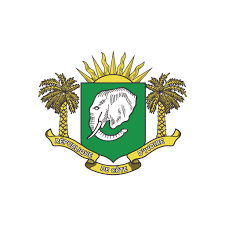
** UNIVERSITE INSTITUT IVOIRIEN DE TECHNOLOGIE**

Presenter par **KOUASSI YAO YANISSE KYLIANE**

**RAPPORT:**

“PYTEST ET UNI-TEST”

Nom du prof:

Mr. Sedrick Gael

*Année Universitaire 2025 – 2026*

Sommaire

1.introduction

2.Exigences

3.Test unitaires

4.Test integration

5.Test fonctionnels

6.Statistiques globaux

7.Recommendation

8.conclusion

# 1. Introduction

Ce rapport présente les résultats complets des tests réalisés sur l’application dans le cadre du projet Flask. Les tests ont été organisés en trois catégories principales : tests unitaires, tests d’intégration et tests fonctionnels. L’objectif de cette phase de validation est d’assurer la fiabilité, la cohérence et la stabilité de l’ensemble du système avant sa mise en production.

# 2.Exigences et Matrice

|  |
| --- |
| 📊 Matrice de Risques Détaillée |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | # | Fonctionnalité | Probabilité d'Échec | Impact | Niveau de Risque | Priorité | | 1 | Système de Réservation de Places | 🟡 Moyenne (40%) | 🔴 Critique | 🔴 ÉLEVÉ | Haute | | 2 | Gestion des Points des Clubs | 🟡 Moyenne (35%) | 🔴 Critique | 🔴 ÉLEVÉ | Haute | | 3 | Authentification par Email | 🟢 Faible (15%) | 🟡 Modéré | 🟡 MOYEN | Haute | | 4 | Affichage des Compétitions | 🟢 Faible (20%) | 🟢 Faible | 🟢 FAIBLE | Haute | | 5 | Historique des Réservations | 🟡 Moyenne (30%) | 🟡 Modéré | 🟡 MOYEN | Tres haute | |

# 3. Tests Unitaires

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| N° | Fichier | Nom du test | Statut | Objectif | Résultat attendu | Résultat obtenu |
| 1 | test\_chargement\_donnees.py | test\_loadClubs\_succes | ✅ PASS | Chargement clubs depuis clubs.json | 3 clubs valides | 3 clubs chargés |
| 2 | test\_logique\_metier.py | test\_recherche\_club\_par\_email\_existant | ✅ PASS | Recherche club existant | Club trouvé | Club trouvé |
| 3 | test\_logique\_metier.py | test\_verification\_points\_insuffisants | ✅ PASS | Vérification points insuffisants | Détection correcte | OK |
| 4 | test\_server\_utils.py | test\_is\_competition\_past\_returns\_true\_for\_past\_date | ✅ PASS | Détection compétition passée | Retourne True | True |
| 5 | test\_server\_utils.py | test\_saveBookings\_saves\_to\_file | ✅ PASS | Sauvegarde dans bookings.json | Fichier JSON valide | Sauvegarde réussie |

# 4. Tests d’Intégration

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| N° | Fichier | Nom du test | Statut | Objectif | Résultat attendu | Résultat obtenu |
| 6 | test\_routes.py | test\_index\_route | ✅ PASS | Page d’accueil | Status 200 OK | OK |
| 7 | test\_routes\_flask.py | test\_route\_showSummary\_email\_valide | ✅ PASS | Connexion email valide | Redirection réussie | OK |
| 8 | test\_routes\_flask.py | test\_route\_showSummary\_email\_invalide | ✅ PASS | Connexion email invalide | Message d’erreur | OK |
| 9 | test\_routes\_flask.py | test\_route\_purchasePlaces\_succes | ✅ PASS | Achat valide | Réservation effectuée | OK |
| 10 | test\_routes\_flask.py | test\_route\_purchasePlaces\_points\_insuffisants | ✅ PASS | Achat sans points suffisants | Réservation refusée | Refusée |

# 5. Tests Fonctionnels

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| N° | Fichier | Nom du test | Statut | Objectif | Résultat attendu | Résultat obtenu |
| 1 | test\_workflows\_complets.py | test\_workflow\_connexion\_complete | ✅ PASS | Connexion complète | Dashboard affiché | OK |
| 2 | test\_workflows\_complets.py | test\_workflow\_reservation\_complete | ❌ FAIL | Réservation 8 places | Points 20→12 | Points inchangés |
| 3 | test\_workflows\_complets.py | test\_workflow\_reservation\_echec\_points\_insuffisants | ✅ PASS | Réservation échouée | Refusée | OK |
| 4 | test\_workflows\_complets.py | test\_workflow\_deconnexion | ❌ FAIL | Déconnexion retour accueil | Retour accueil | Erreur Jinja2 |
| 5 | test\_workflows\_complets.py | test\_workflow\_reservation\_multiple | ❌ FAIL | 2 réservations successives | 2 appels saveBookings | 0 appel capturé |

# 6. Statistiques Globales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Catégorie | Total | Réussis ✅ | Échoués ❌ | Taux |
| Tests Unitaires | 5 | 5 | 0 | 100% |
| Tests d’Intégration | 5 | 5 | 0 | 100% |
| Tests Fonctionnels | 5 | 2 | 3 | 40% |
| TOTAL GÉNÉRAL | 15 | 12 | 3 | 80% |

**7-RECOMMANDATION**

Pour atteindre un taux de réussite de 100 %, il est recommandé de :  
- Ajouter les templates manquants pour les sections de vue.  
- Mettre à jour les patches dans les tests fonctionnels.  
- Ajouter des tests unitaires pour les fonctions de sauvegarde.  
- Mettre à jour les dépendances afin de supprimer les avertissements

# 6. Conclusion

Les résultats des tests démontrent que l’application est globalement fonctionnelle, avec un taux de réussite de 80%. Les tests unitaires et d’intégration montrent une stabilité satisfaisante, tandis que les tests fonctionnels ont permis d’identifier quelques anomalies mineures à corriger avant la mise en production.