

RAPPORT DE STAGE



Du 30/09/2024 au 25/10/2024

Joy Huré

Table des matières

RAPPORT DE STAGE.....	1
Table des matières.....	2
Introduction.....	3
Présentation de l'entreprise.....	4
Thanks-Boss : Mission et objectifs	4
Organisation des équipes.....	5
Infrastructure technique.....	6
Missions et tâches réalisées	7
Organisation et gestion des tâches via Rock.....	7
Création des pages de prévisualisation pour les recruteurs.....	9
Implémentation	11
Implémentation des pages « profil » et « prévisualisation » du recruteur.....	11
Processus de collaboration et gestion de la branche.....	15
Implémentation des pages de navigation vers les candidatures pour les recruteurs	18
Création de jobs de test en préproduction	20
Test des pages en production.....	21
La page Alerte	21
La page Offre d'emploi (parcours candidat).....	23
Compétences acquises.....	25
Compétences techniques.....	25
Compétences organisationnelles	25
Compétences en travail d'équipe	25
Conclusion	26



Introduction

J'ai eu l'opportunité d'effectuer mon stage au sein de Creadev, une entreprise qui participe au développement de la plateforme **Thanks-Boss**, dédiée au recrutement.

Cette plateforme utilise des technologies avancées, notamment l'intelligence artificielle, pour améliorer le processus de recrutement et faciliter la mise en relation entre recruteurs et candidats.

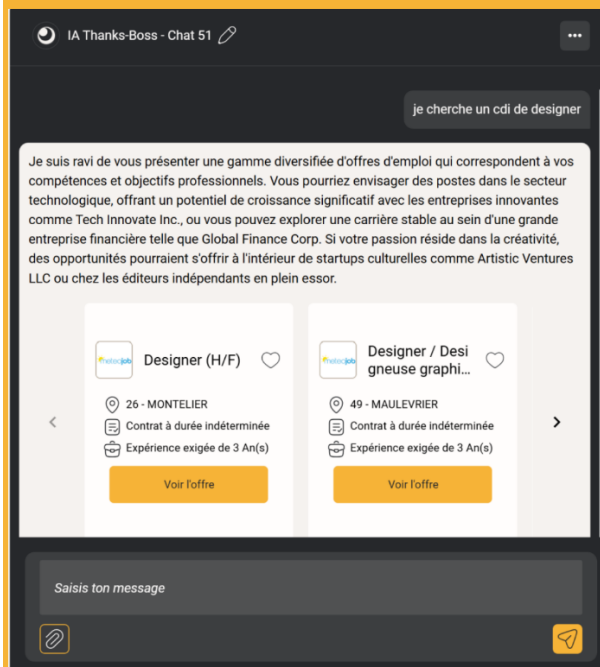
Intégrée à l'équipe frontend, j'ai participé à l'implémentation et à l'amélioration des pages web utilisées par les recruteurs et candidats sur la plateforme.

Mon rôle a consisté à développer des fonctionnalités en **HTML**, **CSS**, **JavaScript**, et **Vue.js**, tout en travaillant en collaboration avec les autres équipes, notamment celles du backend et de l'intelligence artificielle.

Ce stage, réalisé en télétravail, m'a permis de me familiariser avec des outils tels que **GitLab** et **Postman** pour tester et intégrer les nouvelles fonctionnalités dans les environnements de **préproduction** et de **production**.

Libère ton talent et trouve le job de tes rêves avec notre IA !

Chez Thanks-Boss, nous croyons que chaque parcours mérite d'être mis en lumière. Notre plateforme propulsée par l'IA est conçue pour t'accompagner dans ta recherche d'emploi. Connecte-toi aux meilleures opportunités qui te correspondent vraiment. Alors, tu es prêt(e) ?



Accède aux fonctionnalités plus avancées :

- Optimisation de CV
- Optimisation de lettre de motivation
- Aide à la préparation des entretiens





Présentation de l'entreprise

Thanks-Boss : Mission et objectifs

Thanks-Boss est une plateforme numérique spécialisée dans le recrutement et la recherche d'emploi. Son objectif est de simplifier et d'optimiser le processus de recrutement et de recherche d'emploi pour les candidats en utilisant des technologies de correspondance intelligente basées sur l'intelligence artificielle. La plateforme permet aux recruteurs de publier des offres d'emploi et d'évaluer rapidement les candidatures en fonction de critères de compétences et d'expérience. Thanks-Boss se distingue par sa capacité à offrir aux candidats des opportunités correspondant à leurs aspirations, tout en proposant aux recruteurs des profils en adéquation avec les exigences du poste. La mission principale est d'améliorer l'efficacité du processus de recrutement, en réduisant le temps nécessaire à la sélection des meilleurs candidats.

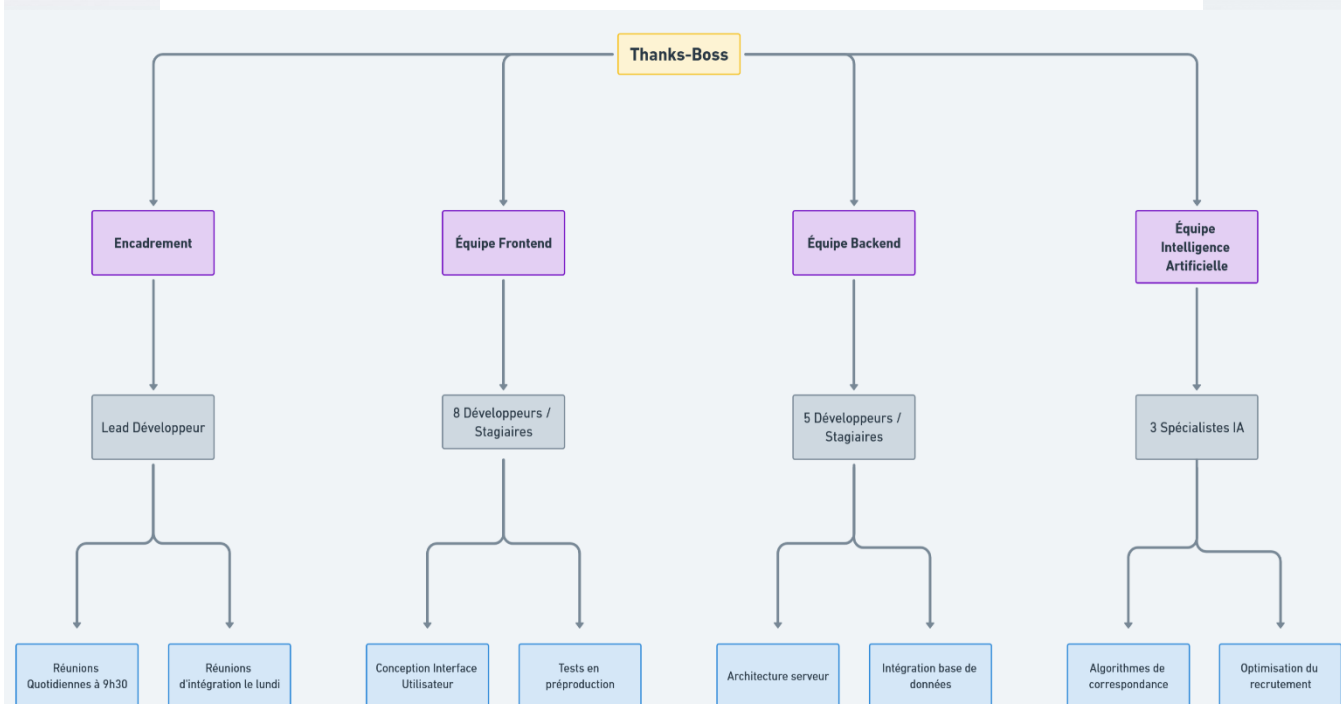
Organisation des équipes

Les équipes qui travaillent sur Thanks-Boss sont divisées en plusieurs pôles :

- **Frontend** : Responsable de la conception et de l'amélioration de l'interface utilisateur de la plateforme.
- **Backend** : Chargé de l'architecture serveur et de l'intégration des fonctionnalités avec la base de données.
- **Intelligence Artificielle** : En charge de développer les algorithmes de correspondance et d'améliorer continuellement la pertinence des résultats.
- **Marketing** : Création des maquettes / réseaux sociaux et communication

Thanks-Boss fonctionne exclusivement en télétravail, avec une réunion quotidienne à 9h30 sur Discord afin de discuter des avancées de chaque équipe, faire le point sur le travail quotidien de chacun et résoudre les éventuels problèmes.

