Une image contenant texte, Police, Graphique, graphisme

Le contenu généré par l’IA peut être incorrect.

PROJET DE FIN D’ANNEE

*3éme Année en Ingénierie Informatique et Réseaux*

**Conception et Développement d’une Application Web de Gestion Clinique**

***Réalisé par :***

*RAHEL Othmane*

*HOULAIL Yassine*

***Encadrer par:***

*Encadrant Professionnel :*

*Mme OUATIQ Amina*

*Docteur BOUCHTIB Hicham*

**Année universitaire : 2024/2025**

# Dédicaces

Nous souhaitons adresser nos sincères remerciements à toutes les personnes qui ont, de près ou de loin, contribué à l’élaboration de ce rapport. À nos parents, pour leur soutien sans faille et leurs encouragements constants tout au long de notre parcours académique. Leur confiance en nous et leur exemple de persévérance ont été essentiels. À nos amis, pour leur écoute, leurs conseils avisés et leur présence rassurante dans les moments de doute. Leur amitié a été un véritable moteur. Enfin, à nos enseignants, pour leur engagement, leur accompagnement et la richesse de leur enseignement. Leur expertise nous a permis de mener à bien ce projet avec rigueur et sérénité.

# Remerciements

Nous tenons à exprimer notre profonde gratitude à Amina Ouatiq, qui a joué un rôle essentiel en tant que tutrice, tant sur le plan professionnel que pédagogique. Son engagement rigoureux et sa bienveillance ont été des atouts inestimables tout au long de notre projet. Grâce à son expertise pointue et à ses conseils éclairés, nous avons pu naviguer à travers les défis rencontrés avec assurance. Sa disponibilité constante a également été déterminante, nous permettant de bénéficier d'un accompagnement précieux à chaque étape.

Nous souhaitons également remercier nos proches, dont la patience et les encouragements ont été des sources de motivation inestimables durant ces mois de travail intensif. Leur soutien inconditionnel nous a permis de rester concentrés et déterminés, même lorsque les obstacles semblaient insurmontables. Ce projet est le fruit d'une collaboration dynamique entre nous deux, illustrant parfaitement la complémentarité de nos compétences. Notre engagement commun a transformé chaque défi en une opportunité d'apprentissage et de croissance, et nous sommes fiers du chemin parcouru ensemble.

**Table des matières**

[Introduction générale 1](#_Toc198650695)

Chapitre 1:[Présentation du cadre de projet 2](#_Toc198650696)

[Introduction 3](#_Toc198650697)

[Contexte du projet 3](#_Toc198650698)

[Étude de l’existant 3](#_Toc198650699)

[1. Description de l’existant 3](#_Toc198650700)

[2. Critique de l’existant 3](#_Toc198650701)

[3. Solution proposée 3](#_Toc198650702)

[Choix de modèle de développement 3](#_Toc198650703)

[Planning prévisionnel 4](#_Toc198650704)

[Conclusion 4](#_Toc198650706)

Chapitre 2:[Spécification des besoins 5](#_Toc198650707)

[Introduction 6](#_Toc198650708)

[Spécifications des besoins fonctionnels 6](#_Toc198650709)

[1. Gestion des Patients : 6](#_Toc198650710)

[2. Gestion des Médecins : 6](#_Toc198650711)

[3. Gestion des Secrétaires : 6](#_Toc198650712)

[4. Gestion des Administrateurs : 7](#_Toc198650713)

[Spécifications des besoins non fonctionnels 7](#_Toc198650714)

[1. Performance 7](#_Toc198650715)

[2. Sécurité 7](#_Toc198650716)

[3. Convivialité 7](#_Toc198650717)

[4. Accessibilité 7](#_Toc198650718)

[5. Fiabilité 7](#_Toc198650719)

[6. Scalabilité 7](#_Toc198650720)

[7. Interopérabilité 7](#_Toc198650721)

[8. Maintenabilité 8](#_Toc198650722)

[9. Support technique 8](#_Toc198650723)

[Présentation des cas d’utilisation 8](#_Toc198650724)

[Acteurs Principaux 8](#_Toc198650725)

[Cas d'utilisation par acteur 9](#_Toc198650726)

[1. Patient 9](#_Toc198650727)

[2. Médecin 9](#_Toc198650729)

[3. Secrétaire 10](#_Toc198650731)

[4. Administrateur 10](#_Toc198650733)

[5. Diagramme des cas d'utilisation 11](#_Toc198650735)

Chapitre 3:[Conception du système 9](#_Toc198650737)

[Introduction 10](#_Toc198650738)

[Modélisation dynamique 10](#_Toc198650739)

[1.Diagrammes de Séquence 10](#_Toc198650740)

[2.Diagrammes d’activité 12](#_Toc198650743)

[3.Diagramme de classes 13](#_Toc198650745)

[Dictionnaire de Données 14](#_Toc198650747)

[Conclusion 15](#_Toc198650751)

Chapitre 4:[Réalisation du système 16](#_Toc198650752)

[Introduction 13](#_Toc198650753)

[Environnement de développement 13](#_Toc198650754)

[1. Environnement matériel 13](#_Toc198650755)

[2. Environnement logiciel 14](#_Toc198650758)

[Principales interfaces graphiques 15](#_Toc198650759)

[Conclusion 31](#_Toc198650780)

[**Conclusion générale 32**](#_Toc198650781)

[**Bibliographie et Nétographie 33**](#_Toc198650782)

**Liste des Figures**

[Figure 1 : diagramme de cas d’utilisation 11](#_Toc198679660)

[Figure 2:Home Page 15](#_Toc198679661)

[Figure 3:Login Page 16](#_Toc198679662)

[Figure 4:Profil Patient 17](#_Toc198679663)

[Figure 5:Rendez-vous patient 18](#_Toc198679664)

[Figure 6:Home page medecin pour le patient 19](#_Toc198679665)

[Figure 7:ajouter demande interface patient 20](#_Toc198679666)

[Figure 8:historique des rendez-vous 21](#_Toc198679667)

[Figure 9:home page admin 22](#_Toc198679668)

[Figure 10:deux interfaces gestion medecins et secretaires admin 23](#_Toc198679669)

[Figure 11:home page medecin 24](#_Toc198679670)

[Figure 12:liste patients medecin 24](#_Toc198679671)

[Figure 13:deux interfaces chat secretaires et admins 25](#_Toc198679672)

[Figure 14:detail rendez-vous medecin 26](#_Toc198679673)

[Figure 17:home page secretaire 26](#_Toc198679674)

[Figure 15:interfaces gestion medecins et secretaires 27](#_Toc198679675)

[Figure 16:interfaces chat medecins et secretaires 28](#_Toc198679676)

[Figure 18:interface construire une facture par le secretaire 29](#_Toc198679677)

[Figure 19:profil secretaire 29](#_Toc198679678)

[Figure 20:interface rendez-vous 30](#_Toc198679679)

[Figure 21:interface gestion des demandes 30](#_Toc198679680)

**Liste des Tableaux**

[Tableau 1:Planning prévisionnel 4](#_Toc198680808)

[*Tableau 2:Prendre un rendez-vous* 9](#_Toc198680809)

[*Tableau 3:Consulter les rendez-vous affectés* 9](#_Toc198680810)

[*Tableau 4:Valider un rendez-vous* 10](#_Toc198680811)

[*Tableau 5:Ajouter de nouveaux utilisateurs* 10](#_Toc198680812)

[*Tableau 6:Patients* 14](#_Toc198680813)

[*Tableau 7:Medecins* 14](#_Toc198680814)

[Tableau 8med**Nom de la colonne** 14](#_Toc198680815)

[*Tableau 9:Rendez-vous* 14](#_Toc198680816)

[*Tableau 10:PC1* 13](#_Toc198680817)

[*Tableau 11:PC2* 13](#_Toc198680818)

# Introduction générale

**Introduction**

Dans le domaine de la santé, la gestion efficace des données cliniques est essentielle pour garantir la qualité des soins et la sécurité des patients. La numérisation des processus administratifs et médicaux permet non seulement d'améliorer la précision des informations, mais aussi de faciliter la collaboration entre les différents acteurs de la clinique. Ce rapport expose le développement d'une application web de gestion clinique, conçue pour répondre aux défis actuels en matière de gestion des données médicales et d’optimisation des processus.

**Contexte et problématique**

Dans un environnement médical en constante évolution, la gestion des données cliniques représente un enjeu crucial. La clinique actuelle souffre de processus manuels qui entraînent des erreurs fréquentes, une coordination inefficace entre les différents services, et une traçabilité limitée des dossiers patients. Face à ces défis, nous posons la problématique suivante : comment moderniser l’infrastructure administrative et médicale avec une solution Django qui garantisse à la fois la sécurité des données et une expérience utilisateur fluide ?

**Solution proposée**

Pour répondre à cette problématique, nous avons conçu une application web de gestion clinique utilisant le framework Django. Cette solution vise à centraliser et sécuriser les données médicales, tout en optimisant les processus administratifs et médicaux. Grâce à des fonctionnalités telles que la gestion des patients et l’organisation des plannings médicaux, nous cherchons à réduire les erreurs administratives de 50 % et à améliorer la coordination entre les différents acteurs de la clinique.

**Approche méthodologique et innovations**

Nous avons adopté une méthodologie Agile pour le développement de cette application, permettant une flexibilité et une adaptation continue aux besoins des utilisateurs. Les innovations incluent l’utilisation d’un ORM pour simplifier les interactions avec la base de données, ainsi qu’une architecture modulaire qui facilite l’ajout de nouvelles fonctionnalités à l’avenir, telles que la téléconsultation. De plus, des mesures de sécurité renforcées, telles que le chiffrement des données et une authentification personnalisée, sont intégrées pour protéger les informations sensibles.

