



# **RAPPORT DE STAGE**

**BTS SIO Option SLAM Année 2025/2026**

**Ali Clarens**

**Du 8 Décembre 2025 au 23 Janvier 2026**

**Tuteur de stage : Clément Lidar**

# SOMMAIRE

1. REMERCIEMENTS..... Page 3
2. INTRODUCTION ..... Page 4
3. PRÉSENTATION DE L'ENTREPRISE..... Page
4. ENVIRONNEMENT TECHNIQUE ET OUTILS..... Page
5. MISSIONS RÉALISÉE..... Page
6. BILAN ET COMPÉTENCE ACQUISES ..... Page
7. CONCLUSION ..... Page

# Remerciements

Avant d'exposer le détail de mes missions, je tiens à exprimer ma profonde gratitude envers l'ensemble de l'entreprise Syslearn pour leur accueil et la bienveillance dont ils ont fait preuve durant ces semaines d'immersion, tout particulièrement mon maître de stage, Monsieur Clément Lidar, qui m'a accordé sa confiance et a pris le temps de nous transmettre à moi et aux autres stagiaire, son expertise technique avec beaucoup de pédagogie.

Cette expérience au sein de Syslearn a été une étape clé dans mon parcours. Elle m'a permis de me confronter aux exigences réelles du métier de développeur, un univers bien différent de mes précédentes expériences. Grâce à l'autonomie qui m'a été confiée sur des sujets comme le développement Fullstack ou la gestion de bases de données NoSQL, j'ai pu consolider mes acquis (à la base très faible) en TypeScript et MongoDB. Ces compétences nouvelles constituent un socle solide pour l'obtention de mon BTS SIO et pour mon insertion future dans la vie active.

Mes remerciements s'adressent également à l'équipe pédagogique pour la qualité de leur suivi et pour les conseils tout au long de la durée du stage.

Enfin, je ne saurais oublier les autres stagiaires avec qui j'ai partagé ce quotidien, ainsi que mes camarades de classe. Nos échanges et l'entraide mutuelle, notamment lors de la phase finale de rédaction de ce rapport, ont été d'un soutien indispensable.

# Introduction

Actuellement étudiant en deuxième année de Brevet de Technicien Supérieur en Services Informatiques aux Organisations (BTS SIO), au sein de l'option SLAM (Solutions Logicielles et Applications Métier), j'ai été amené à réaliser un second stage en immersion professionnelle afin de valider mes compétences techniques et de découvrir les exigences du métier de développeur.

Après une première expérience de stage l'année passée chez Powerlab, qui m'avait permis de me familiariser avec le support matériel, la maintenance et la relation client, j'ai souhaité pour cette deuxième année m'orienter exclusivement vers la conception logicielle et le développement "Fullstack". C'est dans cette optique que j'ai intégré l'entreprise Syslearn, sous la supervision de mon tuteur, Monsieur Clément Lidar.

L'objectif principal de cette mission était de passer de la théorie apprise en formation à une pratique concrète sur des technologies modernes et professionnelles. Durant ces semaines, j'ai eu l'opportunité de travailler sur le projet Deeelroom, un outil complexe nécessitant une maîtrise fine de langages comme TypeScript et de frameworks comme Next.js. Ce stage ne s'est pas limité à la simple écriture de code ; il a été une véritable immersion dans le cycle de vie d'une application, de l'analyse du besoin fonctionnel à l'administration de bases de données NoSQL via MongoDB Atlas.

Ce rapport détaille les différentes missions qui m'ont été confiées. À travers ce document, je présenterai les outils utilisés, les problématiques rencontrées, ainsi que les solutions techniques que j'ai pu déployer pour répondre aux attentes de l'entreprise, confirmant ainsi mon projet professionnel dans le domaine de l'ingénierie logicielle.

# Présentation de l'entreprise

Syslearn est une entreprise spécialisée dans l'ingénierie logicielle et les solutions de transformation digitale. Son cœur de métier réside dans l'accompagnement des entreprises vers l'optimisation de leurs processus internes grâce à des outils numériques innovants. Contrairement à ma première expérience de stage chez Powerlab, qui se concentrait sur la distribution de matériel et le support technique direct, Syslearn se positionne sur le segment de la conception applicative "sur mesure".

L'entreprise se distingue par une culture de développement moderne, privilégiant les architectures agiles et les technologies de pointe (Stack JavaScript/TypeScript). Elle offre un environnement de travail stimulant où l'autonomie et la veille technologique sont encouragées, permettant ainsi de répondre aux besoins complexes de clients issus de secteurs variés (industrie, service, retail).

