



Rapport de stage

12 mai 2025 - 4 juillet 2025

EMBOUAZZA ADNANE SIO 1B



BTS SIO SLAM Première Année

Tuteur de stage : Pascal ATSU

Professeurs principaux : Jean-François VERT & Hervé PIERRON VIALARD

SOMMAIRE

• Page de Garde	1
• Remerciements	3
• Introduction	4

Présentation de l'entreprise :

• Présentation de l'entreprise	5
• Organigramme et Hiérarchie	6

Taches Orienté Architecture et Réseaux Informatique :

• Déplacement de serveur et Gestion / Administration des VPS	7
• Audit des recommandations informatique pour l'entreprise	8
• Projet d'Observabilité	9

Taches Orienté Développement Informatique :

• Projet d'intelligence artificiel pour la sélection de CV	11
• Développement d'une application Web de comptabilité	12
• Conclusion	21

REMERCIEMENTS

Avant de débuter ce rapport de stage , je tiens tout d'abord à présenter mes plus sincères remerciements à l'équipe de Vademi , à Melun , pour l'accueil qu'ils m'ont réservés ainsi que pour leur conseil , pour l'accompagnement et la patience qu'il ont pu faire preuve à mon égard , tout particulièrement mon maître de stage et dirigeant de l'entreprise , Monsieur Pascal ATSU , qui fut d'une grande aide durant toute la période du stage .

Je profite également de cette section pour remercier aussi les équipes pédagogiques du lycée Turgot , notamment les professeur principaux de la classe de SIO 1B , Monsieur Pierron Vialard et Monsieur Vert pour leur enseignement , leur conseil et leur disponibilité avant , pendant et après la période de stage .

INTRODUCTION

Dans le cadre de ma formation, j'ai eu l'opportunité d'effectuer un stage au sein de l'entreprise Vademi , situé sur Melun ,sous la tutelle de Monsieur Pascal ATSU , dirigeant de l'entreprise , et ce sur une période allant du 12 Mai 2025 au 4 Juillet 2025. Ce stage s'inscrit dans une démarche d'apprentissage visant à mettre en pratique les connaissances acquises durant mon cursus, tout en découvrant le fonctionnement concret d'un environnement professionnel, en particulier dans un domaine directement lié à mes études.

Intégré au sein du service informatique de Vademi, ma mission principale a été la conception d'une application web dédiée à la gestion comptable de l'entreprise. Ce projet revêtait une réelle dimension professionnelle, car il avait pour objectif, si sa qualité le permettait, d'être proposé par la suite comme offre commerciale destinée à d'autres entreprises ou particuliers.

Par ailleurs, j'ai également pu participer à diverses missions en lien avec le développement informatique ainsi qu'avec l'architecture et l'infrastructure réseau, ce qui m'a permis de renforcer mes compétences techniques et d'élargir mon champ de connaissances. Parmi ces tâches, on peut citer :

- Le déplacement d'un serveur physique et la gestion/administration des VPS qui y sont associés ;
- L'audit des recommandations informatiques de l'entreprise et de ses clients ;
- Un projet d'observabilité numérique, comprenant le déploiement et l'automatisation des journaux de serveurs et services ;
- Un projet d'intelligence artificielle pour la sélection automatisée de CV ;
- Et enfin, comme mentionné ci-dessus, le développement de l'application web de comptabilité.

Cette expérience m'a permis d'approfondir mes connaissances, de découvrir les réalités du milieu professionnel, et de mieux cerner mes aspirations et ambitions pour l'avenir.

Ce rapport a donc pour objectif de présenter dans un premier temps l'entreprise, son environnement et ses activités, afin de poser un cadre clair, avant de détailler le déroulement de mon stage, les missions qui m'ont été confiées, les compétences mobilisées ainsi que les enseignements que j'en ai tirés.

PRESENTATION DE L'ENTREPRISE

.CONTEXTE

Vademi est une entreprise située sur Melun, créée en 2014 par son PDG actuel, monsieur Pascal Atsu. Il s'agit d'une entreprise spécialisée dans la protection numérique et la cybersécurité de ses clients. Elle met en œuvre pour ses partenaires, grande, moyenne et petite entreprise des solutions et propositions pour leur protection et leur développement numérique.

Elle étend ses activités sur 4 domaines particulier :

- Comme précisé précédemment, la cybersécurité
- Les systèmes d'informations (SI)
- L'audit et le conseil informatique
- la télécommunication

.PLUS QU'UN CONSEILLER INFORMATIQUE

En 2020, son chiffre d'affaire s'élevait à 2,6 millions d'euro et parmi ses partenaires commerciaux et collaboratif important on peut citer notamment la banque de France, l'entreprise Socos Services, l'entreprise Safran ou encore la Banque BNP PARIBAS.

Grâce notamment à leur partenariat stratégiques, Vademi est aussi reconnu par la DRIEETS comme un centre de formation, sous le numéro 11770891777, et qui propose les certifications et formations suivante :

- | | |
|--------------------------------|--|
| - RGPD Certified DPO | - ISO/IEC 27001 Lead Implementer |
| - ISO 27001 Foundation | - ISO/IEC 27701 - Lead Privacy Implementer |
| - ISO/IEC 27001 Lead Auditor | - ISO/IEC 27032 - Lead Cybersecurity Manager |
| - ISO/IEC 2700 A5 Risk Manager | |

