Shipt - QA Engineer - Interview Exercise

Upon completion of this exercise, push all your files into github and send us a link: <https://github.com/natewilliams1/homework>

1. Navigate to www.shipt.com. Choose a feature of the site and write a description or test

case to describe how to test the feature.

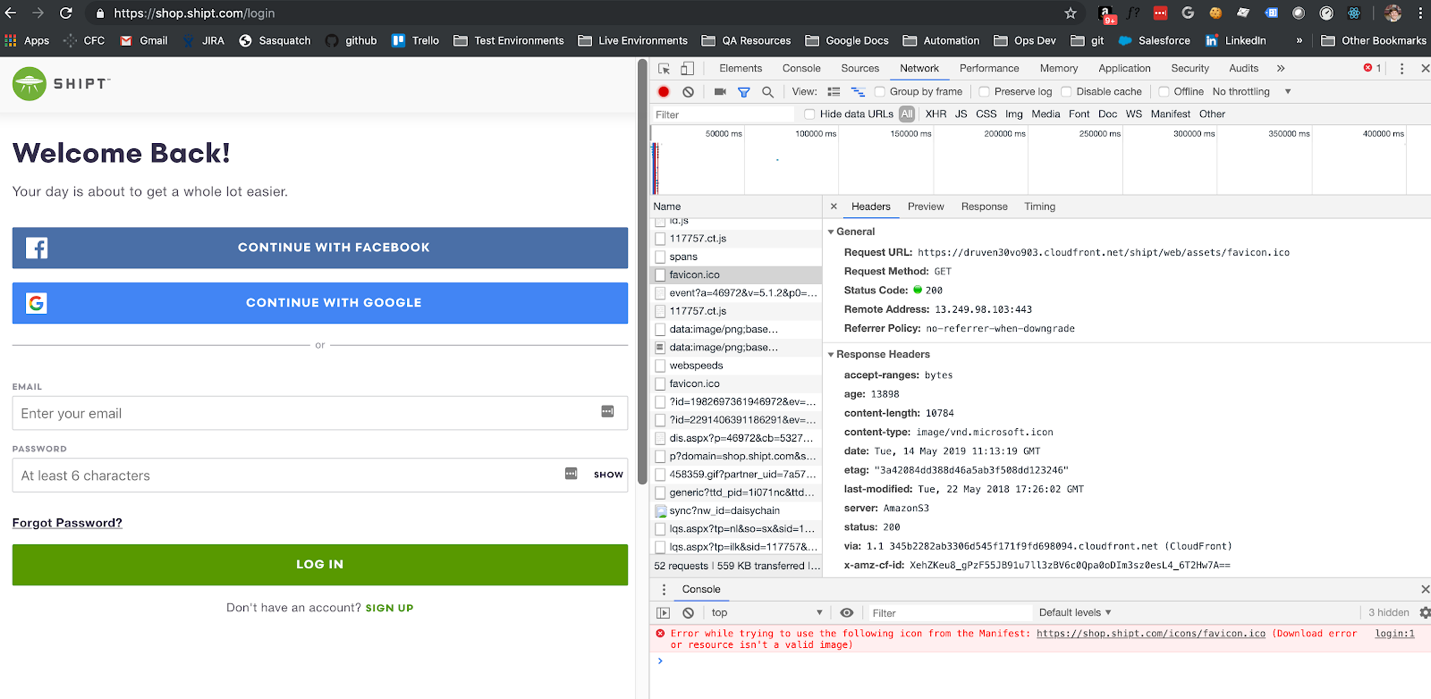
Test Case Summary: Test the Member Login Functionality

Requirements:

* User is able to login with valid credentials
* User is unable to login with invalid credentials
* Page contains valid Forgot Password option
* Page contains validation that prevents invalid entries
* Pages are mobile responsive

