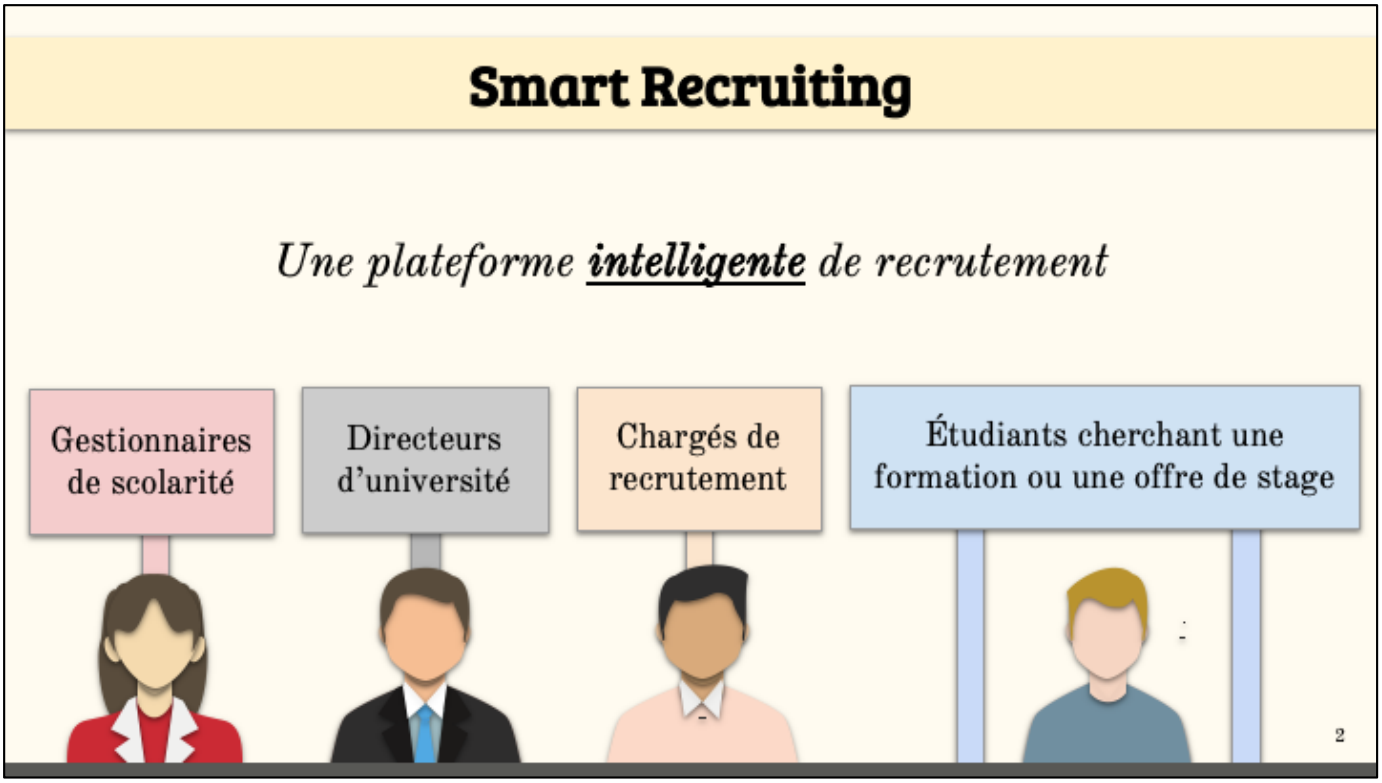
Bonjour nous sommes le groupe de projet SmartRecruiting. Nous allons vous parler du projet sur lequel nous avons travaillé pendant 9 semaines. Ce projet est en lien avec l'entreprise Disrupt Campus qui a pour but de répondre aux besoins liés à l'université.
Il faut savoir que nous sommes la quatrième équipe à participer au développement de ce projet, avant nous, c'était une équipe ENSIMAG/GEM.

Durant cette présentation, nous allons vous expliquer le but du projet, nos missions données, les techniques utilisées, l'architecture du projet, nos réalisations techniques.
On va également vous expliquer notre gestion du projet, les métriques logicielles et pour finir vous faire une démonstration de ce que nous avons réalisé.



Le projet Smart Recruiting répond à la problématique de trouver une relation entre une formation et un stage.

Ce projet correspond donc au développement d'une plateforme intelligente de recrutement qui permet la simplification des relations entre l'université et les entreprises.
"Intelligente" car nous utilisons le deep learning pour que les recruteurs puissent trouver les formations qui correspondent le mieux à l'offre qu'ils postent.

Cette application web a pour but d'être utilisée par différents utilisateurs. Cela peut être :

- Une gestionnaire de scolarité ou directeur d'université qui souhaite avoir des informations et statistiques sur les offres et les formations.
- Un chargé de recrutement qui souhaite rendre visible une offre mais aussi peut faire des recherches pour connaître les formations associées à l'offre postée. Cela permet de mettre en contact plus facilement les chargés de recrutement avec des profils qui leur correspondent.
- Un étudiant qui cherche des stages ou des formations sur la plateforme.

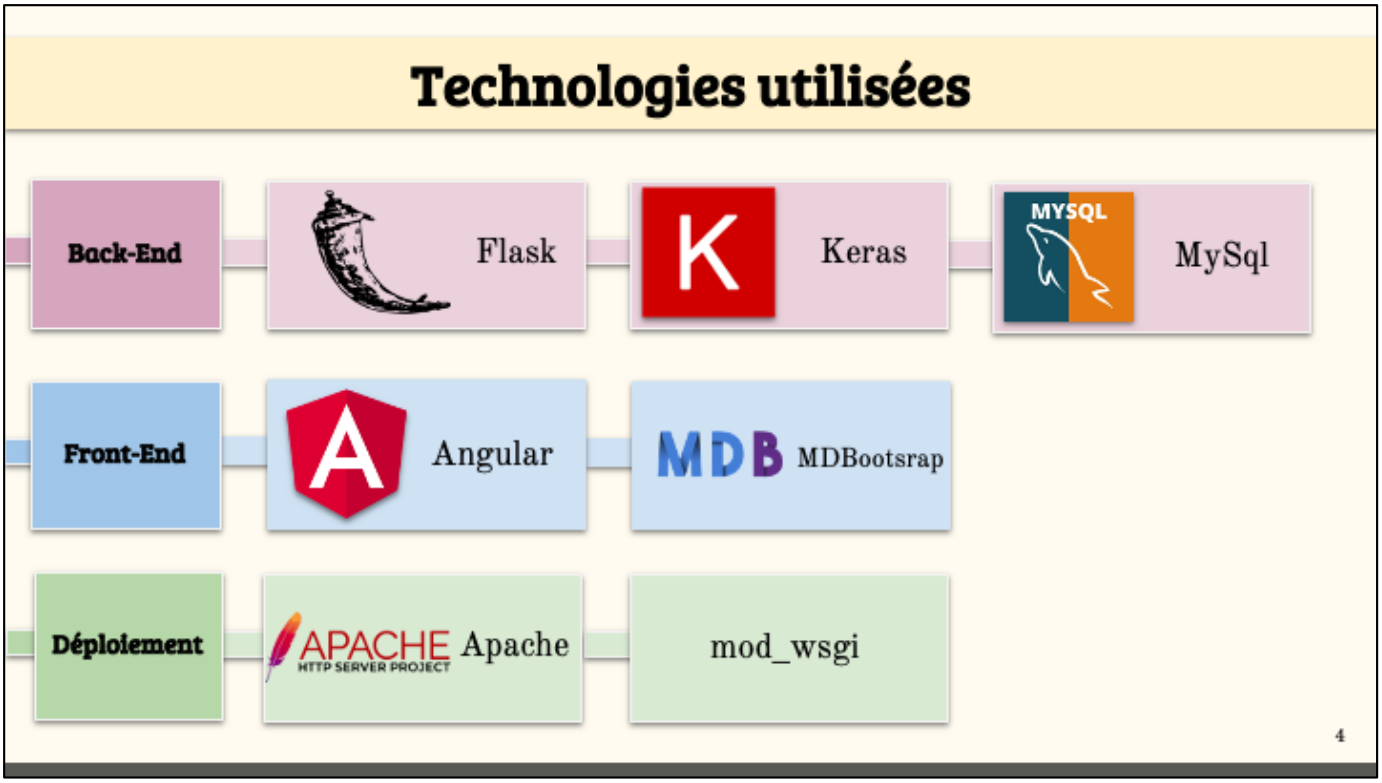
L'application sert donc pour l'ensemble du processus de recrutement ainsi que pour faire une étude du marché.

Maintenant, nous allons parler de nos missions qui ont légèrement changées après la soutenance de mi-parcours.

Pendant le sprint 0, après avoir pris en main le projet, nous avons amélioré le lancement du serveur donnant la possibilité de choisir le corpus utilisé. Nous nous sommes également concentrés sur l'élaboration des maquettes pour répondre aux besoins du persona Gestionnaire. Nous avons également réorganisé la base de données dans le but de répondre aux besoins des nouveaux personas.

Pour le sprint 1, nous nous sommes alors concentrés sur la réalisation de la partie du site correspondant au persona Gestionnaire avec les différentes statistiques demandées.

Pour le sprint 2, les objectifs ont changé. En effet, de base nous devions nous focaliser sur l'aspect Green IT de l'application. Cependant, après en avoir parlé avec notre Product Owner, Anthony Geourjon, nous avons décidé de refaire le persona Recruteur dans le but d'harmoniser le site mais également pour permettre aux offres de stage d'être renseignées correctement dans la base de données. Nous avons aussi refait la page liée à la connection et l'inscription en fonction des différents utilisateurs.

