Projet WEB II - 2022-2023

HELBManager



POPADIUC CLAUDIU

Monsieur Riggio, Jonathan

**Table des matières**

[Introduction 3](#_Toc123758790)

[Description des technologies utilisées 4](#_Toc123758791)

[Fonctionnalités de base 5](#_Toc123758792)

[Fonctionnalités supplémentaires 11](#_Toc123758793)

[Analyse 12](#_Toc123758794)

[Limitations et développement futur 13](#_Toc123758795)

[Conclusion 14](#_Toc123758796)

[Source 15](#_Toc123758797)

# **Introduction**

Dans le cadre du cours de WEB Q III, il nous a été demandé de réaliser individuellement une application web permettant à des utilisateurs de faire de la gestion de projets via un système simplifié de planification de tâches.

Pour avoir une certaine base fonctionnelle, M. Riggio nous a proposé de regarder les vidéos de M. Corey Shafer (Source n°1), c’est ce que j’ai fait, j’ai ensuite codé moi-même la base fonctionnelle.

Dans ce rapport, je vais commencer par expliquer les technologies utilisées, ensuite les fonctionnalités qui ont été implémentées et leur aspect technique. Je continuerais par faire la même chose mais cette fois-ci pour les fonctionnalités supplémentaire. Il y aura ensuite l’analyse de mon projet présentée avec l’aide d’un diagramme. Les limitations et le développement futur du projet suivront et je finirais par une conclusion.

Une page source sera ajouté à la fin de ce rapport, ainsi j’indiquerais au fur et a mesur dans mon rapport lorsque les sources ont été utilisé.

# **Description des technologies utilisées**

Pour ce projet j’ai utilisé le Framework « Django » qui est un cadre de développement web Python de haut niveau. Je l’ai utilisé car c’est ce qui a été imposé par mon client. J’ai aussi utilisé bootstrap car il était dans les video de la base fonctionelle. Pour finir j’ai utilisé aussi du javaScript car j’en ai eu besoin pour principalement réaliser le drag & drop ainsi que d’autre petis détails.

Voici leurs déscription :

Django :

Django est un framework de développement web en Python qui facilite la création de sites web. Il offre plusieurs outils pour les développeurs, comme la gestion de bases de données, l'authentification de l'utilisateur et la gestion de formulaires.

Django utilise le modèle MVC et met l'accent sur la séparation des tâches, ce qui signifie que chaque composant d'une application ne s'occupe que d'une seule chose. Cela rend le code plus facile à maintenir et à étendre. Django est connu pour être sécurisé, scalable et rapide, ce qui en fait un choix populaire pour les développeurs de sites web professionnels.

JavaScript :

JavaScript est un langage de programmation de scripts principalement utilisé dans les navigateurs web pour rendre les pages web interactives. Il peut être utilisé pour ajouter des effets dynamiques sur les pages, comme des animations, des formulaires interactifs et des cartes. JavaScript peut également être utilisé pour construire des applications côté client, des jeux et des applications mobiles.

Bootstrap :

Bootstrap est une collection de styles prédéfinis pour la création de sites web et d'applications. C'est un framework qui vous permet de mettre en place rapidement un site web ou une application en utilisant des styles de base et des composants prédéfinis qui sont prêts à l'emploi.

# **Fonctionnalités de base**

* **Un système d’inscription/connexion sur la plateforme.**

**Inscription :**

**Une image contenant texte

Description générée automatiquement**

**Connexion :**

**Une image contenant texte

Description générée automatiquement**

Pour que les utilisateurs puissent enregistrer leur passage sur la plateforme, un système d'inscription et une connexion ont été mis en place. Dans le fichier "forms.py", il y a des champs tels que "username", "email", "password1" et "password2" afin que l'utilisateur puisse entrer ces informations de connexion, qu'il pourra modifier plus tard dans son profil s'il le souhaite.

Dans le fichier "views.py", j'ai ajouté une fonction "register" qui, si le formulaire mentionné ci-dessus est correct, enregistre les informations et redirige l'utilisateur vers la page de connexion. Tout est enregistré dans la base de données de Django, et l'administrateur a accès à toutes ces informations.

