

RAPPORT PROJET SYSTEME D'INFORMATION

En utilisant Django

BOUTER RAMZI
MEKIDECHE WALID

L3 ISIL B
GROUPE 1

27/01/2024

TABLE DES MATIERES

Table des matières

Table of Contents

Introduction.....	1
Objectif	1
Description des Utilisateurs	1
Présentation du Système	2
Environnement.....	2
Contraintes	2
Adaptabilité.....	2
Optimisation des Performances.....	2
Sécurité	2
Accessibilité	2
Maintenance Facilitée.....	2
Etude de faisabilité et spécifications fonctionnelles.....	3
Besoins fonctionnelles.....	3
Besoins non-fonctionnelles	4
Sécurité	4
Facilité d'utilisation.....	4
Fiabilité	4
Performances	4
MERISE.....	5
Modèle Conceptuel de Données (MCD)	5
Modèle Relationnel (MLD)	6
Schéma de la base de donnée.....	7
Description des tables	7
Les tables dans la base de données	7
Description des relations entre les tables	7
Dictionnaire de données.....	7
Structure de code	9

TABLE DES MATIERES

Organisation du code source	9
Description de la structure des dossiers et des fichiers.....	9
Explication des principaux dossiers.....	9
Modèles de données	10
Description des modèles Django représentant les données	10
Relations entre les modèles	12
Utilisation.....	13
Connexion patient	13
Dashboard du Médecin	14
Dossier Médical	14
Admin Médecin.....	15
Conclusion	16

Introduction

OBJECTIF

Ce document vise à décrire les besoins, les exigences et les fonctionnalités de l'application **clinique** pour la gestion d'une clinique.

Le projet vise à créer un système de gestion médicale basé sur la plateforme Django avec un objectif principal est d'optimiser la gestion des rendez-vous, des dossiers médicaux, des prescriptions médicales, et des informations relatives aux patients et aux professionnels de la santé.

DESCRIPTION DES UTILISATEURS

Ce logiciel est destiné aux utilisateurs qui sont :

- Médecins
- Patients
- Personnel Administratif

Présentation du Système

ENVIRONNEMENT

Le système de gestion médicale repose sur une architecture moderne basée sur le Framework Django de Python, offrant ainsi une structure robuste et évolutive. L'environnement technique comprend une base de données relationnelle SQLite, assurant la persistance des données médicales de manière sécurisée. Le côté client est développé en utilisant des technologies web standard telles que HTML et CSS, assurant ainsi une compatibilité étendue avec différents navigateurs.

CONTRAINTES

Adaptabilité

La conception du code côté vue doit prendre en compte la diversité des utilisateurs, des patients aux professionnels de la santé, en assurant une interface conviviale adaptée à chaque profil.

Optimisation des Performances

La gestion des rendez-vous et des dossiers médicaux peut générer un volume important de données. Ainsi, le code côté vue doit être optimisé pour garantir des temps de changement minimaux, assurant une expérience utilisateur fluide.

Sécurité

Les interactions entre le client et le serveur doivent être sécurisées, en utilisant des protocoles de cryptage pour protéger les données sensibles transitant entre le navigateur et le serveur Django.

Accessibilité

Le code HTML et CSS doit être conforme aux normes d'accessibilité du web, garantissant l'utilisation du système par des individus ayant des besoins spécifiques.

Maintenance Facilitée

La structure du code côté vue doit être modulaire et bien documentée pour faciliter la maintenance future, permettant aux développeurs de comprendre rapidement le fonctionnement du système et d'apporter des modifications si nécessaire.

