

# Expression du besoin – Projet TeamBoard

## 1. Contexte

Le projet **TeamBoard** est une mini-application web permettant à une petite équipe de gérer ses tâches collaboratives.

L'objectif est de fournir une interface *simple, ergonomique et efficace* pour la création, le suivi et la gestion des tâches au sein de l'entreprise.

L'application devra proposer une approche moderne inspirée des méthodologies agiles, avec un système de **colonnes Kanban** pour une visualisation claire de l'avancement des travaux.

---

## 2. Objectifs

- *Permettre* aux membres de l'équipe de se connecter avec un système d'**authentification** simplifié.
  - *Fournir* un espace collaboratif centralisé pour la **gestion des tâches**.
  - *Faciliter* l'attribution des tâches aux différents membres connectés.
  - *Offrir* une vue claire de l'avancement grâce au suivi des statuts avec **interface Kanban**.
  - *Proposer* des options de **filtrage avancé** pour mieux organiser le travail.
  - *Assurer* une expérience utilisateur fluide avec des interactions modernes.
- 

## 3. Fonctionnalités demandées

### 3.1 Authentification

- **Connexion** d'un utilisateur existant (nom d'utilisateur, mot de passe).
- Gestion de la *persistance de session* (reconnexion automatique si déjà connecté).
- Mise à disposition de **comptes de démonstration** pour les tests.
- Interface d'inscription préparée pour évolution future.

### 3.2 Gestion des tâches

- **Création** d'une tâche avec un *titre* et une *description*.
- **Suppression** d'une tâche.
- **Attribution** d'une tâche à un utilisateur connecté.
- Changement de statut d'une tâche (à faire, en cours, annulé, terminé).
- Édition complète des tâches existantes.
- Catégorisation par type de tâche (Bug, Feature, Improvement).

### 3.3 Organisation et filtrage

- **Interface Kanban** avec 4 colonnes de statut distinctes.
  - **Drag & Drop** pour déplacer les tâches entre les colonnes.
  - Filtrer les tâches par *utilisateur assigné*.
  - Filtrer les tâches par *statut*.
  - Possibilité de **combiner les filtres**.
  - Affichage clair et lisible des tâches avec **badges visuels**.
  - Création rapide de tâches depuis chaque colonne.
-

## 4. Contraintes et attentes

- **Logo visible** en haut à gauche quand connecté, au milieu sur la page de connexion.
  - Design moderne et épuré utilisant des composants UI standardisés.
  - **Interface responsive** adaptée aux différents types d'écrans.
  - Données stockées et persistantes côté client (`localStorage`).
  - Navigation fluide et intuitive avec animations appropriées.
  - Expérience utilisateur optimisée avec *notifications* et *feedback visuel*.
  - Accessibilité respectant les bonnes pratiques web.
- 

## 5. Exigences et critères d'acceptation

### 5.1 Sécurité et accès

- Un utilisateur non connecté ne peut pas accéder à la liste des tâches.
- La session utilisateur est maintenue entre les rechargements de page.

### 5.2 Gestion des tâches

- Une tâche nouvellement créée est *immédiatement visible* dans la liste.
- Le statut d'une tâche est modifiable par **drag & drop** et reflété instantanément.
- L'attribution d'une tâche à un utilisateur est persistante et visible.
- Les modifications de tâches sont sauvegardées automatiquement.

### 5.3 Interface et navigation

- L'**interface Kanban** permet une navigation fluide entre les 4 colonnes.
- Les types de tâches (Bug, Feature, Improvement) sont clairement identifiables.
- La création rapide depuis chaque colonne fonctionne avec le bon statut pré-sélectionné.
- Le filtrage affiche uniquement les tâches correspondant aux critères sélectionnés.

### 5.4 Compatibilité

L'application fonctionne de manière responsive sur *desktop*, *tablette* et *mobile*.