



## Rapport du stage de fin d'études

POUR L'OBTENTION DU DUT

FILIERE: GÉNIE INFORMATIQUE

---

# Développement d'un chatbot et d'un système de gestion des rendez-vous pour une maison médicale

---



*Du 01/04/2024 au 31/05/2024*

*Réalisé par :*

Iliass EL AISSAOUI

*Encadré par :*

Soufyane BOUKHRISS

*Soutenu le 05 Juillet 2024, devant le jury composé de :*

Mme. Hafida ZROURI ESTO - Examinatrice

Mme. Amal BOUAICHA ESTO - Examinatrice

ÉCOLE SUPÉRIEURE DE TECHNOLOGIE OUJDA

Promotion : 2023/2024



# Remerciements

Je tiens à remercier dans un premier temps, toute l'équipe pédagogique de l'École Supérieure de Technologie Oujda et les intervenants professionnels responsables de la formation. je tiens à remercier nos professeurs de nous avoir incités à travailler en mettant à notre disposition leurs expériences et leurs compétences.

Avant d'entamer ce rapport, J'adresse mes remerciements à mes proches qui m'ont sans cesse soutenu dans l'élaboration de mon projet professionnel et m'ont aidé à chaque étape de ce rapport de stage. L'accompagnement dont j'ai bénéficié m'ont permis de trouver rapidement un stage, dans le but d'affiner mes compétences techniques.

Je pense également à Mr. Soufyane BOUKHRISS, qui a cru en mon potentiel et m'a accueilli au sein de son entreprise, qui m'a épaulé et conseillé et qui m'a surtout transmis son expertise dans le domaine de Développement Informatique. Ainsi que son équipe.

# Résumé

Mon rapport de stage de fin d'études, sous la direction de Soufyane BOUKHRISS, porte sur le développement d'un chatbot et d'un système de gestion des rendez-vous pour une maison médicale. Réalisé au sein de l'agence digitale FCPO du 1 avril 2024 au 31 mai 2024, ce stage visait à obtenir le Diplôme Universitaire de Technologie (DUT) en Génie Informatique.

Le projet avait pour objectif de développer un chatbot pour une maison médicale afin d'automatiser les réponses aux questions fréquentes des patients et de faciliter la prise de rendez-vous. Ce projet répondait aux besoins croissants de digitalisation dans le secteur médical, visant à améliorer l'efficacité et la qualité du service aux patients.

Pendant ce stage, plusieurs concepts théoriques ont été approfondis, notamment l'architecture des applications web utilisant le modèle MVC (Model-View-Controller) avec Symfony et API Platform pour structurer les applications web, ainsi que la conception et la gestion de bases de données relationnelles avec MySQL.

Des compétences pratiques essentielles ont été développées, incluant le développement frontend avec la création d'interfaces utilisateur interactives en utilisant React, et le développement backend avec Symfony et API Platform pour créer des APIs RESTful. La gestion de version avec Git et GitHub a également été une compétence clé acquise, permettant une gestion efficace de la version et une collaboration en équipe. L'optimisation de l'environnement de développement avec des outils comme Arch Linux, Neovim, Alacritty, ZSH, tmux, et fzf a également été significative.

Le stage a offert une perspective précieuse sur le fonctionnement et la culture d'entreprise. La collaboration avec une équipe de développeurs, l'utilisation de méthodologies agiles pour la gestion de projet comme les sprints et les réunions quotidiennes, et l'adaptation aux exigences des clients ont été des aspects cruciaux. La compréhension de l'importance de répondre aux besoins des clients et de s'adapter rapidement aux changements de spécifications a été renforcée, tout en intégrant les valeurs et pratiques professionnelles de FCPO.

Le développement du chatbot a présenté plusieurs défis, tels que la maîtrise de nouvelles technologies

comme Symfony, API Platform et React, la configuration complexe pour une réponse personnalisée aux interactions des utilisateurs, et la résolution de problèmes techniques liés à des drivers manquants par l'installation de packages et la vérification systématique des dépendances. Des problèmes de configuration et de gestion des domaines ont également été rencontrés, nécessitant une transition fluide entre les environnements de développement et de production.

En conclusion, le stage chez FCPO a été une expérience riche et formatrice, consolidant des connaissances théoriques et des compétences pratiques. Il a offert une immersion dans un environnement professionnel exigeant, développant des compétences en collaboration d'équipe et en gestion de projet, préparant ainsi efficacement à de futurs défis professionnels.

# Liste des figures

1.1	FCPO logo . . . . .	2
2.1	Flux d'ajout des fonctionnalités en Git . . . . .	12
2.2	Collaborer sur github . . . . .	13
3.1	Flux général du Chatbot . . . . .	18
3.2	Flux du processus: F.A.Q . . . . .	18
3.3	Flux du processus: Prise de rendez-vous . . . . .	19
3.4	Flux du processus: Prise de rendez-vous (Prendre les informations de l'utilisateur) . . . . .	19
3.5	Flux du processus: Prise de rendez-vous (Choisir une spécialité) . . . . .	20
3.6	Diagramme de conception . . . . .	22
4.1	Structure générale du configuration Chatbot . . . . .	26
4.2	Structure générale du Chatbot . . . . .	27
4.3	Structure générale du Chatbot (exemple de "render" element) . . . . .	28
4.4	Fonctionnement d'un API . . . . .	29
4.5	Flux général de création d'une entité API . . . . .	30

# Table des matières

Remerciements	i
Résumé	ii
Liste des figures	iv
Table des matières	vii
<b>1 Contexte Général du Projet</b>	<b>1</b>
1.1 Présentation de l'entreprise FCPO . . . . .	2
1.1.1 L'agence digitale FCPO . . . . .	2
1.1.2 Les Services Proposés . . . . .	3
1.2 Présentation du projet . . . . .	3
1.2.1 Contexte du projet . . . . .	3
1.2.1.1 Les objectifs principaux du projet étaient les suivants : . . . . .	4
1.2.2 Introduction à un Chatbot . . . . .	4
1.2.2.1 Fonctionnement des Chatbots . . . . .	4
1.2.2.2 Types de Chatbots . . . . .	4
1.2.2.3 Applications des Chatbots . . . . .	5
1.2.2.4 Avantages des Chatbots . . . . .	5
<b>2 Environnement de Développement</b>	<b>7</b>
2.1 Système d'Exploitation: Arch Linux (BTW) . . . . .	8
2.1.1 Simplicité et Contrôle . . . . .	8
2.1.2 Documentation Complète . . . . .	8
2.1.3 Mise à Jour Continue . . . . .	8
2.1.4 Communauté Active . . . . .	8

2.2	Éditeurs de texte: (Neo)Vim . . . . .	9
2.2.1	Inspiration . . . . .	9
2.2.2	Configuration Personnalisée . . . . .	9
2.2.3	Avantages de Neovim . . . . .	9
2.3	Terminal: Alacritty avec Zsh . . . . .	10
2.3.1	Alacritty . . . . .	10
2.3.2	Zsh . . . . .	10
2.3.3	Outils supplémentaires . . . . .	11
2.4	Contrôle de Version: Git et GitHub . . . . .	11
2.4.1	Git . . . . .	11
2.4.2	GitHub . . . . .	12
2.5	Autres Outils . . . . .	13
2.5.1	L <sup>A</sup> T <sub>E</sub> X . . . . .	13
2.5.2	Postman . . . . .	14
2.5.3	Eraser . . . . .	14
2.5.4	Figma . . . . .	14
<b>3</b>	<b>Conception et développement de l'application</b>	<b>17</b>
3.1	Description du flux de Chatbot . . . . .	18
3.1.1	FAQ . . . . .	18
3.1.2	Prise de rendez-vous . . . . .	19
3.2	Explication des choix technologiques . . . . .	20
3.2.1	React . . . . .	20
3.2.1.1	Interface Utilisateur Réactive et Performante . . . . .	20
3.2.1.2	Composants Réutilisables et Modulaire . . . . .	20
3.2.1.3	Intégration Facile avec les APIs . . . . .	21
3.2.1.4	Écosystème et Communauté . . . . .	21
3.2.2	Symfony . . . . .	21
3.2.2.1	Framework PHP Puissant . . . . .	21
3.2.2.2	Gestion de l'API avec API Platform . . . . .	21
3.2.2.3	Sécurité Renforcée . . . . .	21
3.3	Diagrammes de conception utilisés . . . . .	22



<b>4 Les missions effectuées</b>	<b>25</b>
4.1 Le Chatbot . . . . .	26
4.1.1 Configuration static . . . . .	26
4.1.2 Configuration dynamique . . . . .	27
4.2 La base de données . . . . .	28
4.2.1 Création des entités . . . . .	28
4.2.2 Configuration d'API . . . . .	28
4.3 Tableau de bord . . . . .	30
4.3.1 La gestion des rendez-vous . . . . .	30
4.3.2 La gestion d'équipe médicale . . . . .	30
4.3.3 La gestion du Chatbot . . . . .	31
<b>5 Les apports du stage</b>	<b>33</b>
5.1 Les apports théoriques et intellectuels . . . . .	34
5.2 Les compétences pratiques développées . . . . .	34
5.3 Les apports en termes de monde de l'entreprise . . . . .	35
<b>6 Les difficultés du stage et les solutions apportées</b>	<b>37</b>
6.1 Les difficultés rencontrées . . . . .	38
6.2 Les solutions apportées à ces difficultés . . . . .	38
<b>Conclusion générale et perspectives</b>	<b>41</b>
<b>A Glossaire</b>	<b>42</b>
<b>B Acronymes</b>	<b>44</b>
<b>Ressources</b>	<b>45</b>

# Chapter 1

## Contexte Général du Projet

### Introduction

Dans le cadre de mon stage chez l'agence digitale FCPO, j'ai eu l'opportunité de participer à un projet innovant de développement d'un chatbot destiné à une maison médicale. Fondée en 2013, FCPO est une entreprise spécialisée dans le marketing digital, offrant une large gamme de services incluant la création de sites web, le référencement naturel (SEO), le community management, et le développement d'applications mobiles. Avec une équipe d'experts et des partenaires nationaux et internationaux, FCPO se distingue par la qualité et la rapidité de ses prestations.

