****

**Projet Fil Rouge**

**2ème année : Full Stack Java / Angular**

**Réalisation d’une application web pour la Prise et la Gestion d’un rendez-vous médical**



**Réalisé par : Encadré par :**

Loubna SOUSSI Mr NACIRI TAOUFIK Mohamed

**Année universitaire :** 2021-202

Table des matières :

* Remerciements…………………………………………………………………..3
* Introduction……………………………………………………………………...4

1. Cahier des charges……………………………………………………………….....5
   1. Description
   2. Objectif
   3. Problématique
   4. Solution
2. Fonctionnalité de l’application ……………………………………………………..6
3. Conception du projet en UML………………………………………………………9
   1. Diagramme de cas d’utilisation
   2. Diagramme de classe
4. Réalisation …………………………………………………………………………15
   1. Outils de développement
      1. PowerAMC
      2. Angular
      3. SpringBoot
      4. PostgreSQL
      5. Tailwind
   2. Logiciels utilisés
   3. Présentation de la solution réalisée ………………………………………..17

Conclusion …………………………………………………………………………31

Remerciements :

Je tiens à remercier dans un premier temps, toute l’équipe pédagogique de Youcode Safi et les intervenants professionnels responsables de la formation.

Je remercie également mon formateur, Mr Kamal et Mr Toufik pour leurs sens d’écoute et leurs conseils qui mon permis d’élaborer ce projet.

Mes vifs remerciements vont aussi à Monsieur Abdelaziz et Madame Siham qui m’ont encouragée toute au long de cette formation, et qui m’ont accordée toute leur confiance afin que je puisse réussir, je leur remercie aussi pour le temps qui m’ont consacré sachant répondre à toute mes interrogations

Pour tous ceux et celles qui ont contribué de près ou de loin à l’accomplissement de ce travail trouvent l’expression de mes remerciements les plus chaleureux.

**Introduction générale :**

Dans le cadre de la formation à YouCode, nous étions amenés à réaliser un projet, mon choix s’est porté sur la réalisation d’une plateforme de Gestion des Rendez-vous « AjiDoc », c’est une plateforme qui facilite aux clients la prise des rendez vous en ligne sans avoir à se déplacer au cabinet.

Ce rapport est composé de trois chapitres :

- La première partie contient la présentation du cahier de charges.

- La deuxième partie est destiné à l’analyse et la conception de notre application

- La troisième partie nous est consacré aux outils utilisés pour la réalisation de ce projet, ainsi que des captures d’écran de l’application avec description.

1. **Cahier des charges :**
   1. Description :

Dans le cadre de notre formation en 2ème année à Youcode filière « Full Stack Java/Angular », il nous a été demandé de travailler sur un projet de fin d’étude, Mon choix, c'est porté sur la réalisation d’une plateforme pour la prise et la gestion des rendez-vous des cabinets médicaux.

* 1. Objectif :

L’objectif principal de notre plateforme, c'est de faciliter le processus de prise de rendez-vous pour le patient ainsi que pour le médecin.

 Concernant le patient, il n'a plus besoin de se déplacer jusqu’au cabinet pour prendre un rendez-vous, il peut maintenant le prendre en ligne.

Pour le médecin, la plateforme l’aide à gérer ces rendez-vous facilement.

* 1. Problématique :
* Le médecin a du mal à gérer les rendez-vous de son cabinet.
* Problème du déplacement pour le patient afin de prendre un rendez-vous.
* Problème au niveau de l’heure du rendez-vous
  1. Solution :
* Notre plateforme propose un système qui va aider le médecin à gérer ces rendez-vous d’une manière simple et efficace.
* Plus besoin pour le patient de se déplacer jusqu’au cabinet pour prendre un rendez-vous, notre plateforme aide les patients à prendre leur rendez-vous facilement (UI, UX) en quelques minutes.
* En prenant un rendez-vous dans notre plateforme, vous aurez votre heure exacte pour faire votre consultation.
  1. Cible :

La plateforme cible deux catégories de personnes :

- **Patient :** C’est une personne qui va bénéficier de la plateforme, car il pourra prendre un rendez-vous facilement et même effectué un paiement en ligne sans avoir à se déplacer au cabinet.

- **Docteur :** C’est une personne qui a besoin de notre plateforme pour l’aider à faciliter la gestion des rendez-vous de son cabinet.

1. **Fonctionnalité de l’application :**
   1. Les besoins fonctionnels :

* **ADMIN :**  C’est la personne qui va prendre en charge de gérer toutes les actions du site. L’admin doit faire la gestion des docteurs ça signifie que lors de l’inscription d’un docteur, c'est l’admin qui va valider l’inscription, la validation se fait après une vérification des données professionnelles du docteur (les diplômes ..)
* **Visiteur du site** : Un visiteur c’est une personne qui va visiter notre plateforme pour prendre un rendez pour faire une consultation chez un docteur précis, il peut voir la liste des docteurs qui existent sur notre BD ainsi que les cabinets disponibles, le visiteur peut faire la recherche (filtrage ) soit par un docteur ou bien par le nom du cabinet  ou par une maladie précis
* **Patient du site :** Afin de devenir patient du site, le visiteur doit faire obligatoirement une inscription en entrant toutes ces informations personnelles.
* Maintenant qu’il est devenu patient du site, le patient doit faire une l’authentification pour accéder à la plateforme.
* Il pourra de plus que faire la consultation du site et le filtrage, il peut prendre un rendez-vous, il a le choix entre un simple rendez-vous et un rendez vous automatisé.

 \*\*) Simple Rendez-vous :  Dans ce type de rendez-vous le patient doit remplir les informations nécessaire pour prendre un rendez-vous , il doit  attendre la confirmation de son rendez-vous de la part de la secrétaire ou bien du médecin ,

Après la confirmation de son rendez-vous le patient est censé payé le rendez-vous pour le valider, le paiement se fait soit sur place dans le cabinet ou bien en ligne

**🡺 La scène :**

* 1. **Patient :** Prendre un rendez-vous

**b) Secrétaire ou médecin :** Acceptation du rendez-vous (Un rendez-vous accepté doit impérativement être validé 'payé' pour avoir lieu )

**c - Patient :** Paiement en ligne ou bien sur place ( cabinet ) 'validation de rendez-vous '

 Paiement :

* **En ligne** : validation systématique
* **Sur place** : validation par secrétaire ou médecin

    \*\*) Rendez-vous automatisé : Ce type de rendez-vous est géré par le médecin , ce dernier doit d’abord définir si ce type de rendez-vous  est  disponible ou pas, dans le cas où le rendez-vous automatisé est disponible le médecin doit préciser la période  d'acceptation de ce type .

Ce type contient un mode de paiement en ligne

Par exemple :  un patient veut prendre un rendez-vous pour faire une consultation chez un médecin après 15 jours, une fois le paiement effectué , le système envoie une facture de paiement dans l’email du patient .

**🡺 La scène :**

**a- Patient :** Prendre un rendez-vous

**b- Système :**

-  Acceptation du rendez-vous directement après le paiement

-  Système va envoyer une facture de paiement dans l’email au patient

Le patient du site peut aussi laisser un commentaire dans la page du médecin pour donner son feedback sur le passage de la consultation. Il peut aussi faire un like ou bien un dislike sur la page d’un médecin et aussi le patient a le droit de partager le profil du médecin ça veut dire les informations (contact, adresse du cabinet , numéro de téléphone) avec ses amis soit sur WhatsApp ou bien Facebook.

* **Docteur :** Le médecin doit faire une inscription du médecin en entrant ses informations personnelles et professionnelles (diplôme, spécialité ..) , cette inscription va être validée par l’administrateur du site  pour être acceptée , pour accéder à son Dashboard, le médecin doit faire une authentification . Le médecin doit effectuer plusieurs tâches à savoir , l’ajout de son cabinet médical et aussi il peut  faire la gestion de profil, ça signifie que le médecin peut modifier des propres informations à tout moment , sauf que la modification est autorisée juste au niveau des informations personnelles , le médecin peut aussi faire la gestion des rendez-vous en acceptant un rendez-vous du patient et en le validant après son paiement , le médecin doit s’occuper de la gestion d’une secrétaire en ajoutant une secrétaire à son cabinet et doit aussi faire la gestion des permissions en précisant les permissions qu’une secrétaire pourra avoir dans son poste.
* **Secrétaire :**    La secrétaire doit faire une authentification pour accéder à la plateforme, elle a comme tâche de faire la gestion des rendez-vous des clients, cette gestion se fait en acceptant un rendez-vous et en le validant après son paiement
  1. Les besoins non fonctionnels :

Les besoins non fonctionnels sont importants, car ils agissent de façon indirecte sur le résultat et sur le rendement de l’utilisateur, ce qui fait qu’ils ne doivent pas être négligés, pour cela il faut répondre aux exigences suivantes :

▪ **Fiabilité :** Notre application doit fonctionner de façon cohérente, sans erreurs et doit être satisfaisante.

▪ **Les erreurs :** Les ambiguïtés doivent être signalées par des messages d’erreurs bien organisés pour bien guider l’utilisateur et le familiariser avec notre plateforme.

**▪ Ergonomie et bonne Interface :** L’application doit être adaptée à l’utilisateur sans qu’il ne fournisse aucun effort (utilisation claire et facile) de point de vue navigation entre les différentes pages, couleurs et mise en textes utilisés.

**▪ Sécurité :** Notre solution doit respecter surtout la confidentialité des données personnelles des clients qui reste l’une des contraintes les plus importantes dans les sites web.

**▪ Aptitude à la maintenance et la réutilisation :** Le système doit être conforme à une architecture standard et claire permettant sa maintenance et sa réutilisation.

**▪ Compatibilité et portabilité :** Un site web quel que soit son domaine, son éditeur et son langage de programmation ne peut être fiable qu’avec une compatibilité avec tous les navigateurs web et tous les moyens que ce soit PC, iPad ou Mobiles.

