

# Bilan de campagne de test

## I. Recommandations

### A. Tests à automatiser

Nous avons sélectionné les scénarios à automatiser en tenant compte de :

- *La fréquence d'utilisation : cibler les fonctionnalités les plus utilisées et répétitives.*
- *La criticité : prioriser les scénarios à fort impact financier, juridique ou d'image.*
- *La complexité et couverture : privilégier les tests simples à mettre en place et compatibles multi-navigateurs, permettant une détection rapide des bugs critiques.*

*Les tests ciblés incluent notamment la connexion utilisateur, les ajouts au panier, la gestion du stock et les limites associées.*

### B. Préconisations pour la suite

Il est conseillé, lorsque les ressources le permettront, d'automatiser également :

- *L'affichage complet des produits sur la page d'accueil, un élément clé récurrent dans le parcours utilisateur.*

## II. Tests automatisés

Document revu par :

| Nom            | Fonction    | Version | Signature     |
|----------------|-------------|---------|---------------|
| Laurie Pfeffer | QA Engineer | 1.0.0   | <i>Laurie</i> |
|                |             |         |               |

## Contexte

Cette campagne a été menée dans le cadre de la création d'un site e-commerce spécialisé en produits de beauté solides éco-responsables.

Équipe : Marie (test manuel), Favio (CTO), Laurie (QA Engineer en automatismes).

Les tests ont été exécutés sur Chrome v143 et Electron v138.

## Objectifs

- Automatiser les scripts avec Cypress pour tester l'API (10 requêtes principales).
- Réaliser des smoke tests.
- Tester la résistance à une faille XSS dans l'espace commentaire.
- Automatiser 2 scénarios fonctionnels critiques.

## Tests effectués

### Tests API

- *GET /orders sans auth* : vérifie retour d'erreur pour données confidentielles.
- *GET /orders avec auth* : récupère la liste des produits du panier.
- *GET /products/{id}* : récupère une fiche produit spécifique.
- *POST /login* : cas utilisateur inconnu (401) et connu (200).
- *POST /orders/add* : ajout produit disponible.
- *PUT /orders/add* : ajout produit disponible.
- *POST /orders/add* : ajout produit en rupture de stock
- *PUT /orders/add* : ajout produit en rupture de stock.
- *POST /reviews* : ajout d'un avis.

### Smoke tests

- Vérifie la présence :
  - des champs et boutons de connexion.
  - des boutons d'ajout au panier quand connecté.
  - des boutons consulter sur les produits.
  - du bouton ajouter un produit au panier.

## Test de sécurité

- *Injection XSS dans l'espace commentaires avec 5 payloads classiques pour vérifier si exécution JS possible.*
  - *Tous les payloads sont affichés comme texte brut, mais aucun ne s'exécute (pas d'alerte).*
  - *Conclusion : pas de vulnérabilité XSS détectée – le serveur sanitise correctement les entrées.*

## Tests fonctionnels (2 critiques sélectionnés)

- *Connexion :*
  - *Clique sur bouton connexion.*
  - *Page formulaire s'affiche.*
  - *Entre les identifiants fournis.*
  - *Vérifie connexion réussie (bouton panier visible).*
- *Panier connecté avec infos ci-dessus : Visite directe d'un produit avec un stock suffisant.*
  - *Vérifie stock >2.*
  - *Test limite valeur négative bloquée.*
  - *Test limite >20 (valeur limitée à 20).*
  - *Ajout des produits au panier.*
  - *Vérifie l'ajout.*
  - *Recharge et vérifie stock diminué.*

## Résultats de tests

*Tous les tests ont été lancés via npx cypress run et cypress open.*

- Anomalies héritées de Marie (non corrigées) :
  - **GET/orders** sans connexion retourne 401 ("JWT Token not found") au lieu de 403 comme attendu par Marie.
    - Analyse : le 401 est en réalité le comportement correct (l'utilisateur n'est pas authentifié). Le 403 serait attendu si l'utilisateur était connecté mais sans droits.
    - Impact : aucun, la sécurité est assurée (accès bloqué). Pas de correction nécessaire, mais la spec Swagger ou le cahier des charges devrait être mis à jour pour attendre 401.
  - **POST/orders/add** retourne 405 (attend PUT au lieu de POST).
  - **PUT/orders/add** à l'ajout d'un stock indisponible, ajout accepté.

- *Impact : risque de paniers invalides et commandes non honorables.*
  
- *Nouvelles anomalies détectées par l'automatisation :*
  - *Bouton "Ajouter" n'est pas grisé, n'est pas utilisable, malgré une quantité négative.*
  - *Absence de limitation stricte au-delà de 20 unités à l'ajout.*

## **Propositions :**

- **Renforcer la validation** côté frontend et backend pour :
  - Bloquer les quantités invalides (négatives ou supérieures au stock).
  - Afficher des messages d'erreur clairs à l'utilisateur.
  - Empêcher que le stock affiché devienne négatif.
- **Compléter avec des tests automatisés** API et UI pour couvrir ces cas et éviter toute régression.

## **Confiance**

*J'ai une confiance modérée sur la version actuelle.*

*L'automatisation a permis de détecter les mêmes anomalies que Marie et de gagner en rapidité et fiabilité.*

*La divergence entre frontend et backend sur le contrôle des ajouts en dépassement de stock nécessite une attention prioritaire pour assurer la cohérence et la fiabilité globale du système*