

Campagne de Tests Automatisés pour une Application E-Commerce

Une approche structurée combinant Selenium, Python et l'IA générative.

Réalisé par :

Med Baha Chekir
Hamdi Abdelkader

Faculté des Sciences de Sfax | Département Informatique
Enseignant : M. Taher Labidi
Année Universitaire : 2025-2026



Le Défi : Contexte et Objectifs du Projet

Les Objectifs

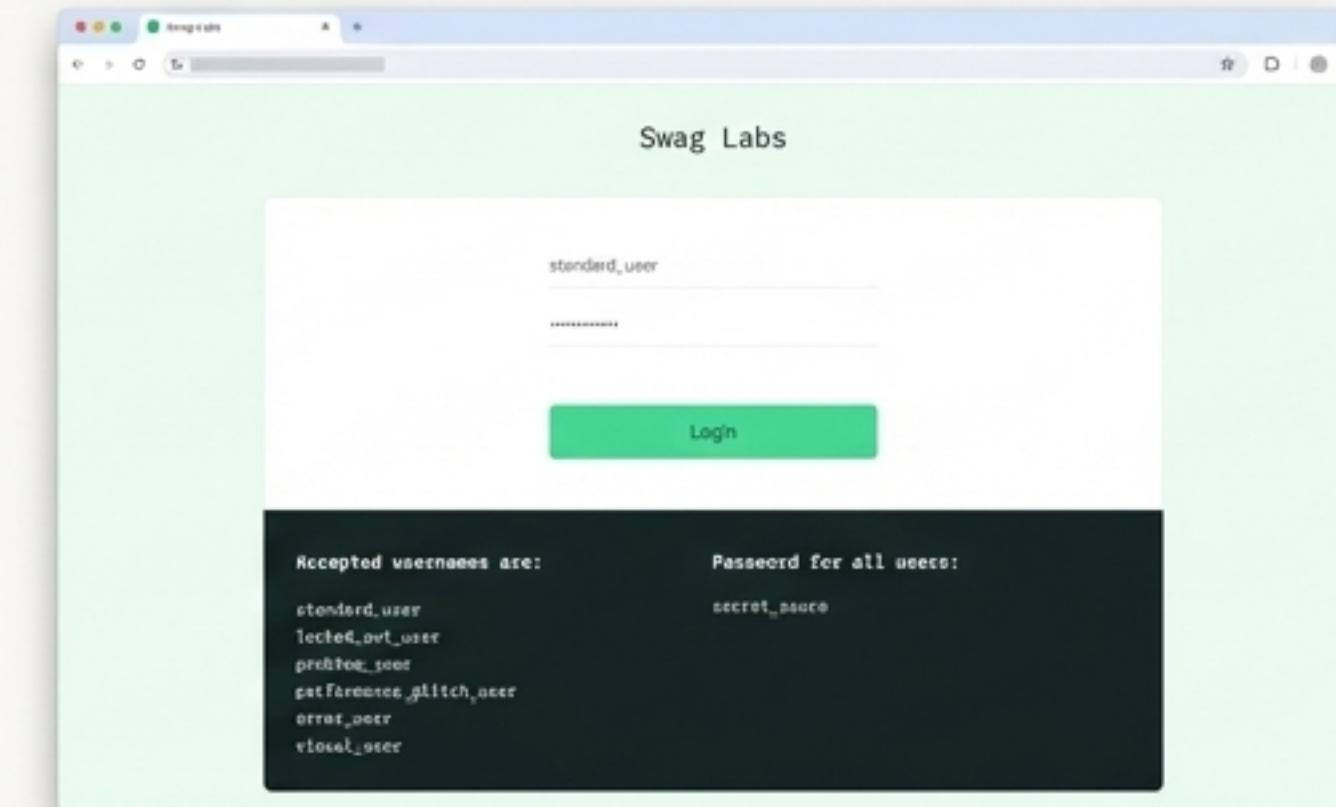
Valider les compétences du syllabus ISTQB.

- **Conception:** Élaborer des cas de tests complets (positifs, négatifs, limites).
- **Automatisation:** Implémenter une campagne de tests robuste avec Selenium WebDriver et Python.
- **Innovation:** Explorer l'apport de l'IA générative pour la création de scénarios.
- **Reporting:** Produire un rapport d'exécution et une gestion des anomalies rigoureux.