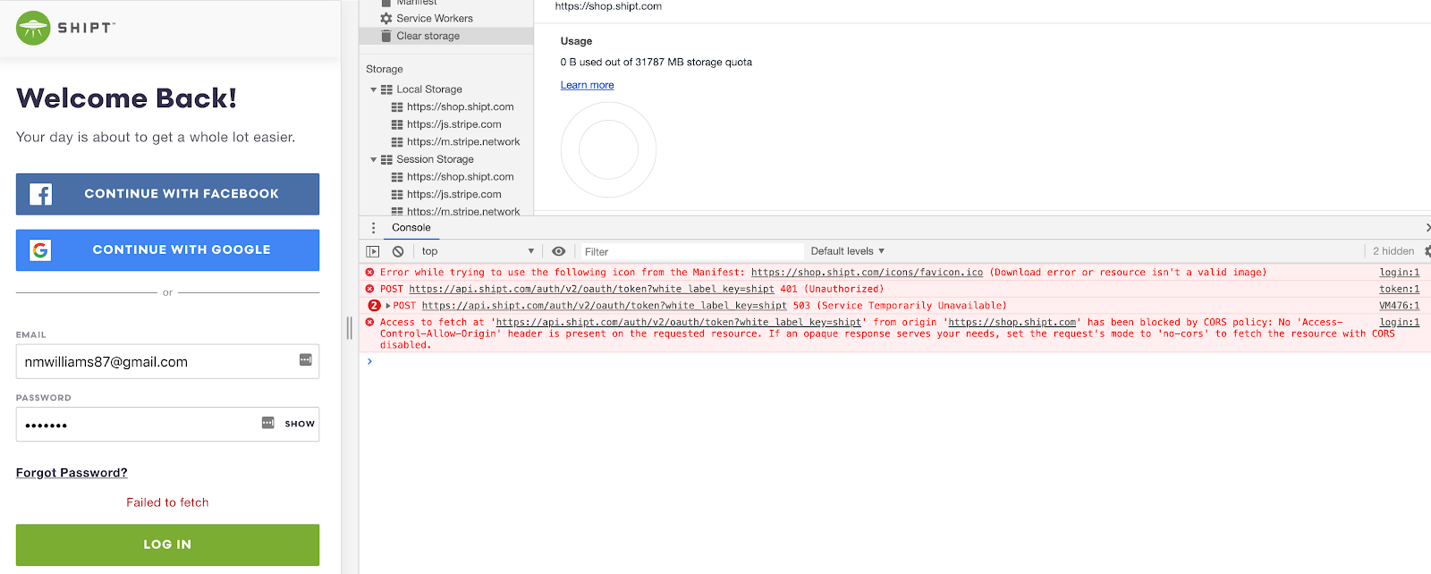
Valid Login:

* Setup a user using the Sign Up feature from <https://www.shipt.com>
* Navigate to <https://shop.shipt.com/login>
* Email: [nmwilliams87@gmail.com](mailto:nmwilliams87@gmail.com)
* Password: Test123
* Log In
* Verified that the login is successful
* Verified that Facebook login works appropriately (directs me to FB login page if not signed into FB; logs me in directly to profile if I am already logged into FB)
* Verified that Google login works appropriately (directs me to select my google account if I have multiple;
* There is an error on the page indicating that <https://shop.shipt.com/icons/favicon.ico> was not downloadable or that the icon isn’t valid.



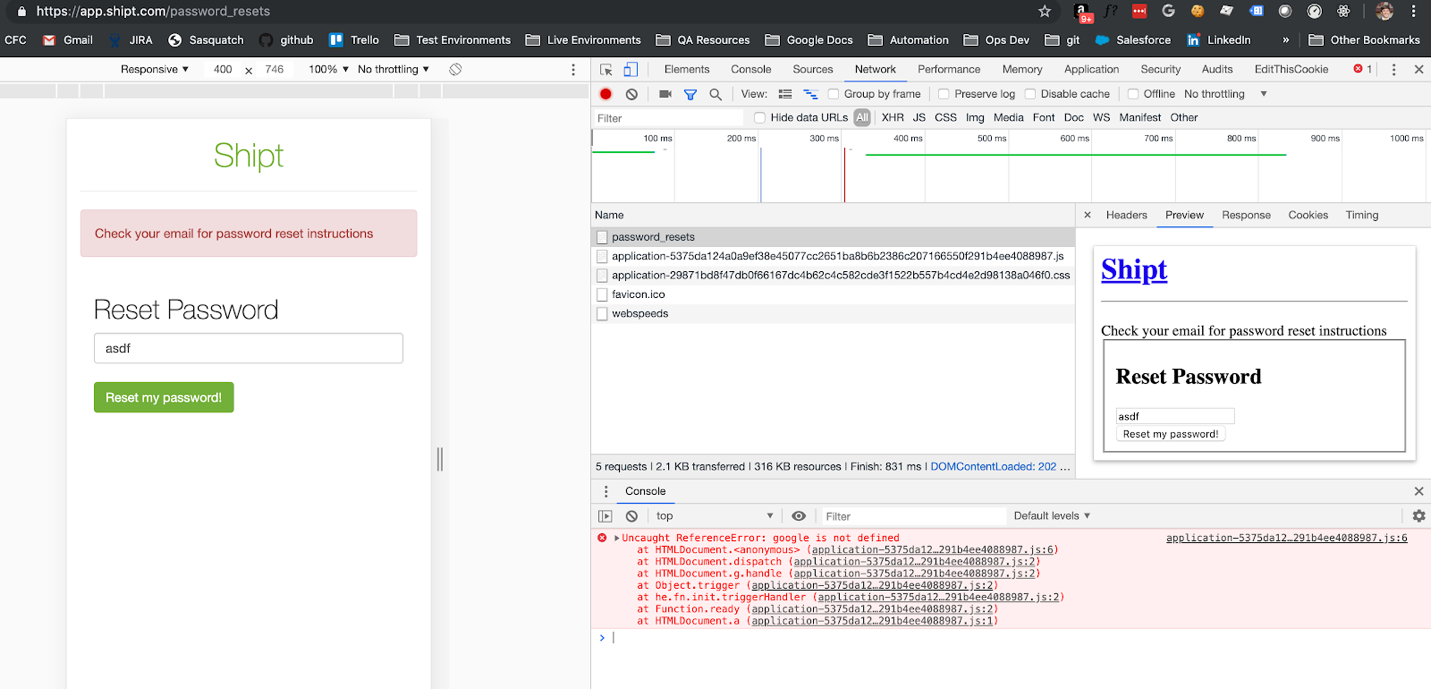
Invalid Login:

