## **Rapport de test et validation de logiciel**

**Nom :** ANE HENRI JOEL

**Projet :** Application de réservation régionale

### **Introduction**

Ce rapport présente les tests effectués sur l'application de réservation régionale, avec pour objectif de garantir sa conformité aux spécifications fonctionnelles, d'assurer une qualité optimale, et de minimiser les risques avant la mise en production.

L’application permet aux clubs de réserver des places pour des compétitions via leurs secrétaires en utilisant un système de points. Les tests se concentrent sur la vérification des fonctionnalités clés et des exigences non fonctionnelles, comme la performance et l’expérience utilisateur.

### **Objectifs du test**

* Valider le bon fonctionnement des fonctionnalités critiques (réservation, connexion, gestion des points).
* Assurer la stabilité et la fiabilité du système sous charge.
* Identifier et corriger les anomalies avant le déploiement.

### **Environnement de test**

* **Outil de test** : pytest
* **Plateforme** : macOS
* **Environnement Virtuel** : .venv
* **Langage** : Python
* **Structure** : Tests unitaires et d'intégration
* **Rapports** : Générés avec pytest-html

### **Plan de test**

#### **1. Cas de test critiques identifiés :**

* **Connexion** : Vérification de l'accès avec les identifiants.
* **Réservation** : Validation du système de points et du plafond de 12 places par club.
* **Gestion des points** : Déduction correcte des points après chaque réservation.
* **Tableau public** : Vérification de la transparence des points entre clubs.

#### **2. Priorisation des tests :**

Les fonctionnalités critiques ont été priorisées, telles que la connexion et la réservation, pour éviter des échecs bloquants.

### **Exécution des tests**

1. **Tests unitaires :**
   1. Ciblé sur les fonctions de calcul de points et d’inscription.
   2. Utilisation de pytest pour vérifier chaque module individuellement.
   3. **Exemple** : Test de la fonction de déduction des points :

python

def test\_home\_page(client):

"""Test d'accès à la page d'accueil."""

response = client.get('/')

