

RAPPORT DE STAGE

CRÉATION D'UNE APPLICATION WEB D'EMAILING

Année de formation 2022/2023

Réalisé par :

BENAYIBA Adam
CHERTI Omar

Encadré par :

RAHMOUNI Oussama
BAHJA Mohammed



SOMMAIRE

Dédicace	2
Remerciements	3
Introduction	5
I - Analyse.....	7
1 – Cahier de charge	7
2 - Méthode Scrum	9
3 – Pert.....	13
4 - DD	14
5 - MLD.....	17
6 - MPD	17
7 – Cas d'utilisation	18
8 – Diagramme de séquence.....	18
9 – Diagramme de Classe	19
II - Réalisation.....	20
1 - Technologies utilisés :	20
2 - Outils utilisés:.....	25
3 – Extrait du code utilisé.....	28
III – Manuel d'utilisateur	32
1 - La maquette initiale :	32
2 - Diagramme de navigation de l'application (jusqu'au cette date)	33
3 - Les écrans finaux :	33
IV - fiche technique pour l'administrateur	35
Conclusion	37
Tableau des abréviations :	39
Table des annexes :	40
Webographie	41

Dédicace

Notre reconnaissance sincère va à nos parents, proches et collègues d'études.

Nos parents, leur éducation, soutien et encouragements ont été inestimables. Leurs sacrifices et amour constants ont été des piliers essentiels dans notre parcours.

Nos proches ont été une source d'inspiration et réconfort.

Nos collègues d'études ont enrichi notre parcours par leur collaboration et partage de connaissances. Cette dédicace témoigne de notre gratitude profonde envers eux.

Nous sommes fiers d'avoir de telles personnes exceptionnelles dans notre vie. Merci pour leur amour, soutien et camaraderie.

Remerciements

Nous souhaitons exprimer notre profonde gratitude à toutes les personnes qui ont joué un rôle primordial dans notre expérience au sein de l'entreprise OPEN ITC.

À nos encadrants et responsables de stage, Jellal Abdelilah, Bahja Mohamed et M. Bendriss, nous tenons à exprimer notre sincère reconnaissance pour leur encadrement éclairé, leur soutien indéfectible et leurs conseils avisés. Leur expertise inestimable et leur disponibilité constante ont été des piliers essentiels pour notre progression et notre acquisition de compétences précieuses.

Nous souhaitons également exprimer notre profonde gratitude envers nos collègues de travail, pour leur accueil chaleureux, leur collaboration fructueuse et leur esprit d'équipe exemplaire. Leur générosité dans le partage de connaissances et d'expériences a considérablement enrichi notre parcours professionnel.

À nos estimés professeurs, nous tenons à exprimer notre reconnaissance la plus sincère. Vous avez joué un rôle fondamental dans notre formation et notre développement professionnel, et nous sommes profondément reconnaissants de votre enseignement rigoureux et de votre dévouement.

En particulier, nous souhaitons remercier chaleureusement Monsieur Rahmouni Oussama, notre Encadrant de stage, ainsi que Monsieur M. Lamnaour, pour leur passion inébranlable et leur expertise dans le domaine de l'informatique. Votre engagement envers l'excellence académique et votre volonté de partager vos connaissances ont été une source d'inspiration pour nous.

Votre patience, votre disponibilité et votre encouragement constant ont été des facteurs déterminants dans notre réussite. Grâce à vous, nous avons acquis des compétences précieuses et développé une vision plus large de notre domaine d'études.

Nous reconnaissons l'impact positif que vous avez eu sur notre parcours et nous vous sommes profondément reconnaissants de nous avoir poussés à nous dépasser, à explorer de nouvelles perspectives et à croire en nos capacités.

Votre dévouement envers notre éducation et notre épanouissement personnel restera à jamais gravé dans nos mémoires. Nous sommes honorés d'avoir été guidés par des mentors aussi exceptionnels que vous.

C'est avec une sincérité profonde que nous adressons nos remerciements les plus chaleureux à toutes les personnes qui ont contribué à faire de ce stage une expérience mémorable et enrichissante. Nous sommes reconnaissants de votre impact positif sur notre vie professionnelle et personnelle.

Ces remerciements solennels sont adressés avec une reconnaissance profonde et un profond sentiment d'honneur. Merci du fond du cœur pour votre engagement et votre dévouement envers notre succès.

Introduction

Le présent rapport de stage est le fruit de notre expérience au sein de la société OPEN ITC, une entreprise informatique basée à Tétouan.

Notre objectif principal lors de ce stage était de participer activement à la création d'une application web dédiée à l'email marketing. L'email marketing joue un rôle crucial dans les stratégies de communication des entreprises, leur permettant de promouvoir leurs produits ou services en envoyant des emails ciblés à leur audience. L'application web que nous avons contribué à développer chez OPEN ITC vise à faciliter et automatiser les processus liés aux campagnes d'email marketing pour son compte.

Dans ce rapport de stage, nous présenterons en détail les différentes tâches auxquelles nous avons participé, les compétences que nous avons acquises et les résultats que nous avons obtenus tout au long de ce projet. Nous partagerons également les défis auxquels nous avons été confrontés et les solutions que nous avons proposées pour les surmonter. Cette expérience de stage nous a permis d'approfondir nos connaissances en développement web et de mettre en pratique les concepts appris dans le cadre de nos études.

Tout au long de ce rapport, nous décrirons les différentes phases du projet, de l'analyse des besoins à la conception, en passant par le développement de l'application.

Nous mettrons en évidence notre collaboration avec l'équipe d'OPEN ITC, en soulignant l'importance du travail d'équipe, de la communication et de la coordination pour atteindre les objectifs fixés.

Ce rapport de stage est donc le reflet de notre implication dans le projet d'application web d'email marketing chez OPEN ITC, en mettant en avant les compétences que nous avons développées et les réalisations que nous avons accomplies tout au long de notre période de stage.

Société

OPEN ITC est une société informatique établie à Tétouan, se positionnant en tant qu'éditeur de logiciels de gestion d'entreprise et fournisseur de solutions web. Forte de son expérience, l'entreprise se concentre sur la création et l'intégration de logiciels de gestion sur mesure, offrant ainsi une gamme complète de solutions informatiques pour répondre aux besoins variés de ses clients.

L'expertise d'OPEN ITC s'étend à différents domaines, tels que le développement de logiciels et de sites web, l'administration des réseaux, la formation, l'assistance et le suivi des clients, ainsi que l'hébergement. En s'adaptant aux évolutions technologiques et aux demandes spécifiques de chaque entreprise, OPEN ITC se positionne en tant que partenaire stratégique pour les entreprises de toutes tailles, en particulier les PME-PMI.

La société se distingue par sa spécialisation dans le développement, la commercialisation, la mise en place et l'assistance technique de logiciels de gestion spécialisés. OPEN ITC propose des solutions de gestion adaptées aux besoins spécifiques de chaque client, couvrant divers secteurs d'activité tels que la comptabilité, la gestion des ressources humaines, la gestion commerciale, la gestion de production, la gestion de la relation client, et bien d'autres.

En travaillant en étroite collaboration avec ses clients, OPEN ITC comprend leurs objectifs, leurs processus métier et leurs contraintes, ce qui lui permet de concevoir et de développer des solutions logicielles sur mesure. L'entreprise met également l'accent sur l'accompagnement des PME-PMI tout au long de leur croissance, en leur fournissant une assistance technique continue et en évoluant leurs solutions pour répondre aux besoins changeants de leur entreprise.

Avec son approche personnalisée, son expertise technique et sa connaissance approfondie des métiers de ses clients, OPEN ITC s'impose comme un acteur de confiance dans le domaine de la gestion informatique.

I - Analyse

1 – Cahier de charge

Nous avons essayé d'élaborer le cahier des charges pour l'application.

L'objectif de ce dernier est de définir les spécifications et les exigences fonctionnelles pour le développement de cette application qui permettra à l'agence de gérer efficacement les campagnes d'email marketing, d'automatiser les processus de gestion des emails et des offres, ainsi que de suivre et d'analyser les performances des campagnes.

Objectifs de l'application

- a. Gestion des campagnes : L'application doit permettre la création, la planification et la gestion des campagnes d'email marketing. Elle devrait offrir une interface conviviale pour la conception des emails, le choix des segments de la liste de diffusion et la planification des envois.
- b. Automatisation des processus : L'application doit automatiser les processus récurrents tels que l'envoi automatique d'emails de bienvenue, de rappels ou de newsletters périodiques.
- c. Suivi des performances : L'application doit fournir des statistiques détaillées sur les taux d'ouverture, les clics, les conversions et autres indicateurs clés de performance des campagnes d'email marketing.
- d. Gestion des offres : L'application doit permettre la création et la gestion des offres promotionnelles. Elle devrait offrir des fonctionnalités pour la création de codes de réduction, la gestion des dates de validité et le suivi des utilisations des offres.

Fonctionnalités principales

- a. Création et édition d'emails : L'application doit permettre la création et l'édition d'emails attrayants et personnalisés à l'aide d'un éditeur intuitif. Il devrait offrir des fonctionnalités de glisser-déposer, de personnalisation dynamique du contenu et d'aperçu avant l'envoi.

b. Gestion des listes de diffusion : L'application doit permettre la gestion des listes de diffusion, y compris l'importation, l'exportation et la segmentation des contacts en fonction de critères spécifiques.

c. Planification et automatisation des envois : L'application doit permettre la planification des envois d'email à des dates et heures spécifiques, ainsi que l'automatisation des envois en fonction de scénarios prédéfinis.

d. Suivi et analyse des performances : L'application doit fournir des rapports détaillés sur les performances des campagnes, y compris les taux d'ouverture, les clics, les conversions, les désabonnements, etc. Elle devrait également offrir des fonctionnalités de suivi des liens et de tests A/B.

e. Gestion des offres promotionnelles : L'application doit permettre la création, la gestion et le suivi des offres promotionnelles, y compris la génération de codes de réduction uniques, la définition de conditions d'utilisation et la génération de rapports sur les utilisations.

Exigences techniques

a. Interface utilisateur conviviale et réactive, compatible avec les navigateurs web courants.

b. Architecture modulaire pour faciliter la maintenance et l'évolutivité de l'application.

c. Performances optimisées pour une gestion efficace des grandes bases de données et des envois massifs d'emails.

d. Gestion des erreurs et des logs pour faciliter le débogage et la surveillance de l'application.