Etude de faisabilité et spécifications fonctionnelles

BESOINS FONCTIONNELLES

Ce logiciel a de nombreuses fonctionnalités qui assure les besoins des utilisateurs :

- Authentification
- Gestion des Médecins
 - Ajouter un Médecin
 - Modifier un Médecin
 - Supprimer un Médecin
 - Chercher un Médecin
- Gestion des patients
 - Ajouter un Patient
 - Modifier un Patient
 - Supprimer un Patient
 - Chercher un Patient
- Gestion des Rendez-vous
 - Ajouter un Rendez-vous
 - Modifier un Rendez-vous
 - Supprimer un Rendez-vous
 - Chercher un Rendez-vous
- Gestion des Dossiers Médicaux
 - Ajouter un Dossier Médical
 - Modifier un Dossier Médical
 - Supprimer un Dossier Médical
 - Chercher un Dossier Médical
- Gestion des Prescriptions Médicales
 - Ajouter une Prescriptions Médicales
 - Modifier une Prescriptions Médicales
 - Supprimer une Prescriptions Médicales
 - Chercher une Prescriptions Médicales
- Gestion des Stocks de Médicaments
 - Ajouter un Médicament
 - Modifier un Médicament
 - Supprimer un Médicament
 - Chercher un Médicament
- Gestion des Actes Médicaux
 - Ajouter un Acte Médical
 - Modifier un Acte Médical
 - Supprimer un Acte Médical

- Chercher un Acte Médical
- Gestion des Chirurgies
 - Ajouter une chirurgie
 - Modifier une chirurgie
 - Supprimer une chirurgie
 - Chercher une chirurgie
- Gestion des Salles d'opérations
 - Ajouter une salle d'opération
 - Modifier une salle d'opération
 - Supprimer une salle d'opération
 - Chercher une salle d'opération

BESOINS NON-FONCTIONNELLES

Sécurité

Le logiciel garde les informations des utilisateurs en sécurité en utilisant l'authentification.

Facilité d'utilisation

Avec l'interface graphique simple à comprendre l'utilisateur ne va pas trouver des difficultés.

Fiabilité

Les besoins sont validés.

Performances

Ce besoin non fonctionnel exige que le système soit capable de garantir une expérience utilisateur satisfaisante et une réponse rapide.

MERISE

MODELE CONCEPTUEL DE DONNEES (MCD)

