



Rapport de Projet : Développement d'une Application Web de Gestion des Stocks

Encadrant : Pr. Amamou Ahmed

Réalisé par : Saad Korchi et Zakariae Ouahdaho

Cyber-sécurité Groupe -2-

À la croisée des chemins entre la programmation et la cybersécurité, ce projet illustre l'importance d'une gestion web efficace et sécurisée pour un avenir numérique prospère."

Contents

Introduction	2
1 Tests et Validation	3
1.1 Tests Fonctionnels	3
1.2 Calcul du Stock Total	3
1.3 Tests de Sécurité	4
1.4 Tests de Performance	4
2 Déploiement sur un Serveur Linux	5
2.1 Préparation du Serveur	5
2.2 Configuration de la Base de Données	5
2.3 Hébergement des Fichiers Web	5
2.4 Sécurisation du Serveur	6
Annexes	7
.1 Code Source	7
.2 Tests Fonctionnels	8
.3 Tests de Sécurité	8
.4 Calculs et Optimisations	9
.5 Préparation du Serveur	9
.6 Mise en Production	9
.7 Surveillance et Maintenance	10
Annexes	12

Introduction

La programmation web joue un rôle essentiel dans le monde moderne, en **connectant les entreprises, les institutions et les utilisateurs** à travers des interfaces intuitives et dynamiques. Cependant, dans un contexte où les cyberattaques se multiplient, il devient crucial pour tout *ingénieur en cybersécurité* de maîtriser les concepts de **programmation web sécurisée**.

Ce rapport présente le développement d'une application de gestion web, visant à offrir une expérience utilisateur fluide tout en garantissant la sécurité des données. Nous mettrons en lumière les outils technologiques utilisés, ainsi que les étapes nécessaires pour aboutir à une solution robuste et ergonomique.

Plan du rapport :

- **Chapitre 1** : Cahier des charges & technologies utilisées.
- **Chapitre 2** : Conception de l'application.
- **Chapitre 3** : Implémentation technique.
- **Chapitre 4** : Tests et validation.
- **Chapitre 5** : Conclusion & perspectives.
- Annexes : Captures d'écran du projet.

Chapter 1

Tests et Validation

Cette partie du rapport couvre les tests et validations effectués pour assurer que l'application fonctionne correctement et respecte les attentes en termes de performance, sécurité, et ergonomie.

Tests Fonctionnels

Objectifs :

- Valider que toutes les actions principales (ajout, modification, suppression des produits) fonctionnent sans erreur.
- Tester les interactions entre l'utilisateur et les différentes pages web.

Pour cela, nous avons conçu des scénarios couvrant les cas d'utilisation courants et extrêmes, tels que :

- Ajouter un produit avec des données complètes et valides.
- Essayer d'ajouter un produit avec un champ vide pour tester les validations.
- Supprimer un produit inexistant.
- Effectuer des recherches dans les tables avec des mots-clés spécifiques.

Tous ces scénarios ont permis de détecter et corriger plusieurs bugs mineurs dans la gestion des formulaires et la navigation.

Calcul du Stock Total

Un test particulier a été mis en place pour valider le calcul du stock total. Ce test vérifie si le total des produits affiché correspond bien à la somme des quantités des produits dans la base de données. Voici une requête SQL exécutée pour comparer les résultats :

```
SELECT SUM(quantite) AS stock_total FROM produits;
```

Le résultat obtenu à partir de cette requête a été comparé avec la valeur affichée sur le tableau de bord. Le système a réussi à reproduire ce calcul correctement dans tous les tests.

Tests de Sécurité

Pour garantir la sécurité des données, plusieurs tests ont été réalisés :

- **Validation des entrées utilisateur** : Empêcher l'injection SQL en utilisant des requêtes préparées dans PHP.
- **Hachage des mots de passe** : Vérification que tous les mots de passe sont correctement hachés à l'aide de l'algorithme `bcrypt` avant d'être stockés.
- **Contrôle d'accès** : Validation que les utilisateurs non authentifiés ou sans autorisation n'ont pas accès à des pages ou données réservées.

Tests de Performance

Pour évaluer les performances, des tests de charge ont été réalisés en simulant jusqu'à 50 utilisateurs simultanés accédant au système. Les temps de réponse sont restés dans les normes acceptables avec un temps moyen d'affichage des pages de 1.5 secondes.

