QA Engineer Technical Task - Estefani Bouvier

Part 1: UI Testing

ID	Title	Description	Steps	Test data	Expected result
UI_TC01	Login - Login with valid credentials.	The user should be able to login successfully with valid credentials.	1- Go to: https://www.saucedemo.com/	-	- The page loads successfully. - The username and password input fields are visible, and the Login button is displayed.
			2- Enter valid credentials in the username and password fields, and click the 'Login' button.	Username: standard_user Password: secret_sauce	- The user is redirected to the products page in less than 1,5 second. URL: https://www.saucedemo.com/inventory.html Title: 'Products'. - All product images are loaded correctly. - All UI elements are properly aligned with UX/UI definition.
UI_TC02	Login - Missing credentials error	Login should fail if credentials are missing.	1- Go to: https://www.saucedemo.com/	-	- The page loads successfully. - The username and password input fields are visible, and the Login button is displayed.
			2- Click the 'Login' button.	-	- The page displays an error message: "Epic sadface: Username is required". - The username and password fields are highlighted in red. - A red cross icon appears next to the fields.

UI_TC03	Login - Missing username error	Login should fail if the username is missing.	1- Go to: https://www.saucedemo.com/	-	- The page loads successfully. - The username and password input fields are visible, and the Login button is displayed.
			2- Enter a valid credential in the password field, and click the 'Login' button.	Password: secret_sauce	- The page displays an error message: “Epic sadface: Username is required”. - The username field is highlighted in red. - A red cross icon appears next to the username field.
UI_TC04	Login - Missing password error	Login should fail if the password is missing.	1- Go to: https://www.saucedemo.com/	-	- The page loads successfully. - The username and password input fields are visible, and the Login button is displayed.
			2- Enter a valid credential in the username field, and click the 'Login' button.	Username: standard_user	- The page displays an error message: “Epic sadface: Password is required”. - The password field is highlighted in red. - A red cross icon appears next to the password field.
UI_TC05	Login - Invalid credentials error	Login should fail if the credentials are invalid.	1- Go to: https://www.saucedemo.com/	-	- The page loads successfully. - The username and password input fields are visible, and the Login button is displayed.
			2- Enter invalid	Username:	- The page

			credentials in the username and password fields, and click the 'Login' button.	wrongusername Password: wrongpassword	displays an error message: “Epic sadface: Username and password do not match any user in this service”. - The username and password fields are highlighted in red. - A red cross icon appears next to the fields.
UI_TC06	Login - Invalid password error	Login should fail if the password is invalid, even when the username is valid.	1- Go to: https://www.saucedemo.com/	-	- The page loads successfully. - The username and password input fields are visible, and the Login button is displayed.
			2- Enter a valid credential in the username field and invalid credential in the password field, and click the 'Login' button.	Username: standard_user Password: wrongpassword	- The page displays an error message “Epic sadface: Username and password do not match any user in this service”. - The username and password fields are highlighted in red. - A red cross icon appears next to the fields.
UI_TC07	Login - Invalid username error	Login should fail if the username is invalid, even when the password is valid.	1- Go to: https://www.saucedemo.com/	-	- The page loads successfully. - The username and password input fields are visible, and the Login button is displayed.
			2- Enter a invalid credential in the username field and valid credential in the	Username: wrongusername Password: secret_sauce	- The page displays: an error message “Epic sadface: Username and

			password field, and click the 'Login' button.		password do not match any user in this service”. - The username and password fields are highlighted in red. - A red cross icon appears next to the fields.
UI_TC08	Login - Locked-out user error	Login should fail if the user is locked-out.	1- Go to: https://www.saucedemo.com/	-	- The page loads successfully. - The username and password input fields are visible, and the Login button is displayed.
			2- Enter valid credentials in the username and password fields for a locked-out user, and click the 'Login' button.	Username: locked_out_user Password: secret_sauce	- The page displays: an error message “Epic sadface: Sorry, this user has been locked out.”. - The username and password fields are highlighted in red. - A red cross icon appears next to the fields.
UI_TC09	Login - Verify error message disappears after closing	The user should be able to close the error message successfully after an invalid login attempt.	1- Go to: https://www.saucedemo.com/	-	- The page loads successfully. - The username and password input fields are visible, and the Login button is displayed.
			2- Click the 'Login' button.	-	- The page displays an error message: “Epic sadface: Username is required”. - The username and password fields are highlighted in red. - A red cross icon

					appears next to the fields.
			3- Click the ‘Close’ button on the error message.	-	- The error message disappears. - The username and password fields return to their original colour. - The red cross next to the fields disappears.
UI_TC10	Cart - Add one product to the cart	The user should be able to add one product to the cart and see it reflected in the cart. — Precondition: the user has an empty cart	1- Go to: https://www.saucedemo.com/. Enter valid credentials in the username and password fields, and click the 'Login' button.	Username: standard_user Password: secret_sauce	- The user is redirected to the products page. URL: https://www.saucedemo.com/inventory.html - All product images are loaded correctly. - All UI elements are properly aligned with UX/UI definition.
			2- Click the 'Add to Cart' button for any product.		- The 'Add to Cart' button changes to 'Remove' and its color changes to red. - The cart displays the number 1.
			3- Click on the cart icon.		- The user is redirected to the cart page. URL: https://www.saucedemo.com/cart.html Title: ‘Your cart’. - The product, quantity, description and price are displayed correctly and match the details from the previous page. - The ‘Remove’, ‘Continue