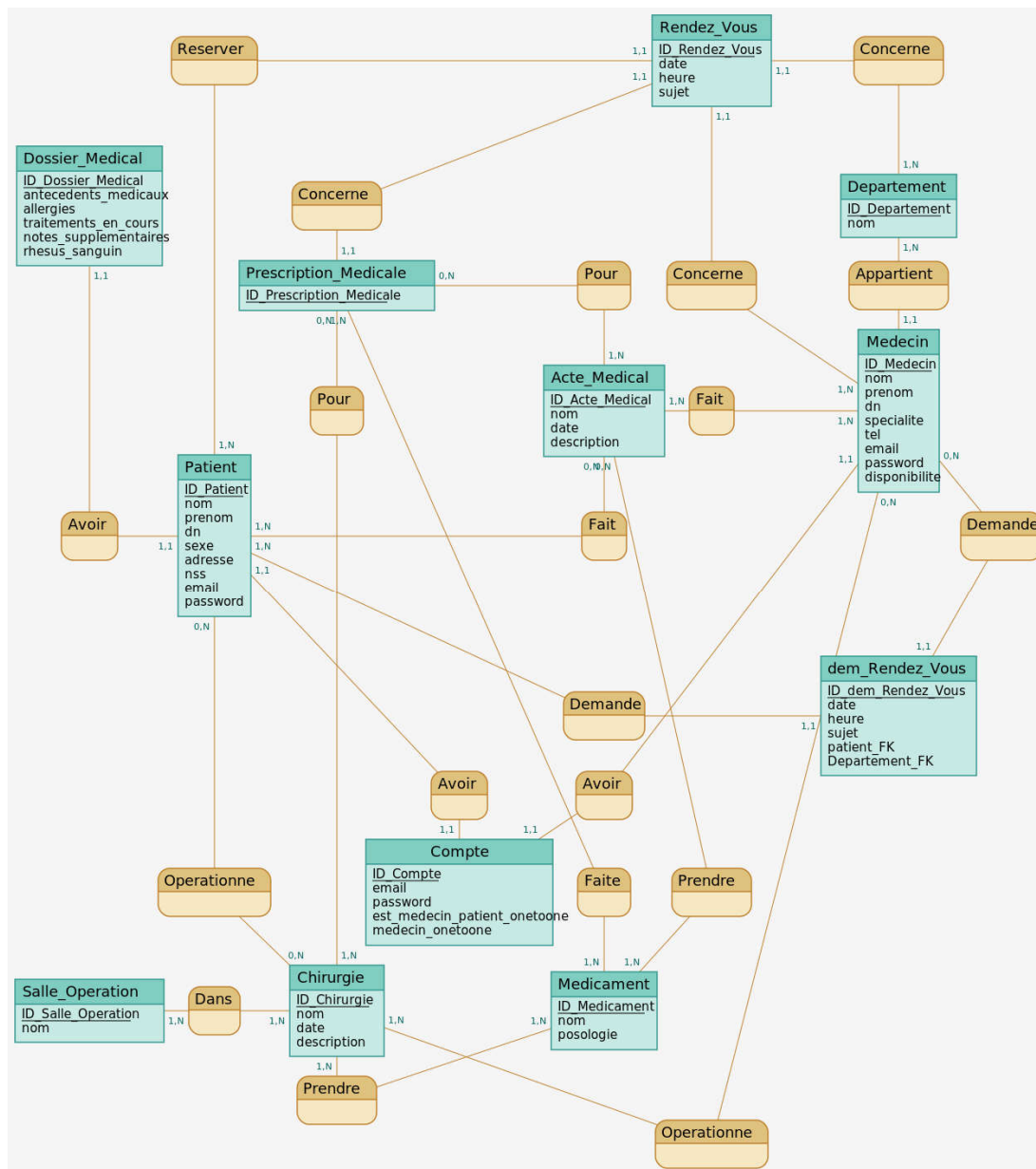


Figure 1: MCD

MODELE RELATIONNEL (MLD)

ACTEMEDICAL (ID ActeMedical, nom, date, description)

CHIRURGIE (ID Chirurgie, nom, date, description)

COMPTE (ID Compte, email, password, est_medecin)

DANS (#ID Chirurgie, #ID SalleOperation)

DEMRENDEZVOUS (ID_demRendezVous, date, heure, sujet, #ID_Departement, #ID_Medecin, #ID_Patient)

DEPARTEMENT (ID Departement, nom)

DOSSIERMEDICAL (ID DossierMedical, antecedents_medicaux, allergies, traitements_en_cours, notes_supplementaires, rhesus_sanguin, #ID_Patient)

FAIT (#ID_Patient, #ID ActeMedical)

FAIT (#ID_Medecin, #ID ActeMedical)

FAITE (#ID PrescriptionMedicale, #ID Medicament)

MEDECIN (ID_Medecin, nom, prenom, dn, specialite, tel, email, password, disponibilite, #ID_Departement, #ID_Compte)

MEDICAMENT (ID Medicament, nom, posologie)

OPERATIONNE (#ID_Patient, #ID Chirurgie)

OPERATIONNE (#ID_Medecin, #ID Chirurgie)

PATIENT (ID_Patient, nom, prenom, dn, sexe, adresse, nss, email, password, #ID_Compte)

POUR (#ID ActeMedical, #ID PrescriptionMedicale)

POUR (#ID Chirurgie, #ID PrescriptionMedicale)

PRENDRE (#ID ActeMedical, #ID Medicament)

PRENDRE (#ID Chirurgie, #ID Medicament)

PRESCRIPTIONMEDICALE (ID PrescriptionMedicale, #ID_RendezVous)

RENDEZ_VOUS (ID_RendezVous, date, heure, sujet, #ID_Patient, #ID_Departement, #ID_Medecin)

SALLEOPERATION (ID SalleOperation, nom)

Schéma de la base de donnée

DESCRIPTION DES TABLES

Les tables dans la base de données

Les tables dans la base de données sont créées selon le fichier **models.py** qui a été déterminé grâce au **MCD** et le **MLD**.

