XEFI GRAND LYON — DIVISION HOPLA

Compte rendu de stage

Projet : Développement de compétences en développement web

Maxime BLANCO
BTS SIO — Année 2025-2026
Entreprise : Xefi Grand Lyon — Division HOPLA

Technologies utilisées

Laravel · Docker · WSL2 · PhpStorm · Postman · VueJS · NuxtJS · Vuetify · i18n · GitLab

Table des matières

Table des matières 2
1. Présentation du stage 3
1.1 L'entreprise 3
1.2 Contexte du stage 3
1.3 Les missions confiées 3
1.4 Les personnes concernées 3
2. Outils et environnement de travail 4
2.1 L'environnement de développement 4
2.2 Les outils utilisés 4
2.3 La base de données du projet scolaire 5
3. Réalisations 6
3.1 Projet 1 — La partie serveur de l'application scolaire 6
3.2 Projet 2 — L'interface machine à sous 7
3.3 Projet 3 — La radio interne de Xefi 8
3.4 Projet 4 — Les guides utilisateurs de WeDrop 9
Organisation du code 9
Les interfaces réalisées 11
Les traductions 13
Livraison du travail 14
4. Difficultés rencontrées 16
4.1 Côté serveur (Projet 1) 16
4.2 Côté interface (Projets 2, 3 et 4) 16
5. Bilan 17
5.1 Ce que ce stage m'a apporté 17
5.2 Compétences validées 17
5.3 Ce que je ferais mieux 17
5.4 Conclusion 17

1. Présentation du stage

1.1 L'entreprise

Xefi est une entreprise spécialisée dans les services informatiques pour les professionnels. Le stage s'est déroulé au sein de la division HOPLA, une équipe interne dédiée au développement de logiciels web utilisés par des entreprises clientes.

HOPLA développe notamment WeDrop, une solution de stockage et partage de fichiers en ligne destinée aux entreprises — un peu comme Google Drive, mais hébergé et contrôlé par Xefi.

1.2 Contexte du stage

Ce stage avait un double objectif : d'un côté, tester et valider des compétences techniques acquises en formation, et de l'autre, contribuer à de vrais projets utilisés par l'entreprise.

Pour cela, plusieurs missions ont été confiées successivement, du plus simple au plus complexe, afin de progresser et de gagner en autonomie.

1.3 Les missions confiées

Le stage s'est organisé autour de quatre projets distincts :

- Projet 1 — Créer la partie « serveur » d'une application de gestion scolaire inspirée de Pronote : gérer les utilisateurs, les classes, les devoirs et les notes
- Projet 2 — Développer une interface de machine à sous pour apprendre à créer des pages web interactives
- Projet 3 — Ajouter une fonctionnalité à la radio interne de Xefi : gérer les utilisateurs bannis depuis une interface d'administration
- Projet 4 — Créer et intégrer les guides utilisateurs de WeDrop, disponibles en 6 langues, directement dans l'application

1.4 Les personnes concernées

- Demandeur : les encadrants techniques de la division HOPLA
- Réalisateur : Maxime Blanco, stagiaire BTS SIO
- Utilisateurs finaux : les équipes Xefi et les clients de WeDrop

2. Outils et environnement de travail

2.1 L'environnement de développement

Tout le travail a été réalisé sur un poste Windows, avec un environnement Linux installé en parallèle (via WSL2, qui permet de faire tourner Linux à l'intérieur de Windows). Cette configuration est celle utilisée par l'équipe en entreprise.

La base de données tournait dans un conteneur Docker — un outil qui permet d'isoler un programme dans une sorte de « boîte virtuelle » afin qu'il ne perturbe pas le reste du système. Ci-dessous, l'interface Docker avec le conteneur actif :

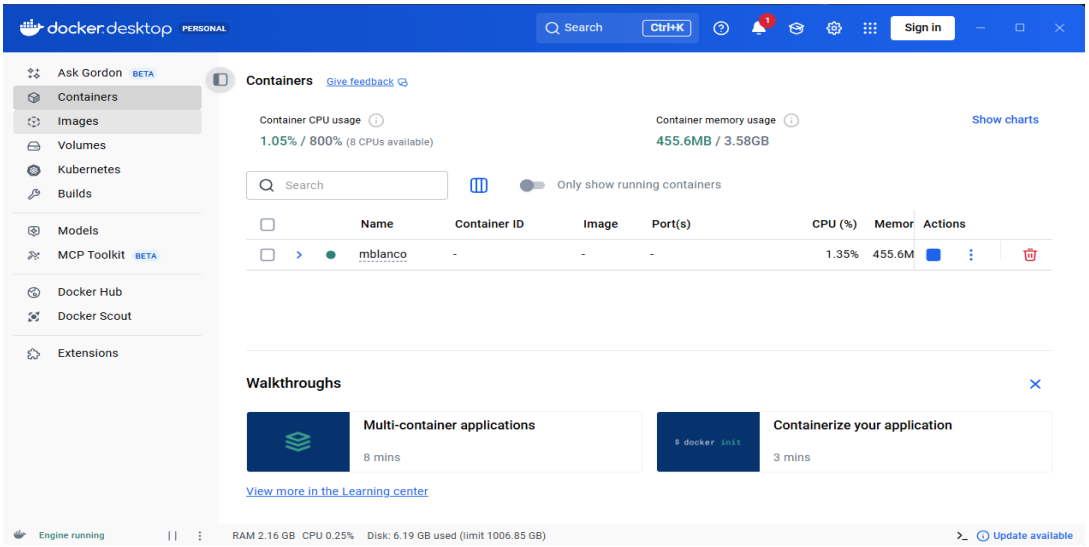

Figure 1 — Interface Docker Desktop avec le conteneur de base de données actif

2.2 Les outils utilisés

Outil	À quoi ça sert ?
Laravel	Outil pour construire la partie invisible d'un site web (ce qui tourne côté serveur : gestion des données, des utilisateurs, des règles métier)
VueJS / NuxtJS	Outils pour construire les pages web et les interfaces que l'utilisateur voit et utilise directement
Vuetify	Bibliothèque de boutons, menus et éléments visuels prêts à l'emploi pour construire des interfaces propres
vue-i18n	Système de traduction automatique : permet d'afficher le site dans plusieurs langues sans tout réécrire
PhpStorm	Logiciel de rédaction de code (comme Word, mais pour programmer)
Docker	Outil qui isole la base de données dans une boîte virtuelle pour ne pas polluer le reste du système
WSL2	Permet d'utiliser Linux (un autre système d'exploitation) directement depuis Windows
Postman	Outil pour vérifier que la partie serveur répond correctement aux demandes

Outil	À quoi ça sert ?
GitLab	Espace de stockage et de partage du code, avec un historique de toutes les modifications effectuées

2.3 La base de données du projet scolaire

Pour le projet de gestion scolaire, une base de données a été conçue pour stocker toutes les informations : les utilisateurs, les classes, les devoirs et les notes. Ces informations sont liées entre elles, un peu comme dans un tableau Excel avec plusieurs onglets qui se référencent les uns les autres.