Le projet Deeelroom, sur lequel j'ai concentré l'intégralité de mes missions, est une plateforme logicielle avancée hybride, combinant les fonctionnalités d'un CRM (Customer Relationship Management) et d'un agrégateur de veille stratégique.

L'enjeu de Deeelroom est de centraliser toutes les informations critiques concernant un client au sein d'une interface unique. Au-delà de la simple gestion des contacts et des opportunités commerciales, la plateforme intègre un module de "suivi d'actualités métier". Ce module permet aux utilisateurs de surveiller en temps réel les annonces publiques, les rapports financiers ou les articles de presse concernant leurs partenaires ou concurrents. L'objectif final est de permettre une prise de décision plus rapide et mieux informée grâce à la donnée.

# Environnements technique et outils

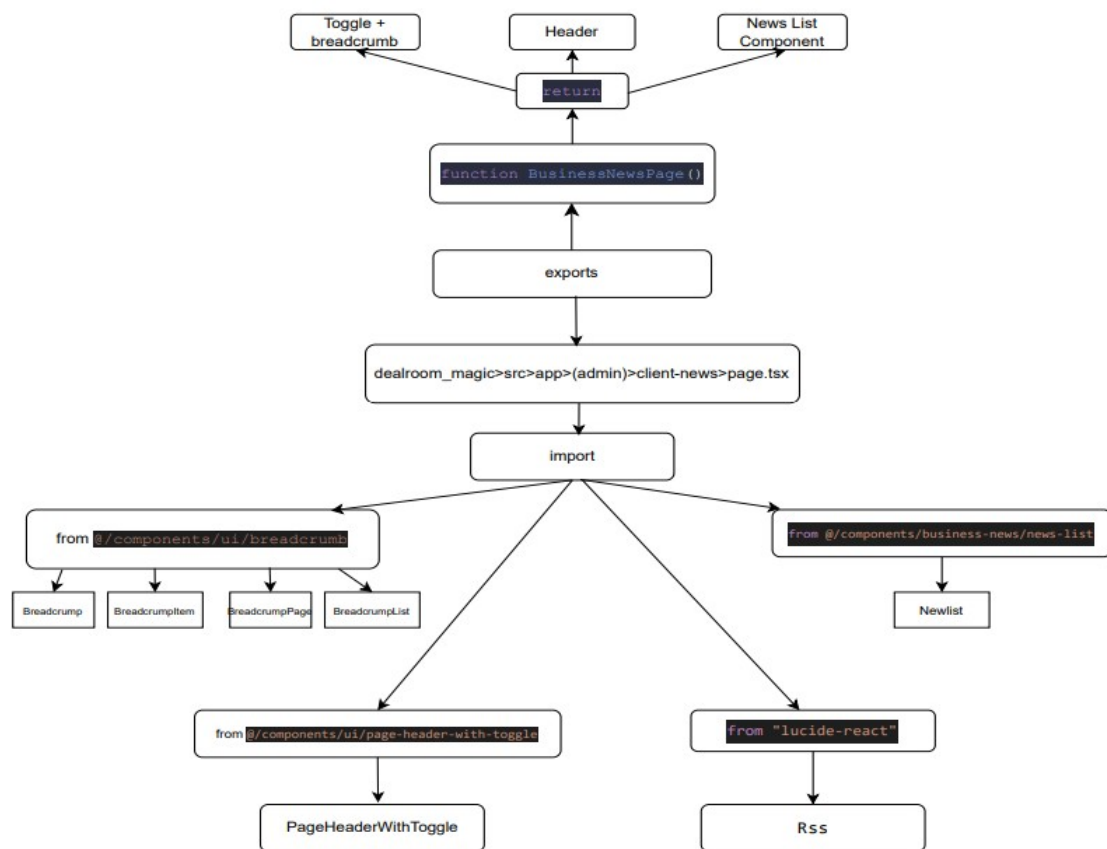
Pour soutenir une telle plateforme, l'environnement technique repose sur des outils et des standards industriels que j'ai dû maîtriser :

