

# Easybee application java:

## Compétence 1.1 :

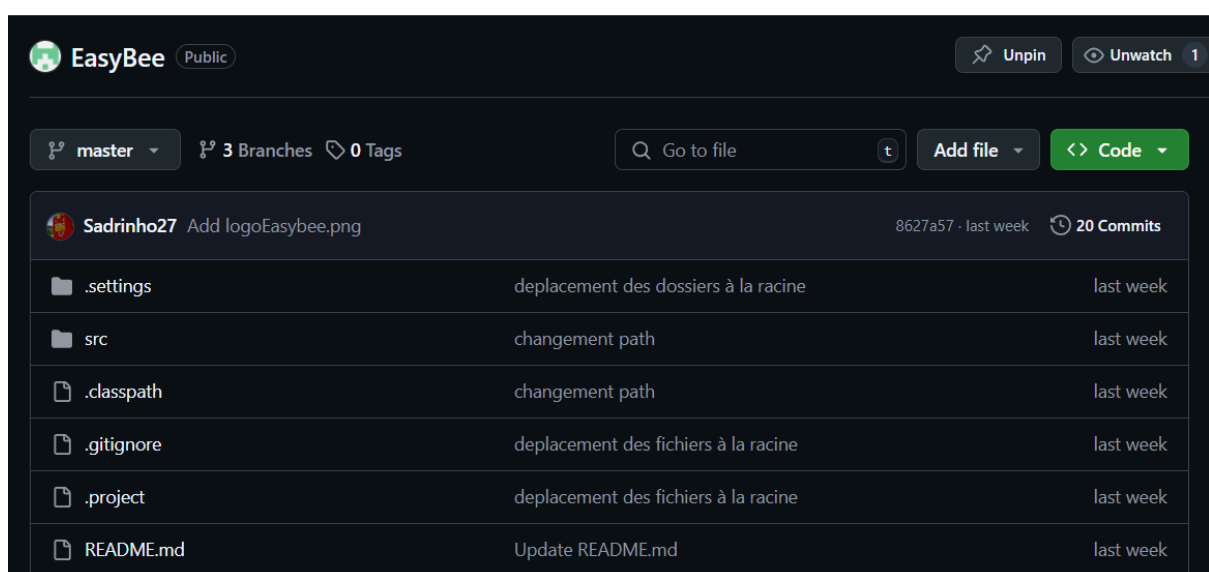
Mettre en place et vérifier les niveaux d'habilitations associés à un service :

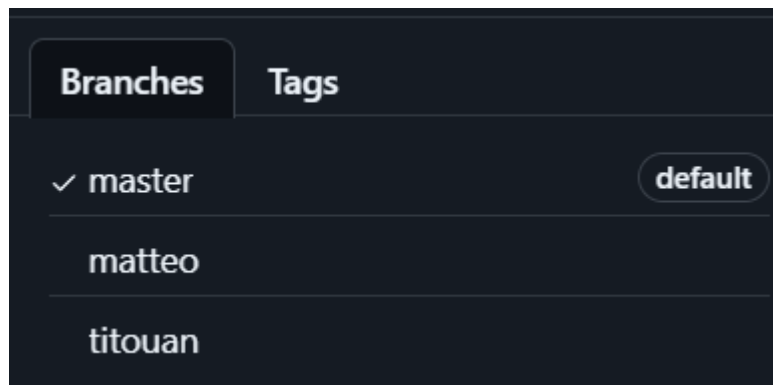
Création de deux comptes concernant la base de données easybee.

<input type="checkbox"/>	admin_easybee	%	spécifique à cette base de données	ALL PRIVILEGES
<input type="checkbox"/>	user_easybee	%	spécifique à cette base de données	SELECT, INSERT

## Utilisation de Github :

Nous avons utilisé Github comme solution de gestion de version. La branche principale du dépôt est la branche "Master", elle contient la version finale de l'application. Chaque membre de l'équipe possède sa branche dédiée. Les nouvelles fonctionnalités sont développées sur la branche de la personne qui l'a développée. A la fin chaque personne a fait un merge pour intégrer les modifications sur la branche "Master".