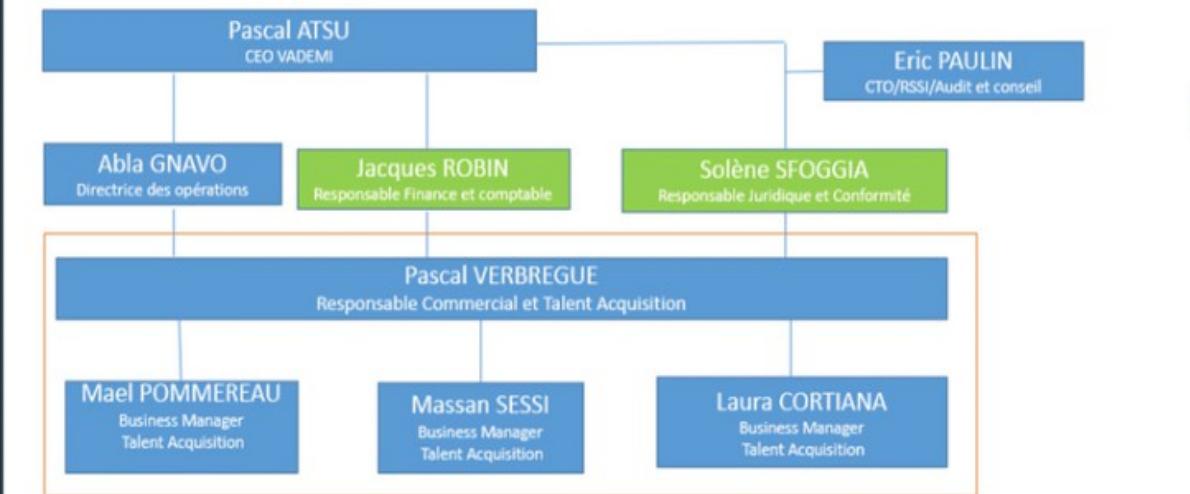
. SERVICES ET SOLUTIONS PROPOSE PAR L'ENTREPRISE

Parmi les services importants proposés par Vademi on retrouve:

- Le service CybersGRC®, qui est là pour aider à mettre en place des politiques de sécurité et confidentialité robustes, et conforme aux réglementations officielles (RGPD...) via des services d'assistance RSSI, d'accompagnement SMSI ou d'analyse SSI
- Le service CyberSOC®, qui assure la détection et la réponse aux incidents de sécurité en temps réel, avec des experts prêts à intervenir à tout moment en réponse aux incidents et menaces
- Le service CyberAudits®, qui permet d'évaluer la sécurité des infrastructures informatiques, en identifiant les failles potentielles et en proposant des solutions concrètes pour y remédier, cela passe par des plans d'action détaillé ou encore des recommandations personnalisées.

NOTRE CAPACITE D'ACCOMPAGNEMENT

Une organisation dédiée aux besoins de nos Clients



Organigramme de l'entreprise

Quelques ressources Utiles :

Interview du PDG de Vademi :

<https://www.lefigaro.fr/economie/vademi-l-expertise-en-systeme-d-information-et-cybersecurite-au-service-des-entreprises-20220127>

Page Linkedin de l'entreprise :

<https://fr.linkedin.com/company/vademi-sas>

Site de présentation de Vademi :

<https://www.vademi.com/>

Données statistiques de l'entreprise et de son activité :

<https://www.pappers.fr/entreprise/vademi-807682232>

Déplacement d'infrastructure réseau (Serveur) et Gestion / Administration des VPS

La première mission qui me fut attribué lors de mon stage fut la suivante : le débranchement et le déplacement d'un serveur d'hébergement pour des VPS (soit des serveurs virtuel , sur des Machine Virtuel) .Le dit Serveur situé en haut de la société au dernier Étage a du donc être déplacé vers le sous sols , dans la salle des serveurs .

Pour cela , j'ai du avant tout m'assurer de la disposition de l'infrastructure réseau , c'est à dire comment le serveur été connecté , à quel fil , sur quel port et enfin vers quel Switch ?

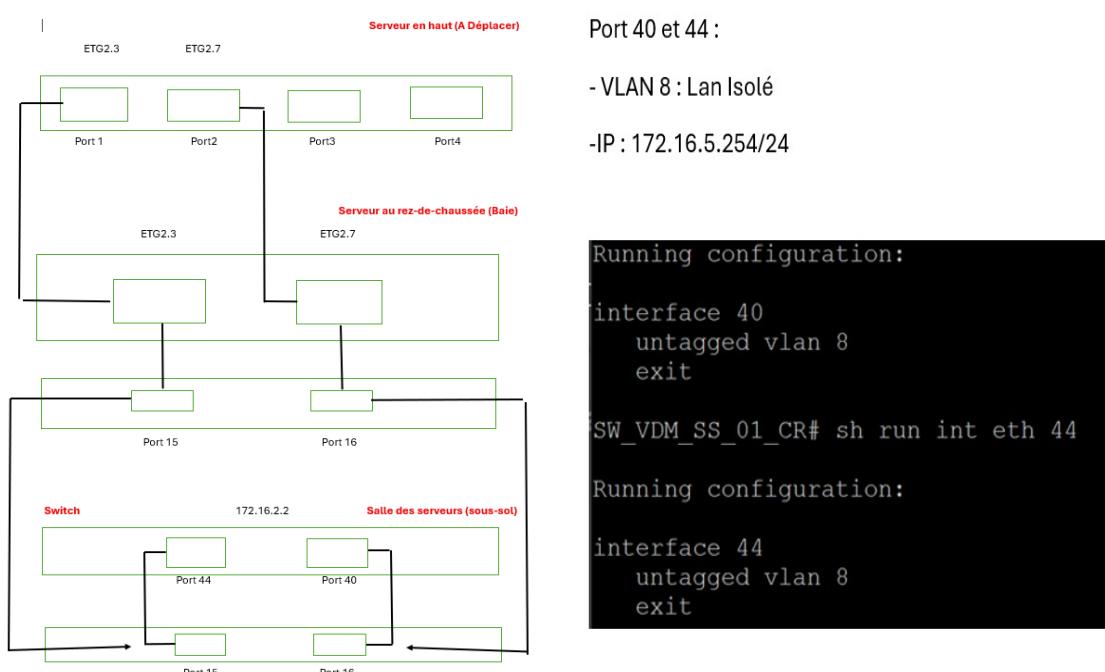
En effet ,Une erreur de ma part aurait pu être problématique lors du déplacement du serveur car un branchement sur un mauvais port aurait pu compromettre la disponibilité d'une certaine partie des services de l'entreprise , et ce pendant un temps indéterminé, empêchant donc mes collègues sur d'autres projets de travailler sur le réseau pendant au moins plusieurs heures ce qui représente une perte de productivité et de bénéfices certaines pour l'entreprise , ce qui n'est en aucun cas envisageable .