L'Application Sous Test (AUT)

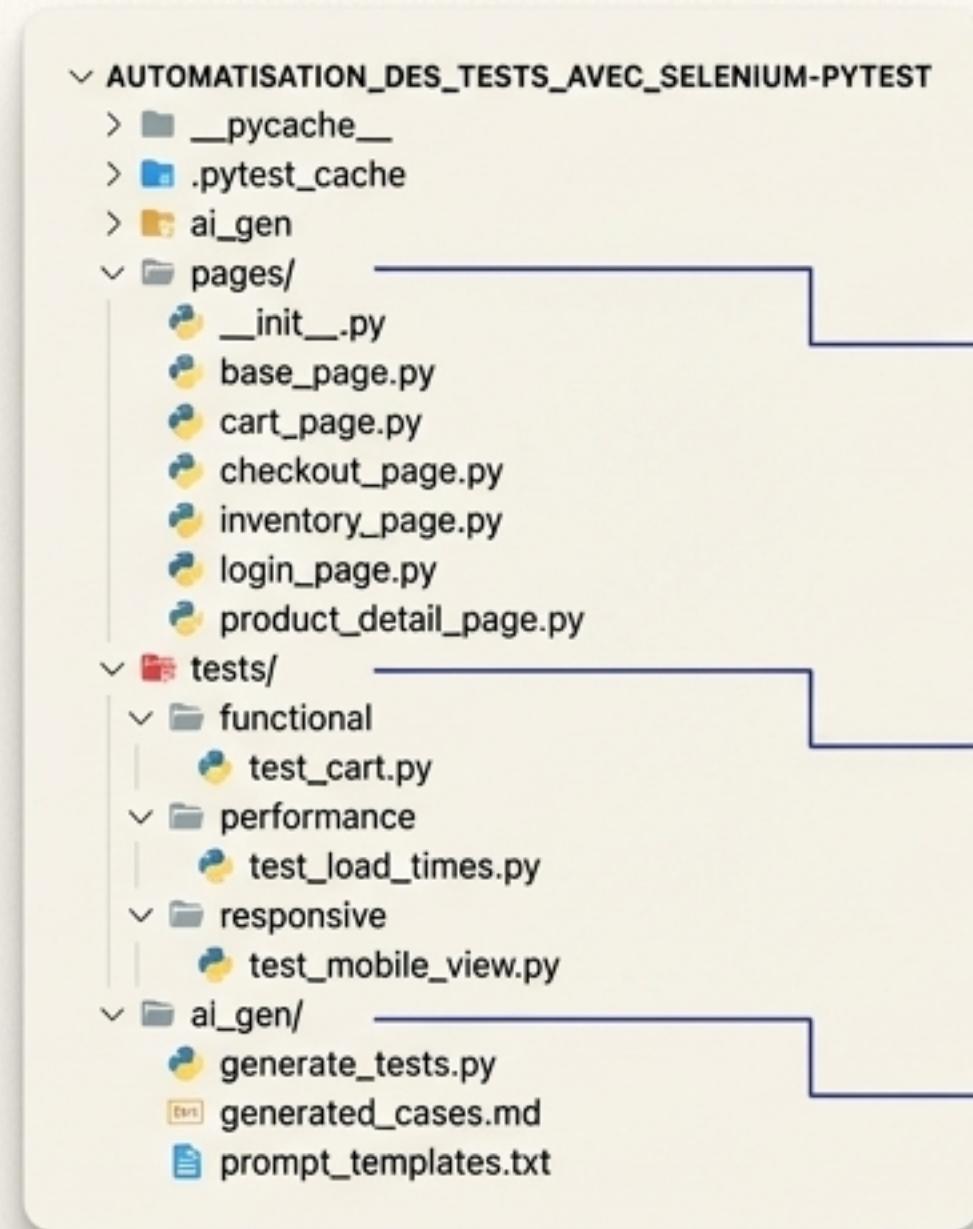
Swag Labs : Un environnement de test e-commerce.

L'application cible est 'Swag Labs' (saucedemo.com), un site "bac à sable" simulant un parcours client complet : Authentification → Sélection de produits → Panier → Paiement.