assert response.status\_code == 200

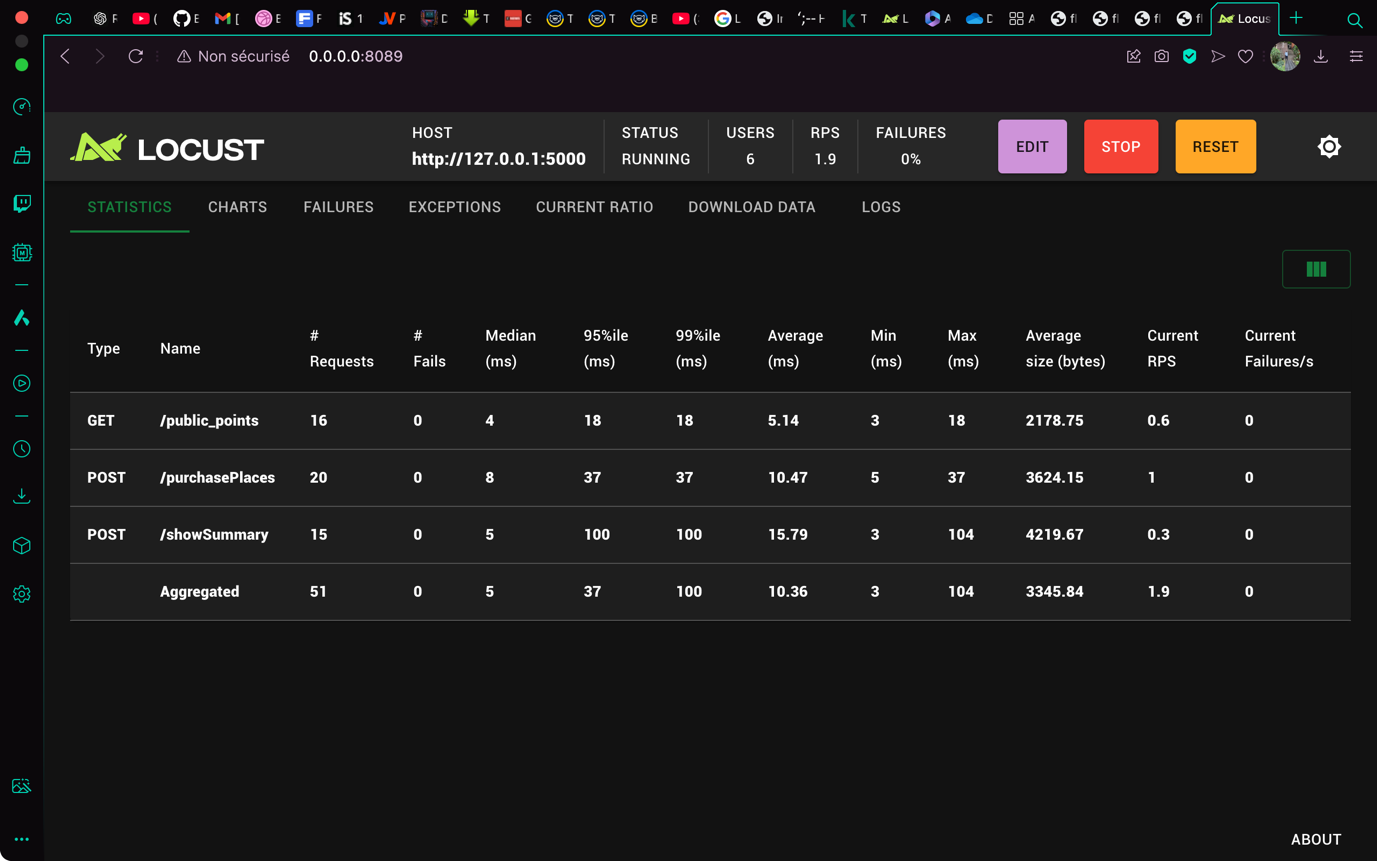
assert (

b"Bienvenue sur le portail d'inscription GUDLFT !"

in response.data

)

1. **Tests d'intégration :**
   1. Assurer l'interaction correcte entre le système de connexion et la réservation.
   2. **Exemple** : Vérification que la déduction des points est mise à jour après une réservation réussie.
2. Tests de performance avec ‘Locust’



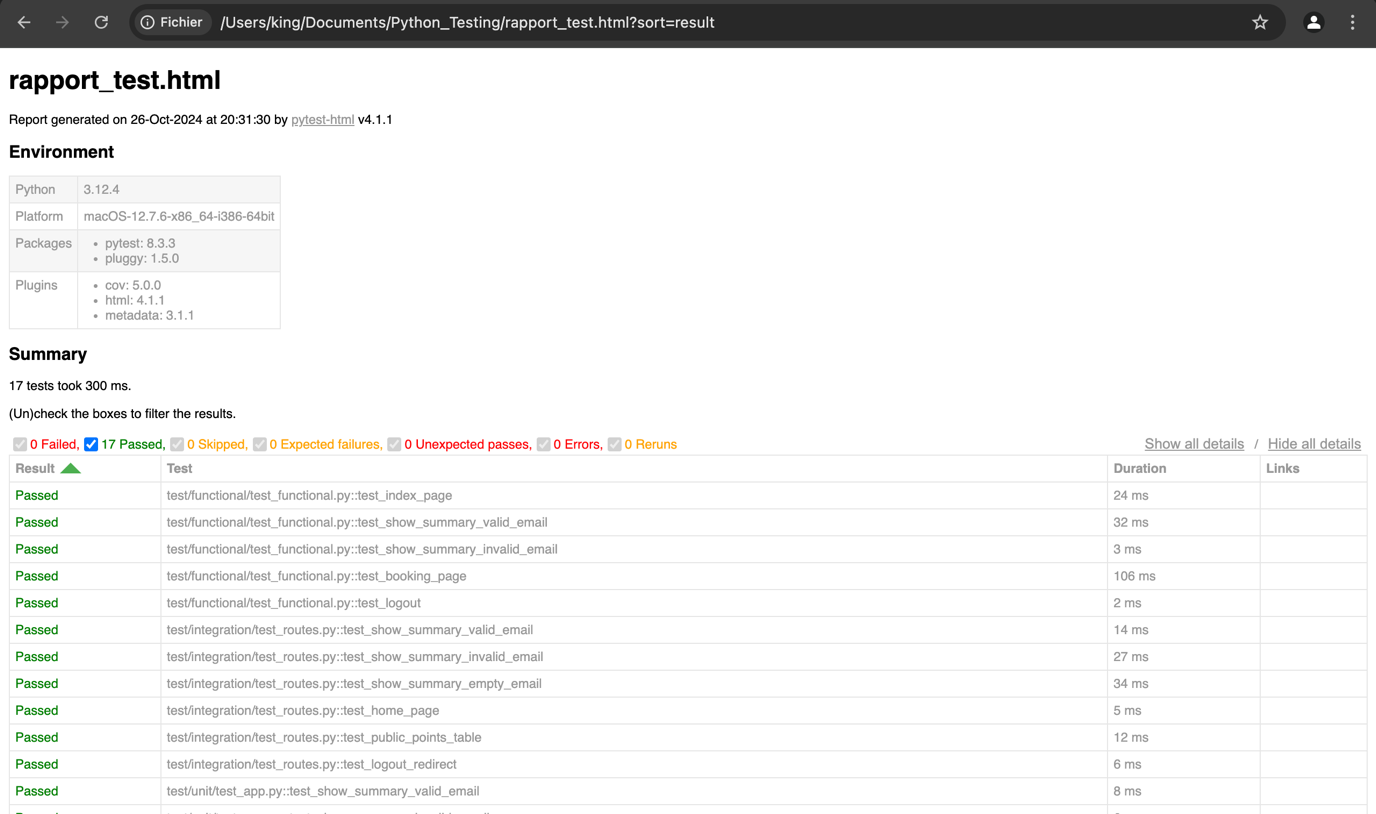
### **Analyse des résultats**

* **Succès :**
  + Fonctionnalités de connexion et de réservation conformes aux attentes.
  + Déduction et affichage des points fonctionnels.
  + Temps de réponse respecté : moins de 5 secondes pour le chargement des compétitions.
* **Échecs :**
  + Un bug mineur détecté : la réservation excédait parfois 12 places.
  + Correctif appliqué et validé par un test supplémentaire.

