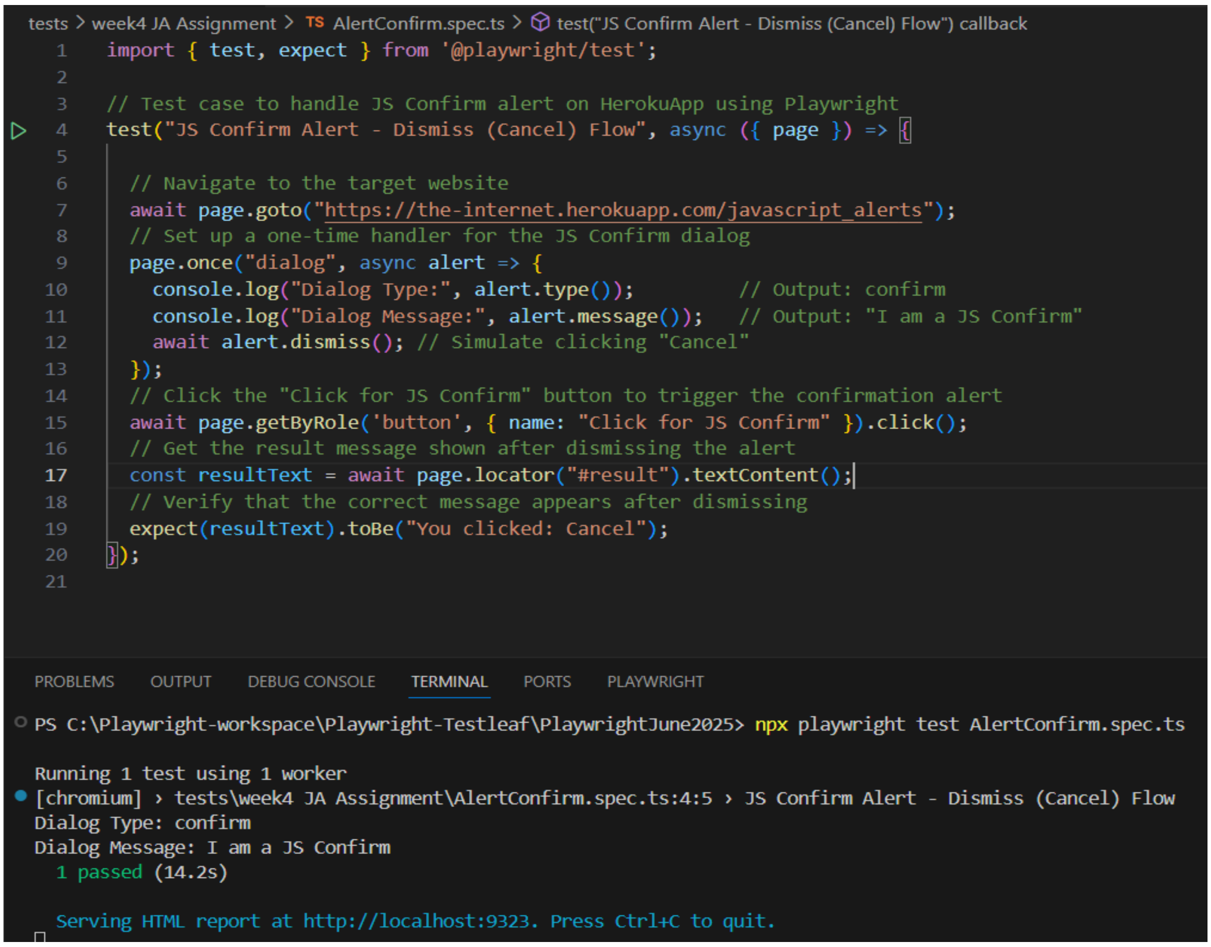
1. JS Alert Confirmation (HerokuApp)

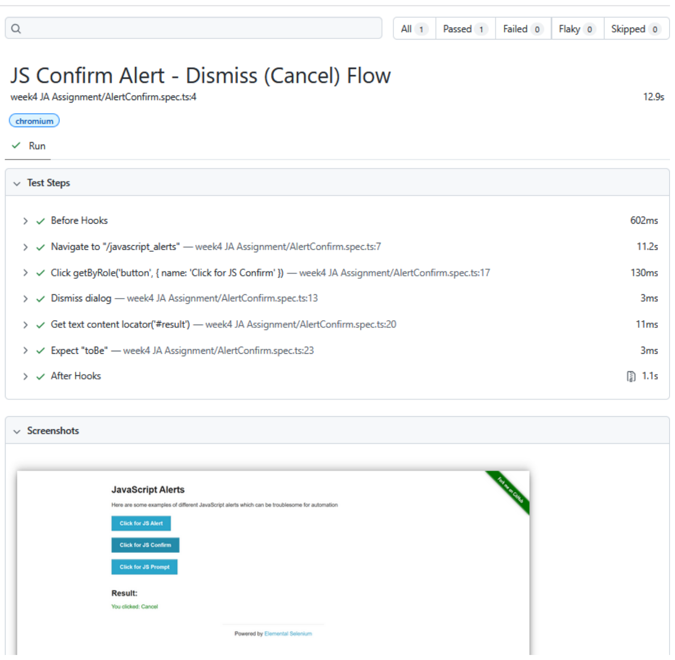


tests > week4 JA Assignment > TS AlertConfirm.spec.ts > test("JS Confirm Alert - Dismiss (Cancel) Flow") callback

```
import { test, expect } from '@playwright/test';

// Test case to handle JS Confirm alert on HerokuApp using Playwright
test("JS Confirm Alert - Dismiss (Cancel) Flow", async ({ page }) => {

  // Navigate to the target website
  await page.goto("https://the-internet.herokuapp.com/javascript_alerts");
  // Set up a one-time handler for the JS Confirm dialog
  page.once("dialog", async alert => {
    console.log("Dialog Type:", alert.type());         // Output: confirm
    console.log("Dialog Message:", alert.message());   // Output: "I am a JS Confirm"
    await alert.dismiss(); // Simulate clicking "Cancel"
  });
  // Click the "Click for JS Confirm" button to trigger the confirmation alert
  await page.getByRole('button', { name: "Click for JS Confirm" }).click();
  // Get the result message shown after dismissing the alert
  const resultText = await page.locator("#result").textContent();
  // Verify that the correct message appears after dismissing
  expect(resultText).toBe("You clicked: Cancel");
});

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS PLAYWRIGHT

