

STAGE

HÔPITAL PRIVÉ DU HAINAUX

04 JANVIER - 14 FÉVRIER 2026

RAPPORT



ELSAN

HÔPITAUX PRIVÉS DU HA
POLYCLINIQUE WALLONNE

FEDNAIL
LECLERCQ

BTS SIO- SLAM

2025-2026



RAPPORT DE STAGE – BTS SIO OPTION SLAM

Année Universitaire : 2025-2026

Sujet : Conception et déploiement d'une application web de gestion de planning (Hainaut Schedule)

Stagiaire : Fednail LECLERCQ

Formation : BTS Services Informatiques aux Organisations

Établissement : Lycée Gaston Berger (Lille)

Entreprise d'accueil : Hôpital Privé du Hainaut (Groupe ELSAN)

Période du stage : Janvier 2026 – Février 2026 (6 semaines)

Lieu : Valenciennes, Nord (59)

Tuteur Académique : M. Julien VILVANDRE

Tuteur Professionnel : M. Florian LICOUR

Table des matières

REMERCIEMENTS	4
Introduction.....	5
Contexte général	5
Objectifs du rapport	5
Périmètre du document	5
Public cible	5
PRÉSENTATION DE L'ENTREPRISE	6
Secteur d'activité et Historique	6
<i>Taille et Organisation</i>	6
<i>Contexte Technologique</i>	6
Contexte de l'établissement.....	7
L'Hôpital Privé du Hainaut.....	7
Le Service Informatique	7
Local du Service Informatique :	7
Situation Initiale	8
Analyse du besoin et spécifications	9
<i>Expression du besoin</i>	9
Modélisation des données	10
Spécifications Fonctionnelles	10
Contraintes Techniques	10
Environnement technique	11
Stack Technique	11
Poste de développement	12
Réalisation technique	12
Développement Backend (Semaine 1).....	12
Développement Frontend (Semaines 2-4)	15
Liste des Erreur Rencontré :	20
Résultats finaux de l'interface :	21
Support et Infrastructure	25
Compétences développées	29
Compétences Techniques	29
Compétences Méthodologiques	29
Compétences Humaines (Soft Skills)	29
Conclusion et perspectives.....	30
Bilan Personnel	30

Annexes	31
Annexe A : Diagramme de séquence : Création de planning avec notification	31
Annexe B Architecture de déploiement.....	32
Annexe C : Base de code modulaire	33
Annexe D : Glossaire.....	34

REMERCIEMENTS

Je tiens à exprimer ma sincère gratitude à M. Florian LICOUR, Technicien Manager /Chef de Service Informatique, pour m'avoir accueilli au sein de l'équipe informatique de l'Hôpital Privé du Hainaut. Sa confiance, ses conseils techniques et son accompagnement ont été déterminants dans la réussite de ce projet.

Je remercie également l'ensemble de l'équipe IT (M. Aurélien VANDAMME, M. Brahim ZABOUB, M. Alexandre BOUCHE) pour leur intégration bienveillante.

Je souhaite adresser mes remerciements à mon professeur référent, pour son suivi pédagogique et ses orientations tout au long de cette période.

Enfin, je remercie la direction de l'Hôpital Privé du Hainaut pour m'avoir permis de réaliser ce stage dans des conditions optimales.

Introduction

Contexte général

Dans le cadre de ma seconde année de BTS SIO option SLAM (Solutions Logicielles et Applications Métiers), j'ai effectué un stage de six semaines au sein du service informatique de l'Hôpital Privé du Hainaut. Ces établissements multisites du groupe ELSAN nécessitent une coordination technique précise entre ses différentes cliniques.

Objectifs du rapport

Ce rapport détaille les activités techniques et organisationnelles mises en œuvre durant cette période. La mission confiée consistait à remplacer la gestion artisanale des plannings (fichiers Excel, emails) par une application web de gestion de planning multisites, destinée au service informatique.

Périmètre du document

Ce rapport couvre l'ensemble du cycle de vie du **Projet Hainaut Schedule**, notamment :

- Le contexte métier et les contraintes techniques identifiés.
- L'architecture logicielle et la justification des choix technologiques.
- Le détail des fonctionnalités implémentées et leurs spécifications.
- Le bilan des compétences acquises et les difficultés rencontrées.
- Les préconisations pour une mise en production pérenne.

Public cible

Ce document s'adresse à trois publics distincts :

- **Encadrement académique** : Pour l'évaluation des compétences du BTS SIO option SLAM.
- **Encadrement professionnel (Tuteur)** : Pour la validation technique et organisationnelle du travail fourni.
- **Équipe IT HPH** : Pour servir de vue d'ensemble fonctionnelle lors de la possible maintenance évolutive.

PRÉSENTATION DE L'ENTREPRISE

Secteur d'activité et Historique

L'Hôpital Privé du Hainaut est un établissement de santé privé à but non lucratif, intégré au groupe ELSAN, leader de l'hospitalisation privée en France. Situé dans le Valenciennois, il regroupe plusieurs cliniques complémentaires (chirurgie, soins de suite, rééducation).

Taille et Organisation

L'établissement emploie environ 760 collaborateurs et 158 médecins, prenant en charge plus de 36 600 patients par an. Le service informatique, rattaché à la Direction des Systèmes d'Information, assure le support technique et le maintien en condition opérationnelle des infrastructures pour l'ensemble des sites.

Contexte Technologique

L'équipe IT est composée de quatre techniciens intervenant sur sept cliniques distinctes (IOS, CBS, SSA, SQT, VAU, SOM, FLA). L'infrastructure repose sur un réseau interne sécurisé, des serveurs locaux et des applications métiers spécifiques.

Contexte de l'établissement

L'Hôpital Privé du Hainaut

Intégré au groupe ELSAN, leader de l'hospitalisation privée en France, l'Hôpital Privé du Hainaut est un établissement de santé multisites desservant le territoire du Haute de France.

L'établissement dispose de sept cliniques réparties géographiquement (IOS, CBS, SSA, SQT, VAU, SOM, FLA).

Le Service Informatique

Le service informatique, rattaché à la Direction des Systèmes d'Information, assure le support technique de niveau 2 pour l'ensemble des sites, gère l'ensemble des sites via des demandes de tickets, pour cela un logiciel de Ticketing, développé en interne par Elsan.

Composition de l'équipe IT :

- Alexandre BOUCHE – Technicien Réseaux
- Aurélien VANDAMME – Technicien Infrastructure
- Brahim ZABOUB – Technicien Support
- Florian LECOURE – Technicien Manager /Chef de Service Informatique

Missions principales :

- Support utilisateur et maintenance du parc (150+ postes)
- Administration des infrastructures réseau et serveurs
- Développement d'outils métiers internes
- Supervision et cybersécurité

Local du Service Informatique :



Situation Initiale

Avant ce projet, la gestion des plannings hebdomadaires de l'équipe IT s'effectuait via des outils disparates et non synchronisés.

Voici à quoi ressemblait l'ancien planning utilisé par l'équipe :

AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB
18-janv	19-janv		20-janv		21-janv		22-janv		23-janv		24	
	AM	PM	AM	PM	AM	PM	AM	PM	AM	PM		
	CBS	CBS	SSA	SSA	SSA	SSA	SSA	SSA	SSA	SSA	VAU	
	VAU	VAU	VAU	VAU	VAU	VAU	VAU	VAU	VAU	VAU	VAU	
	SQT	SQT	SQT	SQT	SQT	SQT	SQT	SQT	SQT	IOS	VAU	
	SSA		SSA		SSA		SSA		SSA			
	CP	CP	SOM	SOM	SOM	SOM	DUN/FLA	FLA	SSA	VAU		
	SOM	SOM	DUN/FLA	DUN/FLA	FLA	FLA	SOM	SOM	FLA	FLA		

Analyse du besoin et spécifications

Expression du besoin

Acteur	Besoin	Priorité
Admin/Manager	Créer/modifier les plannings, gérer les membres, notifier les changements	Haute
Techniciens	Consulter leur planning personnel, recevoir les notifications	Haute
Directeurs de cliniques	Consulter les plannings de leur site, recevoir les affectations	Moyenne

Condensé de la demande :

Il faudrait qu'il y ait deux pages, une page pour faire les plannings et une page pour les consulter