Livrables attendus

a. Application web de gestion d'email marketing fonctionnelle et documentée.

b. Documentation technique détaillée, comprenant les instructions d'installation, la description de l'architecture, les manuels d'utilisation et de maintenance.

2 - Méthode Scrum

Au cours de notre stage, nous avons adopté la méthodologie Agile pour la gestion de projet, en nous concentrant plus précisément sur la méthode Scrum. Scrum est un cadre de travail itératif et collaboratif qui favorise la flexibilité et la livraison continue de résultats de haute qualité. En utilisant Scrum avec Jira, nous avons organisé notre travail en itérations appelées "sprints", généralement d'une durée de deux semaines. Au début de chaque sprint, nous avons tenu une réunion de planification où nous avons défini les objectifs et les fonctionnalités à livrer. Ensuite, nous avons utilisé Jira pour créer et gérer notre backlog de produits, qui contenait toutes les tâches et les exigences du projet.

Pendant le sprint, nous avons eu des réunions quotidiennes de coordination appelées "daily stand-ups", où chaque membre de l'équipe a partagé ses progrès, ses défis et ses plans pour la journée. Ces réunions nous ont permis de rester synchronisés et d'ajuster notre travail en fonction des obstacles rencontrés. Grâce à Jira, nous avons pu suivre l'avancement des tâches, mettre à jour leur statut et effectuer des ajustements en temps réel.

À la fin de chaque sprint, nous avons effectué une revue de sprint pour démontrer les fonctionnalités complétées à l'équipe et aux parties prenantes. Nous avons également organisé une rétrospective pour réfléchir à notre processus de travail, identifier les points forts et les axes d'amélioration, et apporter des ajustements pour les sprints suivants. L'utilisation de Jira dans le cadre de la méthodologie Scrum a permis une gestion efficace des tâches, une communication transparente et une meilleure collaboration au sein de l'équipe. Il nous a également offert une visibilité accrue sur l'avancement du projet, en facilitant la prise de décision et l'adaptation aux changements. En résumé, l'association de la méthodologie Scrum et de Jira a grandement contribué au succès de notre gestion de projet tout au long de mon stage.

Création des composants

Dans un premier temps, nous avons commencé par créer les composants nécessaires dans Jira pour structurer notre projet selon la méthode Scrum.

Projets / openitc

Composants

Créer un composant

Rechercher

Composant	Description	Responsable de composant	Responsable par défaut	Tickets	
Gestion verticaux	CRUD GESTION PARTIE VERTICALS		Projet par défaut	1 ticket	...
gestion Server Providers	CRUD GESTION PARTIE SERVER PROVIDERS		Projet par défaut	1 ticket	...
gestion MTA	CRUD GESTION PARTIE MTA		Projet par défaut	1 ticket	...
Gestion Headers	CRUD GESTION PARTIE HEADERS		Projet par défaut	1 ticket	...
Gestion des utilisateurs, rôles, permissions, équipes	CRUD GESTION PARTIE UTILISATEURS / rôles / permissions / équipes		Projet par défaut	1 ticket	...
Apprentissage bibliothèque Vue et son Framework Quasar	Auto Formation sur le langage de programmation		Projet par défaut	1 ticket	...

Figure 1

Définir les épics et les ajouter dans le backlog :

Nous avons soigneusement choisi les user stories en tenant compte de la capacité de l'équipe et de la complexité des tâches. Nous avons estimé l'effort requis pour chaque user story afin de mieux planifier notre travail.

Projets / openitc / Tableau OP

Backlog

Partager

Rechercher dans le backlog

Uniquement mes tickets

Récemment mis à jour

Analyses

0 tickets Estimation 0

Créer un sprint

Backlog 6 tickets

En tant que stagiaire, je veux apprendre à créer une application de gestion en utilisant Vue.js et Quasar.	APPRENTISSAGE V...	OP-10	15
En tant que stagiaire, je dois créer un système de gestion des utilisateurs , leurs rôles, permissions, et équipes.	CRÉATION D'UN SY...	OP-16	15
En tant que stagiaire, je dois créer un système de gestion des headers	CRÉATION UN SYS...	OP-18	12
En tant que stagiaire, je dois créer un système de gestion des server providers	CRÉATION D'UN SY...	OP-19	15
En tant que stagiaire, je dois créer un système de gestion des Mtas	CRÉATION UN SYS...	OP-20	17
En tant que stagiaire, je dois créer un système de gestion des verticaux	CRÉATION D'UN SY...	OP-21	15

Figure 2

Démarrage des Sprints :

Une fois les épics sont inclus dans le backlog

On a déterminé la date de début du sprint, en choisissant une période où toute l'équipe serait disponible. Ensuite, lors de la réunion de planification du sprint, nous avons discuté des objectifs à atteindre et sélectionné les user stories à inclure dans le sprint.

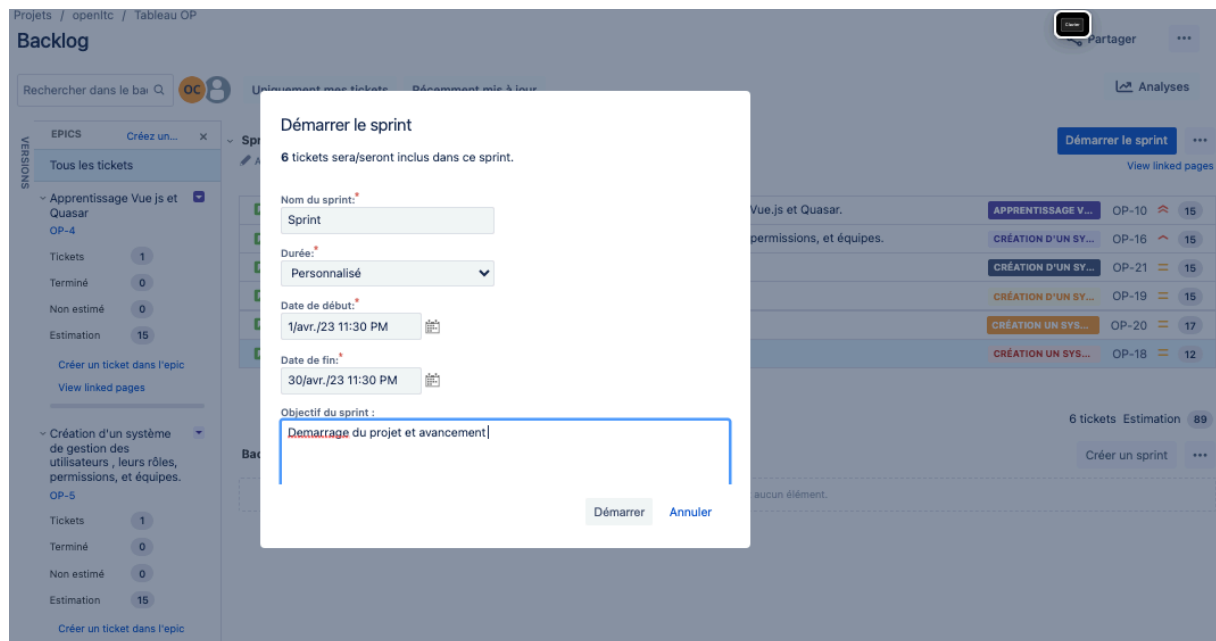


Figure 3

Pour organiser le sprint sur 4 semaines, nous avons opté pour une approche en deux itérations de 2 semaines chacune. Cela nous a permis de répartir les user stories de manière équilibrée et de mieux gérer notre charge de travail.

Suivi du Sprint:

Tout au long du sprint de 4 semaines, nous avons suivi notre avancement, ajustant si nécessaire pour maximiser la valeur livrée à la fin du sprint.

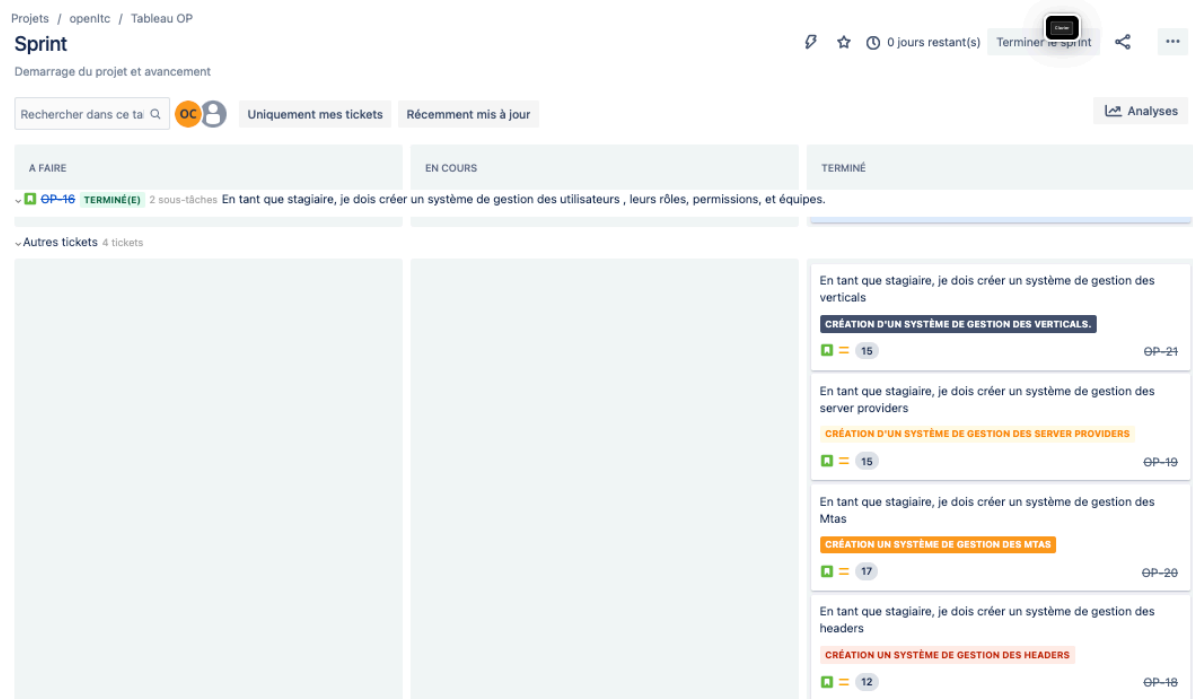


Figure 4

Graphique Burnup :

En utilisant le graphique de burnup pour évaluer l'évolution de l'effort accompli, on a pu obtenir une vision claire des résultats obtenus lors du sprint. Cela nous a permis d'identifier les points forts, de résoudre les problèmes potentiels et d'optimiser nos futurs sprints pour une meilleure performance.

