

SYSTÈME DE GESTION DES ÉTUDIANTS AVEC PYTHON - MONGODB & REDIS

A rendre au plus tard le 05/04/2025

Contexte

Vous êtes chargé de développer une application permettant de gérer les informations des étudiants d'un établissement. L'application devra permettre l'ajout, la modification, la recherche et la suppression d'étudiants, tout en utilisant une base de données MongoDB pour le stockage et Redis pour l'optimisation des accès.

L'objectif est de fournir un outil efficace et rapide, avec des fonctionnalités avancées telles que le tri, les statistiques et la gestion des utilisateurs.

Objectifs

- Développer une application interne permettant aux administrateurs et enseignants de gérer les étudiants.
- Utiliser MongoDB pour stocker les données de manière structurée.
- Intégrer Redis pour optimiser l'accès aux informations les plus demandées.
- Assurer la validation des données (numéro unique, notes valides, etc.).
- Offrir un menu interactif pour naviguer entre les différentes fonctionnalités.
- Permettre l'exportation et l'importation des données.

Fonctionnalités demandées

1. Gestion des étudiants
 - Ajout d'un étudiant :
 - ✓ Saisie des informations (Nom, Prénom, Téléphone, Classe, Notes).
 - ✓ Vérification des contraintes (Téléphone unique, notes entre 0 et 20).
 - ✓ Enregistrement dans MongoDB et mise en cache dans Redis.
 - Affichage des étudiants :
 - ✓ Récupération des données depuis Redis (si disponibles) ou MongoDB.
 - ✓ Affichage sous forme de tableau.
 - Tri des étudiants par moyenne (ordre décroissant).
 - Recherche d'un étudiant selon nom, prénom, téléphone ou classe.
 - Modification des notes d'un étudiant (en mettant à jour MongoDB et Redis).
 - Suppression d'un étudiant avec mise à jour automatique des bases.
 - Exportation des données (CSV, JSON, Excel, PDF).
 - Importation d'une liste d'étudiants à partir d'un fichier CSV ou Excel.
2. Gestion des utilisateurs
 - Ajout d'un système d'authentification avec rôles :
 - ✓ Admin (gestion complète des étudiants et utilisateurs).
 - ✓ Enseignant (ajout/modification des notes uniquement).
 - ✓ Étudiant (consultation de ses propres notes).
 - Stockage sécurisé des mots de passe (ex: bcrypt).
 - Gestion des sessions utilisateurs via Redis (connexion rapide).
3. Statistiques et rapports
 - Calcul de la moyenne générale d'une classe.
 - Classement des étudiants (top 10).
 - Génération de rapports PDF avec résumé des résultats.
4. Notifications et alertes

- Envoi de notifications par e-mail/SMS pour :
 - ✓ Nouvelle note ajoutée.
 - ✓ Alerte si la moyenne est en dessous de 10/20.

Contraintes techniques

- MongoDB pour le stockage permanent des étudiants et des utilisateurs.
- Redis pour optimiser l'accès aux données (mise en cache des étudiants et sessions).
- Python POO pour la gestion de l'application.
- Validation stricte des données (téléphone unique, notes entre 0 et 20).
- Toutes les interactions doivent se faire via un menu interactif en ligne de commande.

Déroulement du projet

Étape 1 : Analyse et conception (**Semaine 1**)

- Définition du modèle de données (MongoDB, Redis).
- Choix des outils et technologies.
- Structuration du projet en POO (classes Etudiant, GestionEtudiants, etc.).

Étape 2 : Implémentation des fonctionnalités de base (**Semaine 1**)

- Création des classes et fonctions pour gérer les étudiants.
- Ajout et modification des étudiants avec validation.
- Connexion entre MongoDB et Redis.

Étape 3 : Ajout des fonctionnalités avancées (**Semaine 2**)

- Recherche et tri des étudiants.
- Système d'authentification et rôles.
- Mise en cache Redis pour améliorer les performances.
- Exportation & Importation des données.

Étape 4 : Ajout de statistiques et rapports (**Semaine 2**)

- Calcul des moyennes générales.
- Création de rapports PDF.

Étape 5 : Tests et validation (**Semaine 3**)

- Vérification des fonctionnalités.
- Tests unitaires et validation des performances.
- Déploiement final et documentation.