

[← My path](#) APPROVED PROJECT

Améliorez une application existante de ToDo & Co

PROJECT BRIEF

COURSES

ASSESSMENT

 100 hours

Last updated on Friday, March 4, 2022



C'est parti ! Il faut faire évoluer l'application de ToDo & Co

Contexte

Vous venez d'intégrer une startup dont le cœur de métier est une application permettant de gérer ses tâches quotidiennes. L'entreprise vient tout juste d'être montée, et l'application a dû être développée à toute vitesse pour permettre de montrer à de potentiels investisseurs que le concept est viable (on parle de Minimum Viable Product ou MVP).

Le choix du développeur précédent a été d'utiliser le framework PHP Symfony, un framework que vous commencez à bien connaître !

Bonne nouvelle ! **ToDo & Co** a enfin réussi à lever des fonds pour permettre le développement de l'entreprise et surtout de l'application.

Votre rôle ici est donc d'améliorer la qualité de l'application. La qualité est un concept qui englobe bon nombre de sujets : on parle souvent de qualité de code, mais il y a également la qualité perçue par l'utilisateur de l'application ou encore la qualité perçue par les collaborateurs de l'entreprise, et enfin la qualité que vous percevez lorsqu'il vous faut travailler sur le projet.

Ainsi, pour ce dernier projet de spécialisation, vous êtes dans la peau d'un développeur expérimenté en charge des tâches suivantes :

- l'implémentation de nouvelles fonctionnalités ;
- la correction de quelques anomalies ;
- et l'implémentation de tests automatisés.

Il vous est également demandé d'analyser le projet grâce à des outils vous permettant d'avoir une vision d'ensemble de la qualité du code et des différents axes de performance de l'application.

Il ne vous est pas demandé de corriger les points remontés par l'audit de qualité de code et de performance. Cela dit, si le temps vous le permet, **ToDo & Co** sera ravi que vous réduisiez la dette technique de cette application.

Description du besoin

Corrections d'anomalies

Une tâche doit être attachée à un utilisateur

Actuellement, lorsqu'une tâche est créée, elle n'est pas rattachée à un utilisateur. Il vous est demandé d'apporter les corrections nécessaires afin qu'automatiquement, à la sauvegarde de la tâche, l'utilisateur authentifié soit rattaché à la tâche nouvellement créée.

Lors de la modification de la tâche, l'auteur ne peut pas être modifié.

Pour les tâches déjà créées, il faut qu'elles soient rattachées à un utilisateur "anonyme".

Choisir un rôle pour un utilisateur

Lors de la création d'un utilisateur, il doit être possible de choisir un rôle pour celui-ci. Les rôles listés sont les suivants :

- rôle utilisateur (*ROLE_USER*) ;
- rôle administrateur (*ROLE_ADMIN*).

Lors de la modification d'un utilisateur, il est également possible de changer le rôle d'un utilisateur.

Implémentation de nouvelles fonctionnalités

Autorisation

Seuls les utilisateurs ayant le rôle administrateur (*ROLE_ADMIN*) doivent pouvoir accéder aux pages de gestion des utilisateurs.

Les tâches ne peuvent être supprimées que par les utilisateurs ayant créé les tâches en question.

Les tâches rattachées à l'utilisateur "anonyme" peuvent être supprimées uniquement par les utilisateurs ayant le rôle administrateur (*ROLE_ADMIN*).

Implémentation de tests automatisés

Il vous est demandé d'implémenter les tests automatisés (tests unitaires et fonctionnels) nécessaires pour assurer que le fonctionnement de l'application est bien en adéquation avec les demandes.

Ces tests doivent être implémentés avec **PHPUnit** ; vous pouvez aussi utiliser Behat pour la partie fonctionnelle.

Vous prévoyez des données de tests afin de pouvoir prouver le fonctionnement dans les cas explicités dans ce document.

Il vous est demandé de fournir un rapport de couverture de code au terme du projet. Il faut que le taux de couverture soit supérieur à 70 %.

Documentation technique

Il vous est demandé de produire une documentation expliquant comment l'implémentation de l'authentification a été faite. Cette documentation se destine aux prochains développeurs juniors qui rejoindront l'équipe dans quelques semaines. Dans cette documentation, il doit être possible pour un débutant avec le framework Symfony de :

- comprendre quel(s) fichier(s) il faut modifier et pourquoi ;
- comment s'opère l'authentification ;
- et où sont stockés les utilisateurs.

S'il vous semble important de mentionner d'autres informations , n'hésitez pas à le faire.

Par ailleurs, vous ouvrez la marche en matière de collaboration à plusieurs sur ce projet. Il vous est également demandé de produire un document expliquant comment devront procéder tous les développeurs souhaitant apporter des modifications au projet.

Ce document devra aussi détailler le processus de qualité à utiliser ainsi que les règles à respecter.

Audit de qualité du code & performance de l'application

Les fondateurs souhaitent pérenniser le développement de l'application. Cela dit, ils souhaitent dans un premier temps faire un état des lieux de la dette technique de l'application.

Au terme de votre travail effectué sur l'application, il vous est demandé de produire un audit de code sur les deux axes suivants : la qualité de code et la performance.

Bien évidemment, il vous est fortement conseillé d'utiliser des outils vous permettant d'avoir des métriques pour appuyer vos propos.

