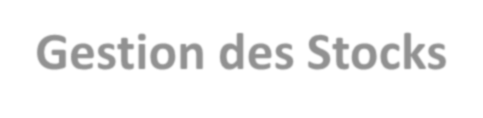
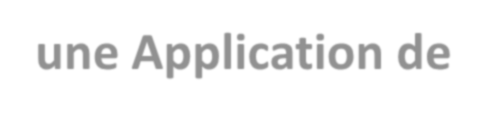
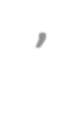
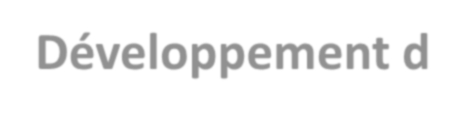


Rapport :



**Développement d’une Application de Gestion des Stocks**

#### Présenté par : Encadré par :

**Chaimaa Bennaouar**

**Et Saad el allam**

Janvier 2025

**Ahmed Amamou**



Introduction

La gestion des stocks est un élément essentiel pour assurer l'efficacité et la rentabilité de toute organisation. Avec l'évolution des technologies, il devient primordial de disposer d'un système moderne et intuitif pour gérer les stocks en ligne. Ce projet vise à répondre à ces besoins en proposant une solution innovante et conviviale.

#### Objectif du Projet :

L’objectif global est de concevoir, développer et déployer une application de gestion qui répond aux besoins suivants :

* Offrir une plateforme sécurisée pour gérer les stocks.
* Fournir des fonctionnalités CRUD (ajout, modification, suppression, et lecture des données).
* Faciliter les recherches dynamiques dans les bases de données.
* Implémenter des rôles utilisateurs (administrateurs et utilisateurs classiques).Afficher des statistiques de base sur un tableau de bord intuitif.



**Fonctionnalités Obligatoires**

1. Authentification Utilisateur

* Système de connexion sécurisé utilisant PHP et MySQL pour gérer les administrateurs et utilisateurs.
* Gestion des sessions pour garantir que seuls les utilisateurs autorisés accèdent à l'application.

1. Gestion des Stocks (CRUD)

* Ajout : Ajouter de nouveaux produits, quantités et fournisseurs.
* Modification : Modifier les informations existantes dans la base de données.
* Suppression : Supprimer des entrées spécifiques.
* Lecture : Afficher les données sous forme de listes organisées.

1. Recherches Dynamiques

* Fonctionnalité permettant de rechercher rapidement des informations spécifiques via des champs de saisie interactifs.
* Utilisation d'algorithmes côté client (JavaScript) ou de requêtes SQL optimisées pour accélérer les résultats.

1. Gestion des Rôles Utilisateurs
   * Administrateurs : Accès complet aux fonctionnalités (ajout, suppression, gestion des utilisateurs).
   * Utilisateurs classiques : Accès limité (consultation des données uniquement).
2. Tableau de Bord
   * Affichage de statistiques de base (nombre total de produits, catégories, etc.).
   * Représentation graphique des données (graphiques à barres ou circulaires simples).



**Technologies Utilisées**

#### Backend

* + - PHP : Pour la logique serveur et l'intégration avec la base de données MySQL.
    - Utilisation de PDO ou MySQLi pour interagir avec la base de données en toute sécurité.

#### Frontend

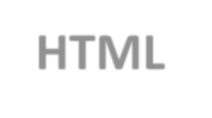
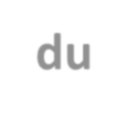
* + - HTML : Structure de l'application.
    - CSS : Stylisation de l'interface utilisateur pour garantir une expérience fluide et moderne.

#### Base de Données

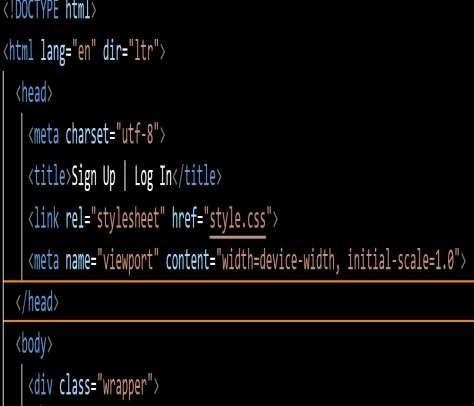
* + - MySQL : Stockage des données (utilisateurs, produits, quantités, fournisseurs).
    - Organisation sous forme de tables bien structurées avec relations entre elles.

#### Optionnel : JavaScript

* + - Améliorations côté client, comme la validation des formulaires et les animations interactives.



**La Partie du HTML**

La section <head> contient les informations essentielles pour configurer la page.

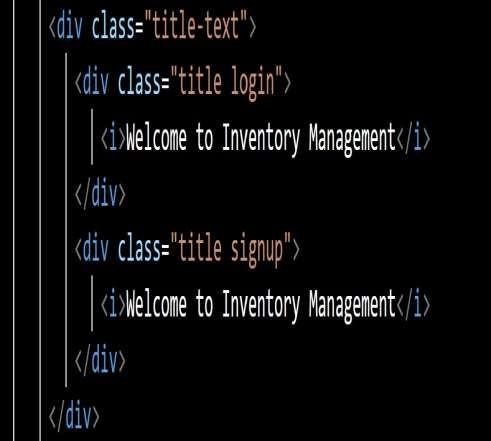
Elle inclut le titre "Sign Up | Log In", visible dans l'onglet du navigateur, ainsi qu'un lien vers la feuille de style externe style.css pour gérer l'apparence de la page.