1. **Conception du projet en UML :**
   1. Diagramme de cas d’utilisation :

L'objectif d'un diagramme de cas d'utilisation est de représenter les différentes façons dont un utilisateur peut interagir avec un système :

1. **Diagramme de cas d’utilisation du patient :**

Un visiteur va consulter notre plateforme pour prendre un rendez-vous chez un médecin , il peut donc voir la lite des docteurs existante aussi la liste des cabinets , il peut même faire une recherche par ville , docteur et aussi cabinet médical .

Pour prendre un rendez-vous il doit faire une inscription en entrant ces informations personnelles afin de devenir patient du site .

Maintenant qu’il est patient il peut faire une authentification avec son email et password afin de prendre un rendez-vous , il existe deux types de rendez vous , rendez vous simple et rendez vous automatisé , il peut même liker ou disliker un profil d’un médecin et même laisser un commentaire sur un médecin à condition qu’il doit en moins avoir passer une consultation chez ce médecin , il peut faire la gestion de son profil a savoir voir la liste des rendez-vous effectué et enfin effectué un paiement .

Pour le paiement il a deux types, un paiement sur place , ça signifie qu’il doit se déplacer au cabinet pour effectué le paiement et valider son rendez vous , sinon il peut effectué un paiement a travers la plateforme et valider son rendez-vous automatiquement .

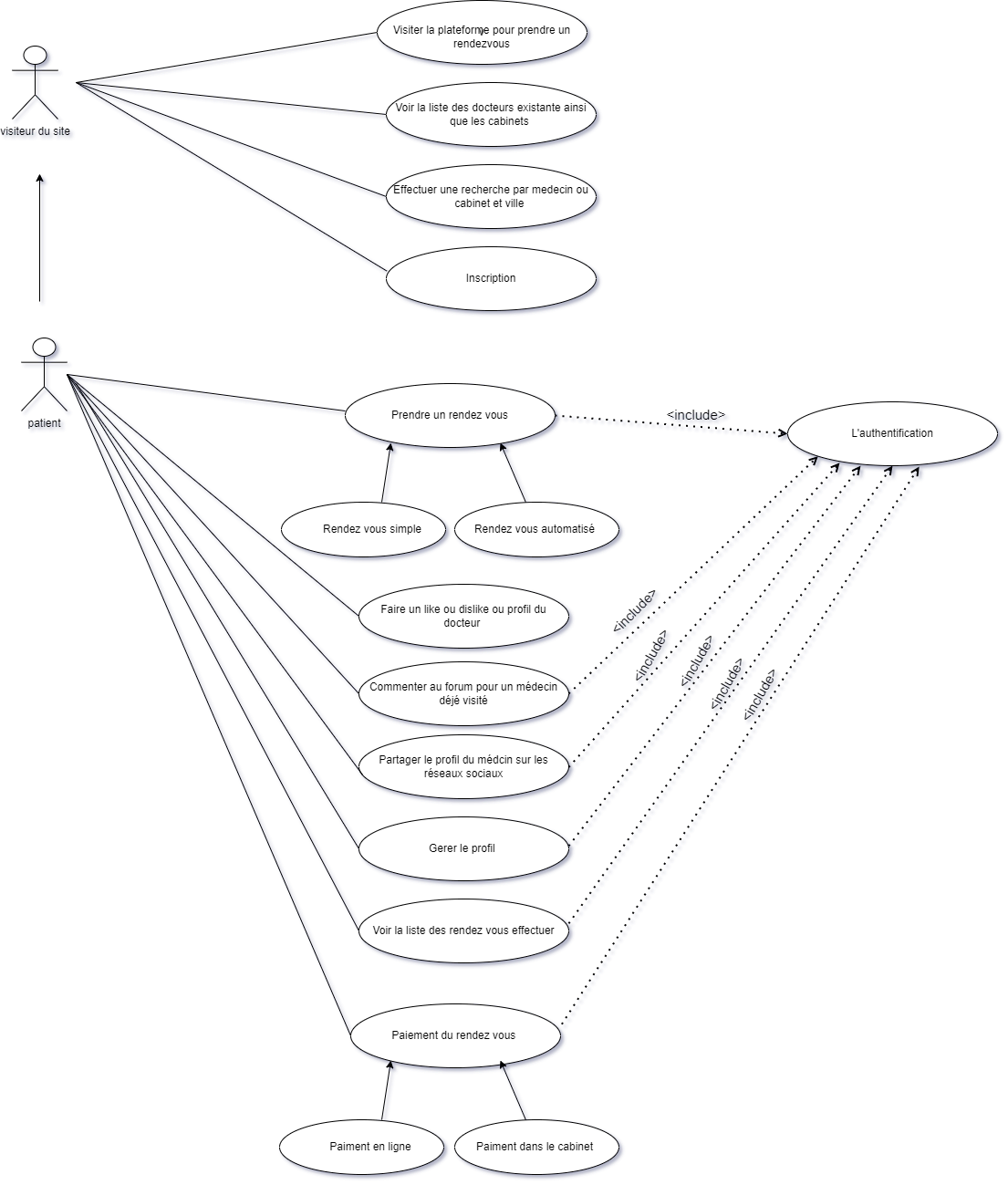
****

Figure 1 : Diagramme cas d’utilisation (visiteur et patient )

1. **Diagramme de cas d’utilisation de la secrétaire :**

Pour la secrétaire après son authentification elle peut gérer les rendez vous a savoir accepter un rendez-vous et le valider ou rejeter.

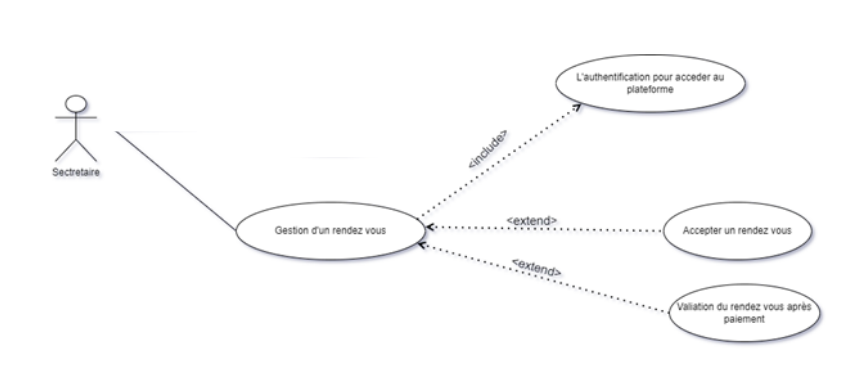
****

Figure 2 : Diagramme cas d’utilisation pour secrétaire

1. **Diagramme de cas d’utilisation du docteur :**

Pour le docteur après son inscription en entrant ces informations personnelles et ces professionnelles (diplôme ..) sur la plateforme , elle a l’accès à son Dashboard ou il va faire la gestion global de son cabinet a savoir effectuer plusieurs tache , d’abord il va ajouter son propre cabinet médical et faire la gestion des secrétaire en ajoutant une secrétaire à son cabinet (modifier, supprimer , consulter ) , il peut faire aussi la gestion de son profil , aussi la gestion des rendez vous en confirmant ou supprimant un rendez vous , il peut faire la gestion

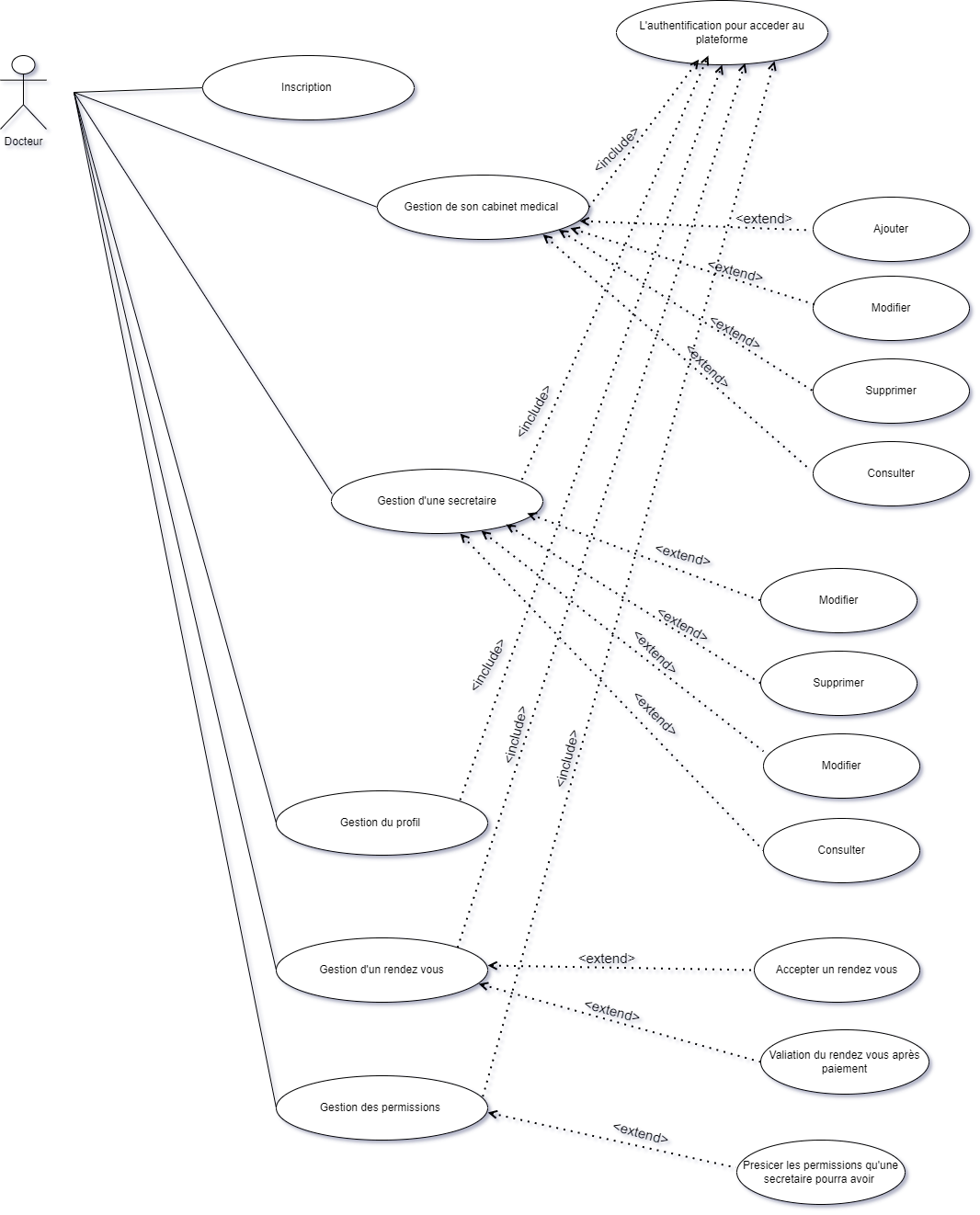
****

Figure 3 : Diagramme cas d’utilisation pour docteur

1. **Diagramme de cas d’utilisation de l’admin :**

Pour l’administrateur du site après avoir fait l’authentification, peut faire la gestion global de la plateforme , après l’inscription d’un médecin sur la plateforme , c’est l’administrateur qui va faire une validation du docteur après avoir fait une vérification des données professionnelles , il peut faire aussi la gestion des contacts , en consultant la liste des contact des patients et en répondant aux messages des patients il peut aussi faire la gestiones permissions ca veut gérer les permissions qu’il va donner à son secrétaire .

