

# Testing with Playwright CRX - Test Cases – Dominos app

---

## Table of Contents

Test ID: 1 .....	1
Test ID: 2 .....	2
Test ID: 3 .....	3
Test ID: 4 .....	4
Test ID: 5 .....	5
Test ID: 6 .....	6
Test ID: 7 .....	7
Test ID: 8 .....	8
Test ID: 9 .....	9
Test ID: 10.....	10
Test ID: 11.....	11
Test ID: 12.....	12
Test ID: 13.....	13

### Test ID: 1

URL being tested: /register

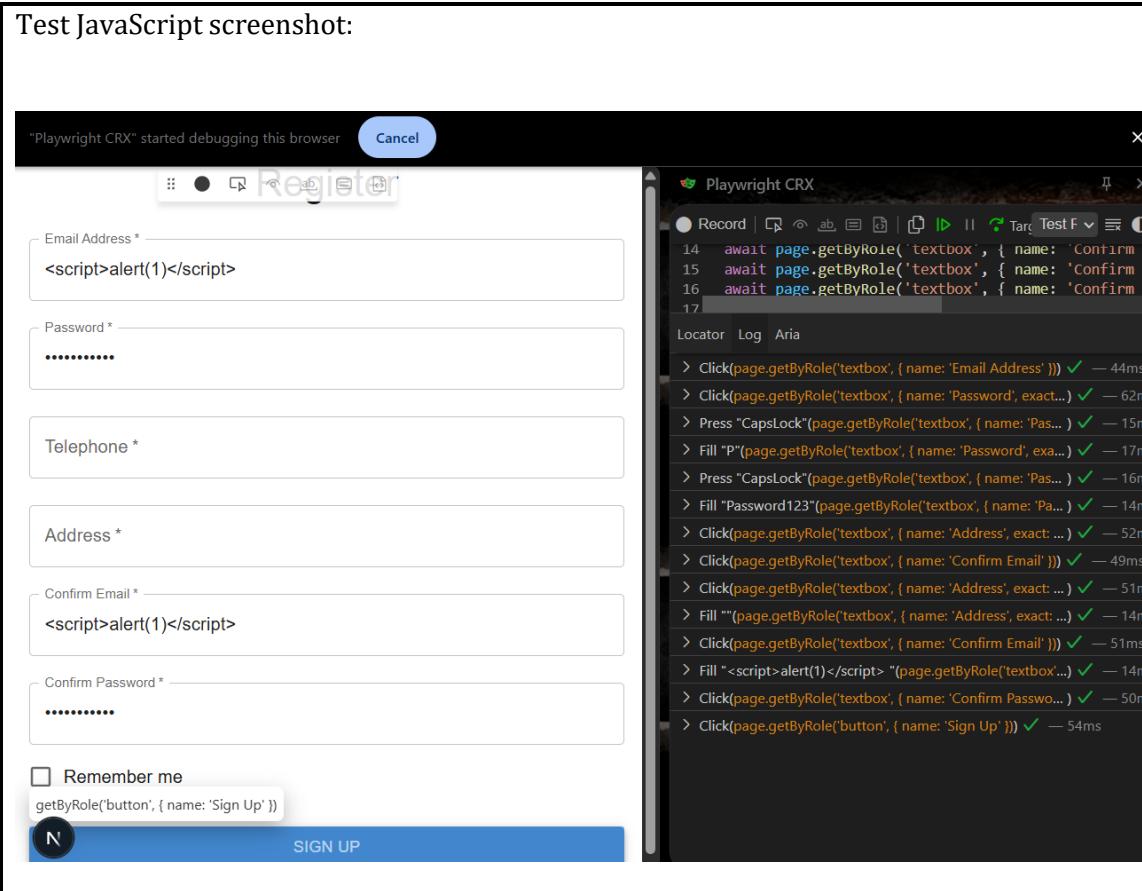
What are you testing: Username and password fields.

Input: admin<script>alert(1)</script> / P@ssword123

Expected Result: email should be rejected

Result: fail email pass on password

## Test JavaScript screenshot:



# Test ID: 2

## URL being tested: /register

## What are you testing: Username field.

## Input:

Expected Result: Too long username should rejected.

## Result:

## Test JavaScript screenshot:

The screenshot shows a browser window with a registration form and a DevTools sidebar.

