

**Thème :** **Système de gestion du fret**

**Présenté par :**

**Nom Prénom : REGRAGUI OTHMANE**

**Matricule: 432**

Jurys :

**Mr.Faraj Youness**

**Mr.ELKZIT Abderrahim**

Encadrant :

**Mr. Abdelaali Jaghboub**

HammadAbir

Année 2023/2024

**Remerciement**

Je tiens à remercier dans un premier temps, toute l'équipe pédagogique de l'École supérieure Sup’ Management ainsi que les intervenants professionnels responsables de la formation Ingénierie informatique. Leur engagement, leur dévouement et leur professionnalisme ont été déterminants dans le succès de cette formation et dans la réalisation de ce projet.

Avant d'entamer ce rapport, je profite de l'occasion pour remercier tout d'abord mon professeur, Monsieur Jaghboub Abdelaali. Son soutien inlassable et ses encouragements constants tout au long de la durée du projet ont été d'une aide précieuse. Sa générosité en matière de formation et d'encadrement a grandement contribué à l'acquisition des compétences nécessaires à la réalisation de ce travail. Je le remercie également pour l'aide et les conseils pertinents qu'il m'a prodigués concernant les missions évoquées dans ce rapport. Ses différentes interventions lors des suivis réguliers ont été sources de motivation et d'inspiration. Sa confiance et son soutien ont été des piliers essentiels dans la concrétisation de ce projet.

Mes remerciements vont aussi à tous les intervenants extérieurs, professionnels du domaine, qui ont partagé avec nous leur savoir-faire et leurs connaissances pratiques. Leurs interventions ont été enrichissantes et ont apporté une dimension concrète à notre formation, nous permettant de mieux appréhender les réalités du terrain et les exigences du métier.

Enfin, je ne saurais oublier mes camarades de promotion, avec qui j'ai partagé des moments de travail intenses et enrichissants. Leur soutien et leur camaraderie ont rendu cette aventure académique plus agréable et moins solitaire. Ensemble, nous avons pu surmonter les défis et célébrer nos réussites, renforçant ainsi notre esprit de groupe et notre solidarité.

En conclusion, ce projet et ce rapport sont le fruit d'un travail collectif et d'un soutien constant de la part de l'ensemble des personnes mentionnées ci-dessus. À tous, je dis un grand merci.

# Sommaire

[Sommaire 3](#_Toc171463100)

[I. Introduction 4](#_Toc171463101)

[● Un aperçu du projet et de ses objectifs 4](#_Toc171463102)

[● L’importance de gérer les entrepôts et les expéditions entre succursales 4](#_Toc171463103)

[II. Analyse des besoins 5](#_Toc171463104)

[● Exigences fonctionnelles 5](#_Toc171463105)

[● Exigences non fonctionnelles 5](#_Toc171463106)

[III.Conception 7](#_Toc171463107)

[employees 7](#_Toc171463108)

[employee\_branche 8](#_Toc171463109)

[products 8](#_Toc171463110)

[shipments 8](#_Toc171463111)

[stock 8](#_Toc171463112)

[transactions 8](#_Toc171463113)

[Relations et Clés Étrangères 9](#_Toc171463114)

[IV. Technologies utilisées 10](#_Toc171463115)

[● PHP 10](#_Toc171463116)

[● JavaScript et jQuery 11](#_Toc171463117)

[● Bootstrap 12](#_Toc171463118)

[● Docker 12](#_Toc171463119)

[● Nginx 13](#_Toc171463120)

[● Nginx Proxy Manager 13](#_Toc171463121)

[V. Mise en œuvre 14](#_Toc171463122)

[● Explication des fonctions et fonctionnalités les plus importantes mises en œuvre 14](#_Toc171463123)

[● Les défis de mise en œuvre et comment les surmonter 15](#_Toc171463124)

[VI. Installation de l'environnement 20](#_Toc171463125)

[● Installation de debian 20](#_Toc171463126)

[● Installation de Docker Compose : 20](#_Toc171463127)

[VII. Sécurité 22](#_Toc171463128)

[● Au niveau du serveur : 22](#_Toc171463129)

[● Au niveau des sites: 22](#_Toc171463130)

[VIII. Maintenance et développement futur 24](#_Toc171463131)

[● Plan de maintenance 24](#_Toc171463132)

[● Suggestions d'améliorations et de fonctionnalités futures 25](#_Toc171463133)

[IX. Conclusion 26](#_Toc171463134)

[● Résumé des réalisations 26](#_Toc171463135)

[● Leçons apprises 26](#_Toc171463136)

# I. Introduction

## Un aperçu du projet et de ses objectifs

Ce projet vise à développer un système de gestion d'entrepôt et d'expédition en ligne pour un groupe de succursales commerciales. Ce système est conçu pour répondre aux besoins des entreprises qui exploitent plusieurs succursales et qui doivent coordonner efficacement les opérations d'inventaire et d'expédition entre elles.

Les principaux objectifs du projet comprennent :

- Fournir une plateforme unifiée pour la gestion des stocks dans toutes les succursales.

- Faciliter le processus de suivi des expéditions entre les différentes succursales.

- Améliorer la précision et la rapidité des opérations d'inventaire et de mise à jour.

- Fournir des rapports et des statistiques en temps réel pour soutenir la prise de décision administrative.

## L’importance de gérer les entrepôts et les expéditions entre succursales

Compte tenu de l’expansion croissante des entreprises et de la multiplicité de leurs succursales, la nécessité d’un système efficace de gestion des entrepôts et des expéditions est devenue vitale pour les raisons suivantes :

* Amélioration de l'efficacité opérationnelle : le système permet de réduire le temps et les efforts consacrés à la gestion des stocks et au suivi des expéditions.
* Réduction des erreurs : en automatisant les processus, les erreurs humaines dans la saisie des données et la gestion des stocks sont réduites.
* Améliorer le service client : fournir des informations précises et en temps réel sur la disponibilité des produits et l'état des expéditions.
* Améliorer la gestion de la chaîne d'approvisionnement : permettre à l'entreprise d'optimiser les opérations d'achat et d'entreposage sur la base de données précises.
* Soutenir la croissance et l'expansion : fournir un système extensible qui peut s'adapter à la croissance de l'entreprise et à l'ajout de nouvelles succursales.

Ce projet vise à répondre à ces besoins en développant une solution logicielle complète et intégrée qui tire parti des dernières technologies en matière de développement Web et de gestion de données.

Souhaitez-vous ajouter d'autres éléments à cette section ou avez-vous des questions sur la manière d'appliquer ces points à votre projet spécifique ?

# II. Analyse des besoins

## Exigences fonctionnelles

Les exigences fonctionnelles décrivent les fonctions et les services que le système doit fournir. Dans le contexte d’un système de gestion d’entrepôt et d’expédition, ceux-ci peuvent inclure :

1. Gestion des stocks :

* Ajouter, mettre à jour et supprimer des produits en stock
* Afficher les niveaux de stock actuels pour chaque succursale

2. Gestion des expéditions :

* Créer des commandes d'expédition entre succursales
* Suivre l'état des expéditions en temps réel
* Enregistrer la réception des expéditions et mettre à jour automatiquement l'inventaire

3. Gestion des utilisateurs :

* Connectez-vous et gérez les autorisations
* Créer et gérer des comptes utilisateurs pour différents rôles (gestionnaire, employé du magasin, superviseur des expéditions)

5. Rechercher et filtrer :

* Rechercher des produits et des expéditions en utilisant plusieurs critères
* Filtrer et trier les résultats par date, succursale, type de produit, etc.

