

Rapport de testing

I. Contexte du projet

Eco Bliss Bath développe un site e-commerce pour la vente de produits de beauté écoresponsables. Une première campagne de tests manuels a été réalisée afin d'identifier les anomalies et les fonctionnalités critiques à tester. L'objectif de ce rapport est de déterminer les tests à automatiser en priorité afin d'optimiser la couverture de test et d'améliorer la réactivité aux bugs.

II. Recommandations

A. Tests à automatiser

1. Connexion (Priorité élevée) → À AUTOMATISER

- Sans connexion, l'utilisateur ne peut pas accéder à des fonctionnalités clés comme le panier ou l'historique de commandes.
- Une panne de connexion signifie une perte immédiate de clients potentiels.
- Tester la connexion permet aussi de vérifier la gestion des erreurs (mauvais identifiants, expiration de session, etc.).
- L'automatisation garantira une détection rapide des bugs critiques et réduira les régressions.

2. Panier (Priorité élevée) → À AUTOMATISER

- Impact direct sur les ventes : si le panier ne fonctionne pas, les utilisateurs ne peuvent pas finaliser leurs achats.
- Plusieurs scénarios critiques à valider :
 - ✓ Ajout de produits
 - ✓ Vérification du stock après ajout
 - ✓ Gestion des limites (quantité négative ou trop élevée)
 - ✓ Passage en caisse
- Un bug sur cette fonctionnalité entraîne une perte de revenus immédiate.

3. Affichage des Produits ✨ (Priorité moyenne 🕒 🕒)

→ À FAIRE MANUELLEMENT

Pourquoi ne pas automatiser ?

- Un bug dans l'affichage impacte l'expérience utilisateur mais n'empêche pas forcément un achat (les produits restent accessibles par recherche ou navigation).
- Les tests d'affichage nécessitent souvent une validation visuelle (exemple : images mal positionnées, textes tronqués), ce qui rend l'automatisation moins efficace.
- Un test manuel régulier suffit pour détecter d'éventuels problèmes d'affichage, surtout si des tests API garantissent que les produits sont bien récupérés.

4. Justification des priorités

Priorité est donnée à l'automatisation de :

- **La connexion** : Essentiel pour que les utilisateurs puissent accéder au site.
- **Le panier** : Impact direct sur les ventes et l'expérience utilisateur.

L'affichage des Produits, moins critique, peut être testé manuellement.

Ce choix maximise le retour sur investissement des tests en automatisant les fonctionnalités qui ont un impact direct sur les conversions et l'expérience utilisateur.

B. Préconisations pour la suite

- Automatiser les tests d'affichage des produits lorsque le budget le permettra, en combinant des tests fonctionnels et des outils de vérification visuelle.
- Mettre en place un système de tests réguliers sur différents navigateurs et appareils pour vérifier la compatibilité.
- Prioriser l'automatisation des tests sur d'autres fonctionnalités critiques comme le paiement et la gestion des commandes.
- Mettre en place un processus automatisé qui exécute les tests après chaque modification du code. Cela permettra de détecter immédiatement les erreurs et de garantir que les nouvelles mises à jour ne perturbent pas les fonctionnalités existantes.

Conclusion : L'automatisation doit se concentrer sur les tests ayant le plus grand impact sur l'expérience utilisateur et les conversions. Les tests manuels resteront pertinents pour les validations visuelles et certains cas complexes. En appliquant ces recommandations, l'équipe QA pourra maximiser la fiabilité des tests tout en optimisant les coûts et les ressources.

III. Bilan de campagne de validation : tests automatisés

Document revu par :

Nom	Fonction	Version	Signature
Marie C.	Testeuse d'automatisation	1.0.0	

A. Contexte :

Cette campagne de tests s'inscrit dans le cadre du lancement de la **version 1 du site e-commerce Eco Bliss Bath**, spécialisé dans la vente de produits de beauté écoresponsables.

