Test Automation Engineer Assessment

Disclaimer

Specops Software provides no compensation for the completion of this technical assessment. This technical assessment is a part of the interview process, and no results from this assessment will be used for other applicants, or for any legitimate tests. This assessment is solely a tool for assessing and getting to better know candidates, and the platform and 'Cybr' company is purely satire.

Contact

517 Wellington St W, Suite 400
Toronto, ON, Canada
M5V 1G1

Contact: david.ketler@specopssoft.com

Changelog

v1.0	2022-08-18	Writeup of assessment instructions.

Table of Contents

Introduction

As part of our Test Automation Engineer recruitment process, we request that any applicants complete a technical asssessment step. This technical assessment is used as the base for the next phase of the interview process. The follow-up interview will discuss:

- The completed test suite: what tests were written and why
- Any issues with the implementation of the tests
- Any issues with the web application itself: were any bugs discovered that would need to be filed?
- And lastly, general questions about previous experience, interests, and questions you have for us

Note: the goal of this assessment is *not* to determine whether you can write code, or how complex you can build a test suite. You are welcome to follow a similar format to the provided base tests; the hope is that it will be a fun exercise to show how you think about testing, how you go about exploring a new product and deciding what to test –and how you plan automating a suite of tests for a new project.

The Web App

Imagine we are building a new Cryptocurrency platform for sale on the general market. The dev-team has completed their initial MVP of the website for the coin. The Blockchain work is still ongoing on the backend, but the QA team has been asked to complete automated UI tests for the website. The website was built by the development team in .NET MVC, with an SQL backend; CI/CD has been implemented and the application is published to an Azure site.

The development team is reasonably junior, since this *is* a cryptocurrency startup, after all, so there are some concerns that they may have missed some security considerations or debug functionaliy, and there may be logic bugs.

The Assessment

The application is hosted on an Azure site dedicated to only this application, and the database is purpose-built for this site. You are welcome to run any tools you wish against this platform, if you have some utilities you like to use during your testing. However, you are not to complete any Denial Of Service attacks against this site or host. The data is backed up and can be restored through a publication, should any data be lost or removed during testing.

A senior member of the team has provided a base test suite that can be used as the basis for the tests. This test suite was written in Cypress. You are not *required* to use it, however it is recommended that you do so, in order to same time. If you would prefer to use another test framework, such as Selenium in a language of your choice, or Cucumber, you are free to do so.

The website is comprised of the following basic functionality; tests for some of this functionality have been completed in the provided test suite, while other portions will need to be completed:

- User login, with Administrator and User roles
- Registration for new users
- A blog where the company will provide news updates –this blog has CRUD functionality when an administrator is logged in
- Some basic informational pages about the platform and how it works, and a privacy policy
- Search functionality that searches the blog

You are welcome (and encouraged) to explore the site, see what issues you can find, and write the appropriate missing tests to verify functionality as you see fit. If you encounter any issues, make note of them. You do not need to file workitems, but make note of your reproduction steps. Treat the site as if you were completing a test cycle; make note of your reproduction steps and symptoms, and move on to the next test.

There also may be some easter eggs or other quirks to discover; make note of anything you come across —that is, if it's not something you feel you should be writing tests for.

Methodology

The application to be tested is located at cybr20200827202321.azurewebsites.net. You are the only concurrent user on this system, so do not worry about how your data might affect other tests.

This assessment can be completed in any language and test framework of your choosing, but if it is especially exotic please include documentation on how to build and run the tests. The functionality that should be tested is your decision, you can write as many or as few tests as you choose; but keep in mind the intent of this assessment. It is expected that the assessment only require an hour or two to work through. Tests should be submitted in a runnable state; should you choose a different test framework, if it requires any instructions to run, please include them.

The tests can be structured however you see fit, in one or many spec files. If you wish to restructure the given tests, feel free to do so, but you are not required to.

Credentials

The credentials are as follows:

- Username: Administrator
- Password: xisE4P5DsjT4Hq
- The URL again: cybr20200827202321.azurewebsites.net

If you run into any issues or questions regarding the assessment, please feel free to email david.ketler@specopssoft.com.