* Navigate to <https://shop.shipt.com/login>
* Email: [nmwilliams87@gmail.com](mailto:nmwilliams87@gmail.com)
* Password: Test1234
* Log In
* Verified that login failed and ‘Invalid Username or Password’ message was surfaced
* Noticed an issue where spamming the login button surfaces ‘Failed to fetch’ message rather than ‘Invalid Username or Password’.  This occurs when API response is 503 (Service Temporarily Unavailable) rather than the normal 401 (Unauthorized).



Forgot Password:

* Forgot password link navigates to <https://app.shipt.com/password_resets/new>
* Field requires text before submission
* Entering email and clicking ‘Reset my password!’ sends email
* I was able to reset my password from reset email
* Validation on the password reset fields prevents spaces
* Validation on amount of times a password reset can be initiated within a 5 minute window.
* No redirect link from the password reset page.  Clicking the Shipt icon takes me to a page with an onion but there is no way to get back to login without clicking back or manually manipulating the URL.
* Password can be set to same password as before.  This could be a security concern.
* Email field does not have the same validation as login page.  We should probably include validation that requires an @ symbol on the reset page.
* Reset email url (<https://app.shipt.com/password_resets/new>) throws a javascript error in Chrome, Firefox, and Safari.  Error message indicates that we need to include the google library before the script calling the library.
* Resetting password results in a page that says ‘Password was successfully updated’ but does not contain a redirect link.  We should probably include a way back to the login page after the password reset process.



Page Validation:

* Verified that email is required.
* Verified that password is required.
* Email field contains email format validation.
* Invalid Username or Password validation triggered with incorrect entries.

Mobile Responsive:

