

Rapport de stage – Ecole Militaire de Saumur



Gabriel Autret

1^{er} Année BTS SIO SLAM

Lycée Chevrollier



Autret
Gabriel
1TSIO1 SLAM

Sommaire :

Sommaire :	2
Présentation de l'entreprise.....	3
Qu'est-ce que l'atelier 3D ?	6
Concernant l'équipe de l'atelier 3D	6
Missions confiées par l'entreprise.....	7
Mes démarches	8
Les difficultés rencontrées et leurs solutions.....	16
La suite	20
Bilan	21



Présentation de l'entreprise

EMS (Écoles Militaire de Saumur),
situé à la 1757 Av. du Maréchal Foch, 49400 Saumur
Mon tuteur est Mr. Gilles Duquenoy, Ingénieur civil de la défense.
Je suis aussi encadré par 2 alternants, Timéo et Yaël, spécialisés dans le web (en master dev. FullStack).

Contacts du tuteur :

- Mail : gilles.duquenoy@intradef.gouv.fr
- Num (direct) : 02 41 83 71 20

Je réalise mon stage au sein des **EMS**, un ensemble d'organisme de formation de l'armée de Terre. Les **EMS** ont pour mission de former les cadres militaires dans plusieurs domaines, notamment :

- L'EC (École de Cavalerie),
- L'ERAT (École du Renseignement de l'Armée de Terre),
- Le CIANRBC (Centre Interarmées de Défense Nucléaire Radiologique Biologique et Chimique),
- L'EEM (École d'État-Major),

L'établissement regroupe environ 600 personnels permanents et forme 4 000 stagiaires par an.

Je suis affecté à la **DFS** (Division de la Formation de Spécialité de l'Ecole de Cavalerie), qui est chargé de la formation technique des cadres dans le domaine du combat blindé, notamment autour du char Leclerc.

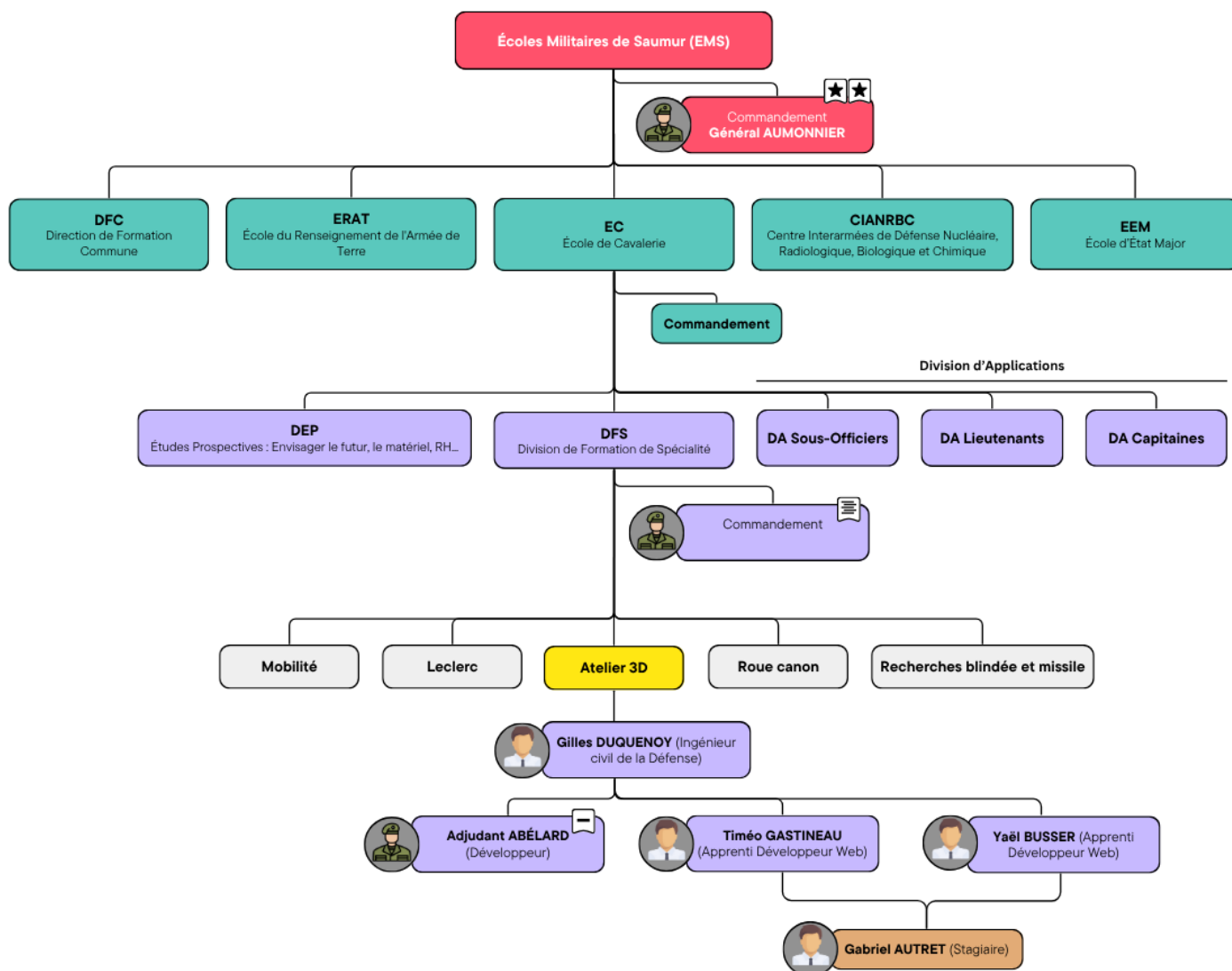
Pour l'instruction ce service utilise :

