

Cahier des Charges pour le Projet de Gestionnaire de Stock et de Livraison(GSL)

1. Contexte du Projet

Ce projet est conçu dans le cadre du programme Fullstack PHP 2024. L'objectif est de développer une application web de gestion de stock et de livraison en utilisant Laravel pour le backend et Vue.js pour le frontend. Le projet doit démontrer les compétences techniques requises dans le référentiel du programme.

2. Objectifs du Projet

- Développer une application robuste de gestion de stock et de livraison.
- Assurer une séparation claire des responsabilités entre le backend (Laravel) et le frontend (Vue.js).
- Garantir la sécurité, la qualité du code, et la performance de l'application.
- Mettre en œuvre des pratiques de développement modernes comme les tests automatisés, l'intégration continue, et la conteneurisation.

3. Fonctionnalités de l'Application

Backend (Laravel)

- **Gestion des utilisateurs** : Authentification et autorisation sécurisées avec JWT.
- **Gestion des stocks** : CRUD pour les produits, gestion des catégories de produits, gestion des fournisseurs.
- **Gestion des livraisons** : Suivi des commandes et des livraisons, gestion des statuts des livraisons.
- **Reporting** : Génération de rapports sur les stocks et les livraisons.
- **Intégration API** : Interface RESTful pour l'intégration avec le frontend.
- **Sécurité** : Middleware pour la gestion des erreurs et la vérification des autorisations.

Frontend (Vue.js)

- **Interface utilisateur** : Composants personnalisés pour gérer les produits, les stocks, les livraisons et les utilisateurs.
- **Navigation** : Utilisation de Vue Router pour une navigation fluide, y compris des routes protégées.
- **Gestion de l'état global** : Utilisation de Vuex pour la gestion de l'état global de l'application.
- **Formulaires** : Gestion des formulaires avec validation en temps réel.
- **Responsivité** : Interface adaptative pour divers appareils (desktop, mobile, tablette).
- **Animations et transitions** : Amélioration de l'expérience utilisateur avec des animations fluides.
- **Tests** : Tests unitaires et de bout en bout avec Jest ou Mocha.

4. Planification du Projet

4.1 Sélection de l'Idée et Élaboration du Cahier des Charges

- **Analyse des besoins** : Identifier les besoins spécifiques du projet en fonction du contexte du client.
- **Définition des fonctionnalités** : Établir une liste exhaustive des fonctionnalités requises.
- **Documentation** : Rédaction d'un cahier des charges détaillé.

4.2 Planification et Gestion de Projet avec Jira

- **Planification des tâches** : Décomposition du projet en phases et tâches détaillées.
- **Suivi des tâches** : Utilisation de Jira pour la gestion des tâches, le suivi de progression, et la gestion des sprints.

4.3 Développement Backend et Frontend

- **Backend (Laravel)**
 - Mise en place de l'environnement de développement.
 - Création de modèles, contrôleurs, et migrations pour la base de données.
 - Développement des API RESTful.
 - Mise en place des tests unitaires et des middlewares de sécurité.
- **Frontend (Vue.js)**
 - Mise en place de l'environnement Vue.js.
 - Développement de composants Vue.js.
 - Mise en œuvre de la gestion d'état avec Vuex.
 - Configuration des routes et protection des routes avec Vue Router.

5. Implémentation de la Sécurité

- **Authentification et autorisation** : Utilisation de JWT pour protéger les routes API et sécuriser l'application.
- **Gestion des erreurs** : Utilisation de middlewares pour la gestion des erreurs.
- **Protection des données sensibles** : Sécurisation des communications et des données sensibles.

6. Tests et Qualité du Code

- **Tests unitaires** : Couverture de tous les contrôleurs et composants clés.
- **Tests de bout en bout** : Scénarios de test couvrant toutes les fonctionnalités principales.
- **Analyse de code** : Utilisation de ESLint pour maintenir un code propre et robuste.

7. Intégration Continue et Déploiement

- **Pipelines CI/CD** : Mise en place de pipelines pour automatiser les tests et le déploiement avec GitHub Actions.

- **Conteneurisation** : Utilisation de Docker pour conteneuriser l'application et faciliter le déploiement.
- **Déploiement** : Déploiement de l'application sur AWS, Heroku, ou une autre plateforme.

8. Documentation et Rapports

- **Documentation du code et de l'architecture** : Documentation exhaustive pour les développeurs.
- **Rapports de performance et sécurité** : Analyses détaillées de la performance, sécurité, et qualité du code.
- **Présentation** : Utilisation de PowerPoint ou Canva pour présenter le projet de manière visuelle.

9. Déploiement et Surveillance

- **Configuration des bases de données** : Sécurisation et configuration des bases de données avec des stratégies de sauvegarde.
- **Surveillance et gestion des performances** : Mise en place d'outils de surveillance pour garantir une disponibilité optimale.

10. Modalités Pédagogiques

- **Travail individuel** : Le projet doit être réalisé en autonomie.
- **Deadline** : 25/09/2024.

11. Modalités d'Évaluation

- **Présentation publique** : 15 minutes pour présenter l'application devant le jury.
- **Revue du code** : Analyse du code et questions sur la culture web.
- **Mise en situation** : 15 minutes pour résoudre un problème ou améliorer une fonctionnalité en temps réel.

12. Livrables

- **Code source** : Lien vers le référentiel GitHub contenant tout le code source.
- **Documentation du projet** : En format PDF, PPT ou sur Canva.
- **Images Docker** : Images Docker du backend et du frontend sur DockerHub.

13. Critères de Performance

- **Qualité du code** : Clarté, performance et commentaires informatifs.
 - **Performance fonctionnelle** : Conformité aux exigences et fiabilité.
 - **Convivialité de l'utilisateur** : Accessibilité et expérience utilisateur optimale.
 - **Contrôle de version Git** : Pratiques de versionnement organisées et cohérentes.
 - **Documentation** : Documentation claire et précise.
-

Ce cahier des charges fournit une base claire pour le développement de votre projet de gestionnaire de stock et de livraison avec Laravel et Vue.js. Assurez-vous de respecter toutes les exigences mentionnées et d'utiliser les outils appropriés pour une réalisation réussie du projet.