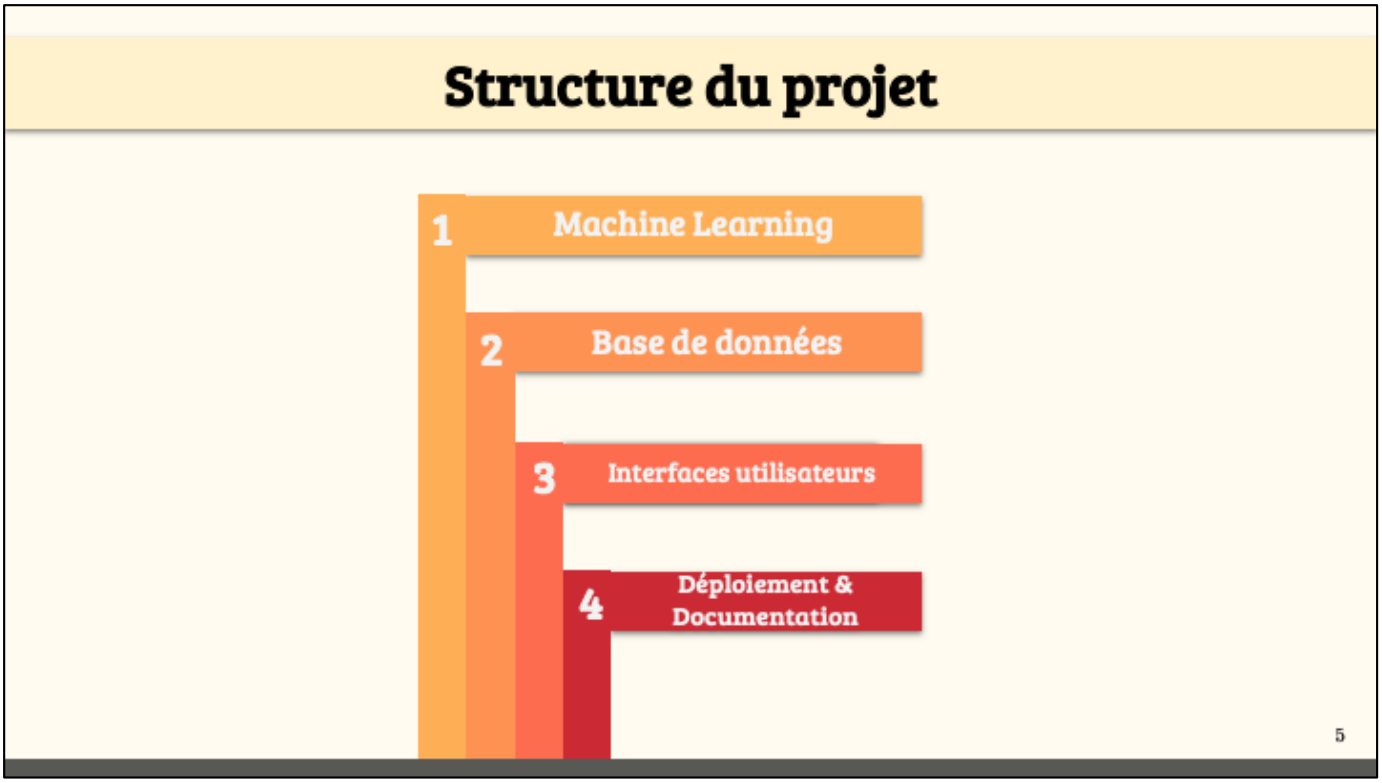


Concernant les technologies utilisées, nous avons utilisé la technologie Flask, qui est un framework web qui fournit des outils, des bibliothèques et des technologies qui vous permettent de construire l'application web. Comme notre application utilise le deep learning, nous avons également utilisé Keras qui est une API de réseaux neuronaux de haut niveau. En ce qui concerne la gestion de la base de données, nous avons utilisé MySql.

Pour le front, nous avons utilisé le framework Angular pour la partie fonctionnelle du projet et MDBootsrap pour la partie visuelle.

Nous utilisons le serveur web Apache et son module mod_wsgi pour déployer notre application flask et notre application Angular sur la VM.

Tout au long du projet, nous avons travaillé avec l'outil de gestion de version GitLab.



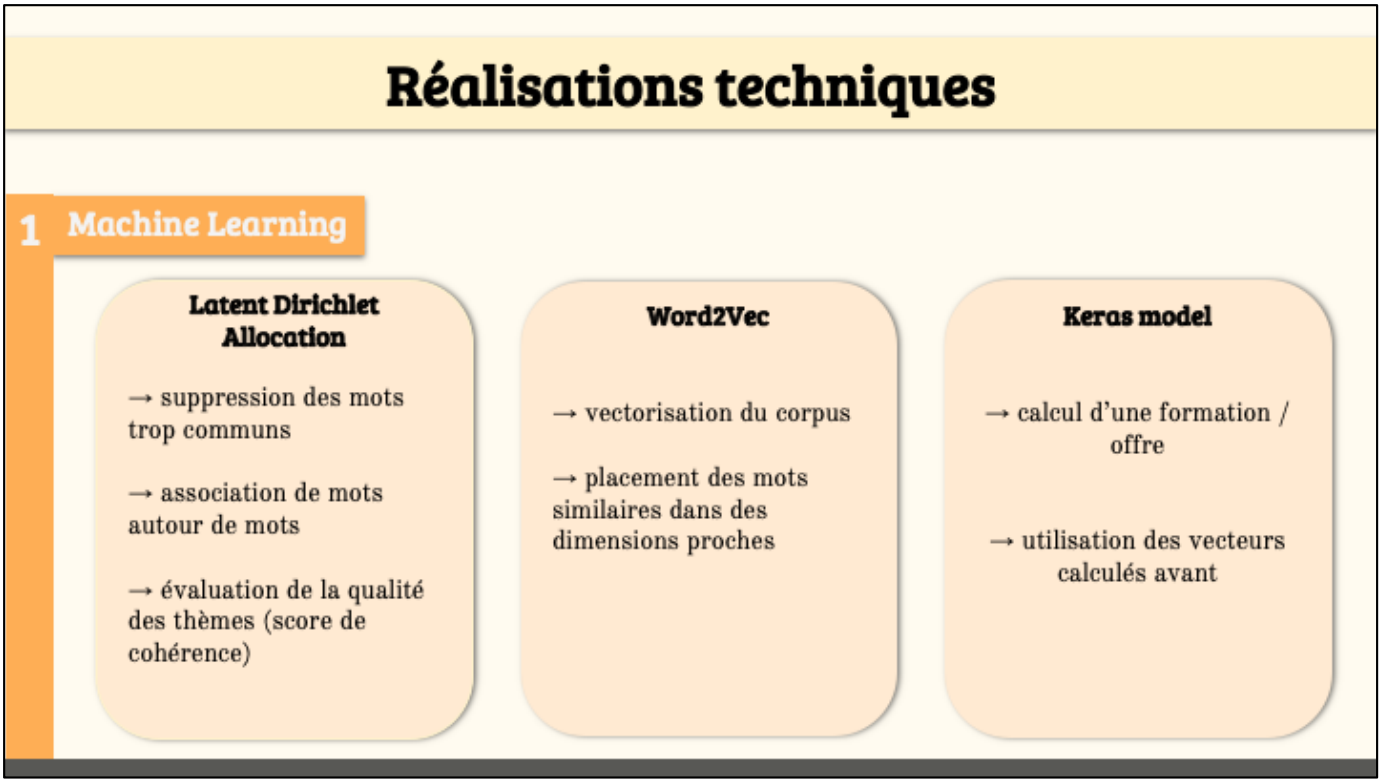
Notre projet peut se diviser en 4 parties différentes.

La première concerne tout ce qui touche le machine learning. C'est là où on utilise les deux modèles différents pour entraîner le modèle de réseau neuronal nous permettant de prédire les bonnes formations aux recruteurs et étudiants.

La deuxième partie concerne la base de donnée MySql. C'est elle qui gère les offres, les entreprises, les utilisateurs, les formations et les composantes.

La troisième partie correspond aux interfaces utilisateurs, c'est à dire l'application web SmartRecruiting. C'est cette application qui permet aux utilisateurs de se connecter et d'avoir accès aux différentes fonctionnalités de SmartRecruiting.

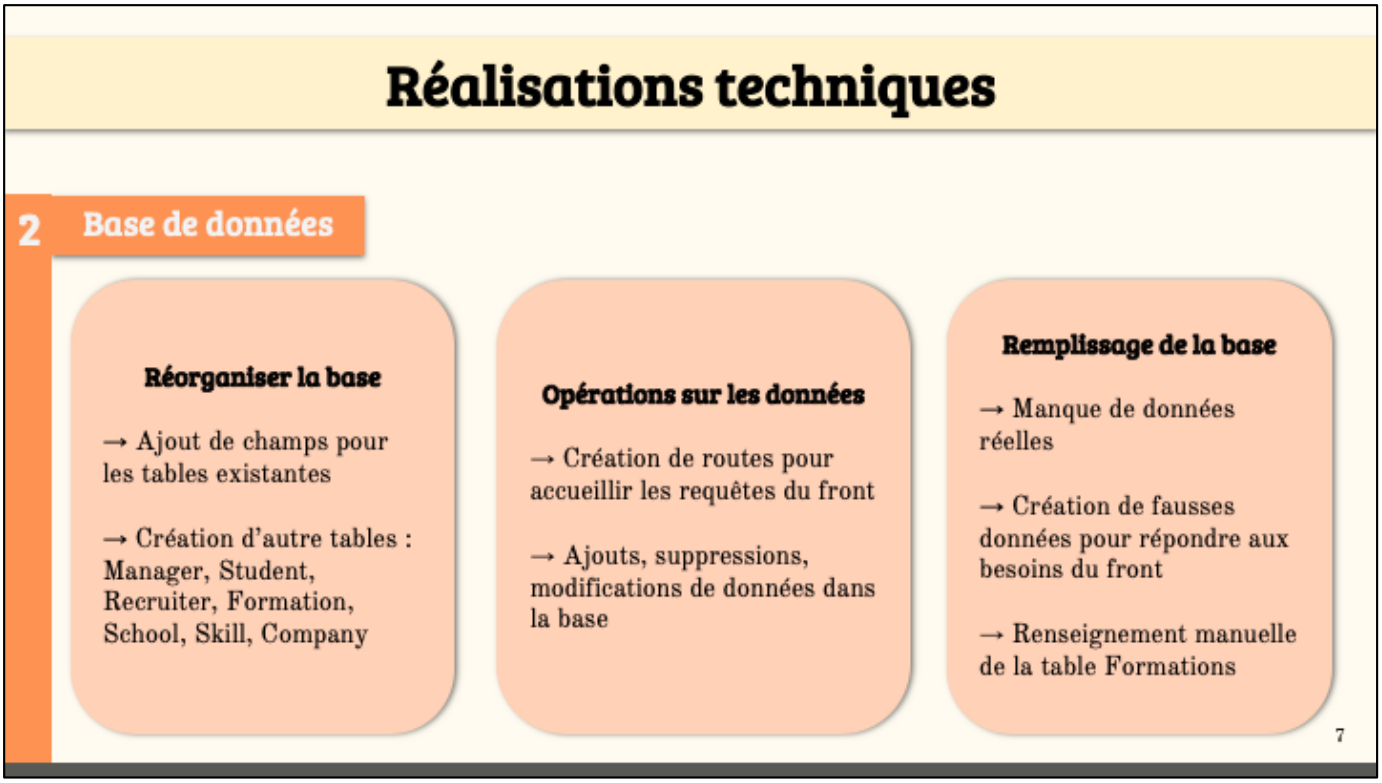
Pour finir, il y a une partie de déploiement et documentation pour rendre disponible notre application à tous et qu'elle soit facilement reprise pour les futurs groupes.



La première chose qui est faite, c'est la création de thèmes concernant les différentes formations. Pour cela, le code commence par supprimer les mots qui sont le plus souvent utilisés comme le, la, de, …
Ensuite, on associe les mots en thème et on calcule la qualité de ces groupes avec le score de cohérence

Une fois que c'est fait, on va vectoriser ces thèmes ainsi que les offres avec le modèle Word2Vec de gensim. Il va créer une matrice de plus de 150 dimensions et plus les mots seront proches, plus ils seront proches dans la matrice.

Enfin, on va utiliser les deux modèles différents pour entraîner le modèle de réseau neuronal. Les datasets seront séparées pour les entraîner et les tester pour vérifier la précision du modèle.