**Browser Form:**

- Email Address \*: aaa
- Password \*: \*\*\*\*\*
- Telephone \*: (empty)
- Address \*: (empty)
- Confirm Email \*: aaa
- Confirm Password \*: \*\*\*\*\*
- Remember me
- `getByRole('button', { name: 'Sign Up' })`
- SIGN UP** button

**DevTools Sidebar:**

- Playwright CRX started debugging this browser
- Cancel button
- Record button
- Log tab (selected):
  - Click(page.getByRole('textbox', { name: 'Email Address' })) — 63m
  - Click(page.getByRole('textbox', { name: 'Confirm Email' })) — 72m
  - Click(page.getByRole('button', { name: 'Sign Up' })) — 87ms
- Locator tab
- Aria tab

**Test ID: 3**

## URL being tested: /register

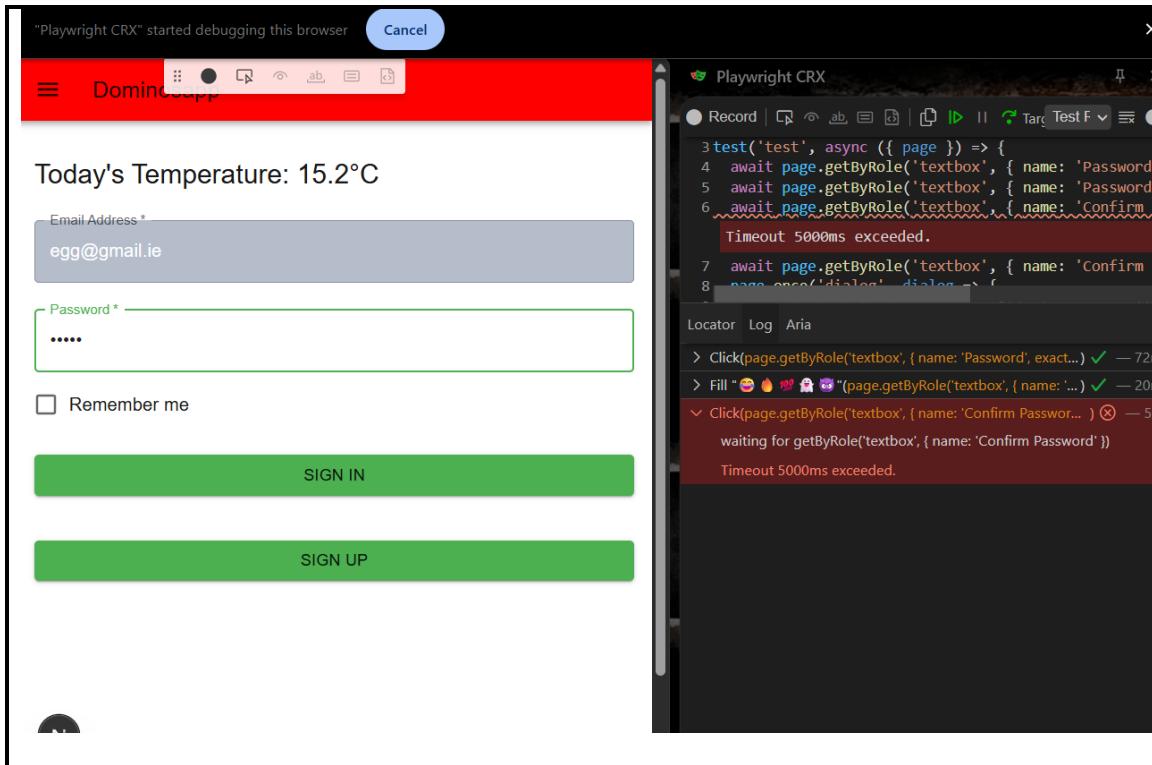
What are you testing: Password field with emoji input.

Input: 😊 🔥 💯 🎃 🐱

Expected Result: Registration should reject.

Result: imoji pass failed

## Test JavaScript screenshot:



#### Test ID: 4

URL being tested: /register

What are you testing: Email field.

Input: test@example.com' DROP TABLE users;--

Expected Result: SQL injection attempt should fail.