Après une première phase de **tests manuels réalisés par Marie (QA manuelle)**, j'ai été chargée de prendre le relais pour mettre en place une **campagne de tests automatisés**, avec pour objectif de fiabiliser les fonctionnalités critiques et garantir une expérience utilisateur fluide dès les premiers usages du site.

L'équipe mobilisée pour cette campagne se compose de :

- Marie (test manuel)
- Marie C. (moi-même, test automatisé)
- Fabio (CTO, superviseur technique)
- Clément (Product Owner)

Les tests ont été réalisés avec l'outil **Cypress**, en environnement de développement local via **Docker**, sur le navigateur **Chrome 135.0.7049.85** (version stable à la date de test), et sur le système **Windows 10**.

B. Objectif(s) :

Alors que le site e-commerce d'Eco Bliss Bath entre dans sa première phase de mise en production, il devient essentiel d'assurer la fiabilité des fonctionnalités critiques afin de garantir une expérience utilisateur fluide et de préserver les performances commerciales.

Dans cette optique, l'équipe technique a choisi d'intégrer **Cypress** comme outil de test automatisé pour renforcer la qualité du produit et accélérer la détection des anomalies.

Sous la supervision du CTO et en collaboration avec la QA manuelle, j'ai été chargée de mettre en œuvre une campagne de tests automatisés.

L'objectif principal est de mettre en œuvre une **première campagne de tests automatisés** afin de couvrir les **fonctionnalités critiques** identifiées lors de la campagne manuelle :

- La connexion
- Le panier

Ces deux fonctionnalités ayant un **impact direct sur l'expérience utilisateur et les conversions**, elles ont été considérées comme prioritaires pour l'automatisation.

En complément, des **tests API** et des **smoke tests** ont été mis en place afin de s'assurer du bon fonctionnement global de l'application à chaque déploiement.

Les objectifs secondaires sont les suivants :

- Vérifier la stabilité globale de l'application à l'aide de **smoke tests**
- Contrôler les **endpoints de l'API** (connexion, panier, produits, avis)
- Simuler des cas limites en UI pour **tester les comportements anormaux** ou incohérents
- Collecter les résultats, détecter les anomalies et produire un **rapport structuré** pour guider les futures actions QA

Cette campagne vise à :

- Accélérer la détection des régressions
- Renforcer la fiabilité du site avant mise en production
- Poser les fondations d'une stratégie de test automatisé durable

C. Tests effectués :

Test 1 : API – authTests.cy.js

- Requêtes POST de connexion :
 - Connexion avec des identifiants valides : renvoie un code 200.
 - Connexion avec des identifiants invalides : renvoie un code 401.

Test 2 : API – cartTests.cy.js

- Requêtes GET sur le panier :
 - Sans connexion : renvoie un code 401 (non autorisé).
 - Avec connexion : renvoie la liste des produits présents dans le panier.
- Requêtes PUT sur le panier :
 - Ajout d'un produit en stock : renvoie un code 200 avec mise à jour du panier.
 - Ajout d'un produit en rupture de stock : renvoie un code 200 au lieu du code d'erreur attendu (*anomalie détectée*).

Test 3 : API – productTests.cy.js

- Requête GET sur une fiche produit aléatoire :
 - Vérifie la présence des propriétés attendues (nom, description, image, disponibilité, prix).

Test 4 : API – reviewsTests.cy.js

- Ajout d'un avis :
 - Avis ajouté si tous les champs sont remplis correctement.
 - L'avis n'est pas ajouté si des champs sont manquants.
 - Test d'une requête avec contenu XSS : renvoie un code 200 au lieu du 400 attendu (*anomalie détectée*).

Test 5 : UI – loginTests.cy.js

- Test de connexion utilisateur :
 - Vérifie l'affichage du formulaire de connexion (champ email, mot de passe, bouton).
 - Simule la saisie des identifiants et la redirection vers la page d'accueil en mode « connectée ».
 - Vérifie la présence du bouton « Mon panier » une fois connecté.

