

RAPPORT DE STAGE DE FIN DE FORMATION POUR L'OBTENTION DU DIPLOME DE TECHNICIEN SUPERIEUR



THEME : Développement d'une plateforme e-commerce B2C pour le marché gabonais et italien : conception et développement

FILIERE : GENIE INFORMATIQUE



**Présenté et soutenu par :
ZIGH CHEICK HAMED TIDIANE
JORDY**

**Sous la direction de :
M. Maty MAMAN**

**Tuteur en entreprise :
M. NKENKE EYEBE
Samuel**



Promotion 2022-2023

DEDICACE

À mes parents,

Pour leur amour inconditionnel, leurs sacrifices et leur soutien indéfectible tout au long de mon parcours académique. Ce rapport est le fruit de vos efforts et de votre confiance en moi.

REMERCIEMENTS

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réussite de ce stage et à l'élaboration de ce rapport.

Mes remerciements s'adressent en premier lieu au directeur général de l'Institut National de la Poste des Technologies de l'information et de la Communication (INPTIC) de Libreville, **Monsieur Landry BADIALI**, pour son engagement dans la formation de ses étudiants.

Ma reconnaissance va également à l'ensemble du corps professoral de l'**Institut National des Postes et Technologies de l'Information et de la Communication (INPTIC)** pour la formation de qualité dispensée durant mon cursus DTS Génie Logiciel, qui m'a permis d'aborder ce stage avec les compétences nécessaires.

Je tiens à remercier **Mr Maty Maman** mon encadrant académique à l'INPTIC, pour son suivi régulier, ses orientations méthodologiques et ses précieux conseils dans la rédaction de ce rapport.

Je remercie aussi **Monsieur NKENKE EYEBE Samuel**, Président Directeur Général d'Alissa IA, pour m'avoir accueilli au sein de son entreprise et m'avoir confié la réalisation du projet GabonMarketHub. Sa confiance, son encadrement et ses conseils avisés ont été déterminants dans la réussite de ce stage.

Je remercie également **Monsieur David TCHIPANGUILA**, Directeur Technique d'Alissa IA, pour son accompagnement technique, sa disponibilité et son expertise qui m'ont permis de surmonter les difficultés rencontrées tout au long du projet.

Mes remerciements vont aussi à l'ensemble de l'équipe d'Alissa IA, notamment **Madame Madlyne MICKALA** (Direction Marketing et Commerciale), **Madame Noemy OBAME** (Direction Financière), et **Monsieur Kenny BIAN** (Direction RH et Affaires Juridiques), pour leur accueil chaleureux et leur collaboration.

Enfin, je remercie ma famille et mes proches pour leur soutien constant, leurs encouragements et leur patience tout au long de ma formation et de ce stage.

À tous, merci infiniment.

Résumé

Ce rapport présente le déroulement de mon stage effectué au sein d'Alissa IA, une startup gabonaise spécialisée dans le développement de solutions numériques intégrant l'intelligence artificielle. Le stage s'est déroulé du 25 juillet au 30 septembre 2025, soit une période de deux mois.

Ma mission principale consistait à contribuer au développement complet de **GabonMarketHub**, une plateforme e-commerce B2B innovante destinée à faciliter les échanges commerciaux entre entreprises gabonaises et fournisseurs locaux et Italiens.

Le développement a nécessité la mise en œuvre d'une architecture technique moderne, avec un backend Node.js/Express.js et MySQL, et un frontend React/TypeScript. La plateforme intègre des fonctionnalités avancées telles qu'un système de messagerie temps réel avec Socket.IO, une gestion complète des produits, des tableaux de bord personnalisés selon les rôles utilisateurs, et un système de notifications.

Le projet a été développé selon une méthodologie itérative, en suivant les principes Agile. J'ai été amené à développer l'ensemble de l'API REST (20+ routes), l'application frontend complète (40+ pages, 100+ composants), et à gérer une base de données complexe avec (30+ tables).

Cette expérience m'a permis d'acquérir une expertise pratique en développement web full-stack, de maîtriser des technologies modernes et recherchées, et de développer des compétences professionnelles essentielles. Le stage a également validé mon choix de carrière dans le développement web et m'a donné confiance pour aborder des projets complexes dans un environnement professionnel.

Mots-clés : Développement web, e-commerce B2B, React, Node.js, MySQL, Socket.IO, TypeScript, architecture logicielle, sécurité web, communication temps réel

Abstract

This report presents the course of my internship at Alissa IA, a Gabonese startup specialized in developing digital solutions integrating artificial intelligence. The internship took place from June 30 to August 30, 2025, a period of two months.

My main mission was to contribute to the complete development of **Gabon Market Hub**, an innovative B2B e-commerce platform designed to facilitate commercial exchanges between Gabonese companies and local and Italian suppliers.

The development required the implementation of a modern technical architecture, with a Node.js/Express.js and MySQL backend, and a React/TypeScript frontend. The platform integrates advanced features such as a real-time messaging system with Socket.IO, complete product and order management, personalized dashboards according to user roles, and a multi-channel notification system.

The project was developed using an iterative methodology, following Agile principles. I was responsible for developing the entire REST API (20+ routes), the complete frontend application (40+ pages, 100+ components), and managing a complex database with (30+ tables).

This experience allowed me to acquire practical expertise in full-stack web development, master modern and sought-after technologies, and develop essential professional skills. The internship also validated my career choice in web development and gave me confidence to approach complex projects in a professional environment.

Keywords : Web development, B2B e-commerce, React, Node.js, MySQL, Socket.IO, TypeScript, software architecture, web security, real-time communication

SOMMAIRE

Table des matières

DEDICACE	I
REMERCIEMENTS	II
Résumé.....	III
Abstract.....	III
Keywords : Web development, B2B e-commerce, React, Node.js, MySQL, Socket.IO, TypeScript, software architecture, web security, real-time communication	III
SOMMAIRE.....	IV
INTRODUCTION	1
PARTIE I : PRÉSENTATION DE L'ENTREPRISE OU DE.....	2
L'ORGANISATION D'ACCUEIL	2
Presentation d'Alissa IA	3
Historique et localisation d'Alissa IA.....	3
Organigramme et organisation d'Alissa IA	3
Activités principales d'Alissa IA.....	5
Missions et rôle de l'entreprise dans le secteur	5
Service concerné par le stage	6
Présentation du Projet : GabonMarketHub.....	7
Contexte et genèse du projet	8
Description fonctionnelle du projet.....	8
Architecture technique du projet	10
Enjeux et défis techniques	11
Résultats attendus	12
Missions confiée	13
2. Méthodologie.....	16
2.1. Méthodologie générale de développement	16
2.2. Méthodologie de travail quotidienne	17
2.3. Gestion des problèmes et difficultés	18
2.4. Communication et collaboration	18
2.5. Apprentissage continu	18
3. Résultats obtenus.....	18
3.1. Résultats techniques	19
3.2. Résultats fonctionnels.....	20
3.3. Résultats quantitatifs	20
3.4. Résultats qualitatifs.....	21
3.5. Apprentissages et compétences développées.....	21
4. Difficultés rencontrées.....	22

4.1. Difficultés techniques.....	22
4.2. Difficultés organisationnelles.....	23
4.3. Difficultés conceptuelles.....	23
4.4. Leçons apprises	24
5. Étude de cas : Implémentation du système de messagerie temps réel	24
5.1. Contexte.....	24
5.2. Objectifs	24
5.3. Analyse technique.....	24
5.4. Implémentation	24
5.5. Résultats.....	26
5.6. Enseignements tirés.....	26
5.7. Améliorations possibles	26
CONCLUSION.....	27
REFERENCES BIBLIOGRAPHIQUES.....	28
ANNEXES	I

INTRODUCTION

Dans un contexte où la digitalisation des entreprises et des administrations gabonaises constitue une priorité stratégique, l'intelligence artificielle et les technologies web modernes jouent un rôle déterminant dans la transformation numérique du continent africain. C'est dans ce cadre que j'ai effectué mon stage au sein d'**Alissa IA**, une startup innovante spécialisée dans le développement de solutions numériques intégrant l'intelligence artificielle.

