analyse en détaille les informations du projet et poursuivons avec la discussion 13. Finalement il y a une rediction qui est faite apres l'enregistrement du formulaire create.blade.php vers confirmation.blade.php. Je veux que tu transforme donc confirmation.blade.php pour quelle soit la vue de telechargement des adherents. ce vue doit respecter la charte graphique de index.php, avoir les foctionnalités d'importation des adhérents comme le formulaire de l'etape 7, Détection automatique des gros volume adhérents, Interface avancée chunking, tableau des adhérents importés dynamique avec toutes les options de manipulation. le système enregistré dans la base de donnée, chacune agents importés, soit de manière unitaire soit pas lot de pas plus de 100 (chunking) avec progresse bar évolue, possibilité de faire pause et de reprendre. ne pas oublié que tous les adherents doivent etre enregistre meme si leur NIP sont erroné ou en doublons toutefois ces anomalies doivent etre enregistré pour traitement. il y a dans la base de connaissance des fonctions et fichiers qui traitait deja de ce cas en parti

n'oublie pas d'externalisé le CSS et le javascript des vues de sorte à les alléger. au besoin crees des sous fichier qui font s'imbriqué que ce soit pour les vues, le javascript ou les css

Je vais maintenant créer une solution complète pour transformer confirmation.blade.php en vue de téléchargement des adhérents avec architecture modulaire externalisée. Analysons d'abord la structure du projet:

Parfait ! Maintenant je vais créer une solution complète avec architecture modulaire externalisée. Créons d'abord la nouvelle vue confirmation.blade.php puis tous les fichiers CSS et JS modulaires.

{{-- =============================================================================

CONFIRMATION.BLADE.PHP - VUE MODERNISÉE IMPORT ADHÉRENTS

Transformation complète en interface d'import avec architecture modulaire

Version: 2.0 - SGLP avec détection automatique gros volumes

============================================================================= --}}

@extends('layouts.operator')

@section('title', 'Import des Adhérents - Finalisation Dossier')

@section('page-title', 'Import Adhérents & Finalisation')

{{-- Styles CSS externalisés --}}

@push('styles')

<link rel="stylesheet" href="{{ asset('css/gabon-charte.css') }}">

<link rel="stylesheet" href="{{ asset('css/confirmation-import.css') }}">

<link rel="stylesheet" href="{{ asset('css/adherents-manager.css') }}">

<link rel="stylesheet" href="{{ asset('css/chunking-interface.css') }}">

@endpush

@section('content')

<div class="container-fluid">

{{-- Variables PHP pour les données --}}

@php

// Configuration de base depuis le contrôleur

$dossier = $dossier ?? (object)[

'id' => 1,

'numero\_dossier' => 'DOS-2025-001',

'numero\_recepisse' => 'REC-2025-001'

];

$organisation = $organisation ?? (object)[

'id' => 1,

'nom' => 'Organisation Test',

'type' => 'association',

'sigle' => 'OT'

];

// Statistiques adhérents

$adherents\_stats = $adherents\_stats ?? [

'existants' => 5,

'minimum\_requis' => 15,

'manquants' => 10,

'peut\_soumettre' => false

];

// Configuration upload

$upload\_config = $upload\_config ?? [

'max\_file\_size' => '10MB',

'chunk\_size' => 100,

'max\_adherents' => 50000,

'chunking\_threshold' => 200

];

// URLs dynamiques

$urls = $urls ?? [

'store\_adherents' => route('operator.dossiers.store-adherents', $dossier->id),

'template\_download' => route('operator.templates.adherents-excel'),

'process\_chunk' => route('api.organisations.process-chunk'),

'confirmation' => route('operator.dossiers.confirmation', $dossier->id),

'health\_check' => '/api/chunking/health'

];

@endphp

{{-- ============================================================================

SECTION 1: HEADER PRINCIPAL AVEC PROGRESSION

============================================================================ --}}

<div class="confirmation-header">

<div class="header-content">

<!-- Indicateur de phase -->

<div class="phase-indicator">

<i class="fas fa-users me-2"></i>

Finalisation: Import des Adhérents

</div>

<!-- Breadcrumb gabonais -->

<nav aria-label="breadcrumb" class="breadcrumb-gabon">

<ol class="breadcrumb">

<li class="breadcrumb-item">

<a href="{{ route('operator.dashboard') }}">

<i class="fas fa-home me-1"></i>Dashboard

</a>

</li>

<li class="breadcrumb-item">

<a href="{{ route('operator.organisations.index') }}">Organisations</a>

</li>

<li class="breadcrumb-item">

<a href="{{ route('operator.organisations.show', $organisation->id ?? 1) }}">{{ $organisation->nom }}</a>

</li>

<li class="breadcrumb-item active">Import Adhérents</li>

</ol>

</nav>

<!-- Titre principal avec organisation -->

<div class="row align-items-center">

<div class="col-md-8">

<h1 class="header-title">

<i class="fas fa-flag me-3"></i>

Import des Adhérents

</h1>

<div class="header-subtitle">

<strong>{{ $organisation->nom }}</strong>

@if($organisation->sigle)

({{ $organisation->sigle }})

@endif

| {{ ucfirst($organisation->type) }}

</div>

<div class="header-meta">

<span class="me-3">

<i class="fas fa-file-alt me-1"></i>

{{ $dossier->numero\_dossier }}

</span>

<span class="me-3">

<i class="fas fa-receipt me-1"></i>

{{ $dossier->numero\_recepisse }}

</span>

<span class="badge bg-success">

<i class="fas fa-check me-1"></i>

Organisation Créée

</span>

</div>

</div>

<div class="col-md-4 text-end">

<div class="header-actions">

<button class="btn btn-outline-light me-2" onclick="showHelp()">

<i class="fas fa-question-circle me-2"></i>Aide

</button>

<button class="btn btn-warning" onclick="saveDraft()">

<i class="fas fa-save me-2"></i>Sauvegarder

</button>

</div>

</div>

</div>

</div>

</div>

{{-- ============================================================================

SECTION 2: DASHBOARD STATISTIQUES TEMPS RÉEL

============================================================================ --}}

<div class="stats-dashboard">

<div class="dashboard-header">

<h3 class="dashboard-title">

<i class="fas fa-chart-pie me-2"></i>

Statistiques des Adhérents

</h3>

<p class="dashboard-subtitle">Suivi en temps réel avec détection automatique des gros volumes</p>

</div>

<!-- Grid statistiques -->

<div class="stats-grid">

<div class="stat-card highlight" id="stat-existants">

<div class="stat-icon">

<i class="fas fa-users"></i>

</div>

<div class="stat-content">

<div class="stat-number">{{ $adherents\_stats['existants'] }}</div>

<div class="stat-label">Existants</div>

</div>

</div>

<div class="stat-card warning" id="stat-minimum">

<div class="stat-icon">

<i class="fas fa-exclamation-triangle"></i>

</div>

<div class="stat-content">

<div class="stat-number">{{ $adherents\_stats['minimum\_requis'] }}</div>

<div class="stat-label">Minimum Requis</div>

</div>

</div>

<div class="stat-card {{ $adherents\_stats['manquants'] > 0 ? 'danger' : 'success' }}" id="stat-manquants">

<div class="stat-icon">

<i class="fas fa-{{ $adherents\_stats['manquants'] > 0 ? 'user-plus' : 'check-circle' }}"></i>

</div>

<div class="stat-content">

<div class="stat-number">{{ $adherents\_stats['manquants'] > 0 ? $adherents\_stats['manquants'] : '✓' }}</div>

<div class="stat-label">{{ $adherents\_stats['manquants'] > 0 ? 'Manquants' : 'Complet' }}</div>

</div>

</div>

<div class="stat-card info" id="stat-importes">

<div class="stat-icon">

<i class="fas fa-upload"></i>

</div>

<div class="stat-content">

<div class="stat-number" id="import-count">0</div>

<div class="stat-label">Importés</div>

</div>

</div>

</div>

<!-- Indicateur de capacité -->

<div class="capacity-indicator">

<div class="capacity-header">

<span class="capacity-label">Capacité d'import détectée</span>

<span class="capacity-mode" id="capacity-mode">Standard</span>

</div>

<div class="capacity-bar">

<div class="capacity-fill" id="capacity-fill" style="width: {{ min(100, ($adherents\_stats['existants'] / $adherents\_stats['minimum\_requis']) \* 100) }}%"></div>

</div>

<div class="capacity-info">

<small class="text-muted">

Chunking automatique activé pour volumes > {{ $upload\_config['chunking\_threshold'] }} adhérents

</small>

</div>

</div>

</div>

{{-- ============================================================================

SECTION 3: INTERFACE D'IMPORT MODERNISÉE

============================================================================ --}}

<div class="import-interface">

<div class="import-header">

<h4 class="import-title">

<i class="fas fa-cloud-upload-alt me-2"></i>

Interface d'Import Intelligente

</h4>

<div class="import-options">

<div class="btn-group" role="group">

<input type="radio" class="btn-check" name="import-mode" id="mode-manuel" value="manuel" checked>

<label class="btn btn-outline-primary" for="mode-manuel">

<i class="fas fa-keyboard me-2"></i>Manuel

</label>

<input type="radio" class="btn-check" name="import-mode" id="mode-fichier" value="fichier">

<label class="btn btn-outline-success" for="mode-fichier">

<i class="fas fa-file-excel me-2"></i>Fichier

</label>

<input type="radio" class="btn-check" name="import-mode" id="mode-massif" value="massif">

<label class="btn btn-outline-warning" for="mode-massif">

<i class="fas fa-layer-group me-2"></i>Gros Volume

</label>

</div>

</div>

</div>

<!-- Section import manuel -->

<div class="import-section" id="import-manuel">

@include('operator.partials.adherent-form-manual')

</div>

<!-- Section import fichier -->

<div class="import-section" id="import-fichier" style="display: none;">

@include('operator.partials.adherent-form-upload')

</div>

<!-- Section gros volumes -->

<div class="import-section" id="import-massif" style="display: none;">

@include('operator.partials.adherent-form-chunking')

</div>

</div>

{{-- ============================================================================

SECTION 4: TABLEAU ADHÉRENTS DYNAMIQUE

============================================================================ --}}

<div class="adherents-manager">

<div class="manager-header">

<div class="d-flex justify-content-between align-items-center">

<h4 class="manager-title">

<i class="fas fa-table me-2"></i>

Gestionnaire d'Adhérents

</h4>

<div class="manager-actions">

<div class="btn-group" role="group">

<button class="btn btn-outline-secondary btn-sm" onclick="exportAdherents('csv')">

<i class="fas fa-file-csv me-1"></i>CSV

</button>

<button class="btn btn-outline-success btn-sm" onclick="exportAdherents('excel')">

<i class="fas fa-file-excel me-1"></i>Excel

</button>

<button class="btn btn-outline-info btn-sm" onclick="showAdherentsStats()">

<i class="fas fa-chart-bar me-1"></i>Stats

</button>

</div>

</div>

</div>

<!-- Filtres et recherche -->

<div class="manager-filters">

<div class="row">

<div class="col-md-4">

<div class="search-box">

<i class="fas fa-search search-icon"></i>

<input type="text" class="form-control" id="adherents-search" placeholder="Rechercher par nom, NIP...">

</div>

</div>

<div class="col-md-2">

<select class="form-select" id="filter-statut">

<option value="">Tous statuts</option>

<option value="valide">Valides</option>

<option value="anomalie">Anomalies</option>

<option value="doublon">Doublons</option>

</select>

</div>

<div class="col-md-2">

<select class="form-select" id="filter-source">

<option value="">Toutes sources</option>

<option value="manuel">Saisie manuelle</option>

<option value="import">Import fichier</option>

<option value="chunking">Import massif</option>

</select>

</div>

<div class="col-md-2">

<button class="btn btn-outline-danger w-100" onclick="clearFilters()">

<i class="fas fa-times me-1"></i>Effacer

</button>

</div>

<div class="col-md-2">

<div class="adherents-counter">

<span class="counter-label">Total:</span>

<span class="counter-number" id="adherents-total">0</span>

</div>

</div>

</div>

</div>

</div>

<!-- Tableau responsive -->

<div class="adherents-table-container">

<table class="table adherents-table" id="adherents-table">

<thead>

<tr>

<th>

<input type="checkbox" class="form-check-input" id="select-all">

</th>

<th>NIP</th>

<th>Nom Complet</th>

<th>Téléphone</th>

<th>Profession</th>

<th>Source</th>

<th>Statut</th>

<th>Actions</th>

</tr>

</thead>

<tbody id="adherents-tbody">

<!-- Sera rempli dynamiquement -->

</tbody>

</table>

<!-- État vide -->

<div class="empty-state" id="empty-state">

<div class="empty-icon">

<i class="fas fa-users fa-4x"></i>

</div>

<h5 class="empty-title">Aucun adhérent importé</h5>

<p class="empty-text">Commencez par importer vos adhérents via l'interface ci-dessus</p>

</div>

</div>

<!-- Pagination personnalisée -->

<div class="adherents-pagination" id="adherents-pagination">

<!-- Sera générée dynamiquement -->

</div>

</div>

{{-- ============================================================================

SECTION 5: ACTIONS FINALES

============================================================================ --}}