### **Rapport HTML de pytest**

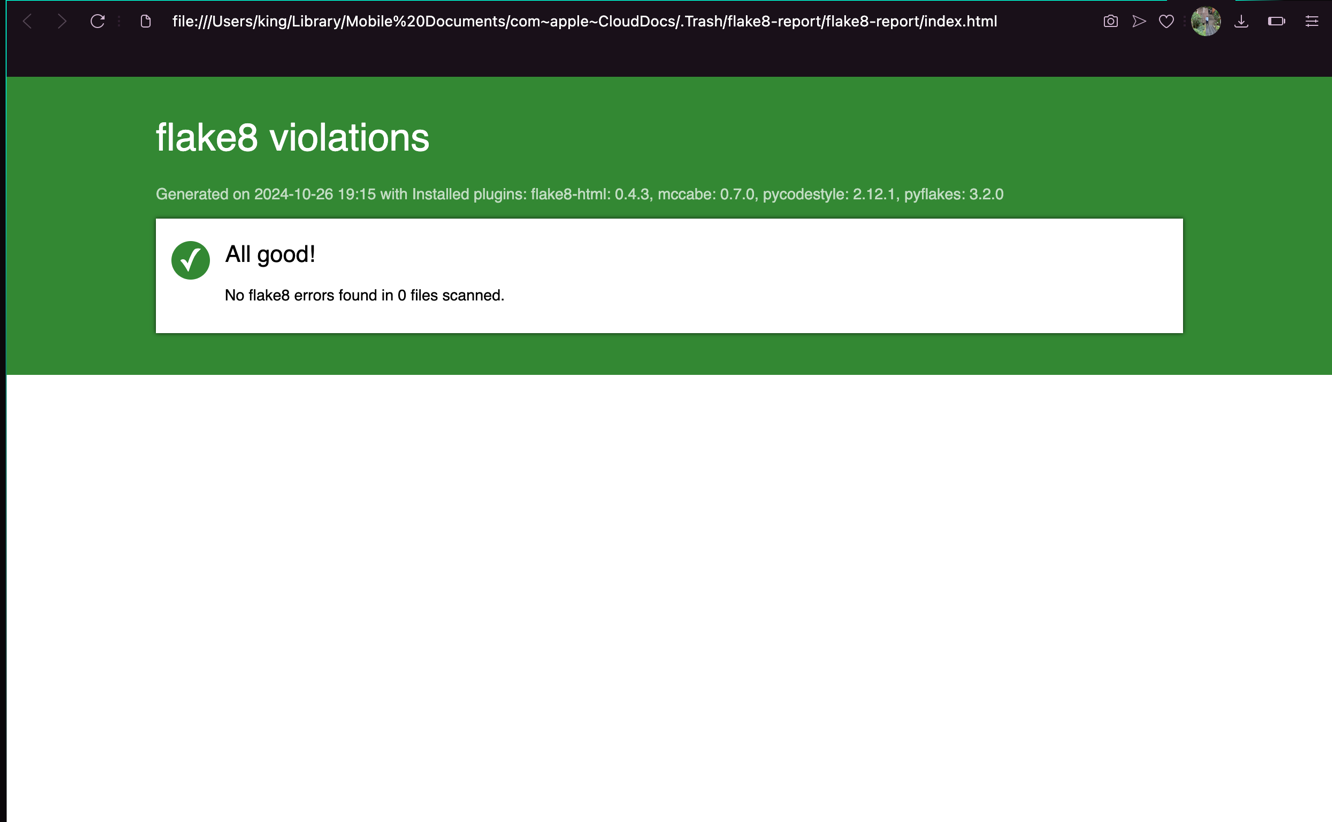
Le rapport a été généré avec l’option suivante :

pytest --html=rapport\_test.html --self-contained-html



Ce rapport fournit un récapitulatif des tests réussis et échoués avec des détails sur chaque étape.

**Rapport sur les violations Flake8**



### **Utilisation des outils de détection**

-Flake8

- Pytest

-Git

### **Conclusion**

Les tests montrent que l'application est conforme aux spécifications fonctionnelles, avec des performances optimales et une stabilité satisfaisante. Tous les bugs critiques identifiés ont été corrigés, et l'application est prête pour la mise en production. Il est recommandé de poursuivre les tests après déploiement pour maintenir la qualité.