TEST AUTOMATION CHALLENGE:

Objective:

Evaluate the skills and approach of the QA Automation Engineer candidate in creating a comprehensive test strategy, developing a based automation framework, and implementing both UI and API tests for the given web application.

Instructions:

1. Test Strategy:

- *Visit https://www.saucedemo.com/ and identify at least 2 features to create a test strategy.*
- *Develop a test plan outlining the test scenarios, test data, test environment, and any assumptions or constraints.*
- *Explain why you chose the features for testing, demonstrating your understanding of risk evaluation.*

Here is base matrix with the main components:

Feature	Probability (1-5)	Impact (1-5)	Risk Level
Login (1)			
- Successful login	1	2	2
- Account lockout	3	4	12
- Problem reporting	3	4	12
Burger Menu (2)			
- Redirect to inv.	3	4	12
- Redirect to web	4	3	12
- Logout	5	5	25

- Reset app state	3	4	12
Inventory (3)			
- View products	1	2	2
- Add to cart	2	3	6
- Remove from cart	2	3	6
- Sort A to Z	1	1	1
- Sort Z to A	1	1	1
- Sort Low to High	1	1	1
- Sort High to Low	1	1	1
- View item details	2	3	6
- Add to cart (det)	2	3	6
- Remove from cart	3	4	12
Cart (4)			
- Display grid	2	3	6
- Remove from cart	2	3	6
- Continue shopping	2	3	6
- Proceed to checkout	1	2	2
Checkout (5)			
- Provide info	2	3	6
- Checkout overview	2	3	6
- Enter payment info	3	4	12
- Display totals	1	2	2
- Finish order	1	2	2
- Cancel checkout	2	3	6
- Return to homepage	1	2	2

2. Automation Framework:

- *Using an automation tool of your preference (e.g., Selenium, Cypress, Puppeteer), create an automation framework that covers the following UI and API tests.*

UI Tests:

- *Verify that an item can be added to the cart and is visible on the cart page.*
- *Ensure the user can complete the purchase/checkout process.*
- *Verify that inventory items can be sorted by price, high-to-low, and the sorting is correct.*
- *Ensure that inventory can be sorted by name, Z-to-A, and the sorting is correct.*

API Tests:

- *A. Create tests for API calls (POST, GET, PUT, DELETE), setting headers, body, and making proper assertions. You can use this Swagger to do some examples of API Testing: https://petstore.swagger.io/*

3. Bonus Task:

- *Implement a function in your automation framework that, given a DOM Element on the page, will visit the element itself and all of its descendants.*
- *For each element, pass that element to a provided callback function.*
- *The arguments to the function should be a DOM element and a callback function (that takes a DOM element as its argument).*

Submission:

- *Push your automation framework code to a public repository (GitHub, GitLab, etc.) for evaluation.*
- *Provide clear documentation on how to set up and run the tests.*

Evaluation Criteria:

- *Test Strategy: Understanding of risk evaluation, clarity in test plan, and rationale for feature selection.*
- *Automation Framework: Proper use of any-based automation tools, completeness of implemented tests, and adherence to best practices.*
- *Bonus Task: Demonstration of advanced JavaScript and DOM manipulation skills.*

Note:

- *Ensure that your repository is well-organized, and the code is commented on for clarity.*
- *Provide any additional instructions or considerations needed for the evaluation of your submission.*