Adrian Sommer

4/12/17

QA ENGINEER CHALLENGE

Write out a test plan for custom ordering a Tesla S at https://www.tesla.com/models/

design). For tests that you feel should be automated, write an automated test suite using nightwatch.js that covers some of the major pain points manual testing would encounter. Explain the reasoning behind your automated test choices.

If any tests fail your expectations, please note them as well.

Browsers to test: Chrome & Firefox

*Part 1: Test plan*

**Choosing custom options**

Test that:

* Selecting an option causes certain updates to occur on the page
  + The price is updated
  + A checkmark, highlighted border, or other visual indication of the selected option manifests itself
  + An option can be selected, unselected, reselected, without causing errors
  + The car image at the top of the page changes when cosmetic options are chosen

**Print spec sheet link**

Test that:

* clicking the link opens a new tab
* the tab shows all the options that are selected on the order page
* the tab has images that reflect the cosmetic choices of the car

**NEXT button**

Test that:

* clicking the button takes the user to the ‘Order or Save Design’ page, without opening a new tab

**Order or Save design page**

* going back a page on the browser will take the user to the previous page and preserve all selected options
* selected options are displayed on the upper right part of the page
* the car image on the upper left matches the selected choices
* the default option under the header ‘Do you prefer to own or lease?” is set to the same option chosen on the previous page
* All fields on the order form are required, with the exception of ‘Billing Address Line 2’ which is optional
* If any required field is omitted, the order cannot be placed
  + All missing fields will be highlighted in red and the word ‘required’ will appear in red in the bottom right hand corner of the field
* additional links on page:
  + ‘Model S Order Agreement’, ‘Customer Privacy Policy’
    - Opens a new tab with relevant info
  + ‘Request a callback’
    - Opens a pop up where all fields are required, with the exception of ‘When do you want us to call you?’
    - Default country selected is ‘United States’

*Part 2: Automatic tests*

Test1.js

* Goto the Tesla S order page
* Select several custom options
* Verify the price is updated correctly
* Click the ‘Next’ button to goto the payment page
* Verify that the details on the payment page match what was selected

Reason for automating:

Between the mechanical and aesthetic options available for the car, there are innumerable combinations of options that can be selected. Performing manual tests of multiple options is time-consuming and has potential for error, since comparing many options between two web pages poses many opportunities to make a mistake.

Test2.js

* Click ‘Next’ button to goto payment page
* Send invalid email address to email field
* Verify that error message appears

Tes3.js

* Click ‘Next’ button to goto payment page
* Send valid-looking Credit Card info
* Verify that no error is thrown

Reason for automating:

If possible, a feature should be tested in both a positive and negative fashion. For example, in a text field that takes user input, invalid input should cause an error message (negative test) and valid input should be accepted (positive test). This is what Test2.js and Test3.js

strive to accomplish. In a full test suite, this type of test would ideally be applied to every input field. The burden of manual testing increases with the number of fields, which is why I chose these type of tests as good candidates for automation.

*Part 3: Do any tests fail expectations?*

The major problem I encountered is that using Nightwatch with Chrome causes difficulty in clicking on elements. In Firefox, I have no problem programmatically clicking on any element, but with Chrome I’ve found that only certain elements can be clicked on. I’ve tried different click methods on different websites with varying degrees of success, but it seems the Tesla site in particular causes some issues in Chrome. This is why the config file I’m sending is set to Firefox by default. I’m curious if you’re able to run these tests in Chrome – to do so, uncomment line 28 of my config file and comment line 29.

I’d like to give credit to the following sites for helping me setup Nightwatch:

http://nightwatchjs.org/gettingstarted/

https://github.com/dwyl/learn-nightwatch