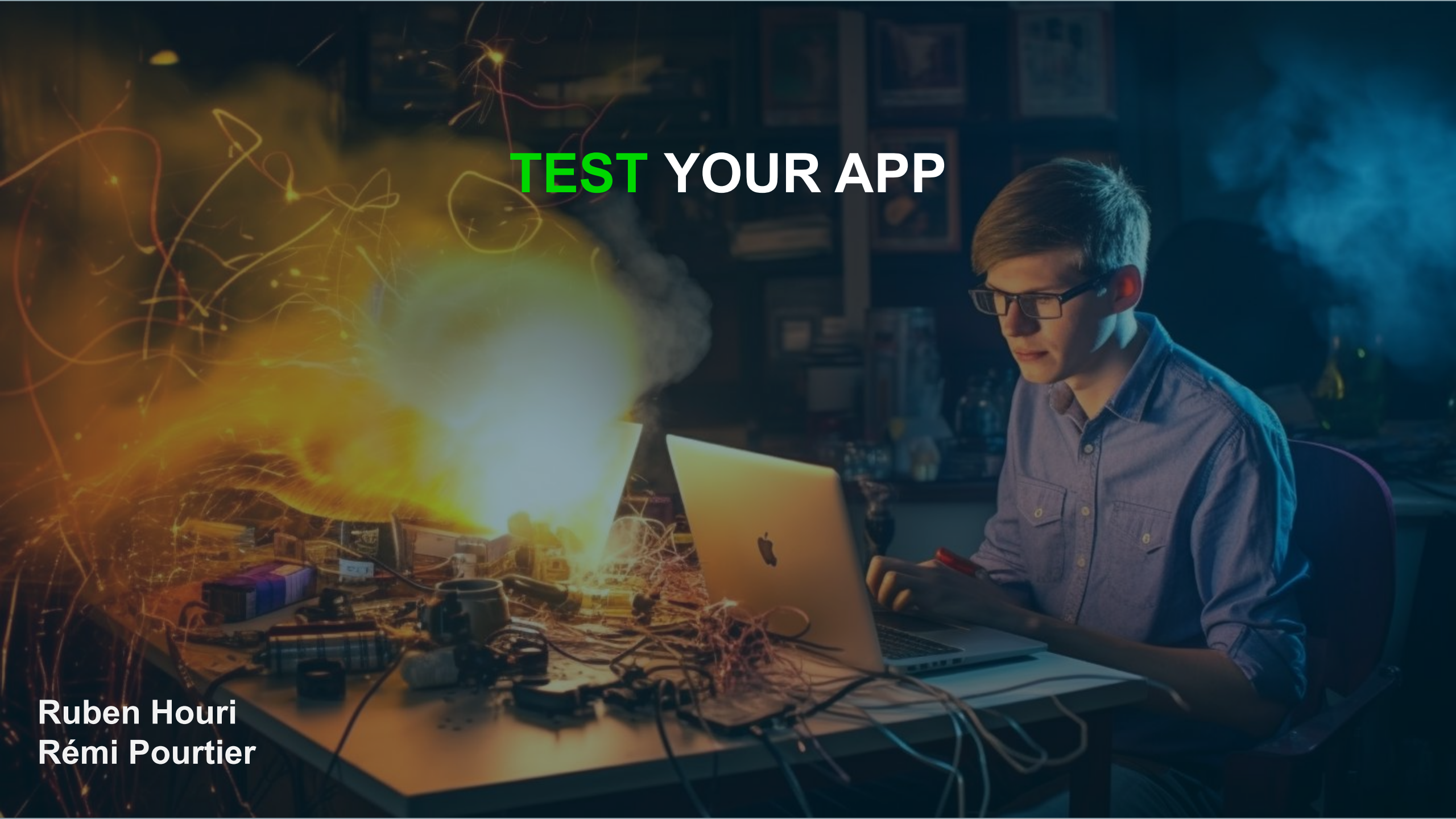
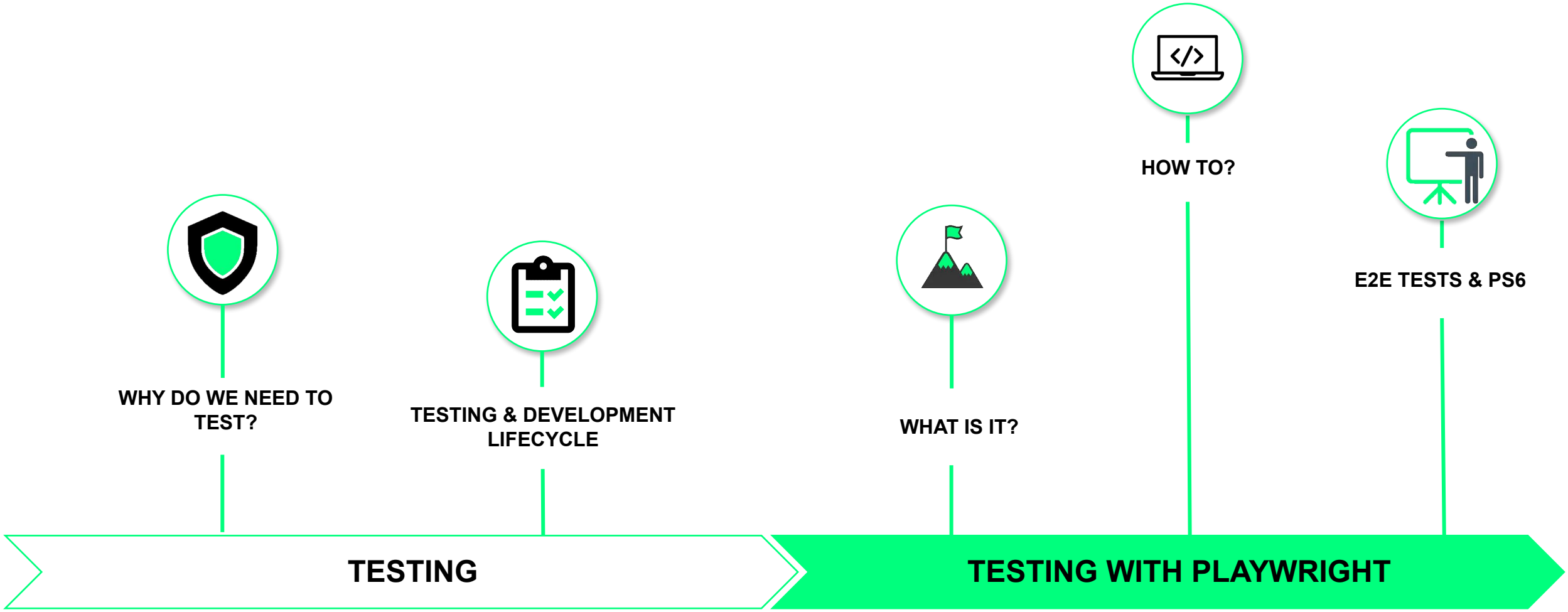