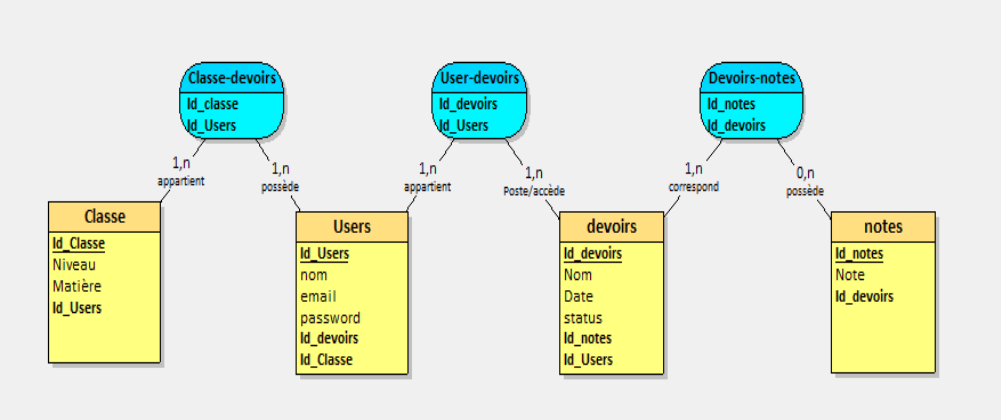

Figure 2 — Schéma de la base de données : les informations et leurs liens

3. Réalisations

3.1 Projet 1 — La partie serveur de l'application scolaire

Le premier projet consistait à construire la partie « invisible » d'une application de gestion scolaire, inspirée de Pronote. Cette partie, qu'on appelle le back-end, est le moteur caché qui gère les données : elle reçoit des demandes, les traite, et renvoie les bonnes informations selon qui les demande.

En pratique, cela signifie qu'un administrateur peut créer des comptes, qu'un professeur peut déposer des devoirs et des notes, et qu'un élève peut consulter ses résultats — et que chacun ne voit que ce à quoi il a accès.

Ce qui a été réalisé :

- Mise en place de la structure de la base de données (les « cases » où sont stockées les informations)
- Création de données de test pour simuler de vrais utilisateurs, classes et devoirs
- Mise en place des droits d'accès selon le rôle : élève, professeur ou administrateur
- Création et vérification de toutes les actions possibles via Postman

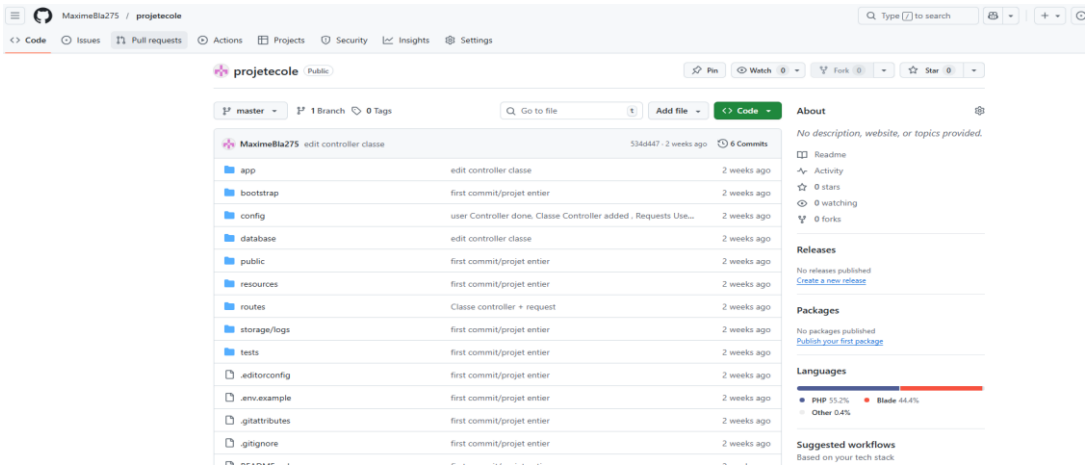

Figure 3 — Organisation du code du projet scolaire sur GitLab

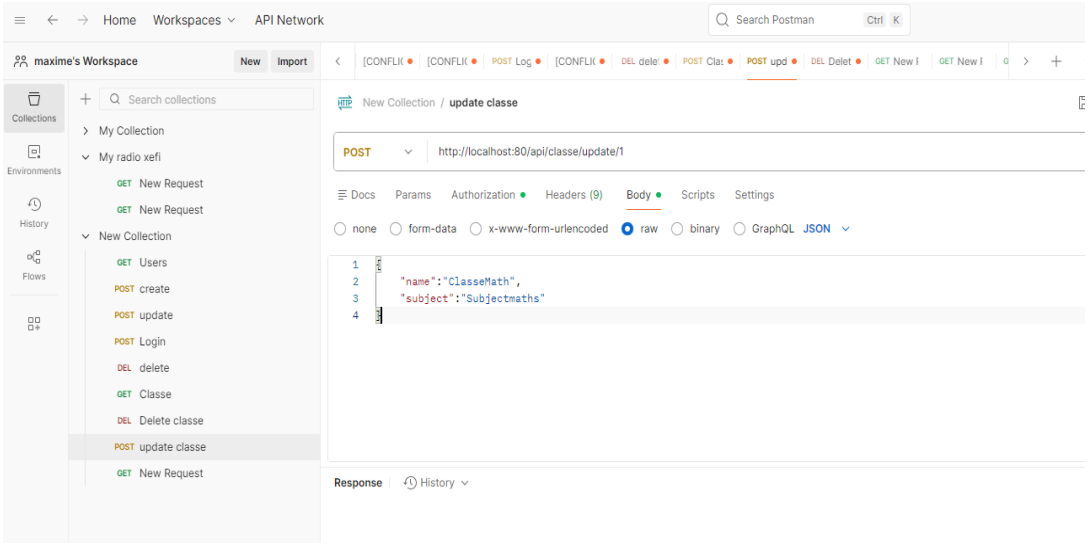

Figure 4 — Postman : vérification que chaque action renvoie le bon résultat

3.2 Projet 2 — L'interface machine à sous

Le deuxième projet était un exercice pédagogique pour apprendre à créer des interfaces web interactives : des pages où l'utilisateur clique sur des boutons et voit l'affichage changer en temps réel, sans recharger la page.

