

Exercice Frontend SpaceX

Objectifs :

- Évaluer les compétences techniques en développement VueJS et Typescript
- Évaluer les capacités à utiliser des systèmes inconnus à partir de leur documentation
- Évaluer les capacités de travail en Autonomie
- Évaluer les capacités à organiser son temps

Ce projet consiste à réaliser une application VueJS simple permettant de présenter des lancements SpaceX.

Il doit utiliser les technologies suivantes :

- VueJS 3 avec Composition API
 - soit en installant [vite.js](https://vitejs.dev/), en utilisant des fichier .Vue (npm init vite@latest my-vue-app----- template vue)
- Typescript
- API SpaceX v5
 - documentation disponible ici : [SpaceX-API](https://api.spacex.com/)
 - utilisation en mode public (sans authentification)
- fonction [Fetch](#) du navigateur ou [Axios](#)
- async/await
- mise en page : Tailwind CSS
- animations : au choix du candidat
- [Youtube embed](#)

La page devra présenter :

- Un titre principal
- Une section affichant "Prochain lancement", la date du prochain lancement (launch), son nom (données chargées depuis l'API SpaceX), un décompte en seconde mis à jour en temps réel
- Un <select> avec 3 choix permettant de filtrer les lancements
 - Tous les lancements
 - Lancements réussis

- Lancements échoués
- Une section listant les 10 derniers lancements en fonction du choix sélectionné
- Lorsqu'on clique sur un lancement, il doit être présenté dans un modal précisant :
 - son nom
 - sa date au format Jour/Mois/Année
 - le texte de détail de la mission (en anglais)
 - une image d'illustration de la mission
 - un lien vers un article de présentation
 - un switch/checkbox permettant d'afficher la vidéo YouTube de la mission dans un composant masqué par défaut juste en dessous du switch/checkbox (selon le composant choisi)
 - nom du lieu de lancement
 - noms des chargements envoyés (payloads)
 - noms des clients qui ont envoyé un chargement dans ce lancement
 - Toutes les données affichées doivent avoir été chargées depuis l'API SpaceX

NB :

Un soin particulier accordé au design (qui ne doit pas nécessairement être trop élaboré) serait apprécié.

Résultat :

- Code source : Un dépôt Git public contenant tout le code source.
- Rapport : Remarques, difficultés rencontrées, explication des choix techniques, ressources utilisées etc
- Guide de déploiement : Instructions détaillées pour déployer l'application (optionnel) Bon

Questions et rendu :

Emile Junior ADA
emilejuniorada@yahoo.com