L'Architecture : Un Cadre Robuste et Maintenable

Le projet est structuré autour du pattern de conception **Page Object Model (POM)** pour garantir la maintenabilité et la clarté du code.



pages/: Centralise les interactions avec les pages web (ex: `login_page.py`, `cart_page.py`). Chaque page est un objet.

tests/: Contient les scripts de test, organisés par catégorie (fonctionnel, performance, responsive).

ai_gen/: Isolé, ce module contient l'expérimentation de génération de tests par IA.

L'Environnement Technique

Langage: Python 3.10+

Framework: Pytest

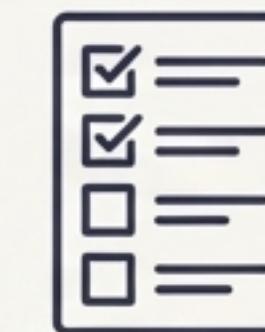
Driver: Selenium WebDriver

IDE: VS Code

Au-delà de l'Automatisation : L'Expérimentation IA

Comment l'IA générative peut-elle assister le testeur en 'Boîte Noire' ?

Objectif : Utiliser un LLM (Large Language Model) pour transformer une **User Story** métier en une matrice de **cas de tests** structurés et pertinents.



Input : User Story Métier

Engine : Module `ai_gen`
(Python + LLM API)

Output : Cas de Tests
Structurés (Markdown)

The Submitted User Story :

“En tant qu'utilisateur, je veux pouvoir ajouter des produits au panier depuis l'inventaire. Le système doit gérer les cas nominaux, les ajouts multiples et les restrictions d'accès.”

De la User Story aux Cas de Test : L'IA en Action

L'IA a généré 3 scénarios de test pertinents, incluant des cas nominaux, de sécurité et de volume.

Transcription de la sortie IA

ID	Description	Résultat Attendu Clé
TC_ADD_001	Ajout unique (Nominal)	Le compteur du panier passe à "1".
TC_ADD_002	Restriction d'accès (Sécurité)	Redirection vers la page de Login.
TC_ADD_003	Ajout total (Volume)	Le compteur du panier affiche "N" (tous les produits).

Preuve de la Génération

```
1 | En tant qu'expert en QA Automation, voici les cas de test en "Boîte Noire" basés sur votre User Story.  
2 |  
3 | 10 | Description | Pré-conditions | Étapes de test | Résultat Attendu |  
4 |-----|  
5 | TC_ADD_TO_CART_001 | Vérifier l'ajout d'un seul produit au panier, la mise à jour du bouton, l'incrémentation du compteur et le visibilité dans le panier. | 1. Un utilisateur est connecté au système. <br> 2. Le panier est vide. <br> 3. L'utilisateur est sur la page : (ligne) affiche "0" ou est absent. <br> 2. Localiser un produit spécifique (ex: "Saocer spâclifque ex: "Saocer Labs Backpack"). <br> 3. Cliquer sur le bouton "Add to cart" associé à ce produit. <br> 4. Vérifier que le bouton "Add to cart" pour ce produit change en "Remove". <br> 5. Vérifier que le compteur du panier affiche "1". <br> 6. Raviguer vers la page du panier ("cart.html"). <br> 7. Vérifier que le produit "Saocer Labs Backpack" est visible dans le panier, avec une quantité de 1. | 1. Le bouton "Add to cart" a changé en "Remove". <br> 2. Le compteur du panier affiche "1". <br> 3. Le produit est correctement listé dans le panier. |  
6 | TC_ADD_TO_CART_002 | Vérifier que l'ajout au panier est restreint pour un utilisateur non connecté. | 1. Un utilisateur n'est pas connecté au système. <br> 2. (Optionnel) L'utilisateur essaie d'accéder directement à "/inventory.html". | 1. Tenter d'accéder à la page "/inventory.html". <br> 2. Si la page est accessible, localiser un bouton et tenter de cliquer sur le bouton "Add to cart". | 1. L'utilisateur est redirigé vers la page d'erreur appropriée indiquant que l'utilisateur n'a pas de connexion. <br> 3. Le compteur du panier ne doit pas s'incrémenter. <br> 4. Le produit n'est pas ajouté au panier. |  
7 | TC_ADD_TO_CART_003 | Vérifier l'ajout de tous les produits disponibles sur la page d'inventaire au panier. | 1. Un utilisateur est connecté au système. <br> 2. Le panier est vide. <br> 3. L'utilisateur est sur la page "/inventory.html". | 1. Identifier que le compteur du panier (ligne) affiche "0" ou est absent. <br> 2. Identifier et compter tous les produits disponibles sur la page "/inventory.html". Soit 'N' le nombre total de produits. <br> 3. Chaque produit listé, cliquer sur son bouton "Add to cart". <br> 4. Après chaque clic, vérifier que le bouton "Add to cart" pour ce produit change en "Remove". <br> 5. Après avoir cliqué sur "Add to cart" pour tous les 'N' produits, vérifier que le compteur du panier affiche "N". <br> 6. Raviguer vers la page du panier ("cart.html"). <br> 7. Vérifier que tous les 'N' produits ajoutés sont présents dans le panier, chacun avec une quantité de 1. | 1. Tous les boutons "Add to cart" sont devenus "Remove". <br> 2. Le compteur du panier affiche le nombre total ("N") de produits ajoutés. <br> 3. Tous les produits ajoutés sont correctement listés dans le panier. |
```