```
PS C:\Playwright-workspace\Playwright-Testleaf\PlaywrightJune2025> npx playwright test AlertConfirm.spec.ts

Running 1 test using 1 worker
[chromium] › tests\week4 JA Assignment\AlertConfirm.spec.ts:4:5 › JS Confirm Alert - Dismiss (Cancel) Flow
Dialog Type: confirm
Dialog Message: I am a JS Confirm
  1 passed (14.2s)

  Serving HTML report at http://localhost:9323. Press Ctrl+C to quit.
```



All 1 | Passed 1 | Failed 0 | Flaky 0 | Skipped 0

JS Confirm Alert - Dismiss (Cancel) Flow
week4 JA Assignment/AlertConfirm.spec.ts:4 12.9s

chromium

✓ Run

Test Steps

✓ Before Hooks	602ms
✓ Navigate to "/javascript_alerts" — week4 JA Assignment/AlertConfirm.spec.ts:7	11.2s
✓ Click getByRole('button', { name: 'Click for JS Confirm' }) — week4 JA Assignment/AlertConfirm.spec.ts:17	130ms
✓ Dismiss dialog — week4 JA Assignment/AlertConfirm.spec.ts:13	3ms
✓ Get text content locator('#result') — week4 JA Assignment/AlertConfirm.spec.ts:20	11ms
✓ Expect "toBe" — week4 JA Assignment/AlertConfirm.spec.ts:23	3ms
✓ After Hooks	1.1s

Screenshots

JavaScript Alerts

Here are some examples of different JavaScript alerts which can be troublesome for automation

Click for JS Alert

Click for JS Confirm

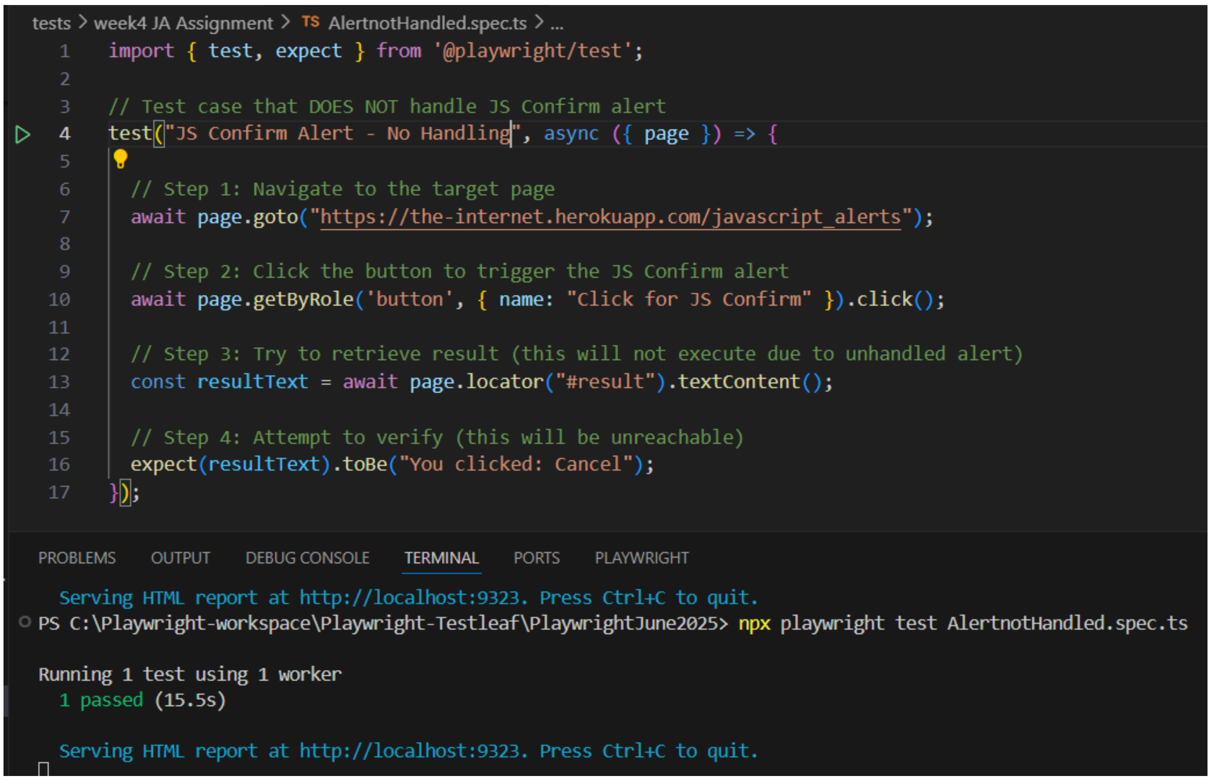
Click for JS Prompt

Result:

You clicked: Cancel

Powered by Elemental Selenium

1(a):Alert not handled - Autodismiss

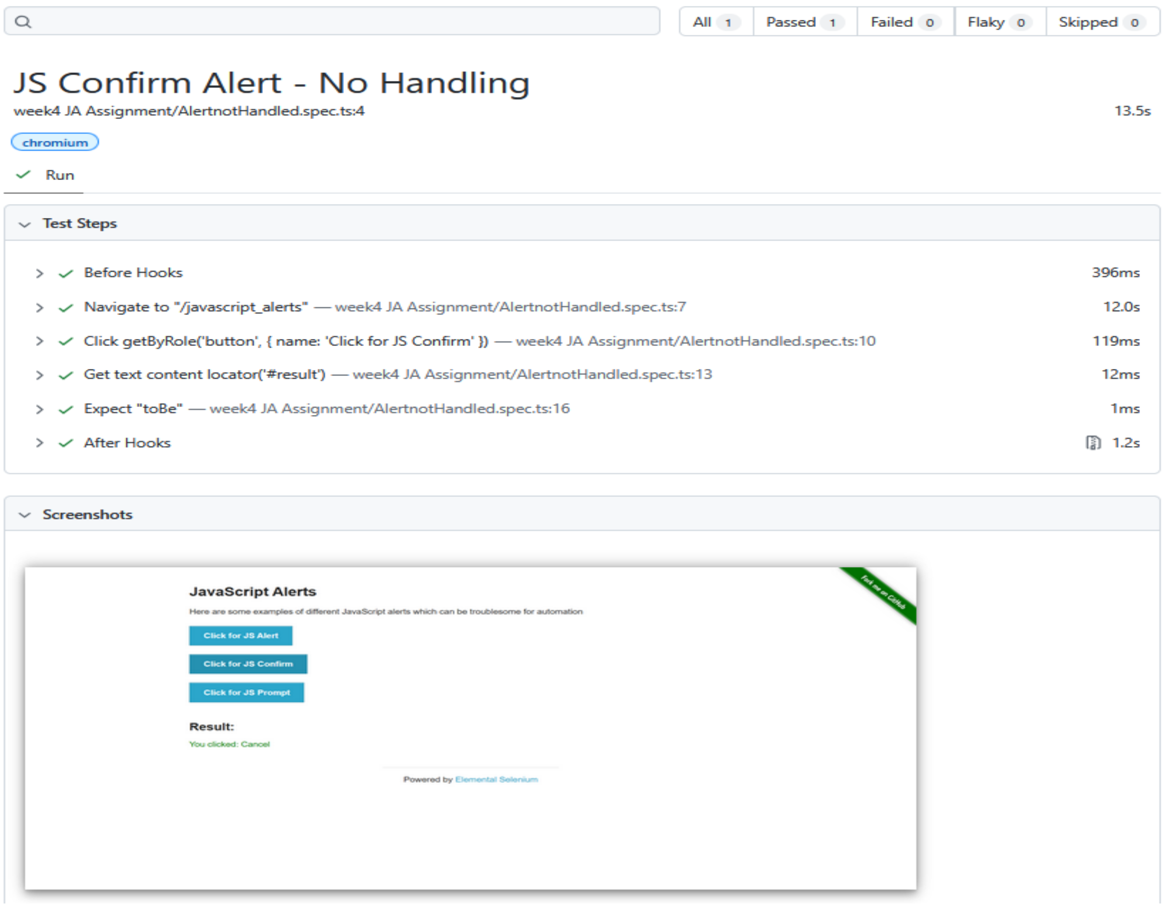


tests > week4 JA Assignment > TS AlertnotHandled.spec.ts > ...