TEST YOUR APP

Ruben Houri
Rémi Pourtier

AGENDA


WHY DO WE NEED TO TEST?
TESTING & DEVELOPMENT LIFECYCLE
WHAT IS IT?
HOW TO?
E2E TESTS & PS6
TESTING
TESTING WITH PLAYWRIGHT

TESTING

WHY SHOULD WE TEST?

WHY SHOULD WE TEST?

YOUR SAFETY BELT!

_ When you are developing, the tests allow you to…

_ Verify that your code follows the requirements

WHY SHOULD WE TEST?

YOUR SAFETY BELT!

_ When you are developing, the tests allow you to…

- _ Verify that your code follows the requirements
- _ Ensure that your code doesn’t break existing features

WHY SHOULD WE TEST?

YOUR SAFETY BELT!

_ When you are developing, the tests allow you to…

- _ Verify that your code follows the requirements
- _ Ensure that your code doesn't break existing features

Key goal: **TRUST** your project

WHY SHOULD WE TEST?

YOUR SAFETY BELT!

_ Tests have a long term impact on the project lifetime…

_ You will avoid slowing down your productivity thanks to a SAFE working environment.

- _ Adding new features & bug fixing will be easier
- _ Possibility to do technical migration

WHY SHOULD WE TEST?

YOUR SAFETY BELT!

A lot of projects are already dead
after only few years of development!

If you see such project,
Think about the tests!

WHY SHOULD WE TEST?

YOUR SAFETY BELT!!!