La balise <meta name="viewport" content="width=device-width, initial-scale=1.0"> est utilisée pour rendre la page responsive, garantissant un affichage adapté à tous les appareils, y compris les smartphones et tablettes.

Dans la section <body>, le conteneur principal .wrapper regroupe tous les éléments interactifs et visuels de la page, comme les formulaires et les contrôles.



**HTML**

Ce bloc contient deux sous-titres distincts :

L'un destiné à la connexion (.title.login) et l'autre à l'inscription (.title.signup).

Ces deux éléments sont conçus pour afficher des messages d'accueil spécifiques à chaque mode, permettant de distinguer clairement les sections "Login" et "SignUp".





**HTML**

Cette section définit les formulaires de connexion (Login) et d'inscription (SignUp), avec une structure interactive et dynamique.



Les onglets sont gérés par des boutons radio invisibles associés à des étiquettes cliquables, permettant de basculer facilement entre les deux modes.

Par défaut, l'onglet "Login" est actif, avec une animation visuelle assurée par l'élément slider-tab.

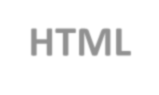
Pour l'inscription, un lien interactif affiche le message "Don't Have Account?" suivi de l'hyperlien "Create A New".

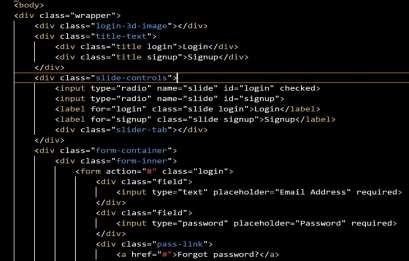
Ce lien permet de guider les utilisateurs vers la création d'un nouveau profil, avec une transition fluide facilitée par JavaScript.

Le formulaire d'inscription comprend trois champs obligatoires : Un pour le nom d'utilisateur, un pour le mot de passe, et un pour confirmer le mot de passe.

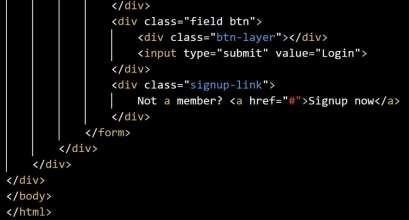
Le formulaire de connexion, quant à lui, inclut deux champs obligatoires pour le nom d'utilisateur et le mot de passe, ainsi qu'un lien "Reset password?" pour la récupération.

**HTML**



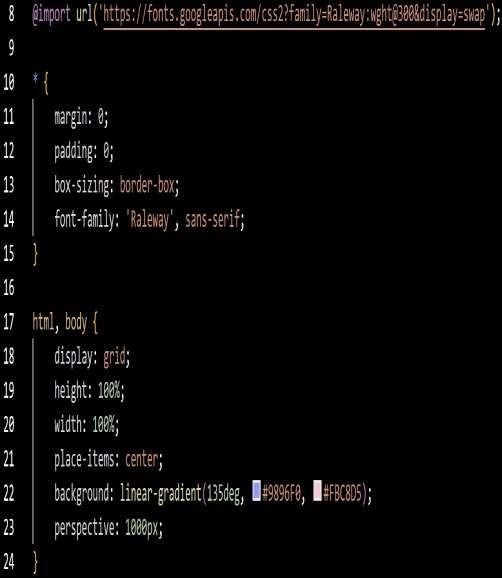
Cette section regroupe le contenu principal de la page, incluant le conteneur principal, les titres, les contrôles de navigation, et les formulaires.

Le conteneur principal, avec la classe wrapper, organise les éléments visuels comme une image stylisée en 3D et deux titres "Login" et "Signup", qui changent dynamiquement selon l'onglet actif.

Les contrôles de navigation sont basés sur des boutons radio invisibles et des étiquettes interactives qui permettent de basculer entre les formulaires, accompagnés d'une animation visuelle gérée par l'élément slider-tab.

Le formulaire de connexion comprend deux champs obligatoires pour l'adresse e-mail et le mot de passe, un bouton stylisé "Login", ainsi qu'un lien "Forgot password?" pour la récupération.

L'ensemble est conçu pour offrir une expérience utilisateur intuitive et fluide grâce aux animations et transitions CSS, tout en garantissant une présentation responsive adaptée à tous les appareils.

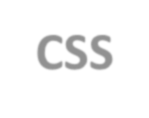
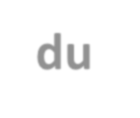




**Cette section la police Raleway est importée pour offrir un design moderne et épuré.**

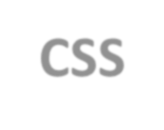
**Le sélecteur universel (\*) est employé pour uniformiser les styles de base sur tous les éléments, garantissant une cohérence visuelle.**

**La mise en page utilise display: grid pour centrer le contenu, tandis qu'un fond en dégradé apporte une touche esthétique professionnelle.**



**La Partie du CSS**

# CSS



Le conteneur .wrapper définit une boîte centrale avec un fond blanc, des coins arrondis, et une ombre subtile.

Un effet de rotation 3D est appliqué avec rotateY pour améliorer l'interactivité.