Pour éviter cela à tout pris , j'ai du donc dressé un schéma réseau de l'architecture de l'entreprise afin de savoir exactement le chemin que prenait mon serveur jusqu'ici entre le dernière Étage et la salle des serveur au sous sols .

l'architecture est composé de 3 niveau :

- En haut se trouve le serveur à déplacer .
- La Baie au rez de chaussé qui permet de relier le serveur à déplacer au port 15 et 16
- La salle des sous-sols ou se trouve les serveurs et les Switch

Voici donc le schéma que j'ai réalisé avant le déplacement du serveur , ainsi que la configuration des port concernés par le déplacement :



Une fois ceci fait , et avant le déplacement du serveur vers le sous sols , les VPS présente sur ce derniers se doivent d'être correctement configuré afin de pouvoir les arrêter et rallumer proprement une fois le serveur physique déplacé . Pour cela ils faut suivre la démarche sur VMWare et éteindre les VPS (serveurs virtualisé) un à un avant de les faire rentrer en maintenance .Durant ce procédé , il faut vérifier tout anomalie lié a ces derniers qui pourrait entraver le rallumage ultérieurement , par la j'entends les problèmes de réseau , de fonctionnalité , de configuration ...

Après quoi le déplacement du serveur est enfin effectué , avant de rallumer tout les VPS , puis pour finir , il faut vérifier la configuration réseau de chaque VPS à nouveau afin de s'assurer qu'il n'y est pas de problème de connexion sur ces derniers , pour se faire on vérifie le paramétrage IP , le DNS et le DHCP sur chaque serveur . Une fois cette étape complété , l'infrastructure concerné peut de nouveau être exploité par les équipes et la production peut reprendre

Dans cette tache j'ai appris l'importance de la connaissance de l'infrastructure informatique et réseau dans laquelle on travaille afin de répondre au besoins des équipes et de l'entreprise sans perturber le système de cette dernière . De plus j'ai pu mettre en pratique en conditions réels les enseignements que j'ai reçu du point de vu SISR (infrastructure et réseau) pour la première fois et me familiariser avec le fonctionnement de serveur et d'une infrastructure réseau informatique professionnels

Audit de la présentation des recommandations informatique de l'entreprise et de ses clients

Au cours de la même semaine que ma première mission ,j'ai reçu une seconde tache. Cette dernière consistait à d'auditer des documents de présentation qui seront utilisé pour expliquer les enjeux, objectifs et mesures futures qui seront mise en place par l'entreprise , dans le cadre de son activité et de ses collaborations avec ses partenaires.

Pour cela , on m'a transmis une présentation sous format powerpoint qui serait utilisé lors d'une réunion et qui comporterait les tenants et aboutissement que j'ai abordé juste précédemment . J'ai donc du relire le document entièrement afin de vérifier la cohérence des informations entre les différentes ressources que l'on a mit a ma disposition pour cette mission (tableur , données à exploiter ...) . J'ai du trier de nouveaux les recommandations et mesure proposés en fonction de plusieurs critères :

- Leur coût globale a l'entreprise
- leur difficulté et les efforts nécessaires à mettre en place
- leur intérêt / valeur Ajouté pour l'entreprise

Ces Trois critères ont donc servi à déterminer l'ordre de priorité de la mise en place des mesures chronologiquement , soit de leur dates sur le calendrier de l'entreprise .

Cela me donne donc 4 catégories :

- **P1** : appellés les Quick Win => ce sont les mesures simples de mise en place et utile à l'entreprise immédiatement (celle qui sont mis en place en priorité => ~ Mai - Juin 2025)
- **P2** : les mesures dont la mise en place et la rentabilité sont moindre mais tout de même intéressant et non négligeable (~ Octobre – Décembre 2025)
- **P3** : les mesures dont la mise en place demande un certains investissement sans être vitales ou extrêmement rentable dans l'immédiat (Janvier – Mars 2026)
- **P4** : les mesures qui demande de gros efforts /investissements pour leurs mise en place mais dont la rentabilité est grande (Juin – Septembre 2026)

Une fois ceci fait , On dresse une ligne d'action à jour avant de la renvoyer prête à l'utilisation . Ce travail peut paraître manqué de pertinence au premier abord mais son importance n'est absolument pas à sous estimer car il permet de mettre en place un plan d'actions clair et optimisé pour l'entreprise , lui permettant donc d'atteindre ses objectifs de la manière la plus efficace et donc de favoriser son développement sur le long terme .

Projet d'Observabilité: Déploiement et automatisations de serveurs et services Pour les Clients

Lors de mon arrivé dans l'entreprise il y avait déjà un projet en cours sur laquelle travaillait la devOps de la boite , que par souci d'anonymat nous appellerons « Sarah » . Cette dernière travaillait sur un projet d'observabilité faisant intervenir plusieurs technologies différentes et variés ,et au vu des compétences que ce dernier faisait intervenir , qui change de ce qu'on a pu apprendre jusqu'ici dans nos apprentissage scolaire aussi bien au niveau de la diversité que de la complexité , j'ai jugé intéressant d'en parler dans mon rapport .

L'Objectif de ce projet est donc de permettre l'observation de toutes les informations relative au serveur et au service (logs , composants , capacité utilisé ...)Pour Vademi et pour ses clients/partenaire et ce à tout moment. Le projet a donc comme l'une de ses contraintes principales une disponibilité constante et en temps réel des informations .

Je n'ai pas participé directement dans ce projet (par la je veux dire que je n'ai pas touché au projet directement moi même) cependant j'ai accompagné Sarah sur le projet pendant une partie de mon stage et est donc pu moi même me familiariser avec les enjeux et aboutissement , ainsi que les spécificité du projet . j'ai également participé à toutes les réunions concernant le projet . De ce fait je me suis tout de même assez investi sur ce dernier pour me permettre d'en parler ici .

Le projet se construit autour de l'utilisation de WorkFlows tout au long du processus. Les Workflows mis en place sont donc la pour automatiser une partie des taches du projet , comme la remonté des logs et des données des infrastructure concernés , l'envoi et la programmation des alerte selon des critères définis préalablement , ainsi que la personnalisation des mails et différent message à envoyer selon les circonstances (message d'avertissement , mail automatique ...) . Leur utilisation est primordiale tant elle permet d'élargir le champs d'action du système d'observabilité , et de minimiser les ressources techniques et humaines à mettre à disposition pour la maintenance , surveillance et utilisation de ce dernier .

Quand à l'application , au delà de l'utilisation de workflows , elle se construit autour de l'utilisation de plusieurs services avec des missions bien précises pour chacun , allant de la réception des données à leur traitement/ mise en forme afin de faciliter et d'expliquer leur lecture, la finalité étant toujours de pouvoir remonter les données de tout les services et infrastructure quel que soit leur types et caractéristiques. Cela inclut donc les composants et matériel informatique (CPU , SNMP , MEM , serveur , caméra ...) mais aussi les services et autres spécificité utilisé par l'entreprise (connexion SSH , service téléphonie ...)