Test 6 : UI – cartUiTests.cy.js

- Ajout au panier :
 - Le produit est ajouté au panier depuis sa fiche.
 - Le stock est mis à jour sur la fiche produit.
 - Le produit est visible dans le panier et également via l'API

- Limites de saisie :
 - Quantité négative : le champ accepte la valeur, aucun blocage (*anomalie détectée*).
 - Quantité supérieure à 20 : le champ accepte la valeur, aucun blocage (*anomalie détectée*).
- Suppression des éléments du panier via l'API

Test 7 : Smoke Tests – loginSmoke.cy.js et productSmoke.cy.js

- Présence des éléments sur les pages clés :
 - Formulaire de connexion : vérifie les champs email, mot de passe et le bouton de validation.
 - Page produits : vérifie la présence du bouton "Ajouter au panier" et l'affichage de la disponibilité du produit.

D. Résultats de tests

Sur les tests automatisés ont permis de détecter 4 anomalies significatives plus les 2 mises en évidence par Marie lors de la réalisation des tests manuels.

Les échecs observés sont listés ci-dessous :

Anomalie 1 – cartTests.cy.js

- Scénario : Ajout au panier d'un produit en rupture de stock.
- Résultat attendu : La requête doit échouer avec un code 400 ou équivalent.
- Résultat obtenu : La requête réussit avec un code 200, ce qui laisse penser que le contrôle de disponibilité n'est pas appliqué côté back-end.
- Impact : Risque de commandes fantômes, mauvaise expérience utilisateur.

Anomalie 2 – reviewsTests.cy.js

- Scénario : Soumission d'un avis contenant une tentative de faille XSS.
- Résultat attendu : La requête doit être rejetée avec un code 400.
- Résultat obtenu : La requête est acceptée avec un code 200.
- Impact : Risque de faille de sécurité majeure, pouvant compromettre l'intégrité du site.

Anomalie 3 – cartUiTests.cy.js

- Scénario : Saisie d'une quantité négative dans le champ de quantité.
- Résultat attendu : Le champ doit empêcher ou corriger la saisie.
- Résultat obtenu : La valeur est acceptée sans blocage ni message d'erreur.
- Impact : Erreur potentielle de commande ou manipulation non prévue.

Anomalie 4 – cartUiTests.cy.js

- Scénario : Saisie d'une quantité supérieure à 20.
- Résultat attendu : Le champ doit être limité à une valeur maximum (ex. : 20).
- Résultat obtenu : La valeur est acceptée sans blocage.
- Impact : Dépassement des règles de gestion du stock, erreurs lors de la commande.

Anomalies mises en évidence par Marie :

Anomalie 5 – apiTests/cartTests.cy.js

- Scénario : Ajout d'un produit au panier via l'API.
- Résultat attendu : La requête doit utiliser le verbe HTTP POST conformément aux bonnes pratiques REST.
- Résultat obtenu : La requête utilise le verbe PUT pour l'ajout d'un produit.
- Impact : Non-conformité avec les conventions REST, pouvant engendrer des erreurs de compréhension ou d'intégration côté front-end.

Anomalie 6 – apiTests/cartTests.cy.js

- Scénario : Suppression d'un produit du panier via l'API.
- Résultat attendu : La route de suppression doit se limiter à DELETE /orders/{id}, sans redondance.
- Résultat obtenu : La route est définie comme DELETE /orders/{id}/delete, ce qui est redondant avec le verbe HTTP utilisé.
- Impact : Mauvaise lisibilité et cohérence de l'API, ce qui peut compliquer la maintenance et l'intégration.

Des captures d'écran et vidéos ont été générées automatiquement pour tous les tests, facilitant l'analyse des résultats et la communication avec l'équipe de développement.

