

QA Engineer: Jorge Rivas

Date: November 10/2025

QA Engineer Assessment

Section 1 – Manual QA

Test Case Design

Scenario: Your company's web app allows users to register and log in using email and password. After login, they can view a dashboard showing their profile information and a list of recent training sessions. Password must be at least 8 characters, include one number, and one special character.

Task: Write three detailed test cases that cover:

- Positive and negative scenarios
- Field validation
- Edge cases

Deliverable: Submit test cases in a table (Test ID, Description, Test Data, Steps, Expected Result).

Test cases design:

Test ID	Description	Test Data	Preconditions	Steps	Expected Result
TC01	<p>Positive Scenario Verify that a user can register and login successfully with a valid email and password</p>	Email: xyz@hotmail.com Password: Testing10\$ (Password > 8 characters, includes 1 number and 1 special character)	- User is new (not previously registered) - Password: must be at least 8 characters, including one number, and one special character.	1. Navigate to the registration page 2. Fill out all required fields with valid data 3. Ensure mandatory fields are requested and the password meets the required format 4. Click on the Register button 5. Login with the new user created	1. The registration page loads correctly 2. Information is added properly in the registration form 3. Mandatory fields are validated and the password is accepted as valid 4. Registration completed successfully 5. Login successfully and the dashboard displays the profile information and recent training sessions
TC02	<p>Negative Scenario Verify the user cannot register with invalid email and password</p>	Email: invalid@hotmail Password: invalidPsw (Invalid: no special character, no number)	- User is new (not previously registered)	1. Navigate to the registration page 2. Fill in invalid email and password 3. Click on the Register button	1. The registration page loads correctly 2. System validates mandatory fields 3. The user cannot register and the error message is displayed "Password must be at least 8 characters, including one number, and one special character."
TC03	<p>Edge Scenario Verify registration works with a password exactly 8 characters long that meets all requirements</p>	Email: xyz@hotmail.com Password: Testing1\$ (Exactly 8 characters, valid format)	- User is new (not previously registered) - Password policy applies	1. Navigate to the registration page 2. Fill out all required fields with valid data 3. Ensure mandatory fields are requested and enter a password that is exactly 8 characters long and meets all requirements 4. Click on the Register button 5. Login with the new user created	1. The registration page loads correctly 2. Information is added properly in the registration form 3. Mandatory fields are validated and the password is accepted as valid 4. Registration completed successfully 5. Login successfully and the dashboard displays the profile information and recent training sessions

Section 2 – Automation QA

Part A: API Testing

Scenario: Use a public API, such as <https://reqres.in>, to create automated test.

Deliverable: Three API test cases using Postman, or a tool you're familiar with. Include validations for status codes and fields.

Test cases design:

Test ID	Description	Method/Endpoint	TestData	Validations	Expected Result
API01	Verify that user details are retrieved correctly for all users	GET: https://reqres.in/api/users?page=2	Headers: key: x-api-key value: reqres-free-v1	- Status code = 200 - Response body includes: id, email, first_name, last_name, avatar for each user	- Api returns a list of all users with correct structure and status code 200
API02	Verify that user details are retrieved correctly for a single user	GET: https://reqres.in/api/users/2	userid: 2 Headers: key: x-api-key value: reqres-free-v1	- Status code = 200 - Response contains correct user data for id = 2	- status code = 200 - Request retrieves information only for the user with id = 2
API03	Verify that a new user can be created successfully	POST: https://reqres.in/api/users	Body: { "email": "eve.holt@reqres.in", "password": "pistol" } Headers: key: x-api-key value: reqres-free-v1	- status code = 201 - Response body includes: id, email, password, createdAt, avatar for each user	- status code = 200 - User is successfully created, and API returns a unique id and timestamp (createdAt)

Part B: Automation Script

Scenario: Using a website you're familiar with, write a Selenium script, or use a tool you're familiar with, to test a capability (such as logging in) and report success or failure.

Deliverable: Submit your test with comments explaining it.

