

Débuggez et testez un SaaS RH

Vous êtes développeur front-end chez *Billed*, une entreprise qui produit des solutions SaaS destinées aux équipes de ressources humaines.

Malheureusement pour *Billed*, Garance, une collègue de la *feature team* “note de frais” a quitté l’entreprise avant d’avoir terminé l’application. Dans une semaine, l’équipe doit montrer la solution qui fonctionne à l’ensemble de l’entreprise. Matthieu, Lead développeur de la *feature team* a demandé à être épaulé pour tenir les délais et vous avez appris hier lors de la réunion d’équipe que c’est vous qui avez été désigné !

The screenshot shows a web application interface for sending expense reports. On the left is a blue sidebar with the 'Billed' logo, a grid icon, an envelope icon, and a power icon. The main content area is titled 'Envoyer une note de frais'. The form contains the following fields:

- Type de dépense:** A dropdown menu with 'Transports' selected.
- Nom de la dépense:** A text input field containing 'Vol Paris Londres'.
- Date:** A text input field with a date picker icon, containing 'jj/mm/aaaa'.
- Montant TTC:** A text input field containing '348'.
- TVA:** Two text input fields, the first containing '70' and the second containing '20'.
- Commentaire:** A large text area for notes.
- Justificatif:** A section with a 'Choisir un fichier' button and the text 'Aucun fichier choisi'.
- Envoyer:** A blue button at the bottom left of the form.

À votre arrivée ce matin, vous avez reçu un e-mail de la part de Matthieu, qui donne plus de détails sur ce qui est attendu de vous.

Objet : Urgent - Informations sur la mission de renfort au sein de la *feature team* “note de frais”

De : Matthieu

À : Moi

Tout d’abord merci de nous prêter main-forte cette semaine pour la mise en place de tests sur la fonctionnalité “note de frais”.

Cette fonctionnalité est très attendue sur le marché et le top management a mis la priorité dessus. L’objectif est de la lancer officiellement auprès de nos clients d’ici 2

semaines. Les délais sont donc très serrés. La *feature team* a beaucoup travaillé ces deux dernières semaines mais le départ de Garance n'arrange pas les choses, et nous avons besoin de ton aide pour la dernière ligne droite.

Voici la description de la [fonctionnalité](#).

L'essentiel a déjà été développé je te rassure. Tu trouveras [le code ici](#). Il y a deux parcours utilisateurs sur cette fonctionnalité : côté administrateur RH et côté employé. Le back-end est prêt (en version alpha). Côté front-end, le *parcours administrateur RH* est complet, entièrement testé et débuggé.

Ce qu'il reste à faire : fiabiliser et améliorer le *parcours employé*. Voici ce que j'attends de toi :

- **[Bug - report]**
Fixer les bugs identifiés dans le rapport de bug fourni par Jest. J'en ai mis une copie dans le [kanban Notion](#).
- **[Bug - hunt]**
Fixer les bugs identifiés par Leila notre QA sur le *parcours employé*. Ils sont décrits dans le kanban également.
- **[Tests unitaires et d'intégration]**
Ajouter des tests unitaires et d'intégration pour les fichiers Bills et NewBill : ces tests vont permettre d'éliminer ces bugs et éviter toute régression lors des prochaines évolutions de la solution. Certains tests sont déjà développés (pour le Login, et pour le Dashboard des administrateurs RH) : ils sont déjà cochés sur le kanban, tu peux t'en inspirer pour les restants. Comme la démo est seulement dans une semaine, nous n'avons pas le droit à l'erreur. Pour cette raison, il faut que tu assures une couverture de branche de 100 % (en dehors des appels au back-end firebase : ces derniers sont indiqués en commentaire dans le code). Tu peux t'appuyer sur le rapport de couverture de Jest (lance l'application avec live-server pour pouvoir le lire et va à l'adresse <http://127.0.0.1:8080/coverage/lcov-report/> : tout est indiqué dans le [readme](#)).
- **[Test E2E]**
Puisque l'on n'a pas le temps d'automatiser des tests End-to-End, les tests seront effectués manuellement par Leila. Il faut donc que tu lui fasses un plan de test End-to-End pour la guider. Tu peux t'inspirer du [plan End-to-End](#) que Garance a déjà rédigé sur le *parcours administrateur RH*.

