**MARKETPLACE S2**

**VISITEUR AUCUN RÔLE**

1. L’inscription

a) Input prénom

-   Regex nombre de caractère >= 3

-   Afficher erreur de validation en dessous de l’input en rouge

b) Input nom

-   Regex nombre de caractère >= 3

-   Afficher erreur de validation en dessous de l’input en rouge

c) Input adresse mail

-   Regex adresse mail valide

-   Afficher erreur de validation en dessous de l’input en rouge

d) Input mot de passe

-   Regex nombre de caractère >= 6

-   Afficher erreur de validation en dessous de l’input en rouge

c) Input mot de passe de confirmation

-   Regex nombre de caractère >= 6

-   Condition, mot de passe === mot de passe de confirmation

-   Afficher erreur de validation en dessous de l’input en rouge

e) Lien déjà un compte ? Redirection vers la page de connexion

f)  Bouton s’enregistrer

-   Vérifier que l’ensemble des champs sont valide avant de tester la création

-   Si le compte est créé, afficher un toaster (L’inscription a été validée)

-   Si le compte n’est pas créé, afficher (Une erreur est survenue lors de l’inscription)

2. La connexion

a) Input adresse mail

-   Regex adresse mail valide

-   Afficher erreur de validation en dessous de l’input en rouge

b) Input mot de passe

-   Regex nombre de caractère >= 6

-   Afficher erreur de validation en dessous de l’input en rouge

d) Lien mot de passe oublié ? Redirection vers la page de réinitialisation

e) Lien pas encore de compte ? Redirection vers la page d’inscription

f)  Bouton de connexion

-   Vérifier que l’ensemble des champs sont valide avant de tester la connexion

-   Si l’utilisateur se connecte, rediriger vers la page d’accueil

-   Si l’utilisateur n’arrive pas à se connecter, afficher un toaster (Une erreur est survenue lors de la connexion)

3. Visualiser l’ensemble des produits

-   L’utilisateur doit pouvoir visualiser tous les produits du site

-   L’ensemble des produits doivent être filtrées côté front via des filtres par l’utilisateur si besoin

-   Afficher l’ensemble des produits par le plus récent

-   Avoir une pagination pour l’ensemble des produits

4. Visualiser un produit

-   Quand je clique sur un produit au hasard, je suis redirigé vers sa page détail

-   Dans la page détail, on visualise l’ensemble des données du produit qu’on aura définit dans la table

5. Ajouter des produits au panier

-   On peut ajouter le produit au panier via sa page de détail

-   Le panier doit être mis à jour après chaque ajout, ce qui veut dire afficher le nombre de produits via un badge qui représente la somme des produits dans le panier présent (style Amazon)

-   Si jamais le visiteur essaie d’acheter les produits du panier alors on lui affichera un bouton, devenir membre afin de continuer le processus.

6. Supprimer des produits du panier

-   Un visiteur peut ajouter des produits au panier sans les acheter forcément mais il peut aussi les supprimer pour en acheter d’autres par exemple

7. Demander à être vendeur

-   Pour devenir vendeur, il faut se rendre dans le footer et cliquer sur le bouton “devenir vendeur”. Cela redirige vers une page POST “seller” ou un formulaire adapté avec les informations qu’on retrouve dans la table (voir schéma).

-   Cela envoie une demande à l’administrateur qui doit valider par la suite. Cela déclenche un envoi de mail à l’administrateur.

-   Si on essaie de se connecter via un compte vendeur et que ça ne marche pas, il faut afficher la raison via un toaster. Soit (Le compte est en cours de validation) si c’est le cas ou lors d' une erreur quelconque (Une erreur est survenue lors de la connexion).

**UTILISATEUR ROLE\_USER**

1. Accéder à la rubrique mon compte

-   Dans le header et dans la navbar, il faut pouvoir cliquer sur le lien qui redirigera l’utilisateur dans la vue de son compte avec ses différentes informations utilisées lors de la création + tout ce qui va concerner (adresse, transporteur, etc.)