La page pour faire les plannings doit être protégé par un login, mdp

Voici les différentes informations :

Personne de l'équipe :

Aurélien VANDAMME

Brahim ZABOUB

Alexandre BOUCHE

Cliniques :

IOS

CBS

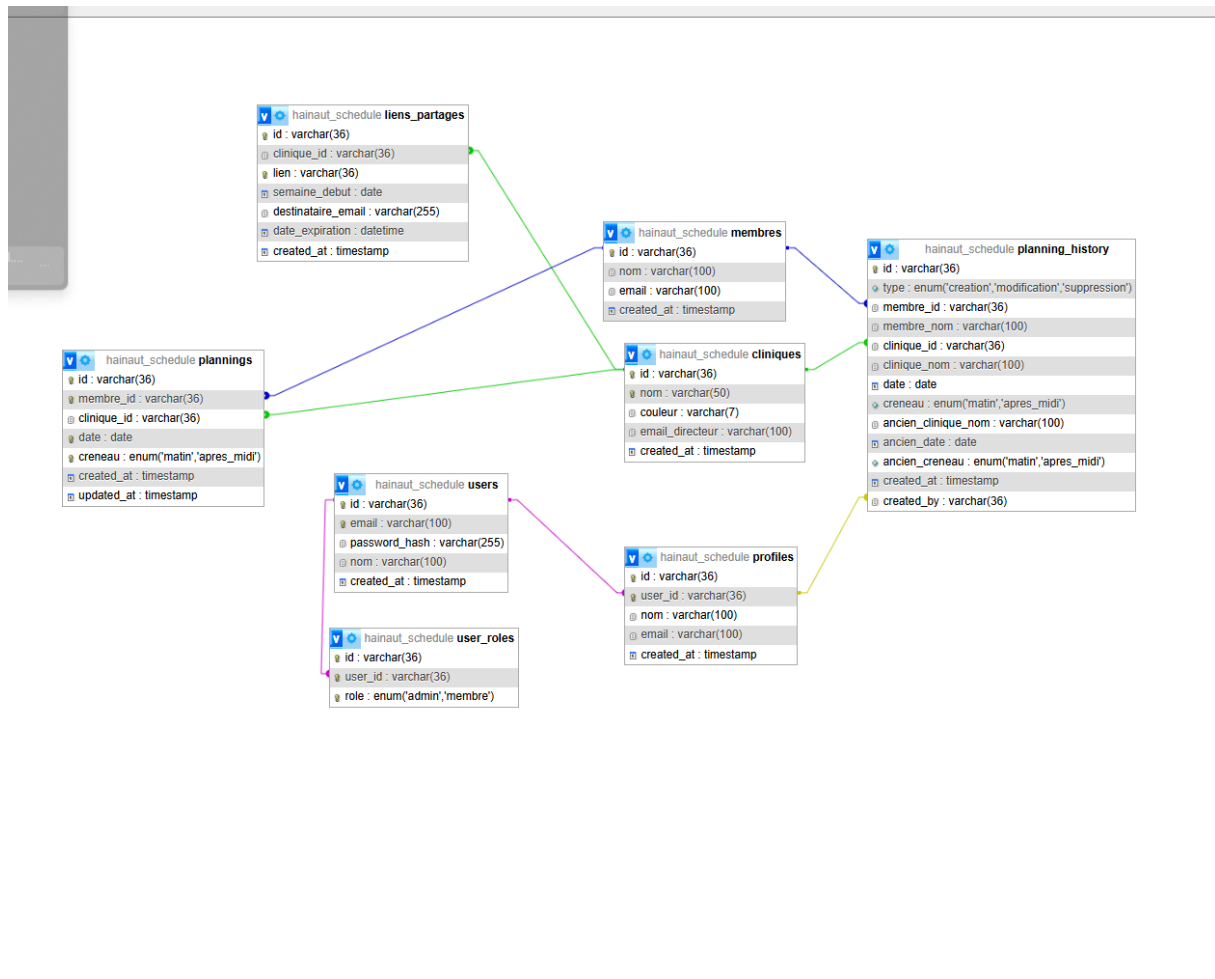
SSA

SQT

VAU

Les différentes cliniques peuvent être positionnées Matin et/ou après-midi sur chaque jour de la semaine sauf Samedi/Dimanche

Modélisation des données



Spécifications Fonctionnelles

- ✓ Authentification sécurisée (JWT + BCrypt)
- ✓ Gestion des rôles : admin / membre / directeur
- ✓ Création de plannings : membre + clinique + créneau (matin/après-midi)
- ✓ Contrainte métier : pas de planning samedi/dimanche
- ✓ Prévention des conflits : unicité membre/créneau
- ✓ Consultation : vue semaine/mois avec filtres
- ✓ Notifications : email + in-app pour modifications
- ✓ Partage : génération de liens sécurisés avec expiration
- ✓ Export : PDF (impression) + CSV (réutilisation Excel)

Contraintes Techniques

- **Pas de planning le week-end** (Samedi/Dimanche).
- **Pas de double réservation** pour un membre à un créneau donné.
- **Déploiement sur votre serveur** (Node.js - express / MySQL)
- **Délai** : MVP fonctionnel en 6 semaines

Environnement technique

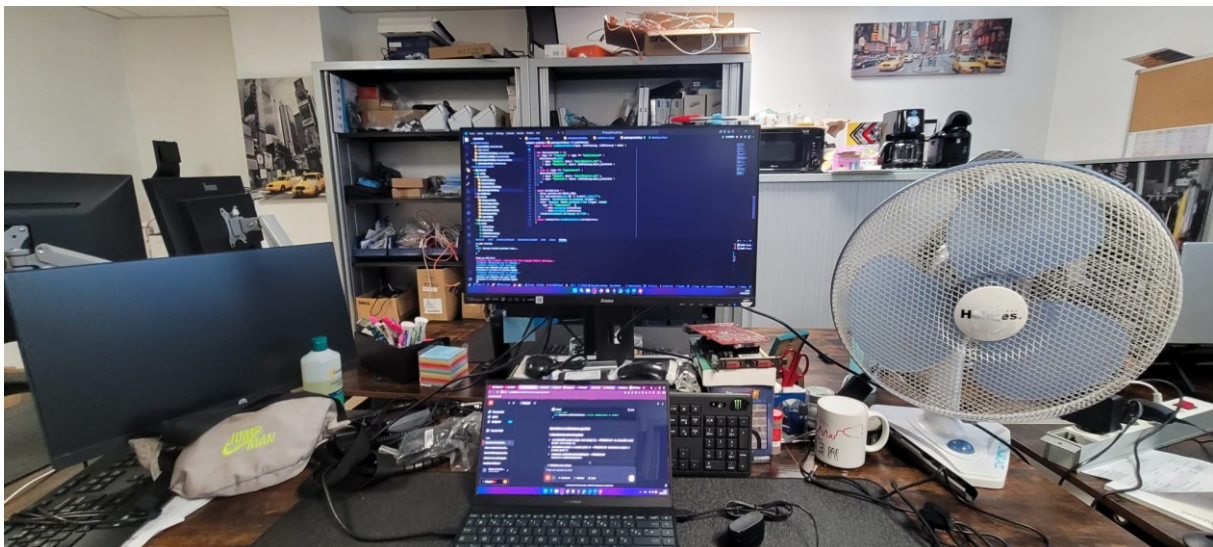
Stack Technique

Composant	Technologie	Détails
Frontend	React.js + NEXT.js	Le calendrier faite Tailwind CSS pour le design.
Backend	Node.js + Express	API REST pour la gestion des plannings, notifications et partage.
Base de données	MySQL	Tables pour utilisateurs, membres, cliniques, plannings, notifications, liens partagés.
Authentification	BCrypt + JWT	Hachage des mots de passe et tokens JWT pour les sessions.
Emails	Nodemailer	Envoi d'emails pour les notifications et partages.
Hébergement	Votre serveur	PM2 pour Node.js, Nginx/Apache pour le frontend.

