

MEGA-PROMPT

Implementation Complete

Espace Utilisateur — SGG Digital

Prompt d'architecture et d'implementation Full-Stack

pour les interfaces de comptes utilisateurs

CAHIER DES CHARGES TECHNIQUE

Stack : React 18 + TypeScript 5.8 + Vite 5.4 + shadcn/ui + Supabase

15 roles | 15 modules | 10 interfaces a implementer

Fevrier 2026 — Version 1.0

TABLE DES MATIERES

PARTIE A LE PROMPT (a copier-coller directement)

- A.1** Role, Contexte et Objectif
- A.2** Instructions d'execution etape par etape
- A.3** Details techniques pre-remplis

PARTIE B SPECIFICATIONS DETAILLEES

- B.1** Les 10 interfaces a implementer
- B.2** Structure des fichiers a creer
- B.3** Types TypeScript a definir
- B.4** Routes et navigation
- B.5** Matrice des permissions par role

PARTIE C CONTRAINTES ET PATTERNS

- C.1** Patterns de code a respecter
- C.2** Composants UI existants a reutiliser
- C.3** Checklist qualite post-implementation

PARTIE A — LE MEGA-PROMPT (Copier-Coller)

Ce qui suit est le prompt complet, pre-rempli avec les details de votre projet SGG Digital. Copiez-collez l'integralite du texte encadre en vert dans votre conversation avec l'IA.

A.1 Role, Contexte et Objectif

Role : Tu es un Architecte Logiciel et Developpeur Senior Full-Stack expert en React 18, TypeScript 5.8, Vite, shadcn/ui, Radix UI, Tailwind CSS et Supabase. Tu maitrises parfaitement React Router v6, React Query, le pattern custom hooks, et les machines a etats pour workflows gouvernementaux.

Contexte : Je construis la plateforme SGG Digital pour le Secretariat General du Gouvernement de la Republique Gabonaise. C'est une SPA React/TypeScript avec :

- 15 roles utilisateurs (admin_sgg, directeur_sgg, sg_ministere, sgpr, premier_ministre, ministre, assemblee, senat, conseil_etat, cour_constitutionnelle, dgjo, citoyen, vice-president, president, professionnel-droit)
- 15 modules proteges avec controle d'accès par role
- Double systeme d'auth : Supabase (production) + mode Demo (sessionStorage)
- Permissions fines : W (Write) > V (Validate) > R (Read) > none
- Interface existante : sidebar 8 sections, 34 items navigation, ProtectedRoute

Objectif : Implemente l'integralite de l'Espace Utilisateur (Mon Compte) avec 10 interfaces fonctionnelles de bout en bout. Chaque interface doit etre prete pour la production : validation Zod, etats de chargement, gestion d'erreurs, toasts de notification, responsive design, et compatible avec le mode Demo existant.

A.2 Instructions d'exécution étape par étape

ETAPE 1 – TYPES ET SCHEMAS (Backend/Shared)

- * Définis les interfaces TypeScript pour : UserProfile, UserSettings, UserSession, LoginHistory, NotificationPreference, SecuritySettings, UserActivity, DeviceInfo
- * Crée les schémas Zod de validation pour chaque formulaire : profileSchema, passwordSchema, settingsSchema, notificationSchema
- * Respecte les types existants dans src/types/index.ts (User, UserRole, Session, AuditLog, AppRole, PermissionType)

ETAPE 2 – HOOKS PERSONNALISES (Logique métier)

- * useUserProfile() : CRUD profil via Supabase + mode Demo (sessionStorage)
- * useUserSettings() : préférences utilisateur (theme, langue, notifications)
- * useUserSecurity() : changement mot de passe, sessions actives, 2FA (TOTP)
- * useLoginHistory() : historique connexions depuis audit_logs
- * useUserActivity() : activités récentes (derniers rapports, validations)
- * Chaque hook DOIT gérer : isLoading, error, isDemo (mode demo detection)

ETAPE 3 – COMPOSANTS UI (10 interfaces)

