59 G.M. Dimtrov blvd.
NV Tower
Sofia, Bulgaria

**Role: QA Engineer Task**

Technical Interview Task

**Instructions:** This task is designed to assess your level of technical abilities. Please attempt to complete as many tasks as you can. Our positions require people with different abilities, so do whatever you feel comfortable doing within your knowledge and the time available to you. This will help us see which part of the organisation you're most likely to fit into. Familiarise yourself with the Amazon.com website and do the following tasks.

1. **Acceptance criteria implementation**

Using Java and Selenium/Webdriver, implement the following tests on the Chrome browser.

- Steps:

  * Enter amazon.com and check the homepage
  * Search by word "laptop"
  * Add the non-discounted products in stock on the first page of the search results to the cart
  * Go to cart and check if the products is the right

2. **Simple Site Crawl**

Write a crawler that opens up the "Shop By Department" dropdown menu on the amazon website, obtains a list of all department links and visits them to make sure that there are no dead links.

Your crawler should keep a list of visited links in a text file in the form (link, page title, status) , where status can be "OK" or "Dead link". After finishing, the crawler should name the file <timestamp>_results.txt.

3. **API Test**

https://jsonplaceholder.typicode.com/posts (GET) is a mock API endpoint which simulates the retrievals of blog posts. Implement the following test scenarios:

**Scenario:** Counting posts for user <user> When I get a list of blog posts using the API endpoint Then user <user> should have <numposts> posts.

The scenario should execute with the following values of (<user>,<numposts>): (5,10), (7,10), (9,10)
**Scenario:** Unique ID per post When I get a list of blog posts using the API endpoint Then each blog post should have a unique ID.

**4.** Git Usage, Best Practices, and Submission Details

Ensure that your codebase is hosted on a Git repository. As you work on the tasks, adhere to the following guidelines:

- Create meaningful commit messages that detail what changes were made.
- Use branches to separate different features or tasks.
- Ensure your code is well-documented, with comments explaining intricate or complicated parts.
- If possible, write a README file for your repository detailing the purpose of the code, how to run it, and any other relevant details.
- Ensure that sensitive information (like API keys, passwords, etc.) is not hardcoded in your codebase. Use environment variables or other secure methods instead.

For submission, push your completed solutions to your GitHub repository and provide us with the repository URL. Ensure that the repository is public so that we can access and review your work.

We are looking forward to receiving your task.

If you have any questions, reach out to the Talent team.