Billed

Mise en place de la fonctionnalité “notes de frais”
Description pratique des besoins

Contexte

- Départ imprévu de Garance qui travaillait initialement sur le projet et remplacement au pied levé.
- Fonctionnalité très attendue et timing serré car lancement dans deux semaines.
- Deux parcours sur la fonctionnalité : 1. Employé et 2. Administrateur RH.
- Où en sommes-nous actuellement ?

	Back-end	Front-end
Parcours employé	✅ Prêt en version alpha	❌ À tester ❌ À débugger
Parcours admin	✅ Prêt en version alpha	✅ Testé ❌ À débugger

Équipe et missions

Matthieu	Garance	Remplaçant(e) de Garance	Leïla
Lead developer Supervise le projet	Feature team A développé, débuggé et testé une partie de la fonctionnalité A quitté l’entreprise	Front-end developer Débugge et teste la fonctionnalité	Quality assurance Identifie les bugs et réalise les tests End-to-End

Tâche	Description de la tâche	Règles/contraintes
1. [Bug - report]	**Fixer les bugs identifiés dans le rapport de bug fourni par Jest.** Une copie est disponible dans le kanban Notion.	Utiliser **Chrome Debugger**.
2. [Bug - hunt]	**Fixer les bugs identifiés par Leila sur le parcours employé.** Ils sont décrits dans le kanban Notion.	Utiliser **Chrome Debugger**.

3. [Tests unitaires et d'intégration]	**Ajouter des tests unitaires et d'intégration pour les fichiers Bills et NewBill.** Ils vont permettre d'éliminer les bugs et d'éviter toute régression lors des prochaines évolutions de la solution. Certains tests sont déjà développés (pour le Login et pour le Dashboard côté administrateur RH) : ils sont déjà cochés sur le kanban. Il faut s'en inspirer pour les restants.	Il faut assurer un **taux de couverture global des containers de 80% minimum (tests unitaires ET tests d'intégration).** **Conseils :** ⇨ **S'appuyer sur le rapport de couverture de Jest** Lancer l'application avec live-server pour pouvoir le lire et aller à l'adresse http://127.0.0.1:8080/coverage/lcov-report/ (tout est indiqué dans le readme). ⇨ Comme pour le fichier Dashboard.js, **s'appuyer sur le mock de l'API**. ⇨ S'assurer d'**utiliser des matchers pertinents,** afin de bien tester l'application et pas simplement obtenir une bonne couverture. ⇨ Ne pas oublier de **tester les erreurs 404 et 500**.
4. [Test End-to-End]	**Rédiger un plan de test End-to-End (E2E) sur le parcours employé** pour guider Leïla.	Manque de temps pour automatiser les tests (E2E). Ils seront effectués manuellement par Leila. S'inspirer du plan E2E que Garance a déjà rédigé sur le parcours administrateur RH.

Autres informations

- L'application contient déjà des données test mais il est nécessaire d'en créer de nouvelles.
- Des comptes administrateur et employé ont été créés pour les tests dans le readme du code front-end. Il faut les utiliser pour pouvoir charger une note de frais côté employé et la consulter côté administrateur RH.