# RAPPORT DE TEST ET VALIDATION LOGICIELLE

## Projet : Online School

### Introduction

Ce rapport présente les tests effectués pour l’application "Online School", une plateforme d'e-learning interactive et centralisée. Le projet vise à numériser les activités éducatives et à améliorer la collaboration entre étudiants, enseignants et administrateurs. Ce document décrit les différents tests réalisés, les résultats obtenus, et les éventuelles erreurs identifiées ainsi que les recommandations pour garantir la qualité du logiciel.

### 1. Contexte et objectifs

#### Contexte du projet

L’application "Online School" permet :  
- La gestion des cours, quiz et devoirs.  
- L’interaction via forums et discussions.  
- Le suivi des évaluations et des performances académiques.

#### Objectifs des tests

1. Garantir la conformité aux spécifications fonctionnelles.  
2. Identifier et corriger les bugs existants.  
3. Valider l’interopérabilité entre les modules.  
4. Améliorer la qualité globale et l’expérience utilisateur.

### 2. Plan de test

#### 2.1. Types de tests

- Tests unitaires : Validation des composants isolés (modèles, vues, URL).  
- Tests d’intégration : Vérification des interactions entre modules (ex : forums et quiz).  
- Tests fonctionnels : Validation des cas d’usage (navigation, soumission de quiz).  
- Tests de validation : Confirmation que le système respecte les exigences initiales.

#### 2.2. Outils utilisés

- Pytest : Framework principal pour les tests unitaires et d’intégration.  
- Coverage : Mesure de la couverture des tests.  
- JsonResponse : Simulation des requêtes HTTP POST et GET.

### 3. Cas de test

#### 3.1. Gestion des utilisateurs

Fonctionnalités testées :

- Authentification des étudiants.  
- Redirection basée sur le type d’utilisateur (enseignant/étudiant).  
- Mise à jour du profil.

|  |  |  |  |
| --- | --- | --- | --- |
| **Cas de test** | **Entrées** | **Attendus** | **Résultat** |
| Connexion utilisateur | Username, password | Redirection vers tableau de bord | Succès |
| Mise à jour du profil | Bio, nom, prénom, email | Profil mis à jour | Succès |
| Accès non authentifié | Aucun | Redirection vers la page login | Succès |

#### 3.2. Gestion des cours

Fonctionnalités testées :

- Accès à la liste des cours.  
- Affichage des détails d’un cours.

|  |  |  |  |
| --- | --- | --- | --- |
| **Cas de test** | **Entrées** | **Attendus** | **Résultat** |
| Liste des cours | Aucune | Affichage des cours disponibles | Succès |
| Détails d’un cours | Slug du cours | Informations du cours affichées | Succès |
| Accès à un cours inexistant | Slug invalide | Page 404 | Succès |

#### 3.3. Gestion des quiz

Fonctionnalités testées :

- Affichage de la liste des quiz.  
- Participation à un quiz.  
- Soumission des réponses.

|  |  |  |  |
| --- | --- | --- | --- |
| **Cas de test** | **Entrées** | **Attendus** | **Résultat** |
| Liste des quiz | Aucune | Liste des quiz affichée | Succès |
| Participation à un quiz | ID du quiz | Questions du quiz affichées | Succès |
| Soumission des réponses | Réponses JSON | Score calculé et sauvegardé | Succès |
| Soumission sans réponses | Réponses JSON vides | Message d’erreur | Succès |

#### 3.4. Gestion des forums

Fonctionnalités testées :

- Création de nouveaux sujets.  
- Participation à un sujet existant.

|  |  |  |  |
| --- | --- | --- | --- |
| **Cas de test** | **Entrées** | **Attendus** | **Résultat** |
| Création de sujet | Titre, question | Sujet créé avec succès | Succès |
| Participation à un sujet | Slug du sujet, réponse | Réponse ajoutée au sujet | Succès |
| Accès à un sujet inexistant | Slug invalide | Page 404 | Succès |

### 4. Analyse des résultats

#### Tests réussis

- Authentification et redirection utilisateur.  
- Gestion des cours (affichage, détails).  
- Participation et soumission des quiz.  
- Gestion des forums (création et participation).

#### Tests échoués

Liste des tests échoués :

|  |  |
| --- | --- |
| **Cas de test** | **Erreur rencontrée** |
| test\_create\_salon | TypeError: Classe() got an unexpected keyword argument 'name' |
| test\_message\_creation | TypeError: Classe() got an unexpected keyword argument 'name' |
| test\_create\_instructor | TypeError: Classe() got an unexpected keyword argument 'name' |
| test\_instructor\_many\_to\_many\_\* | TypeError: Classe() got an unexpected keyword argument 'name' |
| test\_dashboard\_view\_as\_instructor | Direct assignment to the reverse side of a many-to-many set is prohibited. |
| test\_course\_add\_view | TypeError: Classe() got an unexpected keyword argument 'titre' |
| test\_messages\_view | TypeError: Classe() got an unexpected keyword argument 'nom' |
| test\_profile\_view | AssertionError: 302 != 200 |
| test\_quiz\_edit\_view | IntegrityError: NOT NULL constraint failed: quiz\_quiz.temps |
| test\_post\_forum | AssertionError: False |
| test\_create\_devoir | IntegrityError: NOT NULL constraint failed: school\_chapitre.matiere\_id |
| test\_create\_student | ValueError: Cannot assign "Licence 1": "Classe.niveau" must be a "Niveau" instance |

### 5. Recommandations

1. Corriger les problèmes liés aux noms de champs mal définis dans les modèles (Classe).  
2. Assurer que toutes les dépendances nécessaires (comme Matiere et Niveau) sont créées avant les tests.  
3. Revoir les paramètres des vues pour gérer correctement les erreurs et les cas de bord.  
4. Mettre à jour les tests échoués pour couvrir les scénarios réalistes.

### 6. Conclusion

Les tests effectués sur "Online School" ont permis d'identifier plusieurs points d'amélioration nécessaires pour stabiliser et compléter l'application. Les résultats des tests montrent que des ajustements sur les modèles, les vues, et les relations entre les entités sont nécessaires avant un déploiement final.