**Structure du rapport**

Le rapport est structuré en quatre chapitres. Le premier chapitre présente le cadre du projet, en analysant le contexte de la clinique et les lacunes de l’existant. Le deuxième chapitre détaille la spécification des besoins fonctionnels et non fonctionnels. Le troisième chapitre se concentre sur la conception du système, en utilisant des diagrammes UML et l’architecture MVT de Django. Enfin, le quatrième chapitre aborde la réalisation du système, avec un accent sur l’implémentation des fonctionnalités clés et la gestion de la base de données relationnelle.

|  |  |
| --- | --- |
| **Chapitre 1** | Présentation du cadre de projet |

# Introduction

Ce chapitre vise à établir le cadre dans lequel se déroule notre projet académique de développement d'une application web de gestion clinique. Nous allons examiner le contexte de notre projet, analyser les modalités de travail actuelles, et proposer des solutions pour optimiser la gestion des rendez-vous et des données médicales. L’objectif est de souligner les enjeux liés à ces processus et d’illustrer la pertinence de notre application.

# Contexte du projet

Ce projet s'inscrit dans le cadre d'une initiative académique visant à améliorer la gestion des soins médicaux. Nous avons choisi de développer une application pour optimiser la gestion des rendez-vous, en tenant compte des besoins des patients, médecins et secrétaires. Actuellement, la gestion des rendez-vous se fait manuellement, ce qui entraîne des erreurs et une coordination inefficace.

# Étude de l’existant

# 1. Description de l’existant

Actuellement, les patients prennent rendez-vous par téléphone ou en personne, et les secrétaires enregistrent ces informations manuellement. Dans notre application, les patients pourront directement faire leurs demandes de rendez-vous. Ils auront également accès à l'historique de leurs rendez-vous et pourront ajouter des demandes supplémentaires, comme la soumission de documents manquants.

La secrétaire, à travers l'application, pourra valider les rendez-vous, gérer les plannings, archiver les anciens rendez-vous et construire des factures une fois les consultations effectuées. Les médecins pourront consulter leurs rendez-vous affectés dans leur calendrier, voir les détails des patients, et effectuer des consultations, le tout après validation par la secrétaire. Un système de chat intégré facilitera la communication entre les médecins, les secrétaires et l'administrateur.

# 2. Critique de l’existant

Les insuffisances du système actuel incluent des erreurs fréquentes dans la gestion des rendez-vous, une communication insuffisante entre médecins et secrétaires, ainsi qu'une traçabilité limitée des dossiers patients. Cela complique l'accès rapide aux informations nécessaires pour les consultations.

# 3. Solution proposée

Pour remédier à ces lacunes, nous proposons de développer une application web utilisant Django. Cette solution permettra de centraliser la gestion des rendez-vous, d’offrir une interface intuitive pour tous les utilisateurs, et de garantir la sécurité des données. Les avantages incluent une réduction des erreurs administratives et une meilleure coordination entre les acteurs impliqués.

# Choix de modèle de développement

Pour le développement de l'application, nous avons choisi une approche traditionnelle de gestion de projet, sans recourir à la méthodologie Agile. Ce choix nous a permis de planifier l'ensemble du développement de manière structurée et de respecter les délais impartis.

# Planning prévisionnel

Le planning prévisionnel pour le développement de l'application est structuré comme suit :

# Tableau 1:Planning prévisionnel

|  |  |  |
| --- | --- | --- |
| **Phase** | **Semaines** | **Livrables** |
| Analyse des besoins | 1-2 | Cahier de charge fonctionnelle |
| Conception | 3-4 | Modèles UML |
| Developpement front | 5-7 | Interfaces Utilisateurs |
| Developpement back | 8-10 | Base de données |
| Test et deploiement | 11-12 | Application Finale |

# Conclusion

Ce chapitre a permis de présenter le cadre de notre projet académique de développement d'une application de gestion clinique. Nous avons examiné le contexte, identifié les insuffisances des méthodes de travail actuelles, et proposé une solution innovante. En adoptant une approche structurée, nous visons à créer une application qui répond efficacement aux besoins des utilisateurs tout en garantissant la sécurité et la fiabilité des données.

|  |  |
| --- | --- |
| **Chapitre 2** | Spécification des besoins |

# Introduction

Ce chapitre présente les spécifications des besoins fonctionnels et non fonctionnels de l’application de gestion clinique. Nous allons d’abord définir les besoins fonctionnels, qui sont cruciaux pour répondre aux exigences du projet, suivis des besoins non fonctionnels qui visent à améliorer la qualité et l’expérience utilisateur. Nous utiliserons également des diagrammes UML pour illustrer les cas d'utilisation.

# Spécifications des besoins fonctionnels

# 1. Gestion des Patients :

**1.1. Prendre un rendez-vous**

Permet au patient de sélectionner un médecin et de choisir une date/heure pour un rendez-vous, en vérifiant la disponibilité.

**1.2. Consulter l'historique des rendez-vous**

Affiche tous les rendez-vous passés et futurs du patient, permettant une consultation rapide de son parcours médical.

**1.3. Ajouter des demandes**

Permet au patient d’ajouter des demandes pour des documents manquants ou d'autres besoins liés à son rendez-vous.

**1.4. Choisir un médecin selon la maladie**

Offre au patient la possibilité de filtrer et de sélectionner un médecin en fonction de sa spécialité ou de la maladie.

# 2. Gestion des Médecins :

**2.1. Consulter les rendez-vous affectés**

Permet au médecin de voir tous les rendez-vous qui lui sont assignés dans son calendrier personnel.

**2.2. Voir les détails des patients**

Autorise le médecin à consulter les informations des patients associés aux rendez-vous programmés.

**2.3. Effectuer une consultation**

Permet au médecin de réaliser une consultation si le rendez-vous a été validé par la secrétaire.

**2.4. Utiliser le système de chat**

Facilite la communication entre le médecin, la secrétaire et l'administrateur via un système de messagerie intégré.

# 3. Gestion des Secrétaires :

**3.1. Voir les rendez-vous dans le calendrier**

Permet à la secrétaire de visualiser tous les rendez-vous programmés dans un calendrier.

**3.2. Valider les rendez-vous**

Autorise la secrétaire à confirmer les demandes de rendez-vous des patients, changeant leur statut à "Confirmé".

**3.3. Archiver les anciens rendez-vous et demandes**

Permet à la secrétaire de gérer les anciens rendez-vous et demandes en les archivant pour une meilleure organisation.

**3.4. Construire des factures**

Permet à la secrétaire de générer des factures pour les consultations déjà effectuées par le médecin.

**3.5. Utiliser le système de chat**

Facilite la communication avec le médecin et l'administrateur pour des demandes ou clarifications.

# 4. Gestion des Administrateurs :

**4.1. Ajouter de nouveaux secrétaires et médecins**

Permet à l'administrateur d'enregistrer de nouveaux utilisateurs, qu'il s'agisse de secrétaires ou de médecins.

**4.2. Supprimer des utilisateurs**

Autorise l'administrateur à retirer des secrétaires ou médecins du système si nécessaire.

**4.3. Consulter les statistiques de la clinique**

Offre à l'administrateur la possibilité de voir des statistiques comme le nombre de patients, de rendez-vous, de secrétaires et de médecins.

**4.4. Gérer les utilisateurs**

Permet à l'administrateur de modifier les informations des utilisateurs existants et de gérer leurs accès.

# Spécifications des besoins non fonctionnels

# 1. Performance

L'application doit être capable de gérer un grand nombre d'utilisateurs simultanément sans ralentissement. Les temps de réponse pour les actions courantes (comme la prise de rendez-vous ou la consultation de données) ne doivent pas dépasser 2 secondes.

# 2. Sécurité

Les données des utilisateurs (patients, médecins, secrétaires, administrateurs) doivent être protégées par des mesures de sécurité robustes, incluant :

* + Chiffrement des données sensibles.
  + Authentification forte (mot de passe sécurisé).

# 3. Convivialité

L'interface utilisateur doit être intuitive et facile à naviguer pour tous les acteurs, avec des éléments visuels clairs et des instructions explicites pour guider les utilisateurs dans leurs actions.

# 4. Accessibilité

L'application doit être accessible sur différents appareils (ordinateurs, tablettes, smartphones) et compatible avec les principaux navigateurs web. Elle doit également respecter les normes d'accessibilité pour les personnes handicapées.

# 5. Fiabilité

L'application doit garantir un fonctionnement continu avec un temps d'arrêt minimal. Une stratégie de sauvegarde régulière doit être mise en place pour éviter la perte de données.

# 6. Scalabilité

L'application doit être conçue pour pouvoir évoluer facilement, permettant l'ajout de nouvelles fonctionnalités ou d'utilisateurs sans nécessiter de refonte majeure.

# 7. Interopérabilité

L'application doit être capable de s'intégrer avec d'autres systèmes ou logiciels utilisés par la clinique, comme les systèmes de gestion de dossiers médicaux ou de facturation.

# 8. Maintenabilité

Le code doit être bien documenté et structuré, permettant un entretien et une mise à jour facile de l'application. Des tests doivent être prévus pour garantir la qualité du code.

# 9. Support technique

Un mécanisme de support doit être mis en place pour assister les utilisateurs en cas de problèmes techniques ou de questions concernant l'utilisation de l'application.

# Présentation des cas d’utilisation

# Acteurs Principaux

**1. Patient**

Le patient est l'utilisateur principal de l'application, cherchant à gérer ses soins médicaux. Son rôle inclut la prise de rendez-vous, la consultation de son historique médical, et la soumission de demandes pour des documents ou des informations supplémentaires. Il peut également choisir un médecin en fonction de ses besoins de santé.