DevOps & Outils

- ✓ Git : Versioning avec stratégie GitFlow (main/develop/feature/*)
- ✓ PM2 : Process manager pour Node.js en production
- ✓ Postman/Thunder Client : Tests manuels des endpoints API
- ✓ MySQL Workbench : Modélisation et administration BDD

Poste de développement



Réalisation technique

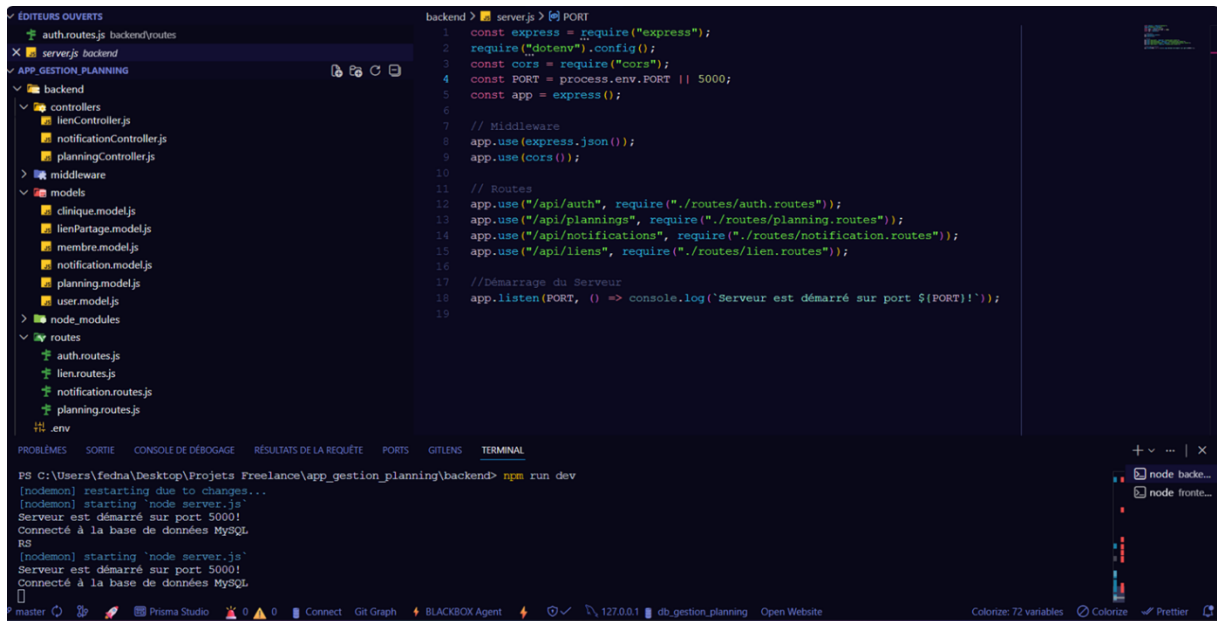
Développement Backend (Semaine 1)

Authentification et Sécurité

Implémentation de l'authentification JWT et hachage BCrypt. Protection des routes via middleware.

Gestion des Plannings (CRUD complet)

Conformément au cahier des charges, le développement d'un socle backend fonctionnel a nécessité une semaine. L'ensemble des routes de l'API de gestion de planning est désormais opérationnel



Définition des routes dans le Backend dans le fichier server.js :

```
// =====  
// ROUTES PRINCIPALES  
// =====  
app.use("/api/auth", authRoutes);  
app.use("/api/membres", membresRoutes);  
app.use("/api/cliniques", cliniquesRoutes);  
app.use("/api/plannings", planningsRoutes);  
app.use("/api/planning-history", planningHistoryRoutes);  
app.use("/api/share-links", shareLinksRoutes);
```

Afin de vérifier si ces routes fonctionnent, j'utilise Postman ou Thunder Client. Voici un exemple de test pour l'affichage des membres :

Home Workspaces API Network

POST Signup POST Login GET notifications POST LienPartage

gestion_planning / plannings

GET http://localhost:5000/api/membres

Docs Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 Ctrl+Alt+P to Ask AI

Body Cookies Headers (9) Test Results

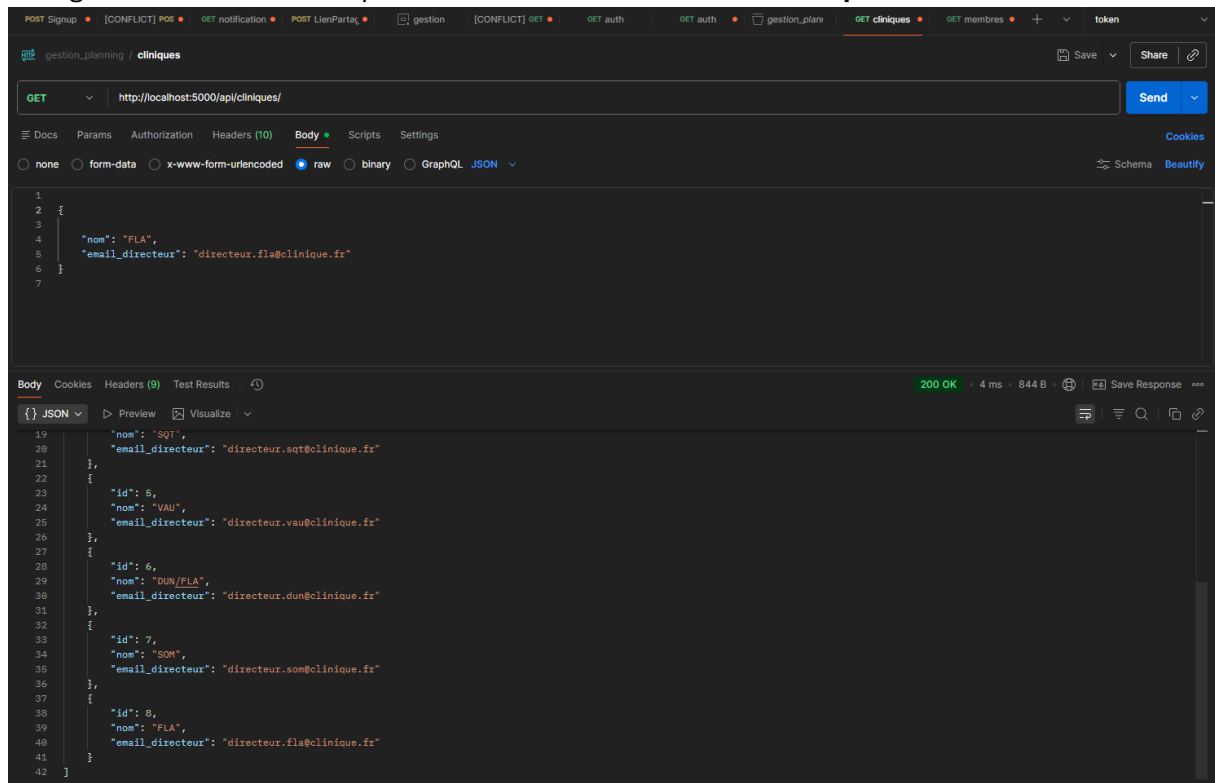
{ } JSON Preview Visualize

```
1 [
2   {
3     "id": 1,
4     "nom": "Aurélien VANDAMME"
5   },
6   {
7     "id": 2,
8     "nom": "Brahim ZABOUB"
9   },
10  {
11    "id": 3,
12    "nom": "Alexandre BOUCHE"
13  },
14  {
15    "id": 4,
16    "nom": "Florian LICOUR"
17  }
18 ]
```

Files BETA

Cloud View Find and replace Console Terminal

J'ai également rencontré un problème lors du test de la route **cliniques** :



```
1 GET http://localhost:5000/api/cliniques/
2
3 {
4   "nom": "FLA",
5   "email_directeur": "directeur fla@clinique.fr"
6 }
7

Body Cookies Headers (9) Test Results
200 OK 4 ms 844 B Save Response

19 {
20   "nom": "SQT",
21   "email_directeur": "directeur.sqt@clinique.fr"
22 },
23 {
24   "id": 5,
25   "nom": "VAU",
26   "email_directeur": "directeur.vau@clinique.fr"
27 },
28 {
29   "id": 6,
30   "nom": "DUN/ELA",
31   "email_directeur": "directeur.dun@clinique.fr"
32 },
33 {
34   "id": 7,
35   "nom": "SOM",
36   "email_directeur": "directeur.som@clinique.fr"
37 },
38 {
39   "id": 8,
40   "nom": "FLA",
41   "email_directeur": "directeur fla@clinique.fr"
42 }
```

Le problème était que j'avais mal écrit email_directeur (j'avais écrit "email_drecteur"), ce qui a créé cette erreur.