Le sujet choisi était une machine à sous simple. À chaque partie, trois chiffres aléatoires s'affichent. Si les trois sont identiques, c'est un jackpot. Les crédits augmentent ou diminuent selon les résultats, et le jeu se bloque quand il n'en reste plus.

Ce qui a été réalisé :

- Interface avec boutons pour ajouter des crédits et jouer
- Affichage dynamique des résultats et des messages (gain, perte, crédits insuffisants)
- Organisation du code pour séparer la logique du jeu de l'affichage visuel

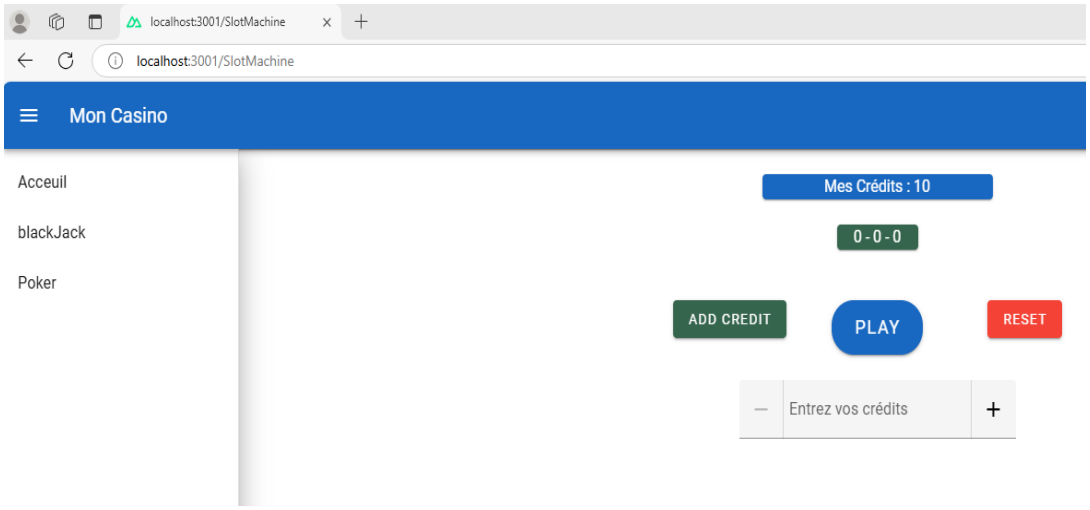

Figure 5 — Interface de la machine à sous

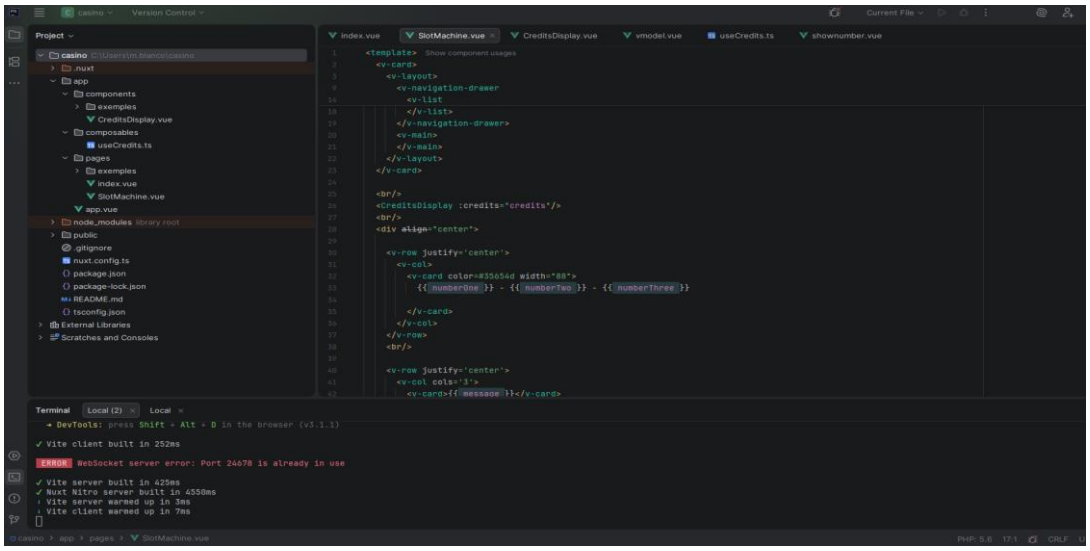

Figure 6 — Organisation du code et logique de gestion des crédits

Une difficulté rencontrée lors de ce projet était la gestion des actions qui prennent du temps (comme récupérer des données depuis un serveur). Le schéma ci-dessous illustre la différence entre une exécution classique, où chaque action attend la fin de la précédente, et une exécution en parallèle :

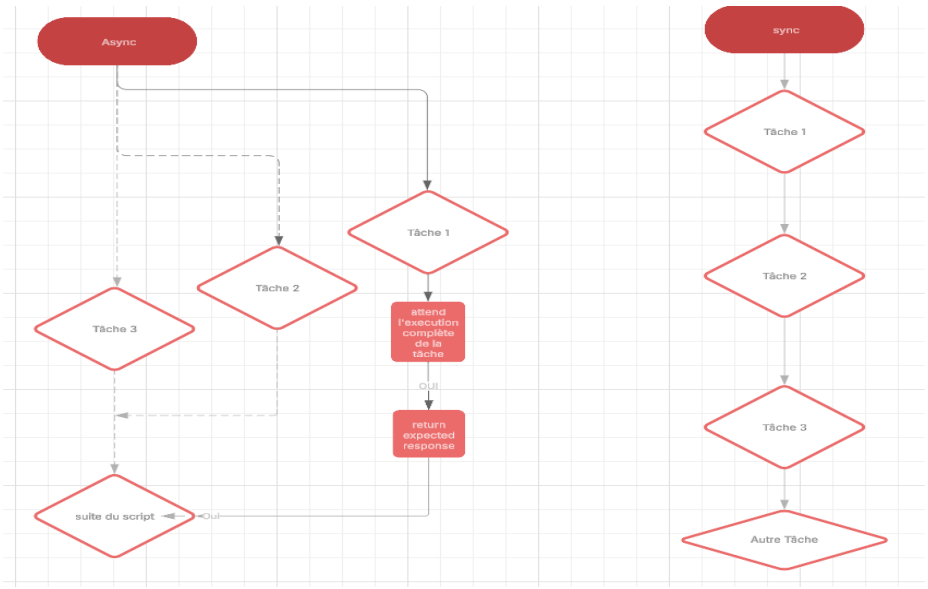

Figure 7 — Exécution classique (droite) vs exécution en parallèle (gauche)

3.3 Projet 3 — La radio interne de Xefi

La radio interne de Xefi est une plateforme où les employés peuvent diffuser et regarder des lives en direct — un peu comme Twitch, mais en interne à l'entreprise.