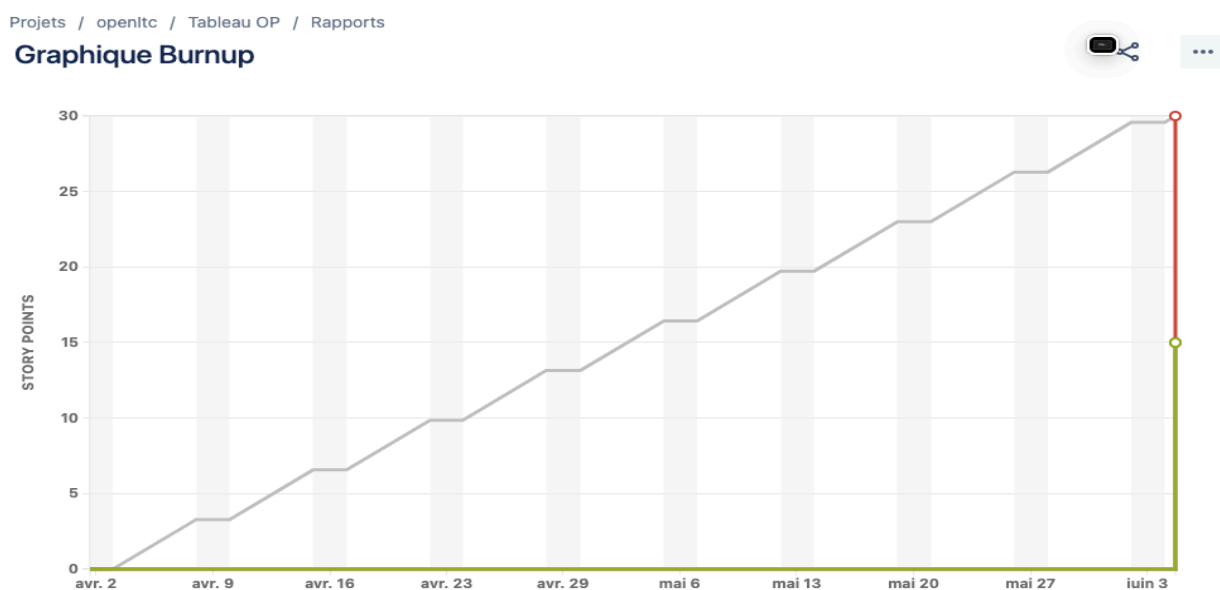


Figure 5

3 – Pert

	Taches	Durée
A	Apprentissage de Vue JS et Framework Quasar	15 jours
B	Gestion des utilisateurs	7 jours
C	Gestion des rôles	2 jours
D	Gestion des teams	2 jours
E	Gestion des permissions	2 jours
F	Gestion des headers	4 jours
G	Gestion des verticales	4 jours
H	Gestion des Server Providers	5 jours
I	Gestion des MTA Servers	7 jours

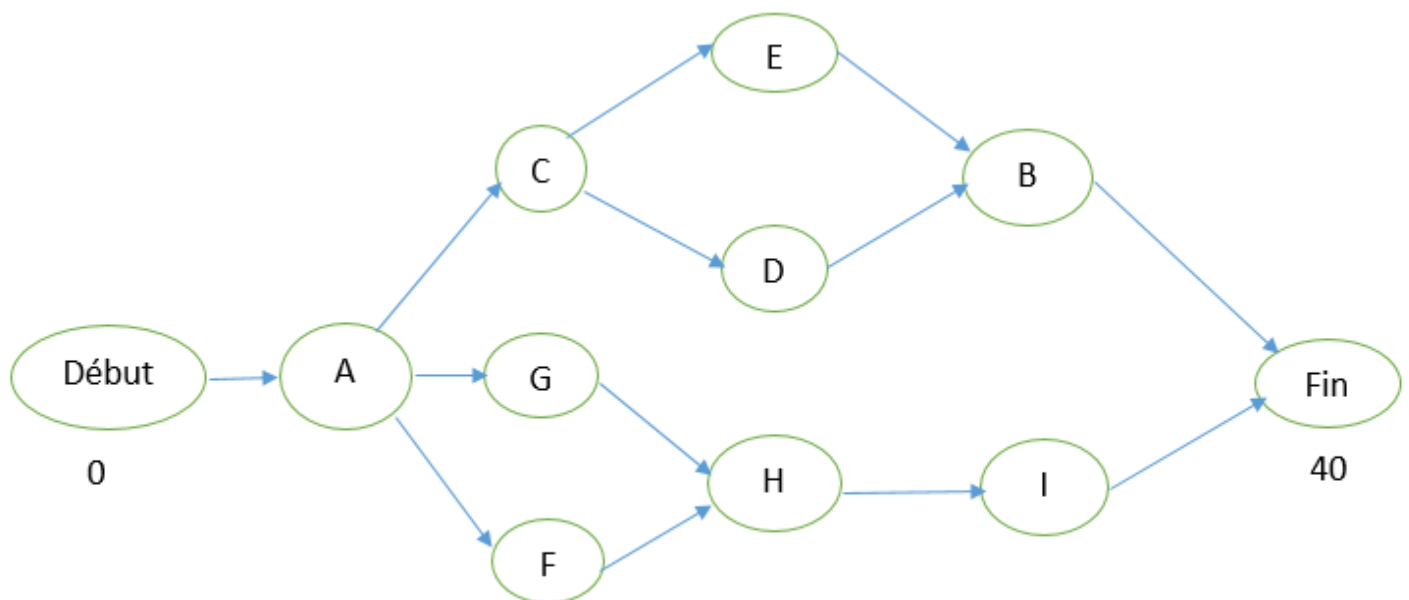


Figure 6

4 - DD

Table: headers

Champ	Type de données	Description
id	Entier	Identifiant unique du header
name	Texte	Nom du header
status	Texte	Statut du header
created_by	Texte	Créateur du header
created_at	Date	Date de création du header
updated_at	Date	Date de mise à jour du header
deleted_at	Date	Date de suppression du header (le cas échéant)

Table: model_has_permissions

Champ	Type de données	Description
permission_id	Entier	Identifiant de la permission
model_type	Texte	Type de modèle associé
model_id	Entier	Identifiant du modèle associé

Table: model_has_roles

Champ	Type de données	Description
role_id	Entier	Identifiant du rôle
model_type	Texte	Type de modèle associé
model_id	Entier	Identifiant du modèle associé

Table: mta_servers

Champ	Type de données	Description
id	Entier	Identifiant unique du serveur
name	Texte	Nom du serveur
status	Texte	Statut du serveur
expiration_date	Date	Date d'expiration du serveur
host_name	Texte	Nom d'hôte du serveur
main_ip	Texte	Adresse IP principale du serveur
ssh_port	Entier	Port SSH du serveur
username	Texte	Nom d'utilisateur pour se connecter au serveur via SSH
password	Texte	Mot de passe pour se connecter au serveur via SSH
server_provider	Texte	Fournisseur de services du serveur
created_at	Date	Date de création du serveur
updated_at	Date	Date de mise à jour du serveur
deleted_at	Date	Date de suppression du serveur (le cas échéant)

Table: permissions

Champ	Type de données	Description
id	Entier	Identifiant unique de la permission
name	Texte	Nom de la permission
guard_name	Texte	Nom du gardien associé à la permission
created_at	Date	Date de création de la permission
updated_at	Date	Date de mise à jour de la permission

Table: permission_cores

Champ	Type de données	Description
id	Entier	Identifiant unique de la permission_core
user_id	Entier	Identifiant de l'utilisateur associé
permission	Texte	Permission associée à l'utilisateur
created_at	Date	Date de création de la permission_core
updated_at	Date	Date de mise à jour de la permission_core

Table: roles

Champ	Type de données	Description
id	Entier	Identifiant unique du rôle
name	Texte	Nom du rôle
guard_name	Texte	Nom du gardien associé au rôle
created_at	Date	Date de création du rôle
updated_at	Date	Date de mise à jour du rôle

Table: role_has_permissions

Champ	Type de données	Description
permission_id	Entier	Identifiant de la permission
role_id	Entier	Identifiant du rôle

Table: server_providers

Champ	Type de données	Description
id	Entier	Identifiant unique du fournisseur de services
name	Texte	Nom du fournisseur de services
status	Texte	Statut du fournisseur de services
created_by	Texte	Créateur du fournisseur de services
created_at	Date	Date de création du fournisseur de services
updated_at	Date	Date de mise à jour du fournisseur de services
deleted_at	Date	Date de suppression du fournisseur de services

Table: users

Champ	Type de données	Description
id	Entier	Identifiant unique de l'utilisateur
name	Texte	Nom de l'utilisateur
email	Texte	Adresse e-mail de l'utilisateur
email_verified_at	Date	Date de vérification de l'e-mail de l'utilisateur
password	Texte	Mot de passe de l'utilisateur
remember_token	Texte	Token de rappel de l'utilisateur
created_at	Date	Date de création de l'utilisateur
updated_at	Date	Date de mise à jour de l'utilisateur
deleted_at	Date	Date de suppression de l'utilisateur (le cas échéant)

Table: verticals

Champ	Type de données	Description
id	Entier	Identifiant unique de la vertical
name	Texte	Nom de la vertical
status	Texte	Statut de la vertical
created_by	Texte	Créateur de la vertical
created_at	Date	Date de création de la vertical
updated_at	Date	Date de mise à jour de la vertical
deleted_at	Date	Date de suppression de la vertical (le cas échéant)