					Shopping’ and ‘Checkout’ buttons are displayed. - All UI elements are properly aligned with UX/UI definition.
UI_TC11	Cart - Remove a product from the cart	The user should be able to remove a product from the cart. — Precondition: the user is logged in and has at least one product in the cart.	1- Go to: https://www.saucedemo.com/cart.html	-	- The product is visible. - The ‘Remove’, ‘Continue Shopping’ and ‘Checkout’ buttons are displayed.
			2- Click the 'Remove' button for any product.	-	- The product is removed from the cart. - The cart badge number decreases by one (or disappears if it was the last item).
UI_TC12	Cart - Remove a product from the products inventory	The user should be able to remove a product from the products inventory. — Precondition: the user is logged in and has at least one product in the cart.	1- Go to: https://www.saucedemo.com/inventory.html	-	- The list of products is visible. - The ‘Add to cart’ and ‘Remove’ buttons are displayed.
			2- Click the 'Remove' button for any product.	-	- The cart badge number decreases by one (or disappears if it was the last item).
			3- Click on the cart icon.	-	- The product is removed from the cart.
UI_TC13	Cart - Verify cart persistence after logout	The cart should retain its contents even after the user logs out and logs back in. — Preconditio	1- Go to: https://www.saucedemo.com/. Enter valid credentials in the username and password fields, and click the 'Login' button.	Username: standard_user Password: secret_sauce	- The user is redirected to the products page. URL: https://www.saucedemo.com/inventory.html Title: Products - The cart badge displays the correct number of

		n: the user has at least one product in the cart.			items that were in the cart before logging out.
			2- Click on the cart icon.	-	- The user is redirected to the cart page. URL: https://www.saucedemo.com/cart.html Title: ‘Your cart’. - The cart contains all the products that the user had previously logged out.
UI_TC14	Cart - Verify ‘Continue Shopping’ button	Verify the ‘Continue Shopping’ button returns to the list of products. — Precondition: the user is logged in.	1- Go to: https://www.saucedemo.com/cart.html	-	- The cart page is displayed. Title: ‘Your cart’. - The ‘Continue Shopping’ and ‘Checkout’ buttons are displayed.
			2- Click the ‘Continue Shopping’ button.	-	- The user is redirected to the products inventory page. URL: https://www.saucedemo.com/inventory.html Title: ‘Products’.
UI_TC15	Cart - Verify ‘Checkout’ button	The user should be able to start the checkout flow. — Precondition: the user is logged in and has at least one product in the cart.	1- Go to: https://www.saucedemo.com/cart.html	-	- The cart page is displayed. Title: ‘Your cart’. - The ‘Continue Shopping’ and ‘Checkout’ buttons are displayed.
			2- Click the ‘Checkout’ button.	-	- The user is redirected to the user information page. URL: https://www.saucedemo.com/checkout-step-one.html Title: ‘Checkout: Your Information’. - The ‘First Name’, ‘Last Name’ and ‘Zip/Postal Code’ input fields are

					displayed. - The ‘Continue’ and ‘Cancel’ buttons are displayed.

I chose Playwright for this task because it is a versatile tool that supports both UI and API testing. Additionally, it provided a great opportunity to learn a new library.

After creating the test plan, I prioritized automating the most critical and blocker test cases.

For the login functionality, I selected *UI_TC01 - Login with valid credentials* since it is a blocker. As QA testers, ensuring that users can access the application with valid credentials is fundamental.

Another critical test case is *UI_TC05 - Invalid credentials error*, which verifies that unauthorized users cannot log in, maintaining security.

Lastly, *UI_TC08 - Locked-out user error* is essential to confirm that blocked users are unable to access the application, maintaining security too.

For the cart functionality, I selected *UI_TC10 - Add one product to the cart* as a blocker test case. Since the primary goal of an e-commerce application is to sell products, it is crucial to ensure that users can successfully add items to their cart.

I also included *UI_TC11 - Remove a product from the cart* as a critical test case to verify that users can manage their cart as needed.

Lastly, *UI_TC15 - Verify ‘Checkout’ button* is another blocker test case. Given that the main objective of the application is completing sales, it is essential to ensure that users can initiate the checkout process smoothly.

These test cases are strong candidates for automation as they cover the core functionality of the application. They should be executed before every new release, ensuring stability across different user types.

In order to make the tests readable and easy to understand and maintain, I implemented the Page Object Model (POM) design pattern to ensure a more structured and reusable codebase. In the class, I defined both locators and methods to interact with the page elements. This approach helps reduce code duplication and improves the overall readability.

In ‘playwright.config.js’, I specified ‘data-test’ as the primary locator attribute since it is a custom attribute designed for testing.

To ensure optimal login performance, I utilized Playwright's ‘performance’ function, measuring the time from clicking the login button to when the page was fully loaded.

For cart testing, I implemented the ‘beforeEach’ function to handle preconditions, ensuring the user is logged in before adding or removing products or initiating the checkout flow.