**📌 MÉMOIRE DU PROJET ET DE L'EXERCICE EN COURS**

**1️⃣ Projet Global : Site Pilote et Architecture Générale**

**🎯 Objectifs généraux du projet**

* **Site Pilote N°1** : Premier prototype fonctionnel intégrant toutes les bases du projet, servant de socle pour le **Site Pilote N°2**.
* **Base de données performante et optimisée**, indexant tous les champs fréquemment utilisés.
* **Gestion avancée des licences**, incluant le statut **free\_diffusion, free\_acquirement**, et d’autres attributs liés aux droits des œuvres.
* **Sécurité et traçabilité rigoureuses** à chaque niveau (gestion des actions, validation en deux étapes, logs).
* **Architecture cloud et déploiement multiservers** (serveurs en France et en Allemagne, complémentés par un Service Cloud).

**🛠️ Architecture de la base de données (MariaDB)**

* **Trois niveaux distincts** :
  1. **BDD Temporaire (SAS de validation)** → Contient des **données brutes** en attente de validation.
  2. **BDD Principale (Production)** → Seules les **données validées** sont stockées ici.
  3. **BDD Analytique (Hypervision)** → Regroupe logs et métriques pour **analyse et cybersécurité**.
* **Entités essentielles et leur rôle** :
  1. contrôle : Collecte, anticipe et gère les actions critiques.
  2. gestion & execution : Évolutifs, ils permettent les mises à jour sans refondre l’ensemble du projet.
  3. dictionnaire\_entité & modèle\_fonctionnel : Pour suivre et structurer les entités.
  4. actants, acteurs, cibles : Relations et permissions.

**2️⃣ Exercice en cours : Gestion des Propriétaires (owners)**

**🔍 Objectif de l’exercice**

* **Développer une interface PHP dynamique** avec trois actions :
  1. Ajouter un propriétaire.
  2. Modifier un propriétaire (sans créer un nouvel enregistrement).
  3. Supprimer un propriétaire (sans destruction définitive immédiate).
* **Implémentation d’un historique** pour permettre **la restauration des propriétaires supprimés**.
* **Sécurisation du processus** (validation en deux étapes, logs d’actions, et vérifications anti-duplication).

**✅ État actuel : Fonctionnalités qui marchent**

* **Ajout d’un propriétaire fonctionne bien** : Les enregistrements sont bien créés en BDD.
* **Photos en Base64 fonctionnent bien** : Si une photo est absente, la valeur "default" est attribuée.
* **Les suppressions ne sont pas destructrices** : deleted = 1 permet de marquer les suppressions.
* **Bouton "Modifier" et "Supprimer" sont affichés uniquement si deleted = 0**.

**⚠️ Problème restant à corriger**

1. **Le bouton "Modifier" crée un nouvel enregistrement au lieu de modifier**.  
   ➜ **Cause probable** : Mauvais paramètre href (db\_owners\_edit.php est appelé en mode ajout au lieu de modification).  
   ➜ **Solution** : Vérifier l’attribut id\_owners dans le lien et ajuster form\_mode.
2. **Le DOM de restauration historique affiche toutes les lignes au lieu de filtrer par id\_owners**.  
   ➜ **Cause probable** : L’écouteur d’événement n’est pas encore implémenté en **JavaScript**.  
   ➜ **Solution** : Ajouter un eventListener pour filtrer dynamiquement les résultats côté front.
3. **Manque de validation en deux étapes pour la restauration**.  
   ➜ **Solution** : Ajouter une **popup de confirmation** pour éviter toute restauration accidentelle.
4. **Suppression définitive non encore implémentée**.  
   ➜ **Solution** : Ajouter un bouton qui purge l’enregistrement directement en BDD.

**3️⃣ Prochaines étapes**

📌 **Correction du bouton "Modifier"**  
📌 **Ajout du filtre id\_owners sur le DOM de restauration**  
📌 **Ajout de la validation en deux étapes sur la restauration**  
📌 **Mise en place du bouton "Suppression Définitive"**