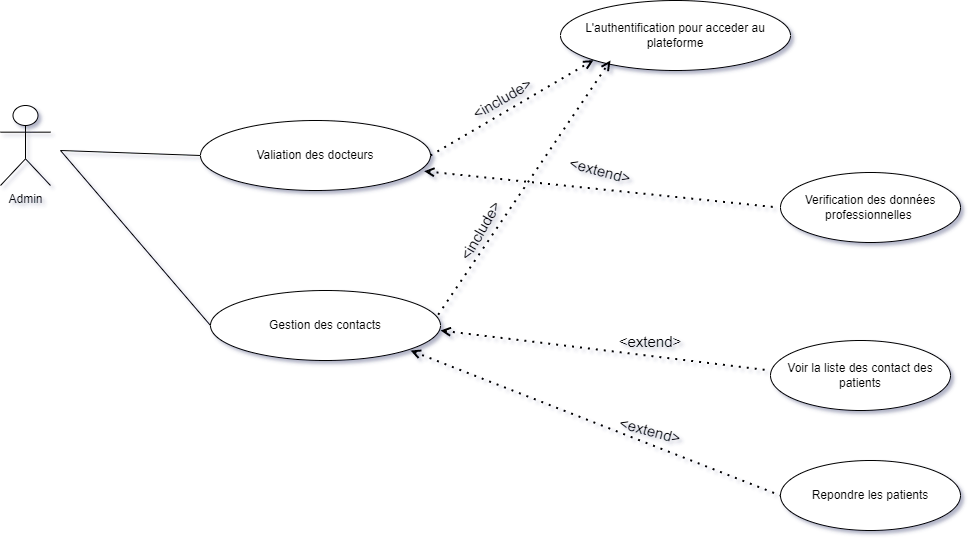
****

Figure 4 : Diagramme cas d’utilisation pour Admin

* 1. **Diagramme de classe :**

Un diagramme de classes fournit une vue globale d'un système en présentant ses classes, interfaces et collaborations, et les relations entre elles.

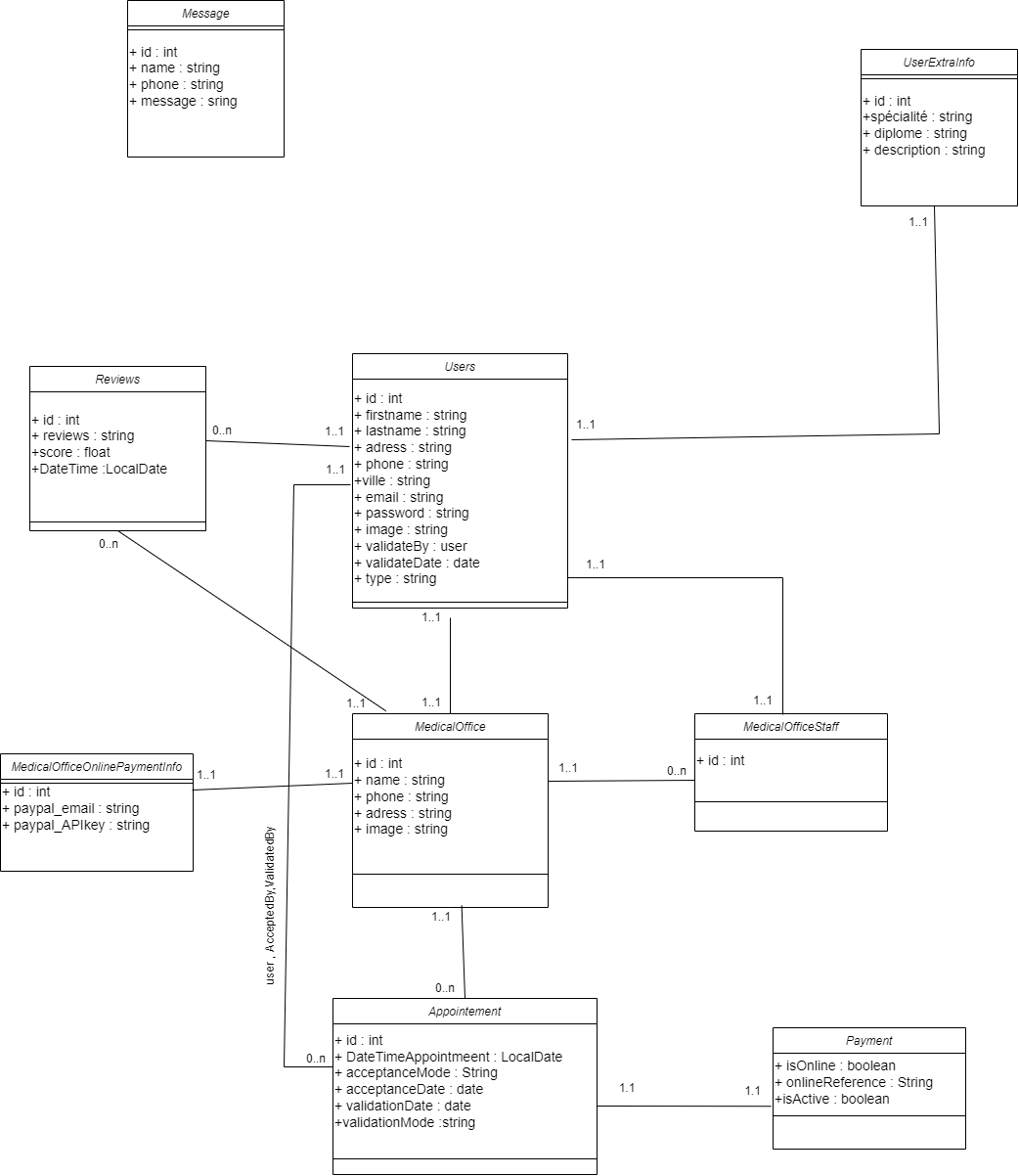
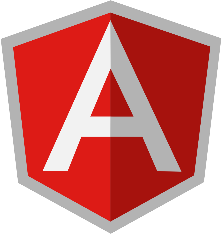
****

Figure 1 : Diagramme de classe de la plateforme

1. **Réalisation :**
   1. Outils de développement :
2. PowerAMC :

PowerAMC : est un des outils majeurs de modélisation des données et des processus. Il a été créé par la société Sybase, est désormais propriété de SAP.

1. Angular :



**Angular** (communément appelé "**Angular 2+**" ou **Angular v2 et plus** « est un framework côté client, open source basé sur typescript, et co-dirigé par l'équipe du projet « Angular » à [google](https://fr.wikipedia.org/wiki/Google) et par une communauté de particuliers et de sociétés. Angular est une réécriture complète d'[AngularJS](https://fr.wikipedia.org/wiki/AngularJS), [cadriciel](https://fr.wikipedia.org/wiki/Cadriciel) construit par la même équipe. Il permet la création d’applications Web et plus particulièrement de ce qu’on appelle des