Le projet de chatbot visait à répondre aux besoins croissants de digitalisation dans le secteur médical, en automatisant les réponses aux questions fréquentes et en facilitant la prise de rendez-vous pour les patients. Ce rapport détaillera le contexte et les objectifs du projet, les outils et technologies utilisés dans l'environnement de développement, ainsi que les étapes de conception et de développement de l'application. Nous explorerons également les différents types de chatbots, leurs fonctionnalités, et les avantages qu'ils offrent dans divers domaines d'application.

## 1.1 Présentation de l'entreprise FCPO

### 1.1.1 L'agence digitale FCPO



Figure 1.1: FCPO logo

Créée en 2013, l'agence FCPO est une entreprise de marketing digital qui accompagne les entreprises dans leurs stratégies web et marketing digital.

L'agence digitale FCPO accompagne les entreprises, les professionnels et les blogueurs à mettre en place une stratégie digitale autour d'un site web. FCPO travaille sur des projets de création de sites web, community management, référencement naturel et payant, gestion des contenus, mise en place des stratégies d'inbound marketing et marketing digital etc.

FCPO travaille avec des partenaires au niveau national et international (Maroc, Afrique du Nord, France, Belgique, Pays-Bas).

Son équipe est composée d'experts dans ce domaine. FCPO garantit un haut niveau de qualité, mais assure également une production en grande quantité de sites Web.

Le développement de l'entreprise se fait par étapes solides basées sur la connaissance et l'expertise.

Toujours à la recherche des meilleures méthodes pour améliorer l'efficacité de ses réalisations, ses équipes souhaitent proposer des services exclusifs avec un très haut niveau de qualité. La fiabilité de ses services lui permet d'offrir des solutions efficaces et pérennes, sans risque de pénalité. À savoir que les commandes sont traitées sous 24H maximum.

### 1.1.2 Les Services Proposés

- **Création site Internet:** Site web clé en main moderne et haut de gamme.
- **Référencement Naturel – SEO:** Site web bien optimisé au SEO et disponible sur la 1ère page du moteur de recherche Google.
- **Création site E-Commerce:** Sites performants et ergonomiques, à l'esth-étique soignée, pour une expérience utilisateur unique.
- **Publicité sur Internet:** Booster l'image de marque de votre entreprise sur internet et sur les réseaux sociaux.
- **Application mobile:** Expertise et innovation dans le développement des applications mobile.
- **Rédaction de contenu:** La prise en charge de la rédaction du contenu des site web.

## 1.2 Présentation du projet

### 1.2.1 Contexte du projet

Dans le cadre de mon stage au sein de l'entreprise FCPO, j'ai participé à un projet visant à développer un chatbot pour une maison médicale. Ce chatbot est destiné à assister les patients en répondant aux questions fréquentes (FAQ) et en facilitant la prise de rendez-vous. Le projet a été réalisé en collaboration avec une équipe de stagiaires comme moi.

Les maisons médicales reçoivent quotidiennement un grand nombre de questions répétitives de la part des patients, ainsi que des demandes de prise de rendez-vous. Cela mobilise une partie significative du temps du personnel administratif, qui pourrait être mieux utilisé pour des tâches nécessitant une intervention humaine directe. De plus, les patients recherchent une solution rapide et accessible pour obtenir des informations et réserver des consultations sans devoir attendre de longues périodes au téléphone.

### 1.2.1.1 Les objectifs principaux du projet étaient les suivants :

- **Automatiser les réponses aux questions fréquentes:** Fournir une assistance instantanée aux patients pour leurs questions courantes sur les horaires, les services offerts, et les procédures.
- **Faciliter la prise de rendez-vous:** Permettre aux patients de réserver des consultations de manière autonome via le chatbot.
- **Gestion centralisée des rendez-vous:** Développer un tableau de bord pour que les administrateurs puissent gérer les rendez-vous, les médecins, et leurs spécialités.

## 1.2.2 Introduction à un Chatbot

Un chatbot, ou agent conversationnel, est un programme informatique conçu pour simuler une conversation humaine avec les utilisateurs, en particulier sur Internet. Les chatbots sont souvent utilisés dans les interfaces de messagerie, les sites web et les applications mobiles pour offrir une assistance instantanée et automatisée. Ils peuvent interagir avec les utilisateurs par le biais de texte ou de voix, répondant à des questions, fournissant des informations, et exécutant diverses tâches de manière autonome.

### 1.2.2.1 Fonctionnement des Chatbots

Les chatbots fonctionnent grâce à une combinaison de règles préétablies et de technologies avancées telles que le traitement du langage naturel (NLP) et l'intelligence artificielle (IA). Les chatbots simples reposent sur des règles et des scripts prédéfinis, tandis que les chatbots plus sophistiqués utilisent le NLP pour comprendre le contexte et l'intention des utilisateurs. Ces derniers peuvent apprendre et s'améliorer avec le temps grâce à des techniques de machine learning.

### 1.2.2.2 Types de Chatbots

Il existe deux principaux types de chatbots :

- Chatbots basés sur des règles:
  - Ils suivent des scripts prédéfinis et des arbres de décision.
  - Idéaux pour des interactions simples et des questions fréquemment posées.
  - Limités par la complexité des scénarios qu'ils peuvent gérer.

- Chatbots basés sur l'IA:
  - Utilisent le NLP pour comprendre et interpréter les requêtes des utilisateurs.
  - Capables de gérer des interactions plus complexes et d'apprendre de nouvelles informations au fil du temps.
  - Plus flexibles et adaptatifs par rapport aux chatbots basés sur des règles.

### 1.2.2.3 Applications des Chatbots

Les chatbots sont utilisés dans une variété de domaines pour améliorer l'expérience utilisateur et optimiser les processus opérationnels. Parmi les applications courantes, on trouve :

- **Service client:** Fournir une assistance 24/7, répondre aux questions fréquentes, et traiter les réclamations.
- **E-commerce:** Aider les clients à trouver des produits, passer des commandes, et suivre les livraisons.
- **Santé:** Répondre aux questions médicales générales, prendre des rendez-vous, et rappeler aux patients de prendre leurs médicaments.
- **Banque et finance:** Fournir des informations sur les comptes, aider à effectuer des transactions, et conseiller sur les produits financiers.

### 1.2.2.4 Avantages des Chatbots

Les chatbots présentent de nombreux avantages pour les organisations et les utilisateurs, notamment:

- **Disponibilité:** Accessibles 24/7, ils offrent une assistance continue sans interruption.
- **Efficacité:** Capables de traiter plusieurs demandes simultanément, réduisant ainsi les temps d'attente pour les utilisateurs.
- **Coût:** Réduction des coûts opérationnels en automatisant les tâches répétitives et en libérant les ressources humaines pour des tâches plus complexes.
- **Personnalisation:** Possibilité de personnaliser les interactions en fonction des préférences et des historiques des utilisateurs.

En somme, les chatbots représentent une avancée significative dans l'interaction homme-machine, offrant des solutions innovantes pour améliorer la communication et les services dans divers secteurs.

## Conclusion

La présentation de l'entreprise FCPO ainsi que le contexte et les objectifs du projet de développement d'un chatbot pour une maison médicale mettent en lumière l'importance croissante du marketing digital et de l'automatisation des services dans divers secteurs. Grâce à son expertise dans le domaine du marketing digital, FCPO s'engage à fournir des solutions de haute qualité, tout en s'adaptant aux besoins spécifiques de ses clients.

Le projet de développement du chatbot répond à un besoin concret dans le domaine médical en offrant une assistance instantanée aux patients et en simplifiant le processus de prise de rendez-vous. Les chatbots, en général, représentent une avancée significative dans l'interaction homme-machine, offrant des solutions innovantes pour améliorer la communication et les services dans divers secteurs.

En conclusion, FCPO, par son engagement envers l'innovation et la qualité, ainsi que le développement du chatbot pour une maison médicale, illustre parfaitement comment le marketing digital et l'automatisation peuvent transformer les processus traditionnels et améliorer l'expérience client dans le monde moderne.

# Chapter 2

## Environnement de Développement

### Introduction

Dans ce chapitre, nous explorons l'environnement de développement utilisé pour ce projet. Un environnement de développement bien choisi est crucial pour assurer une productivité élevée, une gestion efficace du code et une collaboration harmonieuse entre les membres de l'équipe. Nous détaillerons les outils et technologies spécifiques qui ont été adoptés, tels que le système d'exploitation, les éditeurs de texte, les terminaux, et les systèmes de contrôle de version.



## 2.1 Système d'Exploitation: Arch Linux (BTW)

Pour le développement de ce projet, j'ai choisi d'utiliser Arch Linux comme système d'exploitation principal. Arch Linux est une distribution Linux réputée pour sa simplicité, sa flexibilité et son approche "rolling release", ce qui permet d'avoir toujours les dernières versions des logiciels et des bibliothèques. Ma décision d'opter pour Arch Linux a été motivée par plusieurs facteurs, notamment:

### 2.1.1 Simplicité et Contrôle

Arch Linux suit la philosophie "Keep It Simple, Stupid" (KISS), offrant un contrôle total sur l'installation et la configuration du système. Cette approche minimaliste m'a permis de créer un environnement de développement sur mesure, en installant uniquement les packages et les outils nécessaires pour le projet, sans superflu ni bloatware.

### 2.1.2 Documentation Complète

L'Arch Wiki, la documentation officielle d'Arch Linux, est une ressource exhaustive qui couvre une large gamme de sujets. Cette documentation détaillée m'a été particulièrement utile pour la configuration du système, la résolution de problèmes techniques et l'apprentissage de nouvelles compétences liées à Linux.

### 2.1.3 Mise à Jour Continue

Grâce à son modèle "rolling release", Arch Linux fournit les versions les plus récentes des logiciels, ce qui est crucial pour un développement moderne et efficace. En ayant toujours accès aux dernières fonctionnalités et aux correctifs de sécurité, j'ai pu travailler avec des outils à jour et bénéficier des dernières avancées technologiques.

### 2.1.4 Communauté Active

La communauté Arch Linux est réputée pour son dynamisme et son engagement. Les forums, les canaux IRC et d'autres plateformes de discussion offrent un soutien continu, des conseils pratiques et des solutions aux problèmes rencontrés. Cet écosystème vibrant a contribué à enrichir mon expérience avec Arch Linux et à résoudre rapidement les défis rencontrés.

