



# Cyril De Graeve

Développeur React / Node.js

XP : 4 ans

contact@degraeve.dev / 06 76 04 54 31

## Synthèse

En tant que développeur freelance, je propose une gamme de services de développement web qui couvrent toutes les étapes du processus de création d'un site, d'un SaaS ou d'une application. Je maîtrise notamment React, Next.js, Node.js et PostgreSQL. Grâce à ma pratique et mes connaissances de l'incroyable écosystème Javascript, je suis en mesure de résoudre des problèmes complexes en créant des solutions fonctionnelles et des interfaces esthétiques modernes adaptées à vos besoins.

## Formation & Certification

**2021** Formation Développeur Fullstack JavaScript (O'Clock)

**2022** Titre Professionnel de Développeur Web et Web Mobile (DWMM)

## Compétences techniques & fonctionnelles

**Langage de programmation** : Javascript, TypeScript, SQL

**Web** : HTML5, CSS3, React, Node.js, Express.js, Next.js, RESTful APIs, Tailwind, Framer-motion, Responsive Design, SEO, Testing

**Databases** : PostgreSQL, Supabase, Prisma, Drizzle, Sequelize

**Generative AI / LLMs** : Intégration LLM (OpenAI), Automatisation de contenus avec IA, Prompt Engineering...

**Devops** : Nginx, Apache, SSL, FTP, PM2

**Cloud Computing** : Docker

**OS** : Linux, Windows WSL

**Source Control** : Git, GitHub

**Méthodologies** : Agile, Scrum

# Expériences professionnelles

## ASSISTANT GEN IA

01/06/25 - 10/07/25 (1 mois 1/2)

Freelance

> <https://assist.cyrildegraeve.dev>

## Développeur Fullstack JavaScript & GenAI

### CONTEXTE :

Dans une logique d'innovation et d'automatisation au service des professionnels du livre (libraires, bibliothécaires, éditeurs), j'ai développé une application web permettant de générer automatiquement du contenu à partir de la 4<sup>e</sup> de couverture d'un ouvrage, en texte ou en image. Objectif : simplifier la création de contenus éditoriaux, accélérer les mises en ligne, améliorer la visibilité des livres tout en respectant les codes de communication du secteur.

### ENVIRONNEMENT TECHNIQUE :

Next.js (App Router), Node.js, React, TypeScript, Tailwind CSS, shadcn/ui, OpenAI API (gpt-3.5-turbo), Tesseract OCR, OpenLibrary Covers API...

The screenshot shows a web application titled "Votre générateur de contenus automatisé". It features a form with the following elements:

- A dropdown menu labeled "> Choisir un type de génération" with the selected option "Fiche produit + SEO + newsletter".
- Two input fields: "Indiquer le nom de l'auteur \*" and "Indiquer le titre du livre \*".
- A section labeled "> Choisir l'une des 2 options" with two radio buttons:
  - Option 1** (selected): "Insérer le texte source". Below it is a large text area labeled "Coller ou recopier ici le texte de la 4<sup>e</sup> de couverture".
  - Option 2**: "Prendre ou importer une photo de la 4<sup>e</sup> de couverture". Below it are two buttons: "Prendre une photo" (with a camera icon) and "Importer une photo" (with a download icon).
- A footnote: "\* Champs obligatoires".
- A large purple button at the bottom labeled "Générer les textes attendus".

## **Développement fullstack avec Next.js :**

- > Stack unifiée front + back + API avec Next.js App Router, sans serveur externe.
- > Utilisation des Route Handlers (*/api/generate* et */api/cover*) pour les appels GPT et la récupération des couvertures de livres (via ISBN).
- > Traitement conditionnel du contenu transmis : texte inséré ou image analysée par OCR (Tesseract) sur le back.
- > Génération des contenus via OpenAI en fonction du mode sélectionné, avec prompts spécialisés et parsing structuré de la réponse (fiche, meta, newsletter, etc.).

## **Front-end :**

- > Interface utilisateur responsive, fluide et accessible.
- > Utilisation de React, TypeScript et Tailwind CSS v4 pour une stylisation rapide, moderne et maintenable.
- > Sélecteur de mode de génération (fiche, critique, traduction) via shadcn/ui + composants personnalisés.
- > Formulaire intelligent avec options : insertion de texte ou import de photo (mobile ou desktop).
- > Intégration d'une capture webcam en desktop (via navigator.mediaDevices) avec traitement et aperçu en direct.
- > Affichage conditionnel des résultats, avec composants réutilisables et système de copie dans le presse-papiers.

## **OCR & traitement image :**

- > Conversion d'image en texte via Tesseract OCR (*node-tesseract-ocr*).
- > Compression et redimensionnement des images avant traitement (résolution max 1500px) pour améliorer la précision.
- > Nettoyage du texte OCR : suppression des artefacts de saut de ligne, normalisation des espaces, suppression des coupures de mots.
- > OCR déclenché uniquement si l'utilisateur fournit une image (photo ou capture).

## **Recherche de couvertures :**

- > Endpoint */api/cover* dédié, utilisant l'API OpenLibrary Covers.
- > Recherche d'une jaquette à partir d'un ISBN saisi manuellement.
- > Affichage dynamique de la jaquette si disponible (avec lien de téléchargement HQ).

## **Prompts & logique métier (OpenAI) :**

- > Trois prompts spécialisés selon le mode de génération choisi :
  - Fiche produit + SEO + newsletter : ton sobre, sans accroches publicitaires, structuré pour e-commerce.
  - Critique littéraire : ton subjectif et littéraire, 700 caractères max, pour blog ou newsletter.
  - Traduction : prompt sec, sans mise en forme ni commentaire, pour un rendu fidèle et naturel.
- > Parsing précis de la réponse GPT pour extraire les blocs de contenu générés.

## **Déploiement & tests :**

- > Projet testé sur différents types de contenu et 4<sup>es</sup> de couverture réelles (copiées ou photographiées).
- > Prêt à être déployé sur un environnement VPS avec support Node.js.
- > Design responsive, performances fluides, temps de génération très courts (OpenAI).
- > Pensé pour un usage quotidien dans des environnements éditoriaux, e-commerce ou bibliothécaires.

L'outil permet à un professionnel du livre de générer des contenus clés en quelques secondes à partir d'une seule source, sans rédaction manuelle. Il automatise un travail éditorial souvent chronophage, avec des résultats directement utilisables.

Freelance

> <https://www.pierre-saunier.fr>**Développeur Fullstack JavaScript & DevOps****CONTEXTE :**

Pour la Librairie Pierre Saunier, spécialisée dans les ouvrages rares et de bibliophilie, j'ai réalisé une refonte complète de son site, historiquement développé en PHP/MySQL. Objectif : moderniser l'interface sans en trahir l'esprit, optimiser les performances, préparer une boutique en ligne, et offrir une interface de gestion simple et fiable, le tout en exploitant une architecture fullstack unifiée sous Next.js App Router.

**ENVIRONNEMENT TECHNIQUE :**

Next.js (App Router), Node.js, React, TypeScript, Tailwind CSS, MySQL, Drizzle, Sharp, Stripe...