Description des relations entre les tables

Les relations représentent les associations qui existent dans le **MCD**, qui seront ensuite représentées dans le **MLD** et les clés étrangères de type **OneToOne**, **OneToMany**, **ManyToMany**, et **ForeignKey**.

DICTIONNAIRE DE DONNEES

Libellé de l'attribut	Signification	Type	Taille	Règle
ID_ActeMedical	Identificateur de l'entité ActeMedical	N	10	
date	Date de l'acte médical	Date		AAAA-MM-JJ
Description	Description de l'acte médical	AN	100	
Nom	Nom de l'acte médical	A	100	
ID_Chirurgie	Identificateur de l'entité Chirurgie	N	10	
Date	Date de la chirurgie	Date		AAAA-MM-JJ
Description	Description de la chirurgie	AN	100	
Nom	Nom de la chirurgie	A	100	
ID_Departement	Identificateur de l'entité Departement	N	10	
Nom	Nom du département	A	100	
ID_DossierMedical	Identificateur de l'entité DossierMedical	N	10	
antecedents_medicaux	Les antecedents medicaux du patient	A	500	
Allergies	Les allergies du patient	A	500	
traitements_en_cours	Les traitements en cours du patient	An	500	
notes_supplementaires	Des notes supplementaire sur le patient	A	500	
rhesus_sangain	Rhesus sangain du patient	AN	1	
ID_Medecin	Identificateur de l'entité Medecin	N	10	
Nom	Nom du médecin	A	100	
Prenom	Prénom du médecin	A	100	
Dn	Date de naissance du médecin	Date	100	AAAA-MM-JJ

Specialite	Spécialité du médecin	A	100	
Tel	Numéro de téléphone du médecin	N	10	
Email	Email du medecin	AN	100	
Password	Mot de passe de compte du medecin	AN	128	
Disponibilite	Disponibilite du medecin	N	1	
ID_Patient	Identificateur de l'entité Patient	N	10	
Nom	Nom du patient	A	100	
Prenom	Prenom du patient	A	100	
Dn	Date de naissance du patient	Date		AAAA-MM-JJ
Sexe	Sexe du patient	A	1	
Adresse	Adresse du patient	AN	100	
Nss	Numero de securite sociale du patient	N		
Email	Email du patient	AN	100	
Password	Mot de passe de compte du patient	AN	128	
ID_Medicament	Identificateur de l'entité Medicament	N	10	
Nom	Nom du medicament	A	100	
Posologie	Posologie du medecament	AN	500	
ID_PrescriptionMedicale	Identificateur de l'entité PrescriptionMedicale	N	10	
ID_RendezVous	Identificateur de l'entité RendezVous	N	10	
Date	Date du rendez-vous	Date		AAAA-MM-JJ
Heure	Heure du rendez-vous	Heure		HH : MM
Sujet	Sujet du rendez-vous	AN	500	
ID_SalleOperation	Identificateur de l'entité SalleOperation	N	10	
Nom	Nom de la salle d'operation	AN	100	

Structure de code

ORGANISATION DU CODE SOURCE

Description de la structure des dossiers et des fichiers

La structure des dossiers et des fichiers de ce projet est soigneusement organisée pour garantir une gestion efficace du code source et des ressources. Voici une brève description de la disposition des dossiers et fichiers :

clinique/ :

admin.py: Contient la configuration pour l'interface d'administration Django.

forms.py: Formulaires utilisés dans l'application.

models.py: Définition des modèles de données.

urls.py: Gestion des URL de l'application.

views.py: Logique de gestion des vues.

templates/

Contient les fichiers HTML pour les pages web.

static/

Contient les fichiers statiques tels que les feuilles de style CSS, les images, etc.

manage.py

Point d'entrée pour les commandes de gestion Django.

db.sqlite3

Base de données SQLite pour stocker les données de l'application.

Cette structure favorise la modularité, la maintenabilité et la compréhension du code source. Les fichiers sont organisés en fonction de leur fonctionnalité, facilitant ainsi le développement, la collaboration et la gestion du projet.

