**📋 PLAN D'ACTION - SAUVEGARDE PAR ÉTAPES PNGDI**

**🔍 ANALYSE PRÉLIMINAIRE**

D'après les documents fournis, le projet PNGDI est à 99% de finalisation avec :

* Formulaire 9 étapes fonctionnel
* Problème : "Requête tronquée" lors de la soumission finale
* Solution proposée : **Sauvegarde automatique à chaque transition d'étape**

**🎯 OBJECTIF**

Implémenter un système de sauvegarde progressive qui enregistre les données à chaque étape pour éviter la perte de données et réduire la charge de la soumission finale.

**📊 PLAN STRUCTURÉ EN 6 PHASES**

**PHASE 1 : AUDIT ET VÉRIFICATION ⏱️ (1 tâche)**

**Objectif :** Vérifier l'existant avant toute modification

* **Tâche 1.1 :** Analyser les fichiers existants (OrganizationDraft.php, routes, contrôleurs)

**PHASE 2 : ARCHITECTURE BACKEND ⏱️ (2 tâches)**

**Objectif :** Préparer la structure de sauvegarde

* **Tâche 2.1 :** Optimiser le modèle OrganizationDraft.php
* **Tâche 2.2 :** Créer les routes et méthodes de sauvegarde par étape

**PHASE 3 : CONTRÔLEUR PRINCIPAL ⏱️ (2 tâches)**

**Objectif :** Implémenter la logique métier

* **Tâche 3.1 :** Méthodes de sauvegarde automatique par étape
* **Tâche 3.2 :** Méthodes de restauration et validation

**PHASE 4 : FRONTEND JAVASCRIPT ⏱️ (2 tâches)**

**Objectif :** Interface utilisateur et automatisation

* **Tâche 4.1 :** Modification organisation-create.js pour auto-save
* **Tâche 4.2 :** Gestion des transitions et feedback utilisateur

**PHASE 5 : SOUMISSION OPTIMISÉE ⏱️ (1 tâche)**

**Objectif :** Finalisation légère

* **Tâche 5.1 :** Nouvelle méthode de soumission finale allégée

**PHASE 6 : TESTS ET VALIDATION ⏱️ (1 tâche)**

**Objectif :** Vérification complète

* **Tâche 6.1 :** Tests workflow complet A→Z

**🔧 SPÉCIFICATIONS TECHNIQUES**

**Contraintes respectées :**

* ✅ Laravel PHP 7.3.29
* ✅ Structure DB existante (COLUMNS\_v2.sql)
* ✅ Formulaire 9 étapes conservé
* ✅ Aucune perte de fonctionnalités

**Innovations apportées :**

* 🚀 **Auto-save transparent** toutes les 30 secondes
* 🚀 **Sauvegarde à chaque transition** d'étape
* 🚀 **Restauration intelligente** en cas de déconnexion
* 🚀 **Soumission finale légère** (références seulement)

**⚡ ESTIMATION**

* **Durée totale :** 9 tâches réparties sur 6 phases
* **Complexité :** Moyenne (utilise l'existant OrganizationDraft)
* **Impact :** Zéro sur l'UX, amélioration performance majeure

**🎯 RÉSULTAT ATTENDU**

* ✅ Formulaire fonctionnel même avec 1000+ adhérents
* ✅ Sauvegarde automatique transparente
* ✅ Aucune perte de données
* ✅ Performance optimale

**🚦 VALIDATION REQUISE**

**Acceptez-vous ce plan d'action ?**

* Le découpage en 9 tâches vous convient-il ?
* Souhaitez-vous des modifications avant de commencer ?
* Dois-je procéder à la **PHASE 1 - Tâche 1.1** (Audit des fichiers existants) ?

