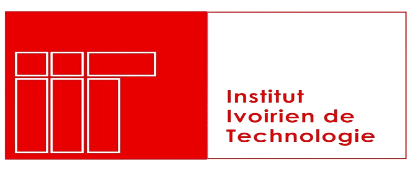
**** 

**THEME :**

**Rapport de Test et Validation de l'Application de Réservation Régionale**

|  |
| --- |
| **COURS :** Méthodes de test et de validation de logiciel |

**Présenté par : Professeur :**

**DJEDJI FRANCK ARMAND Sedrick KOUAGNI**

**Objet du rapport**

Ce rapport détaille les tests effectués et les validations réalisées pour garantir la conformité de l'application de réservation régionale, en suivant les spécifications fonctionnelles fournies.

**Introduction**

**1. Contexte du projet**

L'objectif de l'application est de permettre aux clubs d'inscrire leurs athlètes aux compétitions régionales, avec un accès réservé aux secrétaires des clubs. L'application vise à alléger la charge administrative des organisateurs et à garantir un déroulement logistique optimal.

**2. Exigences fonctionnelles principales**

- Les secrétaires doivent pouvoir se connecter, visualiser le nombre de points de leur club et acheter des places.

- Les compétitions doivent limiter le nombre d'inscriptions par club à 12.

- Une page publique affichant le solde de points des clubs est accessible sans authentification.

**3. Objectifs des tests**

Les tests visent à valider la conformité de l’application aux spécifications et à assurer une expérience utilisateur fluide et sans erreurs, en vérifiant notamment les limites d’inscription et l’affichage des informations.

**Plan de Test**

1. **Phases de test**

- Phase 0 : Lancement du projet pour recueillir les besoins des utilisateurs.

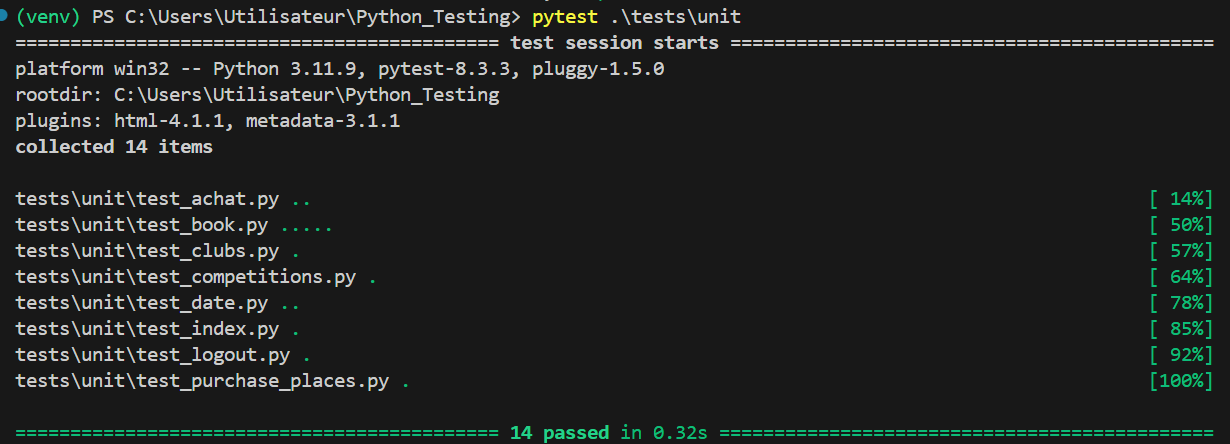
- Phase 1 : Création d’environnement virtuelle et installation des extensions et packages pour le testing

- Phase 2 : Tests d'accès et d’achat de places pour vérifier les restrictions d’inscription (ex. limite de 12 places par club).

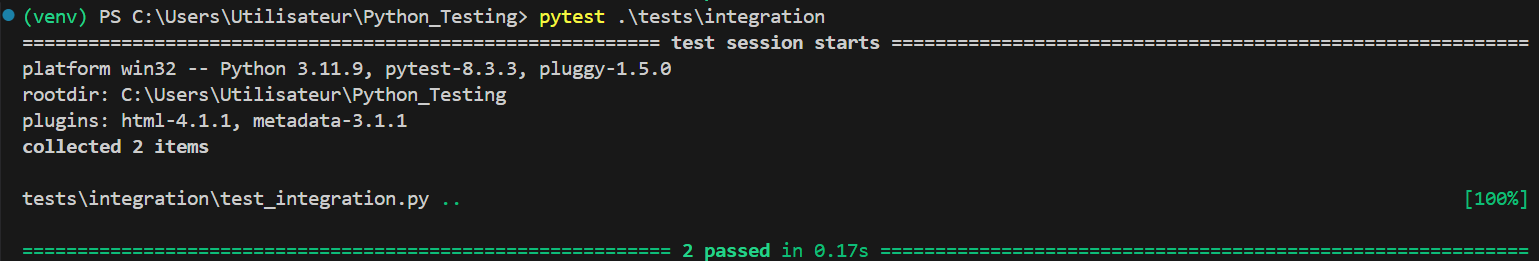
- Phase 3 : Vérification de l’affichage public du tableau des points et test de performance pour des temps de réponse inférieurs à 5 secondes.