## Compétence 1.4 :

Analyser les objectifs et modalités d'organisation d'un projet :

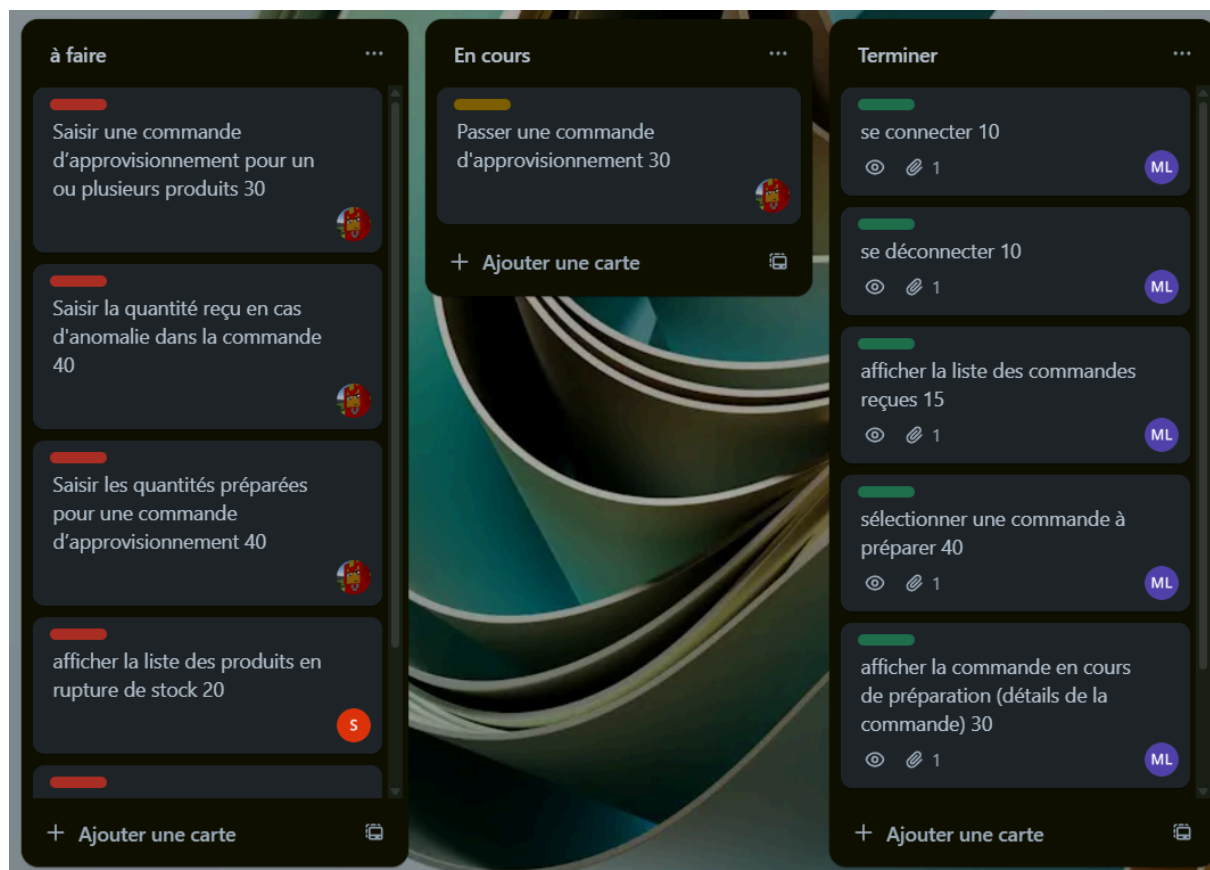
On a utilisé la méthode Scrum pour la gestion de projet. On a défini les user stories, puis réalisé le story mapping.