Result: result failed

Test JavaScript screenshot:

The screenshot shows a browser window with a registration form titled "Register". The form includes fields for Email Address\*, Password\*, Telephone\*, Address\*, Confirm Email\*, and Confirm Password\*. A "Remember me" checkbox and a "SIGN UP" button are also present. To the right of the browser is a dark-themed developer tool panel titled "Playwright CRX". The panel has tabs for "Record", "Log", and "Aria". The "Log" tab displays a sequence of API calls and their results. At the bottom of the log, there is an error message: "Timeout 5000ms exceeded." The "Aria" tab is currently empty.

```
4 await page.getByRole('textbox', { name: 'Email Address' })
5 await page.getByRole('textbox', { name: 'Email Address' })
6 await page.getByRole('textbox', { name: 'Password' })
7 await page.getByRole('textbox', { name: 'Password' })
8 await page.getByRole('textbox', { name: 'Confirm Email' })
9 await page.getByRole('textbox', { name: 'Confirm Email' })
10
```

Locator Log Aria

> Click(page.getByRole('textbox', { name: 'Email Address' })) ✓ — 83ms  
> Fill "test@example.com" DROP TABLE users;--"(page.get... ) ✓ — 22ms  
> Click(page.getByRole('textbox', { name: 'Password' }, example)) ✓ — 112ms  
> Fill "123"(page.getByRole('textbox', { name: 'Password' }, example)) ✓ — 15ms  
✓ Click(page.getByRole('textbox', { name: 'Confirm Email' })) ✘ — 5.05s  
waiting for getByRole('textbox', { name: 'Confirm Email' })  
Timeout 5000ms exceeded.

Test ID: 5

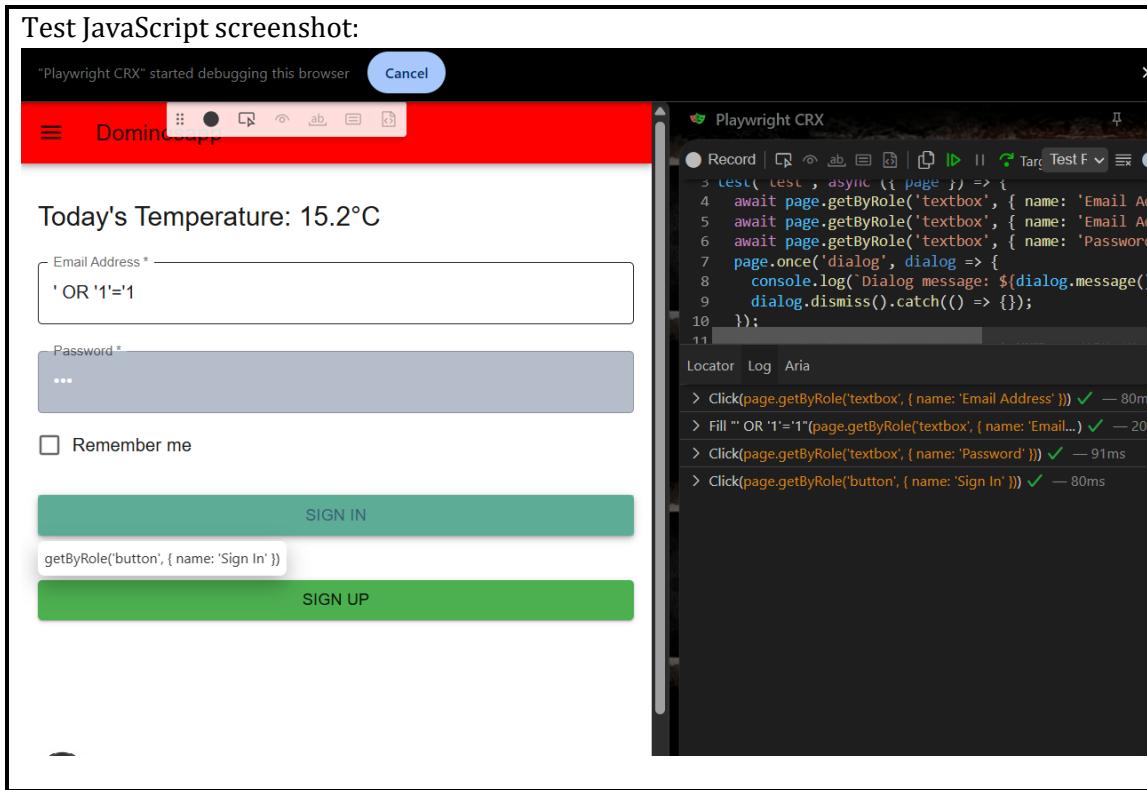
## URL being tested: /login

## What are you testing: Username field.

Input: ' OR '1'='1

Expected Result: SQL injection should fail.