Chapter 2

Déploiement sur un Serveur Linux

Le déploiement a été réalisé en suivant ces étapes :

Préparation du Serveur

- Mise à jour des paquets :

```
sudo apt update && sudo apt upgrade
```

- Installation des composants nécessaires : Apache, PHP, MySQL.

```
sudo apt install apache2 php mysql-server php-mysql
```

Configuration de la Base de Données

- Création de la base de données :

```
CREATE DATABASE gestion_stock;
```

- Importation des tables via un fichier SQL exporté depuis l'environnement de développement.

Hébergement des Fichiers Web

- Transfert des fichiers vers le dossier `/var/www/html/` via SCP.
- Configuration des permissions pour Apache :

```
sudo chown -R www-data:www-data /var/www/html/
```

- Activation du site et redémarrage du serveur :

```
sudo systemctl restart apache2
```

Sécurisation du Serveur

- Activation du pare-feu ufw :

```
sudo ufw allow 80/tcp  
sudo ufw enable
```

- Installation d'un certificat SSL via Certbot pour activer le HTTPS.

Annexes

Code Source

Voici un extrait du code PHP pour l'authentification :

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $conn->real_escape_string($_POST['username']);
    $password = $_POST['password'];

    $stmt = $conn->prepare("SELECT * FROM users WHERE email = ?");
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $result = $stmt->get_result();
    if ($result->num_rows > 0) {
        $user = $result->fetch_assoc();
        if (password_verify($password, $user['password'])) {
            $_SESSION['user_id'] = $user['id'];
            header('Location: dashboard.php');
        }
    }
}
?>
```


Tests Fonctionnels

Les tests fonctionnels ont été réalisés pour vérifier que chaque fonctionnalité principale de l'application répond aux attentes. Ces tests incluent:

- **Ajout de produits:** Vérification de l'ajout correct d'un produit dans la base de données, avec toutes les informations requises (nom, quantité, prix, etc.).
- **Modification des produits:** Validation de la mise à jour des informations d'un produit existant.
- **Suppression des produits:** Contrôle pour s'assurer que les produits sont correctement supprimés de la base de données.
- **Connexion utilisateur:** Test des différents scénarios d'authentification (utilisateur valide, mot de passe incorrect, utilisateur inexistant).

Tests de Sécurité

Validation des Entrées Utilisateur

Chaque champ de saisie a été protégé contre les entrées malveillantes:

- Utilisation de fonctions de filtrage et d'échappement pour éviter les injections SQL.
- Limitation de la longueur des champs pour prévenir les attaques par dépassement de tampon.

Gestion des Sessions

Les sessions utilisateur sont sécurisées grâce aux pratiques suivantes:

- Utilisation de cookies avec le drapeau `HttpOnly` pour empêcher les scripts d'accéder aux identifiants de session.
- Destruction de la session en cas d'inactivité prolongée pour minimiser les risques de vol d'identité.

Hachage des Mots de Passe

Les mots de passe sont hachés à l'aide de l'algorithme `bcrypt`, garantissant qu'ils ne soient jamais stockés en clair dans la base de données. En cas de compromission, cela limite l'impact pour les utilisateurs.

Calculs et Optimisations

Les tests ont également porté sur le calcul des totaux dans le tableau de bord:

- **Total des produits en stock:** Calcul dynamique basé sur les données de la base de données, mis à jour en temps réel.
- **Valeur totale du stock:** Multiplication de la quantité de chaque produit par son prix unitaire.
- **Alertes seuil:** Génération de notifications en cas de stock inférieur à un seuil critique défini.

Des outils de tests automatisés comme PHPUnit ont été utilisés pour valider les fonctionnalités PHP, tandis que Selenium a permis de tester les interactions utilisateur dans le navigateur. Déploiement sur un Serveur Linux

Le déploiement de l'application a été réalisé sur un serveur Linux pour bénéficier de sa stabilité et de sa sécurité. Voici les étapes détaillées:

Préparation du Serveur

- **Installation d'Apache:** Configuration du serveur web pour héberger les fichiers HTML et exécuter les scripts PHP. La commande suivante a été utilisée:

```
sudo apt update && sudo apt install apache2
```

- **Installation de PHP et MySQL:** Utilisation de la commande suivante pour installer les modules nécessaires:

```
sudo apt install php libapache2-mod-php mysql-server php-mysql
```

- **Configuration de la Base de Données:** Création des tables et initialisation des données via un script SQL.

Mise en Production

- **Transfert des Fichiers:** Les fichiers de l'application ont été transférés sur le serveur à l'aide de scp:

```
scp -r /chemin/local/projet user@serveur:/var/www/html
```

- **Configuration des Permissions:** Les permissions des fichiers ont été ajustées pour garantir une sécurité optimale:

