Rapport de stage
1 décembre 2025 - 25 janvier 2026

Hernani CASTRO DE ALMEIDA

BTS SIO SLAM Deuxième Année

Tutrice de stage : Manuella OSTER

Professeurs principaux : Herve Yves LE GUERN & Samir BELKHIR

REMERCIEMENT

Je tiens à remercier personnellement la société Groupe GEFOR pour leur accueil, leur bienveillance et leur accompagnement tout au long de mon stage.

Je remercie également l'ensemble du personnel enseignant du BTS SIO pour la qualité de leur formation, leur disponibilité, ainsi que leurs précieux conseils qui m'ont permis d'aborder ce stage avec des bases solides et une vision claire des attentes professionnelles.

SOMMAIRE

1. Introduction

Dans le cadre de ma deuxième année de **BTS Services Informatiques aux Organisations** (option SLAM - Solutions Logicielles et Applications Métiers), j'ai effectué un stage de huit semaines en entreprise. Ce stage s'inscrit dans un parcours visant à développer mes compétences en tant que développeur et à consolider les acquis de ma formation en cours.

Mon objectif principal était double :

- consolider mes compétences en **développement fullstack,**
- et renforcer ma capacité à apprendre de manière autonome

Cette approche m'a permis de progresser dans un cadre professionnel tout en découvrant le fonctionnement concret d'une entreprise.

J'ai été accueilli par **Manuella OSTER** au sein de l'entreprise **Groupe GEFOR**, où j'ai effectué mon stage de deuxième année.

Ma mission principale a consisté à concevoir et à développer la présence en ligne de la nouvelle activité audiovisuelle du Groupe GEFOR. Cela a impliqué la création d'un site vitrine responsive ainsi que la mise en place de son système de gestion interne **(back-office)** pour le traitement des contacts. En parallèle, j'ai consacré une part importante de mon temps à renforcer mes compétences en développement web, notamment par l'apprentissage autodidacte d'un **framework front-end**.

Ce rapport présentera tout d'abord l'entreprise d'accueil, puis les missions qui m'ont été confiées. Ensuite, j'expliquerai les démarches mises en œuvre pour les réaliser, avant de conclure par un bilan personnel de cette expérience professionnelle.

2. Présentation de l'entreprise

Le **Groupe GEFOR** est une organisation française spécialisée dans la formation professionnelle pour adultes, dont l'activité principale remonte à **1993** (via le Groupe Européen de Formation) et le nom actuel a été adopté en **2010**.

Il est rattaché à **KHEDER Holding**, créé en **2010**, dont le siège social est situé à **7 rue du Louvre à Paris (75001)**.

Son objectif principal est d'organiser des formations adaptées aux besoins des chefs d'entreprise, des salariés et des demandeurs d'emploi. Le Groupe est un acteur reconnu dans son secteur, certifié **Datadock** et validé conforme par **Pôle Emploi**.

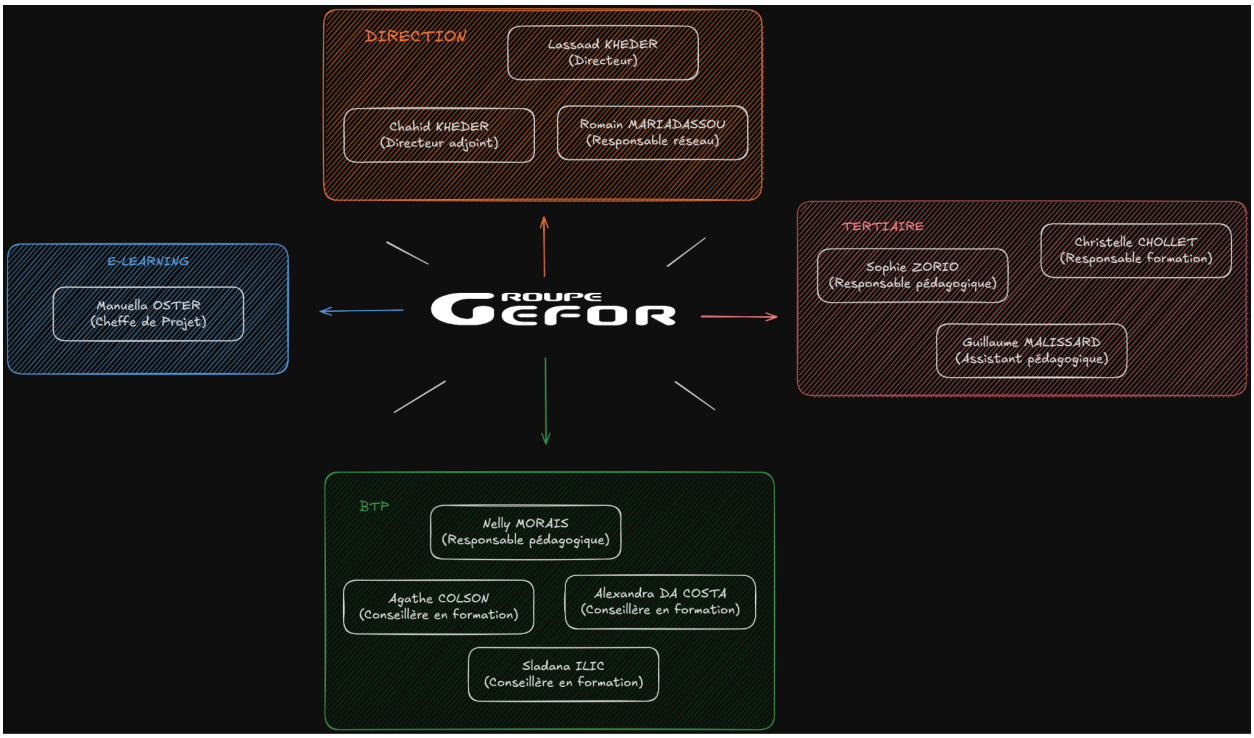
Le Groupe GEFOR opère dans quatre domaines d'activités majeurs :

- **Formation Tertiaire** (BTS en Banque, Assurance, Gestion, Informatique, etc.) : activité exercée à Paris.
- **Formation BTP** (préparation aux CACES R482 et R487) : activité exercée à Changis-sur-Marne (77).
- **Français sur Objectifs Spécifiques (FOS)**.
- **Socle de Compétences en Contexte Professionnel**.

En 2025, le Groupe GEFOR a élargi son offre en développant une nouvelle activité audiovisuelle, suite au rapatriement de la chaîne de télévision Cartage Plus. Cette nouvelle entité propose des services de **Production**, **Location de studios**, **Événementiel** et **Formations** spécifiques, et s'appuie sur deux studios situés à Paris et à Changis-sur-Marne.

Mon stage de deuxième année a été effectué au sein de cette nouvelle activité, sous la tutelle de **Manuella OSTER**, Directrice de Projets au sein du Groupe GEFOR.

Voici un aperçu de l'organisation du Groupe GEFOR par pôles d'activités :



3. Missions confiées

3.1. Introduction

Pour faire suite à l'introduction du rapport de stage, les missions réalisées au cours de mon stage s'articulent autour de deux axes principaux :

- **la création d'un site vitrine pour l'entreprise,**
- **et la réalisation d'un back-office pour le traitement des contacts**

3.2. Création d'un site vitrine (Front-end)

3.2.1 Contexte

Le Groupe GEFOR, après le rapatriement de la chaîne de télévision Cartage Plus en 2025, a lancé une nouvelle activité audiovisuelle. Afin de promouvoir les services de **Production, Location de studios, Événementiel** et **Formations**, l'entreprise avait besoin d'une vitrine numérique professionnelle.

3.2.2 Objectif et problématique

La mission confiée était de concevoir un nouveau site clair, moderne et professionnel, apte à :

- Valoriser les services et les deux studios (Paris et Changis-sur-Marne).
- Renforcer la visibilité en ligne de l'activité.
- Faciliter la prise de contact et la prise de rendez-vous.

3.2.3 Cahier des charges

Le site a été développé sur la base d'un cahier des charges détaillé qui fixait notamment :

- **L'architecture** : 8 pages principales (Accueil, Promotion, Production, Location de studios, Événementiel, Formations, Notre vision, Contact).
- **L'identité visuelle** : Style élégant, minimaliste, avec les couleurs Noir, Blanc et Doré.
- **Les contenus fonctionnels** : Présentation des studios, carrousel d'images pour l'Événementiel, blocs vidéo, et des formulaires de contact et de prise de RDV (avec menu déroulant de sélection du service).
- **Le développement** : Exigence d'un développement fullstack sans CMS et d'un responsive design optimisé (Desktop, Tablette, Mobile

3.2.4 Choix techniques

Le cahier des charges demandait un site au rendu **moderne**, avec une interface lisible et ergonomique, ainsi qu'un responsive design complet (desktop, tablette, mobile).

Afin de répondre à ces exigences tout en gardant une application légère et adaptée à un site vitrine, j'ai fait le choix d'utiliser **Vue.js** côté front-end : Angular me paraissait surdimensionné pour ce type de projet.

Ce choix répondait également à un objectif personnel : apprendre un framework populaire afin de pouvoir le réutiliser dans mes futurs projets, quelle que soit leur taille ou leur complexité.

J'ai retenu Vue.js pour sa popularité et pour sa syntaxe minimaliste, que j'ai trouvée plus accessible que Angular pour démarrer rapidement sur ce projet.

Pour la mise en forme, j'ai utilisé **TailwindCSS**, ce qui m'a permis d'implémenter rapidement un design cohérent et responsive, tout en conservant une bonne cohérence visuelle entre les pages et les sections.

En complément, j'ai utilisé **shadcn/ui** comme base de composants UI afin de disposer de composants réutilisables (boutons, éléments de navigation, composants de formulaire) et d'obtenir une interface homogène plus rapidement.
Enfin, j'ai mis en place le versioning avec **Git** et un dépôt distant sur **GitLab**, afin de suivre l'évolution du projet, sécuriser les modifications et conserver un historique clair des évolutions du code.

