SQA Automation Engineer Assessment

Objective: The objective of this assessment is to evaluate the candidate's ability to automate different web scenarios using Selenium WebDriver and their knowledge of programming, test framework implementation, and problem-solving skills.

Instructions:

- Use any programming language of your choice (Java, C#, Python).
- Write clean, modular, and well-documented code.
- Implement best practices for automation, including handling dynamic elements and using explicit waits where necessary.
- Use appropriate test frameworks (TestNG, JUnit, SpecFlow, etc.).
- Ensure that your code runs successfully and achieves the expected results.
- Bonus points for implementing CI/CD integration strategies.

Automation Tasks

Task 1: Login Automation

URL: https://the-internet.herokuapp.com/login
Requirements:

1. Automate login functionality with valid and invalid credentials.
2. Verify if login is successful by checking the presence of a logout button or page title.
3. Capture a screenshot if login fails.
4. Implement exception handling and error reporting.
5. Implement **Page Object Model (POM)** for the login page.

Task 2: Dynamic Table Handling

URL: https://the-internet.herokuapp.com/tables
Requirements:

1. Extract and print all company names from the table.
2. Verify if a specific company (e.g., "Jason Doe") exists in the table.
3. Implement a generic function to handle data extraction for any table.

Task 3: JavaScript Alerts & Pop-ups

URL: https://the-internet.herokuapp.com/javascript_alerts
Requirements:

1. Click on each alert type (JS Alert, JS Confirm, JS Prompt) and automate handling them.
2. Validate alert messages and ensure correct handling for each type.
3. Implement a generic method to handle different types of JavaScript pop-ups.

Task 4: File Upload Automation

URL: https://the-internet.herokuapp.com/upload
Requirements:

1. Automate file upload using Selenium WebDriver.
2. Verify that the uploaded file name appears after a successful upload.
3. Handle file upload without using sendKeys() (e.g., using Robot class in Java).

Submission Guidelines:

- Submit your source code in a ZIP file or as a GitHub repository link.
- Include a README file with setup instructions and any dependencies required to execute the test scripts.
- Ensure your scripts run successfully on a standard Chrome or Firefox browser.

Evaluation Criteria:

- ✔ Code Quality & Readability
- ✔ Correctness & Functionality
- ✔ Exception Handling & Error Reporting
- ✔ Use of Best Practices (e.g., POM, Wait Strategies, Logging)
- ✔ Performance & Efficiency
- ✔ CI/CD & Test Framework Knowledge

Good Luck! 🚀