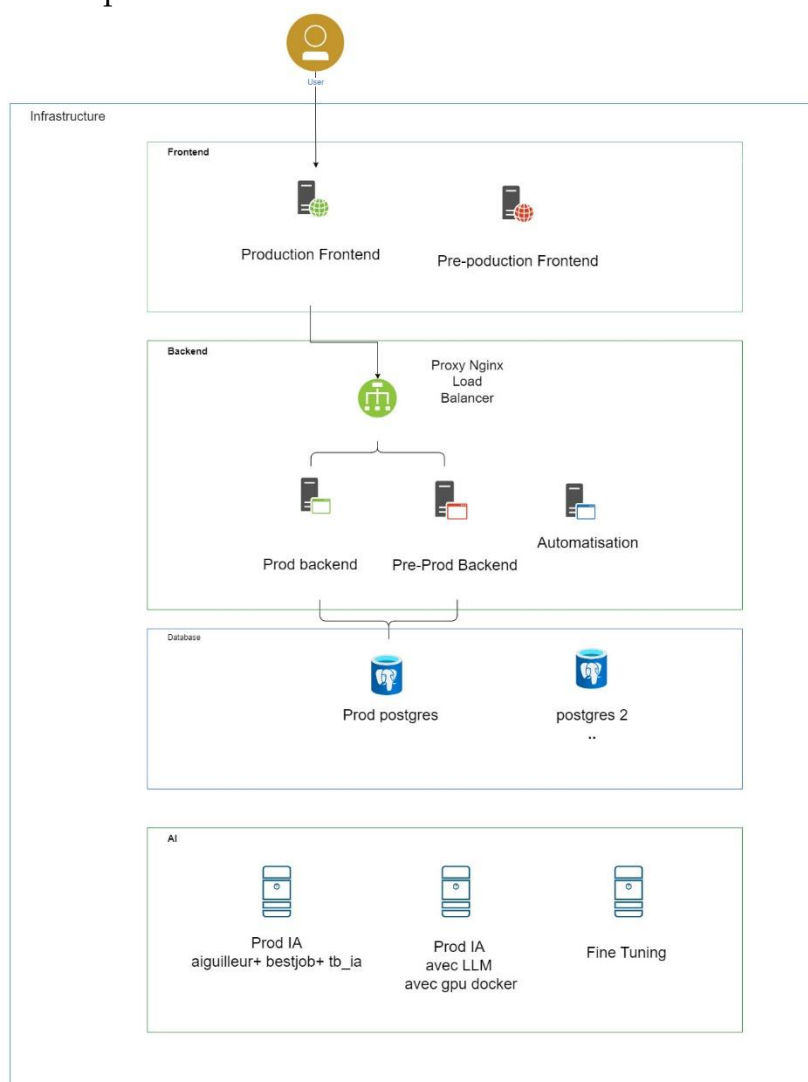
Le lundi matin est dédié à l'intégration des nouveaux stagiaires, facilitant ainsi leur prise en main des outils et des projets.



Infrastructure technique

L'infrastructure technique est divisée entre deux serveurs principaux :

- **Le serveur de production** qui gère l'environnement accessible aux utilisateurs finaux.
- **Le serveur de préproduction** où sont testées les nouvelles fonctionnalités avant leur déploiement en production.



Cette organisation permet aux développeurs de travailler dans des environnements isolés, garantissant ainsi la qualité et la stabilité des fonctionnalités avant leur mise en ligne.

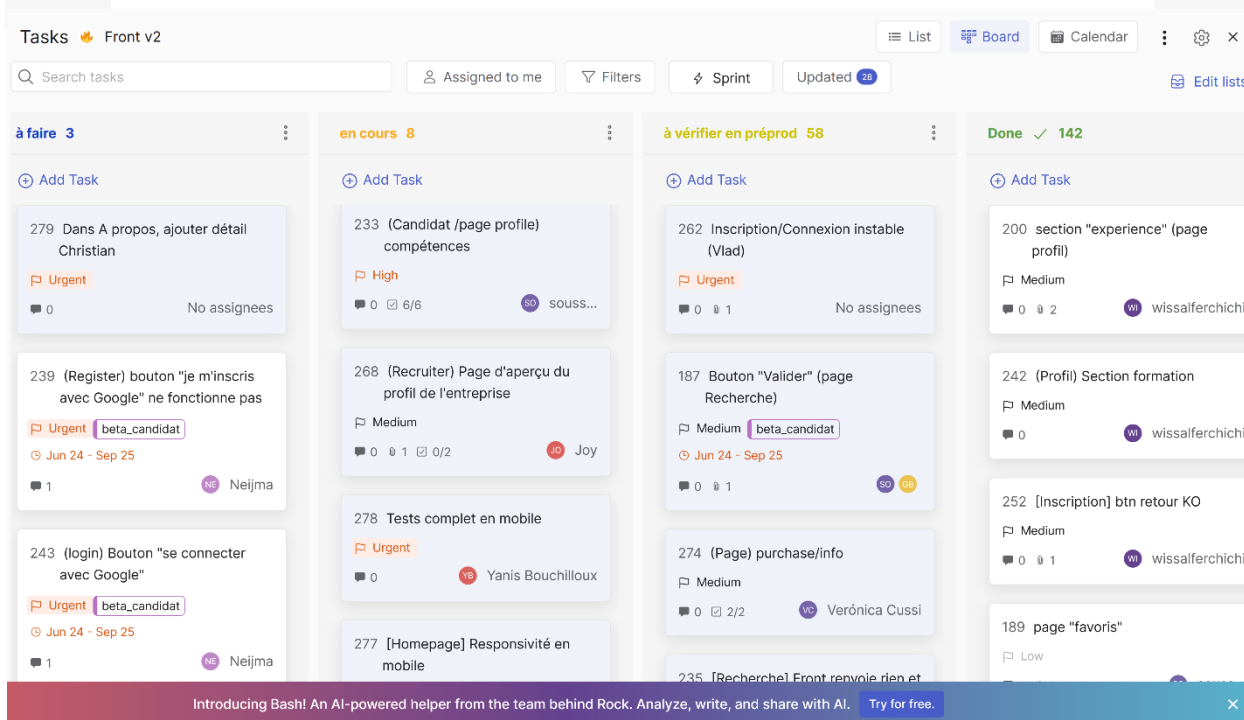
La plateforme repose sur une architecture utilisant un **Proxy Nginx** comme répartiteur de charge, permettant de gérer efficacement le trafic des utilisateurs.

Missions et tâches réalisées

Organisation et gestion des tâches via Rock

Chez **Thanks-Boss**, la gestion des missions et des tâches est centralisée dans un système de tickets sur la plateforme **Rock**.

Chaque mission ou fonctionnalité est enregistrée sous forme de **ticket**, qui représente une tâche à accomplir dans le cadre d'un sprint de développement.