Mon stage, d'une durée de deux mois du 2 juillet au 25 Septembre 2025 s'est déroulé au sein de **Direction Technique**, sous la supervision du Directeur Général, **Monsieur NKENKE EYEBE Samuel**. Durant cette période, j'ai participé activement au développement de **GabonMarketHub**, une plateforme e-commerce B2B visant à faciliter les échanges commerciaux entre entreprises gabonaises et fournisseurs locaux ou Italiens. Ce projet s'inscrit pleinement dans la vision d'Alissa IA : proposer des solutions technologiques innovantes et adaptées aux besoins spécifiques du marché gabonais, tout en contribuant à la digitalisation du tissu économique national.

L'objectif principal de ce rapport est de retracer mon parcours au sein d'Alissa IA et de présenter de manière structurée les différentes missions qui m'ont été confiées, les outils et méthodes employés, ainsi que les résultats obtenus. Il s'agit également d'analyser les compétences techniques et professionnelles acquises au cours de cette expérience et d'en dégager les enseignements personnels et académiques. Sur le plan académique, ce rapport témoigne de ma capacité à mobiliser les connaissances acquises lors de ma formation en **DTS Génie Informatique à l'INPTIC Libreville** dans un cadre professionnel concret. Sur le plan professionnel, il met en lumière la mise en œuvre d'un projet complet de développement d'application web, de la conception à la mise en production, en respectant les bonnes pratiques du développement moderne. Enfin, sur le plan personnel, il constitue un bilan d'apprentissage, d'autonomie et d'orientation pour la suite de mon parcours.

Le projet **GabonMarketHub** soulevait plusieurs défis techniques et fonctionnels majeurs. La problématique centrale était la suivante : **comment concevoir une plateforme e-commerce B2B performante, sécurisée et évolutive**, capable de gérer des milliers de produits, d'utilisateurs et de transactions simultanées, tout en garantissant une expérience utilisateur fluide et responsive sur différents supports. Cette problématique a orienté l'ensemble du travail technique, depuis la conception de l'architecture jusqu'à la mise en production.

Pour y répondre, j'ai adopté une démarche méthodologique rigoureuse et structurée. Avant chaque phase de développement, j'ai mené des recherches sur les meilleures pratiques et documenté toutes les décisions techniques prises. Le projet a été réalisé de manière **itérative**, suivant une approche **agile**, afin de permettre une amélioration progressive des fonctionnalités. Chaque composant a été développé, testé et intégré progressivement pour assurer la qualité et la stabilité du système. Le déploiement a été accompagné de mécanismes de **monitoring** et de **tests manuels approfondis** afin de garantir la fiabilité du produit final.

Ma mission principale consistait à concevoir et développer l'architecture complète de l'application. Sur le plan **backend**, j'ai implémenté une API REST avec **Node.js** et **Express.js**, intégré un système d'authentification sécurisé avec **JWT**, et conçu une base de données **MySQL** optimisée pour la performance. J'ai également mis en place un système de messagerie en temps réel grâce à **Socket.IO**. Sur le plan **frontend**, j'ai développé une application moderne avec **React** et **TypeScript**, en veillant à créer une interface ergonomique et responsive grâce à **Tailwind CSS**. L'ensemble des fonctionnalités principales a été intégré : gestion des produits et catégories, messagerie instantanée, tableaux de bord personnalisés selon le type d'utilisateur et administration des contenus. Des validations robustes et des mesures de sécurité ont été mises en œuvre pour assurer la fiabilité du système.

Ce stage m'a permis de mettre en pratique l'ensemble des compétences acquises au cours de ma formation, tout en développant de nouvelles aptitudes techniques et professionnelles. Il m'a offert une expérience complète du cycle de vie d'un projet web de la conception initiale à la mise en production et m'a sensibilisé

à l'importance du travail collaboratif, de la rigueur méthodologique et de la veille technologique dans le métier de développeur logiciel.



PARTIE I : PRÉSENTATION DE L'ENTREPRISE OU DE L'ORGANISATION D'ACCUEIL

Presentation d'Alissa IA

Historique et localisation d'Alissa IA

Alissa IA (Agence de création de Logiciels Innovants pour Systèmes et Solutions d'Administration, grâce à l'Intelligence Artificielle) est une startup gabonaise fondée en **février 2024** par trois entrepreneurs visionnaires : **Monsieur NKENKE EYEBE Samuel**, **Monsieur ELOKO William** et **Monsieur MANIMA Kévine**. Leur ambition était de faire de l'intelligence artificielle (IA) un pilier essentiel pour la digitalisation des entreprises et des administrations gabonaises.

Dès sa création, Alissa IA a établi des partenariats stratégiques avec plusieurs entités de premier plan : - **Direction Générale de la Statistique** : Partenariat pour le développement de solutions d'analyse de données - **Harmonie Home** : Collaboration sur des projets d'automatisation - **Sikul** : Partenariat technologique - **Société d'Incubation Numérique du Gabon (SING SA)** : Incubation et soutien au développement - **Pam Cakes** : Projet de digitalisation commerciale

Ces collaborations ont permis à la startup de développer des solutions d'IA innovantes, adaptées aux besoins spécifiques des entreprises et administrations gabonaises. La vision d'Alissa IA est de permettre à l'Afrique de maîtriser et d'exploiter l'intelligence artificielle, non seulement pour s'adapter aux évolutions technologiques, mais aussi pour innover et imposer sa propre vision du futur numérique.

La startup est une **Société par Actions Simplifiée (SAS)** au capital de **2 000 000 FCFA**, ayant son siège social à **BP 2280, Rue Pecqueur, Libreville, Gabon**.

Organigramme et organisation d'Alissa IA

L'organigramme d'Alissa IA reflète une **structure hiérarchique claire et fonctionnelle**, favorisant la coordination entre les différentes directions et la bonne exécution des projets. Cette organisation garantit une communication fluide, une répartition équilibrée des responsabilités et une cohérence dans les actions menées par l'entreprise.

2.1. Direction Générale

La **Direction Générale**, dirigée par le **CEO**, est responsable de l'orientation stratégique globale de l'entreprise. Elle supervise l'ensemble des activités et prend les décisions majeures relatives à la croissance, à la stratégie et au développement d'Alissa IA. Le Directeur Général assure la cohérence entre les différents pôles : techniques, juridiques, commerciaux et financiers afin d'aligner les objectifs de chaque service sur la vision globale de l'entreprise.

2.2. Direction des Ressources Humaines et des Affaires Juridiques

Cette direction a pour mission principale la **gestion du capital humain** et la **conformité juridique** de l'entreprise. Elle s'occupe du recrutement, de la gestion administrative du personnel, du suivi des contrats, ainsi que du respect des obligations légales et sociales en vigueur au Gabon. Elle veille à instaurer un **environnement de travail sain, conforme aux normes** et propice au développement professionnel des employés et des stagiaires. Son rôle est essentiel dans la création d'un cadre organisationnel équilibré et juridiquement sécurisé.

2.3. Direction Technique

La **Direction Technique** constitue le **cœur de la production numérique** d'Alissa IA. Elle est chargée de la **conception, du développement et de la maintenance** des solutions technologiques proposées par l'entreprise. Elle encadre les équipes de développeurs, d'ingénieurs et de techniciens, et coordonne leurs travaux afin d'assurer la qualité et la fiabilité des produits numériques. Des employés et stagiaires techniques sont régulièrement intégrés à ce service pour renforcer les capacités de production. C'est au sein de cette direction que j'ai principalement collaboré durant mon stage, notamment sur le **développement du projet Gabon Market Hub**.

