

Amélioration du projet

Des améliorations ont été apportées au projet initial.

Corrections d'anomalies

Attacher une tâche à un utilisateur lors de la création

1. Lorsqu'une tâche est créée, elle doit être automatiquement rattachée à l'utilisateur authentifié.
2. Les tâches déjà créées doivent être rattachées à un utilisateur "anonyme".
3. L'auteur d'une tâche ne peut pas être modifié lors de la modification de la tâche.

Choisir un rôle pour un utilisateur lors de la création

1. Lors de la création d'un utilisateur, permettre la sélection d'un rôle (ROLE_USER ou ROLE_ADMIN).
2. Permettre la modification du rôle d'un utilisateur existant.

Implémentation de nouvelles fonctionnalités

Autorisation d'accès aux pages de gestion des utilisateurs

1. Restreindre l'accès aux pages de gestion des utilisateurs aux administrateurs (ROLE_ADMIN).
2. Suppression de tâches par les utilisateurs autorisés
3. Les tâches rattachées à l'utilisateur "Anonyme" peuvent être supprimées uniquement par les administrateurs (ROLE_ADMIN).

Amélioration du code

Mise à jour du projet

Le projet a récemment subi une mise à jour majeure, passant de Symfony 3.3 à Symfony 7.1, ainsi que de PHP 5.3 à PHP 8.3. Cette évolution représente un saut significatif en termes de fonctionnalités, de performances et de sécurité. Avec Symfony 7.1, nous bénéficions des dernières avancées du framework, telles que des améliorations dans la gestion des composants, une meilleure prise en charge des

technologies modernes et une optimisation des performances. De même, la mise à jour vers PHP 8.3 nous permet d'exploiter les nouvelles fonctionnalités du langage, comme les améliorations de typage, les performances accrues, et les nouvelles syntaxes qui simplifient le développement. Ces changements assurent la pérennité du projet tout en permettant une meilleure maintenabilité et évolutivité du code.

Mise en place de Webpack Encore

Webpack Encore qui est un bundle symfony nous permettant de profiter de la gestion performante des assets de Webpack. Nous avons pu aussi minifier le code javascript et css en production pour gagner en performance.

Audit de qualité et maintenabilité du code

Utilisation de Codacy, un outil de qualité de code automatisé qui permet de mesurer, surveiller et améliorer la qualité du code. Utilisation de Code Climate, un outil de qualité de code automatisé qui permet de mesurer la maintenabilité du code.

Implémentation de tests

1. Tests unitaires avec PHPUnit.
2. Tests fonctionnels.
3. Données de test pour vérifier les différents cas d'utilisation.
4. Rapport de couverture de code.

Ajout d'une documentation technique

Une documentation technique a été ajoutée au projet pour fournir des directives claires sur plusieurs aspects essentiels. Elle explique en détail quels fichiers doivent être modifiés lors des mises à jour ou des personnalisations, en précisant pourquoi ces changements sont nécessaires pour maintenir la cohérence du système. Le processus d'authentification y est également détaillé, décrivant comment les utilisateurs s'authentifient dans l'application, depuis l'envoi des informations d'identification jusqu'à la validation de l'accès. Enfin, la documentation indique précisément où les utilisateurs sont stockés, en mettant en évidence la structure des données et les mécanismes de gestion des utilisateurs dans le projet.