## Exigences non fonctionnelles

Les exigences non fonctionnelles décrivent les caractéristiques et les contraintes que le système doit respecter. Dans le cas de votre système, ils peuvent inclure :

1. Performances :

* Temps de réponse rapide (moins de 2 secondes) pour la plupart des opérations
* Capacité à gérer au moins 1000 utilisateurs simultanés

2. Sécurité :

* Crypter toutes les données transmises en utilisant HTTPS
* Enregistrez toutes les activités sensibles dans les journaux de sécurité

3. Évolutivité :

* Conception extensible pour s'adapter à une augmentation du nombre de succursales et d'utilisateurs
* Possibilité d'ajouter de nouvelles unités au système dans le futur

4. Facilité d’utilisation :

* Interface utilisateur intuitive et facile à naviguer
* Compatibilité avec différentes tailles d'écran (responsive design)

5. Fiabilité :

* La disponibilité du système est d'au moins 99,9 %
* Sauvegarde régulière des données avec possibilité de récupération rapide

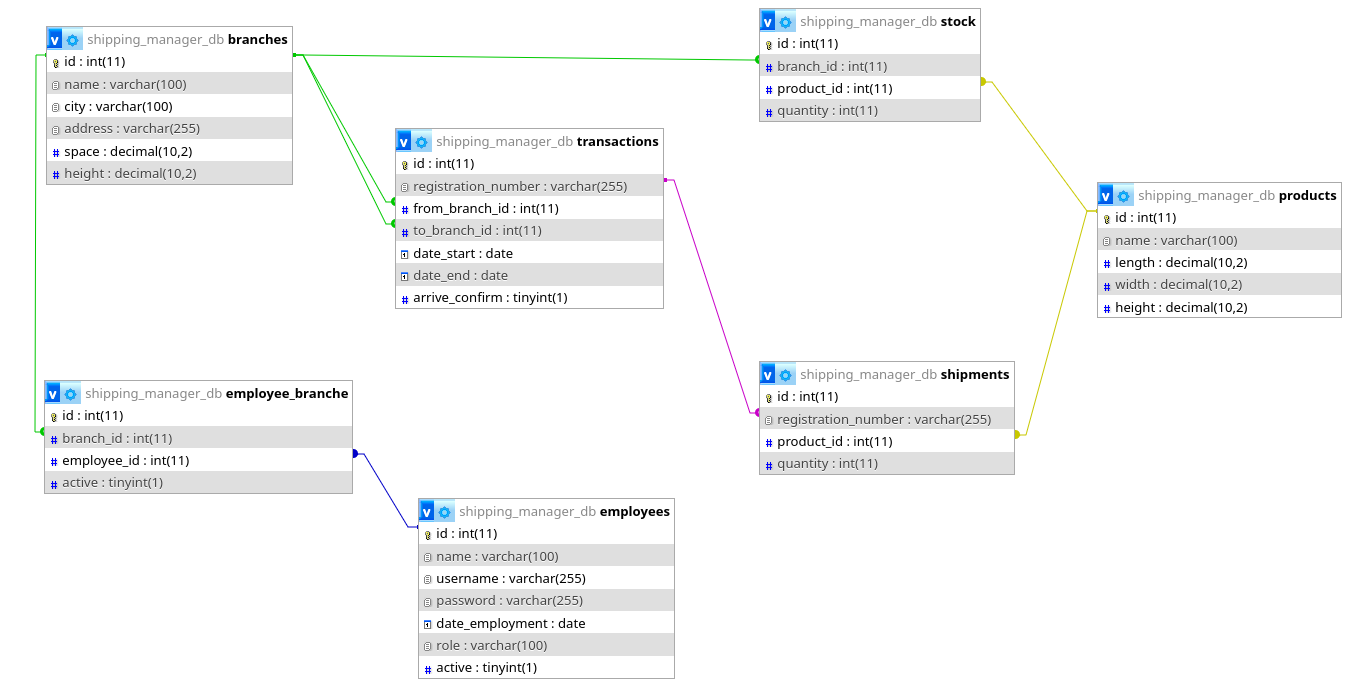
6. Compatibilité :

* Compatibilité avec les principaux navigateurs Web (Chrome, Firefox, Safari, Edge)
* Possibilité d'intégration avec les systèmes ERP existants

7. Entretien :

* Documentation complète du code et des processus
* Facilité de mise à jour et de maintenance du système

# III.Conception

****

La base de données shipping manager db est conçue pour gérer les informations liées à la gestion des expéditions dans une entreprise. Elle est composée des tables suivantes :

**branches**

Cette table stocke des informations sur les différentes branches de l'entreprise.

* id : Identifiant unique de la branche (clé primaire).
* name : Nom de la branche.
* city : Ville où se situe la branche.
* address : Adresse complète de la branche.
* space : Espace disponible dans la branche, exprimé en mètres carrés.
* height : Hauteur de la branche, exprimée en mètres.

### employees

Cette table contient des informations sur les employés.

* id : Identifiant unique de l'employé (clé primaire).
* name : Nom de l'employé.
* username : Nom d'utilisateur utilisé pour la connexion.
* password : Mot de passe de l'employé, généralement stocké sous forme de hachage.
* date\_employment : Date d'embauche de l'employé.
* role : Rôle de l'employé au sein de l'entreprise (par exemple, admin, manager, employé).
* active : État d'activité de l'employé, 1 pour actif et 0 pour inactif.

### employee\_branche

Cette table établit une relation entre les employés et les branches, indiquant dans quelle branche travaille chaque employé et s'il est actif.

* id : Identifiant unique de la relation (clé primaire).
* branch\_id : Identifiant de la branche (clé étrangère vers branches(id)).
* employee\_id : Identifiant de l'employé (clé étrangère vers employees(id)).
* active : État d'activité de la relation, 1 pour actif et 0 pour inactif.

### products

Cette table contient des informations sur les produits.

* id : Identifiant unique du produit (clé primaire).
* name : Nom du produit.
* length : Longueur du produit, exprimée en centimètres.
* width : Largeur du produit, exprimée en centimètres.
* height : Hauteur du produit, exprimée en centimètres.

### shipments

Cette table stocke les informations sur les expéditions.

* id : Identifiant unique de l'expédition (clé primaire).
* registration\_number : Numéro d'enregistrement unique de l'expédition.
* product\_id : Identifiant du produit expédié (clé étrangère vers products(id)).
* quantity : Quantité du produit expédié.

### stock

Cette table gère les stocks de produits dans chaque branche.

* id : Identifiant unique du stock (clé primaire).
* branch\_id : Identifiant de la branche (clé étrangère vers branches(id)).
* product\_id : Identifiant du produit (clé étrangère vers products(id)).
* quantity : Quantité du produit en stock dans la branche.

### transactions

Cette table enregistre les transactions entre les branches.

* id : Identifiant unique de la transaction (clé primaire).
* registration\_number : Numéro d'enregistrement de la transaction, souvent utilisé pour lier avec les expéditions.
* from\_branch\_id : Identifiant de la branche d'origine (clé étrangère vers branches(id)).
* to\_branch\_id : Identifiant de la branche de destination (clé étrangère vers branches(id)).
* date\_start : Date de début de la transaction.
* date\_end : Date de fin de la transaction.
* arrive\_confirm : Indicateur de confirmation d'arrivée, 1 pour confirmer et 0 pour non confirmé.

### Relations et Clés Étrangères

* **employee\_branche** :
  + branch\_id référence branches(id).
  + employee\_id référence employees(id).
* **shipments** :
  + product\_id référence products(id).
  + registration\_number référence transactions(registration\_number), avec suppression en cascade pour maintenir l'intégrité des données.
* **stock** :
  + branch\_id référence branches(id).
  + product\_id référence products(id).
* **transactions** :
  + from\_branch\_id référence branches(id).
  + to\_branch\_id référence branches(id).

