**RV QA Assignment**

Tests as the appear in Project.js:

1. Cart Open
2. Check Page Title
3. Check Phone Number
4. Check Shop Frontier Internet Plans Button
5. Check to See If Drop Down Menu is Enabled
6. Check Internet Availability
   * Set input value
   * Click Check Availability Button
7. Verify Drop Down Menu Items

The order of the tests that I created was not based on importance but rather just how I put them together as I looked through the test website. After I was finished putting together the tests I tried to rearrange them in the code file based on importance but it slowed down the execution of the test to the point that it timed out every time, even when I adjusted the timeout time in the config file. For all test I used the Test Runner with Mocha to perform tests on Chrome and Firefox.

By Test Importance:

1. Check Page Title
2. Cart Open
3. Check Internet Availability

* Set Input Value
* Click Check Availability Button

1. Check Shop Frontier Internet Plans Button
2. Check Phone Number
3. Check to See If Drop Down Menu is Enabled
4. Verify Drop Down Menu Items

Check Page Title

I used this test to verify that the page could be opened in the browser and to send back the name of the page that was opened. Found that the INFR homepage worked fine in Chrome but would not open in Firefox for some reason and so any further tests would not execute.

Cart Open

The cart function is of huge importance to the RV Frontier Business as it is a primary driver of third party sales and data collection and analysis. Used this test to try to verify that the cart opens with the button is clicked. I wanted to add more functionality to the test to verify that the page that loaded was the cart page but when I tried to add this part with exactURL or notExactURL the test broke.

Check Internet Availability

When new customers come to the site they usually want to either check to see if service is available at their address or see what products are available to them. Taking this into account, I attempted to have a test to 1. enter a zip code first to test that values could be entered and then 2. after the value was added click the availability button to verify that an availability check would run. I’m sure I could have combined this into one test but this was how I initially wrote it out.

Check Shop Frontier Internet Plans Button

After checking availability through zip code, the next test in my opinion was to check that the plans button worked. This was a simple test like checking the cart open button test from above.

Check Phone Number

I wanted this test to verify that the phone number was visible on the homepage. I first attempted to getText from the span element that had the INFR phone number but the test wouldn’t execute because there isn’t any text. It appears that the phone number links to an application that will call the number for the customer so I attempted to verify the link instead. What ended up happening was that the test verified and returned the name of all the links on the homepage.

Check to See If Drop Down Menu is Enabled

This was the first test that I tried to put together to verify the links in the sub-menu “Plans and Pricing.” I thought that the isEnabled protocol would work to do this but couldn’t ever get the test to recognize the elements that I listed from the id’s taken from the homepage.

Verify Drop Down Menu Items

I tried the menu test from above from a different angle by first getting the test to click the menu header element, wait for the drop down to become visible, and then click each of the links underneath. As above, I couldn’t get this test to work with the id’s that I was able to pull from the menu elements.

These tests are by no means exhaustive as I’m sure there are many other aspects of the homepage to test with more complex ways of testing them.