**Тестовое задание**

*Описание:*

Вам необходимо автоматизировать три теста (описание тестов ниже). Третий тест необходимо автоматизировать в браузере. Желательно использовать для автоматизации браузер Chrome последней версии.

Подготовка окружения для выполнения тестов:

1. Создать SQLite базу данных(в виде файла), в ней создать таблицу Countries
2. Заполнить таблицу данными

|  |  |  |
| --- | --- | --- |
|  | Population | Area |
| Ukraine | 41588354 | 603628 |
| France | 67399000 | 640679 |
| USA | 328239523 | 9833520 |
| China | 1400050000 | 9596961 |

1. Сохранить изображение как example.jpeg по ссылке - <http://apimeme.com/meme?meme=Alarm-Clock&top=Top+text&bottom=Bottom+text>

**TEST-1(API)**

1. Через GET запрос получить изображение по ссылке <http://apimeme.com/meme?meme=Alarm-Clock&top=Top+text&bottom=Bottom+text>
2. Проверить, что изображение соответствует example.jpeg в стандарте base64

**TEST-2(DB)**

1. Используя данные из SQLite базы данных проверить:

* плотность населения ниже 50 чел./кв.км. только у США. Плотность = Население / Площадь.
* сумма население всех 4-х стран меньше 2 млрд человек

**TEST-3 (iFrame)**

1. Открыть в браузере https://www.w3schools.com/tags/tryit.asp?filename=tryhtml\_iframe
2. В iframe src атрибут заменить на [https://www.bing.com](https://www.bing.com/)
3. Кликнуть на Run
4. В поле поиска вписать “Redmond”,дождаться выпадающего списка с рекомендованными запросами
5. Проверить, что в выпадающем списке есть строка “redmond washington”
6. Кликнуть по “redmond washington”
7. Проверить, что в результатах поиска есть ссылка https://www.bing.com/travelguide?q=Redmond

Требования:

1. Опубликовать на GitHub.
2. Использование тестового фреймворка (любого).
3. В описании на GitHub должно быть руководство к запуску вашего теста.
4. В описании на GitHub должно быть руководство как подготовить окружение(желательно если подготовка будет автоматически выполняться коммандой/скриптом).
5. Это должен быть именно тест, а не набор скриптов. Проект должен быть рабочим и запускаться. Для проверок используйте assert из различных библиотек (в зависимости от выбранного языка): JUnit, TestNG, Chai, PyTest и т.д.
6. Опционально (по желанию): Добавьте скриншоты в ваши тесты.
7. Опционально (по желанию): Добавьте возможность параллельного запуска тестов в вашем проекте (добавьте команду или инструкцию для параллельного запуска в описание проекта)
8. Тесты должны иметь возможность запускаться как индивидуально, так и все вместе