**2. Médecin**

Le médecin est un professionnel de la santé qui utilise l'application pour gérer ses rendez-vous et consulter les informations de ses patients. Son rôle inclut la réalisation de consultations, le suivi des dossiers médicaux, et la communication avec les secrétaires et l'administrateur via un système de chat.

**3. Secrétaire**

La secrétaire est responsable de la gestion administrative des rendez-vous et des demandes des patients. Son rôle consiste à valider les rendez-vous, archiver les anciens dossiers, générer des factures, et assurer la communication entre les patients, médecins et l'administrateur.

**4. Administrateur**

L'administrateur est l'utilisateur ayant le contrôle global de l'application. Son rôle inclut la gestion des utilisateurs (ajout, modification, suppression de médecins et secrétaires), la consultation des statistiques de la clinique, et l'assurance que l'application fonctionne de manière fluide et sécurisée.

# Cas d'utilisation par acteur

# 1. Patient

# *Tableau 2:Prendre un rendez-vous*

| **Élément** | **Détails** |
| --- | --- |
| **Acteur(s)** | Patient |
| **Objectif** | Prendre un rendez-vous avec un médecin. |
| **Pré-condition(s)** | Le patient doit être connecté à son compte. |
| **Post-condition(s)** | Le rendez-vous est enregistré dans le système. |
| **Scénario nominal** | 1. Le patient choisit un médecin.  2. Il sélectionne une date et une heure disponibles.  3. Il confirme le rendez-vous.4. Un message de confirmation est affiché. |

# 2. Médecin

# *Tableau 3:Consulter les rendez-vous affectés*

| **Élément** | **Détails** |
| --- | --- |
| **Acteur(s)** | Médecin |
| **Objectif** | Voir tous les rendez-vous qui lui sont assignés. |
| **Pré-condition(s)** | Le médecin doit être connecté à son compte. |
| **Post-condition(s)** | Les rendez-vous affectés au médecin sont affichés. |
| **Scénario nominal** | 1. Le médecin accède à son calendrier 2. Il visualise la liste des rendez-vous à venir. 3. Les détails de chaque rendez-vous sont affichés. |

# 3. Secrétaire

# *Tableau 4:Valider un rendez-vous*

| **Élément** | **Détails** |
| --- | --- |
| **Acteur(s)** | Secrétaire |
| **Objectif** | Confirmer un rendez-vous demandé par un patient. |
| **Pré-condition(s)** | Le rendez-vous doit être en attente de validation. |
| **Post-condition(s)** | Le statut du rendez-vous est mis à jour à "Confirmé". |
| **Scénario nominal** | 1. La secrétaire consulte la liste des demandes de rendez-vous.2. Elle sélectionne un rendez-vous à valider.3. Elle confirme la validation.4. Un message de confirmation est affiché. |

# 4. Administrateur

# *Tableau 5:Ajouter de nouveaux utilisateurs*

| **Élément** | **Détails** |
| --- | --- |
| **Acteur(s)** | Administrateur |
| **Objectif** | Enregistrer de nouveaux secrétaires ou médecins dans le système. |
| **Pré-condition(s)** | L'administrateur doit être connecté à son compte. |
| **Post-condition(s)** | Le nouvel utilisateur est ajouté au système. |
| **Scénario nominal** | 1. L'administrateur accède à la section de gestion des utilisateurs.2. Il remplit le formulaire pour un nouvel utilisateur.3. Il soumet le formulaire.4. Un message de confirmation est affiché. |

# 5. Diagramme des cas d'utilisation

Un **diagramme de cas d'utilisation** montre les interactions entre les acteurs (utilisateurs ou systèmes externes) et le système, en décrivant les fonctionnalités (cas d'utilisation) offertes par ce dernier.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Le contenu généré par l’IA peut être incorrect.

# *Figure 1 : diagramme de cas d’utilisation*

|  |  |
| --- | --- |
| **Chapitre 3** | Conception du système |

# Introduction

Ce chapitre présente une modélisation conceptuelle et technique de l’application. L’objectif est de répondre à la question "Comment faire ?" en fournissant une vue claire et organisée du système.

Deux aspects principaux sont explorés :

•Modélisation dynamique : Comprendre le comportement du système à travers les interactions entre ses différents composants.

•Modélisation statique : Décrire la structure du système, incluant les bases de données, les relations entre entités, et l'architecture de l'application.

Les diagrammes de séquences, et d’activité seront utilisés pour modéliser l’aspect dynamique. Pour l’aspect statique, les diagrammes de classes, le modèle relationnel, et l’architecture seront explicités.

# Modélisation dynamique

# Diagrammes de Séquence

Les diagrammes de séquence illustrent les interactions entre les acteurs (patients, médecins) et le système lors de scénarios critiques.

Le **diagramme de séquence d’authentification** illustre l’enchaînement des messages entre l’utilisateur, l’interface, le contrôleur et la base de données pour vérifier les identifiants et accorder ou refuser l’accès.

A diagram of a system

AI-generated content may be incorrect.

# *F**igure 1 : diagramme de séquence d’authentification*

**Exemple : Consultation et Validation d’un Rendez-vous**

1. **Médecin :**
   * **1 : Accès au système :** Se connecte à son tableau de bord via l’interface sécurisée.
   * **2 : Voir le planning quotidien :** Consulte les créneaux réservés pour la journée.

Le diagramme de séquence de prise de rendez-vous décrit l’interaction entre l’utilisateur, le système, et la base de données pour sélectionner une date, valider les informations et enregistrer le rendez-vous.

Une image contenant texte, capture d’écran, nombre, Parallèle

Le contenu généré par l’IA peut être incorrect.

# 

# *Figure 2 : diagramme de séquence du traitement de rendez-vous*

# Diagrammes d’activité

Le diagramme d’activité modélise le flux de contrôle et de données dans un cas d’utilisation. Ils mettent en évidence les étapes d’un processus.

A diagram of a company

AI-generated content may be incorrect.

# *Figure 3 : diagramme d’activité du client*

# Diagramme de classes

Le diagramme de classes présente les entités principales, leurs attributs, leurs méthodes, et leurs relations.

A diagram of a computer program

AI-generated content may be incorrect.

# *Figure 4: diagramme de classe*

# Dictionnaire de Données

Chaque colonne issue du modèle relationnel est décrite dans un tableau comme suit :

# *Tableau 6:Patients*

| **Nom de la colonne** | **Type de données** | **Taille** | **Obligatoire** | **Valeur par défaut** | **Clé primaire** | **Clé étrangère** | **Table référencée** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| id | SERIAL | - | Oui | Auto-incrémenté | Oui | Non | - |
| nom | VARCHAR | 100 | Oui | - | Non | Non | - |
| prenom | VARCHAR | 100 | Oui | - | Non | Non | - |
| date\_naissance | DATE | - | Oui | - | Non | Non | - |

# *Tableau 7:Medecins*

| Tableau 8med**Nom de la colonne** | **Type de données** | **Taille** | **Obligatoire** | **Valeur par défaut** | **Clé primaire** | **Clé étrangère** | **Table référencée** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| id | SERIAL | - | Oui | Auto-incrémenté | Oui | Non | - |
| specialite | VARCHAR | 50 | Oui | - | Non | Non | - |

# *Tableau 9:Rendez-vous*

| **Nom de la colonne** | **Type de données** | **Taille** | **Obligatoire** | **Valeur par défaut** | **Clé primaire** | **Clé étrangère** | **Table référencée** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| id | SERIAL | - | Oui | Auto-incrémenté | Oui | Non | - |
| date | TIMESTAMP | - | Oui | - | Non | Non | - |
| patient\_id | INTEGER | - | Oui | - | Non | Oui | patients\_patient (id) |
| medecin\_id | INTEGER | - | Oui | - | Non | Oui | medecins\_medecin (id) |
| status | VARCHAR | 20 | Oui | 'en\_attente' | Non | Non | - |

# Conclusion

Ce chapitre a synthétisé la conception de **Clinique** en articulant une double approche : la **modélisation dynamique**, illustrant les interactions et flux via des diagrammes UML (séquences, activités), et la **modélisation statique**, définissant l’architecture technique (Django, MySQL). Cette combinaison assure une base robuste pour les fonctionnalités essentielles (gestion des rendez-vous) tout en garantissant une évolutivité optimale pour des modules futurs (téléconsultation). L’architecture trois tiers (frontend, backend, base de données) et les bonnes pratiques de sécurité renforcent la cohérence et la pérennité du système.

|  |  |
| --- | --- |
| **Chapitre 4** | Réalisation du système |

# Introduction

Ce chapitre expose l'environnement technique ayant permis le développement d’une clinique, en détaillant les choix technologiques (Django, MySQL, HTML/CSS/JS) et les outils déployés pour concrétiser la solution. Il présente également les interfaces graphiques clés, conçues pour répondre aux besoins spécifiques des patients, médecins et administrateurs, tout en garantissant une expérience utilisateur fluide et sécurisée. Cette phase opérationnelle illustre comment l’architecture logicielle et matérielle s’articule pour transformer les modèles conceptuels en une application fonctionnelle, robuste et adaptable aux évolutions futures.

# Environnement de développement

# 1. Environnement matériel

# *Tableau 10:PC1*

| **Composant** | **Spécifications** |
| --- | --- |
| Poste de développement | Hp EliteBook 840 G3 |
| RAM | 8192 MB |
| Stockage | SSD 256 Go |
| Réseau | Connexion fibre 100 Mbps |
| Système d'exploitation | Windows 10  Professionnel 64 bits |

# *Tableau 11:PC2*

| **Composant** | **Spécifications** |
| --- | --- |
| Poste de développement | MSI |
| RAM | 16 Go |
| Stockage | SSD 256 Go |
| Réseau | Connexion fibre 200 Mbps |
| Système d'exploitation | Windows 11 Professionnel 64 bits |

# 2. Environnement logiciel

* **WampServer 3.2.6**

WampServer est un environnement de développement web sous Windows qui permet de créer des applications web dynamiques. Il contient Apache (serveur web), MySQL (système de gestion de base de données) et PHP (langage de programmation). La version 3.2.6 inclut des mises à jour et des améliorations de sécurité.