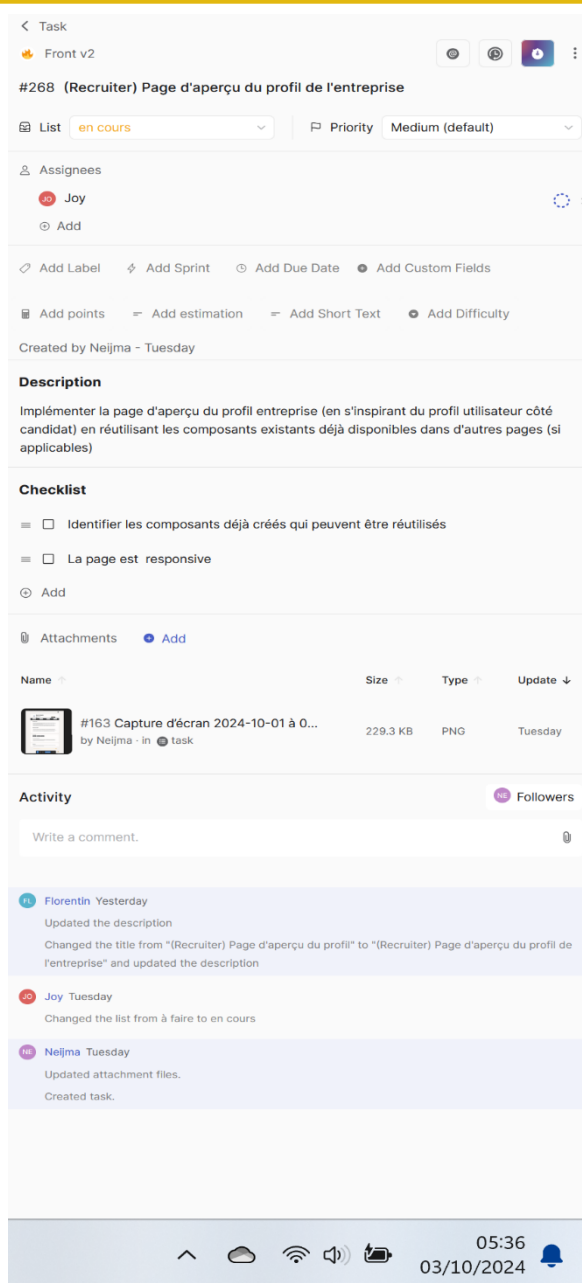


Les tickets sont organisés en plusieurs colonnes :

- **À faire** : Liste des tickets à démarrer, incluant les nouvelles fonctionnalités ou corrections de bugs.
- **En cours** : Tickets actifs sur lesquels les développeurs travaillent actuellement.
- **À vérifier en préprod** : Tickets terminés et en attente de validation sur le serveur de préproduction pour s'assurer de leur stabilité avant déploiement en production.
- **Done** : Tickets finalisés et validés, prêts pour la mise en production ou déjà en production.

Chaque ticket inclut des détails tels que le niveau de priorité (urgent, medium, low), le statut (ex. : à faire, en cours), les personnes assignées, ainsi que les objectifs intermédiaires et les commentaires pour documenter les étapes importantes ou problèmes rencontrés.

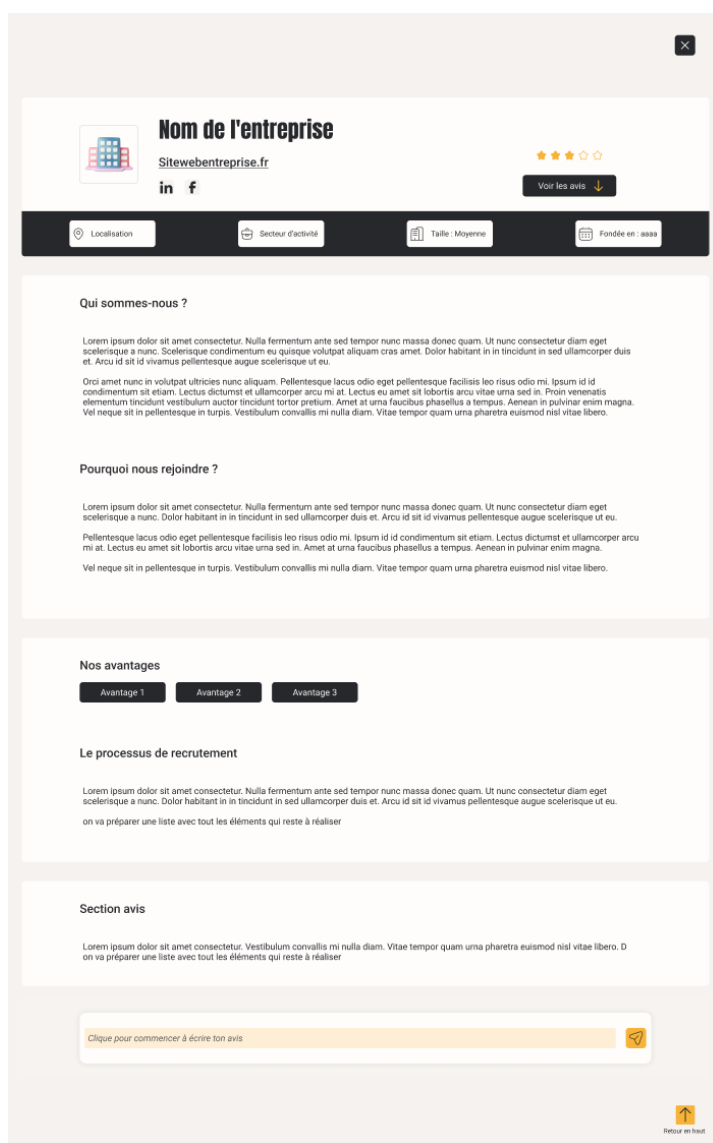
Ce système permet une gestion efficace des tâches et facilite la collaboration entre les membres de l'équipe en offrant une vue d'ensemble des projets et de leur avancement.



Création des pages de prévisualisation pour les recruteurs

Pendant mon stage, j'ai travaillé à la création de deux pages de prévisualisation pour les recruteurs : une page pour le **profil recruteur** et une autre pour les **candidatures**.

Ces pages offrent aux recruteurs un aperçu complet de leur profil et des candidatures reçues, permettant une gestion simplifiée et une évaluation rapide des candidats. En utilisant **Vue.js** et **html** pour la structure, **JavaScript** pour les interactions et le **CSS** pour la mise en forme, j'ai mis en place une interface conforme aux attentes, respectant la mise en page de la maquette fournie et responsive afin de s'adapter aux différents appareils des utilisateurs.