Dans le fichier "Base.html", il y a un bouton pour accéder aux pages de connexion, d'inscription et de déconnexion. Il y a aussi un fichier "login.html" qui donne à l'utilisateur une interface pour se connecter s'il n'est pas connecté, et un autre bouton qui lui permet de se déconnecter s'il est déjà connecté. Il y a également un fichier "register.html" qui donne une belle vue sur l'inscription de l'utilisateur et qui utilise le "formIcrispy" pour donner à l'utilisateur la possibilité de s'inscrire grâce aux 4 champs du fichier "forms.py".

J'ai pu réaliser tout cela grâce aux vidéos de M. Corey Shafer (Source 1). Je n'ai pas vraiment eu de problème, si ce n'est qu'à un moment donné, un mauvais recopiage du code a été mis en place et l'inscription ne fonctionnait pas, mais tout a pu être réglé.

* **Un système de profil pour les utilisateurs inscrits.**

**Une image contenant texte

Description générée automatiquement**

Pour que les utilisateurs puissent visualiser et modifier leurs données, comme leur email et leur nom, un système de profil a été mis en place. Il y aura également une photo de profil par défaut que l'utilisateur pourra modifier s'il le souhaite.

Dans le fichier "Base.html", comme pour l'inscription, il y aura un bouton qui permettra d'accéder au profil de l'utilisateur s'il est connecté à la plateforme. Une page "profile.html" est créée avec les données de l'utilisateur et les formulaires "u\_formIcrispy" et "p\_formIcrispy" sont inclus dans cette page.

Dans le fichier "forms.py", une classe "UserUpdateProfil" permet la modification de l'email et du nom d'utilisateur, et une autre classe "ProfileUpdateForm" permet la modification de l'image de profil. Dans le fichier "models.py", une fonction permet de sauvegarder ces modifications et de mettre toute image à une hauteur et une largeur convenable, quelle que soit l'image importée. Dans le fichier "views.py", une fonction vérifie, comme pour l'inscription, si le formulaire de modification est valide et, s'il l'est, enregistre les modifications dans la base de données.

* **Un système de création et de gestion de projets conforme à la description donnée dans cet énoncé (Project, task, subtasks, status).**

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Pour pouvoir crée un projet, j’ai simplement repris la fonction de création de post qui étais déjà implémenter pour cela, j’ai modifié les posts afin que ce soit un projet qui soit créé. Lorsque l’utilisateur connecter voudra créer un projet, il devra simplement appuyer sur le bouton prevu pour cela (New Project).

Une fois que cela sera fait, l’utilisateur arrivera sur une page ou il devra ajouter le titre, les contributeurs les taches, sous taches ainsi que les statuts souhaités pour son projet dans des char « Fields » ou « Tex Fields ».

(J’expliquerais le système de priorité dans la section prevu pour cela)

Je défini les champs titre, contributeurs, statuts ainsi que les taches à l’interieur du « models.py » ensuite j’ajoute ces champs dans le « views.py », plus exactement dans une fonction « PostCreateViews » qui prendra en compte les champs nécessaires pour la création du post et les affichera. L’utilisateur qui créera le projet deviendra automatiquement le chef du projet (l’auteur du projet). Ceci sera fait encore une fois dans une autre fonction de « PostUpdateViews » afin de pouvoir modifier ou supprimer les éléments du projet, une fois qu’il a été créé. Pour donner vie à cela, les migrations vont être effectué via le PowerShell.

Si l’utilisateurs souhaite ajouter plusieurs contributeurs, taches, sous taches ou statuts, il devra simplement terminer par un « ; » à chaque ajout supplémentaire. Voici un exemple :

**Une image contenant texte

Description générée automatiquement**

Ceci est possible grace aux codes que j’ai ajouté dans « models.py », une fonction def « get\_all\_tasks », qui prendra tout ce que j’ai écrit dans l’image ci-dessus, à l’aide d’un for il ajoutera dans un tableau toutes les taches et sous tache une par une dès que l’élément « ; » est retrouve, quand cela sera fait, il stockera tout cela dans le tableau et dans le « post-detail.html », un simple for parcourra toutes les taches et les affichera. Le même principe est appliqué pour les statuts ainsi que les contributeurs.

Une fois que tout cela sera effectué correctement un simple bouton « submit » sera affiché afin de sauvegarder le projet avec la description donnée dans l’énoncé et le projet s’affichera dans le « Home.html » qui est la page sur la premiere photo ci-dessus.