Result: passed



### Test ID: 6

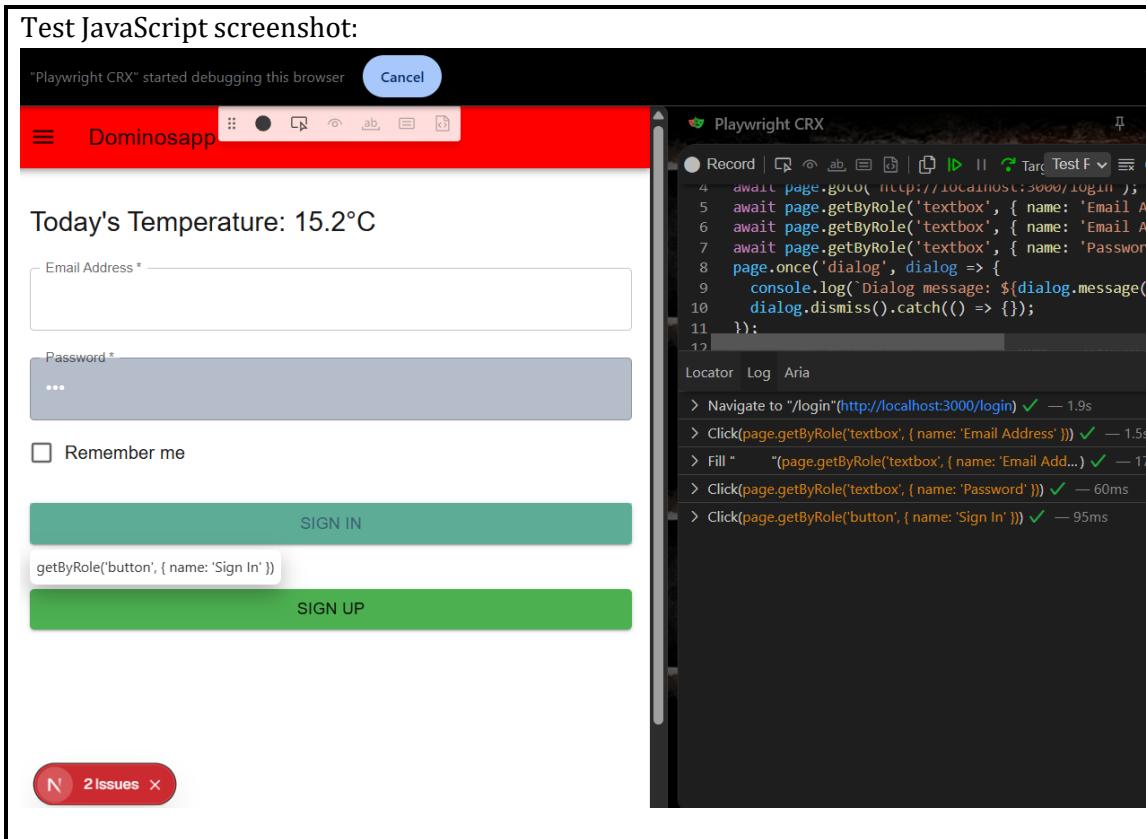
URL being tested: /login

What are you testing: Password field.

Input: " " (all spaces)

Expected Result: Rejected cannot log in with blank input.