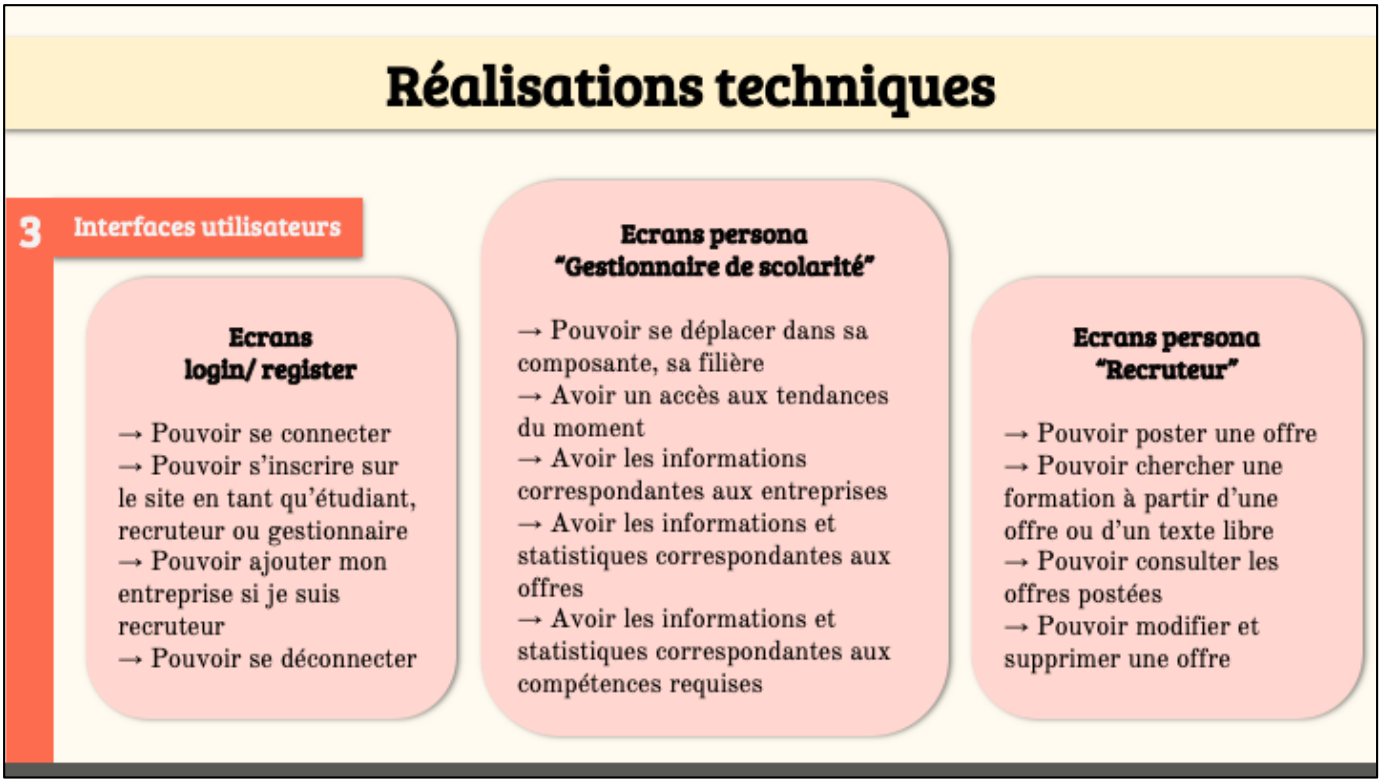


Pour l'équipe Back-End, plusieurs réalisations techniques étaient liées à la base de données.

Dans un premier temps, nous avons dû réorganiser la base. En effet, la base était restreinte pour la recherche de formation en fonction d'une offre de stage. Nous avons donc dû ajouter les tables Manager, Student, Recruiter, Formation, School, Skill et Company, dans le but de pouvoir faire les différents personas cités plus tôt. Nous avons également ajouté de nouveaux champs aux tables existantes. Nous avons essayé de penser à l'ensemble des informations nécessaires pour faire les statistiques demandées pour le persona Gestionnaire mais aussi les informations nécessaires pour le persona Student.

Dans un second temps, nous avons créé l'ensemble des routes pour répondre aux besoins du front ainsi que toute la gestion de la base avec l'ajout, la suppression, et la modification de données.

Et pour finir, nous avons essayé de pallier le manque de données réels en créant de fausses données pour que le front puisse marcher. Nous avons aussi récolté manuellement l'ensemble des informations liées aux formations pour avoir la table Formation complète.



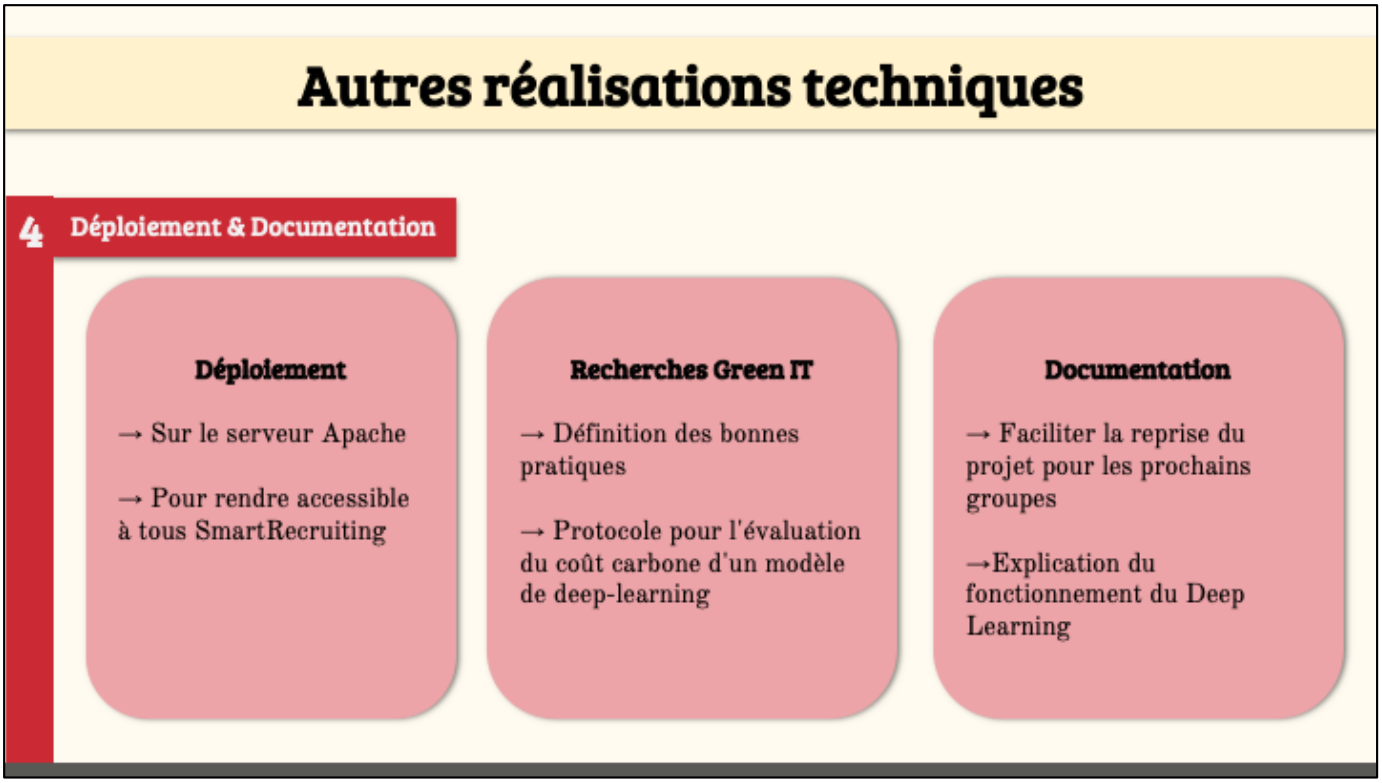
Pour l'équipe Front-End, nous avons fait le choix de repartir de 0 par rapport à la dernière itération du projet. Nous avons réalisé ce choix suite à la redéfinition des différents personas avec le Product Owner. Néanmoins nous ne sommes pas totalement partis de 0, nous avons utilisé un template de dashboard libre de droit de MDB Angular que nous avons par la suite adapté à notre convenance.

Dans un premier temps, nous nous sommes intéressés au persona "Gestionnaire de scolarité" et nous avons donc réalisé le dashboard de ce dernier avec les features définis avec notre Product Owner. Nous avons donc une barre de navigation sur le côté afin de sélectionner la composante voulue ainsi que la filière voulue, et nous avons ensuite 4 onglets permettant d'afficher diverses informations. Un onglet tendance permettant de récapituler les différentes tendances du moment. Un onglet entreprise récapitulant les informations sur ces dernières. Un onglet offres récapitulant les informations et statistiques sur ces dernières et enfin un onglet compétences montrant les principales compétences que l'on retrouve dans les offres postées, cela sous forme de diagramme à bulle pour une meilleure lisibilité.

Dans un second temps nous nous sommes intéressés au persona "Recruteur" qui correspond aux recruteurs d'entreprises. Nous avons donc réalisé un second dashboard comprenant trois onglets différents, un onglet pour poster une offre, un onglet pour chercher une formation à partir d'une offre déjà postée ou bien d'un

texte libre, cela utilisant le machine learning, et enfin un onglet pour consulter ses offres afin de pouvoir les modifier ou les supprimer.

Pour finir, nous nous sommes ensuite penchés sur la connexion et l'inscription des utilisateurs sur la plateforme web. Pour cela, nous avons une page pour se connecter avec un mail et un mot de passe, et une autre page pour s'inscrire. Pour l'inscription, des champs génériques sont demandés tel que le nom, le prénom le mail, etc, puis en fonction du rôle/persona choisi, d'autres informations sont demandées comme la composante, le nom de l'entreprise, etc. Lorsque le rôle "recruteur" est choisi pour l'inscription, si l'entreprise n'est pas déjà rentrée dans la base de donnée, il est possible de la créer via une page de création d'entreprise. Enfin, il y a également un système de déconnexion et de gardes pour garantir une séparation des différents personas au sein de l'application, et également un système de jeton pour réserver l'accès aux pages autres que connexion et inscription uniquement aux membres connectés.