**2. Techniques et outils de test**

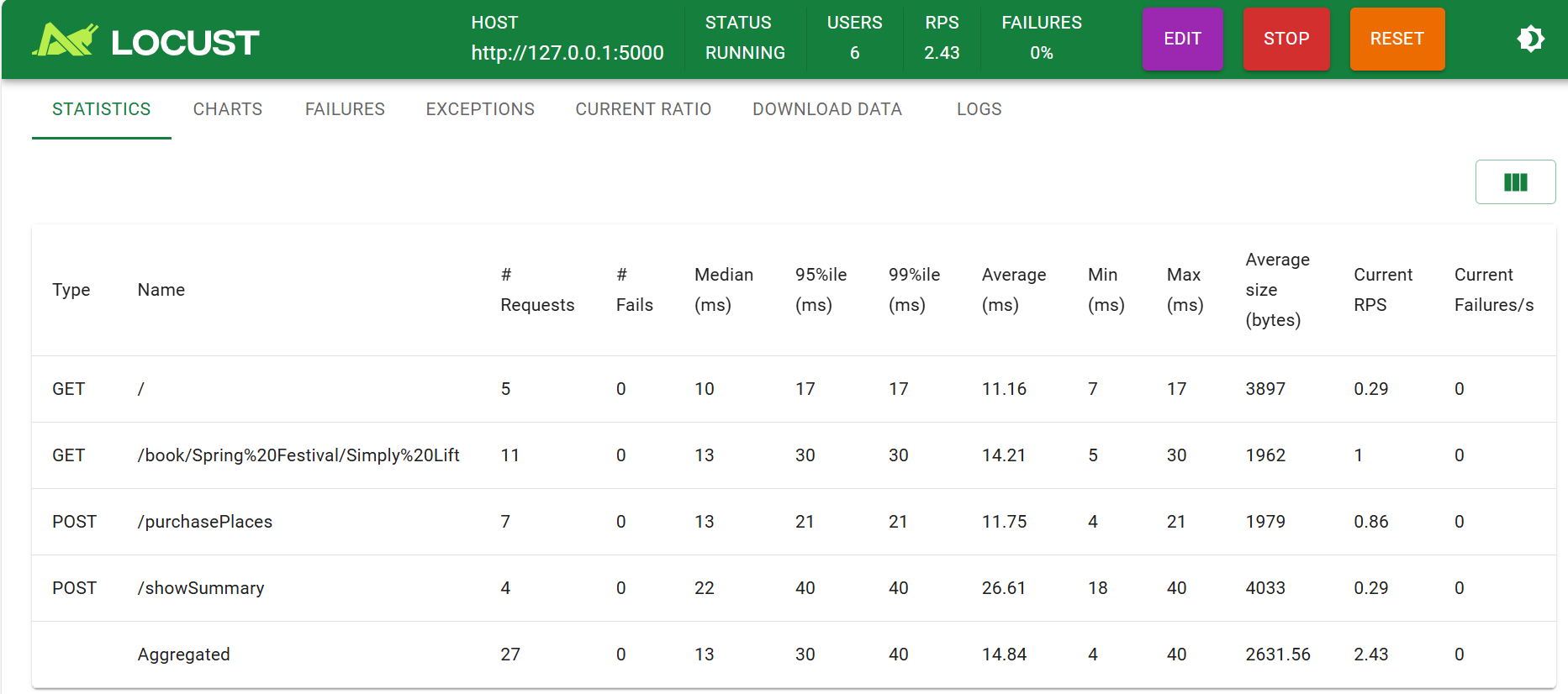
- **Tests unitaires** avec `pytest` pour vérifier le fonctionnement des modules de l’application.