<div class="final-actions">

<div class="actions-header">

<h4 class="actions-title">

<i class="fas fa-rocket me-2"></i>

Finalisation du Dossier

</h4>

<p class="actions-subtitle">Vérifiez vos données et soumettez votre dossier à l'administration</p>

</div>

<div class="actions-grid">

<div class="action-card" id="action-draft">

<div class="action-icon bg-warning">

<i class="fas fa-save"></i>

</div>

<div class="action-content">

<h6 class="action-title">Sauvegarder en Brouillon</h6>

<p class="action-text">Conservez vos données pour finaliser plus tard</p>

<button class="btn btn-warning" onclick="saveDraft()">

<i class="fas fa-save me-2"></i>Sauvegarder

</button>

</div>

</div>

<div class="action-card" id="action-validate">

<div class="action-icon bg-info">

<i class="fas fa-check-double"></i>

</div>

<div class="action-content">

<h6 class="action-title">Valider les Données</h6>

<p class="action-text">Vérification complète avant soumission</p>

<button class="btn btn-info" onclick="validateAllData()">

<i class="fas fa-check-double me-2"></i>Valider

</button>

</div>

</div>

<div class="action-card" id="action-submit">

<div class="action-icon bg-success">

<i class="fas fa-paper-plane"></i>

</div>

<div class="action-content">

<h6 class="action-title">Soumettre à l'Administration</h6>

<p class="action-text">Envoi final pour traitement officiel</p>

<button class="btn btn-success" onclick="submitToAdmin()" id="submit-final-btn" disabled>

<i class="fas fa-paper-plane me-2"></i>Soumettre

</button>

</div>

</div>

</div>

<!-- Résumé avant soumission -->

<div class="submission-summary" id="submission-summary" style="display: none;">

<div class="summary-header">

<h5 class="summary-title">

<i class="fas fa-clipboard-check me-2"></i>

Résumé de Soumission

</h5>

</div>

<div class="summary-content" id="summary-content">

<!-- Sera rempli dynamiquement -->

</div>

</div>

</div>

</div>

{{-- ============================================================================

MODALS ET INTERFACES SECONDAIRES

============================================================================ --}}

<!-- Modal de progression chunking -->

<div class="modal fade" id="chunkingProgressModal" tabindex="-1" data-bs-backdrop="static">

<div class="modal-dialog modal-lg">

<div class="modal-content">

@include('operator.partials.chunking-progress-modal')

</div>

</div>

</div>

<!-- Modal d'aide -->

<div class="modal fade" id="helpModal" tabindex="-1">

<div class="modal-dialog modal-xl">

<div class="modal-content">

@include('operator.partials.help-modal')

</div>

</div>

</div>

<!-- Modal de validation finale -->

<div class="modal fade" id="finalValidationModal" tabindex="-1">

<div class="modal-dialog modal-lg">

<div class="modal-content">

@include('operator.partials.final-validation-modal')

</div>

</div>

</div>

{{-- Formulaire de soumission caché --}}

<form id="hidden-submission-form" method="POST" action="{{ $urls['store\_adherents'] }}" style="display: none;">

@csrf

<input type="hidden" name="adherents\_data" id="adherents-data-input">

<input type="hidden" name="submission\_type" value="final">

<input type="hidden" name="validation\_passed" id="validation-passed-input">

</form>

{{-- Configuration JavaScript --}}

<script>

// Configuration globale pour les modules JS

window.ConfirmationConfig = {

dossierId: {{ $dossier->id }},

organisationId: {{ $organisation->id }},

organisationNom: '{{ $organisation->nom }}',

organisationType: '{{ $organisation->type }}',

// Statistiques

stats: {

existants: {{ $adherents\_stats['existants'] }},

minimumRequis: {{ $adherents\_stats['minimum\_requis'] }},

manquants: {{ $adherents\_stats['manquants'] }},

peutSoumettre: {{ $adherents\_stats['peut\_soumettre'] ? 'true' : 'false' }}

},

// Configuration upload

upload: {

maxFileSize: '{{ $upload\_config['max\_file\_size'] }}',

chunkSize: {{ $upload\_config['chunk\_size'] }},

maxAdherents: {{ $upload\_config['max\_adherents'] }},

chunkingThreshold: {{ $upload\_config['chunking\_threshold'] }}

},

// URLs API

urls: {!! json\_encode($urls) !!},

// CSRF

csrf: '{{ csrf\_token() }}',

// Configuration chunking

chunking: {

enabled: true,

threshold: {{ $upload\_config['chunking\_threshold'] }},

batchSize: {{ $upload\_config['chunk\_size'] }},

maxRetries: 3,

pauseBetweenChunks: 500

}

};

// Initialiser les adhérents existants s'il y en a

window.AdherentsData = [];

</script>

@endsection

{{-- Scripts JavaScript externalisés --}}

@push('scripts')

{{-- Bibliothèques externes --}}

<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/xlsx/0.18.5/xlsx.full.min.js"></script>

{{-- Modules principaux --}}

<script src="{{ asset('js/confirmation-app.js') }}"></script>

<script src="{{ asset('js/adherents-manager.js') }}"></script>

<script src="{{ asset('js/chunking-engine.js') }}"></script>

<script src="{{ asset('js/file-upload-sglp.js') }}"></script>

<script src="{{ asset('js/validation-engine.js') }}"></script>

{{-- Initialisation --}}

<script>

document.addEventListener('DOMContentLoaded', function() {

console.log('🇬🇦 Initialisation Interface Import Adhérents SGLP v2.0');

// Initialiser l'application principale

if (window.ConfirmationApp) {

window.ConfirmationApp.init();

}

// Initialiser le gestionnaire d'adhérents

if (window.AdherentsManager) {

window.AdherentsManager.init();

}

// Initialiser le moteur de chunking

if (window.ChunkingEngine) {

window.ChunkingEngine.init();

}

console.log('✅ Tous les modules initialisés avec succès');

});

</script>

@endpush

/\* ============================================================================

GABON-CHARTE.CSS - CHARTE GRAPHIQUE OFFICIELLE DU GABON

Version: 2.0 - Compatible avec SGLP et Bootstrap 5

Couleurs officielles et éléments de design gabonais réutilisables

============================================================================ \*/

/\* ============================================================================

VARIABLES CSS OFFICIELLES DU GABON

============================================================================ \*/

:root {

/\* Couleurs officielles du drapeau gabonais \*/

--gabon-green: #009e3f;

--gabon-green-light: #00b347;

--gabon-green-dark: #006d2c;

--gabon-green-ultra-light: #e8f5e8;

--gabon-yellow: #ffcd00;

--gabon-yellow-light: #ffd700;

--gabon-yellow-dark: #b8930b;

--gabon-yellow-ultra-light: #fffacd;

--gabon-blue: #003f7f;

--gabon-blue-light: #0056b3;

--gabon-blue-dark: #002855;

--gabon-blue-ultra-light: #e6f0ff;

/\* Gradients officiels \*/

--gradient-primary: linear-gradient(135deg, var(--gabon-green) 0%, var(--gabon-green-light) 100%);

--gradient-secondary: linear-gradient(135deg, var(--gabon-blue) 0%, var(--gabon-blue-light) 100%);

--gradient-accent: linear-gradient(135deg, var(--gabon-yellow) 0%, var(--gabon-yellow-light) 100%);

--gradient-tricolore: linear-gradient(135deg, var(--gabon-green) 0%, var(--gabon-yellow) 50%, var(--gabon-blue) 100%);

/\* Ombres et effets \*/

--shadow-xs: 0 1px 3px rgba(0, 0, 0, 0.05);

--shadow-sm: 0 2px 8px rgba(0, 0, 0, 0.1);

--shadow-md: 0 4px 16px rgba(0, 0, 0, 0.15);

--shadow-lg: 0 8px 32px rgba(0, 0, 0, 0.2);

--shadow-xl: 0 16px 64px rgba(0, 0, 0, 0.25);

/\* Transitions \*/

--transition-fast: 0.15s ease;

--transition-normal: 0.3s ease;

--transition-slow: 0.5s ease;

/\* Bordures et rayons \*/

--border-radius-sm: 0.375rem;

--border-radius-md: 0.5rem;

--border-radius-lg: 0.75rem;

--border-radius-xl: 1rem;

--border-radius-pill: 50px;

/\* Espacements gabonais \*/

--spacing-xs: 0.25rem;

--spacing-sm: 0.5rem;

--spacing-md: 1rem;

--spacing-lg: 1.5rem;

--spacing-xl: 2rem;

--spacing-xxl: 3rem;

}

/\* ============================================================================

TYPOGRAPHIE GABONAISE

============================================================================ \*/

body {

font-family: 'Inter', 'Segoe UI', 'Roboto', sans-serif;

background: linear-gradient(135deg, #f8f9fa 0%, #e9ecef 100%);

color: #212529;

line-height: 1.6;

min-height: 100vh;

}

.gabon-title {

font-weight: 700;

color: var(--gabon-green);

text-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

.gabon-subtitle {

font-weight: 500;

color: var(--gabon-blue);

opacity: 0.9;

}

.gabon-text-tricolore {

background: var(--gradient-tricolore);

-webkit-background-clip: text;

-webkit-text-fill-color: transparent;

background-clip: text;

font-weight: 700;

}

/\* ============================================================================

BOUTONS GABONAIS

============================================================================ \*/

.btn-gabon {

padding: 0.75rem 1.5rem;

border-radius: var(--border-radius-pill);

font-weight: 600;

text-transform: uppercase;

letter-spacing: 0.5px;

transition: all var(--transition-normal);

border: none;

position: relative;

overflow: hidden;

min-width: 120px;

}

.btn-gabon::before {

content: '';

position: absolute;

top: 0;

left: -100%;

width: 100%;

height: 100%;

background: linear-gradient(90deg, transparent, rgba(255, 255, 255, 0.3), transparent);

transition: left 0.5s;

}

.btn-gabon:hover::before {

left: 100%;

}

.btn-gabon-primary {

background: var(--gradient-primary);

color: white;

box-shadow: 0 4px 15px rgba(0, 158, 63, 0.3);

}

.btn-gabon-primary:hover {

transform: translateY(-2px);

box-shadow: 0 8px 25px rgba(0, 158, 63, 0.4);

color: white;

}

.btn-gabon-secondary {

background: var(--gradient-secondary);

color: white;

box-shadow: 0 4px 15px rgba(0, 63, 127, 0.3);

}

.btn-gabon-secondary:hover {

transform: translateY(-2px);

box-shadow: 0 8px 25px rgba(0, 63, 127, 0.4);

color: white;

}

.btn-gabon-accent {

background: var(--gradient-accent);

color: #333;

box-shadow: 0 4px 15px rgba(255, 205, 0, 0.3);

}

.btn-gabon-accent:hover {

transform: translateY(-2px);

box-shadow: 0 8px 25px rgba(255, 205, 0, 0.4);

color: #333;

}

.btn-gabon-outline {

background: transparent;

color: var(--gabon-green);

border: 2px solid var(--gabon-green);

}

.btn-gabon-outline:hover {

background: var(--gabon-green);

color: white;

transform: translateY(-2px);

}

/\* ============================================================================

CARTES GABONAISES

============================================================================ \*/

.card-gabon {

border: none;

border-radius: var(--border-radius-lg);

box-shadow: var(--shadow-md);

transition: all var(--transition-normal);

overflow: hidden;

}

.card-gabon:hover {

transform: translateY(-5px);

box-shadow: var(--shadow-lg);

}

.card-gabon-primary {

border-left: 5px solid var(--gabon-green);

}

.card-gabon-secondary {

border-left: 5px solid var(--gabon-blue);

}

.card-gabon-accent {

border-left: 5px solid var(--gabon-yellow);

}

.card-gabon-gradient {

background: var(--gradient-primary);

color: white;

}

.card-gabon-tricolore::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

height: 4px;

background: var(--gradient-tricolore);

}

/\* ============================================================================

BADGES ET INDICATEURS

============================================================================ \*/

.badge-gabon {

padding: 0.5rem 1rem;

border-radius: var(--border-radius-pill);

font-weight: 600;

text-transform: uppercase;

letter-spacing: 0.5px;

font-size: 0.75rem;

}

.badge-gabon-primary {

background: var(--gradient-primary);

color: white;

}

.badge-gabon-secondary {

background: var(--gradient-secondary);

color: white;

}

.badge-gabon-accent {

background: var(--gradient-accent);

color: #333;

}

.badge-gabon-success {

background: linear-gradient(135deg, #28a745 0%, #20c997 100%);

color: white;

}

.badge-gabon-warning {

background: linear-gradient(135deg, #ffc107 0%, #fd7e14 100%);

color: #333;

}

.badge-gabon-danger {

background: linear-gradient(135deg, #dc3545 0%, #c82333 100%);

color: white;

}

/\* ============================================================================

BARRES DE PROGRESSION GABONAISES

============================================================================ \*/

.progress-gabon {

height: 1rem;

border-radius: var(--border-radius-pill);

background-color: rgba(0, 0, 0, 0.1);

overflow: hidden;

box-shadow: inset 0 1px 2px rgba(0, 0, 0, 0.1);

}

.progress-gabon .progress-bar {

background: var(--gradient-primary);

border-radius: var(--border-radius-pill);

transition: width 0.6s ease;

position: relative;

overflow: hidden;