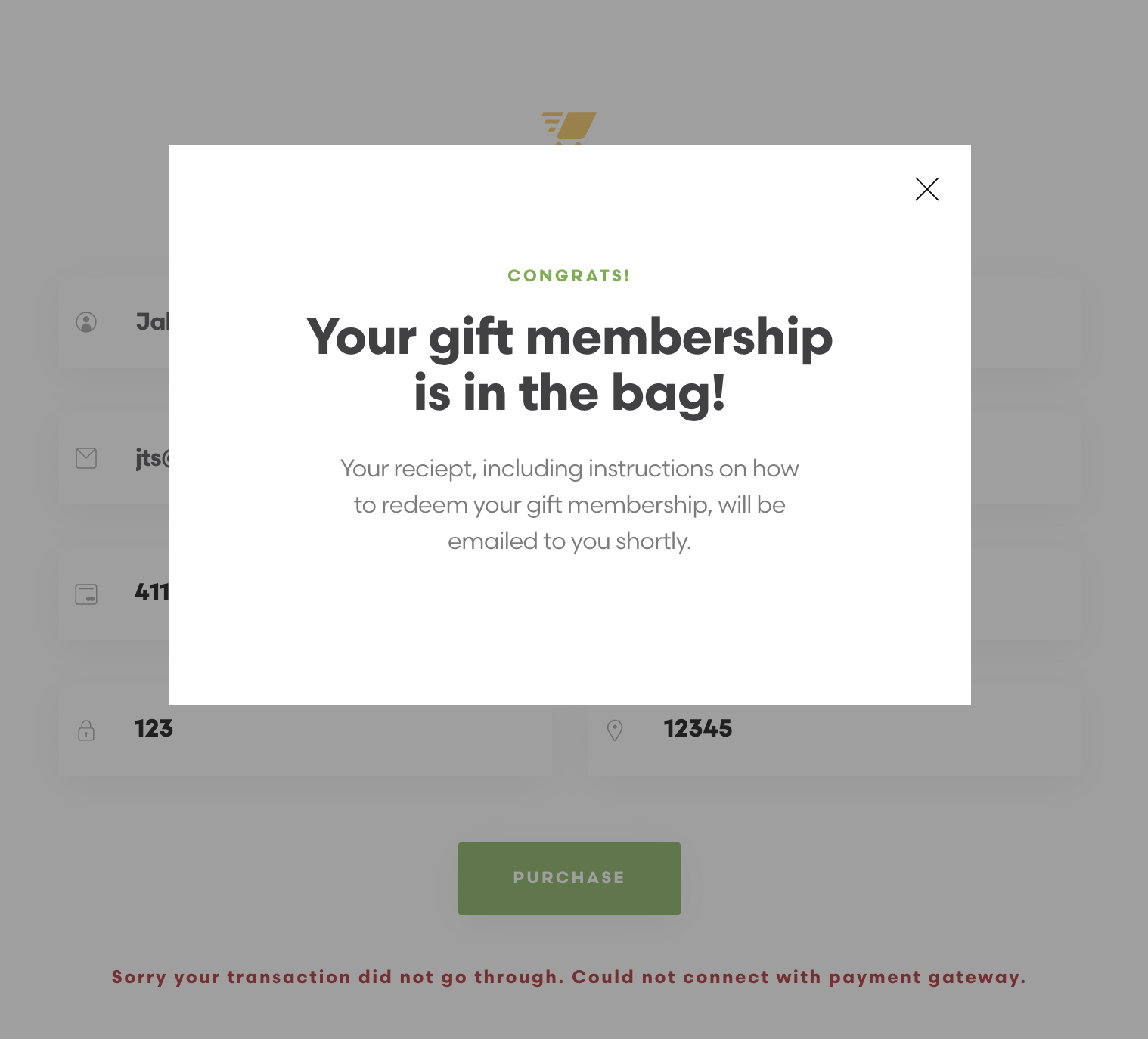
* Verified that the page is mobile responsive and works appropriately on ios and android devices.
* Verified that the page restructured at 767px.
* Verified that the look and feel of the page was appropriate.
* Verified that all embedded link worked appropriately before and after page restructure.

2. Locate one bug or bad workflow within the app.

a. Explain the behavior you are seeing

When attempting to purchase a gift with a test credit card number, the transaction fails but a success modal pops up that says “Congrats!  Your gift membership is in the bag! Your receipt, including instructions on how to redeem your gift membership, will be emailed to you shortly.”

b. Include any useful errors or screenshots



c. Explain why and how it needs to be corrected

We need to prevent the gift completion validation from prompting the modal until after the API call (<https://api.stripe.com/v1.tokens>) successfully posts with a 400 status (rather than a 402).

d. What are the steps you would take to report the issue?

Ideally, this would be something that I could sit and talk with a Developer about.  I could walk them through the steps to replicate the issue and then discuss the user experience in this type of situation.  I would also document it this way.

Title: Unsuccessful Gift Purchase Still Prompts Success Modal to Pop Up

Replication:

1. Home page > gifts > fill all other fields > enter CC number ‘4111 1111 1111 1111’
2. Inspect the page and open your Network tab
3. Click Purchase

\*Notice the modal with the success message\*

\*Notice the tokens entry in red on the network tab.  If you look at the response then

you will see that the card was declined because it is a known test card.\*

Expected Resolution:

If possible, we need to prevent the success modal from prompting until after the API

call (<https://api.stripe.com/v1.tokens>) successfully posts with

a 400 status (rather than a 402).

e. What priority would you give this bug (Scale of 1-5, 1 being highest) and why?

I would rate this a 1.  Financials are not actually being affected.  This is really just generating a poor user experience.  That being said, this would only occur if someone was attempting to use a known test credit card number.  Because of that, I’d rate this low in severity and low in priority.

3. What are the possible reasons for the following defect? How would you go about

debugging the problem and gathering more information?

On a web application, a user adds a phone number to their account. The user then changes the phone number. Upon trying to re-enter the first phone number, the user is allowed to click Save, and it seems to work, but the saved number remains the second number rather than updating to the more recently entered number. A page refresh does not change the result.