- Des salles de cours
- Des simulateurs
- Des travaux pratiques sur matériel



Voici un organigramme représentant la DFS et l'équipe avec laquelle je travail :

Organigramme des EMS





Voici le bâtiment de la DFS :





Qu'est-ce que l'atelier 3D ?

- Il s'agit de la création d'environnement virtuels pour les simulateurs de chars
- Du développement d'outils pour aider à l'instruction :
 - Cartographie
 - Outils Web
 - Impression 3D
- Du développement de composant de simulateur et de simulateurs

Concernant l'équipe de l'atelier 3D

Dans l'équipe, nous sommes 7 au total.

- Il y a 2 permanents :
 - Gilles Duquenoy (*Ingénieur civil de la défense*)
 - L'adjudant Jérôme Abélard (*Développeur*)
- L'équipe se compose aussi de 2 alternants :
 - Yaël Busser (*master dev. FullStack*)
 - Timéo Gastineau (*master dev. FullStack*)
- Et enfin, 3 stagiaires :
 - Théo Denys (*BTS CIEL*)
 - Elvann Jolivel (*BTS CIEL*)
 - Moi, Gabriel Autret (*BTS SIO*)



Missions confiées par l'entreprise

Les élèves et les instructeurs utilisent plusieurs applications web notamment pour les notes, les Quiz en temps réel (similaire à Kahoot ou Socrative) et pour de la documentation. Ces applications sont disponibles via différents identifiants. Cela engendre une perte de temps pour les utilisateurs.

- Ma première mission a consisté à centraliser l'accès aux différentes applications web. L'objectif était de relier ces applications à un identifiant unique, permettant aux utilisateurs de ne se connecter qu'une seule fois pour accéder à l'ensemble des outils disponibles. Cette solution, développée en PHP, visait à simplifier la navigation, améliorer la gestion des accès et faire gagner du temps aux utilisateurs.
- Afin de faire évoluer ces applications, une mission préparatoire m'a été confiée afin d'acquérir les compétences techniques nécessaires à la bonne réalisation de la suite de mon stage. Elle portait sur l'apprentissage de Node.js, un environnement JavaScript utilisé aussi côté serveur et l'utilisation de React pour structurer le code des 2 côtés.
- Nous avons ensuite envisagé de faire évoluer aussi, l'interface utilisateur des applications. En effet, la technologie utilisée actuellement pour le style visuel des applications est peu attirant (Bootstrap), il faudra envisager sur une technologie qui pourra la remplacer.



Mes démarches

Pour mener à bien ma première mission, j'ai d'abord pris le temps de comprendre le fonctionnement existant : la structure des fichiers, leur nommage, leur emplacement, ainsi que leur contenu.

Une fois familiarisé avec l'environnement, j'ai analysé la base de données pour mieux en cerner la logique. Dès que l'ensemble m'était clair, j'ai mis en place une base de données de test à l'aide de MariaDB, une solution open source, afin d'éviter toute modification directe sur l'environnement de production. Pour cela, j'ai utilisé l'interface phpMyAdmin, ce qui m'a permis de gérer la base plus facilement : création des tables, insertion de données d'essai, et ajustement de la structure afin qu'elle corresponde aux nouveaux besoins.