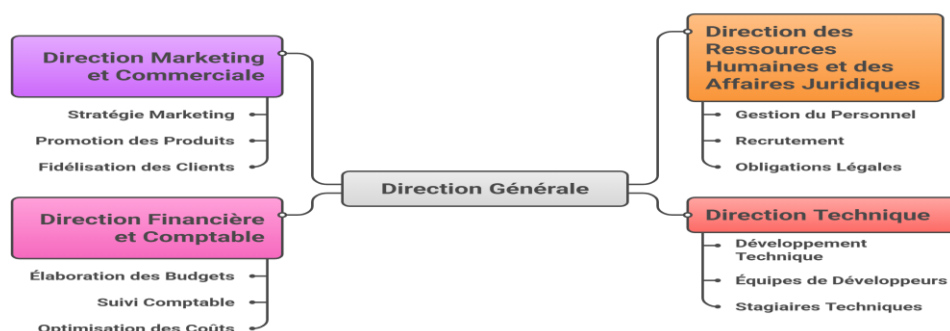
2.4. Direction Financière et Comptable

La **Direction Financière et Comptable** assure la **gestion et l'optimisation des ressources financières** de l'entreprise. Elle est responsable de l'élaboration des budgets, du suivi comptable, de la production des bilans financiers et du contrôle des dépenses. Son rôle est de garantir la **viabilité économique** d'Alissa IA et d'assurer une gestion saine et transparente des ressources disponibles. Elle participe également à la planification stratégique en fournissant des analyses financières utiles à la prise de décision.

2.5. Direction Marketing et Commerciale

La **Direction Marketing et Commerciale** définit et met en œuvre la **stratégie de communication et de développement commercial** d'Alissa IA. Elle travaille sur la **visibilité de la marque**, la **promotion des produits** (notamment par le biais du marketing digital), la **prospection de nouveaux clients** et la **fidélisation de la clientèle existante**. Elle encadre une équipe composée d'employés et de stagiaires, et collabore étroitement avec les autres services pour adapter l'offre commerciale aux besoins du marché. Bien que j'aie été initialement affecté à cette direction, mes compétences techniques m'ont conduit à collaborer de manière régulière avec la **Direction Technique** sur des projets de développement web.

Structure Organisationnelle d'Alissa IA



Activités principales d'Alissa IA

Alissa IA concentre ses efforts sur trois activités principales, chacune représentant un pilier stratégique de l'entreprise :

3.1. Développement web, mobile, desktop

La startup propose des services complets de développement d'applications pour différentes plateformes :

- **Développement web** : Création de sites internet dynamiques, de plateformes web interactives, de sites e-commerce, d'applications web progressives (PWA), etc. Toutes ces solutions sont conçues dans l'optique d'offrir une présence digitale professionnelle aux entreprises et aux entrepreneurs.
- **Développement mobile** : Conception d'applications pour smartphones et tablettes (Android/iOS), répondant aux besoins spécifiques des utilisateurs mobiles, avec une attention particulière sur l'ergonomie et l'expérience utilisateur.
- **Développement desktop** : Développement de logiciels installables sur PC ou Mac, souvent utilisés pour des solutions internes (gestion de facturation, automatisation, etc.).

Ces services permettent à Alissa IA d'apporter des solutions technologiques sur mesure et adaptées à des supports numériques variés.

3.2. Développement intégrant l'IA générative et automatisation

Cette activité est au cœur de l'identité d'Alissa IA et représente son principal avantage concurrentiel :

- **IA générative** : Utilisation de technologies d'intelligence artificielle capables de générer du contenu (images, texte, vidéos, code, etc.) à partir de données ou d'instructions. Cela inclut par exemple la création de Chatbots intelligents comme leur dernière invention "Whats'Bot", un chatbot intégré à WhatsApp.
- **Automatisation** : Mise en place de systèmes permettant de simplifier ou de remplacer des tâches humaines répétitives. Cela comprend l'automatisation des réponses aux clients, la prise de rendez-vous, la gestion des workflows, etc.

Ces solutions permettent aux entreprises de gagner en temps, d'augmenter leur productivité et d'optimiser leurs processus opérationnels.

3.3. Formations

Alissa IA propose également des formations destinées à différents profils (étudiants, professionnels, entreprises) sur les domaines suivants :

- Le développement web et mobile
- L'utilisation de l'intelligence artificielle
- L'automatisation des processus
- La communication et le marketing digital

Ces formations permettent de transmettre des compétences techniques et numériques à ceux qui souhaitent évoluer dans le métier du digital et contribuent ainsi à la montée en compétences du secteur technologique gabonais.

Missions et rôle de l'entreprise dans le secteur

Alissa IA occupe une position unique dans le paysage technologique gabonais. En tant que startup spécialisée dans l'intelligence artificielle et le développement de solutions numériques, elle joue un rôle pionnier dans la transformation digitale des entreprises et administrations locales.

4.1. Positionnement stratégique

L'entreprise se positionne comme un acteur de référence dans le domaine de l'intelligence artificielle appliquée au contexte africain et gabonais. Elle développe des solutions adaptées aux spécificités locales, ce qui constitue un avantage concurrentiel majeur face aux solutions génériques importées.

4.2. Contribution à l'écosystème numérique gabonais

Alissa IA contribue activement au développement de l'écosystème numérique gabonais en : - Formant des compétences locales en IA et développement web - Créant des emplois dans le secteur technologique - Participant à l'incubation de projets innovants via son partenariat avec SING SA - Développant des solutions open-source et propriétaires

4.3. Vision et impact

La vision d'Alissa IA est de faire en sorte que l'Afrique soit capable de maîtriser et d'exploiter l'intelligence artificielle pour non seulement s'adapter mais aussi innover et imposer sa propre vision du futur. Cette ambition se traduit par un engagement fort dans la recherche, le développement et l'innovation technologique.

Service concerné par le stage

Durant mon stage, j'ai été principalement affecté à la **Direction Technique** d'Alissa IA. Ce département constitue le **cœur opérationnel** de la startup, car il est directement responsable de la conception, du développement et de la maintenance des solutions numériques innovantes de l'entreprise.

La **Direction Technique** joue un rôle stratégique dans la croissance d'Alissa IA. Elle transforme les idées et concepts issus de la recherche en produits technologiques concrets et performants. Ses missions principales consistent à :

- **Concevoir et développer** les plateformes et applications web ;
- **Assurer la maintenance et la mise à jour** des solutions existantes ;
- **Garantir la sécurité, la performance et la fiabilité** des produits numériques ;
- **Collaborer avec les autres directions**, notamment la Direction Marketing et Commerciale, pour valoriser les solutions développées.

C'est dans ce cadre que j'ai participé activement au **développement du projet GabonMarketHub**, une plateforme numérique visant à mettre en relation les acteurs du marché gabonais. Mon travail a principalement porté sur le **développement full stack**.

Cette expérience m'a offert une vision d'ensemble du fonctionnement d'une startup technologique et m'a permis de comprendre comment les équipes techniques et commerciales travaillent de manière complémentaire pour atteindre les objectifs stratégiques de l'entreprise.

PARTIE II : CORPS DU RAPPORT

Présentation du Projet : GabonMarketHub

Contexte et genèse du projet

GabonMarketHub est une plateforme e-commerce B2B (Business-to-Business) innovante conçue pour faciliter les échanges commerciaux entre entreprises gabonaises et fournisseurs locaux et internationaux. Ce projet s'inscrit dans la stratégie d'Alissa IA de développer des solutions technologiques adaptées aux **besoins spécifiques du marché gabonais**.

1.1- Contexte économique et technologique

Le marché B2B gabonais présente plusieurs défis :

- **Fragmentation** : Les entreprises ont du mal à identifier et contacter les fournisseurs adaptés à leurs besoins
- **Manque de visibilité** : Les fournisseurs locaux peinent à se faire connaître auprès des entreprises acheteuses
- **Processus complexes** : Les échanges commerciaux traditionnels sont souvent lourds et peu efficaces
- **Digitalisation limitée** : Peu de plateformes adaptées au contexte gabonais

Face à ces constats, Alissa IA a décidé de créer une plateforme numérique qui permettrait de : - Centraliser l'offre et la demande commerciale B2B - Faciliter les mises en relation entre acheteurs et fournisseurs - Moderniser les processus commerciaux - Contribuer au développement économique local

1.2- Objectifs stratégiques du projet

Le développement de Gabon Market Hub répond à plusieurs objectifs stratégiques :

Pour Alissa IA : - Développer une expertise en plateformes e-commerce complexes - Créer une solution réutilisable et adaptable à d'autres marchés africains - Démontrer la capacité de la startup à livrer des projets d'envergure - Générer une nouvelle source de revenus

Pour le marché gabonais : - Faciliter l'accès des entreprises aux fournisseurs - Promouvoir les produits et services locaux - Réduire les coûts de transaction commerciaux - Contribuer à la digitalisation de l'économie gabonaise

Description fonctionnelle du projet

GabonMarketHub est une plateforme web complète offrant un ensemble de fonctionnalités avancées pour gérer tous les aspects d'un écosystème e-commerce B2B.