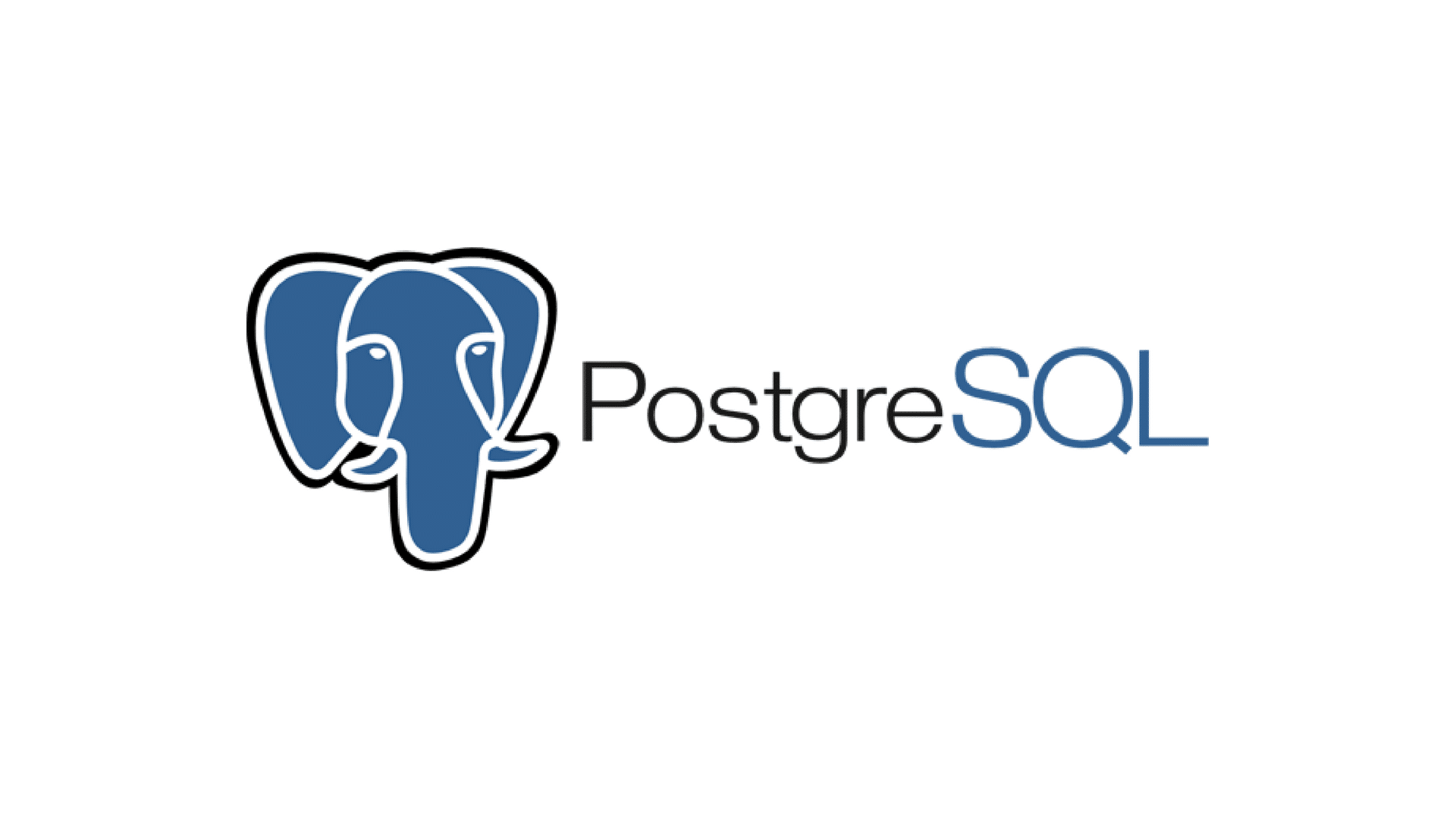
« *Single Page Applications* » :

1. SpringBoot :



**Spring Boot** est un nouveau Framework créé par l'équipe de chez Pivotal, conçu pour simplifier le démarrage et le développement de nouvelles applications **Spring**.

1. PostgreSQL :



PostgreSQL est un système de gestion de base de données relationnelle et objet. C'est un outil libre disponible selon les termes d'une licence de type BSD. Ce système est comparable à d'autres systèmes de gestion de base de données, qu'ils soient libres, ou propriétaires

1. Tailwind :



**Tailwind Css** est un framework **CSS** complètement personnalisable, basé sur le principe de classes utilitaires, dont la version 2.0 a été annoncée hier avec encore plus de nouveautés sympathiques. Le site officiel nous annonce la couleur immédiatement : construisez rapidement des sites sans quitter votre code HTML.

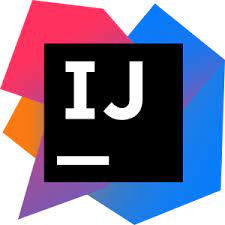
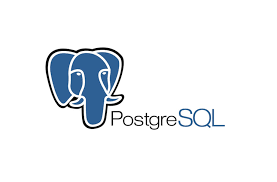
Pour la réalisation de ce projet, on va travailler avec :

**▪ Coté backend :** L’utilisation du spring boot pour la création de l’API Rest

**▪ SGBD :** PostgreSQL pour la création de la base de données

**▪ Coté Frontend :** Consommation de l’API Rest avec Angular

* 1. Logiciels utilisés :

Frontend : Backend : Gestion base de donnée :

* 1. Présentation de la solution réalisée :

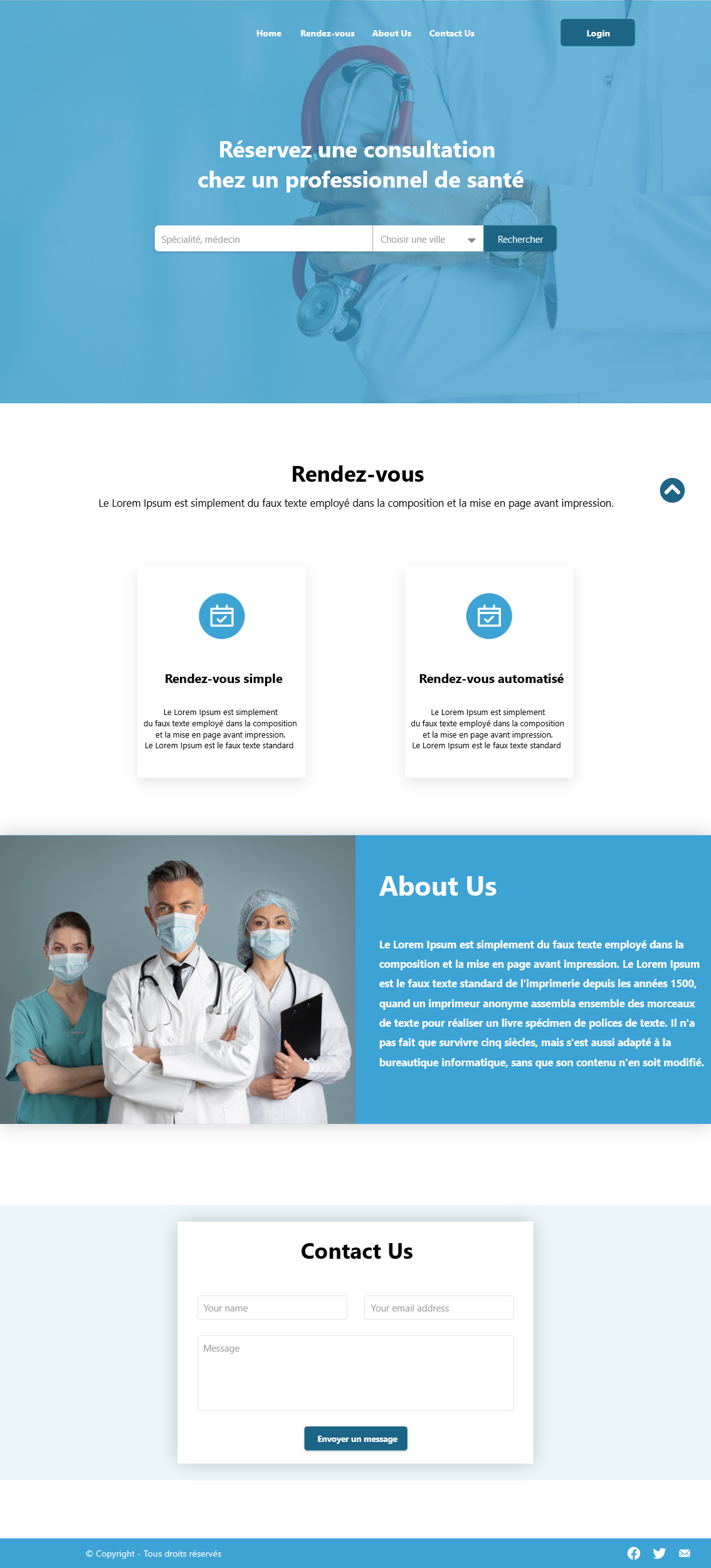
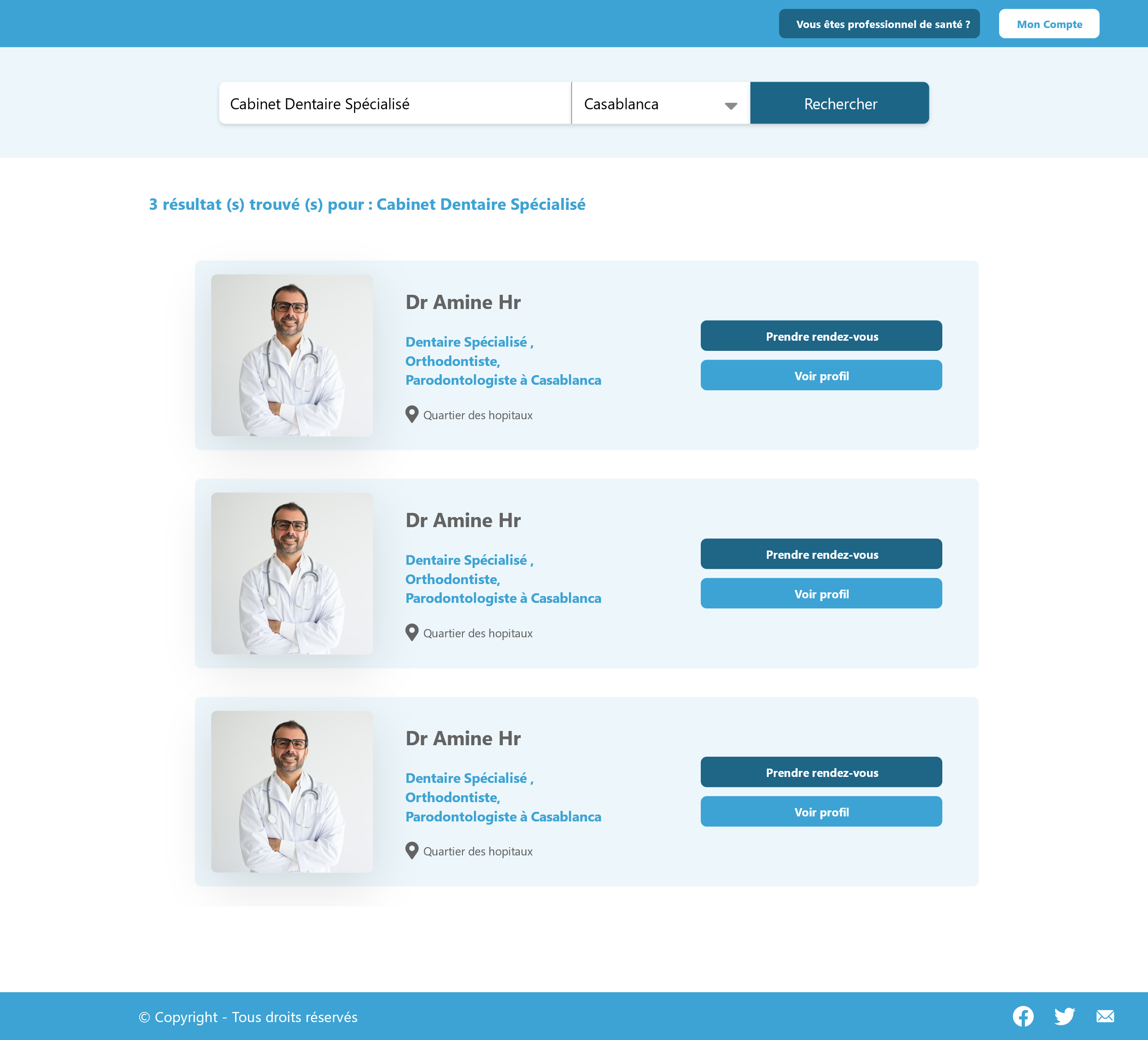
1. **Page d’accueil :**