3.2.5 Apprentissage et montée en compétences

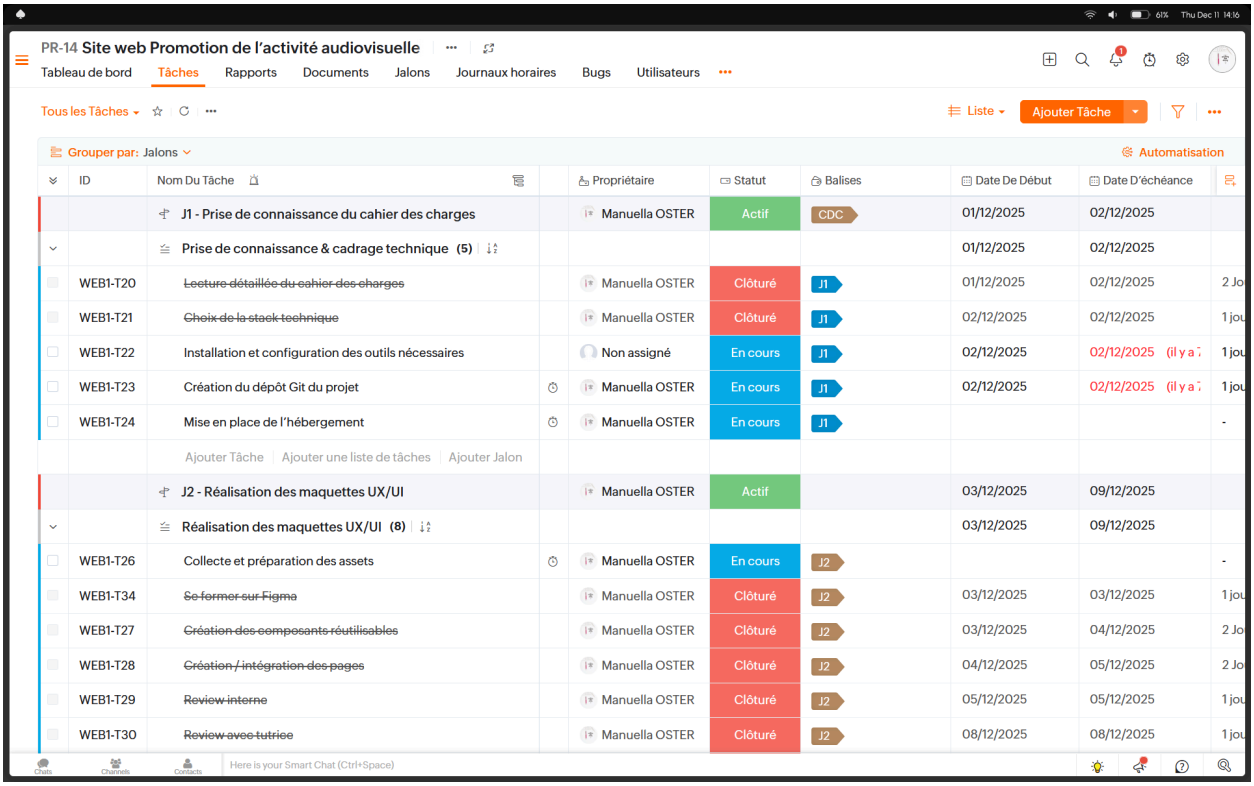
Cette phase m'a surtout permis de prendre en main les outils et la méthode de travail du projet, afin de respecter le planning et de produire un site clair, moderne et responsive.

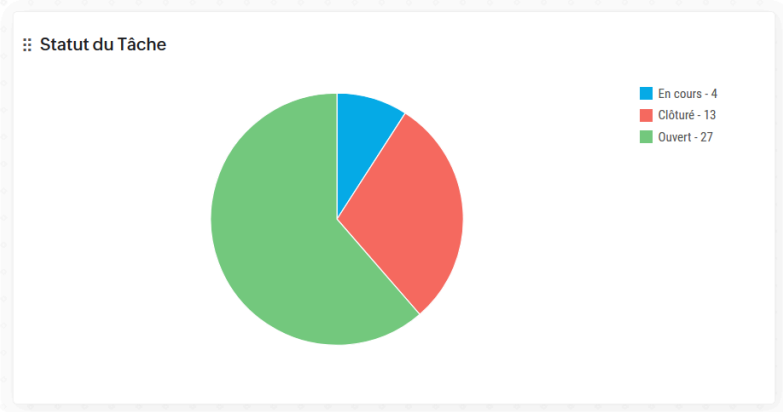
I. Gestion de projet (Zoho Project)

J'ai découvert **Zoho Project**, l'outil utilisé pour gérer le projet (présenté comme l'équivalent d'un Jira), et j'ai compris l'intérêt de transformer un cahier des charges en tâches concrètes et planifiées.

Cette approche m'a poussé à structurer mon travail de A à Z, avec une logique de jalons et d'échéances, plutôt que d'avancer au feeling au quotidien.

L'idée principale retenue est que ce temps de préparation paraît long sur le moment, mais évite ensuite une micro-planification permanente et améliore la régularité de l'avancement.





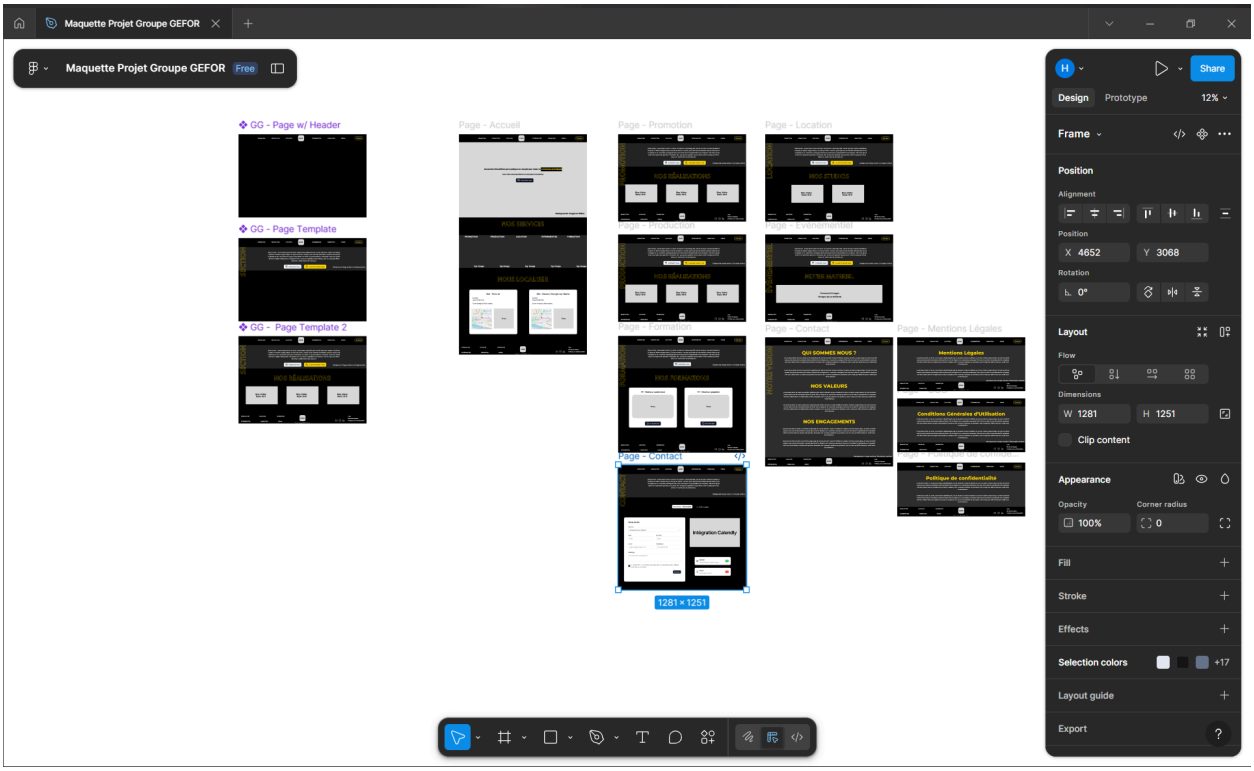
II. Maquettage (Figma)

J'ai utilisé cette période pour monter en compétence sur **Figma**, un outil que je connaissais de nom, mais que je n'avais jamais exploité sérieusement.

Jusqu'ici, je faisais plutôt de petits croquis rapides, sans entrer dans les détails, ce qui me permettait déjà de gagner en clarté et de ne pas perdre de temps à réfléchir au positionnement de chaque élément au moment du développement.

Figma reprend ce principe, mais en bien plus puissant : l'outil propose un système "**d'auto-layout**" (l'équivalent du **flexbox-css**) et un système de composants réutilisables, très proche de ce que permettent les frameworks front end comme Angular.

Figma demande plus de travail qu'un simple croquis, mais en échange j'ai une sorte de "**vision 3D**" du projet, car je visualise facilement la structure en "**box-model**" utilisée par le HTML, combinée à la philosophie des **composants réutilisables** des frameworks front end.