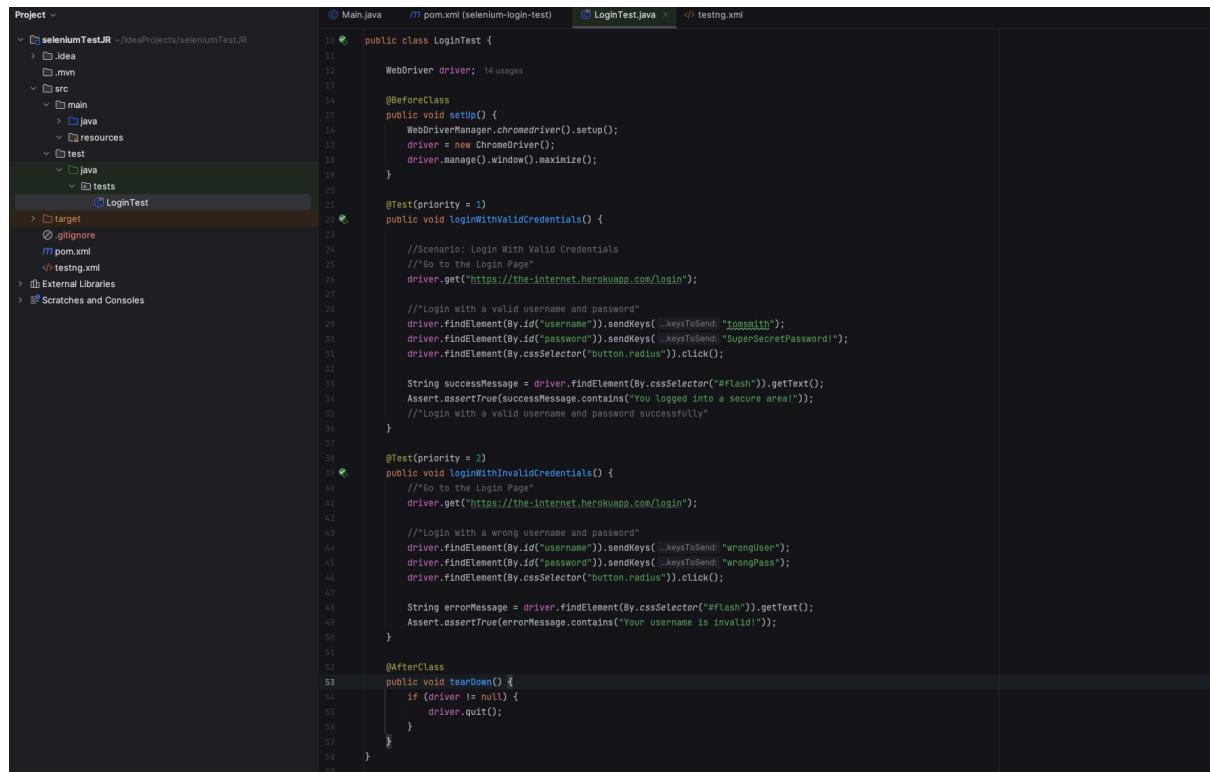
Submission Instructions

- Submit files and documentation to a GitHub repo.
- Include a short video explaining your tests.

Automated test

Positive Test: [loginWithValidCredentials](#)

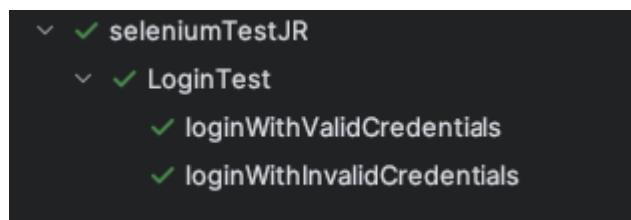
Negative Test: [loginWithInvalidCredentials](#)



The screenshot shows an IDE interface with a project tree on the left and a code editor on the right. The project tree for 'seleniumTestJR' includes '.idea', '.mvn', 'src' (containing 'main' and 'test' packages), 'target', '.gitignore', 'pom.xml', 'testing.xml', and 'External Libraries'. The code editor displays 'LoginTest.java' with the following content:

```
1  package com.example;
2
3  import org.openqa.selenium.By;
4  import org.openqa.selenium.WebDriver;
5  import org.openqa.selenium.WebElement;
6  import org.openqa.selenium.chrome.ChromeDriver;
7  import org.openqa.selenium.interactions.Actions;
8  import org.openqa.selenium.support.ui.ExpectedConditions;
9  import org.openqa.selenium.support.ui.WebDriverWait;
10 import java.util.List;
11
12 public class LoginTest {
13
14     WebDriver driver;
15
16     @BeforeClass
17     public void setup() {
18         WebDriverManager.chromedriver().setup();
19         driver = new ChromeDriver();
20         driver.manage().window().maximize();
21     }
22
23     @Test(priority = 1)
24     public void loginWithValidCredentials() {
25
26         //Scenario: Login With Valid Credentials
27         //Go to the Login Page
28         driver.get("https://the-internet.herokuapp.com/login");
29
30         WebElement usernameField = driver.findElement(By.id("username"));
31         WebElement passwordField = driver.findElement(By.id("password"));
32         WebElement loginButton = driver.findElement(By.cssSelector("button.radius"));
33
34         Actions actions = new Actions(driver);
35         actions.sendKeys(usernameField, "tomsmith");
36         actions.sendKeys(passwordField, "SuperSecretPassword!");
37         loginButton.click();
38
39         String successMessage = driver.findElement(By.cssSelector("#flash")).getText();
40         Assert.assertTrue(successMessage.contains("You logged into a secure area!"));
41         //Login with a valid username and password successfully
42     }
43
44     @Test(priority = 2)
45     public void loginWithInvalidCredentials() {
46
47         //Go to the Login Page
48         driver.get("https://the-internet.herokuapp.com/login");
49
50         WebElement usernameField = driver.findElement(By.id("username"));
51         WebElement passwordField = driver.findElement(By.id("password"));
52         WebElement loginButton = driver.findElement(By.cssSelector("button.radius"));
53
54         Actions actions = new Actions(driver);
55         actions.sendKeys(usernameField, "wrongUser");
56         actions.sendKeys(passwordField, "wrongPass");
57         loginButton.click();
58
59         String errorMessage = driver.findElement(By.cssSelector("#flash")).getText();
60         Assert.assertTrue(errorMessage.contains("Your username is invalid!"));
61     }
62
63     @AfterClass
64     public void tearDown() {
65         if (driver != null) {
66             driver.quit();
67         }
68     }
69 }
```

Result



The screenshot shows a terminal window displaying the execution results of the Selenium test. It shows two successful test cases: 'loginWithValidCredentials' and 'loginWithInvalidCredentials' under the 'LoginTest' class, which is itself under the 'seleniumTestJR' project.

```
1  ✓ seleniumTestJR
2    ✓ LoginTest
3      ✓ loginWithValidCredentials
4      ✓ loginWithInvalidCredentials
```