

1. Introduction

Ce projet consiste à développer une application web qui permet aux utilisateurs de gérer leurs tâches quotidiennes et suivre leurs projets, en équipe ou individuellement. L'application fournira des notifications en temps réel, gérées par Laravel, pour les rappels de tâches et les mises à jour. Elle visera à offrir une gestion collaborative des tâches dans un environnement simple et intuitif.

2. Objectifs

- Faciliter la gestion des tâches et projets personnels et professionnels.
- Optimiser la collaboration entre les membres d'une équipe avec une gestion fluide des tâches.
- Envoyer des notifications pour les rappels et les délais.
- Offrir une interface conviviale pour une gestion efficace des tâches.

3. Technologies

- **Backend** : Laravel (PHP Framework).
- **Base de données** : MySQL ou PostgreSQL.
- **Notifications** : Notifications Laravel via WebSockets.
- **Frontend** : Blade Templates avec Bootstrap ou TailwindCSS.
- **Authentification** : Laravel Breeze ou Laravel Jetstream.
- **API** : RESTful API avec Laravel Sanctum pour la gestion des sessions sécurisées et l'authentification par tokens.

4. Fonctionnalités principales

4.1 Gestion des utilisateurs

- **Inscription et connexion** : Les utilisateurs peuvent créer des comptes et se connecter de manière sécurisée.
- **Rôles utilisateurs** : Gestion des rôles d'utilisateur (administrateur, membre de l'équipe, etc.).
- **Authentification** : Utilisation de Laravel Breeze/Jetstream pour une gestion complète des sessions.

4.2 Gestion des projets et des tâches

- **Création et gestion de projets** : Les utilisateurs peuvent créer des projets avec des descriptions, des dates de début/fin et des priorités.
- **Tâches associées à un projet** : Chaque projet peut avoir plusieurs tâches, avec la possibilité d'ajouter une description, une date d'échéance, et de suivre l'état de la tâche (en attente, en cours, terminé).
- **Modification et suppression** : Les utilisateurs peuvent modifier ou supprimer des projets et des tâches.

4.3 Attribution des tâches aux membres de l'équipe

- **Collaboration** : Les utilisateurs peuvent créer des équipes, attribuer des membres à un projet, et assigner des tâches spécifiques à chacun.
- **Suivi des contributions** : Historique des modifications et contributions des membres pour chaque tâche.

4.4 Notifications pour les délais et rappels de tâches

- **Rappels de tâches** : Notifications générées à l'approche de la date d'échéance d'une tâche.
- **Alertes de délai** : Notifications automatiques lorsque des tâches dépassent la date limite.
- **Notifications Laravel** : Utilisation des fonctionnalités de notification de Laravel pour envoyer des emails ou des notifications en temps réel.

4.5 Interface utilisateur simple et efficace

- **Vue en temps réel des tâches** : Mises à jour des tâches en utilisant Laravel Echo et WebSockets pour des interactions en temps réel.
- **Design responsive** : Interface adaptable aux mobiles et tablettes, conçue avec Bootstrap ou TailwindCSS.
- **Suivi des modifications** : Affichage des changements en temps réel lorsqu'une tâche est modifiée ou mise à jour par un membre de l'équipe.

5. Architecture de l'application

- **Backend** : Laravel gérera toute la logique de gestion des tâches, la collaboration et les notifications.
- **Frontend** : Utilisation des templates Blade de Laravel avec un design moderne (Bootstrap ou TailwindCSS).
- **WebSockets** : Laravel Echo pour la gestion des notifications en temps réel.
- **Sécurité** : Utilisation de Laravel Sanctum pour sécuriser l'accès aux API et gérer les sessions.

6. Rôles des utilisateurs

6.1 Utilisateur standard

- Peut créer, modifier, et gérer ses propres projets et tâches.
- Peut recevoir des notifications et rappels sur les tâches à effectuer.
- Peut collaborer sur des projets attribués en tant que membre d'équipe.

6.2 Administrateur de projet

- Gestion complète des projets et des tâches.
- Attribution des rôles aux membres d'équipe.

- Suivi des statistiques et de l'avancement des projets.

7. Livrables

- **Application web** fonctionnelle avec les modules cités.
- **Code source** du projet Laravel.
- **Documentation utilisateur** expliquant les fonctionnalités de l'application.
- **Manuel d'installation et de configuration** pour déployer l'application.
- **Tests unitaires** intégrés pour garantir la robustesse de l'application.

8. Contraintes et sécurité

- **Protection des données** : Utilisation du hashage bcrypt pour les mots de passe et gestion des accès via CSRF.
- **Performances** : L'application doit être optimisée pour des interactions rapides et une gestion fluide des projets avec de nombreux utilisateurs.
- **Sécurité des sessions** : Laravel Sanctum sera utilisé pour la gestion des tokens et des sessions sécurisées.

9. Maintenance et évolutivité

- **Plan de maintenance** pour corriger les bugs et améliorer les fonctionnalités.
- **Évolutivité** : Prévoir des mises à jour futures telles que l'intégration avec des outils tiers (calendriers Google, Slack, etc.).