Faire tout cela a été très compliqué pour moi, cela m’a pris plusieurs semaines, je n’arrivais à faire en sorte de crée un nombre de tache définit par l’utilisateur seulement, a la base j’avais tout hard coder et j’avais mi seulement un certain nombre de champs suffisant pour répondre à l’énonce du projet, mais si Alice voulait supprimer une tache elle n’aurait pas pu le faire. Alors j’ai généré ce système de tableau qui m’a demandé énormément de temps, et c’est mon client M. Riggio qui m’a proposé de faire en sorte que dans un seul Text Field il y es tous nos taches. Cette conversation a eu lieu lors de notre dernière itération.

* **Un système d’inscription des utilisateurs à un projet créé par un chef de projet.**

Comme expliqué précédemment, il y aura un champ pour les contributeurs, le chef de projet pourra lui seul choisir les utilisateurs qu’il souhaite afin qu’eux seuls puisse accéder au projet. Ce qui donnera un effet d’inscription automatique à un projet si un utilisateur fait partie des contributeurs du projet en question. Si un utilisateur ne fait pas partie des contributeurs il n’aura pas accès aux informations du projet, il ne pourra pas changer le drag & drop.

Ce qu’il en est des contributeurs et de son affichage il sera similaire qu’aux tâches explique lors de la fonctionnalité précédente.

Une image contenant texte

Description générée automatiquement

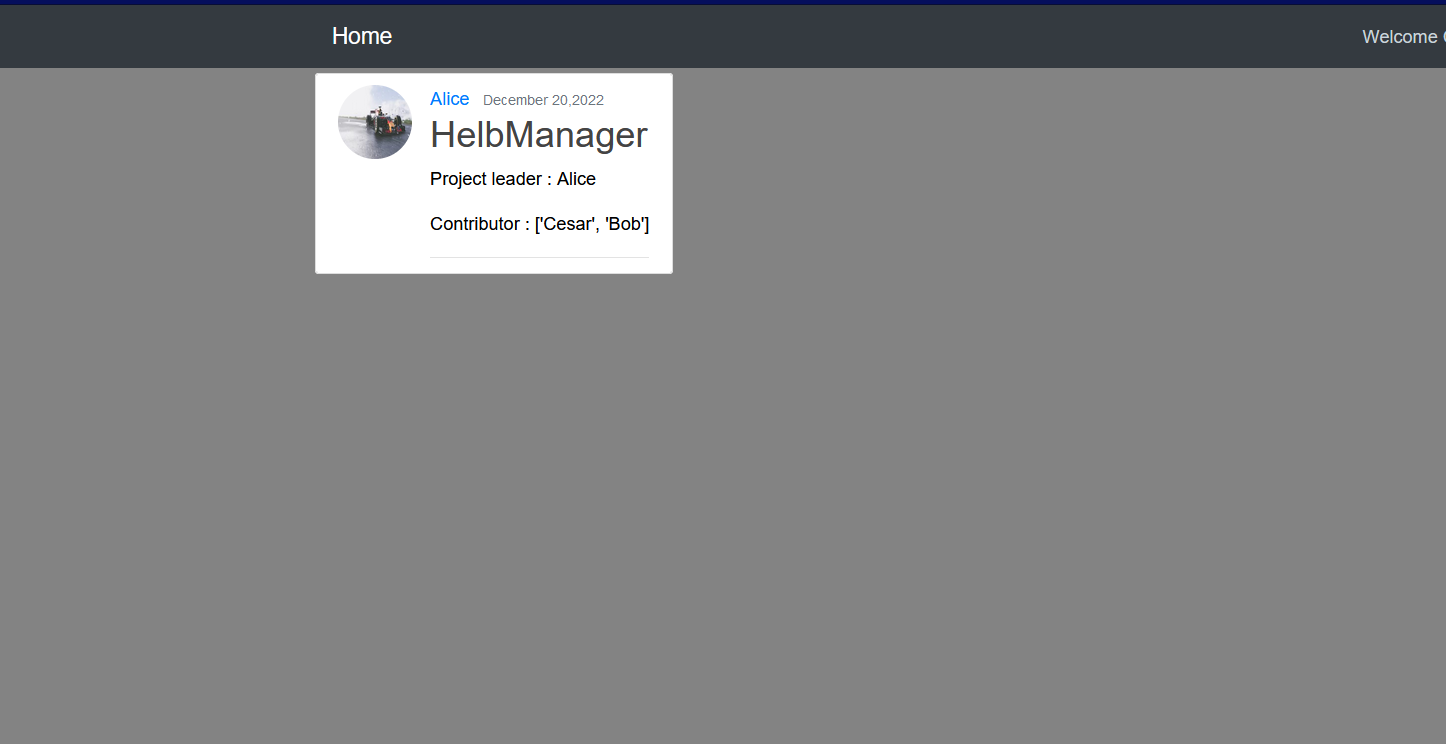
Pour faire en sorte que seul le contributeur puisse voir le drag & drop, j’ai dans mon « post-detail.html » une boucle for qui parcourra le tableau des contributeurs et qui vérifiera pour chaque contributeur si c’est l’utilisateur connecte en ce moment même, si c’est le cas il affichera toutes les taches (drag & drop) sinon il n’affichera tout simplement rien.