## **Développement fullstack avec Next.js :**

- > Stack unifiée (front + back + API + admin) grâce à Next.js App Router.
- > Rendu des pages publiques en Server-Side Rendering (SSR) pour améliorer les performances et le SEO, Server Actions et Route Handlers (app/api/) pour gérer la logique serveur et les interactions avec la base de données, le tout directement dans l'architecture Next.js App Router, sans backend externe.
- > Drizzle ORM pour gérer les requêtes SQL en TypeScript de manière typée et sécurisée.

## **Front-end :**

- > Refonte graphique fidèle à l'univers du site d'origine : design responsive, épuré et élégant.
- > Développement des pages publiques :
  - Page "Splash" et page d'accueil avec mise en avant des sélections.
  - Fiches livres avec galeries d'images adaptée.
  - Création des pages "Vitrine", "Jadis et naguère", "Panier", "Commande" et "Paiement".
- > Moteur de recherche avancée multicritères (auteur, année, éditeur, lieu, mots-clés...).
- > Ajout d'une fonctionnalité d'autocomplétion et de suggestions dynamiques :
  - Composant React côté front (affichage immédiat des résultats et navigation au clavier).
  - API Next.js (Route Handler) interrogeant la base de données via Drizzle ORM pour retourner les suggestions pertinentes en temps réel.
- > Navigation UX repensée : menu fixe, composants, footer, carrousels.

## **Back-office personnalisé :**

- > Développement d'un Back-office élaboré, personnalisé et optimisé 100 % en Next.js.
  - Gestion autonome des ouvrages, de leur mise en vitrine et du stock par le client.
  - Interface claire et épurée adaptée à ses besoins réels.
- > Script d'optimisation des images pour un rendu optimal avec Sharp / Uploading automatique avec Busboy.
- > Scripts d'export des contenus en .docx et .pdf calibrés l'édition d'un catalogue papier.
- > Sécurisation de l'accès à l'administration avec NextAuth.js et Bcrypt.

## **Base de données :**

- > Migration de l'ancienne base MySQL (MyISAM) vers une structure propre en InnoDB, avec ajout d'index, des Primary et Foreign keys adéquates.
- > Ajouts de nouveaux champs dans la BDD.
- > Refactoring du schéma Drizzle ORM et des relations pour correspondre aux nouvelles fonctionnalités.
- > Écritures et utilisation de requêtes typées et évolutives.
- > Élaboration des migrations envoyées à la base de données ("drizzle-kit generate" + "drizzle-kit push").

## **Fonctionnalités e-commerce :**

- > Développement d'un panier utilisateur avec possibilité de retrait d'un ouvrage en librairie.
- > Intégration complète et testée de Stripe Element pour les paiements CB, Google Pay, Apple Pay.

## **Déploiement et DevOps :**

- > Déploiement sur VPS Linux avec NGINX, PM2, certificats SSL via Let's Encrypt.
- > Configuration serveur optimisée pour les performances et la sécurité.
- > Opération de migration du DNS de l'ancien site vers le nouveau.

Le client peut désormais administrer ses ouvrages depuis **une interface multifonctionnelle claire et moderne**, avec des temps de chargement optimisés et une visibilité accrue sur les nouveaux contenus.

Freelance

> <https://www.serious-publishing.fr>**Développeur Fullstack JavaScript & DevOps****CONTEXTE :**

Pour la maison d'édition parisienne Serious Publishing, j'ai été chargé de la refonte complète de leur site web, passant d'une stack PHP/MySQL à une architecture moderne basée sur JavaScript et PostgreSQL. L'objectif était d'améliorer les performances, la sécurité, l'expérience utilisateur et d'ajouter de nouvelles fonctionnalités pour accroître la visibilité de l'éditeur, mettre en avant ses activités et augmenter les conversions (achats de livres en ligne).

**ENVIRONNEMENT TECHNIQUE :**

React, Next.js, Node.js, Express, Tailwind, Shadcn, Framer-motion, TypeScript, PostgreSQL, NextAuth, JWT...





## **Front-end :**

- > Technologies utilisées : React, Next.js, TypeScript, Tailwind CSS, Framer Motion.
- > Conception et développement de l'interface utilisateur :
  - Mise en place de Next.js pour le rendu côté serveur (SSR) et l'optimisation des performances.
  - Utilisation de TypeScript pour une meilleure maintenabilité et la réduction des erreurs.
  - Intégration de Tailwind CSS pour un design réactif et modulable.
  - Ajout d'animations avec Framer Motion pour améliorer l'expérience utilisateur.
- > Optimisation SEO et conformité RGPD :
  - Implémentation des meilleures pratiques SEO : balises méta, sitemap dynamique, URLs optimisées.
  - Mise en conformité avec le RGPD.
- > Création d'un Back-office personnalisé et adapté aux besoins de gestion des contenus du client.

## **Back-end :**

- > Technologies utilisées : Node.js, Express.js, PostgreSQL, NextAuth.js, JWT.
- > Développement de l'API RESTful :
  - Création d'une API REST avec Express.js pour gérer les opérations CRUD (pour le Back-office).
  - Structuration du code selon le modèle MVC pour une meilleure organisation.
- > Gestion de l'authentification et de la sécurité :
  - Implémentation de NextAuth.js pour l'authentification utilisateur (sessions sécurisées).
  - Utilisation de JWT pour la gestion des tokens d'accès et de rafraîchissement.
- > Base de données :
  - Migration vers PostgreSQL pour profiter de ses fonctionnalités avancées.
  - Conception du schéma de la base de données, optimisation des requêtes et utilisation d'index pour améliorer les performances.
- > Sécurisation du Back-office.

## **Fonctionnalités e-commerce :**

- > Boutique en ligne complète :
  - Développement du système de panier utilisateur avec persistance des données.
  - Calcul dynamique des frais de port en fonction du poids, de la destination et des tarifs postaux en vigueur.
  - Gestion des produits avec possibilité d'ajout/update/suppression via le Back-office.
- > Intégration des paiements sécurisés :
  - Intégration de l'API de PayPal pour accepter les paiements via PayPal ou CB.

## **Responsive Design :**

- > Conception mobile-first pour assurer une expérience optimale sur les smartphones.
- > Utilisation et optimisation de Tailwind pour une adaptabilité sur tous les écrans.
- > Tests sur différents navigateurs et appareils pour garantir la compatibilité cross-platform.

## **Déploiement et DevOps :**

- > Configuration du serveur :
  - Mise en place d'un serveur VPS sous Linux (Ubuntu).
  - Installation et configuration de Nginx comme proxy inverse pour servir l'application Next.js et gérer les certificats SSL.
  - Utilisation de PM2 pour le déploiement et la gestion des processus Node.js.
- > Sécurité et performance :
  - Configuration des pare-feu et des règles de sécurité (UFW).
  - Installation de certificats SSL/TLS via Let's Encrypt pour sécuriser les communications.
  - Optimisation des réglages de Nginx pour améliorer les temps de réponse et gérer le caching.