2.1- Fonctionnalités principales

a- Gestion des utilisateurs et authentification

La plateforme supporte trois types d'utilisateurs avec des droits et fonctionnalités distincts :

- **Acheteurs (role_id: 1)** : Particuliers ou professionnels recherchant des produits et services
 - ❖ Inscription et création de profil
 - ❖ Recherche et navigation dans le catalogue de produits
 - ❖ Ajout de produits aux favoris
 - ❖ Communication avec les fournisseurs via messagerie
 - ❖ Gestion de l'historique d'achats
 - ❖ Tableau de bord personnalisé
- **Fournisseurs (role_id: 2)** : Entreprises proposant des produits ou services

- ❖ Inscription avec validation des documents (certificats, statuts)
- ❖ Gestion complète du catalogue de produits
- ❖ Gestion des stocks et des prix
- ❖ Communication avec les clients
- ❖ Statistiques de vente et performance
- ❖ Tableau de bord avec indicateurs commerciaux

➤ **Administrateurs (role_id: 3)** : Équipe de gestion de la plateforme

- ❖ Gestion globale des utilisateurs
- ❖ Modération du contenu (produits, avis)
- ❖ Validation des inscriptions fournisseurs
- ❖ Gestion des catégories et paramètres système
- ❖ Analytics et statistiques globales
- ❖ Support technique

b- Gestion des produits et catégories

- **Catalogue hiérarchique** : Système de catégories et sous-catégories avec navigation intuitive
- **Recherche avancée** : Moteur de recherche avec filtres multiples (prix, catégorie, localisation, note)
- **Gestion des produits** : CRUD complet avec upload d'images multiples, description riche, spécifications techniques
- **Système d'offres** : Gestion des promotions, réductions et offres spéciales avec dates de validité
- **Avis et évaluations** : Système de notation et commentaires pour les produits et fournisseurs

c- Messagerie et communication

- **Messagerie temps réel** : Communication instantanée entre acheteurs et fournisseurs via Socket.IO
- **Conversations structurées** : Système de conversations liées aux produits
- **Indicateurs de statut** : Messages lus/non lus, indicateurs de frappe
- **Notifications** : Alertes en temps réel pour nouveaux messages, commandes, etc.

d- Notifications

- **Système de notifications multi-canal** :
 - ❖ Notifications in-app
 - ❖ Notifications par email
 - ❖ Notifications temps réel via Socket.IO
- **Types de notifications** : Nouveaux messages, mises à jour de commande, nouveaux avis, alertes produits

e- Tableaux de bord personnalisés

Chaque type d'utilisateur dispose d'un tableau de bord adapté à ses besoins :

- **Dashboard Acheteur** : Statistiques personnelles, favoris récents, commandes en cours, recommandations.
- **Dashboard Fournisseur** : Statistiques de vente, produits populaires, commandes à traiter, performance
- **Dashboard Administrateur** : Vue d'ensemble de la plateforme, analytics, gestion des utilisateurs

f- Fonctionnalités avancées

- **Système de favoris** : Sauvegarde des produits et fournisseurs préférés
- **Blog et événements** : Section d'actualités commerciales et événements
- **Recherche avancée** : Filtres complexes avec recherche par mots-clés, catégorie, prix, localisation
- **Responsive design** : Interface optimisée pour desktop, tablette et mobile
- **Gestion des documents** : Upload et validation de documents pour l'inscription fournisseur

Architecture technique du projet

3.1- Stack technologique

Le projet utilise une architecture moderne basée sur des technologies JavaScript/TypeScript, avec une séparation claire entre le frontend et le backend.

3.1.1. Backend (API REST)

- **Technologies principales** :
 - ❖ **Node.js** : Runtime JavaScript côté serveur
 - ❖ **Express.js** : Framework web pour la création de l'API REST
 - ❖ **MySQL2** : Base de données relationnelle
 - ❖ **Socket.IO** : Communication temps réel (WebSocket)
- **Sécurité et validation** :
 - ❖ **JWT (jsonwebtoken)** : Authentification par tokens
 - ❖ **bcryptjs** : Hachage sécurisé des mots de passe
 - ❖ **Helmet** : Sécurisation des en-têtes HTTP
 - ❖ **CORS** : Gestion des requêtes cross-origin
 - ❖ **express-rate-limit** : Protection contre les abus (rate limiting)
 - ❖ **Joi / express-validator** : Validation des données
 - ❖ **isomorphic-dompurify** : Protection XSS
- **Utilitaires** :
 - ❖ **Multer** : Upload de fichiers
 - ❖ **Nodemailer** : Envoi d'emails (vérification, notifications)
 - ❖ **Axios** : Requetes HTTP
 - ❖ **dotenv** : Gestion des variables d'environnement

3.1.2. Frontend (Application React)

- **Technologies principales** :
 - ❖ **React 18.3.1** : Bibliothèque UI moderne
 - ❖ **TypeScript 5.5.3** : Typage statique pour la robustesse
 - ❖ **Vite 5.4.20** : Build tool ultra-rapide
- **UI et styling** :
 - ❖ **Shadcn/ui** : Composants UI accessibles et personnalisables
 - ❖ **Radix UI** : Primitives UI headless
 - ❖ **Tailwind CSS 3.4.11** : Framework CSS utilitaire
 - ❖ **Lucide React** : Bibliothèque d'icônes modernes
- **État et données** :
 - ❖ **React Query (@tanstack/react-query)** : Gestion du cache et synchronisation avec le serveur
 - ❖ **React Hook Form** : Gestion performante des formulaires
 - ❖ **Zod** : Validation des schémas TypeScript

- **Communication :**
 - ❖ **Axios** : Client HTTP pour les requêtes API
 - ❖ **Socket.IO Client** : Communication temps réel
 - ❖ **React Router DOM** : Routing pour Single Page Application

3.1.3. Base de données

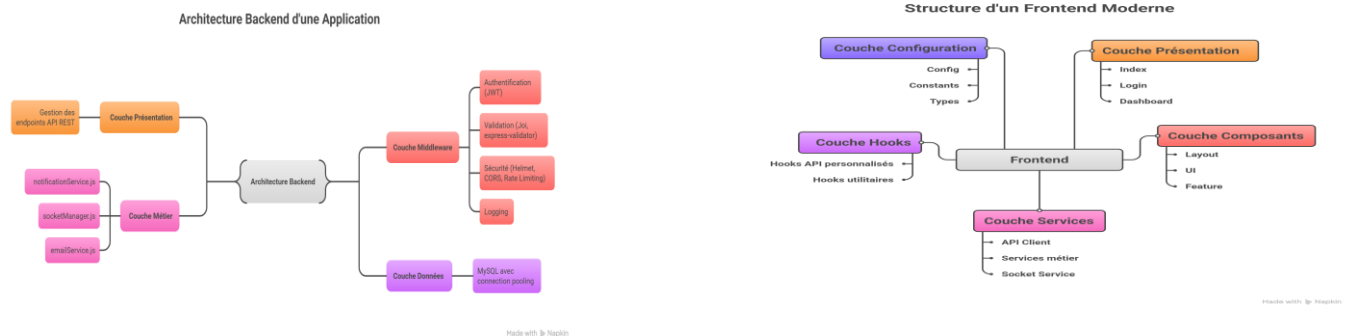
MySQL avec une architecture relationnelle complexe comprenant :

- **30+ tables** organisées en modules fonctionnels
 - ❖ **Relations** avec clés étrangères pour l'intégrité référentielle
 - ❖ **Indexes** optimisés pour les performances des requêtes
 - ❖ **Contraintes** d'intégrité et validation au niveau base

Tables principales : - utilisateurs, entreprises, produits, messages, avis, notifications - categories, favoris, adresses, documents, devis, certifications, etc.