5 - MLD

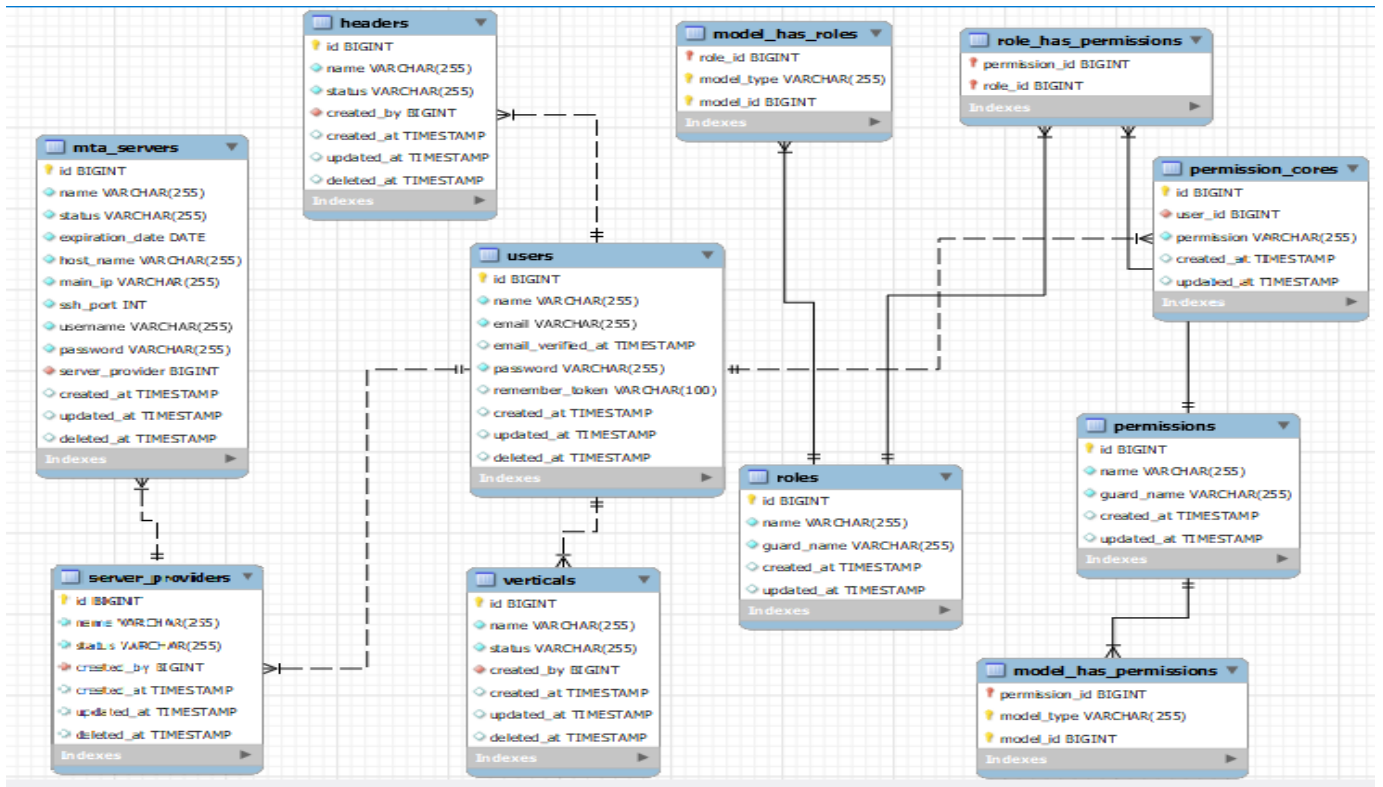


Figure 7

6 - MPD

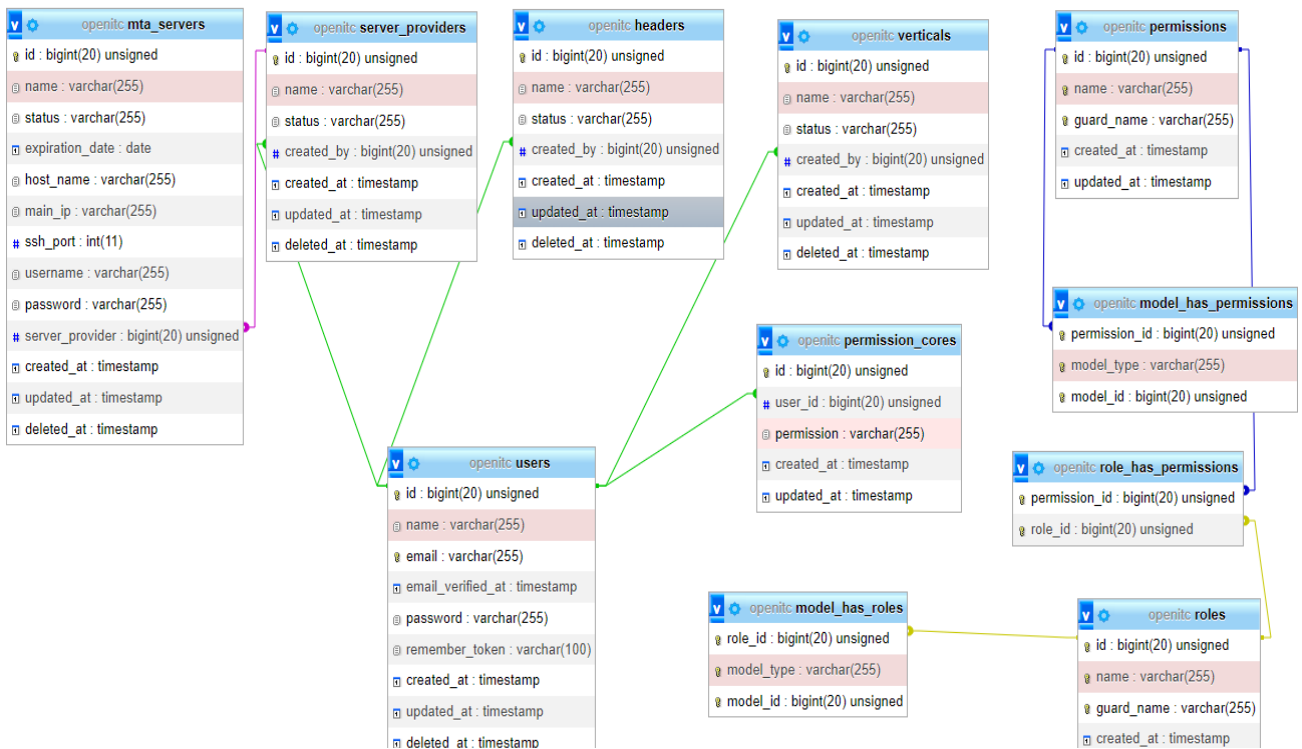


Figure 8

7 – Cas d'utilisation

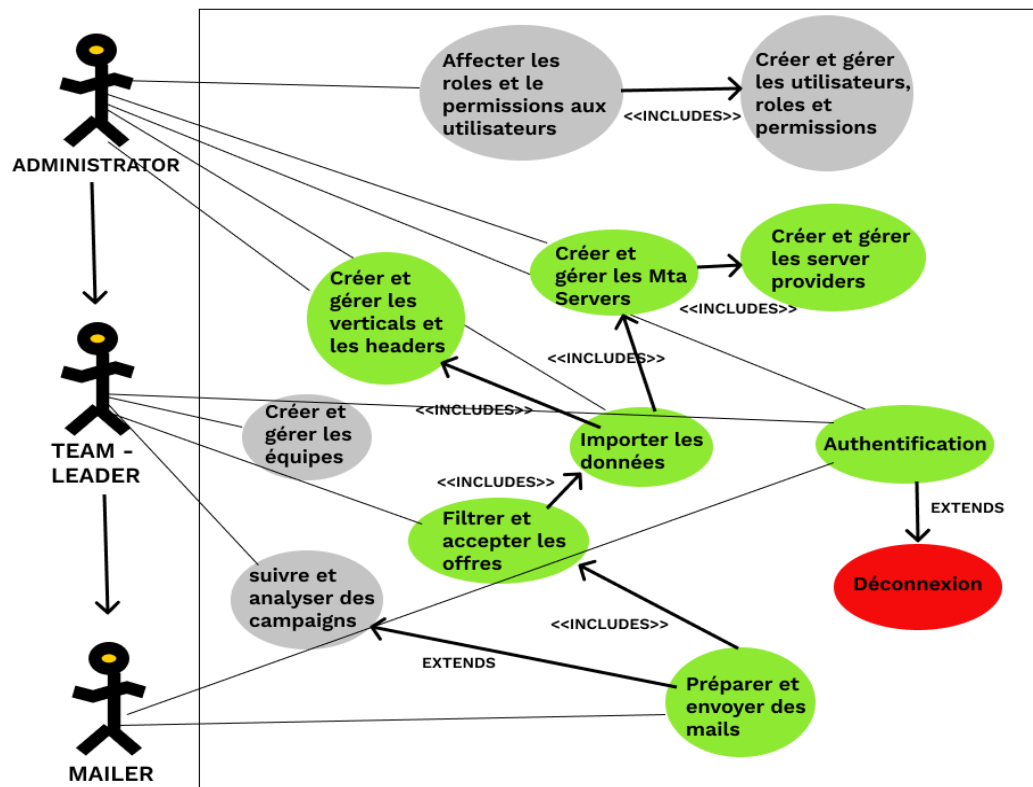


Figure 9

8 – Diagramme de séquence

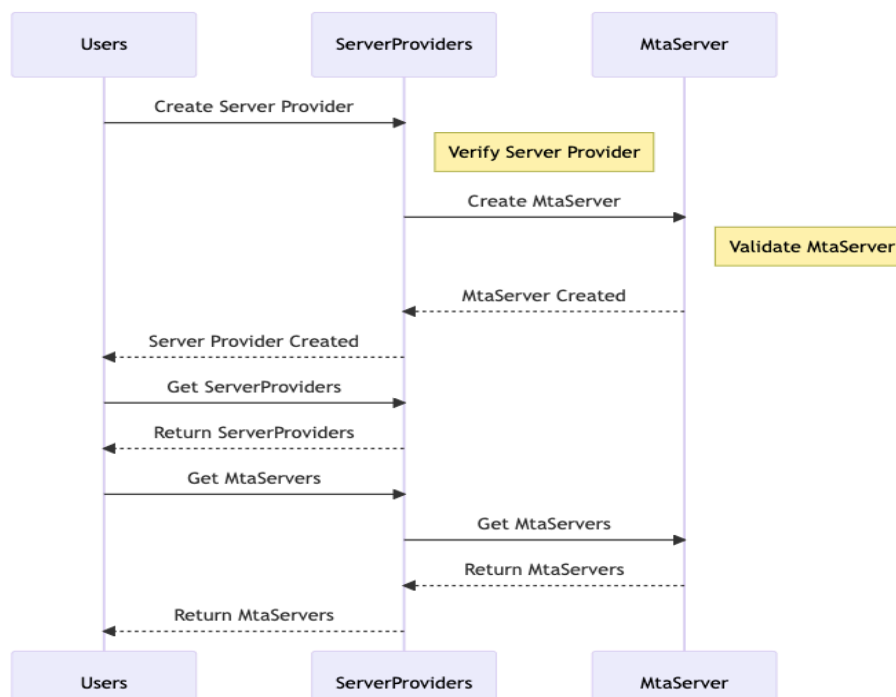


Figure 10

9 – Diagramme de Classe

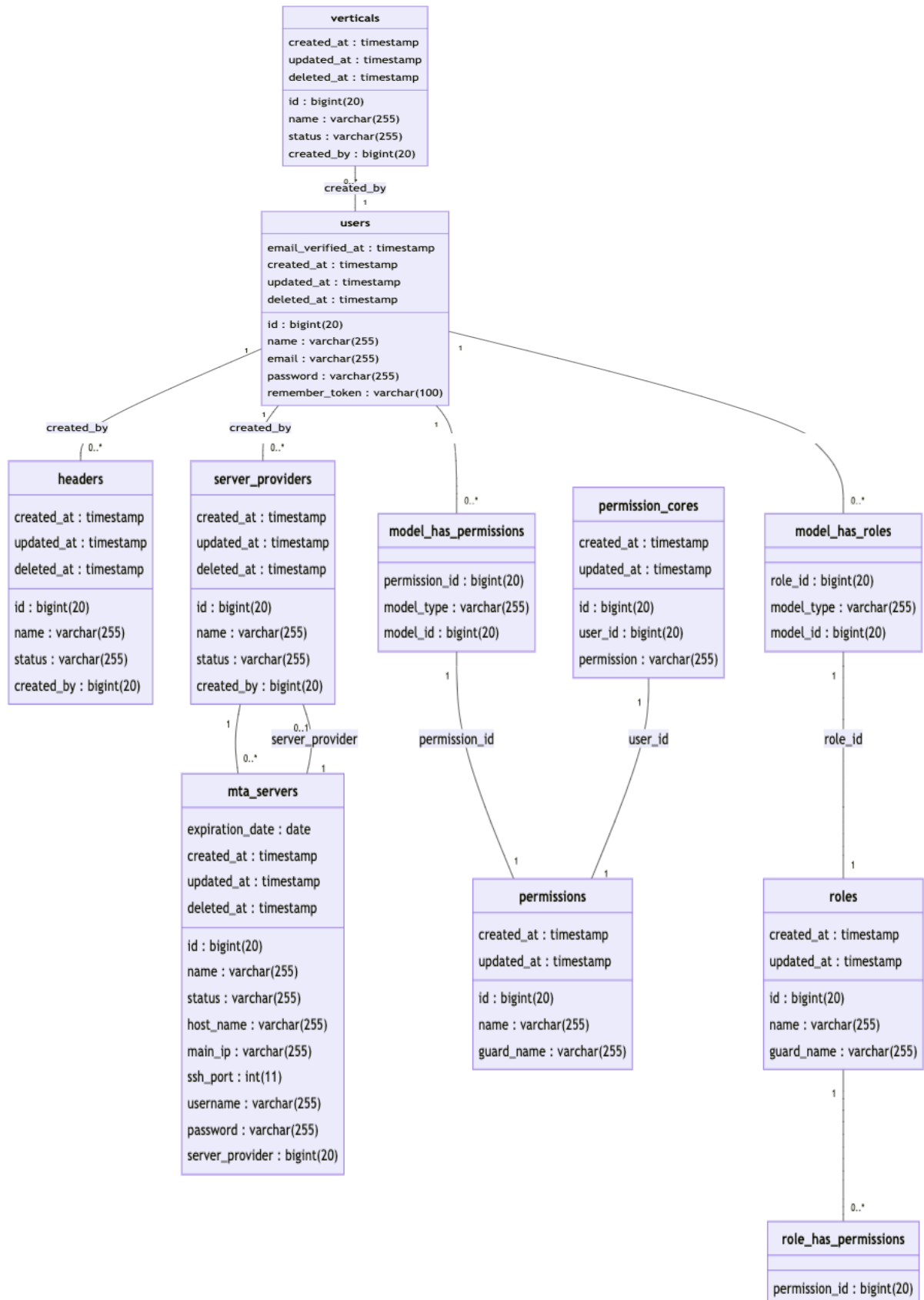


Figure 11

II - Réalisation

1 - Technologies utilisés :

Dans le cadre de notre stage, on a eu l'opportunité de travailler avec plusieurs technologies clés pour le développement de l'application. Voici un aperçu des technologies utilisées :

Laravel :

Laravel est un framework PHP moderne et puissant qui offre une structure de développement robuste et efficace. Il facilite la création de fonctionnalités avancées telles que la gestion des routes, l'authentification des utilisateurs, la gestion des bases de données et bien plus encore. On a utilisé Laravel comme framework de développement backend pour l'application, ce qui a permis de simplifier et d'accélérer le processus de développement.

Vue.js 3 :

Vue.js est un framework JavaScript progressif et réactif utilisé pour la création d'interfaces utilisateur interactives. On a utilisé Vue.js 3 pour développer le frontend de l'application. Grâce à sa structure basée sur les composants, Vue.js m'a permis de créer des interfaces utilisateur dynamiques et réutilisables, offrant ainsi une expérience utilisateur fluide et agréable.

Quasar :

Quasar est un framework basé sur Vue.js qui permet de développer des applications multiplateformes. On a utilisé Quasar pour créer une application qui fonctionne sur plusieurs plateformes, notamment sur le web, les appareils mobiles (iOS et Android) et les ordinateurs de bureau. Grâce à Quasar, On a pu développer une application cohérente et réactive, offrant une expérience utilisateur homogène sur différentes plateformes.

MySQL et PostgreSQL :

On a utilisé deux systèmes de gestion de bases de données relationnelles (SGBDR) pour l'application. MySQL et PostgreSQL sont tous deux des choix populaires et éprouvés en matière de gestion des données. On a utilisé MySQL et PostgreSQL pour stocker, gérer et interroger les données de l'application. Ces bases de données ont offert une performance élevée, une fiabilité et une sécurité, garantissant ainsi l'intégrité des données de l'application.

L'utilisation de ces technologies a été essentielle pour le développement de l'application lors de notre stage. Laravel a fourni un cadre solide pour la création du backend, tandis que Vue.js 3 et Quasar ont permis de développer un frontend réactif et multiplateforme. Les bases de données MySQL et PostgreSQL ont joué un rôle central dans le stockage et la gestion des données. Grâce à cette combinaison de technologies, on a pu contribuer au développement d'une application moderne, performante et conviviale.

Voici les raisons pour lesquelles Laravel et VueJS constituent une excellente combinaison pour le développement full-stack :

Applications réactives et événementielles :

Dans une application réactive, les événements sont poussés vers les composants lorsqu'ils se produisent, plutôt que d'attendre une interrogation ou une requête de l'interface utilisateur. Cela permet de créer des applications plus réactives qui ne nécessitent pas une consultation constante des données par les composants de votre application.