Autre chose : l'application contient déjà des données de test, mais si tu veux en créer de nouvelles sur le parcours employé et pouvoir les consulter sur le parcours administrateur RH, il faudra que tu utilises ton e-mail perso pour te connecter sur les deux parcours. C'est parce que le code côté back end est encore en bêta et on a dû mettre ça en place pour faciliter les tests.

Voilà, bon courage pour résoudre ces bugs et mettre en place les tests manquants ! On compte sur toi.

Matthieu

Lead Developer @Billed

Ça y est, vous avez toutes les informations pour démarrer la correction de cette application. C'est parti !

Livrables

- Lien vers la codebase à jour sur un repo GitHub public.
- Un screenshot avec le rapport de tests Jest sur l'ensemble des fichiers d'UI (*src/views*) et des fichiers d'UX (*src/containers*).
- Un screenshot du rapport de couverture *Jest*.
- Un document au format PDF du plan E2E pour le *parcours employé*.

Le repo GitHub doit être nommé avec la convention suivante :

Nomcomplet_#_Datedémarrage. Le # correspond au numéro du projet sur le parcours et la date doit être au format jjmmaaaa. Par exemple, FrancoisLenotre_5_05032020.

Soutenance

Durant la présentation orale, votre mentor jouera le rôle de Matthieu le lead développeur du projet avec qui vous faites une code review :

•Présentation des livrables (15 minutes)

- Vous commencerez par présenter le rapport de test *Jest* pour montrer les tests que vous avez ajoutés, ainsi que le rapport de couverture actualisé. (5 minutes)
- Ensuite vous aurez 10 minutes pour expliquer comment vous avez corrigé chaque bug et comment vous avez rédigé les tests unitaires et d'intégration sur le parcours Employé.
- Enfin, vous pourrez rapidement illustrer le plan de tests End-to-End que vous avez rédigé sur le *parcours employé*.

•Discussion (10 min)

- Pendant 5 à 10 minutes votre évaluateur vous questionnera sur votre compréhension des bugs, et vos choix dans la manière de rédiger les tests unitaires, d'intégration et End-To-End.

Votre évaluateur questionnera vos choix, soyez donc prêt à défendre votre travail. À la fin de la session, votre évaluateur reprendra son rôle de mentor pour que vous puissiez débriefer ensemble.

Définissez les besoins pour une app de soutien scolaire

Cours :

<https://openclassrooms.com/fr/courses/4507926-initiez-vous-a-la-gestion-de-projet-agile>

<https://openclassrooms.com/fr/courses/4192086-gerez-votre-projet-informatique-facilement>

<https://openclassrooms.com/fr/courses/6739646-realisez-un-cahier-des-charges-fonctionnel>

<https://openclassrooms.com/fr/courses/5249006-concevez-une-interface-cliquable>

<https://openclassrooms.com/fr/courses/5647281-appliquez-le-principe-du-domain-driven-design-a-votre-application>



Vous êtes employé depuis 5 ans chez Dev4U, une entreprise de services du numérique de 120 collaborateurs comportant deux entités de développement logiciel :

- une entité spécialisée dans le développement d'applications mobiles, qui comprend 10 personnes ;
- une dédiée au Web, qui comprend 8 personnes.

Vous êtes le lead développeur de l'entité web.

Vous recevez un message de votre manager avec une bonne nouvelle...

Thomas : Hello ! Ça va ? Je sors d'une réunion avec Learn@Home ! Et j'ai un projet pour toi !

Vous : Ça va et toi ? Super ! C'est quoi ce projet ? 😊

Thomas : Learn@Home est une association qui met en relation des enfants en difficulté scolaire avec des tuteurs bénévoles. En gros, leur objectif est de permettre à tout élève, où qu'il soit, d'avoir accès à un soutien scolaire à distance. Ils nous ont confirmé qu'ils voulaient travailler avec nous sur leur projet de site web. 📧🔗

Vous : Génial ! À quoi servira leur site ?