La mission était d'ajouter une fonctionnalité à l'espace d'administration : permettre aux modérateurs de voir la liste des utilisateurs qui ont été bannis d'un live, et de choisir de les débannir ou de maintenir leur bannissement.

Tout le travail a été réalisé sur la partie visible de l'interface (le front-end), c'est-à-dire les boutons, les listes et les actions accessibles depuis l'écran. Les données étaient déjà gérées côté serveur par l'équipe.

Ce qui a été réalisé :

- Affichage de la liste des utilisateurs bannis pour chaque live
- Boutons d'action : débannir un utilisateur ou conserver son bannissement
- Mise à jour automatique de l'affichage après chaque action

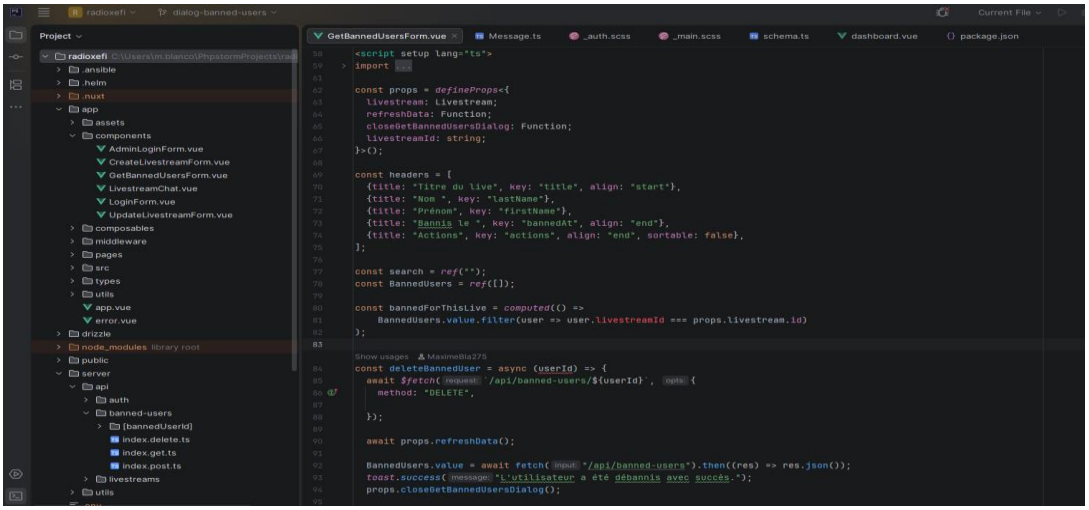

Figure 8 — Code de récupération et affichage des utilisateurs bannis

Les captures suivantes montrent la structure des échanges entre l'interface et le serveur, à titre de référence technique :

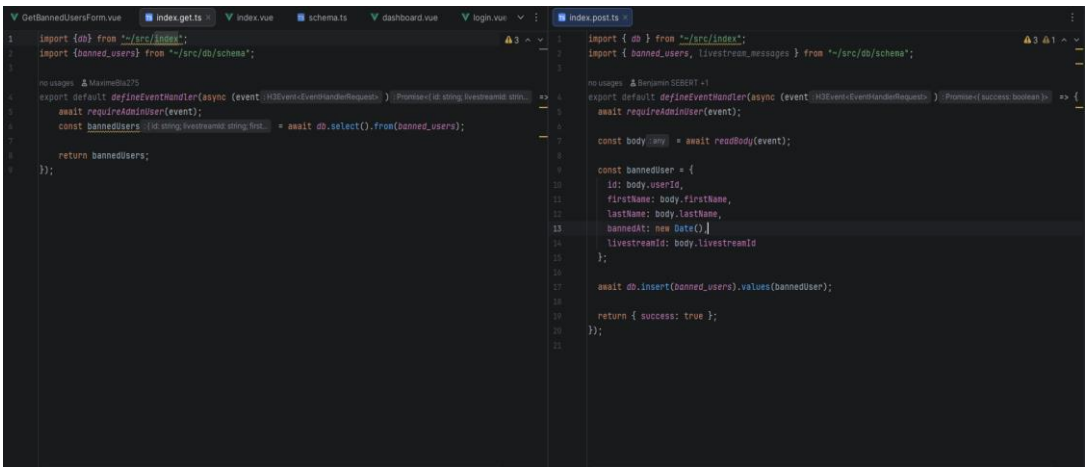

Figure 9 — Les différents types d'échanges possibles avec le serveur

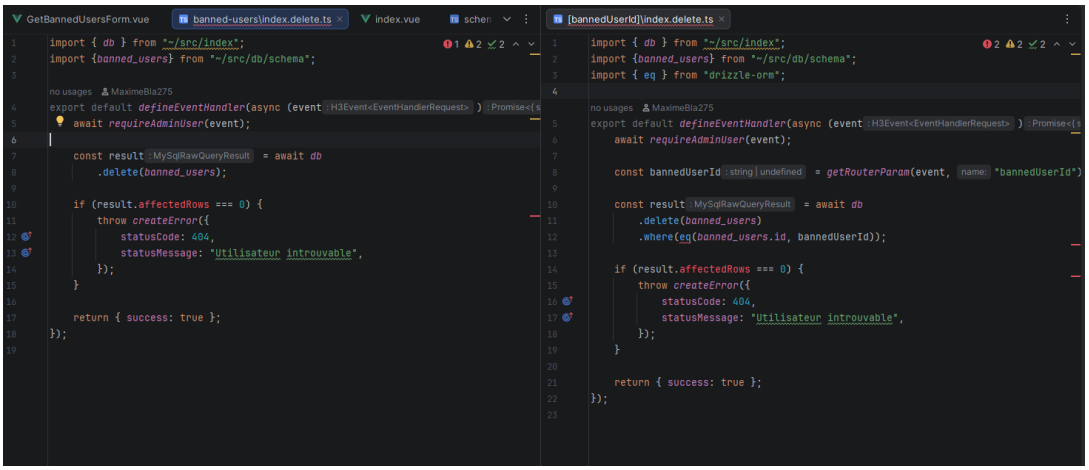

Figure 10 — Suppression d'un bannissement ou de tous les bannissements

3.4 Projet 4 — Les guides utilisateurs de WeDrop

WeDrop est une solution de partage de fichiers en ligne développée par l'équipe HOPLA. C'est un peu comme Google Drive ou Dropbox, mais hébergé par Xefi pour les entreprises qui souhaitent garder le contrôle de leurs données.

