

Patrimoine Informatique MarieTeam

Norme de programmation

- Suivre le modèle MVC (Model-View-Controller) : s'assurer que la logique d'affichage d'une page, la logique métier et la logique d'accès aux données sont bien séparées.
- Écrire un code lisible : utiliser une indentation et une mise en forme appropriées pour faciliter la lecture du code et son maintien.
- Commenter le code : ajouter des commentaires pour expliquer le but et le fonctionnement de chaque fonction ou méthode.
- Utiliser des fonctions et des méthodes modulaires : diviser le code en fonctions et en méthodes réutilisables pour faciliter la maintenance.
- Éviter les noms de variables trop courts ou trop longs : utiliser des noms de variables clairs et concis.
- Éviter la duplication de code : ne copier pas et ne coller pas de code similaire plusieurs fois, créez des fonctions ou des méthodes réutilisables.
- Effectuer des tests unitaires : s'assurer que chaque fonction ou méthode est testée et fonctionne comme prévu.
- Pour les fonctions et procédure : première lettre du premier mot en minuscule ensuite première lettre des autres mots en Majuscule

```
function getUtilisateurById($id) {  
    ...  
}
```

- Pour les classes : Première lettre en majuscule

```
class MaSuperClasse {  
    ...  
}
```

- Pour les variables : première lettre du premier mot en minuscule ensuite première lettre des autres mots en Majuscule

```
$idClient = ...
```

Mise en place d'utilisateur BDD MySQL.

Pour éviter de se connecter à la base de données en root ce qui est déconseillé j'ai créé 3 utilisateurs pour les membres du projet et leur ai affecter des droits.

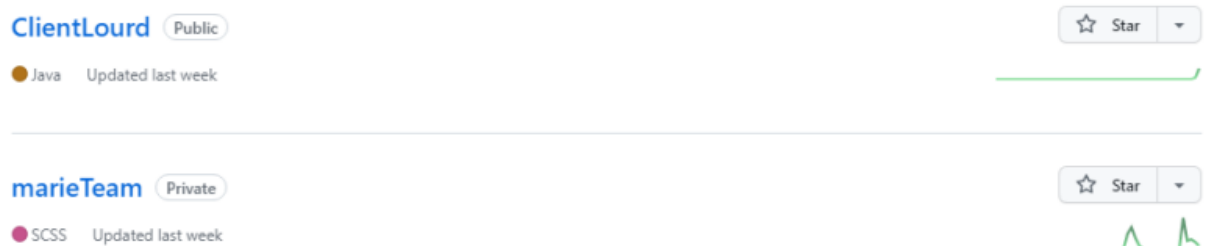
Pour la création de mot de passe fort, j'ai utilisé l'outil fourni par avast.

```
CREATE USER 'yannis'@'localhost' IDENTIFIED BY '^2$RMg$#S';
CREATE USER 'redouane'@'localhost' IDENTIFIED BY '7HhXboxT.F';
CREATE USER 'jawad'@'localhost' IDENTIFIED BY 'ibX]mDv&(8';

GRANT ALL PRIVILEGES ON *.* TO 'yannis'@'localhost' WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* TO 'redouane'@'localhost' WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* TO 'jawad'@'localhost' WITH GRANT OPTION;
```

Procédure d'utilisation d'une solution de gestion de version : Github

- 1) Créer un dépôt pour le projet : une fois connecté à GitHub, cliquer sur le bouton "New" pour créer un nouveau dépôt. Donner un nom au dépôt (Dans notre cas MarieTeam et Client Lourd).



- 2) Initialiser le dépôt avec les fichiers du projet : ouvrir une invite de commande et naviguer jusqu'au répertoire contenant les fichiers du projet MarieTeam. Puis entrer la commande suivante pour initialiser le dépôt Git :

```
git init
```

- 3) Ajouter les fichiers au dépôt : utiliser la commande git add pour ajouter les fichiers de votre projet au dépôt. Par exemple, pour ajouter tous les fichiers du projet MarieTeam, entrer la commande suivante :

```
git add .
```

- 4) Créer un commit : une fois que tous les fichiers sont ajoutés, utiliser la commande git commit pour créer un commit qui représente l'état actuel de votre projet. Assurez-vous d'ajouter un message clair qui décrit les modifications apportées au projet.

```
git commit -m "Ajout du module de Reservation"
```