E. Rapport d'incident

1. Ajout de produits en rupture de stock

1. Environnement

- Outil utilisé : Cypress
- Fichier de test : apiTests/cartTests.cy.js

2. Étapes pour reproduire le bug

- Se connecter à l'application
- Tenter d'ajouter un produit signalé comme indisponible (stock à 0)

3. Comportement attendu

- Le produit ne doit pas être ajouté au panier
- L'API doit renvoyer une erreur (code 400 ou message bloquant)

4. Comportement observé

- Le produit est ajouté
- L'API renvoie un code 200

5. Capture d'écran

- Fichier : Test PUT sur le panier -- N'ajoute pas un produit en rupture de stock (failed).png



- Fichier : 8-FicheProduit-Avant-Ajout-API.png

Sentiments printaniers

Savon avec une formule douce à base d'huile de framboise, de citron et de menthe qui nettoie les mains efficacement sans les dessécher.

PEAU

Propre, fraîche

AROMES

Frais et fruité

INGRÉDIENTS CLÉS

Framboise, zeste de citron et feuille de menthe

60,00 €
-8 en stock

1


Ajouter au panier

- Fichier : 9-Panier-Après-Ajout-API.png

[Accueil](#) [Produits](#) [Avis](#) [e c o . b l i s s . b a t h](#) [Mon panier](#) [Déconnexion](#)

Commande


Panier



Sentiments printaniers

Savon avec une formule douce à base d'huile de framboise, de citron et de menthe qui nettoie les mains efficacement sans les dessécher.

1

 60,00 €

Total

60,00 €

Frais de livraison offerts

00,00 €

6. Message d'erreur

- Aucun message bloquant affiché dans l'interface

- Fichier : 5-Saisie-Négative.png

Milkyway

Savon fabriqué à partir d'ingrédients naturels tels que l'huile d'olive, la cire d'abeille et l'huile essentielle de lavande.

PEAU

Mature

AROMES

Lavande, rose

INGRÉDIENTS CLÉS

Huiles essentielles

15,00 €
6 en stock

-1

Ajouter au panier

- Fichier : Empêche l'ajout au panier avec une quantité supérieure à 20 (failed).png

```
Specs
[✓] 1 [✗] 1 [🔄] 0 [⌵] [⌵]

cartUiTests.cjys 00:31

2 visit '##/products/10'
3 get [data-cy='detail-product-name']
4 -assert expected <h1> to contain Aurore boréale
   (xhr) GET 200 http://localhost:8081/products/10
   (xhr) GET 200 http://localhost:8081/products/random
5 get [data-cy='detail-product-stock']
6 -assert expected <p.stock> to contain 19
7 -screenshot cartUiTests/6-FicheProduit-Avant-Saisie-21
8 get [data-cy='detail-product-quantity']
9 -clear
10 -type 21
11 -screenshot cartUiTests/7-Saisie-Supérieure-À-20
12 get [data-cy='detail-product-quantity']
13 -assert expected <input.ng-untouched.ng-dirty.ng-valid> to have value '20', but the value was '21'

❑ Produit ajouté au panier, présent dans le panier via l'API
```

- Fichier : 7- Saisie-Supérieure-À-20.png

Aurore boréale

Ses propriétés apaisantes aident à calmer les nerfs et à améliorer votre bien-être général.

PEAU

Irritée

AROMES

Vanille

INGRÉDIENTS CLÉS

Agile verte

26,00 €
19 en stock

21

Ajouter au panier

6. Message d'erreur

- Aucun message d'erreur affiché

7. Autres informations

- L'absence de contrôle rend possible la manipulation du stock et peut provoquer des erreurs côté serveur.

Priorité : Haute

3. Fail XSS dans les avis

1. Environnement

- Outil utilisé : Cypress
- Fichier de test : apiTests/reviewsTests.cy.js

2. Étapes pour reproduire le bug

- Remplir tous les champs obligatoires du formulaire d'avis
- Injecter un script JavaScript dans le champ commentaire
- Soumettre l'avis