Explication des principaux dossiers

Le projet Django suit une architecture de type Modèle-Vue-Contrôleur (MVC), où chaque composant a son propre dossier pour une organisation claire. Voici une explication des principaux dossiers du projet :

Modèles (models/)

Fichier clé : **models.py**

Ce dossier contient la définition des modèles de données de l'application, tels que les classes pour les patients, les médecins, les rendez-vous, etc. Ces modèles définissent la structure de la base de données.

Vues (views/)

*Fichier clé : **views.py***

Responsable de la logique de présentation et de l'interaction avec les utilisateurs. Les vues traitent les requêtes HTTP et retournent les réponses appropriées, généralement en utilisant des templates HTML pour rendre les pages.

Contrôleurs (controllers/)

Dans le contexte de Django, les contrôleurs sont gérés par le framework lui-même et sont représentés par les fonctions définies dans le fichier **views.py**. Ces fonctions (vues) agissent comme des contrôleurs, traitant les demandes utilisateur et coordonnant les actions nécessaires.

Formulaires (forms/)

*Fichier clé : **forms.py***

Ce dossier contient la définition des formulaires utilisés pour collecter les données des utilisateurs. Les formulaires peuvent inclure des champs pour saisir des informations telles que le nom, la date de naissance, etc.

Templates (templates/)

Ce dossier contient les fichiers HTML utilisés pour générer les pages web. Les templates sont utilisés par les vues pour présenter les données et les rendre visuellement.

Statiques (static/)

*Sous-dossiers : **css/**, **images/**, etc.*

Contient les fichiers statiques tels que les feuilles de style CSS, et les images utilisées dans les pages HTML.

Cette organisation suit les conventions de Django, permettant une séparation claire des responsabilités et une maintenance facilitée du code. Les principaux fichiers pour chaque composant sont indiqués, offrant une orientation rapide pour les développeurs qui contribuent au projet.

MODELES DE DONNEES

Description des modèles Django représentant les données

La description des modèles Django constitue un élément central de la conception de la base de données et de la représentation des données dans le projet. Voici un aperçu des principaux modèles Django utilisés dans l'application :

Patient

Représente les informations sur les patients, telles que leur nom, prénom, date de naissance, sexe, adresse, numéro de sécurité sociale, e-mail et mot de passe.

Compte

Contient les informations d'authentification des utilisateurs, avec des champs tels que l'e-mail et le mot de passe.

Departement

Décrit les différents départements médicaux auxquels les médecins peuvent être associés.

Medecin

Contient les données relatives aux médecins, y compris leur nom, prénom, date de naissance, spécialité, numéro de téléphone, e-mail, disponibilité et département associé.

RendezVous

Modélise les rendez-vous entre les patients et les médecins, avec des informations telles que la date, l'heure, le département et un sujet facultatif.

DossierMedical

Représente le dossier médical associé à chaque patient, avec une description générale.

ActeMedical

Décrit un acte médical, avec un nom, une date et une description.

Chirurgie

Modèle pour enregistrer les détails d'une chirurgie, y compris le nom, la date et la description.

SalleOperation

Représente les salles d'opération disponibles, avec un nom unique

Medicament

Contient les informations sur les médicaments, y compris le nom et la posologie.

PrescriptionMedicale

Associe des médicaments, actes médicaux et chirurgies à un rendez-vous, permettant de générer des prescriptions médicales.

demRendezVous

Modélise les demandes de rendez-vous des patients avec des informations telles que la date, l'heure, le département et un sujet, avant la confirmation.

Chaque modèle est conçu pour représenter une entité spécifique du domaine médical, permettant ainsi de structurer les données de manière logique et cohérente. Ces modèles sont utilisés pour créer et interagir avec la base de données du projet Django.

Relations entre les modèles

Les modèles Django interagissent entre eux à travers des relations qui reflètent les liens logiques entre les différentes entités du système de gestion médicale. Voici un aperçu des principales relations entre les modèles :

Relation Patient - RendezVous

Un patient peut avoir plusieurs rendez-vous, mais chaque rendez-vous est associé à un seul patient.