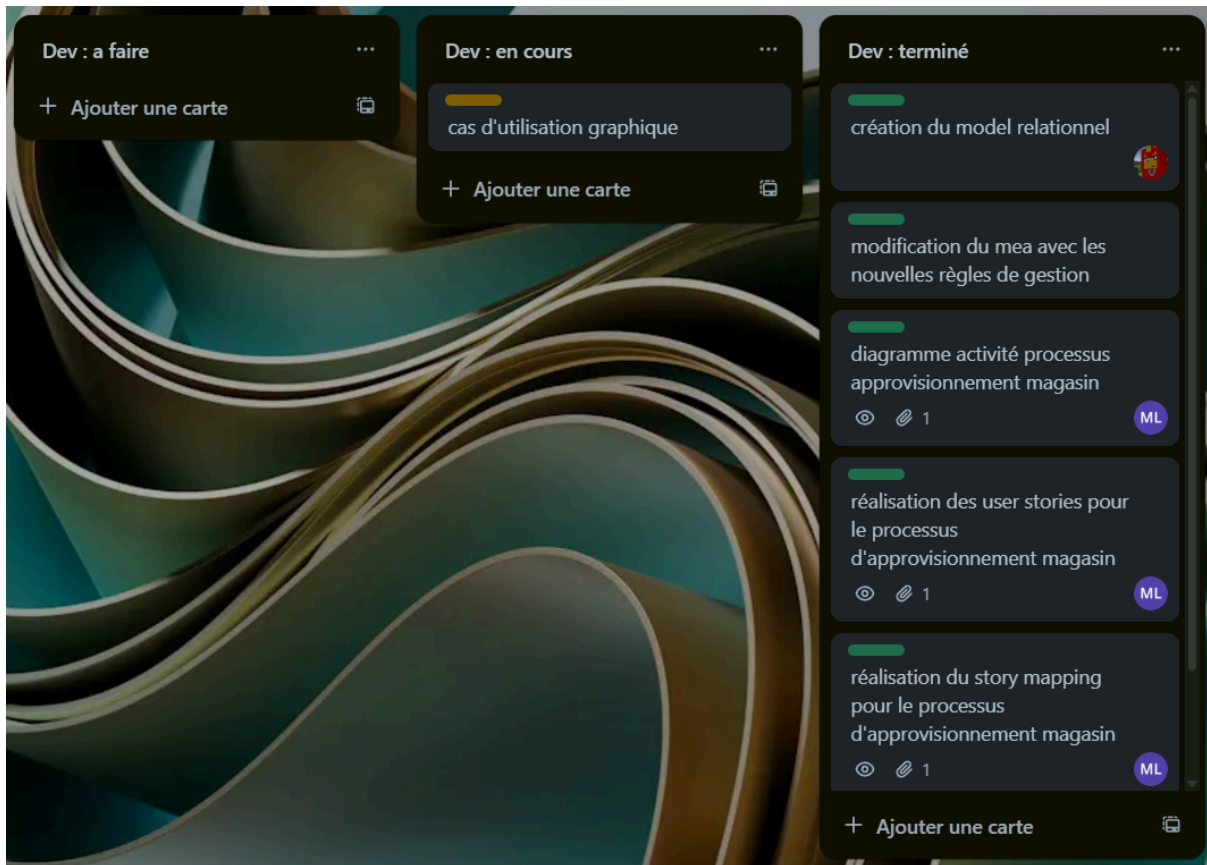
En tant que	Je souhaite	Afin de
Vendeur conseil	Passer une commande d'approvisionnement pour un ou plusieurs produits	avoir du stock pour assurer une vente dans la journée
Vendeur conseil	Afficher une commande d'approvisionnement reçue	d'effectuer une validation complète ou partielle de la livraison et mettre à jour le stock du magasin
Vendeur conseil	Valider la réception d'une commande	mettre à jour le stock et clôturer la commande d'approvisionnement
Vendeur conseil	saisir les quantités reçues en cas d'anomalies ou de livraison partielle	mettre à jour le stock <del>rellement</del> reçu
Vendeur conseil	envoyer un état récapitulatif de la quantité réceptionnée à l'entrepôt	signaler une erreur entre la quantité livrée et la quantité inscrite sur le bon de livraison
Responsable entrepôt	afficher la liste des produits en rupture de stock au magasin	effectuer un <del>réapprovisionnement</del> dans la journée
Responsable entrepôt	Saisir une commande d'approvisionnement pour un ou plusieurs produits	pouvoir livrer rapidement le magasin
Préparateur de commande	<del>selectionner</del> une commande à préparer	pouvoir livrer rapidement le magasin
Préparateur de commande	afficher la commande d'approvisionnement en cours de préparation	saisir les quantités à livrer
Préparateur de commande	saisir les quantités préparées pour une commande d'approvisionnement	pouvoir livrer rapidement le magasin
Responsable entrepôt	imprimer les bons de livraisons des commandes préparées	déclencher la livraison au magasin
Responsable entrepôt	modifier la quantité en stock d'un produit	afin de corriger une <del>erruer</del> de stock