1. This could be related to a caching issue, where the initial phone number was stored in a cookie and then when the user attempts to save a new phone number, the cookie uses the cached phone number instead.  You could test this by manually deleting cached data on the page and then attempting the save. The solution in this situation would be to implement a cache buster that removes the stored phone number from the cookie when the user is editing the field.
2. This could be related to an endpoint being down.  That means that the server the API call is originating from is not properly processing the change to the phone number field.  You could test this by inspecting the page and looking at the Network tab to see the status of the API call. If it is resulting in a 404 (or a 500) then it could mean that the server is not responding.  The fix in this situation would be to restart the server and verify that the endpoints are running.
3. If the phone number save is controlled by a microservice then it could be related to an issue with the microservice.  You could test this by verifying that the micro service is running. If it is then you can check the microservice database to see which number is being stored.  The fix in the case that the microservice was not running would be to restart it.
4. If the phone number also an area code field then it could be a database issue (coupled with bad validation).  I have seen cases where area code was added to the phone field on the front end but the database was limited to only seven characters.  That, along with the fact that there was only UI validation but no server side validation, resulted in a change appearing to be made from the user’s perspective but an error when the new phone number (that was 9 characters) was being added to the database.  The fix in this situation was to update the database to handle area code (either by adding a new column to the table or by upping the character limit on the existing phone field) and by adding server side validation to the page.

SQL

Using, the following schema, answer the questions below:

create schema interview;

Create table interview.stores ( id integer, name text, allowed\_alcohol boolean ); Insert into interview.stores values (1,'Gettar',true); Insert into interview.stores values (2,'Waysafe',false);

Create table interview.products (

id integer, name text, upc text, created\_at date ); Insert into interview.products values (1,'Apple', '123', '2018-01-01');

Insert into interview.products values (2,'Banana', '456', '2018-01-02'); Insert into interview.products values (3,'Grapes', '789', '2018-01-03'); Insert into interview.products values (4,'Golden Banana', '456', '2018-02-04'); Insert into interview.products values (5,'Bouquet Flowers', '9213123', '2018-02-05');

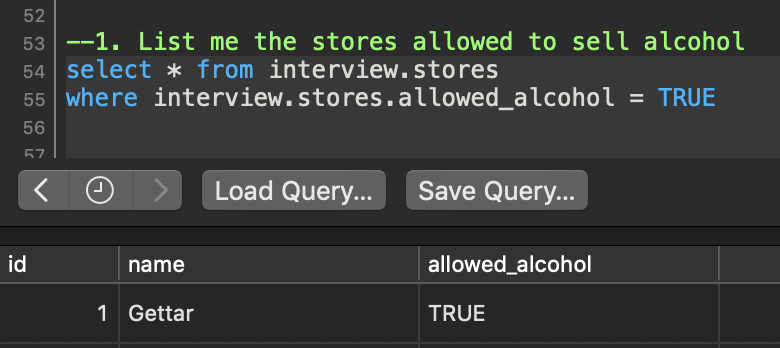
create table interview.store\_prices (

id integer, product\_id integer, store\_id integer, price numeric ); Insert into interview.store\_prices values (1,3,1,2.59); Insert into interview.store\_prices values (2,2,1,3.32); Insert into interview.store\_prices values (3,4,1,3.59); Insert into interview.store\_prices values (4,3,2,2.34); Insert into interview.store\_prices values (5,1,2,1.56);

create table interview.order\_lines (

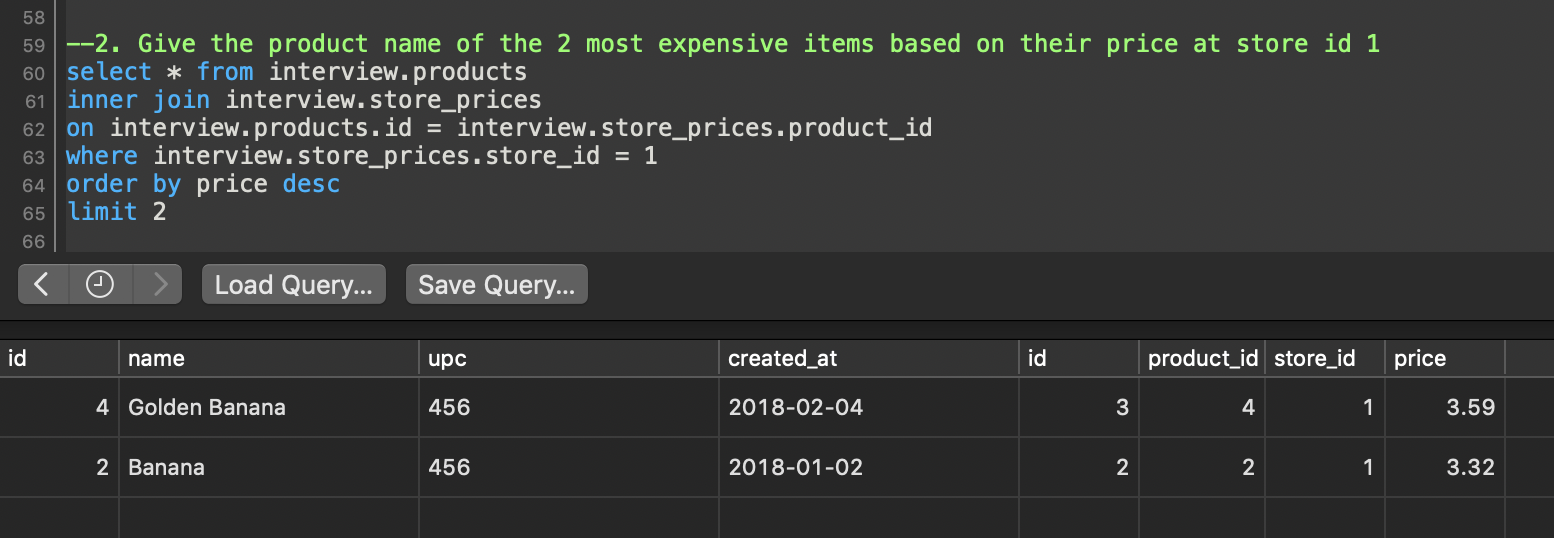
id integer, product\_id integer, store\_id integer, qty integer, line\_total numeric ); Insert into interview.order\_lines values (1,1,2,3,NULL); Insert into interview.order\_lines values (2,2,1,50,NULL); Insert into interview.order\_lines values (3,2,1,1,NULL); Insert into interview.order\_lines values (4,3,2,4,NULL);