- * Layout : ProfilLayout.tsx avec sidebar navigation interne (onglets verticaux)
- * Page 1 : MonProfil.tsx – avatar, nom, email, institution, rôle (read-only si demo)
- * Page 2 : EditerProfil.tsx – formulaire édition avec validation Zod et upload avatar
- * Page 3 : Sécurité.tsx – mot de passe, 2FA toggle, sessions actives, déconnexion
- * Page 4 : Notifications.tsx – préférences par canal (email, app, sms) et par module
- * Page 5 : HistoriqueConnexions.tsx – tableau des connexions avec IP, navigateur, date
- * Page 6 : ActivitéRécente.tsx – timeline d'activités (rapports soumis, validations)
- * Page 7 : MesAccès.tsx – vue des modules accessibles avec permission par module
- * Page 8 : Préférences.tsx – thème (clair/sombre), langue, timezone, format date
- * Page 9 : ExportDonnées.tsx – export des données personnelles (RGPD)
- * Page 10 : Aide.tsx – FAQ, contact support, documentation liens

ETAPE 4 – INTEGRATION SYSTEME

- * Ajoute les routes dans App.tsx sous /profil/* avec ProtectedRoute
- * Ajoute l'entrée 'Mon Compte' dans Sidebar.tsx (icone User, lien /profil)
- * Connecte au AuthContext existant (useAuth) ET au mode Demo (useDemoUser)
- * Ajoute un menu utilisateur dans le header (avatar + dropdown : Profil, Paramètres, Déconnexion)

ETAPE 5 – ANALYSE POST-IMPLEMENTATION (CRITIQUE)

- * Fais une revue de sécurité : données sensibles exposées ? XSS ? CSRF ?
- * Vérifie la compatibilité mode Demo : chaque page fonctionne sans Supabase
- * Identifie les optimisations : React.memo, useMemo, lazy loading
- * Confirme zero erreur TypeScript (npx tsc --noEmit)

A.3 Details techniques pre-remplis

Composant	Détail technique (à respecter strictement)
Framework Frontend	React 18.3 + TypeScript 5.8 + Vite 5.4
Composants UI	shadcn/ui + Radix UI primitives (Sheet, Dialog, Tabs, Accordion, Badge, Card, Table, Select, Switch, Avatar, DropdownMenu)
Styling	Tailwind CSS 4.x (classes utilitaires uniquement)
Routing	React Router v6 (BrowserRouter, Routes, Route, Navigate)
State Management	React Query (QueryClientProvider) + hooks personnalisés
Authentification	Supabase Auth (@supabase/supabase-js) + sessionStorage (demo)
Validation	Zod (@hookform/resolvers/zod + react-hook-form)
Icônes	lucide-react (Shield, User, Settings, Bell, Clock, Activity, Key, etc.)
Visualisation	Recharts (BarChart, PieChart, LineChart pour statistiques)
Notifications	Sonner (toast) + @/components/ui/toast
Base de données	Supabase (PostgreSQL) — tables : profiles, user_roles, sessions, audit_logs, role_permissions
Auth Context	src/contextes/AuthContext.tsx — useAuth() hook
Demo Hook	src/hooks/useDemoUser.ts — useDemoUser() hook
Theme	src/contextes/ThemeContext.tsx — useTheme() hook
Permission Hooks	useMatricePermissions(), usePTMPPermissions()
Route Guard	src/components/ProtectedRoute.tsx
Layout	src/components/layout/Layout.tsx + Sidebar.tsx

Tableau 1 : Stack technique complète du projet

CONTRAINTE ABSOLUE : Le code généré DOIT être compatible avec le mode Demo (sessionStorage, pas de Supabase). Chaque hook doit détecter si demoUser existe et retourner des données mock si oui.

FORMAT DE REPONSE : Utilise des blocs de code séparés pour chaque fichier. Commente la logique complexe. Si la réponse est trop longue, arrête-toi à un point logique et demande 'Continuer ?'.

PARTIE B — SPECIFICATIONS DETAILLEES

B.1 Les 10 interfaces à implementer

Interface 1 : Mon Profil (MonProfil.tsx) — *Vue lecture seule du profil*

Avatar, nom complet, email, telephone, role (badge couleur), institution, date inscription, dernière connexion, nombre connexions, statut vérification

Accès : Tous rôles

Interface 2 : Editer Profil (EditerProfil.tsx) — *Formulaire d'édition*

Champs éditables : nom, téléphone, avatar (upload). Non-editables : email, rôle. Validation Zod (nom min 3 caractères, tel format +241). Toast succès/erreur.