Ensuite, en utilisant PHP, j'ai dupliqué la page de connexion d'une des applications existantes afin de l'adapter à ma solution. Cette nouvelle version redirige l'utilisateur vers une page d'accueil centralisée, appelée dashboard. Cette interface permet de naviguer facilement entre les différentes applications, sans devoir passer par chaque page de connexion, tout en conservant les informations de session grâce à la gestion côté serveur via PHP.

Après avoir conçu ce tableau de bord, j'ai établi la connexion avec la base de données afin que les identifiants soient reconnus et que l'accès soit sécurisé.

Enfin, j'ai adapté les autres applications pour qu'elles utilisent ce nouveau système d'authentification centralisé.

En rappelant que tout cela utilise la technologie PHP pour faire la gestion à la base de données et pour le backend en général



Voici à quoi ressemble la page de connexion et le dashboard :

Accueil



Identifiez-vous

Identifiant

Mot de passe

Entrez

Bienvenue, Gabriel Autret !

Accueil



FlashNote



BuzzR



Felix



FlashEval



Documentation



EAD 3D



Le site côté frontend utilise seulement du HTML, CSS (bootstrap) et JavaScript

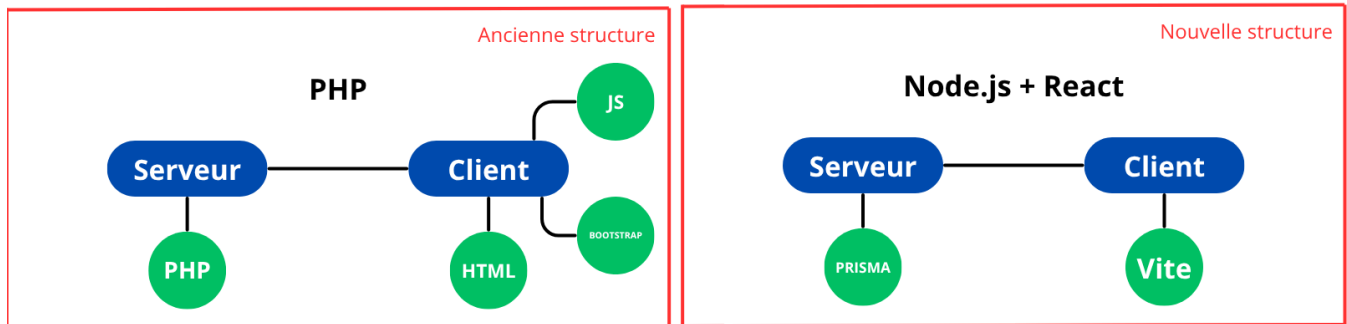


Après cette première mission, la suite du projet nécessitait une modernisation complète du site, tant au niveau de l'interface que des technologies utilisées. Le PHP natif étant progressivement remplacé par Node.js, un exercice pratique avait été préparé par les alternants afin de découvrir et maîtriser ce nouvel environnement.

L'exercice consistait à créer une page de chat en ligne en utilisant différentes technologies : Node.js et Express pour gérer le côté serveur et créer des API (interfaces permettant d'échanger des données entre le client et le serveur), Prisma pour faciliter l'interaction avec la base de données, ainsi que React et Vite pour concevoir l'interface utilisateur côté client.



Voici à un schéma représentant les anciennes et nouvelles technologies utilisées :



Ce type de projet possède une architecture bien plus structurée qu'un simple site statique. Contrairement à un projet PHP classique, l'utilisation de Node.js avec Prisma permet d'éviter d'écrire manuellement des requêtes SQL en utilisant des requêtes préparées, plus sûres et plus faciles à maintenir.

En effet, Prisma génère un fichier copiant l'entièreté de la base de données, ce qui permet d'utiliser des requêtes préparées plus lisibles et sécurisées.

Travailler avec React m'a permis de comprendre le fonctionnement des interfaces dynamiques. Contrairement à une page HTML classique, React met automatiquement à jour les éléments visibles lorsqu'une donnée change, sans recharger la page entière.



Voici à quoi ressemble la page de chat :

Chat en temps réel

Chat

Statistiques

Timéo

Salut

09:08:42

Pascal

Yo

09:08:51

Timéo

ça va?