Figure 1 : Page d’Acceuil

La page d’Acceuil c’est une page représentative de l’application pour donner au visiteur une idée sur le service proposé par la plateforme

1. **Page recherche d’un médecin ou cabinet médicale :**

Figure 2 : Page Recherche « par nom du médecin ou cabinet ou ville  »

Dans la page recherche un visiteur peut chercher un médecin ou bien un cabinet médical ou bien par ville, puis il aura l’affichage des docteurs avec un bouton pour voir le profil du médecin et un autre bouton pour prendre un rendez-vous.

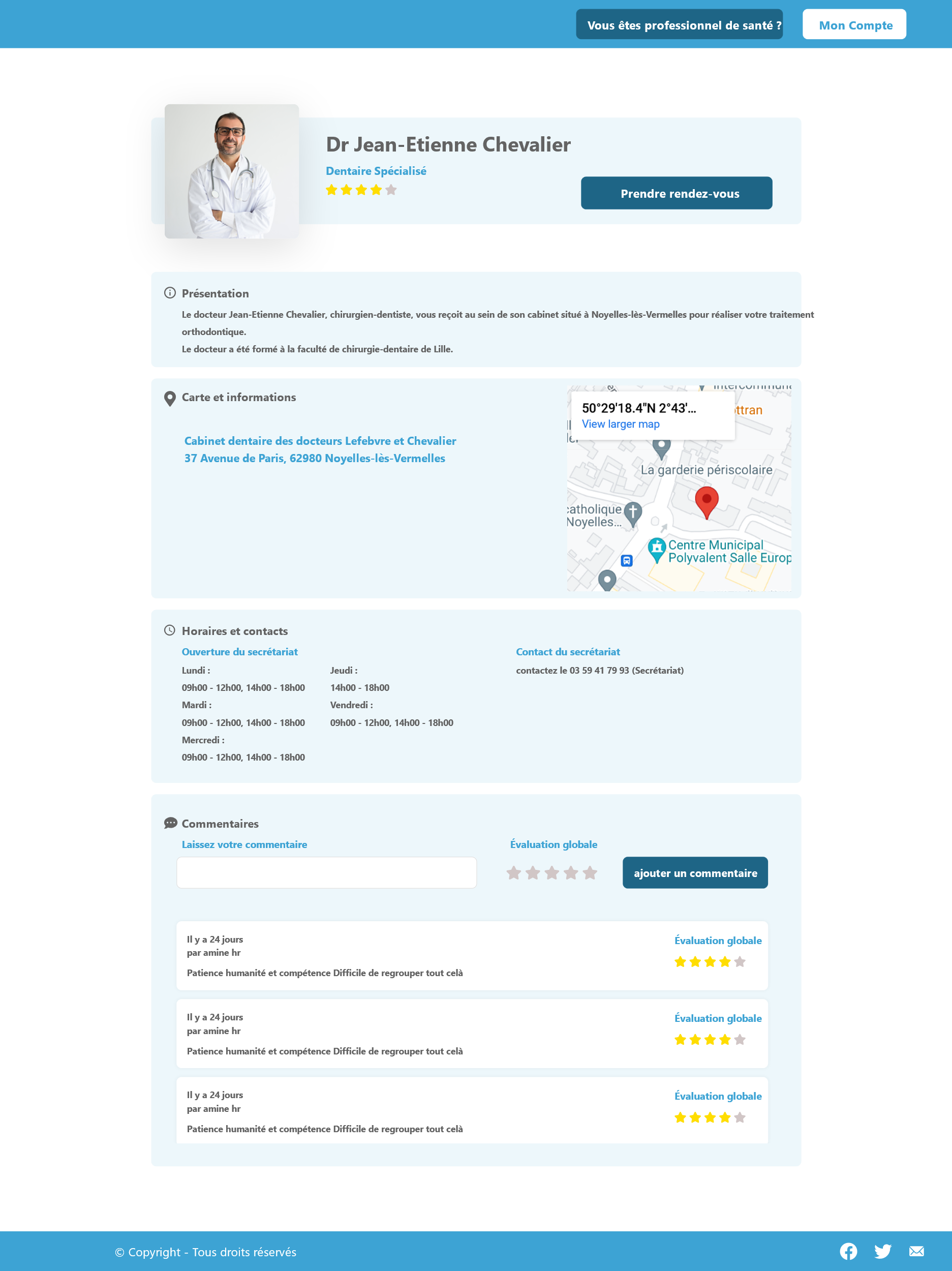
1. **Page d’information sur un médecin :**

Figure 3 : page pour les informations d’un médecin

Cette page représente les informations d’un médecin à savoir les horaires du travail, une petite représentation sur le médecin et aussi les commentaires des patients sur un médecin

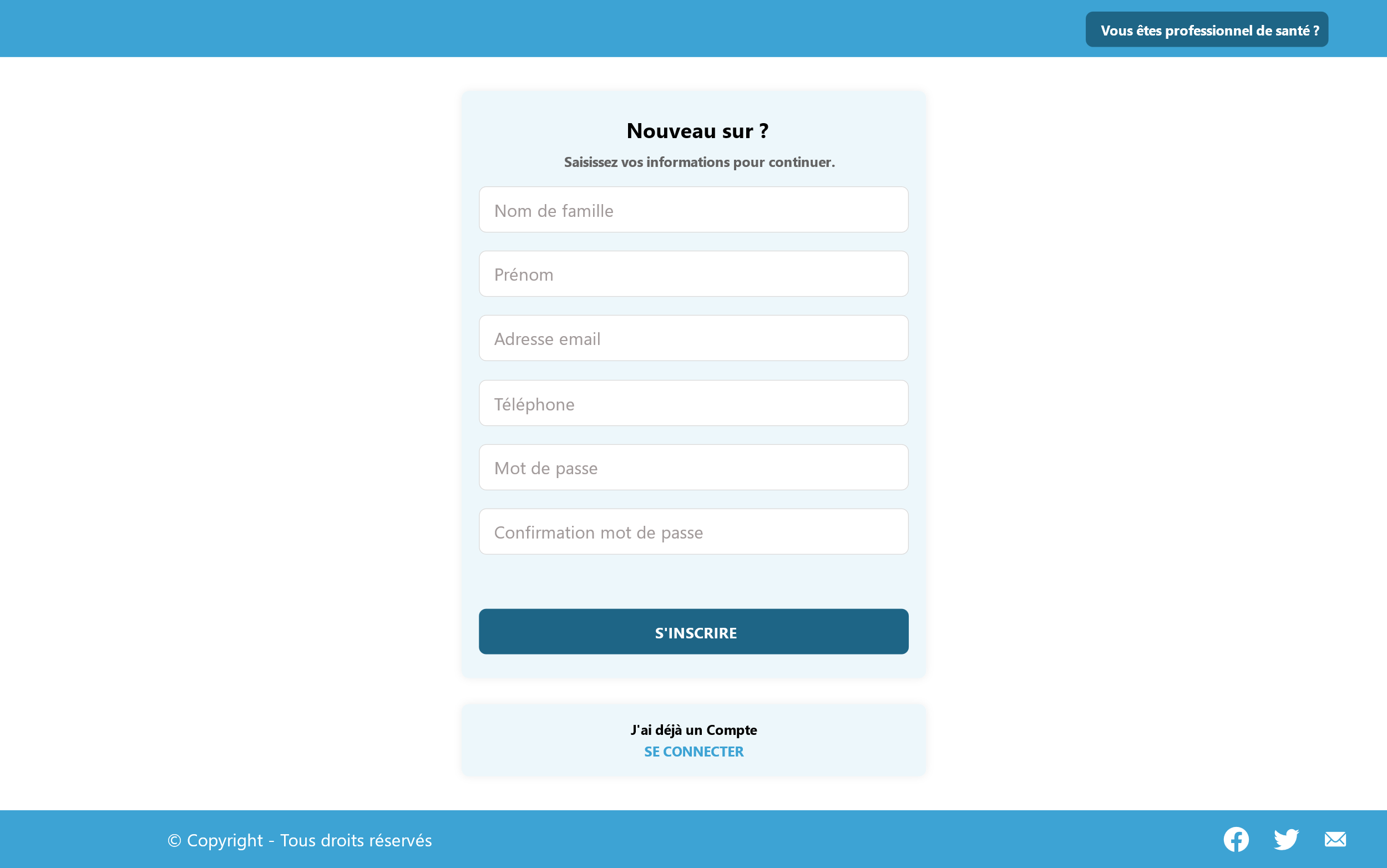
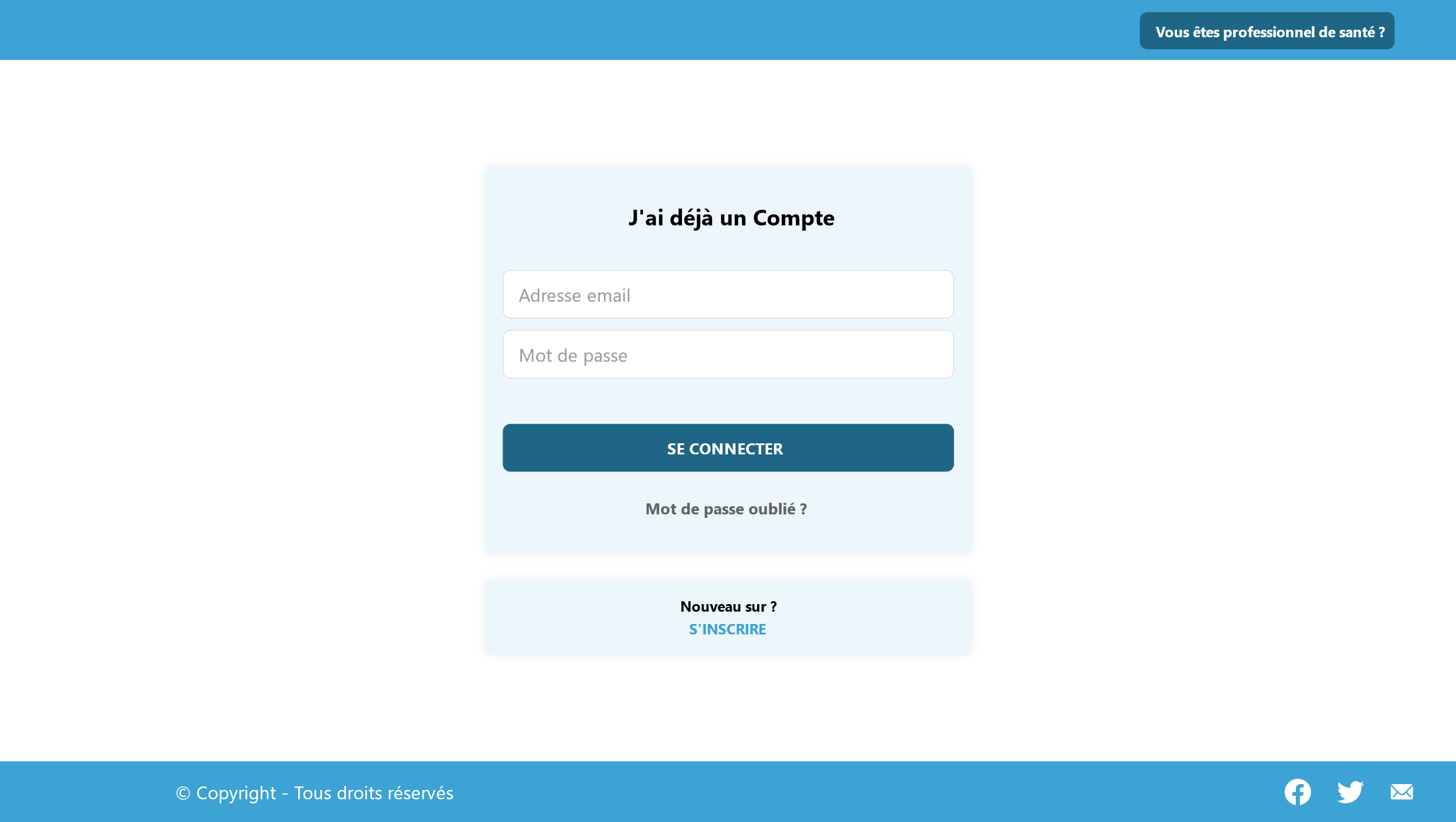
1. **Inscription pour un visiteur :**

