

Rapport de projet

Vinci Store

1. Evaluation du résultat par rapport au planning des tâches et des cas d'utilisation

Veillez décrire si vous avez pu atteindre vos objectifs fonctionnels et autres tâches. Si vous avez dévié des objectifs fonctionnels identifiés initialement, veuillez décrire les raisons de ce changement.

Ce paragraphe est à rédiger sur environ 15 lignes.

Nous avons réussi à implémenter la grande majorité de nos objectifs ce qui rend notre application fonctionnelle. Mais certains n'ont pas pu être atteints, celles-ci faisaient toutes office de "bonus" et n'empêche pas donc pas l'utilisateur d'utiliser "VinciStore" comme nous l'avions imaginé en nous lançant dans ce projet.

Premièrement, les statistiques.

Nous n'avons en effet pas pu implémenter cette page pour manque de données concernant les performances des vendeurs.

La manière dont nous avons architecturé notre base de données a beaucoup évolué tout au long du projet, ce qui nous a donc éloigné de cet objectif qui n'était pas une priorité à nos yeux.

Deuxièmement, les paiements en cryptomonnaie.

Ne voulant pas passer par des plateformes tierces dont l'implémentation demandait des connaissances qui nous auraient demandé trop de temps pour pouvoir être assimiler et par conséquent, freiner dans l'avancement de tâches prioritaires, nous avons décidés de ne pas implémenter ce mode de paiement.

De plus, ces plateformes tierces prennent une commission sur les transactions, ce qui enlève tout l'intérêt que nous avons à implémenter ce type de paiements.

Et pour finir, nous voulions aussi déployer un espace commentaires et réponses. L'espace commentaires et fonctionnel seules les réponses ne le sont pas.

2. Audit ergonomique de votre projet

Veuillez utiliser les outils vus en cours pour analyser l'ergonomie de votre projet et si les règles de GDPR sont respectées. Créer ensuite ci-dessous un rapport à rédiger d'environ 10 lignes.

Nous sauvegardons l'email, le sexe, le nom, le prénom et si l'utilisateur est un vendeur, nous sauvegardons également son adresse.

Ces données étant nécessaire au bon fonctionnement du site, l'utilisateur s'engage en s'inscrivant à respecter nos conditions d'utilisations.

La case pour accepter nos conditions n'est pas pré cochée et si l'utilisateur ne les acceptent pas, il ne sera pas inscrit sur le site.

Les conditions d'utilisations sont bien au nom de VinciStore et ne sont plus valables en dehors de notre application.

Si l'utilisateur le désire, il pourra facilement supprimer définitivement son compte à n'importe quel moment sur sa page personnelle.

3. Difficultés techniques rencontrées

Indiquez ici les difficultés techniques que vous auriez rencontrées au cours de votre projet.

Ce paragraphe est à rédiger sur environ 15 lignes.

Après discussion avec tous les membres du groupe, nous sommes arrivés à une conclusion commune :

Premièrement, la gestion des données a été, au départ, une vraie source de problèmes.

En effet, l'architecture pour le transit de nos données à travers l'application était tout à fait nouvelle pour nous et nous a demandé un certain temps d'adaptation.

Le transit de nos données à partir de la DB vers l'API n'a pas posé de problèmes.

Cependant, le transfert des données de l'API vers le frontend nous a, en revanche, donné du fil à retordre. L'utilisation du fetch et du JWTToken nous étant nouvelle, nous avons eu du mal à exploiter les données renvoyées.

Secondement, l'affichage dynamique de l'HTML, notamment pour celui des objets ayant un style récurrent mais des données différentes.

Par exemple, l'injection de script avec "innerHTML" ne déclenche pas ceux-ci, il a donc fallu trouver un moyen de contourner cette contrainte. Une des solutions a été

de générer un script avec “createElement” et de lui fournir la source voulue pour ensuite le “appendChild”.

Dernièrement, faire la liaison entre notre frontend, la librairie Paypal (frontend) et son API n'a pas été facile. Il a fallu comprendre l'outil de développement PaypalDeveloper et ses subtilités qui sont parfois mal documentées pour la manière dont nous voulions l'implémenter (SPA et html dynamique).

4. Conseils pour appliquer cette technologie

Veuillez résumer tout ce qui serait intéressant pour quelqu'un qui souhaiterait appliquer le même genre de technologie, sur base de ces questions :

Qu'est-ce que vous auriez aimé savoir avant de démarrer ?

Des conseils pour éviter certains problèmes rencontrés ?

Des liens vers les sites les plus intéressants ?

Etc.

Ce paragraphe est à rédiger sur environ 15 lignes.

ElephantSQL & PostgreSQL (backend) : Son grand avantage est d'être une base de données gratuite assez facile à intégrer, en effet, il suffit de créer un fichier “.env” à la root de votre API et y coller 5 lignes fournies par la documentation.

Par contre, le nombre de connexions simultanées dans le plan gratuit étant de 5.

Nous devons souvent fermer notre API pour permettre aux autres membres de notre groupe de pouvoir l'utiliser.

Paypal-JS (frontend) : Paypal est assez facile à implémenter, mais l'affichage du bouton se fait à l'aide d'un script qui doit se trouver dans l'html et comme mentionné au point 3, les scripts ne sont pas lus avec “innerHTML”, il faut donc trouver un moyen de contourner cette contrainte.

ChartJS (frontend => non présent dans version finale) : Cette librairie requiert une quantité minimale de données et nous pensons donc qu'il faut l'implémenter en fin de projet.

Attention, lier l'API et ChartJS n'est pas facile, toutes les architectures n'étant pas représentées dans la doc.

Cela paraît évident mais le meilleur conseil reste de se référer au cours de Web2, cela nous a permis de résoudre un certain nombre de problèmes sans soucis.

Sources :

<https://www.elephantsql.com/docs/index.html>

<https://www.chartjs.org/>

<https://developer.paypal.com/>

<https://e-vinci.github.io/web2/>

5. Quels sont les points positifs à la manière dont s'est déroulée la collaboration au sein du groupe ?

Veillez donner des arguments justifiant votre réponse. Cette réponse est à rédiger sur environ 10 lignes.

-Une fois que l'architecture de base est en place, tout le monde a réussi à plus ou moins trouver sa place et c'est à partir de ce moment que tout le monde à commencer à travailler sérieusement.

-La progression du projet a été exponentielle à partir du moment où tous les membres du groupe ont atteint un niveau de connaissances techniques qui leurs ont permis de devenir autonome dans leur travail.

-Nous avons bien communiqué durant tout le projet, et chacun a pu travailler sur la partie qu'il souhaitait implémenter.

-Nous nous sommes tous rendu disponible pour nous entraider et résoudre les problèmes de que chacun pouvait rencontrer.

Ce qui a permis de lier nos connaissances et de progresser plus rapidement.

6. Quels sont les points qui seraient à améliorer pour de futures collaborations ?

Les points à améliorer seraient les suivants :

-Commencer à travailler dès le premier jour du projet.

-Communiquer AVEC CALME afin de garantir un esprit de travail de meilleure qualité.

-Se mettre d'accord sur les objectifs du projet et donc être rigoureux dans la méthodologie (ex : Scrum, Agile, etc....) choisie.

-Prendre connaissances des avancements du projet avant de commencer à implémenter de nouvelles fonctionnalités.

-Mieux apprendre à utiliser le débogueur. Le temps pris pour résoudre les bugs est largement réduis quand l'utilisation de celui-ci est maitrisée.