- 5) Ajouter le dépôt distant : pour synchroniser le dépôt local avec le dépôt GitHub, utilisez la commande git remoteadd suivi du lien du dépôt.

```
git remote add origin https://github.com/Yannis-Alouache/marieTeam
```

- 6) Pousser les modifications vers GitHub : utilisez la commande git push pour pousser les modifications vers votre dépôt distant.

```
git push -u origin master
```

- 7) Mettre à jour le dépôt : pour apporter des modifications au projet MarieTeam, utiliser les commandes git add et git commit pour ajouter et enregistrer ces modifications dans votre dépôt local. Une fois que je suis satisfait des modifications, j'utilise la commande git push pour mettre à jour le dépôt distant sur GitHub.

```
git add .  
git commit -m "Ajout de fonctionnalité à MarieTeam"  
git push
```

Les Normes pour les messages de commit

Pour les messages de commit j'ai utilisé une norme qui permet de se repérer facilement et de suivre efficacement les modifications apportées au projet.

Pour les ajouts de code :

```
git add .  
git commit -m "[+] added reservation module"  
git push
```

Pour les suppressions de code :







```
git add .  
git commit -m "[-] removed files"  
git push
```

Pour les modifications de code :

```
git add .  
git commit -m "[~] modified reservation module"  
git push
```

Respect des normes Github

Commits on Nov 29, 2022

[+] added connexion  Yannis-Alouache committed on Nov 29, 2022	 6b3d0ef <>
[~] fixed template icon  Yannis-Alouache committed on Nov 29, 2022	 9bbe113 <>
[+] add accueil.html.php and img  RedCHK committed on Nov 29, 2022	 f210e4f <>

J'ai réussi à respecter les normes établies pour les commits sur GitHub.

J'ai pris soin d'écrire des messages de commit concis et précis, en fournissant des informations utiles sur les changements que j'ai apportés au code.

En utilisant une syntaxe cohérente et en suivant les conventions établies pour la structure des messages de commit, j'ai pu rendre le processus de collaboration plus facile pour moi-même et pour mes collègues de travail.

J'espère continuer à suivre les normes établies pour les commits de mes projets à l'avenir.

Respect des normes de Programmation

Yannis-Alouache [+] added error handling for reservation e6e7516 last week 83 commits		
controleur	[+] added error handling for reservation	last week
css	[+]	last month
img	[+]	last month
js	[+] add header, style.css, folder js and bootstrap and others and resp...	5 months ago
modele	[+] added error handling for reservation	last week
vue	[+] added error handling for reservation	last week
.DS_Store	[+]	5 months ago
getRacine.php	[+]	5 months ago
index.php	[+] add header, style.css, folder js and bootstrap and others and resp...	5 months ago

```
function getBoatByName($name) {
    $boat = null;

    try {
        $connexion = connexionPDO();
        $query = "SELECT * FROM `bateau` where bateau.nom = ?";
        $stmt = $connexion->prepare($query);

        $stmt->execute([$name]);
        $boat = $stmt->fetch(PDO::FETCH_ASSOC);
    } catch (PDOException $e) {
        print "Erreur !: " . $e->getMessage();
        die();
    }

    return $boat;
}
```

J'ai réussi à garder un environnement de code sain en respectant les normes de programmation que j'avais prévu.

Gestion des demandes et incidents MarieTeam

Lors de la phase de développement j'avais besoin d'un moyen efficace de gérer les incidents et les demandes des utilisateurs. J'ai donc décidé de mettre en place des issues sur Github.

J'ai créé un template d'issue personnalisé pour les incidents et les demandes qui se présentent comme ceci.

- Titre
- Description
- La personne assigné à la tâche
- Le badge Descriptif

The screenshot shows the GitHub 'New Issue' form. At the top, there's a 'Title' field. Below it are 'Write' and 'Preview' tabs, followed by a rich text editor toolbar. The main text area contains the placeholder 'Leave a comment'. Below the text area is a file upload section with the text 'Attach files by dragging & dropping, selecting or pasting them.' and a 'Submit new issue' button. On the right side, there are sections for 'Assignees' (showing 'RedCHK') and 'Labels'. A 'Labels' dropdown menu is open, showing a search bar 'Filter labels' and a list of labels: 'bug' (red circle), 'documentation' (blue circle), and 'duplicate' (grey circle). The 'bug' label is selected.

