

İSG Denetim Sistemi - Kullanıcı ve Grup Yönetimi Implementasyonu

✓ Tamamlanan Özellikler

1. Reusable UI Components (11 adet)

Tüm bileşenler `/frontend/src/components/ui/` dizininde:

- **Modal.tsx** - Özelleştirilebilir modal penceresi (sm, md, lg, xl boyutları)
- **Button.tsx** - Varyant ve loading durumları ile buton
- **Input.tsx** - Label, error, helper text destekli input
- **Select.tsx** - Özelleştirilebilir dropdown select
- **ConfirmDialog.tsx** - Onay dialog'u (silme işlemleri için)
- **LoadingSpinner.tsx** - Yükleme animasyonu
- **EmptyState.tsx** - Boş durum bileşeni
- **Table.tsx** - Tablo bileşenleri (Table, TableHead, TableBody, TableRow, TableHeader, TableCell)
- **Pagination.tsx** - Sayfalama bileşeni
- **Badge.tsx** - Durum göstergesi rozeti

2. Kullanıcılar Sayfası (/users)

Lokasyon: `/frontend/src/pages/UsersPage.tsx`

Tamamlanan 10 Özellik:

1. ✓ **Kullanıcı Listesi** - Tablo görünümü (username, email, tam ad, rol, grup, durum, işlemler)
2. ✓ **Arama/Filtreleme** - İsim, email, rol bazlı gerçek zamanlı arama
3. ✓ **Sayfalama** - Her sayfada 10 kullanıcı, dinamik sayfa sayısı
4. ✓ **Yeni Kullanıcı Ekleme** - Modal ile form (username, email, password, fullName, role, groupId)
5. ✓ **Kullanıcı Düzenleme** - Modal ile form (tüm alanlar düzenlenebilir, şifre hariç)
6. ✓ **Kullanıcı Silme** - Onay dialog'u ile güvenli silme
7. ✓ **Şifre Değiştirme** - Ayrı modal ile (sadece yeni şifre)
8. ✓ **Aktif/Pasif Durum** - Toggle ile hızlı değiştirme
9. ✓ **Rol Bazlı Yetkilendirme** - Sadece ADMIN kullanıcı işlem yapabilir
10. ✓ **Form Validasyonları** - Email formatı, şifre uzunluğu (min 6), zorunlu alanlar

API Entegrasyonu:

- GET `/api/users` (arama, filtreleme, sayfalama parametreleri ile)
- GET `/api/users/:id`
- POST `/api/users`
- PATCH `/api/users/:id`
- DELETE `/api/users/:id`
- PATCH `/api/users/:id/password`
- PATCH `/api/users/:id/status`

3. Gruplar Sayfası (/groups)

Lokasyon: `/frontend/src/pages/GroupsPage.tsx`

Tamamlanan 10 Özellik:

1. ☒ **Grup Listesi** - Tablo görünümü (grup adı, açıklama, üye sayısı, işlemler)
2. ☒ **Arama/Filtreleme** - Grup adı bazlı gerçek zamanlı arama
3. ☒ **Sayfalama** - Her sayfada 10 grup, dinamik sayfa sayısı
4. ☒ **Yeni Grup Ekleme** - Modal ile form (name, description)
5. ☒ **Grup Düzenleme** - Modal ile form
6. ☒ **Grup Silme** - Onay dialog'u ile güvenli silme
7. ☒ **Grup Üyelerini Görüntüleme** - İki yöntem:
 - Satırda genişletilebilir (chevron ile)
 - Ayrı modal pencere
8. ☒ **Gruba Kullanıcı Ekleme** - Dropdown ile kullanıcı seçimi
9. ☒ **Gruptan Kullanıcı Çıkarma** - Üye listesinde çıkar butonu ile
10. ☒ **Form Validasyonları** - Grup adı zorunlu (min 3 karakter)

API Entegrasyonu:

- GET /api/groups (arama, sayfalama parametreleri ile)
- GET /api/groups/:id (üye detayları dahil)
- POST /api/groups
- PATCH /api/groups/:id
- DELETE /api/groups/:id
- POST /api/groups/:id/members
- DELETE /api/groups/:id/members/:userId

4. Tasarım Özellikleri

Modern ve Temiz Arayüz:

- ☒ Tailwind CSS ile responsive tasarım
- ☒ Modal'lar için backdrop ve animasyonlar (fadeIn)
- ☒ Loading state'leri (spinner ile)
- ☒ Error handling ve toast bildirimleri (react-hot-toast)
- ☒ Boş state'ler (EmptyState component)
- ☒ Confirm dialog'ları (ConfirmDialog component)
- ☒ Lucide-react ikonları (Plus, Search, Edit2, Trash2, Key, UserCheck, UserX, vb.)
- ☒ Badge'ler ile durum göstergeleri
- ☒ Hover efektleri ve transition'lar