Pour cela , différentes technologies rentre en compte :

- Ansible : Ansible est utilisé dans le cadre de la configuration des ressources . Il permet d'automatiser les configurations utilisé par les serveurs et les services lors de leur déploiement. Il permet un gains de temps considérable sur la mise en place des services et ressources nécessaire pour le projet .
- Terraform :Terraform quand à lui ,sert à automatiser le déploiement direct des ressources . Tout comme Ansible son utilisation permet de gagner un temps non négligeable tout en permettant d'éviter le déploiement manuelles des serveurs et services .
- Telegraph : Il s'agit du serveur qui permet la transmissions des informations et des journal aux serveur AnfluxDB
- AnfluxDB : le serveur AnfluxDB est celui qui reçoit les journal et les trie en fonction de leur origine (SNMP , CPU , SSH ...) de cette manière il peut ensuite les envoyer au serveur Grafana
- Grafana : le serveur Grafana est celui qui gère l'affichage des Journal remonté par le serveur AnfluxDB de manière claire et esthétique , il est celui qui forme les tableau de bord qui pourront être exploité par les clients utilisateur du servies et qui gère le système de statuts des système (RUN , WARNING , ERROR , DEAD) et le système d'alerte en cas de problème

Un système d'automatisation est aussi mise en place pour le hachage des mots de passe et la gestion des informations utilisateur (mdp.yml) , ainsi que pour l'ajout de sources de logs à remonter (pour de potentiel client à qui proposer le services , avec le traitement de leur propre serveur).

Dans ce projet , la maîtrise des lignes de commande Linux et des WorkFlows ainsi qu'une connaissance raisonnable des service de déploiement et de traitement utilisé est primordiale car il permettent la mise en place des différents système pour la gestion des fonctionnalité et des problèmes / erreur potentiellement rencontré .

Dans la prochaine partie nous nous concentrerons maintenant sur les taches orienté vers le développement informatique car ce sont celle qui sont censé être les plus enrichissante et intéressante au vu de la spécialité du BTS choisi (option SLAM) , en commençant par le projet de développement d'une intelligence Artificiel pour faciliter la sélection de CV , pour l'entreprise . Après quoi nous parlerons de la mission qui ma demandé le plus d'investissement personnel et de temps , à savoir le développement d'une application web pour la gestion de la comptabilité de l'entreprise .

Projet d'intelligence artificiel pour la sélection de CV

Je ne m'attarderais pas sur ce projet car, comme celui d'observabilité, je n'y ai eu qu'une influence superficielle et n'est pas vraiment participé aux opérations pour la mise en place de ce dernier. De plus à l'heure où j'écris ce rapport, le projet est encore assez loin d'être complètement abouti et n'en est encore qu'à sa genèse. Cependant j'ai bien pu assister aux réunions sur le projet et je trouve qu'il est tout de même intéressant de s'y pencher quelque temps car il m'a donné une perspective sur ce à quoi peut ressembler un projet ambitieux et collectif en entreprise.

Avant tout, je me dois de présenter le projet, ses objectifs et ses enjeux vis-à-vis de l'entreprise afin de donner une idée plus précise de la situation. Comme le titre l'annonce, le but de ce projet est de produire un produit : une intelligence artificielle dont le but serait de permettre l'établissement d'une présélection de Curriculum Vitae (des CV) selon des critères prédefinis par l'entreprise en fonction de ses besoins, attentes et de ses objectifs. Le but est donc d'automatiser une partie du processus grâce à cette IA et de rendre le travail des commerciaux chargés du recrutement plus simple, plus efficaces et plus productifs. De plus l'intelligence artificielle a aussi pour but, en plus d'établir cette pré-sélection dans les bases de données déjà existante du groupe d'alimenter ces dernières en pouvant aller chercher elle-même les profils intéressants et encore une fois correspondant à la direction de l'entreprise, et ce directement sur des plate-formes regroupant de la main d'œuvre et des profils en recherche d'emplois (Indeed, HelloWork, LinkedIn ...)

Pour répondre à cette demande, l'entreprise a fait appel à un professionnel externe, que nous appellerons ici par des souci d'anonymat « Axel ».

Donc après être en contact avec Axel, qui lui-même à récupérer cette tache d'un de ses collègues avec lequel l'entreprise avait aussi été mis en contact, Axel a pu exposer les différentes solutions, restrictions et potentielles mis au point à faire sur le projet afin de lui donner un cadre clair et précis (ex : est ce qu'il était préférable d'utiliser pour l'entreprise , dans le cadre de la source de stockage des CV et de leur tri / classement , un service externe populaire et efficace mais coûteux ou une solution développé en internes , mais potentiellement plus restrictive en termes de fonctionnalité . Comment délimiter concrètement le champ d'action de l'IA, jusqu'à quelle étape du processus est-il judicieux de considérer l'automatisation par cette dernière et quelles parties doivent tout de même absolument être laissé à la surveillance humaine, avec quel outil la construire ...).

Une fois cette étape passée et les questions concernant la facturation et le budget pour ce type de projet réglé, Axel a pu passer à l'étape suivante et nous présentait le workflow qui servirait de base pour la construction du robot (le dit workflows à été repris de l'ancienne personne et réadapté pour mieux correspondre à la vision de l'entreprise) Pour cela, Axel a utilisé un outil en ligne qui se nomme n8n, préconisé au sein même de la boite.

N8N est donc un outil de « low-code » qui permet la construction de workflows complexes, afin d'automatiser certains processus ou manœuvre dans la chaîne de production d'un projet. Il est très populaire et prisé en entreprise, en plus de permettre la compatibilité avec énormément d'outils et d'application externes grâce à l'utilisation d'API. Il s'agit donc de l'outil qui servira de base pour ce projet.

J'ai trouvé assez intéressant de parler de ce projet même si je n'y participerais pas directement (le projet étant encore à sa phase de préproduction) car ils a pu me servir d'introduction personnelle au workflows et montrent leur importance dans la méthodologie professionnelle et à quel point ce sont des outils avec lesquels il faut se familiariser pour pouvoir ouvrir son éventail de compétence et de possibilité au sein de projet qui requiert des tâches automatisées ou des processus répétitif et plutôt complexe comme ici.