3.2. Architecture en couches

Le projet suit une architecture en couches qui garantit la séparation des responsabilités :



3.3. Communication temps réel

Le projet intègre **Socket.IO** pour la communication temps réel, permettant :

- Envoi et réception de messages en temps réel
- Notifications instantanées
- Mise à jour des statuts de commande en direct
- Indicateurs de présence et de frappe

Enjeux et défis techniques

Le développement de Gabon Market Hub a présenté plusieurs défis techniques majeurs :

4.1. Scalabilité

La plateforme doit être capable de gérer :

- Des milliers d'utilisateurs simultanés
- Des dizaines de milliers de produits
- Des centaines de transactions par jour
- Des milliers de messages temps réel

Solutions apportées :

- Connection pooling MySQL pour optimiser les connexions base de données
- Architecture modulaire permettant la montée en charge horizontale
- Mise en cache (implémenter avec Redis en production)

- Optimisation des requêtes SQL avec indexes appropriés

4.2. Sécurité

Les enjeux de sécurité sont critiques pour une plateforme gérant des données sensibles :

- Informations personnelles des utilisateurs
- Documents confidentiels

Solutions apportées : -

- Authentification JWT sécurisée
- Hachage bcrypt pour les mots de passe
- Protection XSS avec DOMPurify
- Protection SQL injection avec prepared statements
- Rate limiting pour éviter les attaques par déni de service
- Validation stricte des données entrantes - HTTPS obligatoire en production

4.3. Expérience utilisateur

L'interface doit être intuitive, rapide et accessible sur tous les supports.

Solutions apportées :

- Design responsive avec Tailwind CSS (mobile-first)
- Optimisation des performances (lazy loading, code splitting)
- Feedback utilisateur en temps réel (toasts, notifications)
- Interface moderne avec Shadcn/ui
- Gestion d'erreurs gracieuse avec messages clairs

4.4. Complexité métier

La gestion des relations complexes entre utilisateurs, produits et messages nécessite une modélisation rigoureuse.

Solutions apportées :

- Architecture de base de données normalisée
- Relations bien définies avec contraintes d'intégrité
- Services métier dédiés pour chaque domaine fonctionnel
- Validation à tous les niveaux (frontend, backend, base de données)

Résultats attendus

À l'issue du développement, Gabon Market Hub doit permettre :

- **Pour les acheteurs** : Faciliter la recherche et l'achat de produits B2C économiser du temps dans la prospection de fournisseurs
- **Pour les fournisseurs** : Augmenter la visibilité de leur offre, faciliter la gestion des clients
- **Pour Alissa IA** : Démontrer sa capacité à livrer des projets complexes, créer une source de revenus, développer une expertise réutilisable

1- Planification et méthodologie de développement

Le projet a été développé selon une approche itérative, en suivant les principes de la méthodologie agile :

- **Phase 1** : Analyse des besoins et conception de l'architecture
- **Phase 2** : Développement du backend (API REST)

- **Phase 3** : Développement du frontend (interface utilisateur)
- **Phase 4** : Intégration et tests
- **Phase 5** : Déploiement et documentation

Cette approche a permis d'obtenir rapidement un prototype fonctionnel, puis d'itérer pour améliorer et enrichir la plateforme.

Missions confiée

1.1. Mission principale

Dans le cadre de mon stage chez Alissa IA, ma mission principale consistait à contribuer au développement complet de **GabonMarketHub**, une plateforme e-commerce B2B. Cette mission s'est déroulée du 30 juin au 30 août 2025, sous la supervision du Directeur Général, et avec la Direction Technique.

1.2. Tâches détaillées effectuées

1.2.1. Développement backend (API REST)

Conception et développement de l'API REST :

- Mise en place de l'architecture backend avec Node.js et Express.js
- Création de plus de 20 routes API couvrant tous les aspects de la plateforme
- Implémentation d'un système d'authentification sécurisé avec JWT (JSON Web Tokens) - Développement d'un système de gestion des rôles utilisateurs (acheteur, fournisseur, administrateur)
- Mise en place de la validation des données avec Joi et express-validator
- Implémentation de mécanismes de sécurité (rate limiting, sanitization XSS, protection SQL injection)

Gestion de la base de données :

- Conception du schéma de base de données avec 30+ tables
- Création des relations entre tables avec clés étrangères
- Optimisation des requêtes SQL avec indexes appropriés
- Mise en place d'un système de connection pooling MySQL
- Développement de migrations pour la gestion des versions de base de données

Système de messagerie temps réel :

- Intégration de Socket.IO pour la communication en temps réel
- Développement d'un système de conversations structurées
- Implémentation des indicateurs de présence et de frappe
- Gestion des salles (rooms) pour organiser les conversations

Système de notifications :

- Développement d'un service de notifications multi-canal
- Intégration de l'envoi d'emails avec Nodemailer
- Création d'un système de notifications in-app
- Mise en place de notifications temps réel via Socket.IO

Upload et gestion de fichiers :

- Configuration de Multer pour l'upload de fichiers
- Développement d'un système de gestion des images produits
- Mise en place d'un système de validation et stockage des documents fournisseurs
- Gestion des photos de profil utilisateurs

1.2.2. Développement frontend (Application React)

Architecture et structure :

- Configuration du projet avec Vite et TypeScript
- Mise en place de la structure de dossiers modulaire
- Configuration de Tailwind CSS pour le styling
- Intégration de Shadcn/ui pour les composants UI
- Configuration de React Router pour la navigation

Développement des pages :

- Page d'accueil avec sections hero, produits populaires, catégories, fournisseurs
- Pages d'authentification (connexion, inscription acheteur, inscription fournisseur)
- Pages de produits (catalogue, détail produit, recherche avancée)
- Pages utilisateur (tableaux de bord personnalisés par rôle)
- Pages de messagerie avec interface de chat temps réel
- Pages administratives (gestion utilisateurs, modération, analytics)

Développement des composants réutilisables :

- Composants UI de base (Button, Input, Card, Modal, etc.)
- Composants métier (ProductCard, SupplierCard, ConversationList, etc.)
- Composants de layout (Header, Footer, Sidebar, Navigation)
- Composants de formulaires avec validation
- Composants de notifications et alertes

Gestion d'état et communication API :

- Configuration de React Query pour le cache et la synchronisation
- Développement de hooks personnalisés pour les appels API
- Intégration d'Axios comme client HTTP
- Configuration du service Socket.IO côté client
- Gestion de l'authentification et des tokens JWT

Design responsive :

- Implémentation d'un design mobile-first
- Adaptation de l'interface pour tablettes et desktop
- Optimisation de l'expérience utilisateur sur tous les supports
- Tests de compatibilité sur différents navigateurs

1.2.3. Fonctionnalités métier implémentées

Gestion des produits :

- CRUD complet (création, lecture, mise à jour, suppression)
- Système de catégories hiérarchique
- Recherche et filtres avancés
- Gestion des images multiples
- Système d'offres et promotions

- Gestion des stocks et quantités minimales (MOQ)

Système de messagerie :

- Création de conversations entre acheteurs et fournisseurs
- Envoi et réception de messages en temps réel
- Indicateurs de messages lus/non lus
- Liens avec produits
- Notifications de nouveaux messages

Système de favoris :

- Ajout/suppression de produits favoris
- Ajout/suppression de fournisseurs favoris
- Affichage dans les tableaux de bord

Système d'avis et évaluations :

- Création d'avis sur les produits
- Évaluation des fournisseurs
- Affichage des notes moyennes
- Modération des avis par les administrateurs

Fonctionnalités administratives :

- Gestion des utilisateurs (validation, suspension)
- Modération du contenu (produits, avis)
- Gestion des catégories
- Analytics et statistiques globales

1.2.4. Sécurité et optimisation

Mesures de sécurité implémentées :

- Authentification JWT sécurisée avec expiration des tokens
- Hachage des mots de passe avec bcrypt
- Protection XSS avec DOMPurify
- Protection SQL injection avec prepared statements
- Rate limiting pour éviter les abus
- Validation stricte des données entrantes
- CORS configuré pour les domaines autorisés
- Headers de sécurité HTTP avec Helmet

Optimisations de performance :

- Optimisation des requêtes SQL avec indexes
- Connection pooling MySQL
- Lazy loading des composants React
- Code splitting pour réduire la taille des bundles
- Mise en cache avec React Query
- Optimisation des images