Accès : Tous sauf citoyen

Interface 3 : Sécurité (Securite.tsx) — *Gestion sécurité du compte*

Changer mot de passe (ancien + nouveau + confirmer, min 8 caractères). Toggle 2FA (TOTP). Liste sessions actives (IP, device, date). Bouton révoquer.

Accès : Tous rôles

Interface 4 : Notifications (Notifications.tsx) — *Préférences notifications*

Matrice : lignes = modules (GAR, PTM, Nominations...), colonnes = canaux (email, app, aucun). Switch toggle par cellule. Sauvegarde automatique.

Accès : Tous rôles

Interface 5 : Historique (HistoriqueConnexions.tsx) — *Journal des connexions*

Tableau paginé : date/heure, IP, navigateur, appareil, localisation estimée, statut (succès/échec). Filtre par période. Export CSV.

Accès : Tous rôles

Interface 6 : Activité Récente (ActiviteRecente.tsx) — *Timeline d'activités*

Timeline verticale : rapports soumis, validations effectuées, documents consultés. Icônes par type d'action. Filtre par module. 50 dernières actions.

Accès : Tous rôles

Interface 7 : Mes Accès (MesAcces.tsx) — *Vue des droits*

Grille de cards par module : nom, icône, permission (badge W/V/R/none). Détail au clic (permissions fines par bloc/action). Lien vers chaque module.

Accès : Tous rôles

Interface 8 : Préférences (Preferences.tsx) — *Paramètres personnels*

Thème (clair/sombre via useTheme), langue (FR par défaut), timezone, format date (JJ/MM/AAAA ou AAAA-MM-JJ), densité tableau (compact/normal).

Accès : Tous rôles

Interface 9 : Export Donnees (ExportDonnees.tsx) — Export RGPD

Bouton 'Exporter mes donnees' : genere JSON avec profil, roles, activites. Bouton 'Supprimer mon compte' (desactive en mode demo). Confirmation dialog.

Acces : Tous roles

Interface 10 : Aide & Support (Aide.tsx) — Centre d'aide

FAQ (Accordion, 10 questions). Contact support (formulaire). Liens documentation. Version application. Guide demarrage rapide par role.

Acces : Tous roles

B.2 Structure des fichiers à créer

Fichier	Description	Action
src/types/user-profile.ts	Types + schemas Zod	Nouveau
src/hooks/useUserProfile.ts	CRUD profil (Supabase + demo)	Nouveau
src/hooks/useUserSettings.ts	Préférences utilisateur	Nouveau
src/hooks/useUserSecurity.ts	Mot de passe, 2FA, sessions	Nouveau
src/hooks/useLoginHistory.ts	Historique connexions	Nouveau
src/hooks/useUserActivity.ts	Activités récentes	Nouveau
src/components/profil/ProfilLayout.tsx	Layout avec nav interne	Nouveau
src/components/profil/ProfilSidebar.tsx	Sidebar onglets profil	Nouveau
src/components/profil/AvatarUpload.tsx	Composant upload avatar	Nouveau
src/pages/profil/MonProfil.tsx	Vue profil (lecture)	Nouveau
src/pages/profil/EditerProfil.tsx	Formulaire édition	Nouveau
src/pages/profil/Securite.tsx	Sécurité du compte	Nouveau
src/pages/profil/Notifications.tsx	Préférences notifications	Nouveau
src/pages/profil/HistoriqueConnexions.tsx	Journal connexions	Nouveau
src/pages/profil/ActiviteRecente.tsx	Timeline activités	Nouveau
src/pages/profil/MesAccès.tsx	Vue permissions	Nouveau
src/pages/profil/Preferences.tsx	Paramètres perso	Nouveau
src/pages/profil/ExportDonnees.tsx	Export RGPD	Nouveau
src/pages/profil/Aide.tsx	Centre d'aide	Nouveau
src/App.tsx	Ajout 10 routes /profil/*	Modifier
src/components/layout/Sidebar.tsx	Ajout lien Mon Compte	Modifier
src/components/layout/UserMenu.tsx	Menu dropdown header	Nouveau