Versions	magasin	Authentification	Entrepôt
Version 1	Passer commande d'approvisionnement	se connecter	afficher liste produit en rupture de stock
	valider la réception d'une commande	se déconnecter	saisir commande d'appro pour un ou plusieurs produits
	saisir quantité reçu en cas d'anomalie		afficher une commande reçu
	envoyer état récapitulatif de la quantité reçu		sélectionner une commande à préparer
Version 2			afficher la commande en cours de préparation
			saisir quantités préparées pour commande d'appro
		modifier mot de passe	imprimer bons de livraisons des commandes préparées
		modifier identifiant	modifier la quantité de stock d'un produit

## Planifier les activités :

On a utilisé Trello pour nous organiser. On a fait une catégorie “à faire”, une autre “en cours” et une “terminé” dans lesquelles on a classé les tâches à réaliser en fonction de leur avancée. Chaque membre de l'équipe s'est assigné plusieurs tâches.





## Compétence 1.5 :

Accompagner les utilisateurs dans la mise en place d'un service :

On a créé un fichier readme dans lequel on a mis le contexte ainsi que les différentes fonctionnalités. Il y a également un guide sur comment installer le projet (prérequis + guide d'installation).

## Contexte

EasyBee est une entreprise fictive spécialisée dans la vente de matériel apicole (ruches, combinaisons, pots, etc.). Face à divers problèmes rencontrés, l'entreprise a décidé d'optimiser son processus d'approvisionnement. Pour ce faire, une solution informatique sous la forme d'une application Java a été mise en place. Cette application, destinée aux salariés de l'entreprise, permet de passer des commandes d'approvisionnement auprès de l'entrepôt lorsqu'un produit est en rupture de stock en magasin.

## Fonctionnalités

### Magasin :

- Passer une commande pour un produit en rupture de stock.
- Confirmer que la commande reçue est correcte.
- Voir l'état d'une commande (En attente, En cours, Terminée).

### Entrepôt :

- Voir la liste des commandes en attente.
- Afficher les détails d'une commande pour la préparer.
- Changer l'état d'une commande (En attente, En cours, Terminée).

## Prérequis

- Java
- IDE compatible (Eclipse, IntelliJ, etc.)
- MySQL ou un autre serveur de base de données relationnelle

## Installation

### Base de données :

1. Importez le fichier *bdd.sql* dans votre serveur de base de données pour créer la structure de la base.
2. Importez le fichier *data.sql* pour y ajouter les données de test.

### Application :

1. Clonez ou téléchargez ce dépôt.
2. Importez le code source dans votre IDE préféré :
  - Ouvrez votre IDE.
  - Sélectionnez l'option "Importer un projet existant" ou équivalent.
3. Configurez les paramètres de connexion à la base de données dans le fichier *ConnexionBdd.java*.

### Lancement :

1. Ouvrez le fichier *pageConnexion.java*.
2. Exécutez le code pour lancer l'application.
3. Connectez-vous avec les identifiants présents dans les données de test.