1.2.5. Documentation et tests

Documentation technique : Documentation de l'API (endpoints, paramètres, réponses) ,Documentation de l'architecture frontend ,Commentaires dans le code ,Guide de déploiement ,Guide d'utilisation pour les développeurs

Tests : Tests manuels de toutes les fonctionnalités, Tests de compatibilité navigateurs , Tests responsive sur différents appareils , Tests de performance , Validation des cas d'erreur

1.3. Technologies et outils utilisés

Backend :

- Node.js, Express.js, MySQL2
- Socket.IO, JWT, bcryptjs
- Multer, Nodemailer
- Joi, express-validator
- Helmet, CORS, express-rate-limit

Frontend :

React, TypeScript, Vite, Tailwind CSS, Shadcn/ui, Radix UI , React Query, React Router , Axios, Socket.IO Client , React Hook Form, Zod

Outils de développement : Git et GitHub pour le versioning , VS Code comme éditeur , Postman pour tester l'API , MySQL Workbench pour la gestion de la base de données

2. Méthodologie

2.1. Méthodologie générale de développement

Afin de mener à bien le développement du projet **Gabon Market Hub**, j'ai adopté une **méthodologie de travail structurée**, inspirée des **principes Agile**, et adaptée aux contraintes techniques, humaines et temporelles du stage. Cette approche itérative a permis d'assurer un suivi régulier de l'avancement du projet, une intégration progressive des fonctionnalités et une réactivité face aux imprévus techniques.

2.1.1. Phase 1 : Analyse et conception (Semaines 1 à 2)

La première étape a consisté à analyser en profondeur les besoins fonctionnels et techniques du projet.

- **Analyse des besoins :** identification des utilisateurs cibles (acheteurs, fournisseurs, administrateurs), définition des cas d'usage, et étude comparative des plateformes e-commerce B2B existantes afin d'en extraire les meilleures pratiques.
- **Conception de l'architecture :** choix de la stack technologique (React, Node.js, MySQL), élaboration du schéma de base de données, définition de l'architecture en couches et structuration des modules.
- **Cahier des charges technique :** définition des fonctionnalités prioritaires, estimation des délais de développement, évaluation des risques techniques et établissement d'un plan de production réaliste.

2.1.2. Phase 2 : Développement backend (Semaines 3 à 6)

Cette phase a marqué le début du développement concret de l'application côté serveur.

- **Mise en place de l'infrastructure :** configuration de l'environnement Node.js, initialisation de la base de données MySQL, structuration du projet et configuration des outils de développement.
- **Développement modulaire :** création des principaux modules du backend, notamment :
 - Authentification sécurisée (inscription, connexion, gestion des rôles, JWT)
 - Gestion des utilisateurs, produits, commandes et catégories

- Système de messagerie en temps réel (Socket.IO)
- Notifications et téléversement de fichiers
- **Approche itérative** : chaque fonctionnalité a été développée, testée, puis validée avant de passer à la suivante, garantissant une progression stable et un code fiable.

2.1.3. Phase 3 : Développement frontend (Semaines 7 à 9)

Une fois le backend opérationnel, le développement du **frontend** a pu débuter avec **React** et **TypeScript**.

- **Configuration initiale** : mise en place du projet avec Vite, intégration de Tailwind CSS et du framework Shadcn/ui, configuration de React Router et de React Query pour la gestion de la navigation et des données.
- **Développement des interfaces** : création des pages publiques (accueil, catalogue, recherche), des interfaces d'authentification, des tableaux de bord personnalisés, des espaces administratifs et du système de messagerie.
- **Intégration avec l'API** : liaison complète avec le backend via Axios, gestion des erreurs et des états de chargement, et intégration de la messagerie en temps réel via Socket.IO.

2.1.4. Phase 4 : Intégration et tests (Semaine 10)

À cette étape, toutes les fonctionnalités développées ont été intégrées et testées dans un environnement complet.

- **Intégration complète** : vérification de la communication entre le frontend et le backend, tests fonctionnels et de sécurité, et évaluation des performances globales.
- **Optimisations** : correction des bugs détectés, optimisation des requêtes SQL, amélioration du rendu React et affinement de l'expérience utilisateur.

2.1.5. Phase 5 : Déploiement et documentation (Semaines 11 à 12)

La dernière phase a consisté à rendre le projet opérationnel et à le documenter pour assurer sa maintenabilité.

- **Déploiement** : configuration des environnements de production, hébergement du backend sur un serveur cloud, déploiement du frontend, gestion des variables d'environnement et tests en conditions réelles.
- **Documentation** : rédaction de la documentation technique (API, architecture, guide d'installation) et de la documentation utilisateur pour faciliter la prise en main de la plateforme.

2.2. Méthodologie de travail quotidienne

2.2.1. Organisation du travail

Mon organisation quotidienne s'appuyait sur un planning rigoureux permettant un suivi clair de mes tâches :

- Réunion matinale avec les encadrants lorsque nécessaire ;
- Planification des objectifs de la journée ;
- Développement et test des fonctionnalités prévues ;
- Documentation systématique des modifications effectuées ;
- Gestion du code source via GitHub, avec commits réguliers et messages explicites ;
- Utilisation de branches dédiées à chaque fonctionnalité et intégration via pull requests après validation.

2.2.2. Bonnes pratiques de développement

J'ai veillé à appliquer rigoureusement les bonnes pratiques du développement logiciel moderne :

- **Standards de code** : respect des conventions de nommage, code clair et commenté, architecture modulaire et réutilisable.
- **Sécurité** : validation systématique des données entrantes, protection contre les injections SQL et XSS, authentification sécurisée et mise en place d'un mécanisme de limitation de requêtes.
- **Performance** : optimisation des requêtes SQL, mise en cache lorsque nécessaire, chargement différé des composants (lazy loading) et optimisation des médias.

2.3. Gestion des problèmes et difficultés

Durant le développement, plusieurs difficultés techniques ont été rencontrées, notamment liées à la configuration du serveur, à la gestion des dépendances et à la synchronisation en temps réel. Pour y faire face, j'ai adopté une démarche structurée :

- **Identification** : journalisation des erreurs rencontrées et recherche de leurs causes profondes ;
- **Résolution** : tests de solutions simples, recours à la documentation officielle, consultation des encadrants en cas de blocage et expérimentation de nouvelles approches lorsque nécessaire.

Cette rigueur m'a permis de développer des réflexes d'autonomie et d'analyse critique face aux problèmes techniques.

2.4. Communication et collaboration

La réussite du projet repose en grande partie sur une communication fluide et une collaboration efficace :

- **Avec les encadrants** : échanges réguliers sur l'état d'avancement, présentation des fonctionnalités développées et validation des choix techniques ;
- **Avec l'équipe technique** : entraide entre stagiaires, partage de bonnes pratiques, relecture de code et collaboration sur les modules complexes.

Cette dynamique collective a favorisé un environnement de travail motivant et productif.

2.5. Apprentissage continu

Durant le stage, j'ai adopté une démarche d'apprentissage continu, essentielle dans un domaine technologique en constante évolution.

- **Veille technologique** : lecture quotidienne de la documentation officielle (React, Node.js, MySQL), consultation de forums (Stack Overflow, Dev.to), et analyse de projets open-source similaires.
- **Expérimentation** : tests de nouvelles approches techniques, prototypage rapide pour valider des idées et amélioration continue basée sur les retours obtenus.

Cette démarche proactive m'a permis de renforcer mes compétences, de comprendre en profondeur les technologies utilisées et d'acquérir une véritable autonomie technique.

3. Résultats obtenus

3.1. Résultats techniques

3.1.1. Développement du backend

Le développement du **backend** s'est articulé autour de la création d'une **API REST complète**, conçue pour assurer la communication entre le serveur et le client tout en garantissant sécurité, performance et modularité.

Plus de **vingt routes API** ont été développées, couvrant l'ensemble des fonctionnalités essentielles de la plateforme :

- Authentification et gestion sécurisée des sessions (inscription, connexion, tokens JWT) ;
- Gestion des utilisateurs, profils et rôles (CRUD complet) ;
- Gestion des produits, catégories, commandes et avis ;
- Système de messagerie et de notifications en temps réel ;
- Fonctions administratives et de supervision.

