



## GiG Code Challenge

Front-end Test Automation Engineer

## **Overview**

This is a code challenge for prospective Frontend Test Automation Engineers in Test to work for Gaming Innovation Group.

It is expected that all points are addressed.

This challenge covers:

- Development of a small test automation framework, preferably using a Python-based framework such as Robot Framework or Javascript based alternative.
- Taking a Behaviour Development Driven (BDD) approach

## **Deliverable**

You are expected to develop a test automation framework aimed at testing websites. The framework should include feature files and step definition classes. The framework could also include any other layer that you deem appropriate such as page object models and configuration files. Commit (regularly) to your git repository.

The project should also include documentation explaining:

- Any taken assumptions
- Any difficulties that were encountered
- A README.md file with instructions on how to install, build and run the tests.

*The final deliverable should be a link to your git repository.*

## **Expected Goals**

- We keep an eye out for the design of the test automation framework so think about the extendibility of your solution and the maintainability of your tests.
- The code is expected to compile and automated tests are expected to pass on any machine with the latest versions of node and npm installed.

Write automated tests for the following features against the website [www.automationpractice.com](http://www.automationpractice.com).

### *Task 1 - Testing the Sign-in feature.*

1. Load the homepage
2. Click on the 'Sign in' button
3. Input value in 'Email address' text box
4. Input value in 'Password' text box
5. Click on the 'Sign in' button

Create tests to cover the Sign-in functionality.

*Task 2 - Testing that the Shopping Cart tracks items correctly.*

1. Load the homepage
2. Search for '*black dress*'
3. Add to cart the *first* dress
4. Add to cart the *second* dress
5. Click on '*Proceed to checkout*'
6. Change the quantity of the second dress to '2'.
7. Remove the *first* dress from the cart
8. Click on '*Proceed to checkout*'

*Create tests to cover the above user journey.*