**Une fois validé, je commencerai par l'audit pour m'assurer de ne rien casser dans l'existant.**

**📋 PLAN D'ACTION AJUSTÉ - SAUVEGARDE PAR ÉTAPES PNGDI**

**🔍 ANALYSE DES FICHIERS DE ROUTES**

D'après l'analyse des fichiers web.php et operator.php, j'ai identifié :

✅ **Existant :**

* Route principale : POST /operator/organisations (store)
* Route confirmation : GET /operator/organisations/confirmation/{dossier}
* APIs de validation déjà présentes dans web.php
* Modèle OrganizationDraft référencé

❌ **Manquant pour la sauvegarde par étapes :**

* Routes de sauvegarde automatique par étape
* Routes de restauration intelligente
* APIs pour gestion adhérents avancée (pagination, édition, suppression en bloc)

**🎯 PLAN STRUCTURÉ AJUSTÉ EN 6 PHASES**

**PHASE 1 : AUDIT ET VÉRIFICATION ⏱️ (1 tâche)**

**Objectif :** Vérifier l'existant avant toute modification

* **Tâche 1.1 :** Analyser OrganizationDraft.php et structure DB existante

**PHASE 2 : ROUTES ET API BACKEND ⏱️ (2 tâches)**

**Objectif :** Ajouter les routes manquantes sans conflit

* **Tâche 2.1 :** Nouvelles routes sauvegarde par étapes dans web.php
* **Tâche 2.2 :** APIs adhérents avancées (pagination, édition, suppression bloc)

**PHASE 3 : CONTRÔLEUR OPTIMISÉ ⏱️ (2 tâches)**

**Objectif :** Logique métier sauvegarde et restauration

* **Tâche 3.1 :** Méthodes sauvegarde automatique par étape
* **Tâche 3.2 :** **Restauration intelligente** + retour étapes précédentes

**PHASE 4 : FRONTEND JAVASCRIPT ENRICHI ⏱️ (2 tâches)**

**Objectif :** Interface utilisateur optimisée

* **Tâche 4.1 :** Auto-save + restauration intelligente dans organisation-create.js
* **Tâche 4.2 :** **Section 8 Adhérents** : pagination, édition, suppression bloc

**PHASE 5 : SOUMISSION LÉGÈRE ⏱️ (1 tâche)**

**Objectif :** Finalisation optimisée

* **Tâche 5.1 :** Nouvelle méthode store allégée (références aux données sauvegardées)

**PHASE 6 : TESTS COMPLETS ⏱️ (1 tâche)**

**Objectif :** Validation workflow A→Z

* **Tâche 6.1 :** Tests avec restauration intelligente + gestion adhérents avancée

**🚀 SPÉCIFICATIONS TECHNIQUES AJUSTÉES**

**Restauration Intelligente :**

* 🔄 **Auto-détection** de brouillon à la connexion
* 🔄 **Restauration contextuelle** selon l'étape de déconnexion
* 🔄 **Navigation arrière** avec conservation des données
* 🔄 **Indicator visuel** du statut de sauvegarde

**Section 8 Adhérents Enrichie :**

* 📄 **Pagination** avec 20 adhérents par page
* ✏️ **Édition inline** ou modal pour chaque adhérent
* 👁️ **Vue détaillée** avec historique et anomalies
* ☑️ **Sélection multiple** avec actions en bloc
* 🗑️ **Suppression en bloc** avec confirmation
* 🔍 **Recherche/filtre** en temps réel
* 📊 **Statistiques** qualité données

**📊 INNOVATIONS TECHNIQUES**

**Auto-Save Intelligent :**

// Sauvegarde à chaque changement significatif

- Transition d'étape : Immédiate

- Modification champ : 3 secondes après arrêt saisie

- Ajout/suppression adhérent : Immédiate

- Toutes les 30 secondes : Sauvegarde complète

**Gestion Adhérents Avancée :**

// Interface moderne avec actions multiples

- Table paginée responsive

- Actions en lot avec progress bar

- Édition modal avec validation temps réel

- Historique des modifications

- Export sélection personnalisée

**🚦 VALIDATION REQUISE**

**Acceptez-vous ce plan ajusté avec :**

1. ✅ **Restauration intelligente** complète
2. ✅ **Section 8 Adhérents** enrichie (pagination, édition, suppression bloc)
3. ✅ **Conservation** de toutes les routes existantes
4. ✅ **Ajout** uniquement des routes manquantes

**Puis-je commencer par la PHASE 1 - Tâche 1.1** (Audit OrganizationDraft.php et structure DB) ?

Cette analyse me permettra de voir exactement ce qui existe déjà pour éviter les doublons et optimiser l'intégration.