

# Analyse des risques

Catégorie	Risque	Impact	Mesures Clés
Technique	Conflits de dépendances	Retards, bugs	Utiliser Docker pour uniformiser l'environnement.
	Base de données mal conçue	Performances médiocres	Modéliser le schéma avant le développement, utiliser un ORM.
	Bugs critiques non détectés	MVP instable	Mettre en place des tests automatisés (Jest), revues de code, checklist OWASP.
Organisationnel	Problèmes de performance	Mauvaise expérience utilisateur	Optimiser les médias (WebP, liens Youtube), lazy loading, cache (Redis).
	Manque de coordination	Perte de temps, doublons	Utiliser un tableau Kanban (Trello), branches Git par fonctionnalité.
	Dérive des objectifs	Retards sur les livrables	Figer le backlog par sprint, désigner un Product Owner.
Fonctionnel	Absence d'un développeur	Perte de connaissances	Documenter le code (README, commentaires), pair programming, rotation des tâches.
	Fonctionnalités mal comprises	Reprise coûteuse en temps	Affiner les user stories avant le sprint, créer des maquettes (Figma).
	Mauvaise expérience utilisateur (UX)	Faible adoption du produit	Réaliser des tests utilisateurs, utiliser un Design System, concevoir en mobile-first.
Externe	Non-respect du RGPD	Risque juridique	Générer une page "Mentions légales", ajouter un banner de consentement pour les cookies, implémenter un bouton "Supprimer mon compte".
	Problèmes avec les APIs externes	Fonctionnalités bloquées	Scaper l'API et générer notre propre API en local