Freelance

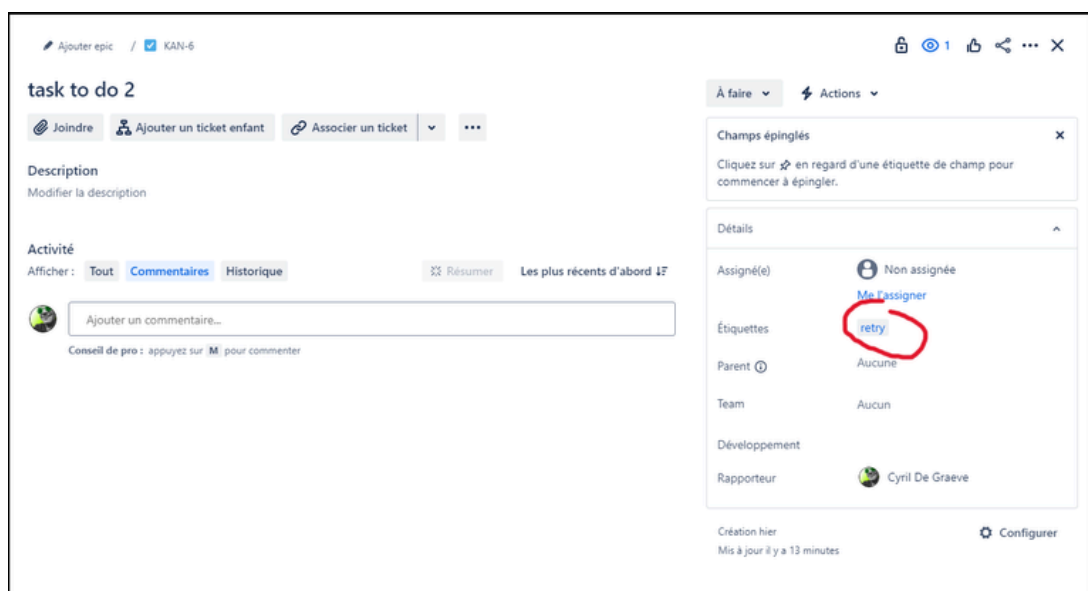
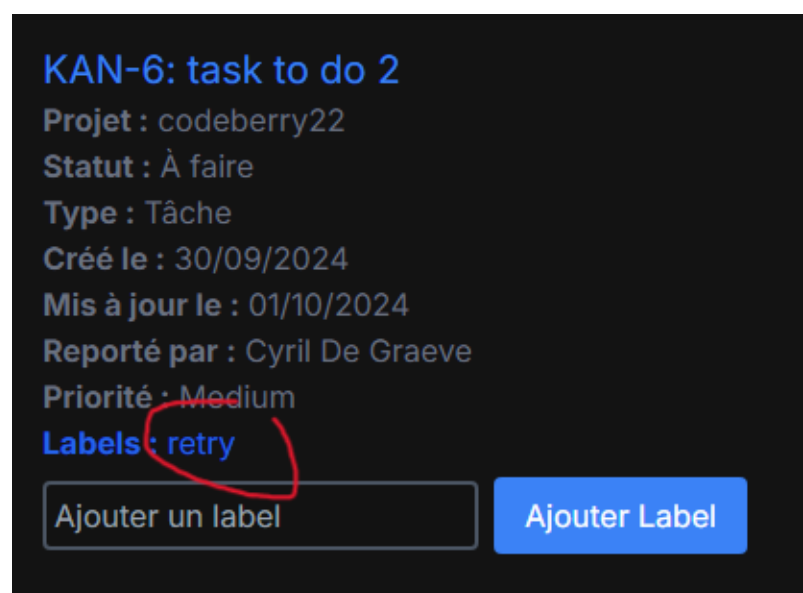
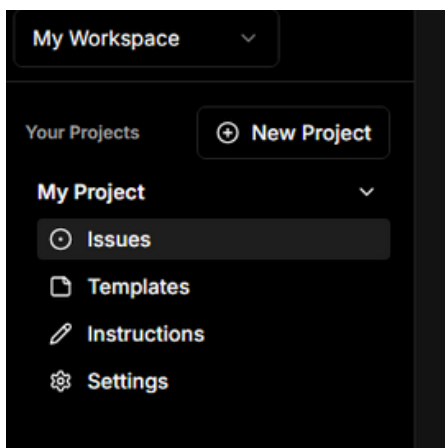
## Développeur Fullstack JavaScript

### CONTEXTE :

Pour une startup en phase de financement, j'ai participé au développement du MVP (Minimum Viable Product) d'un SaaS destiné à générer automatiquement des Pull Requests à partir de tickets Jira et/ou Linear. Mon rôle principal était d'intégrer Jira dans l'application pour récupérer certains tickets et automatiser leur traitement, ce qui facilite le flux de travail des équipes de développement et améliore l'efficacité du processus de gestion de code.

### ENVIRONNEMENT TECHNIQUE :

React, Next.js, TypeScript, Tailwind, Shadcn, Framer-motion, Supabase, Drizzle, Server Actions, Clerk, Vercel...





## **Installation et gestion de l'environnement :**

- > Mise en place de l'environnement de développement avec Next.js pour bénéficier des dernières fonctionnalités.
- > Configuration du déploiement continu sur Vercel pour un hébergement rapide et fiable.
- > Gestion de l'authentification utilisateur avec Clerk pour sécuriser l'accès à l'application.

## **Intégration de Jira dans l'application :**

- > Base de données :
  - Utilisation de Supabase pour la gestion des données.
  - Création de la table 'jira\_issues' pour stocker les informations des tickets Jira.
  - Réalisation des migrations de schéma avec Drizzle ORM pour faciliter les modifications et maintenir la cohérence de la base de données.
- > Back-end :
  - Développement d'une API avec Next.js API Routes pour interagir avec l'API de Jira.
  - Récupération des tickets Jira via l'API REST de Jira, avec gestion de l'authentification et des permissions.
  - Création de fonctions pour automatiser la génération de Pull Requests basées sur les tickets récupérés.
- > Front-end :
  - Affichage des tickets Jira dans l'application en temps réel.
  - Utilisation de Server Actions de Next.js pour gérer les actions côté serveur directement depuis les composants React.
  - Mise en place d'une interface utilisateur intuitive avec Tailwind CSS et Shadcn/UI.

## **Ajout d'un champ 'label' dans l'application et les tickets Jira :**

- > Base de données :
  - Ajout d'un champ 'label' dans la table jira\_issues pour associer des étiquettes aux tickets.
  - Gestion des migrations correspondantes dans Supabase.
- > Back-end :
  - Création d'une route API pour ajouter et mettre à jour les labels des tickets Jira depuis l'application.
  - Développement de fonctions pour synchroniser les labels entre l'application et Jira.
- > Front-end :
  - Mise à jour de l'interface utilisateur pour permettre aux utilisateurs de gérer les labels.
  - Utilisation de Framer-Motion pour ajouter des animations fluides lors de l'ajout ou de la modification des labels.

Avec une base solide pour le MVP, j'ai pu aider à la livraison rapide d'un produit fonctionnel **en seulement 5 jours**, permettant à la startup de présenter une démo aux investisseurs potentiels.