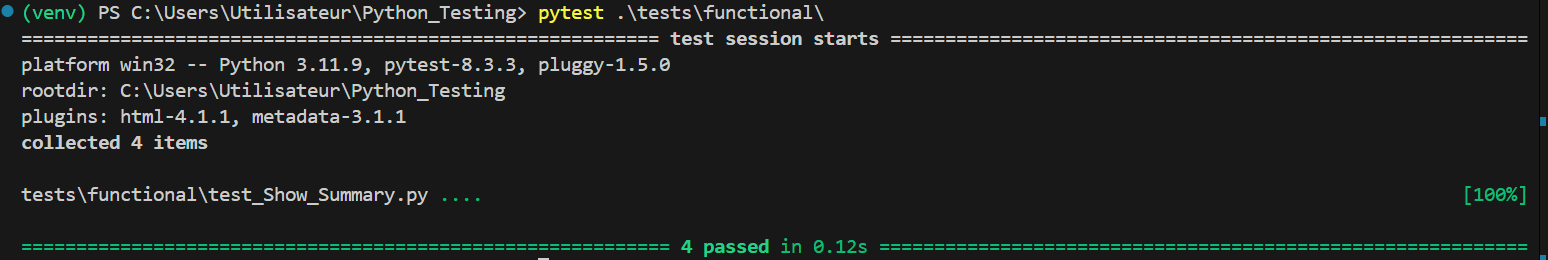
- **Tests d'intégration** pour vérifier l’interaction entre l’achat de places et la mise à jour des points.



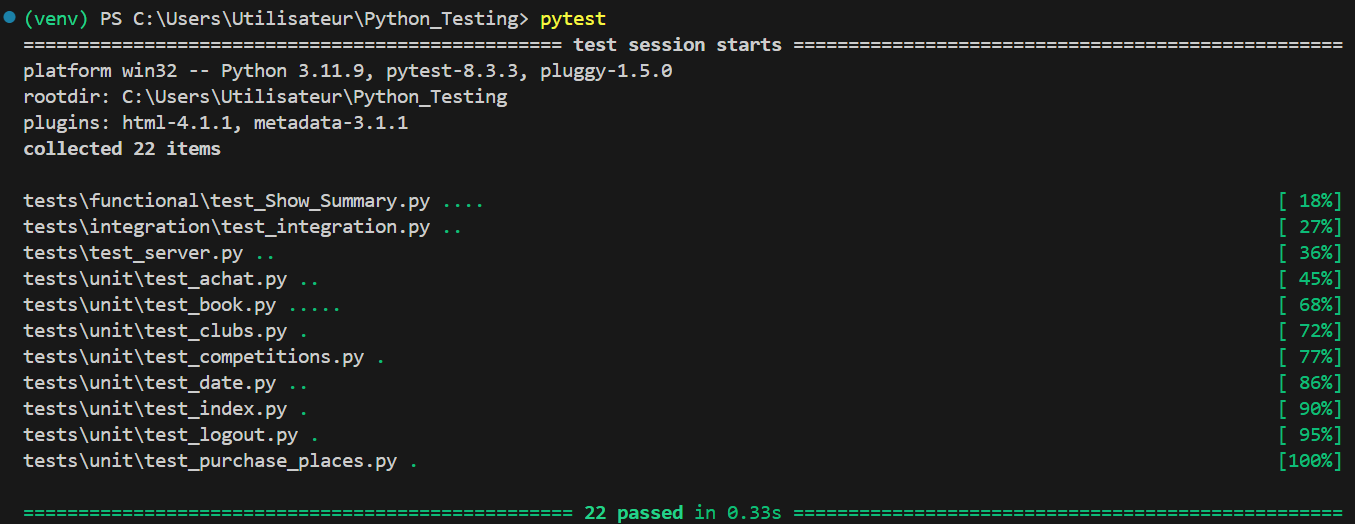
- **Tests de performance** avec `Locust` pour simuler des accès multiples et mesurer le temps de réponse.



- **Tests fonctionnels**



**- All tests**

****

**3. Critères de succès des tests**

Les tests sont réussis si les fonctionnalités de connexion, d’achat de places et d’affichage public des points fonctionnent sans erreur. Le temps de réponse lors des tests de performance doit être inférieur à 5 secondes.