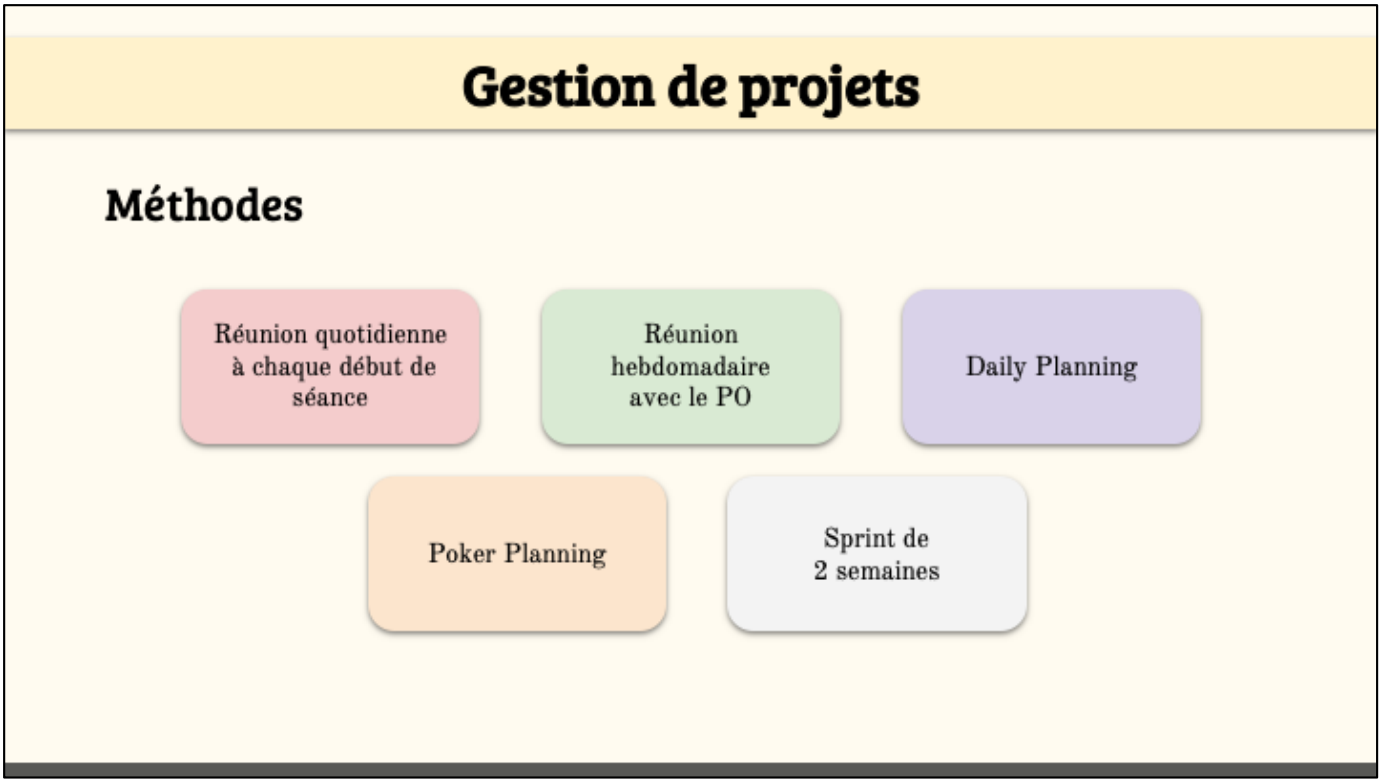


Une fois que nous avons eu une version stable et montrable de notre application. Il nous a fallu mettre en production l'application.
Nous avons donc dû déployer l'application sur une machine virtuelle mise à notre disposition.
Nous avons choisi d'utiliser le serveur HTTP Apache2 pour mettre en place notre site.
Nous avons également dû utiliser mog_wsgi pour mettre en environnement de production notre application Flask.
Le déploiement s'est passé en deux étapes. D'abord, on a déployé le back-end. Puis on a déployé le front end. Nous avons mis à votre disposition un tutoriel pour que les futurs groupes puissent facilement déployer l'application Web et configurer correctement le serveur http. Et ne pas faire face aux mêmes difficultés que nous avons rencontré.

Nous avons également fait plusieurs recherches sur le Green IT. Le Green IT est une pratique qui vise à réduire l'impact qu'à l'informatique sur l'environnement. Nous avons fait un compte-rendu de nos recherches que vous pouvez trouver sur le wiki.

Dans le but de faciliter la reprise du projet pour les prochains groupes, nous avons produit une documentation et également commenté le code.
Nous avons également réalisé une documentation sur le fonctionnement du deep learning en expliquant les parties telles que la vectorisation, Tf-idf, et l'allocation de

Dirichlet latente et la méthode de classification de type “boîte blanche”. L’ensemble de ces explications se trouve dans le rapport technique.

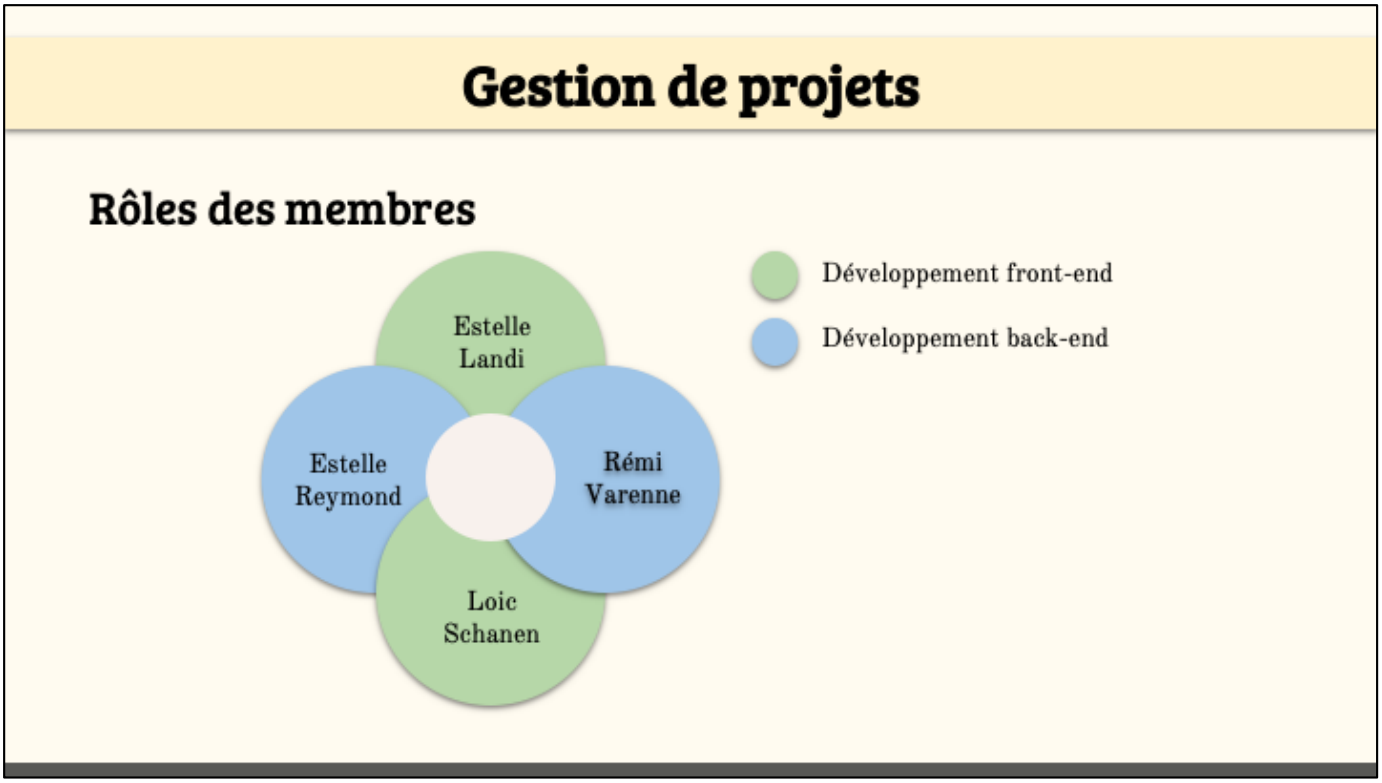


En ce qui concerne la gestion de projets, nous avons utilisé plusieurs méthodes. Nous avons travaillé en méthode agile avec des sprints de 2 semaines.

Pour définir les difficultés des tâches listées par le Product Owner, nous avons fait un poker planning. Cela nous a permis de se mettre d'accord sur la vision et la réalisation des tâches. Et de savoir quelle tâche était réalisable ou non.

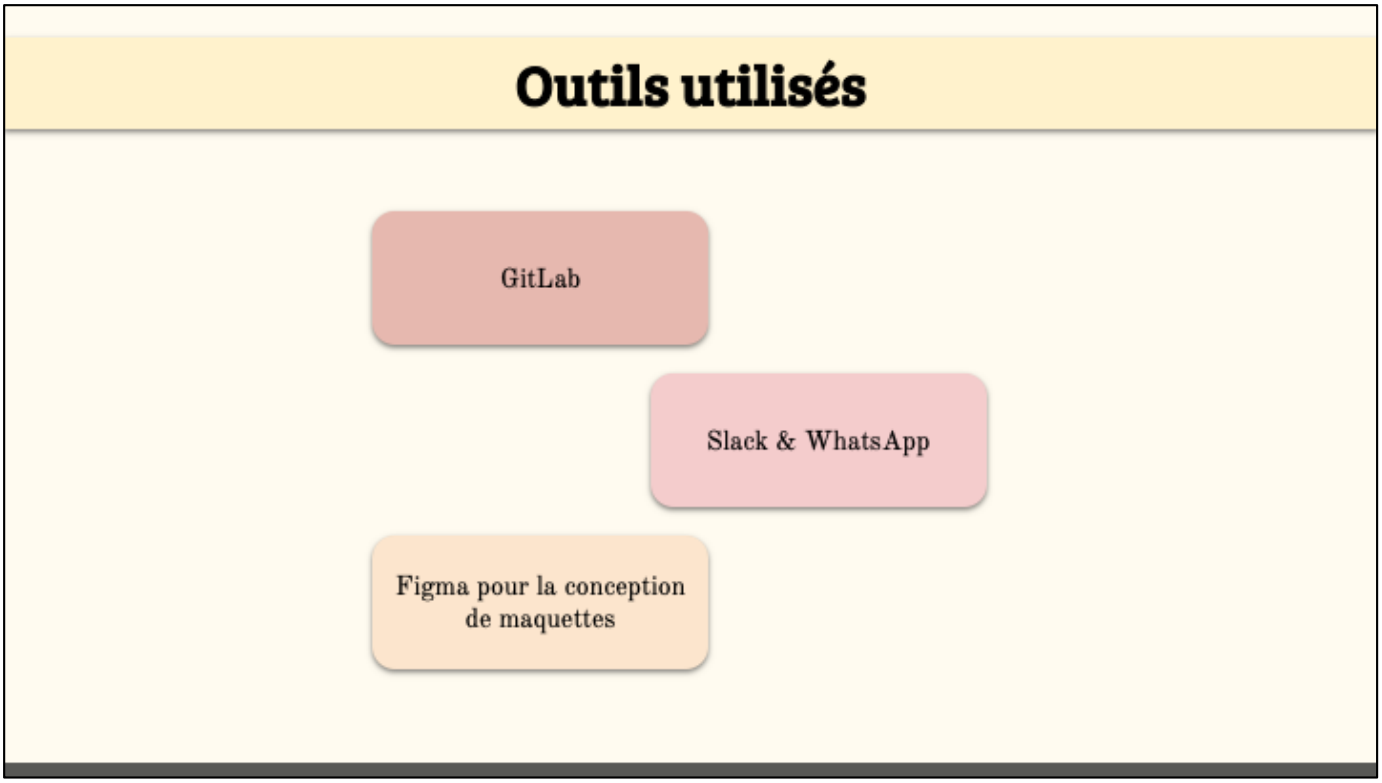
Nous avons mis en place des réunions quotidiennes à chaque début de séance, pour faire le point sur l'évolution du travail produit le jour d'avant. Durant ces réunions, nous établissons également le daily planning pour cadrer et définir les objectifs de la journée. Cela nous a permis de bien se rendre compte de l'avancement de chacun et de bien gérer notre temps.