Thomas : Ils veulent permettre à leurs élèves et à leurs bénévoles de communiquer plus facilement, directement depuis leur site. Jusqu'alors ils utilisaient surtout WhatsApp et les SMS !

Vous : Ah oui c'est un beau challenge !

Thomas : Je voulais te proposer de gérer ce projet : ton rôle serait de cadrer ce projet puis de coordonner tous les développements avec l'équipe. C'est un projet ambitieux et intéressant !

Vous : Merci ! 😊 Très beau projet en effet, j'ai hâte de démarrer. Tu m'envoies plus d'éléments ?

Thomas : Oui bien sûr, je t'envoie un mail dès demain.

Le lendemain, vous recevez bien le mail suivant...

Objet : Retour réunion Learn@Home

De : Thomas

À : Vous

Hello,

Comme promis je t'envoie plus d'informations, suite à la réunion que nous avons eue hier avec Learn@Home.

Learn@Home souhaite travailler avec nous sur toute la conception de son site web.

Notre première étape est de bien définir les besoins client, avant de démarrer le développement logiciel pur. Nous avons prévu une nouvelle réunion dans quelques semaines, l'objectif est que tu leur présentes les éléments suivants :

1. Les diagrammes de cas d'usage pour chacune des fonctionnalités majeures de chaque page (connexion, chat, calendrier, gestionnaire de tâches, tableau de bord), pas besoin d'utiliser l'UML.
2. Les user stories avec critère(s) d'acceptation pour chacune des fonctionnalités, partagées dans un format standard (PDF, DOCX, XLSX).
3. Les maquettes du site avec un design simple et au moins une maquette par page (tu peux utiliser [Figma](#) ou [Sketch](#)).

4. Un Kanban découpant le projet, de manière macro (détails techniques exclus) en blocs de fonctionnalités et sous-fonctionnalités pour le développement. J'ai démarré le découpage dans [ce kanban](#) sur Notion, à toi de finir de compléter la colonne "ANALYZE" sur le même modèle. Tu peux travailler sur Notion, Trello ou GitHub.

Je t'envoie [un document](#) dans lequel j'ai synthétisé les attentes du client. Tu y trouveras nos prises de notes et quelques croquis. Bien sûr, il faudra fournir les maquettes pour desktop et mobile.

Pour les délais, pas de stress, on a réussi à négocier avec le client pour que tu aies suffisamment de temps pour concevoir tous les documents et préparer ta réunion.

Si tu as des questions, n'hésite pas !

Thomas

Vous avez désormais toutes les cartes en main : c'est le moment de vous lancer sur ce projet !

Livrables

- Les diagrammes de cas d'usage.
- Les user stories avec critère(s) d'acceptation.
- Les maquettes du site.
- Le tableau Kanban (Notion, Trello, or GitHub).

Soutenance

Votre soutenance correspond à la réunion avec votre client : Learn@Home. L'objectif de cette réunion est de valider le cadrage du projet avec le client, avant de démarrer le développement fonctionnel. En tant que responsable du projet, c'est vous qui allez présenter les différents éléments sur lesquels vous avez travaillé. L'évaluateur de votre soutenance jouera le rôle du client. Votre présentation sera structurée de la façon suivante :

- **Présentation des livrables** (10 minutes). Après avoir résumé le besoin client et la solution envisagée, vous présentez vos livrables :
 - les maquettes ;
 - les user stories avec critère(s) d'acceptation ;
 - les diagrammes de cas d'usage ;
 - le tableau kanban (Notion, Trello ou GitHub).
- **Discussion** (15 minutes). L'évaluateur, jouant toujours le rôle du client, peut vous poser quelques questions sur votre présentation. Vous devez être capable de justifier vos choix. Si des erreurs ou oublis sont mis en lumière, vous devrez proposer des corrections ou solutions.
- **Débrief** (5 minutes). L'évaluateur sortira de son rôle pour débriefer avec vous.