3. Comportement attendu

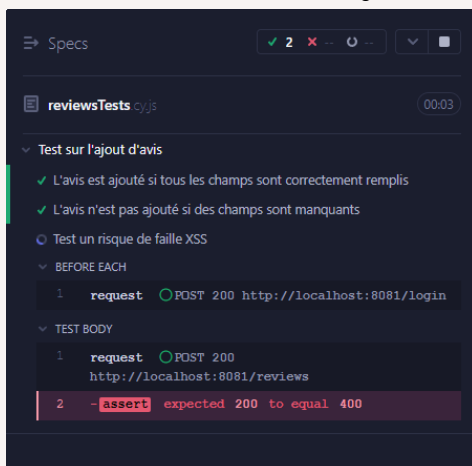
- L'API doit bloquer la requête et renvoyer une erreur (400)

4. Comportement observé

- L'API accepte l'entrée et renvoie un code 200

5. Capture d'écran

- Test sur l'ajout d'avis -- Test un risque de faille XSS (failed).png



6. Message d'erreur

- Aucun message bloquant affiché

7. Autres informations

- Une attaque XSS peut compromettre l'ensemble du site et nuire gravement aux utilisateurs.

Priorité : Très haute

4. Mauvais verbe HTTP pour l'ajout d'un produit au panier

1. Environnement

- Outil utilisé : Swagger
- Méthode testée : /orders/add

2. Étapes pour reproduire le bug

- Lancer une requête PUT /orders/add depuis Swagger
- Vérifier le type de verbe utilisé pour l'ajout

3. Comportement attendu

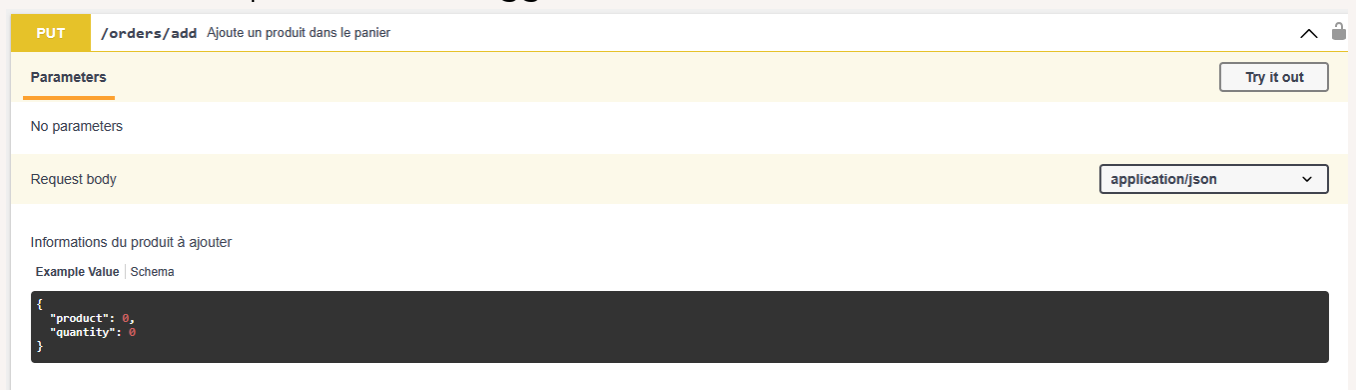
- L'ajout d'un produit doit s'effectuer avec la méthode POST, conformément aux conventions REST (création de ressource).

4. Comportement observé

- La méthode PUT est utilisée pour cette action.

5. Capture d'écran

- Disponible via Swagger.



6. Message d'erreur

- Aucun message d'erreur bloquant

7. Autres informations

- Cette mauvaise implémentation n'empêche pas le fonctionnement, mais nuit à la clarté et la maintenabilité de l'API.

