# État des lieux rapide

**Solide / prêt :**

* Catalogue (catégories flexibles, variants, images).
* Devis/Commandes/Factures + historiques de statut.
* Mouvements de stock + raisons.
* Rôles & permissions, audit/logs, exports PDF/Excel.
* Base UI/UX Inertia réutilisable et performante.

**Partiel / à renforcer :**

* **Inventaire** : réservations de stock pendant le checkout, inventaires cycliques, transferts d’emplacements.
* **Tarification** : règles de prix (prix promo, paliers B2B, listes de prix par segment/client).
* **Taxes** : moteur de TVA par pays/région (et cas Maroc), arrondis, exonérations B2B, incoterms (si besoin).
* **Notifications** : emails & in-app sur événements (commande payée, stock bas, facture échue).
* **Perfs** : index DB systématiques, queues (emails/PDF), cache config/queries, CDN médias.

**Manque encore pour un e-commerce complet :**

* **Panier & Checkout** (panier invité + client, adresses, frais de port, récap, validation).
* **Paiements** (intégrations Stripe/PayPal/CIH/M2M… + webhooks, remboursements).
* **Expédition** (calcul des frais, transporteurs, suivi, états d’expédition).
* **Promos & coupons** (codes, remises catégorielles, BOGO, fenêtres de validité).
* **Retours/RMA & remboursements** (flux retour, remise en stock, avoirs).
* **Comptes clients** (inscription, adresses, commandes, factures, RGPD).
* **Recherche catalogue** (Scout + Meilisearch/Elastic, filtres, tri).
* **API Storefront** (REST/GraphQL) pour une vitrine séparée (ex. Next.js).
* **Internationalisation & multi-devise** (taux, arrondis, affichage).

# Recommandation de roadmap (pragmatique)

1. **Checkout complet + Paiements + Webhooks**
   * Panier, adresse(s), livraison, récap, paiement (Stripe d’abord), gestion des statuts payés/échoués + remboursements.
2. **Réservations de stock & WMS light**
   * Réserver le stock au checkout, libérer à l’abandon/échec paiement, emplacements simples (entreposage).
3. **Expédition**
   * Modèle Expédition, tarifs (poids/zones), numéros de suivi, états partiel/complet.
4. **Promotions/Coupons**
   * Engine de règles (scope produit/cat/carte, %/fixe, min panier, dates, usage).
5. **Comptes clients & RMA**
   * Espace client (commandes, factures PDF), demandes de retour, avoirs.
6. **Recherche & API Storefront**
   * Scout + Meilisearch, facets; exposer /api/v1 (auth publique en lecture).

# MVP “production-ready” checklist

* Panier & Checkout (invité + connecté)
* Stripe (carte + webhooks), statut ordre ↔ paiement
* Réservation de stock pendant checkout
* Calcul TVA (Maroc + base UE générique si besoin)
* Frais de livraison (table simple zones/poids)
* Emails transactionnels (commande, facture, expédition)
* Coupons simples (pourcentage / fixe)
* Espace client (adresses, commandes, factures)
* Logs webhooks + retries en queue
* Index DB & queues & cache (perfs)

🎯 **Verdict** : oui, ta base est **au-dessus de la moyenne** pour démarrer un backend e-commerce. On peut passer en mode livraison de features.  
Je te propose d’attaquer directement par le **module “Checkout + Paiements (Stripe) + Webhooks”** : je te fournis migrations, modèles, routes, contrôleurs Inertia, et l’enchaînement d’états (draft → pending\_payment → paid/failed/canceled) avec réservations de stock. Ensuite on enchaîne sur **expédition** puis **promos**.

Si ça te va, je commence tout de suite par ce module avec un plan technique et les premiers fichiers.

**. Multi-tenancy manquant**

* Aucune notion de "tenant" ou "organisation"
* Les données sont partagées entre tous les utilisateurs
* Pas d'isolation des données par client SAAS

**2. Catégories trop rigides**

* Les catégories sont globales et fixes
* Le système de spécialisation dans config/catalog.php n'est pas flexible
* Chaque tenant devrait pouvoir créer ses propres catégories

**3. Produits pas adaptés au e-commerce**

* Manque de champs essentiels : variations, SEO, gestion des stocks avancée
* Pas de support pour les produits virtuels/téléchargeables
* Système d'images trop basique

**4. Architecture e-commerce incomplète**

* Pas de gestion des paniers
* Pas de système de commandes clients (front-end)
* Pas de gestion des paiements
* Pas de système de livraison

**1. Les Attributs = Propriétés de tes produits**

Les attributs définissent les caractéristiques que peuvent avoir tes produits dans cette catégorie.

**Exemples concrets :**

* **Catégorie "Ordinateurs"** → Attributs : RAM, Processeur, Carte graphique, Couleur
* **Catégorie "Smartphones"** → Attributs : Stockage, Couleur, Marque, Écran
* **Catégorie "Vêtements"** → Attributs : Taille, Couleur, Matière, Marque

**2. Les Options = Valeurs possibles pour un attribut**

Pour les attributs de type select ou multiselect, tu définis les choix possibles.

**Exemple attribut "Couleur" :**

* Option 1 : label: "Rouge", value: "rouge", color: "#ff0000"
* Option 2 : label: "Bleu", value: "bleu", color: "#0000ff"
* Option 3 : label: "Vert", value: "vert", color: "#00ff00"

**3. Les Types d'attributs**

* **text** : Texte libre (ex: nom du modèle)
* **select** : Liste déroulante (ex: couleur, marque)
* **multiselect** : Choix multiples (ex: compatibilités)
* **number** : Nombre entier (ex: quantité RAM en GB)
* **boolean** : Oui/Non (ex: Bluetooth activé)

**4. Les Champs de configuration**

* **is\_required** : Obligatoire lors de la création produit
* **is\_filterable** : Apparaît dans les filtres de recherche
* **is\_searchable** : Inclus dans la recherche textuelle
* **show\_in\_listing** : Affiché dans la liste des produits
* **validation\_rules** : Règles de validation (min/max, regex...)