La mission confiée était de créer et d'intégrer directement dans l'application un système de guides utilisateurs interactifs — des tutoriels pas-à-pas qui guident l'utilisateur dans la découverte des fonctionnalités du logiciel, sans qu'il ait besoin de lire une notice séparée.

Il fallait également que ces guides soient disponibles en six langues différentes : français, anglais, espagnol, allemand, italien et néerlandais.

Organisation du code

Le code a été organisé en petits blocs réutilisables, du plus simple au plus complexe — un peu comme des pièces de Lego qu'on assemble pour former une interface complète. Cette méthode, utilisée par toute l'équipe HOPLA, facilite la maintenance et les évolutions futures.

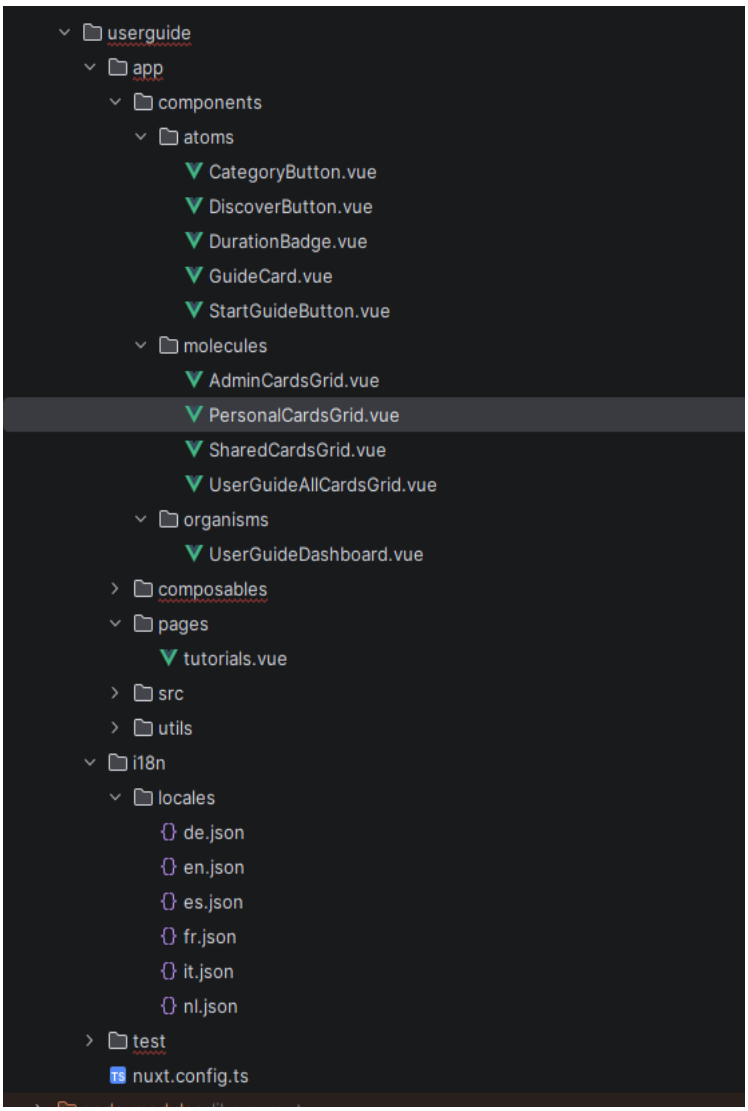

Figure 11 — Organisation des fichiers du module guides utilisateurs

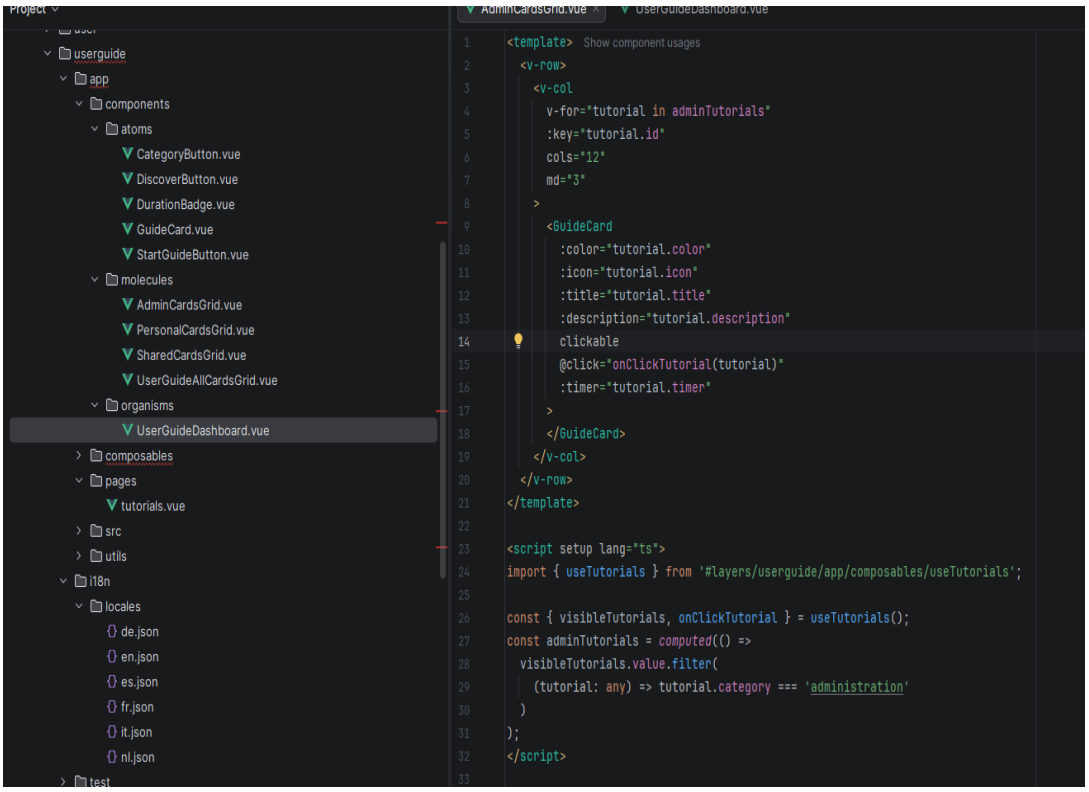

Figure 12 — Exemple de code d'un bloc affichant les guides d'administration

Les interfaces réalisées

La page principale des guides, telle qu'elle apparaît pour un utilisateur ordinaire :