"Key Takeaway": L'IA a identifié un cas de test de sécurité (TC_ADD_002) non prévu initialement dans la campagne manuelle.

Bilan de la Campagne de Tests

15

Exigences

Testées au total
(Fonctionnelles, Non-
Fonctionnelles, Sécurité)

12

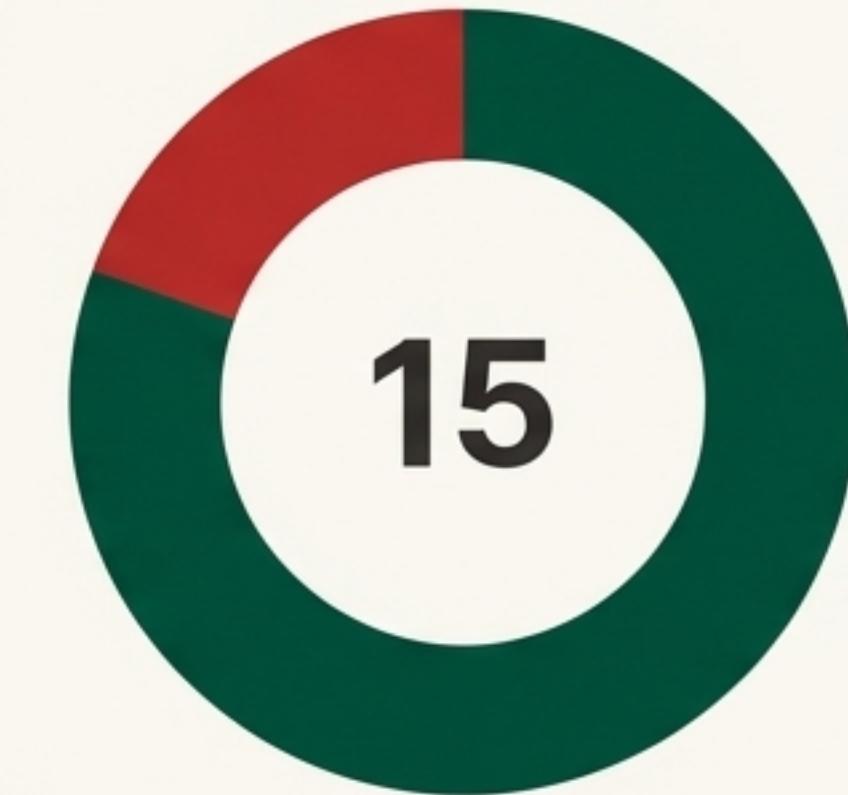
Passées (80%)

Indiquant une bonne
santé générale de
l'application.

3

Échouées (20%)

Révélant des anomalies
critiques à corriger.



Scope of Testing



Tests Fonctionnels:

- Login
- Inventory
- Cart
- Checkout
- Products



Tests Non-Fonctionnels:

- Performance (temps de chargement)
- Responsive (affichage mobile)



Tests Générés par IA:

- Sécurité (accès non-connecté)
- Volume (ajout de tous les produits)

Couverture Complète : La Matrice de Traçabilité des Exigences (RTM)



100% des exigences identifiées sont couvertes par au moins un cas de test.

Summary of Coverage

- REQ-01 to REQ-11:** Couvrent le parcours utilisateur fonctionnel de bout en bout.
- REQ-PERF & REQ-MOB:** Valident les aspects non-fonctionnels clés.
- REQ-SEC & REQ-VOL:** Exigences de sécurité et de volume identifiées **grâce à l'IA**, et couvertes avec succès.