La **base de données**, construite avec **MySQL**, comprend plus de **cinquante tables** interconnectées par des **relations bien définies** et des **contraintes d'intégrité référentielle**. Le schéma a été **normalisé** afin d'éviter la redondance des données et optimisé grâce à l'ajout d'index, garantissant ainsi des temps de réponse rapides même en cas de forte charge.

Sur le plan de la **sécurité**, plusieurs mécanismes ont été mis en place :

- Authentification basée sur **JWT**,
- Hachage des mots de passe avec **bcrypt**,
- Protection contre les **injections SQL** grâce aux requêtes préparées,
- Sécurisation contre les attaques **XSS** via **DOMPurify**,
- **Rate limiting** pour limiter les abus d'accès à l'API,
- Validation stricte de toutes les entrées utilisateurs.

Enfin, un **système de messagerie temps réel** a été développé à l'aide de **Socket.IO**, permettant la communication instantanée entre utilisateurs. Ce module inclut la gestion des conversations, les indicateurs de présence et de frappe, ainsi que l'envoi de **notifications push en temps réel**.

3.1.2. Développement du frontend

Côté client, l'application a été entièrement développée avec **React** et **TypeScript**, selon une architecture modulaire et réutilisable.

Elle comprend plus de **quarante pages fonctionnelles**, parmi lesquelles :

- Une **page d'accueil** présentant les principales sections du site,
- Des pages d'**authentification** (connexion et inscription pour acheteurs et fournisseurs),
- Des pages **produits** (catalogue, recherche, détails),
- Des **tableaux de bord personnalisés** selon le rôle de l'utilisateur,
- Une interface de **messagerie interactive**,
- Des **espaces administratifs** et de gestion de profils.

L'application intègre plus de **cent composants réutilisables**, classés en trois catégories :

- **Composants UI génériques** (boutons, cartes, modales, champs de saisie, etc.) ;
- **Composants métiers** (fiches produits, fournisseurs, conversations, etc.) ;
- **Composants de layout** (header, footer, navigation).

Le design, développé avec **Tailwind CSS** et **Shadcn/UI**, est **entièrement responsive**, assurant une expérience fluide sur ordinateurs, tablettes et smartphones. Des tests de compatibilité ont été effectués sur plusieurs navigateurs afin de garantir une accessibilité optimale.

L'intégration du frontend avec l'API s'appuie sur **Axios** et **React Query**, permettant un **cache intelligent**, une gestion fluide des états de chargement et une **optimisation des performances** lors des échanges de données.

De plus, la **messagerie en temps réel** a été connectée à l'aide de **Socket.IO**, pour offrir un retour instantané aux utilisateurs.

3.2. Résultats fonctionnels

3.2.1. Fonctionnalités implémentées

L'ensemble des fonctionnalités prévues dans le cahier des charges a été **développé, testé et validé**. Les principales réalisations fonctionnelles sont :

- **Gestion des utilisateurs** : Inscription multi-rôles (acheteur, fournisseur), authentification sécurisée via JWT, gestion des profils et documents fournisseurs, rôles et permissions hiérarchisés.
- **Gestion des produits** : CRUD complet, catégories hiérarchiques, filtres et recherche avancée, gestion d'images multiples, gestion des stocks et promotions.
- **Gestion des commandes** : Panier fonctionnel, processus de commande complet, gestion des adresses, suivi des statuts et historique des transactions.
- **Messagerie et notifications** : Conversations et messages en temps réel, indicateurs de lecture, notifications in-app et email, préférences personnalisables.
- **Fonctionnalités avancées** : Système de favoris, avis et évaluations, recherche intelligente, tableaux de bord dynamiques et interface d'administration complète.

3.3. Résultats quantitatifs

Le projet **Gabon Market Hub** se distingue par son ampleur technique et son niveau d'aboutissement. Voici quelques chiffres clés représentatifs du travail accompli :

Élément	Quantité estimée
Lignes de code backend	~15 000
Lignes de code frontend	~25 000
Total	≈ 40 000 lignes
Fichiers backend	80+
Fichiers frontend	150+
Fichiers de documentation	10+
Tables de base de données	50+
Routes API	20+

Élément	Quantité estimée
Pages frontend	40+
Composants React	100+

Ces chiffres traduisent la complexité du projet et la rigueur mise en œuvre pour garantir sa robustesse et sa maintenabilité.

3.4. Résultats qualitatifs

Au-delà de la quantité de travail, les résultats qualitatifs témoignent d'une réelle **maîtrise technique** et d'une **attention portée à la qualité logicielle**.

- **Qualité du code** : Code clair, modulaire, bien documenté et conforme aux bonnes pratiques du développement web moderne.
- **Sécurité** : Aucune faille identifiée lors des tests, validation stricte des entrées, et mise en place de protections contre les principales attaques (SQL, XSS, CSRF).
- **Performance** : Optimisation du temps de chargement, requêtes SQL rapides, interface fluide et réactive.
- **Expérience utilisateur** : Navigation intuitive, design moderne et responsive, retours visuels instantanés et accessibilité sur tous les supports.

3.5. Apprentissages et compétences développées

Ce projet m'a permis d'acquérir et de consolider un **ensemble complet de compétences techniques, méthodologiques et professionnelles**.

Compétences techniques :

- Maîtrise avancée de **React**, **TypeScript**, **Node.js** et **Express.js** ;
- Bonne compréhension de la **modélisation de bases de données MySQL** et des requêtes complexes ;
- Utilisation efficace de **Socket.IO** pour la communication temps réel ;
- Familiarisation avec les outils modernes tels que **Vite**, **Tailwind CSS** et **React Query**.

Compétences méthodologiques :

- Application d'une méthodologie de développement agile et itérative ;
- Planification et organisation du travail selon les priorités ;
- Capacité à tester, valider et documenter chaque étape du développement.

Compétences professionnelles :

- Travail en équipe et communication efficace avec les encadrants ;
- Autonomie et résolution de problèmes techniques ;
- Rédaction de documentation claire et structurée ;
- Sens de la qualité et du respect des délais.

4. Difficultés rencontrées

4.1. Difficultés techniques

4.1.1. Gestion de la complexité de la base de données

Problème

La base de données comprenait plus d'une cinquantaine de tables interreliées. La gestion de ces relations, notamment lors de l'écriture de requêtes **JOIN** complexes, a constitué un défi majeur, rendant parfois le débogage et l'optimisation difficiles.

Solution :

- Élaboration d'un **schéma de base de données documenté** pour clarifier la structure ;
- Utilisation de **diagrammes entité-association (ER)** afin de visualiser les relations entre tables ;
- Optimisation des requêtes via la création d'**index adaptés** ;
- Mise en place de **vues SQL** pour simplifier certaines requêtes répétitives et complexes.

4.1.2. Synchronisation en temps réel avec Socket.IO

Problème

La mise en œuvre d'un système de messagerie en temps réel fiable s'est révélée complexe, notamment dans la gestion :

- des connexions et déconnexions multiples,
- de la synchronisation des messages entre plusieurs clients,
- de la gestion des salles (**rooms**) de conversation,
- et de la performance globale lors de nombreuses connexions simultanées.

Solution :

- Mise en place d'un **système de gestion d'état** dédié à Socket.IO ;
- Organisation des conversations via des **rooms** ;
- Déploiement d'un mécanisme de **heartbeat** pour détecter automatiquement les déconnexions ;
- Optimisation des **événements émis** pour réduire la charge serveur et améliorer la réactivité.

4.1.3. Optimisation des performances frontend

Problème

Avec plus de quarante pages et une centaine de composants React, l'application risquait de subir des lenteurs (temps de chargement initial, re-renders inutiles, taille excessive des bundles).

Solution :

- Utilisation du **lazy loading** via `React.lazy()` pour charger les pages à la demande ;
- **Code splitting** avec **Vite** afin de réduire la taille du bundle principal ;
- Limitation des re-renders inutiles grâce à `React.memo()` ;
- Gestion du cache et des états de chargement avec **React Query** ;
- **Optimisation des images** (compression, formats modernes, lazy loading visuel).

4.1.4. Gestion de l'authentification multi-rôles

Problème :

L'implémentation d'un système d'authentification multi-rôles (acheteur, fournisseur, administrateur) avec des droits différenciés s'est avérée complexe, tant au niveau du backend que du frontend.