## 2.2 Éditeurs de texte: (Neo)Vim

Pour le développement de ce projet, j'ai choisi d'utiliser Neovim comme éditeur de texte principal. Neovim est une version améliorée de Vim, un éditeur de texte puissant et hautement configurable, largement utilisé dans la communauté des développeurs. Ma décision d'opter pour Neovim a été motivée par plusieurs facteurs, notamment:

### 2.2.1 Inspiration

Je me suis inspiré de créateurs de contenu renommés tels que ThePrimeagen, TJ DeVries, Typecraft, et Dreams of Code, qui partagent leur expertise et leurs astuces sur Neovim à travers des tutoriels, des vidéos et des configurations. Leur maîtrise de Neovim et leurs conseils ont été une source d'inspiration et d'apprentissage pour moi, m'encourageant à explorer davantage les fonctionnalités avancées de Neovim et à personnaliser mon propre environnement de développement.

### 2.2.2 Configuration Personnalisée

Ma configuration Neovim a été entièrement personnalisée avec plus de 1100 lignes de code. Cette configuration comprend des réglages de base, des raccourcis clavier, des plugins, des thèmes et des intégrations avec d'autres outils de développement. En m'appuyant sur les bonnes pratiques et les astuces partagées par la communauté, j'ai pu créer un environnement de développement optimisé et ergonomique, adapté à mes besoins spécifiques et à mon flux de travail.

### 2.2.3 Avantages de Neovim

- **Performance:** Neovim est conçu pour offrir des performances supérieures à Vim, notamment en termes de vitesse de chargement des fichiers et de réactivité des commandes.
- **Extensibilité:** La structure modulaire de Neovim permet d'ajouter facilement des fonctionnalités supplémentaires via des plugins et des scripts Lua, offrant ainsi une grande flexibilité et une adaptabilité aux besoins changeants du développement logiciel.
- **Communauté Active:** La communauté Neovim est dynamique et engagée, offrant un soutien continu, des mises à jour fréquentes et des contributions constantes à l'écosystème Neovim, ce qui garantit un développement continu et une évolution constante de l'éditeur.

## 2.3 Terminal: Alacritty avec Zsh

Pour le développement de ce projet, j'ai utilisé Alacritty comme terminal principal, associé à Zsh comme shell par défaut. Alacritty est un émulateur de terminal GPU-accelerated écrit en Rust, connu pour sa réactivité et ses performances élevées. Rust est un langage de programmation moderne qui offre à la fois la sécurité du code et des performances exceptionnelles, ce qui en fait un choix idéal pour les applications sensibles aux performances comme Alacritty. Cette combinaison a été choisie pour améliorer mon flux de travail et ma productivité, et elle a été enrichie par plusieurs utilitaires et configurations personnalisées:

### 2.3.1 Alacritty

Alacritty a été choisi pour sa légèreté et sa rapidité. Étant un émulateur de terminal écrit en Rust, Alacritty offre une expérience fluide et réactive, ce qui est essentiel pour un développement efficace. Ses principaux avantages sont:

- **Performances élevées:** Alacritty est conçu pour être rapide, avec un temps de démarrage minimal et une réponse instantanée aux commandes.
- **Langage de Programmation Rust:** Alacritty est écrit en Rust, un langage de programmation moderne qui offre à la fois la sécurité du code et des performances exceptionnelles.
- **Personnalisable:** Alacritty offre de nombreuses options de personnalisation, notamment la possibilité de configurer les raccourcis clavier, les thèmes et les polices.

### 2.3.2 Zsh

Zsh a été utilisé comme shell interactif principal, enrichi par une configuration personnalisée comprenant plusieurs plugins et utilitaires pour améliorer l'expérience utilisateur. Quelques éléments notables de cette configuration sont:

- **Oh My Zsh:** J'ai utilisé Oh My Zsh, un framework de gestion de configuration Zsh, pour faciliter l'installation et la gestion des plugins et des thèmes.
- **Plugins Zsh:** J'ai intégré plusieurs plugins Zsh, tels que zsh-syntax-highlighting, zsh-completions, zsh-autosuggestions, et Aloxaf/fzf-tab, pour améliorer la syntaxe en surbrillance, la complétion automatique, et l'expérience de suggestion de commandes.

- **Zsh Theme personnalisé:** J'ai implémenté un thème Zsh personnalisé, adapté à mes préférences et à mon flux de travail, pour afficher des informations utiles telles que le répertoire de travail actuel, le statut du git, et d'autres informations système.

### 2.3.3 Outils supplémentaires

En plus d'Alacritty et Zsh, j'ai utilisé plusieurs autres utilitaires pour améliorer mon flux de travail:

- **Tmux:** Tmux a été utilisé pour la gestion des sessions et la possibilité de diviser l'écran en plusieurs panneaux. Cela m'a permis de travailler de manière plus efficace en ayant plusieurs terminaux ouverts simultanément pour différentes tâches.
- **Fzf:** Fzf a été utilisé pour la recherche rapide et interactive dans les fichiers, les historiques de commandes, et d'autres listes de données.
- **Zoxide:** Zoxide a été utilisé pour la navigation rapide dans le système de fichiers en se basant sur les répertoires les plus fréquemment utilisés.
- **Scripts Bash personnalisés:** J'ai écrit plusieurs scripts Bash personnalisés pour automatiser des tâches répétitives et simplifier les opérations courantes dans mon environnement de développement.

## 2.4 Contrôle de Version: Git et GitHub

Pour la gestion du code source de ce projet, j'ai utilisé Git comme système de contrôle de version et GitHub comme plateforme d'hébergement et de collaboration. Git est un système de contrôle de version distribué largement utilisé, conçu pour suivre les modifications apportées au code source tout au long du développement d'un projet. GitHub, quant à lui, est une plateforme de développement collaboratif qui permet aux développeurs de partager leur code, de travailler en équipe et de contribuer à des projets open source. Voici quelques aspects de mon utilisation de Git et GitHub dans ce projet:

### 2.4.1 Git

- **Gestion des Versions:** Git a été utilisé pour suivre l'évolution du code source tout au long du développement du projet. J'ai créé des commits réguliers pour enregistrer les modifications apportées au code et maintenir un historique clair et organisé.

- **Branches:** J'ai utilisé des branches Git pour travailler sur des fonctionnalités isolées et des correctifs de bugs sans perturber le code principal. Les branches m'ont permis de développer de manière collaborative tout en maintenant la stabilité du code.
- **Fusion (Merge):** Une fois les fonctionnalités développées et testées, j'ai fusionné les branches de développement dans la branche principale (main). Cette fusion a consolidé les modifications et intégré les nouvelles fonctionnalités dans le code principal du projet.

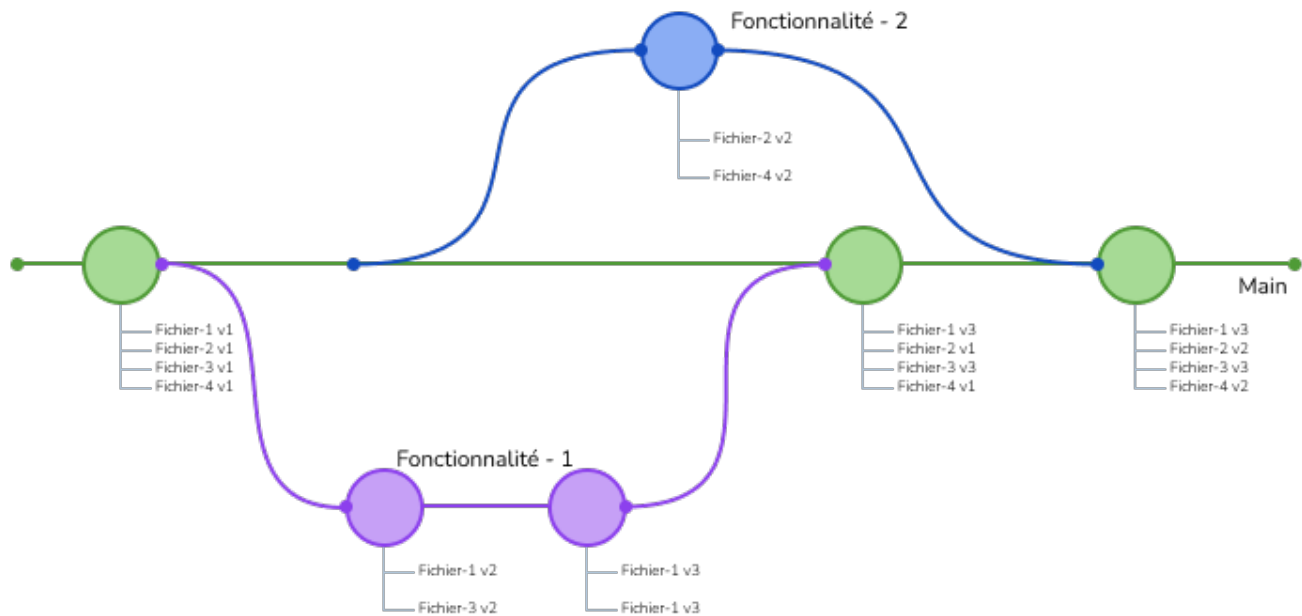


Figure 2.1: Flux d'ajout des fonctionnalités en Git

### 2.4.2 GitHub

- **Hébergement du Code:** J'ai hébergé le code source du projet sur GitHub, ce qui m'a permis de le partager facilement avec mes collaborateurs et mon encadrant.
- **Suivi des Problèmes (Issues):** J'ai utilisé la fonctionnalité des Issues sur GitHub pour suivre les tâches, les bogues et les demandes de fonctionnalités tout au long du développement du projet. Les Issues ont permis une communication efficace entre les membres de l'équipe et ont contribué à une gestion transparente des problèmes.
- **Pull Requests (Demandes de Tirage):** Pour intégrer les modifications dans le code principal du projet, j'ai créé des Pull Requests sur GitHub. Les Pull Requests ont facilité la révision du code par les pairs, les tests d'intégration et les validations avant la fusion finale.