Compétences évaluées

- Contribuer à un projet en utilisant une méthodologie agile
- Créer une maquette pour un client
- Identifier les besoins de l'application à partir d'un cahier des charges
- Modéliser une solution technique pour un client

Projet 13 :

Ressources :

Vous venez de commencer à travailler chez Remede Agency, une agence spécialisée dans le développement d'applications web. L'équipe est principalement composée de développeurs back-end et vient de commencer à recruter d'autres spécialistes front-end.

Remede Agency

Logo de Remede Agency

Après quelques jours dans l'entreprise, vous recevez le courriel suivant :

Objet: Bienvenue !

De: Mila

À: Moi

Bienvenue dans l'équipe !

Je m'appelle Mila Lavigne et je suis ta nouvelle cheffe de projet. Nous sommes très heureux que tu rejoignes l'équipe ! J'ai une excellente nouvelle : le contrat dont nous t'avons parlé lors de l'entretien a été accepté et le client est prêt pour qu'on démarre.

Le projet concerne une nouvelle banque qui démarre, Argent Bank, qui essaie de percer dans le secteur et qui a besoin d'aide pour mettre en place son application. Nous avons obtenu un contrat en deux parties qui se décompose en plusieurs phases :

- **Phase 1 : Authentification des utilisateurs** - Création d'une application web permettant aux clients de se connecter et de gérer leurs comptes et leur profil.
- **Phase 2 : Transactions** - Il s'agirait de spécifier les endpoints d'API nécessaires pour une éventuelle deuxième mission une fois que nous aurons terminé la première.

À la fin du projet, tu présenteras les deux livrables à Avery Moreau, qui gère l'équipe technique d'Argent Bank.

Nous attendons encore des informations, mais je veillerai à ce que tu sois inclus dans toutes les communications pour que tu sois à jour sur le projet.

Au plaisir de travailler avec toi !

Mila

Cheffe de projet

Quelques heures plus tard, vous recevez l'e-mail suivant :

ARGENT BANK

Logo de Argent Bank

Objet: Brief mission profil Argent Bank

De: Avery

À: Moi, Mila

Bonjour Mila et l'équipe,

Je suis Avery Moreau VP Engineering chez Argent Bank. Nous sommes ravis de vous avoir avec nous pour créer notre application web React pour le nouveau système d'authentification des utilisateurs. Nous avons des délais très serrés et nous aurons besoin d'aide !

Voici un aperçu de ce dont nous avons besoin pour la **phase 1 : Authentification des utilisateurs**.

- Créer l'application web complète et responsive avec React. Comme point de départ, nous vous avons fourni le [HTML statique et le CSS](#) pour la page d'accueil, la page de connexion et la page de profil.
- Utiliser Redux pour gérer le state de l'ensemble de l'application.
- Ce que doit faire l'application (voir les détails pour chacune sur [nos modèles de GitHub Issues](#)) :
 - L'utilisateur peut visiter la page d'accueil
 - L'utilisateur peut se connecter au système
 - L'utilisateur peut se déconnecter du système
 - L'utilisateur ne peut voir les informations relatives à son propre profil qu'après s'être connecté avec succès
 - L'utilisateur peut modifier le profil et conserver les données dans la base de données.

Vous pouvez commencer par forker [notre repo existant](#) et suivre l'avancement du travail grâce aux modèles d'Issues GitHub que nous y avons incluses.

Nos ingénieurs back-end ont déjà créé toutes les API dont vous avez besoin. Vous trouverez toute la documentation Swagger à l'intérieur du repo.

Pour la **phase 2 : Transactions**, nous sommes encore en phase de conception. De notre côté, nous mettons au point une fonctionnalité pour les transactions qui doit pouvoir permettre aux utilisateurs :

- de visualiser toutes leurs transactions pour le mois en cours ;
- de visualiser les détails d'une transaction dans une autre vue ;
- d'ajouter, de modifier ou de supprimer des informations sur une transaction.

Puisque vous gérez déjà l'application web pour la phase 1, nous voulons connaître votre avis sur la façon dont vous pensez que les API devraient être modélisées du côté back end. Nous avons besoin que vous nous fournissiez un document décrivant les API proposées pour les transactions, en suivant les directives de Swagger.

