**QA Automation Coding Test (2.0 hrs)**

***For both the tests below, upload the test project(s) on GitHub for review.***

***Also, ensure the project can be run during the next interview session. Questions will be based on the implementation of the Test Solution(s).***

1. ***The below test applies to both API and UI Test Automation Development positions***

Develop a test method in Selenium to browse from [www.labcorp.com](http://www.labcorp.com) through the Careers link to a specific LabCorp job listing confirming information on the job posting. Please be sure to accomplish the following tasks:

1. Start the test by opening a browser to [www.labcorp.com](http://www.labcorp.com)
2. Find and click Careers link
3. Search for any position (that is currently active on the site) example “QA Test Automation Developer”
4. Select and browse to the position
5. Add assertions to confirm
   1. Job Title
   2. Job Location
   3. Job Id
   4. Any 3 other assertions of your choice (could be text in the requirements / introduction)

Example: (Text in quotes “” from below example should be updated by candidate to match that in the position chosen for the automation solution):

* Confirm first sentence of third paragraph under Description/Introduction as “The right candidate for this role will participate in the test automation technology development and best practice models.”
* Confirm second bullet point under Management Support as “Prepare test plans, budgets, and schedules.”
* Confirm third requirement as “5+ years of experience in QA automation development and scripting.”
* Confirm first suggested automation tool to be familiar with contains “Selenium”

1. Click Apply Now and confirm Job Title, Job Location, Job ID and another text of choice to match the previous page in the proceeding page.
2. Click to Return to Job Search

Be sure to include the proper wait methods, at least three different “By” types when searching elements, and that it is tested in Chrome. For the

For the UI Test Automation Developer positions

* The project should be written in C# Selenium
* The structure of the test should be written in BDD format as well.

**Gherkin:**

Scenario: Validate job description on careers page

Given I open google chrome

When I open [www.labcorp.com](http://www.labcorp.com)

And I wait 3 seconds

When I search Careers

Then search results for Careers should appear

When I search term as QA Test Automation Developer

And I wait 3 seconds

Then search results for QA Test Automation Developer should appear

Then I validate “The right candidate for this role will participate in the test automation technology development and best practice models.”

Then I validate second bullet point under Management Support as “Prepare test plans, budgets, and schedules.”

Then I validate third requirement as “5+ years of experience in QA automation development and scripting.”

**Script file:**

from selenium import webdriver

#chrome driver

driver = webdriver.Chrome()

#open webpage

driver.get("https:// [www.labcorp.com](http://www.labcorp.com)")

sleep(3)

careerPage = driver.find\_element\_by\_link\_text("Careers").click()

sleep(3)

#assert text on page

element1 = driver.find\_element\_by\_tag\_name('h1')

assert element1.text == ' The right candidate for this role will participate in the test automation technology development and best practice models.’

Element2 = driver.find\_element\_by\_tag\_name('h2')

assert element2.text == ' Prepare test plans, budgets, and schedules.’

Element3 = driver.find\_element\_by\_tag\_name('h3')

assert element3.text == ‘5+ years of experience in QA automation development and scripting.’

Driver.close()

For the API Test Automation Developer position

* The project should be written in Java with BDD implementation
* Attempt the Test on Rest API Test Automation in point II below.

1. ***The below test applies to ONLY API Test Automation Development position***

Automate the below sample API using REST Assured for both POST and GET calls.

For POST use this sample json: <https://6143a99bc5b553001717d06a.mockapi.io/testapi/v1//Users>

{  
        "createdAt": 1631825833,  
        "employee\_firstname": "TestData12345",  
        "employee\_lastname": "TestData12345",  
        "employee\_phonenumbe": "264-783-9453",  
        "ademployee\_emaildress": "ademployee\_emaildress 1",  
        "citemployee\_addressy": "citemployee\_addressy 1",  
        "stateemployee\_dev\_level": "stateemployee\_dev\_level 1",  
        "employee\_gender": "employee\_gender 1",  
        "employee\_hire\_date": "2025-10-31T16:35:45.426Z",  
        "employee\_onleave": true,  
        "tech\_stack": [],  
        "project": []  
    }  
  
After it got posted, get the information calling get service, and verify the record is retried successfully with all the details.

<https://6143a99bc5b553001717d06a.mockapi.io/testapi/v1//Users>

Attempt at least 3 if not more scenarios for above services.

Note: Make sure you have the same json object when sending the request.

**Gherkin:**

Scenario: Validate job description on careers page

Given: user calls post API to load employee data with request <json>

When I make the post call

Then I validate a 201 response code is returned by the API

Then I validate the database fields with JSON request

When I make the second get call to get data back from server

Then I validate a 200 response code is returned by API

Then I validate response matches input json request

Example:

|request|

|employee.json|

|Invalidemployee.json|

**Script file:**

import requests

import json

postURL = “<https://6143a99bc5b553001717d06a.mockapi.io/testapi/v1//Users>”

# read input file

file = open (file location)

json\_input = json.loads(json\_input)

request\_json = json.loads(json\_input)

#make post call

response = requests.post(postURL, request\_json)

#validate response code

assert response.status\_code == 201

#getURL = “<https://6143a99bc5b553001717d06a.mockapi.io/testapi/v1//Users>”

#send get request

getResponse = requests.get(getURL)

#validate response

assert getResponse.content ==employee.json