Page
profilPreview

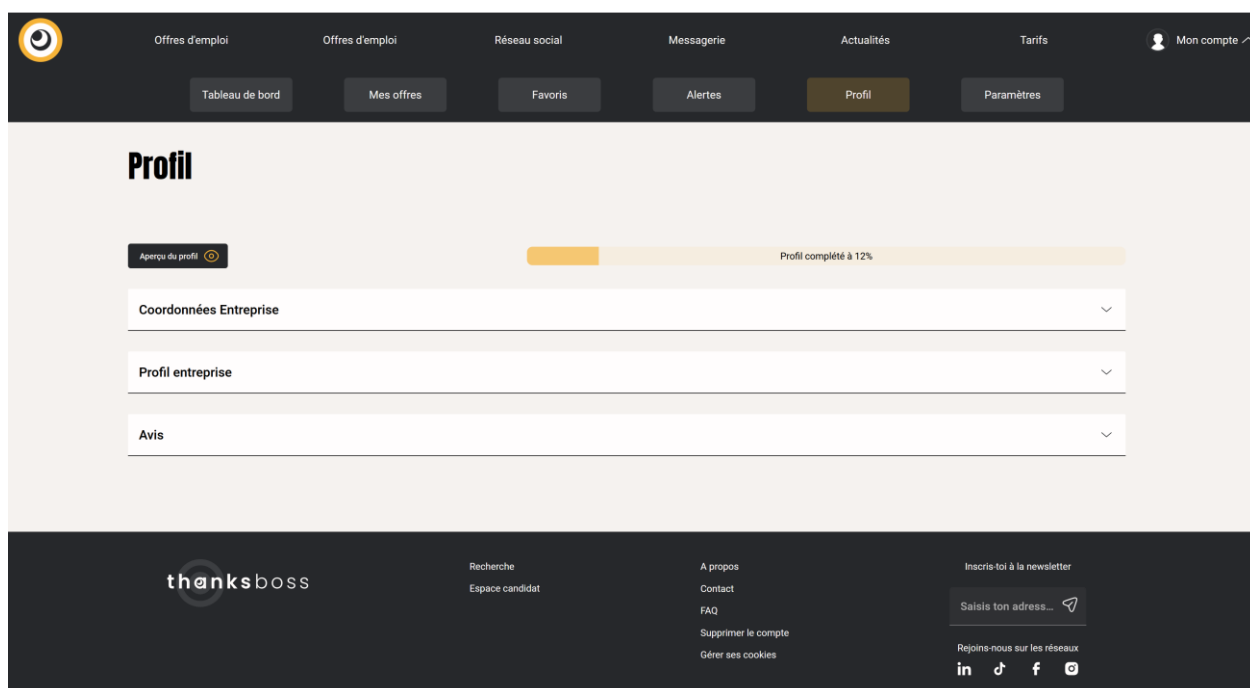


*Page profil-
preview,
simulation de la
visualisation sur
smartphone
grâce à
l'inspecteur du
navigateur*

Implémentation

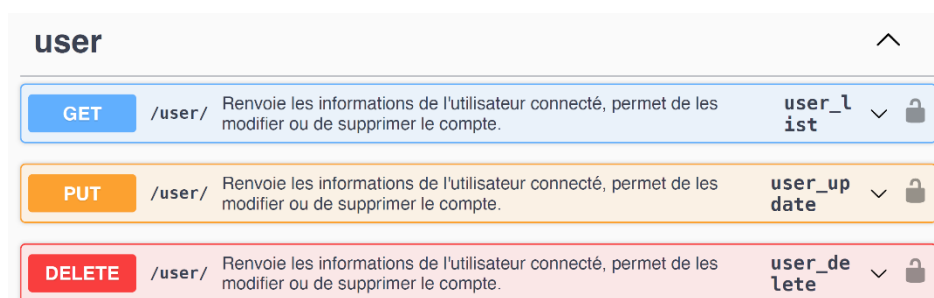
Implémentation des pages « profil » et « prévisualisation » du recruteur

Après avoir créé les pages de prévisualisation, la logique et l'implémentation des pages « profil » et « profil preview » pour les recruteurs devaient être mises en place. Cette mission m'a été assignée en collaboration avec Franck, un autre stagiaire de l'équipe frontend. Pour optimiser notre travail, nous nous sommes répartis les tâches : j'étais en charge de la page profil dont le rôle est de pouvoir récupérer et afficher les données « user » déjà disponibles dans la base de données ainsi que de modifier ces informations en cas de changement par l'utilisateur. Franck s'occupait de la page de prévisualisation qui récupère les données « user » localement, depuis la page profil.



Page profil, menus déroulants clos

Les routes pour ces pages étaient déjà définies et fonctionnelles dans le **Swagger**, ce qui nous a permis de gagner du temps en suivant la logique déjà en place pour le parcours candidat.



Le Swagger et ses routes « user »

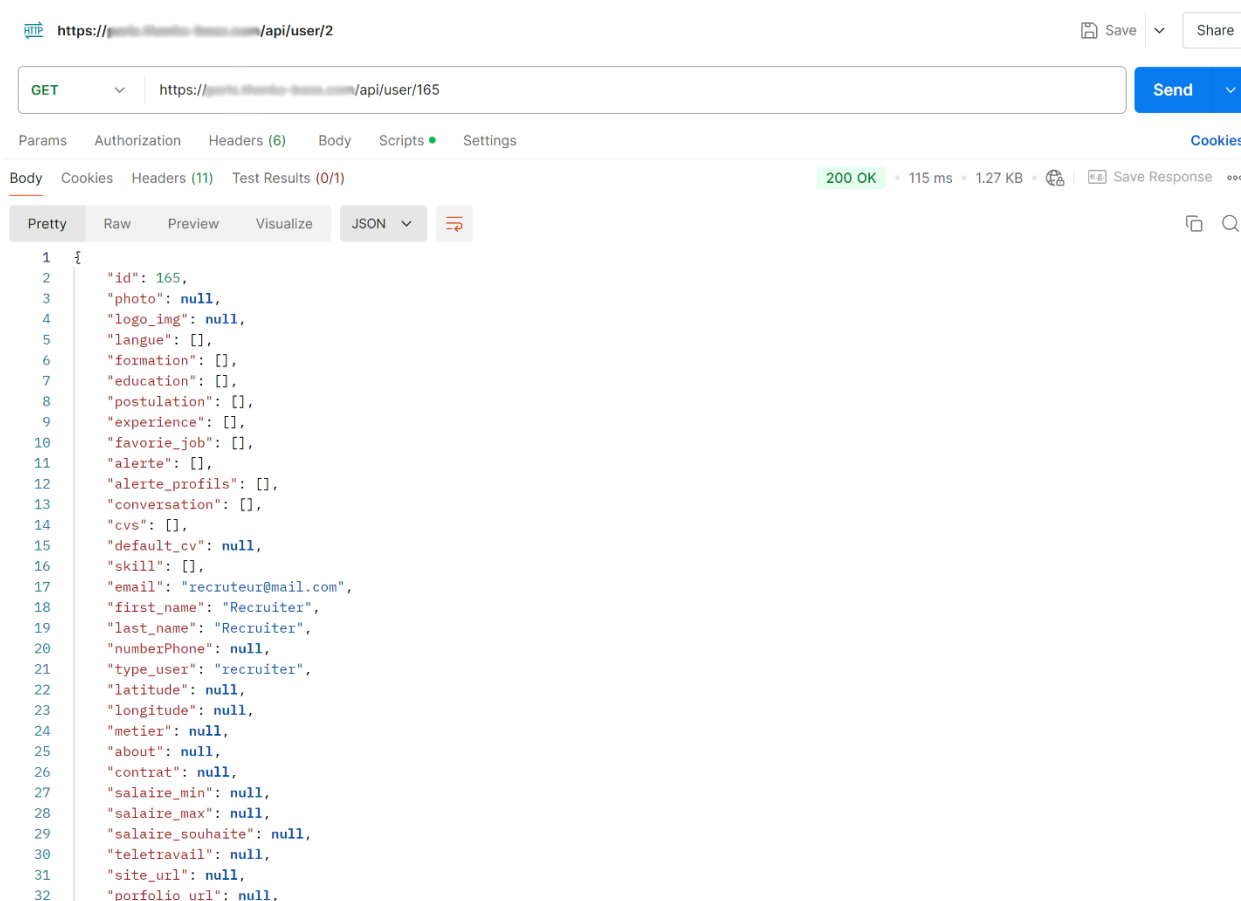
Cependant, dans la structure de la base de données et plus spécifiquement dans la table « user » regroupant les données de tous les utilisateurs, qu'ils soient candidats ou recruteurs, nous avons constaté que certains champs spécifiques aux recruteurs, tels que « *Pourquoi nous rejoindre* » ou « *Avantages* », n'étaient pas encore disponibles.

Il a donc été nécessaire de collaborer avec l'équipe backend pour ajouter ces champs afin de pouvoir les stocker puis les récupérer ultérieurement.

