

# AIRNEIS



ASI 3 DEV A

JUIN 2024



Erdal KARAER | ErdalKARAER : [karaer.erd@gmail.com](mailto:karaer.erd@gmail.com)

Thibault SENE | thiba93 : [thibaultsene4@gmail.com](mailto:thibaultsene4@gmail.com)

Romain VIGNARD | Romain78780 : [romain78780@gmail.com](mailto:romain78780@gmail.com)

Jeremy GRAILLOT | JGRAILLOT  
: [jeremy.grailot@supdevinci-edu.fr](mailto:jeremy.grailot@supdevinci-edu.fr)



# Airneis

Github



<https://github.com/ErdalKARAER/airneis>



# AIRNEIS

HOME & DECO

## Contact Us



+33 6 52 02 61 23



123 Anywhere Street, Any City



reallygreatsite.com

# Remerciements

Avant tout, nous tenons à exprimer notre profonde gratitude à toutes les personnes qui ont contribué à la réussite de ce projet.

Tout d'abord, nous remercions chaleureusement Thibault SENE pour son engagement constant et ses idées innovantes qui ont grandement enrichi notre travail.

Un grand merci à Romain VIGNARD pour son soutien indéfectible et ses contributions techniques qui ont été essentielles à l'avancement du projet.

Nous sommes également reconnaissants envers Jeremy GRILLOT.

Enfin, nous adressons nos sincères remerciements à Avétis KAZARIAN, qui a toujours su répondre à nos questions avec patience et compétence, nous guidant avec précision à chaque étape du projet.

Votre collaboration et votre soutien ont été inestimables et nous vous en sommes profondément reconnaissants.

## Introduction

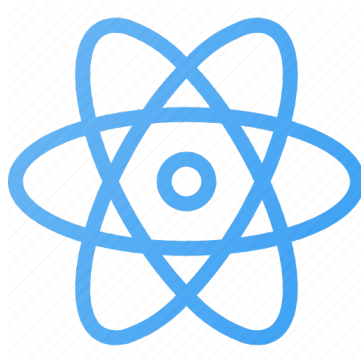
### Contexte

La réalisation de ce projet s'est appuyée sur une collaboration étroite entre les équipes de développement web et mobile, les experts en UX/UI, et les spécialistes du marketing digital. Chaque étape, de la conception initiale à la phase de déploiement, a été marquée par un engagement à l'excellence et à l'innovation. Des tests rigoureux et des itérations basées sur les retours d'utilisateurs ont permis de peaufiner chaque détail, garantissant ainsi une expérience d'achat en ligne de premier ordre.

Le contexte de ce projet est enraciné dans la volonté de l'entreprise d'étendre son rayonnement au-delà des frontières, en capturant l'attention des clients modernes qui privilégient les achats en ligne. Grâce à cette transformation numérique, l'entreprise Airneis se positionne comme un leader dans le secteur de la vente de meubles, prêt à offrir une expérience client enrichie et adaptée aux exigences du commerce électronique contemporain.

# Choix des technologies

## Front-end



En choisissant ReactJS pour le développement de notre projet web, nous avons opté pour une technologie moderne, performante et fiable qui nous permet de créer une plateforme e-commerce réactive, maintenable et évolutive. Les avantages offerts par ReactJS, tels que l'optimisation des performances, la modularité des composants, et la richesse de l'écosystème, nous assurent de pouvoir offrir une expérience utilisateur de qualité supérieure, tout en facilitant la maintenance et l'évolution future de notre application.



En choisissant Tailwind CSS pour le développement des styles de notre projet web, nous avons opté pour une approche moderne et efficace qui nous permet de construire des interfaces cohérentes, maintenables et performantes. Les avantages offerts par Tailwind CSS, tels que la productivité accrue, la cohérence des styles, la personnalisation facile et les performances optimisées, nous assurent de pouvoir offrir une expérience utilisateur de qualité supérieure tout en simplifiant la maintenance et l'évolution future de notre application.