09:08:55

Pascal

oui et toi

09:06:55

Utilisateurs

T

Timéo

P

Pascal

Votre nom

Votre message

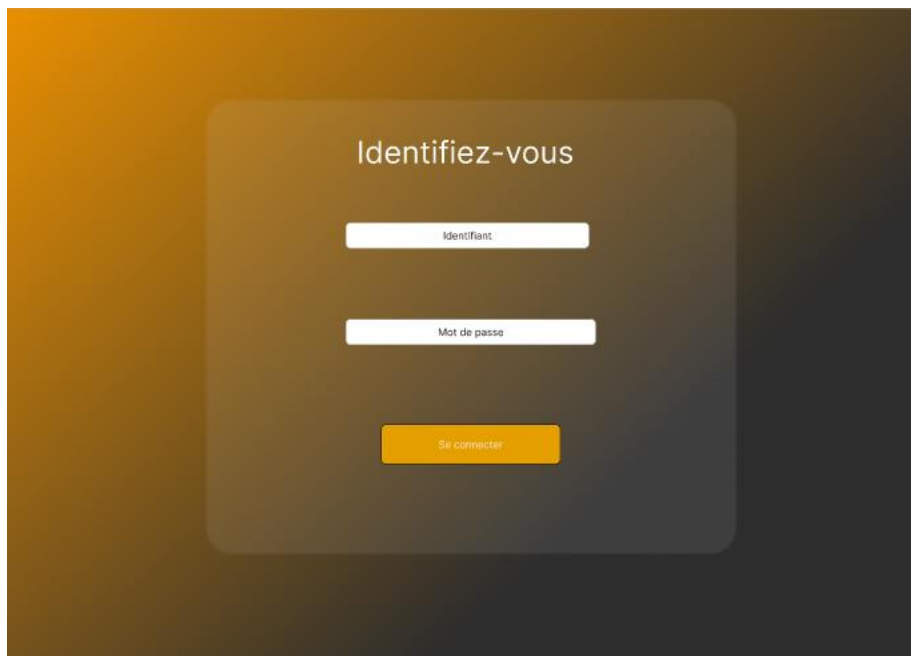
Envoyer

Après avoir mené à bien la phase préparatoire consistant à me familiariser avec Node.js et à mettre en place les bases nécessaires, j'ai pu aborder la mission suivante, moderniser le site.

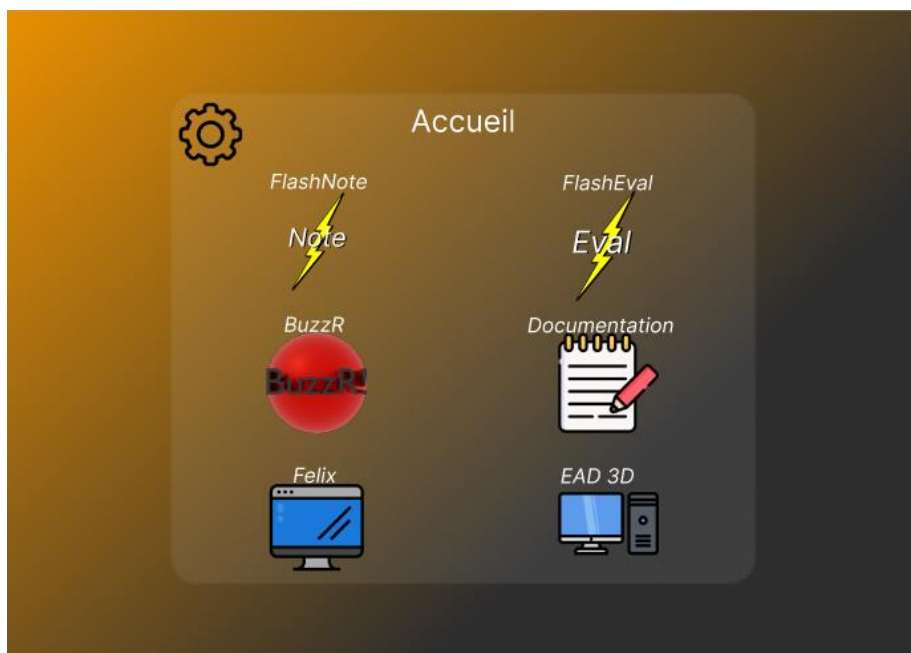
Mais avant de commencer, même si on sait quelle technologie le nouveau site va utiliser, l'interface utilisateur n'a pas encore été conçue. Pour cela j'ai utilisé un logiciel en ligne nommé Figma, ça permet de créer des maquettes de site web avec des prototypes interactifs.