Lorsqu'un incident était signalé, je commençais par confirmer que le problème existait bien et qu'il s'agissait bien d'un problème avec le site web et non une extension navigateur par exemple. J'attribuais ensuite une priorité appropriée à l'issue en fonction de l'impact sur les utilisateurs et de la complexité de la résolution.















The screenshot shows a GitHub issue titled 'Problème de réservation' with a red 'bug' label. The issue is assigned to 'RedCHK' and has a 'Labels' dropdown menu open, showing the same list of labels as the previous screenshot. The issue is marked as 'opened 8 minutes ago by RedCHK'.

Une fois que le problème était réglé par un push sur le dépôt, je passais l'issue en "complété" accompagné d'un compte rendu d'intervention



Voici un extrait des commits du site web :

Commits on Nov 28, 2022

[+] add Footer RedCHK committed on Nov 28, 2022	 bc8a0ad	
[+] add header, style.css, folder js and bootstrap and others and resp... RedCHK committed on Nov 28, 2022	 5bcddee	
[+] fixed typo Yannis-Alouache committed on Nov 28, 2022	 78eab32	
[~] fixed typo && rm old accueil Yannis-Alouache committed on Nov 28, 2022	 a3aa32b	
[+] added accueil controleur Yannis-Alouache committed on Nov 28, 2022	 3144fe6	
[~] fixed typo Yannis-Alouache committed on Nov 28, 2022	 bdc9464	
[~] rm useless model file Yannis-Alouache committed on Nov 28, 2022	 ebd8c53	

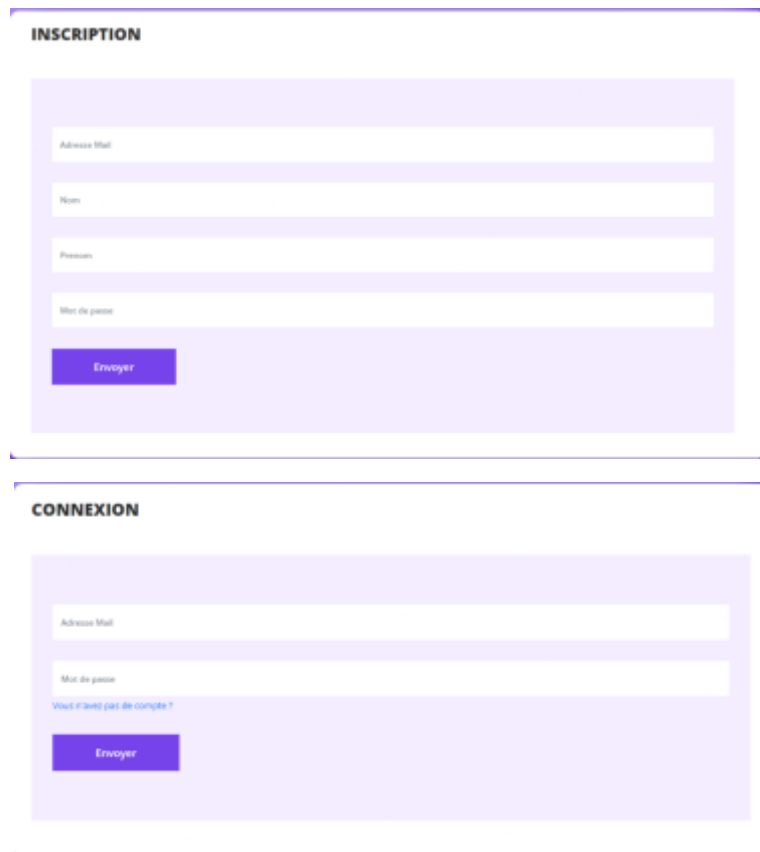
L'utilisation des issues de Github m'a permis de gérer efficacement les incidents et les demandes des utilisateurs de la solution web MarieTeam. Cela m'a également permis de me concentrer sur le développement, en sachant que les problèmes seraient rapidement identifiés et résolus grâce à ce système de gestion d'issues.