```ts
import { test, expect } from '@playwright/test';

// Test case that DOES NOT handle JS Confirm alert
test("JS Confirm Alert - No Handling", async ({ page }) => {

  // Step 1: Navigate to the target page
  await page.goto("https://the-internet.herokuapp.com/javascript_alerts");

  // Step 2: Click the button to trigger the JS Confirm alert
  await page.getByRole('button', { name: "Click for JS Confirm" }).click();

  // Step 3: Try to retrieve result (this will not execute due to unhandled alert)
  const resultText = await page.locator("#result").textContent();

  // Step 4: Attempt to verify (this will be unreachable)
  expect(resultText).toBe("You clicked: Cancel");
});
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS PLAYWRIGHT

```
  Serving HTML report at http://localhost:9323. Press Ctrl+C to quit.
PS C:\Playwright-workspace\Playwright-Testleaf\PlaywrightJune2025> npx playwright test AlertnotHandled.spec.ts

Running 1 test using 1 worker
  1 passed (15.5s)

  Serving HTML report at http://localhost:9323. Press Ctrl+C to quit.
```



All 1 | Passed 1 | Failed 0 | Flaky 0 | Skipped 0

JS Confirm Alert - No Handling

week4 JA Assignment/AlertnotHandled.spec.ts:4 — 13.5s

chromium

✓ Run

Test Steps

```
✓ Before Hooks                                                                                        396ms
✓ Navigate to "/javascript_alerts" — week4 JA Assignment/AlertnotHandled.spec.ts:7                    12.0s
✓ Click getByRole('button', { name: 'Click for JS Confirm' }) — week4 JA Assignment/AlertnotHandled.spec.ts:10   119ms
✓ Get text content locator('#result') — week4 JA Assignment/AlertnotHandled.spec.ts:13               12ms
✓ Expect "toBe" — week4 JA Assignment/AlertnotHandled.spec.ts:16                                       1ms
✓ After Hooks                                                                                         1.2s
```

Screenshots

JavaScript Alerts

Here are some examples of different JavaScript alerts which can be troublesome for automation