Des réunions hebdomadaires avec le Product Owner ont été mis en place pour faire un compte rendu de nos avancements mais aussi de redéfinir les objectifs si besoin.



Pour le projet, nous avons décidé de nous scinder en deux équipes principales : une front-end et une back-end. Le choix s'est fait naturellement, Estelle Landi et Loïc Schanen se sont occupés du développement front-end. Estelle Reymond et Rémi Varenne se sont occupés du développement back-end.

Rémi a été le Scrum Master.

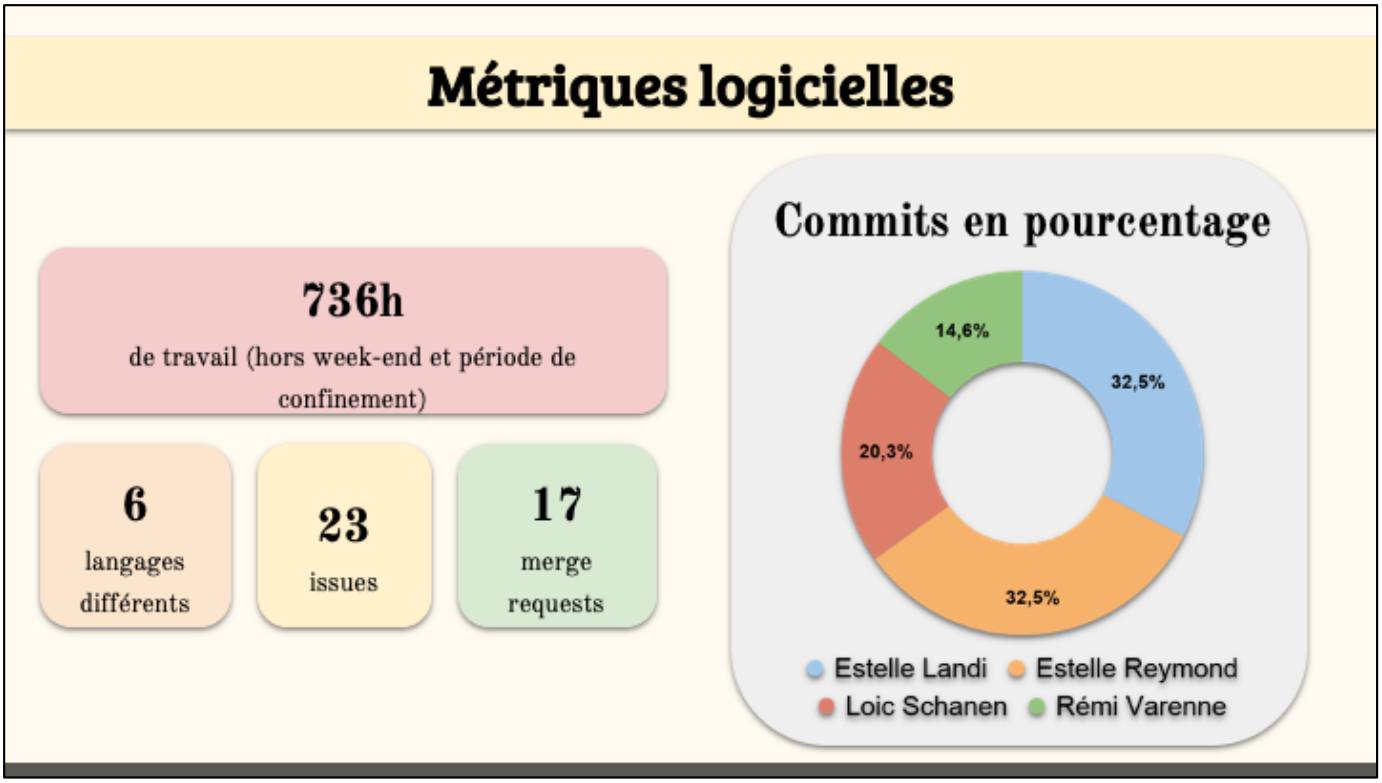


Tout au long du projet, nous avons plusieurs outils pour la communication, le travail en équipe.

Pour communiquer facilement avec le groupe, nous avons pris l'initiative de créer un groupe sur WhatsApp pour pouvoir communiquer en toute simplicité dans le groupe. Mais nous avions aussi un groupe sur Slack avec notre product owner pour garder un contact constant avec Anthony Geourjon.

Vu que nous travaillons sur le même code mais sur des parties différentes, le logiciel GitLab nous a permis de versionner notre code et de faire des merge request. C'est également là où a pu utiliser "l'issue board" de gitlab au même titre qu'une kanban board.

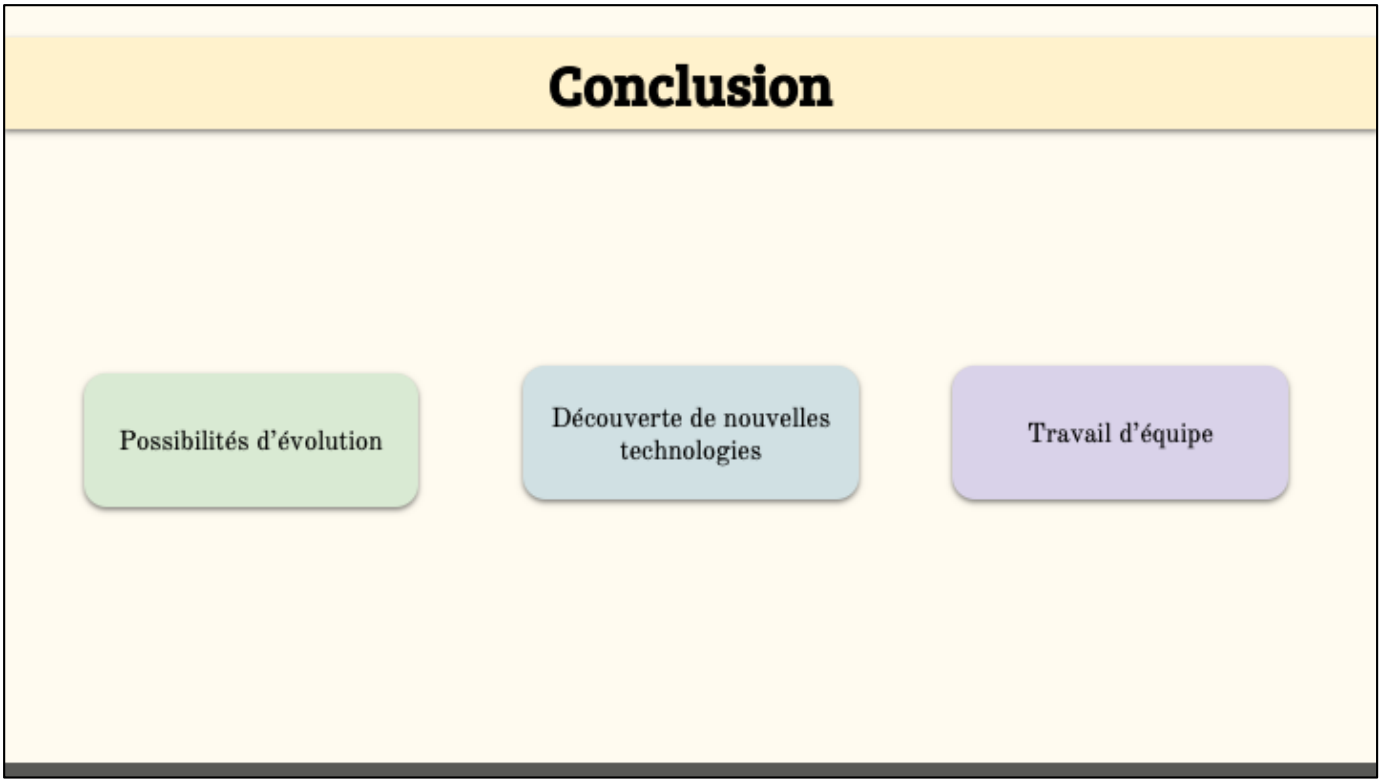
Pour la conception de nos maquettes, nous avons utilisé le logiciel Figma qui nous a permis de se mettre d'accord sur les données nécessaires pour le site ainsi que le visuel attendu.



Nous allons maintenant vous montrer quelques métriques logicielles représentatives de notre travail réalisé sur ce projet. Dans un premier temps sur la durée du travail que nous avons fourni tout au long de ce projet avec plus de 736h de travail.

Notre projet regroupe ensuite 6 langages de programmation différents dont principalement typescript, html, css, python et javascript.

Nous avons ensuite des métriques liées à GitLab : 23 issues et 17 merges requests pour le back et le front, et nous pouvons ensuite voir la représentation en % des commits pour chaque membre du groupe, et nous pouvons voir que l'ensemble du groupe à participer de manière égale à l'avancement du projet. Il est néanmoins important de noter que le nombre de commit réalisé n'est pas suffisant comme indicateur pour représenter la participation d'un membre au projet car selon les tâches à réaliser, plus ou moins de commit seront fait.