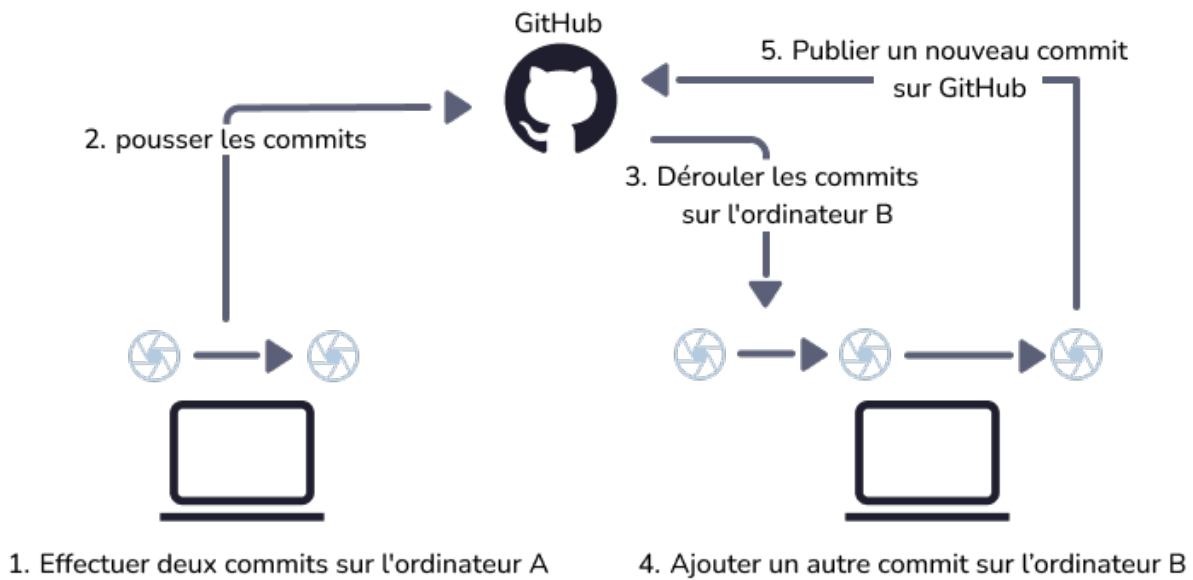


Figure 2.2: Collaborer sur github

## 2.5 Autres Outils

### 2.5.1 $\text{\LaTeX}$

Pour la rédaction du rapport de ce projet, j'ai choisi d'utiliser  $\text{\LaTeX}$ , un système de composition de documents largement utilisé dans le domaine académique et technique pour sa capacité à produire des documents de haute qualité typographique.  $\text{\LaTeX}$  offre de nombreux avantages pour la rédaction de rapports techniques, notamment:

- **Typographie de haute qualité:**  $\text{\LaTeX}$  produit des documents avec une typographie professionnelle et esthétique, offrant un rendu précis des symboles mathématiques, des formules, des tableaux et des graphiques.
- **Gestion avancée des références:**  $\text{\LaTeX}$  simplifie la gestion des références croisées, des citations bibliographiques et des tableaux des matières, ce qui facilite l'organisation et la navigation dans le rapport.
- **Séparation du contenu et de la mise en page:**  $\text{\LaTeX}$  permet de séparer le contenu du document de sa mise en page, ce qui facilite la modification et la réorganisation du contenu sans affecter la présentation visuelle.

## 2.5.2 Postman

Postman a été un outil essentiel dans le processus de développement de l'API pour notre projet. En tant que plateforme de collaboration pour le développement d'API, Postman nous a permis de tester facilement les points de terminaison de notre API RESTful. Avec Postman, nous avons pu envoyer des requêtes HTTP, inspecter les réponses, et valider le comportement de notre API à différentes étapes du développement. L'interface conviviale de Postman nous a également permis de partager nos collections d'API avec d'autres membres de l'équipe et de collaborer efficacement sur les tests et les validations. En résumé, Postman a grandement facilité le processus de développement et de test de notre API, contribuant ainsi à la qualité et à la fiabilité de notre solution logicielle.

## 2.5.3 Eraser

Eraser a été un élément essentiel de notre processus de conception pour visualiser l'architecture et le flux de notre application. Voici comment nous avons utilisé cet outil:

- **Création de Diagrammes:** Eraser nous a permis de créer des diagrammes de classe, des diagrammes de séquence et d'autres types de schémas pour représenter les différents aspects de notre application.
- **Personnalisation:** Nous avons bénéficié d'une large gamme d'outils et de fonctionnalités offerts par Eraser, nous permettant ainsi de personnaliser nos diagrammes selon nos besoins spécifiques et de les rendre aussi clairs et informatifs que possible.
- **Collaboration:** L'interface intuitive d'Eraser et sa capacité à exporter des diagrammes dans divers formats ont facilité la collaboration au sein de l'équipe. Nous avons pu partager nos diagrammes avec les membres de l'équipe et les parties prenantes pour recueillir des commentaires et assurer une compréhension commune du système.

## 2.5.4 Figma

Figma a été un outil polyvalent dans notre processus de conception, nous permettant de créer à la fois des diagrammes détaillés et des présentations visuellement attrayantes. Voici comment nous l'avons utilisé:

- **Création de Diagrammes:** Comme Eraser, Figma nous a permis de créer des diagrammes de classe, des diagrammes de séquence et d'autres types de schémas pour représenter l'architecture

et le flux de notre application. Nous avons bénéficié d'une interface conviviale et d'outils intuitifs pour réaliser ces diagrammes avec précision.

- **Présentations Visuelles:** En plus de la création de diagrammes, Figma a également été utilisé pour créer des présentations visuellement attrayantes. Nous avons pu intégrer des diagrammes, des maquettes d'interface utilisateur et d'autres éléments visuels dans nos présentations pour communiquer efficacement nos idées et nos concepts.



## Conclusion

Ce chapitre a couvert les différents composants de l'environnement de développement, incluant Arch Linux comme système d'exploitation, Neovim pour l'édition de texte, Alacritty comme terminal, et Git pour le contrôle de version. Chaque choix a été justifié par ses avantages spécifiques pour le projet.

# Chapter 3

## Conception et développement de l'application

### Introduction

Ce chapitre se concentre sur la conception et le développement de l'application. Nous commencerons par une description détaillée du flux de l'application, en expliquant comment elle guide les utilisateurs à travers ses différentes fonctionnalités. Ensuite, nous aborderons les choix technologiques majeurs, notamment l'utilisation de React pour l'interface utilisateur et Symfony pour le backend. Enfin, nous présenterons les diagrammes de conception utilisés pour visualiser et planifier l'architecture de l'application.

### 3.1 Description du flux de Chatbot

Le flux de l'application du chatbot pour la maison médicale se déroule en plusieurs étapes clés, organisées pour optimiser l'expérience utilisateur:

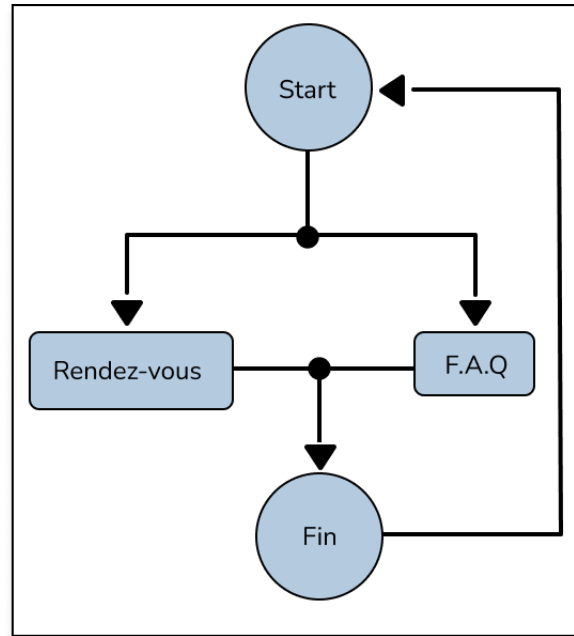


Figure 3.1: Flux général du Chatbot

#### 3.1.1 FAQ

Le chatbot propose à l'utilisateur des sujets prédéfinis sur lesquels il pourrait vouloir en savoir plus, et lors de la sélection d'un, le chatbot répond avec une réponse prédéfinie.

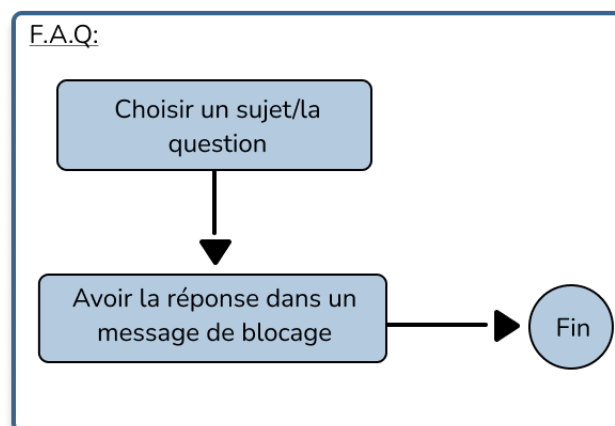


Figure 3.2: Flux du processus: F.A.Q

### 3.1.2 Prise de rendez-vous

Le chatbot guide l'utilisateur avec un flux prédéfini dans ce cas, depuis la demande d'informations de base sur l'utilisateur jusqu'aux opérations facultatives telles que répondre à des questions spécialisées ou joindre des documents, avec une validation intégrée pour chaque petite information que nous lui demandons de fournir (si la validation est possible dans le premier lieu).

