Automation Testing using Playwright

Introduction

I used Playwright, a cross-browser framework, to complete this assignment. We need to perform a test on the Skyscanner website.

Github link: https://github.com/Saipreetham7/ScaniaAssignment

Assignment Description

1. Open www.skyscanner.net
2. Give origin as "Stockholm" and destination as "Everywhere" (Överallt in Swedish)
3. Choose travel dates 22 Dec 2023 and return date 06 Jan 2024 and 1 adult as a passenger.
4. Search for flights
5. Click on the first entry and see the lowest price shown there is same as the lowest price shown when clicking on the link.

For ex: if it shows 500 SEK for Gothenburg as the cheapest as the first link, when clicked on the link it should show the same 500 SEK as the cheapest. If it shows a different price, then the test case should be failed. Otherwise, it should show as passed.

Playwright

Playwright is a framework for Web Testing and Automation. It allows testing Chromium, Firefox and WebKit with a single API. Playwright is built to enable cross-browser web automation that is ever-green, capable, reliable and fast.

Installation

Playwright has its own test runner for end-to-end tests, we call it the Playwright Test.

Using init command

The easiest way to get started with the Playwright Test is to run the init command.

```
# Run from your project's root directory
npm init playwright@latest
# Or create a new project
npm init playwright@latest new-project
```

This will create a configuration file, optionally add examples, a GitHub Action workflow and a first test example.spec.ts. You can now jump directly to the writing assertions section.

Procedure

Step 1: Go to the **test** folder and create a new test file with the **.spec.js** extension to add the test script.

Step 2: Add the following script and save the file

Step 3: Now open the terminal and execute the following command to run the test case.

```
$ npx playwright test ./tests/test1.spec.js --project firefox --headed
```

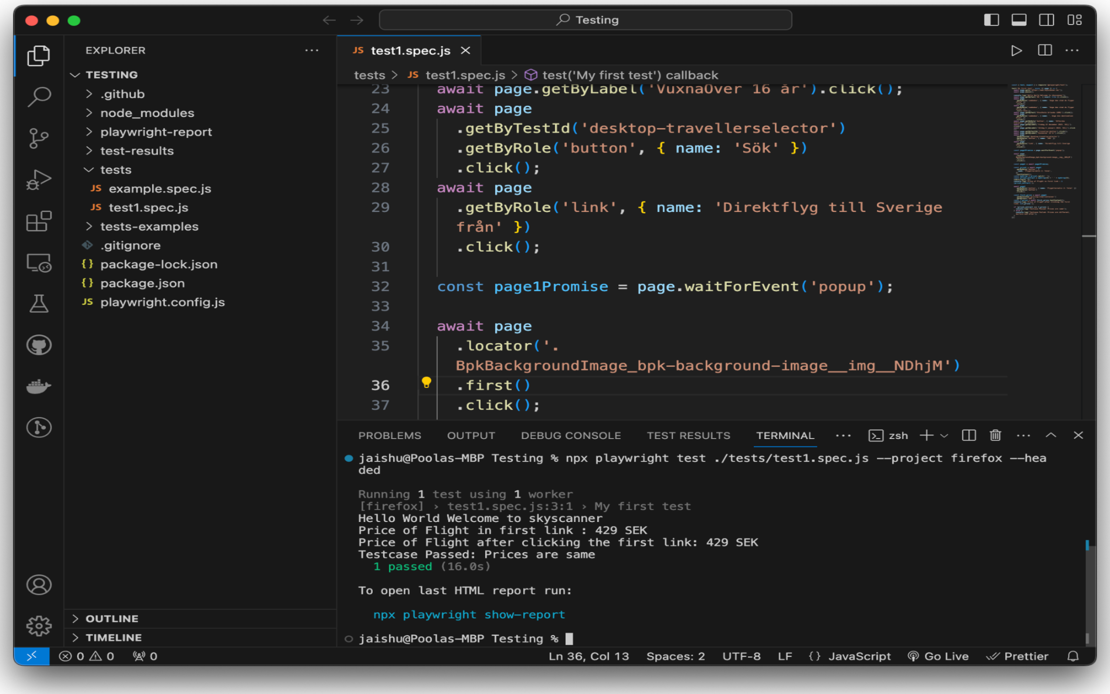
Results

Here we are testing Skyscanner UI end-to-end testing, where we will check the flight rates of the first entry by comparing the prices before clicking the link and after clicking the link.

