

QA Engineer - Technical Challenge

Dear candidate,

Thank you again for your application for the position as a **Quality Assurance Engineer** (m/f/d). Today, we want to get a deeper understanding of your knowledge and problem-solving skills.

Automation Test Challenge

Consider the following situation, where you have a new website launched, and you are asked to automate a suite of test cases. The suite should run on any browser (i.e. cross-browser testing support) and should run against different environments (e.g. local, test, staging or in a Jenkins pipeline leveraging Docker containers). For example, the test cases should run against in the following environments:

- **Local:** <http://localhost:4000/fashionhub/>
- **Staging (dummy environment):** <https://staging-env/fashionhub/>
- **Production:** <https://pocketaces2.github.io/fashionhub/>

You can run the application locally as a container from this Docker image: [Fashionhub Demo App](#).

The expectations are (read carefully):

- The program must run and return result
- Implement the solutions using Playwright or Cypress.
- The code must be well-structured to support future maintenance, evolutions and scalability. Try to use the best practices that you have learned from your experiences approaching production like quality is more important than implementation speed and feature completeness.
- A build procedure whenever applicable (compiled language)
- Any environment option needed to run the test could be passed from the command line or from a config file (the system should verify which one is the preferred option, and select the other one if the primary one is not present)
- As a deliverable, we expect you to push the code to a repository and share the link. Please add also a README file containing instructions so that we know how to build and run your code

For this challenge, please implement all of the test cases.

Test Case 1

- As a tester, I want to make sure there are no console errors when you visit <https://pocketaces2.github.io/fashionhub/>

Hint: you can use the about page to test your implementation as this contains an intentional error

Test Case 2

- As a tester, I want to check if a page is returning the expected status code
 - Fetch each link (e.g. on

<https://pocketaces2.github.io/fashionhub/>) and visit that link to verify that:

- the page returns 200 or 30x status codes
- the page returns no 40x status codes

Test Case 3

- As a customer, I want to verify I can log in to <https://pocketaces2.github.io/fashionhub/login.html>

Hint: use the following details to login: **Username:** demouser **Password:** fashion123

Test Case 4

- As a product owner, I want to see how many open pull requests are there for our product. You can use <https://github.com/appwrite/appwrite/pulls> as an example product
- Output is a list of PR in CSV format with PR name, created date and author