Click for JS Alert

Click for JS Confirm

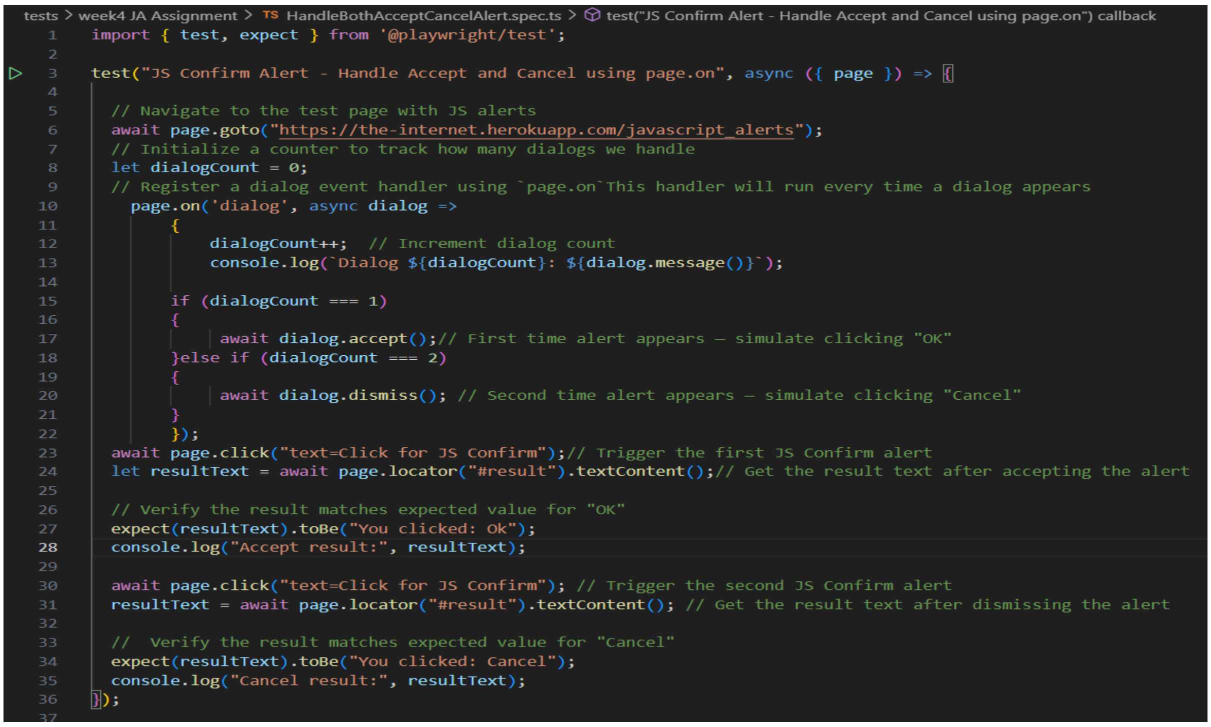
Click for JS Prompt

Result:

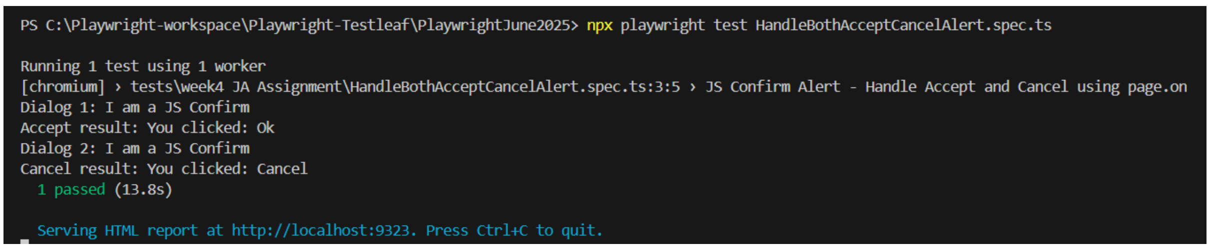
You clicked: Cancel

Powered by Elemental Selenium

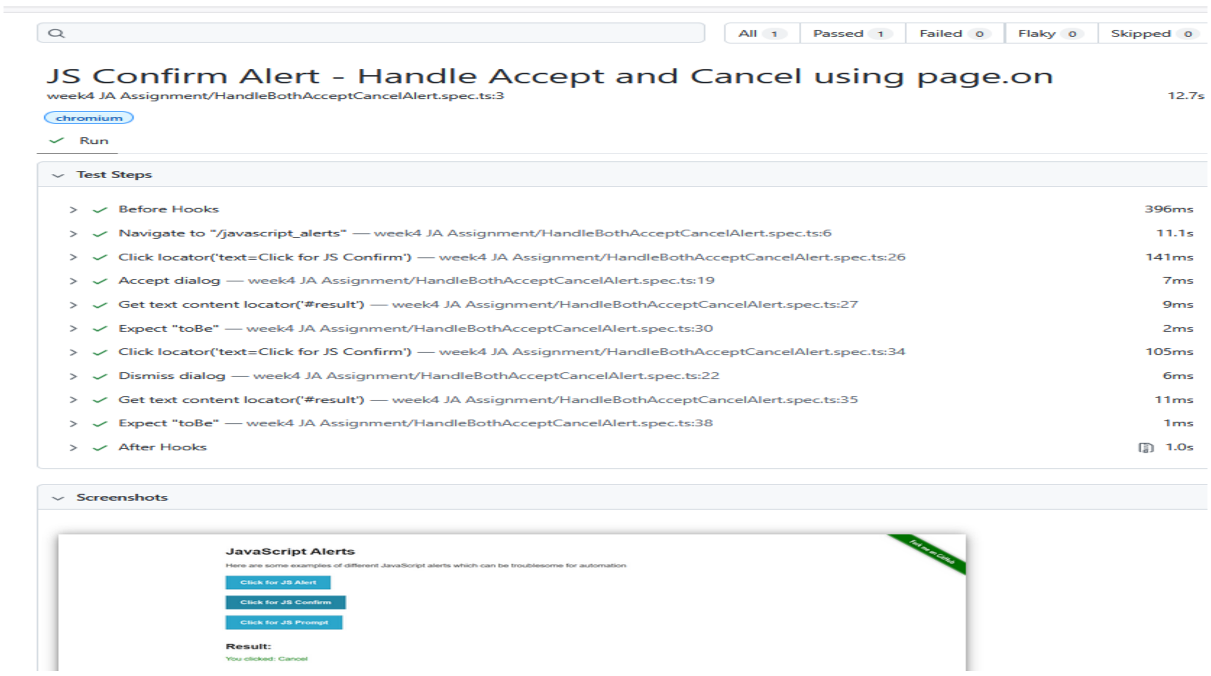
1(b) Simulating both accept and cancel Alerts – Page.on



```
tests > week4 JA Assignment > TS HandleBothAcceptCancelAlert.spec.ts > test("JS Confirm Alert - Handle Accept and Cancel using page.on") callback
import { test, expect } from '@playwright/test';

test("JS Confirm Alert - Handle Accept and Cancel using page.on", async ({ page }) => {

  // Navigate to the test page with JS alerts
  await page.goto("https://the-internet.herokuapp.com/javascript_alerts");
  // Initialize a counter to track how many dialogs we handle
  let dialogCount = 0;
  // Register a dialog event handler using `page.on`This handler will run every time a dialog appears
    page.on('dialog', async dialog =>
        {
            dialogCount++;  // Increment dialog count
            console.log(`Dialog ${dialogCount}: ${dialog.message()}`);

        if (dialogCount === 1)
        {
             await dialog.accept();// First time alert appears — simulate clicking "OK"
        }else if (dialogCount === 2)
        {
             await dialog.dismiss(); // Second time alert appears — simulate clicking "Cancel"
        }
        });
  await page.click("text=Click for JS Confirm");// Trigger the first JS Confirm alert
  let resultText = await page.locator("#result").textContent();// Get the result text after accepting the alert

  // Verify the result matches expected value for "OK"
  expect(resultText).toBe("You clicked: Ok");
  console.log("Accept result:", resultText);

  await page.click("text=Click for JS Confirm"); // Trigger the second JS Confirm alert
  resultText = await page.locator("#result").textContent(); // Get the result text after dismissing the alert

  //  Verify the result matches expected value for "Cancel"
  expect(resultText).toBe("You clicked: Cancel");
  console.log("Cancel result:", resultText);
});
```



```
PS C:\Playwright-workspace\Playwright-Testleaf\PlaywrightJune2025> npx playwright test HandleBothAcceptCancelAlert.spec.ts

Running 1 test using 1 worker
[chromium] › tests\week4 JA Assignment\HandleBothAcceptCancelAlert.spec.ts:3:5 › JS Confirm Alert - Handle Accept and Cancel using page.on
Dialog 1: I am a JS Confirm
Accept result: You clicked: Ok
Dialog 2: I am a JS Confirm
Cancel result: You clicked: Cancel
  1 passed (13.8s)

  Serving HTML report at http://localhost:9323. Press Ctrl+C to quit.
```