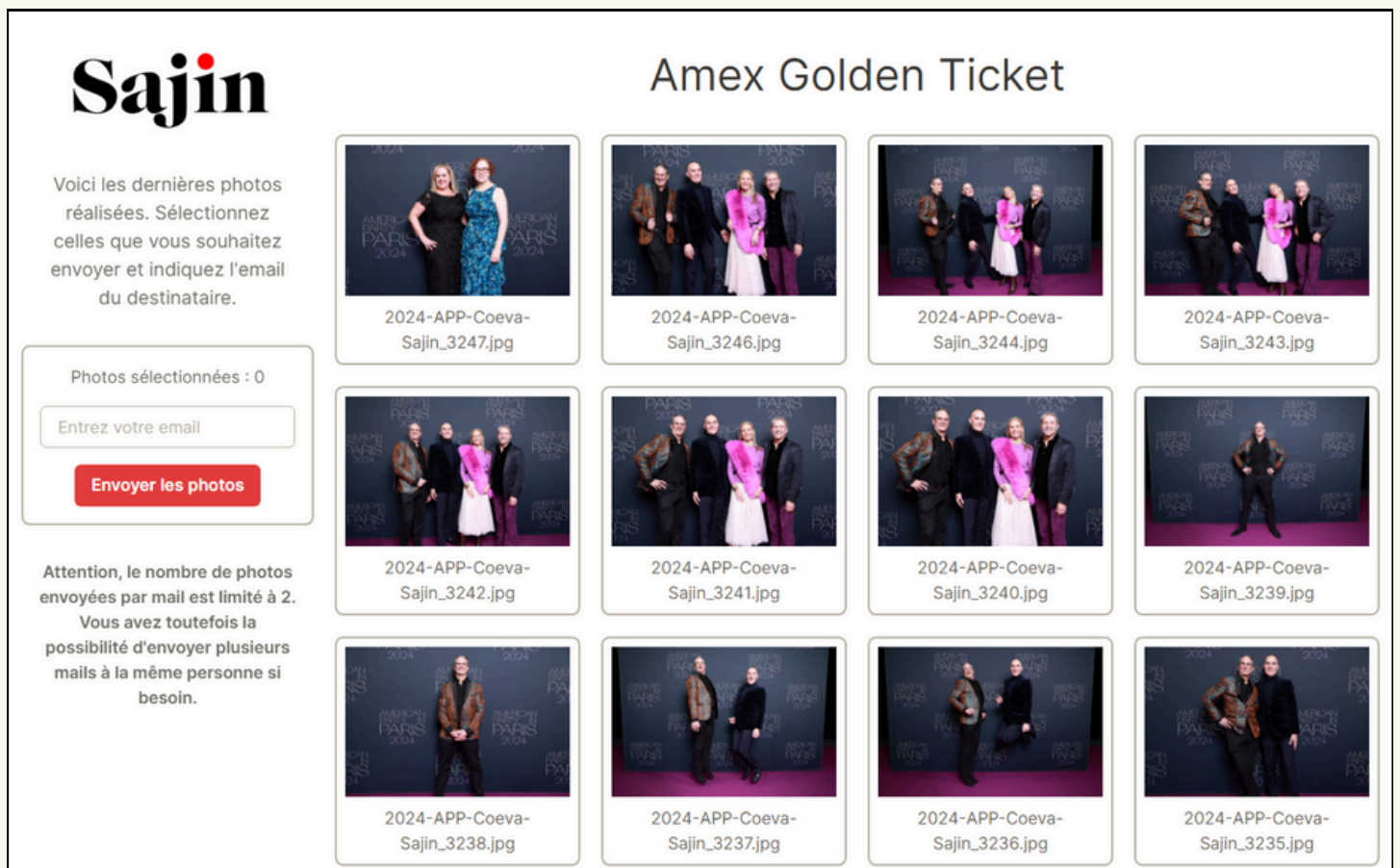
Freelance

**Développeur Fullstack JavaScript & DevOps****CONTEXTE :**

Pour un photographe professionnel souhaitant offrir à ses clients une expérience de partage de photos en temps réel lors d'événements, j'ai conçu et développé une application web et mobile de gestion et de partage de photos événementielles. L'objectif était de permettre au photographe de partager instantanément les photos prises lors d'événements avec les participants, améliorant ainsi l'engagement client et la satisfaction globale.

**ENVIRONNEMENT TECHNIQUE :**

React, Node.js, Express, Next.js, TypeScript, Bcrypt, Chakra UI...



## **Gestion des événements :**

### > Fonctionnalités :

- Création et gestion d'événements personnalisés avec des métadonnées (date, lieu, description).
- Organisation hiérarchique des albums photos par événement pour une navigation simplifiée.

### > Implémentation :

- Conception du modèle de données pour les événements et les albums dans la base de données.
- Utilisation de Next.js pour le routage dynamique et le rendu côté serveur des pages d'événements.

## **Upload et stockage sécurisé :**

### > Fonctionnalités :

- Interface intuitive pour l'upload massif de photos haute résolution par le photographe.
- Stockage sécurisé des images avec contrôle d'accès pour garantir la confidentialité.

### > Implémentation :

- Intégration avec un service de stockage cloud (type AWS S3) pour une scalabilité optimale.
- Mise en place d'un système d'authentification et d'autorisation avec JWT et Bcrypt.
- Gestion des uploads asynchrones avec feedback en temps réel à l'utilisateur.

## **Visualisation intuitive :**

### > Fonctionnalités :

- Galerie photo épurée avec fonctionnalités de tri, filtrage et recherche.
- Affichage en grille avec support du zoom et du mode plein écran.

### > Implémentation :

- Utilisation de Chakra UI pour construire une interface réactive, accessible et responsive.
- Optimisation du chargement des images avec lazy loading et techniques de compression.

## **Envoi de photos personnalisé :**

### > Fonctionnalités :

- Possibilité pour les utilisateurs de sélectionner des photos et de les partager par e-mail.
- Personnalisation des messages avec des templates dynamiques.

### > Implémentation :

- Intégration d'un service d'envoi d'e-mails transactionnels (SendGrid).
- Création d'API sécurisées pour gérer les demandes d'envoi et suivre les statuts.
- Gestion des files d'attente pour l'envoi d'e-mails afin d'assurer la fiabilité.

## **Administration simplifiée :**

### > Fonctionnalités :

- Tableau de bord pour le photographe permettant de gérer les events, les photos et les users.
- Statistiques en temps réel sur l'engagement des utilisateurs (vues, partages, téléchargements).

### > Implémentation :

- Développement d'un back-office avec React et Chakra UI, sécurisé par authentification.
- Gestion des rôles et permissions pour différencier les accès entre administrateur et utilisateurs.
- Utilisation d'API RESTful avec Express.js pour les opérations CRUD.

## **Déploiement et DevOps :**

### > Configuration du serveur :

- Mise en place d'un serveur VPS sous Linux (Ubuntu) pour héberger l'application.
- Installation et configuration de Nginx comme proxy inverse et serveur web.
- Utilisation de PM2 pour la gestion des processus Node.js et le déploiement continu.

### > Sécurité et performance :

- Installation de certificats SSL/TLS via Let's Encrypt pour sécuriser les communications.
- Configuration de pare-feu (UFW) et des règles de sécurité pour protéger le serveur.
- Optimisation des paramètres de Nginx pour améliorer les performances et gérer le caching.