- Visual Studio Code (VS Code) : Cet environnement de développement (IDE) a été mon outil quotidien. Je l'ai utilisé non seulement pour l'écriture du code en TypeScript, mais aussi pour le débogage en temps réel et la gestion des versions via le terminal intégré (Git Bash).
- Architecture basée sur les API REST : Le fonctionnement de Deeelroom repose sur une séparation claire entre le Frontend (interface utilisateur) et le Backend (logique serveur). La communication entre les deux se fait via des API REST. J'ai notamment travaillé sur la sécurisation et l'optimisation de ces routes, en utilisant les méthodes HTTP (GET, POST, DELETE) pour manipuler les données.
- MongoDB Compass & Atlas : Pour la gestion des données, nous avons utilisé MongoDB, une base de données NoSQL. MongoDB Atlas nous a servi d'infrastructure Cloud pour l'hébergement du cluster, tandis que MongoDB Compass a été l'outil graphique indispensable pour visualiser la structure des documents JSON, effectuer des requêtes de filtrage complexes et vérifier l'intégrité des données de tests que j'ai insérées (collection customernews).
- GitHub et le travail collaboratif : Pour la gestion du code source, nous avons utilisé GitHub, une plateforme de forge logicielle basée sur Git. Cet outil a été indispensable pour centraliser le développement du projet Deeelroom et faciliter le travail en équipe. GitHub m'a permis de sauvegarder l'historique de mes modifications, de gérer les différentes versions de l'application et de collaborer efficacement avec mon tuteur via des systèmes de "commits" et de "pushs". C'est également à travers cette plateforme que j'ai pu observer les standards de qualité du code appliqués en entreprise.

# Missions réalisée

Les missions à réalisées étaient diverses et variées :

Analyse et architecture de code :



Annotation de code :

## Voici l'annotation du code :

```
useEffect(() => { /* Le useEffect se lance après le rendu du composant */
  fetchNews(true); /* Appel de l'API grâce à fetch ce qui déclenche une "relance" de l'API */
}, [customerId, limit, sourceFilter]); /* Tableau de dépendances */

const fetchNews = async (reset: boolean = false) => { /* Cette fonction appelle l'API (fetch) et met à jour son état en fonction de la valeur du booléen */
  try {
    if (reset) {
      setLoading(true);
      setOffset(0);
    } else {
      setLoadingMore(true);
    }

    const currentOffset = reset ? 0 : offset;

    const params = new URLSearchParams({ /* Création des paramètres de la requête HTTP (car il y'a URL) */
      limit: limit.toString(),
      offset: currentOffset.toString(),
    });

    if (customerId) { /* Si un customerId est fourni, on l'ajoute aux paramètres de la requête */
      params.append("customerId", customerId);
    }

    if (sourceFilter) { /* Si un filtre de source est appliqué, on l'ajoute aux paramètres de la requête */
      params.append("sourceType", sourceFilter);
    }

    const response = await fetch(`/api/business-news?${params.toString()}`); /* Appel de l'API avec les paramètres construits */

    if (!response.ok) { /* Si la réponse est différente (!) de ok, on affiche "Erreur lors du chargement des actualités" */
      throw new Error("Erreur lors du chargement des actualités");
    }

    const data: NewsResponse = await response.json();

    if (reset) { /* Mise à jour des données si reset = true */
      setNews(data.news);
    } else {
      setNews((prev) => [...prev, ...data.news]);
    }

    setTotal(data.total); /* Mise à jour du total des éléments de l'API */
    setHasMore(data.hasMore); /* Mise à jour de l'état hasMore en fonction de l'API */
  }
}
```

Ajout dans l'API d'un bouton de suppression de news :

### Étape 1 : Dans le fichier news-card.tsx (src/components/business-news/news-card.tsx)

```
return (
  <Card className="hover:shadow-md transition-shadow h-full flex flex-col relative">
    {onDelete && (
      <Button
        variant="ghost"
        size="sm"
        onClick={() => {
          console.log(`Suppression de la news: ${news._id}`);
          onDelete(news._id);
        }}
        className="absolute top-2 right-2 z-10 h-8 w-8 p-0 text-muted-foreground"
      >
        <X className="h-4 w-4" />
      </Button>
    )}
  </Card>
)
```



```
// List view
return (
  <Card className="hover:shadow-md transition-shadow relative">
    {onDelete && (
      <Button
        variant="ghost"
        size="sm"
        onClick={() => {
          console.log(`Suppression de la news: ${news._id}`);
          onDelete(news._id);
        }}
        className="absolute top-2 right-2 z-10 h-8 w-8 p-0 text-muted-foreground"
      >
        <X className="h-4 w-4" />
      </Button>
    )}
  </Card>
);
```

**Étape 2 : Dans le fichier route.ts (src\app\api\business-news\route.ts)**

```
export async function DELETE(request: NextRequest) {
  try {
    await connectDB();

    // Extraction de l'ID depuis les query params
    const { searchParams } = new URL(request.url);
    const id = searchParams.get("id");

    // Validation de l'ID
    if (!id) {
      return NextResponse.json(
        { success: false, message: "ID manquant dans la requête." },
        { status: 400 }
      );
    }

    // Appel du service
    await CustomerNewsAggregatorService.deleteNews(id);

    return NextResponse.json({ success: true }, { status: 200 });
  } catch (error: any) {
    console.error("API Error:", error);
    return NextResponse.json(
      { success: false, message: error.message || "Erreur interne serveur" },
      { status: 500 }
    );
  }
}
```