Développer la présence de MarieTeam

Documentation Technique

La solution applicative web MarieTeam est un site de réservation maritime en ligne développé en PHP en respectant le modèle MVC (Modèle Vue Contrôleur). Ce document décrit les fonctionnalités du site, les technologies utilisées et les processus de développement et de déploiement.

- Connexion et inscription : Les utilisateurs peuvent créer un compte et se connecter pour accéder aux fonctionnalités du site.



The image displays two screenshots of the MarieTeam web application. The top screenshot is the 'INSCRIPTION' (Registration) form, which includes input fields for 'Adresse Mail', 'Nom', 'Prénom', and 'Mot de passe', followed by an 'Envoyer' (Send) button. The bottom screenshot is the 'CONNEXION' (Login) form, which includes input fields for 'Adresse Mail' and 'Mot de passe', a link for 'Vous n'avez pas de compte ?' (You don't have an account?), and an 'Envoyer' button.

Sélection des secteurs : Les utilisateurs peuvent sélectionner un secteur de liaison



Sélection des liaisons et de la date : Les utilisateurs peuvent sélectionner la date de leur traversée et la liaison qu'il souhaite emprunter.



Sélection des traversées : Les utilisateurs peuvent choisir leur traversée avec des informations complémentaires sur les places restantes.

TRAVERSÉES
Voici les Traversées pour le 2023-03-04

N°	Heure	Bateau	Passager	Véhicule inf. 20t	Véhicule sup. 20t	
4	16:00-00	17	54	105	234	Réserver
5	17:00-00	22	56	145	130	Réserver

Formulaire de réservation : Les utilisateurs peuvent finaliser leur réservation en entrant leur information personnelle

RESERVATION

Nom

Prénom

Adresse

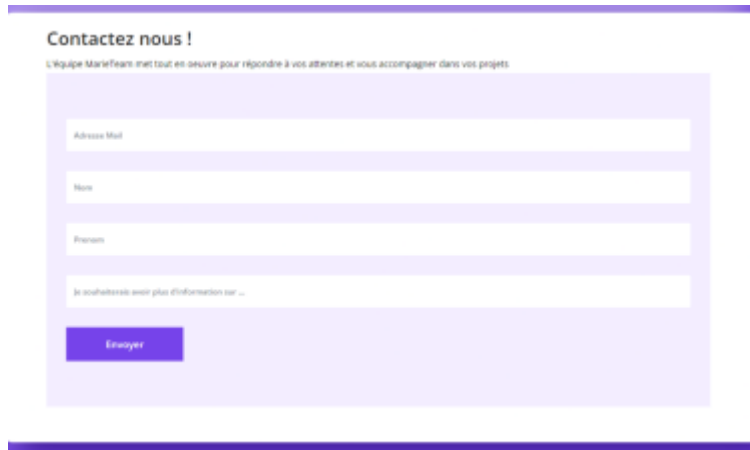
Codpostal

Ville

	Prix en €	Quantité
Adulte	20.00	<input type="text"/>
Enfant 0-10 ans	10.00	<input type="text"/>
Bébé 0-2 ans	5.00	<input type="text"/>
Véhicule inférieur 20t	95.00	<input type="text"/>
Véhicule supérieur 20t	140.00	<input type="text"/>
Porteur	200.00	<input type="text"/>
Changement de car	25.00	<input type="text"/>
Garçon	400.00	<input type="text"/>

[Réserver](#)

Formulaire de contact : Les utilisateurs peuvent remplir ce formulaire pour demander plus d'information, ce qui nous envoie un mail.



The image shows a contact form titled "Contactez nous !" (Contact us!). Below the title, a subtitle reads: "L'équipe MarieTeam met tout en oeuvre pour répondre à vos attentes et vous accompagner dans vos projets." (The MarieTeam team does everything to respond to your expectations and accompany you in your projects). The form contains four input fields: "Adresse Mail" (Email Address), "Nom" (Name), "Prénoms" (First Names), and "Je souhaiterais avoir plus d'information sur ..." (I would like to have more information about ...). A purple "Envoyer" (Send) button is located at the bottom of the form.