# IV. Technologies utilisées

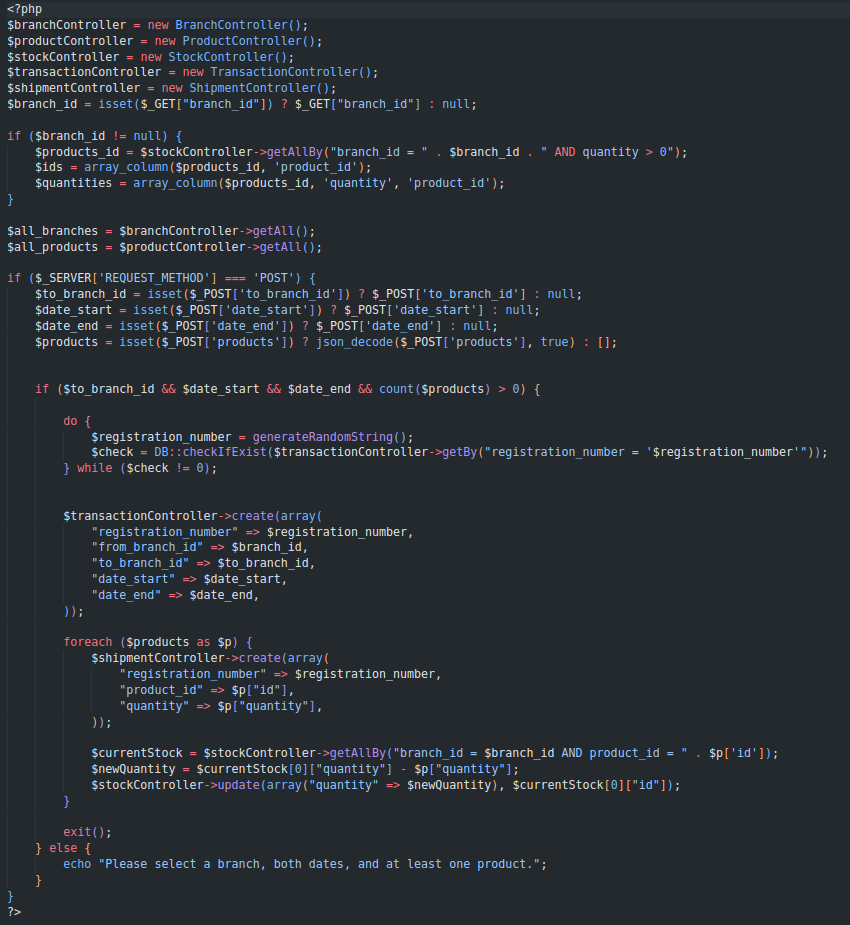
Dans ce projet, diverses technologies modernes ont été utilisées pour garantir l'efficacité, l'évolutivité et la maintenabilité du système. Vous trouverez ci-dessous une explication de chaque technologie et de son rôle dans le projet :

## PHP

**Rôle : Développer le backend du site internet.**

Utilisation dans le projet :

* Traiter les demandes des utilisateurs et mettre en œuvre la logique métier
* Interagissez avec la base de données pour stocker et récupérer les données d'inventaire et d'expéditions
* Créer une interface de programmation d'application (API) pour communiquer avec le front-end

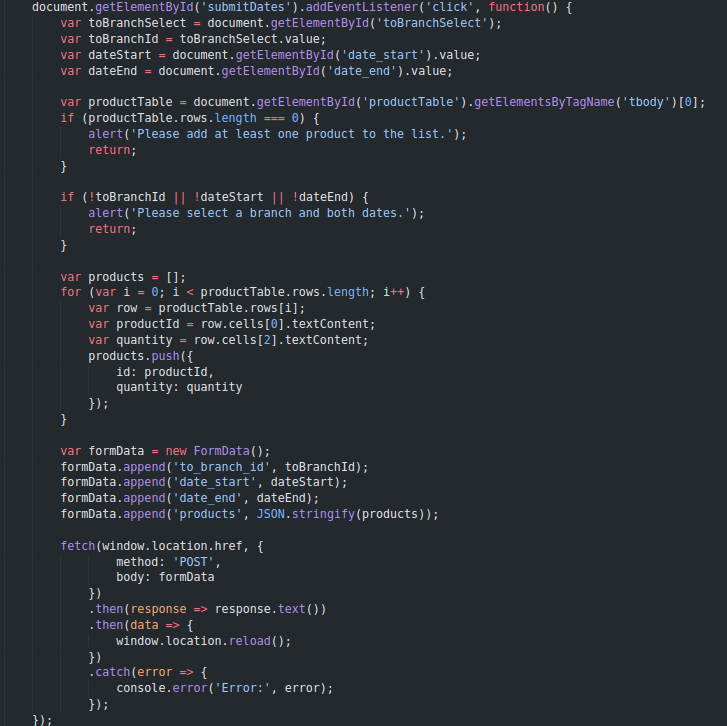


## JavaScript et jQuery

**- Rôle : Améliorer l'interactivité de l'interface utilisateur (Frontend)**

Utilisation dans le projet :

* Ajoutez des mises à jour dynamiques aux pages Web sans les recharger
* Validation des données côté client
* Créez des effets visuels et des transitions fluides

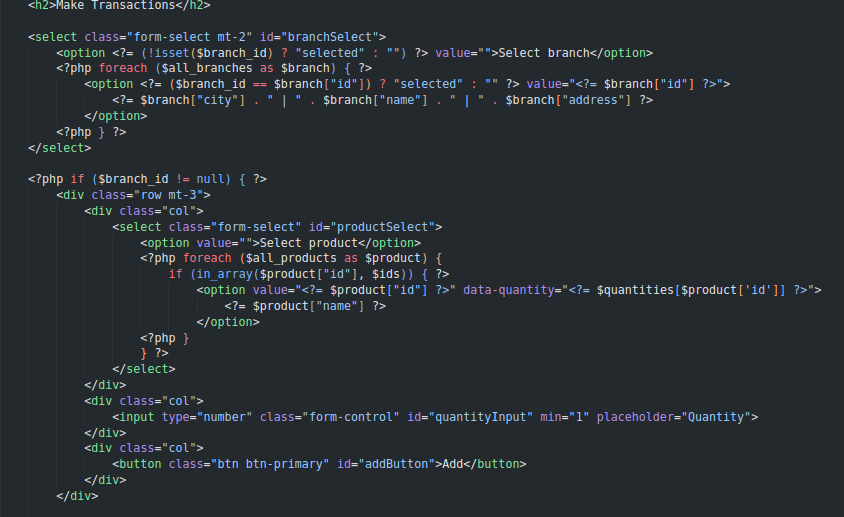


## Bootstrap

**Rôle : Concevoir l'interface utilisateur et sa réactivité**

Utilisation dans le projet :

* Créez une mise en page réactive qui fonctionne sur différentes tailles d'écran
* Utilisez des composants prêts à l'emploi tels que des boutons, des tableaux et des formulaires
* Appliquer un style uniforme sur toutes les pages de l'application



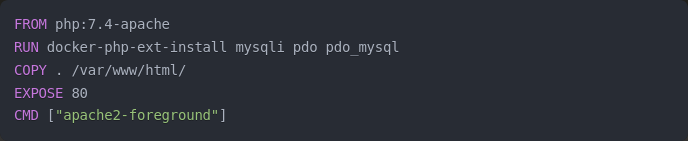
## Docker

**Rôle : Gérer et distribuer l'application**

Utilisation dans le projet :

* Établir un environnement de développement et de production cohérent
* Faciliter le déploiement de l'application sur le serveur
* Gérer les dépendances et les paramètres de manière unifiée

Exemple : Dockerfile pour configurer l'environnement PHP :



## Nginx

**Rôle : Serveur Web et proxy inverse**

Utilisation dans le projet :

* Diriger les demandes vers les services appropriés
* Améliorer les performances grâce à la mise en cache
* Gérer les connexions SSL/TLS

Exemple : configuration de base de Nginx :



## Nginx Proxy Manager

**Rôle : Gérer facilement le proxy inverse**

Utilisation dans le projet :

* Gérer automatiquement les certificats SSL
* Routage des requêtes vers différents services en fonction du nom de domaine
* Fournir une interface graphique pour gérer les paramètres Nginx

# V. Mise en œuvre

## Explication des fonctions et fonctionnalités les plus importantes mises en œuvre

Dans le cadre de ce projet de gestion des expéditions, plusieurs fonctions et fonctionnalités cruciales ont été mises en œuvre pour assurer une gestion efficace et fluide des opérations. Voici une explication des principales fonctionnalités :

**Gestion des branches :**

Création, modification et suppression des branches : Les administrateurs peuvent ajouter de nouvelles branches, mettre à jour les informations existantes ou supprimer des branches obsolètes. Cela inclut la gestion des détails tels que le nom, la ville, l'adresse, l'espace disponible et la hauteur.

Visualisation des branches : Les utilisateurs peuvent visualiser une liste de toutes les branches avec leurs détails respectifs, facilitant ainsi la gestion et l'allocation des ressources.

**Gestion des employés :**

Authentification et autorisation : Implémentation d'un système d'authentification robuste permettant aux employés de se connecter avec un nom d'utilisateur et un mot de passe sécurisés. Le système gère également les rôles des utilisateurs (administrateur, manager, employé) et les permissions associées.