Solution :

- Création d'un **middleware d'authentification réutilisable** ;
- Mise en place d'un **système de rôles hiérarchisé** (ex. : 1 = acheteur, 2 = fournisseur, 3 = admin) ;
- **Protection des routes** selon le rôle utilisateur ;
- Gestion des **permissions et restrictions** directement dans la logique applicative et les composants React.

4.2. Difficultés organisationnelles

4.2.1. Gestion du temps

Problème :

Le stage, d'une durée de deux mois, imposait un délai restreint pour développer une plateforme complète. Il a donc fallu hiérarchiser les fonctionnalités pour assurer la livraison d'un produit fonctionnel dans les temps.

Solution :

- Identification des **fonctionnalités critiques (must-have)** ;
- Adoption d'une approche **itérative et incrémentale** ;
- Priorisation des modules essentiels avant les fonctionnalités secondaires ;
- Report des développements non prioritaires à des versions ultérieures.

4.2.2. Apprentissage de nouvelles technologies

Problème :

Certaines technologies utilisées, comme **Socket.IO**, **React Query** ou encore **TypeScript avancé**, étaient nouvelles pour moi et nécessitaient un apprentissage rapide.

Solution :

- Lecture approfondie de la **documentation officielle** ;
- Consultation de **tutoriels et guides techniques** ;
- Réalisation de **tests expérimentaux** pour valider la compréhension des concepts ;
- Sollicitation de l'aide des encadrants en cas de blocage.

4.3. Difficultés conceptuelles

4.3.1. Modélisation des données métier

Problème :

La modélisation d'un système e-commerce B2B intégrant des entités complexes (produits, commandes, messages, notifications, utilisateurs multi-rôles) a exigé une réflexion approfondie pour assurer la cohérence et la scalabilité du modèle.

Solution :

- Analyse de **modèles de bases de données e-commerce** existants ;
- Échanges réguliers avec les encadrants pour affiner la conception ;

- Multiples **itérations de refactorisation** du schéma de données ;
- Normalisation du modèle pour limiter la redondance et améliorer la maintenance.

4.4. Leçons apprises

Ces différentes difficultés ont permis de renforcer plusieurs compétences clés :

- L'importance de la **planification et de la priorisation** des tâches ;
- La nécessité de **tester fréquemment** au cours du développement ;
- L'utilité de **documenter chaque étape** pour mieux comprendre les choix passés ;
- La valeur du **travail collaboratif** et de la recherche d'aide en cas de blocage ;
- L'efficacité d'une approche **progressive et itérative** dans le développement logiciel.

5. Étude de cas : Implémentation du système de messagerie temps réel

5.1. Contexte

Le système de messagerie représente une composante stratégique de la plateforme **Gabon Market Hub**. Il permet aux acheteurs et fournisseurs d'échanger directement sur la plateforme, facilitant ainsi les **négociations commerciales** et le **suivi des commandes**.

5.2. Objectifs

L'objectif de cette fonctionnalité était de concevoir une messagerie qui permette :

- La **communication instantanée** entre utilisateurs ;
- La **création de conversations** liées à des produits ou des commandes ;
- L'affichage des **indicateurs de présence** et des **statuts de lecture** ;
- L'**envoi de notifications** en temps réel pour chaque nouveau message.

5.3. Analyse technique

Choix technologique :

Le choix de **Socket.IO** s'est imposé pour plusieurs raisons :

- API simple et bien documentée,
- Gestion automatique de la reconnexion,
- Prise en charge native des **rooms** pour segmenter les conversations,
- Compatibilité complète avec **Node.js** et **React**.

Architecture du système :

- **Backend** : Serveur Socket.IO intégré au serveur Express existant ;
- **Frontend** : Client Socket.IO intégré dans React ;
- **Base de données** : Tables `conversations` et `messages` pour stocker l'historique ;
- **Stockage** : Persistance des messages en base pour consultation ultérieure.

5.4. Implémentation

5.4.1. Côté backend

Création d'un serveur Socket.IO intégré à Express :

```
// socket/simpleSocketServer.js
const { Server } = require('socket.io');
```

```

class SimpleSocketServer {
  constructor(server) {
    this.io = new Server(server, {
      cors: {
        origin: process.env.FRONTEND_URL,
        credentials: true
      }
    });
    this.setupEvents();
  }

  setupEvents() {
    this.io.on('connection', (socket) => {
      // Connexion de l'utilisateur
      socket.on('join_room', (roomId) => {
        socket.join(roomId);
      });

      socket.on('send_message', async (data) => {
        // Sauvegarde en base
        // Envoi du message à la room correspondante
        this.io.to(data.conversationId).emit('new_message', message);
      });
    });
  }
}

```

Routes API associées :

- GET /api/messages/conversations : liste des conversations de l'utilisateur ;
- POST /api/messages/conversations : création d'une nouvelle conversation ;
- GET /api/messages/conversations/:id/messages : récupération des messages ;
- POST /api/messages/conversations/:id/messages : envoi d'un message.

5.4.2. Côté frontend

Service de gestion de Socket.IO :

```

// services/socketService.ts
import { io, Socket } from 'socket.io-client';

class SocketService {
  private socket: Socket | null = null;

  connect(token: string) {
    this.socket = io(API_BASE_URL, { auth: { token } });
  }

  joinRoom(roomId: string) {
    this.socket?.emit('join_room', roomId);
  }

  sendMessage(message: any) {
    this.socket?.emit('send_message', message);
  }
}

```

Interface utilisateur :

- Liste des conversations avec indicateurs de nouveaux messages ;
- Zone de chat affichant les messages en temps réel ;
- Indicateur de frappe et de présence ;
- Zone de saisie pour l'envoi de messages.

5.4.3. Base de données

Deux tables principales :

- **conversations** : (id, acheteur_id, fournisseur_id, produit_id, sujet)
- **messages** : (id, conversation_id, expéditeur_id, contenu, lu, created_at)

5.5. Résultats

Fonctionnalités réalisées :

- Communication en temps réel fluide et fiable ;
- Création automatique de conversations liées à un produit ;
- Historique complet des messages en base ;
- Indicateurs de lecture et de présence ;
- Notifications instantanées ;
- Interface de chat moderne et responsive.

Performances :

- Latence moyenne : < **100 ms** ;
- Gestion de plusieurs conversations simultanées sans perte de performance ;
- Reconnexion automatique en cas de déconnexion.

Expérience utilisateur :

- Interface intuitive et agréable ;
- Retour visuel instantané sur les actions ;
- Indicateurs de présence clairs et cohérents.

5.6. Enseignements tirés

Cette implémentation m'a permis de :

- Comprendre le fonctionnement des **WebSockets** et de **Socket.IO** ;
- Gérer l'état temps réel au sein d'une application **React** ;
- Synchroniser efficacement les données entre **API REST** et **Socket.IO** ;
- Anticiper les problèmes liés à la latence et à la gestion des connexions multiples.

5.7. Améliorations possibles

- Ajout de **notifications push** pour les messages reçus hors ligne ;
- Intégration d'un **moteur de recherche** dans les conversations ;
- Ajout de **pièces jointes** (images, fichiers, documents) ;
- Support des **messages vocaux** ;
- Implémentation d'un **module de visioconférence**.

CONCLUSION

REFERENCES BIBLIOGRAPHIQUES

ANNEXES

SOMMAIRE

Dédicace

Remerciements

Résumé / Abstract

INTRODUCTION

PARTIE I: PRÉSENTATION DE L'ENTREPRISE D'ACCUEIL

1. Historique de l'entreprise
2. Organigramme et organisation d'Alissa IA
3. Activités principales d'Alissa IA
4. Missions et rôle de l'entreprise dans le secteur
5. Service concerné par le stage

PARTIE II: CORPS DU RAPPORT

1. Présentation du Projet : GabonMarketHub
2. Mission confiée
3. Méthodologie
4. Résultats obtenus
5. Difficultés rencontrées
6. Étude de cas : Implémentation du système de messagerie temps réel
7. Bilan du stage

Conclusion

Bibliographie / Références

Annexes

Liste des figures

Liste des tableaux

Glossaire