III. Framework front-end (Vue.js)

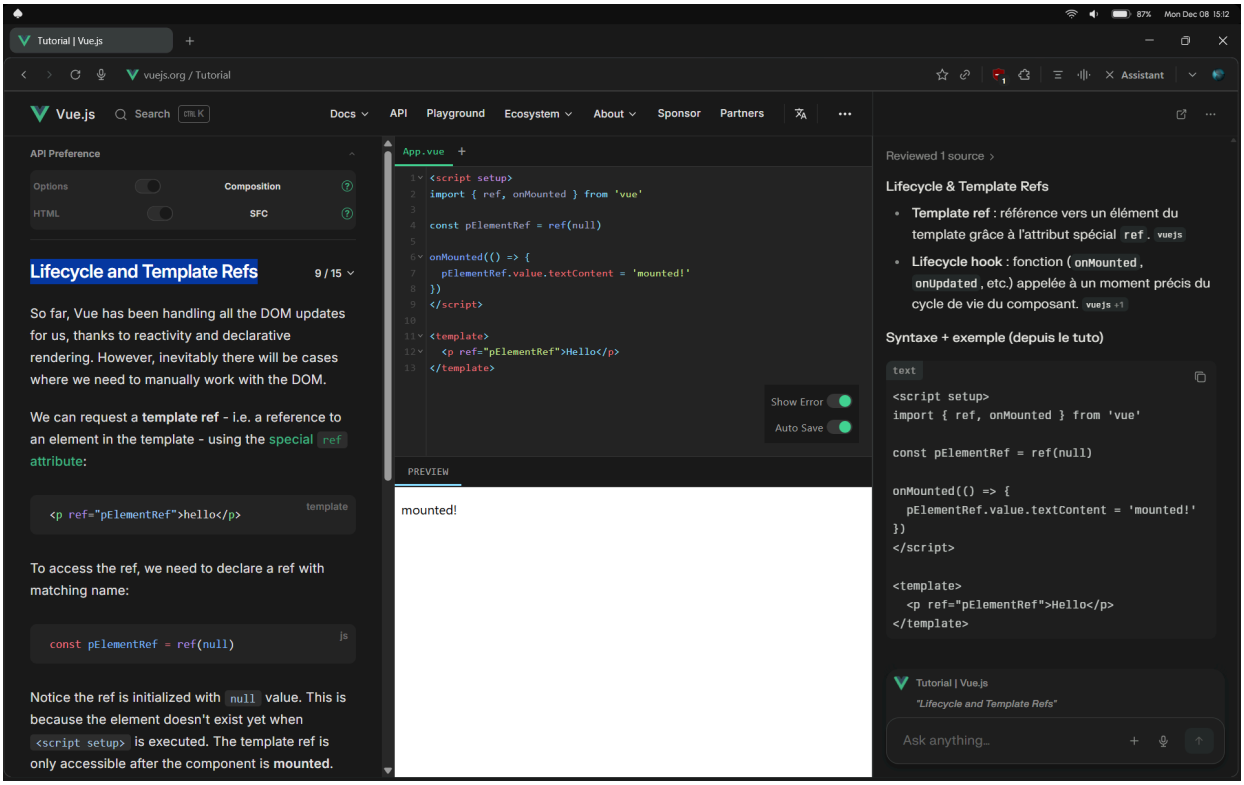
Le cahier des charges autorisait l'utilisation d'un framework front-end et l'objectif était d'obtenir une expérience utilisateur claire, fluide et moderne. Dans ce contexte, j'ai profité de cette phase pour commencer l'apprentissage de **Vue.js**, afin d'être opérationnel avant l'intégration front-end.

En parallèle du stage, je consacrais environ 30 minutes par jour à mettre en pratique **Angular** et **Express.js** vus en cours, à travers un petit projet full stack (refonte d'une mission réalisée lors de mon stage de première année).

Cette pratique régulière m'a permis de mieux comprendre la logique d'un framework front-end, ce qui a facilité l'apprentissage de Vue.js, car plusieurs concepts m'étaient déjà familiers (organisation, composants, logique de framework).

Pour apprendre plus vite, j'ai aussi utilisé le navigateur **Comet (le navigateur web de Perplexity, une sorte de moteur de recherche boosté à l'IA)** comme "mentor" : il m'a permis d'obtenir rapidement des pistes et des sources (documentation, articles, retours d'expérience) en lien direct avec mes questions, ce qui a rendu l'apprentissage plus efficace.

Grâce à cette méthode (documentation + mise en pratique + recherche assistée), il ne m'a fallu qu'environ trois jours pour acquérir les bases de Vue.js.



3.2.6 Arborescence et architecture de l'application

Pour répondre à cet objectif, j'ai organisé le dépôt front-end de manière à séparer les fichiers statiques, le code source, les composants, les pages (routes) et les services :

```
gefor-front/
├── public/  # Fichiers statiques
│   ├── fonts/  # Polices personnalisées
│   ├── formations/  # PDF des formations
│   └── logos/  # Logos et favicon
├── src/
│   ├── assets/  # Ressources compilées par Vite
│   │   ├── main.css  # Styles globaux
│   │   ├── hero-animations.css  # Animation pour AccueilPage
│   │   ├── images/
│   │   └── videos/
```

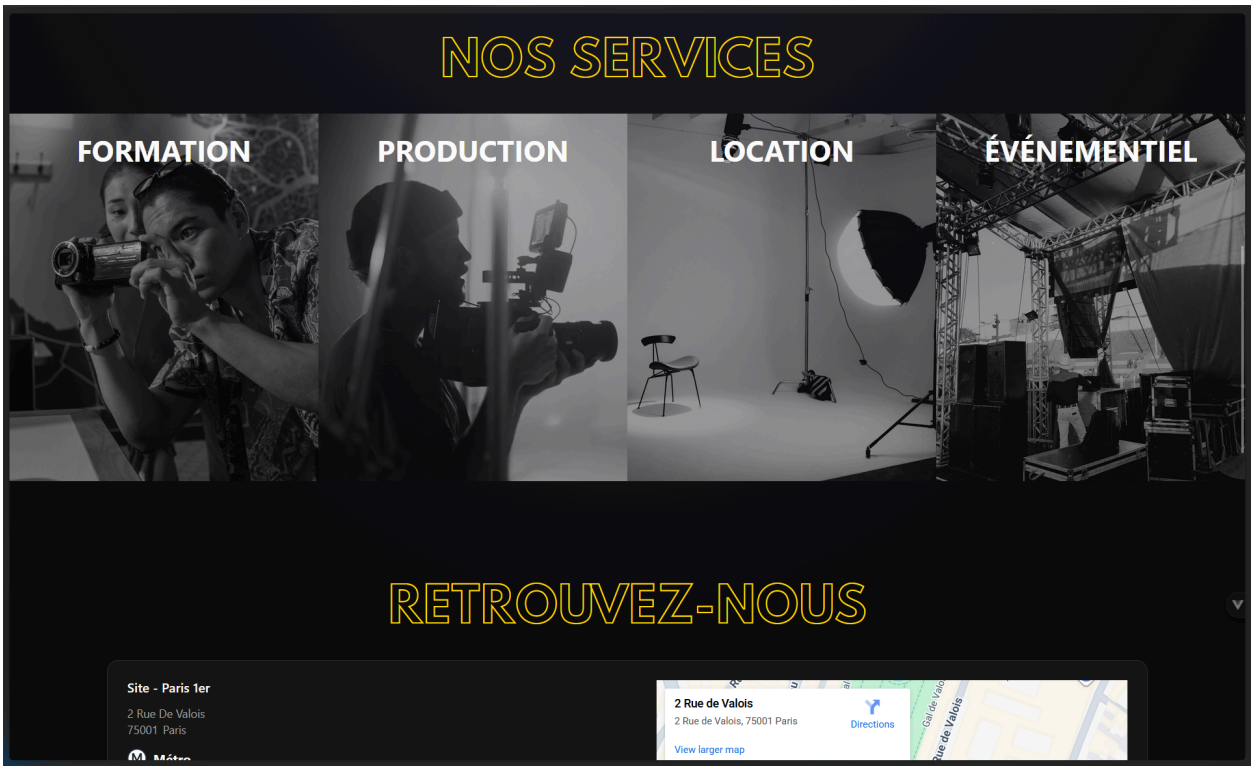
```
│   ├── components/
│   │   ├── layout/  # Composants de mise en page
│   │   │   ├── AppHeader.vue  # Navigation principale
│   │   │   └── AppFooter.vue  # Pied de page
│   │   └── ui/
│   │       ├── custom/  # Composants UI personnalisés
│   │       └── shadcn/  # Composants UI issu d’une librairie
│   ├── interfaces/  # Interfaces/Types TypeScript
│   ├── lib/
│   │   └── utils.ts  # Utilitaire pour tailwindCSS
│   ├── pages/  # Pages (routes)
│   │   ├── AccueilPage.vue  # Page d’accueil (/)
│   │   ├── VisionPage.vue  # Notre vision (/vision)
│   │   ├── ProductionPage.vue # Service (/production)
│   │   ├── LocationPage.vue  # Service (/location)
│   │   ├── EvenementielPage.vue  # Service (/evenementiel)
│   │   ├── FormationPage.vue  # Service (/formation)
│   │   ├── ContactPage.vue # Page contact (/contact)
│   │   ├── MentionsPage.vue
│   │   ├── PolitiquePage.vue
│   │   └── CGUPage.vue
│   ├── router/
│   │   └── index.ts  # Configuration de Vue Router
│   ├── services/  # Services API
│   │   ├── api.ts  # Instance préconfigurée
│   │   └── formResponseService.ts  # Service formulaire de contact
│   ├── App.vue  # Composant racine
│   └── main.ts  # Point d’entrée
├── .env.example  # Template des variables d’environnement
├── components.json # Configuration shadcn/ui
├── index.html  # Page initial
├── package.json  # Dépendances
├── tsconfig.json  # Configuration TypeScript
└── vite.config.ts  # Configuration Vite
```