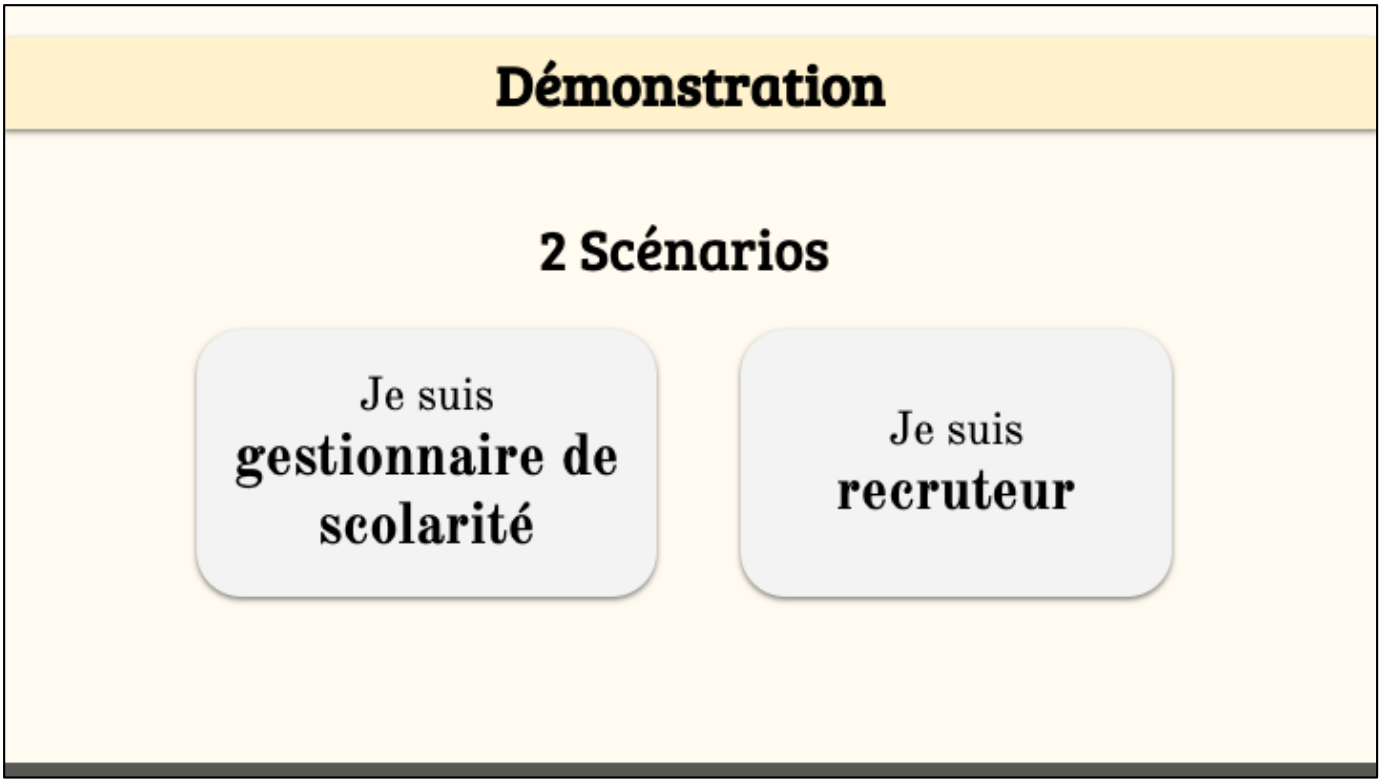
En conclusion, le projet SmartRecruiting est un projet ambitieux avec des possibilités d'évolutions multiples car il utilise des nouvelles technologies et répond à une demande actuelle.

Nous savons qu'un tel site a de l'avenir car chaque année, plus d'étudiants souhaitent trouver des formations ou des stages, que les entreprises sont toujours à la recherche de nouveaux profils et que les gestionnaires de scolarité aiment de plus en plus connaître l'état du marché des offres de stages.

Ce projet nous a également permis de monter en compétence sur des technologies que nous ne connaissions pas ou très peu. Il fut très enrichissant car nous avons pu découvrir d'avantage le monde du web ainsi que le deep-learning.

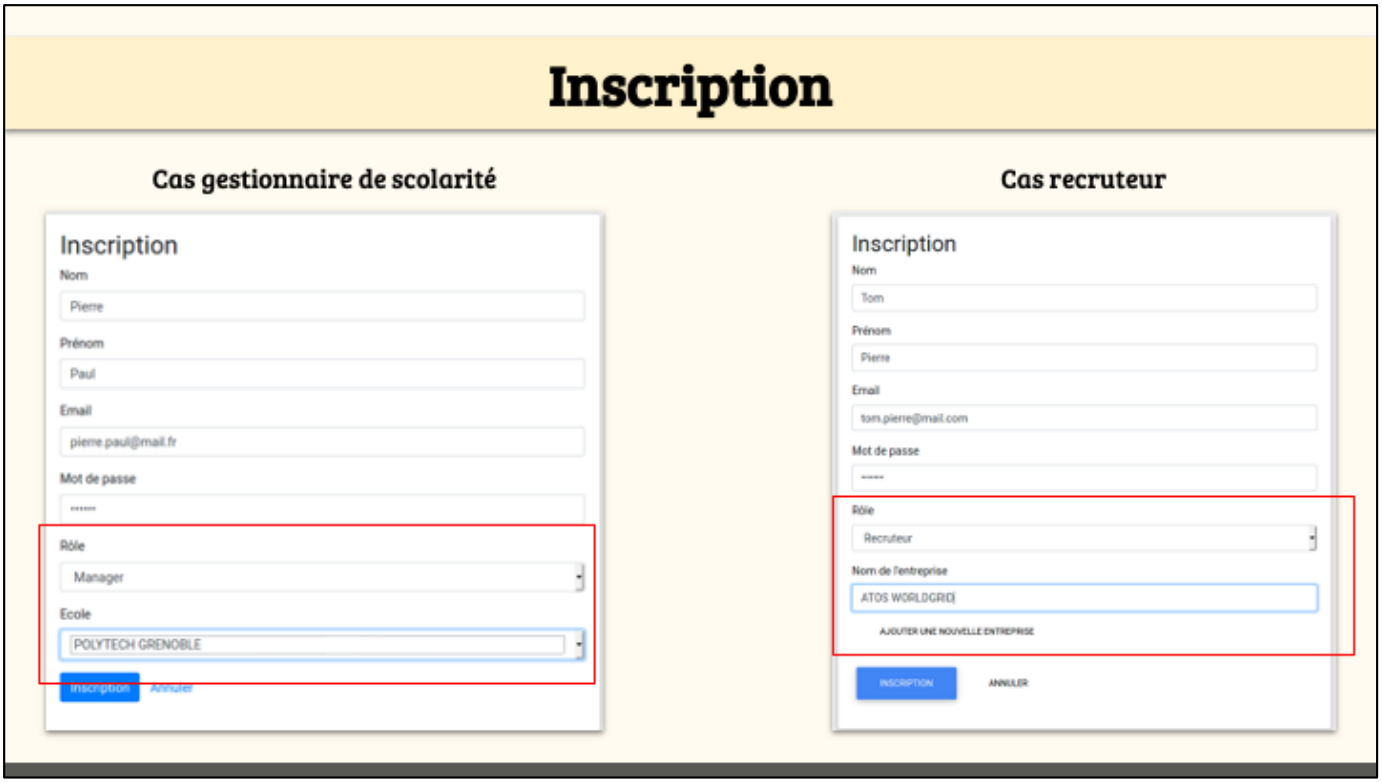
Nous avons aussi apprécié travailler en équipe car nous avons su bien définir le rôle de chacun et mettre à profit les compétences de chacun pour que le projet avance au mieux.

De plus, l’ambiance sérieuse et amicale du projet nous a permis de nous épanouir facilement dans le groupe.



Nous allons vous faire une démonstration de notre projet.

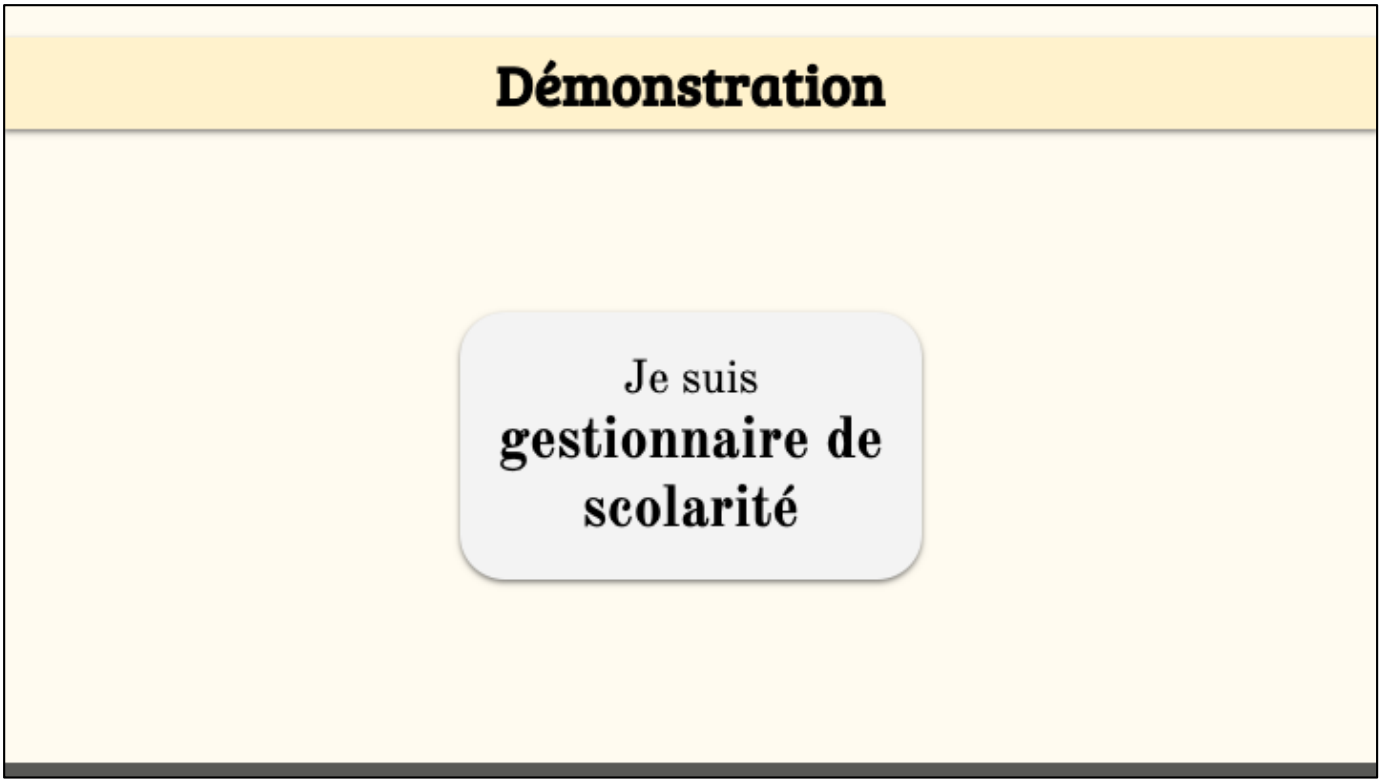
Il y a deux scénarios possible : celui où nous sommes un gestionnaire de scolarité et celui où nous sommes un recruteur.



A l'inscription, tous les utilisateurs doivent remplir des informations communes, comme le nom, prénom et adresse mail et mot de passe.

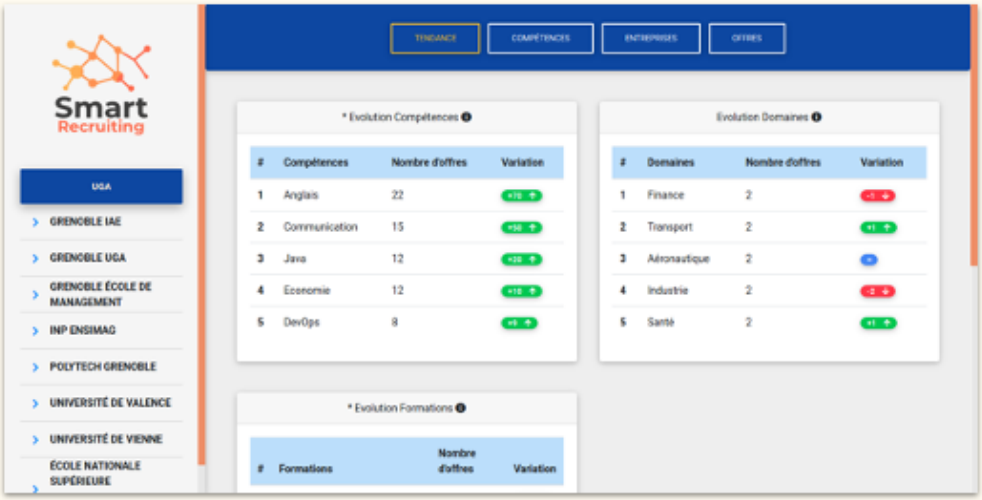
Et en fonction du rôle choisi, le formulaire évolue de deux façons. Une pour le cas gestionnaire où on demande à quelle école il est associé, et une autre pour le recruteur à qui on demande à quelle entreprise il appartient.