Les titres "Login" et "Signup" sont affichés côte à côte sur une largeur totale de 200%, permettant un basculement fluide.

Une transition avec une courbe cubic-bezier est utilisée pour un effet de glissement.

La classe .slide-controls organise les éléments horizontalement avec display: flex et les espace uniformément.

Elle a une hauteur de 50px, des bordures légères, des coins arrondis, et utilise overflow: hidden pour une mise en page propre et élégante.





**Ce code gère les onglets "Login" et "Signup" avec des styles**

**dynamiques et des transitions fluides.**

**Les éléments .slide occupent toute la largeur et la hauteur disponibles, avec une police de 18px, un texte centré, et un curseur interactif.**

**L'animation de la barre .slider-tab se déplace selon l'onglet actif, créant un effet visuel fluide grâce à transition.**

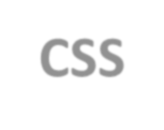
**Les boutons radio, invisibles avec display: none, contrôlent l'état des onglets, tandis que les étiquettes changent de couleur pour indiquer l'onglet sélectionné.**

**La classe .form-container organise le contenu et masque les débordements avec overflow: hidden pour une présentation soignée.**



**CSS**

# CSS



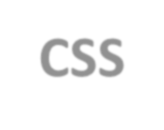
Ce code définit le style et l'organisation des formulaires dans la section .form-container.

La classe .form-inner utilise display: flex pour disposer les formulaires côte à côte sur une largeur totale de 200%, chaque formulaire occupant 50%.

Les champs de saisie .field input sont stylisés avec des bordures arrondies, un espacement interne de 15px, et une transition douce pour changer la couleur de la bordure au focus (border-color: #12e8f0).

Les placeholders ont également un effet de transition, changeant légèrement de couleur lorsqu'un champ est actif. Les liens de récupération de mot de passe et d'inscription sont centrés, colorés en bleu (color: #0e45dd), et leur apparence change subtilement au survol grâce à une décoration soulignée.

# CSS



Ce code stylise les boutons de soumission et l'image 3D décorative.

Le bouton de soumission (form .btn input[type="submit"]) est entièrement responsive, sans bordure ni fond, avec un texte blanc de 20px et des coins arrondis.

Lorsqu'il est survolé, l'effet visuel est animé grâce au déplacement de .btn-layer. L'image 3D (.login-3d-image) est une illustration ronde de 100x100 pixels, centrée avec une ombre subtile pour un effet flottant.

Elle ajoute une touche visuelle moderne et esthétique à l'interface.



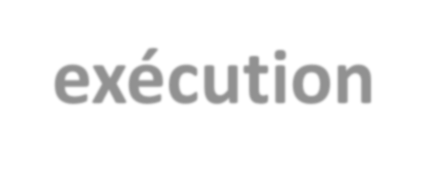
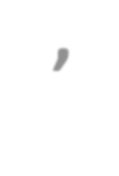
**Voilà l'exécution du code, qui produit une interface fluide et**

**moderne.**

**Les onglets "Login" et "Signup" sont fonctionnels avec une**

**transition fluide entre les formulaires.**

**Les champs de saisie sont clairs, bien organisés, et les boutons animés avec un effet de dégradé ajoutent une touche professionnelle.**