Développement Frontend (Semaine : 2-4)

Authentification

J'ai initié le développement frontend en utilisant la stack Next.js, Tailwind CSS, FullCalendar et l'API Fetch. Lors des tests d'authentification, une erreur indiquant des identifiants incorrects a été retournée. Bien qu'il s'agisse d'une erreur fonctionnelle, cette réponse a confirmé l'établissement de la communication entre le frontend et le backend.

Côté BACKEND :

```
39
40 // Fonction pour la connexion
41 exports.login = async (req, res) => {
42   try {
43     const { email, password } = req.body;
44
45     // Trouve l'utilisateur par son email
46     const user = await users.findByEmail(email);
47     if (!user) {
48       return res
49         .status(400)
50         .json({ message: "Email ou mot de passe incorrect." });
51     }
52
53     // Compare les mots de passe
54     const isMatch = await bcrypt.compare(password, user.password);
55     if (!isMatch) {
56       return res
57         .status(400)
58         .json({ message: "Email ou mot de passe incorrect." });
59     }
60
```

Côté FRONTEND :



Connexion

Email ou mot de passe incorrect.

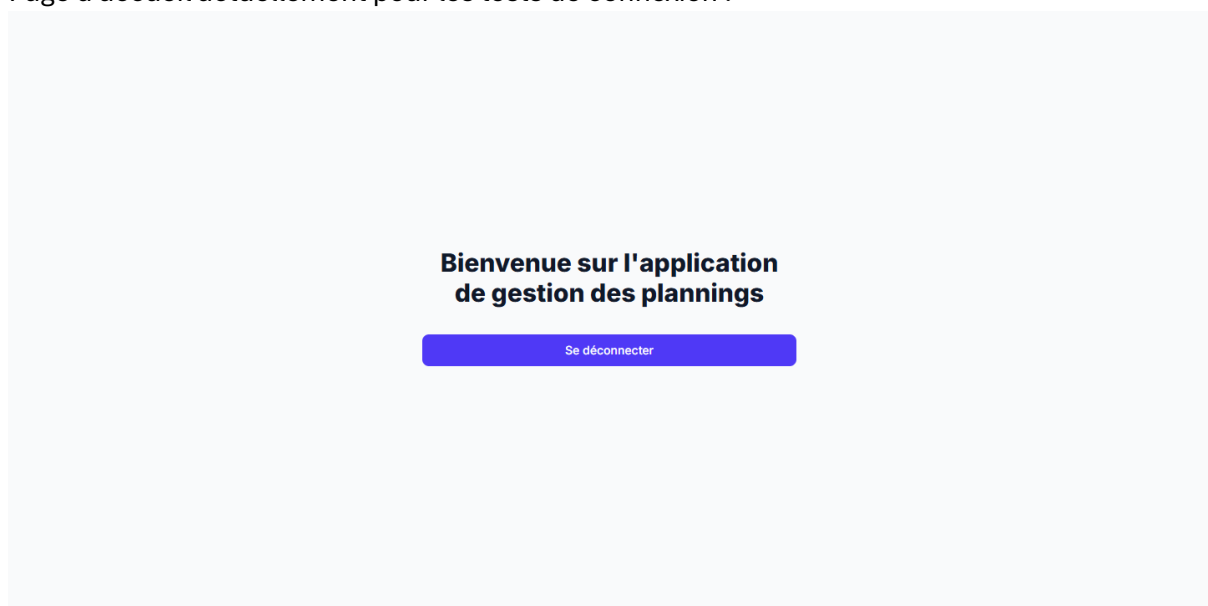
[Pas encore de compte ? S'inscrire](#)

Le problème a été résolu : Le problème venait des intercepteurs que j'avais ajoutés pour inclure les tokens JWT et gérer les erreurs. Cette configuration empêchait les requêtes de connexion et d'inscription d'aboutir.

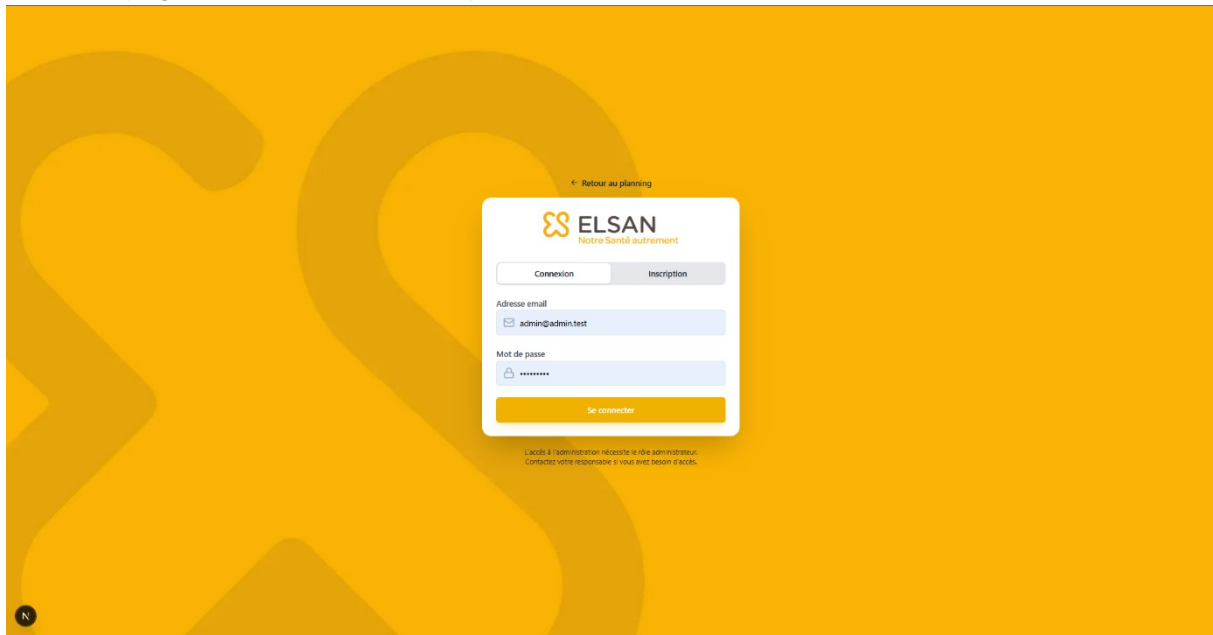
Voici un exemple d'intercepteur que j'avais ajouté :

```
api.interceptors.request.use (  
  (config) => {  
    const token = localStorage.getItem("token");  
    if (token) { config.headers.Authorization = `Bearer ${token}` ; }  
    return config;  
  }, (error) => {  
    return Promise.reject(error);  
  }  
);
```

Page d'accueil actuellement pour les tests de connexion :



Nouvelle page de connexion et inscription :



Code de la page de connexion :