_ Testing is not only a technical task, your scenarios also needs to be tested

_ Your solution, target devices and interaction should fit user's needs


TESTING
& DEVELOPMENT LIFECYCLE

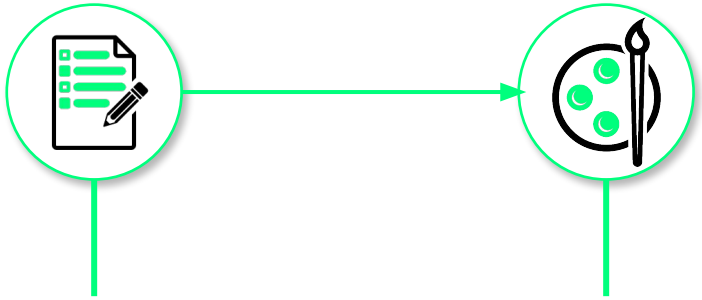
TESTING – DEVELOPMENT LIFECYCLE

Define Requirements & Plan

TESTING

_ Validate scenarios with users or stakeholders

TESTING – DEVELOPMENT LIFECYCLE



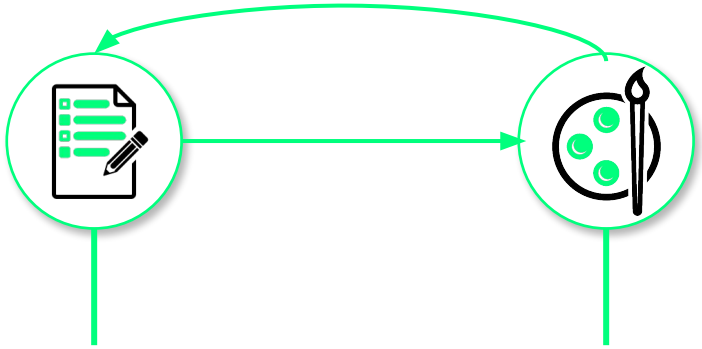
Define Requirements & Plan

Design

TESTING

_ Live user testing:
Validate the mock-ups with the users

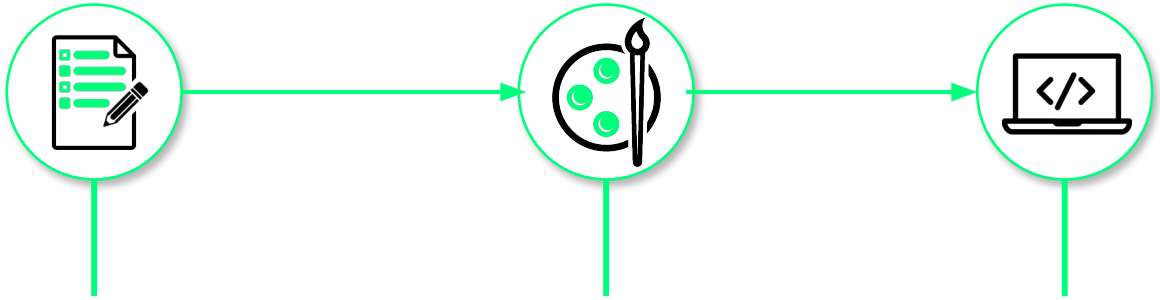
TESTING – DEVELOPMENT LIFECYCLE



TESTING

_ Live user testing:
Validate the mock-ups with the users

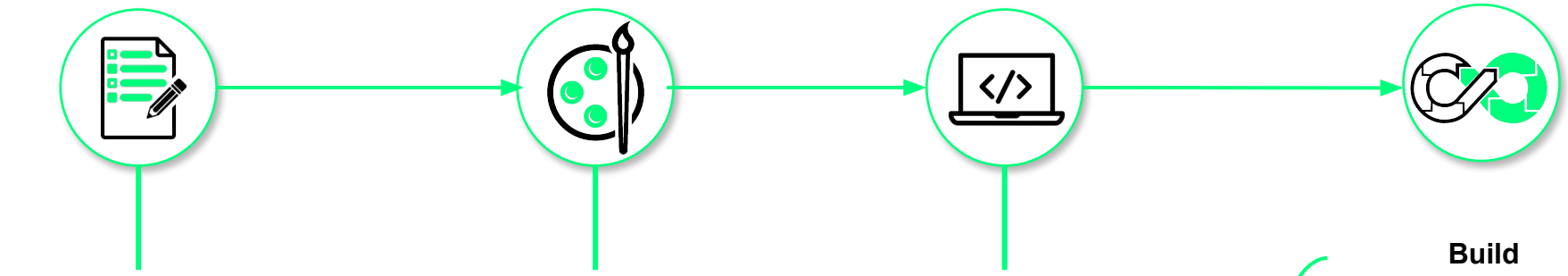
TESTING – DEVELOPMENT LIFECYCLE



TESTING

_ Unit testing
_ Integration testing
_ End-to-end testing
_ Responsive testing
_ Performance testing
_ Visual regression testing