Tableau 2 : Les 22 fichiers à créer/modifier

B.3 Types TypeScript à définir

Fichier : src/types/user-profile.ts

```
interface UserProfile {
  id: string;
  email: string;
  fullName: string;
  phone?: string;
  avatarUrl?: string;
  role: AppRole;
  roleLabel: string;
  institution: string;
  institutionId?: string;
  isActive: boolean;
  isVerified: boolean;
  totpEnabled: boolean;
  lastLogin?: string;
  loginCount: number;
  createdAt: string;
  category: DemoCategory;
}

interface UserSession {
  id: string;
  ipAddress: string;
  userAgent: string;
  deviceFingerprint?: string;
  createdAt: string;
  expiresAt: string;
  isCurrent: boolean;
}

interface LoginHistoryEntry {
  id: string;
  date: string;
  ipAddress: string;
  browser: string;
  device: string;
  location?: string;
  success: boolean;
}

interface UserActivityEntry {
  id: string;
  action: string; // 'soumission' | 'validation' | 'consultation' | 'modification'
  module: string; // 'matriceReporting' | 'ptmptq' | 'nominations' | ...
  description: string;
  timestamp: string;
  metadata?: Record;
}

interface NotificationPreference {
  moduleKey: string;
  moduleLabel: string;
  email: boolean;
  inApp: boolean;
  sms: boolean;
}

interface UserPreferences {
```

```
theme: 'light' | 'dark' | 'system';
language: 'fr' | 'en';
timezone: string;
dateFormat: 'DD/MM/YYYY' | 'YYYY-MM-DD';
tableDensity: 'compact' | 'normal' | 'comfortable';
}
```

B.4 Routes et navigation

Route	Composant	Description	Acces	Module
/profil	MonProfil	Vue profil	Tous	—
/profil/editer	EditerProfil	Edition profil	Tous sauf citoyen	—
/profil/securite	Securite	Securite	Tous	—
/profil/notifications	Notifications	Notifications	Tous	—
/profil/historique	HistoriqueConnexions	Connexions	Tous	—
/profil/activite	ActiviteRecente	Activites	Tous	—
/profil/acces	MesAcces	Permissions	Tous	—
/profil/preferences	Preferences	Parametres	Tous	—
/profil/export	ExportDonnees	Export RGPD	Tous	—
/profil/aide	Aide	Aide	Tous	—

Tableau 3 : Les 10 routes à ajouter dans App.tsx

Navigation interne du ProfilLayout :

Icône (lucide)	Label	Lien	Description
User	Mon Profil	/profil	Vue d'ensemble
Edit	Editer	/profil/editer	Modifier les informations
Shield	Securite	/profil/securite	Mot de passe, 2FA
Bell	Notifications	/profil/notifications	Preferences alertes
Clock	Historique	/profil/historique	Journal connexions
Activity	Activite	/profil/activite	Actions recentes
Key	Mes Acces	/profil/acces	Droits et permissions
Settings	Preferences	/profil/preferences	Theme, langue
Download	Export	/profil/export	Donnees personnelles
HelpCircle	Aide	/profil/aide	Support et FAQ

Tableau 4 : Items de navigation du ProfilSidebar

B.5 Matrice des permissions par role

Cette matrice definit ce que chaque role peut faire dans l'Espace Utilisateur. Le hook useUserProfile doit respecter ces permissions.

Role	Editer profil	Changer MdP	Gerer 2FA	Notif. pref	Export donnees	Voir historique	Suppr. compte
admin_sgg	W	W	W	W	W	R (all)	W
sgg-directeur	W	W	W	W	R	R (own)	R
sgpr	W	W	W	W	R	R (own)	R
premier-ministre	R	R	R	W	R	R (own)	R
president	R	R	R	W	R	R (own)	R
ministre	W	W	W	W	R	R (own)	R
sg-ministere	W	W	W	W	R	R (own)	R
assemblee	R	R	R	W	R	R (own)	R
dgjo	W	W	W	W	R	R (own)	R
citoyen	R	none	R	W	R	R (own)	R

Tableau 5 : Permissions Espace Utilisateur par role

IMPORTANT : En mode Demo, les operations d'ecriture (editer profil, changer mot de passe, supprimer compte) doivent afficher un toast 'Mode Demo : modifications simulees' sans effectuer de vraie operation Supabase.