1. List me the stores allowed to sell alcohol



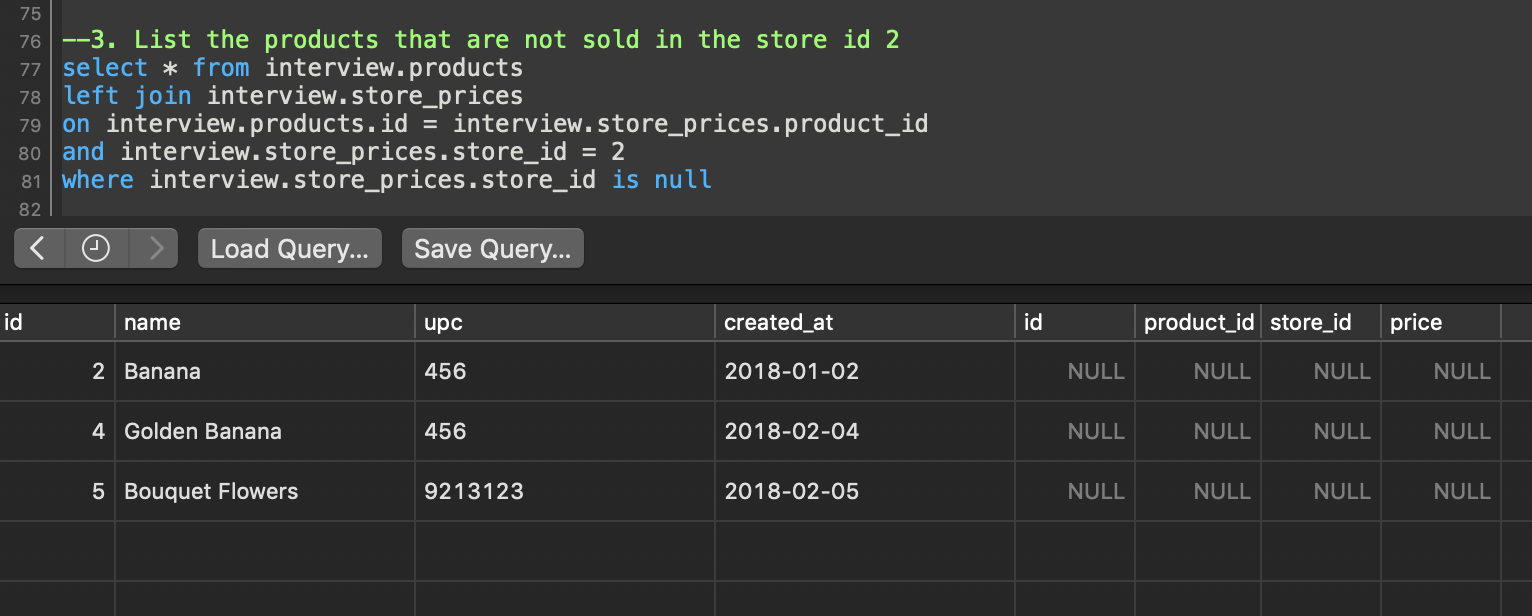
Gettar

2. Give the product name of the 2 most expensive items based on their price at store id 1



