Site e-commerce Stripe

I. Contexte du projet

A. Les objectifs

L'objectif du site est de proposer une solution e-commerce intuitive, claire, et avec un accent fort porté sur la sécurité des données et du code. L'application dissociera deux types d'utilisateurs distincts : les administrateurs, et les clients. Le site proposera un système de page produit dynamique, de panier, d'auto-login, et une solution de paiement.

- La partie administrateur permettra de gérer l'inventaire des marchandises, mais aussi d'accéder aux profils des différents utilisateurs, de récupérer leurs commandes et de mettre à jour le statut de celles-ci.
- La partie client permettra à l'utilisateur d'accéder à son profil et de récupérer l'historique de ses commandes, et de consulter leur suivi. Il pourra également gérer ses adresses, modifier son mot de passe ou supprimer son compte.
- La partie catalogue se dissociera via deux items de navigation : Les marques, et les produits. Chaque item sera accessible depuis l'autre (Une marque affichera les produits, un produit affichera sa marque).

Concernant la partie front, la navigation sera simplifiée et le design épuré pour concentrer l'effort autour de la sécurisation de l'application. Le but restera de proposer une facilité et une rapidité d'accès pour l'utilisateur aux différents items du site.

B. Périmètre du projet

Le projet se limite à la réalisation d'un site d'e-commerce, responsive, avec une partie back admin, une partie back client, une partie front avec une page d'accueil présentant le site et certains produits, une page affichant les marques, une page dynamique par marque affichant les produits de la marque,

une page dynamique par produit, une page de panier, et une page de paiement. L'effort est concentré sur la sécurité.

II. Contexte technique

A. Les objectifs

En vue de réaliser un code compréhensible, maintenable, de respecter un court délai de création et de produire une application en cohérence avec l'actualité technologique, la réalisation du projet se découpera comme suit :

- La partie back API sera réalisée sous PHP grâce à l'utilisation du framework Symfony, l'ORM Doctrine, le composant Console et le bundle Maker afin de produire rapidement un code clair, documenté et automatiquement en relation avec la base de données.

- La base de données utilisera MySQL et sera encodée en utf8mb4.

- Le jeu de données json envoyé par l'api sera interprété grâce à l'utilisation du framework Angular basé sur TypeScript.

- La partie front sera complétée par l'utilisation de fichiers SCSS/CSS et le framework Ngx-Bootstrap

B. Environnement de travail

Le développement du projet se fera sous un environnement standardisé afin d'avoir une compatibilité optimale entre nos travaux.

L'équipe développera donc avec les outils suivants :

- Windows en OS
- L'IDE sera l'outil proposé par Jetbrains : PHPStorm

- GitHub Board en outil de gestion du projet et de la méthodologie agile

De plus, elle utilisera une méthodologie agile inspiré de SCRUM, c'est à dire :

- Un déploiement divisé en sprints d'une semaine composée de :
    - Quatre jours de production, et de bug fixing
    - Un jour de Sprint Review/Planification
- Un Daily Meeting tous les matins

Ces choix permettent de ne pas avoir de Scrum Master, ni de Product Owner, impliquant donc toute l'équipe dans la gestion du projet, et permettant d'avoir un point de vue plus reculé.

Tout ceci restant bien évidemment modulable, l'avantage de la méthodologie agile.

III. Ergonomie et graphisme

A. Structure du site

Le site entièrement responsive, se décompose en 8 parties : La page d'accueil, la page des marques, une page dynamique de marque, une page dynamique de produit, une page d'administration, une page client, une page panier et une page de recherche. Chaque page possèdera un header et un footer fixes facilitant la navigation entre les différentes parties.

- Le header permet d'accéder à l'accueil via le logo du site, à la page des marques, au panier, à l'espace client et à l'outil de recherche.

- Le footer permet d'accéder à l'accueil via le logo du site, et permet l'accès aux items standards (non livrables) des conditions générales, de la politique de protection des données, de la souscription à la newsletter, du formulaire

de contact, des coordonnées de l'entreprise et des liens vers les différents réseaux sociaux.
La page d'accueil permettra d'afficher les produits mis en avant, ainsi qu'une sélection de marques et/ou produits du moment, ainsi que des promotions en cours. La navigation est verticale, avec une bannière carrousel et une suite de sections vitrines.

- La page des marques permettra de naviguer entre les différentes marques représentées par des vignettes. L'organisation se fait de manière alphabétique et les marques se chargeront 8 par 8 à l'aide d'un bouton "Voir plus".

