

## Stori QA Automation Engineer Challenge

Thanks for your interest in joining Stori. At this stage, we ask you to solve the following challenge.  
Best of luck!

For this challenge you should use **Python**. Also you can create “pure” code or use an automation framework like **Selenium, Robot, etc.**

### Test Cases

All of the following test cases must be modeled in BDD using **Gherkin** with different scenarios, following the standard BDD structure.

### Tests Execution

Make a parameterized script to run your test script in a specific browser in a mobile device or mobile emulator Android or IOS.

The script should accept the argument `--browser THE_NAME_OF_THE_BROWSER`. You should use Chrome, Firefox and Opera for the browser. For example `test_stori_challenge.py --browser firefox`

### Tests Components

The test script should execute the following tasks:

1. Open the desired browser in the mobile device or mobile emulator and go to the following page:  
<https://rahulshettyacademy.com/AutomationPractice/>
2. In the Suggestion Class Example, enter the word “**Me**” and select Mexico then enter the word “**Uni**” and select United States (USA) then write again “**Uni**” and select United Arab Emirates.  
(Bonus: use only xpath and select other country you like)
3. In the Dropdown Example, select option 2 and then option 3. The user should be able to see the change.  
(Bonus: use only xpath without using ids, text or values)
4. In the Switch Window Example, click the Open Window button. If the 30 day money back guarantee text (example below) is not shown, fail the test and close the opened window.  
(Bonus: Review the text of SELF PACED ONLINE TRAINING, IN DEPTH MATERIAL, LIFETIME INSTRUCTOR SUPPORT and RESUME PREPARATION and find the bugs here and document them in the RTM)

## Stori QA Automation Engineer Challenge



### 30 DAY MONEY BACK GUARANTEE

We would never want you to be unhappy! If you are unsatisfied with your purchase, contact us in the first 30 days and we will give you a full refund.

5. In the Switch Tab Example, click the Open Tab button. Scroll on the new tab until you see the button below. Then take a screenshot that includes the button and save it with the name of the test case that you gave in the Gherkin file. Don't close the window and return to the first window.  
(Bonus: use xpath and css selector)

[VIEW ALL COURSES](#)

6. In the Switch To Alert Example, type this string “**Stori Card**” and click the Alert button. Print the text in the alert and then click on OK. Then type the same string and click on the Confirm button and print the text. Make sure that the string shown in the alert equals this “**Hello Stori Card, Are you sure you want to confirm?**” then click on OK.  
(Bonus: use xpath and css selector)
7. In the Web Table Example, print how many courses cost \$25 and then print their course names. Then print all the same things for the courses that cost \$15.  
(Bonus: use only the css selector child to parent)
8. Print the names of all the **Engineers** in the Web Table Fixed header. Then print all the names of all the **Businessmans**.  
(Bonus: use xpath and css selector child to parent)
9. In the iFrame example, get the text highlighted in blue in the following image and print it.  
(Bonus: use xpath and print only the odd indexes)

## why we are **Leaders in market?!**

- The academy is led by Rahul Shetty, a Test evangelist and Test architect with over 10+ years in the IT industry.
- Rahul Shetty is a pioneer and authority in software testing space.
- He has taught over 300,000 students in 195 countries from across the world.
- Wide coverage of topics like Selenium, Appium, Cucumber BDD, Cypress, Protractor, RestAPI, SoapUI and JMeter etc.
- His mentorship program is most after in the software testing community with long waiting period.

10. Make an HTML report with all the test cases statuses and save it in a folder called “Reports”.  
(Bonus: Make an XML, report too)

### **Delivery and code requirements**

Your project must meet these requirements:

1. Add A README in the root of your code with clear instructions on how to compile and run your code.
2. All steps should be completed programmatically and well documented in the Gherkin file.
3. Only code files and the reports should be included in your deliverable. Please don't include any screenshots you took while developing your code. Please use the gitignore file.
4. Upload your code to your github in a repo called “Stori-QA-Automation-Challenge” and make it a private repo and add the QA engineer that will review your challenge
5. Submit your code using [this form](#)