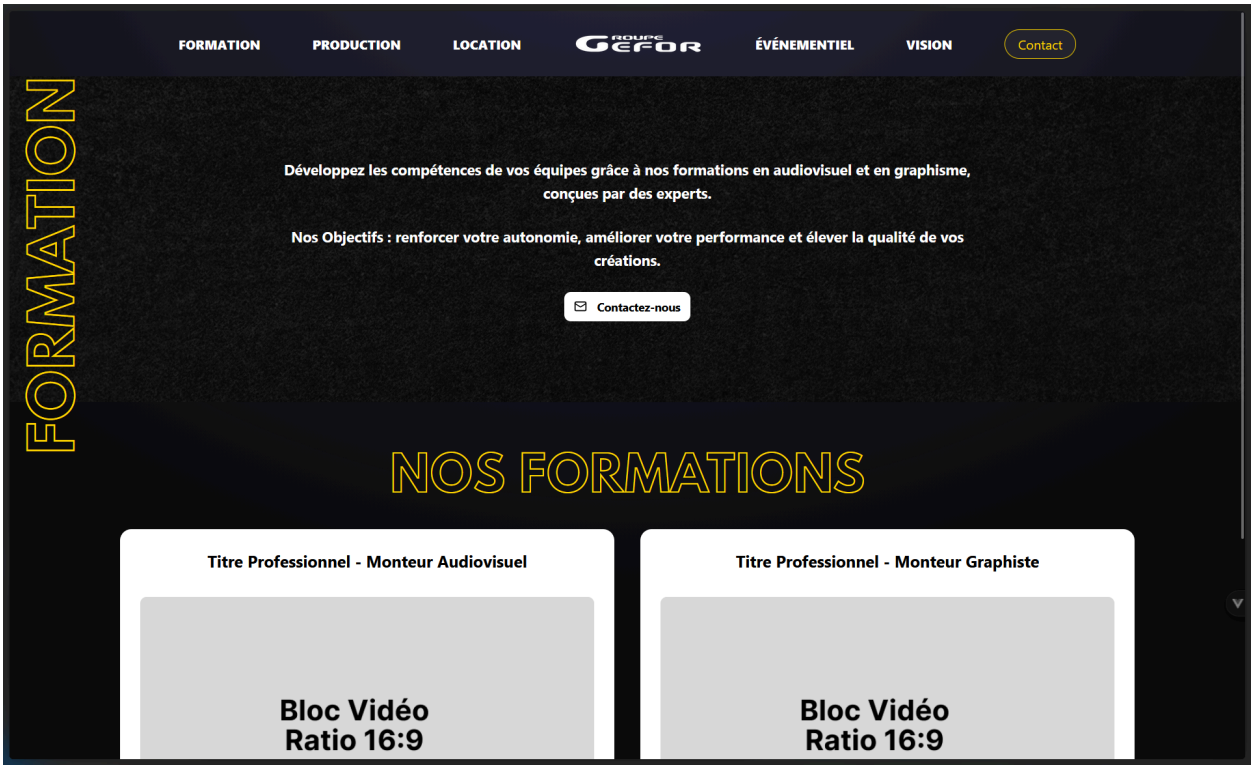
3.2.7 Réalisation

Après la phase de montée en compétences, j'ai commencé la **réalisation** du site en suivant le planning du cahier des charges : mise en place du squelette front-end, intégration progressive des pages, puis ajustements pour obtenir un rendu responsive et cohérent.

L'objectif était de livrer une interface claire et moderne, conforme à l'identité visuelle définie, tout en respectant la structure des 8 pages attendues et les éléments clés (navigation, pages services, contact).

Voici un aperçu du rendu final :


NOS SERVICES
FORMATION
PRODUCTION
LOCATION
ÉVÉNEMENTIEL
RETROUVEZ-NOUS
Site - Paris 1er
2 Rue De Valois
75001 Paris
2 Rue de Valois
2 Rue de Valois, 75001 Paris
Directions
View larger map


FORMATION
PRODUCTION
LOCATION
Groupe GEFOR
ÉVÉNEMENTIEL
VISION
Contact
FORMATION
Développez les compétences de vos équipes grâce à nos formations en audiovisuel et en graphisme, conçues par des experts.
Nos Objectifs : renforcer votre autonomie, améliorer votre performance et élever la qualité de vos créations.
Contactez-nous
NOS FORMATIONS
Titre Professionnel - Monteur Audiovisuel
Bloc Vidéo
Ratio 16:9
Titre Professionnel - Monteur Graphiste
Bloc Vidéo
Ratio 16:9

3.2.8 Bilan et compétences développées

L'utilisation de Zoho Project m'a permis d'avoir une **vision globale** du projet, comme une "carte routière" avec des étapes et des échéances clairement identifiées.

Même si la création initiale des tâches m'a paru longue et parfois répétitive (il est difficile de prévoir précisément, plusieurs semaines à l'avance, tout ce qu'il faudra faire "de A à Z"), je considère ce travail comme un investissement rentable.

Grâce à cette planification, je ne perdais plus de temps chaque matin à définir la prochaine action : je pouvais me concentrer sur l'exécution, la qualité du rendu et la résolution des problèmes plutôt que sur de la micro-planification quotidienne.

Sur le plan technique, ma pratique d'Angular m'a aidé à apprendre Vue.js plus rapidement, car certains principes de base (organisation d'un projet, logique de framework front-end, approche par composants) m'étaient déjà familiers.

Ce projet m'a aussi fait prendre conscience de l'intérêt des frameworks front-end : même si une version en HTML/CSS "vanilla" aurait pu être développée rapidement, un framework apporte une structure, des conventions et des composants réutilisables, ce qui améliore la maintenabilité et l'évolution du code.

Enfin, j'ai compris l'impact du mode de rendu sur un site vitrine.

Vue.js est souvent utilisé en **SPA/CSR** : le serveur envoie une page initiale très légère via le point d'entrée index.html, puis le navigateur exécute JavaScript pour générer l'affichage et gérer la navigation, ce qui peut être moins favorable au référencement **SEO** si le contenu n'est pas disponible dès le chargement initial.

Comme le SEO est un enjeu pour la visibilité d'un site vitrine, j'ai donc migré vers une génération statique (**SSG**) : les pages sont générées lors du build npm run build puis servies comme des fichiers statiques, ce qui répond mieux à l'objectif de visibilité en ligne.

3.3. Création d'un back-office

3.3.1 Contexte

Le site vitrine GEFOR Audiovisuel permet de présenter les services et de collecter des demandes via les formulaires de contact et de prise de rendez-vous.
Cependant, toutes ces demandes arrivent uniquement par email, ce qui rend le suivi difficile à centraliser et complique la gestion à plusieurs personnes.
Dans ce contexte, le groupe souhaite disposer d'un back-office interne pour structurer la gestion des réponses et mieux contrôler l'accès aux données recueillies.

3.3.2 Objectif et problématique

L'objectif principal de cette mission est de concevoir un back-office permettant à l'équipe habilitée de consulter, filtrer et gérer les demandes envoyées depuis le site vitrine (contact et prise de rendez-vous), dans un environnement réservé au personnel.

La problématique est double : d'une part, éviter de laisser les données éparpillées dans des boîtes mail individuelles, et d'autre part, respecter les contraintes RGPD en maîtrisant mieux qui accède à quelles informations et pendant combien de temps.

3.3.3 Cahier des charges

Le back-office doit offrir un espace de connexion réservé (avec comptes et rôles) permettant de gérer les demandes issues des formulaires de contact et de prise de rendez-vous.

Les données doivent être organisées de manière claire (réponses récentes, archives, comptes utilisateurs, journal d'activité), avec des droits différenciés selon les profils (super administrateur, administrateur, staff) pour limiter chaque utilisateur à ce dont il a réellement besoin.

Le système doit également intégrer des règles de conservation et de suppression automatique des données (par exemple séparation "récentes/archives" et purge au-delà de la durée définie), afin de rester conforme au RGPD tout en gardant un suivi des actions réalisées dans le back-office.

3.3.4 Choix techniques

Pour le front-end du back-office, j'ai choisi de rester sur **Vue.js** avec **Tailwind CSS** et la librairie de composants **shadcn/ui**, afin de conserver la même stack que pour le front-office (site vitrine) et de réutiliser au maximum la logique de composants et les styles déjà en place.

Pour le back-end, j'ai retenu la combinaison **Node.js / Express.js / PostgreSQL**, orchestrée avec **Docker** pour la base de données.

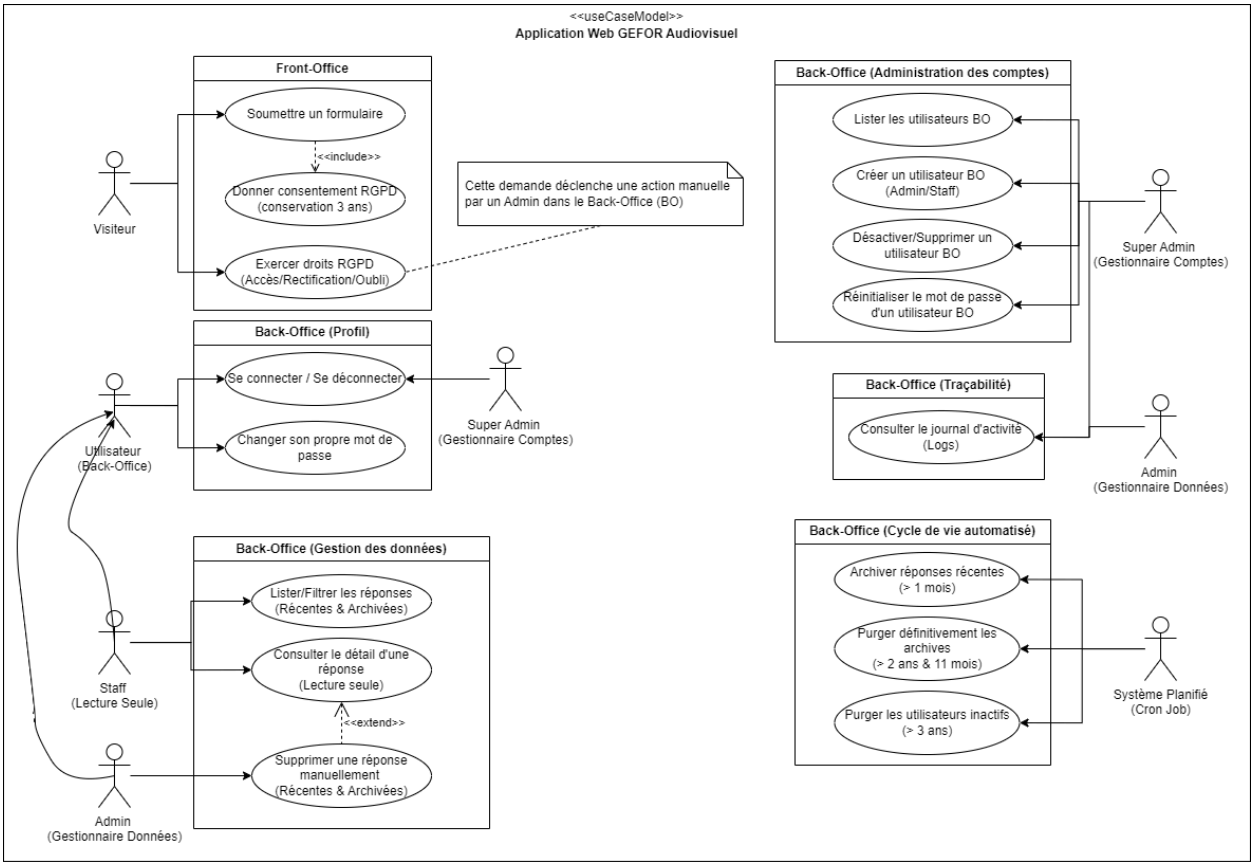
J'avais travaillé en parallèle sur un projet personnel utilisant Node.js, Express.js et Docker, ce qui m'a permis de savoir rapidement comment structurer l'API et comment isoler PostgreSQL dans un conteneur dédié, sans avoir à installer la base de données directement sur la machine hôte.