Relation Medecin - RendezVous

Un médecin peut avoir plusieurs rendez-vous, mais chaque rendez-vous est associé à un seul médecin.

Relation Patient - DossierMedical

Chaque patient a un unique dossier médical, et chaque dossier médical est associé à un seul patient.

Relation Departement – Medecin

Un département peut avoir plusieurs médecins, mais chaque médecin est associé à un seul département.

Relation PrescriptionMedicale - Medicament, ActeMedical, Chirurgie

Une prescription médicale peut inclure plusieurs médicaments, actes médicaux ou chirurgies, et chaque médicament, acte médical ou chirurgie peut être inclus dans plusieurs prescriptions médicales.

Relation Medecin - Compte

Chaque médecin est associé à un compte utilisateur pour l'authentification.

Relation demRendezVous - Patient

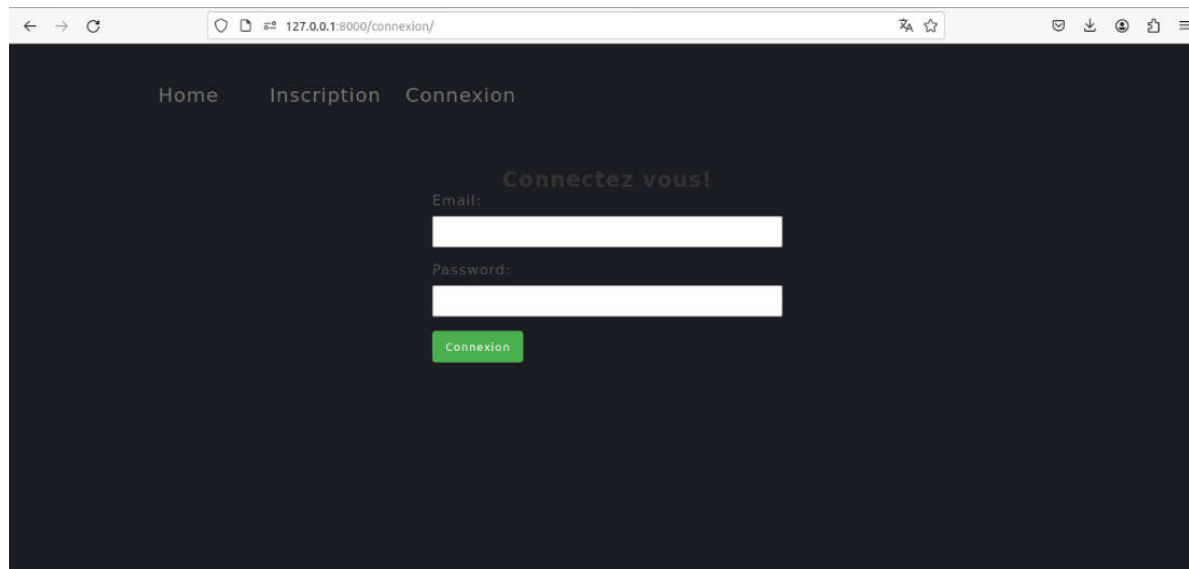
Chaque demande de rendez-vous est associée à un seul patient, mais un patient peut avoir plusieurs demandes de rendez-vous.

Ces relations permettent de modéliser les interactions complexes entre les entités du système de gestion médicale, facilitant ainsi la récupération et la manipulation des données de manière efficace et cohérente. Ces associations sont définies à l'aide des champs ForeignKey, OneToOneField et ManyToManyField dans les modèles Django.

Utilisation

Voici quelques captures qui démontrent la structure de la page avec une explication sur comment l'utiliser :

CONNEXION PATIENT



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/connexion/'. The page has a dark blue background. At the top, there is a navigation bar with three links: 'Home', 'Inscription', and 'Connexion'. Below this, the heading 'Connectez vous!' is centered. Under the heading, there are two input fields: 'Email:' and 'Password:'. Below the 'Password:' field is a green button labeled 'Connexion'.