Detailed RTM (visualized as a compact table for reference)

ID Req.	Description de l'exigence	ID Test	Statut
REQ-01	Connexion valide	TC_LOG_01	PASSED
REQ-05	Retrait produit (inventaire)	TC_INV_02	FAILED
REQ-07	Navigation Panier -> Détails	TC_CART_04	FAILED
REQ-11	Format monétaire (Regex)	TC_PROD_02	FAILED
REQ-SEC	Restriction d'accès (IA)	TC_PROD_02	PASSED
REQ-VOL	Ajout total (IA)	TC_IA_01	PASSED
REQ-VOL	Ajout total (IA)	TC_IA_02	PASSED
... (etc.)			

Rapport d'Anomalies : Les 3 Découvertes Clés

La campagne de tests automatisés a permis d'identifier et de documenter 3 anomalies, dont deux de严重性 majeure.



BUG-001

Navigation Interrrompue

Module: Panier (`Cart`)

Description: Le clic sur un produit dans le panier ne redirige pas vers la page de détails correcte.

Sévérité: Majeure



BUG-002

Suppression Inopérante

Module: Inventaire (`Inventory`)

Description: Le bouton 'Remove' sur la page d'inventaire ne met pas à jour le compteur du panier.

Sévérité: Majeure



BUG-003

Format de Prix Incorrect

Module: Produits (`Products`)

Description: Le prix d'un article spécifique ne respecte pas le format monétaire attendu (\$XX.XX).