Étape 3 : Dans le fichier `customer-news-aggregator.service.ts` (`src\services\customer-news-aggregator.service.ts`)

```
static async deleteNews(id: string): Promise<any> {
  try {
    if (!id) {
      throw new Error("ID manquant pour la suppression.");
    }

    // 1. Appel NewsModel.findByIdAndDelete(id)
    const result = await CustomerNews.findByIdAndDelete(id);

    if (!result) {
      // 2. Erreur claire renvoyée si l'ID n'existe pas dans la base
      throw new Error(`Actualité avec l'ID ${id} non trouvée.`);
    }

    // 3. Renvoyer le résultat de la suppression (le document supprimé)
    return result;
  } catch (error) {
    console.error(
      `Erreur lors de la suppression de l'actualité ${id}:`,
      error
    );
    throw error;
  }
}
```

Étape 4 : Dans le fichier `new-list.tsx` (`src\components\business-news\news-list.tsx`)

```
const handleDeleteNews = async (id: string) => {
  try {
    const response = await fetch(`/api/business-news?id=${id}`, {
      method: "DELETE",
    });

    if (!response.ok) {
      throw new Error("Erreur lors de la suppression de l'actualité");
    }

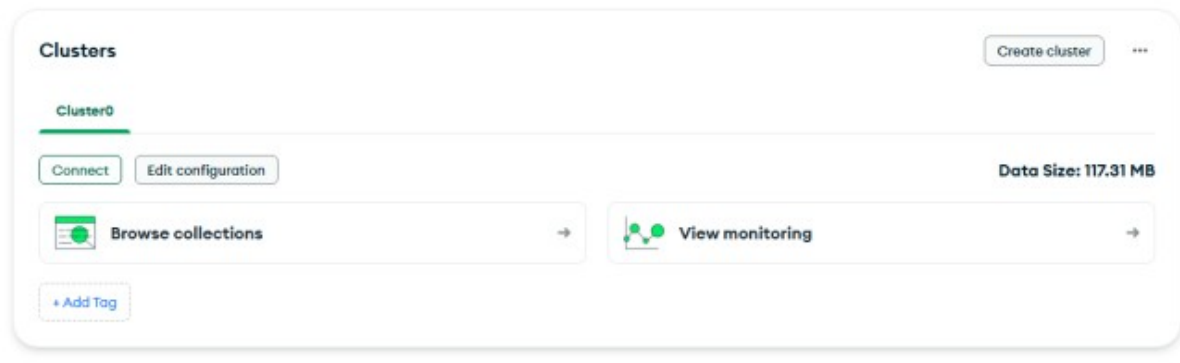
    // Remove the deleted news from the state
    setNews((prev) => prev.filter((item) => item._id !== id));
    setTotal((prev) => prev - 1);

    toast({
      title: "Actualité supprimée",
      description: "L'actualité a été supprimée avec succès.",
    });
  } catch (err) {
    console.error("Erreur:", err);
    toast({
      variant: "destructive",
      title: "Erreur",
      description:
        err instanceof Error
          ? err.message
          : "Impossible de supprimer l'actualité",
    });
  }
};
```

## Initialisation d'une Base De Données NoSQL (MongoDB) :

Tout d'abord il faut aller sur MongoDB Atlas pour avoir accès à notre liste de cluster et pour dans notre cas en créer un.

### Project 0 Overview



Ensuite, durant la création de notre cluster, on nous propose plusieurs méthodes de connexion, dans notre cas on a choisis la connexion par MongoDB Compass. Une fois que c'est fait on arrive sur la page ci-dessous qui contient le « connexion string » qu'il faut conserver et auquel on doit insérer un mot de passe qui nous servira à nous connecter à notre base de donnée depuis notre code en l'implémentant dans notre dossier .env.local

## Connect to Syslearn

✓

✓

3

Set up connection security

Choose a connection method

Connect

### Connecting with MongoDB Driver

#### 1. Select your driver and version

We recommend installing and using the latest driver version.