Result: passed



**Test ID: 7**

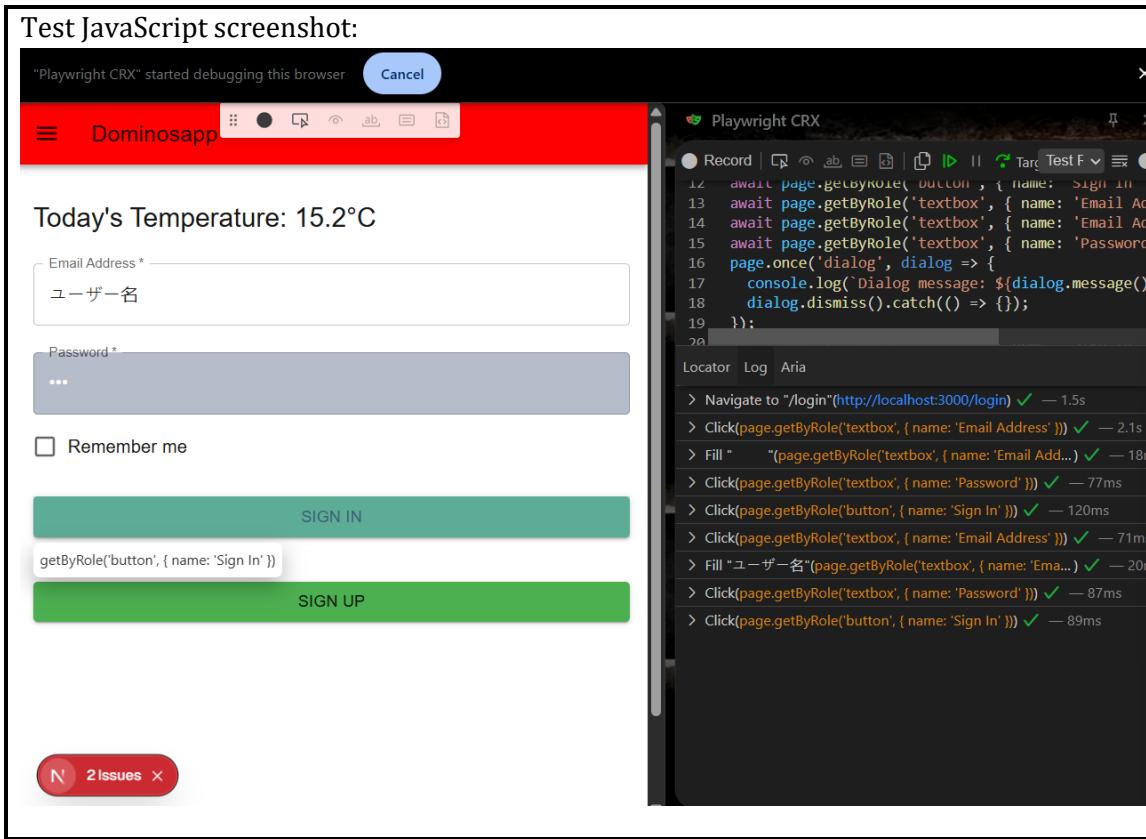
## URL being tested: /login

What are you testing: Username with Unicode.

Input: ユーザー名

Expected Result: Either rejected or handled safely.