Enfin, l'ensemble du code est versionné avec **Git** et hébergé sur **GitLab**, ce qui permet de suivre l'évolution du projet, de revenir en arrière en cas de besoin et d'envisager plus facilement un travail collaboratif ultérieur.

3.3.5 Fonctionnalités / cas d'utilisation

Pour cette mission, les fonctionnalités du back-office ont été définies à partir des besoins concrets de gestion des formulaires (contact) et des différents profils d'utilisateurs internes.

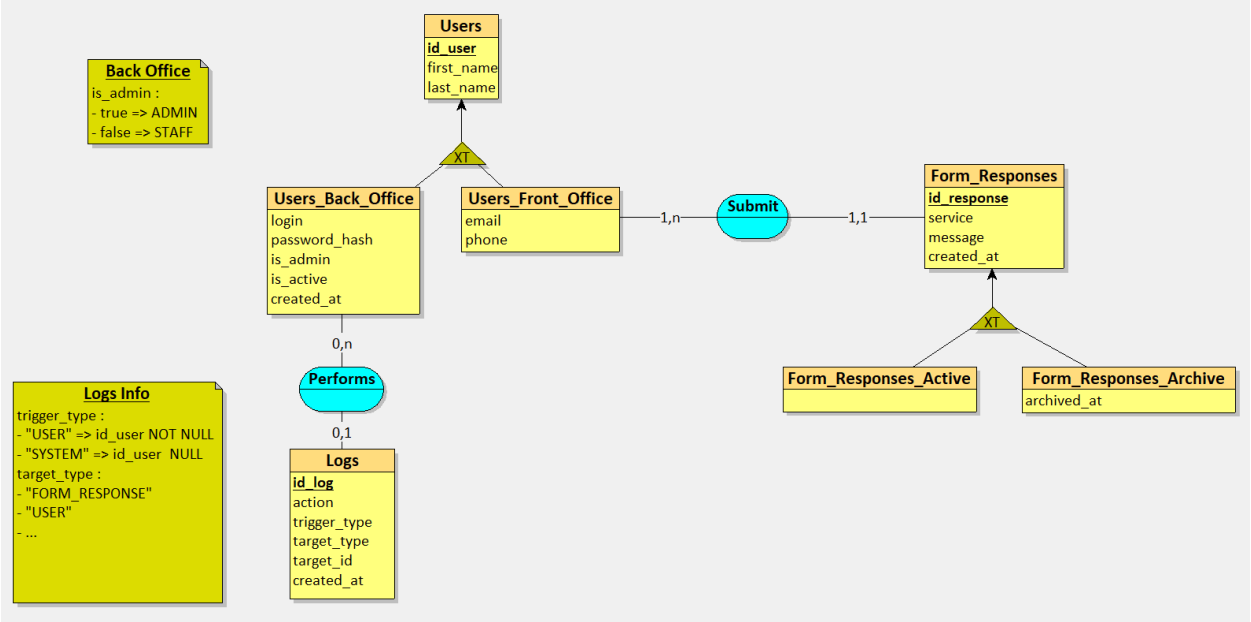
Le diagramme de cas d'utilisation ci-dessous présente les principaux acteurs (super administrateur, administrateur, staff) et les actions associées, comme la gestion des comptes, la consultation des réponses récentes, l'accès aux archives ou encore la suppression de réponses par les profils autorisés.



3.3.6 Modèle conceptuel de données (MCD)

Le MCD du back-office s'articule autour des entités principales nécessaires à la gestion des réponses aux formulaires (contact), des comptes utilisateurs et du suivi d'activité.

Il met notamment en évidence la séparation entre les données métiers (réponses récentes, archives, comptes) et le journal d'activité, qui trace les actions réalisées dans le système.



Ici, la table **Log** est indirectement en association avec les différentes entités métiers qu'elle journalise : cette relation est implémentée par deux attributs, target_type et target_id, qui indiquent respectivement le type d'entité concernée et son identifiant, ce qui rend le mécanisme de journalisation extensible à de nouveaux types d'objets si nécessaire.

3.3.7 Architecture du projet

I. Architecture globale

Lors d'un projet personnel précédent (plateforme e-commerce minimaliste pour la vente de livres), l'application était organisée en deux dépôts GitHub distincts : un pour le front-end Angular et un pour le back-end Express.js, ce qui correspond à une architecture dite **polyrepo**.

Pour cette mission, j'ai fait le choix d'une **monorepo**, en regroupant le back-office et l'API Node.js/Express.js dans un seul dépôt, ce qui me donne une vision globale de l'application et me permet de passer facilement du front au back sans changer de projet.

Je suis volontairement resté sur une utilisation minimaliste de la monorepo, mais ce choix ouvre la possibilité de refactoriser encore plus le code, par exemple via un répertoire partagé entre back-end et back-office pour y stocker des types et interfaces TypeScript communs (comme le type User), afin d'éviter les doublons et les modifications répétées.

```
gefor-back/
├── .env                    # Variables d'environnement (RACINE)
├── .env.example            # Modèle de configuration
├── docker-compose.yml      # Docker pour PostgreSQL
├── Makefile                # Raccourci commandes utilitaires
│
├── back-end/               # API REST
│   └── README.md           # <-- Documentation détaillée
│
├── back-office/            # Interface admin Vue.js
│   ├── .env                # Variables Vite (SÉPARÉ pour le
bundler)
│   └── README.md           # <-- Documentation détaillée
│
└── database/
    └── init.sql            # Script d'initialisation BDD pour
Docker
```

II. Architecture back-end

Le back-end est organisé autour d'une API REST développée avec Node.js et Express.js, connectée à une base de données PostgreSQL exécutée dans un conteneur Docker.

L'arborescence distingue clairement les couches de l'application (routes, contrôleurs, services éventuels, modèles et configuration de la base), ce qui facilite la lisibilité du code et l'évolution future du projet.

```
back-end/
├── src/
│   ├── app.ts                  # Configuration Express (CORS,
Helmet, middlewares)
│   ├── server.ts               # Point d'entrée du serveur
│   ├── constants.ts            # Variables et constantes globales
│   ├── env.ts                  # Chargement des variables
d'environnement
│   │
│   ├── config/                 # Configuration (base de données)
│   │   └── database.ts
│   │
│   ├── controllers/            # Logique métier
│   │   ├── formResponseController.ts
│   │   ├── logController.ts
│   │   ├── userBOController.ts   # Utilisateurs Back-Office
│   │   └── userFOController.ts   # Utilisateurs Front-Office
│   │
│   ├── docs/                   # Documentation API
│   │   └── swagger.yaml
│   │
│   ├── interfaces/             # Types/Interfaces TypeScript
│   │   ├── TokenPayload.ts
│   │   ├── ValidationResult.ts
│   │   └── ...
```

```
│   │
│   ├── middlewares/
│   │   ├── authMiddleware.ts     # Vérification JWT
│   │   └── errorHandler.ts       # Gestion centralisée des erreurs
│   │
│   ├── models/                   # Accès base de données
│   │   ├── formResponseModel.ts
│   │   ├── logModel.ts
│   │   ├── userBOModel.ts
│   │   └── userFOModel.ts
│   │
│   ├── routes/                   # Définition des routes
│   │   ├── authRouter.ts
│   │   ├── formResponseRouter.ts
│   │   ├── logRouter.ts
│   │   ├── userBORouter.ts
│   │   └── userFORouter.ts
│   │
│   └── utils/                    # Utilitaires
│       ├── apiResponseUtils.ts
│       ├── authentificationUtils.ts # Validation des entrées
│       ├── authorizationUtils.ts    # Vérification des rôles
│       ├── cronUtils.ts             # Tâches planifiées (archivage,
purge)
│       ├── emailUtils.ts            # Envoi d'emails
│       └── loggerUtils.ts           # Journalisation
│
├── package.json
└── tsconfig.json
```

III. Architecture back-office

Le back-office est une application Vue.js qui consomme l'API du back-end et s'appuie sur Tailwind CSS et shadcn/ui pour la partie interface.

L'architecture front-end est structurée en composants (pages, layouts, composants génériques) afin de garder un code modulaire et facilement maintenable.

```
back-office/
├── public/                    # Fichiers statiques
│
├── src/
│   ├── App.vue                # Composant racine
│   ├── main.ts                # Point d'entrée
│   ├── constants.ts           # Constantes globales
│   │
│   ├── assets/                # Assets (CSS, images)
│   │   └── main.css           # Styles globaux + TailwindCSS
│   │
│   ├── components/            # Architecture par feature
│   │   ├── admin/             # Composants page administration
│   │   ├── dashboard/         # Composants tableau de bord
│   │   ├── layout/            # Layout principal
│   │   ├── logs/              # Composants page logs
│   │   ├── users/             # Composants page utilisateurs
```

```
│   │   └── ui/                    # Composants UI (shadcn-vue)
│   │       └── shadcn/
│   │
│   ├── interfaces/                # Types TypeScript
│   │
│   ├── lib/
│   │   └── utils.ts               # Utilitaires (cn, etc.)
│   │
│   ├── pages/                     # Pages de l'application
│   │   ├── AdminPage.vue          # Page super admin
│   │   ├── ArchivesPage.vue       # Archives des formulaires
│   │   ├── DashboardPage.vue      # Demandes récentes
│   │   ├── LoginPage.vue          # Connexion
│   │   ├── LogsPage.vue           # Historique des actions
│   │   └── UsersPage.vue          # Gestion utilisateurs
│   │
│   ├── router/
│   │   └── index.ts               # Configuration Vue Router + guards
│   │
│   └── services/                  # Services API
│       ├── api.ts                 # Configuration Axios
│       ├── authService.ts         # Authentification
│       ├── formResponseService.ts
│       ├── logService.ts
│       ├── userBOService.ts
│       └── userFOService.ts
│
├── components.json                # Configuration shadcn-vue
├── vite.config.ts                 # Configuration Vite
├── package.json
└── tsconfig.json
```