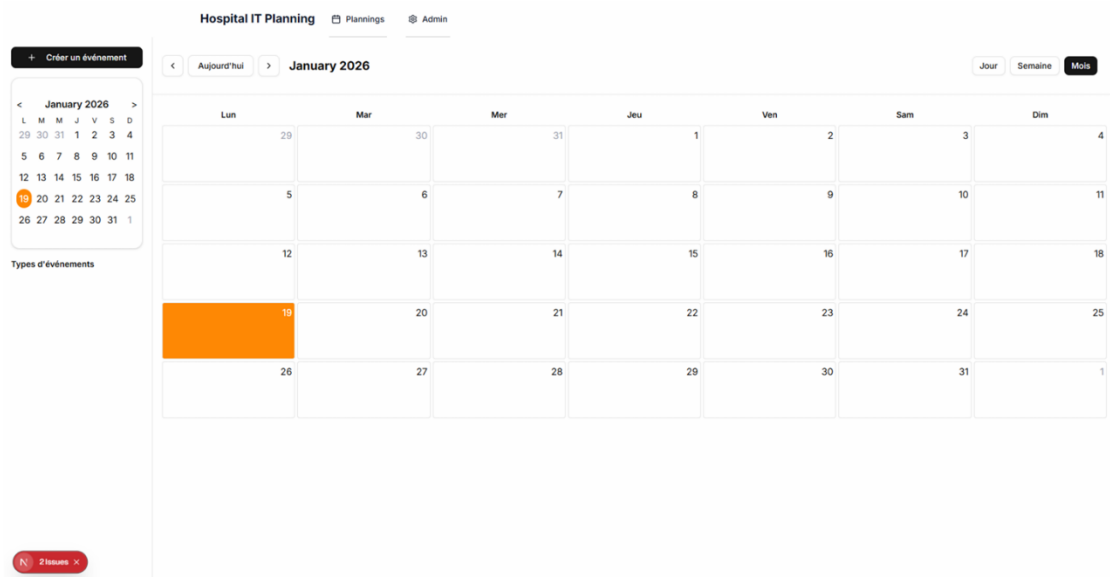
```
15 const handleChange = (e: React.ChangeEvent<HTMLInputElement>) => {
21   };
22
23   const handleSubmit = async (e: React.FormEvent) => {
24     e.preventDefault();
25     try {
26       const response = await fetch("/api/auth/login", {
27         method: "POST",
28         headers: {
29           "Content-Type": "application/json",
30         },
31         body: JSON.stringify(formData),
32       });
33
34       if (!response.ok) {
35         throw new Error("Email ou mot de passe incorrect.");
36       }
37
38       const data = await response.json();
39       localStorage.setItem("token", data.token);
40       router.push("/");
41     } catch (error) {
42       console.error("Erreur lors de la connexion:", error);
43       setError("Email ou mot de passe incorrect.");
44     }
45   };
46
47   return (
48     <div
49       className="min-h-screen flex items-center justify-center py-12 px-4 sm:px-6 lg:px-8"
50       style={{
51         backgroundImage: "url('/bg-elsan-jaune.png')",
52         backgroundSize: "cover",
53         backgroundPosition: "center",
54       }}
55     >
56     <div className="max-w-md w-full space-y-8 relative z-10 bg-white bg-opacity-90 p-8 rounded-xl shadow-2xl">
57       <div className="flex justify-center">
58         <Image
59           src="/elsan-logo.png" // - Assure-toi que le fichier existe dans public/
60           alt="Logo"
61           width={600}
62         />

```

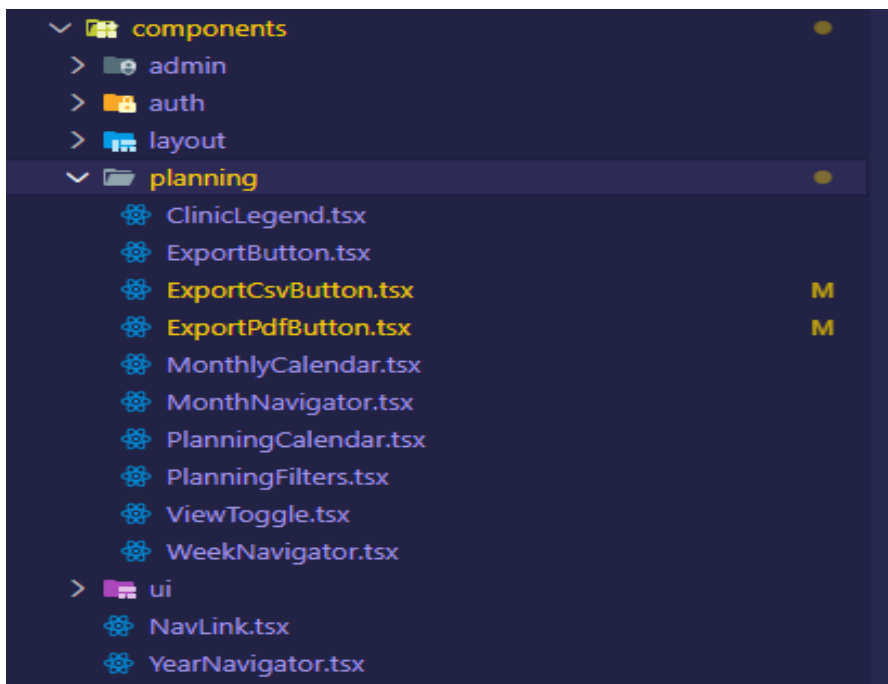
Interface de Planning

L'implémentation du module de planning est fonctionnelle. Actuellement, je dois encore résoudre des problèmes liés à l'affichage des données dynamiques (issues de l'API) et à l'adaptabilité du design (responsive) sur les différents écrans.

Image du calendrier encore avec fullCalendar



J'avais rencontré des problèmes avec FullCalendar alors j'ai décidé de refaire manuellement les composants pour créer un calendrier, ainsi avoir le design et la bonne structure, d'implémentation du backend, ce que fullCalendar n'accepte pas.



Liste des Erreur Rencontré :

Problème rencontré lors de la création d'un planning, alors qu'il n'est pas affiché en frontend :

```
Auto Minimax M2.1 (Free) 342 { /* Body */
PROBLÈMES 23 SORTIE CONSOLE DE DÉBOGAGE PORTS GITLENS RÉSULTATS DE LA REQUÊTE POSTMAN CONSOLE
PS C:\Users\fedna\Desktop\Projets Freelance\hainaut-schedule\backend> npm run dev
> nodemon src/server.js

[nodemon] 3.1.11
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node src/server.js`
[dotenv@17.2.3] injecting env (15) from .env -- tip: 🔒 prevent building .env in docker:
[dotenv@17.2.3] injecting env (0) from .env -- tip: ✅ audit secrets and track compliance
[dotenv@17.2.3] injecting env (0) from .env -- tip: 🕵️ add observability to secrets: http
API running on http://localhost:5000
Connecté à la base de données MySQL
{ status: 409, message: 'Ce créneau est déjà occupé à cette clinique' }
{ status: 409, message: 'Ce créneau est déjà occupé à cette clinique' }
{ status: 409, message: 'Ce créneau est déjà occupé à cette clinique' }
{ status: 409, message: 'Ce créneau est déjà occupé à cette clinique' }
{ status: 409, message: 'Ce créneau est déjà occupé à cette clinique' }
{ status: 409, message: 'Ce créneau est déjà occupé à cette clinique' }
}
```

Erreur de syntaxe SQL rencontrée :

```
PS C:\Users\fedna\Desktop\Projets Freelance\hainaut-schedule\backend> npm run dev
code: 'ER_PARSE_ERROR',
errno: 1064,
sql: '\n' +
  'SELECT\n' +
  '  p.id,\n' +
  '  p.membre_id,\n' +
  '  p.clinique_id,\n' +
  '  p.creneau,\n' +
  '  p.date,\n' +
  '  p.created_at,\n' +
  '  p.updated_at,\n' +
  'FROM plannings p\n' +
  'WHERE p.id = '100'\n' +
  '\n',
sqlState: '42000',
sqlMessage: "You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'FROM plannings p\n' +
  'WHERE p.id = '100'' at line 9"
}
{ status: 409, message: 'Member already booked for this slot' }
{ status: 409, message: 'Member already booked for this slot' }
{ status: 409, message: 'Planning conflict: creneau already assigned' }
{ status: 409, message: 'Planning conflict: creneau already assigned' }
[nodemon] restarting due to changes...
[nodemon] starting `node src/server.js`
[dotenv@17.2.3] injecting env (15) from .env -- tip: 🗃️ backup and recover secrets: https://dotenvx.com/ops
[dotenv@17.2.3] injecting env (0) from .env -- tip: 🔐 encrypt with Dotenvx: https://dotenvx.com
[dotenv@17.2.3] injecting env (0) from .env -- tip: 🔐 encrypt with Dotenvx: https://dotenvx.com
blackboxai/adapt-backend-joins Prisma Studio 22 Connect Git Graph BLACKBOX Agent RHDA analysis has failed 127.0.0.1 db_gestion_planning Open Website Generate Commit Message
```

Résultats finaux de l'interface :

Vue principale de l'application, Vue Semaine depuis la page Consultation :