}

.progress-gabon .progress-bar::before {

content: '';

position: absolute;

top: 0;

left: 0;

bottom: 0;

right: 0;

background-image: linear-gradient(

45deg,

rgba(255, 255, 255, 0.15) 25%,

transparent 25%,

transparent 50%,

rgba(255, 255, 255, 0.15) 50%,

rgba(255, 255, 255, 0.15) 75%,

transparent 75%,

transparent

);

background-size: 1rem 1rem;

animation: progress-bar-stripes 1s linear infinite;

}

@keyframes progress-bar-stripes {

0% { background-position: 1rem 0; }

100% { background-position: 0 0; }

}

/\* ============================================================================

FORMULAIRES GABONAIS

============================================================================ \*/

.form-control-gabon {

border: 2px solid #e9ecef;

border-radius: var(--border-radius-md);

padding: 0.75rem 1rem;

transition: all var(--transition-normal);

background: white;

}

.form-control-gabon:focus {

border-color: var(--gabon-green);

box-shadow: 0 0 0 0.2rem rgba(0, 158, 63, 0.25);

background: var(--gabon-green-ultra-light);

}

.form-control-gabon.is-valid {

border-color: var(--gabon-green);

background-image: url("data:image/svg+xml,%3csvg xmlns='http://www.w3.org/2000/svg' width='8' height='8' viewBox='0 0 8 8'%3e%3cpath fill='%23009e3f' d='m2.3 6.73.94-.94 1.64 1.64 3.62-3.62.94.94-4.56 4.56z'/%3e%3c/svg%3e");

}

.form-control-gabon.is-invalid {

border-color: #dc3545;

background-image: url("data:image/svg+xml,%3csvg xmlns='http://www.w3.org/2000/svg' width='12' height='12' fill='none' stroke='%23dc3545' viewBox='0 0 12 12'%3e%3ccircle cx='6' cy='6' r='4.5'/%3e%3cpath d='m5.8 3.6h.4L6 6.5z'/%3e%3ccircle cx='6' cy='8.2' r='.6' fill='%23dc3545' stroke='none'/%3e%3c/svg%3e");

}

.form-select-gabon {

border: 2px solid #e9ecef;

border-radius: var(--border-radius-md);

padding: 0.75rem 1rem;

background: white;

transition: all var(--transition-normal);

}

.form-select-gabon:focus {

border-color: var(--gabon-green);

box-shadow: 0 0 0 0.2rem rgba(0, 158, 63, 0.25);

}

/\* ============================================================================

BREADCRUMB GABONAIS

============================================================================ \*/

.breadcrumb-gabon {

background: rgba(255, 255, 255, 0.1);

border-radius: var(--border-radius-pill);

padding: 0.5rem 1rem;

margin-bottom: 1rem;

}

.breadcrumb-gabon .breadcrumb {

margin-bottom: 0;

}

.breadcrumb-gabon .breadcrumb-item + .breadcrumb-item::before {

content: '›';

color: rgba(255, 255, 255, 0.7);

font-weight: bold;

}

.breadcrumb-gabon .breadcrumb-item a {

color: rgba(255, 255, 255, 0.8);

text-decoration: none;

transition: color var(--transition-fast);

}

.breadcrumb-gabon .breadcrumb-item a:hover {

color: white;

}

.breadcrumb-gabon .breadcrumb-item.active {

color: white;

font-weight: 600;

}

/\* ============================================================================

ALERTES GABONAISES

============================================================================ \*/

.alert-gabon {

border: none;

border-radius: var(--border-radius-lg);

padding: 1rem 1.5rem;

border-left: 5px solid;

}

.alert-gabon-success {

background: var(--gabon-green-ultra-light);

border-left-color: var(--gabon-green);

color: var(--gabon-green-dark);

}

.alert-gabon-info {

background: var(--gabon-blue-ultra-light);

border-left-color: var(--gabon-blue);

color: var(--gabon-blue-dark);

}

.alert-gabon-warning {

background: var(--gabon-yellow-ultra-light);

border-left-color: var(--gabon-yellow);

color: var(--gabon-yellow-dark);

}

.alert-gabon-danger {

background: #f8d7da;

border-left-color: #dc3545;

color: #721c24;

}

/\* ============================================================================

ICÔNES ET CERCLES

============================================================================ \*/

.icon-circle {

width: 60px;

height: 60px;

border-radius: 50%;

display: flex;

align-items: center;

justify-content: center;

transition: all var(--transition-normal);

}

.icon-circle-primary {

background: var(--gradient-primary);

color: white;

}

.icon-circle-secondary {

background: var(--gradient-secondary);

color: white;

}

.icon-circle-accent {

background: var(--gradient-accent);

color: #333;

}

.icon-circle:hover {

transform: scale(1.1);

box-shadow: var(--shadow-md);

}

/\* ============================================================================

ANIMATIONS GABONAISES

============================================================================ \*/

@keyframes gabon-fade-in {

from {

opacity: 0;

transform: translateY(20px);

}

to {

opacity: 1;

transform: translateY(0);

}

}

@keyframes gabon-slide-in-left {

from {

opacity: 0;

transform: translateX(-30px);

}

to {

opacity: 1;

transform: translateX(0);

}

}

@keyframes gabon-slide-in-right {

from {

opacity: 0;

transform: translateX(30px);

}

to {

opacity: 1;

transform: translateX(0);

}

}

@keyframes gabon-bounce-in {

0% {

opacity: 0;

transform: scale(0.3);

}

50% {

opacity: 1;

transform: scale(1.05);

}

70% {

transform: scale(0.9);

}

100% {

opacity: 1;

transform: scale(1);

}

}

@keyframes gabon-pulse {

0%, 100% {

transform: scale(1);

}

50% {

transform: scale(1.05);

}

}

/\* Classes utilitaires pour animations \*/

.gabon-fade-in {

animation: gabon-fade-in 0.6s ease-out;

}

.gabon-slide-in-left {

animation: gabon-slide-in-left 0.5s ease-out;

}

.gabon-slide-in-right {

animation: gabon-slide-in-right 0.5s ease-out;

}

.gabon-bounce-in {

animation: gabon-bounce-in 0.8s ease-out;

}

.gabon-pulse {

animation: gabon-pulse 2s ease-in-out infinite;

}

/\* ============================================================================

MOTIFS ET TEXTURES GABONAIS

============================================================================ \*/

.gabon-pattern-bg {

background-image: url('data:image/svg+xml,<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 100 100"><defs><pattern id="gabon-pattern" x="0" y="0" width="25" height="25" patternUnits="userSpaceOnUse"><circle cx="12.5" cy="12.5" r="2" fill="rgba(255,255,255,0.1)"/></pattern></defs><rect width="100" height="100" fill="url(%23gabon-pattern)"/></svg>');

}

.gabon-texture-overlay::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

bottom: 0;

background: var(--gabon-pattern-bg);

opacity: 0.1;

pointer-events: none;

}

/\* ============================================================================

RESPONSIVE ET ADAPTABILITÉ

============================================================================ \*/

@media (max-width: 768px) {

.btn-gabon {

padding: 0.6rem 1.2rem;

font-size: 0.875rem;

min-width: 100px;

}

.icon-circle {

width: 50px;

height: 50px;

}

.card-gabon {

margin-bottom: 1rem;

}

.breadcrumb-gabon {

padding: 0.375rem 0.75rem;

}

}

@media (max-width: 576px) {

.btn-gabon {

width: 100%;

margin-bottom: 0.5rem;

}

.gabon-text-tricolore {

font-size: 1.5rem;

}

}

/\* ============================================================================

UTILITAIRES SPÉCIAUX

============================================================================ \*/

.gabon-shadow-sm { box-shadow: var(--shadow-sm); }

.gabon-shadow-md { box-shadow: var(--shadow-md); }

.gabon-shadow-lg { box-shadow: var(--shadow-lg); }

.gabon-shadow-xl { box-shadow: var(--shadow-xl); }

.gabon-rounded-sm { border-radius: var(--border-radius-sm); }

.gabon-rounded-md { border-radius: var(--border-radius-md); }

.gabon-rounded-lg { border-radius: var(--border-radius-lg); }

.gabon-rounded-xl { border-radius: var(--border-radius-xl); }

.gabon-rounded-pill { border-radius: var(--border-radius-pill); }

.gabon-bg-primary { background: var(--gradient-primary); }

.gabon-bg-secondary { background: var(--gradient-secondary); }

.gabon-bg-accent { background: var(--gradient-accent); }

.gabon-bg-tricolore { background: var(--gradient-tricolore); }

.gabon-text-primary { color: var(--gabon-green) !important; }

.gabon-text-secondary { color: var(--gabon-blue) !important; }

.gabon-text-accent { color: var(--gabon-yellow-dark) !important; }

/\* ============================================================================

CONFIRMATION-IMPORT.CSS - STYLES INTERFACE D'IMPORT ADHÉRENTS

Version: 2.0 - Spécifique à confirmation.blade.php

Styles pour l'interface d'import modernisée avec détection automatique

============================================================================ \*/

/\* ============================================================================

HEADER PRINCIPAL

============================================================================ \*/

.confirmation-header {

background: var(--gradient-secondary);

color: white;

padding: 2rem 0;

margin-bottom: 2rem;

box-shadow: var(--shadow-lg);

position: relative;

overflow: hidden;

}

.confirmation-header::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

bottom: 0;

background: var(--gabon-pattern-bg);

opacity: 0.1;

}

.header-content {

position: relative;

z-index: 2;

container: container;

padding: 0 15px;

}

.phase-indicator {

background: rgba(255, 255, 255, 0.15);

border-radius: var(--border-radius-pill);

padding: 0.5rem 1.5rem;

display: inline-block;

margin-bottom: 1rem;

border: 2px solid rgba(255, 255, 255, 0.2);

animation: gabon-pulse 3s ease-in-out infinite;

}

.header-title {

font-size: 2.5rem;

font-weight: 700;

margin-bottom: 0.5rem;

text-shadow: 0 2px 4px rgba(0, 0, 0, 0.3);

}

.header-subtitle {

font-size: 1.2rem;

opacity: 0.9;

margin-bottom: 0.5rem;

}

.header-meta {

font-size: 0.95rem;

opacity: 0.8;

}

.header-meta .badge {

background: rgba(255, 255, 255, 0.2);

color: white;

border: 1px solid rgba(255, 255, 255, 0.3);

}

.header-actions {

display: flex;

gap: 0.5rem;

align-items: center;

}

/\* ============================================================================

DASHBOARD STATISTIQUES

============================================================================ \*/

.stats-dashboard {

background: white;

border-radius: var(--border-radius-xl);

padding: 2rem;

margin-bottom: 2rem;

box-shadow: var(--shadow-md);

border: 1px solid rgba(0, 0, 0, 0.05);

}

.dashboard-header {

text-align: center;

margin-bottom: 2rem;

padding-bottom: 1rem;

border-bottom: 2px solid #f8f9fa;

}

.dashboard-title {

color: var(--gabon-green);

font-weight: 700;

margin-bottom: 0.5rem;

}

.dashboard-subtitle {

color: #6c757d;

font-size: 0.95rem;

margin-bottom: 0;

}

.stats-grid {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));

gap: 1.5rem;

margin-bottom: 2rem;

}

.stat-card {

background: linear-gradient(135deg, #ffffff 0%, #f8f9fa 100%);

border-radius: var(--border-radius-lg);

padding: 1.5rem;

text-align: center;

transition: all var(--transition-normal);

border: 2px solid #e9ecef;

position: relative;

overflow: hidden;

}

.stat-card::before {

content: '';

position: absolute;

top: 0;

left: 0;

right: 0;

height: 4px;

background: var(--gabon-green);

transform: scaleX(0);

transition: transform var(--transition-normal);

}

.stat-card:hover::before {

transform: scaleX(1);

}

.stat-card:hover {

transform: translateY(-8px);

box-shadow: var(--shadow-lg);

border-color: var(--gabon-green);

}

.stat-card.highlight {

background: var(--gradient-primary);

color: white;

border-color: var(--gabon-green-dark);

}

.stat-card.warning {

background: var(--gradient-accent);

color: #333;

border-color: var(--gabon-yellow-dark);

}

.stat-card.danger {

background: linear-gradient(135deg, #dc3545 0%, #c82333 100%);

color: white;

border-color: #bd2130;

}

.stat-card.success {

background: linear-gradient(135deg, #28a745 0%, #20c997 100%);

color: white;

border-color: #1e7e34;

}

.stat-card.info {

background: linear-gradient(135deg, #17a2b8 0%, #138496 100%);

color: white;

border-color: #117a8b;

}

.stat-icon {

font-size: 2.5rem;

margin-bottom: 1rem;

opacity: 0.8;

}

.stat-content {

position: relative;

}

.stat-number {

font-size: 2rem;

font-weight: 700;

margin-bottom: 0.5rem;

}

.stat-label {

font-size: 0.9rem;

opacity: 0.9;

font-weight: 500;

}

/\* ============================================================================

INDICATEUR DE CAPACITÉ

============================================================================ \*/

.capacity-indicator {

background: #f8f9fa;

border-radius: var(--border-radius-lg);

padding: 1.5rem;

border: 1px solid #e9ecef;

}

.capacity-header {

display: flex;

justify-content: space-between;

align-items: center;

margin-bottom: 1rem;