Figure 4 : Inscription d’un visiteur

Cette page c’est pour faire l’inscription afin de devenir patient du site, l’inscription se fait en ajoutant l’ensemble des informations personnelles.

1. **Login pour prendre un rendez-vous :**

 Figure 5 : Page login

Cette page c’est une page pour faire l’authentification pour un patient, donc afin de prendre un rendez -vous le patient doit faire une authentification en entrant son email et password

* **Le cas où il a choisi un rendez-vous simple :**

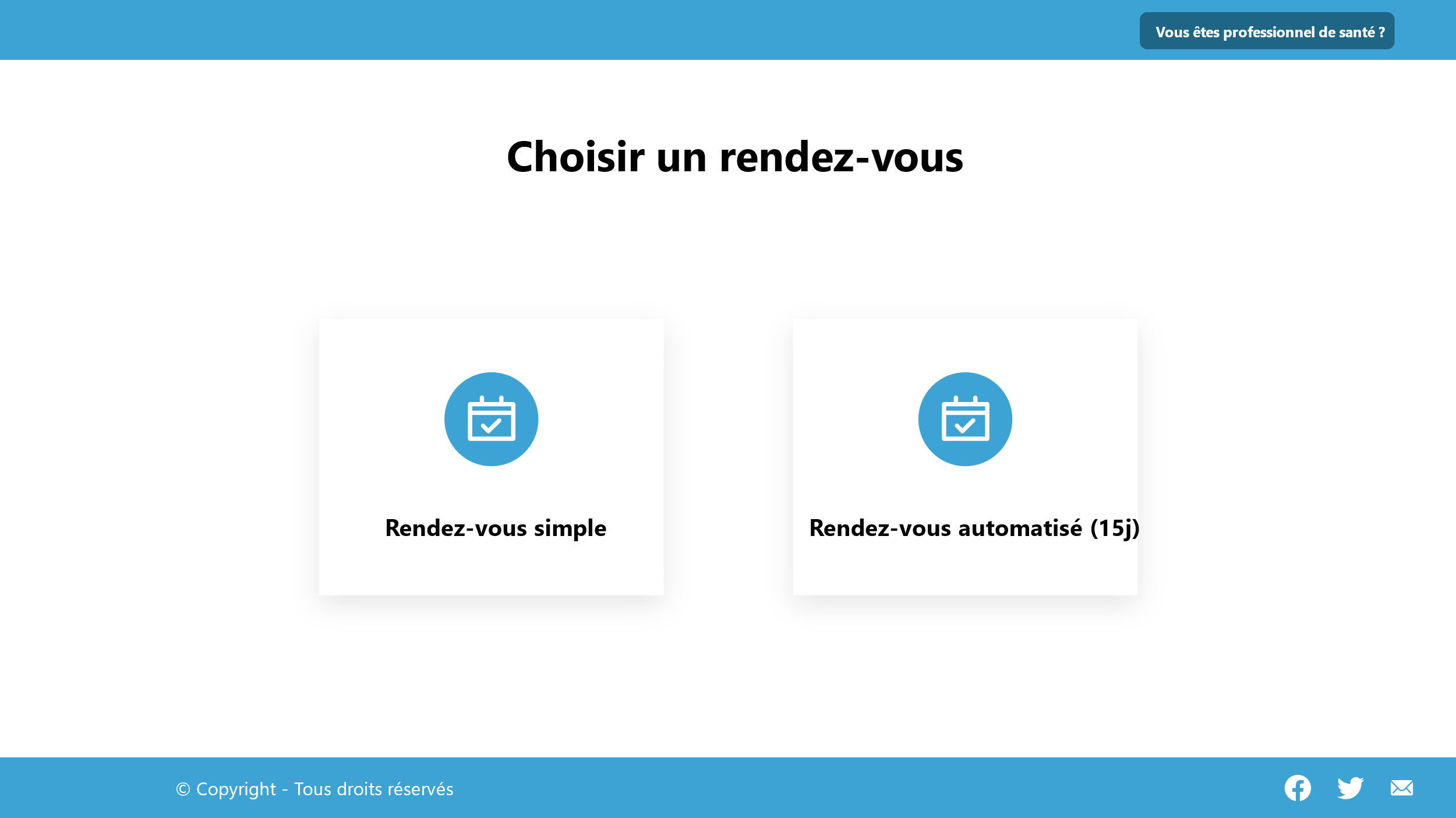
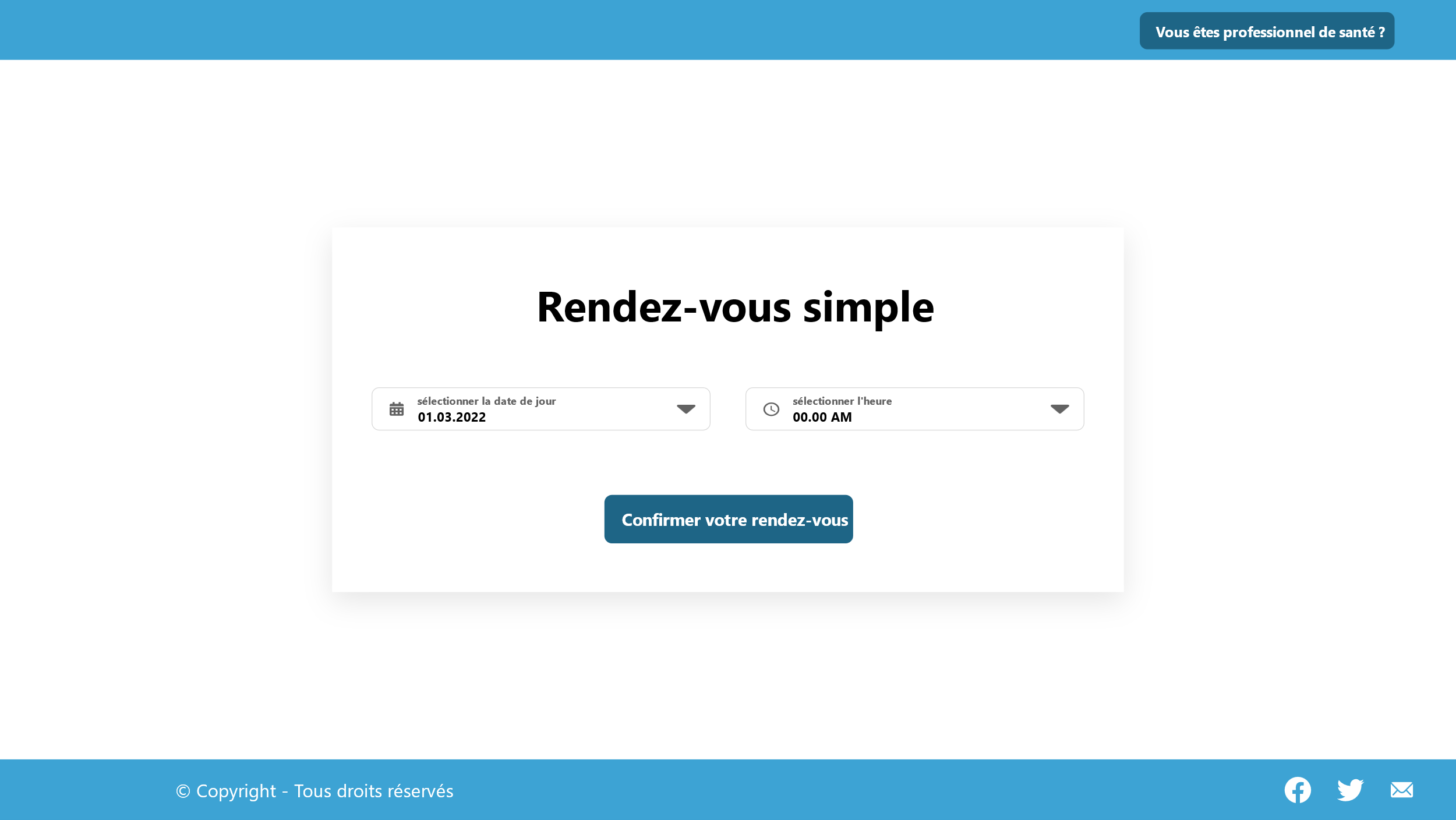
1. **Choix de rendez-vous :**

