Here is the project….Let me know if you have questions.

Test Problem:

From the following two questions (one for frontend, the other for backend), choose ONE that you feel more comfortable with, and automate the scenarios as described:

1. Automate an end-to-end user e-commerce transaction flow using any open source tool for [www.walmart.com](http://mobile.walmart.com/" \t "_blank) with an existing customer on Chrome or Safari browser.

2. Automate an end to end user e-commerce transaction flow for backend Web services API layer utilized by mobile app client:

iOS: <https://itunes.apple.com/us/app/walmart-savings-catcher-shopping/id338137227>

Android: <https://play.google.com/store/apps/details?id=com.walmart.android&hl=en>

You should be able to capture the API calls made by the mobile client by using a proxy tool like Charles Proxy ([http://www.charlesproxy.com](http://www.charlesproxy.com" \t "_blank)). For capturing HTTPS calls made by client you will have enable SSL proxying within Charles by adding SSL certs (http://www.charlesproxy.com/documentation/using-charles/ssl-certificates/).

------------------------------------------------------------------------------------------------------------------------  
  
Scenario to automate:

1. Login using existing account

2. Perform a search on home page from a pool of key words given below  
3. Identify an item from the result set that you can add to cart  
4. Add the item to cart

5. Validate that item added is present in the cart and is the only item in the cart  
  
Test Data:  
• Account details: create your own account  
• Search terms: tv, socks, dvd, toys, iPhone  
  
Testing tools and Programming language to be utilized:

Any open source testing tool such as Webdriver, Watir etc. would work. Programming language for the test can be of your choice.

**Attributes of a good test:**

• Readability and clarity of intent  
• Re-usability of code to implement more tests  
• Eliminate duplication  
• Robustness in test execution

* Put the code in github and please just send back the github link.
* Make sure you have a clear README file:
  + Reasoning behind your technical choices. Trade-offs you might have made, anything you left out, or what you might do differently if you were to spend additional time on the project.
* We’ll review your code based on:
  + First of all, you solution needs to run smoothly following the instructions in README
  + Clarity: does the README clearly explain the problem and solution?
  + Correctness: does the application do what was asked? If there is anything missing, does the README explain why it is missing?
  + Code quality: is the code simple, easy to understand, and maintainable?

Good luck