Les champs de texte n'existant pas en base de données : « Année de création », « pourquoi nous rejoindre », « Avantages » et « processus de recrutement »



A ce moment de mon stage, je ne possédais pas les identifiants d'accès à la base de données. Pour consulter les champs de la table « user » et tester les requêtes j'ai donc utilisé Postman avec les requêtes GET et PUT renseignées dans le swagger.



Affichage avec postman et une requête GET du contenu des données de l'user d'id 165, mon compte de test



```

/**
 * Update company information in database, ignoring identical values
 * @param {Object} data - The new company data to update
 * @return {void}
 * @throws {Error}
 */
const updateCompanyProfile = async (data) => {
  try {
    // Récupérer les informations actuelles de l'utilisateur
    let user = await getUser();

    // Créer un nouvel objet pour les informations de l'entreprise à mettre à jour
    let newCompanyProfile = {};

    // Liste des champs à comparer
    const fieldsToCompare = {
      activitySector: 'secteur',
      taille: 'taille',
      whyJoinUs: 'rejoindre',
      foundationYear: 'company_creation_year',
      aboutme: 'about',
      advantages: 'avantages',
      process: 'processus_recrutement',
    };

    // Comparer chaque champ et n'inclure que les champs modifiés
    for (const [dataKey, userKey] of Object.entries(fieldsToCompare)) {
      if (data[dataKey] !== user[userKey]) {
        newCompanyProfile[userKey] = data[dataKey];
      }
    }

    // Afficher les données transmises à la base de données
    console.log('Données Profil transmises à la base de données:', newCompanyProfile);

    // Mettre à jour les informations de l'utilisateur dans la base de données
    await axiosInstance.put("/user/", newCompanyProfile, {
      headers: {
        "Content-Type": "multipart/form-data",
      },
    });

    // Mettre à jour les informations de l'utilisateur dans le store pour une utilisation locale
    user = { ...user, ...newCompanyProfile };
    store.dispatch("handleUserChange", { type: null, payload: user });

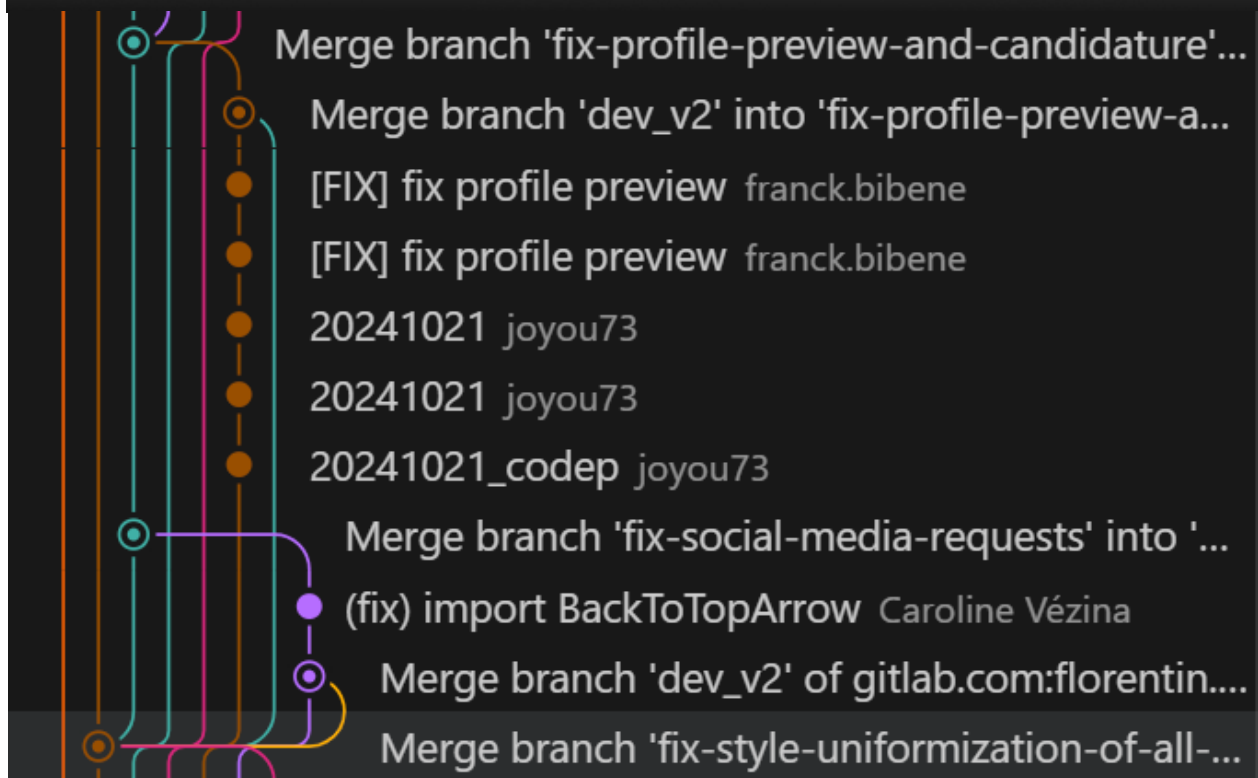
    // Afficher une notification de succès
    toaster.showSuccessPopup("Informations sauvegardées.");
  } catch (error) {
    // Gérer les erreurs
    console.log("Erreur lors de la sauvegarde des données.", error);
    toaster.showErrorPopup("Erreur lors de la sauvegarde des données.");
  }
};

```

Création de la fonction `updateCompanyProfile` dans le fichier `profile.service.js` regroupant toutes les fonctions concernant les données des utilisateurs

Processus de collaboration et gestion de la branche

Pour assurer une bonne gestion de version et faciliter la collaboration, j'ai créé une branche spécifique nommée **fix-profile-preview**. Nous avons adopté un processus de pull/push régulier pour synchroniser notre code, ce qui nous a permis de rester alignés et d'intégrer nos modifications respectives sans conflit. Franck et moi avons travaillé de manière itérative, réalisant des commits pour chaque étape de développement et pullant les modifications de l'autre afin de vérifier la cohérence de nos éléments respectifs. Ce travail d'équipe m'a permis de renforcer mes compétences en gestion de version et d'apprendre à gérer efficacement des branches en collaboration.