Parmi les éléments clés à spécifier pour chaque endpoint de l'API il faudra :

- La méthode HTTP (ex. : GET, POST, etc.)
- La route (ex. : /store/inventory)
- La description de ce à quoi correspond l'endpoint (ex. : Retour de l'inventaire des animaux de compagnie)
- Les paramètres possibles pour tenir compte des différents scénarios (ex. : itemId (facultatif) = ID de l'article spécifique à demander à la base de données d'inventaire).
- Les différentes réponses avec les codes de réponse correspondants qui ont un sens pour cet endpoint (ex. : 404 : réponse d'erreur d'article inconnu).

Vous pouvez utiliser la page des transactions présentée dans les maquettes pour guider vos choix (mais vous n'avez pas besoin d'implémenter cette page). Assurez-vous simplement que le document est exporté vers un fichier YAML (c'est-à-dire Fichier > Enregistrer sous YAML) en utilisant la syntaxe Swagger, qui peut être exportée dans [l'outil d'édition de Swagger](#).

Nous ferons une revue de code et discuterons ensemble de la proposition d'API une fois que tout sera terminé.

Au plaisir de travailler avec vous !

Avery Moreau

Vous remarquez que l'interface est en anglais. Même si vous n'êtes pas très à l'aise dans cette langue, vous vous dites que c'est l'occasion de pratiquer un peu ! C'est toujours bien pour un développeur de lire l'anglais, car la documentation professionnelle se fait souvent dans cette langue. Et Google Traduction est votre ami 😊

Toutes ces informations en main, vous vous mettez au travail dès que possible !

Livrables

Pour ce projet, vous devrez créer les éléments suivants :

- Un dépôt GitHub qui contient la base de code complète pour remplir toutes les exigences de la phase 1 : Authentification de l'utilisateur.
- Un fichier swagger.yml des modèles d'API proposés avec les endpoints requis pour la phase 2 : Transactions.

Le repo GitHub doit être nommé avec la convention suivante :

Nomcomplet_#_Datedémarrage. Le # correspond au numéro du projet sur le parcours et la date doit être au format jjmmaaaa. Par exemple, FrancoisLenotre_5_05032020.

Soutenance

La présentation sera structurée comme suit, l'évaluateur jouant le rôle d'Avery Moreau :

- Présentation et explication des produits livrables (10 minutes)
- Discussion (15 min)

- L'évaluateur vous posera des questions sur votre méthodologie et vos résultats.
- Soyez prêt à défendre votre travail, car l'évaluateur contestera vos décisions.

L'évaluateur cessera de jouer le rôle d'Avery Moreau cinq minutes avant la fin de la session afin que vous puissiez debriefer ensemble.

Projet 14 :

Vous travaillez pour le département technologique d'une grande société financière, WealthHealth.

Cette société utilise une application web interne, appelée HRnet, qui gère les dossiers des employés. L'application est ancienne et utilise jQuery côté front end, ce qui entraîne des bugs considérables et une augmentation des plaintes en interne. Votre équipe de développement s'efforce depuis un certain temps déjà de mettre à niveau l'application.



WEALTH HEALTH

Logo de Wealth Health

Un matin, vous vous rendez au travail et remarquez les messages suivants de Jade, une collègue développeuse de l'équipe :

Jade : Salut, toi ! J'espère que tu as passé un bon week-end ! Tu as entendu la nouvelle ? La direction va ENFIN nous laisser mettre à jour HRnet et le convertir en React ! Plus de maintenance du vieux code jQuery !! Youhouuu !! 🎉 😊

Vous : Oh, vraiment ? Trop bien ! Le jour est enfin arrivé ! 🙌 🙌

Jade : Oui, c'est tellement excitant ! Et puisque tu t'es donné à fond et que tu as fait un travail incroyable ces derniers temps, tu seras l'un des premiers développeurs à commencer à convertir le code ! Félicitations !