1. Modifier ses informations personnelles (à définir)

-   Comme précédemment, dès que l’utilisateur est sur la vue de son profil, il peut donc visualiser les différentes informations de son compte telles que celle utilisée pour enregistrer et les nouvelles (adresses, transporteur par défaut, etc. ...)

-   Vérifier l’ensemble des champs via des regex et validations avant de les mettre à jour (vérification selon leur type)

-   Si les informations sont mises à jour, afficher un toaster (Le compte a bien été mis à jour)

-   Si une erreur survient lors de la mise à jour, afficher un toaster (Une erreur est survenue lors de la mise à jour du compte)

1. Ajouter des adresses

-   Un utilisateur peut ajouter des adresses, pour cela directement sur le profil de l’utilisateur au moment où il visualise l’ensemble de ses données pouvoir créer des données via un modal par exemple ou alors ajouter un bloc d’input pour les adresses

-   Regex sur l’ensemble des propriétés d’une adresse (rue, code postal, pays, etc. …)

4. Visualiser l’ensemble des produits

-   L’utilisateur doit pouvoir visualiser tous les produits du site

-   L’ensemble des produits doivent être filtrées côté front via des filtres par l’utilisateur si besoin

-   Afficher l’ensemble des produits par le plus récent

-   Avoir une pagination pour l’ensemble des produits

1. Ajouter des produits au panier

-   On peut ajouter le produit au panier via sa page de détail

-   Le panier doit être mis à jour après chaque ajout, ce qui veut dire afficher le nombre de produits via un badge qui représente la somme des produits dans le panier présent (style Amazon)

-   Si jamais le client essaie d’acheter les produits, le rediriger vers le paiement

1. Supprimer des produits du panier

-   Un utilisateur peut ajouter des produits au panier sans les acheter forcément mais il peut aussi les supprimer pour en acheter d’autres par exemple

1. Acheter des produits

-   L’achat de produit se fait donc via un bouton dans la vue du panier

-   Un utilisateur ne peut acheter quelque chose seulement si le panier n’est pas vide

-   Ne pas afficher le bouton si le panier est vide ou alors le rendre disabled

-   Si tout le formulaire pour l’ensemble de l’achat est valide alors on peut passer au paiement via stripe

-   Regex sur l’ensemble des champs afin de bien vérifier que les données sont valides

-   Quand l’achat est validé, la commande sera en pending ce qui veut dire qu’elle sera en cours de préparation + étape de livraison et c’est à l’admin de la valider depuis son tableau de bord et ensuite notifier le client par mail qu’il l’a bien reçu à tel ou tel adresse

1. Consulter l’historique de ses commandes

-   En tant qu’utilisateur, je me dois d’accéder à l’ensemble de mes commandes soit terminées ou en cours. Avoir un lien côté sidebar qui permettra la redirection et la visualisation des commandes quand je clique dessus.

-   Imaginer un datatable avec un tableau des commandes avec les statuts etc.

1. Demander un retour pour ses produit et commandes

-   Quand une commande a été validée par l’admin alors l’utilisateur peut choisir de renvoyer un ou des produits pour la raison qu’on lui demandera de spécifier

-   La demande de retour doit être validée côté admin

-   Si la demande de retour est validée ou refusée par l’administrateur alors notifier l’utilisateur par mail afin qu’il soit au courant

-   Selon le statut de la demande de retour lors du mail qu’on envoie à l’utilisateur pensait à un mail style amazon

-   Visualiser l’ensemble des demandes de retour et il ne faut pouvoir renvoyer un produit déjà renvoyé d’une commande

**VENDEUR ROLE\_SELLER**

1. Implémenter un CRUD pour les produits

-   Un vendeur doit pouvoir avoir tout le workflow que fournit un CRUD

-   Ajouter un produit

-   Visualiser les produits

-   Modifier les produits

-   Supprimer les produits

2. Consulter la liste de ses produits achetés

-   En tant que vendeur, implémenter une vue ou il peut visualiser l’ensemble de ses produits déjà achetés par un client, son statut, etc.