Evolution de la branche fix-profile-preview

Dans les cas habituels, lorsque je travaillais seule sur un ticket, je commençais par mettre à jour mon environnement local avec les commandes **git fetch** puis **git pull** depuis la branche principale de développement (dev_v2).

```

● PS C:\Users\joyou\Informatique\thankboss\front_tb_vue> cd v2
● PS C:\Users\joyou\Informatique\thankboss\front_tb_vue\dev_v2> git fetch
● PS C:\Users\joyou\Informatique\thankboss\front_tb_vue\dev_v2> git pull
Updating 40358791..e4a888c0
Fast-forward
 v2/.env | 2 +-
 .../cards/candidate-card/LargeCandidateCard.vue | 6 +-
 v2/src/components/cards/job-cards/LargeJobCard.vue | 6 +-
 v2/src/components/chat/ChatApp.vue | 2 +-
 v2/src/components/layout/navbar/Navbar.vue | 37 +-
 v2/src/components/layout/newsletter/Newsletter.vue | 13 +
 v2/src/components/modal/alert/recruiter/Alert.vue | 11 +-
 .../modal/application/JobApplication.vue | 397 ++++++
 v2/src/components/views-models/blog/HeroBanner.vue | 13 +-
 .../dashboard/GraphSectionRecruiter.vue | 79 +-
 .../views-models/dashboard/RecentJobs.vue | 39 +-
 .../views-models/dashboard/SearchProfile.vue | 32 +-
 v2/src/components/views-models/home/AboutUs.vue | 2 +-
 v2/src/components/views-models/home/Feedbacks.vue | 1 -
 .../components/views-models/home/HeroSection.vue | 33 +-
 .../components/views-models/home/HowDoesItWork.vue | 4 +-
 v2/src/components/views-models/home/OurValues.vue | 2 +-
 .../recruiter/profil/MyProfileEntreprise.vue | 6 +-
 .../recruiter/subscription-page/Facturation.vue | 37 +
 v2/src/components/views-models/search/Alert.vue | 87 +-
 .../views-models/social-media/createPost.vue | 6 +-
 v2/src/router/index.js | 7 +-
 v2/src/services/alert.service.js | 8 +-
 v2/src/services/job.service.js | 28 +-
 v2/src/services/post.service.js | 87 +-
 v2/src/services/profile.service.js | 2 +-
 v2/src/views/all/article/Article.vue | 2 +-
 v2/src/views/all/blog-section/BlogSection.vue | 41 +
 v2/src/views/all/cgu/CGU.vue | 33 +-
 v2/src/views/all/cgv/CGV.vue | 45 +-
 .../views/all/mentions-legales/MentionsLegales.vue | 32 +-
 .../PolitiqueConfidentialite.vue | 32 +-
 v2/src/views/all/social-media/Reseaux.vue | 231 +++--
 v2/src/views/candidate/contact/Contact.vue | 4 +-
 v2/src/views/candidate/faq/Faq.vue | 28 +-
 v2/src/views/candidate/joboffer/Joboffer.vue | 74 +-
 v2/src/views/candidate/messaging/Messaging.vue | 350 +++---
 v2/src/views/candidate/search/Search.vue | 20 +
 v2/src/views/candidate/settings/Settings.vue | 10 +-
 .../candidate/subscriptions/Subscriptions.vue | 58 +-
 v2/src/views/recruiter/alerts/Alerts.vue | 92 +-
 v2/src/views/recruiter/all-offer/Offer.vue | 164 ++-
 .../recruiter/candidate-page/CandidatePage.vue | 134 ++-
 v2/src/views/recruiter/dashboard/Dashboard.vue | 53 +-
 v2/src/views/recruiter/messaging/Messaging 2.vue | 1044 -----
 v2/src/views/recruiter/messaging/Messaging.vue | 357 +++---
 v2/src/views/recruiter/new-offer/NewOffer.vue | 413 +++---
 .../views/recruiter/new-offer/PreviewNewOffer.vue | 34 +-
 v2/src/views/recruiter/prices/Prices.vue | 67 +-
 v2/src/views/recruiter/profil/Profil.vue | 86 +-
 .../profilpreview/RecruiterProfilPreview.vue | 54 +-
 v2/src/views/recruiter/search/Search.vue | 11 +-
52 files changed, 2327 insertions(+), 2089 deletions(-)
create mode 100644 v2/src/components/modal/application/JobApplication.vue
delete mode 100644 v2/src/views/recruiter/messaging/Messaging 2.vue


```

Mise à jour de l'environnement local dans le terminal de VSCode

Ensuite, je créais une nouvelle branche avec un préfixe indiquant la nature de la tâche : **fix-** pour les corrections de bugs ou **feat-** pour les nouvelles fonctionnalités. Cela permet de garder une bonne organisation des branches et de faciliter le suivi des modifications. Une fois les modifications terminées, je soumettais une **merge request** pour intégration, qui était ensuite validée par un développeur plus expérimenté.



Florentin / front_tb_vue / Requêtes de fusion / 1412



Fix profile preview and candidature


 Joy Huré a demandé de fusionner `fix-profile-preview-and-ca...` vers `dev_v2` il y a 2 minutes

Vue d'ensemble 0 Validations 11 Pipelines 0 Modifications 12


 0  0 

 **Afin que vous puissiez exécuter des pipelines, nous devons vérifier votre compte.** 
Nous ne vous demanderons plus ces informations. Elles ne seront jamais utilisées à des fins de marketing.
[Vérifier mon compte](#)

 **Il semble ne pas y avoir de pipeline ici.** 
GitLab CI/CD peut construire, tester et déployer automatiquement votre application. Il ne vous faudra que quelques minutes pour vous lancer et nous pouvons vous aider à créer un fichier de configuration de pipeline.
[Essayez les Pipelines GitLab](#)

8  L'approbation est facultative 

 **Prêt pour la fusion !** 

☐ Supprimer la branche source ☐ Combiner (squash) les commits  ☐ Modifier le message de validation

11 validations et 1 validation de fusion seront ajoutées à dev_v2.

[Fusionner](#)

Demande de fusion de la branche fix-profil-preview avec dev_v2, en attente d'approbation



Implémentation des pages de navigation vers les candidatures pour les recruteurs

Ma dernière mission a été de rendre fonctionnelle la consultation d'une candidature sur une offre d'emploi spécifique par le recruteur. L'objectif de cette page « candidature » est de permettre aux recruteurs d'accepter ou de refuser une candidature reçue. De nombreuses étapes étaient nécessaires pour y parvenir.

The screenshot displays a web interface for managing job applications. At the top left is a 'Retour' button with a left arrow. The main content area is divided into two sections. The top section, titled 'Intitulé du poste' with a building icon, shows 'Chez Nom entreprise'. Below this are four fields: 'Localisation' (with a location pin icon), 'Type de contrat' (with a document icon), 'Type de travail' (with a calendar icon), and 'Réf : AB00050' (with a document icon). Below these fields are two buttons: 'Refuser la candidature' (dark grey) and 'Contacter le candidat' (orange). The bottom section, titled 'Nom, Prénom candidat' with a person icon, shows candidate details. It includes an 'E-mail' field with 'monemail@thanks-boss.comm', a 'Téléphone' field with '0X XX XX XX XX', and a 'Localisation' field with 'Ville'. A heart icon is visible in the top right corner of the candidate section.

Le haut de la page candidature, permettant de refuser la candidature ou de contacter le candidat

La première étape impliquait la page « newOffer », où le recruteur peut créer une nouvelle offre d'emploi. Bien que cette page ne soit pas explicitement mentionnée dans ce ticket, elle s'est avérée indispensable pour alimenter la page « offer » des candidats qui leur permet de visualiser puis de postuler aux offres disponibles.

Une fois la candidature soumise, le recruteur peut enfin accéder à ses détails si toutefois il est le créateur du poste. C'est à cette étape que mon stage a pris fin, je n'ai malheureusement pas eu la possibilité de réaliser les étapes suivantes de cette mission que je vais néanmoins présenter.

Ensuite, il aurait fallu implémenter la page « offer » puis la page « offer-application ». Depuis la page « offer », les recruteurs peuvent cliquer sur l'offre de leur choix pour afficher la liste des candidatures soumises. Enfin depuis la page « offer-application »,

chaque candidature est sélectionnable, permettant au recruteur d'accéder à une page de consultation de candidature individuelle : la page « candidature ».

Ce ticket n'a pas été finalisé, car des problèmes de base de données ont ralenti l'avancement.

En effet, certains champs nécessaires pour enregistrer les informations d'un job étaient absents en base de données et d'autres ne s'enregistraient pas, tel que le champ « author » pourtant indispensable, rendant impossible l'enregistrement complet des offres et par conséquent leur restitution ultérieure.

```
{
  "id": 2096,
  "author": null,
  "question": [],
  "savoir": [
    {
      "libelle": "Habillage / déshabillage du patient",
      "id": 5334
    },
    {
      "libelle": "Identification des signes de maltraitance",
      "id": 5335
    },
    {
      "libelle": "Règles d'hygiène et de propreté",
      "id": 5336
    },
    {
      "libelle": "Accompagner une personne lors d'un déplacement",
      "id": 5337
    }
  ]
}
```

Affichage des données d'un job crée dans Postman, le champ author est « null », le job en fonction de l'auteur n'est pas récupérable

Pour résoudre cette situation, l'équipe backend a dû réinitialiser la base de données en y ajoutant les éléments manquants. Cependant, cette opération a occasionné des instabilités : les données n'étaient pas toujours enregistrées correctement, et la synchronisation avec les pages frontend était parfois compromise, ce qui a perturbé le fonctionnement de l'application toute entière. La réinitialisation et les correctifs ont partiellement restauré la fonctionnalité, mais de nouveaux tests et ajustements étaient encore nécessaires pour assurer la fiabilité de ce processus.

