Portfolio - Stage

Avignon Université

Application Web de Suivi de Projets

pour le CHU Mohammed VI d'Oujda

Aimad Hamdaoui

Année Universitaire 2024-2025

Table des matières

1 Contexte du Projet

Ce projet a été réalisé durant mon stage de fin d'études au Centre Hospitalier Universitaire (CHU) Mohammed VI d'Oujda. Le constat de départ était simple : le suivi des projets internes reposait sur des outils bureautiques classiques (fichiers Excel, échanges d'e-mails), ce qui entraînait des pertes d'information, des difficultés de collaboration et un manque de vision globale pour le management.

Mon objectif a donc été de développer une application web centralisée pour pallier ces manques. L'idée était de fournir un outil unique où les équipes pourraient gérer leurs projets, assigner des tâches et suivre leur avancement en temps réel. Pour cela, j'ai utilisé une stack technique moderne : **Node.js** avec **Express.js** pour le serveur, **MongoDB** comme base de données NoSQL (avec l'ODM **Mongoose**), et des vues rendues côté serveur avec **EJS**.

Ce rapport montre comment la réalisation de ce projet m'a permis de mettre en pratique et de valider un ensemble de compétences en développement et en science des données.

2 Validation des Compétences par le Projet

Chaque fonctionnalité de l'application a été une occasion de travailler une compétence précise. La grille ci-dessous résume les compétences que ce projet m'a permis de valider. Les paragraphes suivants détaillent comment, concrètement, j'ai abordé chaque point.

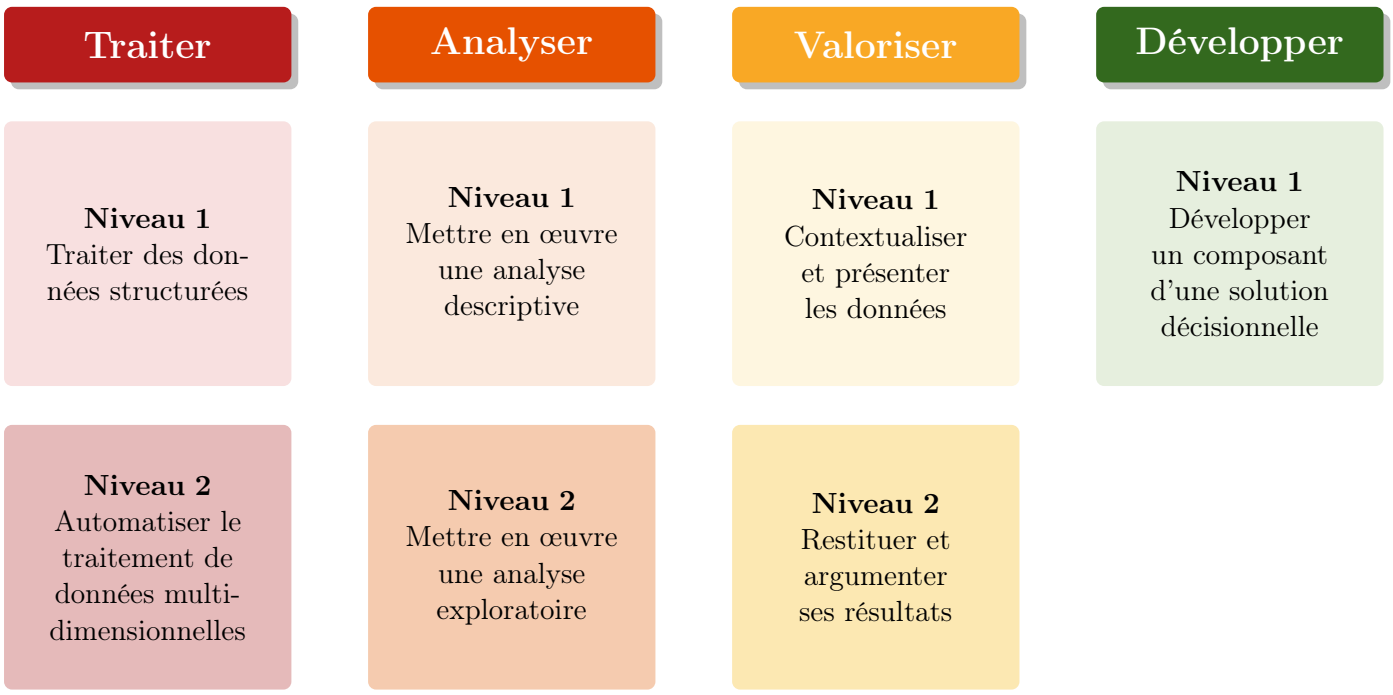


FIGURE 1 – Grille des compétences validées par le projet.

