My Understanding of Given Assignments:

Assignment # 1: This test case deals with performing a functional testing. The requirement is to retrieve quote for different security ticker and verify that the all the fields are returned by the GetQuery quote. That is, the fields "Previous Close", "Open", "Daily Range", "52wk Range" etc. are returned by the query.

This functional test case can be implemented using 3 different ways.

1. Using Selenium IDE Tool for Firefox: With one click, this tool can record the steps and these recorded steps can be played back at will. It is simple and easy to use tool but it has its drawbacks. This IDE tool works only for Firefox. It is good for very simple test cases but falls short for complicated test cases. It does not work well with opening child windows, dealing with Pop ups and data driven testing.

2. Using Selenium RC: Using a client library, test scripts can be developed using java, python, Ruby, Perl and so forth. It communicates with selenium Remote Control (RC) which in turns sends command to browser to test the application. It is a very good tool and it can handle much more complicated test cases as compare to Selenium IDE. Selenium RC requires to run Selenium Remote Control server and architecture of Selenium RC is bit complicated. This tool is being replaced by Selenium Webdriver which is much faster than Selenium RC. Industry trend is to move towards Selenium Webdriver.

3. Selenium Webdriver: This is a latest tool in Selenium suite. It works across different browsers. It allows to use a programming language of choice. It is faster than Selenium RC. Its architecture is simple compare to RC. It cannot readily support new browser but Selenium RC can.

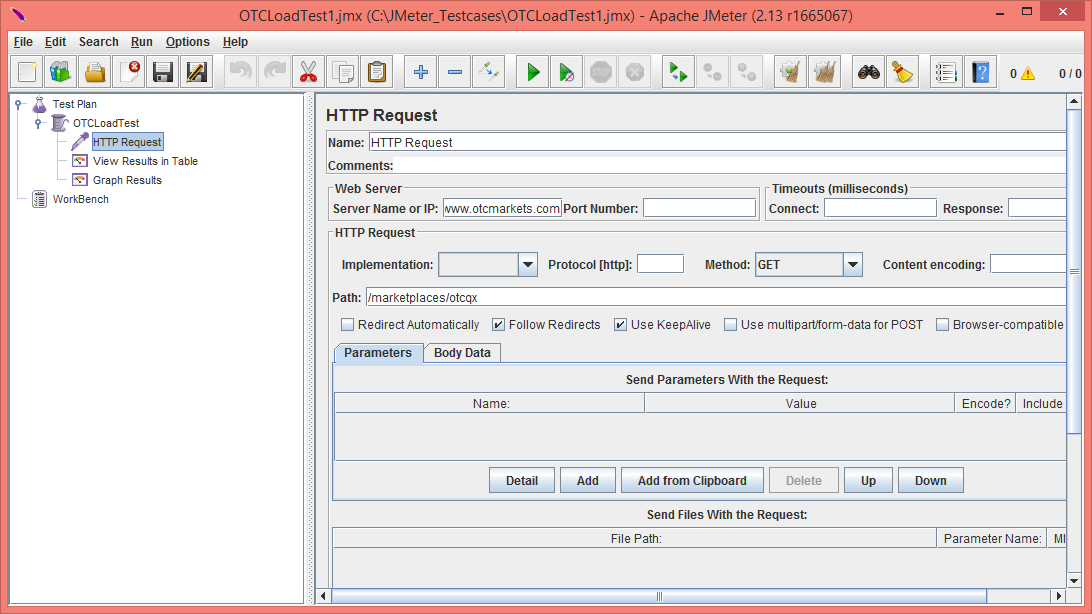
For the implementation of this test case, I have chosen to use the Selenium Webdriver and used Java language to develop the automation code.

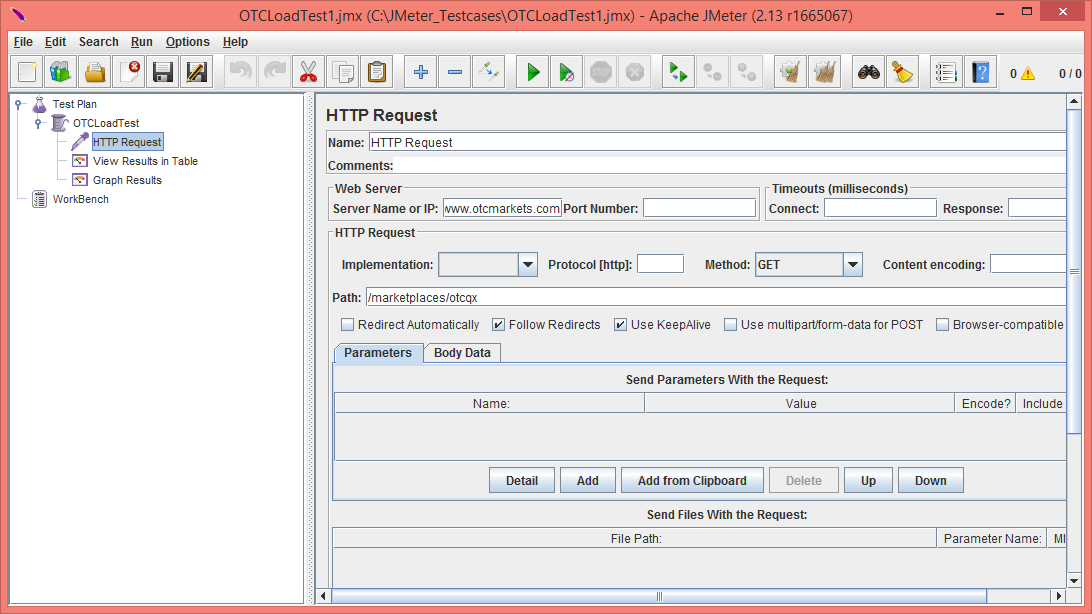
Test Flow:

1. Invoke appropriate browser.
2. Read ticker data from excel file
3. Wait for webpage to load
4. If SurveyMonkey pops up displayed then close it.
5. Retrieve data for each ticker
6. Close browser

Assignment # 2: This test case deals with load testing. The requirement is to simulate 10, 50 or 100 concurrent user accessing the given URL. This load test case ca also be implemented in 3 different ways.

1. Download and install JMeter tool. Create the thread group and http request as shown below to simulate the concurrent users.





1. Install Junit plug-in for JMeter. Develop code in Java with combination of Selenium and Junit. Generate runnable jar file and install this jar file into JMeter and configure JMeter to simulate number of users. The Junit load jar file is include in the package but good amount of configuration requires in JMeter to make it work.
2. Develop Java code combine with Selenium Webdriver and TestNG. TestNG frame work has capability to simulate the load test. For this option, you don’t need JMeter tool. It can be run from the windows command prompt. In this package, runnable jar file is include to run from the command prompt.

I have developed Java code in combination with Webdriver and TestNG frame work. The final runnable jar file is included in the package and can be invoked from the command prompt.