

Definition of Done – Backend Sana

1. Fonctionnalités Utilisateur

CreateUser (POST)

- Validation stricte des données d'entrée (username \geq 3 caractères, email valide).
- Hachage sécurisé du mot de passe avec **Blake2b + salt**.
- Connexion MongoDB réussie et fermée proprement (finally).
- Création de l'utilisateur avec gestion d'erreurs (try/except).
- Unicité garantie pour username et email.
- Logs détaillés de l'opération.
- Tests unitaires (>80% coverage).

Authenticate (POST)

- Hachage du mot de passe avec le même salt.
- Recherche sécurisée de l'utilisateur en base.
- Validation des credentials.
- Retour des données utilisateur sans mot de passe.
- Gestion des erreurs d'authentification.
- Logs de sécurité dédiés.
- Tests d'authentification réussie/échouée.

Update Profile (PUT)

- Vérification de l'existence de l'utilisateur.
- Validation des champs à mettre à jour.
- Vérification d'unicité sur username et email.
- Mise à jour sélective des champs uniquement modifiés.
- Logs complets des modifications.
- Tests de mise à jour partielle/complète.

2. Gestion de la Santé

Health Info (POST/PUT)

- Vérification de l'utilisateur existant.

- Création ou mise à jour de l'historique médical.
- Validation des données médicales (taille, poids, etc.).
- Mise à jour du last_updated.
- Sauvegarde en base avec gestion d'erreurs.
- Retour des données mises à jour.
- Tests avec données valides/invalides.

/health_info_v2/{username} (GET)

- Récupération de l'utilisateur par username.
- Récupération de l'historique médical associé.
- Gestion du cas "pas d'historique".
- Retour au format dict JSON.
- Logs de debug pour troubleshooting.
- Tests avec et sans données.

3. Diagnostic & Admin

User (GET)

- Récupération de tous les utilisateurs.
- Masquage des données sensibles (password).
- Logs de diagnostic complets.
- Comptage des utilisateurs.
- Tests avec base vide/peuplée.
- Accès restreint en production (admin-only).

API Health (GET)

- Endpoint de santé disponible.
- Retour simple confirmant le bon fonctionnement de l'API.
- Logs d'accès.
- Tests de disponibilité (ping).

4. Whisper STT

STT (POST)

- Upload et validation des fichiers audio.

- Transcription via **Whisper STT**.
- Analyse IA des symptômes (parse_medical_text).
- Nettoyage des fichiers temporaires.
- Gestion des erreurs de transcription.
- Tests avec fichiers audio réels.
- Validation de la précision médicale minimale.

5. DoD Technique Backend

Infrastructure & Environnements

- Variables d'environnement configurées (OPENAI_API_KEY, MONGO_URI, SALT, etc.).
- Configuration **CORS** pour le frontend.
- Logging structuré et fonctionnel.
- Gestion claire des environnements (dev, staging, prod).

Base de Données MongoDB

- Connexion MongoDB fonctionnelle avec gestion des erreurs.
- Modèles User et MedicalHistory validés.
- Indexation optimisée pour les requêtes fréquentes.
- Sauvegarde et récupération sécurisées.

Sécurité & Conformité

- Hachage sécurisé des mots de passe (Blake2b).
- Validation et expiration des tokens d'authentification.
- Protection contre les injections et attaques courantes.
- **Rate limiting** configuré sur les endpoints sensibles.
- Logs de sécurité centralisés.

Déploiement & Monitoring

- Docker build et exécution réussis.
- Variables d'environnement disponibles dans l'image.
- Tests sur **staging validés avant prod**.
- Rollback possible en cas d'échec.
- Documentation de déploiement claire.

Monitoring & Observabilité

- Logs structurés et consultables.
- Métriques de performance (latence, erreurs, QPS).
- Alertes configurées (erreurs critiques, downtime).
- Health checks automatisés.
- Traçabilité des requêtes (correlation ID).