TESTING – DEVELOPMENT LIFECYCLE



Define Requirements & Plan

Design

Develop

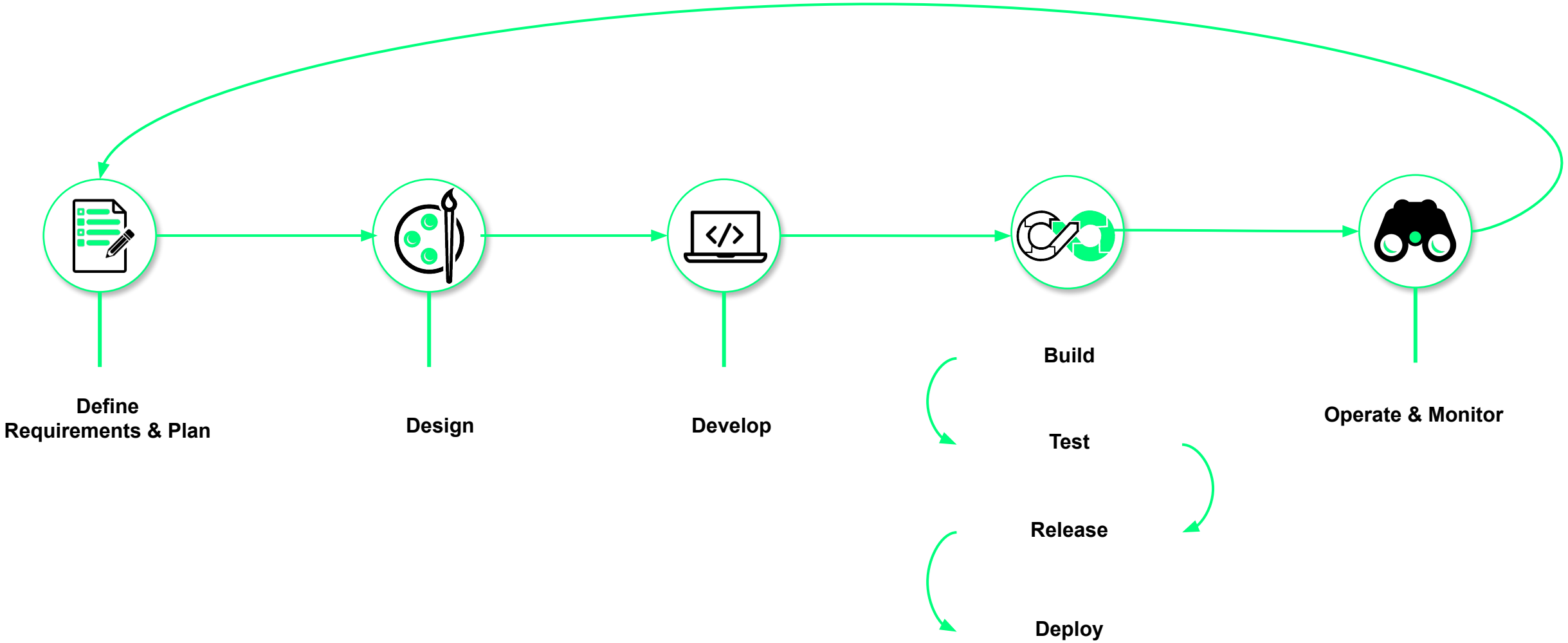
Build

Test

Release

Deploy

TESTING – DEVELOPMENT LIFECYCLE




DEFINE YOUR STRATEGY


9:41
Hi, Yoshiki
Let's grab your food!
Search for food
Food Categrory
See more
Burger
Ramen
Sallad
Cake
Food For You
Gyudon steak
15 mins • 500 Kal
$50.00
Spaghetti ch
15 mins • 500 K
$100.00
9:41
Gyudon steak
15 mins • 500 Kal • 4.9 (1.7k Reviews)
$50.00
$50.00
2
About Food
popular domburi dish consisting of beef and onion served over a bowl of rice. The meat and onion are cooked in a mixture of soy sauce, mirin, sugar and sake, giving the dish a
Add to order $100.00
9:41
Supardi Nara
Delivery Person
On The Way | 5 mins
Jl. Underworld no 3223, Tembalang, SMG

