

TP3 : Automatisation des tests de la calculatrice avec Selenium WebDriver

Rendu :

Le rendu se fera à la fin de la formation, il s'agira d'un rapport en format PDF.

Il y aura dans le rapport une section par TP, chaque section aura 5 sous-sections :

- Objectifs
- Démarche
- Vos résultats
- Les difficultés rencontrés
- Conclusion

Tous vos choix devront être justifiés.

Utiliser judicieusement des captures d'écran pour illustrer vos propos.

Quand c'est approprié, pensez à mettre des liens vers vos travaux.

Travail à effectuer :

Configuration de l'environnement de test :

Installer Selenium WebDriver et le navigateur web driver correspondant (par exemple, ChromeDriver).

Commencez avec "base.html".

Gestion des navigateurs :

Maximiser la fenêtre du navigateur avant d'exécuter les tests.

Écriture des cas de test :

Écrire des cas de test pour couvrir différentes fonctionnalités de la calculatrice, y compris l'addition, la soustraction, la multiplication, la division, l'effacement de l'écran, etc.

Mettez en place le plus de tests possibles.

Détaillez ces tests dans le rapport.

Utilisation des attentes explicites et implicites :

Utiliser des attentes explicites pour attendre que des éléments spécifiques soient visibles ou cliquables avant d'interagir avec eux.

Utiliser des attentes implicites pour attendre que la page soit complètement chargée avant d'exécuter les actions.

Captures d'écran à des étapes clés :

Prendre des captures d'écran à des étapes clés des tests, par exemple avant et après chaque opération, pour vérifier visuellement les résultats.

Mettez dans le rapport le code qui s'occupe de cette partie.

Dans le rapport, signaler clairement quand les images utilisées ont été générées par votre code.

Exécution des tests :

Exécuter les tests automatisés localement sur la machine de développement.

Écrire des rapports de test détaillés pour documenter les résultats des tests.

Détection et gestion des erreurs :

Gérer les erreurs et les exceptions de manière appropriée dans les tests.

Ajouter des mécanismes de gestion des erreurs pour récupérer les tests en cas d'échec.

Avec "alternative.html"***Analyse des tests existants :***

Examiner les cas de test existants pour identifier ceux qui peuvent être réutilisés dans la nouvelle version de la calculatrice.

Identifier les tests qui sont spécifiques à des fonctionnalités qui ont été modifiées ou supprimées dans la nouvelle version et décider s'ils doivent être mis à jour, supprimés ou remplacés.

Refactoring des tests existants :

Modifier les sélecteurs d'éléments et les actions dans les tests pour s'aligner sur la nouvelle interface utilisateur de la calculatrice.

Réutilisation des méthodes de test :

Identifier les méthodes de test qui peuvent être réutilisées dans la nouvelle version de la calculatrice, telles que les méthodes de saisie de chiffres et d'opérations mathématiques. Créer des méthodes de test génériques et réutilisables qui peuvent être utilisées pour effectuer des actions communes sur la calculatrice, indépendamment de la version.

Tests de régression :

Exécuter les tests existants sur la nouvelle version de la calculatrice pour vérifier si les fonctionnalités existantes sont toujours opérationnelles et si les modifications n'ont pas introduit de régressions.

Avec "final.html"***Analyse des tests existants :***

Examiner les cas de test existants pour identifier ceux qui peuvent être réutilisés dans la nouvelle version de la calculatrice.

Identifier les tests qui sont spécifiques à des fonctionnalités qui ont été modifiées ou supprimées dans la nouvelle version et décider s'ils doivent être mis à jour, supprimés ou remplacés.

Refactoring des tests existants :

Adapter les cas de test existants pour prendre en charge les nouvelles fonctionnalités de la calculatrice, telles que l'utilisation de parenthèses.

Modifier les sélecteurs d'éléments et les actions dans les tests pour s'aligner sur la nouvelle interface utilisateur de la calculatrice.

Réutilisation des méthodes de test :

Identifier les méthodes de test qui peuvent être réutilisées dans la nouvelle version de la calculatrice, telles que les méthodes de saisie de chiffres et d'opérations mathématiques. Créer des méthodes de test génériques et réutilisables qui peuvent être utilisées pour effectuer des actions communes sur la calculatrice, indépendamment de la version.

Extension des tests existants :

Étendre les cas de test existants pour couvrir les nouvelles fonctionnalités et les scénarios d'utilisation de la calculatrice, y compris l'utilisation des parenthèses et les modifications de l'interface utilisateur.

Tests de régression :

Exécuter les tests existants sur la nouvelle version de la calculatrice pour vérifier si les fonctionnalités existantes sont toujours opérationnelles et si les modifications n'ont pas introduit de régressions.

Tests d'acceptation :

Écrire de nouveaux tests d'acceptation pour valider les fonctionnalités spécifiques de la nouvelle version de la calculatrice, en mettant l'accent sur les fonctionnalités ajoutées telles que l'utilisation de parenthèses.