Vous : C'est génial ! J'ai hâte de me mettre dedans pour résoudre (enfin) ce problème ! HRnet à lui seul doit représenter environ 90% de notre dette technologique totale, lol 🤖

Jade : M'en parle pas ! Je me suis dit que je t'enverrais des informations pour t'aider à démarrer. Comme tu le sais sûrement, les plus gros problèmes pour les utilisateurs de HRnet en ce moment sont les sélecteurs de date, les fenêtres modales, les menus déroulants et les tableaux. Nous avons reçu plusieurs plaintes disant que ces plugins jQuery sont très lents.

Nous voulons donc créer nos propres composants React à la place de ces plugins jQuery tiers qui sont utilisés dans l'interface utilisateur. Et on espère que la conversion de ces plugins jQuery en composants React améliorera les performances et la stabilité.

Vous : Ok, compris. Tous ces appels réseau et toute cette manipulation DOM nécessaire pour les plugins jQuery s'additionnent et ralentissent les choses. Et ce sera bien de commencer à avoir nos propres composants React pour avoir plus de contrôle sur l'application.

Jade : Exactement ! La direction ne veut pas que cette première partie du processus de conversion prenne trop de temps, alors s'il te plaît, choisis UN SEUL des quatre plugins jQuery et convertis-le en composant React. (Je te les envoie plus tard dans la journée). Tu devras créer un repo GitHub séparé pour le code converti.

Vous : Bien sûr ! C'est une bonne manière d'avoir une version initiale de la nouvelle application plus rapidement, et nous pourrons toujours itérer dessus plus tard.

Jade : Exactement ! Pour tester manuellement tes composants React convertis, tu devras aussi convertir l'application HRnet principale en une application React. Comme tu sais que jQuery et React ne vont pas bien ensemble, nous voulons donc que la nouvelle version de l'application soit 100% React et 0% jQuery. Nous ne voulons pas créer un genre de monstre jQuery/React !

Vous : C'est compris. Merci de le préciser, je commençais déjà à trembler à l'idée de créer un hybride React/jQuery 🤖

Jade : Non 🙅🙅🙅🙅🙅🙅 On ne veut vraiment pas ça ahah

Je t'envoierai le repo actuel de HRnet pour que tu puisses commencer. Nous aurons aussi besoin que tu fasses des tests de performance et que tu nous fournisses des rapports.

Vous : Bien vu ! Je veillerai à inclure ces rapports de performance dans mes livrables.

Jade : Aussi, garde à l'esprit que nous voulons suivre un paradigme de programmation fonctionnel en écrivant ces bibliothèques dans React, donc évite d'utiliser des classes quand tu le convertis. Et essaie d'écrire des morceaux de code plus petits et modulaires et des fonctions autonomes pour une modularité et une maintenabilité optimales.

Aussi, assure-toi de documenter ton composant React converti avec une description générale de ce que fait le composant et des commentaires expliquant à quoi chaque accessoire est destiné et comment il est utilisé. Le reste de l'équipe te remerciera !



Vous : D'accord, je m'occupe de ça !

Jade : Perfecto ! Cela devrait être tout ce dont tu as besoin pour commencer - je t'envoie aussi un e-mail récapitulatif tout à l'heure. Si tu as des questions ou si tu as besoin d'autres détails ou ressources, n'hésite pas à me le faire savoir. Allons nous débarrasser de ce code jQuery ! 🍷

Vous : Je suis sur le coup ! Tu peux compter sur moi, Jade ! 🍷

Comme promis, un peu plus tard dans la journée, Jade vous envoie le mail suivant :

Objet: Détails sur le projet de refonte

De: Jade

À: Moi

Salut à toi !

J'ai hâte que tu puisses démarrer ce projet ! Voici un récapitulatif des principales tâches :

- Convertir l'ensemble du projet HRNet en React.
- Convertir l'un des quatre plugins jQuery actuels en React. Remplacer les 3 plugins jQuery restants par des composants React.
- Effectuer des tests de performance Lighthouse en comparant l'ancienne et la nouvelle application.