- If the Prices are the same in both cases, then we will print the Testcase Passed
- If the prices are different, then we will print the Testcase Failed.

Here is the logic for implementation:

```
if (price_before === price_after) {
   console.log('Testcase Passed: Prices are same');
 } else {
   console.log('Testcase Failed: Prices are different, before and after’);
}
```



```
  await page.getByLabel('VuxnaOver 16 år').click();
  await page
    .getByTestId('desktop-travellerselector')
    .getByRole('button', { name: 'Sök' })
    .click();
  await page
    .getByRole('link', { name: 'Direktflyg till Sverige
    från' })
    .click();

  const page1Promise = page.waitForEvent('popup');

  await page
    .locator('.
    BpkBackgroundImage_bpk-background-image__img__NDhjM')
    .first()
    .click();
```

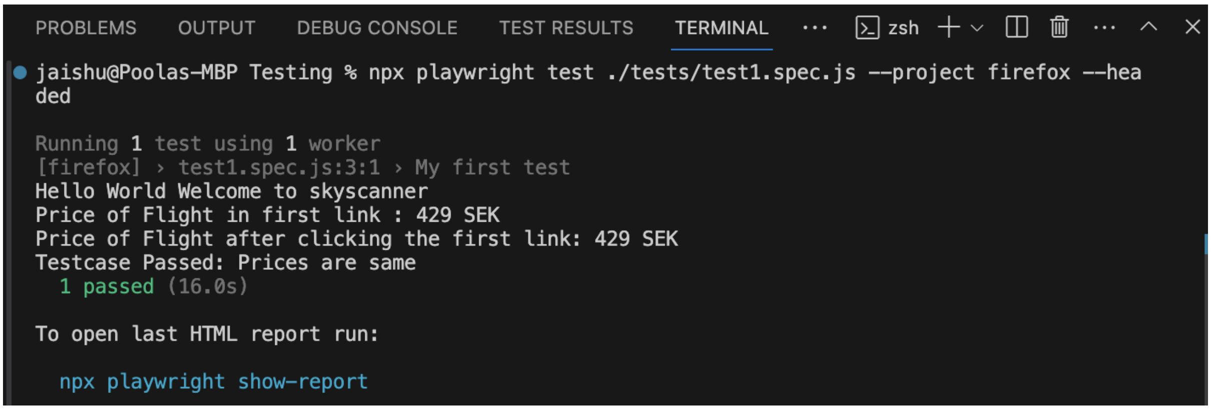
```
jaishu@Poolas-MBP Testing % npx playwright test ./tests/test1.spec.js --project firefox --hea
ded

Running 1 test using 1 worker
[firefox] › test1.spec.js:3:1 › My first test
Hello World Welcome to skyscanner
Price of Flight in first link : 429 SEK
Price of Flight after clicking the first link: 429 SEK
Testcase Passed: Prices are same
  1 passed (16.0s)

To open last HTML report run:

  npx playwright show-report

jaishu@Poolas-MBP Testing %
```