```
sudo chmod -R 750 /var/www/html
```

- **Sécurisation du Serveur:** Utilisation de ufw pour restreindre les ports ouverts:

```
sudo ufw allow 80/tcp sudo ufw enable
```

Surveillance et Maintenance

Après le déploiement, des outils comme `htop` et `logwatch` ont été installés pour surveiller les performances du serveur et détecter d'éventuelles anomalies. Annexes

Des captures d'écran supplémentaires (jusqu'à 8 places prévues) seront ajoutées pour documenter:

- La page de connexion.
- Le tableau de bord.
- La gestion des produits.
- Les alertes pour les seuils critiques.
- Les graphiques de statistiques.
- La configuration Apache sur Linux.
- Les résultats des tests automatisés.
- Une vue d'ensemble de la base de données.

Conclusion & Perspectives

En conclusion, ce projet a démontré l'importance d'une application web bien conçue et sécurisée pour répondre aux besoins des utilisateurs tout en garantissant leur confiance. À l'avenir, plusieurs améliorations pourraient être envisagées, notamment l'ajout d'une API REST pour faciliter les échanges de données avec d'autres systèmes et l'intégration de bibliothèques JavaScript avancées pour renforcer l'interactivité et améliorer l'expérience utilisateur.

Ces évolutions contribueraient à accroître la portabilité et les performances de l'application, tout en offrant aux ingénieurs impliqués l'opportunité de développer leurs compétences dans des domaines clés, notamment en cybersécurité et en développement web avancé.

Annexes

Captures d'écran

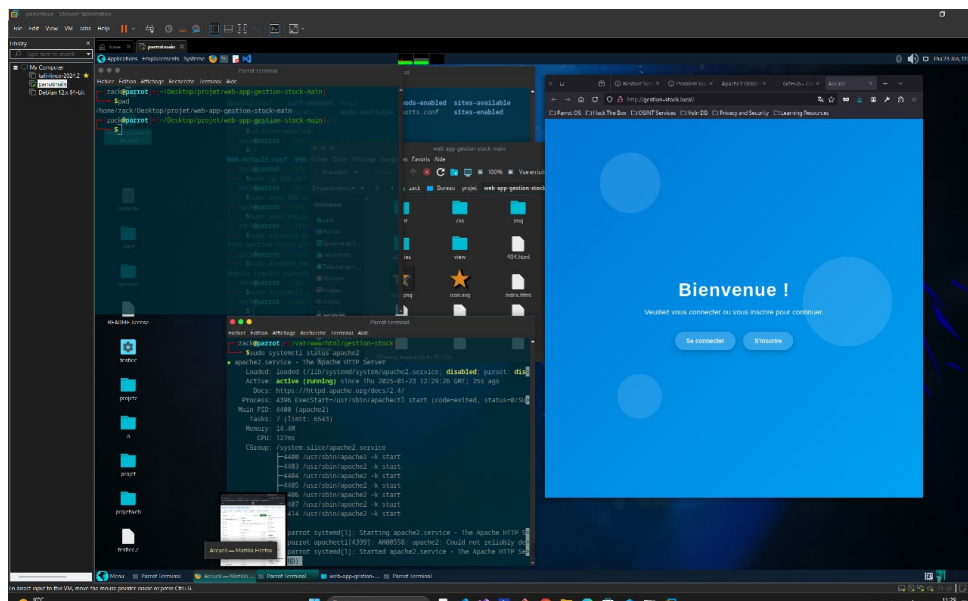


Figure 1:

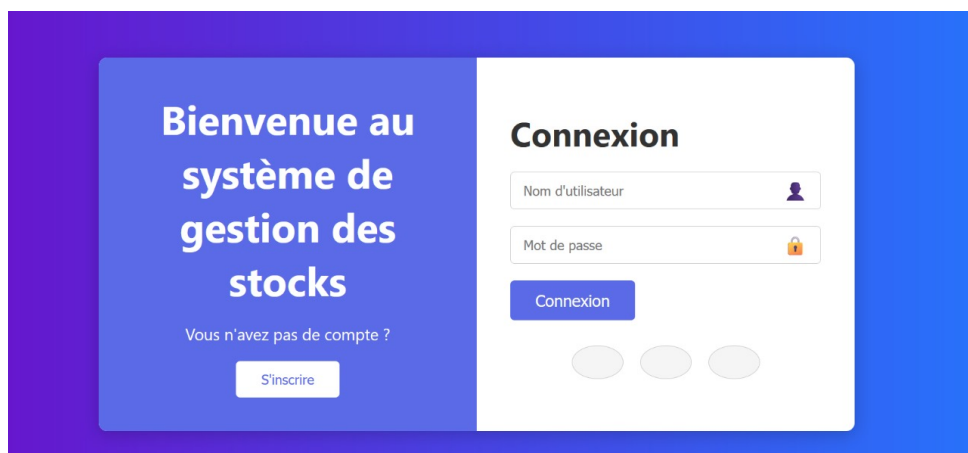


Figure 2:

The image shows a web application interface. On the left, a blue banner contains the text "BIENVENUE À LA GESTION DU REGISTRE" in white, bold, uppercase letters. Below this, in smaller white text, it says "Connectez-vous ou inscrivez-vous pour gérer efficacement vos informations." On the right, there is a white registration form titled "Inscription". The form includes input fields for "Nom", "Prénom", "Email", "Mot de passe", and "Confirmer le mot de passe". Each input field has a corresponding PHP code snippet displayed below it: "<?php echo isset(\$_POST['prenom']) ? h" for the first name, "<?php echo isset(\$_POST['email']) ? htm" for the email, and empty