Affectation des employés aux branches : Les employés peuvent être affectés à des branches spécifiques. Cette fonctionnalité permet de suivre les employés et de gérer leur activité au sein de différentes branches.

**Gestion des produits :**

CRUD des produits : Les administrateurs peuvent créer, lire, mettre à jour et supprimer des informations sur les produits, y compris les dimensions (longueur, largeur, hauteur).

Visualisation des produits : Les utilisateurs peuvent consulter les détails des produits stockés, facilitant ainsi la gestion des stocks et des expéditions.

**Gestion des expéditions :**

Suivi des expéditions : Chaque expédition est enregistrée avec un numéro d'enregistrement unique, le produit expédié, et la quantité. Cette fonctionnalité permet de suivre les expéditions de manière précise et détaillée.

Lien avec les transactions : Les expéditions sont associées aux transactions pour maintenir une traçabilité complète du déplacement des produits entre les branches.

**Gestion des stocks :**

Suivi des stocks par branche : Les stocks de chaque produit dans chaque branche sont suivis de manière détaillée. Les administrateurs peuvent visualiser et mettre à jour les quantités disponibles.

Alerte de stock faible : Implémentation d'un système d'alerte pour informer les gestionnaires lorsque le stock d'un produit dans une branche tombe en dessous d'un seuil critique.

**Gestion des transactions :**

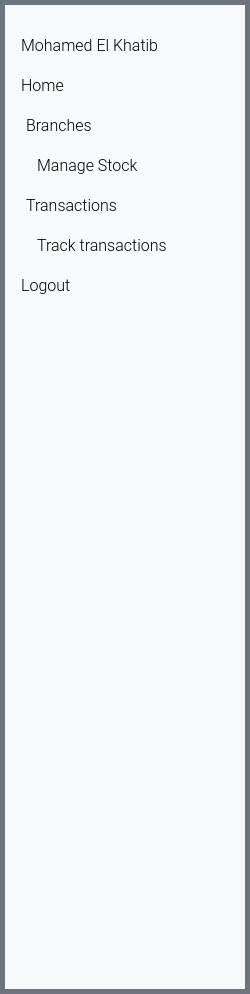
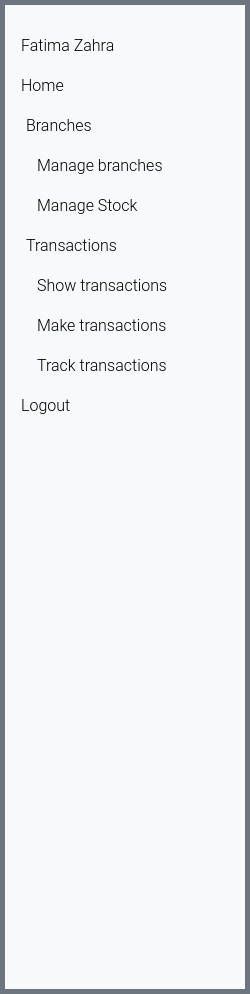
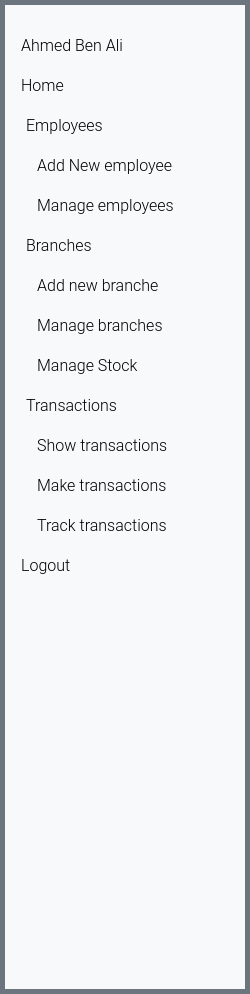
Enregistrement des transactions : Chaque transaction entre branches est enregistrée, incluant les branches d'origine et de destination, les dates de début et de fin, et la confirmation d'arrivée.

Suivi des envois : Les envois sont suivis grâce à leurs numéros d'enregistrement, permettant ainsi une gestion transparente et efficace des déplacements de produits.

## Les défis de mise en œuvre et comment les surmonter

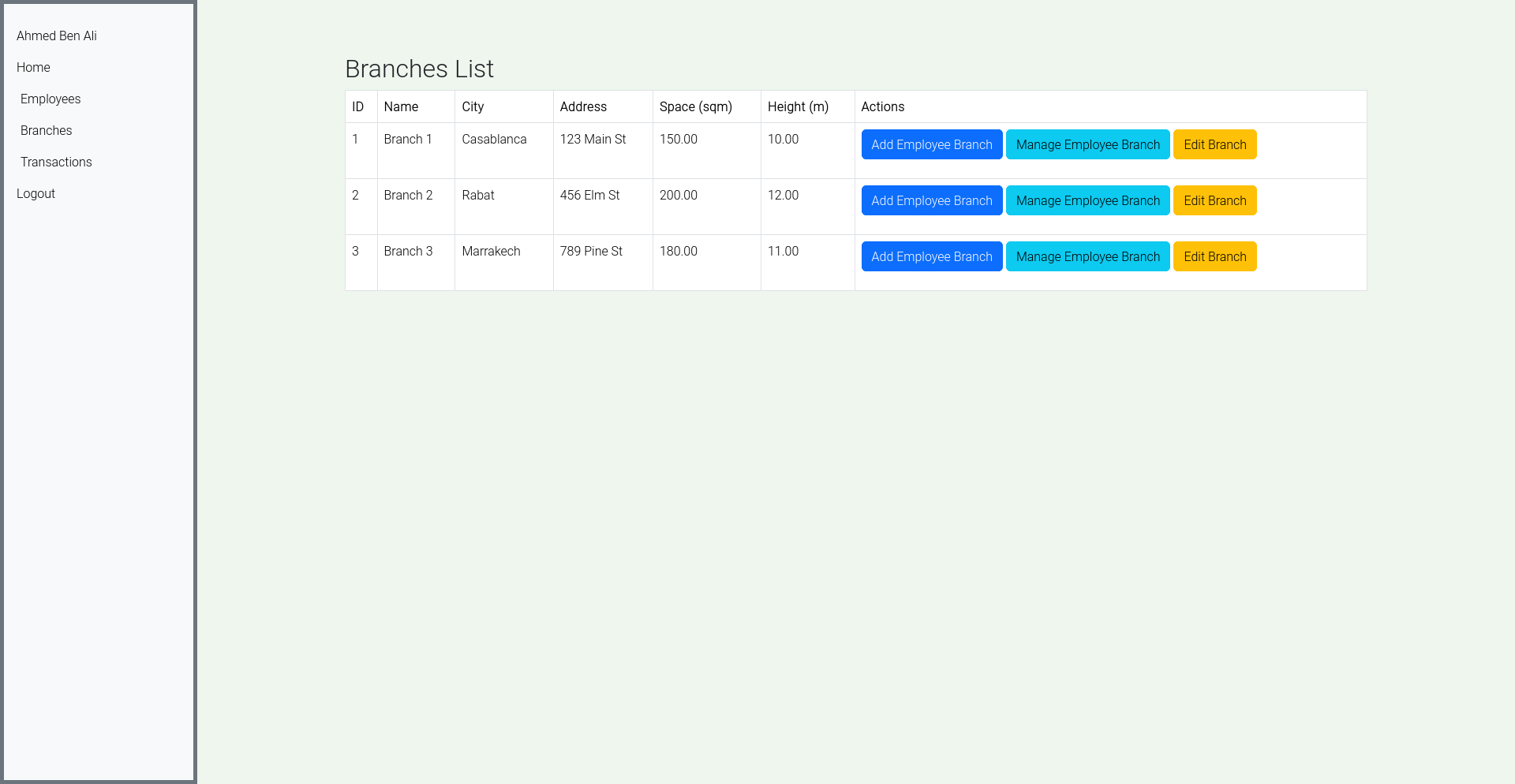
* **Sécurité des données :**
* Défi : Assurer la sécurité des données sensibles telles que les mots de passe des employés.
* Solution : Utilisation de techniques de hachage robustes pour stocker les mots de passe et implémentation de mécanismes d'authentification et d'autorisation sécurisés pour restreindre l'accès aux informations sensibles.