Responsive Design:

- ☒ Mobile-first yaklaşım
- ☒ Grid layout'lar (md:grid-cols-2 vb.)
- ☒ Overflow scroll desteği
- ☒ Flexible container'lar

5. Teknik Özellikler

React Best Practices:

- ☒ Hooks kullanımı (useState, useEffect)

- ☒ Zustand store entegrasyonu (useAuthStore)
- ☒ TypeScript type safety
- ☒ Reusable component'ler
- ☒ Custom CSS animations

API Client:

- ☒ Axios client yapılandırması
- ☒ Request interceptor (JWT token)
- ☒ Response interceptor (error handling, 401 redirect)
- ☒ Try-catch blokları
- ☒ Error toast bildirimleri

Type Definitions:

- ☒ User interface (groupId ve group relation eklendi)
- ☒ Group interface (members ve _count eklendi)
- ☒ PaginatedResponse generic tipi

Dosya Yapısı

```

frontend/
├── src/
│   ├── components/
│   │   └── ui/
│   │       ├── Modal.tsx
│   │       ├── Button.tsx
│   │       ├── Input.tsx
│   │       ├── Select.tsx
│   │       ├── ConfirmDialog.tsx
│   │       ├── LoadingSpinner.tsx
│   │       ├── EmptyState.tsx
│   │       ├── Table.tsx
│   │       ├── Pagination.tsx
│   │       └── Badge.tsx
│   ├── pages/
│   │   ├── UsersPage.tsx (☒ Tam özellikli)
│   │   └── GroupsPage.tsx (☒ Tam özellikli)
│   ├── types/
│   │   └── index.ts (☒ Güncellenmiş)
│   ├── api/
│   │   └── client.ts (☒ Mevcut)
│   ├── store/
│   │   └── authStore.ts (☒ Mevcut)
│   └── index.css (☒ Custom animations)

```

Özellik Durumu

Kullanıcılar Sayfası:  10/10

Gruplar Sayfası:  10/10

UI Components:  11/11

Toplam:  31/31

Çalıştırma Talimatları

Frontend Dev Server:

```
cd /home/ubuntu/isg_denetim_sistemi/frontend
npm run dev
```

Frontend: http://localhost:5173

Backend API:

```
cd /home/ubuntu/isg_denetim_sistemi/backend
npm run start:dev
```

Backend: http://localhost:3000/api

Test Kullanıcısı:

- Username: admin
- Password: Admin123!
- Role: ADMIN

Tasarım Kararları

1. **Modüler Yapı:** Tüm UI bileşenleri ayrı dosyalarda, tekrar kullanılabilir
2. **Type Safety:** TypeScript ile tam tip güvenliği
3. **Error Handling:** Tüm API çağrılarında try-catch ve toast bildirimleri
4. **Loading States:** Tüm async işlemlerde loading göstergeleri
5. **Confirmation Dialogs:** Kritik işlemler (silme) için onay dialog'ları
6. **Empty States:** Boş liste durumları için bilgilendirici mesajlar
7. **Responsive Design:** Tüm ekran boyutlarında çalışan responsive layout
8. **Accessibility:** Proper labels, aria attributes, keyboard navigation
9. **Performance:** Pagination ile büyük veri setlerinde performans
10. **UX:** Smooth animations, instant feedback, clear error messages

Notlar

- Tüm arayüz metinleri Türkçe
- Sadece ADMIN rolündeki kullanıcılar CRUD işlemleri yapabilir
- Şifre değiştirme ayrı endpoint ile yapılıyor

- Grup üyeleri expandable row ve modal ile görüntülenebiliyor
- Form validasyonları client-side yapılıyor
- Backend validasyonları toast ile gösteriliyor

Bağımlılıklar (Mevcut)

- react: ^18.2.0
- react-router-dom: ^6.21.1
- axios: ^1.6.5
- zustand: ^4.4.7
- react-hot-toast: ^2.4.1
- lucide-react: ^0.309.0
- clsx: ^2.1.0
- tailwindcss: ^3.4.1

Sonuç

Kullanıcı ve Grup yönetimi sayfaları **tam özellikli** olarak tamamlanmıştır. Tüm gereksinimler karşılanmış, modern ve kullanıcı dostu bir arayüz oluşturulmuştur.