Driver	Version
Node.js	6.7 or later

#### 2. Install your driver

Run the following on the command line

```
npm install mongodb
```

[View MongoDB Node.js Driver installation instructions.](#)

#### 3. Add your connection string into your application code

Use this connection string in your application

☐ View full code sample ☒ Show Password ⓘ

```
mongodb+srv://cali_db_user:zHGDMN4YjUVNIHd@sysLearn.p9v3iwj.mongodb.net/?appName=Syslearn
```

The password for **cali\_db\_user** is included in the connection string for your first time setup. This password will not be available again after exiting this connect flow.

#### RESOURCES

[Get started with the Node.js Driver](#)

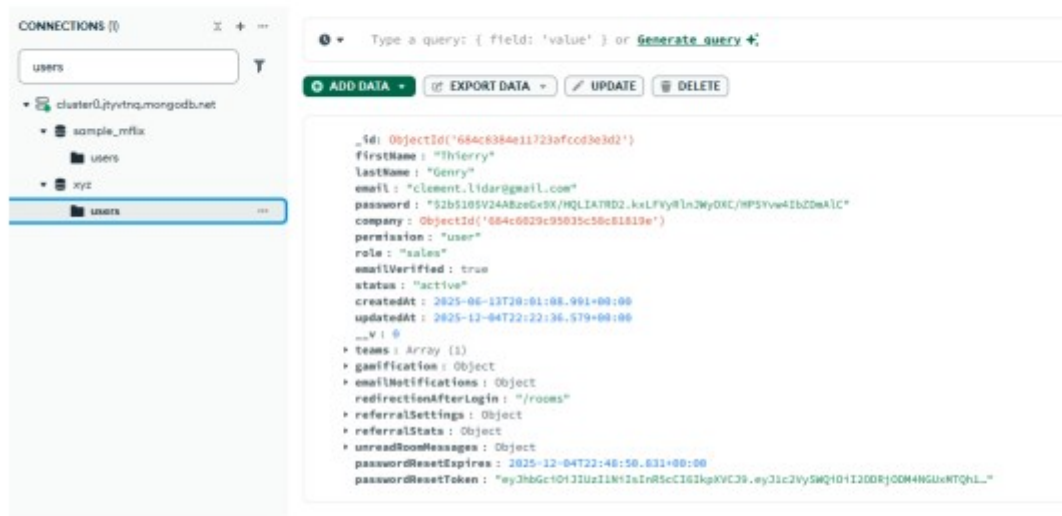
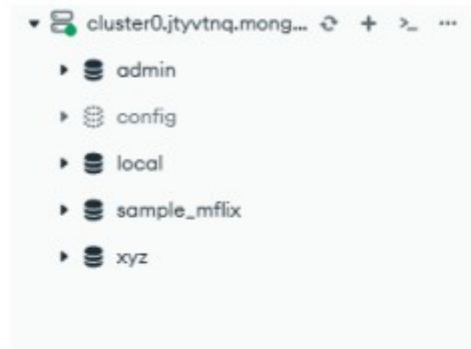
[Node.js Starter Sample App](#)

[Access your Database Users](#)

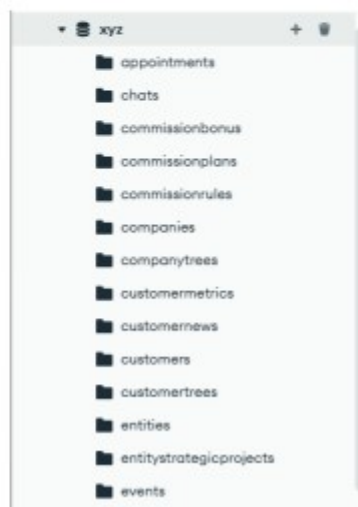
[Troubleshoot Connections](#)

```
.env.local
1 APP_URL=http://localhost:3000
2 MONGODB_URI=mongodb+srv://clarensalipro_db_user:yN2Tc7jW3CMHZYG1@cluster0.jtyvtmq.mongodb.net/
3 MONGODB_DB_NAME=xyz
4 JWT_SECRET=2GQd3VUG5wBIOAlhFUFUh6egzrDxpYqKGv6GgrgOCRbx35M0dZRux0wR1KazExUJ
```

Une fois cela fait et après avoir lancer notre base de donnée et nous être connecté sur le port localhost:3000 pour s'assurer que tout fonctionnait bien, on a créé un user (en appuyant sur ADD DATA et ensuite insert document) dans le fichiers users pour pouvoir nous connecter à l'API et y naviguer.