All 1 | Passed 1 | Failed 0 | Flaky 0 | Skipped 0

JS Confirm Alert - Handle Accept and Cancel using page.on
week4 JA Assignment/HandleBothAcceptCancelAlert.spec.ts:3 — 12.7s
chromium
✓ Run

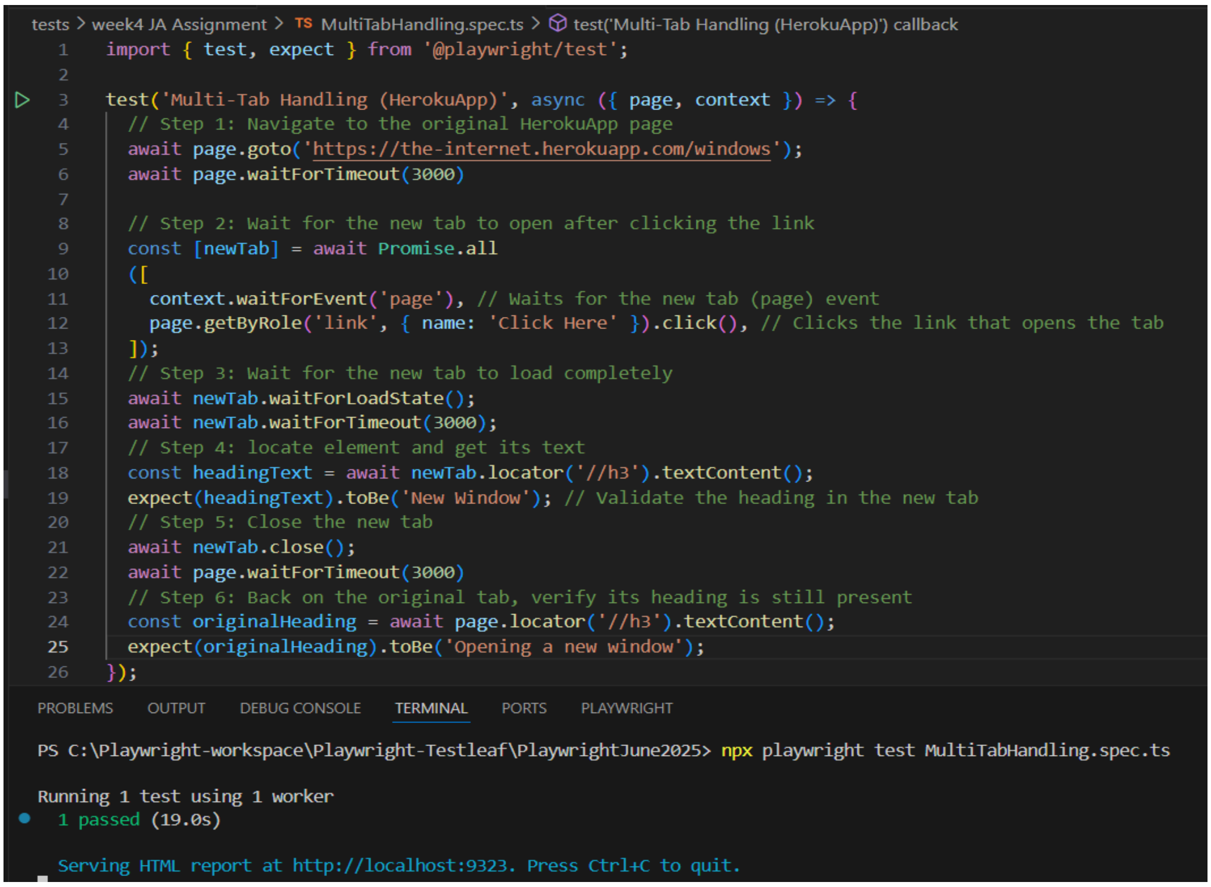
Test Steps

```
> ✓ Before Hooks                                                                                                   396ms
> ✓ Navigate to "/javascript_alerts" — week4 JA Assignment/HandleBothAcceptCancelAlert.spec.ts:6                    11.1s
> ✓ Click locator('text=Click for JS Confirm') — week4 JA Assignment/HandleBothAcceptCancelAlert.spec.ts:26        141ms
> ✓ Accept dialog — week4 JA Assignment/HandleBothAcceptCancelAlert.spec.ts:19                                        7ms
> ✓ Get text content locator('#result') — week4 JA Assignment/HandleBothAcceptCancelAlert.spec.ts:27                  9ms
> ✓ Expect "toBe" — week4 JA Assignment/HandleBothAcceptCancelAlert.spec.ts:30                                        2ms
> ✓ Click locator('text=Click for JS Confirm') — week4 JA Assignment/HandleBothAcceptCancelAlert.spec.ts:34        105ms
> ✓ Dismiss dialog — week4 JA Assignment/HandleBothAcceptCancelAlert.spec.ts:22                                       6ms
> ✓ Get text content locator('#result') — week4 JA Assignment/HandleBothAcceptCancelAlert.spec.ts:35                 11ms
> ✓ Expect "toBe" — week4 JA Assignment/HandleBothAcceptCancelAlert.spec.ts:38                                        1ms
> ✓ After Hooks                                                                                                      1.0s
```

