Hi Team,

I have written code to automation web based application for automating login page using playwright Typescript

**Instruction**:

To install playwright type command in terminal – “npm init playwright@latest”

Node-modules, playwrightconfig.ts, config.ts, package.json and example test files will be installed

Create a test file, write code and run the code after the execution will get report both in HTML and XML file.

**Code for automating login page**

import { test, expect, Page, Locator } from playwright@latest;

import { chromium } from 'playwright-core';

test.describe('Automation for login', async() =>{

test.configure.describe({mode: 'serial'}){

test('Create Browser', async() => {

broweser = await chromium.launch();

context = await browser.newContext();

page = await context.newPage();

});

test('Go to URL', async() => {

await this.page.goto('www.amazon.com');

});

test('Create Browser', async() => {

await this.page.locator('#Username).fill('User');

await this.page.locator('#Passowrd).fill('Pass');

await this.page.locator('#Submit').click();

expect(await page.locator('#Success Msg')).toHaveText('Logged in Successfully');

});

});

**GitHub commands:**

git clone ‘copied url’

git checkout –b “branch-name”

git commit –m “Automated scripts for login”

git push

git pull

wait for review and approval, once approved merge branch into master branch

**Jenkins (CI/CD)**

Jenkins file will be configured with path and parameters in above automation folder once code is merged into master the code will be seamlessly integrated into jenkins path.

**Questions**

1. **What other tests would you suggest could be written? You do not need to write them out in detail or code them.**

I have automated scripts for Login Page, where in the username, password fields and submit button is automated. I have to automate more by giving assertion to make sure things are working look as per the requirements also after signing should verify the windows navigation if any.

1. **If this test was part of a much larger test set, how would you speed it up?**

I would suggest to use playwright Automation framework with JavaScript/TypeScript language, because playwright has a very good speed, auto-waits and less flaky features.

1. **Sometimes UI tests can fail unpredictably. For example, the page may not have fully loaded before test automation attempts to click a button on a webpage. How would you improve reliability of these tests?**

I would suggest to use playwright Automation framework with TS to avoid such problems Since I have real time experience with playwright I have come across all those problem with simple solution by giving waits like (waitForLoadState, waitForNavigation), also giving assertions for the particular task to be performed….,etc.

**Thanks**