Setup Instructions

If you would like to use a different language or test framework from the given example, you can ignore the following sections. Simply build the test suite as you choose and submit it, providing any necessary instructions. However, we do recommend using this sample project as it will save you a lot of time. Since the sample project is built in Cypress, a testing environment will require the following:

- An installed web-browser: chrome, firefox, chromium etc.
- A working recent node.js install
- A working recent npm install
- A text editor/IDE of the candidate's chocie: vim, vscode, etc.

If you require any of the above, please follow the installation procedures for your operating system. Once your environment is setup, navigate to the `SpecopsSoftwareAssessment` folder, and run npm install.
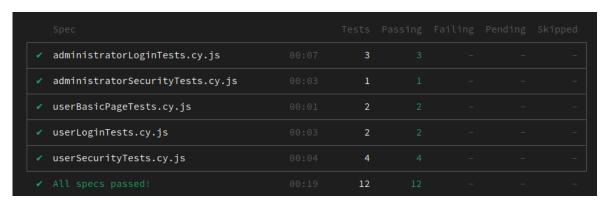
Running Tests

If you're new to Cypress, this method of writing tests may be new to you. There are a couple of ways to work with the Cypress framework:

- Run: headlessly execute the test suite, record video and screenshots, and return the results: similar to just running a junit or mstest suite
- Open: interactive mode: keep the test runner open, run in a browser —this will also re-run the tests as you develop; on file change the open spec will execute

Commands

To simply run the tests headlessly, open a terminal or command prompt, navigate to the folder you extracted the test suite to, ie: ~/*SpecopsSoftwareAssessment*, type `npx cypress run`, and press Return. You will see results similar to the following:

	Spec		Tests	Passing	Failing	Pending	Skipped
✓	administratorLoginTests.cy.js	00:07	3	3	–	–	–
✓	administratorSecurityTests.cy.js	00:03	1	1	–	–	–
✓	userBasicPageTests.cy.js	00:01	2	2	–	–	–
✓	userLoginTests.cy.js	00:03	2	2	–	–	–
✓	userSecurityTests.cy.js	00:04	4	4	–	–	–
✓	All specs passed!	00:19	12	12	–	–	–

To run the tests interactively, and have the tests run when you make changes, navigate to the folder you extracted the test suite to, ie: ~/*SpecopsSoftwareAssessment*, type `npx cypress open`, and press Return. You will then see the Cypress UI appear. Click the `E2E Testing` button, select your browser of choice, and click `Start E2E testing in <browser>`. You will then see the list of test specs that can be run; you can return to the list by clicking on the `Specs` button on the left-hand-side. As you make changes to the spec you last ran, Cypress will execute the changes upon detecting a file change. If you want to run a specific test only, you can add the command `only` to the test, ie `it.only("Attempt to login`).

Test Project

This sample project is provided as an optional, albeit recommended, base for the test suite. A set of classes have been to provide some initial coverage for the site, this code can be used as a base for the rest of the tests. These sample spec files are are non-exhaustive set of starting tests, you are recommended to consider what other test scenarios are required, both within these given specs (as they are incomplete) as well as any other types of tests you feel should be run. We recommend starting by doing some manual exploratory testing to get a feel for the website, before you work on coding. If you're new to Cypress, it's good to note the following files that are part of the test project:

- `SpecopsSoftwareAssessment/cypress/support/commands.js` contains custom test commands, you'll see a couple of these used in the implemented tests.
- `SpecopsSoftwareAssessment/cypress/fixtures/users.json` the fixture file containing the existing users, you can add to this if you wish to add more test data.
- `SpecopsSoftwareAssessment/cypress/e2e/` contains the implemented test spec files.

Add new tests, and test specs as you wish, you're free to restructure the test suite if you believe it needs it —build the test suite however you see fit.

Submission

Once you have completed your test suite to a level you feel is satisfactory, please run a final test run to ensure the tests work as expected. After that, please zip the whole folder (SpecopsSoftwareAssessment) and either:

- Compress into an archive of your choice, upload to a file sharing site, share the file, and email the link to the file
- Compress into an archive of your choice, attach the file to an email, and send the email

Good luck with the assessment, and we hope you have fun exploring Cybr Coin's website. We look forward to meeting you and talking about joining the Specops Team.