- La page d'une marque permettra d'accéder à tous les produits de la marque. Les produits seront représentés sous forme de vignette avec l'image du produit, son nom, son prix, et un bouton d'ajout au panier. En cliquant sur la vignette on pourra accéder à la page du produit. L'organisation se fait de manière alphabétique ou par ordre de prix (croissant - décroissant), ou par pourcentage de réduction (promotion en cours), et les produits se chargeront 10 par 10 à l'aide d'un bouton "Voir plus".

- La page d'un produit permettra d'accéder aux différentes spécificités du produit, ainsi que sa marque. En bas de page seront présentes des suggestions de produits de la même marque sous la même forme qu'à la page d'une marque.

- La page du panier permettra d'afficher tous les produits ajoutés au panier par l'utilisateur sous forme de liste. Il sera possible d'ajuster la quantité, de consulter le sous-total et d'accéder au bouton de paiement.

- La page de recherche permettra de lister les produits recherchés par l'utilisateur sous forme de vignettes identiques au format préalablement mentionné.

L'affichage se fera par ordre alphabétique et les produits se chargeront 10 par 10 à l'aide d'un bouton "Voir plus".

- La page de l'espace client permettra à l'utilisateur de consulter ses informations personnelles, de modifier son mot de passe, de se déconnecter, de gérer ses adresses et de consulter son historique de commandes.

- La page d'administration permettra à l'utilisateur de gérer la totalité du catalogue et la totalité des utilisateurs (clients comme administrateurs). La page se découpera donc en 4 menus : Clients, Administrateurs, Produits, et Modification du mot de passe (courant).
    - Le menu Clients permettra d'afficher les informations personnelles non sensibles du client (nom, prénom, mail, téléphone) ainsi qu'un bouton pour accéder à son historique d'achat.
    - Le menu Administrateurs permettra d'afficher la totalité des administrateurs enregistrés, d'en supprimer ou d'en ajouter.
    - Le menu Produits se décomposera en 3 sous-menus : Promotions, Produits, Marques. Ces 3 sous-menus permettrons d'accéder à la liste de tous les items relatifs. Sur cette page seront présents 3 boutons permettant de rechercher un item d'un sous-menu précis, et 3 boutons permettant d'ajouter un item d'un sous-menu précis.

B. Charte graphique

La charte graphique du site reposera sur le code couleur suivant :

	blanc	#FFFFFF
	noir	#000000
	bleu	#3366CC
	bleu	#88ABE6
	orange	#FFA264
	orange	#E6AE88

Le style des pages restera épuré afin de miser sur la clarté de l'information et la simplicité de navigation. Les couleurs majoritaires resteront donc blanc et noir et les items à mettre en valeur utiliseront le bleu et l'orange dont les nuances sélectionnées sont complémentaires.

Les polices utilisées pour le site seront :

Montserrat
Nunito

Ces polices sobres et supportées par la totalité des navigateurs permettront d'accentuer la clarté du rendu visée, et contribuera à la réduction des temps de chargement.

Toujours dans un souci de lisibilité mais aussi d'éco-responsabilité, le site utilisera peu d'images, et chaque produit ne sera lié qu'à une seule et unique image. Les images seront toutes compressées à l'aide du site tinypng & compressjpg.

Le header et le footer prendront chacun 30% de l'écran, les éléments resteront scrollables et non pas fixés, une page prendra toujours au moins 100% de l'écran.

IV. Sécurité

A. Base de données

La base de données utilisera le système MySQL, les tables seront encodées en utf8mb4_general_ci qui permet par exemple le stockage d'emojis contrairement à l'utf8 classique, et le type de stockage sera InnoDB afin de permettre l'utilisation de clés étrangères.

La base de données contiendra les tables suivantes :

- admins :
    - id : int, primary
    - login : varchar(255)
    - password : varchar(255)
    - token : varchar(255)
    - creation_date : date
    - iv : varchar(255)
    - key : varchar(255)
    - iv_time : varchar(255)
    - key_time : varchar(255)

  Cette table permet de stocker les différents administrateurs. Les colonnes iv, key, iv_time, et key_time permettent de générer un lien chiffré (et déchiffrable) sur la boîte mail d'administration du site, afin de permettre à un administrateur de se connecter.

- customers :
    - id : int, primary
    - mail : varchar(255)
    - lastname : varchar(100)
    - firstname : varchar(100)
    - phone : varchar(50)
    - password : varchar(255)
    - token : varchar(255)
    - creation_date : date
    - active : tinyint(1)

  Cette table permet de stocker les différents clients du site. La colonne active, ayant par défaut la valeur 0 permet de pallier au problème de subscription bombing, un compte ne devient actif qu'après avoir validé l'adresse mail via un lien dédié envoyé à l'utilisateur. Les comptes restés inactifs pourront alors être supprimés.

