Manual to Automation QA - Playwright Beginner Guide

This guide is for manual QA testers transitioning into automation using Playwright with TypeScript. It breaks down concepts in plain language and maps them directly to what you already know from manual testing.

What is Playwright?

Playwright = automation tool that controls a web browser using code.

Instead of clicking and typing with your mouse and keyboard, you write scripts that do it for you.

You're still doing QA - just faster, repeatable, and scalable.

What is a Test?

In Playwright, every test is a script that tells the browser what to do and what to check.

Example:

```
test('check example.com', async ({ page }) => {
  await page.goto('https://example.com');
  await expect(page).toHaveTitle(/Example Domain/);
});
```

What is a Locator?

Locators tell Playwright how to find something on the screen.

Examples:

```
page.getByRole('button', { name: 'Submit' });

page.getByPlaceholder('Full Name');

page.locator('#email');
```

What Are Assertions?

Assertions = how you check if something is correct.

Examples:

```
expect(page).toHaveURL(/\/dashboard/);

expect(page.locator('#output')).toBeVisible();

expect(page.locator('#email')).toContainText('jacob@example.com');
```

What Jacob Has Built So Far

- Page Navigation
- Element Interaction
- Locators: getByRole, locator, getByPlaceholder
- Assertions: toBeVisible, toHaveURL, toContainText
- Negative Tests
- Hooks (beforeEach)
- GitHub Repo
- Documentation (README.md, this guide)

Manual QA -> Automation Mapping

Manual Testing	=>	Playwright Automation
Open browser	=>	page.goto()
Click a button	=>	click()
Type in a field	=>	fill()
Look for a message	=>	toBeVisible()
Check the URL	=>	toHaveURL()
Retest manually	=>	Run with npx playwright test
Write test steps	=>	Write steps in code

Interview Talking Point

"I've built a real automation suite in Playwright using TypeScript. I've automated page navigation, form submission, and validation using real-world selectors and assertions. I've used hooks for structure, pushed to GitHub, and I understand how to read and debug locator failures."

This is now 100% true for Jacob.