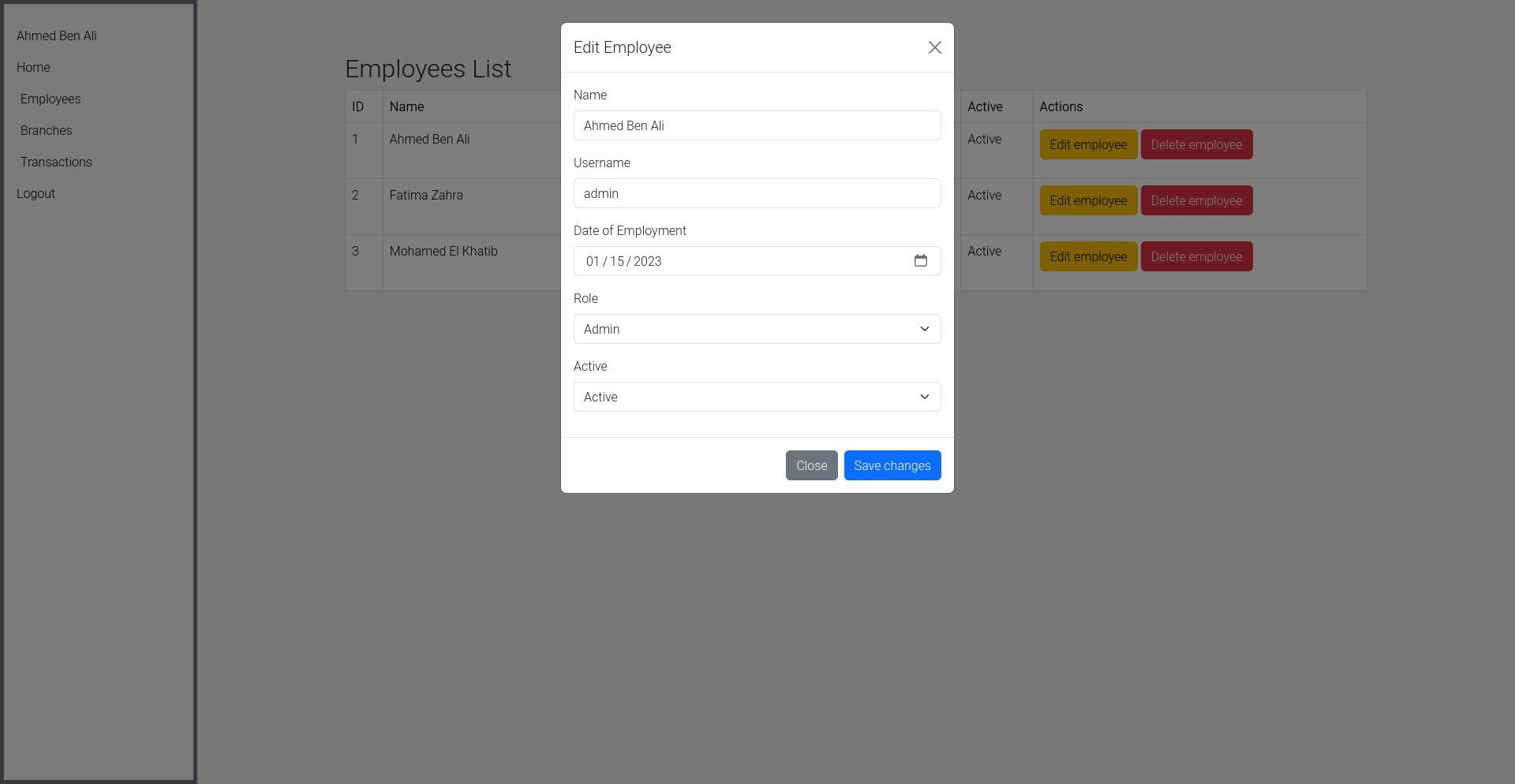
|  |  |  |
| --- | --- | --- |
| **admin** | **manager** | **employees** |

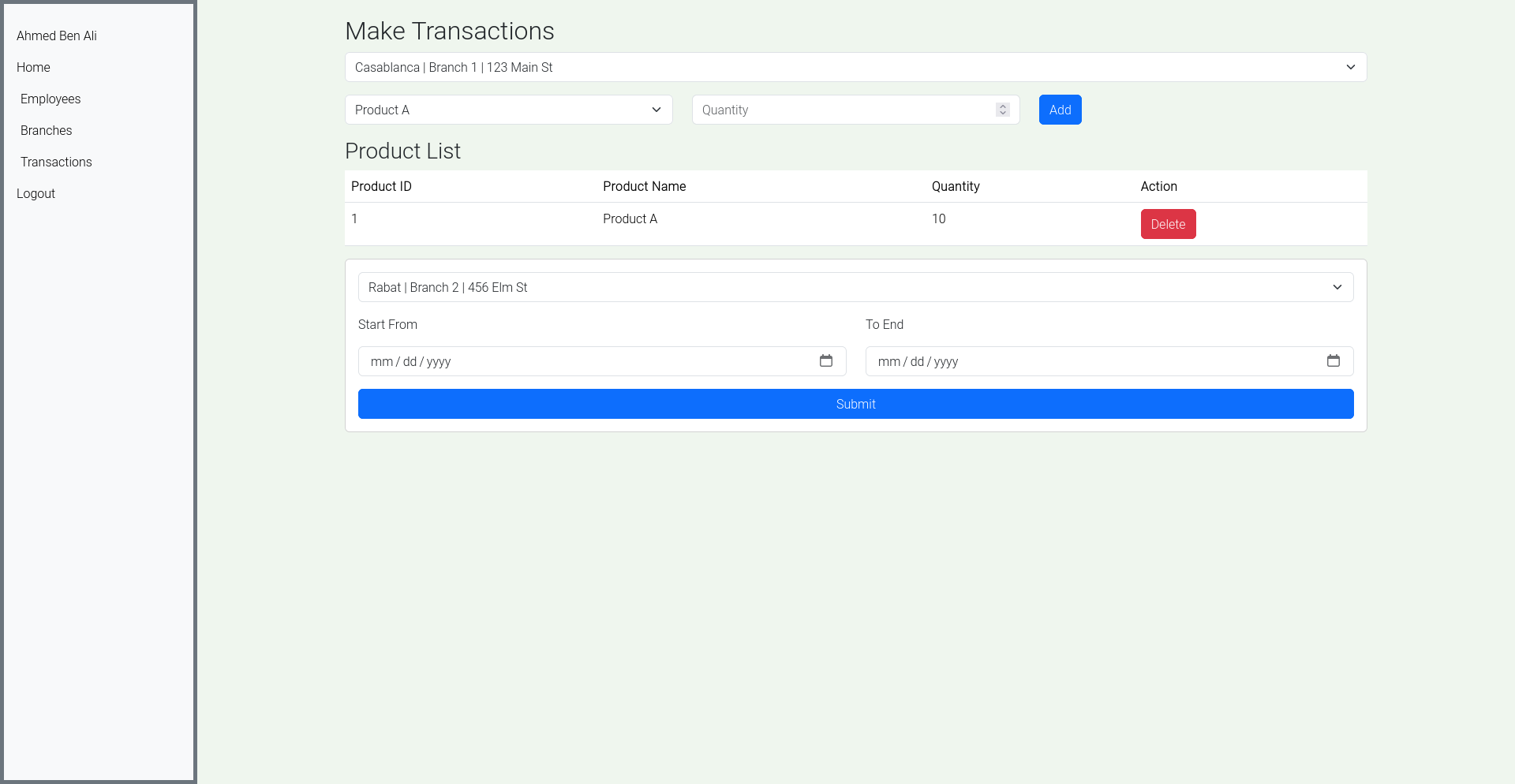
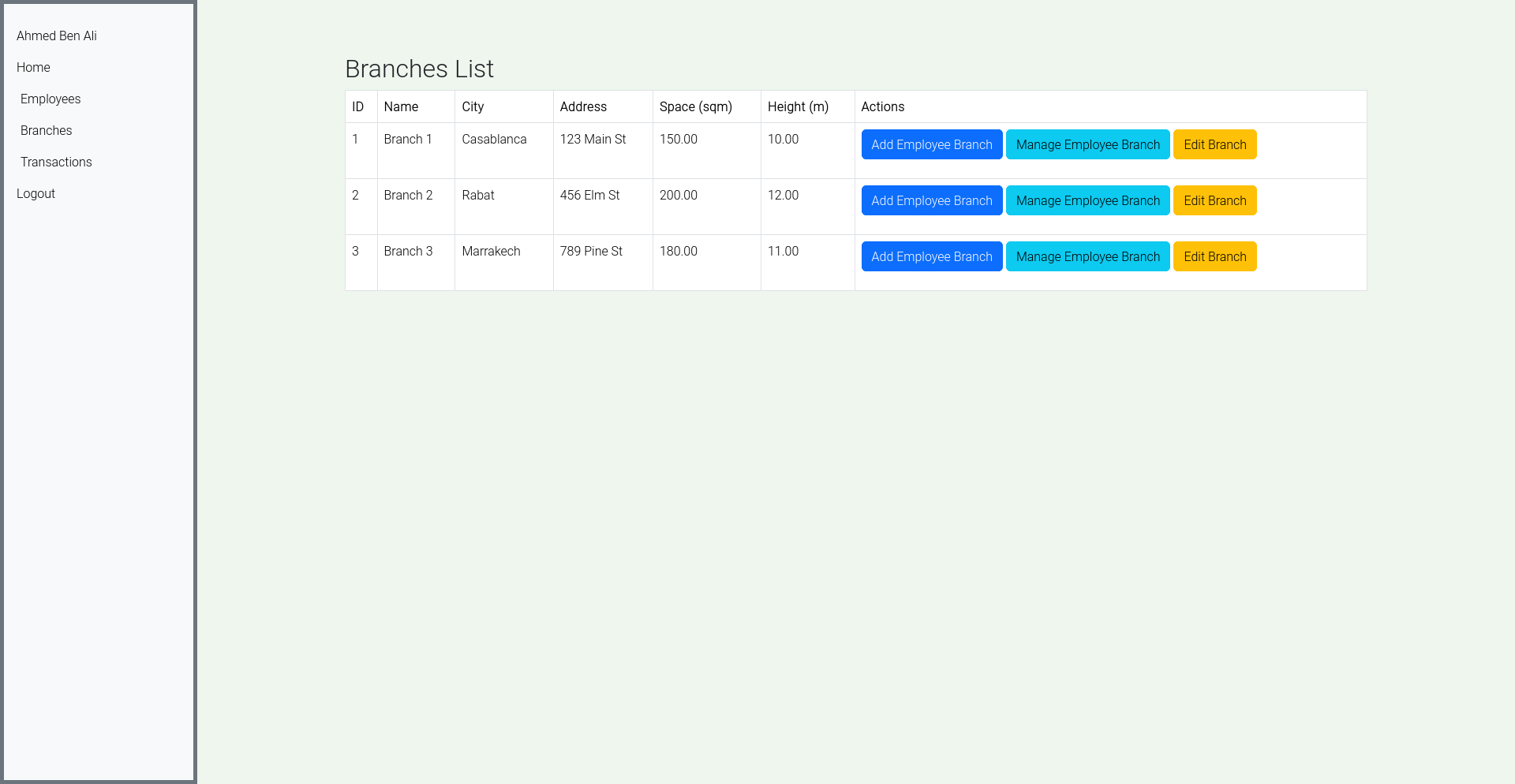