Vu d’un contributeur d’un projet inscrit par alice :

Une image contenant texte

Description générée automatiquement

Vu d’un utilisateur qui n’es pas contributeur :



Cependant pour le chef de projet il y aura une autre condition qui vérifiera que si c’est l’utilisateur connecté alors il affichera en plus, la possibilité de modifier les taches, sous taches, contributeur, statuts ou même de supprimer le projet complètement. Ici « l’update » permettra d’accéder à la même page que pour la création de projet mais avec les informations du projet en question afin de pouvoir le modifier.

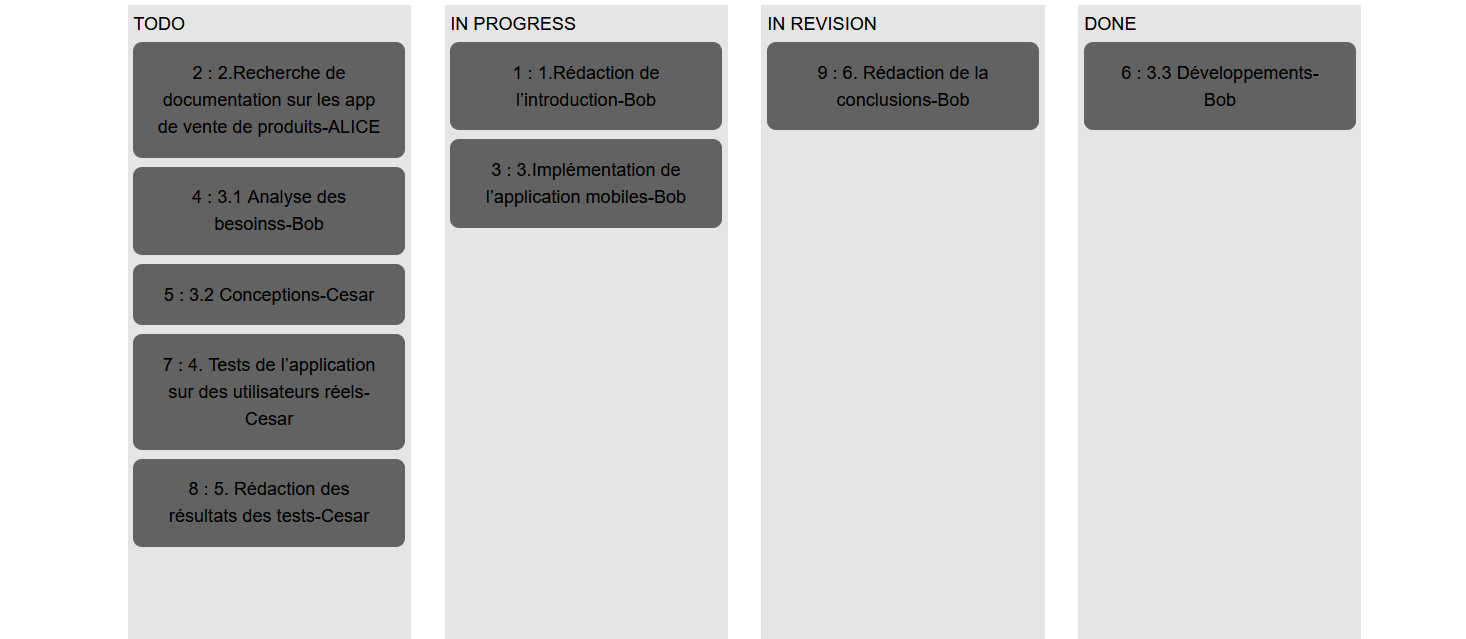
Une image contenant texte

Description générée automatiquement

* **Une interface de type « drag & drop » permettant de notifier le statut de tâches et sous tâches du projet.**