**HYPER-FREE**

31/07/2023 – Aujourd'hui

Freelance

> <https://hyper-free.cyrildegraeve.dev>

**Développeur Fullstack JavaScript & Copywriter**

### **CONTEXTE :**

En tant que développeur Fullstack JavaScript et Copywriter, j'ai créé HYPER-FREE, une plateforme présentant mes offres de services packagées en développement web et en copywriting. Mon objectif est de combiner la puissance du code et l'art des mots pour aider des indépendants, des artistes, des coachs, des photographes, des startup, des PME... à booster leur présence en ligne et à accroître leur réputation. Cette initiative me permet de proposer des solutions complètes qui répondent précisément aux besoins de mes clients, tout en mettant en avant mon expertise unique issue de 20 ans d'expérience en journalisme.

### **ENVIRONNEMENT TECHNIQUE :**

React, Next.js, TypeScript, Panda CSS, Ark UI, Framer-Motion, i18next...



## **New Life**

Tu veux développer ta marque et/ou ton activité sur les réseaux ? Tu cherches à établir une présence professionnelle marquante sur Internet ?

Lance-toi dans le monde digital avec **New Life** ! Avec cette formule, je crée de toutes pièces ton site et ton application web mobile en intégrant à la fois le développement Front et Back-end. Grâce à mes compétences éditoriales, je m'occupe également de l'editing, en rédigeant un contenu captivant et engageant.

Ton branding, ton service et ton business bénéficieront d'une visibilité et d'une présence en ligne optimales, avec une plateforme personnalisée reflétant parfaitement ton image et tes objectifs.



Fais découvrir au monde ton talent, ton service ou business unique grâce à 'New Life', ta porte d'entrée dans le monde digital !

## **Développement de la plateforme web :**

### **> Conception de l'interface utilisateur :**

- Création d'un design moderne et responsive, assurant une expérience utilisateur optimale sur tous les appareils.
- Utilisation de Panda CSS (et Ark UI) pour une mise en page flexible et une personnalisation rapide du style.
- Implémentation de composants réutilisables avec TypeScript pour améliorer la maintenabilité du code.
- Création d'un formulaire de contact permettant au visiteur de préciser au mieux sa demande.
- Internationalisation des contenus avec la library i18next pour gérer les traductions dynamiques (français / anglais).

### **> Optimisation des performances :**

- Mise en place du rendu côté serveur (SSR) avec Next.js pour améliorer le temps de chargement des pages.
- Utilisation de techniques d'optimisation telles que le code splitting et le lazy loading.

## **Copywriting et contenu marketing :**

### **> Création de contenus percutants :**

- Rédaction de textes engageants pour présenter mes services en développement web et en copywriting.
- Mise en avant de mon expérience de journaliste pour établir une connexion avec le public cible.

### **> Stratégie de marketing de niche :**

- Élaboration de messages adaptés aux besoins spécifiques des clients potentiels.
- Optimisation du contenu pour le SEO afin d'améliorer la visibilité en ligne.

## **Offres de services personnalisée :**

### **> Élaboration de quatre formules de services (création from scratch et ou refonte intégrale).**

### **> Présentation claire des offres :**

- Création de composants dédiés pour chaque formule détaillant les avantages et les spécificités.
- Intégration d'appels à l'action (CTA) pour encourager les prises de contact.

## **Optimisation SEO et performance :**

### **> Amélioration de la visibilité en ligne :**

- Utilisation des meilleures pratiques SEO : balises méta, URLs optimisées, sitemap XML.
- Mise en place de contenus structurés avec schema.org pour améliorer le référencement.

### **> Analyse et suivi :**

- Intégration de Google Analytics et Google Search Console pour suivre le trafic et les performances du site.
- Ajustements continus basés sur les données pour améliorer le taux de conversion.

En offrant **une combinaison rare de compétences** en développement web et en copywriting, je réponds aux besoins de clients recherchant une solution complète pour leur présence en ligne.

# ARTIFICIAL LIFE COACH

09/10/2023 – 30/11/2023 (2 mois)

Freelance

> <https://artificial-life-coach.vercel.app>

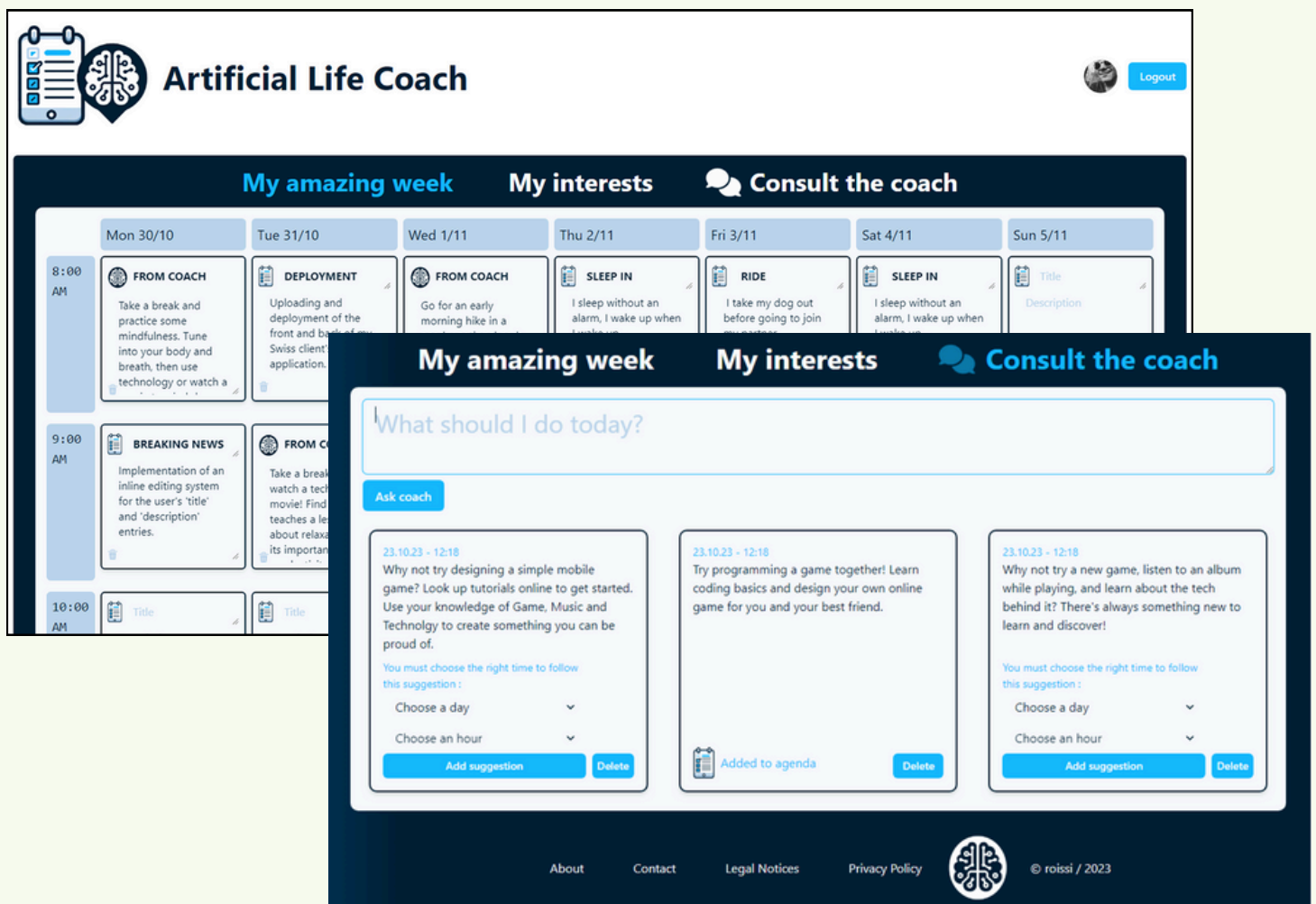
Développeur Fullstack JavaScript

## CONTEXTE :

J'ai conçu de A à Z (Front-end et Back-end) "Artificial Life Coach", une application web et mobile offrant aux utilisateurs un coach de vie artificiel pour gérer leur agenda interactif hebdomadaire ou quotidien. L'objectif était de créer une expérience personnalisée permettant aux utilisateurs de planifier leurs activités en fonction de leurs intérêts et besoins actuels, tout en bénéficiant de conseils sur mesure.