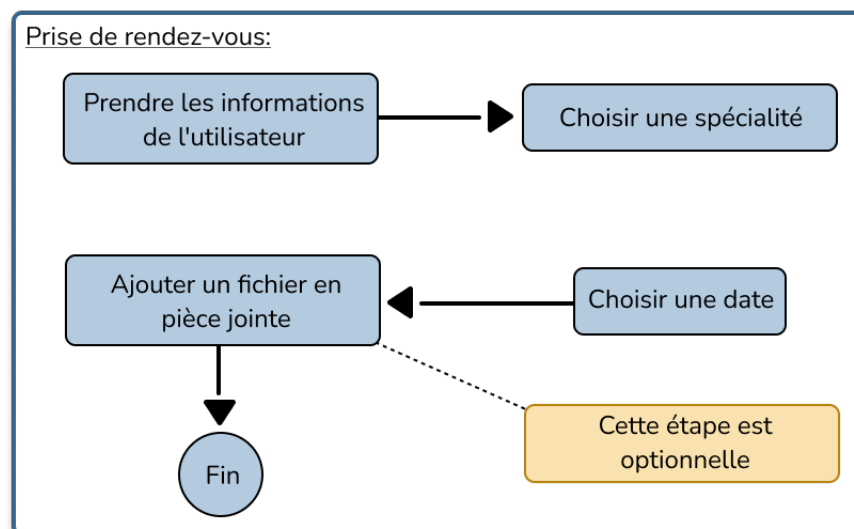


Figure 3.3: Flux du processus: Prise de rendez-vous

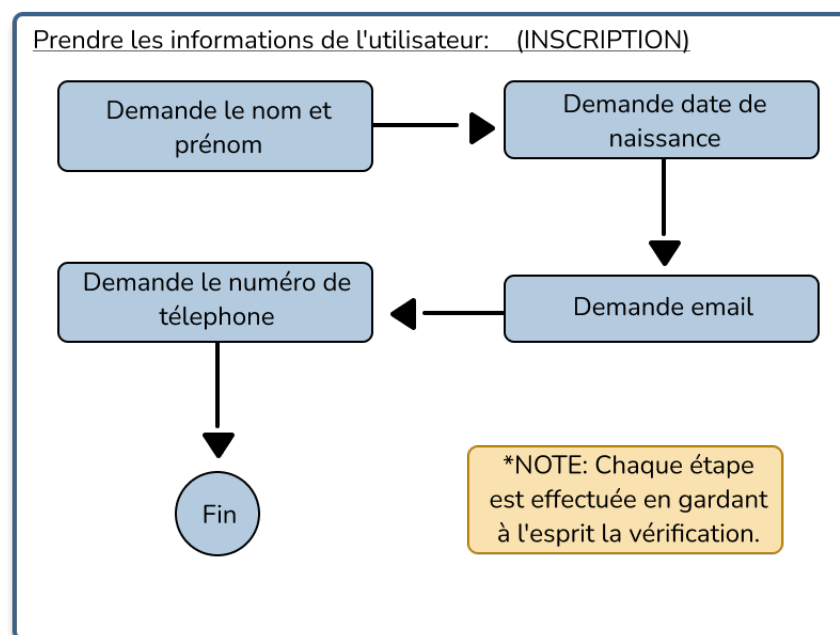


Figure 3.4: Flux du processus: Prise de rendez-vous (Prendre les informations de l'utilisateur)

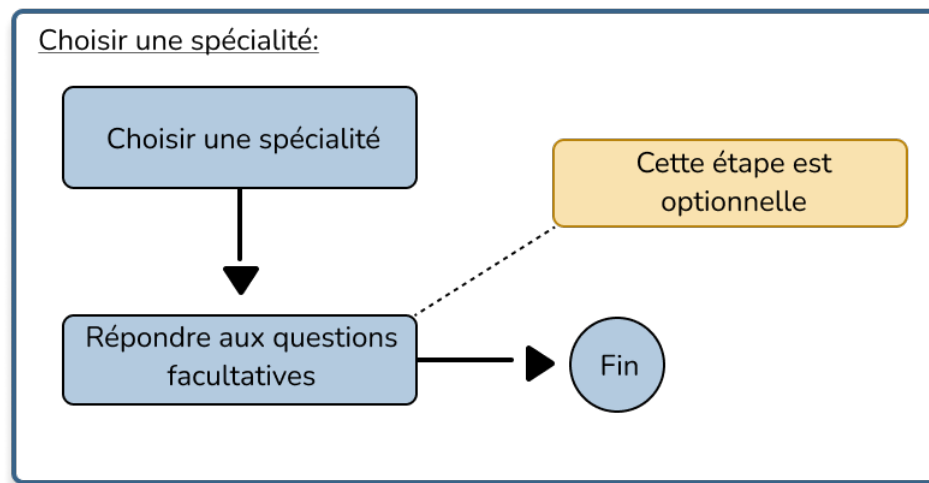


Figure 3.5: Flux du processus: Prise de rendez-vous (Choisir une spécialité)

## 3.2 Explication des choix technologiques

Pour les technologies, nous n'avons pas beaucoup de choix, on nous proposait de travailler soit avec Django de Python, soit avec React (Front-end) et Symfony\* (Back-end).

### 3.2.1 React

#### 3.2.1.1 Interface Utilisateur Réactive et Performante

- **DOM Virtuel:** React utilise un DOM virtuel pour des mises à jour rapides et efficaces de l'interface utilisateur, garantissant une expérience utilisateur fluide et sans latence.
- **Rendu Dynamique:** Les capacités de rendu dynamique de React permettent de gérer efficacement les interactions en temps réel, cruciales pour un chatbot.

#### 3.2.1.2 Composants Réutilisables et Modulaire

- **Composants:** La structure basée sur les composants de React facilite la création de widgets et de modules réutilisables pour différentes parties du chatbot, comme les fenêtres de conversation, les formulaires de saisie, et les notifications.
- **Modularité:** Permet de construire l'application de manière modulaire, rendant le code plus maintenable et évolutif.

### 3.2.1.3 Intégration Facile avec les APIs

- **Hooks:** Les hooks de React, comme `useEffect` et `useState`, simplifient les appels API et la gestion des états, permettant une intégration fluide avec les services backend du chatbot.
- **Interopérabilité:** React facilite l'intégration avec des APIs RESTful ou GraphQL, nécessaires pour les fonctionnalités de traitement du langage naturel (NLP) et la gestion des dialogues.

### 3.2.1.4 Écosystème et Communauté

- **Support et Documentation:** La vaste communauté de React et sa documentation exhaustive offrent un soutien continu et des ressources abondantes pour résoudre les problèmes et optimiser le développement.
- **Outils et Bibliothèques:** Un large éventail de bibliothèques et d'outils complémentaires, tels que les bibliothèques de calendriers et même la bibliothèque de chatbots avec laquelle nous avons travaillé.

## 3.2.2 Symfony

### 3.2.2.1 Framework PHP Puissant

- Symfony est un framework PHP mature et puissant, offrant une structure solide et bien organisée pour le développement d'applications web complexes telles que le chatbot médical. Sa stabilité et sa maturité en font un choix fiable pour la création d'un back-end robuste.

### 3.2.2.2 Gestion de l'API avec API Platform

- Symfony offre une intégration transparente avec API Platform, une solution robuste pour la création et la gestion d'API RESTful. En configurant l'API avec Symfony et API Platform, nous bénéficions d'une documentation automatique, d'une gestion avancée des opérations CRUD, et d'une sérialisation/désérialisation automatique des données.

### 3.2.2.3 Sécurité Renforcée

- Symfony intègre des fonctionnalités avancées de sécurité, telles que la protection contre les attaques CSRF, XSS et SQL injection. Cela garantit un niveau élevé de sécurité pour les données sensibles des utilisateurs du chatbot médical, ce qui est essentiel dans le domaine médical.

### 3.3 Diagrammes de conception utilisés

Dans le cadre du développement du chatbot pour la maison médicale, nous avons créé un Modèle Conceptuel de Données (MCD) pour concevoir la structure de la base de données. Le MCD ci-dessous illustre les principales entités et leurs relations dans notre système.

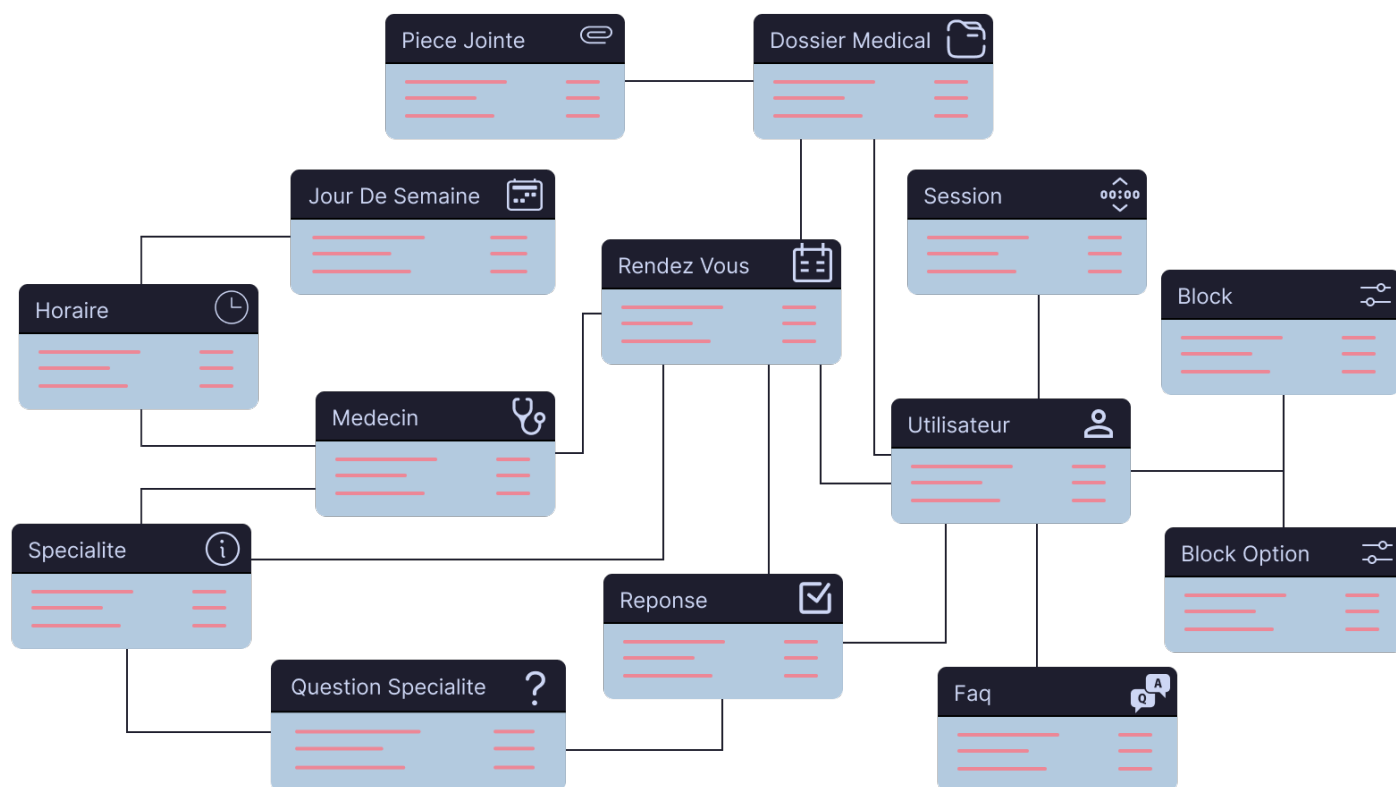


Figure 3.6: Diagramme de conception

Ce MCD représente les principales entités de notre base de données pour le chatbot médical. Voici une brève explication de chaque entité :

- **Utilisateur:** Stocke les informations sur les utilisateurs du chatbot, telles que leur identifiant, leur nom et leurs informations de contact. Les utilisateurs peuvent avoir des rôles spécifiques, comme patient ou admin.
- **Medecin:** Stocke les informations sur les médecins, telles que leur identifiant, leur nom et leurs informations de contact.
- **Specialite:** Décrit les différentes spécialités médicales disponibles, telles que la cardiologie, la dermatologie, etc.