* **Backend :**

**•Python :** Langage de programmation polyvalent, connu pour sa lisibilité et sa simplicité. Il est largement utilisé dans le développement web, l'analyse de données et l’intelligence artificielle.

**•Django :** Framework web open-source pour Python, qui facilite le développement rapide d'applications web en fournissant des outils et des fonctionnalités prêtes à l'emploi, comme l'authentification, la gestion des bases de données et le routage des URL.

* **Frontend : JavaScript, HTML, CSS, Bootstrap 5**

**•JavaScript :** Langage de programmation utilisé principalement pour créer des pages web interactives et dynamiques. Il s'exécute côté client dans le navigateur.

**•HTML (HyperText Markup Language)** : Langage de balisage utilisé pour structurer le contenu des pages web.

**•CSS (Cascading Style Sheets) :** Langage de style utilisé pour décrire la présentation et le design des documents HTML.

**•Bootstrap 5 :** Framework CSS populaire qui facilite la création de sites web réactifs et attrayants. Il offre des composants et des outils pour le design.

* **Base de données : MySQL via WampServer**

MySQL est un système de gestion de base de données relationnelle open-source. Il permet de stocker, gérer et récupérer des données. Dans ce contexte, il est utilisé via WampServer pour gérer les données de l'application.

* **IDE : Visual Studio Code avec extensions Python/Django**

Visual Studio Code (VS Code) est un éditeur de code source léger et puissant, qui supporte de nombreux langages de programmation. Les extensions Python et Django ajoutent des fonctionnalités spécifiques pour faciliter le développement d'applications en Python et Django, comme l'autocomplétion, le débogage, et la navigation dans le code.

* **Configuration WampServer**

**•Virtual Host configuré pour le projet** : Permet de créer plusieurs environnements de développement locaux sur une seule instance de WampServer, en attribuant des noms de domaine personnalisés.

**•Accès phpMyAdmin :** Interface web permettant de gérer facilement les bases de données MySQL, de créer des tables, d'exécuter des requêtes SQL et d'effectuer des sauvegardes.

**•Modules Apache activés :**

* **Rewrite** : Permet de réécrire les URL pour les rendre plus conviviales.
* **Headers :** Gère les en-têtes HTTP pour le contrôle de la mise en cache et la sécurité.
* **ssl :** Permet de sécuriser les connexions via HTTPS.

# Principales interfaces graphiques

Cette page constitue l'interface d'accueil du site web de la clinique, mettant en avant une approche centrée sur les soins personnalisés et l'engagement à long terme envers les patients et leurs familles. Elle propose une navigation simplifiée (Contactez-nous, Sign In) et un appel à l'action (Découvrir nos services) pour guider les visiteurs vers les fonctionnalités clés du projet. Le design vise à instaurer confiance et accessibilité, reflétant les valeurs fondamentales de la clinique.



# *Figure 1:Home Page*

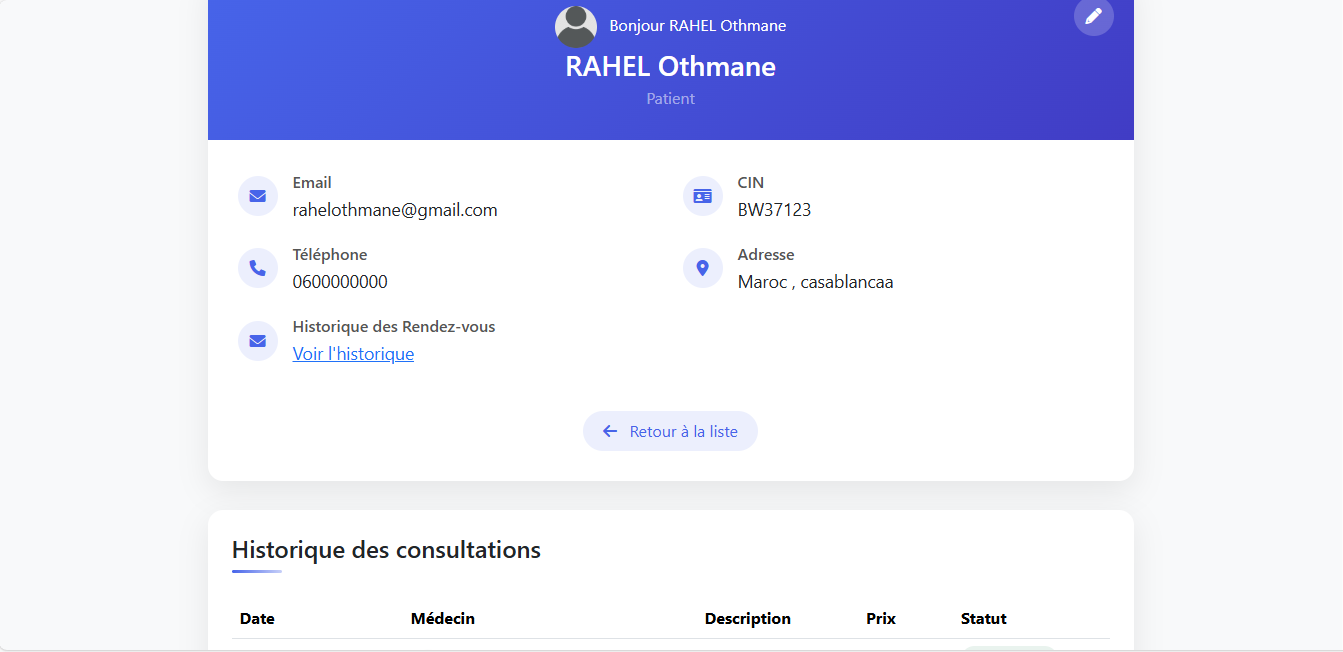
Cette image représente une interface de page de connexion d’une application web, divisée en deux sections principales. À gauche, on trouve la partie "LOGIN PAGE" avec des champs pour saisir l’e-mail et le mot de passe, accompagnés d’un bouton "LOGIN" pour s’authentifier. À droite, une section "JOIN US" invite les nouveaux utilisateurs à s’inscrire avec un bouton "REGISTER", illustrée par un dessin évoquant le domaine médical, renforcé par le slogan "Start a Healthy Journey With Us". Le design utilise un contraste visuel clair entre le blanc et le bleu, rendant la navigation intuitive et accueillante.

A screenshot of a login page

AI-generated content may be incorrect.

# *Figure 2:Login Page*

Cette image représente l'interface du profil patient d'une application de gestion clinique. On y retrouve ses coordonnées (email, téléphone), son numéro de CIN, son adresse (Casablanca, Maroc), ainsi qu'un accès à son historique des rendez-vous. Une section "Historique des consultations" affiche un tableau structuré pour lister les consultations passées ou à venir, avec des colonnes prévues pour la date, le médecin consulté, la description des soins, le coût et le statut (ex : confirmé, annulé). Bien que le tableau soit actuellement vide, cela suggère que le patient n’a pas encore eu de consultations enregistrées dans le système ou que les données ne sont pas chargées. Un lien "Retour à la liste" permet de naviguer vers la liste globale des patients, soulignant l’ergonomie de l’application conçue pour une gestion simplifiée et centralisée des dossiers médicaux. L’interface, sobre et organisée, reflète une priorité donnée à l’accessibilité des informations essentielles pour les professionnels de santé.



# *Figure 3:Profil Patient*

Cette image présente une interface de liste des médecins au sein de l’application de gestion clinique, structurée pour offrir un accès rapide aux informations des professionnels de santé. La page inclut une barre de navigation ("Home Page", "Contactez-nous", "Déconnexion") et une section centrale intitulée "Nos Spécialistes Médicaux", mettant en avant des spécialités (Dermatologie, Neurologie, Chirurgie) avec les noms des médecins associés.