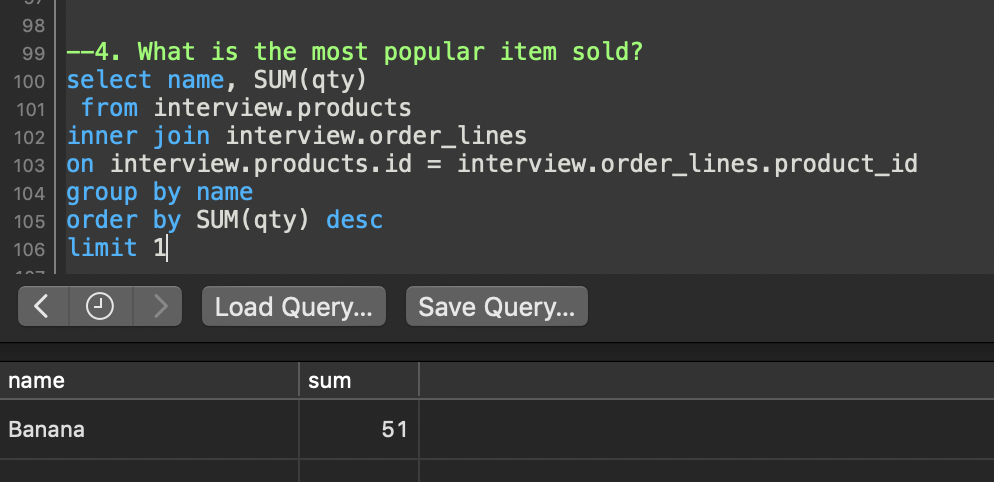
Golden Banana and Banana

3. List the products that are not sold in the store id 2



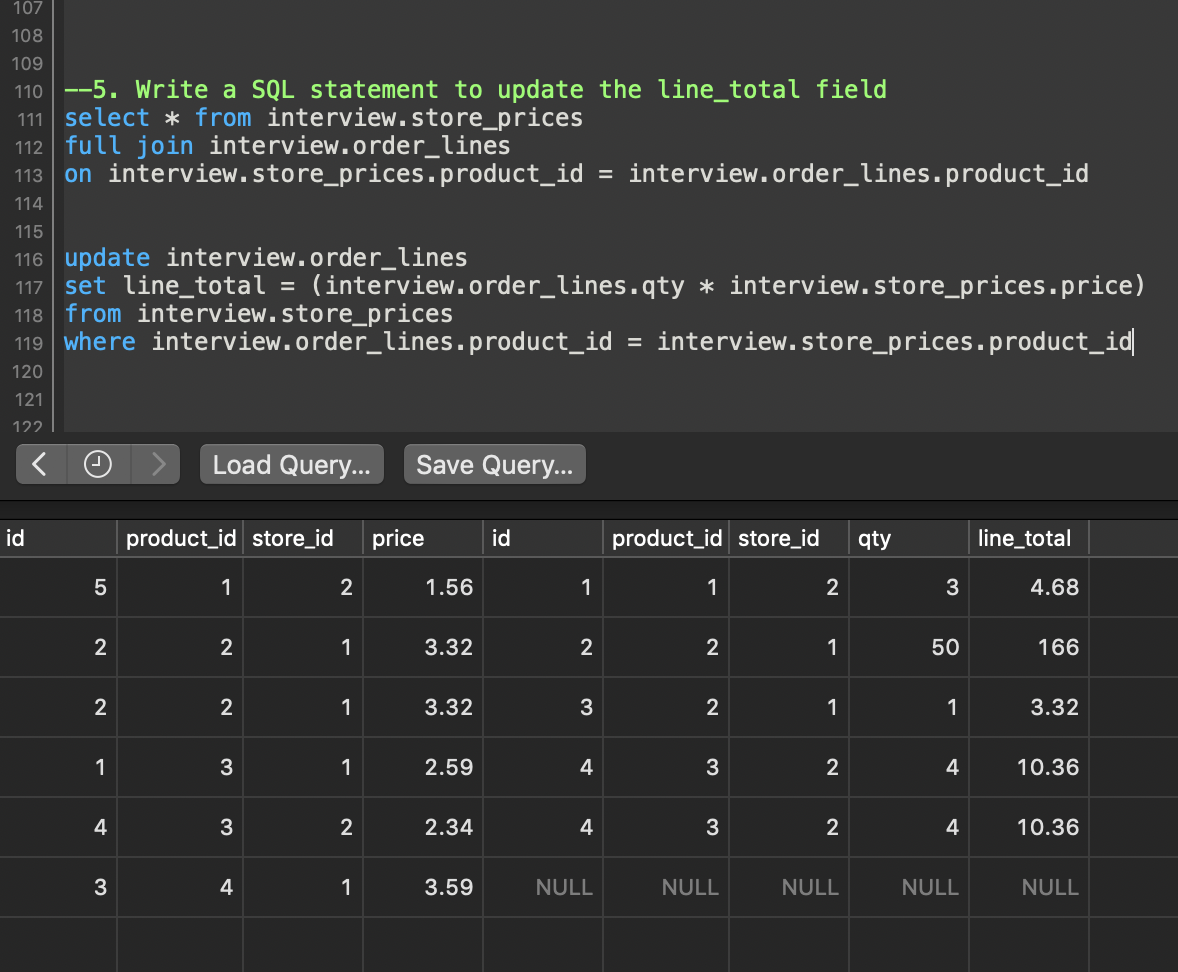
Bouquet Flowers, Banana, and Golden Banana