Et donc voici ce que j'ai proposé à l'équipe :



Nouvelle page de connexion



Nouveau Dashboard



J'ai aussi rajouté quelques fonctionnalités, notamment le fait de voir les infos personnelles de l'utilisateur et les modifier via l'onglet option (la modification ne sera peut-être pas incluse) :

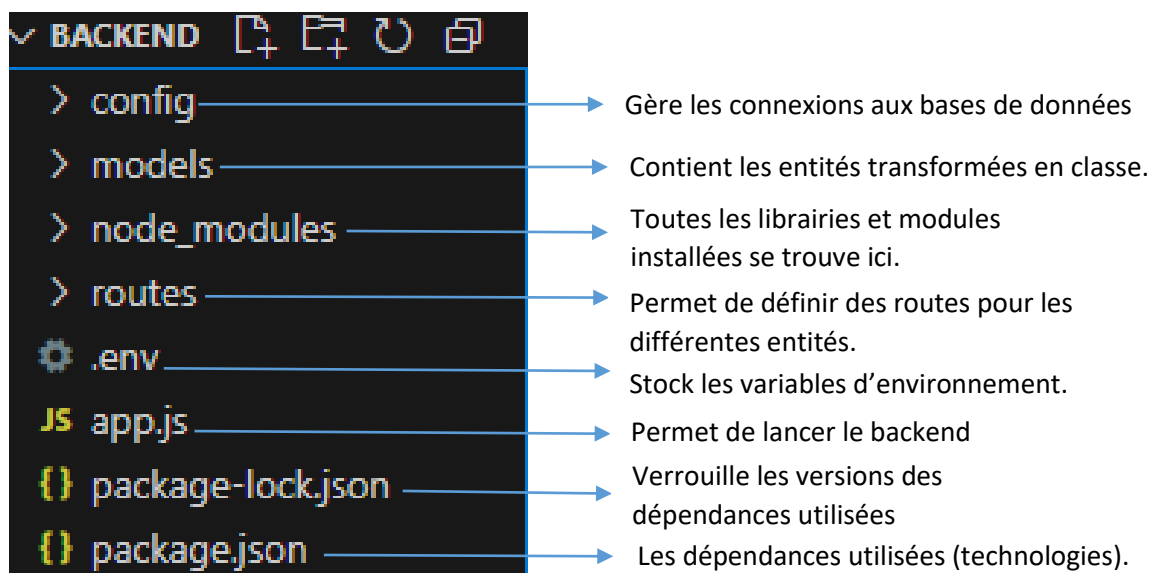
Ce qui est affiché lorsqu'on consulte les infos de l'utilisateur

Ce qui est affiché lorsqu'on souhaite modifier les infos de l'utilisateur



Et c'est donc après cela que j'ai commencé à créer le projet, en commençant par le backend et par la connexion à la base de données via un ORM (Objet-Relational Mapping, Prisma en est un).

Après avoir fini l'installation de tout le backend du projet, l'arborescence des fichiers devraient ressembler à ça :



C'est une installation avec l'ORM Sequelize, qui comprend un dossier "models" contenant les différentes entités de la base de données sous forme de classes ou d'objets, ainsi que le dossier "routes" qui définit les routes HTTP de l'entité (GET, POST, PUT, DELETE, etc...). Cette organisation permet de séparer la logique métier (models) de la gestion des requêtes côté serveur (routes), rendant le code beaucoup plus maintenable.



Les difficultés rencontrées et leurs solutions

Problème : Compréhension de la structure du projet

Lors de ma première mission, la principale difficulté a été de comprendre la structure du projet, notamment l'organisation des fichiers et la fonction de chaque dossier.

Solution :

Une fois cette étape franchie, la modification des unités s'est avérée plus simple à réaliser.

Problème : Manque de données dans la base de test

Le plus compliqué a été de modifier les fichiers qui utilisaient l'entièreté de la base de données. En effet, la base de données de test ne contenait que les données pour l'identification, ce qui empêchait certaines fonctionnalités de fonctionner correctement.

Solution :

J'ai dû faire une copie de la base en production pour compléter celle de test. Cela m'a permis d'assurer le bon fonctionnement des fonctionnalités dépendantes de données plus complètes.