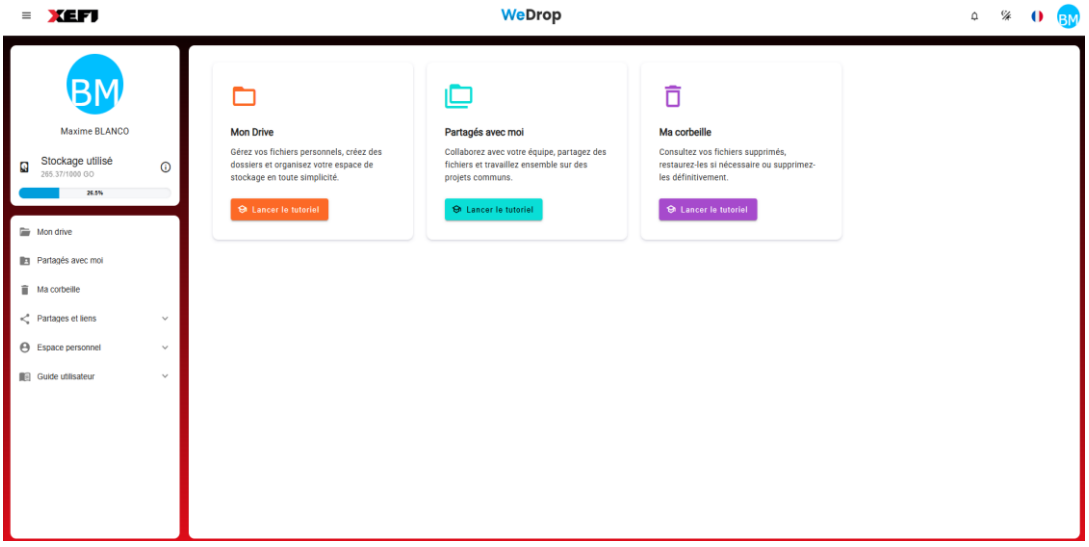

Figure 13 — Page guides vue par un utilisateur (Maxime Blanco)

La même page dans l'espace administrateur, avec davantage de guides disponibles et un filtrage par catégorie :