Tu trouveras toutes les informations clés pour chaque tâche ci-dessous :

Conversion du projet HRNet

Voici [le repo HRnet](#) actuel. N'oublie pas que toute l'application HRNet doit être convertie en React :

- Tu devras faire une nouvelle version des pages "Create Employee" et "Employee List" avec React.
- Tu devras ajouter un système de gestion d'état (la version actuelle utilise un stockage local).
- Tu dois aussi t'assurer que tout est cohérent au niveau du style. Pour cela, tu n'es pas obligé de refaire le design de l'application, mais si tu veux changer le style pour quelque chose de plus moderne, tu es le bienvenu.
- Si tu as le temps, tu peux tester le code React avec une suite de tests unitaires. Sinon, seuls des tests manuels sont nécessaires.

Conversion des plugins

Pour les plugins, n'hésite pas à consulter l'onglet Issues pour avoir plus de contexte sur les problèmes que les utilisateurs rencontrent avec les plugins jQuery existants : [issue sur les sélecteurs de date](#), [issues de fenêtres modales](#), [issue sur menus déroulants](#), et [issue sur les tableaux](#).

Voici la liste des plugins jQuery actuellement utilisés qui doivent être convertis :

- [Plugin de sélection de date](#)
- [Plugin de fenêtre modale - jQuery.modal.js](#)
- [Menus déroulants](#)
- [Plugin pour les tables de données](#)

De plus, lors de la conversion d'un plugin jQuery en un composant React, il faut garder à l'esprit de ne convertir que le code qui traite de la fonctionnalité réelle de l'interface utilisateur du plugin. Par exemple, si un plugin jQuery inclut du code AJAX, tu n'as pas besoin de le convertir. Si tu convertis un plugin jQuery pour une fenêtre modale, concentre-toi sur la création d'un composant React qui fonctionne comme une fenêtre modale, et rien d'autre.

Tests de performance

Nous voulons également mesurer des données quantifiables (ex. : temps de chargement des pages, appels réseau) pour nous assurer que la conversion de l'application à React améliore effectivement les performances. Pour cela, fais bien des audits de performance Lighthouse. Pour comparer, tu devras en faire un pour

l'application jQuery HRnet actuelle, puis une autre fois que l'application et le plugin jQuery choisis seront convertis en React.

C'est tout ! Une fois que l'application HRnet en React fonctionne, tu peux publier le composant React sur npm sous forme de package et partager le lien pour que nous puissions l'utiliser si nécessaire. Si tu as du mal avec npm, tu peux utiliser les paquets GitHub comme alternative.

Aussi, fais-moi savoir quand tout est fait et nous ferons une révision du code pour répondre aux commentaires restants.

J'ai hâte de voir ce que tu vas faire !

Jade

Ça y est, vous avez toutes les informations dont vous avez besoin. Maintenant, le plus dur sera de choisir quel plugin jQuery convertir !

Livrables

- Un **repo GitHub** contenant :
 - le code source du plugin jQuery sélectionné converti en ReactJS (un seul plugin doit être converti).
 - Un lien vers la bibliothèque React convertie publiée sur npm (ou GitHub Packages) dans le fichier README.
- Un **second repo GitHub** contenant le code source de l'application HRnet converti en React.
- Un **rapport de performance Lighthouse** de HRnet fonctionnant avant et après la conversion de la bibliothèque, au format JSON.

Le repo GitHub doit être nommé avec la convention suivante : Nomcomplet_#_Datedémarrage. Le # correspond au numéro du projet sur le parcours et la date doit être au format jjmmaaaa. Par exemple, FrancoisLenotre_12_05032020.

Soutenance

La présentation sera structurée comme suit, l'évaluateur jouant le rôle de Jade :

- Présentation et explication des livrables (10 minutes)
- Discussion (15 min)
 - L'évaluateur vous posera des questions sur votre méthodologie et vos résultats.
 - Soyez prêt à défendre votre travail, car l'évaluateur mettra en question vos décisions.

L'évaluateur cessera de jouer le rôle de Jade cinq minutes avant la fin de la session afin que vous puissiez faire un compte rendu ensemble.