Puis par la suite nous avons insert deux autres data dans le fichier companies à la demande de notre tuteur







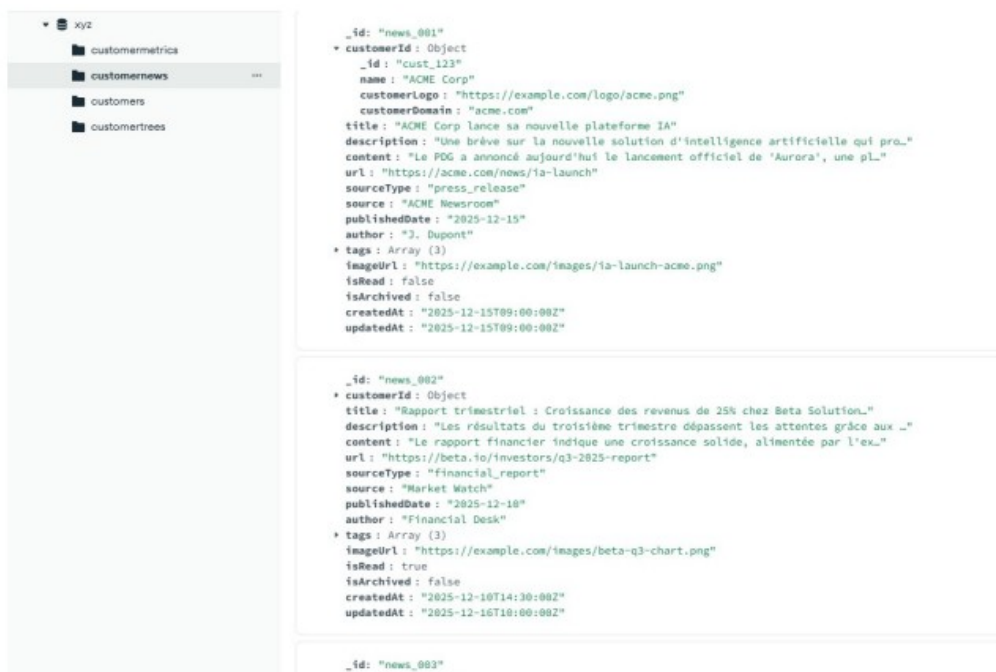
Tout d'abord on lance l'API depuis VisualStudioCode en allant sur notre projet en ouvrant un terminal gitbash et grâce à la commande npm run dev

```
$ npm run dev
> deeeelroom@1.1.0 dev
> tsx server.ts

> Ready on http://localhost:3000
> WebSocket server running on the same port
✔ MongoDB scheduler prêt (utilise la connexion existante)
🕒 Démarrage du scheduler d'agrégation des actualités
📅 Planification : 0 1 * * * (tous les jours à 1h du matin)
✔ Scheduler démarré avec succès
```

Une fois lancé ce message nous est renvoyé et on sait que tout la connexion à la base de donnée à été effectué avec succès et que notre API fonctionnent correctement.

Une fois l'API lancé, on se connecte et on va dans l'onglet « Actualités clients » puis une fois cela fait on va dans MongoDB Compass dans la collection customernews et on y ajoute nos 10 documents :



The screenshot shows the MongoDB Compass interface. On the left, a sidebar displays a folder structure under 'xyz' with subfolders: 'customermetrics', 'customernews' (selected), 'customers', and 'customertrees'. The main area displays two documents from the 'customernews' collection.

**Document 1 (news\_001):**

- `_id`: "news\_001"
- `customerId`: Object
  - `_id`: "cust\_123"
  - `name`: "ACME Corp"
  - `customerLogo`: "https://example.com/logo/acme.png"
  - `customerDomain`: "acme.com"
- `title`: "ACME Corp lance sa nouvelle plateforme IA"
- `description`: "Une brève sur la nouvelle solution d'intelligence artificielle qui pro..."
- `content`: "Le PDG a annoncé aujourd'hui le lancement officiel de 'Aurora', une pl..."
- `url`: "https://acme.com/news/ia-launch"
- `sourceType`: "press\_release"
- `source`: "ACME Newsroom"
- `publishedDate`: "2025-12-15"
- `author`: "J. Dupont"
- `tags`: Array (3)
- `imageUrl`: "https://example.com/images/ia-launch-acme.png"
- `isRead`: false
- `isArchived`: false
- `createdAt`: "2025-12-15T09:00:00Z"
- `updatedAt`: "2025-12-15T09:00:00Z"

