

Préconisations

Mise en place de la revue de code :

- *Description* : La revue de code est une pratique où les développeurs examinent le code source les uns des autres pour identifier et corriger des erreurs, des problèmes de conception et des incohérences. Cette revue peut être effectuée de manière formelle ou informelle, selon les besoins de l'équipe.
- *Intérêt* : La revue de code permet de détecter et de corriger les erreurs plus tôt dans le processus de développement, réduisant ainsi les coûts et le temps consacrés à la correction de bogues plus tard dans le cycle de développement. Elle contribue également à améliorer la qualité globale du code, à partager les connaissances entre les membres de l'équipe et à garantir que les normes de codage sont respectées.

Formation aux bonnes pratiques :

- *Description* : La formation aux bonnes pratiques consiste à fournir aux membres de l'équipe des formations et des ressources pour les aider à adopter des méthodologies et des approches de développement de logiciels de haute qualité. Cela peut inclure l'apprentissage des principes de conception, des normes de codage, des méthodes de test, etc.
- *Intérêt* : La formation aux bonnes pratiques permet d'améliorer les compétences des membres de l'équipe en matière de développement de logiciels. Cela garantit que l'équipe travaille de manière plus efficace, produit un code de meilleure qualité, et suit des normes de développement cohérentes.

Installation d'un logiciel d'analyse de code (type Sonar) :

- *Description* : L'installation d'un logiciel d'analyse de code, comme Sonar, permet d'automatiser la détection de problèmes potentiels dans le code source. Ces outils analysent le code à la recherche d'erreurs, de vulnérabilités, de violations des normes de codage et de mauvaises pratiques.
- *Intérêt* : Les outils d'analyse de code aident à identifier les problèmes de manière automatisée, ce qui permet de réduire le temps nécessaire pour trouver des erreurs et des bogues. Ils améliorent également la cohérence du code en s'assurant que les normes de codage sont respectées.

Mise en place de tests unitaires :

- *Description* : Les tests unitaires sont des tests automatisés qui vérifient le bon fonctionnement des composants individuels (unités) d'un logiciel. Ils permettent de s'assurer que chaque petite partie du code fonctionne comme prévu.
- *Intérêt* : Les tests unitaires aident à identifier les erreurs et les défauts de manière précoce. Ils garantissent que les modifications ultérieures du code ne cassent pas les fonctionnalités existantes, ce qui facilite la maintenance et l'évolution du logiciel.

Création d'une documentation :

- *Description* : La documentation consiste en des descriptions écrites des fonctionnalités, de l'architecture, des processus et des composants du logiciel. Elle peut prendre la forme de manuels, de commentaires dans le code, de guides d'utilisation, etc.
- *Intérêt* : La documentation facilite la compréhension du logiciel par les membres de l'équipe et les utilisateurs finaux. Elle contribue à la maintenance, à la résolution de problèmes et à la collaboration au sein de l'équipe de développement.

Créer un système de CI/CD (Intégration Continue / Déploiement Continu) :

- *Description* : CI/CD est une pratique qui automatise la construction, les tests, et le déploiement d'applications. L'intégration continue consiste à fusionner régulièrement les modifications de code dans un référentiel partagé, tandis que le déploiement continu automatise la livraison du logiciel aux environnements de production.
- *Intérêt* : CI/CD accélère le cycle de développement en automatisant des tâches répétitives. Cela garantit que les nouvelles fonctionnalités ou les correctifs sont rapidement disponibles pour les utilisateurs. De plus, cela réduit le risque d'erreurs humaines lors du déploiement.

Hashage des mots de passes avec Bcrypt

- *Description* : Le hashage des mots de passe est une technique de sécurité qui consiste à convertir un mot de passe en une chaîne de caractères aléatoires, appelée hachage, à l'aide d'une fonction de hachage. Le résultat du hachage est stocké dans la base de données au lieu du mot de passe en clair.
- *Intérêt* : Le hashage des mots de passe est essentiel pour renforcer la sécurité des systèmes informatiques, en particulier des applications qui nécessitent une authentification des utilisateurs.

Utilisation d'un singleton pour la connexion à la BDD