Création de jobs de test en préproduction

Dans le cadre de l'optimisation et des tests de la plateforme Thanks-Boss, j'ai été chargée de créer plusieurs offres d'emploi de test sur l'environnement de préproduction et dans la continuité du ticket concernant l'implémentation de la page candidature (voir section précédente).

J'ai pu créer 5 offres d'emploi malgré les difficultés mentionnées précédemment.

```
"results": [
  {
    "id": 459233,
    "author": 165,
    "question": [],
    "savoir": [],
    "savoir_pro": [],
    "expertise": [],
    "permis": [],
    "langue_recruteur": [],
    "formation_recruteur": [],
    "is_favorite": false,
    "external_id": null,
    "created_at": null,
    "updated_at": null,
    "title": "Dev FS",
    "nom_recruteur": "Company Name",
    "recruiter_url": null,
    "entreprise": null,
    "descriptif": "Create cool stuff",
    "keywords": null,
    "company": null,
    "logo_url": "",
    "url": null,
    "industrie": null,
    "jours_teletravail": 0.0,
    "contract": "CDI",
    "qualification": null,
```

*Enregistrement du champ « author » en BDD,
le job crée est enfin exploitable*

Ces offres fictives avaient pour objectif de :

- simuler des situations réelles de candidature pour tester l'intégration des fonctionnalités liées au parcours de recrutement (partiellement fonctionnelles à cet instant)
- vérifier que postuler à un job est fonctionnel côté candidat (plus urgent)
- pouvoir tester la nouvelle fonctionnalité d'import des jobs de France Travail qui tous les jours importe de nouveaux jobs, supprime les jobs obsolètes... Mais ne doit avoir aucune action sur les jobs créés dans l'application.

Cette phase est essentielle pour identifier des ajustements nécessaires avant le passage en production.

Test des pages en production

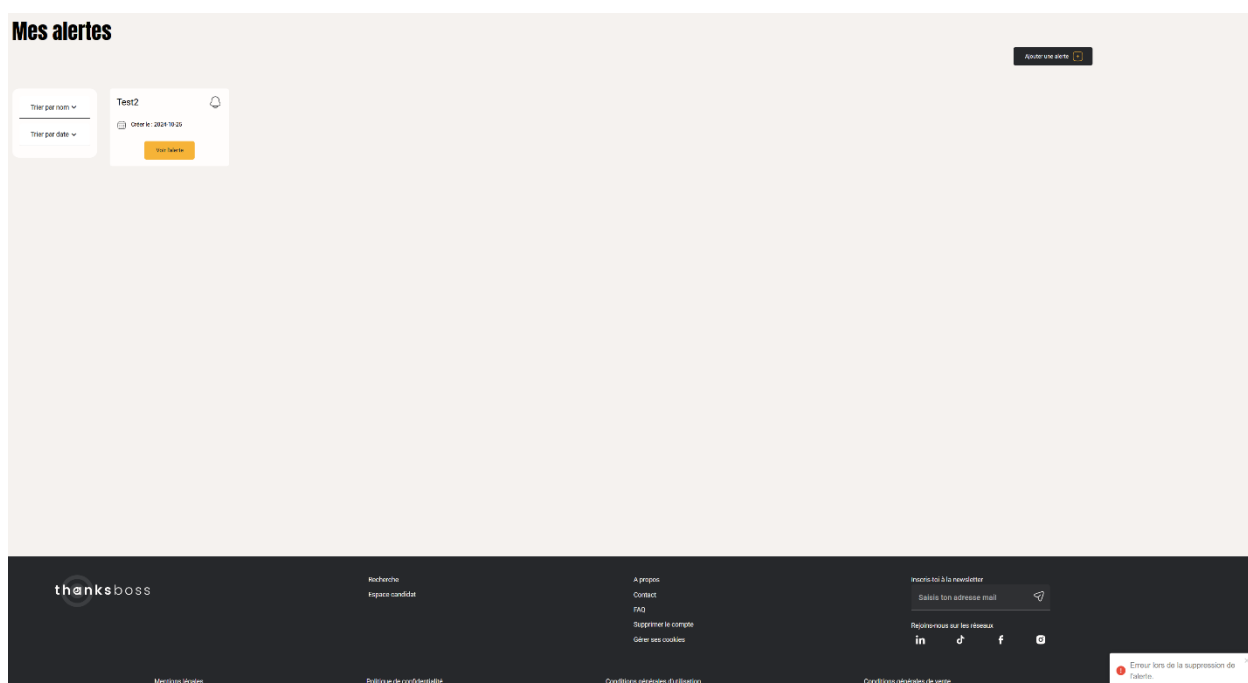
Suite aux problèmes rencontrés sur le serveur de préproduction, notamment la nécessité de réinitialiser la base de données en raison de champs manquants et d'autres dysfonctionnements, l'environnement de préproduction est devenu temporairement inexploitable, paralysant l'activité des développeurs Frontend.

Durant cette période, il a été décidé de réaliser des tests directement sur le serveur de **production** pour certaines fonctionnalités critiques, afin de garantir la stabilité de l'expérience utilisateur.

Il m'a été confié la responsabilité des pages **Alerte** et **Offres d'emploi** du côté candidat.

À partir de ces tests en conditions réelles, plusieurs anomalies ont été détectées sur ces pages. J'ai documenté ces anomalies et créé deux tickets pour chaque ensemble de corrections à apporter.

La page Alerte



Echec de désactivation de l'alerte, l'icône de cloche se désactive mais le toaster d'erreur apparaît

Ticket #340 : Corrections pages « Alerte » côté candidat :

- **Problème de tri** : Il était impossible de décocher un critère de tri une fois sélectionné. Lorsqu'un utilisateur sélectionnait deux tris, le dernier sélectionné prenait cependant le dessus, permettant le tri mais impactant la clarté et l'expérience utilisateur.
- **Problème de suppression et mise à jour** : La suppression et la mise à jour des alertes échouaient une fois sur deux, introduisant une instabilité dans la gestion

des préférences pour les utilisateurs. Je n'ai pas réussi à déterminer le facteur déclenchant

l'erreur, ayant cependant reproduit la manipulation plusieurs fois avec un enchaînement d'actions différent (désactiver / activer l'alerte avant suppression, connexion puis suppression directement sur la page des alertes, créer l'alerte puis la supprimer...)

- **Champ incorrect dans la page de création d'alerte** : Le champ « poste » se présentait comme une zone de saisie libre remplaçant le menu déroulant prévu par la maquette, ce qui créait une incohérence et une expérience utilisateur confuse.

