**GLOBAL KINETIC TECHNICAL ASSESSMENT : QA AUTOMATION ENGINEER**

**INSTRUCTIONS:**

* Candidates should complete all sections if possible and must indicate the total time taken per section.

*We estimate between 8-16 hours to complete each individual section (possibly less for experienced QA Engineers)*

* **ALL** solutions should **must** be submitted via [Github](https://github.com/) , [Bitbucket](https://bitbucket.org/product) or any other source code management solution.

***Note****: Bitbucket provides a free private repository*

* You may complete the test in a language of your choice, though Java is preferred.
* Please provide documentation and instructions for all your assessment solutions. *Preferably in README.md format*
* Please tell us how to execute your solution. We expect to be able to run it and for it to generate an output of test results.
* Please ensure that any automated tests you submit can be executed easily and successfully (hint: use a build tool such a Gradle).
* For **‘Assessment 3’ (API automation),** please provide .jmx or .json files.

**Assessment will be based on:**

* Proficiency using the required technologies
* Source code management (SCM) usage
* Ease of project integration with CI e.g. Jenkins
* Project portability (i.e. avoid hard-coded directory locations)
* Documentation on the tool and how to execute tests , dependencies, preferred IDE etc.
* Improvisation
* Code quality metrics such as cleanliness, readability, maintainability and simplicity.

**ASSESSMENT:**

**1 - Web Automation (SELENIUM)**

Use any public of a website and preferably a testing framework based on Selenium/WebDriver.

*Demonstrate testing of:*

* Basic navigation
* Basic functionality
* Use of parameterization / variables

**2 - Mobile Automation (APPIUM)**

Automate testing of an Android application :

<https://github.com/googlesamples/android-UniversalMusicPlayer>

If you are unable to build the APK, Global Kinetic will provide an APK on request.

*Demonstrate testing of:*

* Basic navigation
* Basic functionality
* Use of parameterization / variables

**3 - API Automation (JMETER or POSTMAN)**

Use any public API or Docker image with a REST API service.

*Demonstrate testing of:*

* API endpoints
* Use of assertions /parameterization / correlation