

Exercice Technique – Application de Gestion de Notes Collaboratives

Objectif

Développer une application web de gestion de notes avec authentification, recherche, filtrage, et partage entre utilisateurs.

Fonctionnalités attendues

1. Authentification (JWT ou session)

- Inscription d'un utilisateur (email + mot de passe)
- Connexion / Déconnexion
- Middleware de sécurisation des routes API

2. Gestion des notes

Chaque note comporte :

- Titre
- Contenu (Markdown autorisé)
- Date de création / modification
- Tag(s) (optionnel)
- Statut de visibilité : privé / partagé / public

Fonctionnalités API :

- Créer une note
- Lister ses notes
- Supprimer une note
- Modifier une note
- Rechercher par titre ou tag
- Filtrer par statut

3. Partage de notes

- Partager une note avec un autre utilisateur (lecture seule)
- Permettre l'accès via lien public (token URL ou identifiant)

4. Interface Web

- Affichage des notes par statut (onglets ou filtres)
- Création/modification avec éditeur Markdown
- Barre de recherche + filtres
- Interface responsive (desktop/mobile)
- Notifications (succès/erreur)

Structure de rendu demandée

L'exercice a été estimé pour 72h par notre CTO

Un repo sous github composé

- Codes
- README.md complet avec instructions d'installation & de test
- Dockerisation (docker compose opérationnel)
- Bonnes pratiques : structure de projet claire, gestion d'erreurs, validation des inputs

Versions techniques demandées

Version 1 : Python + FastAPI

- Backend : FastAPI
 - JWT Auth
 - Pydantic pour validation
 - SQLite ou PostgreSQL
- Frontend : Next.js
 - API Routes côté client pour consommation
 - Auth context
 - Styled avec TailwindCSS (ou autre)
- Bonus : Swagger UI pour l'API

Version 2 : Node.js + React.js

- Backend : Node.js avec Express.js
- JWT Auth via `jsonwebtoken`
 - Validation via `Joi` ou `zod`
 - BDD SQLite ou MongoDB
- Frontend : React.js (CRA ou Vite)
 - Auth context / React Router
 - Styled Components ou autre
- Bonus : Documentation Postman