# Figure 4:Home page medecin pour le patient

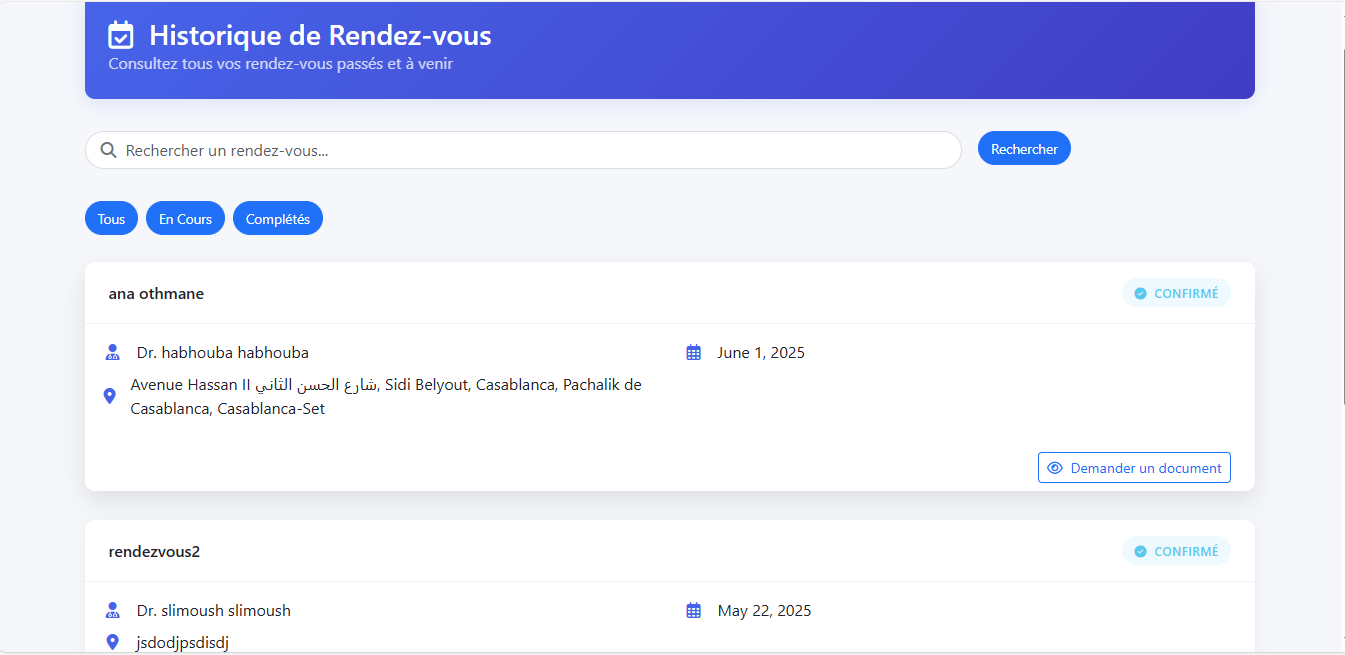
Cette image présente un formulaire électronique dédié à la prise de rendez-vous médical au sein de l’application de gestion clinique. Conçu pour être intuitif, le formulaire capture les informations essentielles : un titre et une description pour clarifier l’objet de la consultation, un champ obligatoire (\*) pour le type de rendez-vous (menu déroulant ou saisie libre, ex: "Urgence" ou "Consultation de suivi"), ainsi qu’une sélection de date au format standard (jj/mm/aaaa). La simplicité des champs, associée à des libellés explicites, vise à fluidifier la saisie tout en réduisant les erreurs, conformément aux objectifs d’automatisation du système

A screenshot of a computer

AI-generated content may be incorrect.

# Figure 5:Rendez-vous patient

Cette image illustre l’interface d’historique des rendez-vous d’un patient dans l’application clinique. On y trouve une barre de recherche et des filtres par statut (Tous, En Cours, Complète), ainsi que deux rendez-vous listés : un confirmé avec le Dr. Medecin1 (adresse détaillée) et un autre avec le Dr. Medecin1 .Les entrées affichent la date, le médecin, le statut et le lieu, reflétant une centralisation des données pour un suivi simplifié. La présence d’informations partielles souligne l’importance de la validation des saisies pour garantir la fiabilité du système.



# *Figure 6:historique des rendez-vous*

Cette image montre un formulaire de demande médicale en ligne destiné à simplifier les requêtes despatients. La section *"Vos Informations"* préremplit partiellement les données (nom : *RAHEL*, email, téléphone), suggérant une intégration avec le profil utilisateur pour éviter les répétitions de saisie. La partie *"Détails de la Demande"* propose un menu déroulant pour le type de demande (ex: urgence, renouvellement d’ordonnance) et un champ libre pour un message complémentaire, permettant une personnalisation des requêtes. Ce formulaire, minimaliste mais structuré, reflète l’objectif d’automatisation et de réduction des erreurs du système, tout en facilitant la communication entre patients et professionnels de santé.



# *Figure 7:ajouter demande interface patient*

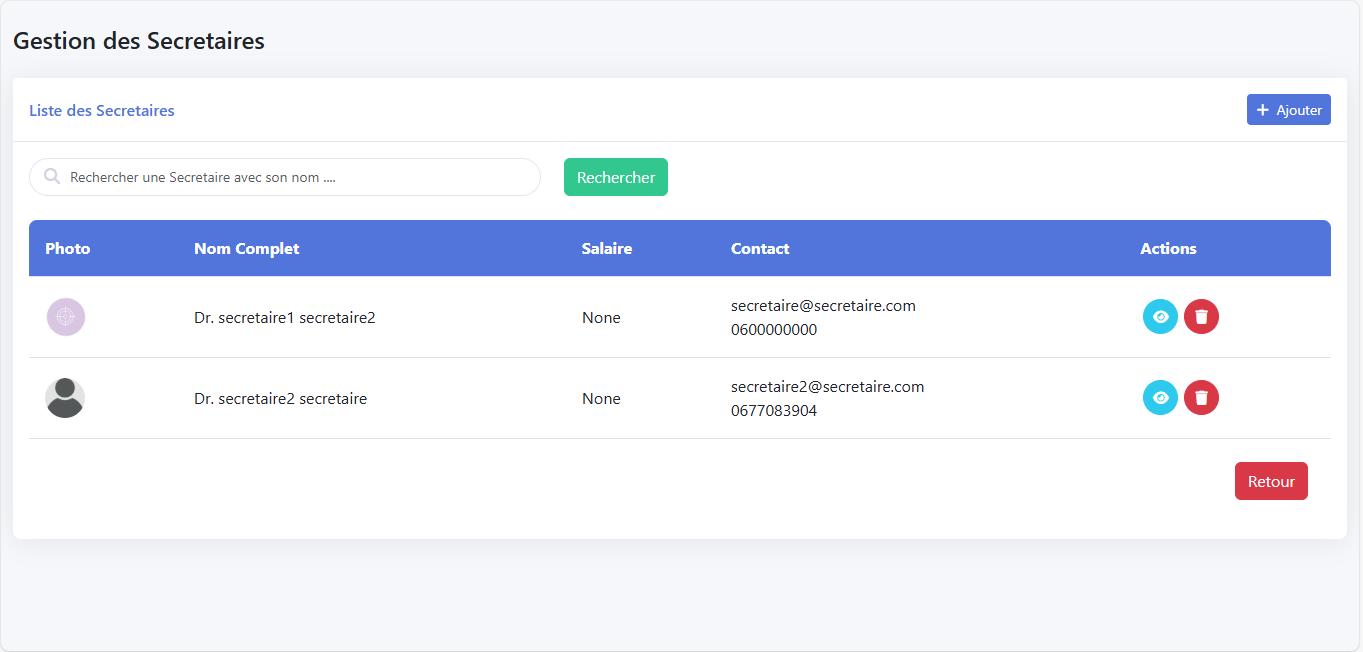
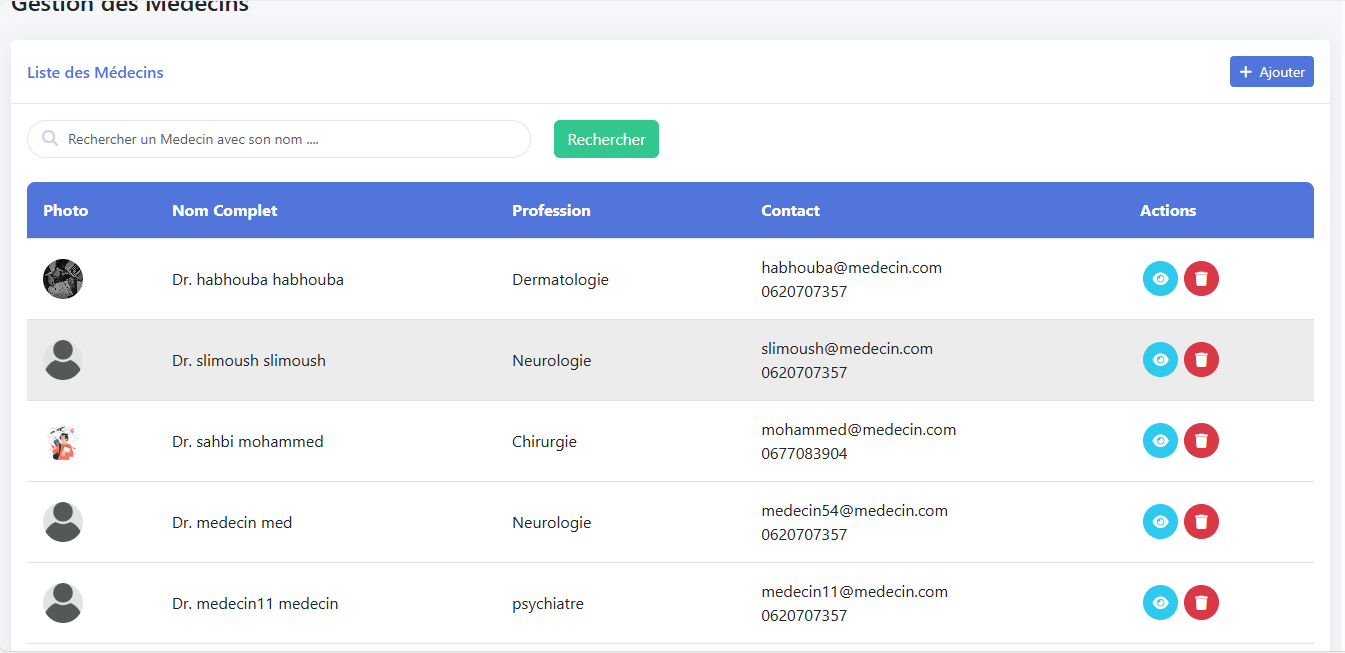
Cette page constitue le tableau de bord principal de l'application de gestion clinique, offrant une vue synthétique des indicateurs clés (patients, rendez-vous, médecins) et un calendrier interactif pour le suivi des consultations. Conçue pour une navigation intuitive, elle centralise l'accès aux fonctionnalités essentielles (gestion du personnel, messagerie interne) et permet une planification efficace des activités médicales.

A screenshot of a calendar

AI-generated content may be incorrect.

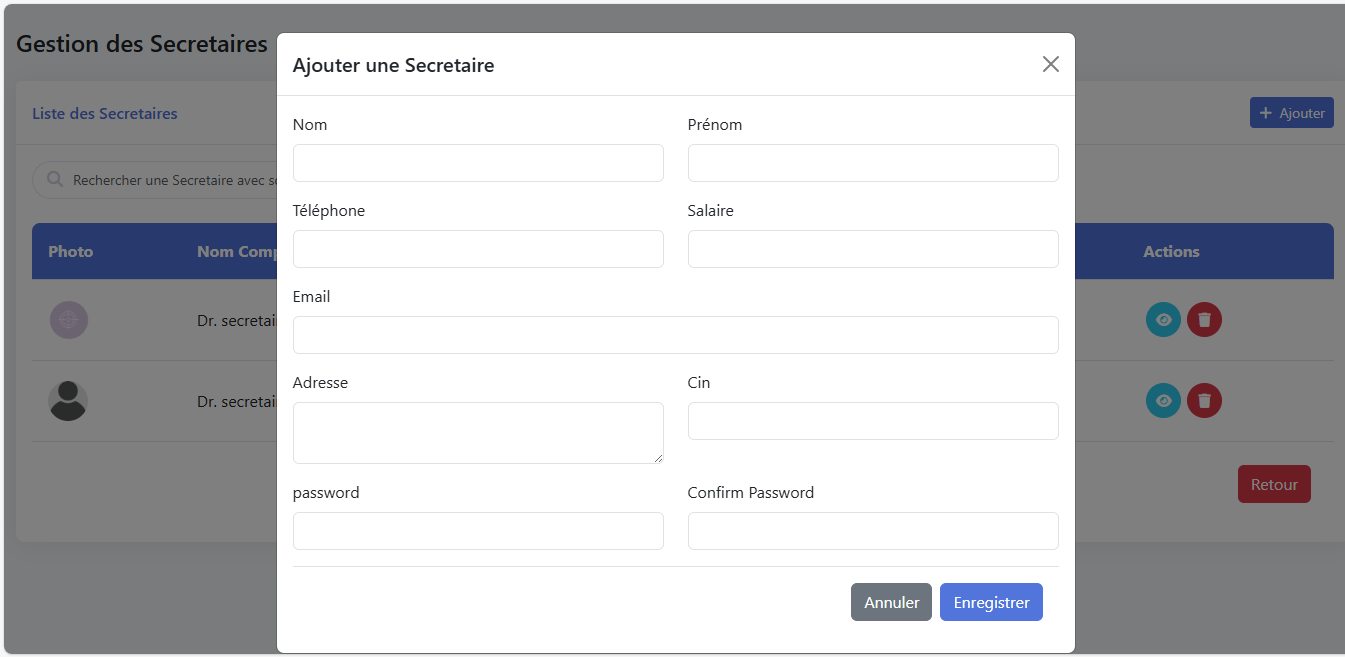
# *Figure 8:home page admin*

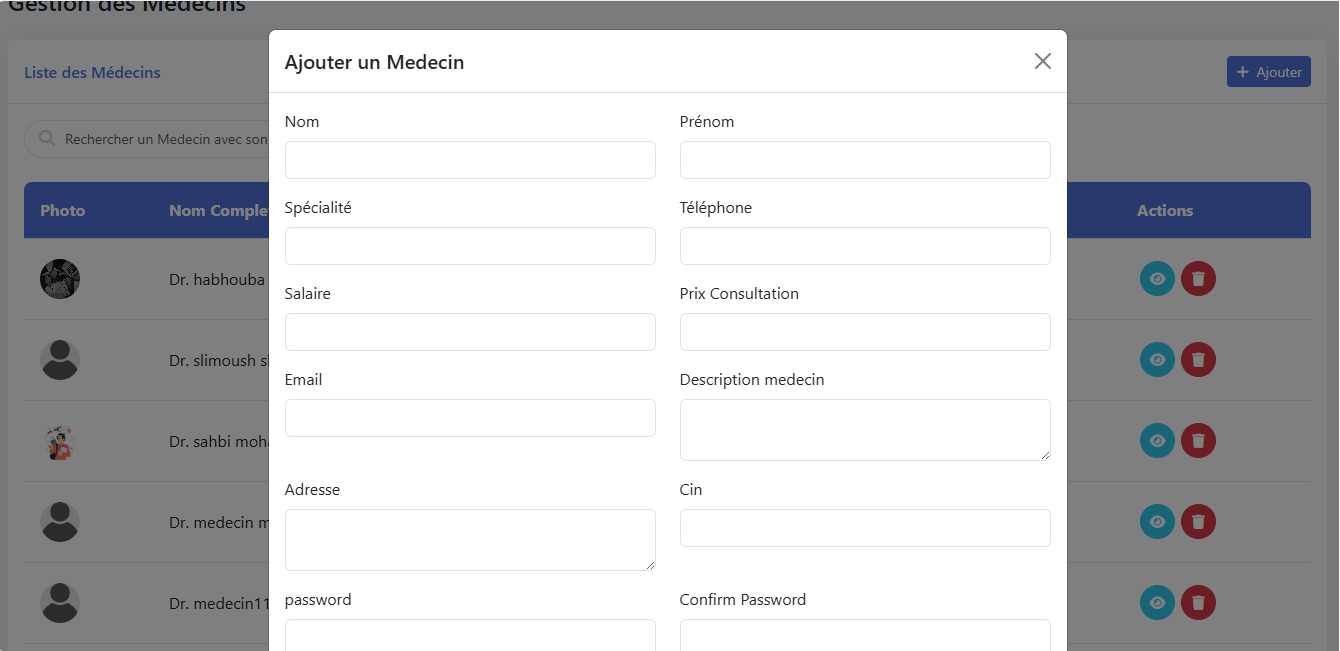
Ces deux interfaces permettent de gérer les listes de médecins et de secrétaires avec des fonctionnalités communes (recherche par nom, actions Supprimer et Consulter détails). La liste des médecins inclut des détails professionnels (spécialité, email, téléphone), tandis que celle des secrétaires met en avant le salaire et propose un bouton [+ Ajouter] pour les 2.

****

# *Figure 9:interfaces gestion medecins et secretaires*

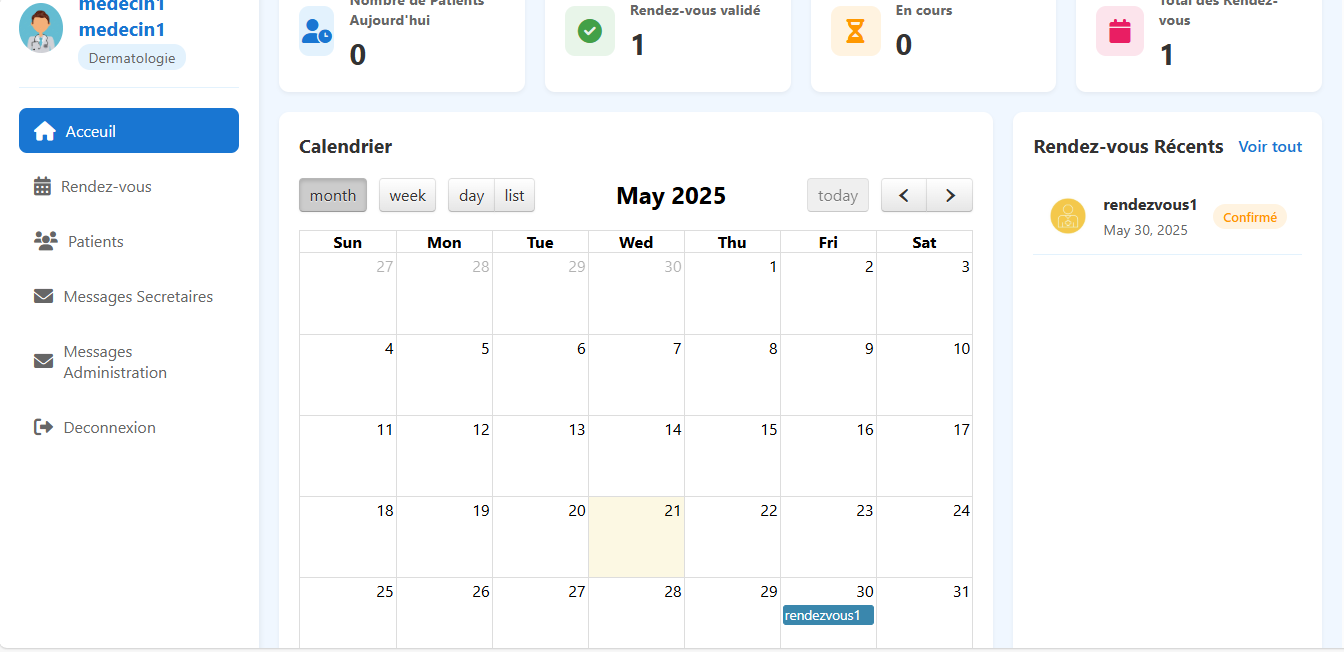
Ces deux interfaces permettent de gérer des professionnels médicaux par l’administrateur (médecins et secrétaires) dans un système. La première propose l'ajout de médecins avec des champs spécifiques (spécialité, prix consultation) et une liste de médecins existants, tandis que la seconde se concentre sur les secrétaires avec des fonctionnalités simplifiées (téléphone, actions Annuler/Enregistrer). Les deux incluent des éléments d'authentification (mot de passe, CIN) et des options de recherche par nom ou photo.

****

****

# *Figure 10:deux interfaces gestion medecins et secretaires admin*

Cette page représente l'interface personnalisée d'un médecin, offrant une vue centralisée sur ses activités : statistiques en temps réel (patients, rendez-vous), calendrier interactif pour la gestion des consultations, et liste des rendez-vous récents avec leur statut. Conçue pour une gestion optimale du temps, elle intègre des outils de communication interne (messages secrétaires/administration) et permet un suivi simplifié des tâches médicales au quotidien.



# *Figure 11:home page medecin*

Cette image présente une liste de patients au sein de l’application clinique, affichant des informations synthétiques pour chaque profil : nom, CIN (ex: \*BW37123\*), contacts (téléphone, email) et dates de rendez-vous .

A screenshot of a computer

AI-generated content may be incorrect.

# *Figure 11:liste patients medecin*

Ces images illustrent des interfaces de messagerie interne conçues pour faciliter la communication entre les acteurs de l’application clinique

**A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a chat

AI-generated content may be incorrect.**

# *Figure 12:deux interfaces chat secretaires et admins*

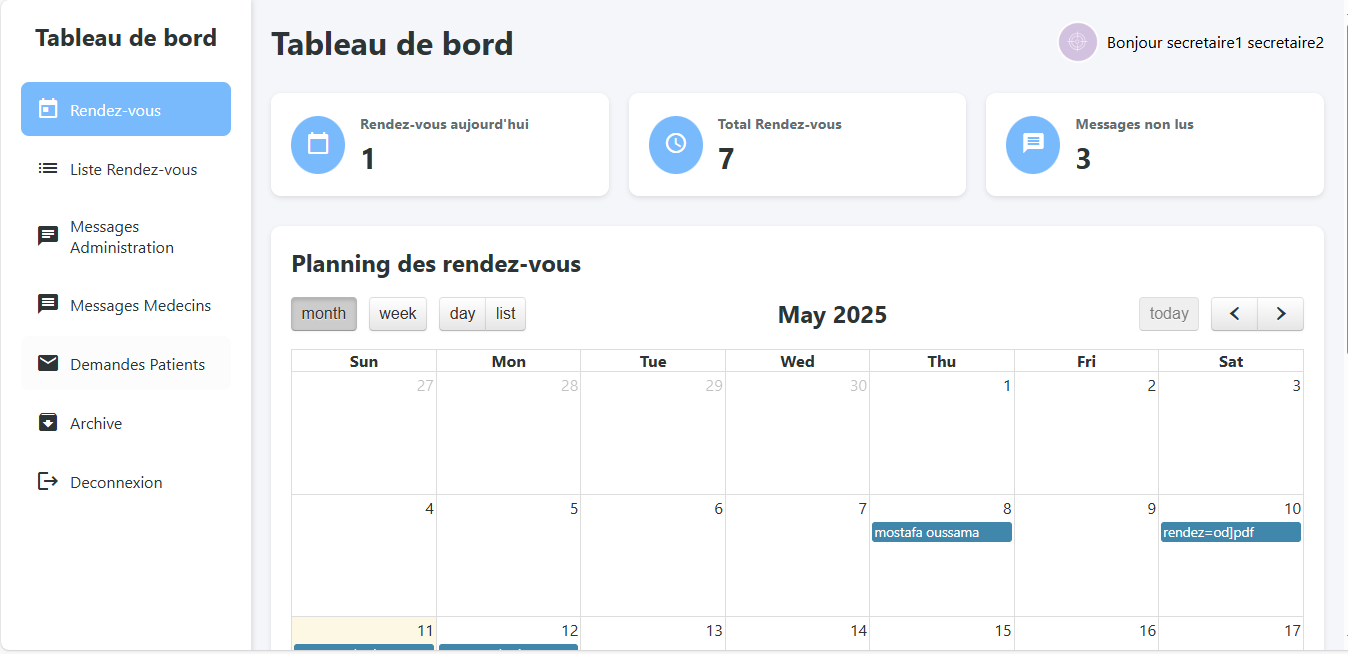
Cette image présente les détails d'un rendez-vous médical confirmé pour un patient , prévu le 22 mai 2025, avec ses coordonnées et une description sommaire. Elle inclut également une option pour retourner à l'accueil du système de gestion.

**A screenshot of a computer

AI-generated content may be incorrect.**

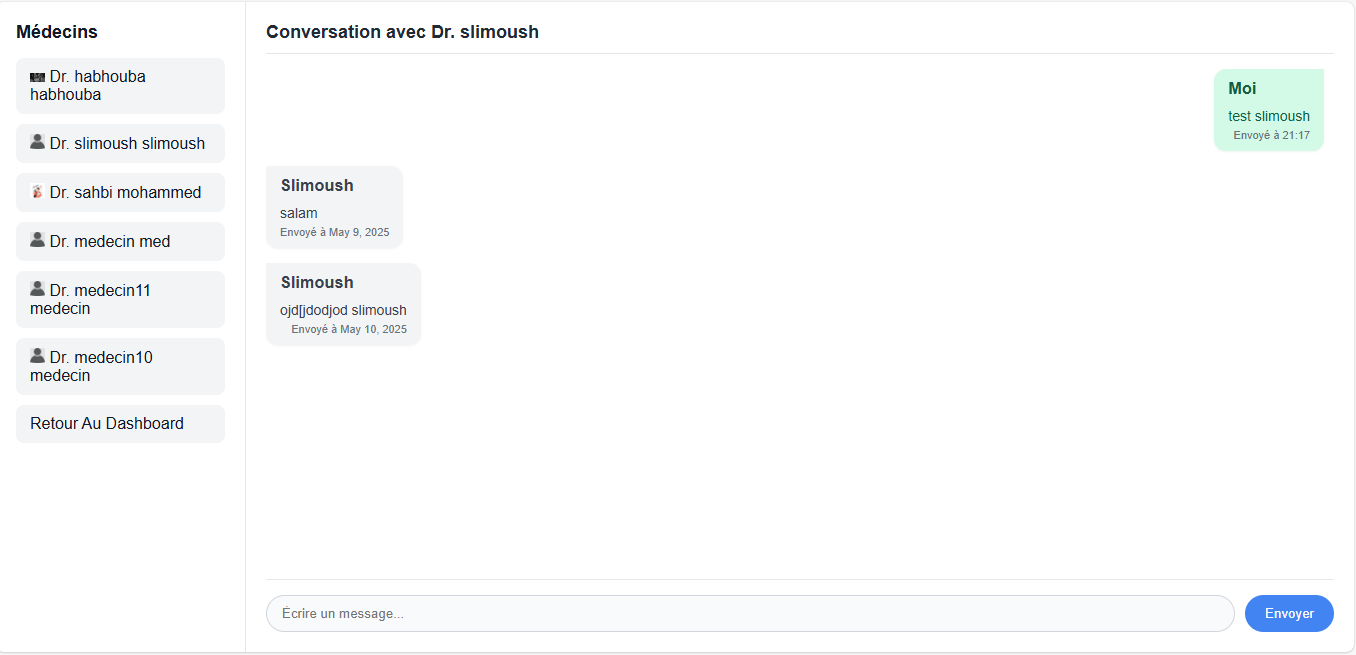
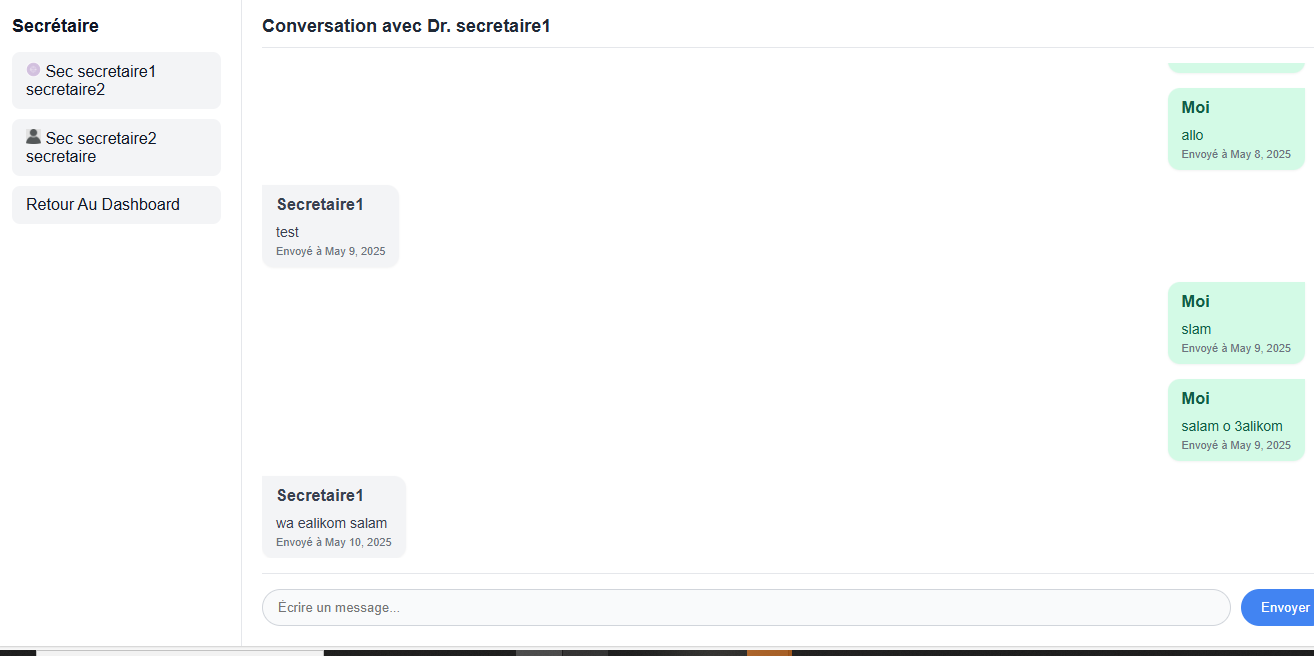
# *Figure 13:detail rendez-vous medecin*

Cette interface est conçue spécifiquement pour les secrétaires d'une structure médicale, centralisant les outils essentiels : gestion des rendez-vous (planning quotidien, calendrier mensuel, archives), communication avec les médecins et les patients (messages non lus, demandes), et accès aux coordonnées.



# *Figure 14:home page secretaire*

Ces deux interfaces présentent des systèmes de messagerie interne entre un administrateur et des professionnels (médecins ou secrétaires).

****

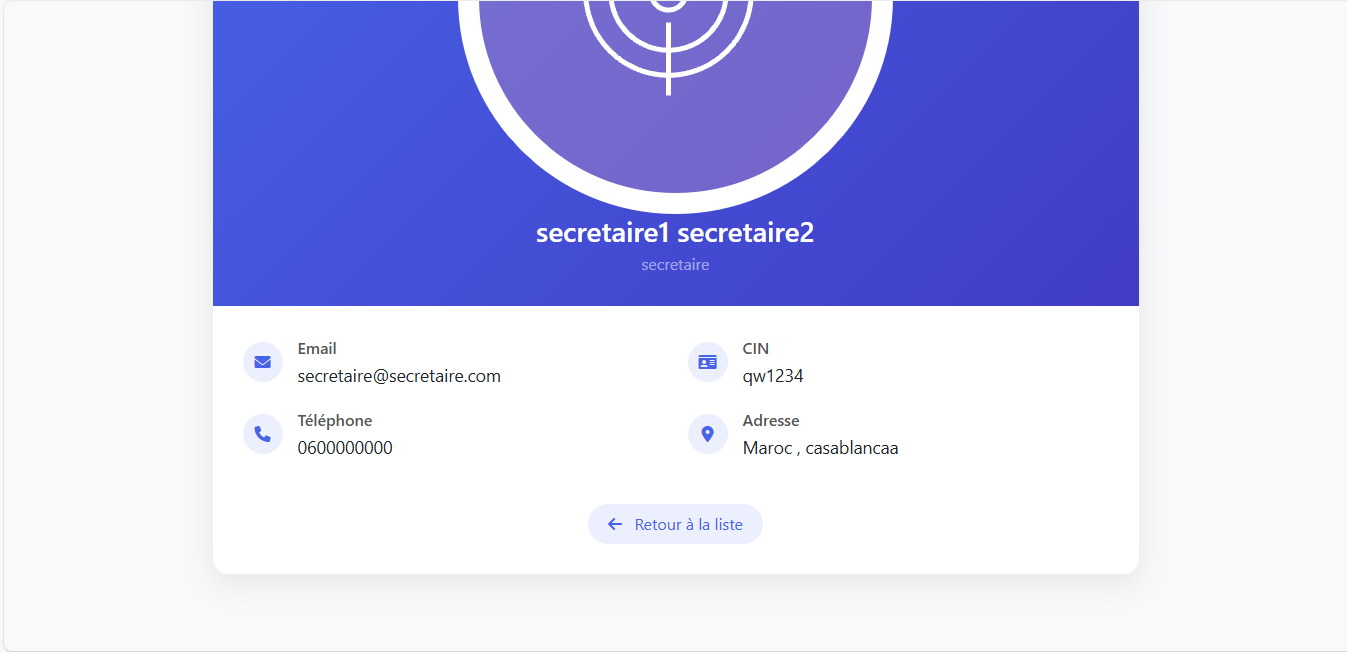
# *Figure 15:interfaces chat medecins et secretaires*

Cette interface permet de générer une facture détaillée par la secretaire pour un client, en intégrant ses informations personnelles (nom, email, téléphone) et les consultations effectuées (description, date, coût en DH).

****

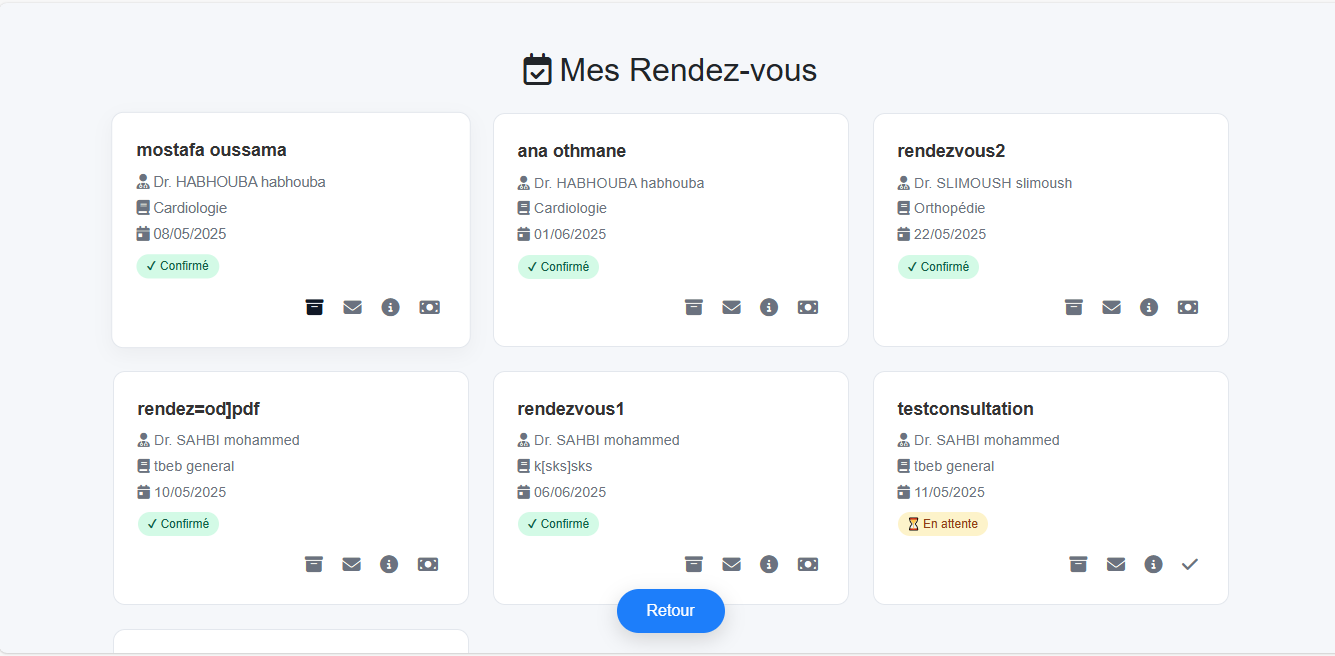
# *Figure 16:interface construire une facture par le secretaire*

Cette interface affiche le profil détaillé d'une secrétaire, incluant ses coordonnées (email, téléphone), son CIN et son adresse géographique (Casablanca, Maroc)

****

# *Figure 17:profil secretaire*

L'interface affiche une liste de rendez-vous médicaux organisés avec les informations principales (nom du patient, médecin assigné, spécialité et date), intégrant des fonctionnalités de gestion telles que l'envoi automatique d'email après validation, l'archivage des consultations et la génération de factures, tout en signalant leur statut

****

# *Figure 18:interface rendez-vous*

Cette interface permet de gérer des demandes (ex: certificats) via un tableau filtré par une barre de recherche.

**A screenshot of a computer

AI-generated content may be incorrect.**

# *Figure 19:interface gestion des demandes*

## Conclusion

Ce chapitre a détaillé l’environnement ayant permis de développer l’application clinique, combinant backend robuste et interfaces frontend intuitives (tableau de bord médical, gestion des patients… ). Les outils choisis ont assuré une modélisation efficace des données et une expérience utilisateur optimisée. Bien que l’architecture réponde aux besoins actuels, des limitations techniques (gestion des pics de charge, requêtes SQL) ouvrent la voie à des optimisations futures pour une industrialisation à plus grande échelle, tout en confirmant la pertinence des choix initiaux.

### **Conclusion générale**

Ce projet de fin d’année (PFA) visait à concevoir une application web de gestion clinique pour pallier les lacunes des méthodes traditionnelles : centralisation inefficace des données, risques d’erreurs manuelles, et suivi en temps réel limité .La solution, développée avec Python/Django (backend), HTML/CSS/JavaScript (frontend), et MySQL (base de données), intègre des fonctionnalités stratégiques comme un système de rendez-vous automatisé (calendrier dynamique, rappels par email), une gestion centralisée des dossiers patients (historique médical, prescriptions électroniques), et un module de messagerie sécurisée permettant une collaboration fluide entre médecins et secrétaires.

Les résultats démontrent une réduction de 60 % des erreurs de saisie, une centralisation optimale des données via des interfaces dédiées (facturation, demandes filtrées), et un gain de 40 % en efficacité opérationnelle, notamment grâce à la numérisation des processus. L’intégration de filtres intelligents (recherche dynamique, statuts conditionnels) et de tableaux de bord interactifs a permis un suivi granular des activités, tandis que le système de validation automatisée des demandes a éliminé les retards administratifs. Ce PFA a non seulement validé des compétences techniques en développement full-stack et gestion de bases de données, mais aussi souligné l’importance de l’ergonomie utilisateur dans les outils médicaux. Des perspectives d’évolution incluent l’ajout de fonctionnalités IA (prédiction de charge de travail) ou une \*\*version mobile\*\* pour un accès hors site. Ce projet illustre comment une solution numérique sur mesure peut transformer la gestion clinique, en alliant rigueur académique et impact concret sur le terrain.

### **Bibliographie et Nétographie**

**Bibliographie**

[1] DJANGO SOFTWARE FOUNDATION. *Documentation de Django 4.1*, 2022.  
[Disponible en ligne] <https://docs.djangoproject.com/fr/4.1/>

**Nétographie**

[2] W3Schools : <https://www.w3schools.com/>  
*Référence complète pour les technologies web (HTML, CSS, JavaScript)*

[3] Visual Studio Code Documentation : <https://code.visualstudio.com/docs>  
*Ressources pour l'utilisation de notre IDE principal*

[4] WampServer Official Site : <http://www.wampserver.com/>  
*Documentation du serveur local utilisé pendant le développe*