La solution web MarieTeam est développée en PHP sans aucun framework.



Les autres technologies utilisées sont : HTML, CSS et JavaScript pour la partie front-end.



MySQL pour la base de données.



Git pour le contrôle de version.



Xampp pour le "déploiement" local.



Audit SEO et performance

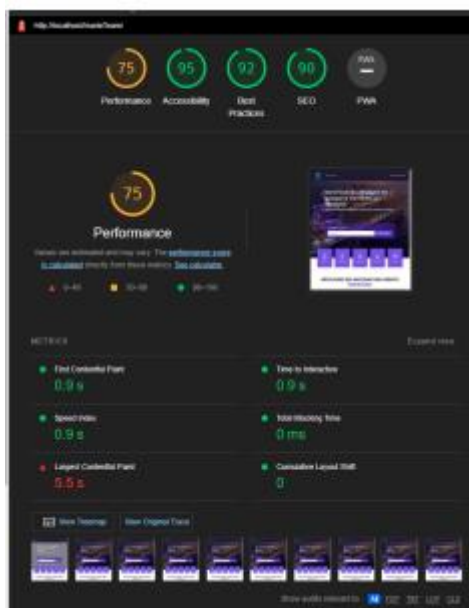
Google Lighthouse est un outil open-source d'audit et de test de performance web. Cet outil, créé par Google, permet aux développeurs et aux propriétaires de sites web d'évaluer la qualité et les performances de leur site en fournissant des recommandations pour améliorer la vitesse, l'accessibilité, la qualité et la sécurité du site web.

Google Lighthouse évalue la performance d'un site web en se basant sur plusieurs critères tels que :

La performance : la vitesse de chargement des pages et les optimisations liées à la performance

L'accessibilité : l'accessibilité du site pour les personnes en situation de handicap et les recommandations pour l'améliorer
Le référencement : les recommandations pour améliorer le référencement du site

La qualité : la qualité du code HTML, CSS et JavaScript, la détection des erreurs et des avertissements.



Mode Projet MarieTeam

Description du besoin et du contexte

MarieTeam est une entreprise de transport maritime basée en France. Elle a obtenu plusieurs contrats avec des conseils généraux pour assurer la desserte maritime d'îles du littoral français par délégation de service public. Parmi les îles desservies, on trouve Belle-Île-en-mer, Houat, Ile de Groix, etc.

MarieTeam a décidé de créer un site web pour faciliter la réservation en ligne des traversées. Le site web sera développé en utilisant le langage de programmation PHP et respectera le modèle MVC (Modèle Vue Contrôleur). Le site web permettra aux utilisateurs de réserver des billets pour les traversées de passagers et de marchandises, de consulter les horaires des traversées et les tarifs.

L'utilisateur aura besoin d'être inscrit et connecté sur la plateforme avant de pouvoir passer une réservation.

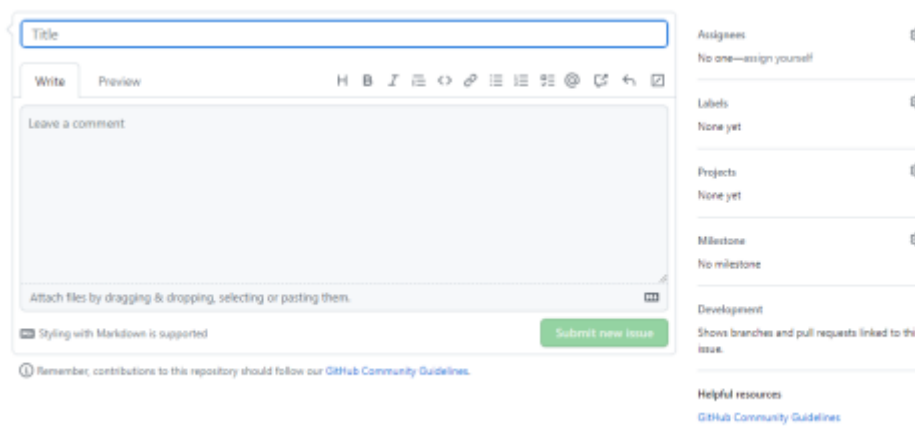
Une fois le site web développé, MarieTeam pourra l'utiliser pour offrir une expérience de réservation en ligne pratique et facile à ses clients, améliorant ainsi la qualité de service offerte par la compagnie. Le site web permettra également à MarieTeam de gérer les réservations et les paiements de manière efficace, réduisant ainsi les tâches administratives et améliorant la productivité globale de l'entreprise.

