

DOCUMENTATION DU PROJET DEFICLINIQUE

Interface conçu : Interface Utilisateur pour le personnel Médical

Fait par : AKAKPO SEFAKO GISELE

DESCRIPTION DU PROJET

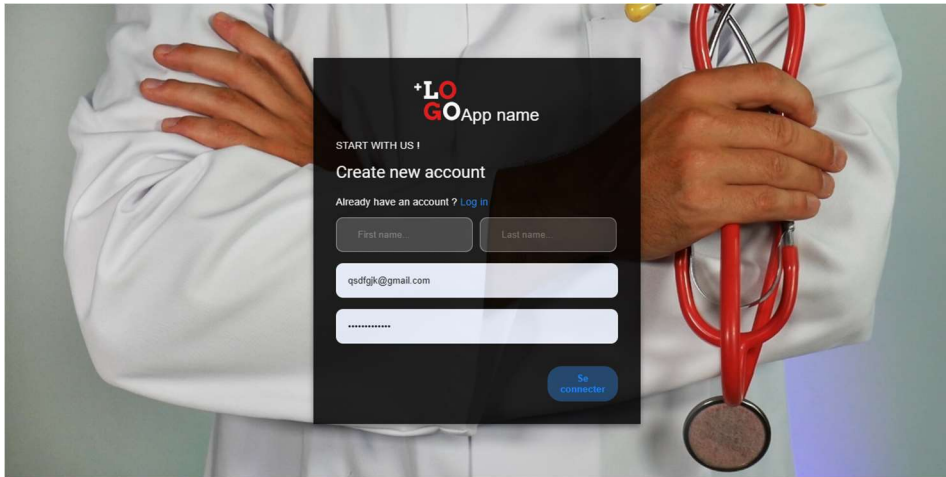
L'application **DéfiClinique** est mise en place pour faciliter la gestion des cliniques de la place .En effet la gestion d'une clinique n'est pas chose facile quand c'est à la main ; elle est beaucoup plus facile et rapide à gérer si la gestion est informatisée d'où la nécessité de notre logiciel

DOCUMENTATION NON TECHNIQUES DES INTERFACES

1-)Page de chargement



Sur cette interface nous avons une brève description de notre application ; après le chargement nous nous retrouvons directement sur la page de connexion



Ici notre utilisateur se connectera avec son nom , son prénom son email et le mot de passe.

Notons que c'est l'administrateur qui vous fournit l'adresse mail et le mot de passe avec lequel vous vous connectiez. C'est ainsi parce que dans notre application nous avons des interfaces spécifiques à chaque type d'utilisateurs (Administrateur, docteur, secrétaire ...)

Voici l'interface sur lequel l'administrateur crée les utilisateurs :

[← Home](#)

Menu

- Dashboard
- Doctors Ma...**
- Departements
- Settings

Doctors managements

search medical records...