## ENVIRONNEMENT TECHNIQUE :

React, Node.js, Express, Next.js, PostgreSQL, Sequelize, Bcrypt, JWT, Chakra UI, Framer-Motion...





## **Fonctionnalités principales :**

### > Personnalisation des centres d'intérêt et des besoins :

- Les utilisateurs peuvent saisir leurs centres d'intérêt (un ou plusieurs) et définir leurs besoins actuels pour une durée spécifique (en jours).
- Ces informations sont transmises au coach virtuel, qui génère des suggestions personnalisées pour enrichir leur quotidien.

### > Interaction avec le coach virtuel :

- Les utilisateurs peuvent poser des questions au coach, telles que :
  - "Je n'ai pas la forme, donne-moi une idée d'activité adéquate pour l'après-midi."
  - "J'ai un rendez-vous important demain matin, comment puis-je m'y préparer pour être au top ?"
- Le coach fournit des réponses et des conseils adaptés en temps réel grâce à l'intégration de l'API OpenAI.

### > Gestion de l'agenda interactif :

- Possibilité d'ajouter les suggestions du coach directement à l'agenda personnel, à la date et l'heure souhaitées.
- Les utilisateurs peuvent également saisir leurs propres programmes et activités, à la manière de Google Agenda.

## **Intégration de l'IA avec l'API OpenAI :**

- > Utilisation de l'API OpenAI GPT-3.5-turbo-1106 pour générer des réponses intelligentes et contextuelles du coach virtuel.
- > Optimisation des prompts pour obtenir des réponses pertinentes tout en respectant les limites de l'API.

## **Développement Front-end :**

- > Conception d'une interface utilisateur intuitive et responsive, adaptée aux ordinateurs de bureau et aux smartphones.
- > Utilisation de CSS Modules ou Styled Components pour une gestion efficace du style.
- > Gestion de l'état de l'application avec le Context API de React.

## **Développement Back-end :**

- > Création d'une API RESTful pour gérer l'authentification, les données des utilisateurs, les interactions avec l'agenda et les requêtes à l'API OpenAI.
- > Mise en place d'un système d'authentification sécurisé avec JWT pour les tokens et Bcrypt pour le hachage des mots de passe.
- > Gestion de la base de données avec PostgreSQL et Sequelize comme ORM.
- > Gestion des CORS :
  - Configuration des CORS (Cross-Origin Resource Sharing) pour permettre une communication sécurisée entre le Front-end et le Back-end.
  - Mise en place de règles précises pour autoriser uniquement les origines fiables, renforçant ainsi la sécurité de l'application.

## **Sécurité et performance :**

- > Validation et sanitation des entrées utilisateur pour prévenir les injections SQL et les attaques XSS.
- > Implémentation de cache pour les réponses fréquentes afin de réduire les temps de réponse et les coûts liés à l'API OpenAI.
- > Optimisation des requêtes à la base de données et utilisation d'index pour améliorer les performances.

Freelance

> <https://clairios.com>**Développeur Front-end + DevOps****CONTEXTE :**

Pour Clairios, une entreprise de services informatiques spécialisée dans l'infogérance et le conseil, j'ai conçu et développé un site web dynamique et moderne. L'objectif principal était de mettre en avant les services proposés, améliorer la visibilité en ligne de l'entreprise et offrir une expérience utilisateur optimale aux visiteurs, en particulier les PME et TPE recherchant des solutions informatiques fiables.

**ENVIRONNEMENT TECHNIQUE :**

React, Next.js, TypeScript, Tailwind, Framer-motion...



## **Conception de l'interface utilisateur :**

### **> Design moderne et responsive :**

- Création d'une interface utilisateur claire, intuitive et accessible, mettant en valeur les services tels que l'infogérance, la sécurité, le support et les solutions cloud.
- Utilisation de Tailwind CSS pour un design réactif et une personnalisation rapide du style.
- Intégration de Framer-Motion pour ajouter des animations fluides, améliorant l'engagement et l'expérience utilisateur.

### **> Accessibilité :**

- Respect des normes WCAG pour assurer une accessibilité optimale aux utilisateurs en situation de handicap.
- Tests d'accessibilité réalisés pour vérifier la conformité.

## **Développement Front-end :**

### **> Optimisation des performances :**

- Mise en place du rendu côté serveur (SSR) avec Next.js pour améliorer les temps de chargement et le référencement.
- Utilisation du code splitting (multiples composants) et du lazy loading pour optimiser le chargement des ressources.

### **> Formulaire de contact optimisé :**

- Création d'un formulaire de contact convivial avec validation côté client et serveur.
- Implémentation d'une API pour traiter les soumissions et envoyer les données de manière sécurisée.
- Ajout de notifications de confirmation pour améliorer l'expérience utilisateur.

## **Optimisation pour le référencement (SEO) :**

### **> Amélioration de la visibilité en ligne :**

- Incorporation de balises méta personnalisées pour chaque page (title, description, keywords).
- Utilisation de données structurées (schema.org) pour améliorer l'interprétation du contenu par les moteurs de recherche.
- Génération automatique d'un sitemap XML pour faciliter l'indexation.
- Optimisation des URL et du contenu pour cibler les mots-clés pertinents du secteur.
- Optimisation des images avec Next.js Image Component pour une meilleure performance.

## **Déploiement et DevOps :**

### **> Configuration du serveur :**

- Mise en place d'un serveur VPS sous Linux (Ubuntu).
- Installation et configuration d'Apache comme serveur web pour servir l'application.
- Configuration de Node.js pour exécuter l'application Next.js en mode production.
- Utilisation de PM2 pour la gestion des processus Node.js, assurant la stabilité et la récupération automatique en cas de panne.

### **> Sécurité et performance :**

- Installation de certificats SSL/TLS via Let's Encrypt pour sécuriser les communications HTTPS.
- Configuration du pare-feu (UFW) et mise en place de règles de sécurité pour protéger le serveur.
- Optimisation des paramètres d'Apache pour améliorer les performances, notamment en activant la compression Gzip et en configurant la mise en cache.

### **> Automatisation du déploiement :**

- Mise en place de scripts de déploiement pour faciliter les mises à jour continues.
- Intégration avec Git pour le contrôle de version et le déploiement continu.

