24. Demo of the Application and Test Scenarios which needs to be automated

Enter email,password -login. select the product and click on “Add To Cart” -> click on cart button

In this cart section So you have to make sure that same item what you added is showing in the My Carte Page so for that we will keep some assumptions in this page to make sure it's the same. So once it is confirmed, then you can actually “checkout”.

25. Write the Script to dynamically find the product to buy from list of products

 so you can also write locator in playwright, based upon that text also

if you know that text for that UI and if you don't know to write exact css Then use this.

26. Add assertions for the actions performed and implement necessary sync steps

there are two types of text based locator.

So one is blindly find a button whatever element on the page which have text add to cart.

Another way it will find that text only for elements which have a tagname

We generally call this as a sudo class. Colon has hyphen text.

For isVisible method don’t have auto waits that’s why we use waitfor() before the isvisible method.

27. Handling Auto suggestive dropdown options with playwright – Example

here when you start typing input based upon your input suggestions will show up.

So instead of typing, if you directly paste the word(Ind) suggestions will not show up

how would you tell playwright to type slowly letter by letter instead of dumping them directly on that?

So in this case fill may not be suitable because the whole purpose of fill is that it enters the whole bunch of the text into edit box.

And here we know that in our case we have to enter one by one sequentially.

So for that they have introduced a new locator which is called press sequentially.

Break method is used to comeout of the loop.

28. Complete E2E flow of Placing the order and grab the OrderID with playwright

30. Dynamically find the order from OrderHistory page using Playwright

const {test, expect} = require('@playwright/test');

test('@Web Client App Login', async ({page})=>    {

    const email = "anshika@gmail.com";

    const productName = 'ZARA COAT 3';

    const products = page.locator(".card-body");

 await page.goto("https://rahulshettyacademy.com/client");

 await page.locator("#userEmail").fill(email);

 await page.locator("#userPassword").fill("Iamking@000");

 await page.locator("[value='Login']").click();

 //await page.waitForLoadState("networkidle");   // loaded all the network but due to some flaky that why we use wait for method

await page.locator(".card-body b").first().waitFor();

 const graballTitles = await page.locator(".card-body b").allTextContents();

// console.log(graballTitles);

 const count = await products.count();

 //ZARA COAT 3

 for(let i=0; i<count ; ++i)

 {

   if(await products.nth(i).locator("b").textContent() === productName ) // if both matches go inside the loop )

   {

    await products.nth(i).locator("text= Add To Cart").click();

    break;

}

 }

 //await page.pause();

 await page.locator("[routerlink\*='cart']").click();

 await page.locator("div li").first().waitFor(); // use of waitfor “is visible method is not supporting autowaits in playwirght

 const bool = await page.locator("h3:has-text('ZARA COAT 3')").isVisible();

 expect(bool).toBeTruthy();

 await page.locator("text=Checkout").click();

 await page.locator("[placeholder\*='Country']").pressSequentially("ind",{delay:100});

 const dropdown = page.locator(".ta-results");

 await dropdown.waitFor();

 const optionsCount = await dropdown.locator("button").count();

 for(let i=0;i<optionsCount;++i)

 {

   const text = await dropdown.locator("button").nth(i).textContent();

    if(text === " India")

    {

        await dropdown.locator("button").nth(i).click();

        break;

    }

 }

 expect(page.locator(".user\_\_name [type='text']").first()).toHaveText(email);

    await page.locator(".action\_\_submit").click(); // place order button

    await expect(page.locator(".hero-primary")).toHaveText(" Thankyou for the order. ");

    const orderId = await page.locator(".em-spacer-1 .ng-star-inserted").textContent();

    console.log(orderId);

// \* means regular expression

    await page.locator("button[routerlink\*='myorders']").click();

   await page.locator("tbody").waitFor(); // whole table

   const rows = await page.locator("tbody tr"); // each and every row

   for (let i = 0; i < await rows.count(); ++i) {

      const rowOrderId = await rows.nth(i).locator("th").textContent();

      if (orderId.includes(rowOrderId)) {

         await rows.nth(i).locator("button").first().click(); // click on view button

         break;

      }

   }

   const orderIdDetails = await page.locator(".col-text").textContent();

   expect(orderId.includes(orderIdDetails)).toBeTruthy();

});

If you want to run the specific test file in the test folder then the command is

npx playwright test tests/filename

if you to debug each and every step then npx playwright test tests/filename –debug

playwright inspector will open and we can debug on step over