Projet Personnel : Développement d'une application Web , permettant de gérer la comptabilité de l'entreprise et de créer / imprimer des Factures / Devis

Pendant la majorité de mon stage, j'ai travaillé en autonomie sur le développement complet d'une application web destinée à la gestion de devis et de factures. Ce projet représente l'axe principal de mon stage, sur lequel je me suis investi durant plusieurs semaines. L'objectif de cette plate-forme est de permettre aux entreprises et à leurs utilisateurs de créer, modifier, visualiser et suivre l'ensemble de leurs documents commerciaux, tout en garantissant une gestion structurée et sécurisée des données. Pour développer cette application, j'ai choisi d'utiliser FastAPI, un Framework Python moderne, rapide et adapté à la création d'API REST qui compose la majorité de mon backend. Les raisons derrière ce choix sont multiples mais simple. Tout d'abord je voulais tout simplement expérimenter et découvrir l'utilisation et les spécificités liés au Framework, de plus je voulais également éviter d'utiliser à nouveau PHP comme on l'a fait tout au long de cette année scolaire, et approfondir mes connaissances sur un autre langage afin d'élargir mon panneau de compétence. Ajoutons aux raisons qui m'ont poussé vers ce choix le fait que FastAPI est facile d'accès pour les amateurs non initié à l'utilisation de Framework et qu'il correspond parfaitement au besoin que j'avais pour cette application, tant il facilite la construction et les tests d'API REST par la présence d'un swagger, documentation graphique auto-complété qui me permet de tester mes routes sans nécessiter le développement d'une vue par mes propres soins :

The screenshot shows the FastAPI Swagger UI interface. At the top, it displays "FastAPI 0.1.0 OAS 3.1" and "openapi.json". On the right side, there is a green "Authorize" button with a lock icon. Below the header, there's a section titled "default". Under "default", there is a list of endpoints for the "utilisateurs" endpoint:

- GET /me** Lire Profil
- GET /utilisateurs/** Get Utilisateurs
- POST /utilisateurs/** Create Utilisateur
- GET /utilisateurs/{utilisateur_id}** Get Utilisateur By Id
- PUT /utilisateurs/{utilisateur_id}** Update Utilisateur

The "Create Utilisateur" (POST) and "Update Utilisateur" (PUT) methods are highlighted with a green background, indicating they are the focus or active methods. The "Delete Utilisateur" (DELETE) method is shown but has no visible highlighting.

Capture d'écran du swaggers et de routes de tests

La base de données relationnelle a quant à elle été modélisée avec **SQLModel**, une bibliothèque / ORM combinant les avantages de SQLAlchemy (un autre ORM python) et de Pydantic, ce qui m'a permis de valider les données tout en structurant efficacement les tables. Les raisons de ce choix sont assez similaires à celles évoquées ci-dessus : une prise en main facilité (SQLModel est fait justement pour cela) ainsi que l'envie de tester et découvrir l'utilisation d'un ORM, technologie de plus en plus démocratisée dans le monde professionnel du développement, et ce au sein d'un projet web avec une base de données.

A la demande de mon responsable direct, la personne à l'origine de l'initialisation du projet, ce dernier devait pouvoir, si le produit final était de bonne qualité, se vendre à des particuliers. De ce fait il devait respecter un certain nombre de contraintes et de conditions et j'ai vite dressé une esquisse de cahier des charges afin de centrer le projet et ses objectifs, et voici donc une petite liste exhaustive des critères que j'ai noté :

HEBERGEMENT

- Service d'hébergements (à déterminer)
- Nom de sous domaine (à déterminer)

PROGRAMMATION

- Langage de programmation (à déterminer) : Node.js, Python, PHP ?
- Utilisation d'un Framework ? (FastAPI, Laravel, Express.js...)
- Architecture Web ? (MVC ...)
- Besoin potentiellement d'API (ex : Authentification forte MFA via envoi de mail de confirmation)
- Utilisation de code procédurale ou de POO ? (À déterminer)
- Utilisation potentiel de workflows (A déterminer) (ex : créer un devis → envoyer → transformer en facture → marquer comme payé)
- Intégration d'un IDE adapté ? (PyCharm pour Python, PHP Storm pour PHP ...)

BESOIN GENERAL DU SITE :

- Possibilité de créer un compte + Système de connexion / Déconnexion
- Opération CRUD sur la plupart des fonctions du site (Ajouter, Modifier, Supprimer et Lire)
- Organisation logique et pertinente des données et informations sur la vue.
- Un système de rôles / droit (ADMIN, Utilisateur ...)

extrait du cahier des charges

Parmi ces éléments, les critères que je me devais de remplir sont donc les suivants :

- Un site dynamique, soit exigeant la présence d'une base de données, pour laquelle j'ai opté pour PostgreSQL, qui après recherche me semble le choix le plus approprié dans ce contexte.
- Un système de rôle (utilisateur / admin ...) pour délimiter les responsabilités, droits et interdictions des différents utilisateurs
- Un système de connexion et déconnexion ainsi qu'une authentification forte MFA (système de code de validation envoyé par mail)
- Transformation des factures/Devis en PDF avec le logo de l'entreprise (classe déjà existante)
- Une interface graphique stylisé et idéalement responsive