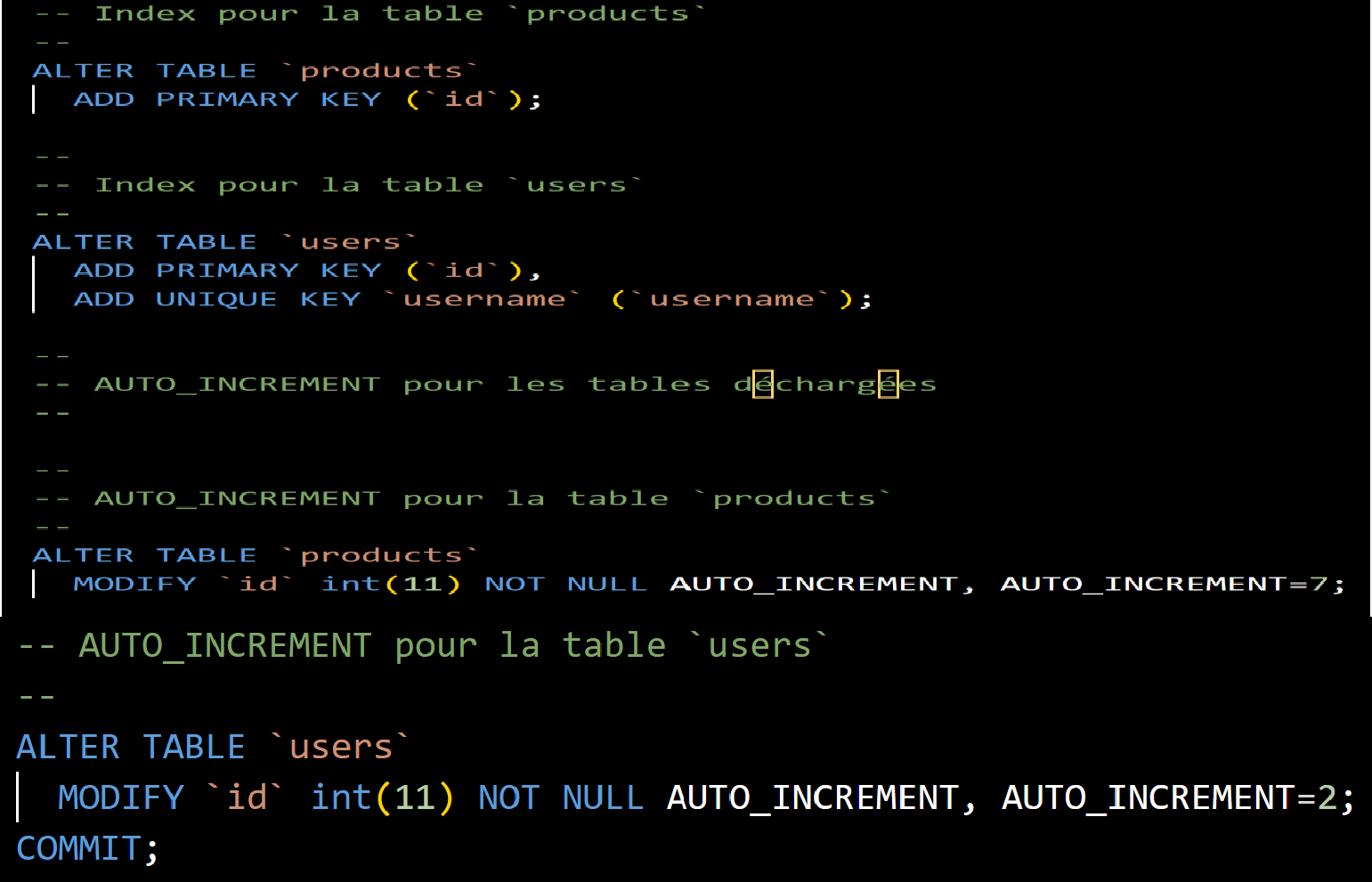
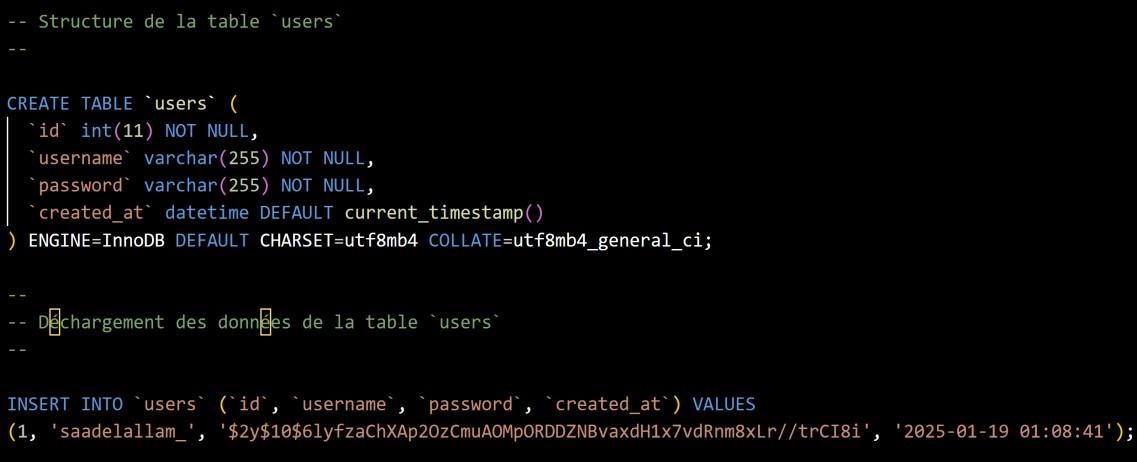


**L ’exécution**

A screenshot of a computer

AI-generated content may be incorrect.

## DataBase Sql



**La base de données inventory\_management se compose de deux tables principales : users et products. La table users est conçue pour gérer les informations des utilisateurs, avec des identifiants uniques et des mots de passe hachés pour garantir la sécurité. Elle inclut également une colonne created\_at pour suivre la date et l'heure de création de chaque compte. La table products, quant à elle, permet de gérer l'inventaire des produits en stock. Elle contient des informations essentielles telles que le nom du produit, la date d'ajout, la quantité disponible et le type de produit. Ce fichier SQL inclut également des données préalablement insérées pour les tests : un utilisateur nommé saadelallam\_ et plusieurs produits, comme "shampoo" et "human". Ces données facilitent l'initialisation et la vérification du bon fonctionnement des fonctionnalités CRUD et des opérations d'authentification dans l'application.**