Description de la méthode d'organisation

La méthode d'organisation sur GitHub pour le projet MarieTeam consiste à utiliser les fonctionnalités intégrées de GitHub pour organiser et gérer le développement du site web de manière efficace.

Pour ce faire, nous avons utilisé les outils suivants :

Les issues : pour identifier les problèmes, les bugs, les demandes de fonctionnalités et les améliorations à apporter au projet. Les issues ont été attribuées aux membres de l'équipe pour qu'ils puissent travailler sur les tâches correspondantes.



Les pulls requests : pour soumettre des modifications de code, permettre aux membres de l'équipe de discuter et de réviser ces modifications avant de les fusionner dans la branche principale.



En utilisant ces outils, nous avons pu suivre l'état du projet en temps réel, attribuer des tâches aux membres de l'équipe, collaborer efficacement et nous assurer que le projet progresse bien. Cette méthode d'organisation sur GitHub nous a permis d'optimiser notre temps de travail et de réaliser le projet MarieTeam de manière efficace.

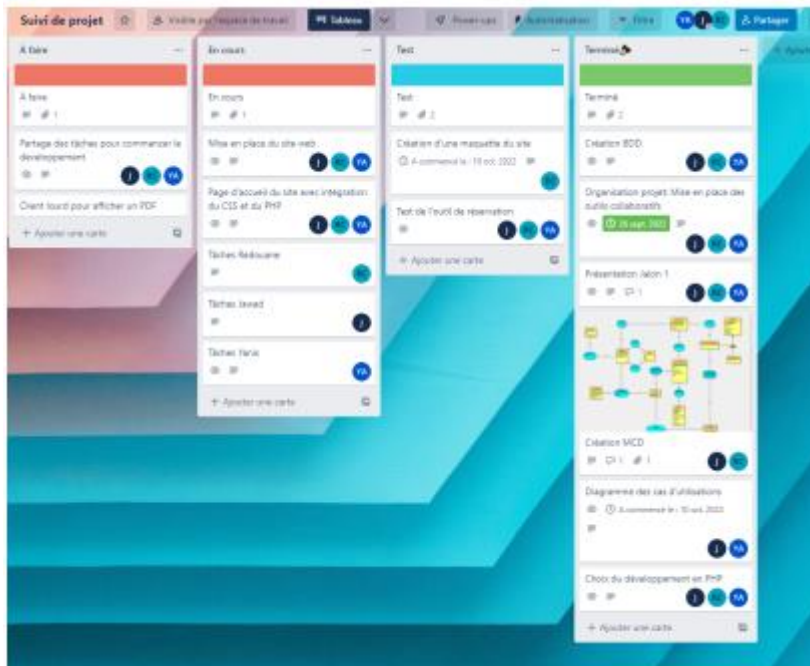
J'ai également utilisé la méthode Kanban. La méthode Kanban est une méthode de gestion de projet qui permet de visualiser et de suivre l'avancement des tâches de manière simple et efficace. La méthode Kanban utilise un tableau à trois colonnes pour visualiser l'état d'avancement des tâches : "À faire", "En cours" et "Terminé". Chaque tâche est représentée par une carte, qui peut contenir des informations sur la tâche, comme une description, une checklist, ou encore le nom de la personne chargée de la réalisation de la tâche.



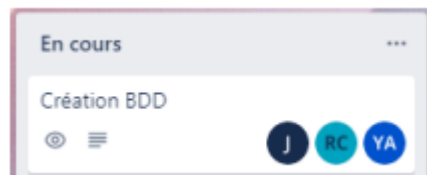
J'ai mis en place la méthode Kanban pour organiser mes tâches de projet et j'ai choisi d'utiliser le logiciel web Trello pour cela.