Concernant l'audit de performance, l'usage de Blackfire est obligatoire. Ce dernier vous permettra de produire des analyses précises et adaptées aux évolutions futures du projet.

De l'aide pour aborder le projet étape par étape

Afin de fluidifier votre avancement voici une proposition de manière de travailler :

- **Étape 1** - Prenez connaissance entièrement de l'énoncé et des spécifications détaillées.
- **Étape 2** - Récupérez le MVP ici : [clonez ce projet](#).

- **Étape 3** - Créez votre repository [GitHub](#) pour le projet et poussez-y l'ensemble du projet.
- **Étape 4** - Créez l'ensemble des issues sur le repository GitHub
(https://github.com/username/nom_du_repo/issues/new).
- **Étape 5** - Faites les estimations de l'ensemble de vos issues.
- **Étape 6** - Entamez le développement de l'application et proposez des pull requests pour chacune des fonctionnalités/issues.
- **Étape 7** - Faites relire votre code à votre mentor (code proposé dans la ou les pull requests), et une fois validée(s) mergez la ou les pull requests dans la branche principale. (Cette relecture sert à valider votre implémentation des bonnes pratiques et la cohérence de votre code. La validation se fera en continu durant les sessions.)
- **Étape 8** - Vérifiez la qualité ainsi que les performances de votre code à chaque commit.
- **Étape 9** - Effectuez une démonstration de l'ensemble de l'application.
- **Étape 10** - Préparez l'ensemble de vos livrables et soumettez-les sur la plateforme.

Prenez le temps de valider chaque étape avec votre mentor afin de vous assurer que vous avancez dans la bonne direction. 😊

Livrables

- Un fichier contenant un lien vers l'ensemble du projet sur un repository GitHub (fichiers PHP/HTML/JS/CSS...)
- Les instructions pour installer le projet (dans un fichier README à la racine du projet)
- Les issues sur le repository GitHub
- Documentation technique concernant l'implémentation de l'authentification (fichier au format PDF)
- L'ensemble des fichiers HTML générés par PHPUnit indiquant le niveau de code coverage de l'application (un minimum de 70 %)
- Document expliquant comment contribuer au projet (fichier markdown ".md")
- Le rapport d'audit de qualité de code et de performance (fichier au format PDF)

Pour faciliter votre passage au jury, déposez sur la plateforme, dans un dossier nommé "*P8_nom_prenom*", tous les livrables du projet. Chaque livrable doit être nommé avec le numéro du projet et selon l'ordre dans lequel il apparaît, par exemple "*P8_01_projet*", "*P8_02_instructionsREADME*", et ainsi de suite.

Soutenance

Il vous sera demandé de présenter le projet en soutenance.

Il vous est demandé de vous mettre en situation réelle : en effet, il s'agit d'un rendez-vous professionnel. Vous vous adresserez à votre directeur technique qui validera que le projet est prêt à être présenté au client ainsi qu'à un développeur qui sera à même de reprendre le projet et/ou de travailler en collaboration avec vous.

Voici, sous forme d'étapes, comment devrait se dérouler la soutenance :

| Étapes | Temps à p: |
|--|--------------|
| Étape 1: <u>Présentation du contexte du projet, analyse du besoin & organisation du</u> | ~ 10 minutes |

projet

- **Présentation du contexte du projet** : il s'agit de rappeler le contexte du projet dans le cadre du parcours que vous suivez.
- **Analyse du besoin** : vous choisirez l'une des fonctionnalités développées et il vous faudra présenter les schémas UML correspondants (diagrammes d'utilisation, de classes et de séquence, modèle de données).
- **Organisation du projet** : vous explicitez la méthodologie adoptée, l'organisation de votre travail (découpage en tâches, priorisation, les estimations...).

Étape 2 : **Démonstration de l'application**

~ 5 minute:

Il s'agit de présenter l'ensemble des fonctionnalités de l'application.

Il vous faudra également présenter les résultats d'audit de qualité et de performance.

Vous exposerez les propositions de solutions pour remédier à la dette technique.

Étape 3 : **Exécution**

~ 5 minute:

Vous présenterez le projet du point de vue technique cette fois-ci. Il s'agit de :

- l'architecture de l'application (où sont les fichiers, et à quoi ils servent);
- les librairies qui ont été choisies pour être mises en place (installation avec Composer) ;
- la documentation expliquant comment l'authentification a été implémentée ;
- la documentation présentant le niveau de couverture de code des tests ;
- la documentation expliquant comment il est possible de contribuer au projet.

Étape 4 : **Questions-Réponses**

~ 5 minute:

Une session de questions-réponses autour de ce que vous aurez exposé sera à mener.

Total**30 minute:**
maximum

Skills



Mettre en œuvre des tests unitaires et fonctionnels



Implémenter de nouvelles fonctionnalités au sein d'une application déjà initiée en suivant un plan de collaboration clair



Lire et retranscrire le fonctionnement d'un morceau de code écrit par d'autres développeurs



Produire un rapport de l'exécution des tests





Analyser la qualité de code et la performance d'une application



Établir un plan pour réduire la dette technique d'une application



Fournir des patches correctifs lorsque les tests le suggèrent



Proposer une série d'améliorations

OPENCCLASSROOMS



OPPORTUNITIES



SUPPORT



FOR BUSINESS



MORE



English



Download on the
App Store