2.1 Traiter : Structurer et Automatiser

Niveau 1 - Données structurées. J'ai mis en place la structure de la base de données avec Mongoose. Concrètement, j'ai défini des schémas stricts pour les projets, les tâches et les utilisateurs. Cela garantit que toutes les données enregistrées sont cohérentes et complètes. L'ensemble des routes de l'API (pour créer, lire, modifier, supprimer) a été développé pour manipuler ces données de façon fiable.

Niveau 2 - Automatisation. Pour aller plus loin, j'ai automatisé certains traitements. Par exemple, quand un chef de projet supprime un projet, une fonction se déclenche automatiquement pour supprimer en cascade toutes les tâches liées. De même, l'assignation d'une tâche à un utilisateur génère une notification, sans aucune action manuelle. Cela illustre bien le traitement de données interdépendantes (projets, utilisateurs, tâches).

2.2 Analyser : De la Description à l'Exploration

Niveau 1 - Analyse descriptive. J'ai développé un tableau de bord statistique pour les administrateurs. Il affiche des indicateurs simples mais essentiels : nombre de projets en cours, tâches en retard, etc. C'est une photographie de l'activité à un instant T, qui permet de voir d'un coup d'œil si tout se passe bien.

Niveau 2 - Analyse exploratoire. Ce même tableau de bord permet d'aller plus loin. En filtrant les données (par exemple par utilisateur ou par période), un manager peut explorer les chiffres pour comprendre pourquoi une équipe est moins performante ou identifier des goulots d'étranglement. J'ai pour cela utilisé des **pipelines d'agrégation MongoDB** (voir Annexe), qui permettent de faire des calculs complexes sur les données à la volée pour répondre à ces questions.

2.3 Valoriser : Donner du Sens aux Données

Niveau 1 - Contextualiser les données. L'enjeu n'était pas seulement de montrer des données, mais de les rendre utiles. J'ai donc conçu des interfaces qui donnent du contexte. Le tableau de bord de chaque utilisateur lui montre *ses* tâches et *ses* projets. La page d'un projet regroupe toutes les informations qui le concernent (participants, tâches, fichiers), au lieu de les disperser.

Niveau 2 - Restituer et argumenter. L'application devient un outil d'argumentation. Avec les graphiques du tableau de bord statistique (Fig. 2), un chef de service peut aller voir sa direction et dire : "Regardez, mon équipe a géré 50% de tâches en plus ce trimestre, j'ai besoin de ressources". Les données ne sont plus de simples chiffres, elles deviennent des preuves pour appuyer une décision.

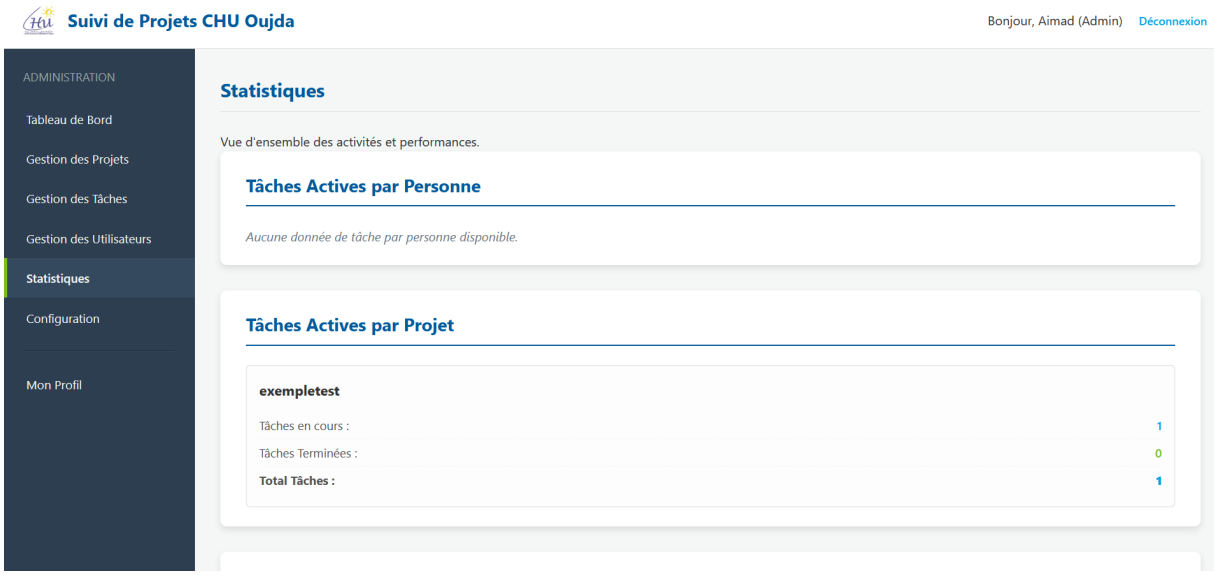
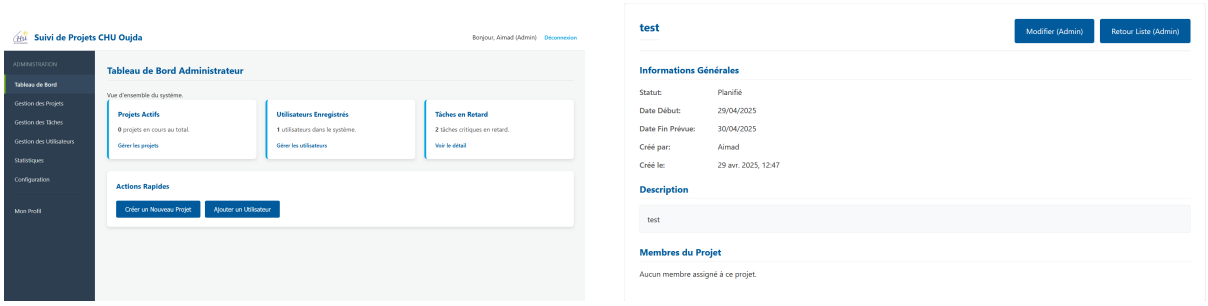