# Easybee application web :

## Compétence 1.1 :

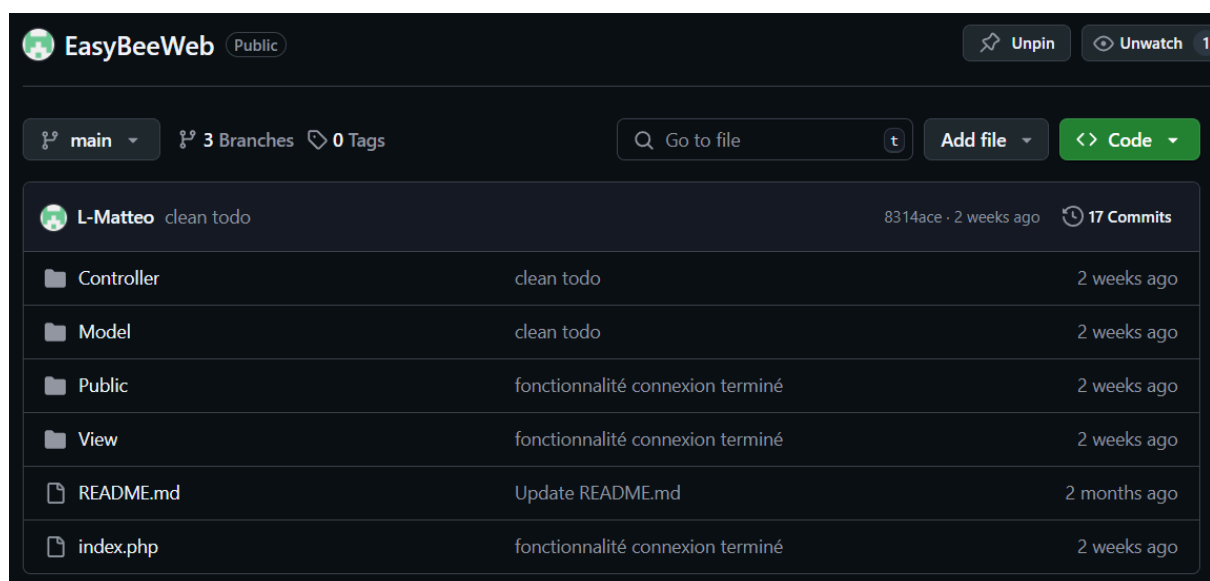
Mettre en place et vérifier les niveaux d'habilitations associés à un service :

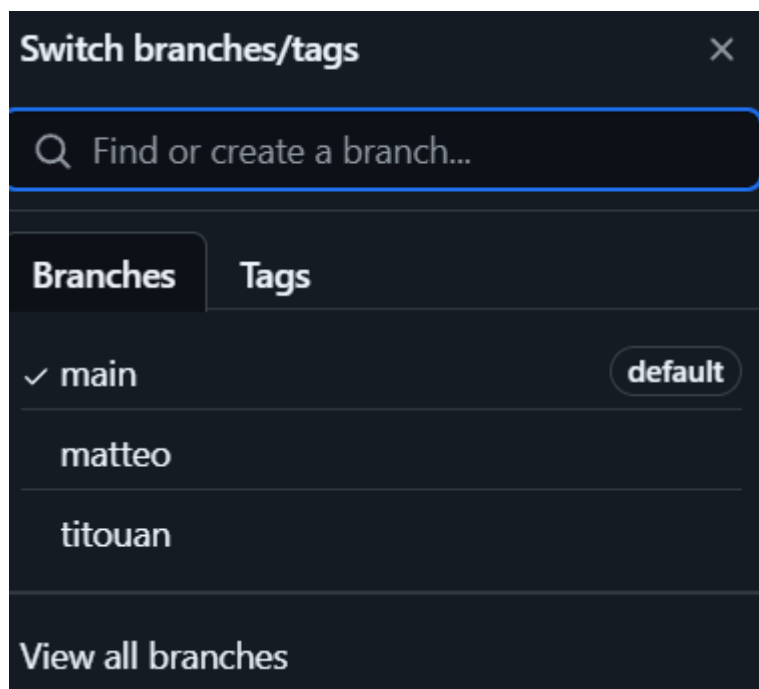
Création de deux comptes concernant la base de données easybee.

<input type="checkbox"/>	admin_easybee	%	spécifique à cette base de données	ALL PRIVILEGES
<input type="checkbox"/>	user_easybee	%	spécifique à cette base de données	SELECT, INSERT

## Utilisation de Github :

Nous avons utilisé Github comme solution de gestion de version. La branche principale du dépôt est la branche "Master", elle contient la version finale de l'application. Chaque membre de l'équipe possède sa branche dédiée. Les nouvelles fonctionnalités sont développées sur la branche de la personne qui l'a développée. A la fin chaque personne a fait un merge pour intégrer les modifications sur la branche "Master".