-   Les produits seront surement éparpillés sur différentes commandes ou les produits ne seront pas seulement au vendeur, alors afficher les produits et les informations auquel ils sont liés

3. Consulter la liste de ses produits retournés

-   Visualiser l’ensemble des produits retournés par les acheteurs avec les informations nécessaires tel que le prix, nom, motif, etc.

4. Consulter son profil

-   En tant que vendeur, une page profil lui aussi dédié ou il retrouvera l’ensemble de ses informations modifiable

-   Comme pour l’ensemble des utilisateurs, avant chaque mis à jour, il faut vérifier la validation des inputs et notifier l’utilisateur via un toaster selon le résultat final

-   L’ensemble des informations seront à définir comme il faut lors de la mise en place de l’architecture

**ADMINISTRATEUR ROLE\_ADMIN**

1. Implémenter un CRUD pour les produits

-   Un administrateur doit pouvoir visualiser l’ensemble des produits du site internet avec toutes les informations possibles que les précédentes mais aussi qui l’a créé, quand, etc.

-   Ajouter un produit

-   Visualiser les produits

-   Modifier les produits

-   Supprimer les produits

2. Implémenter un CRUD pour les transporteurs

-   En tant qu’administrateur, il faudrait comme pour l’ensemble des autres vues qui lui sont propre implémenter une vue ou il visualise l’ensemble des transporteurs disponibles, etc.

-   Ajouter un transporteur au site

-   Visualiser les transporteurs

-   Modifier un transporteur

-   Supprimer un transporteur

3. Gérer les demandes de retour

-   Implémenter une vue avec les différents retours et les utilisateurs qui lui sont associés

-   Une validation ou annulation d’un retour d’un client doit lui être notifié par adresse électronique quoi qu’il arrive afin qu’il connaisse le suivit de sa commande même si dans son interface cela lui sera fourni

-   Implémenter donc un datatable avec tous les retours en cours, les statuts, les utilisateurs, etc.

-   Accepter ou refuser seront les deux choix sur cette interface en plus de la visualisation des détails

4. Gérer les demandes pour devenir vendeur

-   Comme pour précédemment, un administrateur doit visualiser la liste des personnes qui souhaitent devenir seller

-   Dans cette interface de visualisation, il pourra accepter ou alors refuser la demande

-   Selon le choix il faudra notifier la personne en question si la demande a été accepté ou refusé et pour qu’elle raison

5. Implémenter un CRUD pour les utilisateurs

-   Comme pour le CRUD sur les produits, l’admin doit pouvoir visualiser les différents clients du site et les gérer facilement

-   Ajouter un utilisateur

-   Visualiser les utilisateurs

-   Modifier un utilisateur

-   Supprimer un utilisateur

6. Gérer les commandes d’un utilisateur

-   L’admin doit pouvoir afficher l’ensemble des commandes payés par les utilisateurs et les statuts qui leurs sont associés (payés, en cours de validation, en transit, livrée) voilà une petite idée mais à améliorer si besoin

-   Il lui suffira de valider la commande afin qu’elle puisse lui être livrée et du coup notifier l’utilisateur ou alors refuser la commande et du coup le rembourser etc.

-   La partie de l’admin qui concerne un utilisateur doit être le plus notifié possible afin qu’il soit au courant de ce qui se passe et avoir une expérience facile

**RÉCAPITULATIF**

-   Ne pas oublier les notifications par mail quand cela est important pour les workflow qui sont important

-   Ne pas oublier les toaster au lieu de faire des catch erreur côté front

-   Vérifier le code avant de commit n’importe quoi et laisser des logs. Vider tous les console.log avant de push.

-   Gérer les messages correctement de retour côté backend afin de les gérer côté frontend facilement.

-    Throw des exception côté back pour avoir le bon format de réponse côté client.

-  Suivre le format rest et le même format de retour d’erreurs côté back.

-   Pour chaque micro-service, il faut implémenter 2 test unitaire et 1 test fonctionnel minimum (à la fin)

-   Ne pas push une branche sur la develop si les commit sont pas signés

-   Ne pas push une branche sur la develop si elle est pas review par une personne localement

- Documenter le plus de choses possible pour une meilleure compréhension (GATEWAY, MÉTHODES, ETC …)