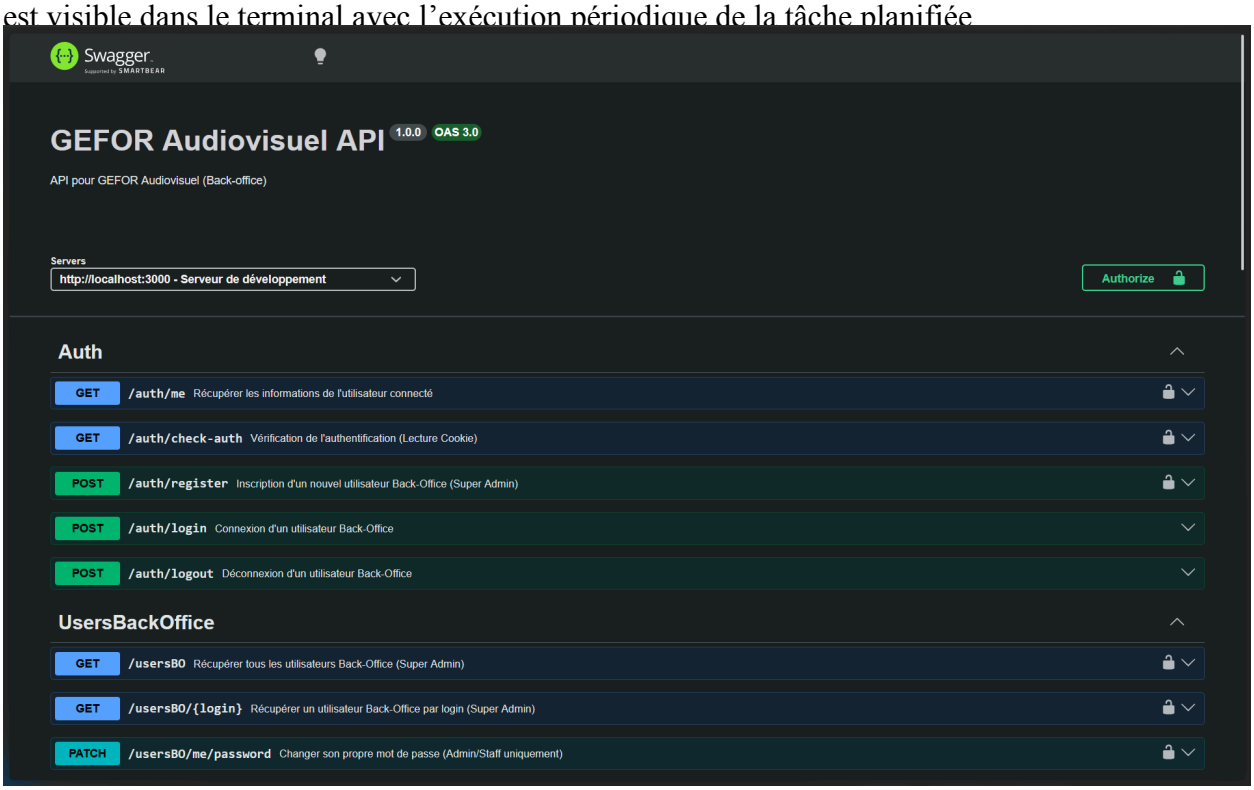
3.3.8 Réalisations

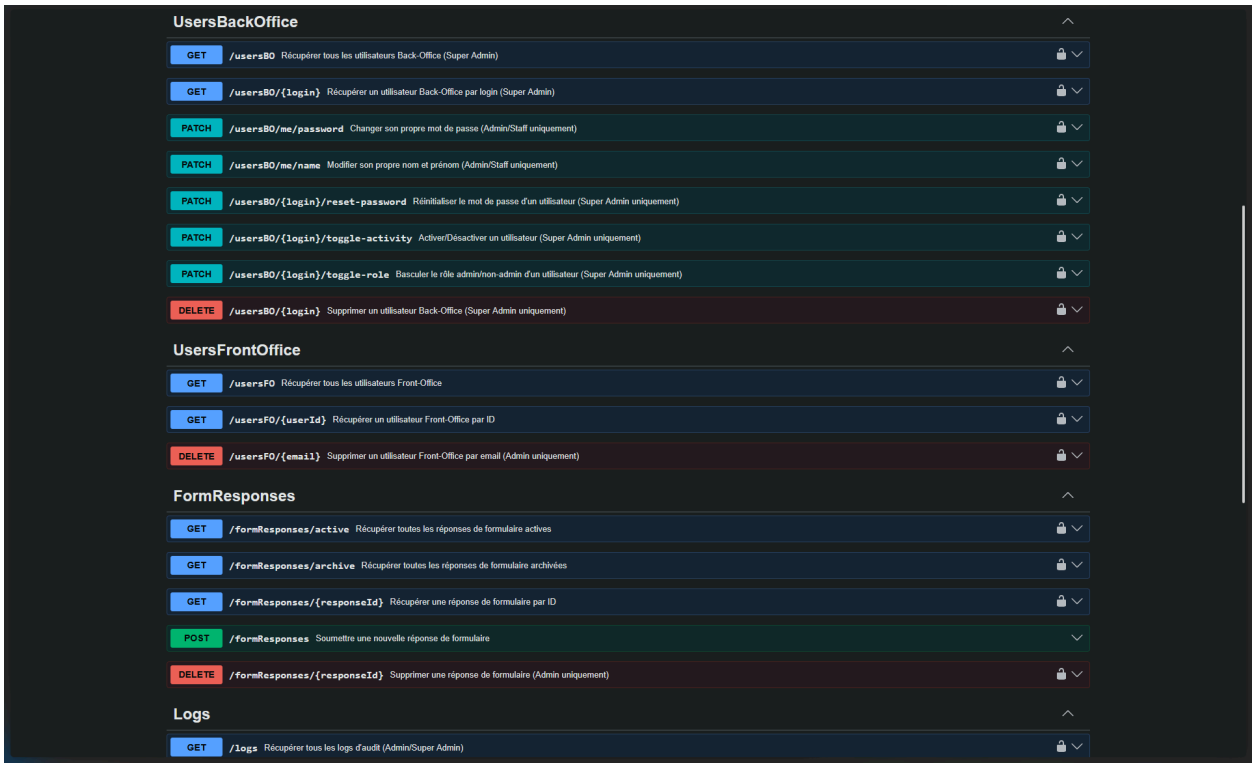
I. Réalisation back-end

L'API REST du back-office a été entièrement documentée avec Swagger, ce qui permet de visualiser l'ensemble des routes, leurs paramètres, les schémas de données ainsi que les statuts de réponse attendus.

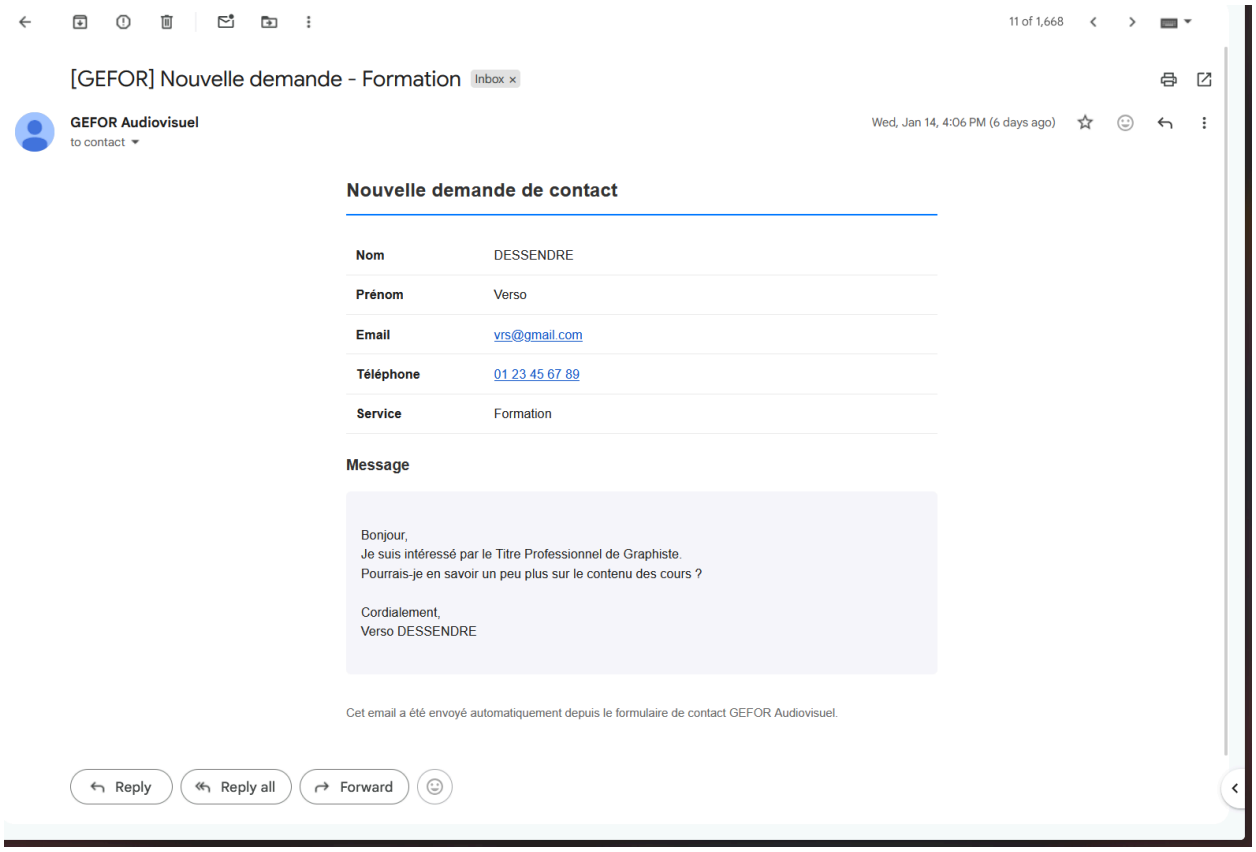
Un cron job a été mis en place côté back-end pour automatiser certaines tâches de maintenance, comme la réinitialisation ou la purge régulière de données de test.

Pendant le développement, ce job a été configuré pour s'exécuter toutes les deux minutes, ce qui est visible dans le terminal avec l'exécution périodique de la tâche planifiée.