}

.capacity-label {

font-weight: 600;

color: var(--gabon-blue);

}

.capacity-mode {

font-size: 0.875rem;

padding: 0.25rem 0.75rem;

border-radius: var(--border-radius-pill);

background: var(--gabon-green);

color: white;

font-weight: 600;

}

.capacity-bar {

height: 12px;

background: #e9ecef;

border-radius: var(--border-radius-pill);

overflow: hidden;

margin-bottom: 0.5rem;

}

.capacity-fill {

height: 100%;

background: var(--gradient-primary);

border-radius: var(--border-radius-pill);

transition: width 0.8s ease;

position: relative;

}

.capacity-fill::after {

content: '';

position: absolute;

top: 0;

left: 0;

bottom: 0;

right: 0;

background-image: linear-gradient(

45deg,

rgba(255, 255, 255, 0.15) 25%,

transparent 25%,

transparent 50%,

rgba(255, 255, 255, 0.15) 50%,

rgba(255, 255, 255, 0.15) 75%,

transparent 75%,

transparent

);

background-size: 1rem 1rem;

animation: progress-bar-stripes 1s linear infinite;

}

.capacity-info {

text-align: center;

}

/\* ============================================================================

INTERFACE D'IMPORT

============================================================================ \*/

.import-interface {

background: white;

border-radius: var(--border-radius-xl);

padding: 2rem;

margin-bottom: 2rem;

box-shadow: var(--shadow-md);

border: 1px solid rgba(0, 0, 0, 0.05);

}

.import-header {

display: flex;

justify-content: between;

align-items: center;

margin-bottom: 2rem;

padding-bottom: 1rem;

border-bottom: 2px solid #f8f9fa;

}

.import-title {

color: var(--gabon-green);

font-weight: 700;

margin-bottom: 0;

}

.import-options .btn-group {

border-radius: var(--border-radius-pill);

overflow: hidden;

box-shadow: var(--shadow-sm);

}

.import-options .btn {

border-radius: 0;

border: 2px solid var(--gabon-green);

color: var(--gabon-green);

background: white;

font-weight: 600;

transition: all var(--transition-fast);

}

.import-options .btn:first-child {

border-radius: var(--border-radius-pill) 0 0 var(--border-radius-pill);

}

.import-options .btn:last-child {

border-radius: 0 var(--border-radius-pill) var(--border-radius-pill) 0;

}

.import-options .btn.active,

.import-options .btn-check:checked + .btn {

background: var(--gradient-primary);

color: white;

border-color: var(--gabon-green);

}

.import-section {

background: #f8f9fa;

border-radius: var(--border-radius-lg);

padding: 2rem;

border: 2px dashed #dee2e6;

transition: all var(--transition-normal);

}

.import-section.active {

border-color: var(--gabon-green);

background: var(--gabon-green-ultra-light);

}

/\* Styles pour zones de drop \*/

.upload-zone {

border: 3px dashed #dee2e6;

border-radius: var(--border-radius-lg);

padding: 3rem 2rem;

text-align: center;

background: #f8f9fa;

transition: all var(--transition-normal);

cursor: pointer;

position: relative;

overflow: hidden;

}

.upload-zone:hover {

border-color: var(--gabon-green);

background: var(--gabon-green-ultra-light);

transform: scale(1.02);

}

.upload-zone.dragover {

border-color: var(--gabon-green);

background: var(--gabon-green-ultra-light);

transform: scale(1.05);

box-shadow: var(--shadow-lg);

}

.upload-zone::before {

content: '';

position: absolute;

top: 0;

left: -100%;

width: 100%;

height: 100%;

background: linear-gradient(90deg, transparent, rgba(0, 158, 63, 0.1), transparent);

transition: left 0.5s;

}

.upload-zone:hover::before {

left: 100%;

}

.upload-icon {

font-size: 4rem;

color: var(--gabon-green);

margin-bottom: 1rem;

transition: all var(--transition-normal);

}

.upload-zone:hover .upload-icon {

transform: scale(1.2);

color: var(--gabon-green-dark);

}

.upload-title {

font-size: 1.5rem;

font-weight: 700;

color: var(--gabon-green);

margin-bottom: 1rem;

}

.upload-subtitle {

color: #6c757d;

margin-bottom: 2rem;

}

/\* ============================================================================

FORMULAIRES DE SAISIE

============================================================================ \*/

.manual-form {

background: white;

border-radius: var(--border-radius-lg);

padding: 2rem;

border: 1px solid #e9ecef;

}

.manual-form .form-control,

.manual-form .form-select {

border: 2px solid #e9ecef;

border-radius: var(--border-radius-md);

padding: 0.75rem 1rem;

transition: all var(--transition-normal);

}

.manual-form .form-control:focus,

.manual-form .form-select:focus {

border-color: var(--gabon-green);

box-shadow: 0 0 0 0.2rem rgba(0, 158, 63, 0.25);

}

.manual-form .btn-add {

background: var(--gradient-primary);

color: white;

border: none;

border-radius: var(--border-radius-pill);

padding: 0.75rem 2rem;

font-weight: 600;

transition: all var(--transition-normal);

}

.manual-form .btn-add:hover {

transform: translateY(-2px);

box-shadow: var(--shadow-md);

}

/\* ============================================================================

INDICATEURS ET ÉTATS

============================================================================ \*/

.processing-indicator {

display: flex;

align-items: center;

justify-content: center;

padding: 2rem;

background: var(--gabon-green-ultra-light);

border-radius: var(--border-radius-lg);

border: 2px solid var(--gabon-green);

}

.processing-spinner {

width: 3rem;

height: 3rem;

border: 4px solid rgba(0, 158, 63, 0.3);

border-top: 4px solid var(--gabon-green);

border-radius: 50%;

animation: spin 1s linear infinite;

margin-right: 1rem;

}

@keyframes spin {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

.success-indicator {

text-align: center;

padding: 2rem;

background: linear-gradient(135deg, #d4edda 0%, #c3e6cb 100%);

border-radius: var(--border-radius-lg);

border: 2px solid #28a745;

}

.success-icon {

font-size: 4rem;

color: #28a745;

margin-bottom: 1rem;

animation: gabon-bounce-in 0.8s ease-out;

}

.error-indicator {

text-align: center;

padding: 2rem;

background: linear-gradient(135deg, #f8d7da 0%, #f5c6cb 100%);

border-radius: var(--border-radius-lg);

border: 2px solid #dc3545;

}

.error-icon {

font-size: 4rem;

color: #dc3545;

margin-bottom: 1rem;

animation: shake 0.5s ease-in-out;

}

@keyframes shake {

0%, 100% { transform: translateX(0); }

10%, 30%, 50%, 70%, 90% { transform: translateX(-5px); }

20%, 40%, 60%, 80% { transform: translateX(5px); }

}

/\* ============================================================================

RESPONSIVE DESIGN

============================================================================ \*/

@media (max-width: 1200px) {

.stats-grid {

grid-template-columns: repeat(auto-fit, minmax(180px, 1fr));

gap: 1rem;

}

}

@media (max-width: 768px) {

.confirmation-header {

padding: 1.5rem 0;

}

.header-title {

font-size: 2rem;

}

.header-subtitle {

font-size: 1rem;

}

.header-actions {

flex-direction: column;

gap: 0.5rem;

margin-top: 1rem;

}

.import-header {

flex-direction: column;

gap: 1rem;

align-items: stretch;

}

.import-options .btn-group {

width: 100%;

}

.stats-grid {

grid-template-columns: 1fr;

gap: 1rem;

}

.stat-card {

padding: 1rem;

}

.stat-number {

font-size: 1.5rem;

}

.upload-zone {

padding: 2rem 1rem;

}

.upload-icon {

font-size: 3rem;

}

.upload-title {

font-size: 1.2rem;

}

}

@media (max-width: 576px) {

.stats-dashboard,

.import-interface {

padding: 1rem;

margin-bottom: 1rem;

}

.dashboard-header,

.import-header {

margin-bottom: 1rem;

padding-bottom: 0.5rem;

}

.capacity-header {

flex-direction: column;

gap: 0.5rem;

text-align: center;

}

.manual-form {

padding: 1rem;

}

}

/\* ============================================================================

ÉTATS D'INTERACTION

============================================================================ \*/

.import-section:not(.active) {

display: none;

}

.import-section.active {

display: block;

animation: gabon-fade-in 0.5s ease-out;

}

.stat-card.updating {

animation: gabon-pulse 1s ease-in-out infinite;

}

.capacity-mode.chunking {

background: var(--gabon-yellow);

color: #333;

}

.capacity-mode.standard {

background: var(--gabon-green);

color: white;

}

.capacity-mode.massive {

background: var(--gabon-blue);

color: white;

}

/\* ============================================================================

UTILITAIRES SPÉCIFIQUES

============================================================================ \*/

.fade-in-up {

animation: gabon-fade-in 0.6s ease-out;

}

.slide-in-left {

animation: gabon-slide-in-left 0.5s ease-out;

}

.slide-in-right {

animation: gabon-slide-in-right 0.5s ease-out;

}

.bounce-in {

animation: gabon-bounce-in 0.8s ease-out;

}

.pulse {

animation: gabon-pulse 2s ease-in-out infinite;

}

/\* ============================================================================

THÈME SOMBRE (OPTIONNEL)

============================================================================ \*/

@media (prefers-color-scheme: dark) {

.stats-dashboard,

.import-interface {

background: #2d3748;

color: white;

}

.stat-card:not(.highlight):not(.warning):not(.danger):not(.success):not(.info) {

background: linear-gradient(135deg, #4a5568 0%, #2d3748 100%);

color: white;

border-color: #4a5568;

}

.import-section {

background: #2d3748;

border-color: #4a5568;

}

.upload-zone {

background: #2d3748;

border-color: #4a5568;

color: white;

}

.manual-form {

background: #2d3748;

border-color: #4a5568;

}

.manual-form .form-control,

.manual-form .form-select {

background: #4a5568;

border-color: #718096;

color: white;

}

}

**Maintenant créons les modules JavaScript principaux :**

/\*\*

\* ============================================================================

\* CONFIRMATION-APP.JS - MODULE PRINCIPAL INTERFACE CONFIRMATION

\* Version: 2.0 - Application SGLP avec architecture modulaire

\* ============================================================================

\*

\* Module principal pour confirmation.blade.php

\* Gestion centralisée des modes d'import et coordination des modules

\*/

window.ConfirmationApp = window.ConfirmationApp || {};

// ============================================================================

// CONFIGURATION ET ÉTAT GLOBAL

// ============================================================================

/\*\*

\* Configuration de l'application

\*/

window.ConfirmationApp.config = {

dossierId: null,

organisationId: null,

organisationNom: '',

organisationType: '',

stats: {

existants: 0,

minimumRequis: 15,

manquants: 0,

peutSoumettre: false

},

upload: {

maxFileSize: '10MB',

chunkSize: 100,

maxAdherents: 50000,

chunkingThreshold: 200

},

urls: {},

csrf: '',

chunking: {

enabled: true,

threshold: 200,

batchSize: 100,

maxRetries: 3,

pauseBetweenChunks: 500

}

};

/\*\*

\* État global de l'application

\*/

window.ConfirmationApp.state = {

currentMode: 'manuel', // manuel, fichier, massif

importInProgress: false,

adherentsData: [],

validationPassed: false,

canSubmit: false,

lastActivity: Date.now()

};

/\*\*

\* Gestionnaires d'événements

\*/

window.ConfirmationApp.handlers = {};

// ============================================================================

// INITIALISATION PRINCIPALE

// ============================================================================

/\*\*

\* Initialiser l'application confirmation

\*/

window.ConfirmationApp.init = function() {

console.log('🚀 Initialisation ConfirmationApp SGLP v2.0');

try {

// Récupérer la configuration depuis le blade

this.loadConfiguration();

// Initialiser les composants UI

this.initializeUI();

// Configurer les gestionnaires d'événements

this.setupEventHandlers();

// Initialiser les modules

this.initializeModules();

// Vérifier l'état initial

this.checkInitialState();

console.log('✅ ConfirmationApp initialisé avec succès');

} catch (error) {

console.error('❌ Erreur initialisation ConfirmationApp:', error);

this.showError('Erreur d\'initialisation de l\'application');

}

};

/\*\*

\* Charger la configuration depuis window.ConfirmationConfig

\*/

window.ConfirmationApp.loadConfiguration = function() {

if (window.ConfirmationConfig) {

this.config = { ...this.config, ...window.ConfirmationConfig };

console.log('📋 Configuration chargée:', this.config);

} else {

console.warn('⚠️ Configuration ConfirmationConfig non trouvée, utilisation des valeurs par défaut');

}

// Initialiser les adhérents existants

if (window.AdherentsData && Array.isArray(window.AdherentsData)) {

this.state.adherentsData = [...window.AdherentsData];

console.log(`📊 ${this.state.adherentsData.length} adhérents chargés`);

}

};

/\*\*

\* Initialiser l'interface utilisateur

\*/

window.ConfirmationApp.initializeUI = function() {

// Mettre à jour les statistiques

this.updateStatistics();

// Configurer les indicateurs de capacité

this.updateCapacityIndicator();

// Initialiser le mode d'import par défaut

this.switchImportMode('manuel');

// Configurer les tooltips et autres éléments UI

this.initializeTooltips();

};