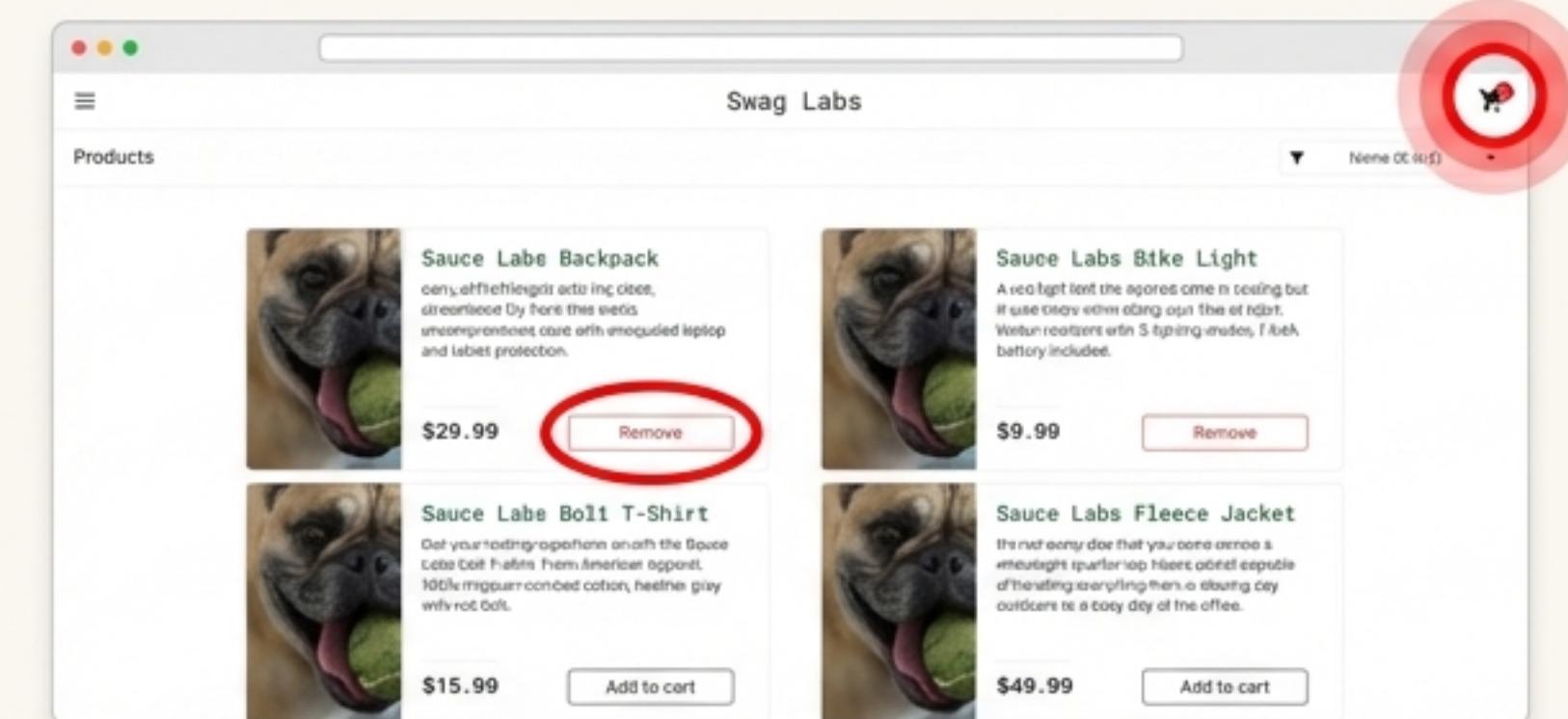
Sévérité: Mineure

Preuves des Anomalies : Analyse Visuelle

BUG-002 - Suppression impossible depuis l'Inventaire

Contexte: L'utilisateur a ajouté deux produits puis cliqué sur 'Remove' pour l'un d'eux.

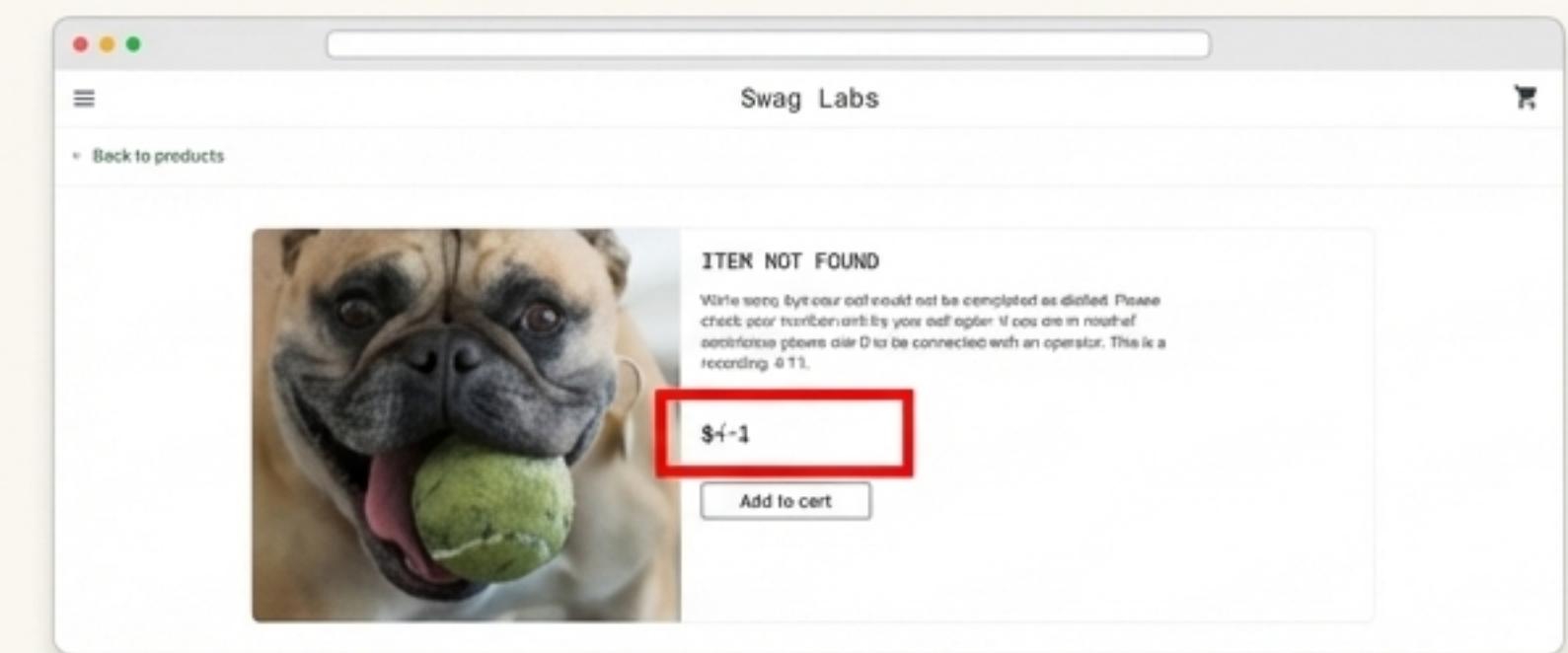
Observation: Le bouton 'Remove' est cliqué, mais le compteur du panier affiche toujours '2'. L'état de l'application est incohérent.