Screenshots

JavaScript Alerts
Here are some examples of different JavaScript alerts which can be troublesome for automation
Click for JS Alert
Click for JS Confirm
Click for JS Prompt
Result:
You clicked: Cancel

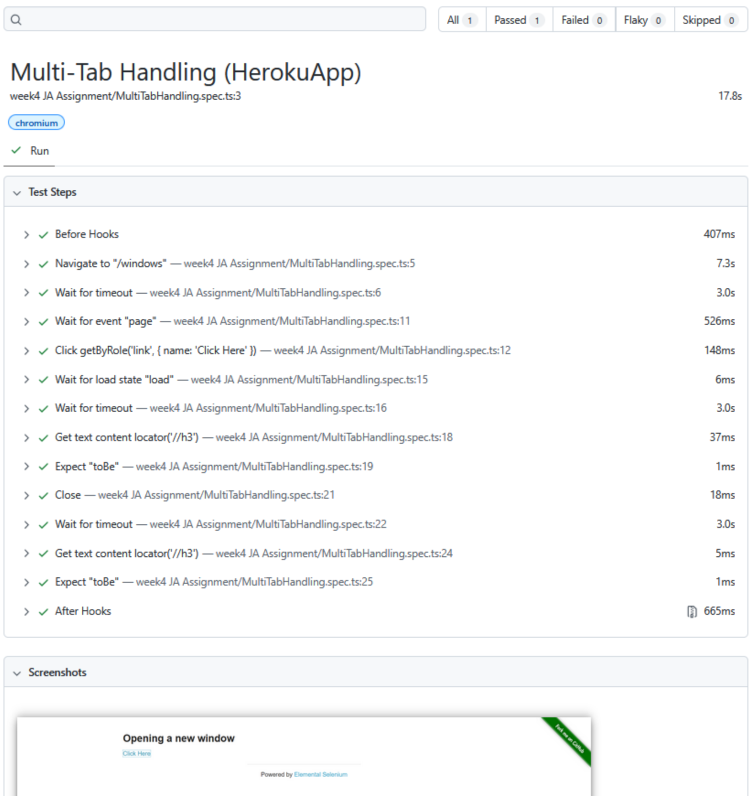
2. Multi-Tab Handling (HerokuApp)



tests > week4 JA Assignment > TS MultiTabHandling.spec.ts > test('Multi-Tab Handling (HerokuApp)') callback

```typescript
import { test, expect } from '@playwright/test';

test('Multi-Tab Handling (HerokuApp)', async ({ page, context }) => {
  // Step 1: Navigate to the original HerokuApp page
  await page.goto('https://the-internet.herokuapp.com/windows');
  await page.waitForTimeout(3000)

  // Step 2: Wait for the new tab to open after clicking the link
  const [newTab] = await Promise.all
  ([
    context.waitForEvent('page'), // Waits for the new tab (page) event
    page.getByRole('link', { name: 'Click Here' }).click(), // Clicks the link that opens the tab
  ]);
  // Step 3: Wait for the new tab to load completely
  await newTab.waitForLoadState();
  await newTab.waitForTimeout(3000);
  // Step 4: locate element and get its text
  const headingText = await newTab.locator('//h3').textContent();
  expect(headingText).toBe('New Window'); // Validate the heading in the new tab
  // Step 5: Close the new tab
  await newTab.close();
  await page.waitForTimeout(3000)
  // Step 6: Back on the original tab, verify its heading is still present
  const originalHeading = await page.locator('//h3').textContent();
  expect(originalHeading).toBe('Opening a new window');
});
```

```
PS C:\Playwright-workspace\Playwright-Testleaf\PlaywrightJune2025> npx playwright test MultiTabHandling.spec.ts

Running 1 test using 1 worker
  1 passed (19.0s)

  Serving HTML report at http://localhost:9323. Press Ctrl+C to quit.
```



Multi-Tab Handling (HerokuApp)

week4 JA Assignment/MultiTabHandling.spec.ts:3   17.8s

chromium

✓ Run

Test Steps

```
✓ Before Hooks                                                                                   407ms
✓ Navigate to "/windows" — week4 JA Assignment/MultiTabHandling.spec.ts:5                          7.3s
✓ Wait for timeout — week4 JA Assignment/MultiTabHandling.spec.ts:6                                3.0s
✓ Wait for event "page" — week4 JA Assignment/MultiTabHandling.spec.ts:11                        526ms
✓ Click getByRole('link', { name: 'Click Here' }) — week4 JA Assignment/MultiTabHandling.spec.ts:12  148ms
✓ Wait for load state "load" — week4 JA Assignment/MultiTabHandling.spec.ts:15                     6ms
✓ Wait for timeout — week4 JA Assignment/MultiTabHandling.spec.ts:16                               3.0s
✓ Get text content locator('//h3') — week4 JA Assignment/MultiTabHandling.spec.ts:18              37ms
✓ Expect "toBe" — week4 JA Assignment/MultiTabHandling.spec.ts:19                                  1ms
✓ Close — week4 JA Assignment/MultiTabHandling.spec.ts:21                                         18ms
✓ Wait for timeout — week4 JA Assignment/MultiTabHandling.spec.ts:22                               3.0s
✓ Get text content locator('//h3') — week4 JA Assignment/MultiTabHandling.spec.ts:24               5ms
✓ Expect "toBe" — week4 JA Assignment/MultiTabHandling.spec.ts:25                                  1ms
✓ After Hooks                                                                                    665ms
```