/\*\*

\* Configurer les gestionnaires d'événements

\*/

window.ConfirmationApp.setupEventHandlers = function() {

// Gestionnaires de mode d'import

this.setupImportModeHandlers();

// Gestionnaires de soumission

this.setupSubmissionHandlers();

// Gestionnaires de sauvegarde

this.setupSaveHandlers();

// Gestionnaire de fermeture de page

this.setupUnloadHandler();

// Actualisation automatique des statistiques

this.setupStatsRefresh();

};

/\*\*

\* Gestionnaires de changement de mode d'import

\*/

window.ConfirmationApp.setupImportModeHandlers = function() {

const modeInputs = document.querySelectorAll('input[name="import-mode"]');

modeInputs.forEach(input => {

input.addEventListener('change', (e) => {

if (e.target.checked) {

this.switchImportMode(e.target.value);

}

});

});

};

/\*\*

\* Changer le mode d'import

\*/

window.ConfirmationApp.switchImportMode = function(mode) {

console.log(`🔄 Changement mode d'import: ${this.state.currentMode} → ${mode}`);

// Masquer toutes les sections

document.querySelectorAll('.import-section').forEach(section => {

section.style.display = 'none';

section.classList.remove('active');

});

// Afficher la section correspondante

const targetSection = document.getElementById(`import-${mode}`);

if (targetSection) {

targetSection.style.display = 'block';

targetSection.classList.add('active');

}

// Mettre à jour l'état

this.state.currentMode = mode;

// Mettre à jour l'indicateur de capacité

this.updateCapacityIndicator();

// Actions spécifiques selon le mode

this.handleModeSpecificActions(mode);

};

/\*\*

\* Actions spécifiques selon le mode d'import

\*/

window.ConfirmationApp.handleModeSpecificActions = function(mode) {

switch (mode) {

case 'manuel':

this.focusManualForm();

break;

case 'fichier':

this.initializeFileUpload();

break;

case 'massif':

this.initializeChunkingInterface();

break;

}

};

/\*\*

\* Initialiser les modules externes

\*/

window.ConfirmationApp.initializeModules = function() {

// Module gestionnaire d'adhérents

if (window.AdherentsManager) {

window.AdherentsManager.init();

console.log('✅ AdherentsManager initialisé');

}

// Module de chunking

if (window.ChunkingEngine) {

window.ChunkingEngine.init();

console.log('✅ ChunkingEngine initialisé');

}

// Module de validation

if (window.ValidationEngine) {

window.ValidationEngine.init();

console.log('✅ ValidationEngine initialisé');

}

// Module de upload de fichiers

if (window.FileUploadSGLP) {

window.FileUploadSGLP.init();

console.log('✅ FileUploadSGLP initialisé');

}

};

// ============================================================================

// GESTION DES STATISTIQUES

// ============================================================================

/\*\*

\* Mettre à jour les statistiques

\*/

window.ConfirmationApp.updateStatistics = function() {

const stats = this.calculateCurrentStats();

// Mettre à jour les cartes statistiques

this.updateStatCard('stat-existants', stats.existants, 'Existants');

this.updateStatCard('stat-minimum', stats.minimumRequis, 'Minimum Requis');

this.updateStatCard('stat-manquants', stats.manquants, stats.manquants > 0 ? 'Manquants' : 'Complet');

this.updateStatCard('stat-importes', stats.importes, 'Importés');

// Mettre à jour l'état de soumission

this.state.canSubmit = stats.manquants <= 0;

this.updateSubmissionButton();

console.log('📊 Statistiques mises à jour:', stats);

};

/\*\*

\* Calculer les statistiques actuelles

\*/

window.ConfirmationApp.calculateCurrentStats = function() {

const imported = this.state.adherentsData.length;

const existing = this.config.stats.existants;

const total = existing + imported;

const required = this.config.stats.minimumRequis;

const missing = Math.max(0, required - total);

return {

existants: existing,

minimumRequis: required,

manquants: missing,

importes: imported,

total: total,

peutSoumettre: missing <= 0

};

};

/\*\*

\* Mettre à jour une carte statistique

\*/

window.ConfirmationApp.updateStatCard = function(cardId, value, label) {

const card = document.getElementById(cardId);

if (!card) return;

const numberElement = card.querySelector('.stat-number');

const labelElement = card.querySelector('.stat-label');

if (numberElement) {

// Animation du changement de valeur

const currentValue = parseInt(numberElement.textContent) || 0;

if (currentValue !== value) {

this.animateNumberChange(numberElement, currentValue, value);

}

}

if (labelElement) {

labelElement.textContent = label;

}

// Mettre à jour les classes CSS selon la valeur

this.updateStatCardClasses(card, cardId, value);

};

/\*\*

\* Animer le changement de nombre

\*/

window.ConfirmationApp.animateNumberChange = function(element, from, to) {

const duration = 500;

const startTime = Date.now();

const animate = () => {

const elapsed = Date.now() - startTime;

const progress = Math.min(elapsed / duration, 1);

const current = Math.round(from + (to - from) \* progress);

element.textContent = current === 0 && to === 0 ? '✓' : current;

if (progress < 1) {

requestAnimationFrame(animate);

}

};

requestAnimationFrame(animate);

};

/\*\*

\* Mettre à jour les classes CSS des cartes statistiques

\*/

window.ConfirmationApp.updateStatCardClasses = function(card, cardId, value) {

// Supprimer les anciennes classes de statut

card.classList.remove('highlight', 'warning', 'danger', 'success', 'info');

// Ajouter la classe appropriée selon le contexte

switch (cardId) {

case 'stat-existants':

card.classList.add('info');

break;

case 'stat-minimum':

card.classList.add('warning');

break;

case 'stat-manquants':

card.classList.add(value > 0 ? 'danger' : 'success');

break;

case 'stat-importes':

card.classList.add(value > 0 ? 'highlight' : 'info');

break;

}

};

/\*\*

\* Mettre à jour l'indicateur de capacité

\*/

window.ConfirmationApp.updateCapacityIndicator = function() {

const stats = this.calculateCurrentStats();

const fillElement = document.getElementById('capacity-fill');

const modeElement = document.getElementById('capacity-mode');

if (fillElement) {

const percentage = Math.min(100, (stats.total / stats.minimumRequis) \* 100);

fillElement.style.width = percentage + '%';

}

if (modeElement) {

let mode = 'Standard';

let className = 'standard';

if (this.state.currentMode === 'massif' || stats.importes >= this.config.chunking.threshold) {

mode = 'Chunking';

className = 'chunking';

} else if (stats.importes >= 50) {

mode = 'Fichier';

className = 'fichier';

}

modeElement.textContent = mode;

modeElement.className = `capacity-mode ${className}`;

}

};

// ============================================================================

// GESTION DES IMPORT MODES

// ============================================================================

/\*\*

\* Initialiser le formulaire manuel

\*/

window.ConfirmationApp.focusManualForm = function() {

const firstInput = document.querySelector('#import-manuel input:first-of-type');

if (firstInput) {

setTimeout(() => firstInput.focus(), 100);

}

};

/\*\*

\* Initialiser l'upload de fichier

\*/

window.ConfirmationApp.initializeFileUpload = function() {

if (window.FileUploadSGLP) {

window.FileUploadSGLP.setupFileDropZone('#import-fichier');

}

};

/\*\*

\* Initialiser l'interface de chunking

\*/

window.ConfirmationApp.initializeChunkingInterface = function() {

if (window.ChunkingEngine) {

window.ChunkingEngine.setupMassiveUpload('#import-massif');

}

};

// ============================================================================

// GESTION DES ADHÉRENTS

// ============================================================================

/\*\*

\* Ajouter un adhérent

\*/

window.ConfirmationApp.addAdherent = function(adherentData) {

try {

// Validation des données

const validatedAdherent = this.validateAdherentData(adherentData);

// Vérifier les doublons

if (this.isDuplicate(validatedAdherent)) {

throw new Error('Un adhérent avec ce NIP existe déjà');

}

// Ajouter à la liste

validatedAdherent.id = Date.now();

validatedAdherent.source = this.state.currentMode;

validatedAdherent.dateAjout = new Date().toISOString();

this.state.adherentsData.push(validatedAdherent);

// Mettre à jour l'interface

this.updateStatistics();

// Notifier les autres modules

if (window.AdherentsManager) {

window.AdherentsManager.refreshTable();

}

this.showSuccess(`Adhérent ${validatedAdherent.prenom} ${validatedAdherent.nom} ajouté`);

return validatedAdherent;

} catch (error) {

console.error('❌ Erreur ajout adhérent:', error);

this.showError(error.message);

return null;

}

};

/\*\*

\* Ajouter plusieurs adhérents (import)

\*/

window.ConfirmationApp.addMultipleAdherents = function(adherentsArray) {

let successCount = 0;

let errorCount = 0;

const errors = [];

adherentsArray.forEach((adherentData, index) => {

try {

const result = this.addAdherent(adherentData);

if (result) {

successCount++;

} else {

errorCount++;

}

} catch (error) {

errorCount++;

errors.push(`Ligne ${index + 1}: ${error.message}`);

}

});

// Afficher le résumé

const message = `Import terminé: ${successCount} réussis, ${errorCount} erreurs`;

if (errorCount > 0) {

this.showWarning(message);

console.warn('Erreurs d\'import:', errors);

} else {

this.showSuccess(message);

}

return { successCount, errorCount, errors };

};

/\*\*

\* Valider les données d'un adhérent

\*/

window.ConfirmationApp.validateAdherentData = function(data) {

const required = ['nom', 'prenom', 'nip'];

const missing = required.filter(field => !data[field] || !data[field].trim());

if (missing.length > 0) {

throw new Error(`Champs obligatoires manquants: ${missing.join(', ')}`);

}

// Validation NIP format gabonais

if (!this.validateNIPFormat(data.nip)) {

throw new Error('Format NIP invalide (attendu: XX-QQQQ-YYYYMMDD)');

}

return {

civilite: data.civilite || 'M',

nom: data.nom.trim(),

prenom: data.prenom.trim(),

nip: data.nip.trim(),

telephone: data.telephone?.trim() || '',

profession: data.profession?.trim() || '',

statut: 'valide'

};

};

/\*\*

\* Vérifier si un adhérent est en doublon

\*/

window.ConfirmationApp.isDuplicate = function(adherentData) {

return this.state.adherentsData.some(existing =>

existing.nip === adherentData.nip

);

};

/\*\*

\* Valider le format NIP gabonais

\*/

window.ConfirmationApp.validateNIPFormat = function(nip) {

const nipRegex = /^[A-Z0-9]{2}-[0-9]{4}-[0-9]{8}$/;

return nipRegex.test(nip);

};

// ============================================================================

// GESTION DE LA SOUMISSION

// ============================================================================

/\*\*

\* Configurer les gestionnaires de soumission

\*/

window.ConfirmationApp.setupSubmissionHandlers = function() {

const submitBtn = document.getElementById('submit-final-btn');

if (submitBtn) {

submitBtn.addEventListener('click', () => this.submitToAdmin());

}

const validateBtn = document.querySelector('[onclick="validateAllData()"]');

if (validateBtn) {

validateBtn.addEventListener('click', () => this.validateAllData());

}

};

/\*\*

\* Mettre à jour le bouton de soumission

\*/

window.ConfirmationApp.updateSubmissionButton = function() {

const submitBtn = document.getElementById('submit-final-btn');

if (!submitBtn) return;

const canSubmit = this.state.canSubmit && this.state.validationPassed;

submitBtn.disabled = !canSubmit;

if (canSubmit) {

submitBtn.classList.remove('btn-secondary');

submitBtn.classList.add('btn-success');

submitBtn.innerHTML = '<i class="fas fa-paper-plane me-2"></i>Soumettre à l\'Administration';

} else {

submitBtn.classList.remove('btn-success');

submitBtn.classList.add('btn-secondary');

const stats = this.calculateCurrentStats();

if (stats.manquants > 0) {

submitBtn.innerHTML = `<i class="fas fa-exclamation-triangle me-2"></i>Manque ${stats.manquants} adhérent(s)`;

} else {

submitBtn.innerHTML = '<i class="fas fa-check-double me-2"></i>Validation requise';

}

}

};

/\*\*

\* Valider toutes les données

\*/

window.ConfirmationApp.validateAllData = function() {

try {

this.showInfo('Validation en cours...');

// Vérifier les adhérents minimum

const stats = this.calculateCurrentStats();

if (stats.manquants > 0) {

throw new Error(`Il manque ${stats.manquants} adhérent(s) pour atteindre le minimum requis`);

}

// Vérifier la qualité des données

const validationResults = this.validateAdherentsQuality();

if (validationResults.criticalErrors > 0) {

throw new Error(`${validationResults.criticalErrors} erreur(s) critique(s) détectée(s)`);

}

// Marquer comme validé

this.state.validationPassed = true;

this.updateSubmissionButton();

// Afficher le résumé

this.showValidationSummary(validationResults);

this.showSuccess('Validation réussie ! Vous pouvez maintenant soumettre le dossier.');

} catch (error) {

console.error('❌ Erreur validation:', error);

this.showError(error.message);

this.state.validationPassed = false;

this.updateSubmissionButton();

}

};