**Freelance**

> <https://neo-telegraphe.vercel.app>

**Développeur Fullstack JavaScript****CONTEXTE :**

J'ai créé "Télégraphe", une plateforme web dédiée à la découverte des boutiques du quartier Télégraphe, situé dans le 20ème arrondissement de Paris. L'objectif était d'aider les résidents et les visiteurs à trouver facilement des boutiques locales et à en savoir plus sur ce qu'elles offrent. Ce projet vise à renforcer le lien communautaire, soutenir les commerces de proximité et dynamiser le quartier.

**ENVIRONNEMENT TECHNIQUE :**

React, Node.js, Express, Next.js, PostgreSQL, Supabase, Bootstrap...

**La Poste**

28 rue du Télégraphe



★ ★ ★ (3.0)

En savoir plus

**Kelbongoo**

18 rue du Borrégo



★ ★ ★ ★ (4.5)

En savoir plus

**Manakea**

21 rue du Borrégo



★ ★ ★ ★ (4.5)

En savoir plus

**Épicerie Tounette**

12 rue du Borrégo



★ ★ ★ ★ (4.3)

En savoir plus

## **Fonctionnalités principales :**

- > Authentification des utilisateurs :
  - Mise en place d'un système d'inscription et de connexion sécurisé pour les membres de la communauté.
  - Gestion des sessions utilisateurs avec JWT pour sécuriser les communications.
- > Localisation des boutiques :
  - Intégration de l'API Google Maps pour afficher une carte interactive du quartier.
  - Les utilisateurs peuvent localiser les boutiques sur la carte avec des marqueurs personnalisés.
- > Notes et avis :
  - Consultation des notes et avis laissés par les autres membres sur les boutiques.
  - Système de notation par étoiles et commentaires détaillés pour évaluer les commerces.
- > Ajout d'avis personnels :
  - Possibilité pour les utilisateurs de rédiger et d'enregistrer leurs propres avis sur les boutiques visitées.
  - Modération des avis pour assurer la qualité et la pertinence du contenu.
- > Interface responsive :
  - Conception d'une application entièrement responsive, adaptée aux desktops, tablettes et smartphones.
  - Utilisation de Bootstrap pour faciliter le design responsive et assurer une expérience utilisateur cohérente.

## **Développement Front-end :**

- > Conception d'une interface utilisateur intuitive et conviviale, mettant en avant les boutiques et les avis de la communauté.
- > Utilisation de Next.js pour le rendu côté serveur (SSR) améliorant les performances et le SEO.
- > Implémentation de composants réutilisables avec React pour une meilleure maintenabilité du code.

## **Développement Back-end :**

- > Création d'une API RESTful avec Express.js pour gérer les opérations CRUD des boutiques, des avis et des utilisateurs.
- > Utilisation de Supabase (basé sur PostgreSQL) pour la base de données et l'authentification des utilisateurs.
- > Gestion des relations entre les tables (boutiques, utilisateurs, avis) pour maintenir l'intégrité des données.

## **Intégration de l'API Google Maps :**

- > Utilisation de l'API JavaScript de Google Maps pour afficher les cartes interactives.
- > Personnalisation des marqueurs pour représenter les différentes boutiques.
- > Implémentation de fonctionnalités telles que le zoom, le filtrage des boutiques par catégorie, et l'affichage des informations détaillées au clic sur les marqueurs.

**CDD****Développeur Fullstack JavaScript****CONTEXTE :**

Dans le cadre d'un CDD de 4 mois, j'ai participé à l'internationalisation de la plateforme de l'association Giving What We Can pour créer la version française du nouveau site de Don Efficace. Don Efficace est une ONG à but non lucratif dont la mission est de financer des actions visant à réduire la souffrance de tous les êtres sensibles (humains et non-humains) de la manière la plus efficace possible, directement ou indirectement.

Au cours de ce projet, j'ai collaboré étroitement avec des développeurs anglais et suisses-allemands, ce qui a impliqué de nombreuses séances de pair programming et une communication interculturelle enrichissante. J'ai également été responsable de la mise à jour de la partie marketing du site en utilisant Sanity, un CMS open-source headless écrit en JavaScript et conçu pour les développeurs.

**ENVIRONNEMENT TECHNIQUE :**

React, Node.js, Express, Next.js, TypeScript, Sanity, i18n, Jest...





## **Internationalisation du site web :**

- > Mise en place de l'internationalisation (i18n) :
  - Configuration de Next.js pour supporter plusieurs langues, en particulier le français et l'anglais.
  - Utilisation de bibliothèques telles que react-i18next pour gérer les traductions dynamiques (anglais / français).
  - Gestion des fichiers de traduction JSON pour structurer le contenu dans différentes langues.
- > Adaptation du contenu :
  - Traduction du contenu existant en français.
  - Ajustement du formatage et du style pour s'adapter aux particularités linguistiques et culturelles du public francophone.
- > Tests et validation :
  - Mise en place de tests unitaires avec Jest pour vérifier le bon fonctionnement des composants multilingues.
  - Réalisation de tests manuels pour s'assurer de la cohérence et de la qualité des traductions.

## **Collaboration internationale :**

- > Communication efficace :
  - Participation à des réunions régulières avec les équipes anglaises et suisses-allemandes.
  - Utilisation de l'anglais comme langue de travail principale pour faciliter la communication.
- > Pair programming :
  - Sessions de pair programming pour résoudre des problèmes complexes, partager des connaissances et harmoniser les pratiques de développement.
  - Utilisation d'outils de collaboration tels que Visual Studio Live Share ou des plateformes de visioconférence avec partage d'écran.

## **Mise à jour de la partie marketing avec Sanity CMS :**

- > Intégration de Sanity CMS :
  - Configuration de Sanity comme CMS headless pour gérer le contenu dynamique du site.
  - Création de schémas personnalisés pour les différents types de contenu marketing (articles, pages, témoignages).
- > Développement Front-end :
  - Mise en place de pages dynamiques dans Next.js pour afficher le contenu du CMS.
- > Optimisation du contenu :
  - Collaboration avec l'équipe marketing pour s'assurer que le contenu est optimisé pour le public cible.
  - Implémentation des meilleures pratiques SEO pour améliorer la visibilité du site.

## **Améliorations techniques et optimisation :**

- > Performance :
  - Optimisation du temps de chargement des pages grâce au pré-rendu statique (SSG) de Next.js.
  - Utilisation de Lazy Loading pour les images et les composants non critiques.
- > Qualité du code :
  - Adoption de TypeScript pour améliorer la maintenabilité et la robustesse du code.
  - Mise en place de standards de codage avec ESLint et Prettier.
- > Tests :
  - Écriture de tests unitaires et d'intégration avec Jest pour assurer la fiabilité du code.
  - Couverture de tests visant à atteindre un niveau élevé de confiance dans le déploiement en production.