< Task

🔥 Front v2

#340 Corrections pages Alerte côté candidat

List bugs Priority Medium (default)

👤 Add Assignees
 ✎ Add Label
 ⚡ Add Sprint
 ⌚ Add Due Date

➕ Add Custom Fields
 📊 Add points
 = Add estimation
 = Add Short Text

🔴 Add Difficulty

Created by Joy - Oct 25

Description

Correction à apporter suite aux tests

Checklist

- ☐ Impossible de décocher un tri lorsqu'il a été coché, lorsque les deux tris sont sélectionnés, il se fait selon la dernière case cochée
- ☐ La suppression et la MàJ de l'alerte échouent une fois sur deux
- ☐ Dans la page de création de l'alerte, le champ "poste" est une saisie libre et non un choix déroulant

⊕ Add

📎 Add Attachment

Activity JO Followers

Ticket crée afin de corriger les bugs constatés sur la page « alerte »

La page Offre d'emploi (parcours candidat)

The screenshot shows a job search interface. On the left is a sidebar with filters. The 'Type de contrat' filter is set to 'CDD'. The main area displays a grid of job listings. Each listing includes a job title, location, contract type, and a 'Voir l'offre' button. The jobs listed are: Chef de projets immobiliers (F/H) (H/F), Chef de Rang (H/F), Manutentionnaire (H/F), Moniteur / Monitrice d'auto-école (H/F), Commis de Pâtisserie (H/F), Commis / Commise de cuisine (H/F), Assistant.e CATEGORY MANAGER H/F (H/F), EMPLOYE D'HOTEL POLYVALENT (H/F), and RESPONSABLE HEBERGEMENT -...

Le filtre « CDD » affiche un résultat non filtré

Ticket #338 : Corrections page Offres d'emploi côté candidat

- **Recherche non intuitive** : Lorsqu'une recherche était lancée sans renseigner le champ « poste », un toaster d'erreur s'affichait, mais la recherche se lançait malgré tout. De plus, quand l'emploi avait été saisi puis une recherche effectuée, il était impossible de réinitialiser la vue à son état d'origine, ce qui bloquait l'utilisateur dans une vue filtrée.
- **Filtres du menu de gauche inactifs** : Les filtres du menu situé à gauche de la page avaient une action imprévisible mais non escomptée sur le résultat de la recherche

L'accès aux offres d'emploi étant disponible aux candidats connectés et aux utilisateurs non connectés, j'ai réalisé ces tests dans ces deux contextes. Le résultat final était identique.



Task

Front v2

@

🕒

#338 Corrections page Offres d'emploi côté candidat

List

à faire

▼

Priority

Medium (default)

▼

👤 Add Assignees

🏷️ Add Label

⚡ Add Sprint

🕒 Add Due Date

⚙️ Add Custom Fields

📅 Add points

⚖️ Add estimation

⚖️ Add Short Text

🔍 Add Difficulty

Created by Joy - 6 mins ago

Description

Corrections page Offres d'emploi côté candidat suite aux tests réalisés connecté et non connecté en tant que candidat.

Checklist

☐ Recherche sans l'emploi affiche de saisir l'emploi mais lance quand même la recherche, cependant impossible de réinitialiser à la vue d'origine une fois qu'un emploi à été saisi et une recherche lancée

☐ Les filtres à gauche ne fonctionnent pas

+

Add

📎 Add Attachment

Activity

JO Followers

Write a comment.

📎

JO Joy 6 mins ago

Created task.

Ticket créé afin de corriger les bugs constatés sur la page « offres d'emploi »

Ces tickets ont permis de documenter précisément les bugs constatés et de prioriser les ajustements pour garantir une expérience utilisateur cohérente et fonctionnelle. Cette période de test en production m'a permis de développer mes compétences en **gestion de bugs** et en **validation fonctionnelle** dans un environnement de production, tout en renforçant ma capacité à anticiper et documenter les problèmes de manière exhaustive.

Compétences acquises

Mon stage chez Thanks-Boss m'a permis de développer un éventail de compétences dans plusieurs domaines, allant de la programmation frontend aux pratiques de collaboration en équipe. Voici un aperçu des principales compétences acquises au cours de ces missions.

Compétences techniques

- **Découverte de Vue.js, un framework frontend** : En travaillant sur la création et l'implémentation de pages, j'ai découvert Vue.js et le fonctionnement d'un framework frontend, notamment en structurant des composants, en utilisant des directives, et en optimisant les interactions utilisateur.
- **Gestion de version avec Git** : J'ai appris à gérer les branches de développement, en utilisant des préfixes spécifiques (**fix-** pour les corrections de bugs et **feat-** pour les nouvelles fonctionnalités) et en synchronisant régulièrement les modifications via des pulls et des pushes.
- **Tests et validation avec Postman** : En testant les requêtes API et en vérifiant la compatibilité entre le frontend et le backend, j'ai renforcé mes compétences en tests fonctionnels, en utilisant Postman pour simuler et vérifier les échanges de données.
- **Identification et correction de bugs** : Grâce aux tests réalisés sur les serveurs de préproduction et production, j'ai développé ma capacité à identifier et documenter les bugs, à analyser leur origine, et à collaborer avec l'équipe pour apporter des corrections.

Compétences organisationnelles

- **Gestion de tickets et priorisation** : En utilisant le système de tickets sur Rock, j'ai appris à organiser mes tâches de manière structurée, à documenter les étapes de développement, et à prioriser les tickets selon leur impact sur la plateforme.
- **Collaboration en environnement distribué** : Le télétravail étant la norme dans l'équipe, j'ai acquis des compétences en communication et en gestion des interactions à distance, notamment en participant aux réunions quotidiennes et en échangeant régulièrement sur l'avancement des tickets.

Compétences en travail d'équipe

- **Collaboration sur des branches partagées** : Mon travail en tandem avec un autre stagiaire, Franck, sur la création des pages profil et preview m'a permis d'expérimenter la collaboration directe sur des branches Git partagées, en réalisant des merges réguliers pour synchroniser nos travaux.
- **Revue de code et validation** : Chaque merge request nécessitait la validation par un développeur plus expérimenté, ce qui m'a permis de comprendre l'importance des bonnes pratiques de revue de code.

Conclusion

Mon stage au sein de Thanks-Boss a été une expérience enrichissante et formatrice, qui m'a permis de consolider mes compétences techniques tout en découvrant les réalités du travail en équipe dans un environnement de développement d'application web.

Travailler au sein de l'équipe frontend m'a offert l'opportunité de participer à des projets concrets, d'explorer des technologies modernes comme **Vue.js**, et d'interagir avec des systèmes complexes via des outils de gestion et de versionnage tels que **Git** et **Postman**.

Il a représenté une étape décisive dans mon parcours professionnel, m'offrant une immersion concrète dans le monde du développement web. J'ai découvert l'ampleur des compétences requises pour mener à bien un tel projet, allant bien au-delà des connaissances théoriques acquises en formation. Chaque mission m'a confrontée à des défis techniques qui m'ont permis de mesurer l'importance de la polyvalence et de l'apprentissage continu dans ce domaine.

La compétence que j'ai continuellement sollicitée est l'autonomie. J'ai souvent dû chercher des solutions par moi-même, que ce soit pour surmonter des difficultés techniques ou pour trouver des approches optimisées. Les outils d'intelligence artificielle, tels que **ChatGPT** et **Copilot**, ont été une aide précieuse, me permettant d'accélérer mes recherches, d'obtenir des pistes de solution et d'enrichir ma compréhension.

Cette autonomie, renforcée par les échanges réguliers avec l'équipe, m'a appris à être plus proactive et à ne pas hésiter à explorer différentes ressources pour atteindre mes objectifs.

Travailler dans ce cadre a été à la fois stimulant et motivant. J'ai particulièrement apprécié cette manière de travailler, où la recherche personnelle et le soutien collaboratif se combinent pour favoriser un apprentissage constant et un développement des compétences. Cette expérience m'a non seulement renforcée dans ma décision de poursuivre dans le développement web, mais elle m'a aussi montré l'importance d'adopter une attitude de curiosité et de rigueur, des qualités essentielles pour évoluer dans ce domaine.

En somme, ce stage m'a offert une vision complète et réaliste des exigences du métier, tout en m'encourageant à poursuivre avec confiance et motivation sur cette voie professionnelle.