Planning de l'équipe
Consultez les plannings de l'équipe informatique

16 Février 2026 - 22 Février 2026
Semaine de travail

Légende cliniques
CBS CP DUN FLA ISO REC SOM
SQT SSA VAU

Membre	Lundi 16 Fevr.	Mardi 17 Fevr.	Mercredi 18 Fevr.	Jeudi 19 Fevr.	Vendredi 20 Fevr.
Alexandre BOU...	ISO Matin	SOM Matin	FLA Matin	CBS Matin	Matin
Alexandre BOU...	ISO Après-Midi	DUN Après-Midi	SLA Après-Midi	SQT Après-Midi	SQT Après-Midi
Aurélien VAND...	CP Matin	DUN Matin	ISO Matin	FLA Matin	Matin
Aurélien VAND...	CP Après-Midi	CBS Après-Midi	Après-midi	FLA Après-Midi	Après-midi
Brahim ZABOUB	Matin	Matin	Matin	Matin	Matin
Brahim ZABOUB	Après-midi	Après-midi	Après-midi	Après-midi	Après-midi
Florian LICOUR	Matin	Matin	Matin	Matin	Matin
Florian LICOUR	Après-midi	Après-midi	Après-midi	Après-midi	Après-midi

Vue Mois de la page Consultation, avec la vue sans le week-end (samedi/dimanche) qui sont grisés :

Planning de l'équipe
Consultez les plannings de l'équipe informatique

Février 2026

Légende cliniques
CBS CP DUN FLA ISO REC SOM
SQT SSA VAU

Lun	Mar	Mer	Jeu	Ven	Sam	Dim
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9 A SQT A SOM F REC F REC	10 A FLA F REC A SOM F REC	11 A SQT	12 A SQT	13 A VAU	14	15
16 A ISO A CP A ISO A CP	17 A DUN A CBS A SOM A DUN	18 A SSA A FLA A ISO A FLA	19 A SQT A FLA A CP A FLA	20 A SQT	21	22
23	24	25	26	27	28	1

Page de notification de l'application :

ELSAN
Historique des modifications de planning

Activité récente
37 modification(s)

- + Création il y a 33 minutes
LICOUR Florian affecté(e) à VAU
mercredi 11 février 2026 Après-midi
- + Création il y a 33 minutes
LICOUR Florian affecté(e) à VAU
mercredi 11 février 2026 Après-midi
- + Création il y a environ 17 heures
LICOUR Florian affecté(e) à CP
mardi 10 février 2026 Matin
- + Création il y a environ 17 heures
LICOUR Florian affecté(e) à CP
mardi 10 février 2026 Matin
- + Création il y a environ 17 heures
BOUCHE Alexandre affecté(e) à SQT
mardi 10 février 2026 Après-midi
- + Création il y a environ 17 heures
VANDAMME Auré lien affecté(e) à REUP
lundi 9 février 2026 Matin
- + Création il y a environ 17 heures
VANDAMME Auré lien affecté(e) à REUP
lundi 9 février 2026 Matin

Page de partage de lien :

ELSAN
Administration
Gérez les plannings, membres, cliniques et liens de partage

Planning Membres Cliniques Partage

Liens de partage

- + Générer un lien
- CBS
Semaine du 8 février 2026
Expire le 11 mars 2026 à 15:03
- SSA
Semaine du 8 février 2026
Expire le 16 févr. 2026 à 15:02

Page gestion des cliniques :

The screenshot shows the 'Administration' page for 'Cliniques' in the ELSAN system. The page header includes the ELSAN logo and navigation links for Consultation, Notifications, Journal, Admin, and Déconnexion. The main content area is titled 'Administration' and includes a sub-header 'Gérez les plannings, membres, cliniques et liens de partage'. Below this, there are tabs for Planning, Membres, Cliniques, and Partage. The 'Cliniques' tab is active, displaying a list of clinics with their names and corresponding edit and delete icons. A green '+ Ajouter' button is located at the top right of the list.

Clinique	Modifier	Supprimer
CBS		
CP		
DUN/FLA		
FLA		
IOS		
RECUP		
SOM		
SQT		

Page de gestion des membres :

The screenshot shows the 'Administration' page for 'Membres de l'équipe' in the ELSAN system. The page header includes the ELSAN logo and navigation links for Consultation, Notifications, Journal, Admin, and Déconnexion. The main content area is titled 'Administration' and includes a sub-header 'Gérez les plannings, membres, cliniques et liens de partage'. Below this, there are tabs for Planning, Membres, Cliniques, and Partage. The 'Membres' tab is active, displaying a list of team members with their names and corresponding edit and delete icons. A green '+ Ajouter' button is located at the top right of the list.

Membre	Modifier	Supprimer
BOUCHE Alexandre		
LICOUR Florian		
VANDAMME Aurélien		
ZABOUB Brahim		

Fonctionnalités d'Export

Export en PDF du planning :

Planning - Semaine du 9 février 2026
Exporté le 9 février 2026 à 09:20

Membre	Lundi 9 févr.	Mardi 10 févr.	Mercredi 11 févr.	Jeudi 12 févr.	Vendredi 13 févr.
AB Alexandre ROUICH	IOS Matin	Matin	Matin	SSA Matin	Matin
	Après-midi	CBS Après-Midi	SQT Après-Midi	Après-midi	VAU Après-Midi
AV Aurélien VANDAM	SSA Matin	VAU Matin	Matin	CBS Matin	Matin
	Après-midi	Après-midi	IOS Après-Midi	Après-midi	SQT Après-Midi
BZ Rrahim ZAROUIR	Matin	VAU Matin	Matin	SQT Matin	Matin
	SSA Après-Midi	Après-midi	IOS Après-Midi	Après-midi	CBS Après-Midi
FL Florian LICOUR	CBS Matin	Matin	VAU Matin	SSA Matin	Matin
	Après-midi	IOS Après-Midi	Après-midi	Après-midi	SQT Après-Midi

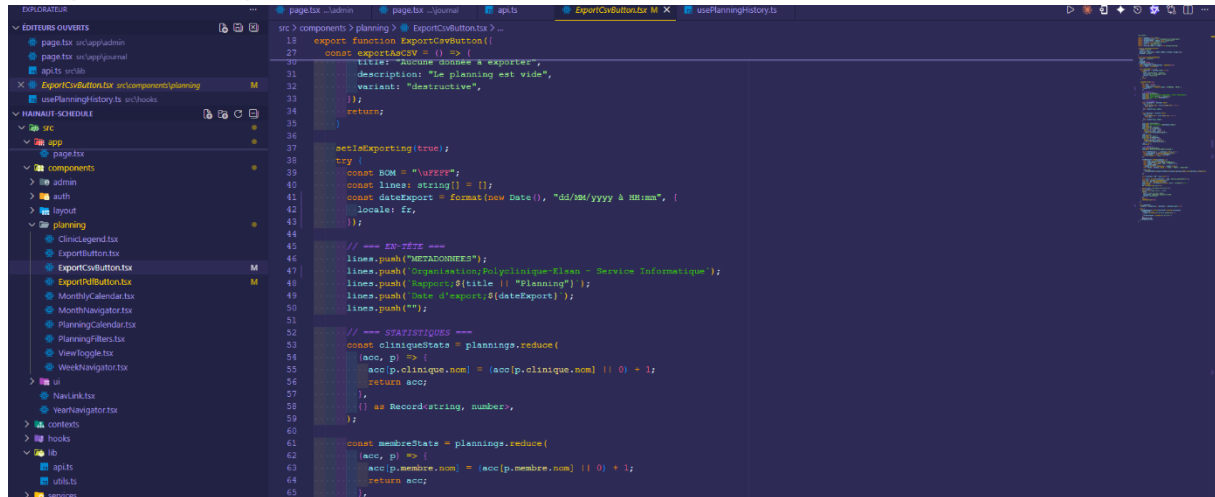
Code du composant ExportPdfButton.tsx qui gère les styles et l'affichage du PDF :

```

src > components > planning > ExportPdfButton.tsx > ...
19 export function ExportPdfButton() {
20   const captureElement = async () => {
21     const canvas = await html2canvas(targetRef.current, {
22       logging: false,
23     });
24   };
25   return canvas;
26 };
27
28 const exportAsPDF = async () => {
29   setIsExporting(true);
30   try {
31     const canvas = await captureElement();
32     const imgData = canvas.toDataURL("image/png");
33     const imgWidth = canvas.width;
34     const imgHeight = canvas.height;
35
36     // Calculate PDF dimensions (A4 landscape or portrait based on aspect ratio)
37     const isLandscape = imgWidth > imgHeight;
38     const pdf = new jsPDF({
39       orientation: isLandscape ? "landscape" : "portrait",
40       unit: "mm",
41       format: "A4",
42     });
43
44     const pdfWidth = pdf.internal.pageSize.getWidth();
45     const pdfHeight = pdf.internal.pageSize.getHeight();
46
47     // Add title if provided
48     if (title) {
49       pdf.setFontSize(16);
50       pdf.setTextColor(30, 41, 59);
51       pdf.text(title, 14, 15);
52       pdf.setFontSize(10);
53       pdf.setTextColor(100, 116, 139);
54       pdf.text(
55         `Exporté le ${format(new Date(), "d MMMM yyyy à HH:mm", { locale: fr })}`,
56       );
57     }
58   } catch (error) {
59     console.error("Erreur lors de l'exportation PDF:", error);
60   }
61   setIsExporting(false);
62 }
63
64 export { ExportPdfButton };