BUG-003 - Format de prix incorrect

Contexte: Test de validation du format du prix sur la page de détail du 4ème produit.

Observation: Le prix est affiché comme `\$-1`, ce qui échoue à la validation par expression régulière (Regex) et présente une information invalide à l'utilisateur.



La Fondation Technique : Qualité et Réutilisabilité du Code

Une architecture de test efficace repose sur un code propre, réutilisable et facile à maintenir.

`base_page.py` - Le Cœur du Framework

Centralise toutes les interactions Selenium (clics, saisies, attentes explicites) pour éviter la duplication de code et fiabiliser les tests.

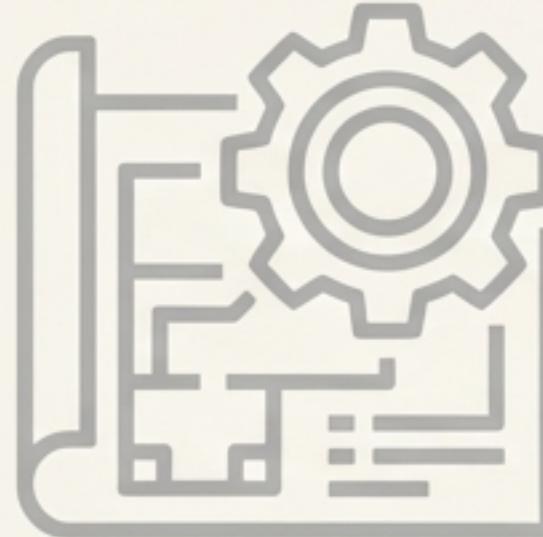
```
class BasePage:  
    def __init__(self, driver):  
        self.driver = driver  
  
    def do_click(self, by_locator):  
        WebDriverWait(self.driver, 10).until(  
            EC.visibility_of_element_located(by_loca  
        ).click()
```

`conftest.py` - Gestion Intelligente du Navigateur

Utilise les 'fixtures' de Pytest pour gérer automatiquement le cycle de vie du navigateur (ouverture, configuration headless, fermeture) pour chaque classe de test.

```
@pytest.fixture(scope="class") ← Gestion de  
def init_driver(request): cycle de vie  
    options = webdriver.ChromeOptions()  
    options.add_argument("--headless")  
    web_driver = webdriver.Chrome(options=options)  
    request.cls.driver = web_driver  
    yield  
    web_driver.quit()
```

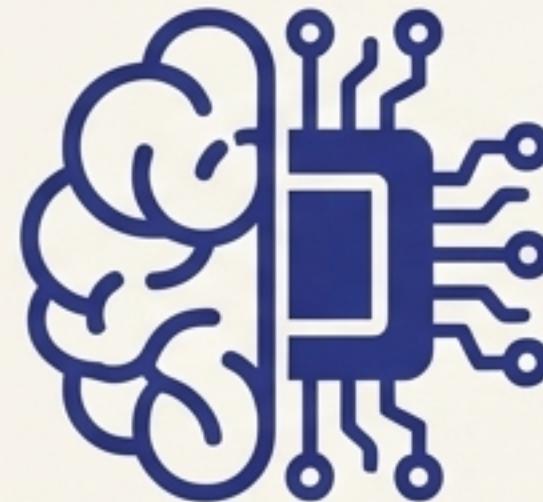
Synthèse : Une Campagne de Tests Complète et Novatrice



Architecture Robuste:
Mise en place d'un framework d'automatisation maintenable et scalable basé sur le Page Object Model.



Couverture Maximale:
Atteinte de 100% de couverture des exigences fonctionnelles et non-fonctionnelles.



Innovation par l'IA:
Démonstration réussie de l'utilisation de l'IA pour enrichir la couverture de tests avec des scénarios de sécurité et de volume.



Impact Concret:
Identification et documentation de **3 anomalies critiques**, fournissant une valeur directe pour l'amélioration de l'application.

Ce projet démontre la maîtrise de l'ensemble du cycle de vie des tests logiciels, de la conception stratégique à l'exécution automatisée et au reporting d'anomalies.