(Source n°2) Pour effectuer l’interface de drag & drop, je me suis inspiré de cette vidéo YouTube, j’ai pu donc ajouter un script.js qui effectue le drag & drop.

J’ajoute la possibilité d’être draggable, cela dans le div qui affichait déjà les taches. Et ces taches en question je dois les ajouter dans un « div app » et ensuite dans un « div lists ». Ensuite dans le css j’ajoute les modifications nécessaires afin d’arranger cela. Je fais ensuite la même chose pour les statuts sauf que les statuts ne seront pas draggable mais seront les cadres dans lequel on pourra ajouter les taches. A Noter que le chef de projet peut ajouter autant de taches et de statuts qu’il le souhaite.



(Source 3,4) Malheureusement le script fourni de la vidéo n’était pas complet, lorsque j’effectuais un drag & drop et que je rafraichissais la page, la page revenait à ce qu’elle était au début, j’ai eu ce probleme pendant une dizaine de jour, ca a été compliqué pour moi de reussir cela, mais j’ai pu comprendre comment faire grace au stockage local de javaScript.

Alors le script a du être modifier afin de pouvoir enregistrer les changements efféctué du drag & drop même si je quittais la page.

Finalement, il fallait ajouter des « ID » à chaque tache et sous taches, pour cela j’ai utilisé un forloop.counter, et il a aussi fallu modifier une partie du code « script.js » pour le dra & drop : Il y a 2 variables qui sélectionnent tous les éléments de la liste et toutes les listes de la page (list\_items, lists).

-La fonction « saveDraggedItemLocation » empêche le comportement par défaut de l'événement de se produire, ensuite, elle récupère l'élément cible de l'événement "drop" et le stocke dans la variable « targetList ». Elle vérifie ensuite que l'élément glissé-déposé (draggedItem) et l'élément cible (targetList) sont bien définis avec une condition. Pour finir la condition enregistre l'emplacement de l'élément déplacé dans le stockage local.

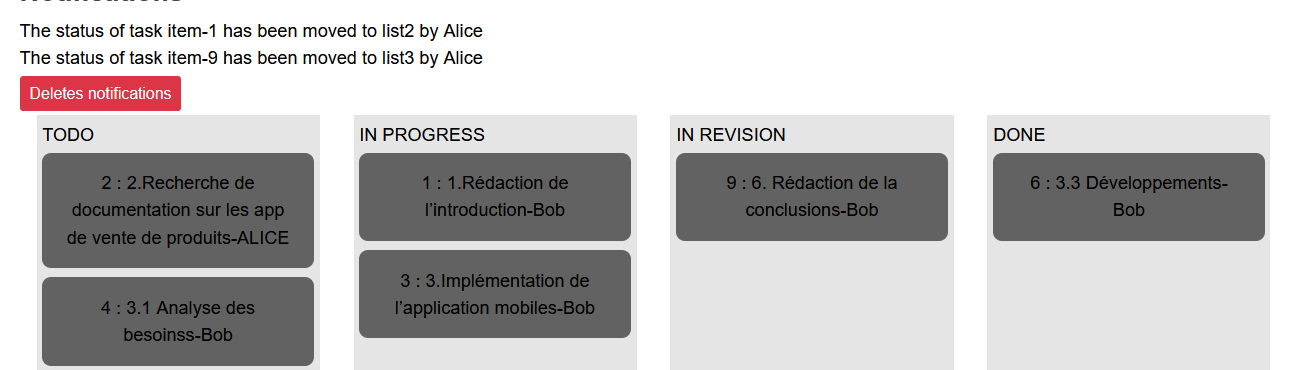
- La fonction « restoreDraggedItemLocations » récupère tous les éléments de liste et les stocke dans la variable « list\_items ». Chaque élément de liste est ajouté avec une boucle for. La varaible « location » récupère l’enplacement enregistré dans le stockage local. Si un emplacement a été enregistré. La variable « targetList » récupère l'élément de liste. Vérifie ensuite que l'élément cible est bien une liste. Pour finir il ajoute l'élément de liste courant à l'élément de liste cible avec la méthode « append() ».