# FORMIK

**En choisissant FormikJS pour la gestion des formulaires de notre projet web, nous avons opté pour une solution robuste, flexible et adaptée aux besoins modernes des applications React. Les avantages offerts par FormikJS, tels que la gestion simplifiée des formulaires, la validation puissante, l'amélioration de l'expérience utilisateur et la modularité, nous assurent de pouvoir créer des formulaires performants, maintenables et faciles à développer. Cette approche nous permet de répondre efficacement aux exigences de notre plateforme e-commerce, en offrant une expérience utilisateur de qualité supérieure tout en facilitant la maintenance et l'évolution future de notre application.**

## Back-end



**En choisissant Node.js pour le développement du backend de notre projet web, nous avons opté pour une solution performante, évolutive et adaptée aux besoins modernes des applications e-commerce. Les avantages offerts par Node.js, tels que son modèle non-bloquant, son environnement JavaScript unifié, sa rapidité de développement, sa gestion efficace des données en temps réel et son large support communautaire, nous assurent de pouvoir construire une plateforme robuste et réactive. Cette approche nous permet de répondre efficacement aux exigences de notre plateforme e-commerce, en offrant une expérience utilisateur de qualité supérieure tout en facilitant la maintenance et l'évolution future de notre application.**



**En choisissant Stripe pour le traitement des paiements de notre projet web, nous avons opté pour une solution de paiement sécurisée, flexible et riche en fonctionnalités. Les avantages offerts par Stripe, tels que la facilité d'intégration, la sécurité et la conformité, le support de multiples méthodes de paiement, les fonctionnalités avancées, les rapports détaillés et l'évolutivité, nous assurent de pouvoir offrir une expérience de paiement fluide et fiable à nos clients. Cette approche nous permet de répondre efficacement aux exigences de notre plateforme e-commerce, en améliorant la satisfaction des clients et en facilitant la gestion des transactions financières.**

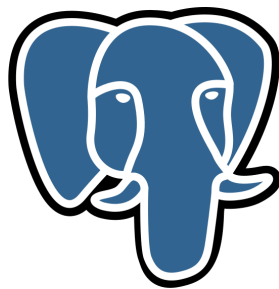


En choisissant Axios pour les requêtes HTTP de notre projet web, nous avons opté pour une solution flexible, puissante et facile à utiliser. Les avantages offerts par Axios, tels que la simplicité d'utilisation, la compatibilité avec les environnements modernes, les fonctionnalités avancées, la gestion simplifiée des données, la sécurité et le support de la pagination et des requêtes multiples, nous assurent de pouvoir construire une application web performante et sécurisée. Cette approche nous permet de répondre efficacement aux exigences de notre plateforme e-commerce, en améliorant l'expérience utilisateur et en facilitant la gestion des communications avec les API backend.

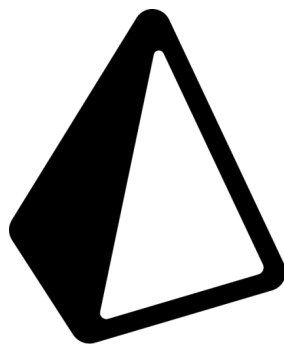


En choisissant Next.js pour le développement de notre application web, nous avons opté pour une solution moderne et performante qui combine les avantages du rendu côté serveur et de la génération de sites statiques. Les avantages offerts par Next.js, tels que l'amélioration des performances et du SEO, l'expérience de développement améliorée, le routage intégré, les fonctionnalités avancées et l'écosystème riche, nous assurent de pouvoir construire une plateforme e-commerce robuste, évolutive et facile à maintenir. Cette approche nous permet de répondre efficacement aux exigences de notre projet, en offrant une expérience utilisateur de qualité supérieure et en facilitant la gestion et l'évolution future de notre application.