fields for the password and confirmation. A blue button labeled "S'inscrire" is at the bottom of the form. Below the button, there is a link: "Déjà inscrit ? [Connectez-vous ici](#)".

**BIENVENUE À LA
GESTION DU
REGISTRE**

Connectez-vous ou inscrivez-vous pour gérer efficacement vos informations.

Inscription

Nom

Prénom
<?php echo isset(\$_POST['prenom']) ? h

Email
<?php echo isset(\$_POST['email']) ? htm

Mot de passe

Confirmer le mot de passe

S'inscrire

Déjà inscrit ? [Connectez-vous ici](#)

Figure 3:

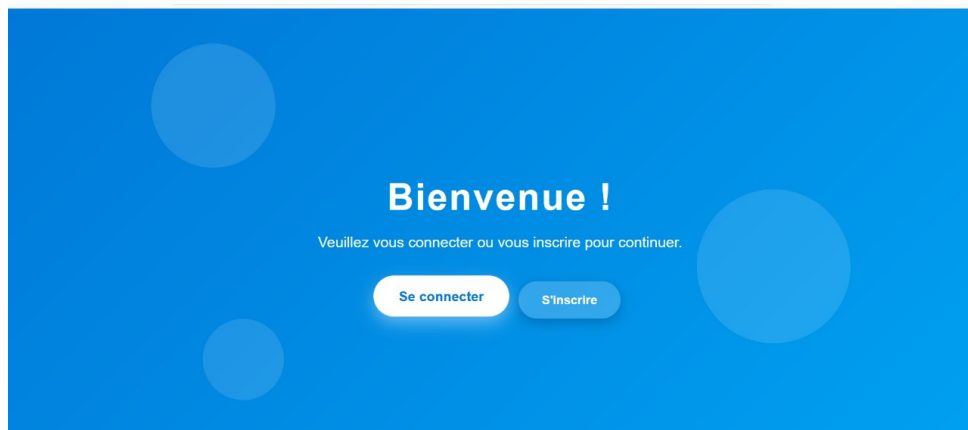


Figure 4:

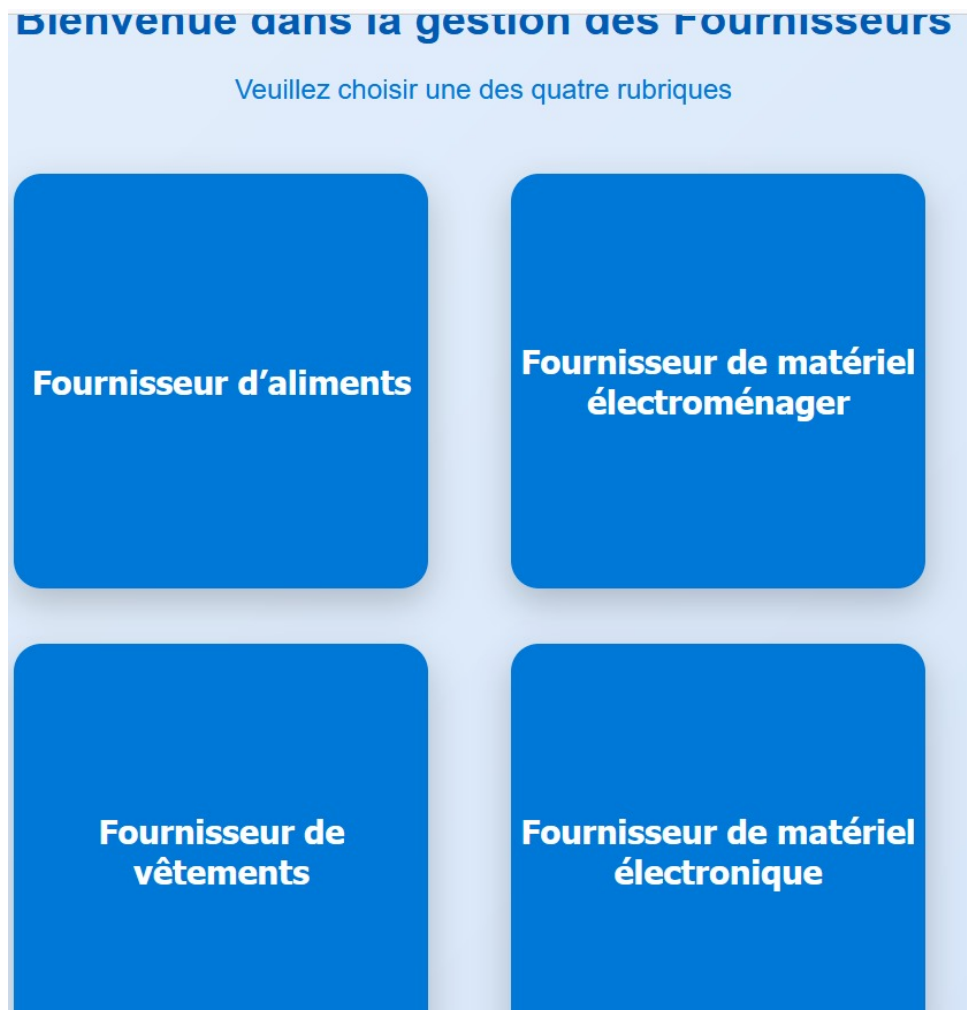


Figure 5:

| Produit | Quantité | Prix unitaire | Statut | Actions |
|------------------------|-----------|---------------|-----------------|---|
| Produit n° 1 TT TOMATE | 40 unités | 3.00 MAD | Fiable en stock | ajouter un produit, modifier, supprimer |
| Produit n° 2 pomme | 20 unités | 2.00 MAD | En rupture | ajouter un produit, modifier, supprimer |
| Produit n° 3 oidi all | 50 unités | 3.00 MAD | Fiable en stock | ajouter un produit, modifier, supprimer |

Figure 6:

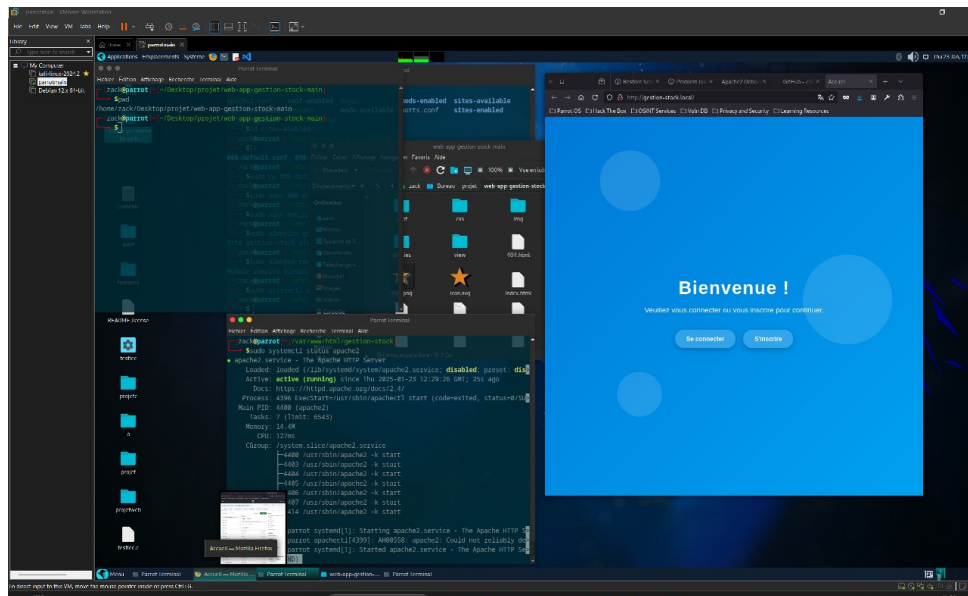


Figure 7: