



RAPPORT DE STAGE

Développement Web Full-Stack et Évolution d'Applications Complexes

Présenté par :

Nicolas Lopez

Étudiant en BUT Informatique

G2S4

Enseignant-référent : Christophe JALOUX

Entreprise d'accueil :

Fazaé

Saint-Didier-au-Mont-d'Or

Tuteur d'entreprise : Romain



Fiche technique

Caractéristiques du stage	Informations issues du rapport
Nom de l'entreprise	Fazaé
Activité de l'entreprise	Solutions web sur-mesure, open-source/ERP et hébergement.
Intitulé du sujet ou des sujets	Développement Full-Stack (FazaCRM, Lecture Jeunesse, Digit-all).
Sujet(s) indépendant(s) ou s'inscrit dans un projet plus large ?	Projets en production (maintenance et évolutions).
Le travail répond à quels besoins ?	Correction d'anomalies et refonte technique/graphique.
Il doit atteindre quels objectifs ?	Qualité, stabilité et respect de l'architecture.
A qui le travail est-il destiné?	Utilisateurs internes et clients B2C/B2B.
Quel sera son devenir ?	Mise en production directe.
Quelles sont les contraintes de temps ?	Délais serrés (ex: 2 semaines pour Digit-all).

Y a-t-il des informaticiens dans l'entreprise ?	Oui.
Est-ce que vous travaillez seul ?	Non, travail collaboratif.
Avec un autre stagiaire, un alternant ?	Oui (Yves).
Sur le même projet ou sur des projets différents ?	Mêmes projets (soutenu sur Digit-all).
Avec une équipe de l'entreprise ?	Équipe technique, graphiste et communication.
Quel est le matériel utilisé ?	PC sous Linux (Kubuntu).
Quels sont les langages ?	PHP (Symfony), Python, JS, XML, SQL.
Avez-vous dû apprendre un nouveau langage, des frameworks, des outils?	Oui, QWeb, OWL (Odoo) et Docker.
Quelle méthode ?	Agilité, règle des 20 min, MVC.
Avez-vous eu des contraintes ou des facilités pour la rédaction de votre rapport de stage ?	Facilités : documentation continue et comptes rendus.

Remerciements

Avant d'entamer la présentation de mes missions techniques, je souhaite exprimer ma gratitude envers les personnes ayant rendu cette immersion en entreprise si enrichissante.

Je tiens tout d'abord à remercier chaleureusement **Monsieur Fabrice PICCINALI**, dirigeant de **Fazaé**, pour m'avoir accueilli au sein de son équipe. Sa confiance et ses orientations stratégiques m'ont permis d'appréhender sereinement des projets de production d'envergure.

Ma reconnaissance s'adresse tout particulièrement à **Romain**, mon tuteur de stage. Son accompagnement quotidien et sa pédagogie ont été des piliers majeurs pour perfectionner mes compétences en développement et mon autonomie.

Je remercie également l'ensemble de l'équipe : **Basile, Loïc, Oriane, Thomas, Isabelle** et **Yves**. Leur accueil et la cohésion régnant au sein de l'agence, notamment lors des rétrospectives hebdomadaires, ont créé un cadre de travail idéal et très stimulant.

Enfin, je remercie **Madame Jocelyne Deboute** pour m'avoir orienté vers cette opportunité, **Christophe JALOUX** pour m'avoir accompagné, ainsi que le corps enseignant de l'IUT Lyon 1. Les bases théoriques et la rigueur acquises durant ces deux années de BUT ont été essentielles pour mener à bien mes travaux.

SOMMAIRE

Fiche technique	1
Remerciements	3
SOMMAIRE	4
Introduction	5
Présentation de l'entreprise	6
1. Description de Fazaé	6
a. Historique	6
b. Activités	7
c. Clientèle et Références	7
2. Environnement de travail	7
a. Philosophie	7
b. Outils	8
Les missions et réalisations durant le stage	9
1. Le projet FazaCRM	9
a. Problème d'erreur sur le suivi	9
b. Champs dynamiques sur les mails	10
c. Autocomplete Symfony UX	11
2. Le projet Lecture Jeunesse	12
a. Configuration	13
b. Ajout de menu dans le panel administrateur	13
c. Duplication de formation	14
3. Le projet Digit-all	15
a. Étude de faisabilité pour odoo19	16
b. Digit-all et Docker	17
c. La refonte de Digit-all	18
Bilan des compétences acquises et mises en œuvre	28
1. Réaliser des applications informatiques (Développement Full-Stack)	28
2. Optimiser des applications informatiques (Qualité de code et Résolution de problèmes)	29
3. Administrer des systèmes informatiques (Conteneurisation et Environnements)	29
4. Gérer des données et des bases de données	30
5. Travailler en équipe et conduire un projet (Savoir-être et Agilité)	30
6. Auto-évaluation globale et gestion des difficultés	31
Conclusion	31
ANNEXE	32
Document 1 :	32

Introduction

Ce stage de BUT informatique se situe à la fin de notre deuxième année de BUT. Il vient terminer et appliquer tout ce que l'on a vu précédemment durant les cours. Durant ces 2 années, de nombreux concepts ont été abordés, bien sûr du développement informatique, comme du C, JAVA, HTML/CSS et beaucoup d'autres langages. Cela nous a permis de comprendre la logique du développement et d'apprendre plus vite et de meilleure façon les prochains langages que nous découvrirons. Nous avons aussi appris la qualité de code, car oui un code qui marche n'est pas forcément un bon code, il faut le structurer, qu'il soit lisible et compréhensible, car ce code-là sera manipulé par nous mais aussi par d'autres personnes. Nous avons aussi vu du management ce qui nous a permis de comprendre les différentes façons de travailler en équipe, les méthodes de travail des informaticiens, des techniques de construction d'un projet, des techniques qui, sans ça, aucun projet informatique ne pourrait aboutir.

L'enjeu de ce stage était donc de sortir du cadre scolaire pour confronter toutes ces notions, comme la qualité de code, l'architecture ou encore la gestion de projet aux exigences d'un vrai environnement de production. Mon objectif était d'appliquer cette base technique à des écosystèmes web et backend professionnels complexes, ce qui m'a permis de faire l'expérience des contraintes réelles et quotidiennes d'une entreprise.

Mon objectif était de rejoindre une entreprise avec des missions orientées backend pour permettre d'approfondir mes connaissances. Je menais une recherche ciblée vers des agences de développement avant qu'une belle opportunité apparaisse. Durant le cours de communication, Madame Deboute a mis au tableau rapidement des entreprises avec lesquelles elle avait déjà travaillé les années précédentes, j'ai donc contacté Fazaé et deux jours après ils me contactaient pour un entretien. Fazaé m'a tout de suite plu. Déjà, l'adéquation technique était là : ils font du web/Symfony, ce que je cherchais, du backend. Ensuite, leur petite équipe soudée était idéale pour un bon apprentissage.

Fazaé m'a confié des missions différentes et variées mais globalement elles se tournent vers du full stack développeur. Je devais découvrir, installer et comprendre chaque différent projet que mon tuteur me demandait d'installer. Soit faire de la maintenance en réglant des bugs (des tickets) ou alors rajouter des fonctionnalités. J'ai également pu avoir la chance de travailler sur des pages du début à la fin.

Si je devais résumer cela en une phrase ce serait : Comment assurer le développement de nouvelles fonctionnalités sur des applications web déjà existantes, tout en garantissant une bonne qualité de code et la stabilité de l'ensemble ?

Présentation de l'entreprise

1. Description de Fazaé

a. Historique

Ancien ingénieur informatique pour de grands noms comme Euronews, Fabrice PICCINALI décide en 2003 de se mettre à son compte et commence à proposer son service de création de solutions web. Il décide en 2004 de créer sa société : Fazaé. Passant d'un statut d'auto-entrepreneur au statut de Société à Responsabilité Limitée (SARL), il commence à embaucher pour pallier la forte demande. Sa volonté n'est pas d'avoir un effectif important : il ne veut pas se consacrer à des tâches administratives ou de gestion de ressources humaines. Il préfère se consacrer à la gestion de projet et au développement. C'est pourquoi l'effectif n'a jamais dépassé 5 salariés. Au fil des années, Fazaé a enrichi son réseau et est aujourd'hui reconnue. L'entreprise compte aujourd'hui une petite équipe très soudée (dont Romain, mon tuteur). L'entreprise a connu de très belles années, dépassant par le passé les 500 000 € de chiffre d'affaires, même si ce chiffre a récemment baissé avec l'arrivée de l'IA qui a fait perdre quelques petits projets WordPress. L'entreprise, qui louait des locaux à Villeurbanne, dispose aujourd'hui de ses propres locaux à Saint-Didier-au-Mont-d'Or.



b. Activités

Depuis ses débuts, la société s'est principalement focalisée sur ces activités de prestations de services web :

1. **Développement de solutions sur-mesure** : E-Commerce, Intranets, Gestion de contenu, et développement d'outils internes complets (comme le projet FazaCRM).
2. **Adaptations de solutions open-source et ERP** : Utilisation de WordPress pour la gestion de contenus en architecture hybride (couplé à du Symfony en backend), ou encore l'intégration et la migration de solutions ERP comme Odoo (projet Digit-all).
3. **Hébergement et Supervision** : L'entreprise propose des solutions d'hébergement à ses clients pour accompagner les prestations web, ce qui représente une part financière non négligeable du chiffre d'affaires.

c. Clientèle et Références

Fazaé travaille aussi bien avec des petites structures que des grandes. En plus de 20 ans, l'entreprise a pu acquérir un panel de références important et surtout varié. En voici un échantillon historique :

1. **Intranet sur mesure** : SIEA (Syndicat Intercommunal d'énergie et d'e-communication de l'Ain), Bouygues Telecom (gestion des formations), Culture et Diversité.
2. **Solution E-Commerce** : Rotorplace, autodistribution pour l'armée de terre.
3. **Gestion de contenu** : Lecture Jeunesse.

2. Environnement de travail

a. Philosophie

À Fazaé, tout le monde applique des règles de bon sens très efficaces :

- **La règle des 20 minutes** : Lorsqu'on bloque plus de 20 minutes sur un problème, on va en parler (de vive voix ou sur Mattermost). Généralement, lorsque l'on bloque, on s'entête et on perd du temps. En en parlant, on profite des conseils des autres, on se libère la tête et au final on y gagne du temps.

- **Refactoring** : Ce n'est pas à la fin du projet que l'on refactorise. On s'en occupe le plus rapidement possible. Il faut que tout soit clair et concis.
- **Abstraction** : Notamment sur le cas du framework Symfony, Fazaé crée des classes et des services qui vont pouvoir être utilisés dans d'autres projets. La logique est isolée (par exemple la gestion des slugs ou le publipostage) pour être réutilisable.
- **Méthodes agiles** : Quand bien même Fazaé est une petite entreprise, les méthodes agiles sont utilisées. Ainsi, les vendredis, à la fin d'une semaine de développement, nous faisons une rétrospection pour faire le point en équipe.

b. Outils

Pour développer les solutions web, nous utilisons des technologies modernes avec une forte volonté d'utiliser des logiciels open source pour ne pas être dépendants.

- **Langages et Frameworks** : Nous utilisons le langage PHP dans ses versions récentes (PHP 8.3). Afin d'assurer une qualité au projet sur-mesure, Fazaé utilise le Framework Symfony (version 7.4). Nous utilisons également Python et XML pour les modules liés à Odoo.
- **Base de données** : Nous utilisons MySQL, mais également PostgreSQL (notamment manipulé via des extensions VS Code).
- **Système d'exploitation** : Le système d'exploitation utilisé par l'équipe est Kubuntu. Ce choix n'est pas anodin : l'objectif est d'être au plus près des configurations de production, c'est-à-dire des serveurs Linux. Nous mettons également en place Apache en local pour refléter au mieux cette architecture de production.
- **IDE** : Afin de faciliter le développement, nous utilisons des éditeurs modernes comme VS Code, qui permettent d'ajouter facilement des extensions pour visualiser les bases de données ou gérer le code.
- **Gestion de version et Déploiement** : Les projets sont tous sous versioning via Git. Aujourd'hui, Fazaé pousse l'automatisation plus loin en utilisant Docker. La création de Dockerfiles et de fichiers `docker-compose.yml` permet de monter des environnements de développement complets en local avec une seule commande. Cela facilite aussi la préparation des pipelines CI/CD.

Fazaé comporte deux développeurs expérimentés qui s'occupent de projets clients, Romain et Loïc. Elle contient un développeur junior (Basile), une graphiste (Oriane) , deux stagiaires, moi et Yves (orienté DevOps), une ressource humaine (Isabelle) et un chargé de prospection (Thomas). Cette équipe technique communique très souvent ensemble pour se partager de nouvelles technologies ou astuces notamment grâce aux rétros, des réunions où l'on se partage des découvertes liées au développement; ou encore durant des projets communs ou même seulement quand quelqu'un a besoin d'aide. Nous priorisons la communication en réel mais nous pouvons aussi communiquer par Mattermost, un logiciel open source pour se partager nos écrans ou encore si une personne est en télétravail.

Le travail découvert par chacun est inscrit dans le "book" de Fazaé. Ce book est un site web et permet de rassembler les connaissances de l'entreprise et de faire gagner du temps à chacun. Plus simplement, le book comporte la documentation de chaque outil utilisé par Fazaé.

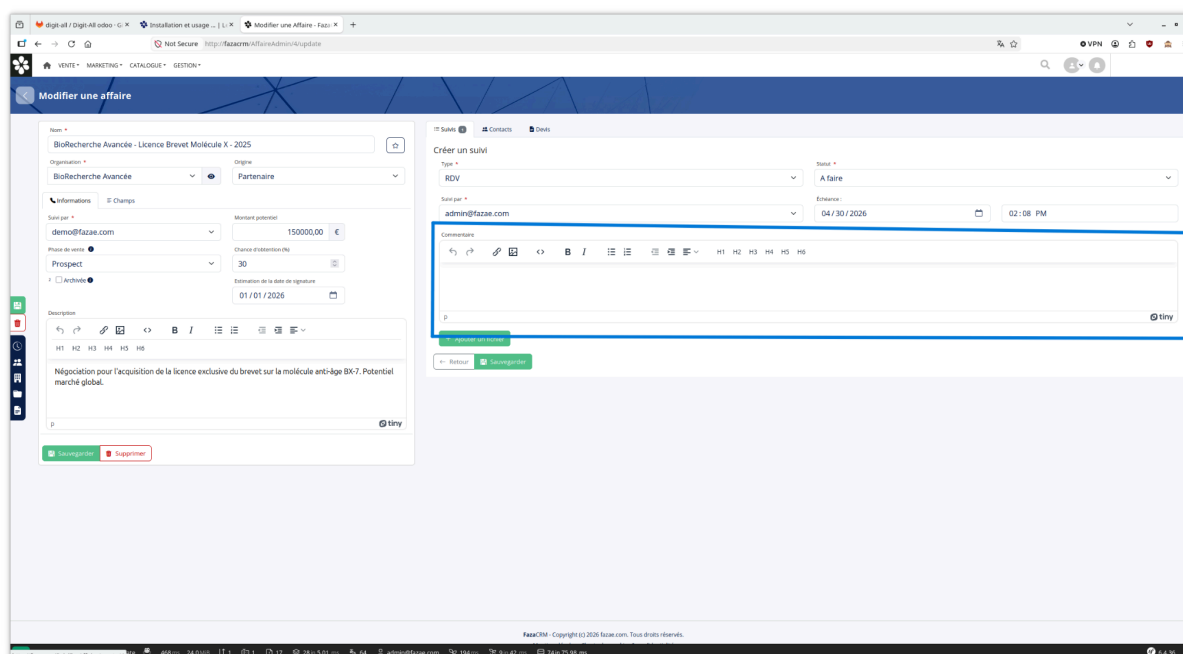
Les missions et réalisations durant le stage

1. Le projet FazaCRM

FazaCRM est le nom de mon premier projet découvert au cours de mon stage chez Fazaé. Ce projet leur a permis de doser mon niveau pour me donner des choses adaptées et me faire découvrir le plus de choses possible. Ce projet est interne à Fazaé, c'est un CRM. Un CRM (Customer Relationship Management) est un outil de gestion de la relation client qui permet de centraliser, traiter et analyser l'ensemble des interactions entre une entreprise et ses clients ou prospects afin d'optimiser la productivité et la réactivité commerciale. Il est développé avec Symfony qui est un framework (structure logicielle qui offre des outils pour simplifier et organiser le développement) de PHP. Ce CRM est utilisé tous les jours par Thomas alors il y a beaucoup de tickets (bugs ou évolution) qui y sont découverts.

a. Problème d'erreur sur le suivi

J'ai commencé par un bug assez simple d'apparence, il n'y avait pas de message d'erreur lorsqu'on ne mettait pas de commentaire dans les suivis, alors qu'il est obligatoire.

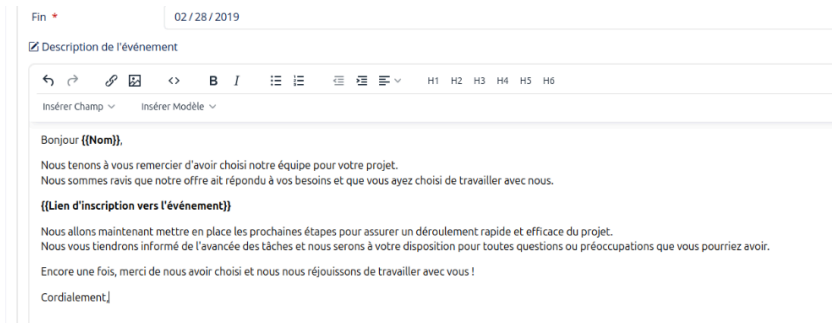


Pour moi, le plus dur a été de trouver les fichiers correspondants. Cela peut paraître simple, mais c'est normal, je n'ai jamais travaillé sur un projet si énorme. Beaucoup de fichiers sont liés, un fichier "vendor" avec des fichiers et des fonctions utilisés de partout chez Fazaé, des fichiers propres au projet, et tout ça dans des sous dossiers, des contrôleurs, des entités, des formulaires. Après 45 min à fouiller de partout, je suis allé voir mon tuteur et il m'a expliqué comment marchait la page, les vendors, et chaque dossier. Il ne m'a pas dit les réponses mais m'a aiguillé et donné des indices pour que je m'oriente bien mieux. J'ai trouvé le controller qui était étendu d'un fichier dans les vendors et avec ça j'ai pu trouver tous les fichiers qu'il me fallait ! Le problème était qu'ils avaient remplacé le textarea par un input personnalisé, mais le message d'erreur, lui, essayait de se mettre sur le textarea qui n'était pas présent. Après avoir réglé le problème, Fabrice a finalement dit qu'un commentaire ne doit pas être obligatoire, alors je le rends non obligatoire. Mais ça n'a pas été du temps perdu, j'ai appris à m'orienter dans un gros Symfony !

b. Champs dynamiques sur les mails

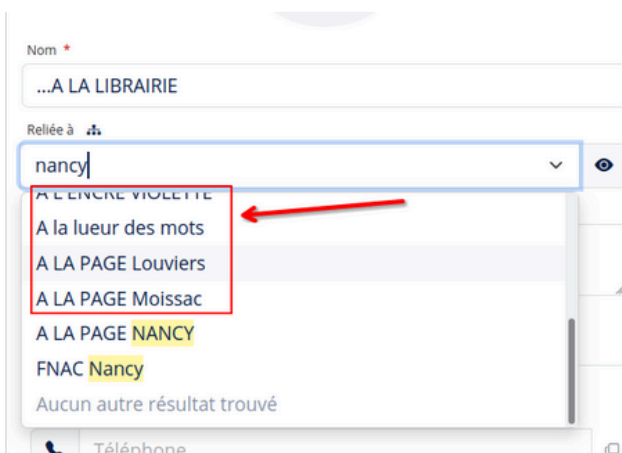
Ensuite un autre ticket m'a été assigné, FazaCRM permet d'envoyer des emails massifs à toutes les personnes inscrites à une "campagne". Ces mails n'étaient pas écrits pour chacun et contenaient donc des champs personnalisés comme `{{Prénom}}`, `{{formations-2024}}` (pour avoir des liens plus simplement), et plein d'autres. Ces

champs dynamiques ne se transformaient pas et déclenchaient une erreur 500 lors d'envois de mails massifs. La cause était qu'aucun code ne remplaçait ces champs, et ces caractères spéciaux étaient exécutés par le script. Pour régler le problème je cherche les champs `{{Prénom}}` dans les mails et je les remplace avec `str_replace` par `$user->getPrenom();`



c. Autocomplete Symfony UX

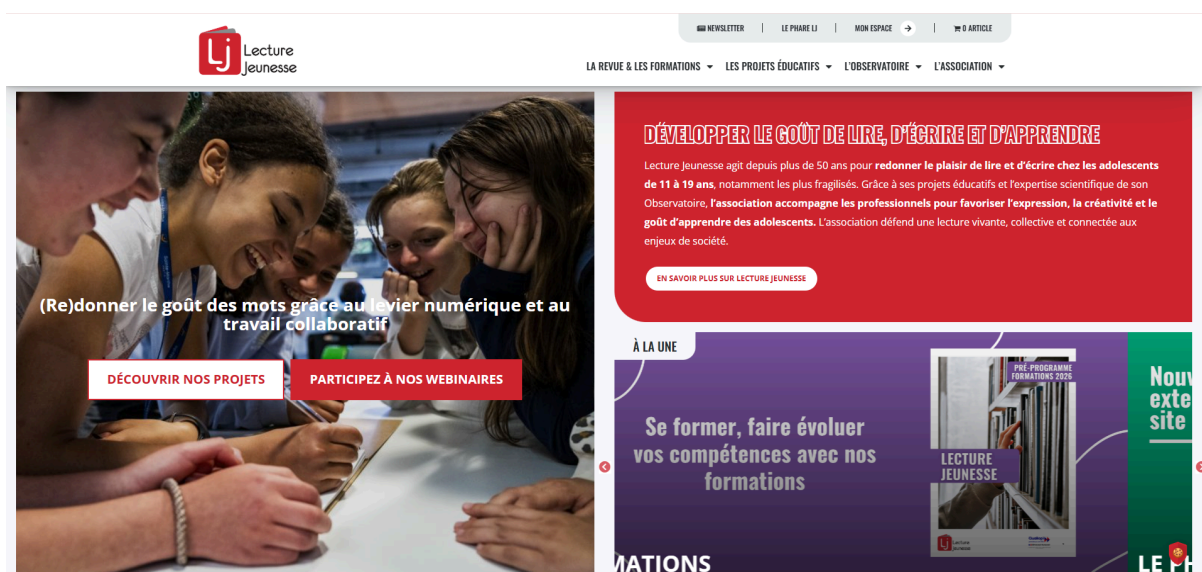
Au milieu de mon stage je suis revenu sur FazaCRM pour corriger de nombreux bugs. J'ai commencé par un problème, le select de l'autocomplete ne fonctionnait pas. Ce select était complètement géré par Symfony UX, on lui disait sur quelle entité chercher, puis il faisait automatiquement apparaître les 10 premiers puis quand on scrollait les 10 d'après. Cette technique est pratique et lisible, on l'appelle simplement et il fait le reste. En plus de ça tout était sécurisé avec les paginations de 10 par 10 qui fonctionnaient grâce à des appels AJAX (des appels au serveur directs en JavaScript). Le problème était que lorsque l'on recherchait dans le select les résultats apparaissaient en bas des 10 premiers résultats. Pour savoir si le problème venait de nous ou de Symfony j'ai créé un projet propre avec Symfony 6.4 et j'ai intégré au plus simple ce select SYMFONY UX. Le problème était toujours là, le problème venait donc de Symfony.



Le problème venait du fait que le paramètre de préchargement (preload) gardait en mémoire les anciennes requêtes AJAX, ce qui venait "polluer" l'affichage des nouveaux résultats de recherche. Pour contourner le problème il y avait deux solutions : soit désactiver le préchargement, soit se passer des appels AJAX et faire une seule requête avec query pour appeler tout d'un coup. Mais cette solution est lourde pour les entités qui contiennent beaucoup de lignes. À partir de 1000 ça devient trop lourd et illisible alors il faut appliquer la solution 1. J'ai réglé le problème de partout où je l'ai trouvé et documenté cela sur le book de Fazaé. Au début j'avais tout fait avec la solution 1 mais Thomas, celui qui s'occupe de la communication et est sur FazaCRM toute la journée, ne trouvait pas cette solution adaptée pour plusieurs champs avec de petits résultats alors j'ai corrigé cela et ai même groupé les selects pour faire apparaître les utilisateurs sous des titres. (par utilisateur actif/inactif par exemple).

2. Le projet Lecture Jeunesse

Lecture Jeunesse, ce projet est un site pour une entreprise qui développe le goût de lire, d'écrire et d'apprendre. Ils vendent des livres, des formations, des événements pour des écoles que l'on peut réserver sur le site via "numook". C'est un projet avec deux technologies, WordPress pour le front-end et Symfony pour le backend. WordPress n'est pas forcément apprécié par les développeurs mais il est très apprécié des clients qui préfèrent l'utiliser. Lecture Jeunesse est un site vitrine et un e-commerce.



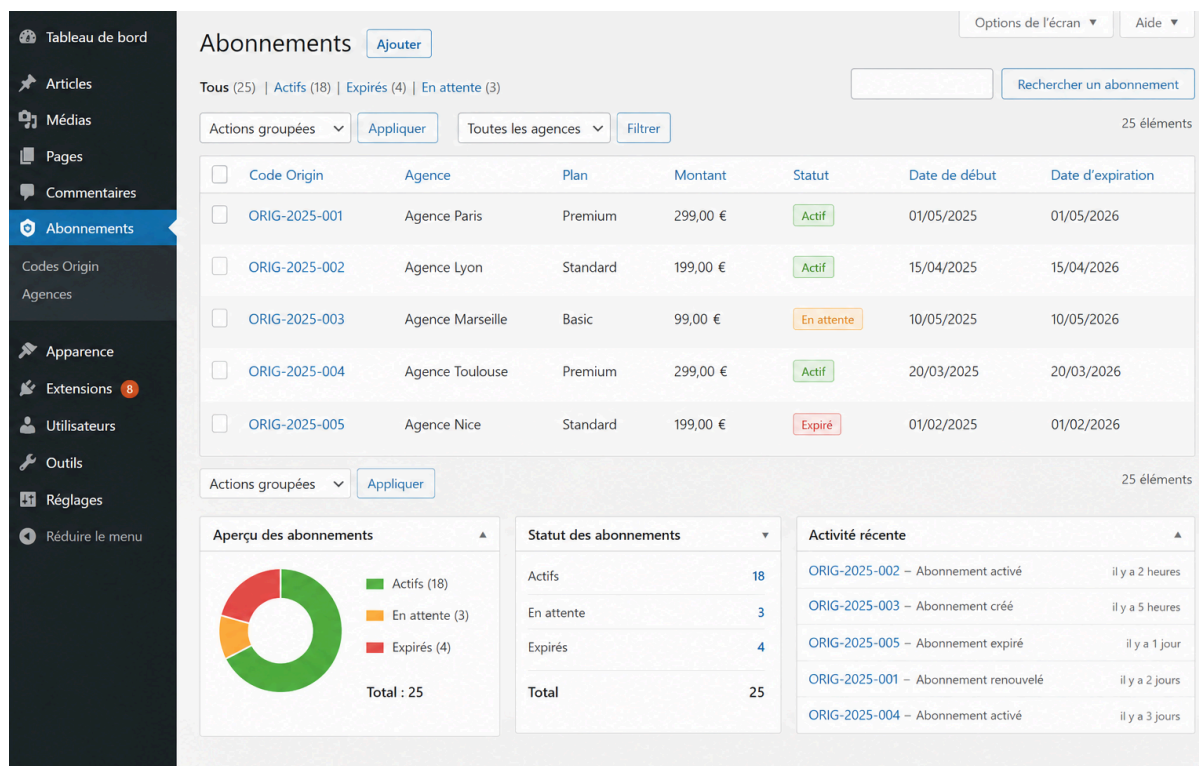
a. Configuration

La première difficulté a été d'installer et de configurer Lecture Jeunesse en local. Après un conflit de mon environnement local sur mon ordinateur, le logiciel Nginx a pris le contrôle et le logiciel Apache ne pouvait plus travailler. Après avoir coupé Nginx, tout ne marchait toujours pas. Et après avoir demandé à mon tuteur il m'a dit de me renseigner sur "varnish". Effectivement si je faisais `systemctl status varnish`, varnish était inactif, le problème était que le port était toujours utilisé par d'autres instances de Nginx et que Apache nécessitait varnish. Après avoir tout éteint, cela marchait ! J'ai fait une commande : `db_prod_to_local` pour récupérer la base de données de production pour avoir des exemples et tout était bon.

b. Ajout de menu dans le panel administrateur

Mon premier ticket a été une demande de Lecture Jeunesse qui souhaitait avoir le contrôle sur 2 informations dans le panel abonnement. Les codes origin et les agences. Ces informations sont très importantes et si elles venaient à être supprimées cela causerait des problèmes, mais ils le voulaient, alors il fallait leur donner l'accès via un menu déroulant. Actuellement ces informations étaient en brut dans le code, il fallait les passer dans la Base de données en taxonomy. Les taxonomies sont un outil puissant pour organiser et classer le contenu d'un site WordPress, cela est parfait pour la demande. En passant ces champs en taxonomy on les rend dynamiques et présents dans la Base de données en plus de pouvoir les intégrer dans des sous-menus plus facilement. Puisque ces données sont importantes j'ai mis en place un `<select>` pour qu'ils sélectionnent dans une liste déroulante et qu'ils ne fassent pas de faute de frappe. Dans WordPress nous ne pouvons pas directement interagir et modifier les menus déroulants du panel administrateur. Pour les intégrer j'ai utilisé les hooks. WordPress, juste avant la génération des sous-menus, demande au code s'il veut rajouter quelque chose, je lui dis justement avec mon hook de rajouter les sous-menus de mon choix. Cette architecture basée sur les Hooks impose une logique très différente de l'architecture MVC d'un framework comme Symfony, ce qui demande un temps d'adaptation certain pour un développeur backend.

Voici le résultat :



Un autre souci persistait, il fallait faire passer toutes les anciennes lignes des abonnements dans la nouvelle table taxonomy et non dans la table classique abonnement. Pour cela j'ai créé un script qui récupère chaque ligne dans abonnement et intègre cela dans la nouvelle table "taxonomy_abonnement". Ce script devra être lancé en prod ensuite puisque je n'ai pas accès à cette base de données-là. Il y avait juste à lancer `php bin/console migrate-taxonomies`, ce script était en PHP (dans le côté Symfony pour avoir plus de libertés et de simplicité).

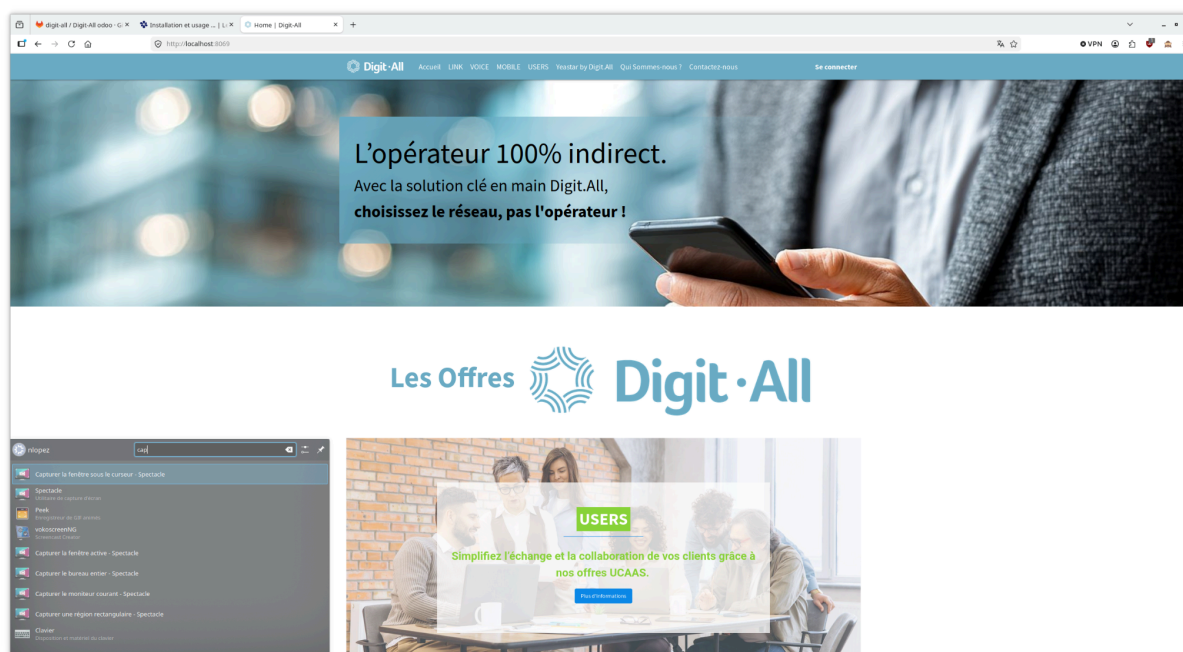
c. Duplication de formation

Je me suis ensuite occupé d'un autre ticket, qui était un bug. Nous pouvons dupliquer des formations en utilisant le même nom, mais on doit pouvoir utiliser le même nom puisque les formations sont renouvelées chaque année. Livre de la jungle 2025, Livre de la jungle 2026... J'ai donc essayé de moi-même dupliquer des formations et effectivement on pouvait, et après regard dans la base de donnée je me suis rendu compte que le slug, lui, était le même. Dans une formation, l'entité Formation possède un titre pour l'affichage (ex: Livre de la jungle), et un 'slug', qui est une chaîne de caractères simplifiée et unique servant d'identifiant dans l'URL "livre-de-la-jungle"

(exemple: lecturejeunesse.com/formation/livre-de-la-jungle). Le slug étant identique pour plusieurs formations cela les dupliquait. Pour régler le problème je suis allé dans le controller qui gérait les formations et à la création je vérifiais si le slug existait déjà, et si oui alors je rajoutais un -2, et s'il existait aussi alors un -3 ainsi de suite. Fabrice a regardé mon travail et m'a précisé que le controller servait seulement à contrôler et appeler, et non pas à agir. Il m'a alors expliqué que Symfony est bien en MVC mais qu'il intégrait quelque chose d'intéressant pour la propreté et la clarté du code : les services. Ils permettent de faire des actions spécifiques que le controller peut appeler, ils sont dans des fichiers différents. Ceci est aussi appelé le Principe de Responsabilité Unique. J'ai donc intégré toute cette logique dans un service, pour chercher les slugs j'ai utilisé Doctrine de Symfony qui permet de chercher dans une base de données facilement : `findOneBy(['slug' => $slug])` et le bug avait disparu !

3. Le projet Digit-all

Digit-all, après m'être occupé de Lecture Jeunesse, mon tuteur Romain m'a mis sur un autre projet : Digit-all. Digit-all est un revendeur de réseaux, de matériel informatique, d'outils réseaux ou de comptes pour des applications spécialisées dans les réseaux. Ils vendent cela pour des agents ou des marques blanches. Les agents achètent en gros et ensuite distribuent à d'autres entreprises, leurs clients. Une marque blanche, elle, ne gère pas des clients mais différents sites. La société Digit-all ne vend pas cela à des particuliers, ils travaillent en B2B (Business to Business, d'entreprise à entreprise). Ces agents et marques blanches ont donc un accès à un site Digit-all pour gérer leurs commandes, ils peuvent créer des tickets, avoir des liens, des trunks (outils réseaux) et plein d'autres choses. Leur site n'était pas très attirant et beaucoup de clients ne signaient pas à cause de de leur site. Cela ne faisait pas une très bonne impression. Ils ont donc contacté Fazaé pour une refonte de leur page accessible par les agents et les marques blanches.



Leur site a été entièrement développé sur Odoo 15 (2021) par d'autres développeurs. Odoo est un ERP assez puissant, un ERP est un type de logiciel que les entreprises utilisent pour gérer leurs activités quotidiennes telles que la comptabilité, les achats, la gestion de projets, la gestion des risques et la conformité, ainsi que les opérations de supply chain.

a. Étude de faisabilité pour odoo19

Ma première mission a consisté à réaliser une étude de faisabilité concernant la migration de l'ERP, de la version Odoo 15 vers Odoo 19. J'ai déjà essayé d'installer simplement le projet Odoo 15 sur un Odoo 19. Mais au moment de l'installation Oodo à généré des erreurs, le projet utilise plus de 80 modules communautaires pour aider au développement. Ces projets sont pour la plupart maintenus en Odoo 19, sauf pour "generic_ressource" qui s'occupe de toute la partie ticketing. Le problème étant que les développeurs sont ukrainiens et depuis le début de la guerre avec la Russie nous n'avons plus aucune nouvelle d'eux pourtant précédemment actifs. Sans une mise à jour de ce module spécifique, c'est l'intégralité du système de ticketing client qui devenait inutilisable, rendant la migration impossible en l'état. Cette idée a été mise de côté pour l'instant, le passage d'Odoo 15 à Odoo 19 n'est donc pas simple.

b. Digit-all et Docker

Il a fallu tout de même travailler sur le projet en Odoo 15, et le téléchargement et la configuration d'Odoo 15 est complexe et nécessite des versions précises de Python, PostgreSQL, et autres. Pour faciliter cela Romain m'a demandé de mettre en place un Docker pour tout gérer. Docker est un outil de conteneurisation qui permet de packager une application complexe avec toutes ses dépendances et services associés (comme une base de données PostgreSQL ou des modules spécifiques). Plutôt que d'installer manuellement les différentes technologies requises par un projet directement sur le système d'exploitation du développeur. Il permet de virtualiser uniquement l'environnement d'exécution de l'application, rendant ainsi le projet portable, léger à déployer, et permettant l'installation d'une architecture complète en une seule ligne de commande.

Le problème, je n'ai jamais poussé aussi loin la conteneurisation avec Docker alors il va falloir que j'apprenne. Je me lance avec une piste que mon tuteur m'a donnée et ma petite expérience avec les cours de Docker en BUT, il faut que je m'intéresse au [docker-compose.yml](#). Ce fichier définit ce qu'est notre image et télécharge des packs tout prêts. Digit-all nécessite un Odoo 15 alors nous faisons `image: odoo:15`. Avec ceci je peux construire mon Odoo muni de sa base de données PostgreSQL. Mais à la construction : `docker compose up -d`, des erreurs apparaissent.

Voici le fichier docker-compose pour Digit-all :

```
1 db:
2   image: odoo15
3   container_name: odoo_db
4   ports:
5     - "5432:5432"
6   environment:
7     - POSTGRES_DB=digitall
8     - POSTGRES_USER=digitall
9     - POSTGRES_PASSWORD=digitall
10  healthcheck:
11    test: ["CMD-SHELL", "pg_isready -U digitall -d digitall"]
12    interval: 10s
13    timeout: 5s
14    retries: 60
15    start_period: 30s
16  volumes:
17    - ./db_data:/var/lib/postgresql/data
18    - ./srv222.sql:/docker-entrypoint-initdb.d/01_backup.sql:ro
19
20
21 web:
22   image: $(IMAGE_WEB:-digit-odoo:latest) #digit-odoo:latest
23   container_name: odoo_web
24   depends_on:
25     - db
26
27
28 volumes:
29   # 1. Stockage des sessions (évite de se faire déconnecter au restart)
30   - odoo_sessions:/var/lib/odoo
31
32   # 2. Ton code source de travail (Repo Git)
33   - ./digitall-repo:/opt/digitall-repo
34
35   # 3. Tes modules externes (Alliance)
36   - ./Alliance-Com/dist:/mnt/extra-addons/alliance
37
38   # 5. Ton FILESTORE (très important pour les images/docs)
39   # On simplifie le chemin car tu n'as plus la structure de l'image officielle
40   - ./preprod:/var/lib/odoo/filestore/digitall/
41
42   environment:
43     - HOST=db
44     - USER=digitall
45     - PASSWORD=digitall
46     - ODOO_RC=/etc/odoo/odoo.conf
47   command: python3 /opt/odoo-source/odoo-bin -c /etc/odoo/odoo.conf -d digitall --dev=all
48
49 volumes:
50   odoo_sessions:
```

Au lancement, pas de soucis. Puis j'essaye de lancer mon site à l'adresse définie, soit localhost:8069. Et le site n'a aucun CSS, absolument aucun. Je cherche et pourtant j'ai bien téléchargé et lié le module avec Odoo, qui devrait définir le CSS. Le problème, le CSS sur le thème entier d'Odoo a été défini dans un autre dossier, et ce dossier était mal lié. Pour qu'Odoo dans Docker accède à n'importe quel fichier ou dossier sur notre

ordinateur (et non Docker), il faut créer des volumes. Les volumes sont des mécanismes de stockage persistant qui permettent de conserver les données en dehors des conteneurs et de les partager entre plusieurs conteneurs si nécessaire. Sans ça impossible de faire une liaison ordinateur -> conteneur Docker. Alors j'écris ce lien dans le docker-compose avec l'adresse physique du dossier de mon ordinateur et là où je veux le mettre dans Docker. Et voilà, au lancement de mon site les CSS sont apparents et le module marche !

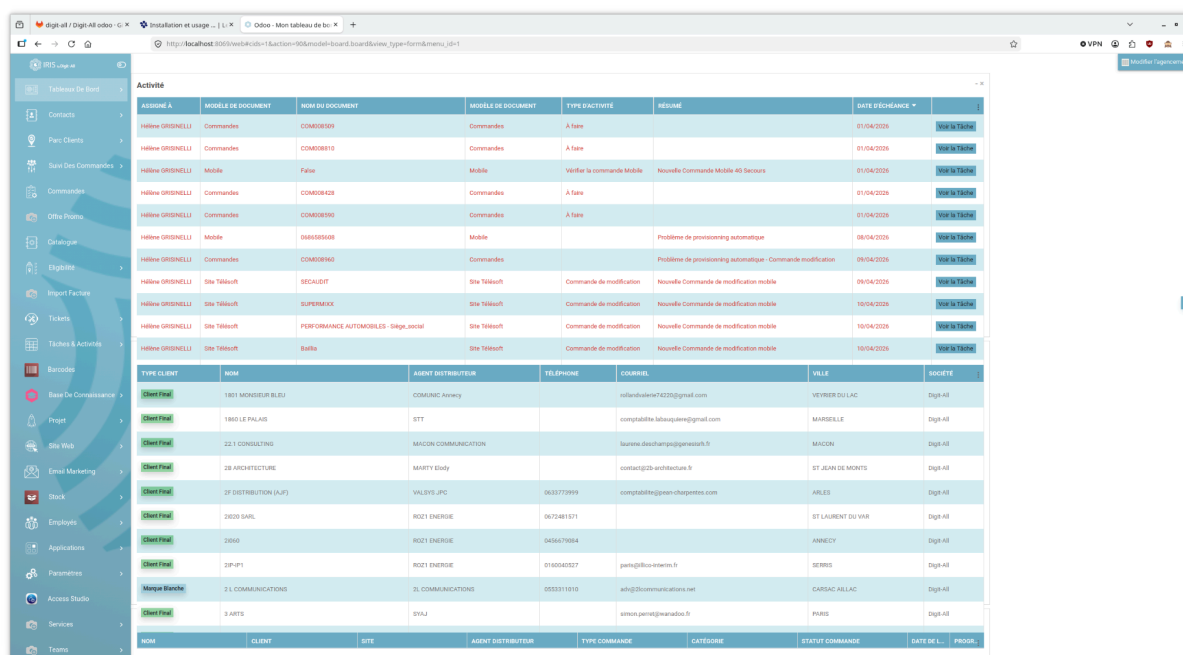


En cours de route nous apprenons avec mon tuteur qu'Yves, l'autre stagiaire, fait aussi le Docker de Digit-all. Alors pour ne pas avoir perdu trop de temps Romain essaye de pousser l'image pour voir si on peut tout intégrer dans le Docker. Après des jours de travail j'y arrive enfin, mais avec une image beaucoup trop complexe et inutile, on a gardé l'image d'Yves. Même s'il y a eu un chevauchement de tâches, cette expérience en a valu la peine car elle m'a beaucoup éduqué sur Docker.

c. La refonte de Digit-all

Toute la logique d'installer Docker pour Odoo avait pour but de travailler dessus, alors nous commençons. Nous sommes 4 sur le projet à développer, Romain, moi, Basile et Oriane. Nous avons 5 pages à faire : le tableau de bord, la page de recherche générale multicritère, une page simple agenda et actualités, une page pour gérer les bandeaux, une pour la page client et détails client. Pour chacune de ces pages et autres fonctionnalités, un ticket y est assigné.

Le site de digit-all avant la refonte graphique était :



The screenshot shows a web application interface for 'digit-all'. On the left is a sidebar menu with icons for 'Tableau De Bord', 'Contacts', 'Pacs Clients', 'Stat Des Commandes', 'Commandes', 'Offre Promo', 'Général', 'Etybille', 'Import Factory', 'Tickets', 'Tâches & Activités', 'Barcodes', 'Base De Connaissance', 'Projet', 'Site Web', 'Email Marketing', 'Stock', 'Employés', 'Applications', 'Paramètres', 'Accès Studio', 'Services', and 'Teams'. The main area displays a table titled 'Activité' with columns: ASSIGNE À, MODÈLE DE DOCUMENT, NOM DU DOCUMENT, MODÈLE DE DOCUMENT, TYPE D'ACTIVITÉ, RÉSUMÉ, and DATE D'ÉCHÉANCE. Below this is another table with columns: TYPE CLIENT, NOM, AGENT DISTRIBUTEUR, TÉLÉPHONE, COURRIEL, VILLE, and STATUT. The data rows show various clients and their associated orders or activities.

ASSIGNE À	MODÈLE DE DOCUMENT	NOM DU DOCUMENT	MODÈLE DE DOCUMENT	TYPE D'ACTIVITÉ	RÉSUMÉ	DATE D'ÉCHÉANCE
Hélène GRENELLE	Commande	COM00009	Commande	À faire		01/04/2026
Hélène GRENELLE	Commande	COM00010	Commande	À faire		01/04/2026
Hélène GRENELLE	Mobile	Falso	Mobile	Vérifier la commande Mobile	Nouvelle Commande Mobile 4G Secours	01/04/2026
Hélène GRENELLE	Commande	COM00028	Commande	À faire		01/04/2026
Hélène GRENELLE	Commande	COM00030	Commande	À faire		01/04/2026
Hélène GRENELLE	Mobile	086585500	Mobile		Problème de provisioning automatique	08/04/2026
Hélène GRENELLE	Commande	COM00030	Commande		Problème de provisioning automatique - Commande modification	09/04/2026
Hélène GRENELLE	Site Télécom	SECALOT	Site Télécom	Commande de modification	Nouvelle Commande de modification mobile	09/04/2026
Hélène GRENELLE	Site Télécom	SUPERMIX	Site Télécom	Commande de modification	Nouvelle Commande de modification mobile	10/04/2026
Hélène GRENELLE	Site Télécom	PERFORMANCE AUTOMOBILES - Siège social	Site Télécom	Commande de modification	Nouvelle Commande de modification mobile	10/04/2026
Hélène GRENELLE	Site Télécom	Belle	Site Télécom	Commande de modification	Nouvelle Commande de modification mobile	10/04/2026

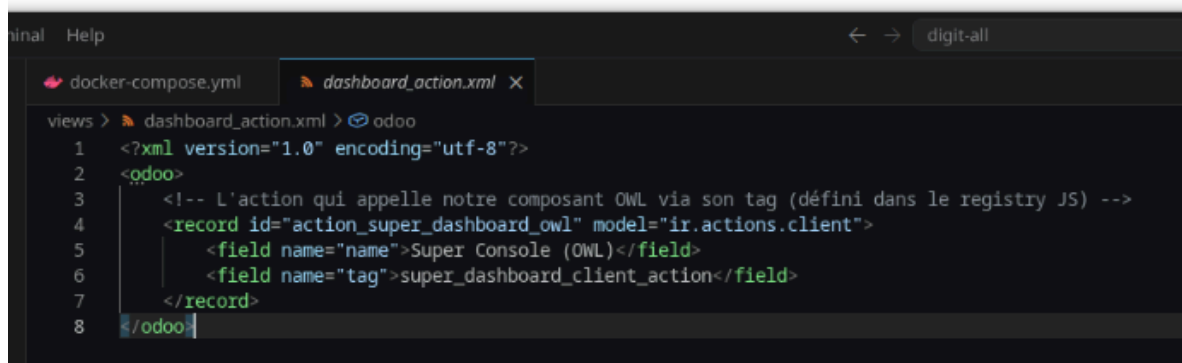
TYPE CLIENT	NOM	AGENT DISTRIBUTEUR	TÉLÉPHONE	COURRIEL	VILLE	STATUT
Client Final	1801 MONSIEUR BLEU	COMUNIC Anway		mlab@univie7420@gmail.com	VEVERRE DU LAC	Digit All
Client Final	1802 LE PALAIS	STT		comptable@lepalais.com	MARSEILLE	Digit All
Client Final	22.1 CONSULTING	MAISON COMMUNICATION		laurence.duchamp@maisonc.com	MAISON	Digit All
Client Final	28 ARCHITECTURE	MARTY Study		contact@gb-architecture.fr	ST JEAN DE MONTS	Digit All
Client Final	2F DISTRIBUTION (AUF)	VALPES_PJC	063779999	comptable@jean-chapettes.com	ARLES	Digit All
Client Final	2020 SARL	RO21 ENERGIE	0670481571		ST LAURENT DU VAR	Digit All
Client Final	2060	RO21 ENERGIE	045647084		ANNÉCY	Digit All
Client Final	2UP-IFI	RO21 ENERGIE	016040537	paris@ro21-energie.fr	SENNES	Digit All
Client Final	31 COMMUNICATIONS	31 COMMUNICATIONS	053311010	info@31communications.net	CARACAS LAC	Digit All
Client Final	3.4075	SNJ		simon.perrin@snj.fr	PARIS	Digit All

Avant de connaître le projet, Romain m'avait fait tester cette technologie que Fazaé ne connaissait pas encore : le Qweb. Pour me faire tester il m'a demandé de faire un test sur une recherche et sur un tableau de bord. Ce que j'avais fait n'était que des tests mais cela m'avait permis de faire de la documentation sur les 2 techniques possibles. Une première en détournant la balise Kanban pour écrire du Qweb tout en gardant les fonctionnalités internes d'Odoo (comme la recherche sur le modèle par exemple). Et la technique JavaScript OWL qui nous refait partir sur une feuille vierge. Cette expérience et ces documentations m'ont permis ensuite et durant le projet d'aider l'ensemble de l'équipe à la compréhension des différentes technologies et à une aide pour faire les différentes pages, même à des développeurs expérimentés comme Basile ou Fabrice.

En toute logique, puisque j'avais déjà commencé la recherche et le tableau de bord, ces 2 tickets me sont attribués. Je commence donc par le tableau de bord qui me paraît le plus simple pour commencer. La maquette du tableau de bord ressemble à cela : Elle est constituée de plusieurs parties pour permettre à l'utilisateur de voir et de manipuler les informations les plus importantes (comme les commandes gelées, les tickets urgents, les rendez-vous dans agenda etc., etc...).

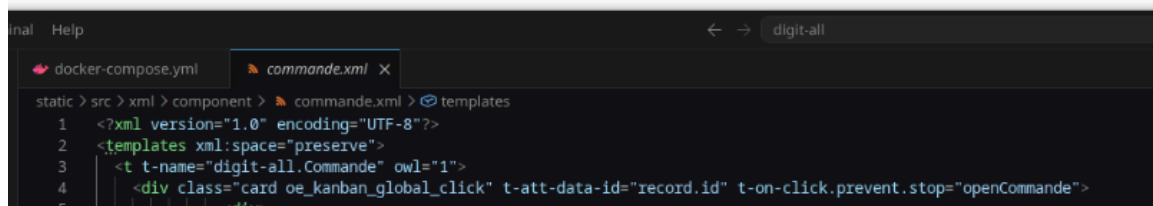
Je commence par les commandes par statut que j'avais déjà un peu testées auparavant à la découverte du Qweb. Cette partie a été un peu complexe car je faisais quelque chose que je ne maîtrisais pas encore complètement mais cela s'est passé plutôt

correctement. Je n'ai pas fait directement la partie CSS car Oriane devrait s'en occuper. Voici comment je structure une page avec Odoo, je fais un fichier `action_dashboard` dans le dossier views.



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <odoo>
3   <!-- L'action qui appelle notre composant OWL via son tag (défini dans le registry JS) -->
4   <record id="action_super_dashboard_owl" model="ir.actions.client">
5     <field name="name">Super Console (OWL)</field>
6     <field name="tag">super_dashboard_client_action</field>
7   </record>
8 </odoo>
```

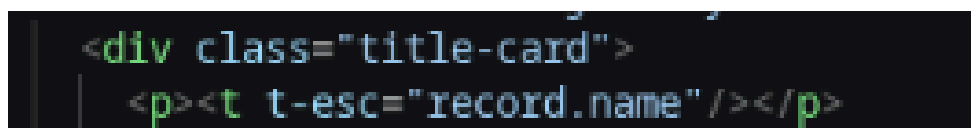
Ce fichier fait simplement l'action pour ouvrir la page, par exemple on peut appeler l'action dans le menu pour rajouter Tableau de bord au menu. Ensuite dans `static, xml`, j'ai un fichier `dashboard.xml`. Ensuite à l'intérieur je structure le début pour déclarer que c'est une page et qu'elle est en Qweb.



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <templates xml:space="preserve">
3   <t t-name="digit-all.Commande" owl="1">
4     <div class="card oe_kanban_global_click" t-att-data-id="record.id" t-on-click.prevent.stop="openCommande">
5       <div>
```

puis je peux coder en HTML classique à l'intérieur avec du Qweb. Le Qweb me permet d'utiliser des balises `<t>` comme `t-for-each` pour faire des boucles, `t-if` pour des conditions et plein d'autres.

Je peux accéder aux données dans le JavaScript avec `t-esc`.



```
<div class="title-card">
  <p><t t-esc="record.name" /></p>
```

Pour le JavaScript j'ai un fichier `dashboard.js` dans `static/javascript`. Ceci est un JavaScript avec des imports. Un `state` pour déclarer les variables accessibles dans la page XML avec le `t-esc`.

```
import { registry } from "@web/core/registry";
import { useService } from "@web/core/utils/hooks";

const { Component } = owl;
const { useState, onWillStart } = owl.hooks;

export class BandeauxCrud extends Component {
  setup() {
    this.rpc = useService("rpc");
    this.action = useService("action");
    this.notification = useService("notification");

    // On récupère l'ID envoyé par la page Liste (s'il existe)
    this.bandeauId = this.props.action.params?.bandeau_id || null;

    this.state = useState({
      name: "",
      description: "",
      type_message: "panne",
      mascotte: "reparateur"
    });
  }
}
```

Puis un `OnWillStart` pour déclencher des fonctions au lancement de la page puis des fonctions classiques.

```
JS bandeaux_crud.js X
static > src > js > JS bandeaux_crud.js > BandeauxCrud > goBack
9   export class BandeauxCrud extends Component {
10     setup() {
25       onWillStart(async () => {
27         const result = await this.rpc("/web/dataset/call_kw/bandeaux.bandeaux/read", {
31           kwargs: {}
32         });
33         if (result.length > 0) {
34           const data = result[0];
35           this.state.name = data.name;
36           this.state.description = data.description;
37           this.state.type_message = data.type_message;
38           this.state.mascotte = data.mascotte;
39         }
40       }
41     });
  }
```

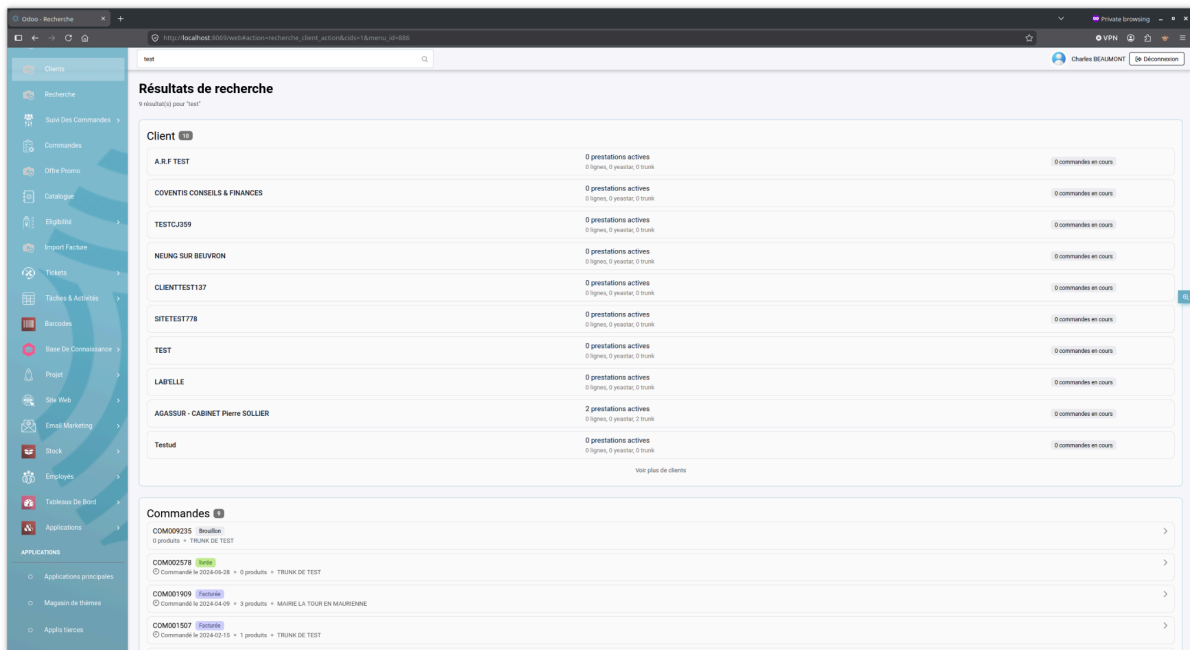
On accède à la base de données avec RPC.

```
if (this.bandeauId) {
  await this.rpc("/web/dataset/call_kw/bandeaux.bandeaux/write", {
    model: "bandeaux.bandeaux", method: "write",
    args: [[this.bandeauId], this.state], kwargs: {},
  });
  this.notification.add("Modifié avec succès !", { type: "success" });
} else {
  await this.rpc("/web/dataset/call_kw/bandeaux.bandeaux/create", {
    model: "bandeaux.bandeaux", method: "create",
    args: [[this.state]], kwargs: {},
  });
  this.notification.add("Créé avec succès !", { type: "success" });
}
```

RPC permet de faire des appels à la base de données comme si on appelait une API, avec une route et des données. On aurait pu le faire en ORM, sans ressemblance à une API et plus proche de la base de données, mais en faisant avec RPC cela pourrait nous permettre de copier-coller les RPC pour y intégrer sur un autre site avec des appels API simplement.

Par la suite, j'ai entrepris la conception de la page de recherche multicritère. Fidèle à la stratégie adoptée, j'ai sollicité le framework **OWL** pour développer un module de recherche universel, totalement affranchi de l'interface native d'Odoo et élaboré à partir d'une page vierge. Une fois l'arborescence des fichiers initialisée, j'ai mis en place l'extraction des entités (clients, commandes et tickets) via des appels **RPC**. Cependant, pour cibler avec précision les données pertinentes, une immersion dans la logique métier de Digit-all s'est avérée indispensable, notamment pour appréhender les notions de *trunks*, *Yeastar*, *voices* et *users*. Suivant la règle des 20 minutes, j'ai sollicité Romain pour obtenir des éclaircissements techniques. Il m'a précisé que les *trunks* représentent des segments réseaux tandis que *Yeastar* se rapporte à la gestion des comptes. Fort de ces orientations sur les champs à interroger, j'ai finalisé le développement fonctionnel de cette vue, tout en réservant l'intégration **CSS** pour une phase ultérieure.

Voici la page recherche multicritère final :



Il y a une partie que je n'ai pas encore abordée, la partie de droite du tableau de bord. J'avais des questions dessus, notamment je n'ai pas accès à l'API de Bouygues pour

tester l'éligibilité. Et je ne savais pas non plus à ce moment si agenda et actualité existaient ou s'il fallait que je les crée. Romain étant en vacances lorsque j'ai commencé alors j'ai fait la page recherche en attendant son retour. Maintenant qu'il est revenu je peux m'entretenir avec lui. Il me dit qu'il faut bien créer un agenda et actualité avec un CRUD, donc une page de création, modification, suppression simplement en Odoo classique, pas besoin de faire une belle page. Il me dit qu'il s'occupe de l'éligibilité et aussi de la recherche en header.

Pour faire ma page j'avais mis la recherche dans les menus mais au il fallait l'enlever et la mettre dans un nouveau header qui apparaîtrait sur toutes les pages. Alors je m'attaque à la création de pages classiques d'Odoo, les pages classiques se déclarent simplement dans une page XML dans views et dans le manifest. Le manifest est un fichier en python ou l'on déclare l'entièreté des fichiers de digit-all.

Il se construit comme ceci :

```
(
    'data': [
        'views/core/stock.xml',
        'views/dashboard.xml',
        'views/main.xml',
        'data/generic_resource_type.xml',
    ],
    'assets': {
        'web.assets_backend': [
            'digit-all/static/src/js/bandeaux_crud.js',
            'digit-all/static/src/js/bandeaux_list.js',
            'digit-all/static/src/css/my_style.css',
            'digit-all/static/src/css/trunk.css',
            'digit-all/static/src/css/yeastar.css',
            'digit-all/static/src/css/dashboard.css',
            'digit-all/static/src/css/bandeaux.css',
            'digit-all/static/src/css/client_detail.css',
            'digit-all/static/src/css/clients.css',
            'digit-all/static/src/css/mobile.css',
            'digit-all/static/src/css/general.css',
            'digit-all/static/src/css/recherche.css',
            'digit-all/static/src/css/dashboard_js.css',
            'digit-all/static/src/js/many2many_tags.js',
            'digit-all/static/src/js/custom_datepicker.js',
            'digit-all/static/src/js/eligibility_tab_links.js',
            'digit-all/static/src/js/dashboard_js.js',
            'digit-all/static/src/js/recherche.js',
            'digit-all/static/src/js/client_detail.js',
        ],
        'web.assets_qweb': [
            'digit-all/static/src/xml/component/ticket.xml',
            'digit-all/static/src/xml/component/yeastar.xml',
            'digit-all/static/src/xml/component/trunk.xml',
            'digit-all/static/src/xml/component/mobile.xml',
            'digit-all/static/src/xml/component/commande.xml',
            'digit-all/static/src/xml/component/commande_par_status.xml',
            'digit-all/static/src/xml/dashboard_js.xml',
            'digit-all/static/src/xml/recherche.xml',
            'digit-all/static/src/xml/bandeaux_crud.xml',
            'digit-all/static/src/xml/bandeaux_list.xml',
            'digit-all/static/src/xml/client_detail.xml',
        ],
    },
    'demo': [],
    'installable': True,
    'application': True,
    'auto_install': True,
)
```

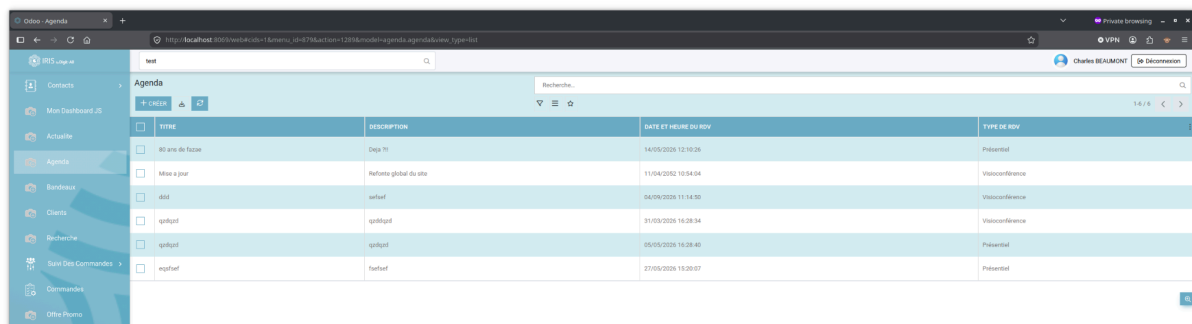
Pour les pages xml classique elle se construit simplement comme ceci :

```
agenda.xml X
1 <?xml version="1.0" encoding="UTF-8"?>
2 <odoo>
3
4   <record id="view_agenda_tree" model="ir.ui.view">
5     <field name="name">agenda.tree</field>
6     <field name="model">agenda.agenda</field>
7     <field name="arch" type="xml">
8       <tree>
9         <field name="name"/>
10        <field name="description"/>
11        <field name="date"/>
12        <field name="type_rdv"/>
13      </tree>
14    </field>
15  </record>
16
17  <record id="view_agenda_form" model="ir.ui.view">
18    <field name="name">agenda.form</field>
19    <field name="model">agenda.agenda</field>
20    <field name="arch" type="xml">
21      <form>
22        <sheet>
23          <div class="oe_title">
24            <label for="name" string="Titre de l'actualité"/>
25            <h1>
26              <field name="name" placeholder="Ex: Nouvelle mise à jour..."/>
27            </h1>
28          </div>
29          <group>
30            <field name="description" placeholder="Description de l'actualité..."/>
31            <field name="date" placeholder="Date et heure de l'actualité..."/>
32            <field name="type_rdv" placeholder="Type de l'actualité..."/>
33          </group>
34        </sheet>
35      </form>
36    </field>
37  </record>
38
39  <record id="action_agenda" model="ir.actions.act_window">
40    <field name="name">Agenda</field>
41    <field name="res_model">agenda.agenda</field>
42    <field name="view_mode">tree,form</field>
43    <field name="target">current</field>
44  </record>
45
```

Ce code permet de faire une page complexe en 50 lignes.

Simplement des appels au cœur de Odoo. Ces pages ont pu être faites très rapidement.

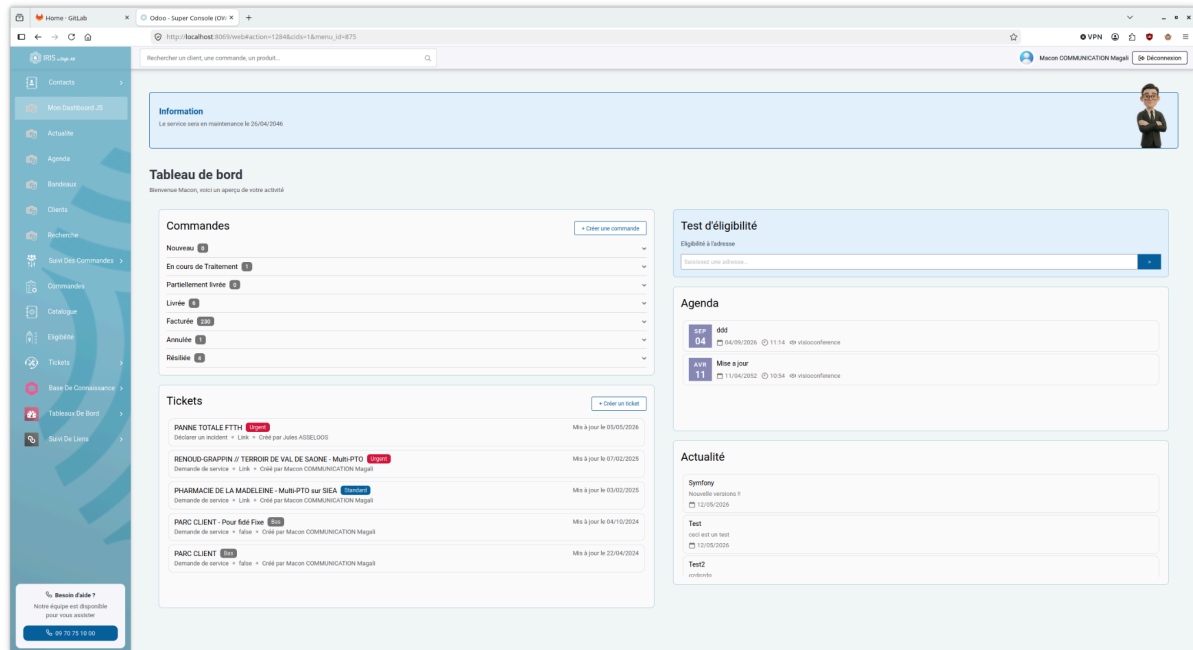
Voici le résultat :



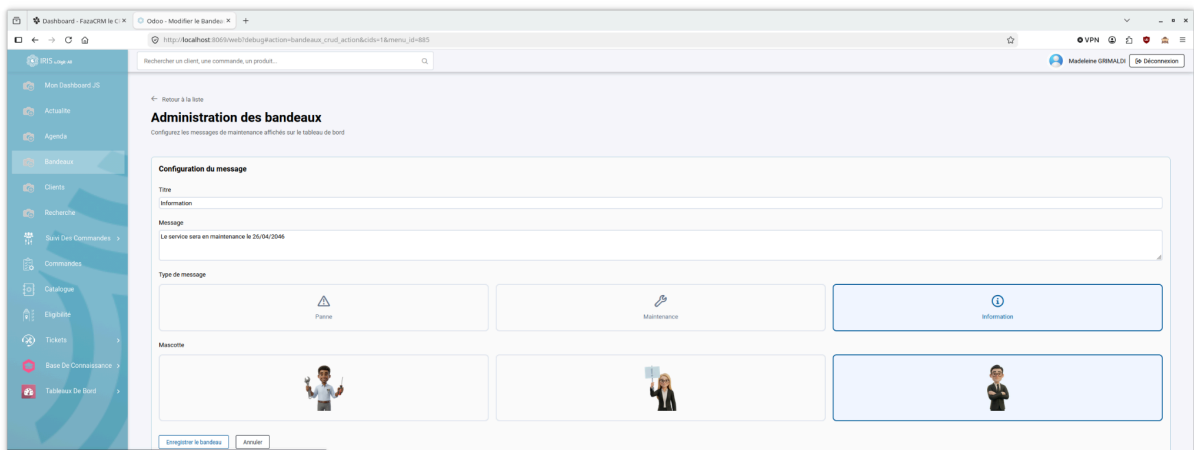
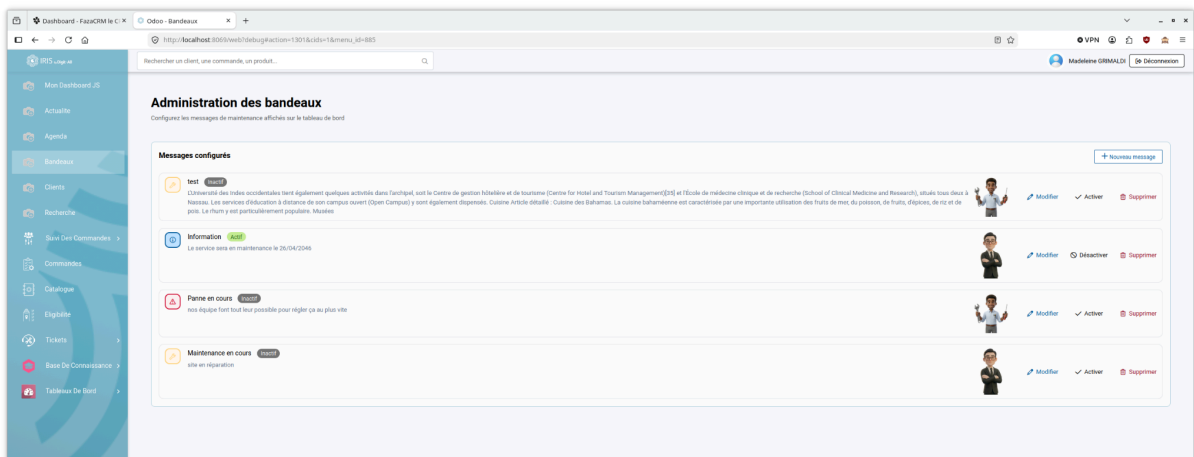
	Titre	Description	Date et heure du RDV	Type de RDV
☐	80 ans de l'ère	Déjà 71	14/05/2025 12:10:26	Présentiel
☐	Mise à jour	Refonte global du site	11/04/2025 10:34:04	Visioconférence
☐	dod	surfat	04/09/2025 11:14:50	Visioconférence
☐	spilipil	spilipil	31/03/2025 16:28:34	Visioconférence
☐	spilipil	spilipil	05/05/2025 16:28:40	Présentiel
☐	spilipil	surfat	27/09/2025 15:20:07	Présentiel

Pendant que je développais sur la page recherche, Basile et Oriane ont commencé sur le projet et Basile a commencé la page client. Il m'a notamment demandé de l'aide pour la compréhension du kanban car la page client est la seule page où l'on voulait garder la recherche client d'Odoo qui est assez complète. Je l'ai aidé mais je n'avais pas poussé le kanban sur des pages aussi complexes alors Basile avait du mal sur cette page-ci.

J'avais donc fini ma page de recherche et le tableau de bord, Oriane travaillait sur la page recherche et était loin d'avoir terminé. Je me suis alors occupé du style du tableau de bord, même si ce n'est pas ce que je préfère j'ai quand même bien aimé, cela a fait une pause dans mon développement intensif. Voici la page terminée, j'en suis fier :



J'ai ensuite pris le ticket des bandeaux car j'avais fini toutes mes tâches. Une question s'est alors posée, le faisons-nous en Kanban ou en JavaScript OWL ? En kanban cela est plus simple car il intègre déjà des fonctionnalités déjà codées de la part d'Odoo, malheureusement après réflexion la page bandeaux contient des images cliquables comme une liste mais horizontale, et trop de détails de style, ce serait au final beaucoup plus compliqué et moins lisible de le faire en Kanban. Je commence donc ce CRUD en JavaScript OWL, ceci est devenu maintenant beaucoup plus simple et agréable de coder sur Odoo avec OWL. J'intègre également le style en même temps car Oriane est encore sur la page recherche. Il faut faire le CRUD, donc logiquement il faut faire 3 pages, mais je n'en ferai que 2. La page de création sera aussi la page de modification, on pré remplira juste les champs quand on modifiera. Ces deux pages ont été faites plutôt rapidement, avec style, et de qualité, ce sont mes meilleures pages de ce projet, le résultat de l'apprentissage des autres pages et des précédents projets.



Il fallait rendre au client le site fonctionnel pour qu'il en fasse la présentation à son équipe. Mais la page client et le détail des clients n'étaient absolument pas finis, Romain me demande alors de faire de mon mieux pour essayer de rattraper ces pages. Je m'active et essaye de faire un rendu propre tout en étant rapide, j'arrive à faire la page client détails avec le style assez rapidement grâce aux commandes que j'ai mises en component ce qui me permet avec un `<t-call>` de les mettre dans n'importe quel fichier Qweb.

L'appel `<t-call>` :

```
<t t-foreach="state.tickets" t-as="cmd" t-key="cmd.id">
  <t t-call="digit-all.Ticket" />
</t>
</div>
```

Ceci m'a fait gagner du temps et cela rend le code plus propre. Je rends aussi la page client fonctionnelle, pas conforme à la maquette car je n'ai pas le temps mais fonctionnelle.

Ce projet est donc fini, entre la réunion et la fin j'ai pu peaufiner un peu la page client et régler un problème sur un retour client : nous n'avions pas bien géré la séparation entre marque blanche et agence, alors j'ai bien séparé pour afficher les clients aux agences et les sites aux marques blanches de partout où j'appelais simplement le client.

Client	Prestations	Commandes	Incidents / Alertes
1801 MONSIEUR BLEU	0 prestations actifs	3 en cours	-
1860 LE PALAIS	1 prestations actifs	1 en cours	-
22.1 CONSULTING	19 prestations actifs 0 table, 14 modules, 2 usine	14 en cours	-
28 ARCHITECTURE	0 prestations actifs	-	-
2F DISTRIBUTION (AJP)	0 prestations actifs	-	-
2020 SARL	0 prestations actifs	-	-
2042	0 prestations actifs	-	-
2060	0 prestations actifs	-	-
2070 SARL	0 prestations actifs	-	-
2IP-IP1	0 prestations actifs	-	-
2L COMMUNICATIONS	2 prestations actifs	-	-
3 ARTS	0 prestations actifs	-	-
3C SAS	0 prestations actifs	-	-
3 TECH	307 prestations actifs 4 matériels, 254 modules, 27 usine, 22 user, 24 parties	101 en cours	-
3S TELECOMS	0 prestations actifs	-	-
A2COM	0 prestations actifs	-	-
A2P ISOLATION	0 prestations actifs	-	-
A4 COPIE	0 prestations actifs	-	-
A5 TELECOM	0 prestations actifs	-	-
AARON AALBOOG	0 prestations actifs	-	-
AARPI BLACKBIRD	0 prestations actifs	-	-
ABAJUE CONSULTING	0 prestations actifs	-	-
Abatek Distribution	1 prestations actifs 1 usine	6 en cours	-

Client	Produit	Quantité
3 ITECH	3 ITECH	3 matériel(s)
3 ITECH	SCP PFR	1 matériel(s)

Commandes et Tickets
Commandes en attente
Tickets
Nouveaux
En cours de Traitement
Partiellement livrés
Livrée
Facturée
Annulée
Résiliée

Nous avons fait une réunion que j'étais chargé de planifier, et d'en faire le compte rendu, une réunion de fin de projet pour comprendre ce qui s'est bien passé, ce qui s'est moins bien passé et pourquoi, pour pouvoir régler ces problèmes la prochaine fois. La réunion se passe bien et nous arrivons à développer ce qui s'est mal passé pour pouvoir y régler. Vous retrouverez l'intégralité de ce compte rendu en Annexe 1 de ce document.

J'ai beaucoup aimé ce projet, j'ai pu coder des pages de zéro et je me suis senti vraiment utile. Je suis content que Romain et Fabrice m'aient fait confiance pour travailler sur Digit-all.

Bilan des compétences acquises et mises en œuvre

Cette expérience au sein de l'agence Fazaé a constitué un levier majeur pour articuler les acquis théoriques du cursus BUT Informatique avec les exigences de la production réelle. En m'appuyant sur le référentiel pédagogique, cette analyse détaille les savoir-faire critiques que j'ai pu consolider tout au long du stage.

1. Réaliser des applications informatiques (Développement Full-Stack)

Il s'agit de la compétence fondamentale de ma mission. J'ai agi sur l'intégralité de la chaîne technique, de la conception initiale de pages spécifiques à la maintenance corrective de noyaux applicatifs complexes.

- **Adaptation à des architectures complexes** : Loin du cadre scolaire, j'ai dû m'immerger dans des environnements d'envergure. Sur FazaCRM, j'ai appris à maîtriser l'arborescence Symfony (notamment les *vendors* et services) pour exploiter le modèle MVC à l'échelle industrielle. Pour Lecture Jeunesse, mon travail s'est concentré sur une logique WordPress articulée autour des *Hooks* et des taxonomies.
- **Maîtrise de nouveaux langages et frameworks** : J'ai élargi mon spectre technique en pratiquant le QWeb et le framework JavaScript OWL pour Digit-all. L'usage d'appels RPC pour interroger PostgreSQL a renforcé ma capacité à lier front-end et back-end de manière asynchrone.
- **Conception d'interfaces utilisateurs (UI/UX)** : J'ai traduit des maquettes graphiques en interfaces dynamiques. L'emploi de composants réutilisables via la balise `<t-call>` a permis d'assurer la modularité et la propreté du code front-end.

2. Optimiser des applications informatiques (Qualité de code et Résolution de problèmes)

Un enjeu majeur était de distinguer un code fonctionnel d'un code de production pérenne. La culture technique de Fazaé a été déterminante dans cet apprentissage.

- **Débogage analytique** : J'ai affiné ma méthodologie de résolution d'anomalies. Pour l'autocomplete de FazaCRM, j'ai isolé le problème sur un projet Symfony vierge pour démontrer que le défaut de cache (*preload*) incombait au composant Symfony UX, validant ainsi une approche rigoureuse du débogage.
- **Application des principes Clean Code** : Sensibilisé au principe de Responsabilité Unique, j'ai appris à décharger les contrôleurs au profit de services dédiés, notamment lors du développement du script de duplication pour Lecture Jeunesse exploitant Doctrine.
- **Refactoring continu** : J'ai adopté la pratique de l'optimisation immédiate du code pour garantir sa clarté et sa maintenabilité dès les premières phases de développement.

3. Administrer des systèmes informatiques (Conteneurisation et Environnements)

La gestion de l'environnement de travail sous Linux (Kubuntu) a été particulièrement instructive pour se rapprocher des standards de production serveurs.

- **Déploiement et conteneurisation** : J'ai approfondi l'usage de Docker en manipulant des fichiers `docker-compose.yaml` pour Digit-all. J'ai intégré l'importance des volumes pour assurer la persistance et la liaison des assets CSS d'Odoo entre l'hôte et le conteneur.
- **Gestion des serveurs web** : Les conflits de ports rencontrés lors de l'installation de Lecture Jeunesse m'ont contraint à manipuler les processus système via `systemctl`, renforçant ma compréhension des interactions entre Nginx, Apache et Varnish.

4. Gérer des données et des bases de données

Le traitement sécurisé de l'information a été au cœur des missions, soulignant l'importance de la rigueur en environnement de production.

- **Migration automatisée** : J'ai conçu des scripts PHP pour automatiser le transfert de données vers de nouvelles structures de taxonomies, tout en mesurant l'impact critique d'une erreur de manipulation sur la continuité du service.
- **Sécurisation applicative** : Le remplacement de saisies libres par des menus `<select>` dynamiques a permis de garantir l'intégrité des données en limitant les risques de fautes de frappe.

5. Travailler en équipe et conduire un projet (Savoir-être et Agilité)

L'intégration dans une équipe agile et à taille humaine m'a permis de participer activement au cycle de vie d'un projet informatique professionnel.

- **Démarche Agile et Rétrospectives** : Ma participation aux réunions hebdomadaires et la rédaction du compte rendu pour Digit-all m'ont appris à analyser collectivement les réussites et les points de vigilance pour optimiser les performances de l'équipe.
- **Communication fluide** : L'application de la règle des 20 minutes a favorisé l'entraide mutuelle. Mes travaux sur OWL m'ont permis d'accompagner des collaborateurs confirmés lors du déploiement de nouvelles fonctionnalités.
- **Documentation partagée** : En alimentant le "book" de Fazaé sur Docker ou Symfony UX, j'ai contribué à la pérennisation du savoir technique au sein de l'agence.

6. Auto-évaluation globale et gestion des difficultés

Le stage a comporté des défis techniques, de la complexité des architectures Symfony à la configuration exigeante des environnements conteneurisés. Ces obstacles ont été surmontés grâce à une résilience constante et une curiosité active.

Ma progression s'est révélée exponentielle, passant de la phase d'appropriation initiale à une capacité de développement en autonomie complète. Cette expérience confirme mon aptitude à m'adapter à des écosystèmes techniques variés par une méthode de travail structurée.

Conclusion

L'objectif principal de ce stage était de mettre en application tout ce que l'on avait appris en cours. Mais également de comprendre la différence entre l'université et le monde du travail. Ce stage m'a appris beaucoup, notamment de la rigueur de programmation pour la qualité de mon code qui est primordiale pour le bon fonctionnement d'un projet. J'ai aussi appris à naviguer et à déboguer des problèmes sur de grosses architectures, ce que je n'avais jamais fait auparavant. J'ai apprécié durant ce stage la petite équipe que compose Fazaé et sa façon de travailler. Mais durant ce stage je n'ai pas seulement développé des fonctionnalités ou corrigé des bugs. J'ai également appris à documenter mes découvertes, faire des comptes rendus, prendre l'importance de ne pas trop faire de réunions, surtout d'inutiles. Le projet Digit-all est celui qui m'a le plus marqué, j'ai pu faire des pages à partir seulement de maquettes et d'idées. Rendre quelque chose au client, qui plus est, a été très content du résultat. Ce stage a complètement confirmé ma voie professionnelle, faire du développement informatique est ma carrière professionnelle. Je vais entamer ma troisième année de BUT confiant et poursuivre pour pouvoir en fin de troisième année intégrer une équipe de développement informatique. Je suis à la recherche d'un métier pour lequel je suis content de me réveiller le matin. Cela a été le cas durant mon stage chez Fazaé. Je remercie toute l'équipe, Romain et Fabrice pour ce stage qui m'a énormément appris.

ANNEXE

Document 1 :

COMPTE RENDU DIGIT-ALL

Compte rendu de la rétrospection de digit-all le 27 mai 2026 à 11h.

Durée de la réunion : 1h

Membre présent : Romain LAGRANGE, Nicolas LOPEZ, Oriane TOURNEROCHÉ, Yves GILLE et Basile BRUSSEAU.

Information

Projet de refonte des pages accessibles par le client, repris d'un projet existant (nous ne sommes pas repartis de zéro).

Le développement a été réalisé sur Odoo 15, avec l'objectif futur de migrer vers Odoo 19. Les pages ont

été conçues en QWeb, une technologie qui s'intègre difficilement sur Odoo 15 mais qui sera parfaitement

adaptée pour Odoo 19.

Le projet a démarré le 4 mai 2026 avec une deadline fixée au 19 mai 2026, et a été livré dans les temps.

La mise en production avec Docker est planifiée pour le 4 juin 2026.

Le positif

- Le client est ravi de la démo, en particulier de la page de recherche (qui l'a laissé « bouche bée »).
- Le projet a été livré dans les temps, malgré un petit manque de finition sur le design et un ou deux retours client pour des modifications.

Le négatif

- Forte pression sur l'équipe, contrainte de s'approprier le projet et de tout réaliser en seulement deux semaines, sans aucune possibilité d'anticipation.
- Peu, presque aucun brief métier pour l'équipe, manque également de curiosité des développeurs qui n'ont posé aucune question.
- La mise en place technique a été compliquée, à cause de la configuration de Docker et la découverte de nouvelles technologies par les développeurs (un problème également lié au manque d'anticipation).
- Double travail inutile sur la configuration Docker pour Odoo, entraînant une perte de temps.

Décisions

- Organiser un brief métier collectif dès le lancement du projet pour que toute l'équipe partage la même vision. Il faudra oser poser davantage de questions dès qu'une zone d'ombre persiste sur le métier.
- Centraliser la communication directement sur le canal du projet plutôt que par messages privés. Une autre personne de l'équipe possède peut-être déjà la réponse.
- Penser plus à la relecture de code, notamment à travers les merge requests.

- Enrichir les maquettes avec des commentaires textuels au lieu de fournir de simples captures d'écran.
- Alimenter plus régulièrement le "book" de Fazae en documentation. Dès qu'une nouvelle difficulté est résolue, rédiger une note pour les suivants.
- Arriver à chaque début de réunion avec l'ordre du jour (les questions à aborder) et partager les documentations récemment rédigées pour que tout le monde soit au même niveau d'information.

Actions

- Pour orianne,basile,romain,nicolas, yves : Rédiger des books sur les savoirs faire acquis sur ce projet : qweb, utilisation de docker et mise en place de l'environnement de dev, utilisation des fichiers odoo, organisation du code, organisation des templates
 - faire relire ces books par fabrice
- Fabrice :
 - mettre à jour le book Conduite de projet avec les décisions prises
 - migrer le book « Conduite de réunion »