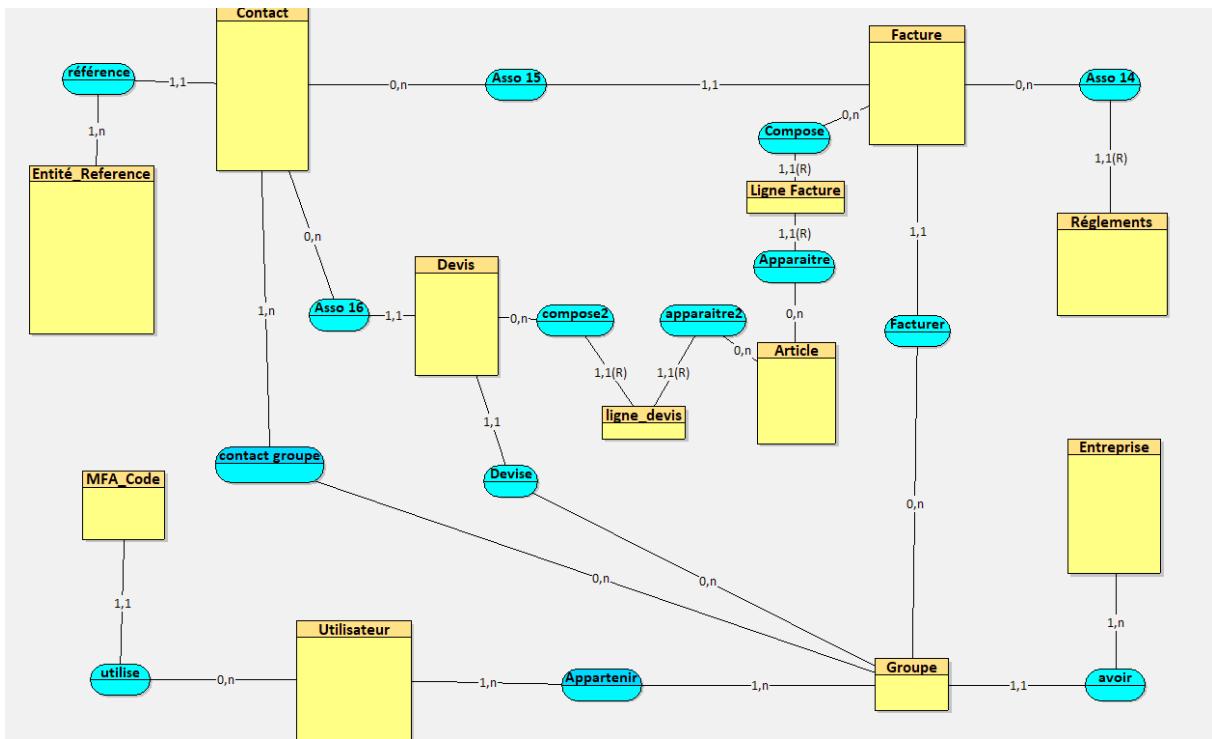
Dans le cadre de mon stage, je ne me suis concentré majoritairement sur la partie Backend et n'ai pas eu le temps de terminer la partie Frontend (bien que j'ai tout de même réussi à en faire une majorité, dont je vous montrerais des extraits via des captures d'écran) Cependant comme j'ai pu le préciser en début de cette partie, j'ai travaillé seul sur ce projet et est donc réalisé l'ensemble des étapes du processus, parmi lesquelles :

- La conception du schéma relationnel et la création de la base de données ;
- L'implémentation des modèles Python avec validations intégrées ;
- Le développement des EndPoints API pour les opérations CRUD (Create, Read, Update, Delete) ;
- L'intégration des règles et contraintes métier à ces dit EndPoints afin de délimiter clairement leur capacité et utilisation .
- La mise en place d'un système d'authentification sécurisé avec vérification en deux étapes (MFA) par email, et d'impression PDF pour les devis et factures.
- La majeur partie de la conception d'un Frontend qui mettent en valeur visuellement les données.

L'utilisation de workflows pour automatiser certains processus (ex : créer un devis → envoyer → transformer en facture → marquer comme payé) a été envisagé mais le manque de compétence et de connaissance ainsi que le manque de temps afin d'avoir un apprentissage suffisant sur le sujet m'ont finalement fait abandonner cette initiative (la fonctionnalité pourra toujours être implémenté par l'entreprise elle-même plus tard)

Pour la conception et la modélisation de la base de données, l'application utilisée jusque-là par l'entreprise et qui m'a servi de site de référence se nomme DBF Audit. Il s'agit donc d'un site qui permet de gérer l'aspect comptable de son entreprise , d'enregistrer des contact et de leur transmettre des documents de comptabilité (Factures , Devis , Article ...).De ce fait , n'ayant bien évidemment pas accès à la base de données de ce site , et ayant pour but afin de satisfaire les attentes de mes responsables , de me rapprocher le plus possible de ce site professionnels en terme d'architecture et de fonctionnalité , la première étape dans ce projet pour moi a été l'étude approfondi du site internet référent (grâce à des droits administrateur fourni sur l'interface réservé à l'entreprise) . Par cela, je veux dire que l'objectif était, en observant le site et son fonctionnement, de comment ses différentes fonctionnalités se comportent individuellement et pris en groupe, de comprendre le fonctionnement potentiel d'une base de données pouvant répondre aux caractéristiques de ce genre de site et d'en émuler une base de données qui servirait de base de ce projet.

Bien sûr pour des raisons de confidentialité pour l'entreprise, je ne vous montrerais pas le modèle de la base de données avec les informations sensible. Cependant, voici une copie de sa structure sans ces dernières (nom des champs notamment) pour que vous puissiez vous faire une idée de la forme de cette dernière tout en préservant l'original de toute exposition :



Modele conceptuel de la base de données (MCD)

Comme vous pouvez l'observer sur cette photo je suis au final arrivé à une base de données composé de 13 tables, ce qui me semblait le plus proche du site référent en termes des fonctionnalités qui devront être programmer et implémenter. C'est donc la structure de base de données sur laquelle je suis partie pour ce projet.

Un fois cela fait , il était temps de passer sur l'environnement de développement et de commencer à construire des routes API sur la base du CRUD (Create , Delete ,Update , Read) .On définit tout d'abord une classe pour la connexion à la base de données que l'on relie grâce à un fichier « .env.» qui contient les information de notre base de données .Pour cela on utilise la librairie **dotenv** ainsi que la fonction **load_dotenv()** pour charger les variable d'environnement du fichier « .env » que l'on associe ensuite à des variable dans notre code grâce à **os.getenv()**. Etant donné qu'il s'agissait de la première fois que j'utilisais un Framework, et que je n'étais pas non plus très à l'aise avec python comme langage de prédilection, les 2 premiers jours se sont résumé à une découverte du fonctionnement du Framework et de l'ORM associé, à savoir pour ce projet SQLModel, qui se base lui-même sur SQLAlchemy. Après m'être accommodé avec ces technologies peu familière pour moi, j'ai enfin pu commencer à construire mon projet avec les premières routes et modèles. Pour les modèles, j'ai les ai construits de cette manière ,comme une succession de classe qui reprennent les attributs des entités de ma base de données, avec l'ORM qui fait le lien entre les champs de la base et ceux des modèles :

Donc par exemple, imaginons une entité utilisateur dans la base de données, avec des champs classique (id, mail, mot de passe, nom, prénom, rôle) voici comment se présenterait le fichier du modèle de cette table :

- Une classe (si besoin) pour gérer les cas particulier (ex : les champs qui dépendent d'un Enum en base de données) Cette classe DOIT porter le même nom que le champ concerné dans la base
- Une classe qui regroupe les attributs partagés entre tous les modèles, que l'on peut nommer UtilisateurBase (SQLModel) (on lui fait hériter de SQLModel la classe qui définit l'ORM)
- Les classes UtilisateurRead / Update / Create à qui on fait hériter soit la classe Base (pour READ et CREATE) ou bien de nouveau SQLModel (pour UPDATE)
- Une classe Utilisateur à qui on transmet UtilisateurBase et qui doit faire le lien entre le modèle et la table SQL (il s'agit du modèle ORM et la caste est très importante ici !)