Figure 2: Connexion Patient

Pour qu'un patient qui est déjà inscrit sur notre site puisse se connecter il appui sur **Connexion** trouve dans la page d'accueil et insère son email et mot de passe et appui sur le bouton **Connexion**.

DASHBOARD DU MEDECIN

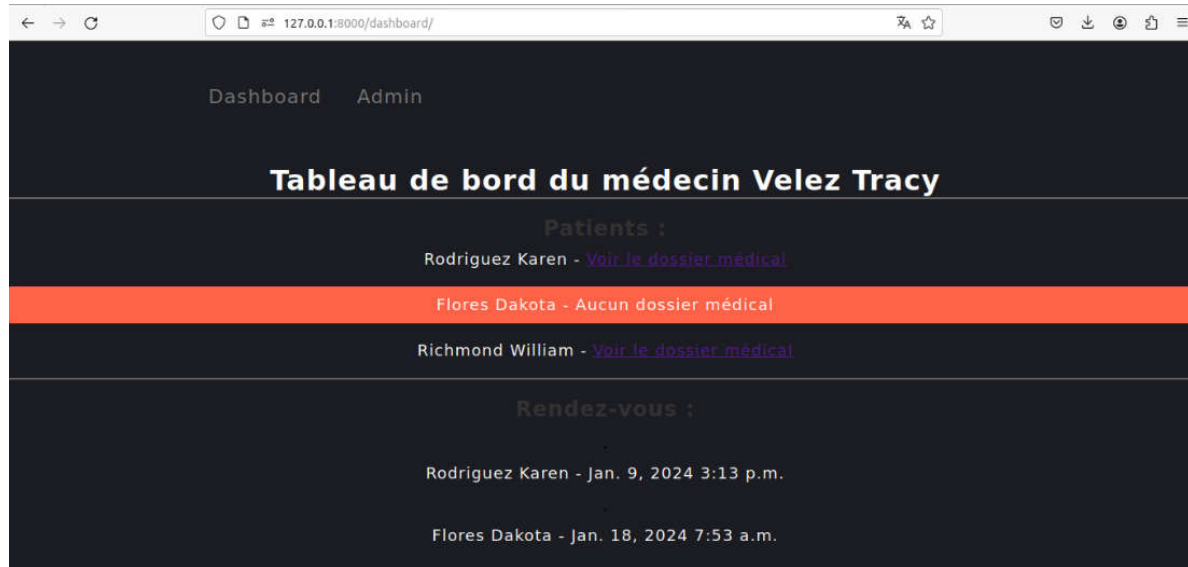


Figure 3: Dashboard Médecin

Après la connexion du médecin, il peut voir comme indiqué la liste des patients avec la possibilité de consulter leur dossier médical.

Il peut aussi consulter sa liste des rendez-vous.

