

Fonctionnalités développées :

- Interface utilisateur interactive sans données avec l'utilisation de Tailwind CSS pour le rendu visuel
- Affichage dynamique des terrains et séances disponibles avec donnée statiques
- Début de la base de donnée (tables player, coach, court) en PostgreSQL
- Création de l'API avec SpringBoot (tables player, coach, court)
- Découverte complète de Timefold
- Tutoriel Timefold effectué
- Mise en place sur Timefold du diagramme de classes

Bilan de la version actuelle :

- **Bugs :**
 - L'affichage de la page administrateur n'est pas responsive
 - Bug sur la table court lors de la modification d'un terrain
 - Problème suite à une review de l'API qui ne correspond pas à tous les critères pour être une API Restful
- **Validation :**
 - Page de connexion responsive
 - Affichage des terrains, et de l'onglet donnée et contrainte validé
 -

Conformité aux choix initiaux du projet :

- **Changement des choix techniques :**
 - L'utilisation de Vue.js et Tailwind CSS pour la conception du frontend suite à l'étude de nos différentes possibilités.
 - Changement du type de base de données, passant de MariaDB à PostgreSQL. On pense que PostgreSQL offre une flexibilité en termes de types de données, de capacité de mise à l'échelle, de simultanéité et d'intégrité des données structurées.

Donc si on fait un rapide bilan de ce que nous avons à la fin de l'itération 1, nous avons mis en place notre maquette avec les technologies web pour pouvoir montrer un début de rendu au client. Qui montre très rapidement comment fonctionne l'application web. Nous avons un début d'API restful. Elle fonctionne et est liée à certaines tables de la base de données. Pour terminer nous pensons avoir compris comment fonctionne TimeFold et ses différentes fonctionnalités