Priorité : Moyenne

5. Route de suppression redondante

1. Environnement

- Outil utilisé : Swagger
- Méthode testée : DELETE /orders/{id}/delete

2. Étapes pour reproduire le bug

- Effectuer une requête pour supprimer un élément du panier
- Observer la route définie dans la documentation de l'API

3. Comportement attendu

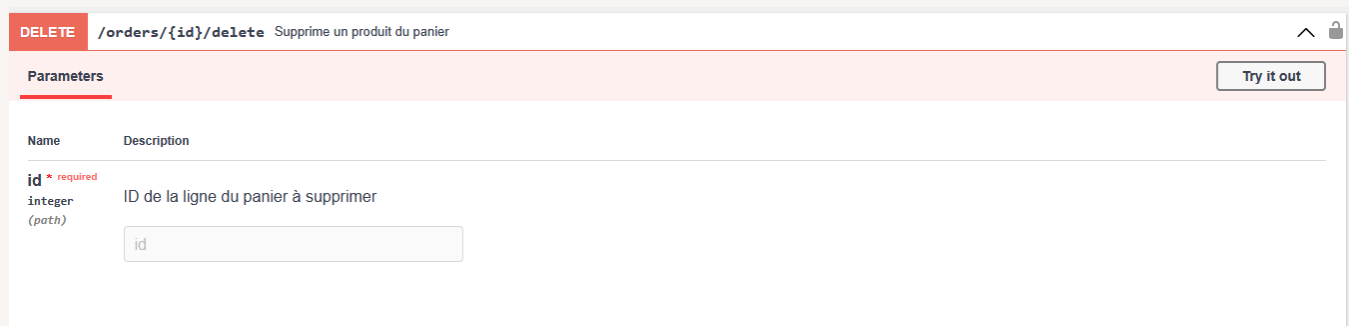
- Une requête DELETE sur /orders/{id} est suffisante et respectueuse des standards REST.

4. Comportement observé

- La route actuelle est DELETE /orders/{id}/delete, ce qui est redondant.

5. Capture d'écran

- Disponible via Swagger.



6. Message d'erreur

- Aucun message d'erreur bloquant

7. Autres informations

- Ce choix de route ne bloque pas le fonctionnement, mais nuit à la cohérence et à la maintenabilité de l'API.

Priorité : Basse (À améliorer pour respecter les standards REST et éviter les redondances inutiles.)

F. Confiance

À ce stade de la campagne de tests, la version actuelle de l'application ne peut pas être considérée comme totalement fiable pour une mise en production sans corrections supplémentaires.

Plusieurs anomalies critiques ont été détectées au cours des tests automatisés, en particulier sur des fonctionnalités essentielles telles que le panier, la validation des quantités ou encore la sécurité des avis utilisateurs.

Criticité des anomalies détectées :

- **Haute priorité (à corriger en urgence) :**
 - Possibilité d'ajouter un produit en rupture de stock au panier (rupture de stock non prise en compte côté back-end).
 - Absence de contrôle sur la quantité ajoutée au panier (valeurs négatives ou excessives acceptées).
 - Absence de protection contre les attaques XSS dans le formulaire d'avis (faille de sécurité majeure).
- **Moyenne priorité :**
 - Utilisation d'un verbe HTTP inadapté pour l'ajout au panier (PUT au lieu de POST).
 - Route de suppression d'un produit redondante (/delete superflu).

Avis sur la stabilité :

- **Points rassurants :**
 - La connexion fonctionne correctement avec gestion des erreurs d'identifiants.
 - Les produits sont correctement affichés et la navigation entre pages ne génère pas d'erreurs.
 - Les tests smoke permettent de valider rapidement la disponibilité des fonctionnalités clés à chaque lancement.
- **Points bloquants :**
 - Les failles identifiées sur la gestion du stock peuvent générer des commandes impossibles à honorer.
 - La faille XSS constitue un risque majeur en matière de sécurité, pouvant impacter l'intégrité du site et la confiance des utilisateurs.

G. Conclusion :

La version actuelle **ne peut pas être validée en l'état**. Une mise en production sans correction exposerait l'entreprise à des bugs visibles côté client, une mauvaise gestion des commandes, et des risques de sécurité importants.

Les corrections à prioriser concernent le panier (stock et quantités) ainsi que la sécurité des avis. Une fois ces points traités, une nouvelle campagne de tests pourra être lancée pour valider les corrections et renforcer la confiance dans la livraison.