Plan de test

1. Introduction :

Ce document décrit l'approche utilisé pour réaliser les tests qui ont comme but de garantir le bon fonctionnement du logiciel

2. Objectif :

Valider et tester les différentes fonctionnalités de l'application Web_app

3. Rôles et responsabilités :

Développeurs SW : Responsables de développer les cas de test et d'implémenter les tests unitaires.

Responsable qualité : Responsable de la validation des cas de test et des tests unitaires, implémentation des tests d'intégration

Chef de projet : Responsable de la planification et de la coordination des tests ainsi que transmettre les résultats au client.

4. Stratégie de test :

Cette stratégie de test couvre différents aspects de l'application " Librairie EPSIC" et vise à garantir sa qualité.
Elle comprend une variété de tests pour couvrir divers scénarios.

Les tests recommandés sont :

Tests Unitaires :

Objectif : Tester chaque composant de l'application individuellement pour vérifier leur bon fonctionnement.

Exemples de tests :

Tester les méthodes des modèles (par exemple, la création, la mise à jour et la suppression des livres).

Tester les vues et les fonctions de contrôle (par exemple, la gestion des utilisateurs, l'affichage des listes de livres).

Tester les fonctions utilitaires

Tests d'Intégration :

Objectif : Tester l'intégration entre différents composants de l'application pour vérifier leur collaboration.

Exemples de tests :

Tester l'intégration entre les modèles et les vues

Tester l'intégration avec des services externes

Tests de Sécurité :

Objectif : Identifier et corriger les vulnérabilités de sécurité potentielles de l'application.

Exemples de tests :

Tester l'injection SQL en essayant d'injecter des commandes SQL dans les formulaires de l'application.

Tester les failles de sécurité liées à l'authentification et à l'autorisation (par exemple, l'accès non autorisé à des fonctionnalités réservées aux admin).

Tests End-to-End (E2E) :

Objectif : Tester l'ensemble du flux de l'application, depuis le début jusqu'à la fin, en simulant le comportement de l'utilisateur final.

5. Environnement de test :

Infrastructure et serveurs :

Les tests sont déployés en local et un pipeline CI/CD a été mise en place qui run les tests à chaque push or pull request, ou si on déclenche le job manuellement

Outils de test :

Les outils utilisés sont :

1 . pytest

2. github (pour le pipeline)

Données de test :

Données simulées, et des données simulées en utilisant des fixtures comme on peut voir ici par exemple :

```
@classmethod
def setUpTestData(cls):
    # Create a book to be deleted in each test method
    cls.book = Book.objects.create(author="Dominique Rogues", title="Pico Bogue")
```

Cas de test IT_indexView

Description	tester le comportement de la view « index » dans des conditions normales d'utlisation
Prérequis	Le logiciel est installé et fonctionnel
Étapes du test	1. Création d'un client de test 2. Le client fait une requête GET 3. Vérification du code de statut 4. Vérification des données
Résultat attendu	Code de réponse 200 et numéro de visites incrémenté en 1

Cas de test IT_DeleteBook (using fixtures)

Description	Tester le comportement de la suppression d'un livre dans l'interface d'administration
Prérequis	Le logiciel est installé et fonctionnel et un super-admin est créé pour accéder à l'interface
Étapes du test	1. Création d'un livre. 2. Vérifier que le livre a bien été crée 3. Supprimer le livre 4. Vérifier que le livre a bien été supprimé
Résultat attendu	Le livre est supprimé en bd

Cas de test UT_book_list (using mocks)

Description	Ce cas de test vise à vérifier le comportement de la view book_list dans l'application.
Prérequis	Le logiciel est installé et fonctionnel
Étapes du test	1. Arrange : Des livres fictifs sont préparés pour simuler les données récupérées de la base de données. 2. Act : La vue book_list est appelée avec des données fictives en entrée. 3. Assert : les asserts sont effectuées pour vérifier que est renvoie une réponse HTTP avec le code 200 et que les noms des auteurs et les titres des livres sont présents dans le contenu de la réponse.
Résultat attendu	code 200 et livres présents dans la réponse