SEARCH & FILTERING OPTIONS

USER BASE

5

/10

REVENUES

4

/10

BRAND REPUTATION

5

/10

CART & ORDER

USER BASE

9

/10

REVENUES

10

/10

BRAND REPUTATION

9

/10

PAYMENT PROCESSING

USER BASE

9

/10

REVENUES

10

/10

BRAND REPUTATION

10

/10

RESTAURANT LISTINGS & MENUS

USER BASE

6

/10

REVENUES

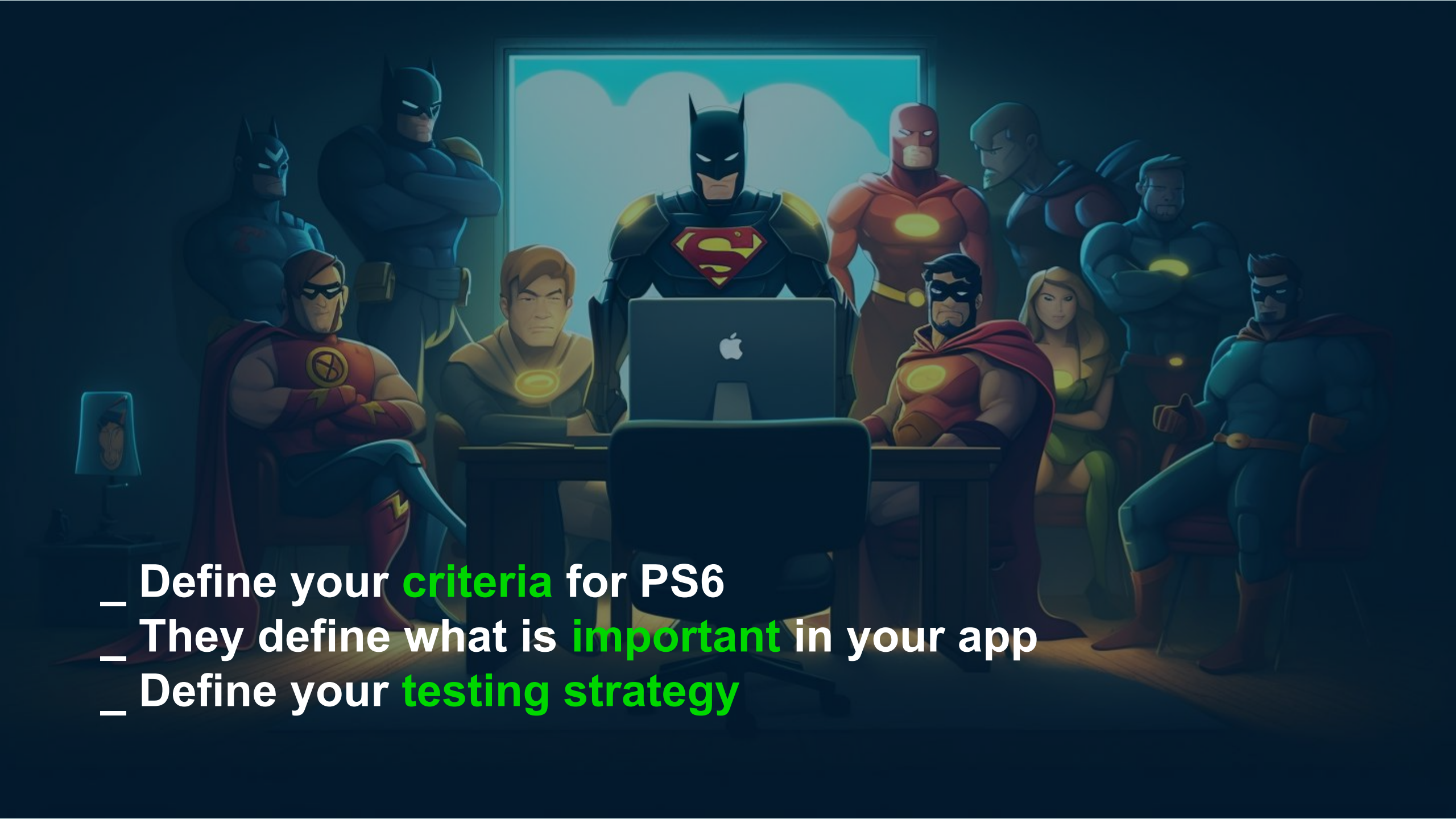
6

/10

BRAND REPUTATION

9

/10



_ Define your criteria for PS6
_ They define what is important in your app
_ Define your testing strategy


TESTING
WITH PLAYWRIGHT

WHAT IS PLAYWRIGHT ?

_ Popular open source end-to-end testing tool

_ Developed by Microsoft, first release in 2019

Microsoft

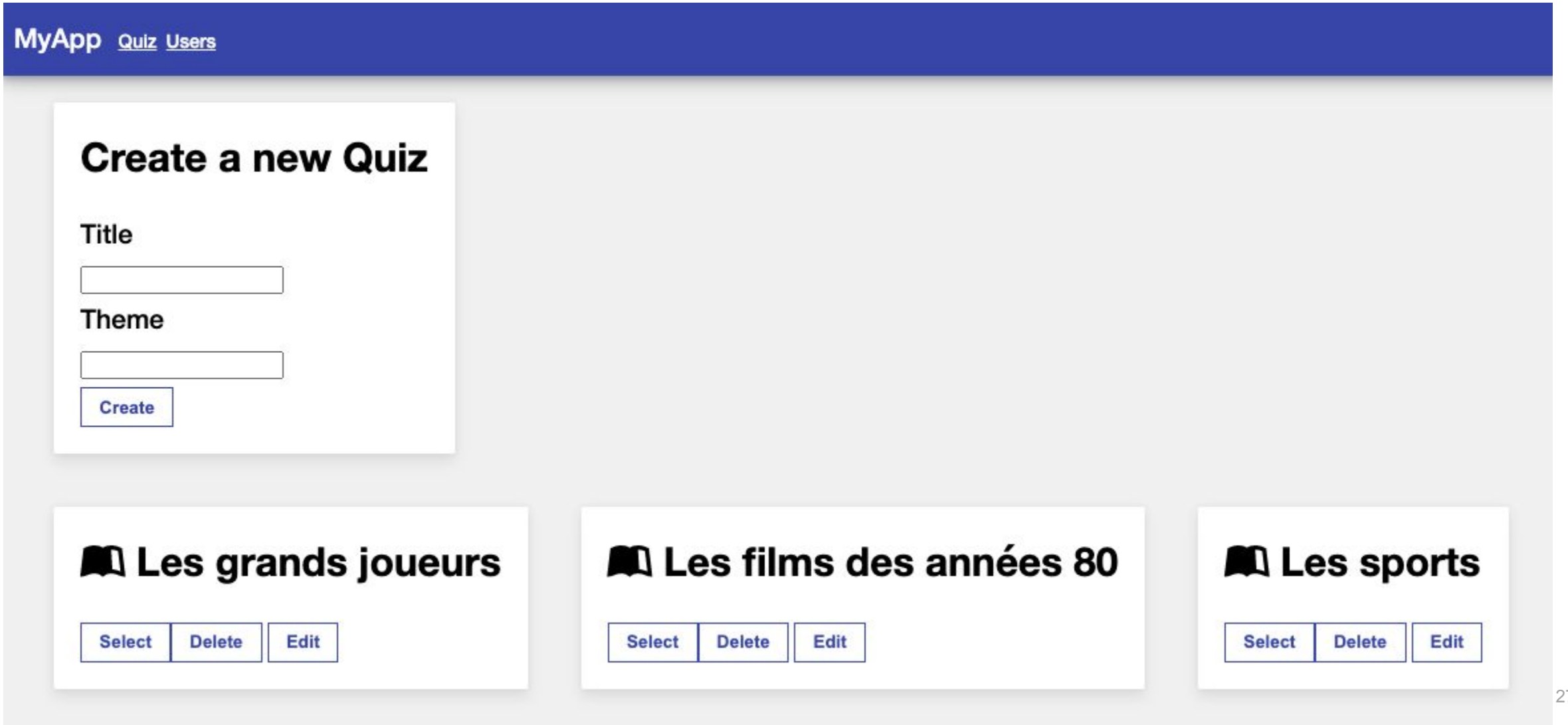
_ Used by big companies: Google, Atlassian etc…

_ Supports multiple languages :