Si l'entreprise n'existe pas dans notre base de données il est possible de la créer.



Nous allons maintenant vous montrer ce que le persona gestionnaire peut visualiser sur la partie du site qui lui est dédié.

Gestionnaire de scolarité : Page d'accueil



Les "*" qui se présentent sur les cards du dashboard correspondent aux cards où les données sont fictives, aucun lien avec la bdd n'est établi par faute de data.

Ici, vous retrouvez la page d'accueil de la gestionnaire de scolarité.

Le premier élément de la page d’accueil est la navigation. Cela permet d’avoir accès aux statistiques à différents niveaux d'abstraction.

Après que notre Product Owner est interrogé une gestionnaire de scolarité pour se mettre d’accord sur les statistiques qui pourrait l'intéresser, quatre parties ont été défini. Nous avons choisi de mettre les parties en quatre différents onglets.

Nous allons expliquer les différents éléments réaliser dans ces onglets dans les diapositives suivantes.

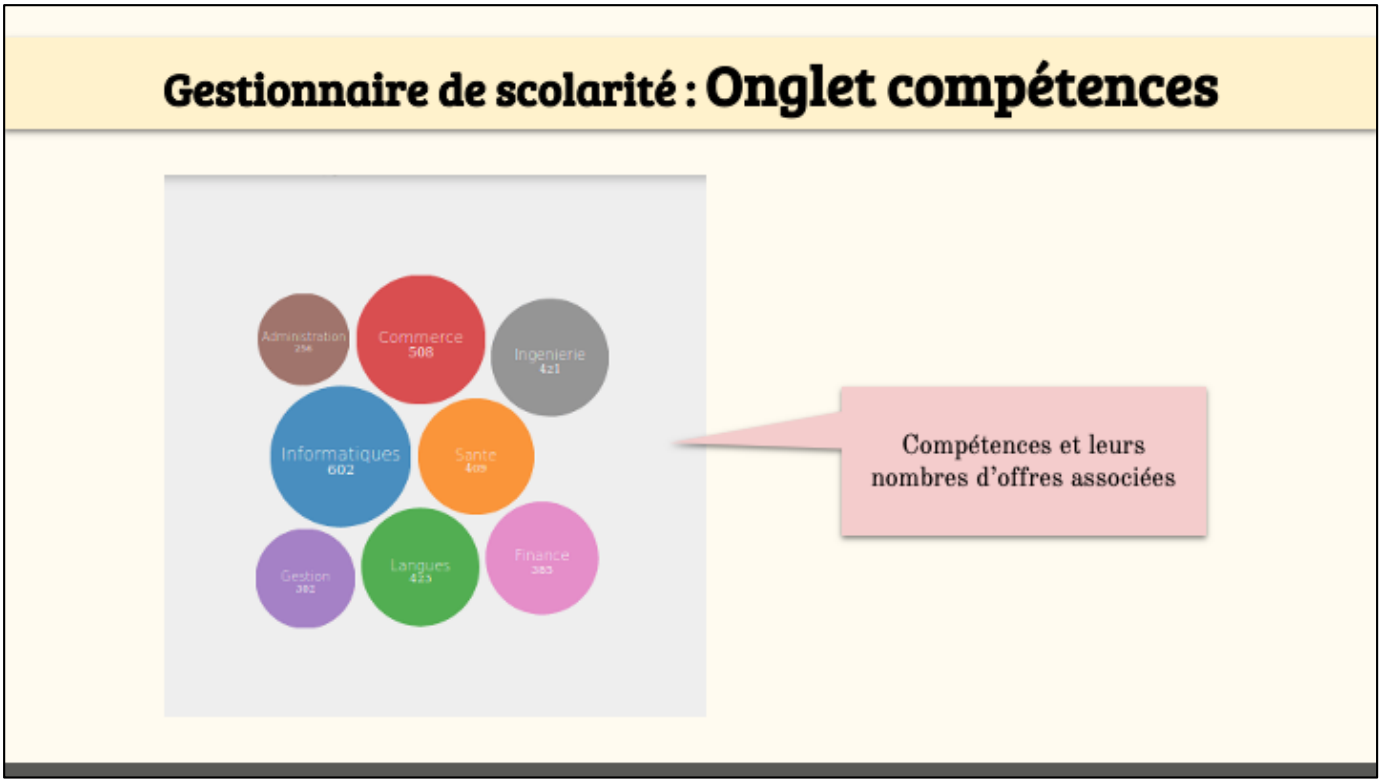
L'onglet Tendance permet d'avoir une vue d'ensemble du marché en fonction des compétences, des domaines, des écoles.

Le premier tableau correspond à l'évolution du nombre d'offres en fonction des compétences.

Le deuxième tableau correspond à l'évolution d'offres en fonction des domaines.

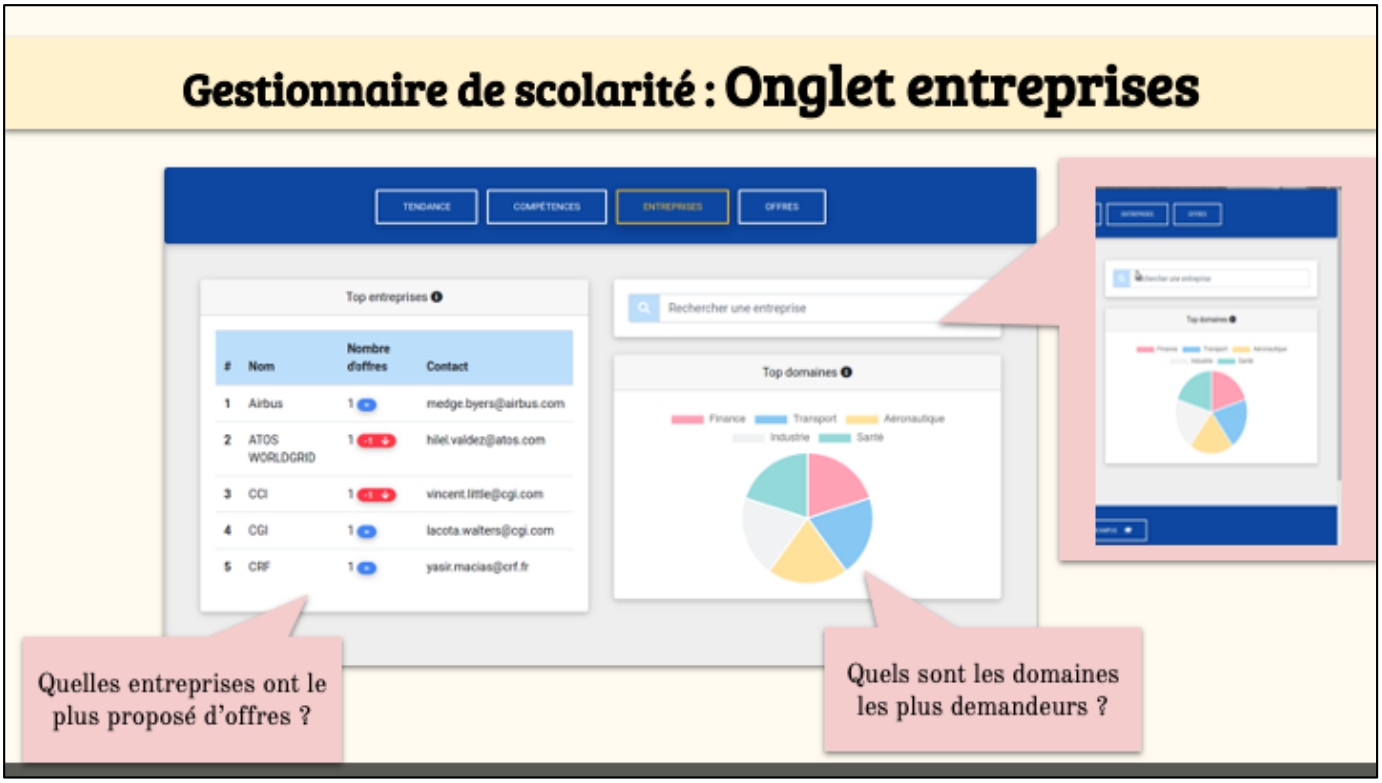
Le troisième tableau correspond à l'évolution des offres pour les différentes composantes.

Malheureusement, nous n'avons pu réaliser que le deuxième tableau avec de fausses données. Car malheureusement, nous avions un manque d'informations pour avoir des statistiques.



L'onglet Compétences permet d'avoir un diagramme à bulle qui répertorie le nombre d'offres en fonction des compétences. Nous avons pensé qu'il y aurait des sous-compétences et que lorsqu'on clique sur une compétence, on puisse avoir un détail plus précis de cette dernière. Par exemple, si on veut savoir en détail comment sont réparties les offres pour le domaine Informatique, il suffit de cliquer sur la bulle et un autre diagramme en bulle apparaît. On peut voir le nombre d'offres en fonction des sous-catégories de la compétence Informatique.

La partie visuel fonctionne mais pas la partie fonctionnelle par manque de données.