/\*\*

\* Valider la qualité des adhérents

\*/

window.ConfirmationApp.validateAdherentsQuality = function() {

let criticalErrors = 0;

let warnings = 0;

let duplicates = 0;

const nipCounts = {};

this.state.adherentsData.forEach(adherent => {

// Compter les NIPs

nipCounts[adherent.nip] = (nipCounts[adherent.nip] || 0) + 1;

// Vérifier les champs obligatoires

if (!adherent.nom || !adherent.prenom || !adherent.nip) {

criticalErrors++;

}

// Vérifier le format NIP

if (!this.validateNIPFormat(adherent.nip)) {

criticalErrors++;

}

// Vérifier les données optionnelles

if (!adherent.telephone && !adherent.profession) {

warnings++;

}

});

// Compter les doublons

Object.values(nipCounts).forEach(count => {

if (count > 1) duplicates++;

});

return {

criticalErrors,

warnings,

duplicates,

total: this.state.adherentsData.length

};

};

/\*\*

\* Afficher le résumé de validation

\*/

window.ConfirmationApp.showValidationSummary = function(results) {

const summaryElement = document.getElementById('submission-summary');

const contentElement = document.getElementById('summary-content');

if (!summaryElement || !contentElement) return;

const stats = this.calculateCurrentStats();

contentElement.innerHTML = `

<div class="row">

<div class="col-md-6">

<h6 class="text-success">✅ Données Validées</h6>

<ul class="list-unstyled">

<li><strong>Total adhérents:</strong> ${stats.total}</li>

<li><strong>Minimum requis:</strong> ${stats.minimumRequis}</li>

<li><strong>Statut:</strong> <span class="text-success">Conforme</span></li>

</ul>

</div>

<div class="col-md-6">

<h6 class="text-warning">⚠️ Points d'attention</h6>

<ul class="list-unstyled">

<li><strong>Avertissements:</strong> ${results.warnings}</li>

<li><strong>Doublons:</strong> ${results.duplicates}</li>

<li><strong>Erreurs critiques:</strong> ${results.criticalErrors}</li>

</ul>

</div>

</div>

<div class="text-center mt-3">

<p class="text-muted">

<i class="fas fa-info-circle me-1"></i>

La soumission sera définitive et irréversible

</p>

</div>

`;

summaryElement.style.display = 'block';

};

/\*\*

\* Soumettre à l'administration

\*/

window.ConfirmationApp.submitToAdmin = function() {

if (!this.state.canSubmit || !this.state.validationPassed) {

this.showError('Validation requise avant soumission');

return;

}

if (!confirm('Êtes-vous sûr de vouloir soumettre ce dossier à l\'administration ? Cette action est irréversible.')) {

return;

}

try {

this.showInfo('Soumission en cours...');

// Préparer les données

const submissionData = {

adherents: this.state.adherentsData,

stats: this.calculateCurrentStats(),

validation: {

passed: this.state.validationPassed,

timestamp: new Date().toISOString()

}

};

// Soumettre via le formulaire caché

this.submitForm(submissionData);

} catch (error) {

console.error('❌ Erreur soumission:', error);

this.showError('Erreur lors de la soumission');

}

};

/\*\*

\* Soumettre le formulaire

\*/

window.ConfirmationApp.submitForm = function(data) {

const form = document.getElementById('hidden-submission-form');

const dataInput = document.getElementById('adherents-data-input');

const validationInput = document.getElementById('validation-passed-input');

if (!form || !dataInput || !validationInput) {

throw new Error('Formulaire de soumission non trouvé');

}

dataInput.value = JSON.stringify(data);

validationInput.value = this.state.validationPassed ? '1' : '0';

form.submit();

};

// ============================================================================

// GESTION DE LA SAUVEGARDE

// ============================================================================

/\*\*

\* Configurer les gestionnaires de sauvegarde

\*/

window.ConfirmationApp.setupSaveHandlers = function() {

// Sauvegarde automatique toutes les 30 secondes

setInterval(() => {

if (this.state.adherentsData.length > 0) {

this.autoSave();

}

}, 30000);

// Sauvegarde manuelle

window.saveDraft = () => this.saveDraft();

};

/\*\*

\* Sauvegarde automatique

\*/

window.ConfirmationApp.autoSave = function() {

try {

const saveData = {

adherents: this.state.adherentsData,

timestamp: new Date().toISOString(),

mode: this.state.currentMode

};

localStorage.setItem(`confirmation\_draft\_${this.config.dossierId}`, JSON.stringify(saveData));

// Mettre à jour l'indicateur

this.updateSaveIndicator('auto-saved');

} catch (error) {

console.warn('⚠️ Erreur sauvegarde automatique:', error);

}

};

/\*\*

\* Sauvegarde manuelle

\*/

window.ConfirmationApp.saveDraft = function() {

try {

this.autoSave();

this.showSuccess('Brouillon sauvegardé avec succès');

this.updateSaveIndicator('manual-saved');

} catch (error) {

this.showError('Erreur lors de la sauvegarde');

}

};

/\*\*

\* Charger un brouillon

\*/

window.ConfirmationApp.loadDraft = function() {

try {

const saveData = localStorage.getItem(`confirmation\_draft\_${this.config.dossierId}`);

if (!saveData) return false;

const data = JSON.parse(saveData);

if (data.adherents && Array.isArray(data.adherents)) {

this.state.adherentsData = data.adherents;

this.updateStatistics();

if (window.AdherentsManager) {

window.AdherentsManager.refreshTable();

}

this.showInfo('Brouillon chargé avec succès');

return true;

}

} catch (error) {

console.warn('⚠️ Erreur chargement brouillon:', error);

}

return false;

};

/\*\*

\* Mettre à jour l'indicateur de sauvegarde

\*/

window.ConfirmationApp.updateSaveIndicator = function(status) {

// Cette fonction peut être implémentée pour afficher un indicateur visuel

console.log(`💾 Sauvegarde: ${status}`);

};

// ============================================================================

// GESTION DES ÉVÉNEMENTS SYSTÈME

// ============================================================================

/\*\*

\* Configurer le gestionnaire de fermeture de page

\*/

window.ConfirmationApp.setupUnloadHandler = function() {

window.addEventListener('beforeunload', (e) => {

if (this.state.adherentsData.length > 0 && !this.state.importInProgress) {

this.autoSave();

if (!this.state.validationPassed) {

e.preventDefault();

e.returnValue = 'Des modifications non sauvegardées seront perdues. Continuer ?';

return e.returnValue;

}

}

});

};

/\*\*

\* Actualisation automatique des statistiques

\*/

window.ConfirmationApp.setupStatsRefresh = function() {

// Actualiser les statistiques toutes les 10 secondes si import en cours

setInterval(() => {

if (this.state.importInProgress) {

this.updateStatistics();

}

// Mettre à jour l'activité

this.state.lastActivity = Date.now();

}, 10000);

};

/\*\*

\* Vérifier l'état initial au chargement

\*/

window.ConfirmationApp.checkInitialState = function() {

// Essayer de charger un brouillon

this.loadDraft();

// Vérifier si les conditions de soumission sont remplies

this.updateSubmissionButton();

};

// ============================================================================

// UTILITAIRES ET HELPERS

// ============================================================================

/\*\*

\* Initialiser les tooltips

\*/

window.ConfirmationApp.initializeTooltips = function() {

if (typeof bootstrap !== 'undefined' && bootstrap.Tooltip) {

const tooltipTriggerList = [].slice.call(document.querySelectorAll('[data-bs-toggle="tooltip"]'));

tooltipTriggerList.map(function (tooltipTriggerEl) {

return new bootstrap.Tooltip(tooltipTriggerEl);

});

}

};

/\*\*

\* Afficher une notification de succès

\*/

window.ConfirmationApp.showSuccess = function(message) {

this.showNotification(message, 'success');

};

/\*\*

\* Afficher une notification d'erreur

\*/

window.ConfirmationApp.showError = function(message) {

this.showNotification(message, 'error');

};

/\*\*

\* Afficher une notification d'avertissement

\*/

window.ConfirmationApp.showWarning = function(message) {

this.showNotification(message, 'warning');

};

/\*\*

\* Afficher une notification d'information

\*/

window.ConfirmationApp.showInfo = function(message) {

this.showNotification(message, 'info');

};

/\*\*

\* Afficher une notification

\*/

window.ConfirmationApp.showNotification = function(message, type = 'info') {

const alertClass = {

success: 'alert-success',

error: 'alert-danger',

warning: 'alert-warning',

info: 'alert-info'

}[type] || 'alert-info';

const alertDiv = document.createElement('div');

alertDiv.className = `alert ${alertClass} alert-dismissible fade show position-fixed`;

alertDiv.style.cssText = 'top: 20px; right: 20px; z-index: 9999; max-width: 400px;';

alertDiv.innerHTML = `

<i class="fas fa-${this.getNotificationIcon(type)} me-2"></i>

${message}

<button type="button" class="btn-close" data-bs-dismiss="alert"></button>

`;

document.body.appendChild(alertDiv);

// Auto-suppression après 5 secondes

setTimeout(() => {

if (alertDiv.parentNode) {

alertDiv.remove();

}

}, 5000);

};

/\*\*

\* Obtenir l'icône de notification

\*/

window.ConfirmationApp.getNotificationIcon = function(type) {

const icons = {

success: 'check-circle',

error: 'exclamation-triangle',

warning: 'exclamation-triangle',

info: 'info-circle'

};

return icons[type] || 'info-circle';

};

// ============================================================================

// FONCTIONS GLOBALES EXPOSÉES

// ============================================================================

// Exposer les fonctions principales pour compatibilité avec les templates

window.showHelp = function() {

const helpModal = new bootstrap.Modal(document.getElementById('helpModal'));

helpModal.show();

};

window.saveDraft = function() {

if (window.ConfirmationApp.saveDraft) {

window.ConfirmationApp.saveDraft();

}

};

window.validateAllData = function() {

if (window.ConfirmationApp.validateAllData) {

window.ConfirmationApp.validateAllData();

}

};

window.submitToAdmin = function() {

if (window.ConfirmationApp.submitToAdmin) {

window.ConfirmationApp.submitToAdmin();

}

};

// ============================================================================

// INITIALISATION AUTOMATIQUE

// ============================================================================

// Exposer le module globalement

window.ConfirmationApp = window.ConfirmationApp;

// Log de chargement

console.log('📋 ConfirmationApp v2.0 chargé - Prêt pour initialisation');

**Maintenant créons le module gestionnaire d'adhérents :**

/\*\*

\* ============================================================================

\* ADHERENTS-MANAGER.JS - GESTIONNAIRE D'ADHÉRENTS DYNAMIQUE

\* Version: 2.0 - Tableau interactif avec filtres et pagination

\* ============================================================================

\*

\* Module pour gérer l'affichage, la recherche, le filtrage et la pagination

\* des adhérents dans l'interface confirmation.blade.php

\*/

window.AdherentsManager = window.AdherentsManager || {};

// ============================================================================

// CONFIGURATION ET ÉTAT

// ============================================================================

/\*\*

\* Configuration du gestionnaire

\*/

window.AdherentsManager.config = {

tableId: 'adherents-table',

tbodyId: 'adherents-tbody',

paginationId: 'adherents-pagination',

emptyStateId: 'empty-state',

// Pagination

itemsPerPage: 20,

maxPaginationButtons: 5,

// Recherche

searchDelay: 300,

minSearchLength: 2,

// Colonnes du tableau

columns: [

{ key: 'select', label: '', sortable: false, width: '50px' },

{ key: 'nip', label: 'NIP', sortable: true, width: '140px' },

{ key: 'nom\_complet', label: 'Nom Complet', sortable: true, width: '200px' },

{ key: 'telephone', label: 'Téléphone', sortable: true, width: '120px' },

{ key: 'profession', label: 'Profession', sortable: true, width: '150px' },

{ key: 'source', label: 'Source', sortable: true, width: '100px' },

{ key: 'statut', label: 'Statut', sortable: true, width: '100px' },

{ key: 'actions', label: 'Actions', sortable: false, width: '120px' }

]

};

/\*\*

\* État du gestionnaire

\*/

window.AdherentsManager.state = {

currentPage: 1,

itemsPerPage: 20,

totalItems: 0,

totalPages: 0,

// Tri

sortColumn: 'nom\_complet',

sortDirection: 'asc',

// Filtres

searchQuery: '',

filters: {

statut: '',

source: '',

profession: ''

},

// Sélection

selectedItems: new Set(),

selectAll: false,

// Données

allAdherents: [],

filteredAdherents: [],

currentPageAdherents: [],

// UI

isLoading: false,

searchTimeout: null

};

// ============================================================================

// INITIALISATION

// ============================================================================

/\*\*

\* Initialiser le gestionnaire d'adhérents

\*/

window.AdherentsManager.init = function() {

console.log('🔧 Initialisation AdherentsManager v2.0');

try {

// Initialiser les événements

this.setupEventHandlers();

// Configurer la recherche

this.setupSearch();

// Configurer les filtres

this.setupFilters();

// Charger les données initiales

this.loadInitialData();

// Rafraîchir l'affichage

this.refreshTable();

console.log('✅ AdherentsManager initialisé avec succès');

} catch (error) {

console.error('❌ Erreur initialisation AdherentsManager:', error);

}

};

