

Débuggez et testez un SaaS RH

 openclassrooms.com/fr/projects/debuggez-et-testez-un-saas-rh/assignment

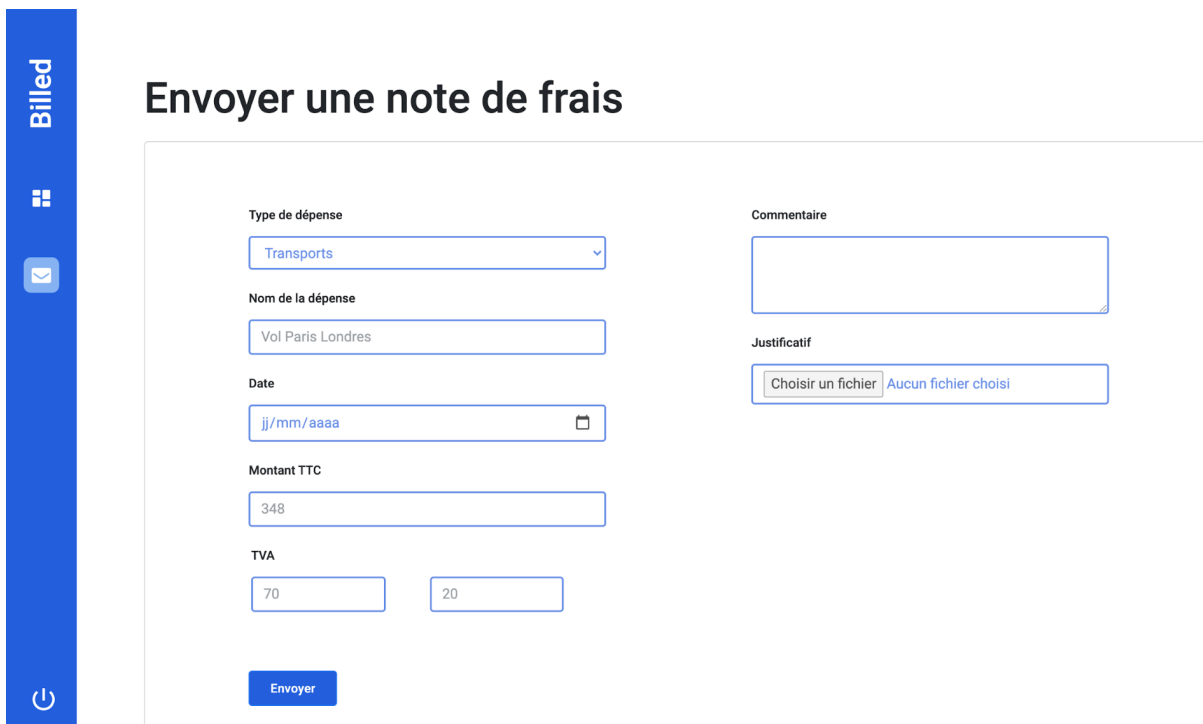
 60 heures Mis à jour le lundi 6 septembre 2021

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



Logo de Billed

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 !



Interface de l'application Billed

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.

Note concernant la connexion à l'application : nous vous recommandons de ne pas utiliser votre adresse email personnelle pour vous connecter à l'application ni y télécharger des documents personnels si vous ne voulez pas que les autres étudiants qui utilisent la base de données puisse accéder à vos données. Cependant, il vous faut utiliser la même adresse côté administration RH et côté employé pour pouvoir voir les notes de frais que vous aurez créées.

-



Ecrire des tests unitaires avec JavaScript

-



Ecrire des tests d'intégration avec JavaScript

-



Débugger une application web avec le Chrome Debugger

-



Rédiger un plan de test end-to-end manuel