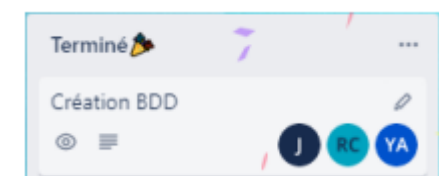
Avec Trello, j'ai pu créer un tableau Kanban personnalisé en utilisant des colonnes pour représenter les différentes étapes de mes tâches. J'ai également pu ajouter des cartes pour chaque tâche, avec des informations détaillées telles que la description, une couleur et une check liste pour la réalisation de chaque tâche.



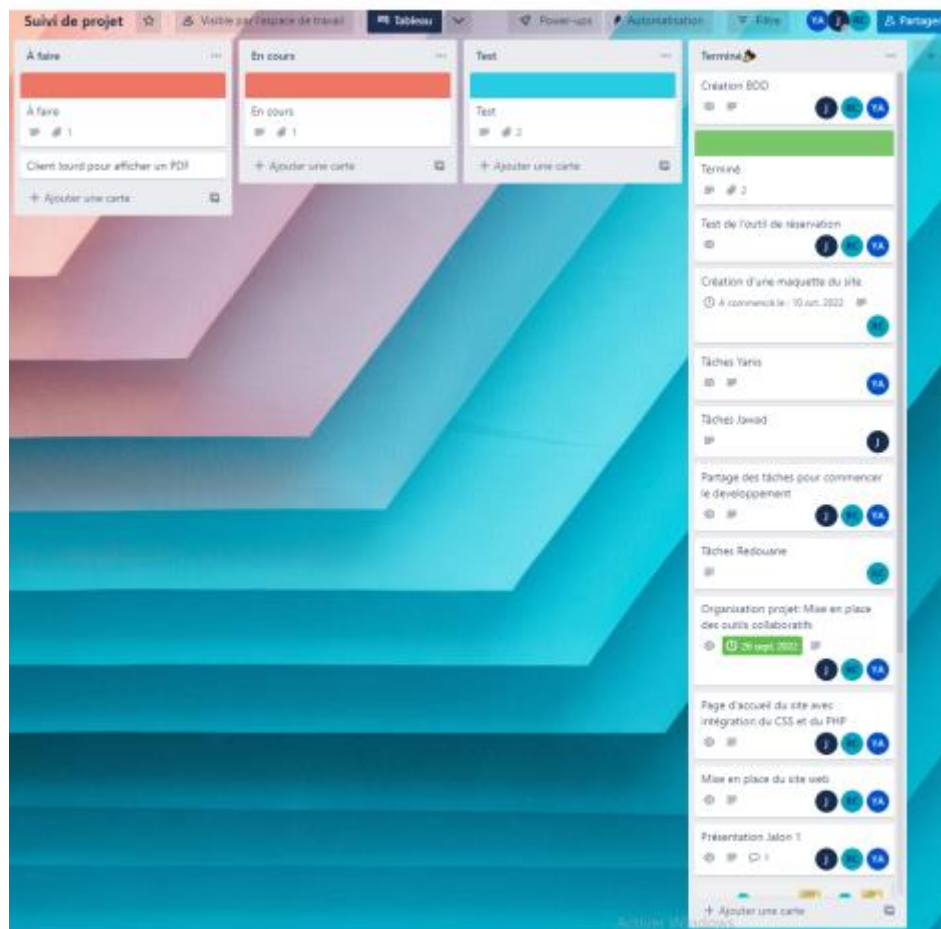
Lorsqu'une tâche est prête à être réalisée, elle est déplacée de la colonne "À faire" à la colonne "En cours". Une fois qu'une sous-étape est terminée je coche sa case.



Lorsque toutes les sous-étapes sont réalisées, la tâche est considérée comme étant terminée et elle est déplacée de la colonne "En cours" à la colonne "Terminé".



Mon Compte Rendu d'activité à la fin du projet



Toutes les tâches prévues au début du projet ont été respectées dans les limites de temps imparti.

Conclusion

Grâce à l'utilisation de Trello et Github, j'ai pu visualiser rapidement l'état d'avancement de chaque tâche, prioriser mes tâches en fonction de leur importance, et limiter le nombre de tâches en cours pour éviter la surcharge de travail. La méthode Kanban en combinaison avec le logiciel web Trello et Git se sont révélées être un moyen efficace pour moi d'organiser mes tâches de projet.