Il est très important aussi de bien gérer les imports nécessaires pour éviter tout import circulaire ! (Python et fastAPI sont très sensibles à cette problématique)

Lors de la construction de nos routes, nous allons donc nous servir de ces modèles, UtilisateurRead pour l'affichage (pour masquer des données sensibles comme le mot de passe) UtilisateurCreate pour la création et UtilisateurUpdate pour la mise à jour / modification. Le Modèle ORM quant à lui ne doit jamais être directement affiché où retourner par nos routes.

```
@router.get("/utilisateurs/", response_model=List[UtilisateurRead])
async def get_utilisateurs(session: AsyncSession = Depends(get_session)) -> List[UtilisateurRead]:
    result = await session.execute(select(Utilisateur))
    utilisateurs = result.scalars().all()
    return utilisateurs
```

Exemple très simple d'une route potentielle selon l'exemple ci-dessus

La même logique est suivie pour toutes les routes de toutes nos tables de la base de données , tout en y appliquant à la fois la gestion des droits(un utilisateur n'a pas accès au même actions qu'un admin) et les règles métiers (exemple : le compte d'un utilisateur ne peut être supprimé que par lui-même , une entreprise que par son créateur ...)Une fois ceci fait , on apporte la route qui gère la connexion de l'utilisateur et on adapte nos routes en conséquence (ex : il faut être connecté pour avoir accès aux contenu sensible du site)

Problème rencontré lors du Processus

Puisque ce stage a pour but avant tout l'apprentissage et la professionnalisation du stagiaire je pense qu'il est important de parler quelque peu des problèmes rencontrés durant ce processus avant de passer à la suite.

Mis à part les erreurs de code basique qui ont pu être corrigé sans trop de difficulté, et qui sont normal dans un projet de cette envergure, le plus gros problème qui m'est arrivé a été une incohérence dans la structure de ma base de données. En effet, l'une des exigences techniques de ce site était qu'une entreprise puisse au sein de sa structure posséder plusieurs groupes, et que ces groupes indépendamment les uns des autres puissent posséder des contacts, tout en rendant les dit contact ajoutable à tous les groupes d'une entreprise.

Autrement dit plusieurs groupes d'une entreprise peuvent posséder des contacts similaires sans que ce ne soit forcément le cas et les contacts associés à cette entreprise ne peuvent cependant pas être ajoutés au groupe d'une autre boîte.

Cela a bien entendu demandé d'être implémenté au niveau de la logique métier de la manière la plus optimisée possible, mais aussi au niveau de la base de données.

Alors que je pensais avoir justement construit ma base de données de tel sorte à prendre en compte toutes les contraintes de ce type, j'ai fini par me rendre compte que le modèle actuel de ma base de données entraînait en contradiction ou tout du moins compliqué grandement et inutilement la circulation des données de la manière attendu et prévu et que donc la logique derrière ce modèle était quelque peu faussée. Le problème venait notamment de l'inversion au niveau du schéma des tables entité_référence et contact (qui était donc dans l'autre sens originalement) et que leur place actuelle ne permettait pas à mon modèle de produire les résultats escomptés. J'ai donc dû faire l'inversion entre les 2 et réadapter mes modèles (import, foreign key ...) et routes de tel sorte à correspondre à ce changement dans la structure de la base de données. Cela m'a fait perdre un temps non négligeable et je trouvais intéressant de le mentionner car cela m'a permis de redoubler de rigueur et d'attention afin d'éviter qu'une telle situation se reproduise à l'avenir.

Une fois donc ce problème réglé, on incorpore dans notre projet le système d'authentification forte MFA qui se fait via l'envoi d'un code à 6 chiffres par mail et qui sert à créer les tokens JWT (avec la librairie **jose** et **jwt.encode()** pour encoder le jeton avec la clé secrète et l'algorithme d'encodage) et qui servent lors de l'authentification de l'utilisateur, pour le reconnaître. Pour tester on fait appel à un service du nom de mailTRAP. Il s'agit d'un service capable d'émuler un serveur SMTP et donc d'émuler et de simuler l'envoi de mail. Il nous suffit pour cela de définir dans notre fichier «.env» les variables d'environnement qui servent à la connexion au serveur SMTP, puis de vérifier lors de l'exécution de nos routes si les mails sont bel et bien envoyés avant de passer à la suite :

The screenshot shows an email inbox with three messages from 'dodo@gmail.com'. The third message, titled 'Votre code de vérification MFA', is selected. The content of this message is displayed below the preview pane: 'Votre code de vérification est : 951575'.

Dans le même principe que le système d'envoi de mail, nous avons aussi le système de conversion des factures / devis en PDF afin de faciliter leur envoi, consultation et leur impression. Pour cela on commence en tout premier par la création de Template HTML qui serviront de structure pour l'agencement des informations et la forme de nos PDF. Puis on passe ensuite à la définition de nos routes qui vont gérer la transformation de nos factures sous format JSON en format PDF (On utilisera pour cela des librairies comme PDF kit, jinja2 ...)

