1. **PRESENTATION DE L'ENTREPRISE**

* Identité : "GentlemanBio", une marque de cosmétiques bio-luxe pour hommes
* Valeurs : Qualité premium, respect de l'environnement, masculinité moderne
* Produits : Gamme complète de soins pour hommes (visage, corps, cheveux, barbe)
* Positionnement : Haut de gamme, écologique, exclusivement masculin

1. Objectifs du site e-commerce

* Générer 100 000 € de chiffre d'affaires la première année
* Atteindre 5 000 visites mensuelles d'ici 6 mois
* Obtenir un taux de conversion de 3% à 12 mois

1. Public cible

Persona principale : Romaric, 21 ans, cadre urbain, soucieux de son apparence et de l'environnement.

1. Architecture technique

* Langage de programmation : PHP 8.2+
* Base de données : MySQL 8.0+
* Serveur web : Apache 2.4+ ou Nginx
* Framework PHP : Laravel 10+ (pour une structure MVC robuste)
* Frontend : HTML5, CSS3 (avec framework Bootstrap 5), JavaScript (avec Vue.js 3)

## 2. Fonctionnalités principales (en PHP)

### 2.1 Gestion des produits

- Système CRUD pour l'ajout, la modification et la suppression de produits

- Catégorisation et étiquetage des produits

- Gestion des variantes (taille, couleur, etc.)

- Gestion des stocks avec alertes de réapprovisionnement

### 2.2 Panier et commandes

- Panier d'achat dynamique avec Ajax

- Tunnel de commande multi-étapes

- Génération de factures PDF (utilisation de la bibliothèque FPDF)

- Suivi des commandes et mises à jour de statut

### 2.3 Gestion des utilisateurs

- Inscription, connexion, et gestion de profil

- Système de rôles et permissions (admin, client, etc.)

- Historique des commandes et liste de souhaits

### 2.4 Paiement

- Intégration de passerelles de paiement (Stripe, PayPal) via leurs API PHP

- Gestion sécurisée des transactions

### 2.5 Recherche et filtrage

- Moteur de recherche avancé avec Elasticsearch ou Algolia

- Filtres dynamiques pour les produits (AJAX)

## 3. Sécurité

- Implémentation de HTTPS

- Protection contre les injections SQL, XSS, CSRF

- Hachage sécurisé des mots de passe (utilisation de password\_hash())

- Validation et assainissement des entrées utilisateur

## 4. Performance

- Mise en cache avec Redis

- Optimisation des requêtes de base de données

- Minification des assets (CSS, JavaScript)

- Utilisation de CDN pour les fichiers statiques

## 5. SEO et marketing

- URLs propres et conviviales pour les moteurs de recherche

- Génération de sitemap XML

- Intégration de métadonnées dynamiques

- Système de notation et d'avis produits

## 6. Intégrations

- API RESTful pour les intégrations futures (mobile, etc.)

- Webhooks pour les notifications en temps réel

- Intégration avec les réseaux sociaux (partage, login)

## 7. Administration

- Tableau de bord administrateur complet

- Rapports et analyses des ventes

- Gestion des stocks et des commandes

- Gestion du contenu (CMS pour blog, pages statiques)

## 8. Tests et déploiement

- Tests unitaires et d'intégration (PHPUnit)

- Environnement de staging pour les tests

- Déploiement automatisé (CI/CD avec GitLab ou Jenkins)

## 9. Maintenance et évolutivité

- Documentation complète du code

- Logs d'erreurs et de performances

- Structure modulaire pour faciliter les futures extensions