VueJS est un excellent framework pour construire des applications réactives grâce à son approche de composant unique, qui facilite l'organisation de votre code en petites fonctionnalités réutilisables à travers toute votre application.

Laravel est construit autour de la programmation événementielle, ce qui facilite la création d'un code modulaire qui réagit efficacement aux événements survenant dans votre application. Les deux frameworks simplifient la création d'applications réactives comme jamais auparavant !

Applications à page unique (SPA) :

Une application à page unique (SPA) est une application qui charge une seule page HTML et met à jour cette page dynamiquement selon les besoins, sans nécessiter un rafraîchissement complet de la fenêtre du navigateur.

Les SPAs offrent une meilleure expérience utilisateur que les sites web traditionnels, car ils permettent aux utilisateurs d'interagir avec votre application de manière similaire à une application mobile native, tout en nécessitant également du code côté serveur pour gérer toutes ces requêtes dynamiques.

En utilisant à la fois Laravel et VueJS, il est facile de construire de puissantes SPAs qui se chargent rapidement et fonctionnent efficacement sur n'importe quel appareil ! Les deux frameworks utilisent JavaScript comme langage principal, ce qui facilite l'apprentissage si vous êtes déjà familier avec l'un ou les deux.

Avec Laravel et VueJS travaillant ensemble, votre application peut être rapidement opérationnelle et avoir un aspect formidable !

Sécurité de l'application :

Laravel est conçu en ayant la sécurité de l'application à l'esprit, ce qui signifie qu'il dispose de plusieurs fonctionnalités intégrées qui rendent votre application plus sûre. Celles-ci incluent la protection contre les attaques CSRF, le hachage des mots de passe, la validation des entrées, et bien plus encore !

La sécurité est une priorité absolue pour de nombreuses entreprises, il est donc bon de savoir que vous pouvez construire une application sur Laravel sans vous soucier de ces aspects dès le premier jour ! Grâce à son accent sur la sécurité dès le départ, Laravel facilite la création d'une application sécurisée.

VueJS dispose de fonctionnalités intégrées qui aident à rendre votre application plus sûre par défaut, notamment son compilateur de templates (avec échappement automatique des balises HTML) et le mode strict par défaut.

Ces fonctionnalités travaillent ensemble pour garantir que votre application est aussi sécurisée que possible sans avoir à vous en soucier ! Bien que les deux frameworks facilitent grandement la sécurité, les utiliser ensemble rend la tâche encore plus facile.

Accès flexible aux bases de données :

Laravel et VueJS vous permettent tous deux d'accéder facilement aux bases de données à partir de votre code, ce qui signifie que vous pouvez construire rapidement des applications web sans avoir une connaissance spécialisée sur le fonctionnement des

bases de données ou sur la manière de les connecter à votre code.

Que vous souhaitiez une fonctionnalité CRUD simple ou des requêtes complexes, les deux frameworks vous offrent un accès facile dès le départ. Avec son support pour les bases de données relationnelles comme MySQL et PostgreSQL, ainsi que les options NoSQL comme MongoDB, CouchDB, Redis, etc., Laravel vous offre tout ce dont vous avez besoin pour vous lancer avec la programmation de bases de données en un rien de temps !