On commence donc par aller chercher la facture et ses relations correspondantes à l'aide de l'ORM et de la fonction **selectinload()** qui permet de simplifier les démarches de jointure entre tables. Après quoi on chargera le Template concerné à l'aide de **FileSystemLoader()**

```
# Charger le template
env = Environment(loader=FileSystemLoader("app/templates")) #Environnement: objet Jinja2 pour gérer le rendu des
template = env.get_template("facture.html") #charge le fichier facture.html pour y injecter les données
```

Avant de passer les données au Template pour voir le rendu HTML

```
# Rendu HTML
html_content = template.render( #on passe tt les données au template pour avoir un rendu HTML complet
|   #données à charger dans le template ( enlevé ici pour des raisons de confidentialité )
)
```

Après quoi il ne nous reste plus qu'à générer le PDF puis de le retourner pour finir notre route.

```
# Générer PDF
pdf = pdfkit.from_string(html_content, False) #convertit le HTML en pdf
pdf_stream = io.BytesIO(pdf) #Crée un flux mémoire contenant le PDF

# Retourner le PDF
return StreamingResponse(
    pdf_stream,
    media_type="application/pdf",
    headers={"Content-Disposition": f"inline; filename=facture_{numero_facture}.pdf"}
    # indique au client d'afficher le PDF directement
)
```

J'ai développé un peu plus en profondeur, dans la partie précédente quelque fonctionnalité et procédé de mon projet afin de pouvoir donner à mon interlocuteur une idée un peu plus précise de la dimension technique de ce projet. Cependant à partir de maintenant, je le ferai de manière moins fréquente , étant donné que cette partie (le front) est moins exigeante d'un point de vue technique et n'a bénéficier que d'un temps limité pour être produite (environ une semaine et demi)
Passons donc sans plus attendre , à la partie visuel du site , soit le Frontend .

Pour le visuel de mon site , je suis parti sur un modèle classique , basé sur des templates HTML à l'aide de Jinja2, stylisé par une fiche de CSS . Pour pouvoir générer et gérer dynamiquement l'affichage des données dynamiquement , j'utilise des fichiers JavaScript qui permettent de faire le lien entre les vue et les route API , grâce aux ids et name HTML . Je suis parti sur une structure classique avec une page d'accueil qui donne la possibilité de se connecter ou de s'inscrire via des boutons . Comme précisé précédemment la connexion est géré par des tokens stocké dans des cookies qui constitue la session de l'utilisateur . Pour afficher des données de manière précise ou permettre l'interaction de l'utilisateur avec les API REST , j'ai utilisé des formulaires sous forme de modales , qui appelle les fonctions écrite en JavaScript .



Page d'accueil une fois connecté

```

> async function loadFactures() { // Fonction pour charger les factures et les afficher dans le tableau...
}

> function closeFactureModal() { // Fonction pour fermer la modale...
}
|
> async function openFactureModal(numero_facture) { // Fonction pour ouvrir la modale avec les détails d'une facture...
}

> function openCreateFactureModal() { ...
}
> function closeCreateFactureModal() { ...
}

> function closeEditFactureModal() { // Fonction pour fermer la modale de modification...
}

> function closeAddLigneFactureModal() { // Fonction pour fermer la modale d'ajout d'une ligne d'article...
}

```

Exemple de fonction javascript pour gérer les données et API lié au facture

Numéro	Date	Client	Montant HT	État	Actions
FAC001	2025-06-23	Madame mia mama	2000.00 €	En attente	Détail Supprimer Modifier PDF
FAC003	2025-06-25	Madame mia mama	2500.00 €	Payée	Détail Supprimer Modifier PDF
FAC005	2025-06-27	Madame mia mama	15000.00 €	En attente	Détail Supprimer Modifier PDF

Visuel pour les factures avec tout les options nécessaires

Une fois le principe compris (les routes API qui sont appelées dans les fonctions JavaScript qui elle même sont appelées dans les templates HTML) il s'applique à tout les fonctionnalité pour chaque page nécessaire du site (les factures , les règlements , les devis , les articles ...) quand à l'utilisation des modales , elles permettent à la fois de garder un visuel moderne tout en évitant de démultiplier le nombre de page nécessaires . Après quoi on ajoute les fonctionnalité annexe , comme le fait de pouvoir trier les données à l'aide de filtres , ou les choix pré-choisi pour certaines données (exemple : dans quel état est la facture ? Soit payée , soit en attente , soit annulée ... , on évite comme cela des données trop éparpillés ou non prévu à la base , exemple : donner comme état à la facture la valeur « cochon-dinde » cela n'aurait aucun sens).

Pour cette partie il fallait bien faire attention au règle de caste , notamment à respecter le nom des URL des routes appelés , des id et name des templates connectant le JavaScript et le HTML , ainsi qu'à la construction des fonctions afin que ces dernières est bien l'effet exactement escompté et ne permette pas de créer de potentielle exploitation de faiblesse avec une fonction trop généraliste .

Une fois toute ces règles et contrainte respecté , on peut considérer que l'application est fin prête à rentrer en phase de production et a être déployé . Cependant , bien qu'extrêmement avancé je n'ai pas pu , par manque de temps , atteindre cette phase et rendre un projet complet . Néanmoins je retire tout de même une expérience plutôt positive de ce projet car cela m'a permis de renforcer mes compétences de FrontEnd tout en expérimentant et consolidant mon apprentissage du JavaScript que j'avais entamé durant mon temps libre pendant le stage grâce au site [TheOdinProject](#) .

Je dirais donc pour conclure mes dires sur cette mission de stage que ce projet fut le plus intéressant de loin , tant il m'a permis de consolider , étendre et diversifier mes capacités informatiques aussi bien sur l'aspect BackEnd (base de données , route API) que le frontEnd (Javascript , modale HTML ...) .

CONCLUSION

Au cours de ce stage de 2 mois , j'ai eu l'occasion de découvrir beaucoup d'aspect du monde informatique avec lesquelles je n'étais pas ou peu familiers , notamment les missions en lien avec le réseau et l'architecture informatique . Ce stage a donc été pour moi de découvrir des taches et missions qui sortent du cadre initiale de ma formations , ce qui pour moi représente le plus grand accomplissement et intérêts de ce stage .

De plus j'ai également pu grandement approfondir mon niveau et mes compétence techniques sur l'aspect développement informatique , soit la spécialité de mon enseignement , et j'ai également pu découvrir comment se construis un projet professionnel , quel exigences à respecter et quelle erreur à surtout éviter .

Au final , ce stage m'a permis de grandir aussi bien sur un plan personnel que professionnel et à mis en avant mes capacité d'apprentissage , d'adaptation et d'autonomie ce qui en fait une expérience enrichissante et qui fut intéressante à vivre .