Add Doctor

Nom	Prénom	Doc num
Spécialité	Mot de passe	Departement

Ajouter

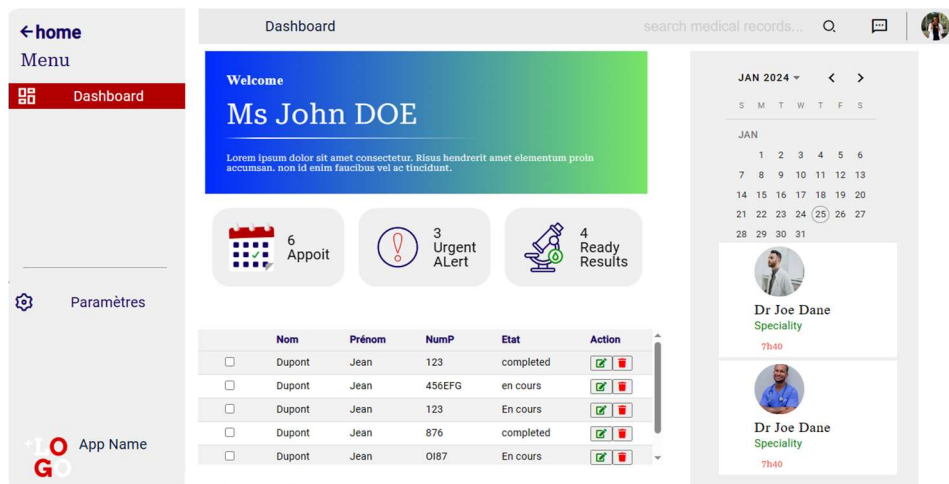
Personnel

	Nom	Prénom	Doc num	Spécialités	mot de passe	Departements
<input type="checkbox"/>	Doc	Doc	000000	Speciality	abcd00	
<input type="checkbox"/>	Doc	Doc	000000	Speciality	abcd00	
<input type="checkbox"/>	Doc	Doc	000000	Speciality	abcd00	
<input type="checkbox"/>	Doc	Doc	000000	Speciality	abcd00	
<input type="checkbox"/>	Doc	Doc	000000	Speciality	abcd00	
<input type="checkbox"/>	Doc	Doc	000000	Speciality	abcd00	

Modifier

Supprimer

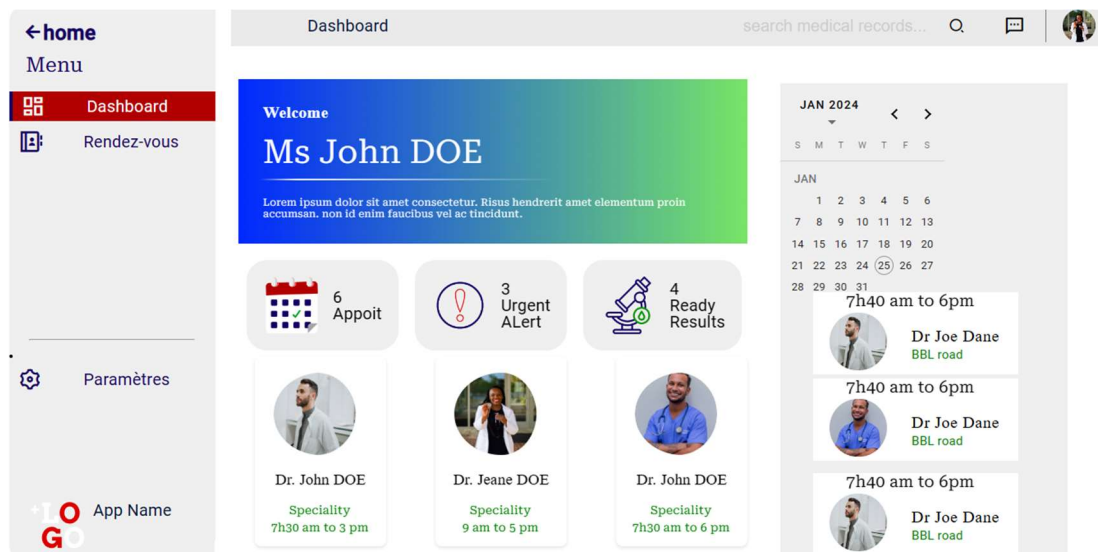
Après votre connexion si vous votre profile correspond à celui d'une secrétaire vous auriez accès à cette interface :



Sur cette interface vous avez un mot de bienvenue et aussi les docteurs de garde (Leurs noms, leurs spécialités et leurs de travail dans la journée)

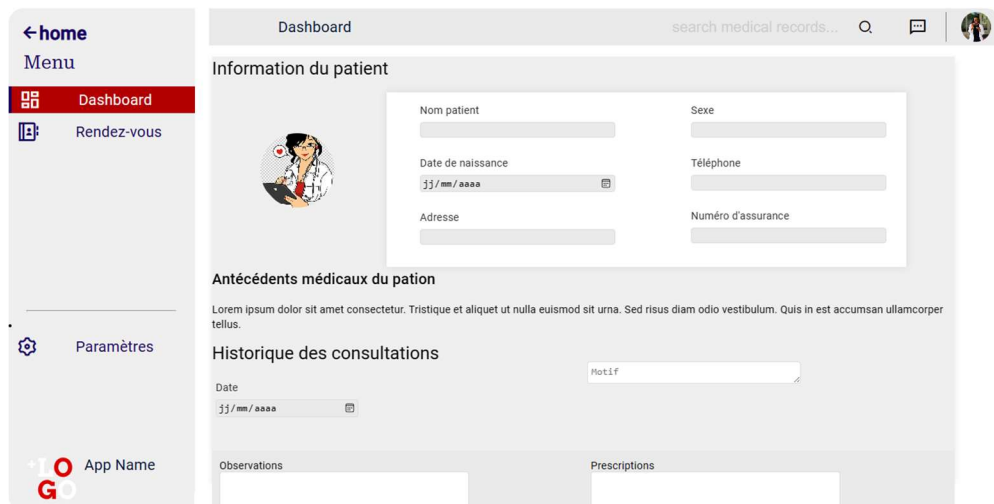
Vous avez aussi les rendez-vous de la journée, les états d'alertes se notifieront et aussi les résultats disponibles seront affichés. Les analyses des patients disponibles ou en cours sont également affichées dans un tableau.