- **Question Specialite:** Contient des questions uniques pour chaque spécialité.
- **Reponse:** Enregistre les réponses fournies par les patients aux questions relatives à une spécialité médicale. Inclut des références à l'utilisateur, à la question de spécialité et à la réponse fournie.
- **Jour De Semaine:** Stocke les jours de la semaine pour aider à gérer les horaires et les disponibilités des médecins et des rendez-vous.
- **Horaire:** Définit les plages horaires disponibles pour les rendez-vous et les sessions avec les médecins.
- **Session:** Contient la séance de travail pour le centre médical.
- **Rendez Vous:** Enregistre les rendez-vous programmés, y compris la date, l'heure, le patient et le médecin assigné.
- **Faq:** Stocke les questions fréquentes et leurs réponses associées, utilisées par le chatbot pour fournir des informations instantanées.
- **Block:** Décrit les blocs de contenu ou de dialogue utilisés dans le chatbot pour structurer les conversations.
- **Block Option:** Contient les options disponibles pour chaque bloc de contenu, permettant au chatbot de guider les utilisateurs à travers différentes branches de la conversation.
- **Piece jointe:** Gère le téléchargement de fichiers. Cette entité stocke les informations sur les fichiers joints aux conversations.
- **Dossier Medical:** Gère les dossiers médicaux des utilisateurs. Cette entité permet le téléchargement de plusieurs fichiers pour chaque utilisateur.



## Conclusion

La conception et le développement de l'application décrits dans ce chapitre montrent comment les choix technologiques et les approches méthodologiques ont permis de créer une solution efficace et robuste. En utilisant des frameworks modernes et des pratiques de conception bien établies, nous avons pu développer une application qui répond aux besoins des utilisateurs de manière optimale.

# Chapter 4

## Les missions effectuées

### Introduction

Ce chapitre est consacré à la description des missions effectuées durant le stage. Chaque mission représente une étape clé dans le développement du chatbot, depuis la configuration initiale jusqu'à la mise en ligne. Nous aborderons la configuration des environnements de développement, la création et la gestion des bases de données, et le développement des interfaces utilisateur. En détaillant ces missions, nous mettrons en lumière les compétences techniques mobilisées et les défis rencontrés. Ce chapitre vise à illustrer le processus de développement et les contributions spécifiques apportées à chaque étape du projet.

## 4.1 Le Chatbot

### 4.1.1 Configuration static

La configuration statique du chatbot concerne les éléments du chatbot qui sont définis à l'avance et qui ne changent pas dynamiquement en fonction des interactions utilisateur. Voici quelques aspects clés de cette configuration:

- **Réponses prédéfinies:** Le chatbot est programmé avec une série de réponses prédéfinies pour les questions fréquentes.
- **Flow du prise de rendez-vous:** Définir un flux statique qui accompagnera l'utilisateur dans sa prise de rendez-vous.
- **Interface utilisateur:** La présentation et les options initiales du chatbot, telles que les boutons de démarrage et les messages de bienvenue, sont configurées statiquement.
- **Connexion:** Configurer un processus statique permettant aux utilisateurs de se connecter avant d'utiliser certaines fonctionnalités du chatbot.
- **Inscription:** Mettre en place un processus d'inscription statique pour les nouveaux utilisateurs, leur permettant de créer un compte avant d'accéder aux fonctionnalités du chatbot.

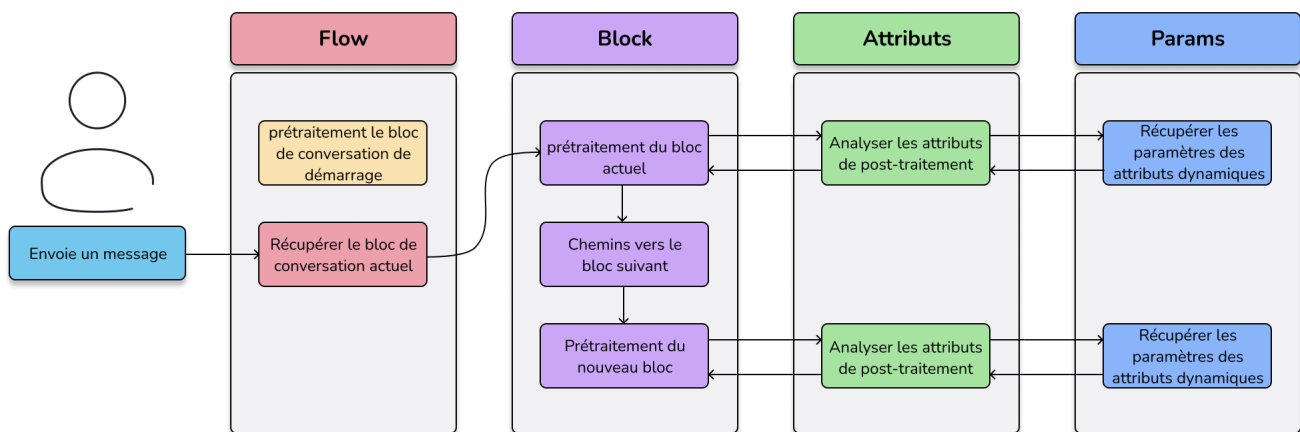


Figure 4.1: Structure générale du configuration Chatbot

### 4.1.2 Configuration dynamique

Concernant les aspects dynamiques de notre chatbot, il nous a été demandé de le rendre « entièrement » personnalisable, ce qui signifie:

- **Construction dynamique:** Chaque message à l'écran doit être configurable via un tableau de bord, y compris les messages d'erreur.
- **Messages dynamiques:** Les messages doivent être modifiés en fonction du contexte.
- **Flux dynamique:** L'utilisateur doit avoir la possibilité de choisir entre où aller ensuite à partir de sa position actuelle (à l'exclusion bien sûr des flux dans lesquels nous demandons des informations critiques pour créer le compte de l'utilisateur)
- **Composant dynamique:** Les composants personnalisés doivent s'adapter aux entrées de l'utilisateur et afficher les informations en conséquence.
- **Données dynamiques:** chaque information contenue par le chatbot provient de l'API qui connecte notre chatbot à la base de données.

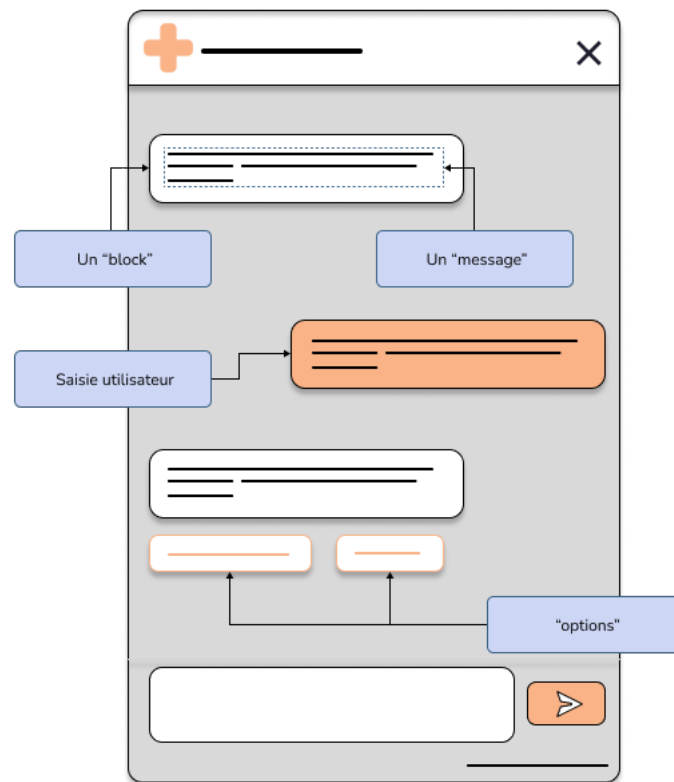


Figure 4.2: Structure générale du Chatbot

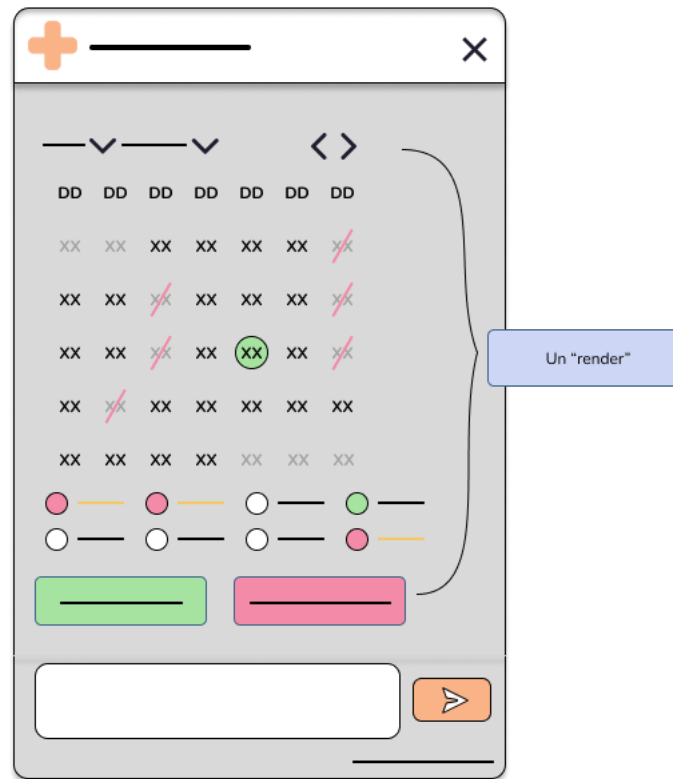


Figure 4.3: Structure générale du Chatbot (exemple de "render" element)

## 4.2 La base de données

### 4.2.1 Création des entités

La création des entités (Figure 3.6) dans la base de données est une étape cruciale pour structurer les informations de manière efficace et logique.