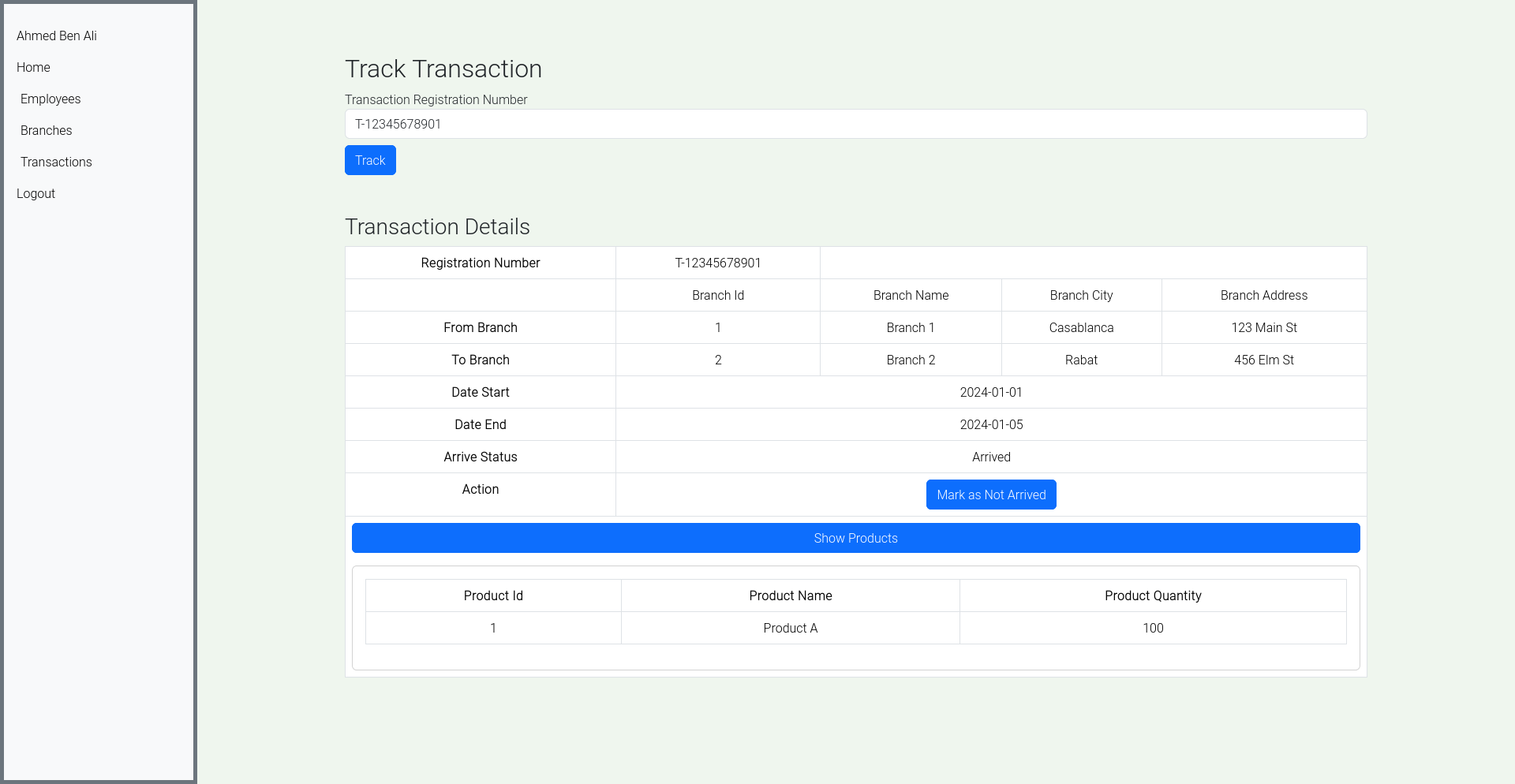
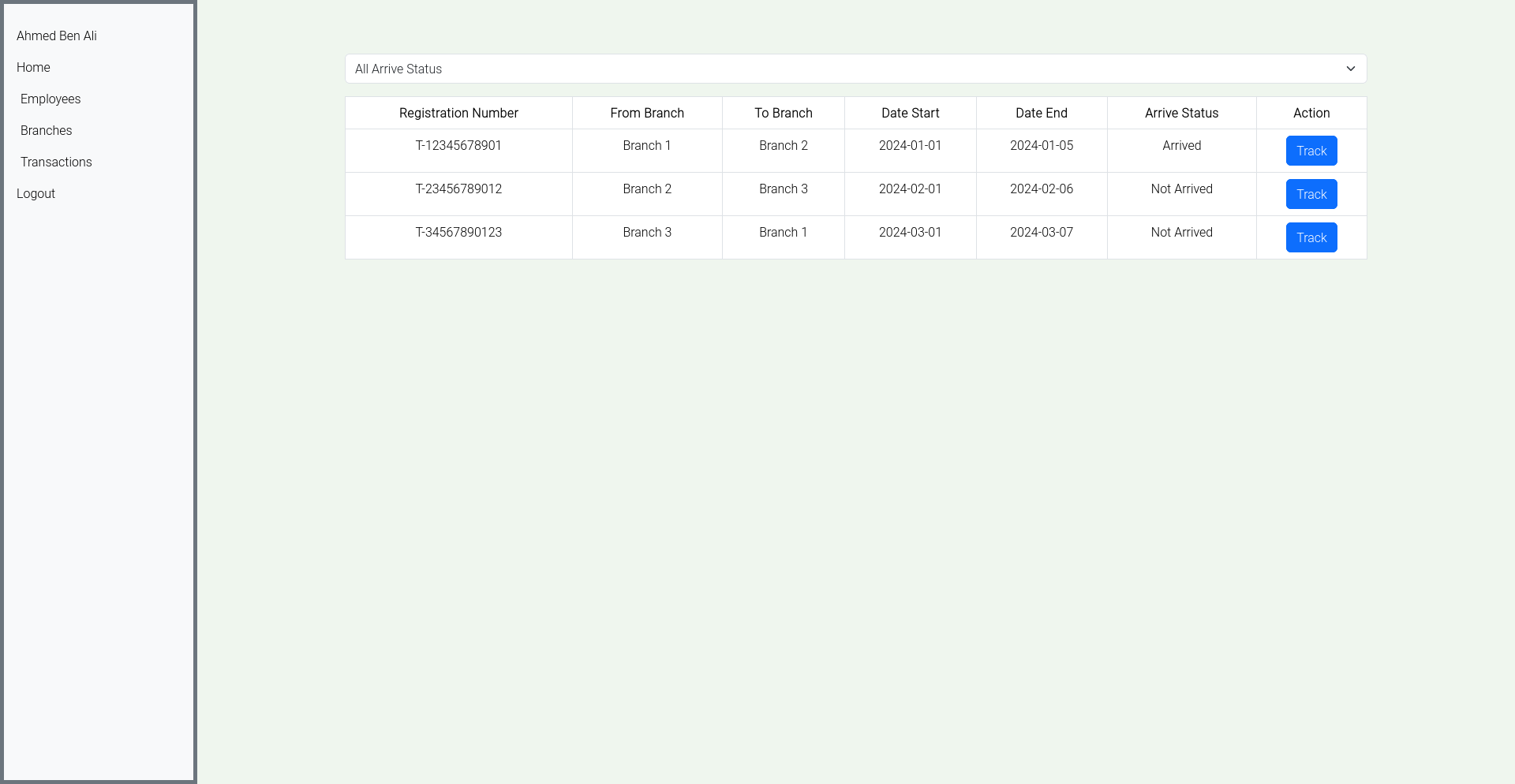
* **Intégrité des données :**
* Défi : Maintenir l'intégrité des données entre les différentes tables et relations.
* Solution : Mise en place de clés étrangères et de contraintes d'intégrité référentielle pour garantir que les relations entre les tables restent cohérentes. Utilisation de transactions pour assurer que les opérations complexes soient exécutées de manière atomique.
* **Gestion des performances :**
* Défi : Optimiser les performances de la base de données pour gérer un grand volume de données et des requêtes fréquentes.
* Solution : Utilisation d'index pour accélérer les requêtes, optimisation des requêtes SQL et mise en cache des résultats fréquemment utilisés. Surveillance des performances et ajustements continus pour éviter les goulets d'étranglement.
* **Scalabilité :**

