**  We can perform the accessibility testing by the below-mentioned steps:-

* We can make sure that when we need to enter a base value, then it would have the ability to sound like “ Enter Base Value”
* Similarly for “Target Currency” and “Amount to Convert” and For Convert buttons.
* So basically we need to test like in this way that, it should have voice guidance like what to do and what is the results

1. **If you were automating the testing of this form for regression test purposes, what would you hope the developer had done that would make the automation easier?**

**Answer 5:-** Developer should write the conversion logic properly so that it wouldn’t fail while writing the automation Test cases.

1. **If this application were used internationally, what further tests would you perform?**

**Answer 6:-** If it is, then we should check like it working for all geo-locations and the language should be suitable for the user in different countries

1. **The screenshot of the form above shows what user interface issues are immediately apparent (you may have mentioned some of these in previous answers)?**

**Answer 7:- Interface should be updated because it is looking like so old**

**Notes**

* In answering the questions, identifying tests is sufficient
* Details of test data and expected results are not required where they can be inferred
* Aim for an answer no longer than 1200 words for this entire section

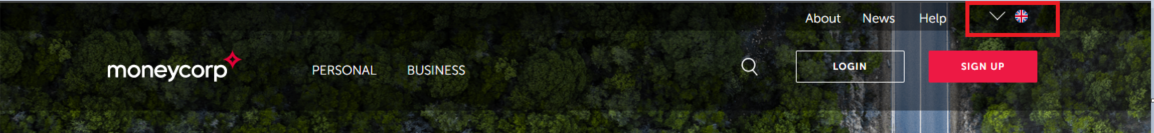
**Question 2**

**Introduction**

For this question, you can use any JavaScript framework of your choice. We want you to automate the scenarios below using the Chrome browser.

**Test Scenario**

1. Open the following URL: <https://www.moneycorp.com/en-gb/>
2. Change the language and region from the top right corner to USA (English). See example snippet.



1. Click Find out more for “Foreign exchange solutions” Validate if you have arrived on the page
2. Search for the word “international payments” using the search box
3. Validate if you have arrived on the result page
4. Validate that each article in the list displays a link that starts with <https://www.moneycorp.com/en-us/>

**Evaluation**

Evaluation will be done on the following criteria:

* Clean and tidiness of the solution
* Design and architecture
* Readability
* Maintainability
* Stability of the tests
* Clarity of setup and execution documentation

Would you please provide instructions for setting up an environment from scratch and how to run the tests?

**Answer:-**

* Create a maven project
* Structure, Organize, & Set-Up Source Control
  + Start by setting up and organizing a folder structure for your test assets.
  + Structuring your project in this manner will allow for them to be quickly referenced by members of your team and will ensure your tests are stable when updates are made.
* Familiarize with the Application
  + Performing exploratory testing on UI to understand the application more clearly
* Determine Your Testing Environments & Gather Data
* Create Utilities for On Screen Actions for example Action class
* Set Up Your Logging & Reporting Mechanism