```

Composant de l'export en format CSV :



```
15 export function ExportCsvButton({
16   exportancev = () => {
17     title: "aucune donnée à exporter",
18     description: "Le planning est vide",
19     variant: "destructive",
20   }) {
21   }
22 }
23
24 // --- EXPORT ---
25
26 // --- EXPORT ---
27
28 // --- EXPORT ---
29
30 // --- EXPORT ---
31
32 // --- EXPORT ---
33
34 // --- EXPORT ---
35
36 // --- EXPORT ---
37
38 // --- EXPORT ---
39
40 // --- EXPORT ---
41
42 // --- EXPORT ---
43
44 // --- EXPORT ---
45
46 // --- EXPORT ---
47
48 // --- EXPORT ---
49
50 // --- EXPORT ---
51
52 // --- EXPORT ---
53
54 // --- EXPORT ---
55
56 // --- EXPORT ---
57
58 // --- EXPORT ---
59
60 // --- EXPORT ---
61
62 // --- EXPORT ---
63
64 // --- EXPORT ---
65
```

Support et Infrastructure

Interventions Terrain

Visite et intervention dans les blocs Opératoires de l'Hôpital, avec la collaboration du service du BioMed, j'ai dû intervenir sur une imprimante qui ne parvenait pas à se connecter au VLAN Imprimante qui lui était dédié.







Déploiement et Gestion d'Incident

Incident Critique

Suite à une erreur de manipulation ayant entraîné la suppression d'une partie significative du projet, j'ai pu restaurer l'intégralité des fonctionnalités. La combinaison des données de la base et d'une sauvegarde externe a permis une récupération complète du projet au cours du week-end.

Mise en Production

- Configuration de PM2 pour assurer le redémarrage automatique du service Node.js.
- Sauvegarde de la base de données avant mise en production.

Compétences développées

Compétences Techniques

- **Développement Fullstack** : Maîtrise de la chaîne complète (Base de données → API → Interface).
- **Sécurité Applicative** : Gestion des sessions (JWT), hachage (BCrypt), protection des routes.
- **Base de Données** : Modélisation relationnelle MySQL, optimisation des requêtes, contraintes d'intégrité.
- **Outils DevOps** : Utilisation de PM2 pour la production

Compétences Méthodologiques

- **Gestion de Projet** : Respect d'un délai court (6 semaines) avec priorisation des fonctionnalités (MVP).
- **Debugging** : Analyse de logs, isolation des bugs (ex : problème d'interceptors Axios), correction de typos SQL.
- **Documentation** : Rédaction de commentaires de code, schémas d'architecture et guide d'installation.

Compétences Humaines (Soft Skills)

- **Communication** : Vulgarisation technique pour les utilisateurs non-experts (techniciens, directeurs de clinique).
- **Autonomie** : Recherche de solutions techniques sans assistance constante.
- **Gestion du Stress** : Récupération de données et de code à la suite d'une suppression accidentelle lors du développement.

Conclusion et perspectives

Bilan Personnel

Ce stage a consolidé mes compétences en développement web dans un contexte exigeant. J'ai pu vérifier que la théorie apprise en formation (architecture MVC, sécurité, base de données) s'applique directement en entreprise, mais avec des contraintes supplémentaires (délais, legacy, utilisateurs réels).

Apports pour l'Entreprise

- Outil opérationnel réduisant les conflits de planning et améliorant la communication inter-sites.
- Documentation technique complète facilitant la maintenance par l'équipe interne.
- Base de code modulaire permettant des évolutions futures sans refonte majeure.

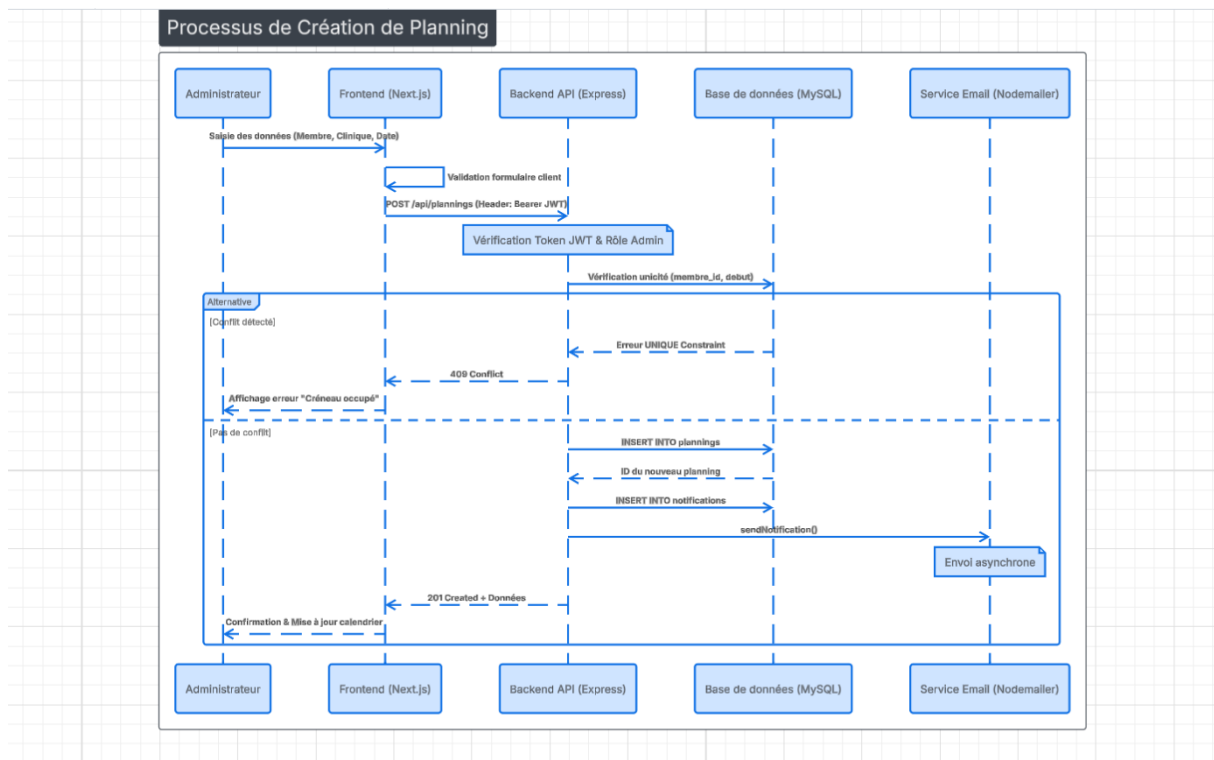
Perspectives Professionnelles

Cette expérience confirme mon souhait d'évoluer vers un poste de Développeur Fullstack. Je souhaite approfondir mes connaissances en DevOps et en architecture cloud pour concevoir des systèmes encore plus résilients.

Annexes

Annexe A : Diagramme de séquence : Création de planning avec notification

Ce diagramme illustre le flux de données lors de la création d'un créneau de planning par un administrateur, incluant la persistance en base de données et l'envoi asynchrone de la notification.