Result: passed



## Test ID: 8

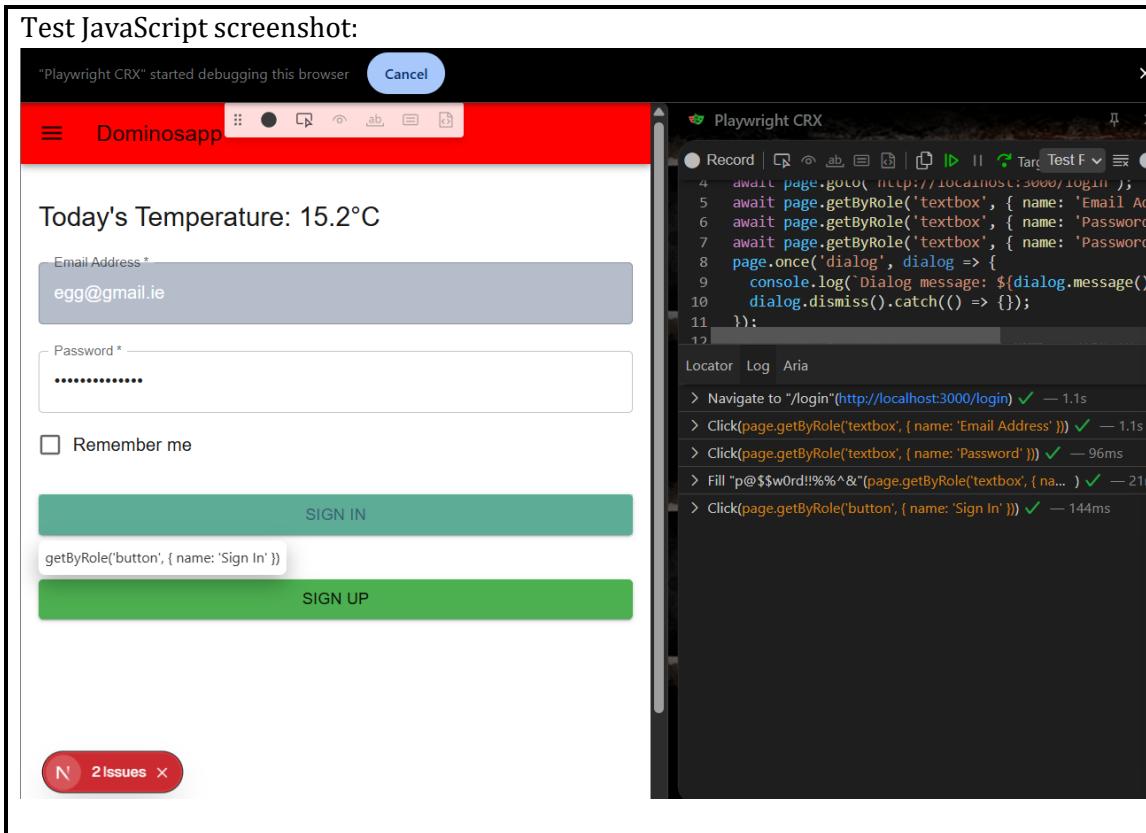
## URL being tested: /login

What are you testing: Password with special chars.

Input: p@\$\$w0rd!!%%^&

Expected Result: Login should accept if account exists, else reject.

Result: passed



# Test ID: 9

## URL being tested: /login

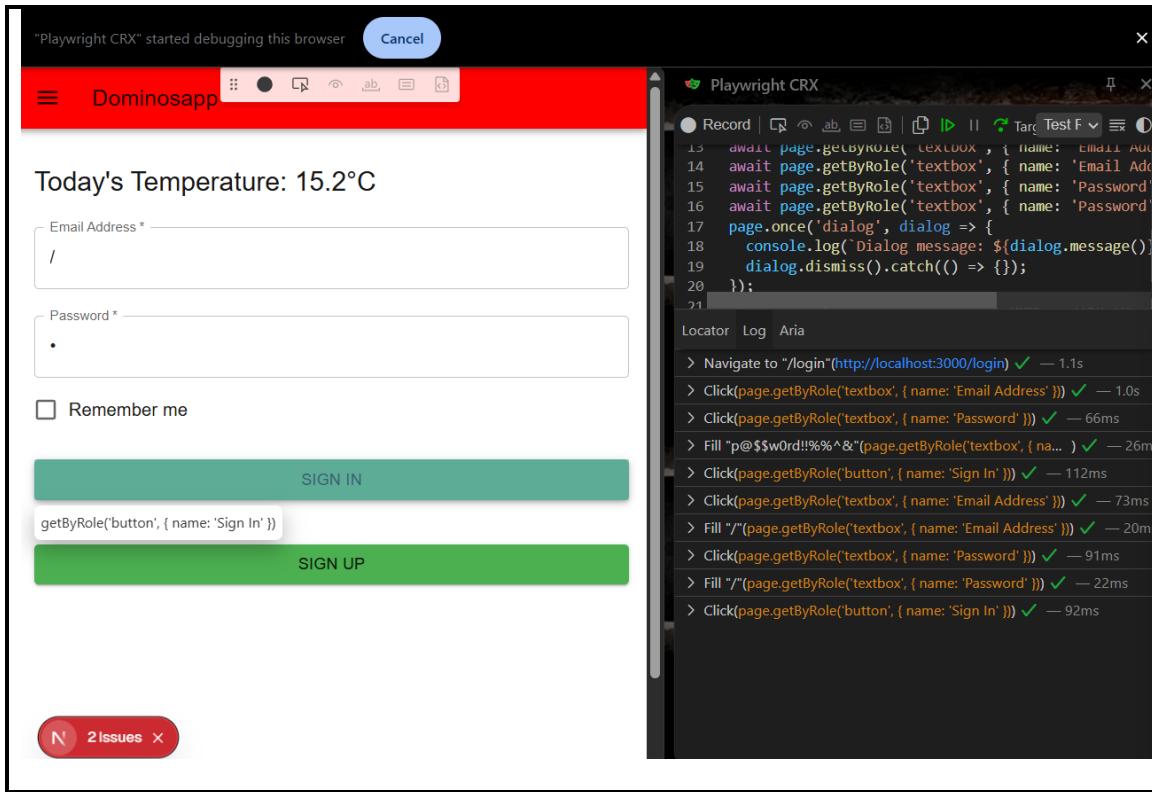
What are you testing: Empty username and password.