En structurant ces entités de manière logique et efficace, nous avons pu créer une base de données robuste et évolutive, capable de gérer les différentes facettes du projet, y compris la gestion des utilisateurs, des médecins, des rendez-vous, et des interactions du chatbot.

### 4.2.2 Configuration d'API

Une API (Application Programming Interface) est un ensemble de règles permettant à des applications de communiquer entre elles. Elle utilise des protocoles comme HTTP/HTTPS pour échanger des données et donne accès à certaines fonctionnalités ou informations d'un service.

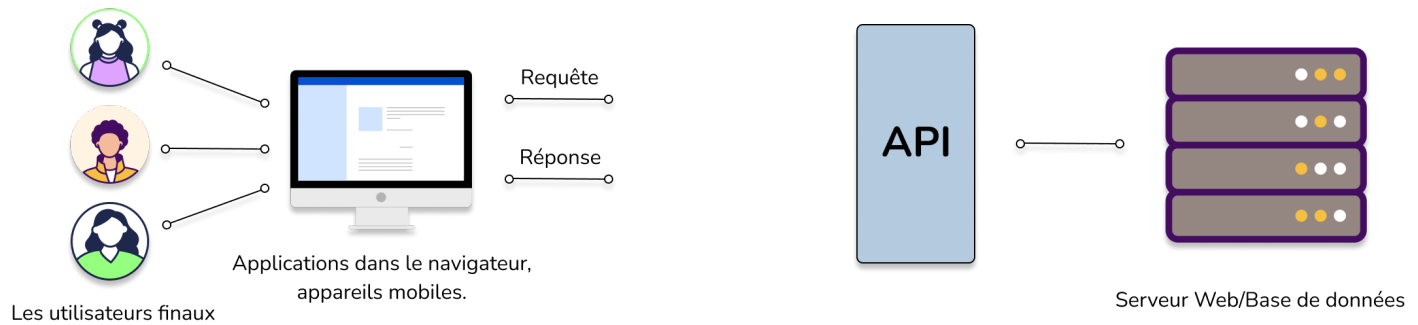


Figure 4.4: Fonctionnement d'un API

La configuration de l'API est une étape essentielle pour permettre une communication efficace entre le frontend et le backend de notre application.

Pour ce projet, nous avons utilisé Symfony avec API Platform pour créer une API RESTful. API Platform offre une grande flexibilité et permet de configurer facilement les opérations CRUD (Create, Read, Update, Delete), la validation des champs, la gestion des relations, les filtres et les groupes de sérialisation. Voici un aperçu de ces fonctionnalités:

- **Opérations CRUD Configurables:** API Platform facilite la configuration des opérations CRUD pour chaque entité. Par défaut, les opérations GET, POST, PUT, DELETE sont disponibles, mais elles peuvent être personnalisées ou restreintes selon les besoins.
- **Validation des Champs:** API Platform intègre le composant de validation de Symfony, ce qui permet de définir des règles de validation directement dans les entités.
- **La gestion des relations entre les entités est simplifiée avec API Platform.** Les relations peuvent être configurées pour être exposées via l'API.
- **API Platform permet d'ajouter facilement des filtres pour les opérations GET.** Ces filtres permettent de rechercher et de trier les données de manière flexible.
- **Groupe de Sérialisation:** Les groupes de sérialisation permettent de contrôler les données qui sont exposées via l'API. Cela est utile pour restreindre les champs visibles selon le contexte (lecture, écriture, etc.).

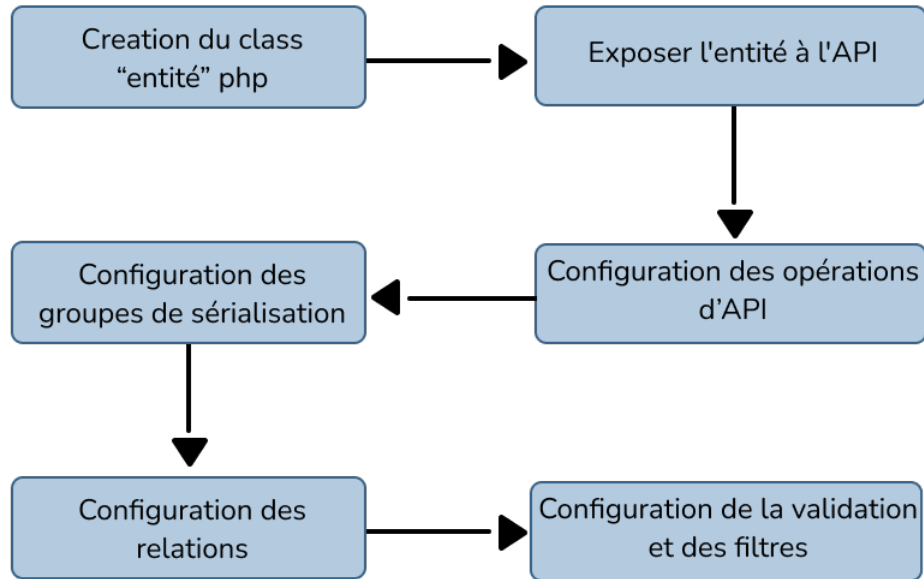


Figure 4.5: Flux général de création d'une entité API

## 4.3 Tableau de bord

### 4.3.1 La gestion des rendez-vous

La gestion des rendez-vous est une fonctionnalité essentielle du dashboard, permettant aux administrateurs et aux médecins de suivre et de gérer efficacement les consultations des patients. Voici les principales caractéristiques de cette fonctionnalité:

- **Vue Calendrier:** Une interface calendrier permet de visualiser les rendez-vous par jour, semaine ou mois. Les rendez-vous sont colorés en fonction de leur statut (confirmé, en attente, annulé).
- **Confirmation et Modification:** Les administrateurs peuvent confirmer, modifier ou annuler des rendez-vous directement depuis le dashboard.

### 4.3.2 La gestion d'équipe médicale

La gestion de l'équipe médicale est une autre fonctionnalité clé du dashboard, assurant une organisation optimale des ressources humaines et des services offerts. Voici comment cette fonctionnalité est mise en œuvre:

- **Liste des Médecins:** Une liste détaillée des médecins avec leurs spécialités, disponibilités et coordonnées. Les administrateurs peuvent facilement ajouter, modifier ou désactiver des médecins.
- **Planification des Horaires:** Un module de planification permet de définir les horaires de travail des médecins.
- **Gestion des Spécialités:** Les administrateurs peuvent gérer les spécialités médicales, en ajoutant ou supprimant des spécialités et en assignant des médecins aux différentes spécialités.
- **Rapports et Statistiques:** Des rapports détaillés sur les activités des médecins, incluant le nombre de consultations, le taux de satisfaction des patients, et les performances par spécialité.

### 4.3.3 La gestion du Chatbot

La gestion du chatbot depuis le dashboard permet de maintenir et d'améliorer en continu l'interaction avec les patients. Voici les fonctionnalités principales de cette section:

- **Configuration des F.A.Q:** Les administrateurs peuvent ajouter ou modifier les réponses pré-définies du chatbot pour les questions fréquentes. Cela inclut la mise à jour des informations sur les horaires, les services et les procédures.
- **Personnalisation:** Des options de personnalisation permettent de modifier le comportement du chatbot, incluant les messages de bienvenue.



## Conclusion

En conclusion, les missions effectuées ont permis de mettre en pratique une large gamme de compétences techniques et organisationnelles. Chaque étape du projet a été l'occasion d'apprendre et de s'adapter, démontrant la complexité et la richesse du processus de développement. Les compétences acquises et les défis surmontés ont non seulement contribué à la réussite du projet, mais ont également enrichi notre expérience professionnelle de manière significative.

# Chapter 5

## Les apports du stage

### Introduction

Le stage chez FCPO a été une expérience riche en apprentissages et en développement professionnel. Ce chapitre examine les différents apports du stage, tant sur le plan théorique qu'intellectuel et pratique. Nous aborderons les compétences techniques acquises, les connaissances théoriques consolidées, et les leçons tirées de l'intégration dans une équipe professionnelle. L'expérience de la gestion de projet et l'application concrète des technologies avancées ont joué un rôle crucial dans notre formation. Ce chapitre vise à offrir une réflexion sur la valeur ajoutée de cette expérience de stage et son impact sur notre développement professionnel.

## 5.1 Les apports théoriques et intellectuels

Durant ce stage, j'ai pu approfondir plusieurs concepts théoriques que j'avais appris lors de ma formation académique. Les principaux apports théoriques et intellectuels incluent:

- **Architecture des applications web:** J'ai consolidé mes connaissances sur les architectures MVC (Model-View-Controller) en travaillant avec Symfony et API Platform. J'ai appris à structurer efficacement une application web pour assurer une bonne séparation des préoccupations et une maintenance facile.
- **Systèmes de gestion de bases de données:** J'ai approfondi mes compétences en conception et gestion de bases de données relationnelles, en particulier avec MySQL. J'ai appris à modéliser des entités complexes et à gérer les relations entre elles de manière optimale.

## 5.2 Les compétences pratiques développées

Le stage m'a offert l'opportunité de développer et de perfectionner des compétences pratiques essentielles pour un développeur. Les compétences pratiques acquises comprennent:

- **Développement Frontend avec React:** J'ai amélioré mes compétences en développement frontend en utilisant React pour créer des interfaces utilisateur interactives et réactives.
- **Développement Backend avec Symfony et API Platform:** J'ai acquis une expérience pratique significative en utilisant Symfony pour le développement backend et API Platform pour la création d'APIs RESTful. J'ai appris à configurer des opérations CRUD, à valider des champs et à gérer des relations entre entités.
- **Gestion de version avec Git et GitHub:** J'ai perfectionné mes compétences en gestion de version avec Git, en utilisant GitHub pour collaborer avec les membres de l'équipe, gérer des branches, résoudre des conflits et suivre l'historique des modifications.
- **Utilisation d'outils de développement:** J'ai optimisé mon environnement de développement en utilisant Arch Linux, Neovim avec une configuration personnalisée, et des outils comme Alacritty, ZSH, tmux, et fzf. Cela m'a permis de travailler de manière plus efficace et productive.