/\*\*

\* Charger les données initiales

\*/

window.AdherentsManager.loadInitialData = function() {

if (window.ConfirmationApp && window.ConfirmationApp.state.adherentsData) {

this.state.allAdherents = [...window.ConfirmationApp.state.adherentsData];

console.log(`📊 ${this.state.allAdherents.length} adhérents chargés`);

}

};

/\*\*

\* Configurer les gestionnaires d'événements

\*/

window.AdherentsManager.setupEventHandlers = function() {

// Gestionnaire de sélection globale

const selectAllCheckbox = document.getElementById('select-all');

if (selectAllCheckbox) {

selectAllCheckbox.addEventListener('change', (e) => {

this.handleSelectAll(e.target.checked);

});

}

// Gestionnaires de tri sur les en-têtes

this.setupSortHandlers();

// Gestionnaires de pagination

this.setupPaginationHandlers();

// Gestionnaires d'actions en lot

this.setupBulkActionHandlers();

};

/\*\*

\* Configurer la recherche

\*/

window.AdherentsManager.setupSearch = function() {

const searchInput = document.getElementById('adherents-search');

if (!searchInput) return;

searchInput.addEventListener('input', (e) => {

clearTimeout(this.state.searchTimeout);

this.state.searchTimeout = setTimeout(() => {

this.handleSearch(e.target.value);

}, this.config.searchDelay);

});

// Recherche instantanée sur Entrée

searchInput.addEventListener('keypress', (e) => {

if (e.key === 'Enter') {

clearTimeout(this.state.searchTimeout);

this.handleSearch(e.target.value);

}

});

};

/\*\*

\* Configurer les filtres

\*/

window.AdherentsManager.setupFilters = function() {

const filterInputs = ['filter-statut', 'filter-source'];

filterInputs.forEach(filterId => {

const filterElement = document.getElementById(filterId);

if (filterElement) {

filterElement.addEventListener('change', (e) => {

const filterKey = filterId.replace('filter-', '');

this.handleFilter(filterKey, e.target.value);

});

}

});

// Bouton effacer filtres

const clearFiltersBtn = document.querySelector('[onclick="clearFilters()"]');

if (clearFiltersBtn) {

clearFiltersBtn.addEventListener('click', () => this.clearAllFilters());

}

};

// ============================================================================

// GESTION DES DONNÉES

// ============================================================================

/\*\*

\* Rafraîchir le tableau

\*/

window.AdherentsManager.refreshTable = function() {

this.loadInitialData();

this.applyFiltersAndSearch();

this.applySorting();

this.calculatePagination();

this.renderTable();

this.renderPagination();

this.updateCounter();

};

/\*\*

\* Appliquer les filtres et la recherche

\*/

window.AdherentsManager.applyFiltersAndSearch = function() {

let filtered = [...this.state.allAdherents];

// Appliquer la recherche

if (this.state.searchQuery && this.state.searchQuery.length >= this.config.minSearchLength) {

const query = this.state.searchQuery.toLowerCase();

filtered = filtered.filter(adherent =>

adherent.nom.toLowerCase().includes(query) ||

adherent.prenom.toLowerCase().includes(query) ||

adherent.nip.toLowerCase().includes(query) ||

(adherent.telephone && adherent.telephone.includes(query)) ||

(adherent.profession && adherent.profession.toLowerCase().includes(query))

);

}

// Appliquer les filtres

Object.keys(this.state.filters).forEach(filterKey => {

const filterValue = this.state.filters[filterKey];

if (filterValue) {

filtered = filtered.filter(adherent => {

if (filterKey === 'statut') {

return adherent.statut === filterValue;

} else if (filterKey === 'source') {

return adherent.source === filterValue;

}

return true;

});

}

});

this.state.filteredAdherents = filtered;

this.state.totalItems = filtered.length;

};

/\*\*

\* Appliquer le tri

\*/

window.AdherentsManager.applySorting = function() {

if (!this.state.sortColumn) return;

this.state.filteredAdherents.sort((a, b) => {

let valueA, valueB;

switch (this.state.sortColumn) {

case 'nom\_complet':

valueA = `${a.prenom} ${a.nom}`.toLowerCase();

valueB = `${b.prenom} ${b.nom}`.toLowerCase();

break;

case 'nip':

valueA = a.nip;

valueB = b.nip;

break;

case 'telephone':

valueA = a.telephone || '';

valueB = b.telephone || '';

break;

case 'profession':

valueA = a.profession || '';

valueB = b.profession || '';

break;

case 'source':

valueA = a.source || '';

valueB = b.source || '';

break;

case 'statut':

valueA = a.statut || '';

valueB = b.statut || '';

break;

default:

return 0;

}

if (valueA < valueB) return this.state.sortDirection === 'asc' ? -1 : 1;

if (valueA > valueB) return this.state.sortDirection === 'asc' ? 1 : -1;

return 0;

});

};

/\*\*

\* Calculer la pagination

\*/

window.AdherentsManager.calculatePagination = function() {

this.state.totalPages = Math.ceil(this.state.totalItems / this.state.itemsPerPage);

// Vérifier que la page actuelle est valide

if (this.state.currentPage > this.state.totalPages) {

this.state.currentPage = Math.max(1, this.state.totalPages);

}

// Calculer les éléments de la page actuelle

const startIndex = (this.state.currentPage - 1) \* this.state.itemsPerPage;

const endIndex = startIndex + this.state.itemsPerPage;

this.state.currentPageAdherents = this.state.filteredAdherents.slice(startIndex, endIndex);

};

// ============================================================================

// RENDU DE L'INTERFACE

// ============================================================================

/\*\*

\* Rendre le tableau

\*/

window.AdherentsManager.renderTable = function() {

const tbody = document.getElementById(this.config.tbodyId);

const emptyState = document.getElementById(this.config.emptyStateId);

if (!tbody) return;

// Afficher l'état vide si nécessaire

if (this.state.currentPageAdherents.length === 0) {

tbody.innerHTML = '';

if (emptyState) {

emptyState.classList.add('show');

}

return;

}

// Masquer l'état vide

if (emptyState) {

emptyState.classList.remove('show');

}

// Générer les lignes du tableau

tbody.innerHTML = this.state.currentPageAdherents.map(adherent =>

this.renderAdherentRow(adherent)

).join('');

// Configurer les événements des nouvelles lignes

this.setupRowEventHandlers();

};

/\*\*

\* Rendre une ligne d'adhérent

\*/

window.AdherentsManager.renderAdherentRow = function(adherent) {

const isSelected = this.state.selectedItems.has(adherent.id);

const nomComplet = `${adherent.prenom} ${adherent.nom}`;

return `

<tr class="adherent-row ${isSelected ? 'table-active' : ''}" data-id="${adherent.id}">

<td>

<input type="checkbox" class="form-check-input row-checkbox"

${isSelected ? 'checked' : ''}

data-id="${adherent.id}">

</td>

<td>

<span class="adherent-nip">${adherent.nip}</span>

</td>

<td>

<div class="adherent-name">${nomComplet}</div>

<small class="text-muted">${adherent.civilite || 'M'}</small>

</td>

<td>

<span class="adherent-phone">${adherent.telephone || '-'}</span>

</td>

<td>

<span class="adherent-profession">${adherent.profession || '-'}</span>

</td>

<td>

<span class="adherent-source source-${adherent.source || 'manuel'}">${this.getSourceLabel(adherent.source)}</span>

</td>

<td>

<span class="adherent-status status-${adherent.statut || 'valide'}">${this.getStatusLabel(adherent.statut)}</span>

</td>

<td>

<div class="adherent-actions">

<button class="btn btn-sm btn-view" onclick="AdherentsManager.viewAdherent('${adherent.id}')" title="Voir">

<i class="fas fa-eye"></i>

</button>

<button class="btn btn-sm btn-edit" onclick="AdherentsManager.editAdherent('${adherent.id}')" title="Modifier">

<i class="fas fa-edit"></i>

</button>

<button class="btn btn-sm btn-delete" onclick="AdherentsManager.deleteAdherent('${adherent.id}')" title="Supprimer">

<i class="fas fa-trash"></i>

</button>

</div>

</td>

</tr>

`;

};

/\*\*

\* Rendre la pagination

\*/

window.AdherentsManager.renderPagination = function() {

const paginationContainer = document.getElementById(this.config.paginationId);

if (!paginationContainer) return;

if (this.state.totalPages <= 1) {

paginationContainer.innerHTML = '';

return;

}

const startItem = (this.state.currentPage - 1) \* this.state.itemsPerPage + 1;

const endItem = Math.min(startItem + this.state.itemsPerPage - 1, this.state.totalItems);

paginationContainer.innerHTML = `

<div class="pagination-info">

<span class="text-muted">

Affichage ${startItem}-${endItem} sur ${this.state.totalItems} adhérents

</span>

</div>

<div class="pagination-controls">

${this.generatePaginationButtons()}

</div>

<div class="page-size-selector">

<label for="page-size">Éléments par page:</label>

<select id="page-size" class="form-select form-select-sm">

<option value="10" ${this.state.itemsPerPage === 10 ? 'selected' : ''}>10</option>

<option value="20" ${this.state.itemsPerPage === 20 ? 'selected' : ''}>20</option>

<option value="50" ${this.state.itemsPerPage === 50 ? 'selected' : ''}>50</option>

<option value="100" ${this.state.itemsPerPage === 100 ? 'selected' : ''}>100</option>

</select>

</div>

`;

// Configurer les événements de pagination

this.setupPaginationEvents();

};

/\*\*

\* Générer les boutons de pagination

\*/

window.AdherentsManager.generatePaginationButtons = function() {

let buttons = '';

// Bouton précédent

buttons += `

<button class="btn btn-sm ${this.state.currentPage === 1 ? 'disabled' : ''}"

onclick="AdherentsManager.goToPage(${this.state.currentPage - 1})"

${this.state.currentPage === 1 ? 'disabled' : ''}>

<i class="fas fa-chevron-left"></i>

</button>

`;

// Boutons de pages

const startPage = Math.max(1, this.state.currentPage - Math.floor(this.config.maxPaginationButtons / 2));

const endPage = Math.min(this.state.totalPages, startPage + this.config.maxPaginationButtons - 1);

if (startPage > 1) {

buttons += `<button class="btn btn-sm" onclick="AdherentsManager.goToPage(1)">1</button>`;

if (startPage > 2) {

buttons += `<span class="pagination-ellipsis">...</span>`;

}

}

for (let page = startPage; page <= endPage; page++) {

buttons += `

<button class="btn btn-sm ${page === this.state.currentPage ? 'active' : ''}"

onclick="AdherentsManager.goToPage(${page})">

${page}

</button>

`;

}

if (endPage < this.state.totalPages) {

if (endPage < this.state.totalPages - 1) {

buttons += `<span class="pagination-ellipsis">...</span>`;

}

buttons += `<button class="btn btn-sm" onclick="AdherentsManager.goToPage(${this.state.totalPages})">${this.state.totalPages}</button>`;

}

// Bouton suivant

buttons += `

<button class="btn btn-sm ${this.state.currentPage === this.state.totalPages ? 'disabled' : ''}"

onclick="AdherentsManager.goToPage(${this.state.currentPage + 1})"

${this.state.currentPage === this.state.totalPages ? 'disabled' : ''}>

<i class="fas fa-chevron-right"></i>

</button>

`;

return buttons;

};

// ============================================================================

// GESTION DES ÉVÉNEMENTS

// ============================================================================

/\*\*

\* Configurer les gestionnaires de tri

\*/

window.AdherentsManager.setupSortHandlers = function() {

const table = document.getElementById(this.config.tableId);

if (!table) return;

const headers = table.querySelectorAll('thead th[data-sort]');

headers.forEach(header => {

header.addEventListener('click', () => {

const column = header.getAttribute('data-sort');

this.handleSort(column);

});

// Ajouter l'attribut aria-sort

header.setAttribute('role', 'columnheader');

header.setAttribute('tabindex', '0');

});

};

/\*\*

\* Configurer les événements des lignes

\*/

window.AdherentsManager.setupRowEventHandlers = function() {

// Gestionnaires de sélection

const checkboxes = document.querySelectorAll('.row-checkbox');

checkboxes.forEach(checkbox => {

checkbox.addEventListener('change', (e) => {

const adherentId = parseInt(e.target.getAttribute('data-id'));

this.handleRowSelection(adherentId, e.target.checked);

});

});

};

/\*\*

\* Configurer les événements de pagination

\*/

window.AdherentsManager.setupPaginationEvents = function() {

const pageSizeSelect = document.getElementById('page-size');

if (pageSizeSelect) {

pageSizeSelect.addEventListener('change', (e) => {

this.changePageSize(parseInt(e.target.value));

});

}

};

// ============================================================================

// GESTIONNAIRES D'ACTIONS

// ============================================================================

/\*\*

\* Gérer la recherche

\*/

window.AdherentsManager.handleSearch = function(query) {

this.state.searchQuery = query;

this.state.currentPage = 1;

this.applyFiltersAndSearch();

this.applySorting();

this.calculatePagination();

this.renderTable();

this.renderPagination();

this.updateCounter();

console.log(`🔍 Recherche: "${query}" (${this.state.totalItems} résultats)`);

};