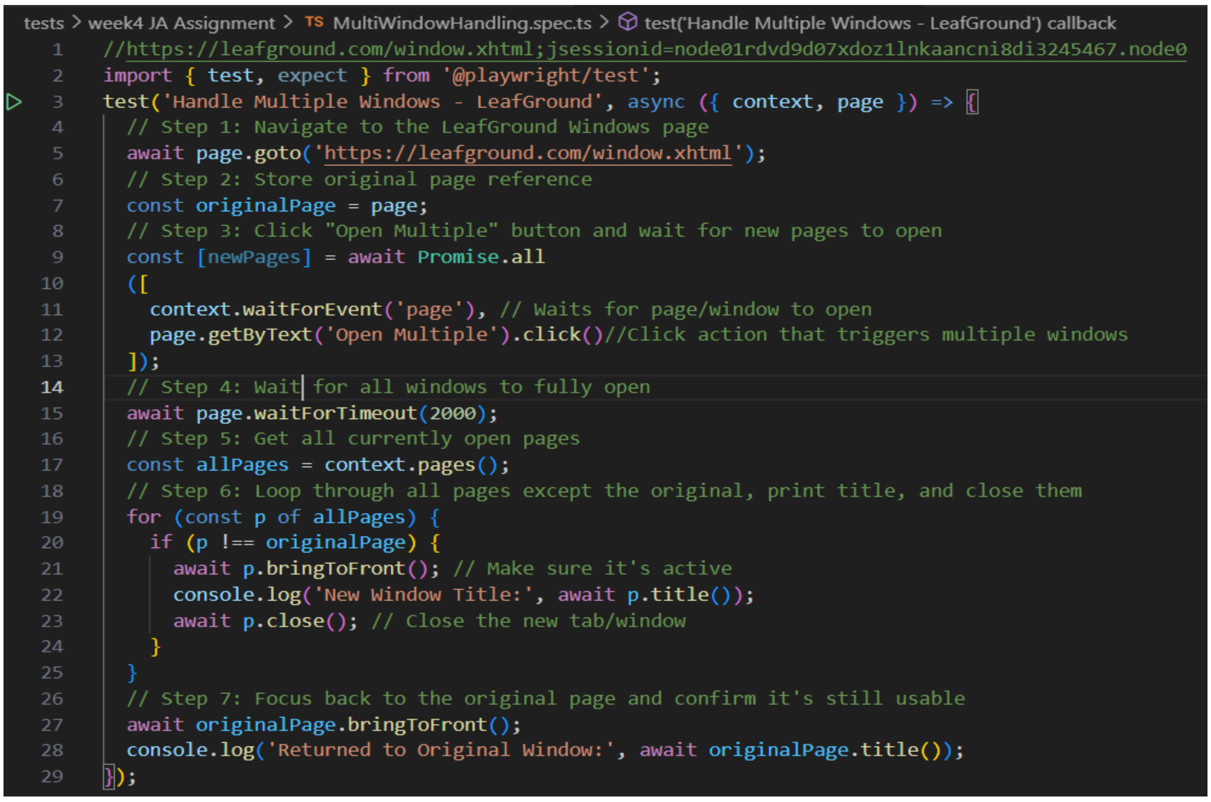
Screenshots

Opening a new window

Click Here

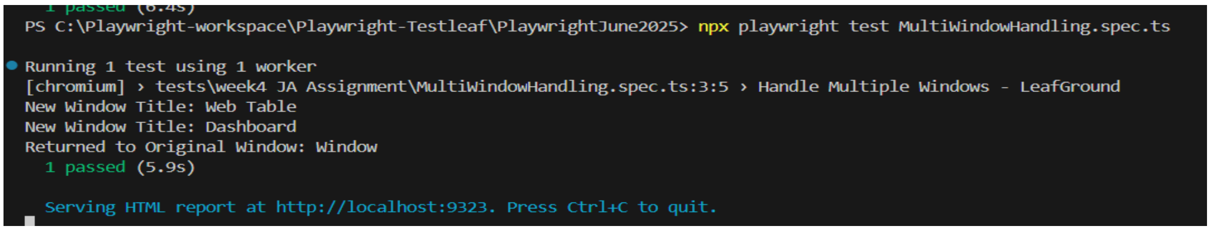
Powered by Elemental Selenium

3. Multi-Window Handling(Leafground.com)

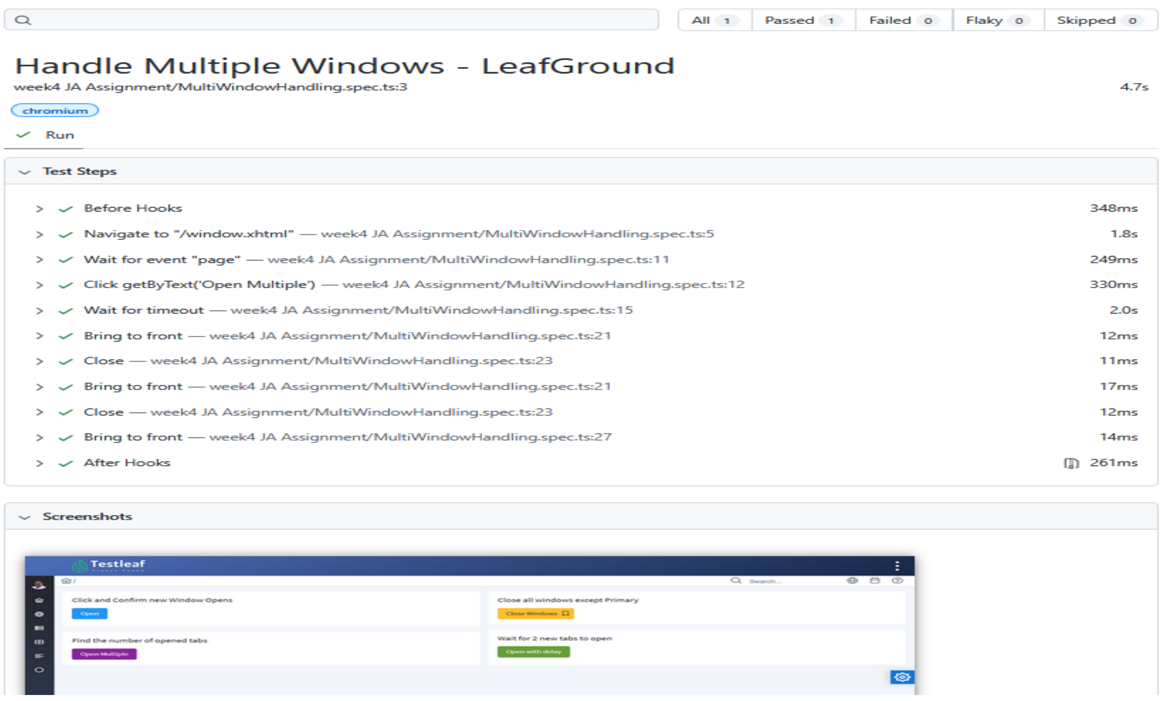


tests > week4 JA Assignment > TS MultiWindowHandling.spec.ts > test('Handle Multiple Windows - LeafGround') callback

```typescript
//https://leafground.com/window.xhtml;jsessionid=node01rdvd9d07xdoz1lnkaancni8di3245467.node0
import { test, expect } from '@playwright/test';
test('Handle Multiple Windows - LeafGround', async ({ context, page }) => {
  // Step 1: Navigate to the LeafGround Windows page
  await page.goto('https://leafground.com/window.xhtml');
  // Step 2: Store original page reference
  const originalPage = page;
  // Step 3: Click "Open Multiple" button and wait for new pages to open
  const [newPages] = await Promise.all
  ([
    context.waitForEvent('page'), // Waits for page/window to open
    page.getByText('Open Multiple').click()//Click action that triggers multiple windows
  ]);
  // Step 4: Wait for all windows to fully open
  await page.waitForTimeout(2000);
  // Step 5: Get all currently open pages
  const allPages = context.pages();
  // Step 6: Loop through all pages except the original, print title, and close them
  for (const p of allPages) {
    if (p !== originalPage) {
      await p.bringToFront(); // Make sure it's active
      console.log('New Window Title:', await p.title());
      await p.close(); // Close the new tab/window
    }
  }
  // Step 7: Focus back to the original page and confirm it's still usable
  await originalPage.bringToFront();
  console.log('Returned to Original Window:', await originalPage.title());
});
```



```
  1 passed (6.4s)
PS C:\Playwright-workspace\Playwright-Testleaf\PlaywrightJune2025> npx playwright test MultiWindowHandling.spec.ts