Problème : Erreurs liées au serveur WebSocket

Certaines pages utilisaient un WebSocket (qui permet à un serveur et un client d'échanger de manière bidirectionnelle sans devoir rouvrir la connexion à chaque fois). Travaillant en local, je n'avais pas accès au serveur WebSocket, ce qui provoquait des erreurs sur ces pages.

Solution :

J'ai dû apprendre à différencier les erreurs liées à mes modifications de celles causées par l'absence du serveur WebSocket. Cela m'a permis d'éviter de corriger à tort des parties du code qui étaient en réalité fonctionnelles.

Problème : Difficulté de configuration avec l'ORM

Dans le cadre de la modernisation du site web, j'ai rencontré beaucoup de difficultés à utiliser l'ORM Prisma, notamment au niveau de sa configuration que je n'ai pas réussi à faire fonctionner correctement.

Solution :

J'ai donc opté pour un autre ORM nommé Sequelize. Il n'y a pas de très grandes différences entre les deux : les commandes préparées sont similaires. Cependant, Sequelize engendre une perte de temps par rapport à Prisma, car contrairement à Prisma qui récupère automatiquement la structure de la base de données, avec Sequelize il faut tout recréer manuellement les tables ainsi que les relations entre elles ce qui est particulièrement long pour les bases ayant une structure complexe.



Voici à quoi ça ressemble :

Jointure : (lien entre plusieurs tables)

```
//relation
Stagiaire.belongsTo(Filiere, { foreignKey: 'nomFiliere' });
Filiere.hasMany(Stagiaire, { foreignKey: 'nomFiliere' });

Stagiaire.belongsTo(Grade, { foreignKey: 'nomGrade' });
Grade.hasMany(Stagiaire, { foreignKey: 'nomGrade' });

Historique.belongsTo(Stagiaire, { foreignKey: 'idStagiaire' });
Stagiaire.hasMany(Historique, { foreignKey: 'idStagiaire' });

Historique.belongsTo(Type_event, { foreignKey: 'idEvent' });
Type_event.hasMany(Historique, { foreignKey: 'idEvent' });

Log_stage.belongsTo(Stagiaire, { foreignKey: 'idStagiaire' });
Stagiaire.hasMany(Log_stage, { foreignKey: 'idStagiaire' });

Log_stage.belongsTo(Stage, { foreignKey: 'idStage' });
Stage.hasMany(Log_stage, { foreignKey: 'idStage' });

Stage.belongsTo(Filiere, { foreignKey: 'nomFiliere' });
Filiere.hasMany(Stage, { foreignKey: 'nomFiliere' });

Thematique.belongsTo(Filiere, { foreignKey: 'nomFiliere' });
Filiere.hasMany(Thematique, { foreignKey: 'nomFiliere' });
```



Entité : (élément représenté dans une base)

```
const Stagiaire = sequelize.define('Stagiaire', {  
  idStagiaire: {  
    type: DataTypes.STRING(30),  
    primaryKey: true,  
    allowNull: false  
  },  
  nomGrade: {  
    type: DataTypes.STRING(5),  
    allowNull: false  
  },  
  nomFiliere: {  
    type: DataTypes.STRING(50),  
    allowNull: false  
  },  
  nomStagiaire: {  
    type: DataTypes.STRING(50),  
    allowNull: false  
  },  
  prenomStagiaire: {  
    type: DataTypes.STRING(50),  
    allowNull: false  
  },  
  mdpStagiaire: {  
    type: DataTypes.STRING(100),  
    allowNull: false  
  },  
  estLogge: {  
    type: DataTypes.BOOLEAN,  
    defaultValue: false  
  }  
})
```



La suite

Mission 1 : déploiement du site

Déployer le site sur le serveur de production afin de tester sa stabilité et sa compatibilité en conditions réelles.

Mission 2 : Modernisation de l'aspect des sites

L'utilisation de Bootstrap limite les possibilités de présentation des sites. Une nouvelle technologie par exemple Tailwind ou du CSS natif rendra la maintenabilité plus simple. Ainsi, un nouveau développeur pourra facilement ajouter des fonctionnalités sans être ralenti par le style du site.

Mission 3 : Modernisation fonctionnelle des sites

L'utilisation de technologies de type Node.js, React, d'outils de type ORM permettent aujourd'hui de développer plus rapidement les sites.

