Réponses au TP4 Playwright

Partie A : Installation et premiers pas

Pourquoi tester sur plusieurs navigateurs ?
Tous les navigateurs ne réagissent pas de la même façon. Tester sur Chrome, Firefox et Safari permet de s'assurer que notre site fonctionne correctement pour tout le monde.

Comment Playwright gère les temps d'attente ?
Playwright attend automatiquement que les éléments soient prêts avant de cliquer ou remplir un champ. Du coup, on n'a pas besoin de mettre des pauses manuelles partout et les tests sont plus fiables.

Partie B : Interactions complexes et assertions

Gérer les éléments qui apparaissent ou disparaissent ?
On utilise des fonctions comme waitForSelector ou des assertions comme toHaveCount pour s'assurer que l'élément est bien là ou disparu avant de continuer.

Quand utiliser toHaveText ou toContainText ?

- toHaveText : quand on sait exactement ce que l'élément doit afficher.
- toContainText : quand le texte peut varier ou contenir des infos supplémentaires.

Les pièges avec les frames et comment les éviter ?
Les frames ne sont pas toujours prêtes immédiatement. Il faut attendre qu'elles soient chargées avant d'interagir, sinon le test plante.

Rendre les tests stables malgré animations ou temps de chargement ?
Toujours attendre que les éléments soient visibles et interactifs. Playwright gère déjà beaucoup de cette logique automatiquement.

Partie C : Page Object Model et Mock API

Pourquoi le Page Object Model est utile ?
Il rend le code plus propre et facile à maintenir. Si un champ ou un bouton change, on modifie juste la page correspondante et tous les tests s'adaptent.

Comment les tests paramétrés aident ?
Ils permettent de tester plusieurs scénarios avec le même test, ce qui évite de répéter du code et améliore la couverture.

Pourquoi utiliser le mocking d'API ?
Ça permet de tester des cas précis sans dépendre de l'API réelle. Même si l'API n'est pas prête ou lente, on peut vérifier que tout fonctionne.

Comment vérifier que tout fonctionne avec des données simulées ?
On s'assure que les éléments affichent correctement les données mockées et que toutes les interactions se passent comme prévu.

Partie D : Debugging, Reporting et CI/CD

Comment analyser un fichier de trace ?
Playwright fournit toutes les étapes du test avec captures d'écran et logs. C'est pratique pour comprendre pourquoi un test a échoué.

Quels avantages la CI apporte ?
La CI lance automatiquement les tests à chaque modification du code. Ça permet de détecter les bugs tôt et de garder le projet stable.