

Qover – QA Test Assignment

Charter	Focus on Test Implementation
Testers	Nikita Danylov
Environment	macOS Monterey, Version 12.7.5, Chrome Browser, Version 126.0.6478.127 (Official Build) (x86_64)
Started Time	02/09/2023 12:07 PM
Duration	10
Part 1	7.5
Part 2	2.5

Test Automation Frontend

I start with choosing what to automate. For that reason, I create Product Coverage Outline. Product coverage outline is a great way to outline what is there to test.

I've chosen to cover specific for frontend area that would allow to showcase certain how we deal with certain issues that appear when we automate: design choice, duplication, readability of the code, choice what to cover with the automation.

Specifically, I've chosen to cover links and flow. However, I did the implementation of the links part only due to the time constraints.

The page is available in multiple languages, so it makes all locators based on text not reliable. Therefore, I chose using a custom CSS locator to locate elements on the page.

What would be your choice for framework when automating the web application and why?

My choice of framework was dictated by the software culture at Qover. According to the position description and what I learned during the interview, the Qover team chose Playwright as an automation tool, and developers write their code in TypeScript.

Another reason I chose Playwright is that the product must support all major browsers—Chrome, Mozilla, Edge, and Safari—as well as smartphones and tablets. Playwright provides the capability to execute tests on all of these browsers and devices.

Additionally, Playwright has less issues with React-built interface components.

Comment on what you would implement in an ideal world (with plenty of time and resources for testing), how and why.

It is vitally important to conduct internationalization testing for this website. The number of issues I encountered while exploring it proves this necessity. Testing manually is time-consuming, repetitive, and prone to missing small mistakes, especially in foreign languages that the tester does not know. The how part may include multiple options. One of them would be using Jasmine-Data-Provider and Playwright. Playwright would scan text from the target pages and Jasmine assertion library will match expected and actual results.

Once, again, because the product must support all major browsers—Chrome, Mozilla, Edge, and Safari—as well as smartphones and tablets, I would invest my ideal world unlimited resources into making sure it supports all major browsers and devices. Playwright is already a good choice for that. The next steps would be to correctly identify scenarios that are appropriate for those browsers and devices.

Additionally, I would cover Flows with automatic checks. I've noticed that the main flow requires many steps, with the user needing to fill in numerous fields and make several choices. I would ensure all flows are covered, as it is definitely worth implementing automation for them. The risk is that maintaining these tests may become to challenging and time-consuming. The design choice of the test framework should be robust enough from the beginning to handle all future changes. Therefore, I would consider automating certain functional checks at the API level and only implement those that are relevant to the UI. I've already seen that validation is done at the UI level, so I would cover this validation using Playwright. I would consider using automatic data-generators to make functional checks more diverse and interesting.

I would use Playwright to test how the product meets accessibility standards. Playwright supports axe, a library specifically created for accessibility testing, which is also available as a browser plugin. Axe and Lighthouse browser plugins can be used to conduct quick accessibility checks.

Test Strategy And Planning

Considering Qover business model and products, what type of tests would you prioritise, automate and execute?

- Function Testing – *what the product's functions can do*
- Flow Testing – *do one thing after another*
- Domain Testing – *data processed by the product, outputs and inputs*
- Stress Testing – *overwhelming the product*
- Security Testing – *protection of the product against malicious usage*

Considering the application provided (url above), could you make a checklist of features/scenarios to test? (No need for deep step by step test cases, but a list of items you would execute to test our application)

#	Items
1	Make sure links are clickable and navigate to the expected page/resource which is available
2	Analyze text for mistakes and typos
3	Click 'See prices' while leaving 'Type of bike' and 'Insured value' fields empty, and keeping the 'Bike' field at its default value
4	Click 'See prices' with 'Type of bike' and 'Insured value' fields filled in with correct values, and keeping the 'Bike' field at its default value

Problems

- The Back button is not clickable on the /bike/quote page in the EN or DE version of the website
- Clicking on the Logo icon redirects to non-existent page in the EN or DE version of the website
- Preview your contract and Summary of your requirements and needs – “The document could not be downloaded. Please try again and contact our customer service if this happens again.”. The same occurs on trying again.
- After clicking on the button Proceed to payment on the page /bike/summary, selected checkboxes “Terms of service” and “Consent electronic letters” (if they were selected) are shown as unselected which may confuse a user. On clicking on Back, they remain to be unselected
- See all FAQs link leads to a page that does not exist /hc/en-be/sections/360002382398-Bike
- When changing the language to any other than English, clicking on links still leads to downloading documents and opening pages in English
- Missing country flags and duplicated Country – LANGUAGE in the Country – LANGUAGE drop-down list in the
- “general.localesWithFlag.cs-ES” is shown instead of Catalan – ES in the Country – LANGUAGE drop-down list and choosing it causes 404 Not Found page to be open
- Placeholders “bike.optionLabel” are shown instead of examples of plans in the Spain – EN version of the website

Issues

- Downloads start automatically on clicking on the link
- New tab is not visible in Time Travel