Une veille sur les nouvelles technologies permettrait de gagner encore plus de temps en développement.

D'autres idées :

- La mise en place de statistiques, telles que le nombre de quiz réalisés par un élève, son temps de connexion, ou encore sa moyenne aux quiz.
- Rendre les interfaces plus interactives et informatives.



Bilan

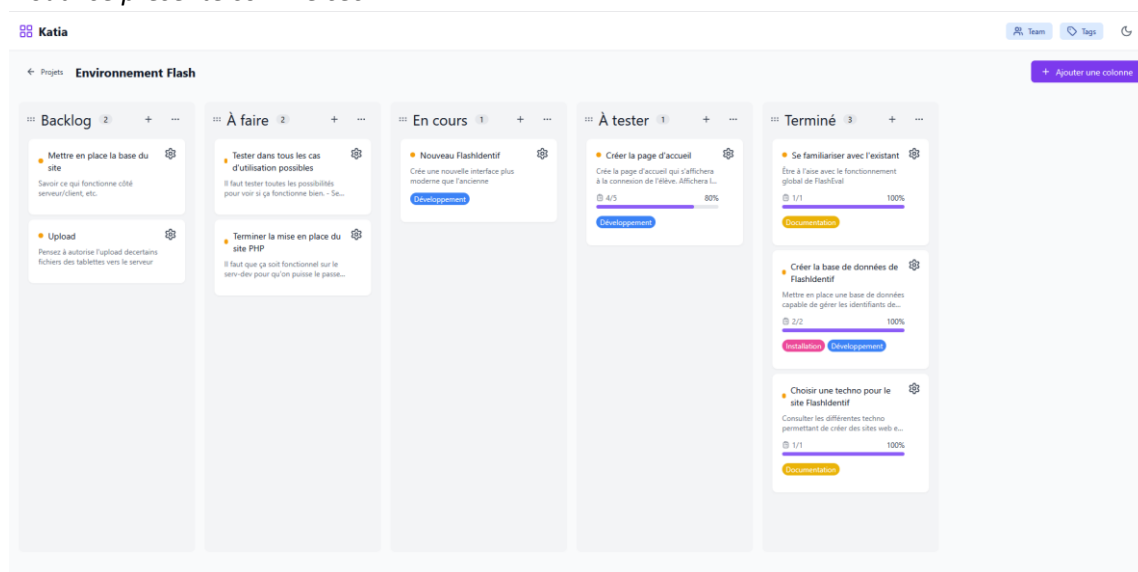
Pendant ces 4 semaines j'ai pu acquérir de nouvelles compétences techniques grâce à l'utilisation de nouveaux outils tels que les ORM ou Node.js et React mais aussi pu renforcer mes compétences avec PHP.

Ce stage m'a permis de consolider les bases acquises durant ma formation, en les mettant en œuvre dans un environnement professionnel, avec des projets concrets.

Au-delà des aspects purement techniques, ces 4 semaines m'ont offert une première immersion dans le fonctionnement d'un service informatique au sein d'une entreprise. J'ai découvert l'importance des relations humaines : l'esprit d'équipe, l'entraide au sein du groupe, ainsi que les moments de convivialité qui contribuent à créer un environnement de travail agréable et motivant.

Durant ce stage, j'ai également pu utiliser des outils privés de gestion de projet, dont Katia, un site créé par les alternants permettant de gérer ses tâches.

L'outil se présente comme ceci :





J'aimerais aussi rappeler que toute l'équipe prenait un moment chaque jour pour faire un résumé des tâches effectuées, afin de savoir où chacun en était.

J'ai également identifié certains axes à améliorer, notamment ma gestion du temps. J'ai parfois sous-estimé le temps nécessaire à l'installation et à la configuration de projets, notamment avec des technologies comme Node.js et Sequelize. Cela m'a appris à mieux planifier mes tâches et à ajuster mes objectifs en conséquence.

Je tiens à remercier toute l'équipe de l'atelier 3D pour son accueil chaleureux, sa disponibilité et son accompagnement tout au long de ce stage.

Un remerciement particulier à Gilles Duquenoy et aux alternants pour leurs patiences, leurs conseils et la confiance qu'ils m'ont accordées dans la réalisation de mes missions.

Je remercie également mon père, sans qui je n'aurais pas eu l'opportunité de réaliser ce stage. Son aide m'a été précieuse pour intégrer cette structure et vivre cette première expérience professionnelle.