Afin d’assigner les taches aux contributeur, alice ajoutera un « - » suivi du nom du contributeur a chaque fin de taches. Si bob est connécté il pourra seulement drag & drop les taches qui lui sont assigné en regardant si son nom est dessus. Pour le reste des taches il pourra les voirs mais modifié leurs statuts. Le chef de projet peut modifier toutes les taches.

Ceci est effectué dans le « post-detail.html », lorsque j’affiche les taches je verifie s’il y a dedans le nom de l’utilisateur connécté, si c’est le cas les taches suivies de l’utilsiateur connécté seront « draggable » et le reste des taches non, ainsi Bob poura effectué le changement de status que pour ces propres taches. Pour ce qui est du chef de projet, si l’autheur du projet est connécté il poura directement modifie le status de toutes les taches.

* **Des notifications en cas de changement du statut d’une tâche ou sous tâche du projet.**





Des notifications ont été ajouté lors d’un changement de status d’une tache, comme vous pouvez le voir, ci-dessus on peut y lire « The status of task item-1 has been moved to list2 by Alice » nous avons le nom de la personne qui a fait le changement ainsi que l’ID de la tache (qui est le premier numero de la tache) ainsi que l’endroit ou la taches a été deposé (chaque status coréspond à une list qui ont chacun un ID). Pour la premiere notification ont peux y lire que la Redaction de l’introduction de Bob à changé de status et est passé à « IN PROGRESS ».

Une notification est prevu aussi lorsque le chef de projet modifie le projet (ajoute, modifie, suprime, une tache) en desous du bouton update on y lit « The project was updated on 07/07/2023 at 14 :08 :03. » ce qui donne la date et l’heure a laquel le chef de projet a modifié le projet.

Pour faire la notification du drag & drop, dans le « script.js » j’ai ajouté à ma fonction du mouvement de drag & drop « saveDraggedItemLocations », un InnerHTML, qui effectue un afichage sur une page html depuis un script la valeur donnée, je reprends donc simplement les ID des taches ainsi que celui des status et je les affiche avec une phrase, j’ajoute aussi le nom de la personne connecter sur le site en ce moment meme. Similaire que pour les taches j’effectue un stockage local pour que les changements sur la page html soit enregistré. J’ajoute auss iun bouton qui suprime ce stockage local et cela sera fessable seulement par le chef de projet.

Pour faire la notification du project update, j’ai ajouté un script qui faisait la meme chose que pour le drag & drop mais cette fois ci seulement lorsque le bouton « update » etait cliqué ainsi une notification apparaitra seulement lorsque le chef aura été sur la page de l’update du projet, malheureusement je n’ai pas réussi à préciser les changements que le chef fera dans la page de l’update du projet.

# **Fonctionnalités supplémentaires**

* **Un système de priorité pour les taches, sous taches.**

Une image contenant texte

Description générée automatiquement

Comme vu precedeemnt, il y avait un champ qui se nomait « Priority » dans la création d’un projet, dans ce champs-là, le chef de projet indiquera l’ordre exact dans lequel les taches devront etre effecuté, et les colaborateur seront tenu de suivre ses instructions.