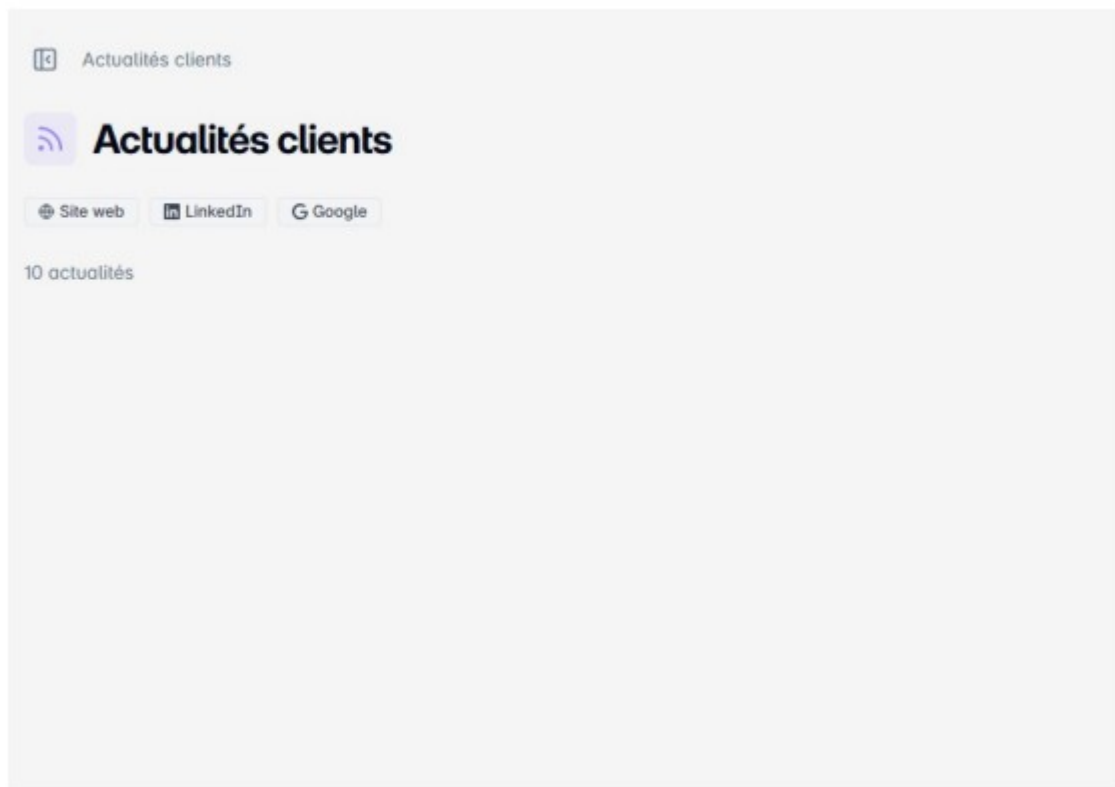
**Document 2 (news\_002):**

- `_id`: "news\_002"
- `customerId`: Object
  - `title`: "Rapport trimestriel : Croissance des revenus de 25% chez Beta Solution..."
  - `description`: "Les résultats du troisième trimestre dépassent les attentes grâce aux ..."
  - `content`: "Le rapport financier indique une croissance solide, alimentée par l'ex..."
  - `url`: "https://beta.io/investors/q3-2025-report"
  - `sourceType`: "financial\_report"
  - `source`: "Market Watch"
  - `publishedDate`: "2025-12-16"
  - `author`: "Financial Desk"
  - `tags`: Array (3)
  - `imageUrl`: "https://example.com/images/beta-q3-chart.png"
  - `isRead`: true
  - `isArchived`: false
  - `createdAt`: "2025-12-16T14:30:00Z"
  - `updatedAt`: "2025-12-16T16:00:00Z"

**Document 3 (news\_003):**

- `_id`: "news\_003"

Une fois cela fait on refresh la page Actualité clients sur l'API et on voit bien qu'il y'a 10 actualités :



Mise en place de prototype en TypeScript permettant la récupération de news grâce au flux RSS de Google News :

Voici ce que j'ai appris et mis en place :

En premier lieu k'ai configuré un projet Node.js et installé les dépendances nécessaires, notamment le package `rss-parser` qui permet de lire et de structurer les données provenant d'un fichier XML (le flux RSS).

```
mkdir bbc-news-proto
cd bbc-news-proto
npm init -y
npm install rss-parser
npm install --save-dev typescript ts-node @types/node
npx tsc --init
```

Pour que le code soit propre et sécurisé, j'ai créé une interface `NewsItem` qui permet de définir exactement quelles données je voulais récupérer (le titre, le lien et la date de publication).

```
5  interface NewsItem {
6      title: string;
7      link: string;
8      pubDate: string;
9  }
```