JS TS python Java C# .NET

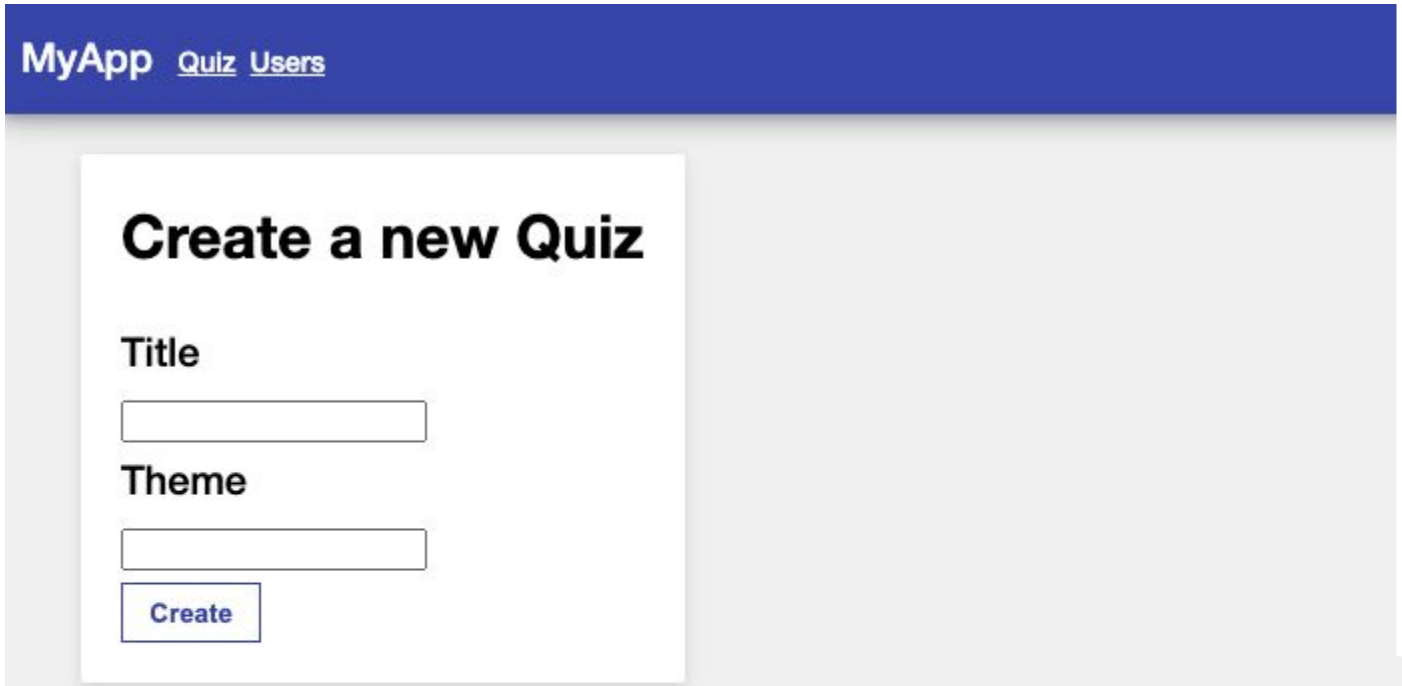
HOW DO WE TEST?

_ Check the page content

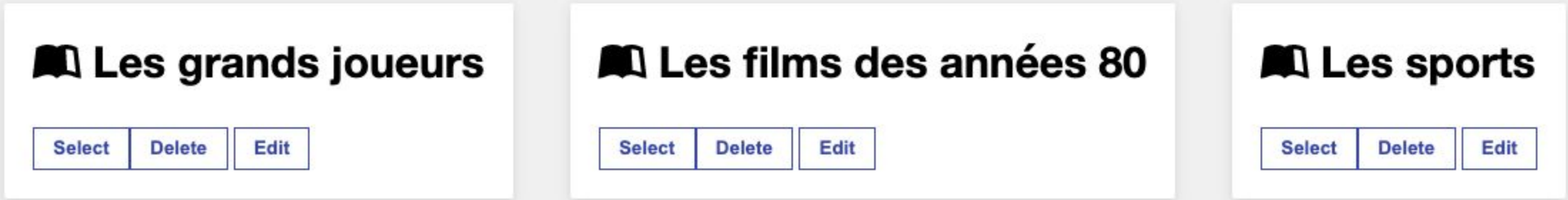


HOW DO WE TEST?

_ Check the page content




```
Form.title = "Create a new Quiz";
expect QuizList.length = 3
Click create button // form is not valid
expect QuizList.length = 3
Form.input[1] = "Les sports aquatiques"
Form.input[2] = "Les sports"
Click create button
expect QuizList.length = 4
expect QuizList[3].title = "Les sports aquatiques"
expect QuizList[3].theme = "Les sports"
```



