Audit de la V1 - Gestion de Stock

Contexte

La version 1 (V1) du système de gestion de stock pour une entreprise de vente de maquettes d'avions en papier a été développée. Cette version est un prototype (POC) permettant les opérations CRUD pour les entités principales : - Produits - Catégories - Fournisseurs - Clients - Commandes et Lignes de Commande

L'objectif de cet audit est d'évaluer la V1 pour identifier les failles de sécurité, les limitations fonctionnelles et les pistes d'amélioration avant le développement de la V2.

1. Failles de Sécurité Identifiées

a) Vulnérabilité aux injections SQL

Description :
- Les requêtes SQL directes dans le code utilisent des chaînes concaténées pour intégrer les paramètres utilisateur.
- Cela rend le système vulnérable à des attaques par injection SQL.

Exemple de code vulnérable :

```
app.get('/produits/:id', (req, res) => {
  const query = `SELECT * FROM Produits WHERE id = ${req.params.id}`;
  db.query(query, (err, results) => {
    if (err) return res.status(500).json({ error: err.message });
    res.json(results);
  });
});
```

Risques :
- Permet à un utilisateur malveillant de manipuler la requête SQL.
- Exemple d'attaque : id=1; DROP TABLE Produits.

b) Manque de validation des données

Description :
- Les entrées utilisateur ne sont pas validées côté serveur avant d'être insérées ou utilisées dans la base de données.

Conséquences :
- Risques d'erreurs ou d'incohérences dans la base de données, par exemple :
  - Quantité négative.

- Champs requis laissés vides.
- Formats incorrects (email, téléphone, etc.).

c) Absence d'authentification et de permissions

Description :
- Tous les endpoints sont accessibles sans restriction.

Conséquences :
- N'importe qui dans le réseau interne peut modifier ou supprimer des données critiques.

d) Exposition des messages d'erreur

Description :
- Les erreurs de la base de données sont renvoyées telles quelles dans les réponses JSON.

Risques :
- Les messages peuvent révéler des informations sensibles sur la structure de la base de données.

2.Limitations Fonctionnelles

a) Vérifications métier absentes

Problèmes identifiés :

1. **Stock insuffisant** :
   - Une commande peut être passée pour un produit qui n'a pas assez de stock.
2. **Suppression incohérente :**
   - Les entités comme les produits ou les catégories peuvent être supprimées même si elles sont référencées dans d'autres tables (par exemple, dans des commandes ou lignes de commande).

b) Pas de recherche avancée

Limitations :

- Les endpoints ne permettent pas de filtres avancés (par exemple, commandes par date ou produits par catégorie).

c) Absence de statistiques

Limitations :

- Aucune fonctionnalité pour générer des statistiques simples, telles que :

- Produits les plus vendus.
- Chiffre d’affaires par période.

3. Pistes d'Amélioration pour la V2

Utiliser des **requêtes paramétrées** pour éviter les injections SQL.

Ajouter une authentification avec JWT ou sessions pour sécuriser les endpoints sensibles.

Vérifier la disponibilité du stock **avant de valider une commande.**

Conclusion

La V1 remplit son objectif principal de démonstration, mais présente des failles critiques en termes de sécurité et des limitations fonctionnelles notables. La V2 doit se concentrer sur :

- La sécurisation des données.
- L’ajout de vérifications métier et de validations.
- L’amélioration des fonctionnalités pour répondre aux besoins pratiques.

Avec ces améliorations, le système sera prêt pour une utilisation en production.