FIGURE 2 – Le tableau de bord statistique, qui passe de la simple description à l'exploration.



(a) Tableau de bord personnel (b) Vue détaillée d'un projet

FIGURE 3 – Interfaces conçues pour contextualiser l'information.

2.4 Développer : Une Solution d'Aide à la Décision

Niveau 1 - Composant d'une solution décisionnelle. L'application, et surtout son module statistique, n'est pas qu'un simple outil de gestion. C'est un premier pas vers une solution d'aide à la décision. Il transforme l'activité de tous les jours (une tâche marquée comme "terminée") en information stratégique. Il permet aux managers de piloter leur activité avec des données à jour, plutôt qu'au feeling.

3 Conclusion

Ce projet a été une expérience de développement très complète. Au-delà des aspects purement techniques, il m'a confronté aux défis réels de la conception d'un outil : comprendre le besoin de l'utilisateur, structurer la donnée pour qu'elle ait du sens, et la présenter de manière à ce qu'elle aide vraiment à prendre des décisions. La solution livrée est fonctionnelle et répond au besoin initial du CHU, ce qui démontre ma capacité à mener un projet de A à Z.

A Annexe : Exemple de Pipeline d'Agrégation

Pour illustrer un aspect technique plus complexe, voici un extrait du code qui génère les statistiques par utilisateur. Ce pipeline MongoDB est un bon exemple de la façon dont on peut transformer des données brutes (des centaines de tâches) en un résumé structuré et utile pour l'analyse.

```
// Contr leur des statistiques
async function getUserTaskStats() {
    const stats = await Task.aggregate([
        // Etape 1: Grouper les taches par utilisateur et par statut
        {
            $group: {
                _id: { userId: "$assignedTo", status: "$status" },
                count: { $sum: 1 }
            }
        },
        // Etape 2: Joindre avec la collection 'users' pour obtenir les noms
        {
            $lookup: {
                from: "users",
                localField: "_id.userId",
                foreignField: "_id",
                as: "userInfo"
            }
```

```
        },
        // Etape 3: Deconstruire le tableau resultant de la jointure
        { $unwind: "$userInfo" },
        // Etape 4: Regrouper a nouveau par utilisateur pour formater la sortie
        {
            $group: {
                _id: "$_id.userId",
                fullname: { $first: "$userInfo.fullname" },
                tasksByStatus: {
                    $push: { // Creer un tableau des statuts et des comptes
                        status: "$_id.status",
                        count: "$count"
                    }
                }
            }
        },
        // Etape 5: Projeter les champs finaux pour un affichage propre
        {
            $project: {
                _id: 0,
                userName: "$fullname",
                stats: "$tasksByStatus"
            }
        },
        // Etape 6: Trier par nom d'utilisateur
        { $sort: { userName: 1 } }
    ]);
    return stats;
}
```

Listing 1 – Pipeline d'agrégation MongoDB pour les statistiques utilisateurs