**DEPARTEMENT MATHEMATIQUES ET INFORMATIQUE**

**Filière Ingénieur :**

**« Ingénierie Informatique : Big Data et Cloud Computing »**

*II-BDCC 1*

**Rapport du Projet de Fin de Module**



Application de Blogging basée sur Node.js et Express : une plateforme web pour les blogs



**Encadré par :**

Pr Abdelaziz DAAIF

**Réaliser par :**

Nouhayla MOUAKKAL

Sommaire :

[Introduction : 3](#_Toc135586607)

[Présentation du projet : 4](#_Toc135586608)

[Cahier de Charge : 5](#_Toc135586609)

[1-Contexte générale du projet : 5](#_Toc135586610)

[2- Contraint fonctionnelles : 6](#_Toc135586611)

[3-Relations entre les entités : 6](#_Toc135586612)

[Diagramme de Classe : 6](#_Toc135586613)

[Backend : 7](#_Toc135586614)

[Article : 11](#_Toc135586615)

[Catégorie : 12](#_Toc135586620)

[Commentaire 13](#_Toc135586625)

[Utilisateur 14](#_Toc135586630)

[Authentification 15](#_Toc135586635)

[Frontend : 17](#_Toc135586638)

[Conclusion : 22](#_Toc135586639)

# 

# Introduction :

Avec la montée en puissance du développement web et l'expansion continue des besoins, Node.js s'est établi comme un pilier incontestable pour la création d'applications cross-plateformes. Que ce soit pour le développement d'applications mobiles avec des Framework tels que Ionic ou la conception d'applications pour ordinateurs portables grâce à Electron, Node.js offre une polyvalence incomparable. En outre, il est largement utilisé dans le domaine du développement de serveurs web et la mise en place d'API REST. Face à cette demande croissante, les développeurs se tournent vers des Framework tels qu'Express.js, qui leur permettent de construire rapidement des bases solides pour leurs applications tout en favorisant une collaboration harmonieuse et le respect de normes communes. Ainsi, ces Framework offrent une solution efficace pour accélérer le processus de développement, répondre aux exigences croissantes et offrir des résultats de qualité supérieure.

# 

# Présentation du projet :

* Objectif :

L'objectif de ce projet est de créer une plateforme de blogs permettant aux utilisateurs d'afficher et de gérer des articles liés à différentes catégories. Les blogs seront accessibles au public, offrant ainsi une plateforme de partage d'informations et d'interactions.

* Description :

Ce projet vise à fournir une solution complète pour les blogs, offrant des fonctionnalités spécifiques en fonction du rôle de chaque utilisateur. Les utilisateurs peuvent être des administrateurs, des utilisateurs normaux ou des blogueurs.

En tant qu'administrateur, l'utilisateur aura le pouvoir de gérer les autres utilisateurs, de gérer les articles existants ainsi que les catégories. Cette fonctionnalité permet de maintenir l'ordre et de garantir que le contenu du blog reste pertinent et de haute qualité.

Les utilisateurs normaux auront la possibilité de consulter les blogs, de lire les articles et d'ajouter des commentaires. Cela favorise l'engagement de la communauté en permettant aux utilisateurs de donner leur avis, de poser des questions ou de partager des idées.

Les blogueurs, quant à eux, auront des privilèges supplémentaires. Ils pourront créer leurs propres articles et les gérer, mais uniquement pour les articles qu'ils ont créés. Cela permet aux blogueurs de partager leurs connaissances et leurs expériences spécifiques dans leur domaine d'expertise.

En résumé, ce projet vise à offrir une plateforme de blogs complète et interactive, permettant aux utilisateurs d'accéder à une multitude de sujets et de catégories, de partager leurs propres articles et de participer à des discussions enrichissantes au sein de la communauté des blogs.

# Cahier de Charge :

## 1-Contexte générale du projet :

Nous avons l'intention de créer un blog dont les parties backend et frontend seront développées et testées de manière indépendante. Le backend sera conçu sous la forme d'une API JSON. Voici les technologies que nous prévoyons d'utiliser pour ce projet :

Du côté **backend** :

Le Framework Node.js Express pour le serveur web.

* L'ORM Prisma et une base de données MySQL/MariaDB pour la persistance des données.

Du côté **frontend** :

* HTML/CSS/JavaScript.

Vous pouvez utiliser la bibliothèque JavaScript jQuery et le Framework CSS Bootstrap.

## 2- Contraint fonctionnelles :

L'application Blog doit respecter les contraintes fonctionnelles suivantes, en gérant quatre entités :

**1. Utilisateur :**

**3. Catégorie :**

- Nom

**4. Commentaire :**

- Email

- Contenu

- Nom

- Email

- Mot de passe

- Rôle (ADMIN || AUTHOR)

