OPENCLASSROOMS

Corrigé

P1C6 - Tests E2E avec Playwright

Le scénario peut être construit de la manière suivante :

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


test(
 "un utilisateur se déconnecte",
 async ({ page }) => {


   await page.goto("/login");


   await page
     .getByLabel("Email")
     .fill("test@example.com");


   await page
     .getByLabel("Mot de passe")
     .fill("secret");
```

```
  await page
    .getByRole("button", {
      name: /se connecter/i
    })
    .click();

  await expect(
    page.getByText("Mes tâches")
  ).toBeVisible();

  await page
    .getByRole("button", {
      name: /se déconnecter/i
    })
    .click();

  await expect(page)
    .toHaveURL("/login");

 }
);
```

Le test vérifie d'abord que l'utilisateur accède bien à son espace personnel grâce à l'assertion portant sur le texte « Mes tâches », titre de la section qui liste les tâches. Comme cette section ne s'affiche qu'une fois les tâches chargées, la web-first assertion attend automatiquement son apparition : cet

état intermédiaire confirme à la fois que l'authentification a réussi et que la page est chargée avant de poursuivre le scénario.

Pour provoquer un échec volontaire, vous pouvez modifier l'assertion finale :

```typescript
await expect(page)

 .toHaveURL("/tasks");
```

Le scénario échouera alors puisque la déconnexion redirige normalement vers `/login`.

En ouvrant le Trace Viewer, vous pourrez visualiser précisément la navigation réalisée par le navigateur et constater à quel moment la redirection s'est produite. Cette analyse vous permettra d'identifier rapidement l'origine du problème sans avoir à modifier le code à l'aveugle.

Avec ce dernier scénario, votre équipe dispose désormais d'une couverture de test allant du niveau unitaire jusqu'au parcours utilisateur complet.