PARTIE C — CONTRAINTES ET PATTERNS

C.1 Patterns de code a respecter

Pattern 1 : Detection mode Demo

```
const { demoUser } = useDemoUser();
const isDemo = !!demoUser;

// Dans chaque hook :
if (isDemo) {
  return { data: MOCK DATA, isLoading: false, error: null };
}

// Sinon : appel Supabase reel
```

Pattern 2 : Hook avec etats

```
const [isLoading, setIsLoading] = useState(false);
const [error, setError] = useState(null);

const updateProfile = async (data: Partial<UserProfile>) => {
  setIsLoading(true); setError(null);
  try {
    if (isDemo) { toast('Mode Demo : modifications simulees'); return; }
    const { error } = await supabase.from('profiles').update(data)...;
    if (error) throw error;
    toast.success('Profil mis a jour');
  } catch (e) { setError(e.message); toast.error('Erreur'); }
  finally { setIsLoading(false); }
};
```

Pattern 3 : Page avec ProtectedRoute

```
// Dans App.tsx :
<Route path='/profil' element={<ProtectedRoute>
  <ProfilLayout />
</ProtectedRoute>
}>
<Route index element={<MonProfil />} />
<Route path='editer' element={<EditerProfil />} />
<Route path='securite' element={<Securite />} />
...
</Route>
```

C.2 Composants UI existants à reutiliser

Composant(s)	Import	Usage dans Espace Utilisateur
Card, CardHeader, CardContent	@/components/ui/card	Container principal
Tabs, TabsList, TabsTrigger	@/components/ui/tabs	Navigation onglets
Badge	@/components/ui/badge	Labels role, statut, permission
Avatar, AvatarImage, AvatarFallback	@/components/ui/avatar	Photo profil
Switch	@/components/ui/switch	Toggle notifications, 2FA
Dialog, DialogTrigger, DialogContent	@/components/ui/dialog	Confirmations
Table, TableHead, TableBody, TableRow	@/components/ui/table	Listes, historique
Select, SelectTrigger, SelectValue	@/components/ui/select	Dropdowns
Input, Label, Textarea	@/components/ui/input	Formulaires
Sheet, SheetContent, SheetTrigger	@/components/ui/sheet	Panneaux lateraux
Accordion, AccordionItem	@/components/ui/accordion	FAQ, details
Progress	@/components/ui/progress	Complétude profil
Separator	@/components/ui/separator	Divisions sections
DropdownMenu, DropdownMenuTrigger	@/components/ui/dropdown-menu	Menu utilisateur header
ScrollArea	@/components/ui/scroll-area	Listes longues
toast / Sonner	sonner	Notifications feedback

Tableau 6 : Composants shadcn/ui à reutiliser (déjà installés)

C.3 Checklist qualite post-implementation

Criterie	Verification	Priorite
TypeScript	npx tsc --noEmit retourne zero erreur	Bloquant
Build	npx vite build se termine sans erreur	Bloquant
Mode Demo	Toutes les pages fonctionnent en mode Demo (sessionStorage, pas de Supabase)	Bloquant
Responsive	Layout fonctionne sur mobile (375px), tablet (768px) et desktop (1280px)	Important
Etats chargement	Chaque bouton affiche isLoading pendant l'operation. Spinner visible, bouton desactive.	Important
Erreurs	Chaque formulaire gere les erreurs serveur et les affiche dans un toast ou inline.	Important
Validation Zod	Chaque formulaire valide cote client avant soumission. Messages d'erreur en francais.	Important
Accessibilite	aria-label sur les boutons icones. Focus trap dans les dialogs.	Recommande
Navigation	Sidebar interne highlight la page active. Breadcrumb optionnel.	Recommande
Securite	Pas d'info sensible dans le DOM. Mot de passe masque. Tokens non exposes.	Bloquant
Performance	Lazy loading des pages (React.lazy). Images avatar optimisees.	Recommande
Cohérence UI	Meme style que MatriceReporting, PTMMatrice : memes badges, memes cards, memes couleurs.	Important

Tableau 7 : Checklist qualite a verifier apres implementation

PRET A COPIER-COLLER : Utilisez la Partie A comme prompt direct.

Ce document constitue le cahier des charges technique complet pour l'implementation de l'Espace Utilisateur de SGG Digital. La Partie A est le prompt a copier-coller directement dans votre conversation IA. Les Parties B et C servent de reference technique pour valider le code genere.