Welcome to the Ceros Ski Test and Automation Challenge

QE Edition, v3

We would like you to take a little time to show your thought processes when testing a product, and show off your skills in both manual and automated testing. Track your time and let us know how long you've spent in each area of the challenge. If you have any difficulties feel free to reach out to us with questions.

There are four sections to the challenge:

1. Creating a test strategy
2. Manual testing/Bug reporting
3. Test automation
4. Bonus

Create a test strategy

For the manual testing portion you can find the Ceros ski game at the following address http://ceros-ski.herokuapp.com/. You will need to create a test strategy for at least 2 existing features of the game, along with a new feature that is about to be developed that is described below. In addition to the test plans, describe why you chose the features that you did so we can get an idea of how you evaluate risk.

New Feature: Jump!

Acceptance Criteria

- Have the skier jump by pressing a key
- Have the skier jump whenever he hits a ramp.
- The skier should be able to jump over some obstacles while in the air.
    - Rocks can be jumped over
    - Trees can NOT be jumped over

Manual testing/Bug Reporting

The Ceros Ski game is known to contain at least a few bugs. Take some time playing the game and testing it to find the hidden bug(s) and write up a bug report that the developer can use to fix it.

Automated Testing

For this section you will be given the opportunity to show off your skills in automated testing. You can use either the basic Javascript/Protractor framework we are providing or a framework you create yourself. You can download our basic framework here: https://ceros-dev-code-challenge.s3.amazonaws.com/ceros-automation-code-challenge-v3.zip

After unpacking the zip file you will want to make sure to go through the README for further installation instructions.

> If you do not have familiarity with Protractor you can find a reference guide at https://www.protractortest.org/#/. If you have no familiarity with JavaScript and would prefer to build your own framework, that is acceptable. We would ask that you still look through the provided code and make note of automation best practices that aren't being followed so you can list those in your submission.
>
> In any event, make sure you use automation and programming best practices!

You will notice that we have included 6 tests (4 required, 2 bonus) in `swaglabsSpec.js`. You'll be writing tests against https://www.saucedemo.com:

1. Verify the standard user can log in (credentials found on the SwagLabs login page).
2. Verify you can add an item to the card and that it's visible on the cart page.
3. Verify that, by default, the inventory page lists 6 items.
4. Ensure the user can complete the purchase/checkout process.

Bonus Automation Challenges:

1. Verify you can sort the inventory items by price, high-to-low, and the sorting is correct.
2. Ensure you can sort the inventory by name, Z-to-A, and the sorting is correct.