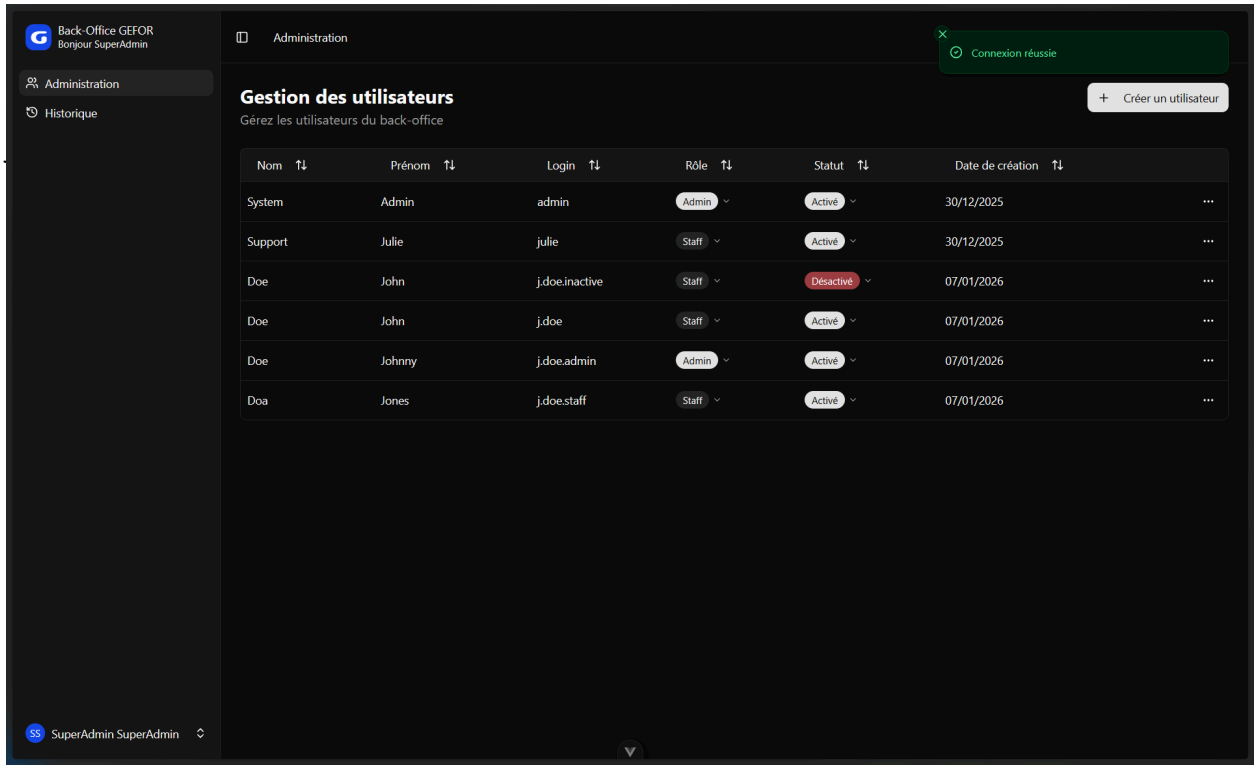

```
---Lancement en mode développement---
[14/01/2026 18:38:29] [INFO] PostgreSQL connecté à la base: gefor_db
[14/01/2026 18:38:29] [INFO] Serveur en cours d'exécution sur http://localhost:3000
[14/01/2026 18:38:29] [INFO] Accès à la documentation sur http://localhost:3000/docs
[14/01/2026 18:38:29] [INFO] [CRON] Cron jobs initialisés (TEST: toutes les 2 minutes)

[14/01/2026 18:40:00] [INFO] [CRON] Démarrage des tâches de maintenance...
[14/01/2026 18:40:00] [INFO] [CRON] 1 réponse(s) archivée(s)
[14/01/2026 18:40:00] [INFO] [CRON] 2 archive(s) supprimée(s)
[14/01/2026 18:40:00] [INFO] [CRON] 2 compte(s) inactif(s) supprimé(s)
[14/01/2026 18:40:00] [INFO] [CRON] Tâches de maintenance terminées.
[14/01/2026 18:42:00] [INFO] [CRON] Démarrage des tâches de maintenance...
[14/01/2026 18:42:00] [INFO] [CRON] 0 réponse(s) archivée(s)
[14/01/2026 18:42:00] [INFO] [CRON] 0 archive(s) supprimée(s)
[14/01/2026 18:42:00] [INFO] [CRON] 0 compte(s) inactif(s) supprimé(s)
[14/01/2026 18:42:00] [INFO] [CRON] Tâches de maintenance terminées.
[14/01/2026 18:44:00] [INFO] [CRON] Démarrage des tâches de maintenance...
[14/01/2026 18:44:00] [INFO] [CRON] 0 réponse(s) archivée(s)
[14/01/2026 18:44:00] [INFO] [CRON] 0 archive(s) supprimée(s)
[14/01/2026 18:44:00] [INFO] [CRON] 0 compte(s) inactif(s) supprimé(s)
[14/01/2026 18:44:00] [INFO] [CRON] Tâches de maintenance terminées.
```

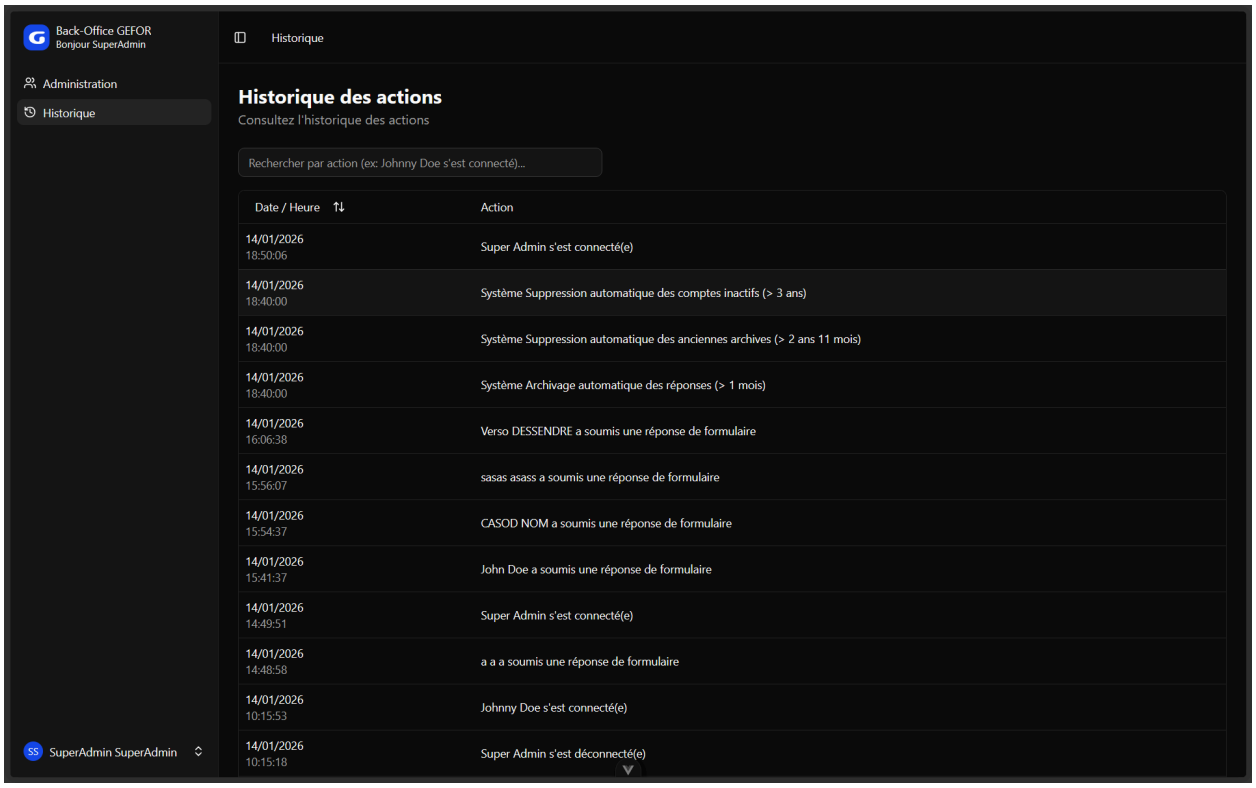

11 of 1,668
[GEFOR] Nouvelle demande - Formation
Inbox ×
GEFOR Audiovisuel
to contact
Wed, Jan 14, 4:06 PM (6 days ago)
Nouvelle demande de contact
Nom
DESSENDRE
Prénom
Verso
Email
vrs@gmail.com
Téléphone
01 23 45 67 89
Service
Formation
Message
Bonjour,
Je suis intéressé par le Titre Professionnel de Graphiste.
Pourrais-je en savoir un peu plus sur le contenu des cours ?
Cordialement,
Verso DESSENDRE
Cet email a été envoyé automatiquement depuis le formulaire de contact GEFOR Audiovisuel.
Reply
Reply all
Forward