Il est également livré avec un support intégré pour les migrations afin de garder une trace des modifications de votre schéma de base de données. Cela facilite la maintenance d'une base de données propre et organisée sans avoir à écrire beaucoup de code à partir de zéro !

Déploiement sans serveur :

Un déploiement sans serveur est une nouvelle approche pour déployer des applications qui élimine bon nombre des tracas liés au déploiement de votre application sur un serveur. Avec un déploiement sans serveur, vous pouvez déployer votre application sans vous soucier de la configuration ou de la maintenance des serveurs pour l'héberger ; vous téléversez simplement votre code et laissez un fournisseur cloud comme Amazon Web Services (AWS) s'occuper de tout le reste !

Cela facilite le déploiement rapide des applications, car vous n'avez pas besoin d'un administrateur de serveur sur site ou d'une équipe DevOps pour commencer. Cela vous aide également à économiser de l'argent en éliminant les coûts inutiles liés à l'exécution de serveurs toute la journée juste pour que votre application puisse être en ligne !

Bien que les deux frameworks facilitent le déploiement des applications, les utiliser ensemble vous permet de tirer parti de leurs points forts lors du déploiement de votre application. Avec le support intégré de Laravel pour les fonctions AWS Lambda et sa capacité à fonctionner avec d'autres services comme Heroku et Digital Ocean dès le départ, le déploiement d'une nouvelle application est plus facile que jamais !

Le framework vous offre tout ce dont vous avez besoin dès le départ pour que vous puissiez vous lancer rapidement et sans douleur, vous permettant ainsi de vous concentrer sur la construction de grandes applications plutôt que de configurer des serveurs.

Migration de base de données :

Lorsque vous construisez une application web, il est souvent nécessaire de travailler avec des bases de données pour stocker les données. L'une de vos tâches essentielles en tant que développeur consiste à vous assurer que le schéma de votre base de données est toujours à jour.

Cela peut être un processus complexe, car il nécessite de suivre manuellement les modifications et de mettre à jour votre base de données en conséquence. Si vous commettez des erreurs, elles peuvent causer de nombreux problèmes à l'avenir ! Pour éviter ces problèmes, de nombreux développeurs choisissent d'utiliser des outils de migration de bases de données tels que Flyway ou Sequel Pro. Ces outils vous permettent d'écrire des scripts qui mettront automatiquement à jour votre base de données lorsque cela est nécessaire, de sorte qu'il n'y ait aucune chance de faire une erreur lors de sa mise à jour ultérieure.

Avec le support des migrations de Laravel et VueJS, vous pouvez utiliser ces outils pour vous aider à maintenir constamment à jour le schéma de votre base de données. Cela facilite l'ajout de nouvelles fonctionnalités ou la modification de celles existantes sans craindre de faire des erreurs !

2 - Outils utilisés:

Outre les technologies et les outils mentionnés précédemment, on a également utilisé plusieurs outils spécifiques pour créer des diagrammes et des modèles lors de mon stage. Voici une présentation de ces outils :

Visual Studio Code (VS Code) :

On a utilisé Visual Studio Code comme mon principal éditeur de code, mais il offre également des extensions et des plug-ins pour la création de diagrammes. On a utilisé des extensions telles que Draw.io Integration et PlantUML pour générer des diagrammes UML, des diagrammes de flux, des organigrammes, etc. directement dans VS Code. Ces extensions ont facilité la création de diagrammes en intégrant les fonctionnalités de diagramme dans mon flux de travail de développement.

phpMyAdmin et pgAdmin :

phpMyAdmin est un outil web pour gérer les bases de données MySQL, tandis que pgAdmin est spécifique à PostgreSQL. Ces outils offrent des fonctionnalités de visualisation des schémas de bases de données, ce qui peut être utilisé pour générer des diagrammes entité-relation (ER) ou des diagrammes de modèle de base de données. On a utilisé ces outils pour examiner la structure des bases de données et créer des diagrammes de modèles de données pour mieux comprendre la relation entre les tables et les entités.

Jira :

Jira est un outil de gestion de projet populaire utilisé pour suivre les tâches, les problèmes et les projets dans un environnement de développement. Bien qu'il ne soit pas spécifiquement conçu pour les diagrammes, il offre des fonctionnalités de visualisation telles que les tableaux Kanban et les diagrammes de Gantt. On a utilisé Jira pour suivre et organiser les tâches liées au développement des diagrammes, ainsi que pour collaborer avec l'équipe sur les différentes étapes du processus.

Figma :

Figma est un outil de conception d'interface utilisateur (UI) et de prototypage collaboratif. Bien qu'il soit principalement utilisé pour la conception d'interfaces utilisateur, il offre également des fonctionnalités de création de diagrammes tels que les organigrammes, les diagrammes de flux, les diagrammes de processus, etc. On a utilisé Figma pour créer des diagrammes d'interface utilisateur, des maquettes interactives et des diagrammes de flux pour mieux visualiser le flux d'utilisation de l'application.

L'utilisation de ces outils a été essentielle pour créer des diagrammes et des modèles précis et visuellement attrayants lors de mon stage. Ils ont facilité la création de diagrammes UML, de diagrammes de modèle de base de données, de diagrammes d'interface utilisateur et d'autres types de diagrammes nécessaires pour communiquer et documenter le processus de développement de l'application.

Github :

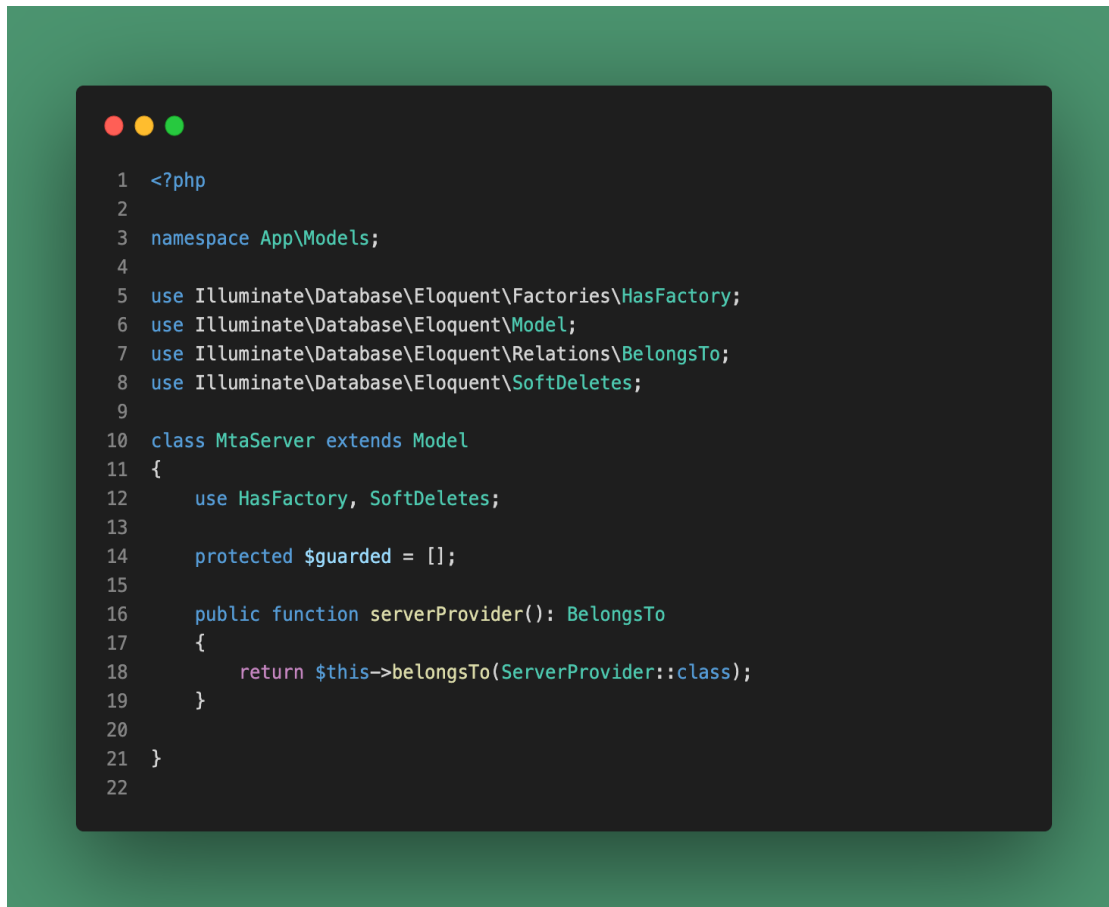
On a utilisé Github comme plateforme de gestion de versions et de collaboration pour héberger le code source de l'application. Grâce à Github, On a pu créer des branches, effectuer des commits et des pushes, fusionner des modifications et gérer les différentes versions du code. Cela a facilité la collaboration avec l'équipe en permettant un suivi précis des modifications, en facilitant les processus de revue de code et en assurant une intégration harmonieuse des modifications apportées par différents membres de l'équipe.

Workbench :

Workbench est un outil visuel de modélisation de bases de données pour MySQL. On a utilisé Workbench pour concevoir et modéliser la structure de la base de données de l'application. Cela m'a permis de créer des schémas de bases de données, de définir les tables et les relations entre elles, et d'optimiser la conception de la base de données. Workbench offre une interface conviviale et des fonctionnalités avancées pour faciliter la conception de la base de données et garantir la cohérence et l'intégrité des données.

En utilisant Visual Studio Code, phpMyAdmin, pgAdmin, Jira et Figma, on a pu bénéficier d'un ensemble complet d'outils pour créer, visualiser et collaborer sur des diagrammes dans le cadre de mon stage. Ces outils ont amélioré la productivité, la communication et la compréhension de l'équipe tout au long du processus de développement.

3 – Extrait du code utilisé



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\Relations\BelongsTo;
8 use Illuminate\Database\Eloquent\SoftDeletes;
9
10 class MtaServer extends Model
11 {
12     use HasFactory, SoftDeletes;
13
14     protected $guarded = [];
15
16     public function serverProvider(): BelongsTo
17     {
18         return $this->belongsTo(ServerProvider::class);
19     }
20
21 }
22
```

Figure 12

Dans Laravel, la propriété `guarded` est utilisée dans les modèles pour spécifier les attributs qui ne peuvent pas être massivement assignés lors de l'utilisation de la méthode `create` ou de la méthode `fill` avec un tableau de données.

Lorsque vous déclarez la propriété `guarded` dans un modèle, vous indiquez à Laravel quels attributs doivent être exclus de l'assignation de masse. Cela signifie que les attributs spécifiés dans la propriété `guarded` ne seront pas remplis automatiquement lorsque vous utilisez des méthodes telles que `create` ou `fill` avec un tableau de données.