**Résolution des Erreurs**

**1. Problèmes rencontrés et corrections appliquées**

- Lignes de code trop longues et erreurs de style :

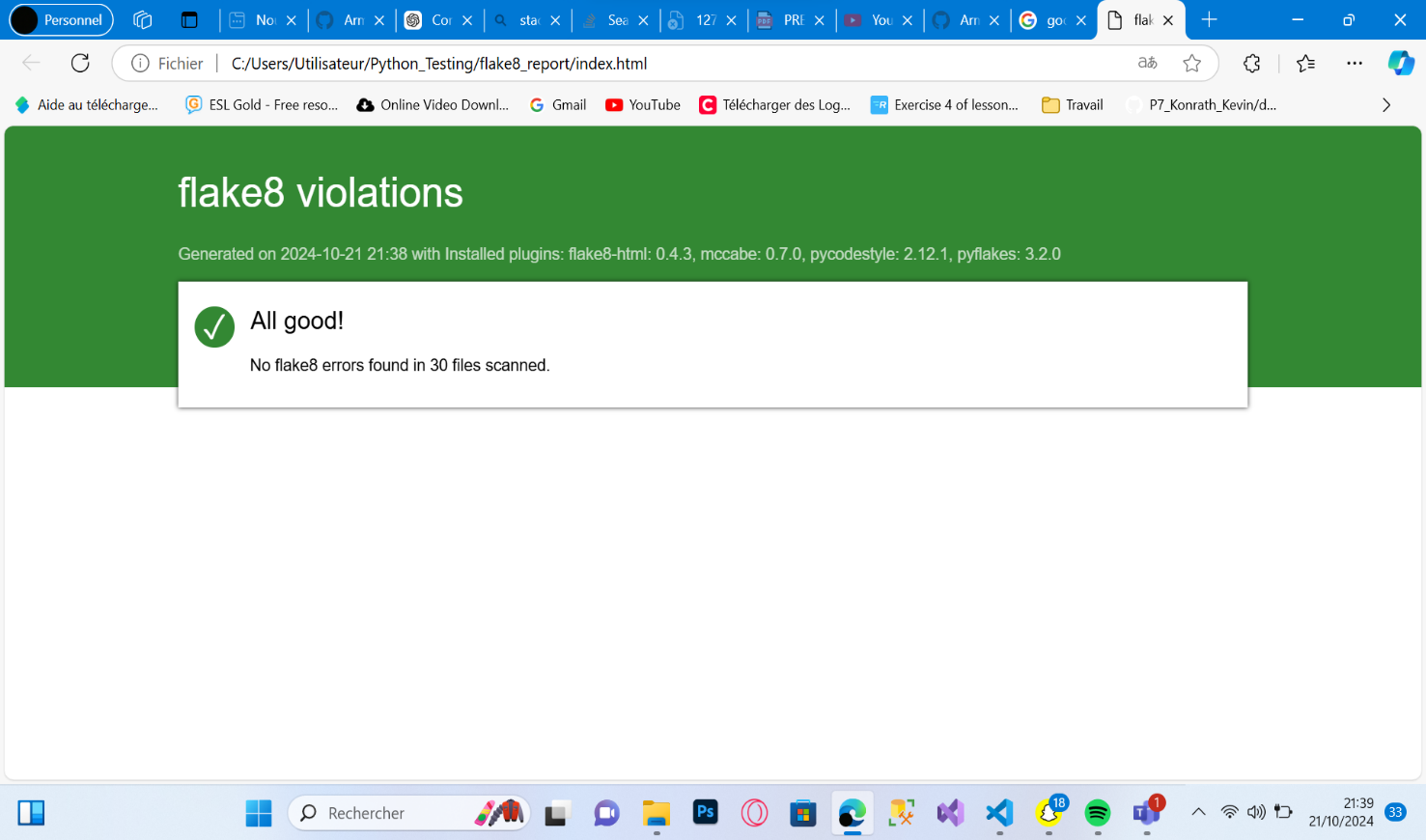
Les erreurs E501 (lignes de plus de 79 caractères) ont été corrigées en fragmentant les lignes trop longues.

- Indentations incorrectes :

Des erreurs de type E127 et E303 ont été corrigées pour une meilleure lisibilité et conformité aux normes PEP8.

- Erreurs de tests :

Lors des tests d'intégration, des erreurs comme `AssertionError` ont été résolues en ajustant les valeurs de données et en améliorant les assertions.



**2. Utilisation des outils de détection d'erreurs**

- **Flake8** pour détecter les erreurs de style.

- **Pytest** pour exécuter les tests et valider les corrections apportées.

- **Git** : Standardisation des branches avec un respect des règles de nommage pour une gestion claire des versions et corrections.

**Conclusion**

Tous les tests ont été réalisés avec succès, et l'application répond aux exigences fonctionnelles. Les recommandations sont de continuer à appliquer ces standards pour maintenir la stabilité et la qualité du code, tout en réduisant les erreurs.