Input: "" / ""

Expected Result: Validation error, login fails.

Result: passed as both name and password

## Test JavaScript screenshot:



## Test ID: 10

**URL being tested: customer-dashboard**

What are you testing: Access control.

Input: Logged in as Customer visit customer-dashboard

Expected Result: passed.

Result: passed

### Test JavaScript screenshot:

The screenshot shows a browser window titled "Dominosapp". Inside, there's a login form with the heading "Today's Temperature: 15.2°C". The form has an "Email Address" field containing "egg@gmail.ie", a "Password" field (partially visible), and a "Remember me" checkbox. Below the form are "SIGN IN" and "SIGN UP" buttons. To the right of the browser is the "Playwright CRX" extension interface, which displays recorded JavaScript code for navigating and filling the form.

```
4 await page.getByRole('textbox', { name: 'Email Address' });
5 await page.getByRole('textbox', { name: 'Email Address' });
6 await page.getByRole('textbox', { name: 'Password' });
7 await page.getByRole('button', { name: 'Sign In' });
8 await page.getByRole('button', { name: 'Logout' });
9 await page.getByRole('textbox', { name: 'Email Address' });
10});
```

Locator Log Aria

- > Click(page.getByRole('textbox', { name: 'Email Address' })) ✓ — 68ms
- > Fill "egg@gmail.com"(page.getByRole('textbox', { name: ... })) ✓ — 33ms
- > Click(page.getByRole('textbox', { name: 'Password' })) ✓ — 75ms
- > Click(page.getByRole('button', { name: 'Sign In' })) ✓ — 113ms
- > Click(page.getByRole('button', { name: 'Logout' })) ✓ — 943ms
- > Fill "egg@gmail.ie"(page.getByRole('textbox', { name: '...'})) ✓ — 331ms

### Test ID: 11

**URL being tested: manger-dashboard**

What are you testing: Access control.

Input: Logged in as Manager visit manager-dashboard

Expected Result: Should pass.

Result: passed

Test JavaScript screenshot:

The screenshot shows the Playwright CRX extension running in a browser. The main window displays a login form for 'Dominosapp' with fields for 'Email Address' (containing 'egg@gmail.ie'), 'Password' (with three dots), and a 'Remember me' checkbox. Below these are 'SIGN IN' and 'SIGN UP' buttons. The top of the browser window has a red bar with the title 'Playwright CRX' started debugging this browser' and a 'Cancel' button. The bottom right corner of the browser window has a circular icon with a white letter 'N'. To the right of the browser is the DevTools panel for the 'Playwright CRX' extension. It includes a toolbar with icons for Record, Log, Aria, and various selection tools. The main pane shows a code editor with the following JavaScript code:

```
4 await page.getByRole('textbox', { name: 'Email Address' })
5 await page.getByRole('textbox', { name: 'Password' })
6 await page.getByRole('button', { name: 'Sign In' })
7 await page.getByRole('heading', { name: 'Manager Dashboard' })
8 await page.getByRole('button', { name: 'Logout' })
9 await page.getByRole('textbox', { name: 'Email Address' })
10});
```

Below the code editor, there is a 'Locator' tab and a log pane containing several entries with green checkmarks and execution times:

- > Click(page.getByRole('textbox', { name: 'Email Address' })) ✓ — 74ms
- > Click(page.getByRole('textbox', { name: 'Password' })) ✓ — 86ms
- > Click(page.getByRole('button', { name: 'Sign In' })) ✓ — 93ms
- > Click(page.getByRole('heading', { name: 'Manager Dashboard' })) ✓ — 1.1ms
- > Click(page.getByRole('button', { name: 'Logout' })) ✓ — 77ms
- > Fill "egg@gmail.ie"(page.getByRole('textbox', { name: 'Email Address' })) ✓ — 918ms