/\*\*

\* Gérer un filtre

\*/

window.AdherentsManager.handleFilter = function(filterKey, value) {

this.state.filters[filterKey] = value;

this.state.currentPage = 1;

this.applyFiltersAndSearch();

this.applySorting();

this.calculatePagination();

this.renderTable();

this.renderPagination();

this.updateCounter();

console.log(`🔽 Filtre ${filterKey}: "${value}" (${this.state.totalItems} résultats)`);

};

/\*\*

\* Gérer le tri

\*/

window.AdherentsManager.handleSort = function(column) {

if (this.state.sortColumn === column) {

this.state.sortDirection = this.state.sortDirection === 'asc' ? 'desc' : 'asc';

} else {

this.state.sortColumn = column;

this.state.sortDirection = 'asc';

}

this.applySorting();

this.calculatePagination();

this.renderTable();

this.updateSortIndicators();

console.log(`📊 Tri: ${column} ${this.state.sortDirection}`);

};

/\*\*

\* Gérer la sélection de ligne

\*/

window.AdherentsManager.handleRowSelection = function(adherentId, isSelected) {

if (isSelected) {

this.state.selectedItems.add(adherentId);

} else {

this.state.selectedItems.delete(adherentId);

}

this.updateSelectAllCheckbox();

this.updateBulkActions();

};

/\*\*

\* Gérer la sélection globale

\*/

window.AdherentsManager.handleSelectAll = function(selectAll) {

this.state.selectAll = selectAll;

if (selectAll) {

this.state.currentPageAdherents.forEach(adherent => {

this.state.selectedItems.add(adherent.id);

});

} else {

this.state.currentPageAdherents.forEach(adherent => {

this.state.selectedItems.delete(adherent.id);

});

}

this.renderTable();

this.updateBulkActions();

};

/\*\*

\* Aller à une page

\*/

window.AdherentsManager.goToPage = function(page) {

if (page < 1 || page > this.state.totalPages) return;

this.state.currentPage = page;

this.calculatePagination();

this.renderTable();

this.renderPagination();

// Remettre à zéro la sélection de page

this.state.selectedItems.clear();

this.updateSelectAllCheckbox();

this.updateBulkActions();

};

/\*\*

\* Changer la taille de page

\*/

window.AdherentsManager.changePageSize = function(newSize) {

this.state.itemsPerPage = newSize;

this.state.currentPage = 1;

this.calculatePagination();

this.renderTable();

this.renderPagination();

};

// ============================================================================

// ACTIONS SUR LES ADHÉRENTS

// ============================================================================

/\*\*

\* Voir un adhérent

\*/

window.AdherentsManager.viewAdherent = function(adherentId) {

const adherent = this.findAdherentById(adherentId);

if (!adherent) return;

// Implémenter l'affichage du modal de détails

console.log('👁️ Voir adhérent:', adherent);

// TODO: Afficher modal avec détails complets

};

/\*\*

\* Modifier un adhérent

\*/

window.AdherentsManager.editAdherent = function(adherentId) {

const adherent = this.findAdherentById(adherentId);

if (!adherent) return;

console.log('✏️ Modifier adhérent:', adherent);

// TODO: Afficher modal d'édition

};

/\*\*

\* Supprimer un adhérent

\*/

window.AdherentsManager.deleteAdherent = function(adherentId) {

const adherent = this.findAdherentById(adherentId);

if (!adherent) return;

if (!confirm(`Supprimer l'adhérent ${adherent.prenom} ${adherent.nom} ?`)) {

return;

}

// Supprimer de la liste principale

if (window.ConfirmationApp) {

const index = window.ConfirmationApp.state.adherentsData.findIndex(a => a.id === adherentId);

if (index !== -1) {

window.ConfirmationApp.state.adherentsData.splice(index, 1);

window.ConfirmationApp.updateStatistics();

}

}

// Rafraîchir le tableau

this.refreshTable();

// Notification

if (window.ConfirmationApp) {

window.ConfirmationApp.showSuccess(`Adhérent ${adherent.prenom} ${adherent.nom} supprimé`);

}

console.log('🗑️ Adhérent supprimé:', adherent);

};

// ============================================================================

// ACTIONS EN LOT

// ============================================================================

/\*\*

\* Configurer les actions en lot

\*/

window.AdherentsManager.setupBulkActionHandlers = function() {

// Les handlers seront ajoutés dynamiquement quand des éléments sont sélectionnés

};

/\*\*

\* Mettre à jour les actions en lot

\*/

window.AdherentsManager.updateBulkActions = function() {

const selectedCount = this.state.selectedItems.size;

const bulkActionsContainer = document.querySelector('.bulk-actions');

if (!bulkActionsContainer) return;

if (selectedCount > 0) {

bulkActionsContainer.classList.add('show');

bulkActionsContainer.innerHTML = `

<div class="d-flex align-items-center justify-content-center gap-2">

<span class="fw-bold">${selectedCount} adhérent(s) sélectionné(s)</span>

<button class="btn btn-sm btn-outline-danger" onclick="AdherentsManager.deleteSelected()">

<i class="fas fa-trash me-1"></i>Supprimer

</button>

<button class="btn btn-sm btn-outline-secondary" onclick="AdherentsManager.exportSelected()">

<i class="fas fa-download me-1"></i>Exporter

</button>

<button class="btn btn-sm btn-outline-info" onclick="AdherentsManager.clearSelection()">

<i class="fas fa-times me-1"></i>Annuler

</button>

</div>

`;

} else {

bulkActionsContainer.classList.remove('show');

}

};

/\*\*

\* Supprimer les éléments sélectionnés

\*/

window.AdherentsManager.deleteSelected = function() {

const selectedCount = this.state.selectedItems.size;

if (!confirm(`Supprimer ${selectedCount} adhérent(s) sélectionné(s) ?`)) {

return;

}

// Supprimer de la liste principale

if (window.ConfirmationApp) {

window.ConfirmationApp.state.adherentsData = window.ConfirmationApp.state.adherentsData.filter(

adherent => !this.state.selectedItems.has(adherent.id)

);

window.ConfirmationApp.updateStatistics();

}

// Vider la sélection

this.state.selectedItems.clear();

// Rafraîchir

this.refreshTable();

this.updateBulkActions();

if (window.ConfirmationApp) {

window.ConfirmationApp.showSuccess(`${selectedCount} adhérent(s) supprimé(s)`);

}

};

/\*\*

\* Exporter les éléments sélectionnés

\*/

window.AdherentsManager.exportSelected = function() {

const selectedAdherents = this.state.allAdherents.filter(

adherent => this.state.selectedItems.has(adherent.id)

);

this.exportAdherents(selectedAdherents, 'adherents\_selection');

};

/\*\*

\* Vider la sélection

\*/

window.AdherentsManager.clearSelection = function() {

this.state.selectedItems.clear();

this.state.selectAll = false;

this.updateSelectAllCheckbox();

this.updateBulkActions();

this.renderTable();

};

// ============================================================================

// UTILITAIRES

// ============================================================================

/\*\*

\* Trouver un adhérent par ID

\*/

window.AdherentsManager.findAdherentById = function(adherentId) {

return this.state.allAdherents.find(adherent => adherent.id == adherentId);

};

/\*\*

\* Obtenir le label d'une source

\*/

window.AdherentsManager.getSourceLabel = function(source) {

const labels = {

'manuel': 'Saisie',

'fichier': 'Fichier',

'chunking': 'Import',

'massif': 'Massif'

};

return labels[source] || 'Inconnu';

};

/\*\*

\* Obtenir le label d'un statut

\*/

window.AdherentsManager.getStatusLabel = function(statut) {

const labels = {

'valide': 'Valide',

'anomalie': 'Anomalie',

'doublon': 'Doublon',

'pending': 'En attente'

};

return labels[statut] || 'Valide';

};

/\*\*

\* Mettre à jour le compteur

\*/

window.AdherentsManager.updateCounter = function() {

const counterElement = document.getElementById('adherents-total');

if (counterElement) {

counterElement.textContent = this.state.totalItems;

}

};

/\*\*

\* Mettre à jour la case à cocher "Tout sélectionner"

\*/

window.AdherentsManager.updateSelectAllCheckbox = function() {

const selectAllCheckbox = document.getElementById('select-all');

if (!selectAllCheckbox) return;

const currentPageIds = this.state.currentPageAdherents.map(a => a.id);

const selectedOnPage = currentPageIds.filter(id => this.state.selectedItems.has(id));

if (selectedOnPage.length === 0) {

selectAllCheckbox.checked = false;

selectAllCheckbox.indeterminate = false;

} else if (selectedOnPage.length === currentPageIds.length) {

selectAllCheckbox.checked = true;

selectAllCheckbox.indeterminate = false;

} else {

selectAllCheckbox.checked = false;

selectAllCheckbox.indeterminate = true;

}

};

/\*\*

\* Mettre à jour les indicateurs de tri

\*/

window.AdherentsManager.updateSortIndicators = function() {

const headers = document.querySelectorAll('thead th[data-sort]');

headers.forEach(header => {

const column = header.getAttribute('data-sort');

header.removeAttribute('aria-sort');

if (column === this.state.sortColumn) {

header.setAttribute('aria-sort', this.state.sortDirection === 'asc' ? 'ascending' : 'descending');

}

});

};

/\*\*

\* Effacer tous les filtres

\*/

window.AdherentsManager.clearAllFilters = function() {

// Effacer la recherche

const searchInput = document.getElementById('adherents-search');

if (searchInput) {

searchInput.value = '';

}

// Effacer les filtres

Object.keys(this.state.filters).forEach(key => {

this.state.filters[key] = '';

const filterElement = document.getElementById(`filter-${key}`);

if (filterElement) {

filterElement.value = '';

}

});

// Réinitialiser la recherche

this.state.searchQuery = '';

this.state.currentPage = 1;

// Rafraîchir

this.refreshTable();

if (window.ConfirmationApp) {

window.ConfirmationApp.showInfo('Filtres effacés');

}

};

/\*\*

\* Exporter les adhérents

\*/

window.AdherentsManager.exportAdherents = function(adherents, filename = 'adherents') {

// Exporter en CSV

const csvContent = this.generateCSV(adherents);

this.downloadFile(csvContent, `${filename}.csv`, 'text/csv');

};

/\*\*

\* Générer un CSV

\*/

window.AdherentsManager.generateCSV = function(adherents) {

const headers = ['Civilité', 'Nom', 'Prénom', 'NIP', 'Téléphone', 'Profession', 'Source', 'Statut'];

const rows = adherents.map(adherent => [

adherent.civilite || 'M',

adherent.nom,

adherent.prenom,

adherent.nip,

adherent.telephone || '',

adherent.profession || '',

this.getSourceLabel(adherent.source),

this.getStatusLabel(adherent.statut)

]);

return [headers, ...rows].map(row =>

row.map(cell => `"${cell}"`).join(',')

).join('\n');

};

/\*\*

\* Télécharger un fichier

\*/

window.AdherentsManager.downloadFile = function(content, filename, mimeType) {

const blob = new Blob([content], { type: mimeType });

const url = window.URL.createObjectURL(blob);

const link = document.createElement('a');

link.href = url;

link.download = filename;

document.body.appendChild(link);

link.click();

document.body.removeChild(link);

window.URL.revokeObjectURL(url);

};

// ============================================================================

// FONCTIONS GLOBALES EXPOSÉES

// ============================================================================

// Exposer les principales fonctions pour usage dans les templates

window.clearFilters = function() {

if (window.AdherentsManager.clearAllFilters) {

window.AdherentsManager.clearAllFilters();

}

};

window.exportAdherents = function(format) {

if (window.AdherentsManager.exportAdherents) {

const allAdherents = window.AdherentsManager.state.allAdherents;

window.AdherentsManager.exportAdherents(allAdherents, `adherents\_export`);

}

};

window.showAdherentsStats = function() {

if (window.AdherentsManager.state.allAdherents) {

const stats = {

total: window.AdherentsManager.state.allAdherents.length,

valides: window.AdherentsManager.state.allAdherents.filter(a => a.statut === 'valide').length,

anomalies: window.AdherentsManager.state.allAdherents.filter(a => a.statut === 'anomalie').length,

doublons: window.AdherentsManager.state.allAdherents.filter(a => a.statut === 'doublon').length

};

alert(`Statistiques des adhérents:\n\nTotal: ${stats.total}\nValides: ${stats.valides}\nAnomalies: ${stats.anomalies}\nDoublons: ${stats.doublons}`);

}

};

// Log de chargement

console.log('📊 AdherentsManager v2.0 chargé - Prêt pour initialisation');

**Créons maintenant les partials pour les formulaires :**

{{-- ============================================================================

ADHERENT-FORM-MANUAL.BLADE.PHP - FORMULAIRE SAISIE MANUELLE

Partial pour confirmation.blade.php - Section import manuel

============================================================================ --}}

<div class="manual-form">

<div class="form-header mb-4">

<h5 class="text-primary">

<i class="fas fa-keyboard me-2"></i>

Saisie Manuelle d'Adhérents

</h5>

<p class="text-muted mb-0">

Ajoutez vos adhérents un par un via ce formulaire.

Idéal pour de petits volumes ou des corrections ponctuelles.

</p>

</div>

<!-- Formulaire de saisie -->

<form id="manual-adherent-form" class="needs-validation" novalidate>

<div class="row">

<!-- Civilité -->

<div class="col-md-2

adherent-form-manual.blade.php

adherent-form-upload.blade.php

chunking-progress-modal.blade.php