

Question 1 - Automated User Flow Testing :

The problem statement was to create a test plan for automating the login process from a user perspective along with proper validations. I then divided it into a two criterias as follows:

1. Happy Path Scenario:

In this the approach was pretty straightforward wherein I wished to automate a script for logging into the site and navigate the site without any errors or exceptions.

To validate this scenario I cross reference the URL of the website using the `expect.toHaveURL` method and the text "Purchase & Opportunity list" present in the page.

Other than that I tried the navigation to different tabs and I validated that by matching the expected and actual URLs of the tabs

2. Negative Testing Scenarios:

For this I wanted to explore out all the error messages that the site gives in various situations and ended up with 3 scenarios which were:

- When the fields are blank:

In this scenario both the email id and password fields were left empty and the submit button was directly pressed. Here we observe that there is an error showing "Required" beneath both the fields giving us a point of validation.

There are two fields with the text "Required" and hence they clash causing a strict mode violation while choosing the locator. This was resolved by mentioning the n-th matching element method.

- When the Email Id and Password are wrong:

This is a scenario where the inputs of both Email Id and Password were put in wrong to test the behavior. Here we saw that there was a message that popped saying "Your Email Id and Password are incorrect."

There was a challenge that was encountered here. At times the page was slow to load and the error message did not show up immediately. This made the validation throw out

a timeout error as the elements were not able to load in time. This was later resolved using the `waitForTimeout` method.

- When the Formats of the Email Id and Password are wrong:

In this scenario, the user puts in the format of the Email Id and Password wrong to see the results, and the observation was that there was a message popping beneath the Email Id box as “Email not Valid” and Password box as “Password Too Short” accordingly.

This was my basic approach towards the automation of the code for login. Furthermore, I observed that my code was using the visiting the URL every time when the code was running, so I used the `BeforeEach` method to iterate the site visit action and called it before every test case.

Question 2 - Finding the Xpath:

Finding the Xpath of an element is straight forward and can be done by inspecting that element, then locating the element, copying the Xpath using right click and making sure it is a unique selector for using it in automation.

In this scenario the logical Xpath would be `“//input[@placeholder=’Enter pizza name’]”`, however we are unable to use this XPath as it is present inside of a Shadow DOM. Shadow DOM is used to encapsulate and hide the elements in the HTML, and can’t be used directly.

I have not worked with Shadow DOMs previously, but brushing up on them led me to believe that we can use them by using the `evaluate` method on the shadow root in Playwright.