### La Partie du PHP

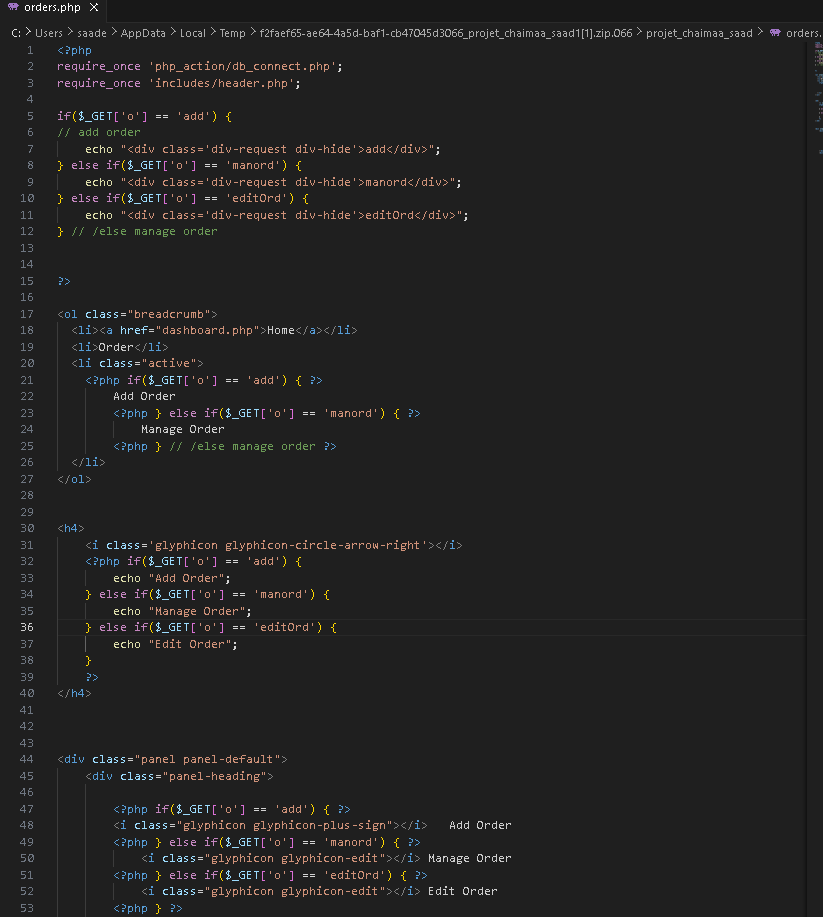
#### db.php

**Ce code établit une connexion sécurisée avec la base de données inventory\_management en utilisant PDO (PHP Data Objects). Il commence par définir les paramètres de connexion, notamment l'hôte (localhost), le nom de la base de données, ainsi que les identifiants d'utilisateur et de mot de passe. Grâce à la classe PDO, la connexion est non seulement efficace mais également sécurisée, offrant la possibilité d'utiliser des requêtes préparées pour éviter les injections SQL. Le code configure également un mode de gestion des erreurs via l'attribut PDO::ATTR\_ERRMODE pour lever des exceptions en cas de problème, facilitant ainsi le débogage. En cas d'échec de connexion, une exception PDOException est capturée, et un message explicatif est affiché. Ce script constitue un point de départ fiable pour intégrer la gestion des données dans une application, en garantissant robustesse et sécurité.**

**Index.php**

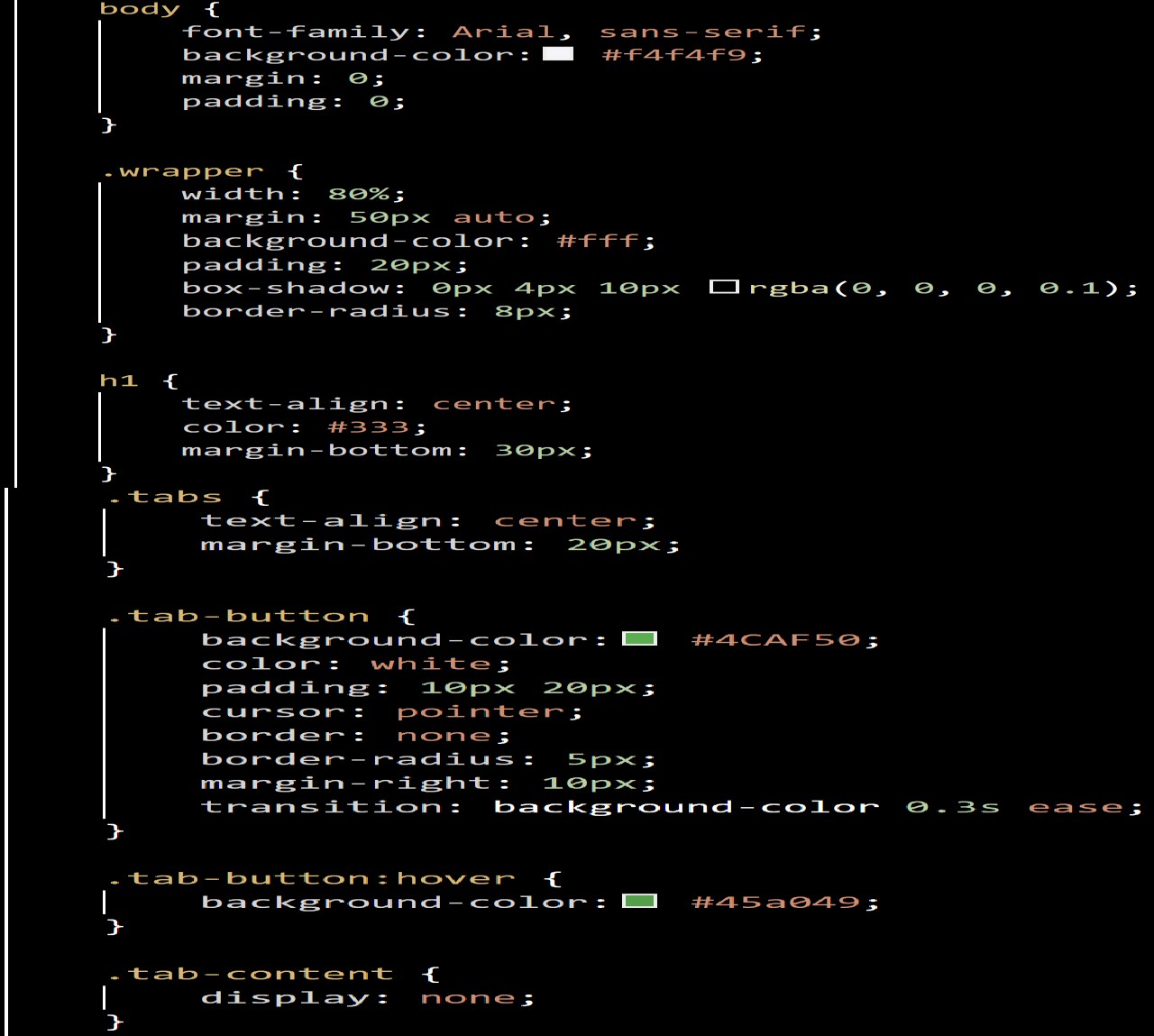
**Ce code implémente un système complet de gestion de l'inscription et de la connexion des utilisateurs dans une application. Il commence par inclure une connexion sécurisée à la base de données via un fichier externe db.php. Lorsqu'un utilisateur soumet un formulaire, le script traite l'action en fonction de la méthode HTTP POST. Pour l'inscription, il vérifie si les mots de passe correspondent, puis hache le mot de passe avec password\_hash() avant de l'enregistrer dans la base de données via une requête préparée. En cas de tentative de connexion, le script recherche l'utilisateur dans la base de données et vérifie la validité du mot de passe en utilisant password\_verify(). Si les informations d'identification sont valides, une session utilisateur est créée, et l'utilisateur est redirigé vers une page protégée (stock.php). En cas d'erreurs, des messages d'avertissement adaptés, comme "Nom d'utilisateur déjà existant" ou "Mot de passe invalide", sont affichés pour une meilleure expérience utilisateur. Ce code offre une solution robuste et sécurisée pour gérer les accès des utilisateurs à l'application.**

