

# Cahier des charges – EpiTrello

## 1. Contexte

Le projet consiste à développer une application web de gestion de tâches inspirée de Trello. L'objectif est de permettre à un utilisateur ou une équipe de créer, organiser et suivre des tâches sous forme de tableaux, de listes et de cartes.

Le projet doit être fonctionnel, ergonomique et accessible depuis un navigateur web. La durée de développement prévue est de 4 mois.

---

## 2. Objectifs

- Créer une application web de type Kanban permettant la gestion de projets personnels ou collaboratifs.
  - Permettre la création de tableaux, de listes et de cartes.
  - Offrir une interface moderne, fluide et responsive.
  - Intégrer une authentification sécurisée des utilisateurs.
  - Création d'une CI/CD, utilisation de Github project, Docker.
  - Création d'une documentation projet / code.
- 

## 3. Public visé

Le projet s'adresse principalement à des étudiants, développeurs ou petites équipes souhaitant organiser leurs tâches de manière simple et efficace.

## 4. Fonctionnalités principales

### 1. Authentification et utilisateurs

- Inscription et connexion via email et mot de passe.
- Possibilité d'effacer son compte
- Gestion de la session utilisateur.

### 2. Tableaux (Boards)

- Création, modification et suppression de tableaux.
- Choix d'un nom et d'une couleur.
- Visualisation en liste des tableaux de l'utilisateur.

### 3. Listes (Columns)

- Ajout, renommage et suppression de listes dans un tableau.
- Réorganisation des listes par glisser-déposer.

### 4. Cartes (Cards)

- Création, modification et suppression de cartes.
- Déplacement des cartes entre les listes (drag-and-drop).
- Détails d'une carte : titre, description, pièce jointe, date limite, tags, etc.

### 5. Interface utilisateur

- Affichage sous forme de tableau Kanban.
- Interface responsive (tablette, ordinateur).
- Design minimaliste et clair avec Tailwind CSS.

## 5. Contraintes techniques

### Front-end :

- Framework : Vue 3 (Composition API)
- Langage : TypeScript
- Outil de build : Vite
- Style : Tailwind CSS v4
- Routing : Vue Router

### Back-end :

- Langage : TypeScript
- Base de données : Supabase (PostgreSQL)
- Authentification : Supabase Auth
- Stockage éventuel : Supabase Storage

### Schéma simplifié de la base de données :

- Users : id, email, nom, avatar
- Boards : id, user\_id, nom, couleur
- Lists : id, board\_id, nom, position
- Cards : id, list\_id, titre, description, date\_limite, position

## 6. Design et expérience utilisateur

- Interface simple et intuitive.
- Utilisation de composants réutilisables (boutons, cartes, modales).
- Animations légères et transitions fluides.

## 7. Sécurité

- Authentification sécurisée avec Supabase Auth.
  - Validation des données côté client et serveur.
  - Protection des routes et des ressources utilisateur.
  - Stockage minimal des informations personnelles
- 

## 8. Tests et qualité

- Linting avec ESLint et Prettier.
  - Intégrations optionnel de tests avec Playwright.
  - Intégration via GitHub Actions.
- 

## 9. Planning prévisionnel (4 mois)

- **Mois 1** : Installation des outils, mise en place du front, du back et de la base de données, création des maquettes et système d'authentification.
  - **Mois 2** : Ajout des cartes, titres, descriptions, labels, checklists, recherche et premières optimisations.
  - **Mois 3** : Ajout des fonctionnalités secondaires, amélioration de l'interface et optimisations techniques.
  - **Mois 4** : Sécurité, tests, corrections, déploiement et présentation finale.
- 

## 10. Livrables

- Code source complet sur GitHub.
- Documentation technique et guide d'installation.
- Présentation finale du projet.