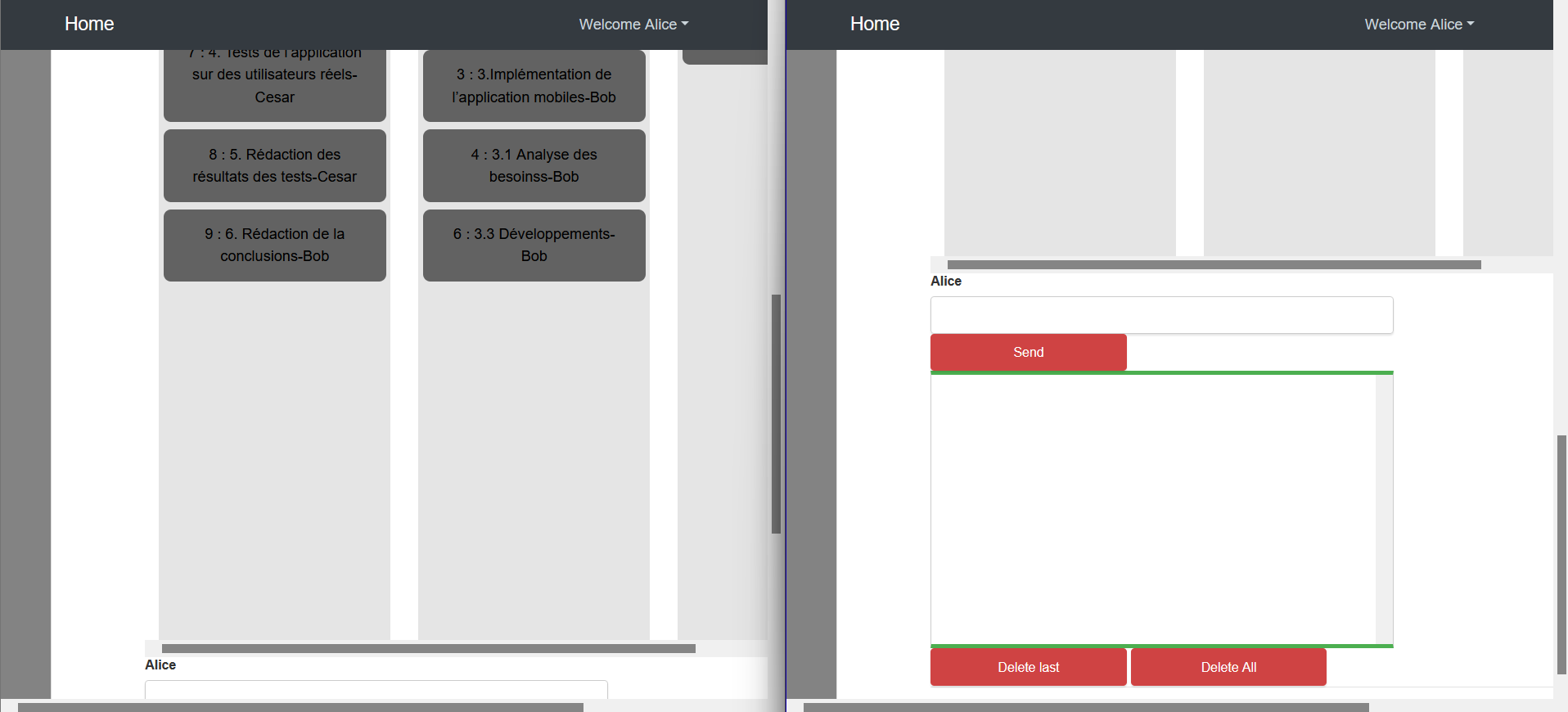
Une image contenant texte

Description générée automatiquement

Voici comment sera affiché les priorités dans la page du projet en question, le nom du chef du projet sera indiqué en prenant l’autheur du projet.

Pour crée cela, j’ai fait la meme chose que pour le système de création de tache mais pour les priorités. Et j’ai juste ajouté les priorites dans un cadre avec un affichage simple au lieu de le mettre dans le drag & drop.

* **Un système de message entre utilisateurs de projet.**

****

En dessous du drag & drop, un système de message est inplementer pour le chef du projet et les colaborateur, 2 boutons seront implementé et seul le chef de projet poura les utilisé, c’est de suprimer le dernier message et de suprimer toute la conversation.