* Défi : Concevoir le système pour qu'il puisse évoluer avec l'augmentation du nombre d'utilisateurs et de données.
* Solution : Utilisation d'une architecture modulaire permettant d'ajouter facilement de nouvelles fonctionnalités. Préparation de la base de données et des applications pour la montée en charge, avec la possibilité de distribuer les charges sur plusieurs serveurs.
* **Expérience utilisateur :**
* Défi : Fournir une interface utilisateur intuitive et facile à utiliser pour les employés de différents niveaux techniques.
* Solution : Conception d'une interface utilisateur claire et conviviale avec des instructions détaillées et des feedbacks immédiats pour les actions effectuées. Réalisation de tests utilisateur pour identifier et résoudre les problèmes d'utilisabilité.









En surmontant ces défis, nous avons réussi à créer un système de gestion des expéditions robuste, sécurisé et efficace, capable de répondre aux besoins de l'entreprise tout en offrant une expérience utilisateur fluide.

# VI. Installation de l'environnement

## Installation de debian

**Téléchargement de Debian :**

* **Téléchargez l'image ISO de Debian depuis le site officiel de Debian.**

**Création d'un support d'installation :**

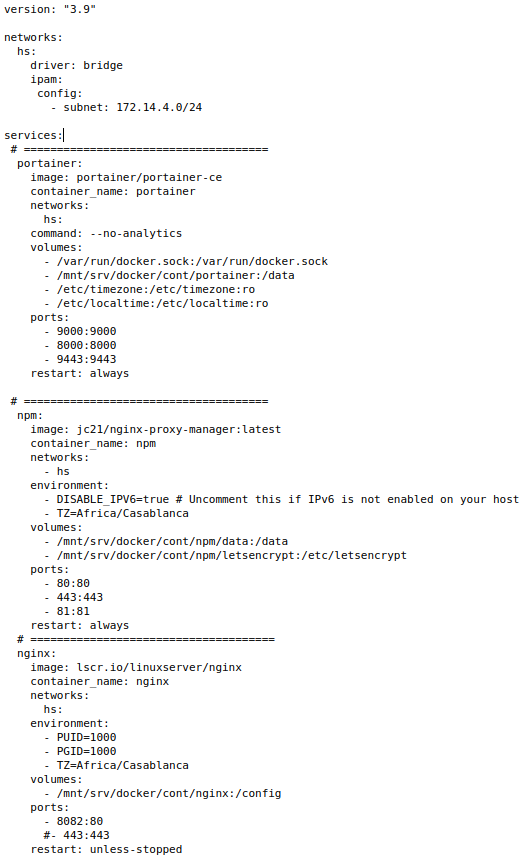
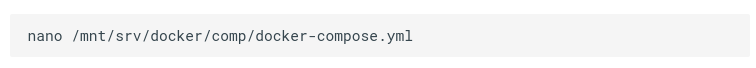
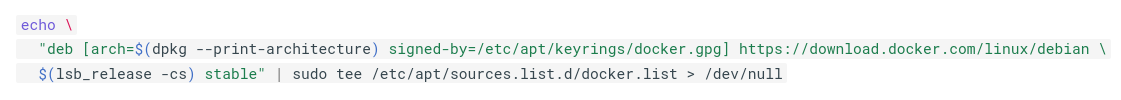
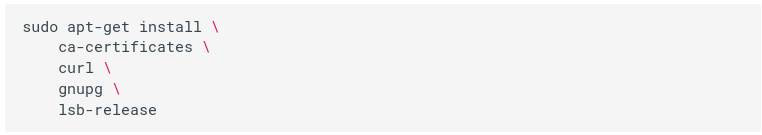
* **Utilisez un outil comme** [**Rufus**](https://rufus.ie/) **(pour Windows) ou dd (pour Linux) pour créer une clé USB bootable avec l'image ISO de Debian.**

**Installation de Debian :**

* **Insérez la clé USB dans le serveur et démarrez-le. Accédez au BIOS/UEFI et configurez-le pour démarrer à partir de la clé USB.**
* **Suivez les étapes de l'installateur Debian :**
* **Sélectionnez la langue, le fuseau horaire et la disposition du clavier.**
* **Configurez le réseau en attribuant une adresse IP statique ou dynamique.**
* **Créez un utilisateur et définissez un mot de passe root.**
* **Partitionnez le disque. Vous pouvez choisir le partitionnement assisté pour simplifier le processus.**
* **Installez le système de base.**

## Installation de Docker Compose :

**Mise à jour du système :**



# VII. Sécurité

## Au niveau du serveur :

1. **Configuration SSH sécurisée :**

* **Port non standard** : Le port par défaut pour SSH (22) est modifié pour réduire les tentatives de connexion non autorisées.
* **Désactivation de l'accès root** : L'accès direct au compte root via SSH est désactivé, limitant ainsi les possibilités d'attaques.
* **Authentification par clé SSH** : L'utilisation de clés SSH pour l'authentification est privilégiée par rapport aux mots de passe, offrant une sécurité renforcée.