Figure 6 : Page choix du rendez-vous

Cette page présente les deux types de rendez-vous gérer par la plateforme, un rendez-vous simple qui a besoin d’être accepter par le médecin et la secrétaire et que le patient va au cabinet pour faire le paiement afin qu’il soit validé, et un rendez-vous automatisé qui doit être d’une date supérieure à 15j et qu’il est validé automatiquement par le système

1. **Choix d’un rendez-vous Simple :**

Figure 7 : Information du rendez-vous simple

Après le choix d’un rendez-vous simple, on a cette interface qui demande au patient d’entrer une date et une heure convenable a ces besoins pour passer un rendez-vous et la confirmer

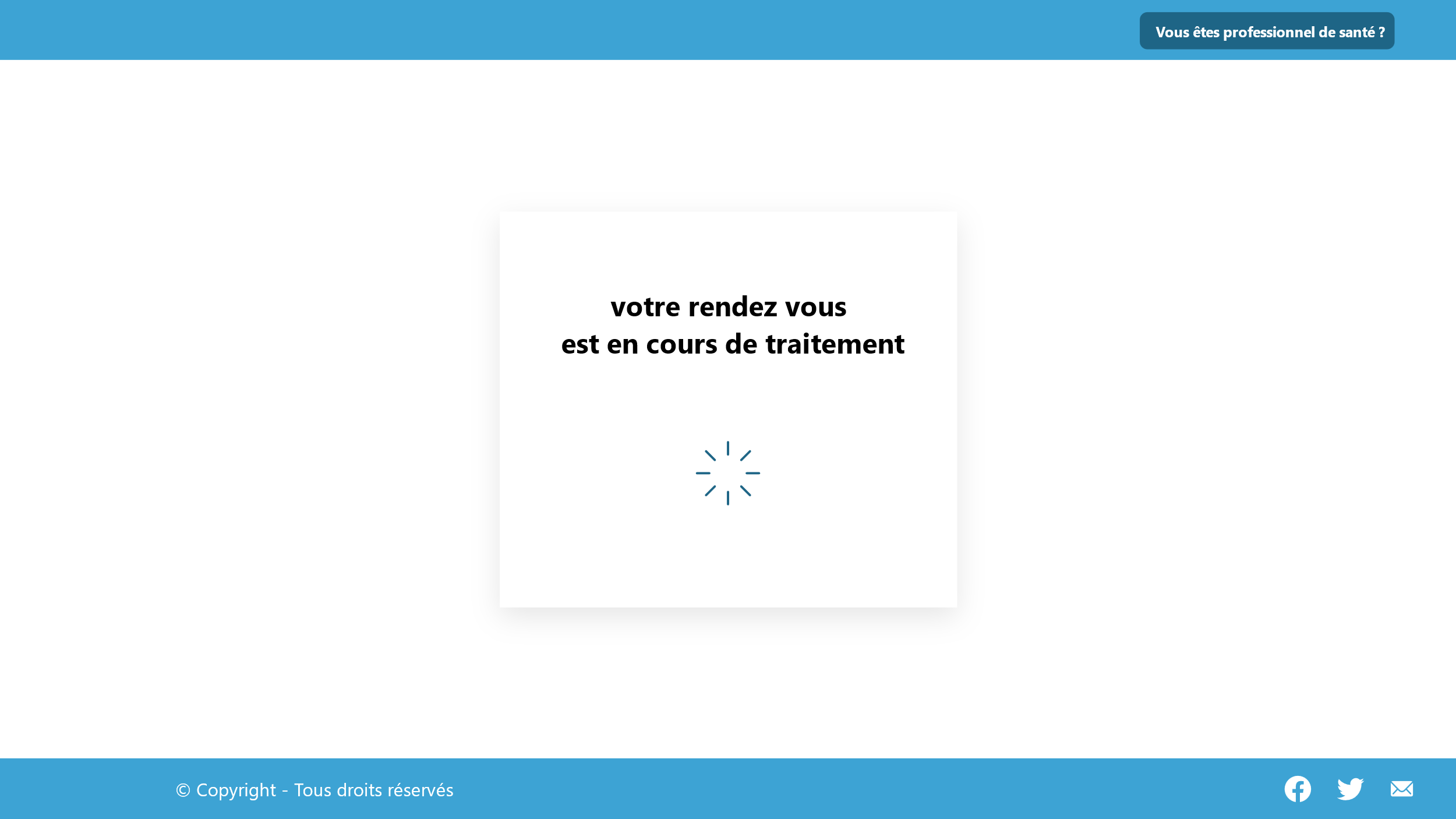
1. **Rendez-vous en cours de traitement :**

Figure 8 : En cours de traitement

Lorsque le patient confirme son rendez-vous on a cette interface qui apparu pour dire au patient que son rendez-vous est en cours de traitement

1. **Rendez-vous confirmer après avoir effectué un paiement dans le cabinet :**

Figure 9 : Rendez-vous confirmé

Cette interface apparue lorsqu’un médecin ou une secrétaire confirme un rendez-vous c’est pour dire au patient que son rendez-vous est confirmé il reste juste de partir au cabinet pour payé le rendez-vous afin d’être validé

* **Le cas où il a choisi un rendez-vous simple :**

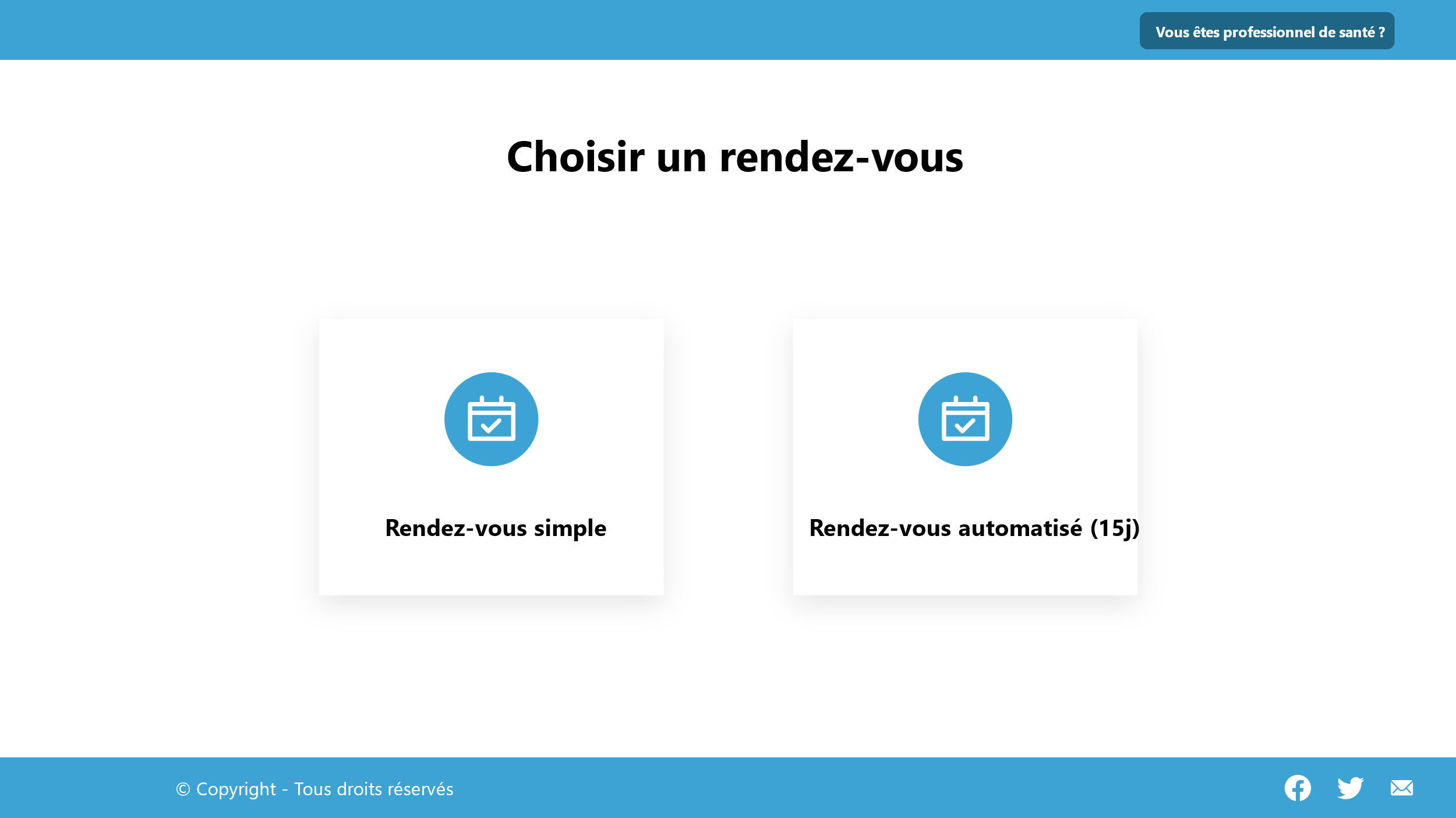
1. **Choix d’un rendez-vous :**

Figure 10 : Page choix du rendez-vous

Cette page présente les deux types de rendez-vous gérer par la plateforme, un rendez-vous simple qui a besoin d’être accepter par le médecin et la secrétaire et que le patient va au cabinet pour faire le paiement afin qu’il soit validé, et un rendez-vous automatisé qui doit être d’une date supérieure à 15j et qu’il est validé automatiquement par le système