Par la suite j'ai codé une fonction asynchrone `fetchBBCNews` qui se connecte à l'url de la BBC et j'ai aussi utilisé une méthode `slice(0,5)` pour limiter l'affichage aux 5 articles les plus récents, afin de ne pas surcharger la console et j'ai ajouté des conditions (comme « sans titre ») pour éviter que le programme ne plante si une information est manquante dans le flux.

```
async function fetchBBCNews() {
  const RSS_URL = "https://feeds.bbc1.co.uk/news/rss.xml";

  try {
    console.log("Récupération des actualités BBC News...\n");
    const feed = await parser.parseURL(RSS_URL);
    console.log(`Nombre d'articles trouvés : ${feed.items.length}\n`);

    const newList: NewsItem[] = feed.items.slice(0, 5).map((item: any) => ({
      title: item.title || "Sans titre",
      link: item.link || "",
      pubDate: item.pubDate || "Date inconnue",
    }));

    newList.forEach((news, index) => {
      console.log(`[${index + 1}] ${news.title}`);
      console.log(`    Date: ${news.pubDate}`);
      console.log(`    Lien: ${news.link}`);
      console.log(".".repeat(60));
    });
  } catch (error) {
    console.error("Erreur lors de la récupération des actualités :", error);
  }
}
```

Et enfin j'ai configuré le fichier `package.json` pour que je puisse lancer tout le processus d'un coup simplement avec la commande `npm start`.

```
{
  "name": "prototype",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "tsc index.ts && node index.js",
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "type": "commonjs",
  "dependencies": {
    "@types/node": "^25.0.3",
    "rss-parser": "^3.13.0",
    "ts-node": "^10.9.2",
    "typescript": "^5.9.3"
  }
}
```

# Bilan et compétences acquises

Ce stage chez Syslearn marque une étape majeure dans mon parcours, me permettant de passer du support matériel effectué l'an dernier à une maîtrise concrète du cycle de vie d'un logiciel professionnel. J'ai consolidé un socle technique Fullstack en développant des fonctionnalités complètes, de la création de routes API sécurisées côté Backend à l'intégration dynamique avec Next.js côté Frontend.

Sur le plan de la donnée, j'ai acquis une autonomie dans l'administration de bases NoSQL en configurant des clusters sur MongoDB Atlas. Cette flexibilité a été essentielle pour gérer des informations variées. J'ai ainsi appris à extraire et structurer des flux RSS externes tout en optimisant les performances de stockage en mémoire vive.

Au-delà de la technique, cette immersion m'a formé à la rigueur méthodologique du métier de développeur. J'ai appris l'importance d'analyser les besoins fonctionnels à travers des User Stories avant chaque phase de codage. L'utilisation systématique de GitHub pour le versioning et le respect des standards de nommage m'ont permis de produire un code propre et collaboratif, conforme aux attentes d'une équipe de développement.

En conclusion, cette expérience valide pleinement mon projet professionnel vers une carrière de développeur Fullstack. En comparant ce stage à ma première année chez Powerlab, j'ai pu mesurer ma progression vers l'ingénierie logicielle et confirmer ma capacité à gérer aussi bien les infrastructures de données que l'expérience utilisateur final.

# Conclusion

Ce stage de sept semaines au sein de l'entreprise Syslearn a été une expérience particulièrement enrichissante qui marque un tournant dans mon parcours de formation. Intégré au projet Deeelroom, j'ai eu l'opportunité de passer de la théorie académique à la pratique professionnelle en relevant des défis techniques stimulants, notamment dans le domaine du développement Fullstack et de la gestion de données NoSQL.

Contrairement à ma première immersion chez Powerlab qui était axée sur le support et la relation client, ce second stage m'a permis de me concentrer exclusivement sur l'ingénierie logicielle. J'ai pu approfondir ma maîtrise de technologies modernes comme Next.js, TypeScript et MongoDB, tout en découvrant l'importance de la rigueur méthodologique à travers l'utilisation de GitHub. Ces compétences ne sont pas seulement techniques ; elles m'ont aussi appris à être plus autonome et à proposer des solutions optimisées face à des problématiques réelles.

Travailler sous la supervision de Monsieur Clément Lidar m'a permis de mieux appréhender les attentes du monde professionnel : la qualité du code, le respect des délais et la capacité à collaborer efficacement en équipe. Cette expérience confirme mon attrait pour le développement logiciel et valide mon projet de poursuivre dans cette voie à l'issue de mon BTS SIO.

En conclusion, ce stage constitue une étape charnière de mon parcours. Il m'a permis de consolider mon identité de développeur SLAM et me donne une vision claire et motivée de mes futures aspirations professionnelles. Je me sens aujourd'hui prêt à aborder les épreuves finales de mon examen avec un bagage technique et une expérience de terrain solides.