Ce code PHP affiche des messages dynamiques à l'utilisateur en vérifiant si la variable $message contient un texte. Si un message est disponible, il est affiché dans une balise <div> avec une classe message pour un style personnalisé. Le code inclut également un fichier JavaScript externe (style.js), qui peut ajouter des fonctionnalités ou des animations pour améliorer l'expérience utilisateur.



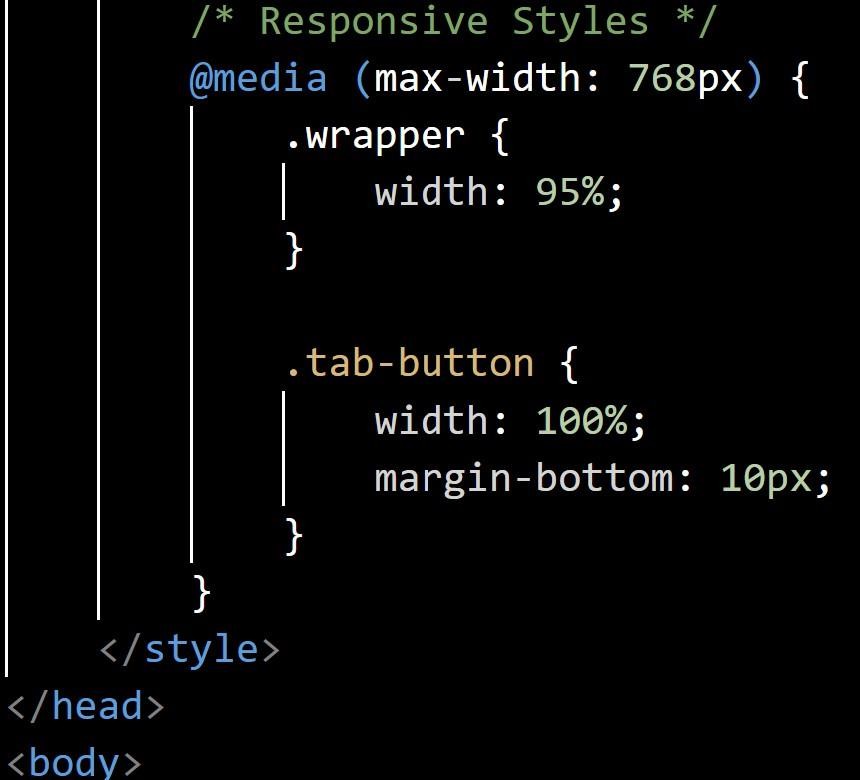
### order.php

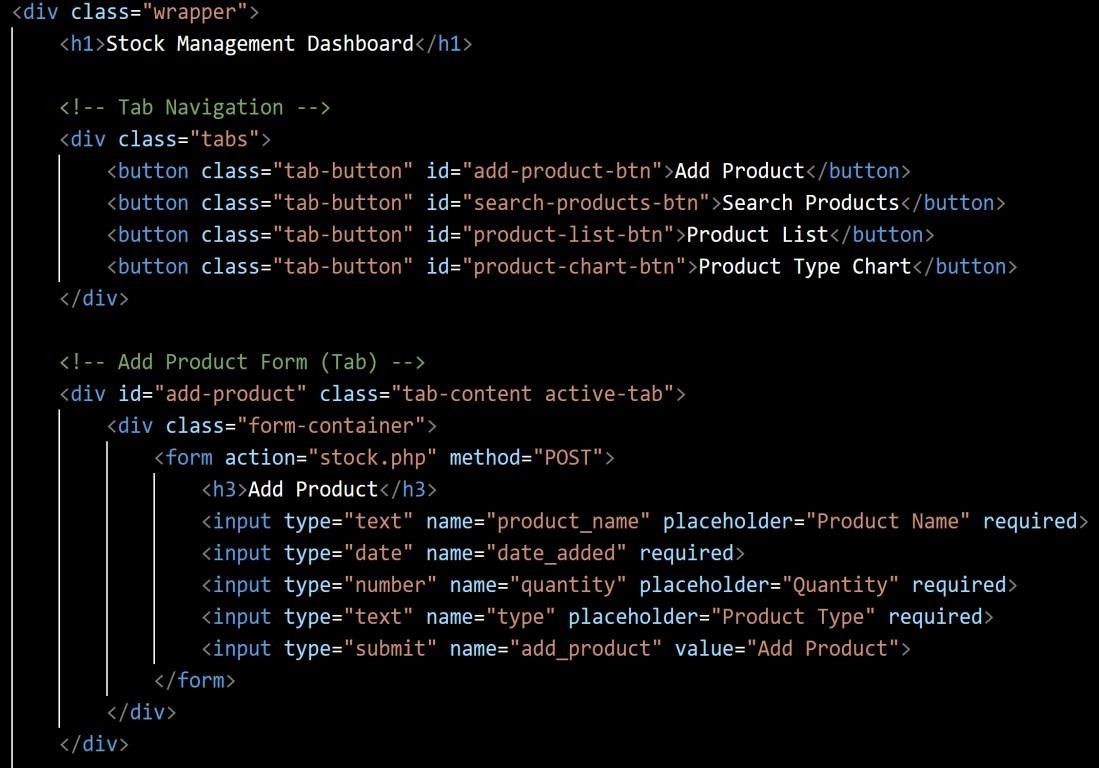
Ce code permet de gérer une interface de gestion des produits avec des fonctionnalités d'ajout et de recherche. Il vérifie d'abord si l'utilisateur est connecté via une session active, sinon il redirige vers index.php. Les produits peuvent être ajoutés en insérant leur nom, date d'ajout, quantité, et type dans la table products via des requêtes préparées pour plus de sécurité. Le script prend également en charge la recherche de produits par nom ou type, ou affiche tous les produits si aucun critère n'est spécifié.

Ce code CSS définit le style d'une interface utilisateur moderne et épurée. La règle globale pour le corps (body) utilise une police sans-serif, un fond clair et une mise en page sans marges pour une apparence propre. Le conteneur principal (.wrapper) est centré avec un fond blanc, des coins arrondis, une ombre subtile et un espacement intérieur confortable, créant un effet visuel professionnel. Les titres (h1) sont centrés avec une couleur sombre pour une meilleure lisibilité. Les onglets interactifs sont stylisés via les classes .tabs et .tab- button, avec des boutons verts arrondis qui changent légèrement de couleur au survol grâce à une transition fluide. Les contenus des onglets (.tab-content) sont initialement masqués, préparant le terrain pour des interactions dynamiques via JavaScript.