## Base de données



En choisissant PostgreSQL pour la gestion de notre base de données, nous avons opté pour une solution fiable, performante et riche en fonctionnalités. Les avantages offerts par PostgreSQL, tels que sa conformité ACID, ses fonctionnalités avancées, ses capacités de performance et de scalabilité, sa sécurité robuste, son statut open source, et son support communautaire actif, nous assurent de pouvoir gérer efficacement les données de notre plateforme e-commerce. Cette approche nous permet de répondre efficacement aux exigences de notre projet, en offrant une expérience utilisateur de qualité supérieure et en facilitant la gestion et l'évolution future de notre application.



En choisissant Prisma pour l'interaction avec notre base de données, nous avons opté pour une solution moderne, performante et facile à utiliser. Les avantages offerts par Prisma, tels que l'API intuitive, la sécurité renforcée, les migrations automatisées, les requêtes optimisées, le support multi-base de données, et la documentation complète, nous assurent de pouvoir gérer efficacement les opérations de base de données de notre plateforme e-commerce. Cette approche nous permet de répondre efficacement aux exigences de notre projet, en offrant une expérience utilisateur de qualité supérieure et en facilitant la gestion et l'évolution future de notre application.

## Objection.js

Nous avons initialement choisi Knex.js pour le SQL Query Builder et Objection.js comme ORM pour notre projet. Cependant, nous avons remarqué que Prisma offre une solution intégrée combinant à la fois les fonctionnalités d'un ORM et d'un SQL Query Builder.





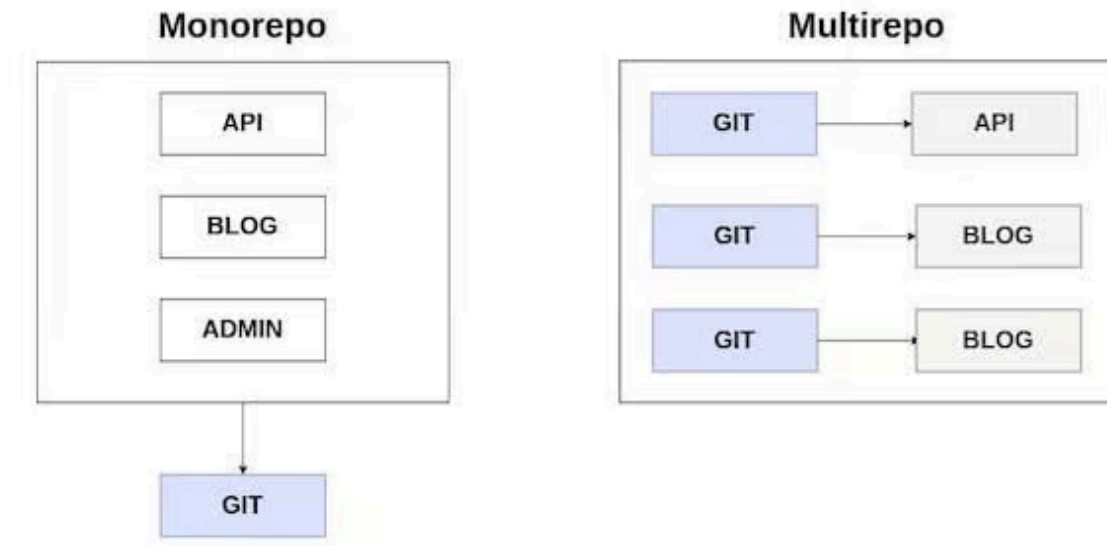
**Avec Prisma, nous bénéficions d'une API unifiée et intuitive pour les opérations CRUD, ce qui simplifie le développement et réduit la complexité du code. De plus, Prisma génère automatiquement des types TypeScript basés sur le schéma de la base de données, offrant ainsi une sécurité des types et une complétion de code améliorée, ce qui n'est pas natif avec Knex.js et Objection.js.**