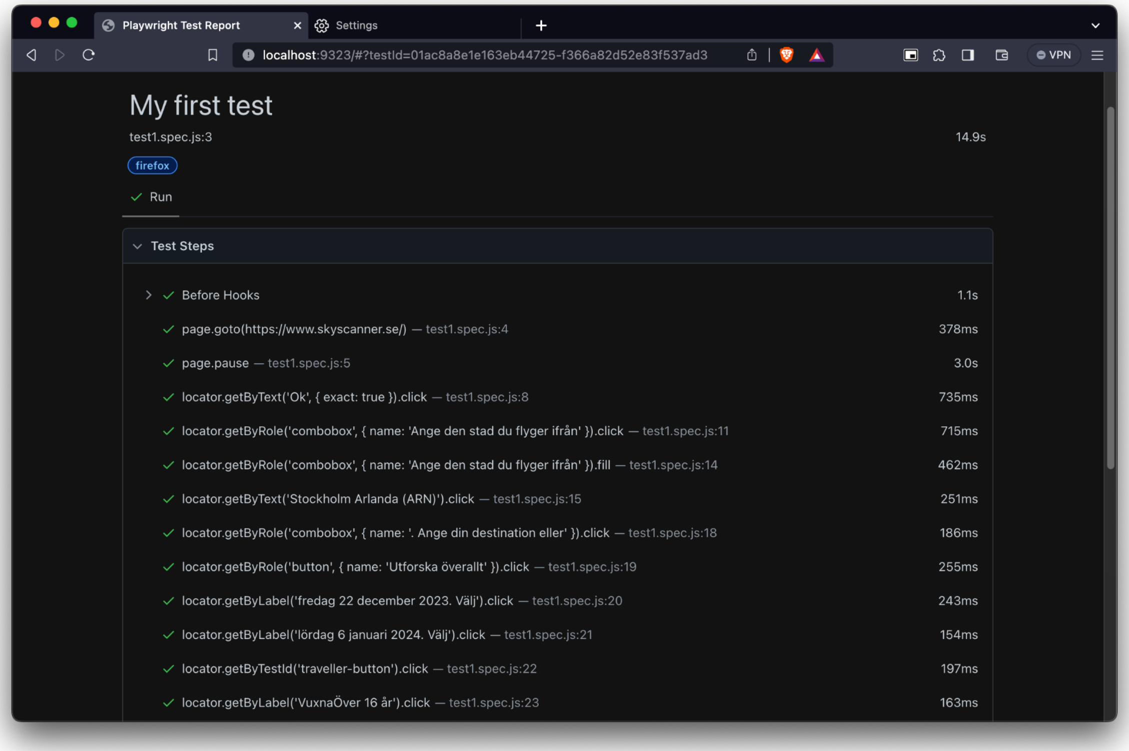


```
jaishu@Poolas-MBP Testing % npx playwright test ./tests/test1.spec.js --project firefox --hea
ded

Running 1 test using 1 worker
[firefox] › test1.spec.js:3:1 › My first test
Hello World Welcome to skyscanner
Price of Flight in first link : 429 SEK
Price of Flight after clicking the first link: 429 SEK
Testcase Passed: Prices are same
  1 passed (16.0s)

To open last HTML report run:

  npx playwright show-report
```


Playwright Test Report
Settings
localhost:9323/#?testId=01ac8a8e1e163eb44725-f366a82d52e83f537ad3
VPN

My first test

test1.spec.js:3 14.9s

firefox

✓ Run

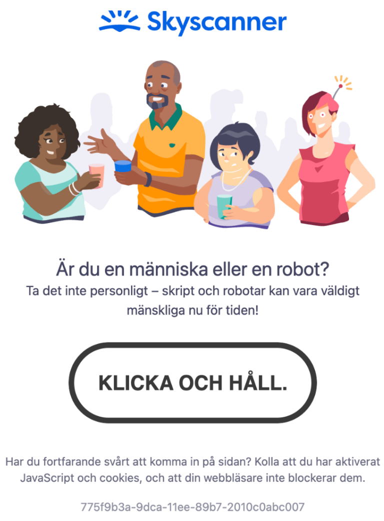
Test Steps

```
✓ Before Hooks                                                                                  1.1s
✓ page.goto(https://www.skyscanner.se/) — test1.spec.js:4                                      378ms
✓ page.pause — test1.spec.js:5                                                                  3.0s
✓ locator.getByText('Ok', { exact: true }).click — test1.spec.js:8                             735ms
✓ locator.getByRole('combobox', { name: 'Ange den stad du flyger ifrån' }).click — test1.spec.js:11   715ms
✓ locator.getByRole('combobox', { name: 'Ange den stad du flyger ifrån' }).fill — test1.spec.js:14    462ms
✓ locator.getByText('Stockholm Arlanda (ARN)').click — test1.spec.js:15                        251ms
✓ locator.getByRole('combobox', { name: '. Ange din destination eller' }).click — test1.spec.js:18    186ms
✓ locator.getByRole('button', { name: 'Utforska överallt' }).click — test1.spec.js:19          255ms
✓ locator.getByLabel('fredag 22 december 2023. Välj').click — test1.spec.js:20                243ms
✓ locator.getByLabel('lördag 6 januari 2024. Välj').click — test1.spec.js:21                  154ms
✓ locator.getByTestId('traveller-button').click — test1.spec.js:22                             197ms
✓ locator.getByLabel('VuxnaÖver 16 år').click — test1.spec.js:23                              163ms
```

Challenges Encountered

Initially, I began automating tasks using the Selenium tool. However, as the test script ran in Selenium, the browser opened a Captcha. I attempted to eradicate it, but it kept popping up. Then I tried some methods to get around the Captcha, but they didn't work.