### Compétence 1.3 :

### Compétence 1.4 :

Analyser les objectifs et modalités d'organisation d'un projet :

On a utilisé la méthode Scrum pour la gestion de projet. Les user stories story étaient déjà réalisées.

### Liste des user stories

---

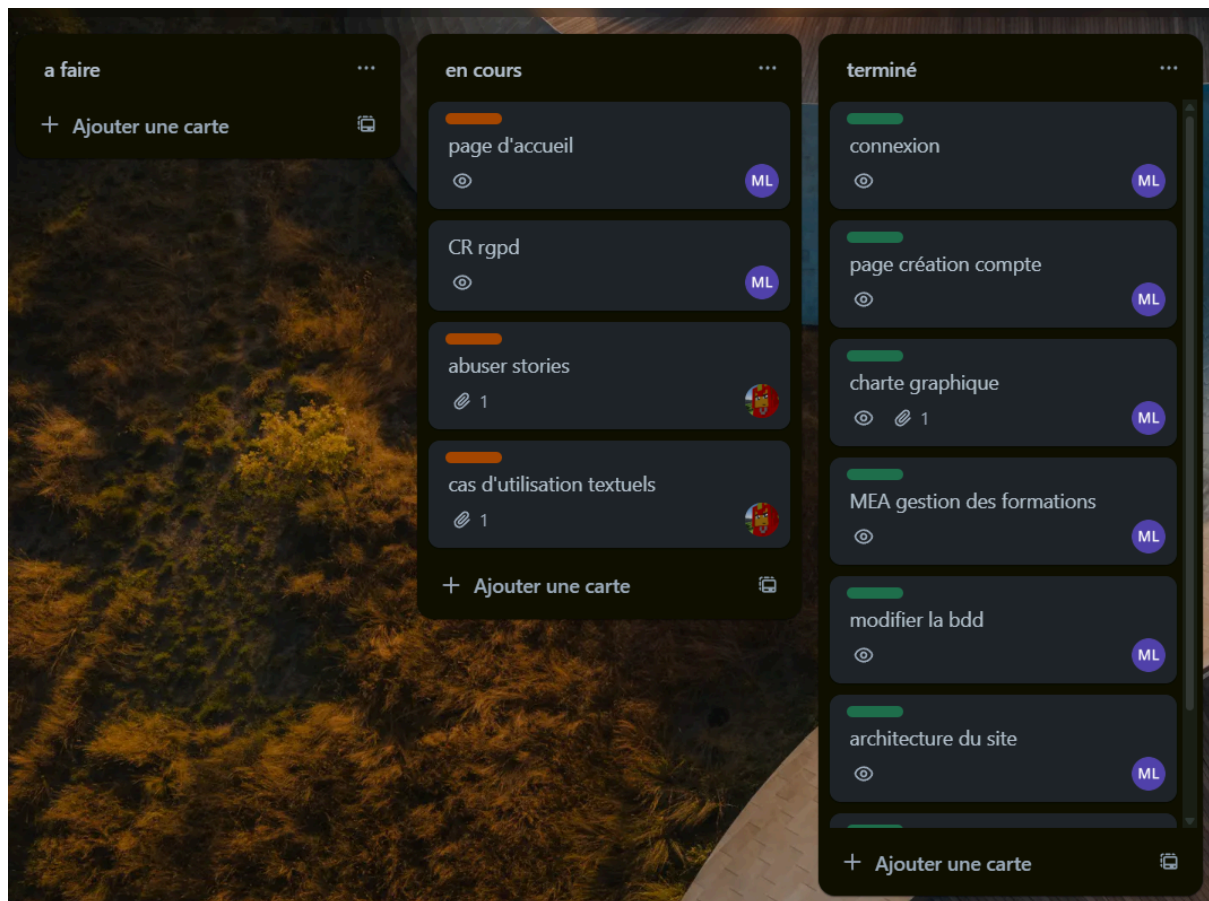
Le Product Owner a écrit les user stories suivantes :