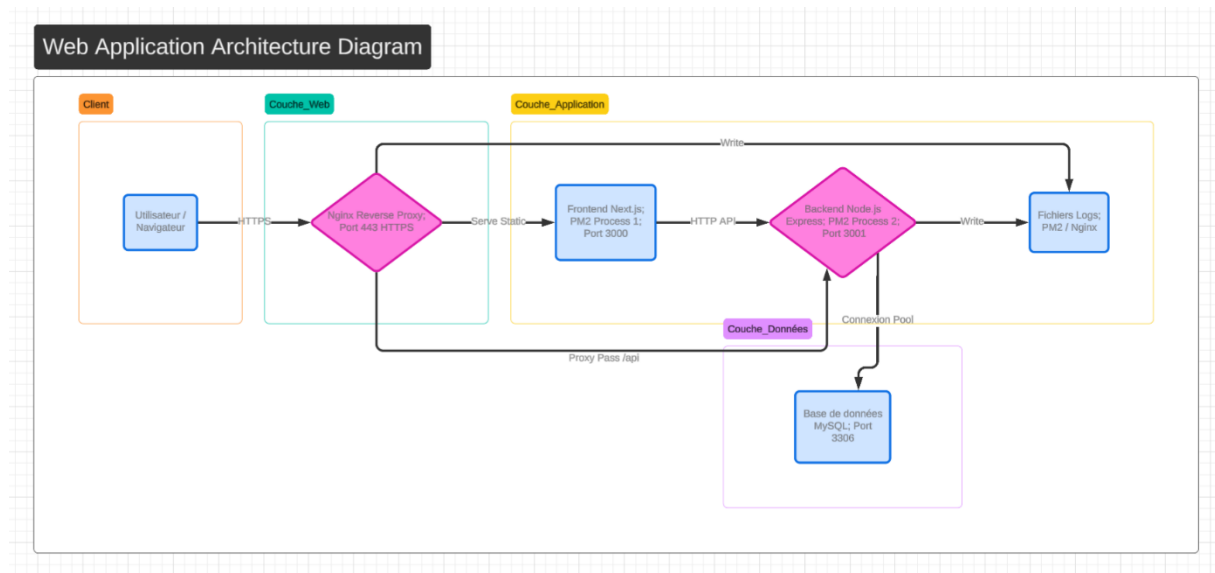


Légende :

- **Flèches pleines** : Requête synchrone.
- **Flèches pointillées** : Réponse ou action asynchrone.
- **Alt** : Condition alternative (Conflit ou Succès).

Annexe B Architecture de déploiement

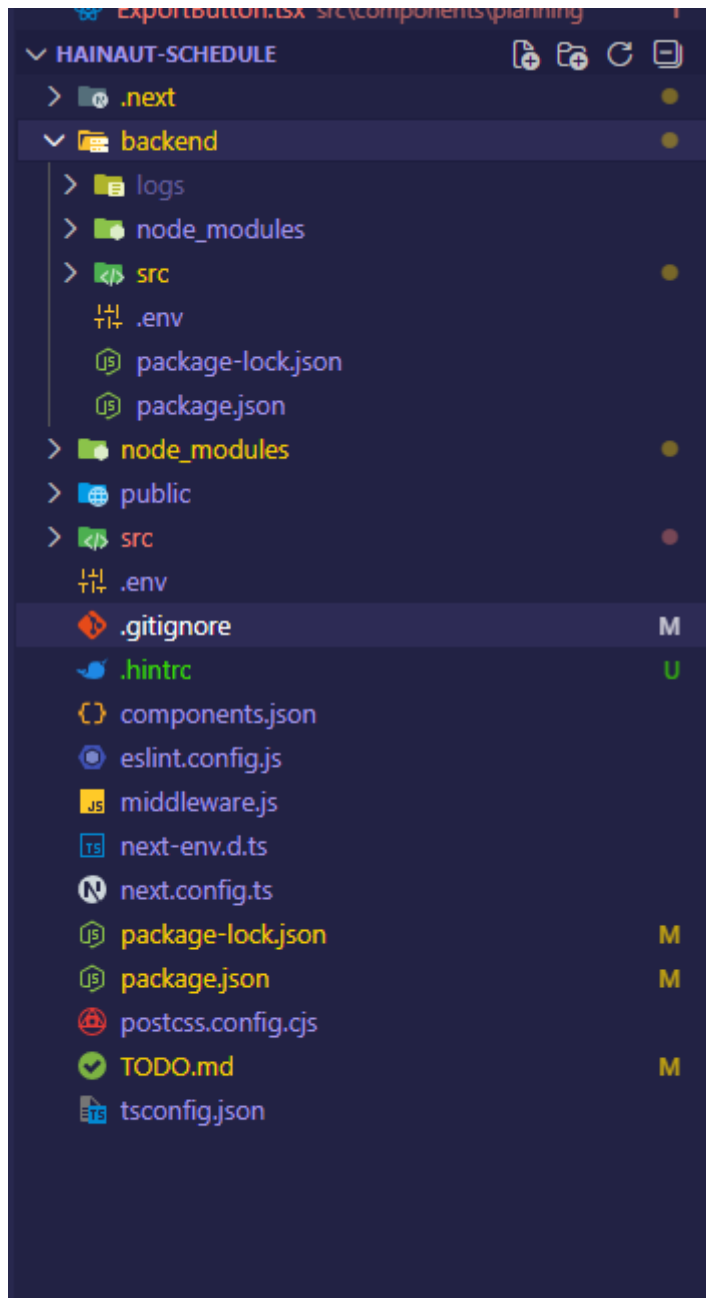
Schéma d'infrastructure montrant l'interaction entre les composants lors de l'exécution en production sur le serveur interne de l'hôpital.



Détails de l'architecture :

- **Nginx :** Gère la terminaison SSL (HTTPS), la compression Gzip et la répartition des requêtes (statiques vers Next.js, API vers Express).
- **PM2 (Process Manager 2) :** Assure le maintien en condition opérationnelle des processus Node.js (redémarrage automatique en cas de crash, gestion des logs).
- **MySQL :** Stockage persistant des données, sécurisé par mot de passe fort et accès local uniquement (bind-address 127.0.0.1).
- **Sécurité :** Le backend n'est pas exposé directement sur Internet/Intranet, il est protégé derrière Nginx.

Annexe C : Base de code modulaire



Annexe D : Glossaire

- **JWT** : JSON Web Token, standard pour l'authentification sécurisée.
- **MVP** : Minimum Viable Product, version minimale fonctionnelle d'un produit.
- **SSO**: Single Sign-On, authentication unique.
- **API REST** : Interface de programmation permettant l'interaction entre services.
- **BioMed** : Le service Bio Médicale gère tout ce qui touche au bloc opératoire
- **MySQL** : Système de gestion de base de données relationnelle utilisé pour stocker les plannings et utilisateurs.
- **Next.js** : Framework React permettant le rendu côté serveur et une optimisation des performances web.
- **Nginx** : Serveur web haute performance utilisé ici comme reverse proxy pour sécuriser les accès.
- **Node.js** : Environnement d'exécution JavaScript côté serveur pour développer l'API backend.
- **PM2** : Gestionnaire de processus pour Node.js assurant la disponibilité de l'application en production.
- **Git** : Système de gestion de versions utilisé pour suivre les modifications du code source.
- **HTTPS** : Protocole de communication sécurisée chiffrant les échanges entre le client et le serveur.
- **Middleware** : Fonction exécutée entre la réception d'une requête et l'envoi de la réponse pour vérifier les permissions.
- **SQL** : Langage standard utilisé pour interroger et manipuler les données dans la base de données MySQL.
- **Token** : Jeton numérique transmis dans les en-têtes HTTP pour prouver l'identité de l'utilisateur connecté.
- **BCrypt** : Algorithme de hachage sécurisé utilisé pour protéger les mots de passe dans la base de données.
- **Backend** : Partie serveur de l'application, responsable de la logique métier et de l'accès aux données.
- **CSV** : Format de fichier texte utilisé pour représenter des données tabulaires, exportable vers Excel.
- **Frontend** : Partie cliente de l'application, visible par l'utilisateur et exécutée dans le navigateur.