● Running 1 test using 1 worker
  [chromium] › tests\week4 JA Assignment\MultiWindowHandling.spec.ts:3:5 › Handle Multiple Windows - LeafGround
  New Window Title: Web Table
  New Window Title: Dashboard
  Returned to Original Window: Window
    1 passed (5.9s)

    Serving HTML report at http://localhost:9323. Press Ctrl+C to quit.
```



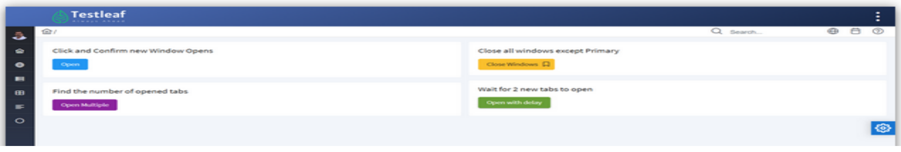
All 1 | Passed 1 | Failed 0 | Flaky 0 | Skipped 0

Handle Multiple Windows - LeafGround
week4 JA Assignment/MultiWindowHandling.spec.ts:3 4.7s
chromium
✓ Run

Test Steps

```
> ✓ Before Hooks                                                                          348ms
> ✓ Navigate to "/window.xhtml" — week4 JA Assignment/MultiWindowHandling.spec.ts:5         1.8s
> ✓ Wait for event "page" — week4 JA Assignment/MultiWindowHandling.spec.ts:11            249ms
> ✓ Click getByText('Open Multiple') — week4 JA Assignment/MultiWindowHandling.spec.ts:12 330ms
> ✓ Wait for timeout — week4 JA Assignment/MultiWindowHandling.spec.ts:15                  2.0s
> ✓ Bring to front — week4 JA Assignment/MultiWindowHandling.spec.ts:21                    12ms
> ✓ Close — week4 JA Assignment/MultiWindowHandling.spec.ts:23                             11ms
> ✓ Bring to front — week4 JA Assignment/MultiWindowHandling.spec.ts:21                    17ms
> ✓ Close — week4 JA Assignment/MultiWindowHandling.spec.ts:23                             12ms
> ✓ Bring to front — week4 JA Assignment/MultiWindowHandling.spec.ts:27                    14ms
> ✓ After Hooks                                                                           261ms
```

Screenshots


Testleaf
Click and Confirm new Window Opens
Open
Close all windows except Primary
Close Windows
Find the number of opened tabs
Open Multiple
Wait for 2 new tabs to open
Open with delay