- addresses :
    - id : int, primary
    - id_customer : int, index FK customers.id

    - lastname : varchar(100)
    - firstname : varchar(100)
    - address_number : varchar(25)
    - address_street : varchar(150)
    - address_addition : varchar(150)
    - postal_code : int(10)
    - city : varchar(100)
    - main : tinyint(1)

  Cette table permet de stocker les adresses indépendamment d'un client afin de lui permettre des livraisons à différents lieux et/ou noms. Il sera également possible grâce à ce système d'implanter un système de commandes en tant qu'invité.

- orders
    - id : int, primary
    - id_customer : int, index FK customers.id
    - address_id : int, index FK addresses.id
    - creation_date : date
    - status : varchar(150)
    - tracking_number : varchar(50)

  Cette table permet de stocker les différentes commandes du site en liant un utilisateur à une adresse, et permet d'afficher le statut de celles-ci ainsi que l'éventuel numéro de suivi.

- order_details :
    - id : int, primary
    - id_order : int, index FK orders.id
    - id_product : int, index FK products.id
    - product_quantity : int
    - price : float
    - total_price : float

  Cette table permet de stocker les détails de chaque commande. Chaque entrée en base de données représente une référence du catalogue présente dans la

commande cible (par exemple, une commande comprenant 3 articles A, 2 articles B et 5 articles C aura 3 lignes dans cette table). Ce système permet par exemple d'enregistrer le prix d'un article à un instant donné dans le cas de promotions.

- ips
    - address : varchar(100), primary
    - id_customer : int, index FK customers.id
    - blacklist: tinyint(1)

  Cette table permet de stocker les différentes adresses IPs qui interagissent avec le site en se connectant. Il sera possible de blacklister manuellement ou automatiquement via le formulaire de connexion toute adresse IP suspecte (adresse IP changeant fréquemment de compte, bruteforce de connexion, pays à risque, etc).

- products
    - id : int, primary
    - id_brand : int, index FK brands.id
    - name : varchar(255)
    - description : varchar(500)
    - image_path : varchar(255)
    - price : float
    - stock : int
    - active : tinyint(1)

  Cette table permet de stocker les différents produits du catalogue. La colonne image_path contiendra l'url du fichier stocké localement après l'upload via le formulaire de création d'article. La colonne active permet d'activer ou de désactiver un produit.

- brands
    - id : int, primary
    - name : varchar(255)
    - description : varchar(500)
    - active : tinyint(1)

Cette table permet de stocker les différentes marques du catalogue, et de les activer ou désactiver via la colonne active. Une marque désactivée permettra de ne pas afficher tous les produits de cette marque.

- discounts
    - id : int, primary
    - product_id : int, index FK products.id
    - name : varchar(255)
    - percentage : float
    - start_date : date
    - end_date : date

  Cette table permet de stocker les différentes promotions du site. Une promotion affecte un ou plusieurs produits dépendamment de l'action effectuée sur la page d'administration. Elle n'est effective que lorsque la date actuelle est incluse entre la date de début et la date de fin de promotion.

- register
    - id : int, primary
    - id_customer : int, index FK customers.id
    - openssl_iv : varchar(550)
    - openssl_key : varchar(550)
- connect
    - id : int, primary
    - id_customer : int, index FK customers.id
    - openssl_iv : varchar(550)
    - openssl_key : varchar(550)
- time
    - id : int, primary
    - id_customer : int, index FK customers.id
    - openssl_iv : varchar(550)
    - openssl_key : varchar(550)

  Ces trois tables permettent de stocker les clés de chiffrage et donc de déchiffrage des mails envoyés aux utilisateurs

pour les authentifier lorsque nécessaire (création d'un compte, connexion depuis un nouvel appareil, adresse IP affectée à un autre compte).

B. Sécurité des données

Aucune donnée sensible (en dehors de la combinaison login / password) n'est stockée en base de données puisqu'aucune donnée sensible n'est demandée lors de la navigation.

Les mots de passe sont hashés de manière classique grâce à Symfony avant d'être stockés en base de données. Le typage des colonnes des différentes tables est bien défini et limité afin d'éviter toute entrée non désirée

Enfin, toute insertion en base de données fera l'objet d'une sanitisation par le biais de la fonction php htmlentities, encoding UTF-8 et flag ENT_QUOTES, évitant la majeure partie des injections SQL et XSS.

C. Validation des données

Tous les champs de toutes les entités seront automatiquement vérifiés et validés grâce aux constraints de Symfony via l'utilisation de la fonction Assert. La validation du typage via api-platform ne sera pas privilégiée car les messages d'erreur ne seront pas toujours explicites ou risquent une exposition sql.

D. Authentification

L'authentification s'effectuera également via Symfony, et l'utilisateur se verra donc définir un rôle adéquat ainsi qu'un jwt.

E. Cookies

F. Stripe