So I used the Playwright tool and only used the Firefox browser for testing because I discovered that other web browsers generate Captcha while running test scripts. Using this I successfully executed the test case in the Firefox browser and received Test reports, of which I attached a screenshot in the preceding section.

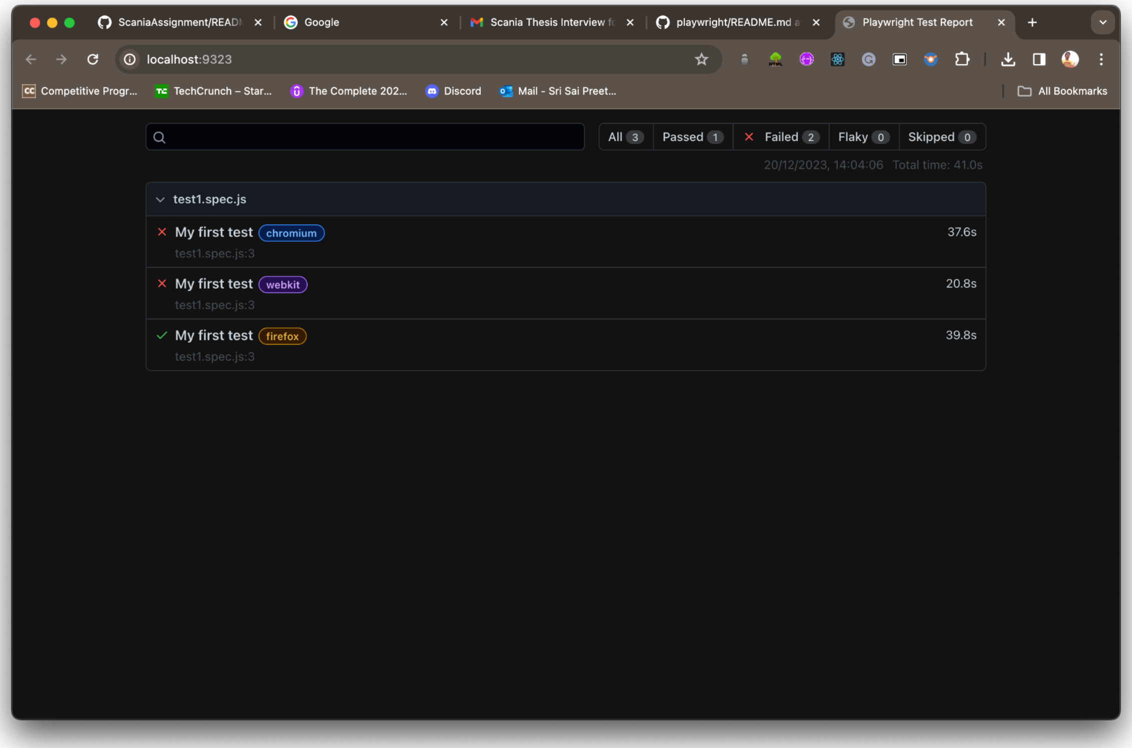


Är du en människa eller en robot?

Ta det inte personligt – skript och robotar kan vara väldigt mänskliga nu för tiden!

Har du fortfarande svårt att komma in på sidan? Kolla att du har aktiverat JavaScript och cookies, och att din webbläsare inte blockerar dem.

775f9b3a-9dca-11ee-89b7-2010c0abc007



All 3 | Passed 1 | ✕ Failed 2 | Flaky 0 | Skipped 0

20/12/2023, 14:04:06 Total time: 41.0s

test1.spec.js

✕ My first test chromium 37.6s
test1.spec.js:3

✕ My first test webkit 20.8s
test1.spec.js:3

✓ My first test firefox 39.8s
test1.spec.js:3