4. What is the most popular item sold?



Bananas

5. Write a SQL statement to update the line\_total field



Automation: github link for automation: <https://github.com/natewilliams1/homework>

Write an automated test which accomplishes the following

1. Access www.shipt.com in the browser

2. Using the feature you chose for step 1, write an automated test 3. The test should validate the feature works as expected

We recommend you choose from the following languages/tools. Feel free to use other tools if you feel they lend better to this type of testing.

● Selenium

● Calabash

● Typescript/Javascript

● Ruby

Automation Assessment

1. If you chose to use a tool or language other than the recommended, briefly explain why.

2. How did you go about locating the elements for your tests?

In my Ruby spec I utilized the page-object gem and tried to use mainly href’s and id’s.  I chose these because they are less likely to change over time. In my Cypress spec I utilized the data-test tags.  The data-test tags show a lot of foresight from a developmental perspective.

3. What do you believe are the most common causes for instability in UI automation?

I would say that the most common reasons I’ve run into with UI instability are:

1. Runtime.  UI tests typically take a long time to run because they are accessing the UI and waiting for pages to load.  This isn’t a huge deal if you have you have a small regression suite. However, if your regression suite is large then it can take a very long time to run. I currently work on a more legacy product that has a regression suite of about 1400 separate tests that run nightly through Jenkins.  Each full run takes around 4 hours to complete. That is while no one else is really hitting the server. During the day, when builds are happening and we are actively testing in those environments, the run times tend to be longer. That makes it time consuming to identify actual issues from the tests.
2. Load Times for Elements/Pages Not Fully Loading.  A lot of the flakiness I’ve seen with UI Automation specs comes from the time it takes for elements to load on a page as it corresponds with when the spec is expecting those elements to be present.  This makes it difficult to parse through failing tests to determine when things are legitimately broken.
3. Browser/Server/Tool Dependencies.  There are multiple places where services can fail with UI tests.  If a server goes down, if there is a slight change to server configurations, if the browser crashes, etc.  These things can all affect UI Automation tests, causing them to fail when there isn’t actually a problem with the UI.
4. UI Code Changes.  If (and when) the UI for a page is updated, it can affect the UI tests that have been running against that page.  This usually causes a lot of the tests to break if the specs themselves are using bad selectors/elements. For instance, if I use text to identify a button and then that text is altered, the spec I was using is now going to break when it tries to identify that button.  This means that I have to spend time digging through my failures to identify which ones are legit and which ones are being caused by flaky specs.

4. How do you make your tests consistent and easy to debug?

I use DRY principles (don’t repeat yourself) to avoid lots of unnecessary work.  I try to create methods in my page objects and then call those methods in my specs.  That keeps the spec clean, especially if the feature itself is complex and requires multiple steps to test.  This needs to be done within reason, though. If a feature is fairly simple and it makes more sense to include the actions in the spec itself then I say go for it.  There is no sense creating a ton of methods that will only be used once in a single spec.

Another great principle I’ve picked up on is to leave things as you find them.  That simply means avoiding static data or worse, creating new data for each test and not deleting it at the end of each test.  At my current job, we use a structure that utilizes a lot of API calls and a datafactory gem that essentially generates and deletes data straight from our database.  I include generation calls in the before (:all) and removal calls in the after (:all). In cases where we don’t have access to those calls, I make sure that each test starts in a neutral state.  That avoids the pitfall of contingencies when trying to test a specific feature. (ex. If I am testing the account profile editor, I don’t want to test the login feature first. I’d rather have the user generated, the login credentials passed through, and the test start directly on the account profile page).

Lastly, I try to keep my tests consistent in structure from a setup level (lib/page folders contain the actual page being tested, with one page.rb file for an actual application page and the spec folder contains the automated checks).  I also try to keep things simple and readable. In my experience, that goes a long way in understanding what is actually happening with each spec.