**En termes de performances, Prisma optimise les requêtes SQL générées, assurant une meilleure gestion des ressources et une exécution plus rapide des requêtes complexes. Les outils de migration de Prisma facilitent également les changements de schéma de manière sécurisée et automatisée, ce qui est plus complexe à gérer manuellement avec Knex.js.**

**Enfin, Prisma intègre des protections automatiques contre les injections SQL et une gestion d'erreurs robuste, renforçant la sécurité et la résilience de notre application. Ces avantages techniques ont conduit à notre décision de choisir Prisma pour sa simplicité, sa légèreté et ses fonctionnalités avancées intégrées**

# Choix de l'architecture

## Next.js Monorepo Architecture

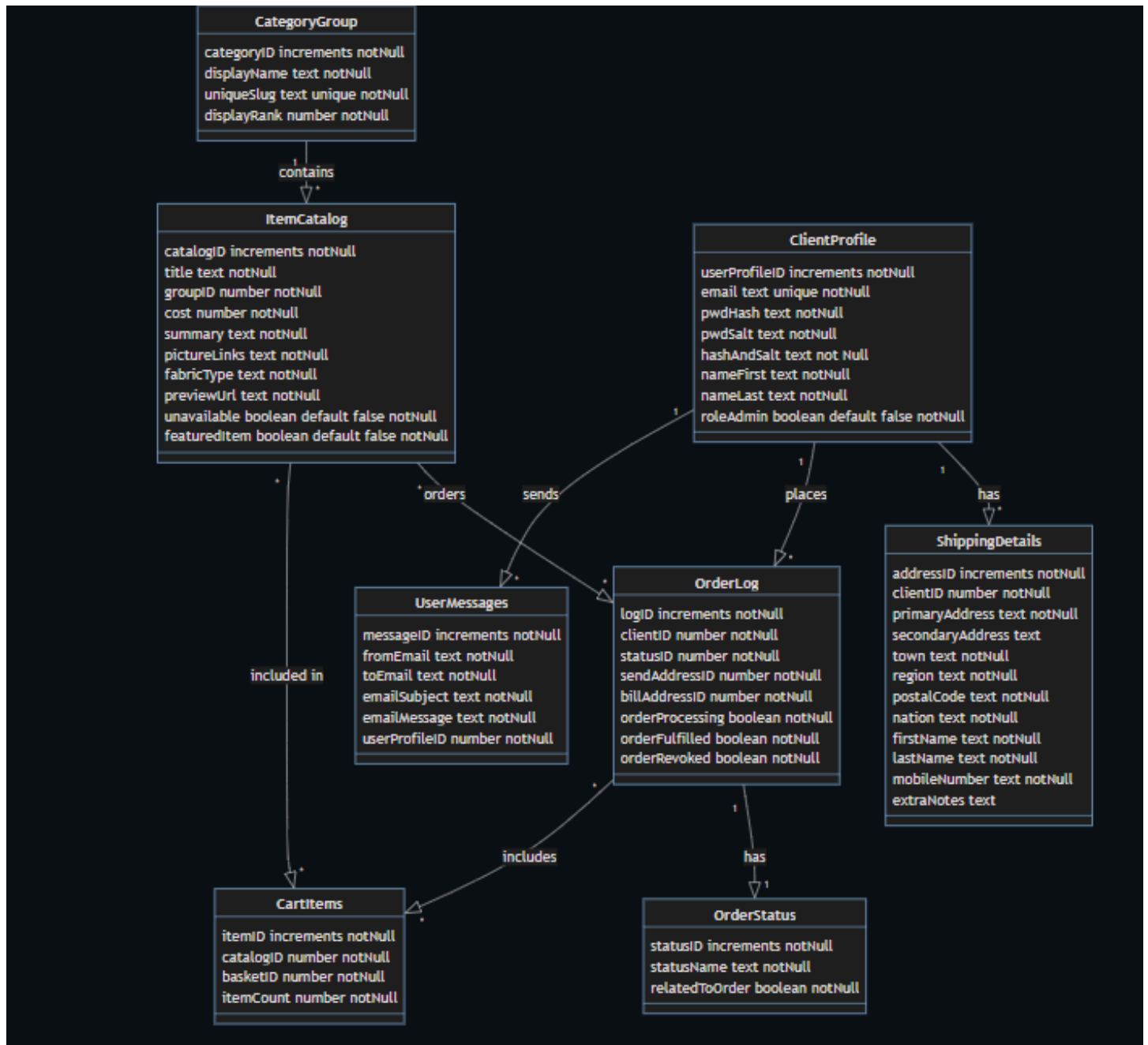


L'architecture choisie pour notre projet web est centrée sur la modularité, la maintenabilité et l'efficacité. Chaque décision de structuration et de choix technologique vise à simplifier le développement, à améliorer les performances et à garantir une évolutivité à long terme. Cette approche nous permet de répondre efficacement aux exigences de notre projet, en offrant une expérience utilisateur de qualité supérieure tout en facilitant la gestion et l'évolution future de notre application.

En choisissant un monorepo pour notre projet, nous avons opté pour une architecture qui favorise la collaboration, la cohérence et la gestion efficace des dépendances et des versions. Les avantages du monorepo, tels que la centralisation des dépendances, la réutilisation du code, et l'optimisation des performances de développement, nous assurent de pouvoir développer, tester et déployer notre application de manière plus fluide et coordonnée. Cette approche nous permet de répondre efficacement aux exigences de notre plateforme e-commerce, en offrant une expérience utilisateur de qualité supérieure tout en facilitant la maintenance et l'évolution future de notre application.

# Diagramme

## Modèle Conceptuel de Données (MCD)



# Difficultés rencontrées

## Début trop rapide sans réflexion à long terme

Au début de notre projet, nous avons pris l'initiative de démarrer rapidement le développement sans une réflexion approfondie sur les besoins à long terme et les implications technologiques. Cette précipitation a conduit à plusieurs difficultés :