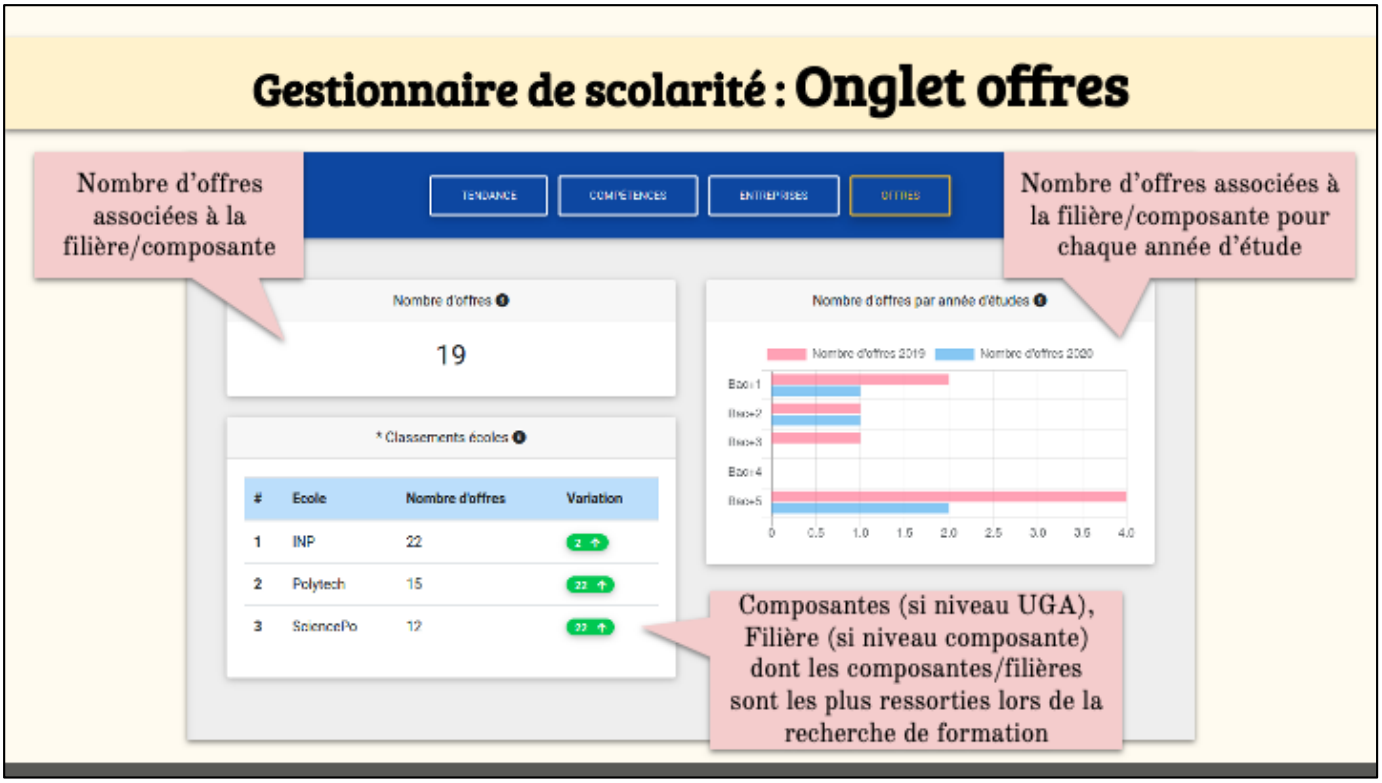


L'onglet Entreprises permet de connaître toutes les informations liées aux entreprises.

Cette onglet est composé de :

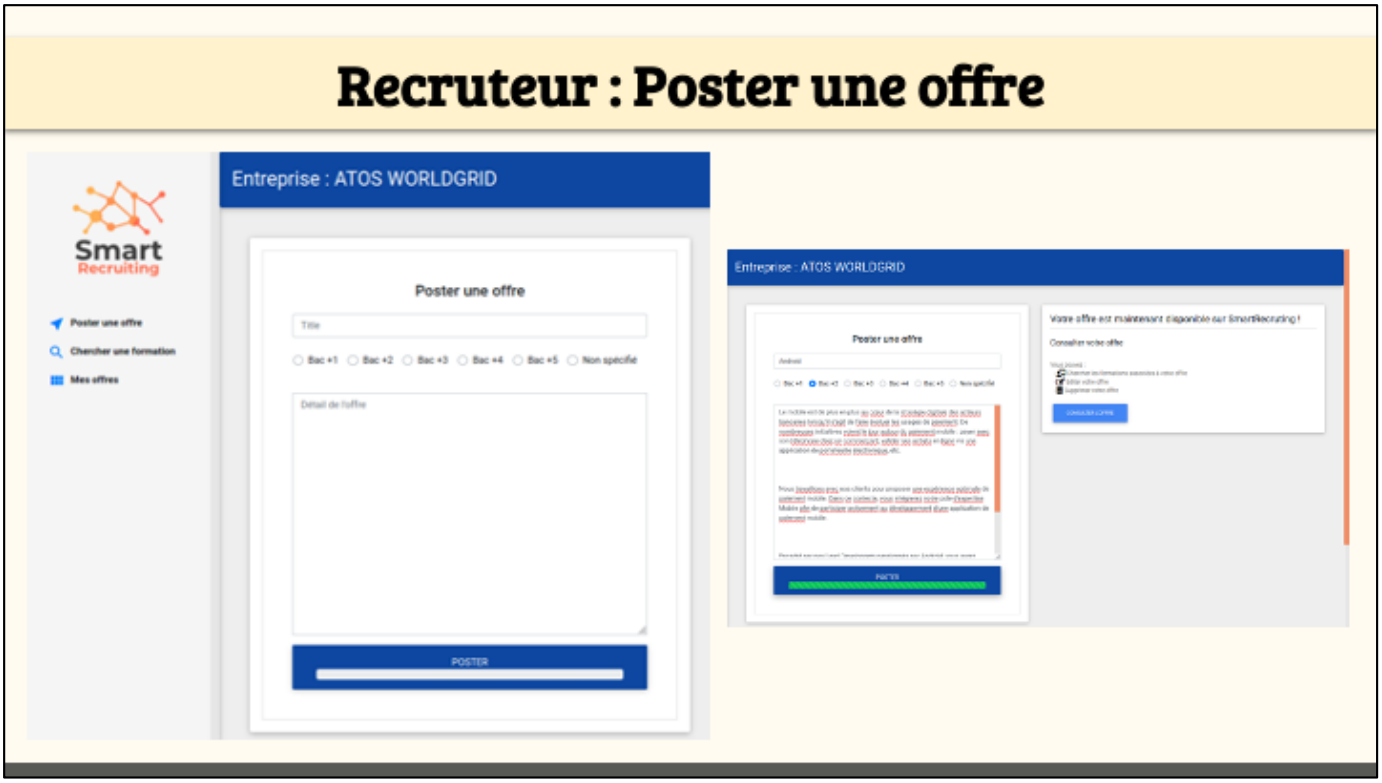
- Un top pour connaître l'évolution du nombre d'offre par entreprises.
- Une barre de recherche qui permet de rechercher les offres liées à une entreprise. On peut ensuite sélectionner une offre pour avoir plus de détail.
- Un diagramme pour connaître le top des domaines.

Tous les parties de cette onglet ont été réalisé mais avec de fausses données.



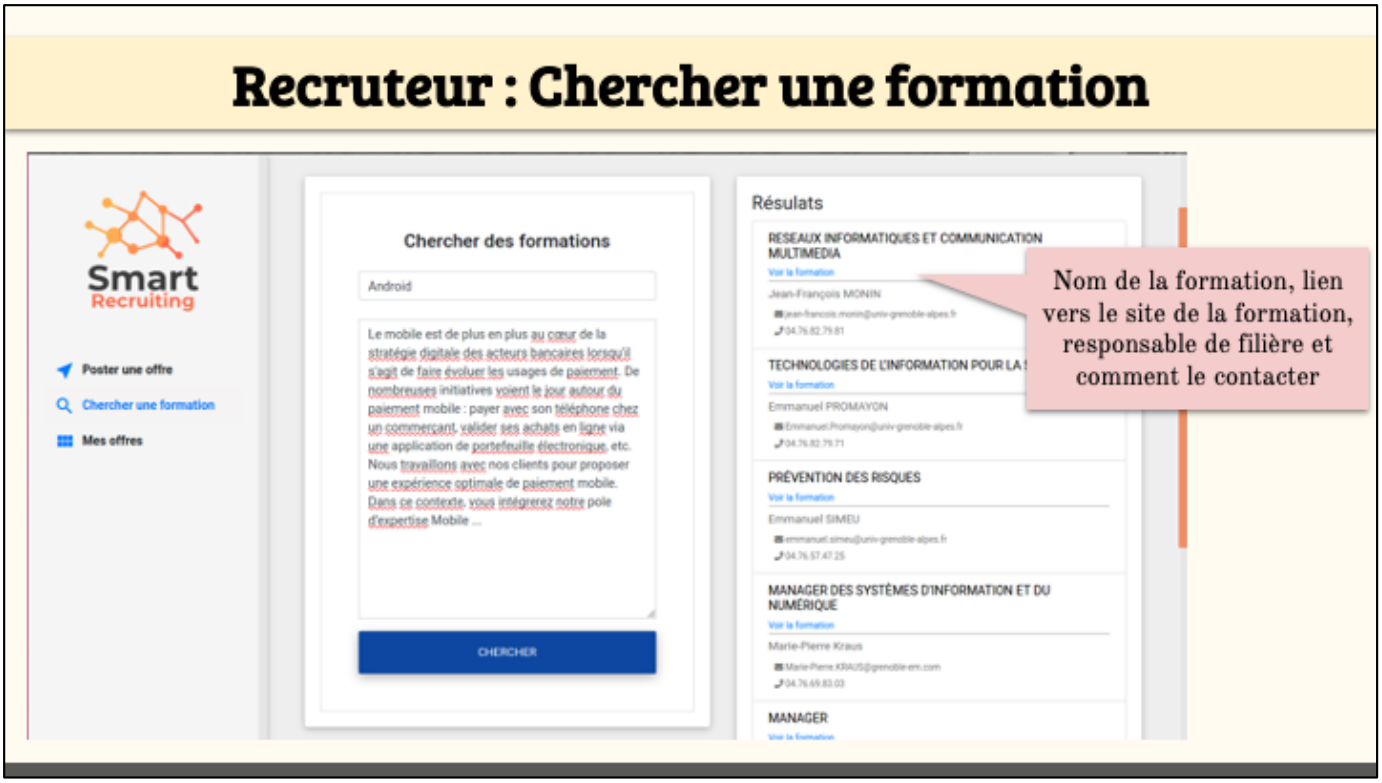
L'onglet Offres permet d'avoir les statistiques sur le nombre d'offres présentes dans la base, le top des écoles et un diagramme à barre pour connaître la répartition des offres en fonction du niveau d'étude demandé.

Nous allons maintenant vous montrer la partie du site lorsque nous sommes un recruteur.



Ici, nous avons la page pour que le recruteur puisse poster une offre. Il faut renseigner le titre, le niveau d'étude nécessaire et la description de l'offre.

Pour cette page nous avons ajouté des champs supplémentaires comme le niveau d'étude pour que l'offre soit renseignée correctement dans la base.



Il y a une page pour chercher les formations associées à une offre.

Il suffit de mettre le titre et la description en texte libre et de cliquer sur “chercher”. Cela permet de trouver les formations liées à l’offre en utilisant le deep-learning.

Chaque formation est affichée avec le contact de la formation ainsi qu’un lien cliquable sur le site web de la formation.

Il est possible pour le recruteur de revoir les offres qu'il a déjà posté. Elles sont affichées sous forme de grid. Il est facile de voir facilement la date, le domaine et le niveau d'étude.

Les offres sont modifiables, supprimables et il est également possible de voir les formations associées.

Le premier logo permet l'accès à la liste des formations associées.

Le deuxième logo permet de modifier l'offre postée.

Et le dernier logo permet de supprimer l'offre du site et de la base de données.

Fin de la présentation

La présentation est maintenant finie.
Nous vous remercions pour votre attention et nous restons à disposition pour plus d’informations.