Après votre connexion si votre profil correspond à celui d'un médecin, vous auriez accès à cette interface :



Sur cette interface vous avez un mot de bienvenue et aussi les docteurs de garde (Leurs noms, leurs spécialités et leurs de travail dans la journée)

Vous avez aussi les rendez-vous de la journée, les états d'alertes se notifieront et aussi les résultats disponibles seront affichés.



Ici c'est l'interface de consultation , le médecin entre le nom du patient, sa date de naissance, le sexe, le numéro de téléphone , l'adresse et le numéro d'assurance.

Le médecin voit les antécédents de son patient , il a aussi l'historique des consultation de son patient .Le docteur peut également mentionner ses observations et aussi faire des prescription nécessaire à son patient

DOCUMENTATION TECHNIQUES DES INTERFACES

FRAMEWORK UTILISE : Angular

LES IMPORTATIONS DES MODULES ET COMPOSANTS

Routes de @angular/router : Un tableau de routes Angular, utilisé pour configurer les chemins de navigation dans l'application.

LoginComponent, AccueilComponent, HeaderComponent, DashbordComponent, ConsultationComponent, PrincipalComponent, DocComponent, SecretaireComponent : Ce sont des composants Angular importés depuis leurs fichiers respectifs. Chaque composant représente une vue ou une page dans l'application.

CONFIGURATIONS DES ROUTES

- **routes: Routes** : Une constante **routes** est déclarée, qui est un tableau de type **Routes**. Chaque élément du tableau représente une route spécifique dans l'application.
- Exemple: { **path: 'accueil', component: AccueilComponent** } - cette route associe le chemin 'accueil' au composant **AccueilComponent**. Lorsque l'utilisateur navigue vers /accueil, le **AccueilComponent** est rendu.
- Routes avec enfants : La route {**path: 'principal', component: PrincipalComponent, children: [...]**} définit une route parent avec des routes enfants. Cela signifie que **PrincipalComponent** aura des sous-routes comme **consultation** et **dashbord**.

```

import { Routes } from '@angular/router';
import { LoginComponent } from '../login/login.component';
import { AccueilComponent } from '../accueil/accueil.component';
import { HeaderComponent } from '../header/header.component';
import { DashbordComponent } from '../dashbord/dashbord.component';
import { ConsultationComponent } from '../consultation/consultation.component';
import path from 'path';
import { PrincipalComponent } from '../principal/principal.component';
import { DocComponent } from '../doc/doc.component';
import { SecretaireComponent } from '../secretaire/secretaire.component';

export const routes: Routes = [
  { path: 'accueil', component: AccueilComponent },
  { path: 'principal', component: PrincipalComponent, children: [
    { path: 'consultation', component: ConsultationComponent },
    { path: 'dashbord', component: DashbordComponent }
  ] },
  { path: 'login', component: LoginComponent },
  { path: 'header', component: HeaderComponent },
  { path: 'consultation', component: ConsultationComponent },
  { path: 'dashbord', component: DashbordComponent },
  { path: 'doc', component: DocComponent },
  { path: 'secretaire', component: SecretaireComponent, children: [
    { path: 'doc', component: DocComponent }
  ] }

  // { path: '', redirectTo: '/accueil', pathMatch: 'full' },
];

export class AppComponent {}

```

Détail des Routes Individuelles:

- **/accueil** : Affiche le **AccueilComponent**.
- **/principal** : Affiche le **PrincipalComponent** et peut avoir des sous-chemins :
 - **/principal/consultation** : Affiche le **ConsultationComponent**.
 - **/principal/dashbord** : Affiche le **DashbordComponent**.
- **/login** : Affiche le **LoginComponent**.
- **/header** : Affiche le **HeaderComponent**.
- **/consultation** : Affiche le **ConsultationComponent**.
- **/dashbord** : Affiche le **DashbordComponent**.
- **/doc** : Affiche le **DocComponent**.

- **/secretaire** : Affiche le **SecretaireComponent** et peut avoir un sous-chemin **/secretaire/doc** qui affiche **DocComponent**.