

# RAPPORT TECHNIQUE

## . Présentation du besoin

Nextmux a besoin d'un Mini ERP afin de structurer et centraliser la gestion de ses activités opérationnelles et financières.

Actuellement, la gestion des clients, des projets, des factures et des dépenses est dispersée, ce qui entraîne :

- Un manque de visibilité globale sur l'activité
- Des risques d'erreurs dans les calculs financiers
- Une difficulté à suivre les paiements
- Une absence d'indicateurs clairs pour mesurer la rentabilité

L'entreprise souhaite donc disposer d'un outil unique permettant :

- De regrouper toutes les informations au même endroit
- D'automatiser le suivi financier
- De mieux piloter les projets
- D'améliorer la prise de décision

Ce projet répond à un besoin stratégique : permettre à Nextmux de gérer son activité de manière plus professionnelle, structurée et rentable.

## . Choix techniques

Le Mini ERP de Nextmux a été développé en utilisant une architecture 3-tiers afin de séparer clairement l'interface utilisateur, la logique métier et la gestion des données.

Le frontend est implémenté en **React**, permettant la création d'interfaces dynamiques grâce à une architecture basée sur les composants et une gestion efficace du rendu.

Le backend est développé avec **Laravel**, un framework PHP structuré selon le modèle MVC, offrant une gestion optimisée des opérations CRUD, des règles métier et de l'interaction avec

La base de données **MySQL**. La **base de données MySQL** permet une gestion relationnelle fiable des entités (clients, projets, factures, paiements, dépenses) grâce aux clés étrangères et aux contraintes d'intégrité.

## . Architecture

Le Mini ERP de Nextmux repose sur une architecture en 3 couches (architecture 3-tiers) afin d'organiser correctement l'application et de séparer les responsabilités.

La première couche correspond au **Frontend**. Elle permet d'afficher les pages et de gérer les interactions avec l'utilisateur.

La deuxième couche correspond au **Backend**. Elle contient la logique métier : traitement des formulaires, validation des données, calculs financiers et communication avec la base de données.

La troisième couche est la **base de données MySQL**, qui assure le stockage des informations (clients, projets, tâches, factures, paiements et dépenses) et garantit l'intégrité des relations entre les données.

Cette organisation permet d'avoir une application structurée, claire, maintenable et évolutive.

## . Difficultés rencontrées

La principale difficulté a été la **formalisation précise des besoins**, notamment concernant le calcul de la rentabilité des projets et la gestion des paiements partiels. Certains ajustements ont été nécessaires en cours de développement afin d'adapter la solution aux exigences réelles de Nextmux.

La **modélisation de la base de données MySQL** a également représenté un point critique. Il était essentiel de structurer correctement les relations entre clients, projets, factures, paiements et dépenses tout en garantissant l'intégrité des données.

Enfin, l'**implémentation des calculs financiers dans le backend Laravel** et la synchronisation avec le frontend React ont nécessité plusieurs phases de test afin d'assurer la fiabilité des résultats et la cohérence des informations affichées.

## . Perspectives d'amélioration

L'amélioration la plus importante serait la mise en place d'un **tableau de bord avec des indicateurs clés** (chiffre d'affaires, dépenses, rentabilité, factures impayées) afin de renforcer l'aide à la décision.

Il serait également pertinent d'ajouter un **système de gestion des rôles et permissions** pour sécuriser l'accès aux données.

Enfin, l'automatisation de certaines fonctionnalités, comme la **génération de factures en PDF et les rappels de paiement**, permettrait d'optimiser davantage la gestion financière.