DOSSIER MEDICAL

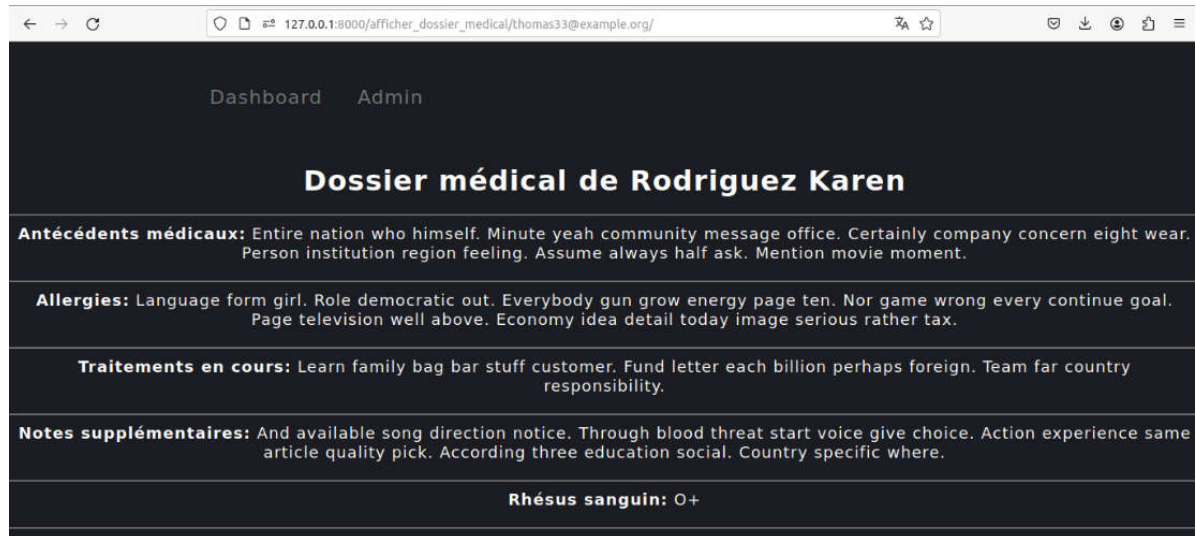


Figure 4:Dossier Medical

On appuyant sur **voir le dossier médical**, une fenêtre s'ouvre affichant les différents champs du dossier médical du patient.

ADMIN MEDECIN

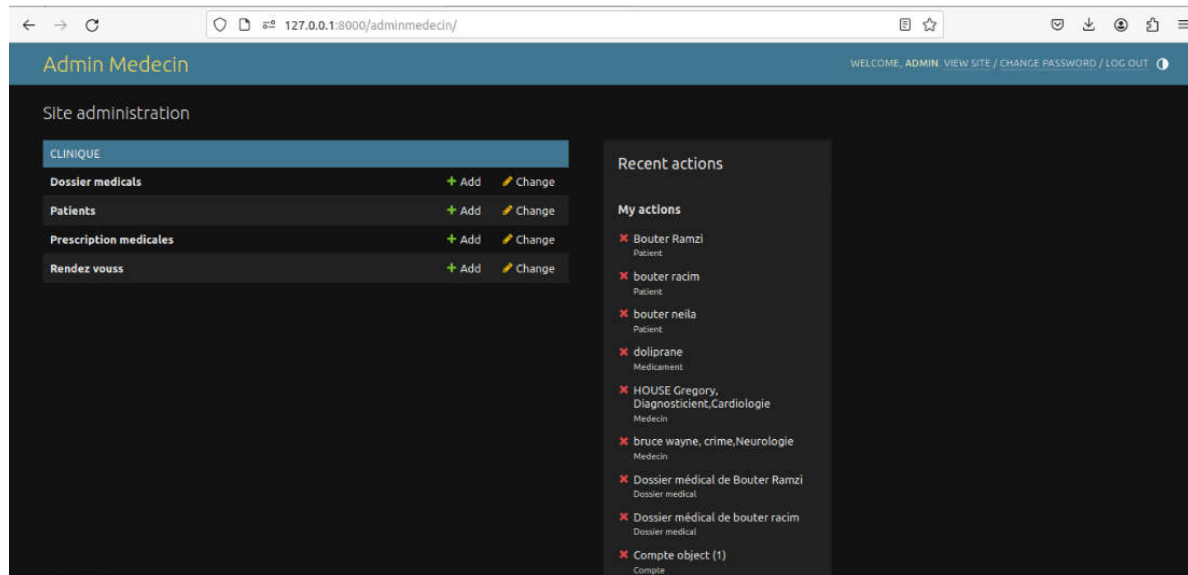


Figure 5: Admin Medecin

Quand un médecin se connecte, il a la possibilité de gérer ses patients et leur dossier médicaux, rendez-vous, et des prescription médicales.

Conclusion

En résumé, le projet aspire à fournir une solution complète et intégrée qui améliore l'efficacité des processus médicaux tout en offrant une expérience utilisateur positive aux professionnels de la santé et aux patients.