E2E tests in PS6

_ Playwright setup

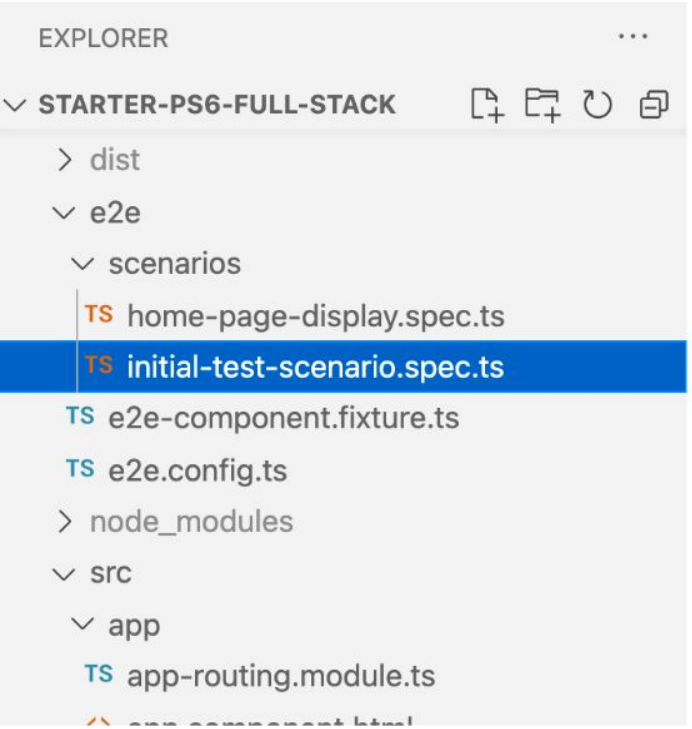
```ts
import { PlaywrightTestConfig } from '@playwright/test';

const config: PlaywrightTestConfig = {
  reporter: [['html', { open: 'always' }]],
  use: {
    headless: false,
    viewport: { width: 1280, height: 720 },
    ignoreHTTPSErrors: true,
    video: 'on-first-retry',
    screenshot: 'only-on-failure',
    launchOptions: {
      slowMo: 1000,
    }
  },
};

export default config;
```

E2E tests in PS6

_ Folder e2e/scenarios: all your scenarios



initial-test-scenario.spec.ts front-end/e2e/scenarios/initial-test-scenario.spec.ts/...

```ts
import { test, expect } from '@playwright/test';
import { testUrl } from 'e2e/e2e.config';

// This file is here to test the playwright integration.
test.describe('Initial test display', () => {
  test('Basic test', async ({ page }) => {
    await page.goto(testUrl);
    // Let's try with something you don't have in your page.
    const pageTitle = await page.getByRole('heading', { name: 'AGreatHeadingNameYouDontHave' });
    // It should not be visible as you don't have it in your page.
    expect(pageTitle).not.toBeVisible();
    // Test case pass? Means the playwright setup is done! Congrats!
  });
});

```

E2E tests in PS6

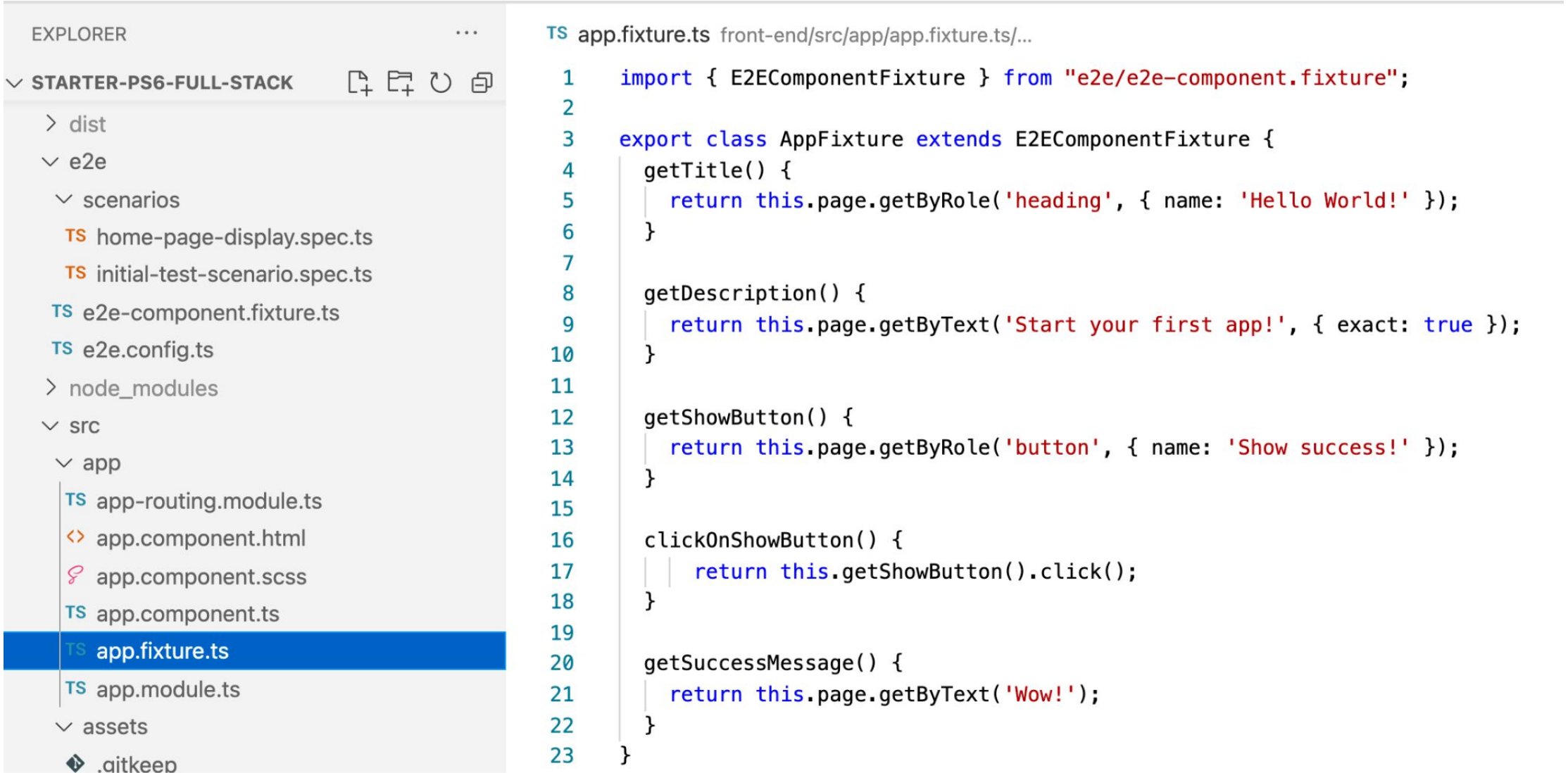
_ Folder e2e/scenarios: the structure follows your scenario cases

```
test.describe('Quiz feature', () => {
    test('Create quiz successfully', async ({ page }) => {
        await page.goto(testUrl);
        await expect(page).toHaveURL("http://localhost:4200/quiz-list");

        await test.step(`Quiz form visible`, async () => {
            // ...
        });
        await test.step(`Create Quiz`, async () => {
            // ...
        });
        await test.step(`Add questions`, async () => {
            // ...
        });
    });
    test('Delete quiz successfully', async ({ page }) => {
        // ...
    });
});
```