# J'ADOpte UN HUMAIN

15/11/2021 – 20/12/2021 (1 mois)

## PROJET DE FIN DE FORMATION

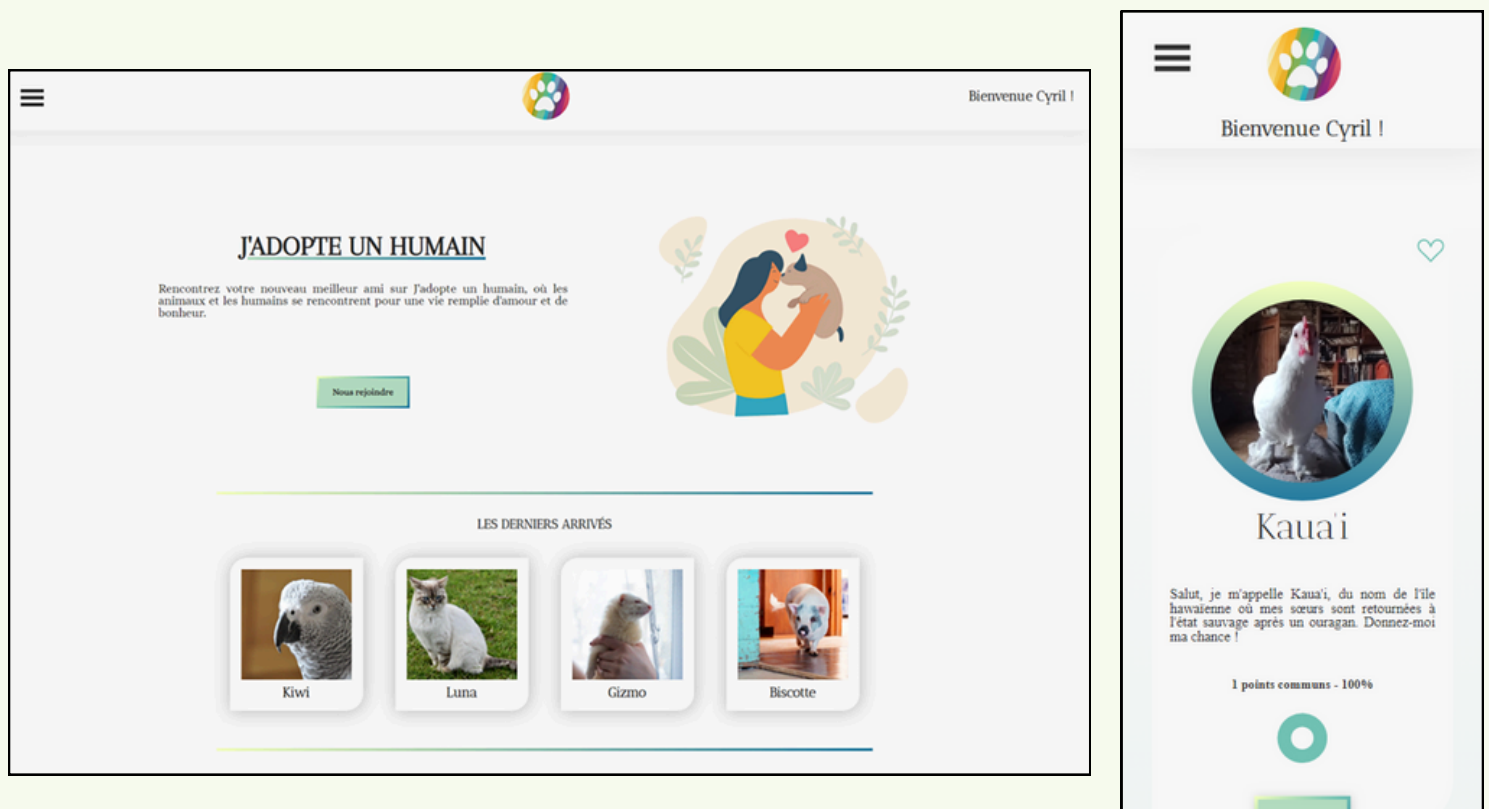
### Product Owner & Développeur Back-end

#### CONTEXTE :

À la fin d'une formation chez O'Clock, j'ai participé en tant que Product Owner et Développeur Back-end à la conception de "J'adopte un humain", une application web responsive pour un refuge fictif d'animaux. L'objectif principal était de permettre aux futurs adoptants de choisir le compagnon idéal en fonction de leurs envies, valeurs et besoins. Ce projet, réalisé en méthode Agile (Scrum) avec une équipe de 4 autres développeurs, visait à moderniser les sites vitrines existants, souvent peu interactifs, et à fournir aux refuges un système de gestion des animaux simplifié et sécurisé.

#### ENVIRONNEMENT TECHNIQUE :

React, Node.js, Express, PostgreSQL, JWT, Bcrypt, Swagger, Jest...



## **Product Ownership :**

### > Gestion de projet Agile :

- Mise en place de la méthodologie Scrum pour organiser le développement sur une période d'un mois.
- Animation des réunions quotidiennes (stand-up meetings), planifications de sprint, revues et rétrospectives.
- Priorisation du backlog produit en fonction des besoins du projet et des ressources disponibles.

### > Recueil des besoins et définition des spécifications :

- Élaboration des user stories avec des critères d'acceptation clairs.
- Collaboration étroite avec l'équipe pour définir les fonctionnalités clés et assurer une vision commune du projet.

## **Développement Back-end :**

### > Conception et développement de l'API RESTful :

- Utilisation de Node.js et Express.js pour créer une API RESTful robuste.
- Structuration du code selon le modèle MVC pour une meilleure organisation et maintenabilité.
- Documentation de l'API avec Swagger pour faciliter l'intégration côté Front-end et la compréhension par l'équipe.

### > Gestion de la base de données :

- Conception du schéma relationnel avec PostgreSQL, modélisant les entités clés : utilisateurs, animaux, profils, etc.
- Implémentation des relations entre les tables pour assurer l'intégrité des données.
- Optimisation des requêtes SQL pour améliorer les performances de l'application.

### > Authentification et sécurité :

- Mise en place de l'authentification avec JSON Web Tokens (JWT) pour sécuriser les endpoints de l'API.
- Hachage des mots de passe utilisateurs avec Bcrypt pour garantir la confidentialité des informations sensibles.
- Gestion des rôles et permissions pour les trois niveaux d'accès :
  - Utilisateur : consultation des animaux, création de profil, demande d'adoption.
  - Staff : gestion des animaux (ajout, modification, suppression), consultation des profils utilisateurs.
  - Administrateur : gestion des utilisateurs, modération, accès complet à toutes les fonctionnalités.

### > Développement du système de matching :

- Création d'un algorithme de matching entre les profils des utilisateurs et des animaux.
- Prise en compte de multiples critères tels que le type d'animal, le comportement, les besoins spécifiques, l'environnement de l'adoptant.
- Retour des résultats sous forme de liste ordonnée et un pourcentage de matching, offrant à l'utilisateur les meilleures correspondances potentielles.

## **Collaboration avec l'équipe Front-end :**

### > Coordination avec les développeurs Front-end travaillant avec React pour l'intégration de l'API.

### > Tests d'intégration pour assurer la cohérence entre le Front-end et le Back-end.

### > Résolution des problèmes techniques et ajustements en fonction des retours de l'équipe.

## **Tests et assurance qualité :**

### > Écriture de tests unitaires et tests d'intégration avec Jest pour vérifier le bon fonctionnement des modules Back-end.

### > Mise en place de l'intégration continue pour automatiser les tests et maintenir la qualité du code.