## 5.3 Les apports en termes de monde de l'entreprise

Ce stage m'a également offert une précieuse perspective sur le fonctionnement et la culture d'entreprise. Voici quelques apports significatifs:

- **Travail en équipe:** J'ai appris à collaborer efficacement avec une équipe de développeurs, à communiquer clairement, à partager des tâches et à résoudre des problèmes ensemble. J'ai compris l'importance de la coordination et de la coopération dans la réussite d'un projet.
- **Gestion de projet:** J'ai été exposé aux méthodologies de gestion de projet agile, telles que les sprints et les réunions quotidiennes. J'ai appris à gérer mon temps, à prioriser les tâches et à respecter les délais tout en assurant la qualité du travail.
- **Adaptation aux exigences des clients:** J'ai acquis une meilleure compréhension de l'importance de répondre aux besoins des clients et de s'adapter rapidement aux changements de spécifications. J'ai appris à être flexible et réactif pour fournir des solutions qui satisfont les attentes des utilisateurs finaux.
- **Culture d'entreprise et éthique professionnelle:** J'ai intégré les valeurs et les pratiques professionnelles de FCPO, telles que le respect des délais, la qualité du travail, la confidentialité des données et l'engagement envers les objectifs de l'entreprise. Cela m'a permis de développer un sens aigu de la responsabilité et de l'éthique professionnelle.

## Conclusion

Les apports du stage chez FCPO sont nombreux et variés. Sur le plan théorique, ce stage a permis de consolider les connaissances acquises au cours de nos études, tout en les appliquant à des situations réelles et complexes. Sur le plan pratique, nous avons développé des compétences techniques avancées, appris à naviguer dans un environnement professionnel exigeant, et à collaborer efficacement au sein d'une équipe. Cette expérience a été formatrice à bien des égards, renforçant notre confiance en nos capacités et nous préparant à affronter les défis futurs avec assurance et compétence.

# Chapter 6

## Les difficultés du stage et les solutions apportées

### Introduction

Chaque projet ambitieux comporte son lot de défis, et notre stage chez FCPO n'a pas fait exception. Ce chapitre se penche sur les difficultés rencontrées tout au long du développement du chatbot, qu'elles soient techniques, organisationnelles ou relationnelles. Nous analyserons les problèmes spécifiques auxquels nous avons été confrontés et les solutions mises en œuvre pour les surmonter. Cette réflexion vise à démontrer notre capacité à résoudre des problèmes complexes et à adapter nos méthodes de travail en fonction des obstacles rencontrés. En partageant ces expériences, nous mettons en lumière l'importance de la résilience et de l'innovation dans la gestion de projet.

## 6.1 Les difficultés rencontrées

Voici les principales difficultés rencontrées au cours de mon stage:

- **Utilisation de nouvelles technologies:** Découvrir et maîtriser des technologies comme Symfony, API Platform et React a été un défi initial majeur, car je n'avais qu'une expérience limitée avec ces outils.
- **Gestion dynamique du chatbot:** Configurer le chatbot pour qu'il réponde de manière dynamique et personnalisée aux interactions des utilisateurs a été un défi technique important.
- **Erreurs dues à des drivers manquants:** Des erreurs imprévues causées par des drivers manquants ont perturbé le développement, en particulier lors de l'intégration des fonctionnalités backend.
- **Gestion des domaines:** La configuration et la gestion des domaines pour l'application web ont posé des problèmes, surtout pour assurer une transition fluide entre les environnements de développement et de production.
- **Installation et configuration des versions précises des technologies:** Éviter les conflits entre différentes versions de bibliothèques et frameworks a nécessité une gestion précise des versions, entraînant des ajustements et des configurations détaillées.

## 6.2 Les solutions apportées à ces difficultés

Des solutions spécifiques ont été mises en place pour résoudre les problèmes.

- **Utilisation de nouvelles technologies:**
  - **Formations et documentation:** J'ai suivi des tutoriels et des formations en ligne pour me familiariser rapidement avec Symfony, API Platform et React. La consultation de la documentation officielle et de forums de développeurs m'a également aidé.
- **Gestion dynamique du chatbot:**
  - **Algorithmes personnalisés:** J'ai implémenté des algorithmes personnalisés, pour maintenir un flux de travail tout en obtenant le résultat escompté.

- **Erreurs dues à des drivers manquants:**
  - **Installation de packages manquants:** J'ai identifié et installé les drivers manquants grâce à l'analyse des logs d'erreurs. Une vérification systématique des dépendances avant chaque déploiement a été mise en place.
  - **Documentation et vérification:** J'ai bien lu une documentation détaillée des prérequis et des configurations nécessaires pour garantir que tous les composants requis étaient correctement installés et configurés.
- **Installation et configuration des versions précises des technologies:**
  - **Gestion des dépendances avec Composer et npm:** J'ai utilisé Composer pour PHP et npm pour JavaScript pour gérer les dépendances et m'assurer que les versions spécifiques requises étaient installées.
  - **Fichiers de configuration détaillés:** Des fichiers de configuration (comme composer.json et package.json) ont été utilisés pour spécifier les versions exactes des bibliothèques et frameworks, réduisant ainsi les risques de conflits et d'incompatibilités.



## Conclusion

En conclusion, les difficultés rencontrées durant le stage ont été des opportunités de croissance et d'apprentissage. Les solutions apportées pour surmonter ces défis ont démontré notre capacité à innover et à persévérer face à l'adversité. Ces expériences ont enrichi notre développement professionnel, renforçant notre aptitude à gérer des projets complexes et à collaborer efficacement dans un environnement professionnel dynamique. Les leçons tirées de ces défis nous préparent à affronter les futurs obstacles avec une approche proactive et résiliente.

# Conclusion générale et perspectives

Le stage chez FCPO a été une expérience extrêmement enrichissante et formatrice. Il m'a permis de mettre en pratique les connaissances acquises durant mes études tout en développant de nouvelles compétences dans un cadre professionnel exigeant. Les défis rencontrés ont été nombreux et variés, couvrant des aspects techniques, organisationnels et interpersonnels. Cependant, chaque difficulté a été une occasion d'apprentissage et d'amélioration continue.

Le projet de développement du chatbot pour une maison médicale a été un véritable test de mes capacités à utiliser des technologies avancées telles que Symfony, API Platform et React. La découverte et la maîtrise de ces outils, initialement intimidants, se sont avérées être des atouts précieux pour le projet. Le processus de configuration dynamique du chatbot, en particulier, m'a permis de comprendre l'importance de la conception soignée et de l'optimisation des réponses pour créer une solution interactive et réactive.

L'intégration des différentes entités dans la base de données, ainsi que la configuration des API, a mis en lumière l'importance de la structuration des données et de la gestion des relations entre celles-ci. Grâce à cela, j'ai pu développer une compréhension approfondie de la conception et de l'implémentation de systèmes de gestion de contenu robustes et évolutifs.

En termes de développement personnel, ce stage m'a permis de renforcer mes compétences en résolution de problèmes et en gestion de projet. J'ai appris à travailler efficacement en équipe, à communiquer de manière claire et concise, et à gérer mon temps de manière optimale.

Enfin, ce stage m'a également offert une vision précieuse du monde de l'entreprise. J'ai pu observer de près comment une agence digitale comme FCPO fonctionne, de la gestion des projets à la relation avec les clients. J'ai compris l'importance de la rigueur et de la qualité dans le travail, ainsi que la nécessité de toujours chercher à innover et à s'améliorer.

En conclusion, ce stage a été une étape cruciale dans mon parcours professionnel. Les défis surmontés et les solutions apportées m'ont préparé à aborder avec confiance les futurs projets. Je suis reconnaissant pour cette opportunité et impatient de développer ces compétences dans les projets à venir.

# Annexe A

## Glossaire

**Chatbot:** Un programme d'intelligence artificielle qui simule une conversation humaine.

**Arch Linux:** Une distribution Linux simple et légère.

**KISS:** Keep It Simple, Stupid (Garde-le simple, stupide), un principe de conception affirmant que les systèmes fonctionnent mieux s'ils sont simples.

**Alacritty:** Un émulateur de terminal rapide et multiplateforme.

**Zsh (Z Shell):** Une version étendue du Bourne Shell avec de nombreuses améliorations.

**Neovim (Neo-Vim):** Un éditeur de texte extensible et hautement configurable basé sur Vim.

**GitHub:** Une plateforme web utilisée pour le contrôle de version.

**Postman:** Une plateforme de collaboration pour le développement d'API.

**Figma:** Un outil de design web pour la conception d'interface et le prototypage.

**L<sup>A</sup>T<sub>E</sub>X:** Un système de composition de documents souvent utilisé pour la documentation technique et scientifique.

**Eraser:** Pourrait faire référence à un outil utilisé dans un contexte spécifique, non défini explicitement.

**API Platform:** Une puissante plateforme pour créer des projets orientés API.

**REST:** Représentation État Transfert, un style architectural pour concevoir des applications en réseau.

**CRUD:** Créer, Lire, Mettre à jour, Supprimer, les quatre fonctions de base du stockage persistant.

**MVC:** Modèle-Vue-Contrôleur, un modèle architectural pour implémenter des interfaces utilisateur.

**React:** A JavaScript library for building user interfaces, maintained by Facebook.

**Symfony:** A PHP framework for web applications.

# Annexe B

## Acronymes

<b>SEO</b>	Optimisation pour les moteurs de recherche.
<b>VIM</b>	VI iMproved
<b>API</b>	Interface de programmation d'application.
<b>FAQ</b>	Foire aux questions.
<b>NLP</b>	Traitement du langage naturel (Natural Language Processing).
<b>IA</b>	Intelligence artificielle.
<b>MCD</b>	Modèle Conceptuel de Données

# Ressources

- [1] React: <https://react.dev>
- [2] Symfony: <https://symfony.com>
- [3] The Odin Project: <https://www.theodinproject.com>
- [4] Symfonycasts: <https://symfonycasts.com>
- [5] React Chatbotify: <https://react-chatbotify.tjtanjin.com>
- [6] React Chatbotify (GitHub): <https://github.com/tjtanjin/react-chatbotify>
- [7] MDN: <https://developer.mozilla.org/en-US>
- [8] ChatGPT: <https://chat.openai.com>
- [9] Google: <https://www.google.com>
- [10] Stack overflow: <https://stackoverflow.com>
- [11] YouTube: <https://www.youtube.com>
- [12] Arch Wiki: <https://wiki.archlinux.org>
- [13] Overleaf: <https://www.overleaf.com/learn>