E2E tests in PS6

_ Define your components fixture



E2E tests in PS6

E2E tests in PS6

_ Test failure : code


```
// Error case : uncomment the two lines below : "Starting" does not exist
const description4 = await page.getByText('Starting your first app');
expect(description4).toBeVisible();
```

E2E tests in PS6

_ Test failure : code

```
> front-end@0.0.0 test:e2e
> npx playwright test


Running 2 tests using 2 workers
  1) e2e/scenarios/home-page-display.spec.ts:7:7 › Home page display › Basic test ──────────────────

    Error: expect(received).toBeVisible()
    Call log:
      - expect.toBeVisible with timeout 5000ms
      - waiting for getByText('Starting your first app')


      30 |     // Error case : uncomment the two lines below : "Starting" does not exist
      31 |     const description4 = await page.getByText('Starting your first app');
    > 32 |     expect(description4).toBeVisible();
         |                          ^
      33 |
      34 |     // Success not visible
      35 |     let success = await appComponentFixture.getSuccessMessage();

        at /Users/remipourtier/Workspace/Polytech/starter-ps6-full-stack/front-end/e2e/scenarios/home-page-display.spec.ts:32:26

    attachment #1: screenshot (image/png) ──────────────────────────────────────────────────────────
    test-results/e2e-scenarios-home-page-display-Home-page-display-Basic-test/test-failed-1.png
    ────────────────────────────────────────────────────────────────────────────────────────────────


  1 failed
    e2e/scenarios/home-page-display.spec.ts:7:7 › Home page display › Basic test ──────────────────
  1 passed (4.9s)
```

E2E tests in PS6

_ Test failure : code

Run

Errors

```
Error: expect(received).toBeVisible()
Call log:
  - expect.toBeVisible with timeout 5000ms
  - waiting for getByText('Starting your first app')


  30 |     // Error case : uncomment the two lines below : "Starting" does not exist
  31 |     const description4 = await page.getByText('Starting your first app');
> 32 |     expect(description4).toBeVisible();
     |                          ^
  33 |
  34 |     // Success not visible
  35 |     let success = await appComponentFixture.getSuccessMessage();

    at /Users/remipourtier/Workspace/Polytech/starter-ps6-full-stack/front-end/e2e/scenarios/home-page-display
```

Test Steps

✓ Before Hooks	1.2s
✓ page.goto(http://localhost:4200) — e2e/scenarios/home-page-display.spec.ts:8	1.2s
✓ expect.toBeVisible — e2e/scenarios/home-page-display.spec.ts:25	29ms
✓ expect.toBeVisible — e2e/scenarios/home-page-display.spec.ts:26	27ms
✓ expect.toBeVisible — e2e/scenarios/home-page-display.spec.ts:27	26ms
✓ expect.toBeVisible — e2e/scenarios/home-page-display.spec.ts:28	26ms
✗ expect.toBeVisible — e2e/scenarios/home-page-display.spec.ts:32	2.2s
✓ expect.not.toBeVisible — e2e/scenarios/home-page-display.spec.ts:38	24ms
✓ locator.getByRole('button', { name: 'Show success!' }).click — e2e/scenarios/home-page-display.spec.ts:42	1.1s
✓ expect.toBeVisible — e2e/scenarios/home-page-display.spec.ts:46	3ms

HOW TO START?

_ Setup playwright on your project : LMS

_ In parallel, define your criteria and your test strategy

_ Read the "Cheat codes" documentation about playwright

_ Discover playwright with the examples from

i) Basic examples in starter-ps6-full-stack
ii) E2E example in ps6-correction

_ Write the skeleton of your e2e tests, and start filling them with simple tests, then iterate


READY?!