

Résumé des projets

Code Hub est un forum de discussion en ligne conçu pour permettre aux utilisateurs de partager des articles et de commenter les autres articles. Le cahier des charges pour le développement de ce forum a été fourni par notre école La plateforme.

L'application permet aux visiteurs sans compte de consulter les articles et les commentaires, mais ils ne peuvent pas en ajouter eux-mêmes. Les utilisateurs avec un compte peuvent ajouter des articles et des commentaires. Ils ont également la possibilité de modifier leur profil et de supprimer leur compte si nécessaire.

Les administrateurs ont les mêmes droits qu'un utilisateur standard, mais disposent également de fonctionnalités supplémentaires. Ils peuvent modifier, supprimer et ajouter des utilisateurs au forum. De plus, ils peuvent supprimer des articles ou des commentaires en cas de violation des règles du forum.

La première étape dans la réalisation d'un projet informatique est de construire une équipe solide de développeurs qui peuvent collaborer efficacement pour atteindre les objectifs du projet, nous avons constitué une équipe de quatre développeurs et utilisé une combinaison de méthodes Scrum et Kanban avec Trello pour suivre les tâches.

Le projet a été divisé en petites tâches pour faciliter le travail et assurer une bonne gestion du temps. Nous avons également utilisé GitHub pour le suivi des versions et le partage du code.

Notre projet consiste en un backend (l'API REST réalisée en Symfony et Api-Platform) et plusieurs frontend (nos interfaces utilisateur).

Le projet CodeHub organisée en quatre couches :

- 1- La couche de présentation : Cette couche comprend toutes les vues de notre application, y compris les écrans que les utilisateurs voient et avec lesquels ils interagissent. Elle est composée de trois interface distincts :
 - Interface Mobile : Nous avons utilisé la plateforme Expo qui facilite le développement d'applications mobiles en fournissant des outils, des fonctionnalités et des services prêts à l'emploi. Il simplifie le processus de développement, de test et de déploiement, Expo est construit sur React Native, une bibliothèque populaire pour le développement d'applications mobiles. Et nous avons installé Expo Go sur nos téléphones android pour tester et de prévisualiser notre application développée avec Expo.
 - Interface desktop : Cette interface est une application Python qui utilise une interface graphique tkinter et le module requests pour se connecter à notre API Symfony. L'API Symfony, à son tour, effectue des appels vers l'API OpenAI pour

générer un article à partir d'un titre donné. Ce service était l'une des tâches que j'ai réalisées au sein de mon entreprise d'accueil, Smart Tribune. L'objectif était de concevoir un service permettant à nos clients de générer des réponses à partir de titres de questions. J'ai appliqué le même principe à notre application et j'ai développé un service offrant aux utilisateurs la possibilité de générer des articles IA à partir de titres, en utilisant l'API OpenAI.

- Interface web : c'est un panneau d'administration qui offre aux utilisateurs disposant du rôle d'administrateur l'accès et la possibilité de gérer efficacement les données relatives aux utilisateurs, aux articles et aux commentaires. La décision d'utiliser le Framework React.js a été motivée par ses avantages en termes de modularité, de réutilisabilité du code et de performances. Grâce à la structure basée sur des composants de React.js, j'ai pu créer des interfaces interactives et dynamiques, offrant ainsi une expérience utilisateur améliorée.
- 2- La couche de logique d'application : Cette couche contient la logique métier de notre application, c'est-à-dire les règles qui déterminent comment les données sont manipulées et gérées. Cette couche est principalement gérée par l'API Platform et Symfony, qui fournissent des fonctionnalités pour gérer les entités (User, Article, Comment) et effectuer des opérations CRUD (créer, lire, mettre à jour, supprimer) sur ces entités.
- 3- La couche d'accès aux données : Cette couche est responsable de la communication avec la source de données, dans notre cas la base de données. Cette couche est également gérée par l'API Platform et Symfony, qui utilisent Doctrine pour communiquer avec la base de données et mapper les données en entités PHP.
- 4- La couche de services : Cette couche comprend les différents services de notre application, qui fournissent des fonctionnalités spécifiques à différentes parties de notre application. Dans notre cas, nous avons des services pour gérer l'authentification des utilisateurs et un service qui appelle l'API OpenAI pour générer un article depuis son titre. Ces services sont également gérés par l'API Platform et Symfony.

Pour faciliter l'accès aux données dans l'application, nous avons utilisé l'ORM Doctrine de Symfony. L'ORM (Object-Relational Mapping) est une technique de programmation qui permet de faire correspondre les objets d'une application avec les tables d'une base de données relationnelle. Avec Doctrine, nous avons créé des entités qui représentent les différents types de données dans l'application (par exemple, les articles, les commentaires, les utilisateurs, etc.) et nous avons défini les relations entre ces entités.

Ce projet peut être adapté et utilisé pour créer des forums spécialisés dans divers domaines tels que la technologie, la cuisine, le sport, etc. Cette flexibilité permet aux utilisateurs de s'engager et d'échanger des idées sur des sujets qui les intéressent.

Je vois dans les prochaines fonctionnalités à prévoir dans codehub, divers éléments :

- ◆ Le nombre de vues des articles.
- ◆ Classement des articles par nombre de vues pour promouvoir les contenus les plus populaires.
- ◆ Ajouter des images d'illustrations dans les articles.