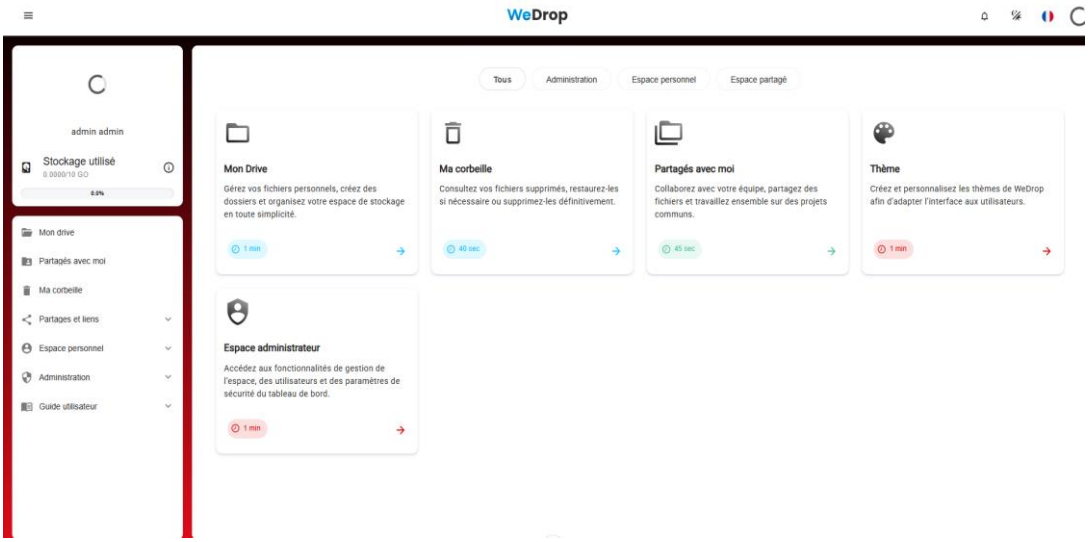

Figure 14 — Page guides vue par un administrateur — catégorie Administration

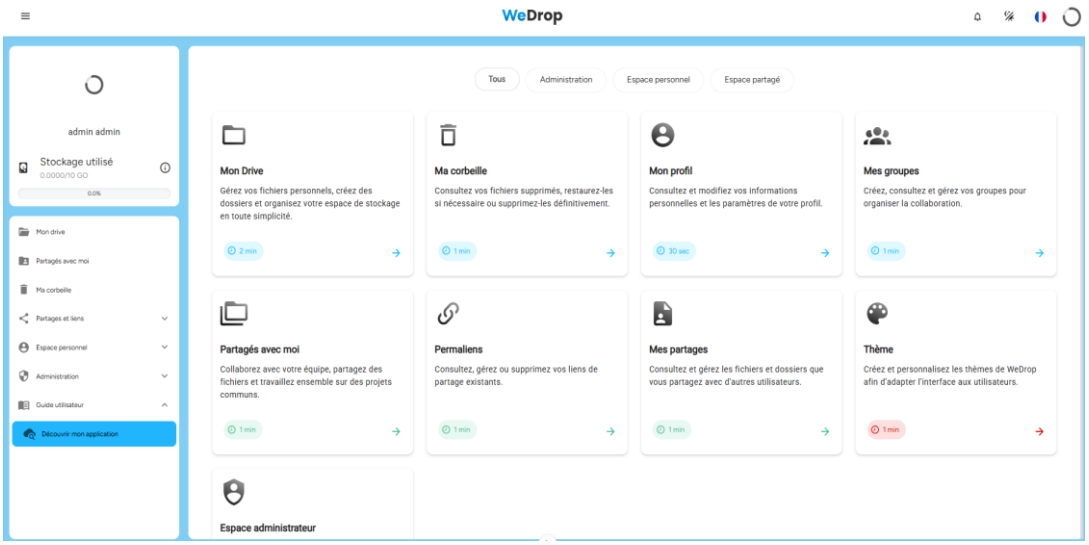

Figure 15 — Tous les guides disponibles

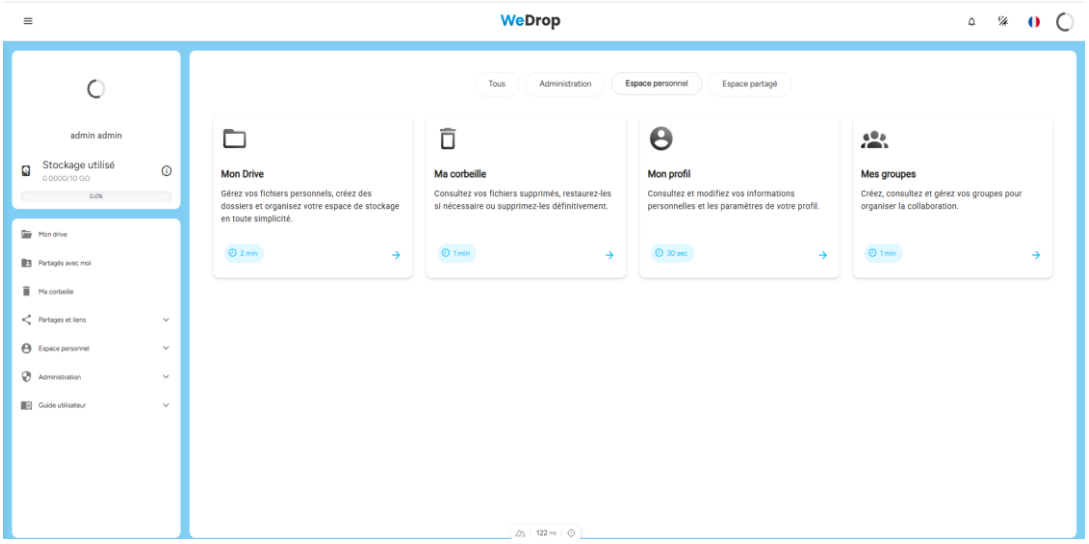

Figure 16 — Guides filtrés sur la catégorie Espace personnel

Exemple d'un guide en cours d'utilisation : une petite fenêtre apparaît et guide l'utilisateur étape par étape sur la fonctionnalité concernée, ici la personnalisation du thème visuel de WeDrop :

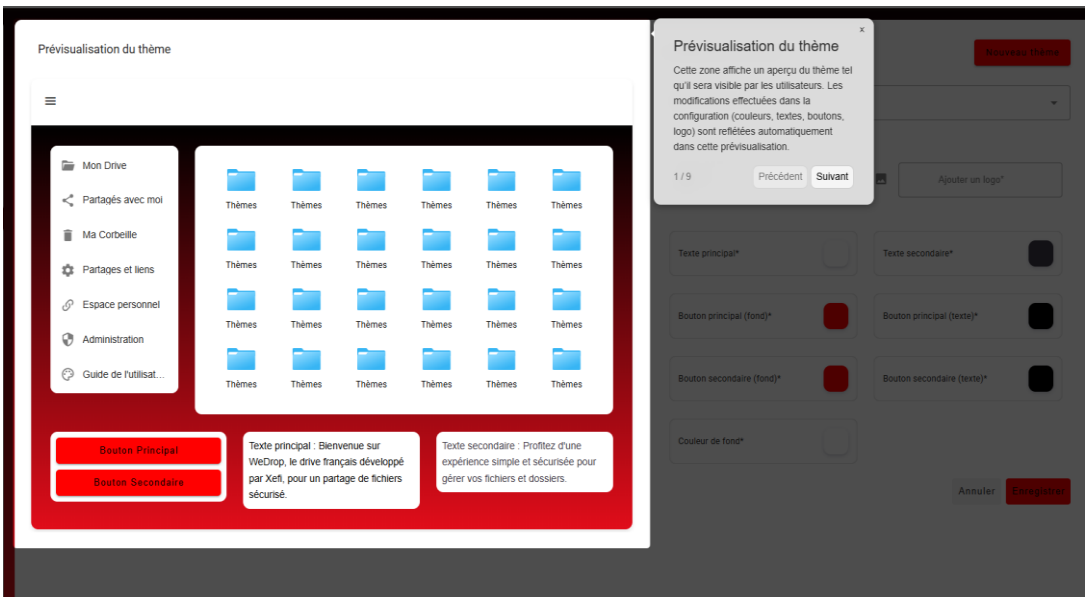

Figure 17 — Guide interactif en cours : l'utilisateur est guidé étape par étape

Les traductions

Chaque guide a été intégralement traduit dans les six langues. Pour cela, un système de fichiers de traduction a été mis en place : chaque langue possède son propre fichier contenant tous les textes. Quand l'utilisateur change la langue de l'application, les textes se mettent à jour instantanément, sans rechargement de la page.

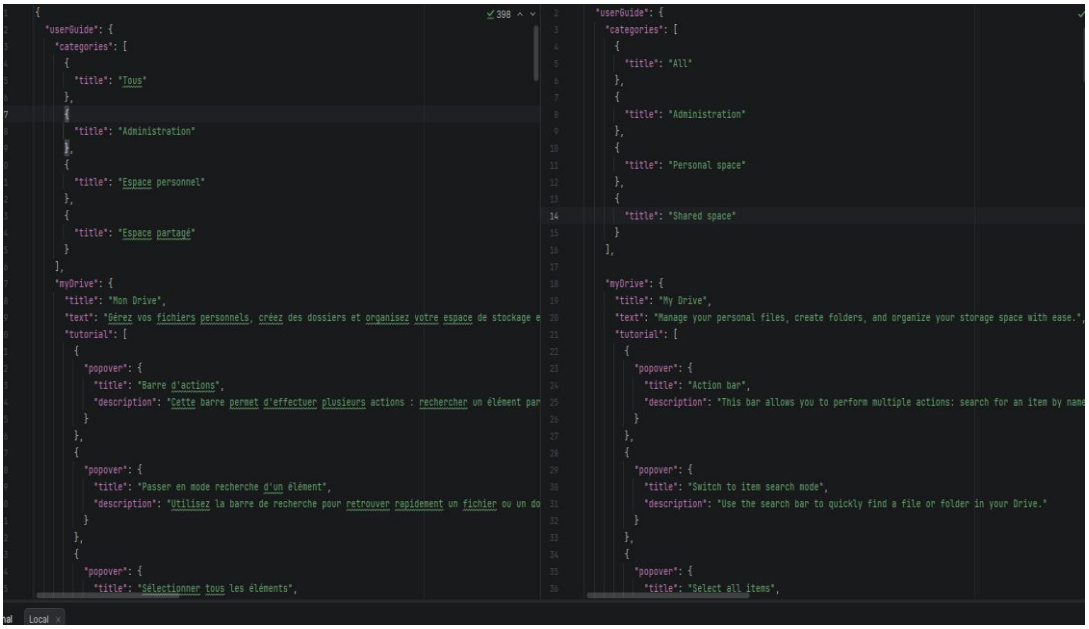

Figure 18 — Fichiers de traduction côte à côte : français (gauche) et anglais (droite)

