📋 Résumé des Améliorations - Système de Feature Requests

Date: 8 Janvier 2025
Branche: `improvements/feature-request-enhancements`

✅ Modifications Implémentées

1. 🔄 Scripts d'Export et d'Import (Dossier `scripts/`)

Quatre nouveaux scripts ont été créés pour faciliter la sauvegarde et la restauration des données :

Scripts pour les Feature Requests

- **`export-requests.ts`** : Exporte toutes les feature requests avec leurs attachments en JSON
  ```bash
  npx ts-node scripts/export-requests.ts [nom-fichier-optionnel]
  ```
- **`import-requests.ts`** : Réimporte les feature requests depuis un fichier JSON
  ```bash
  npx ts-node scripts/import-requests.ts <fichier-json>
  ```

Scripts pour le Suivi des Tests

- **`export-test-pages.ts`** : Exporte l'état de validation des pages de test
  ```bash
  npx ts-node scripts/export-test-pages.ts [nom-fichier-optionnel]
  ```
- **`import-test-pages.ts`** : Réimporte l'état des pages de test
  ```bash
  npx ts-node scripts/import-test-pages.ts <fichier-json>
  ```

Caractéristiques:
- Gestion d'erreurs complète
- Validation des données avant import
- Rapports détaillés (succès/échecs)
- Statistiques par statut/rôle

2. 📊 Nouvelle Page de Suivi des Tests

Localisation: `/feature-requests/test-tracking`

Fonctionnalités:
- Interface organisée par rôle (Super Admin, Admin, Contractor, Agency, Payroll)
- Liste automatique de toutes les pages de la plateforme par rôle
- Checkbox pour marquer chaque page comme validée
- Sauvegarde de l'état dans la base de données
- Statistiques en temps réel (total, validées, pourcentage de complétion)
- Recherche par nom ou URL de page
- Sauvegarde groupée avec détection des changements

Pages identifiées par rôle:
- **SUPER_ADMIN:** 9 pages (Dashboard, Analytics, Users, Tenants, etc.)
- **ADMIN:** 25 pages (Dashboard, Contracts, Invoices, Settings, etc.)
- **CONTRACTOR:** 13 pages (Dashboard, Timesheets, Expenses, Invoices, etc.)
- **AGENCY:** 10 pages (Dashboard, Contractors, Contracts, Payments, etc.)
- **PAYROLL:** 10 pages (Dashboard, Payslips, Invoices, Remittances, etc.)

3. 🔧 Amélioration du Système de Statut

Ancien comportement:
- Statut par défaut: `SUBMITTED`

Nouveau comportement:
- Statut par défaut: `PENDING`
- Permet de changer librement entre les statuts
- Workflow plus flexible (peut repasser en PENDING après modification)

Statuts disponibles:
- `PENDING` (par défaut)
- `SUBMITTED`
- `WAITING_FOR_CONFIRMATION`
- `CONFIRMED`
- `REJECTED`

4. 🌱 Amélioration du Fichier de Seed (PRIORITÉ MAXIMALE)

Nouvelles données créées automatiquement au premier lancement:

Tenant Company

- Nom: “Tenant Company”
- Type: `ownerType: "tenant"` (marquée comme TENANT)
- Pays: United States
- Ville: New York
- Statut: Active

Compte Bancaire par Défaut

- Nom: “Default Tenant Bank Account”
- Banque: “Default Bank”
- Titulaire: “Tenant Company”
- Numéro de compte: 1234567890
- Devise: USD
- Usage: GROSS
- Lié à la tenant company

16 Contrats (4 de chaque type)

Chaque type de contrat est créé 4 fois :
- **GROSS** (4 contrats)
- **PAYROLL** (4 contrats)
- **PAYROLL_WE_PAY** (4 contrats)
- **SPLIT** (4 contrats)

Propriétés des contrats:
- Type: SOW (Statement of Work)
- Statut: Active
- Taux: 5000 USD/mois
- Marge: 10%
- Durée: 1 an
- Liés au compte bancaire de la tenant company
- Créés par l'utilisateur Admin

📦 Modifications de Base de Données

Nouveau Modèle: `PageTestStatus`

```
model PageTestStatus {
  id          String    @id @default(cuid())
  tenantId    String
  pageUrl     String
  pageName    String
  pageRole    String
  isValidated Boolean   @default(false)
  testedBy    String?
  testedAt    DateTime?
  notes       String?
  createdAt   DateTime  @default(now())
  updatedAt   DateTime
}
```

Migration: `20250108000000_add_page_test_status_and_pending_default`

- Création de la table `page_test_status`
- Modification du statut par défaut de `feature_requests`
- Ajout des index nécessaires
- Contrainte unique sur (tenantId, pageUrl, pageRole)

🚀 Prochaines Étapes

1. Déploiement

```
# Se placer dans le répertoire du projet
cd /home/ubuntu/github_repos/payroll-saas

# Installer les dépendances si nécessaire
npm install

# Générer le client Prisma avec le nouveau schéma
npx prisma generate

# Appliquer la migration
npx prisma migrate deploy

# Lancer le seed pour créer les données
npx ts-node scripts/seed.ts
```

2. Pousser les Modifications sur GitHub

```
# Pousser la branche sur GitHub
git push origin improvements/feature-request-enhancements

# Créer une Pull Request sur GitHub
# Comparer: feature/feature-request-system ← improvements/feature-request-enhancements
```

3. Vérification Post-Déploiement

1. **Vérifier la page de suivi des tests:**
   - Se connecter en tant qu'Admin
   - Naviguer vers `/feature-requests/test-tracking`
   - Tester la validation des pages

2. **Vérifier les nouvelles feature requests:**
   - Créer une nouvelle feature request
   - Vérifier que le statut par défaut est bien `PENDING`

3. **Vérifier les scripts d'export/import:**
   - Exporter les feature requests
   - Exporter les statuts de test
   - Vérifier les fichiers JSON générés

4. **Vérifier les données du seed:**
   - Vérifier l'existence de la tenant company
   - Vérifier le compte bancaire par défaut
   - Vérifier les 16 contrats (4 de chaque type)

📝 Documentation

Un fichier `scripts/README.md` complet a été créé avec :
- Description de tous les scripts
- Instructions d'utilisation
- Exemples de commandes
- Gestion des erreurs
- Bonnes pratiques

⚠️ Notes Importantes

1. **Migration de Base de Données:** La migration doit être appliquée avant d'utiliser la nouvelle page de suivi des tests.

2. **Compatibilité:** Les changements sont rétrocompatibles avec les feature requests existantes.

3. **Permissions:** L'accès à la page de suivi des tests nécessite les permissions appropriées (Admin/Super Admin).

4. **Seed Idempotent:** Le script de seed peut être relancé sans créer de doublons (il vérifie l'existence avant de créer).

🎯 Objectifs Atteints

- ✅ Scripts d’export/import JSON fonctionnels
- ✅ Page de suivi des tests complète et interactive
- ✅ Système de statut amélioré avec PENDING par défaut
- ✅ Seed amélioré avec tenant company, compte bancaire et contrats
- ✅ Migration Prisma créée et testée
- ✅ Documentation complète
- ✅ Code commité sur la branche dédiée

📞 Support

Pour toute question ou problème :
1. Vérifier les logs de migration Prisma
2. Vérifier les logs des scripts (messages d’erreur détaillés)
3. Consulter la documentation dans `scripts/README.md`

Branche Git: `improvements/feature-request-enhancements`
Commit: `fc2f88a` - feat: Amélioration du système de feature requests et ajout du suivi des tests