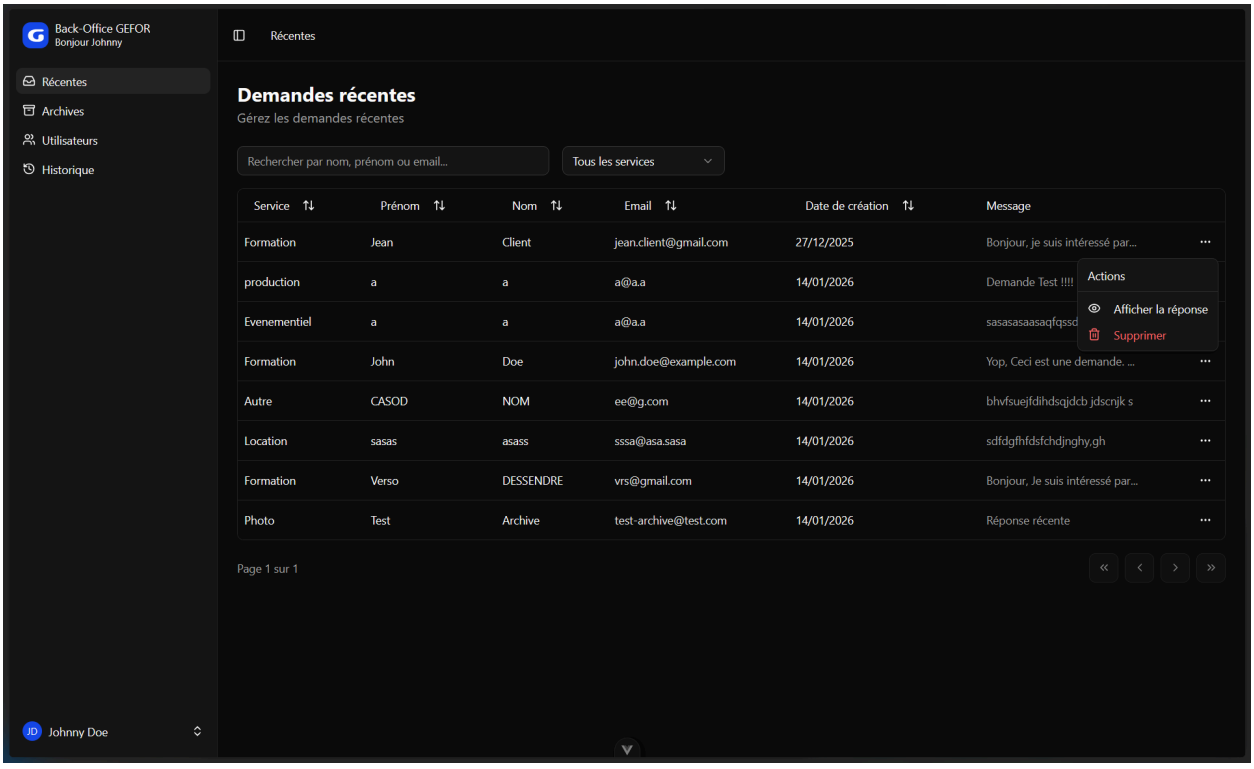
II. Réalisation back-office

Le back-office développé avec Vue.js propose une interface de connexion permettant de restreindre l'accès aux seuls utilisateurs autorisés, en s'appuyant sur le système d'authentification exposé par l'API.

Une fois connecté, l'utilisateur accède à un tableau de bord présentant les principales sections du back-office (réponses récentes, archives, journal d'activité, gestion des comptes selon le rôle).



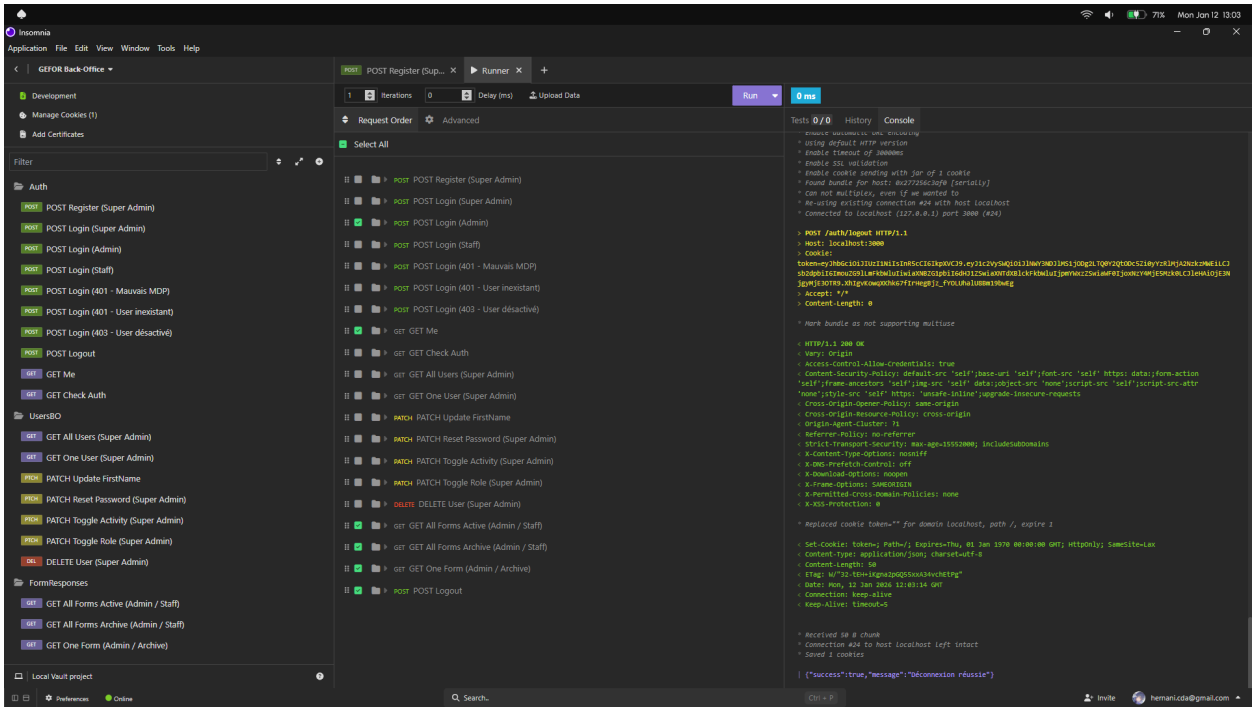

Back-Office GEFOR
Bonjour SuperAdmin
Administration
Historique
Historique
Historique des actions
Consultez l'historique des actions
Rechercher par action (ex: Johnny Doe s'est connecté)...
Date / Heure
Action
14/01/2026 18:50:06
Super Admin s'est connecté(e)
14/01/2026 18:40:00
Système Suppression automatique des comptes inactifs (> 3 ans)
14/01/2026 18:40:00
Système Suppression automatique des anciennes archives (> 2 ans 11 mois)
14/01/2026 18:40:00
Système Archivage automatique des réponses (> 1 mois)
14/01/2026 16:06:38
Verso DESSENDRE a soumis une réponse de formulaire
14/01/2026 15:56:07
sasas asass a soumis une réponse de formulaire
14/01/2026 15:54:37
CASOD NOM a soumis une réponse de formulaire
14/01/2026 15:41:37
John Doe a soumis une réponse de formulaire
14/01/2026 14:49:51
Super Admin s'est connecté(e)
14/01/2026 14:48:58
a a a soumis une réponse de formulaire
14/01/2026 10:15:53
Johnny Doe s'est connecté(e)
14/01/2026 10:15:18
Super Admin s'est déconnecté(e)
SuperAdmin SuperAdmin


Back-Office GEFOR
Bonjour Johnny
Récentes
Archives
Utilisateurs
Historique
Récentes
Demandes récentes
Gérez les demandes récentes
Rechercher par nom, prénom ou email...
Tous les services
Service
Prénom
Nom
Email
Date de création
Message
Formation Jean Client jean.client@gmail.com 27/12/2025 Bonjour, je suis intéressé par...
production a a a@a.a 14/01/2026 Demande Test !!!!
Evenementiel a a a@a.a 14/01/2026 sasasasaasaqfqssd
Formation John Doe john.doe@example.com 14/01/2026 Yop, Ceci est une demande. ...
Autre CASOD NOM ee@g.com 14/01/2026 bhvfsuejfdihdsqjdcb jdscnjk s
Location sasas asass sssa@asa.sasa 14/01/2026 sdfdgfhfdsfchdjnghy,gh
Formation Verso DESSENDRE vrs@gmail.com 14/01/2026 Bonjour, Je suis intéressé par...
Photo Test Archive test-archive@test.com 14/01/2026 Réponse récente
Actions
Afficher la réponse
Supprimer
Page 1 sur 1
Johnny Doe

3.3.9 Bilan et compétences développées

Sur ce projet, une approche **feature par feature** a été adoptée plutôt que “finir le back-end avant le front-end”, ce qui a permis de rester concentré sur une fonctionnalité à la fois (API, interface, tests) et de limiter l’anticipation inutile alors que l’application était susceptibles d’évoluer.

Pour les tests, Insomnia a été utilisé en complément de Swagger : là où Swagger est surtout orienté documentation et exploration visuelle de l’API, Insomnia a permis d’enchaîner rapidement les requêtes, de définir des variables (tokens, identifiants, mots de passe de test) et de les réutiliser automatiquement, ce qui a fait gagner du temps sur les scénarios répétitifs.



Enfin, le back-end a été sécurisé en utilisant des cookies **httpOnly** pour stocker le token d'authentification, ce qui empêche le JavaScript côté client de lire directement le jeton et réduit ainsi les risques d'attaques de type XSS.

Cette mise en place a permis de consolider les compétences en sécurité web (gestion des tokens, cookies sécurisés, séparation front/back) et d'appliquer concrètement des bonnes pratiques couramment utilisées en production.

4. Conclusion

Ce second stage a été une expérience particulièrement riche, puisqu'il m'a permis de passer d'un simple site vitrine à la conception d'un véritable écosystème complet : front-office, back-office et API sécurisée autour d'un besoin métier concret pour le groupe GEFOR.

En travaillant sur la gestion des formulaires, la structuration des données, la journalisation des actions et la sécurisation des accès, le projet de back-office m'a fait découvrir des problématiques plus proches d'une application professionnelle en production.

Sur le plan technique, ce stage m'a permis de consolider l'usage de technologies modernes déjà abordées en autonomie (Node.js, Express, Docker, PostgreSQL, Vue.js, Tailwind, shadcn/ui) et de les mettre en pratique dans un contexte cohérent, avec une architecture pensée pour évoluer.

Même si le front-end n'est pas ce qui m'attire le plus, l'usage combiné de TailwindCSS et de librairies UI rend cette partie du travail nettement plus agréable.

Sur le plan personnel, ce stage m'a appris à mieux gérer mon travail dans la durée, à documenter mes choix, à accepter que le projet puisse évoluer et à rester pragmatique dans les solutions retenues en fonction des besoins réels de l'entreprise.

Enfin, je tiens à remercier l'entreprise pour sa bienveillance, sa confiance et l'opportunité qui m'a été offerte. Le fait qu'ils aient répondu à ma candidature en moins de deux jours a été un vrai signe de confiance et de considération, qui m'a particulièrement marqué. Ce stage a été formateur sur tous les plans et représente pour moi une étape importante dans mon parcours.