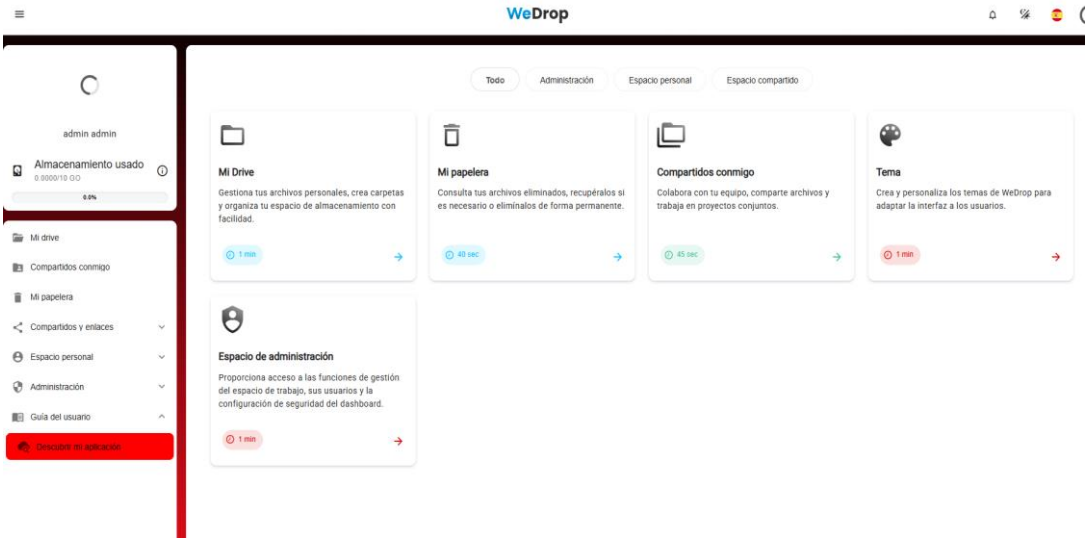

Figure 19 — Interface affichée en espagnol

Figure 20 — Menu de navigation traduit en allemand

Livraison du travail

Une fois le développement terminé, le code a été soumis à l'équipe pour relecture et validation — une étape incontournable en entreprise pour s'assurer de la qualité du travail avant qu'il ne soit intégré au produit final. Deux développeurs seniors ont relu le code, formulé des retours, et ont donné leur accord après corrections.

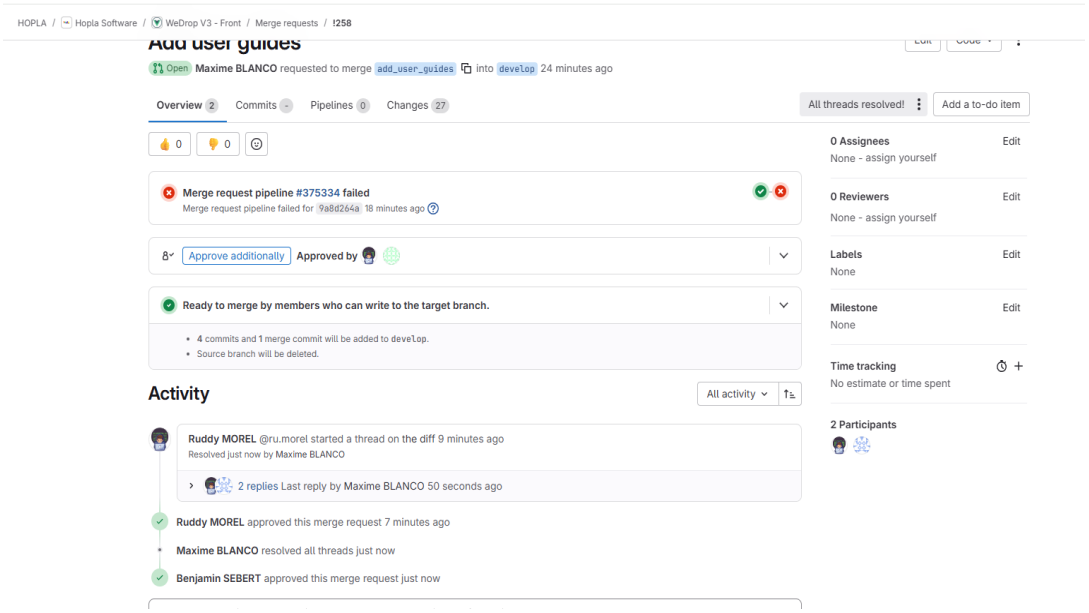

Figure 21 — Validation du travail par l'équipe sur GitLab

4. Difficultés rencontrées

4.1 Côté serveur (Projet 1)

- Comprendre comment relier les différentes informations entre elles dans la base de données
- Mettre en place un système de droits qui empêche chaque type d'utilisateur d'accéder à ce qui ne le concerne pas
- Configurer correctement l'environnement de travail (Linux, Docker, variables de configuration)

4.2 Côté interface (Projets 2, 3 et 4)

- Comprendre comment gérer des informations qui changent en temps réel dans une interface (les crédits, les résultats, la liste des bannis...)
- Faire communiquer correctement l'interface avec le serveur, surtout pour les actions qui prennent du temps
- S'intégrer dans un projet existant déjà développé par une équipe, avec ses propres règles et sa propre organisation
- Gérer 6 langues différentes tout en maintenant une cohérence dans tous les textes de l'application

Ces difficultés ont été surmontées progressivement, grâce à une approche par petits pas, des tests réguliers, et l'aide et les conseils de l'équipe HOPLA.

5. Bilan

5.1 Ce que ce stage m'a apporté

Ce stage a été une expérience très enrichissante, aussi bien techniquement qu'humainement. Avoir travaillé sur un vrai produit utilisé par de vraies entreprises, avec une vraie équipe et de vraies contraintes de délais et de qualité, est quelque chose que la formation seule ne peut pas apporter.

La progression dans les missions — du projet d'entraînement jusqu'à l'intégration dans un produit professionnel — a permis de gagner en autonomie et en confiance.

5.2 Compétences validées

- Construire la partie serveur d'une application web (gestion des données, des utilisateurs, des droits d'accès)
- Créer des interfaces web interactives et modernes
- Intégrer un système de traduction multilingue dans une application existante
- Travailler en équipe avec un outil de gestion de code partagé (GitLab)
- S'adapter à un environnement de travail professionnel

5.3 Ce que je ferais mieux

- Mieux anticiper les problèmes liés aux actions qui s'exécutent en parallèle
- Prendre plus de temps pour lire et comprendre le code existant avant de commencer à modifier
- Améliorer la gestion des données partagées entre différentes parties d'une interface

5.4 Conclusion

Ce stage chez Xefi Grand Lyon au sein de la division HOPLA a pleinement rempli ses objectifs : tester et valider des compétences en développement web, contribuer à un produit réel, et découvrir le fonctionnement d'une équipe de développement professionnelle.

Il confirme mon orientation vers le développement web et m'a donné une base concrète et solide pour la suite de mon parcours.