

# CollabTask — One-Pager

## Pitch

**CollabTask** est une application Web collaborative de gestion de projets et de tâches. Elle vise à offrir une expérience claire et moderne : création de projets, attribution de tâches aux membres, suivi visuel de l'avancement via un tableau Kanban et statistiques synthétiques. Le but : démontrer la maîtrise d'une architecture full-stack moderne, du design d'API jusqu'à l'interface réactive déployée en ligne. Prototype complet *front + back* attendu d'ici **15 déc. 2025**.

## Piliers

- Clarté / Ergonomie / Cohérence visuelle
- Architecture full-stack propre (Spring Boot + React)
- Persistance fiable et déploiement durable

## Fonctionnalités clés

- Authentification et gestion d'utilisateurs
- Création et organisation de projets
- Gestion de tâches : titre, description, état, échéance
- Tableau Kanban interactif (drag & drop)
- Filtres, tri et recherche intégrée
- Historique et notifications internes

## Architecture technique

- **Backend** : Spring Boot 3.5 • Java 21 • API REST + validation Jakarta
- **Base** : PostgreSQL 18 (Supabase en production)
- **Frontend** : React + Vite 16 (TypeScript, hooks)
- **Déploiement** : Vercel (front) • Supabase (BDD) • GitHub Actions (CI/CD)

## Objectifs d'apprentissage

- Concevoir une application Web full-stack cohérente
- Développer une API REST sécurisée avec Spring Boot
- Déployer une base PostgreSQL dans un environnement distant
- Construire une interface moderne et performante avec React
- Assurer CI/CD, documentation et maintenabilité

## Livrables

- Application Web fonctionnelle (front + back déployés)
- Base de données persistante avec données de démonstration
- Documentation technique : schéma BDD • architecture • procédure de déploiement
- Présentation finale et post-mortem

## Public & usage visé

- Équipes étudiantes ou petites entreprises souhaitant suivre leurs projets
- Usage via navigateur (Web responsive desktop + mobile)

## Style visuel & UX

- Interface claire, palette neutre avec couleur accent turquoise
- Icônes vectorielles • police Segoe UI
- Animations douces et transitions cohérentes
- Design system léger pour composants UI réutilisables

## Tech & prod

- Spring Boot • React • TypeScript • Docker Compose (local)
- Gestion Git (GitHub • Git CLI)
- Suivi de versions • tests unitaires Jest / JUnit 5
- Monitoring et logs sur Supabase

## Risques & parades

- **Intégration front/back** : standardiser les endpoints REST et les DTO.
- **Performance front** : utiliser React memo + lazy loading Vite.
- **Scope** : se limiter à la gestion des tâches et à un tableau Kanban unique pour le prototype.
- **Déploiement** : automatiser avec GitHub Actions et variables d'environnement sécurisées.

## Valeur ajoutée (USP)

- Application full-stack aboutie • architecture documentée
- UX fluide et moderne, pensée pour la collaboration
- Outil léger et accessible via le Web, sans installation