1. **Pare-feu :**

* UFW (Uncomplicated Firewall) est configuré pour permettre uniquement le trafic nécessaire, notamment les ports SSH, HTTP et HTTPS.

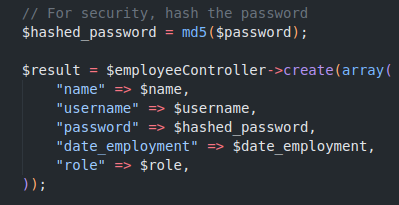
1. **Mises à jour régulières :**

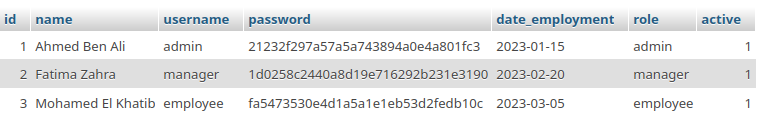
* Les paquets du système et des applications sont régulièrement mis à jour pour appliquer les derniers correctifs de sécurité.

****

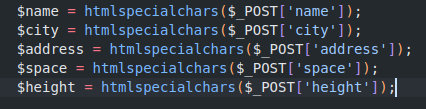
## Au niveau des sites:

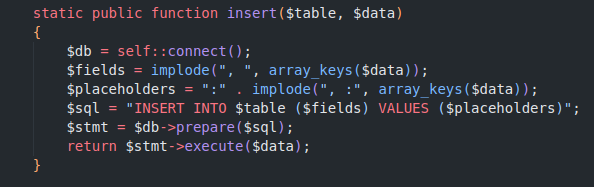
* **Cryptage de mot de passe:**

****

****

* **Eviter SQL injection:**

****

****

# VIII. Maintenance et développement futur

## Plan de maintenance

* **Mises à jour régulières du système :**
* Système d'exploitation : Effectuer des mises à jour régulières du système d'exploitation pour appliquer les derniers correctifs de sécurité et les mises à jour des paquets.
* Applications et dépendances : Assurer la mise à jour des dépendances de l'application (bibliothèques, frameworks) pour maintenir la sécurité et les performances optimales.
* **Surveillance continue :**
* Surveillance des performances : Utiliser des outils de surveillance (comme Prometheus et Grafana) pour suivre les performances du système et des applications, et identifier les goulets d'étranglement ou les problèmes de ressources.
* Surveillance de la sécurité : Configurer des outils de surveillance de la sécurité (comme Fail2Ban et Wazuh) pour détecter et réagir aux tentatives d'intrusion et autres activités suspectes.
* **Sauvegardes régulières :**
* Base de données : Mettre en place des sauvegardes automatiques quotidiennes de la base de données, avec vérification régulière de l'intégrité des sauvegardes.
* Fichiers critiques Sauvegarder régulièrement les fichiers critiques et la configuration de l'application, et stocker les sauvegardes dans des emplacements sécurisés et redondants.
* **Gestion des utilisateurs :**
* Revue des accès : Réaliser des revues périodiques des accès des utilisateurs pour s'assurer que seuls les utilisateurs autorisés ont accès aux ressources nécessaires.
* Rotation des mots de passe : Imposer la rotation régulière des mots de passe pour tous les utilisateurs, en particulier ceux ayant des privilèges élevés.
* **Documentation :**
* Mise à jour de la documentation : Maintenir à jour la documentation technique et utilisateur de l'application, incluant les procédures de maintenance, les guides de dépannage et les changements récents.

## Suggestions d'améliorations et de fonctionnalités futures

* **Améliorations des fonctionnalités existantes :**
* Interface utilisateur : Améliorer l'interface utilisateur pour la rendre plus intuitive et réactive, en intégrant des Framework modernes comme React ou Vue.js.
* Optimisation des performances : Optimiser les requêtes de la base de données et le code backend pour réduire les temps de réponse et améliorer les performances globales de l'application.
* **Nouvelles fonctionnalités :**
* Rapports et analyses avancés : Ajouter des fonctionnalités de génération de rapports et d'analyse des données, permettant aux utilisateurs de visualiser les tendances et les performances de l'entreprise.
* Notifications en temps réel : Mettre en place un système de notifications en temps réel pour informer les utilisateurs des événements importants, comme l'arrivée de nouveaux stocks ou les expéditions en cours.
* **Sécurité renforcée :**
* Audit de sécurité : Effectuer des audits de sécurité réguliers par des tiers pour identifier et corriger les vulnérabilités potentielles.
* **Intégrations avec d'autres systèmes :**
* API publiques : Développer des API publiques pour permettre à d'autres applications et services de s'intégrer facilement avec l'application.
* **Évolutivité :**
* Mise à l'échelle horizontale : Concevoir l'application pour qu'elle puisse être mise à l'échelle horizontalement, en ajoutant des instances supplémentaires pour gérer une charge accrue.
* Utilisation de micro services : Adopter une architecture de micro services pour permettre une évolution et une maintenance plus faciles des différentes parties de l'application.

# IX. Conclusion

## Résumé des réalisations

A l’issue de ce projet, plusieurs réalisations importantes ont été réalisées qui méritent d’être soulignées :

Développer un système intégré de gestion des entrepôts et des expéditions :

* Une plateforme unifiée a été créée qui permet une gestion efficace des stocks dans toutes les succursales de l'entreprise.
* Le système a réussi à automatiser les processus de suivi des expéditions entre les différentes succursales.

Application des technologies modernes :

* Une combinaison de technologies modernes (PHP, JavaScript, jQuery, Bootstrap) a été utilisée pour développer une interface utilisateur interactive et facile à utiliser.
* L'utilisation de Docker a réussi à fournir un environnement de développement et de production cohérent et évolutif.

Performances et sécurité améliorées :

* Nginx est implémenté en tant que serveur Web et proxy inverse, ce qui améliore les performances du système.
* Des mesures de sécurité strictes ont été mises en œuvre pour protéger les données d'inventaire et d'expédition.

Interface utilisateur simple :

* Une interface utilisateur intuitive est conçue pour permettre aux employés de gérer plus facilement et efficacement les stocks et les expéditions.

Évolutivité :

* Le système est conçu de manière à pouvoir être facilement étendu pour s'adapter à la croissance de l'entreprise et à l'ajout de nouvelles succursales.

## Leçons apprises

Au cours du développement de ce projet, de nombreux enseignements précieux ont été tirés :

L’importance d’une planification minutieuse :

* Le projet a démontré l'importance d'élaborer un plan détaillé et une analyse complète des exigences avant de commencer la mise en œuvre.

La valeur de l’intégration entre les technologies :

* Nous avons appris à combiner différentes technologies (telles que PHP et JavaScript) pour créer une solution complète et intégrée.

Importance des tests de performances :

* Nous avons découvert l'importance d'effectuer des tests de performances approfondis, en particulier lorsqu'il s'agit de grandes quantités de données.

Défis de la gestion des bases de données :

* Nous avons été confrontés à des défis dans la conception d'une base de données efficace pour prendre en charge des opérations complexes, ce qui a souligné l'importance d'une bonne planification de la structure des données.

L’importance des retours des utilisateurs :

* L’obtention des commentaires des utilisateurs au cours du processus de développement s’est avérée cruciale pour améliorer l’expérience utilisateur finale.

La valeur d’une documentation complète :

* Nous avons réalisé l'importance de documenter chaque aspect du système pour faciliter la maintenance et le développement futur.

En conclusion, ce projet représente une étape importante dans le domaine de la gestion des entrepôts et des expéditions, car il apporte une solution technique avancée qui répond aux besoins des entreprises contemporaines. Forts des leçons apprises et des expériences acquises, nous sommes impatients de continuer à développer et à améliorer le système pour répondre aux besoins changeants du marché.

**X. Références**

* [**Docker**](https://www.docker.com/)
* [**Nginx**](https://nginx.org/en/)
* [**Nginx Proxy Manager**](https://nginxproxymanager.com/)
* [**Debian**](https://www.debian.org/)
* [**Php**](https://www.php.net/)
* [**JavaScript**](https://www.javascript.com/)
* [**Jquery**](https://jquery.com/)
* [**BootStrap 5**](https://getbootstrap.com/)
* [**Mysql**](https://www.mysql.com/)