1. En tant que visiteur je veux pouvoir m'inscrire à une formation pour pouvoir y participer
2. En tant que visiteur je veux pouvoir m'inscrire sur liste d'attente à une formation pour pouvoir y participer si il y a des désistement
3. En tant que visiteur je veux pouvoir créer un compte afin de pouvoir m'inscrire à une formation.
4. En tant que visiteur je veux pouvoir m'authentifier afin de pouvoir m'inscrire à une formation.
5. En tant que visiteur je veux pouvoir modifier l'adresse mail de mon compte afin de pouvoir recevoir les informations à la bonne adresse.
6. En tant que visiteur je veux pouvoir afficher la liste des formations par niveau pour pouvoir affiner ma recherche et m'inscrire à une formation.
7. En tant que visiteur je veux pouvoir afficher la liste des formations par thème pour pouvoir affiner ma recherche et m'inscrire à une formation.
8. En tant que visiteur je veux pouvoir afficher le détail d'une formation pour pouvoir vérifier que cette formation me convient et m'inscrire.
9. En tant que visiteur je veux pouvoir afficher la liste des formations par ordre chronologique (plus récent au plus ancien) afin de visualiser les prochaines formations proposées.
10. En tant que responsable de formation, je veux que le système supprime automatiquement les formations qui n'ont pas été réglées afin de pouvoir libérer des places
11. En tant que responsable de formation, je veux pouvoir afficher les inscriptions sur liste d'attente qui peuvent passer en inscription définitive afin de contacter les visiteurs pour confirmer leur demande.
12. En tant que responsable de formation, je veux pouvoir afficher une inscription sur liste d'attente pour la passer en attente de paiement pour inscrire de nouveaux participants.
13. En tant que responsable je veux pouvoir afficher la liste des stagiaires à une formation afin de pouvoir préparer le matériel nécessaire.

### Planifier les activités:

On a utilisé Trello pour nous organiser. On a fait une catégorie “à faire”, une autre “en cours” et une “terminé” dans lesquelles on a classé les tâches à réaliser en fonction de leur avancée. Chaque membre de l'équipe s'est assigné plusieurs tâches.





## Compétence 1.5 :

Accompagner les utilisateurs dans la mise en place d'un service :

On a créé un fichier readme dans lequel on a mis le contexte ainsi que les différentes fonctionnalités. Il y a également un guide sur comment installer le projet (prérequis + guide d'installation).

## Contexte

EasyBee est une entreprise fictive spécialisée dans la vente de matériel apicole (ruches, combinaisons, pots, etc.) et la formation en apiculture. Pour optimiser la gestion des formations, une application web a été développée. Cette application est destinée :

- aux **clients**, qui peuvent consulter les formations disponibles et s'y inscrire,
- aux **administrateurs**, qui peuvent gérer les inscriptions et trier les participants selon divers critères.

## Fonctionnalités

**Clients :**

- Voir les détails des formations proposées par l'entreprise (dates, contenu, prix, etc.).
- S'inscrire à une formation.

**Administrateurs :**

- Afficher la liste des clients inscrits à une formation spécifique.
- Trier les inscriptions selon différents critères (sur liste d'attente, en attente de paiement, date.).

## Prérequis

- PHP (8.2 minimum)
- IDE compatible (VS Code, PHPStorm, etc.)
- Serveur web (WAMP, XAMPP, etc)
- MySQL ou un autre serveur de base de données relationnelle

## Installation

**Base de données :**

1. Importez le fichier *bdd.sql* dans votre serveur de base de données pour créer la structure de la base.
2. Importez le fichier *data.sql* pour y ajouter les données de test (formations, utilisateurs, etc.).

**Application :**

1. Clonez ou téléchargez ce dépôt.

**Lancement :**

1. Accédez à l'application : ouvrez votre navigateur et rendez-vous sur <http://localhost>.

## Notes supplémentaires :

Pour vous connecter, utilisez les identifiants présents dans les données de test.