Test ID: 12

**URL being tested: /add to cart**

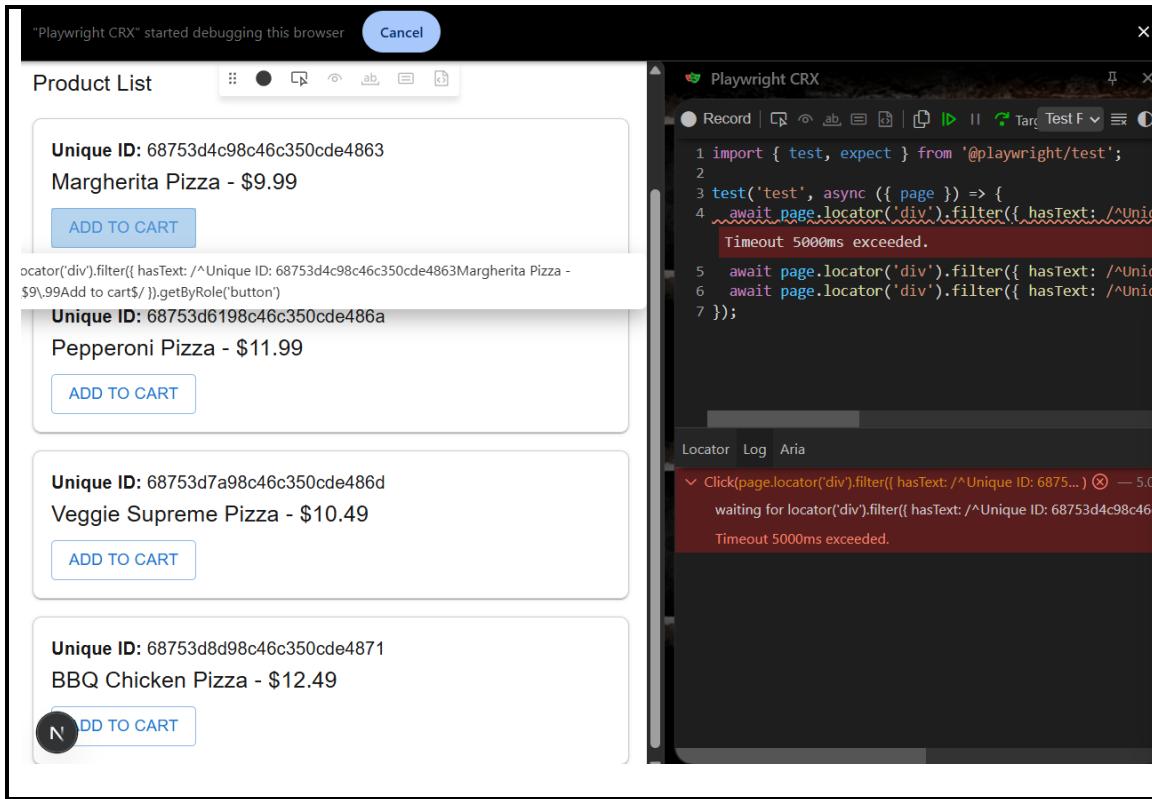
## What are you testing: button adds to cart

Input: click

Expected Result: successful

Result: failed cart is already full

## Test JavaScript screenshot:



### Test ID: 13

URL being tested: /checkout

What are you testing: checkout button

Input: click

Expected Result: pass by alert.

Result: faild no functionality after clicked

## Test JavaScript screenshot:

The screenshot shows a browser window titled "Test JavaScript screenshot:" containing a pizza ordering interface. The interface lists several pizzas with their names and prices. At the bottom, it shows a total price of \$115.89 and a blue "CONFIRM CHECKOUT" button. A "N" icon is visible above the checkout button. To the right of the browser window is an open developer tools DevTools panel titled "Playwright CRX". The DevTools panel displays a test script in the "Script" tab and a log entry in the "Log" tab.

```
import { test, expect } from '@playwright/test';
test('test', async ({ page }) => {
  await page.getByRole('button', { name: 'Cart' }).click();
  Timeout 5000ms exceeded.
  page.once('dialog', dialog => {
    console.log(`Dialog message: ${dialog.message()}`);
    dialog.dismiss().catch(() => {});
  });
  await page.getByRole('button', { name: 'Confirm Checkout' }).click();
});
```

Locator Log Aria

Click(page.getByRole('button', { name: 'Cart', exact: true })) — 5.0s  
waiting for getByRole('button', { name: 'Cart', exact: true })  
Timeout 5000ms exceeded.