1. **Choix d’un rendez-vous Automatisé :**



Figure 11 : Information Rendez-vous automatisé

Pour un rendez-vous automatisé le patient doit remplir une date supérieure à 15j et sélectionner une heure convenable, et une méthode de paiement

1. **Finalisation du paiement :**

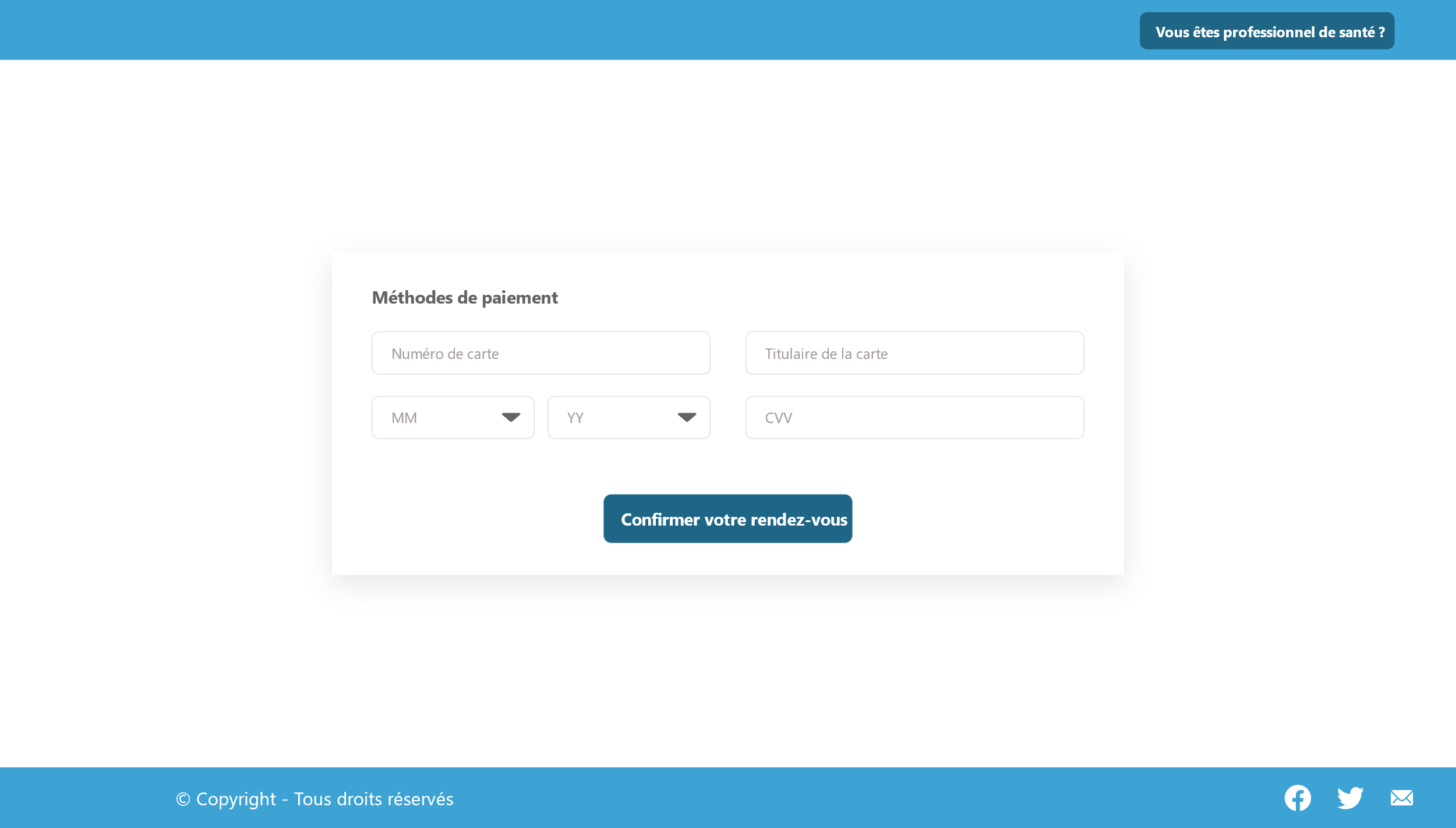


Figure 12 : Méthode de paiement Rendez-vous automatisé

Dans cette étape le patient doit entrer le numéro de la carte pour effectuer un paiement par carte bancaire et confirmer son paiement .

1. **Confirmation automatique du rendez-vous :**



Figure 13 : Rendez-vous confirmé

Cette interface apparue lorsqu’un médecin ou une secrétaire confirme un rendez-vous c’est pour dire au patient que son rendez-vous est confirmé

1. **Paiement en ligne du rendez-vous pour la validation :**

Figure 14 : Rendez-vous confirmé

Cette interface montre au patient que son paiement à été effectué avec succès

**Dashboard Docteur** :



Figure 15 : Dashboard

Cette interface montre le Dashboard du docteur et de l’admin

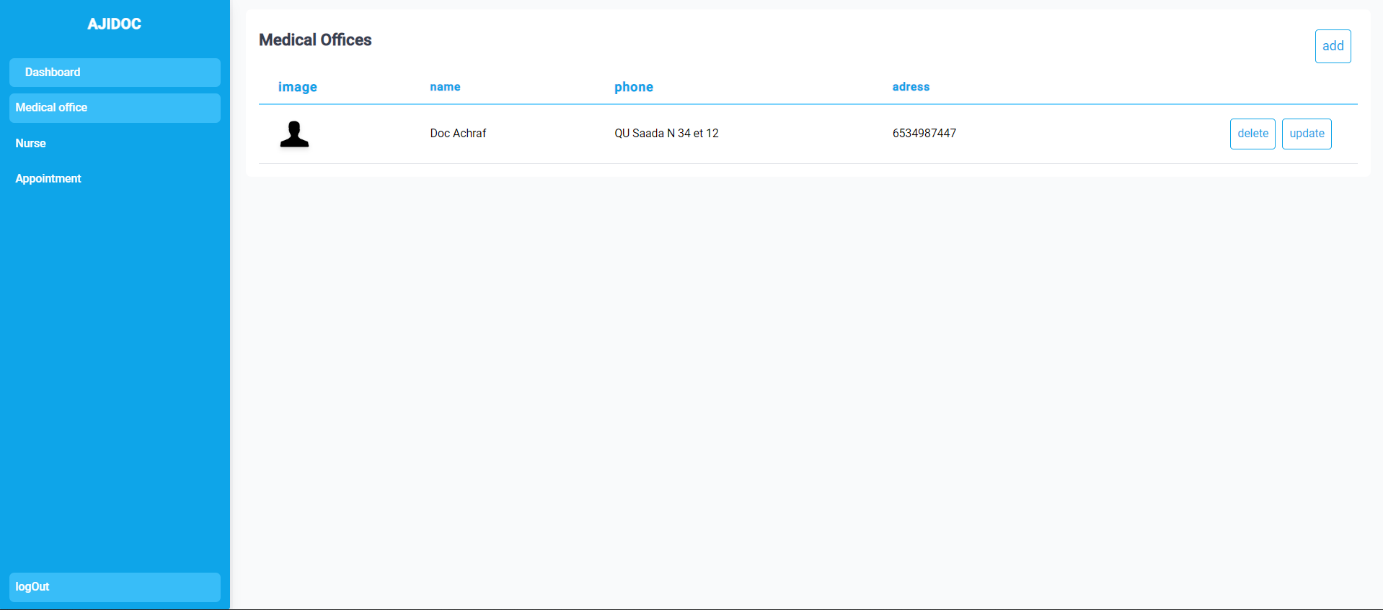
**Dashboard DOCTOR** :

Figure 16 : Dashboard docteur

**Conclusion :**

Pour conclure , ce projet à pour but de faciliter aux patients la prise de rendez vous , ainsi que aider le docteur à faire la gestion de son cabinet ainsi que des rendez-vous , La réalisation de ce projet m’as permis d’acquérir de développer mes compétences acquises tout au long de la formation à Youcoachrafloubna

de , ainsi que de mettre en pratique mon savoir faire , Pour le moment le projet « AjiDoc  » n’est pas complet , certes il englobe la majorité des fonctionnalités cité dans le cahier des charges , Par la suite je compte développer l’ensemble des fonctionnalités et ajouté d’autres afin que je puisse le lancer au maché .

Ce projet à pour but de faciliter aux clients l’achat d’un cadeau selon un événement, ainsi de gagner le temps dans le déplacement à une boutique ou à une autre. La réalisation de se projet m’a permis d’acquérir de nouvelles connaissances et de développer mes compétences acquises au cours de ma formation à Youcode. Pour le moment « HADAYA BY LOUBNA » est un site dynamique et fonctionnel, par la suite je compte ajouter d’autres fonctionnalités au site afin que je puisse le lancer au marché.

**Références et bibliographie :**

<https://spring.io/projects/spring-boot>

<https://tailwindcss.com>

<https://angular.io/docs>