Son credo ? Se démarquer des grands site e-commerce comme Amazon en créant des applications thématiques ne vendant qu’un seul groupe de produits. Il y a par exemple Oribook pour les livres ou Oritextil pour les vêtements.

Vos compétences en développement web et votre personnalité ont plu à Paul, le fondateur de l’entreprise.

Dans un premier temps, Paul souhaite créer un premier MVP pour démontrer le fonctionnement de ses applications à ses investisseurs.

L’équipe est constituée de Jeanne, développeuse back-end travaillant sur les API et vous, pour la partie front-end.

Définition. Une **API est** un ensemble de définitions et de protocoles qui facilite la création et l'intégration de logiciels d'applications. **API est** un acronyme anglais qui signifie « Application Programming Interface », que l'on traduit par interface de programmation d'application.

À votre arrivée, vous trouvez un mail de Paul dans votre boîte mail.

**Objet : Bienvenue !**

Bonjour et bienvenue chez Orinoco ! J’espère que tu te plairas parmi nous. J’ai vraiment hâte de montrer ce MVP aux investisseurs. Il faut qu’on les rassure car ils misent sur nous.



Jeanne s’occupera du back-end et toi du front.

Avec l’équipe, on a déjà pas mal bossé sur le cahier du MVP ( Minimum Viable Product - Produit Minimum Viable) version d'un produit qui permet d'obtenir un maximum de retours client avec un minimum d'effort. Cela te fera une bonne base pour avancer. Tu trouveras en pièce jointe les spécifications.

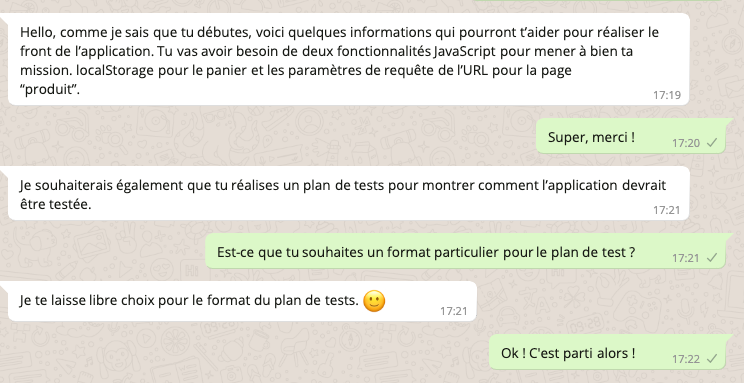
N'hésite pas à venir me voir si tu as la moindre question, ma porte est toujours ouverte.

Paul

Fondateur de Orinoco

Paul vous envoie également des messages sur le système de messagerie instantanée de l’entreprise pour vous apporter quelques précisions complémentaires.

La conversation instantanée entre Paul et vous



Vous gardez en tête toutes ces informations et attaquez le développement immédiatement.

**Livrables**

* Le**lien** vers un dépôt Git public contenant le code de l'application web.
  + Le mentor et l'évaluateur doivent être en mesure de cloner le référentiel, d'ouvrir index.html, d'utiliser l'application entièrement opérationnelle et de confirmer que n’importe quel input utilisateur est validé.
* Un **plan de tests.**

Pour faciliter votre passage au jury, déposez sur la plateforme, dans un dossier nommé “*P5\_nom\_prenom*”, tous les livrables du projet. Chaque livrable doit être nommé avec le numéro du projet et selon l'ordre dans lequel il apparaît, par exemple “*P5\_01\_lienGithub*”, “*P5\_02\_plantests*”, et ainsi de suite.

Une question sur ce projet et votre mentor n’est pas disponible ? Rendez-vous sur le post Workplace dédié aux questions sur ce projet. **Le lien est disponible dans l’onglet Ressources du projet**. Un mentor est disponible pour répondre à vos questions. Une fois par semaine, ce même mentor organisera une session live d'une heure sur Workplace pour répondre à vos questions en direct.

**Soutenance**

La soutenance, d’une durée maximum de 30 minutes, se déroulera en deux parties :

**Partie 1 – 20 minutes : Simulation d'une réunion professionnelle.**

*L’évaluateur jouera le rôle de Paul, le fondateur d’Orinoco (il pourra vous interrompre pour vous poser des questions).*

* + [~ 20 minutes] Vous réaliserez une présentation de votre application.

**Partie 2 – 10 minutes : Retour sur la soutenance**

* + Échange de questions/réponses : l’évaluateur pourra revenir sur certains points pour vous questionner sur vos choix.
  + L’évaluateur vous fera un retour sur votre prestation en soutenance.

Vous vous demandez par où commencer ? Suivez nos conseils dans cette video pour vous organiser et faire le point avant votre première session de mentorat !

Projet a deux concepts clés principaux :

1. Javascript

**Implémenter le comportement du site**

* Gérer les évènements ( exemples : ajout d’un élément dans le panier…)
* Implémenter un formulaire de contact,
* Communiquer avec le serveur via une API

1. Les API ( s’utilisent à travers du code JavaScript)

Gérer les items à acheter sur le site

Conseil : manipuler d’abord l’API seule avant de l’intégrer dans le code JS

* + - Dédier les séances de mentorat à la prise en main de l’API et du JS

**Les étapes :**

* + 1. **Faites un bechmarck des sites de e-commerce**
    - Comparer les sites e-commerce
    - S’inspirer des sites de e-commerce pour voir les fonctionnalités
    - Analyse des sites déjà existants pour mieux se projeter dans le travail à réaliser
    1. **Découper le travail en tâches pour bien s’organiser**
    - Ne pas se lancer directement dans le développement
    - Identifier d’abord les fonctionnalités de l’application ( ce qu’elle est sensé faire afin de couper le travail en taches pour planifier les taches du développement).

Exemple : chaque page du site peut représenter deux taches ( une pour récupérer les données de l’API et l’autre pour afficher les données sur la page)

* + - Pour planifier les tâches, on peut utiliser des outils de gestion de projets comme ( otableaucombon avec trilo) ou rédiger des ichou sur github
    1. **Structurer les développements en identifiant les fonctions à coder**
    - Découper le code en fonction avant de l’écrire
    - Chaque fonction doit avoir un objectif clairement identifié et unique
    - Eviter les fonctions qui font plusieurs choses à la fois pour avoir un code claire et modulaire plus facile à maintenir et surtout établir le plan de test facilement par la suite sans avoir à revenir sur le code
    1. **Découvrir et tester l’API du projet**
    - Pour utiliser l’API (commencer par lire sa documentation, l’installer, la tester, comprendre son fonctionnement avant de l’utiliser dans le projet)

**Voir l’outil swagger petstore**

* + 1. **Rédiger le code**
    - Privilégier un code de qualité qui respecte les conventions de nommage, l’indentation, les commentaires… et qui utilise les apports de S6 ( exemples : les classes, les fonctions fléchées…)
    - Faire des commits réguliers afin d’avoir un suivi étape par étape de l’évolution du code
* **Compétences évaluées**
*  Interagir avec un web service avec JavaScript
*  Valider des données issues de sources externes
*  Créer un plan de test pour une application
*  Gérer des événements JavaScript