

Analyse des impacts et des risques - Application web de gestion de commandes

Nom : Khalil Lamiaa

1. Utilisateurs : Inscriptions et authentications

Risques :

- Fuites de données, attaques par force brute
- Problèmes de performance en cas de pic de connexions

Recommandations :

- Tests de sécurité (ex : tests de pénétration, validation des tokens OAuth2)
- Tests de charge
- Rate-limiting pour bloquer les abus

2. Commandes : Gestion et suivi

Risques :

- Perte de données si le système tombe
- Ralentissements en cas de forte charge

Recommandations :

- Sauvegardes automatiques
- Mécanismes de récupération
- Tests de charge et de résilience

3. Catalogue : Produits et stocks

Risques :

- Incohérences si plusieurs mises à jour en même temps
- Ralentissements sous charge

Recommandations :

- Mécanismes de verrouillage ou gestion des conflits
- Redis pour le cache
- Tests de performance

4. Paiement : Service tiers

Risques :

- Panne du service de paiement
- Problèmes de sécurité avec les infos bancaires

Recommandations :

- Tests d'intégration
- Sécurisation des données (ex : SSL/TLS)
- Fallback en cas de panne

5. Notifications : Emails et messages

Risques :

- Messages perdus si surcharge
- Retards dans l'envoi

Recommandations :

- Utiliser Kafka
- Tests de résilience et de performance

6. Back-office : Interface admin

Risques :

- Bugs après mises à jour
- Lenteurs avec beaucoup de données

Recommandations :

- Tests unitaires et d'intégration
- Pagination et filtres
- Tests de performance

7. Frontend (React.js)

Risques :

- Lenteur de l'interface
- Bugs après modification

Recommandations :

- Tests unitaires et d'intégration
- Lazy loading, code splitting
- Tests de performance

8. Backend (FastAPI)

Risques :

- Erreurs sous charge
- Régressions après mise à jour

Recommandations :

- Tests unitaires et d'intégration
- Monitoring + logging
- Tests de charge

9. Base de données (PostgreSQL)

Risques :

- Requêtes lentes
- Indisponibilité si panne

Recommandations :

- Optimisation SQL
- Réplication + basculement
- Sauvegardes régulières

10. Cache (Redis)

Risques :

- Incohérences entre cache et BDD
- Saturation

Recommandations :

- Stratégies d'invalidation
- Tests de performance
- Surveillance de la charge

11. Message Queue (Kafka)

Risques :

- Perte de messages
- Retards de traitement

Recommandations :

- Tests de résilience
- Monitoring
- Tests de performance

12. Authentification (OAuth2 avec FastAPI)

Risques :

- Fuites de tokens

- Force brute

Recommandations :

- Tests de sécurité
- Rate-limiting
- Tests de charge