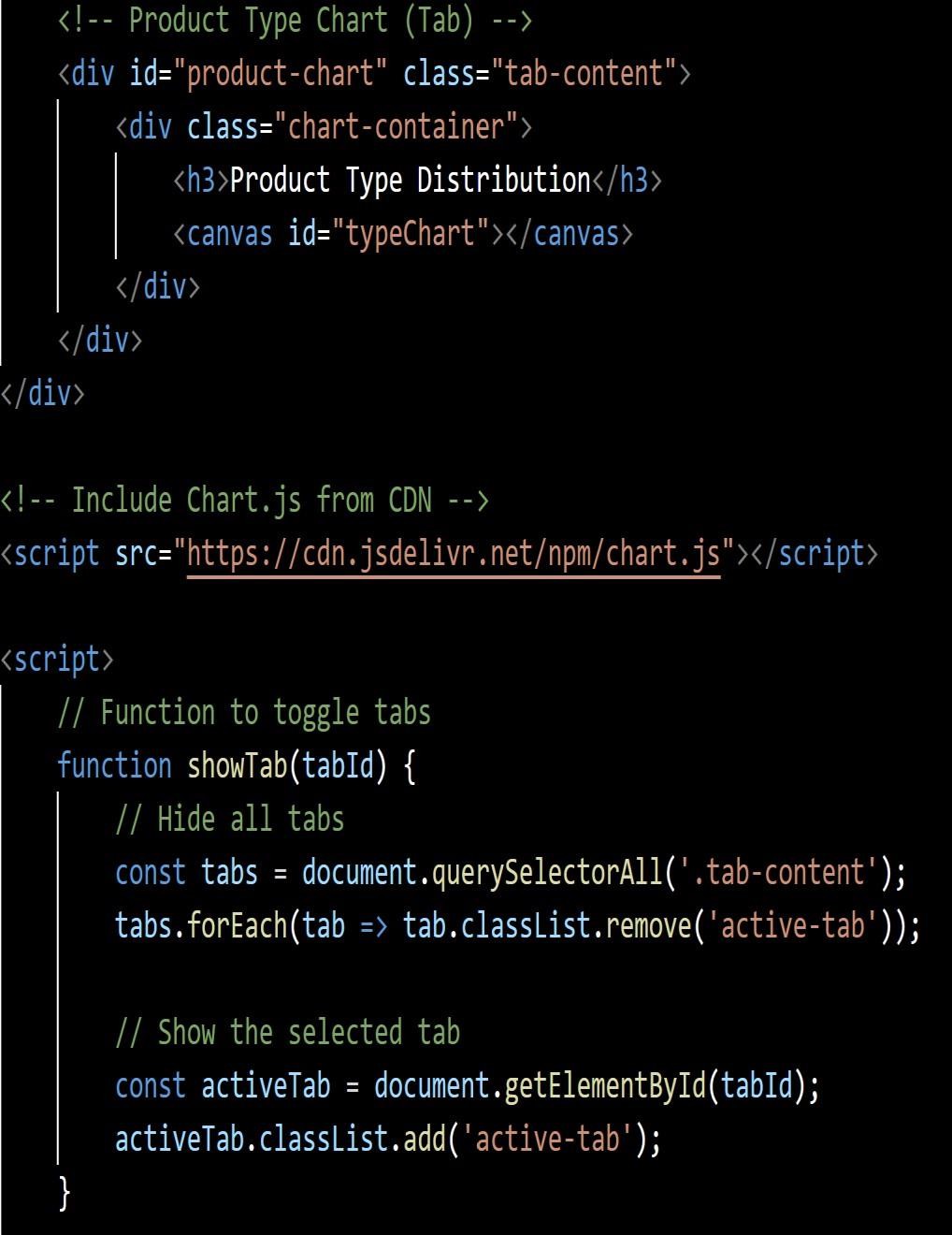


Ce code stylise les formulaires et les tableaux pour une interface propre et organisée. Les formulaires disposent d’un fond clair, de bordures arrondies, et d’une ombre légère pour un design moderne. Les champs de saisie et les listes déroulantes sont larges, avec des bordures simples et un espacement généreux, tandis que les boutons de soumission verts changent légèrement de teinte au survol. Les tableaux sont affichés en pleine largeur, avec des bordures simples et des en-têtes colorés pour une meilleure lisibilité.

Ce code ajoute des styles responsifs pour garantir une bonne expérience utilisateur sur les appareils à écran étroit, comme les tablettes et les smartphones. Lorsque la largeur de l'écran est inférieure à 768 pixels, le conteneur principal (.wrapper) s'ajuste à G5% de la largeur pour une meilleure adaptation. Les boutons d'onglets (.tab-button) s'étendent en pleine largeur et disposent d'un espacement inférieur pour faciliter l'interaction tactile.

Ce code structure une interface utilisateur pour un tableau de bord de gestion des stocks. Le conteneur principal (.wrapper) contient un titre centralisé et une navigation par onglets avec des boutons pour différentes fonctionnalités : ajouter un produit, rechercher des produits, afficher la liste des produits et visualiser un graphique des types de produits. L'onglet actif par défaut affiche un formulaire pour ajouter un produit, comprenant des champs pour le nom, la date d'ajout, la quantité, et le type, avec un bouton pour soumettre les données.

Ce code ajoute deux sections : une pour rechercher des produits et une autre pour afficher la liste des produits. La section "Search Products" contient un formulaire permettant de rechercher des produits par nom ou type, avec un champ pré-rempli pour améliorer l'expérience utilisateur. La section "Product List" affiche un tableau dynamique listant les produits avec leurs noms, dates, quantités et types.

Ce code ajoute un onglet pour afficher un graphique illustrant la répartition des types de produits. La section "Product Type Chart" contient un élément <canvas> pour le rendu du graphique, géré par la bibliothèque Chart.js incluse via un CDN. Une fonction JavaScript, showTab(), permet de basculer dynamiquement entre les différents onglets en masquant les autres et en affichant le contenu sélectionné. Ce composant améliore l'expérience utilisateur en offrant une visualisation claire et interactive des données liées aux types de produits.

Ce code JavaScript gère les interactions des onglets et la visualisation des données de produits avec Chart.js. Des gestionnaires d'événements sont configurés pour les boutons d'onglets, permettant de basculer dynamiquement entre les sections du tableau de bord. Pour la répartition des types de produits, un graphique circulaire est généré à l'aide de Chart.js. Les données pour le graphique (types de produits et quantités totales) sont passées dynamiquement depuis le serveur via PHP et affichées avec des couleurs distinctes pour chaque catégorie.

## L'exécution

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a calendar

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Conclusion**

Ce rapport de projet de gestion web, qui contient les éléments essentiels de fonctionnement de l’application, met en pratique les technologies PHP, HTML, CSS, et MySQL pour développer une application efficace et conviviale axée sur la gestion des stocks. L'application répond aux exigences clés, notamment l'authentification des utilisateurs, les opérations CRUD, la recherche dynamique, et la gestion des rôles. Le tableau de bord interactif, avec ses statistiques et graphiques simples, offre une vue claire des données.