- **Choix technologiques précipités :** Nous avons initialement opté pour des technologies qui semblaient convenir aux besoins immédiats sans évaluer leur capacité à évoluer avec le projet. Par exemple, nous avons commencé avec Knex.js et Objection.js pour la gestion de la base de données, mais avons réalisé plus tard que Prisma, avec son intégration d'ORM et de SQL Query Builder, offrait une solution plus simple et légère. Ce changement a nécessité un effort supplémentaire pour migrer le code existant et adapter notre structure de projet.
- **Manque de planification :** L'absence de planification détaillée a entraîné des réévaluations fréquentes des choix technologiques, perturbant ainsi la continuité du développement et augmentant le temps nécessaire pour atteindre nos objectifs.

## Changements fréquents de technologies

En raison de notre départ précipité, nous avons dû changer plusieurs fois de technologies pour aligner notre stack avec les exigences du projet :

- **Impact sur le développement :** Ces changements fréquents ont perturbé le flux de travail et ont nécessité des périodes d'apprentissage et d'adaptation pour les nouvelles technologies. Chaque migration technologique a impliqué non seulement des ajustements de code, mais aussi des révisions des configurations, des tests, et des processus de déploiement.
- **Réécriture de parties du code :** À chaque changement, certaines parties du code devaient être réécrites ou adaptées, entraînant des retards et une augmentation de la charge de travail.

## Problèmes de collaboration

**La dynamique de l'équipe a également posé des défis :**

- **Participation inégale :** Un de nos collègues n'a pas participé activement au travail, ce qui a mis une pression supplémentaire sur le reste de l'équipe. L'absence de contribution de sa part a nécessité une redistribution des tâches et une réorganisation des responsabilités pour compenser le manque de participation.
- **Coordination et communication :** La collaboration efficace a été entravée par des problèmes de communication et de coordination, exacerbés par les changements technologiques et la nécessité de réévaluer constamment nos approches.