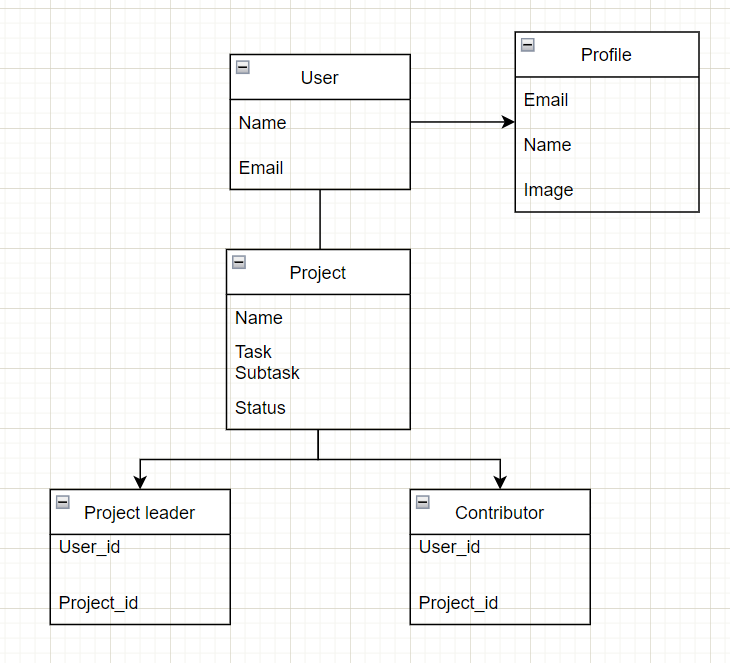
Pour les boutons, ils s’affichent seulement si l’autheur du projet est sur le projet. Pour le système du chat, j’ai ajouté un container pour l’affichage avec une bande de texte dans l’HTML. Mais tout se passe dans le « script.js ». Il y aura comme pour le drag & drop, un système d’enregistrement des mesages envoyé dans le stockage local, (comprendre cela ma aidé enormement pour tout le projet) et tout simplement lorsqu’un utilisateur ecrit et envoie un message il est affiché dans la zone prevu pour cela, le nom de l’utilsiateur est affiché avant le message pour savoir qui envoie le message.

Une image contenant texte

Description générée automatiquement

Pour ce qui est de la supression des messages, c’est simple, on vide le stockage local en ajoutant a la place de tous les messages une donné vide et il y n’aura juste plus rien. Pour suprimer le dernier message, on va juste prendre dans une fonction le dernier message envoye et le suprimmer de l’affichage.

# **Analyse**



Tous les utilisateurs auront un nom et un email, ils auront un profil, et ils pourront avec ses données ainsi que leur image de profil par défaut.

Ils peuvent tous crée un projet avec un nom, des taches, des sous taches ainsi que des statuts, et en fonction de cela, un utilisateur est soit un chef de projet soit un contributeur, ils peuvent être les deux ou aucun des deux en fonctions de chaque projet crée. Ils peuvent donc participer à plusieurs projets. Chaque chef de projet aura son id enregistré en tant que chef de projet et assignera des contributeurs et enregistra leur id comme tel. Chaque utilisateur aura donc un accès pour leur projet auquel ils font partie.

# **Limitations et développement futur**

Les limites de votre application web,

par exemple : dans quels cas d’utilisation votre application pourrait ne pas

fonctionner comme prévu ? Y a-t-il des aspects techniques du site qui n’ont

pas été traité ? Si vous aviez plus de temps pour le projet, qu’auriez-vous

amélioré ? Plusieurs points de vue sont possibles, il revient au groupe

d’étudiant de choisir les points qu’il considère les plus pertinents pour réaliser

son autocritique.

# **Conclusion**

Il est vrai que j’ai pu énormément appris lors de ce projet, l’utilisation de Django et python, ce sont des choses très intéressante et surtout très importante pour mon développement en tant que programmeur.

Mais je pense contrairement au projet de java que celui-ci aurait dû être fait en groupe car ce projet a été à mon avis beaucoup plus compliqué que celui de java, énormément de chose ne fonctionnait pas, j’ai pris beaucoup de temps a faire des recherches et finalement je passais plus de temps a cherché quelque chose qui fonctionne que coder moi-même. Je pense donc que ce projet avec une autres personne aurait pu être vraiment beaucoup plus simple et tout aussi intéressant à le faire.

# **Source**

M.Corey Shafer:

1 : <https://www.youtube.com/watch?v=UmljXZIypDc&list=PL-osiE80TeTtoQCKZ03TU5fNfx2UY6U4p>

Drag & Drop :

2 : <https://www.youtube.com/watch?v=tZ45HZAkbLc>

Local Storage:

3 : <https://openclassrooms.com/fr/courses/7697016-creez-des-pages-web-dynamiques-avec-javascript/7911201-sauvegardez-les-donnees-dans-le-localstorage>

4 : <https://chat.openai.com/chat>

Notification :

https://www.youtube.com/watch?v=6WUJdEfftK8