

QA Automation Engineer Interview Assessment

Introduction [🔗](#)

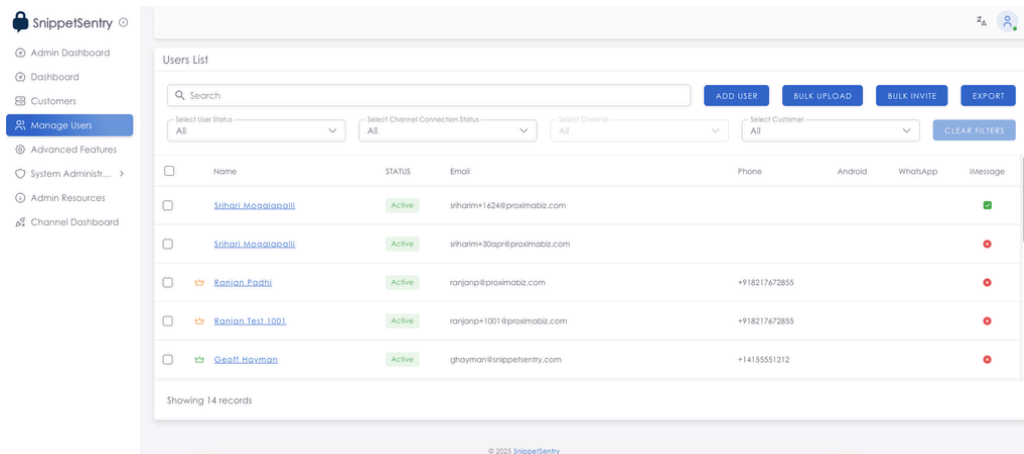
Thanks for taking the time to interview with us! We'd like to assess your QA skills, both from a manual/exploratory point of view, as well as from a technical automation/coding perspective. The goal of this assignment is not to give you any busywork or take up too much of your day, but rather see how you would go about solving the following problems that we have had to solve already, and discuss why you chose to approach it the way you did. Please timebox yourself to 2-4 hours - do your best to provide a complete solution, but don't waste time on anything extraneous, like setting up test reporters for your automation framework or trying to exhaustively capture every test case you can think of. Feel free to send a message back to us with any questions you have after you've read through this document.

All of the exercises relate to our own product (referred to here as "SnippetSentry"), and you can access our "beta" environment for your testing at: <https://beta.snippetsentry.com>

Your credentials to log in have been uniquely created for you and an email to set your password should have been sent to your email address; let us know if you need us to re-send the email.

Part 1: Manual/Exploratory [🔗](#)

We'd like to see how you would test some functionality found under the "Manage Users" section of our website. Specifically, please create a test plan for how you would handle the "CRUD" operations for our "users". I would expect no less than 5-7 test cases in this plan, but don't go past 15.



Here is some advice that can help you succeed:

- You're being dropped into a project without any explanation for how it works or its purpose, which is exactly what happened to me when I first joined. Thus, part of the fun is trying to figure out how this feature is supposed to work and what its purpose is. All I can ask is for you to embrace it; perfection or a full understanding is not expected.
- For this exercise, we don't care about testing the other buttons on the page (ie. "Bulk Upload", "Export", etc.) - just stick with "Add User" and the Users list.
- Feel free to make as many users as you want (within reason), provided you have a way to get the emails for each one you make!
- At minimum, this should cover the basic "happy path", but we encourage you to explore the functionality and see what happens when you try modifying a user's configuration in certain ways.
- Can you figure out the different statuses a user can have, and what each one means?
- Think not just about the user you create in isolation, but also how other created users could impact it. And what about the user you're logged in as?
- Functional testing is a requirement, but is that all there is to a complete test plan?

Part 2: UI Automation [🔗](#)

Now that you have a test plan, we should automate some of these tests so that we don't have to keep checking them manually!

Pick 3-5 test cases that you wrote in part 1 and automate them in a modern, end-to-end UI automation framework. Pick any framework and programming language combination you'd like: what's most important is that you're comfortable working with your choice and that you can explain how it works to us if it's not using a framework and/or language we're familiar with.

That being said, if you have a wide range of experience and/or no preference, we recommend using either the Playwright, Cypress, or WebDriverIO frameworks, where tests are written in JavaScript/TypeScript (or additionally Python/Java if you choose Playwright).

Some more helpful advice to succeed here:

- Your primary goal is to translate your test cases into readable code in your framework. Then, get it to pass (or fail expectedly) to the best of your ability.
- What does clean test code mean to you? How do you accomplish that?
 - What are the minimum amount of dependencies you need to install to achieve this vision?
- Will you stick with the tried and true Page-Object Model (POM), or do you prefer a different architectural pattern?
- How can you best take advantage of your framework's features to simplify/reduce the code you write?
- How do you know it is running the way you expect it to (in other words, catching false positives/negatives)? How do you go about debugging your work if it doesn't run correctly the first time (it'd be a miracle if it succeeded every time)?
- You've probably received guidance in the past not to under-comment on your work... but what about over-commenting?

Conclusion [🔗](#)

When you're ready to show us your work, simply provide one of the following through the contact method you've been using to communicate with us:

- A link to a code repository that contains your automation framework, as well as a text document that captures all of your test cases
- An attachment that contains your automation framework and test case document in a compressed file format (ZIP/RAR)

Best of luck, and thank you again for your time. We hope you enjoy the challenge!