```

1  /**
2   * Display a listing of the resource.
3   *
4   * @param Request $request
5   * @return AnonymousResourceCollection
6   */
7  public function index(Request $request): AnonymousResourceCollection
8  {
9      $query = MtaServer::query()
10         ->with('serverProvider');
11
12      // Search by different columns
13      if ($request->has('filter')) {
14          ( $searchTerm = $request->input('filter');
15            $query->where('name', 'like', "%$searchTerm%")
16              ->orWhere('status', 'like', "%$searchTerm%")
17              ->orWhere('created_by', 'like', "%$searchTerm%");
18          }
19
20      // Sorting
21      if ($request->has('sort_by')) {
22          ( $sortBy = $request->input('sort_by');
23            $sortDirection = $request->boolean('descending') ? 'desc' : 'asc';
24
25            $query->orderBy($sortBy, $sortDirection);
26          }
27
28      // Pagination
29      $mtas = $query->paginate($request->input('per_page'));
30
31      return MtaServerResource::collection($mtas)->additional([
32          'meta' => [
33              'sort_by' => $request->input('sort_by'),
34              'descending' => $request->boolean('descending'),
35          ]
36      ]);
37  }
38

```

Figure 13

Ce code représente une méthode du contrôleur qui récupère une liste paginée de serveurs MtaServer avec des options de recherche, de tri et de pagination. Voici une description ligne par ligne :

La méthode `index` prend un objet `Request` en paramètre et renvoie une `AnonymousResourceCollection`, qui est utilisée pour représenter une collection de ressources transformées.

Le commentaire initial décrit la fonctionnalité de la méthode et spécifie les paramètres d'entrée et de sortie.

La variable `$query` est initialisée en créant une requête de base pour le modèle `MtaServer` et en préchargeant la relation `serverProvider`.

Si le paramètre `filter` est présent dans la requête, une recherche est effectuée en utilisant les termes de recherche fournis dans les colonnes `name`, `status` et `created_by`. Les résultats sont filtrés en conséquence.

Si le paramètre `sort_by` est présent dans la requête, les résultats sont triés en fonction de la colonne spécifiée et de l'ordre de tri (ascendant ou descendant). La méthode `orderBy` est utilisée pour effectuer le tri.

Le paramètre `per_page` est utilisé pour paginer les résultats. La méthode `paginate` est appelée sur la requête pour récupérer les résultats paginés en fonction du nombre d'éléments par page spécifié.

Les résultats paginés sont transformés en une collection de ressources `MtaServerResource` à l'aide de la méthode `collection`.

Enfin, des métadonnées supplémentaires sont ajoutées à la réponse en utilisant la méthode `additional`. Ces métadonnées incluent les paramètres de tri (`sort_by`) et l'ordre de tri (`descending`) pour fournir des informations supplémentaires à l'application cliente.

Ce code permet de récupérer une liste paginée de serveurs `MtaServer` en fonction des options de recherche, de tri et de pagination fournies. Les résultats sont transformés en ressources `MtaServerResource` et renvoyés en tant que réponse JSON.

```

1  <?php
2
3  namespace App\Http\Resources;
4
5  use Illuminate\Http\Request;
6  use Illuminate\Http\Resources\Json\JsonResource;
7
8  /** @mixin \App\Models\MtaServer */
9  class MtaServerResource extends JsonResource
10 {
11     /**
12      * Transform the resource into an array.
13      *
14      * @return array<string, mixed>
15      */
16     public function toArray(Request $request): array
17     {
18         return
19             'id' => $this->id,
20             'name' => $this->name,
21             'status' => $this->status,
22             'server_provider' => new ServerProviderResource($this->whenLoaded(
23                 'serverProvider')),
24             'expiration_date' => $this->expiration_date,
25             'host_name' => $this->host_name,
26             'main_ip' => $this->main_ip,
27             'ssh_port' => $this->ssh_port,
28             'created_at' => $this->created_at->diffForHumans(),
29             'username' => $this->username,
30             'password' => $this->password
31         ];
32     }
33 }
34 }
35

```

Figure 14

Ce code définit une méthode `toArray` qui transforme une ressource `MtaServer` en un tableau associatif. Les propriétés du serveur, telles que l'identifiant, le nom, le statut, la date d'expiration, l'adresse IP, etc., sont incluses dans le tableau retourné. La ressource `ServerProviderResource` est également incluse pour la relation `'serverProvider'` préchargée. La date de création est formatée en utilisant la méthode `diffForHumans()` pour afficher une version lisible de la date.

III – Manuel d'utilisateur

1 - La maquette initiale :



Figure 15

La page de connexion (Login) est la première étape pour accéder au tableau de bord (Dashboard) de l'application web de gestion d'email marketing. Elle permet aux utilisateurs autorisés de se connecter à leur compte pour accéder aux fonctionnalités de l'application.

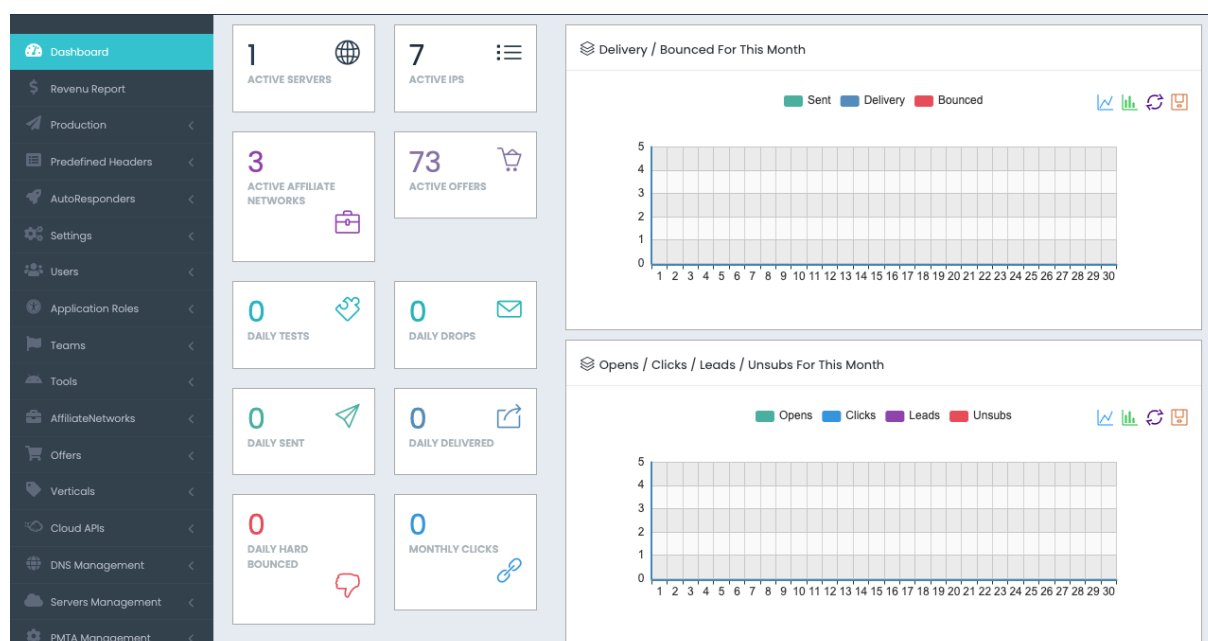


Figure 16

2 - Diagramme de navigation de l'application (jusqu'à cette date)

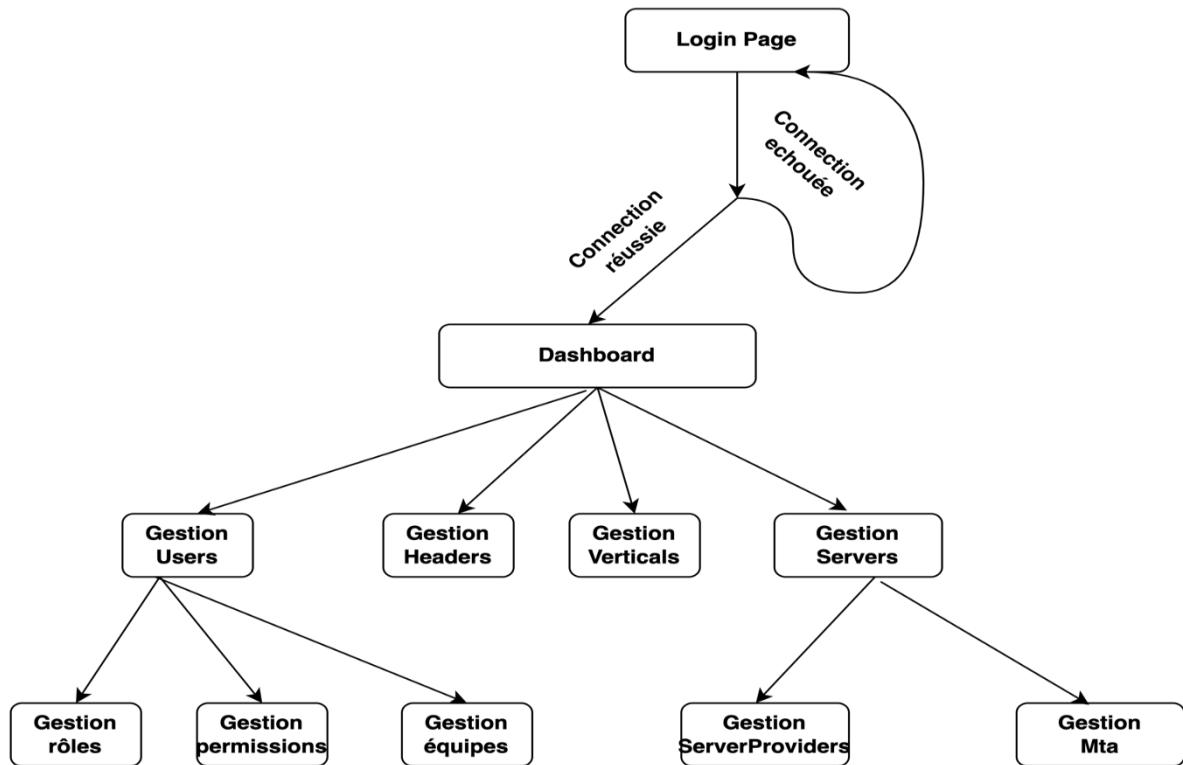


Figure 17

3 - Les écrans finaux :

Page d'authentification

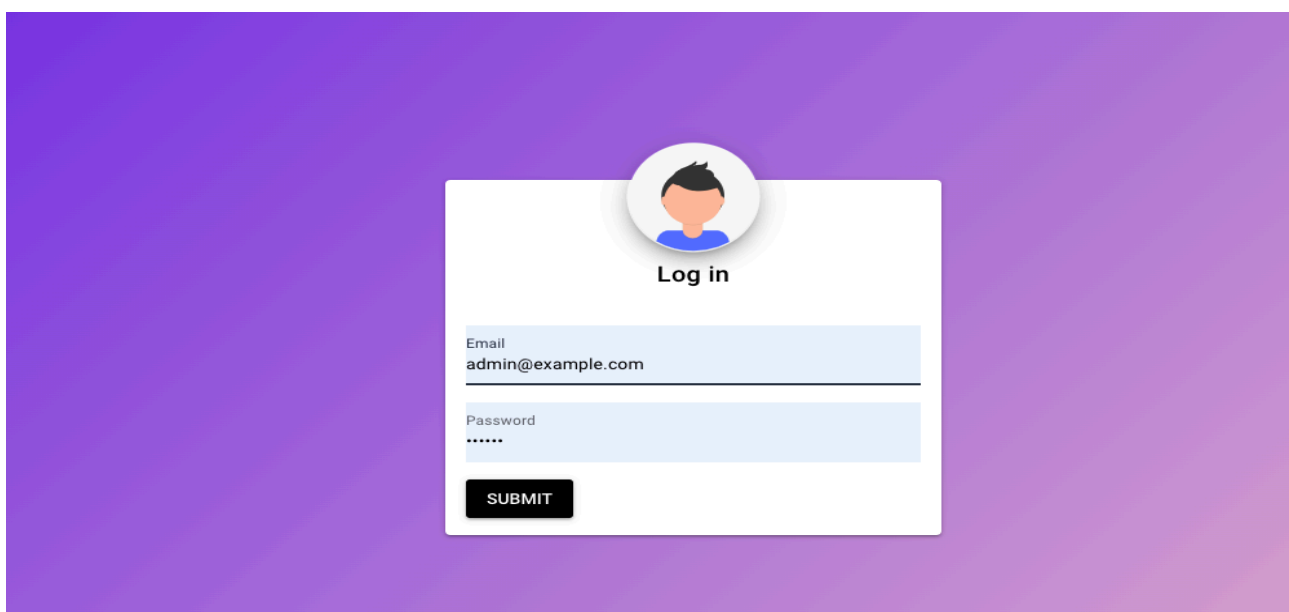


Figure 18

Page MTA Servers list

The screenshot shows a dashboard with a dark sidebar on the left containing navigation links: Dashboard, Gestion Users, Permissions, Roles, Verticals, Headers, Server Management, Server Providers, MTA SERVERS, and App Features. The main content area has a top bar with a user profile 'John Doe' and a 'LOGOUT' button. Below this is a table with columns: ID, Name, Status, Provider Name, Expiration Date, Host Name, Main IP, SSH Port, and Create. A single row is visible with ID 7, Name Mta1, Status active, Provider Name server1, Expiration Date 2023-06-13, Host Name Host1, Main IP 123.234.234.65, SSH Port 22, and Create 1 second. A '+ ' button is at the top left of the table, and a search bar is at the top right. At the bottom right, it says 'Records per page: 5' and '1-1 of 1'.

ID	Name	Status	Provider Name	Expiration Date	Host Name	Main IP	SSH Port	Create
7	Mta1	active	server1	2023-06-13	Host1	123.234.234.65	22	1 second

Figure 19

Ajouter un MTA Server

The screenshot shows the same dashboard as Figure 19, but with a modal form open for adding a new MTA server. The form has fields for: active (a dropdown menu), Provider (server1), Expiration Date (2023/06/13 with a calendar icon), Host Name (Host1), Main IP (123.234.234.65), SSH Port (22), username (test1), and password (pw1). The background table is dimmed.

ID	Name	Status	Provider Name	Expiration Date	Host Name	Main IP	SSH Port	Create
7	Mta1	active	server1	2023-06-13	Host1	123.234.234.65	22	1 second

Figure 20

IV - fiche technique pour l'administrateur

La méthode d'installation:

Voici les étapes générales pour installer notre application :

1. Prérequis :

- Assurez-vous d'avoir installé PHP sur votre machine, avec les extensions requises pour Laravel (par exemple, PDO, OpenSSL, Mbstring, etc.).
- Installez Node.js et npm (Node Package Manager) pour gérer les dépendances de Vue.js Quasar.
- Assurez-vous d'avoir une base de données compatible (MySQL, PostgreSQL, etc.) installée et configurée.

2. Clonage du projet :

- Utilisez Git pour cloner le projet depuis un référentiel Git distant (par exemple, GitHub). Vous pouvez utiliser la commande suivante : ``git clone <url_du_projet>``.
- Assurez-vous d'avoir les droits d'accès appropriés au référentiel.

3. Installation des dépendances du projet :

- Ouvrez une fenêtre de terminal et accédez au répertoire racine du projet.
- Exécutez la commande ``composer install`` pour installer les dépendances PHP définies dans le fichier `composer.json`.
- Ensuite, exécutez la commande ``npm install`` pour installer les dépendances JavaScript définies dans le fichier `package.json`.

4. Configuration de l'environnement :

- Dupliquez le fichier ``.env.example`` et renommez-le en ``.env``.
- Configurez les variables d'environnement dans le fichier ``.env`` selon votre configuration de base de données et d'autres paramètres spécifiques à votre application.

5. Génération de la clé d'application :

- Exécutez la commande ``php artisan key:generate`` pour générer une clé d'application unique.

6. Migration de la base de données :

- Exécutez la commande ``php artisan migrate`` pour exécuter les migrations et créer les tables de base de données requises.

7. Compilation des ressources frontales :

- Exécutez la commande ``npm run dev`` pour compiler les ressources frontales de Vue.js Quasar.

8. Lancement du serveur de développement :

- Exécutez la commande ``php artisan serve`` pour démarrer le serveur de développement Laravel.

Une fois ces étapes terminées, vous devriez pouvoir accéder à votre application web en ouvrant un navigateur et en accédant à l'URL indiquée par le serveur de développement Laravel.

Conclusion

En conclusion, notre stage chez OPEN ITC a été une expérience enrichissante et formatrice. Nous avons eu l'opportunité de travailler sur un projet concret d'application web d'email marketing, ce qui nous a permis d'approfondir nos connaissances en développement web et de mettre en pratique les compétences acquises lors de nos études.

Au cours de notre stage, nous avons pu collaborer avec une équipe compétente et passionnée, qui nous a guidés et soutenus tout au long du projet. Nous avons également pu découvrir le fonctionnement d'une entreprise spécialisée dans l'informatique, en comprenant les différentes étapes du développement logiciel, de l'analyse des besoins à la conception, en passant par le développement et les tests.

Nous avons relevé des défis tout au long du projet, notamment en termes de gestion du temps, de résolution de problèmes techniques et de coordination avec les membres de l'équipe. Grâce à notre détermination et à notre esprit d'équipe, nous avons su surmonter ces défis et atteindre les objectifs fixés.

Ce stage nous a également permis d'acquérir de nouvelles compétences techniques, de renforcer nos capacités de résolution de problèmes et d'améliorer notre capacité à travailler en équipe. Nous avons pu appliquer les connaissances théoriques que nous avons acquises lors de nos études à des situations réelles, ce qui a été une expérience précieuse pour notre développement professionnel.

En conclusion, notre stage chez OPEN ITC a été une expérience marquante qui nous a permis de grandir sur le plan professionnel. Nous sommes reconnaissants envers l'entreprise pour nous avoir offert cette opportunité et envers l'équipe pour son soutien et son encadrement. Nous sommes fiers des résultats que nous avons obtenus et des compétences que nous avons développées au cours de ce stage.

Nous tenons également à exprimer notre gratitude envers nos encadrants, nos professeurs, nos familles, nos amis et tous ceux qui nous ont soutenus tout au long de ce parcours. Leur soutien indéfectible, leurs encouragements et leurs conseils ont été d'une importance capitale dans notre réussite.

Enfin, ce rapport de stage témoigne de notre engagement, de notre détermination et de notre passion pour l'informatique. Nous sommes impatients de continuer à développer nos compétences et de relever de nouveaux défis dans notre carrière professionnelle.

Tableau des abréviations :

Abréviation	Signification
SGBDR	Système de Gestion de Base de Données Relationnelles
PERT	Program Evaluation and Review Technique (Technique d'Évaluation et de Révision de Programme)
DD	Dictionnaire de données
MLD	Modèle Logique de Données
MPD	Modèle Physique de Données
UML	Unified Modeling Language (Langage de Modélisation Unifié)
MYSQL	My Structured Query Language (Langage de Requête Structuré MySQL)
PGSQL PostgreSQL	(Système de Gestion de Base de Données Relationnelles)
PME	Petite et Moyenne Entreprise
PMI	Petite et Moyenne Industrie
PHP	Hypertext Preprocessor (Préprocesseur Hypertexte)

Table des annexes :

Numéro figure	Description	Pages
1	Composants en Jira	11
2	Epics en backlog	11
3	Lancer un sprint	12
4	Suivi du sprint	13
5	Burn Up	13
6	Diagramme Pert	14
7	MLD	18
8	MPD	18
9	Diagramme de Cas d'utilisation	19
10	Diagramme de séquence	19
11	Diagramme de classe	20
12	Modele MTA Server	29
13	Controller MTA Server	30
14	Resources	32
15	Login maquette initiale	33
16	Dashboard maquette initiale	33
17	Diagramme de navigation	34
18	Page authentication	34
19	MTA Servers list	35
20	Ajouter MTA Server	35

Webographie

<https://medium.com/javarevisited/why-laravel-and-vuejs-make-a-perfect-combination-for-full-stack-development-bea8d1fd6c3f>

<https://laravel.com/>

<https://vuejs.org/>

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

<https://www.postgresql.org/>

<https://www.ofppt.scholarvox.com/>