

# Mini Projet Technique – Application de gestion de tâches collaboratives

---

## Contexte

Développez une mini-application de gestion de tâches partagées pour une petite équipe. Les utilisateurs doivent pouvoir créer, éditer, supprimer et consulter leurs tâches, filtrer et trier les résultats, et interagir via une interface animée et moderne.

## Objectifs

### Backend – Node.js + Express

#### Fonctionnalités

- REST API pour la gestion des tâches
- Routes à implémenter :
  - GET /tasks — Liste des tâches
  - GET /tasks/:id — Détails d'une tâche
  - POST /tasks — Création
  - PUT /tasks/:id — Mise à jour
  - DELETE /tasks/:id — Suppression

#### Stack technique

- Node.js + Express
- Middleware de validation (ex : express-validator)
- Gestion centralisée des erreurs
- Structure en modules : routes/, controllers/, services/, models/

#### Bonus

- Logger (type morgan)
- Seeds de données (avec Prisma)
- Tests d'API (bonus avec Supertest)

## Frontend – Next.js + TypeScript + TailwindCSS + Framer Motion

### Fonctionnalités

- Liste des tâches avec affichage sous forme de cartes ou tableau
- Filtres dynamiques : statut, date d'échéance
- Tri : priorité ou date
- Animations à l'apparition/suppression (Framer Motion)
- Formulaire d'ajout/édition avec validation (React Hook Form + Zod)
- Recherche instantanée
- Navigation fluide avec App Router
- UI responsive, dark mode toggle, animations avec Framer Motion
- Aucun composant externe : tout est fait avec TailwindCSS pur

### Stack

- Next.js 14+ (App Router)
- TypeScript
- TailwindCSS
- Framer Motion
- State management : useState, useEffect, useReducer
- Appels API via fetch ou axios

### Bonus

- Dark mode (via Tailwind dark:)
- Gestion du cache avec SWR ou react-query (facultatif)
- Authentification simulée (si voulu)

### Livrables attendus

Dépôt Git public avec :

- Code clair, modulaire et commenté
- Durée maximale de réalisation : 48h