**2. Article :**

- Titre

- Contenu

- Image

- Date de création

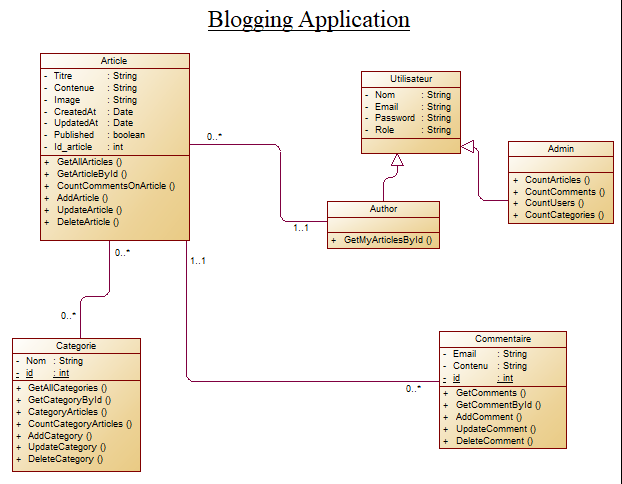
- Date de mise à jour

- Statut de publication sous forme booléenne

## 3-Relations entre les entités :

* Un **article** est associé **à un et un seul** **utilisateur** (Cet utilisateur devrait avoir le role AUTHOR)
* Un **utilisateur** (ayant le rôle AUTHOR) peut écrire **zéro ou plusieurs** **articles**
* Un **article** est associé à **zéro ou plusieurs catégories**
* Une **catégorie** est associée à **zéro ou plusieurs** **articles**
* Un **commentaire** est associé à **exactement un** **article**
* Un **article** est associé à **zéro ou plusieurs** **commentaires**.

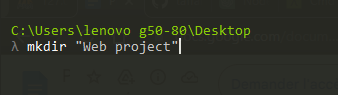
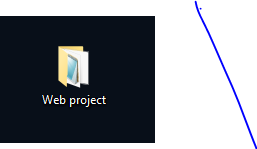
# Diagramme de Classe :



Réalisation du projet :

# Backend :

1. J’ai créé le répertoire de notre projet :



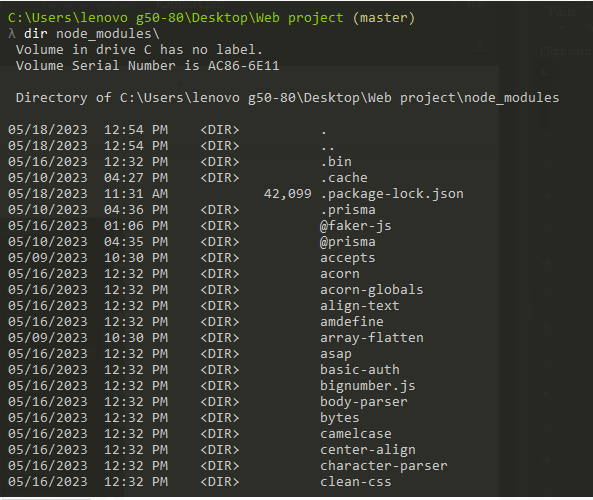
1. J’ai saisi les commandes nécessaires pour commencer :



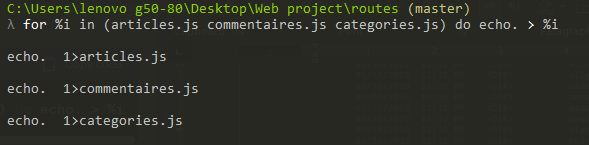


* Pour installer le répertoire node modules :





🡺Création des trois fichiers



1. Prisma ORM :



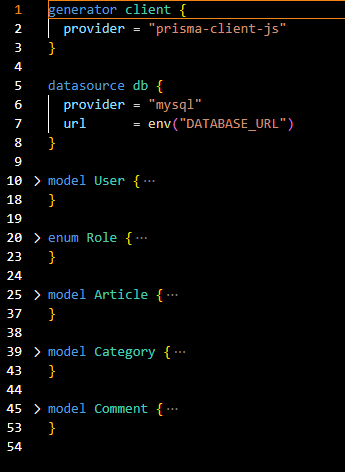




1. **Env**

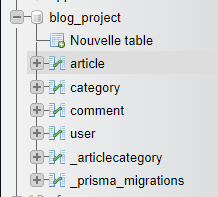


1. L’ajout des modèles reflétant les entités et les relations entre elles et la migration :



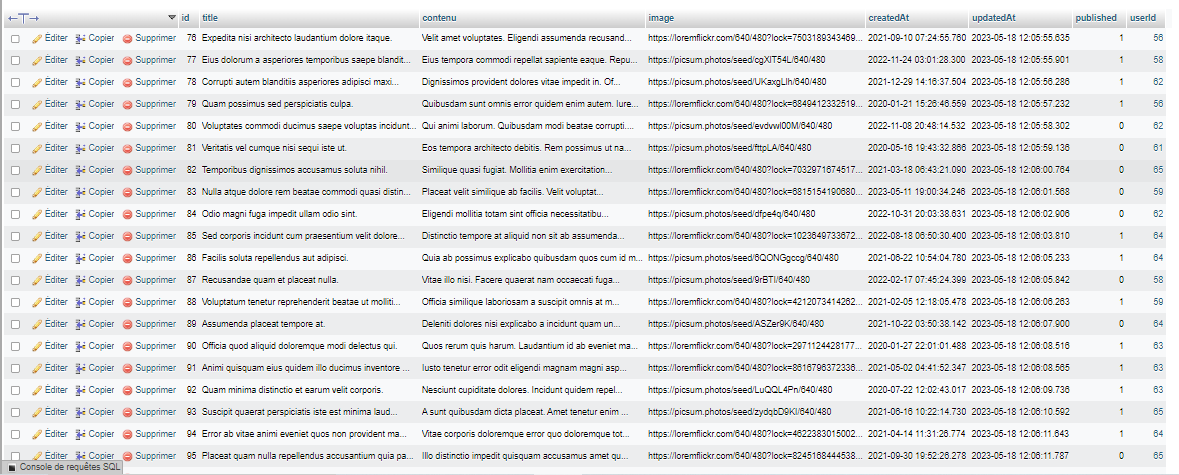
1. On exécute la commande : **npx prisma migrate dev --name init**

* **Donc notre base de données a été créer :**



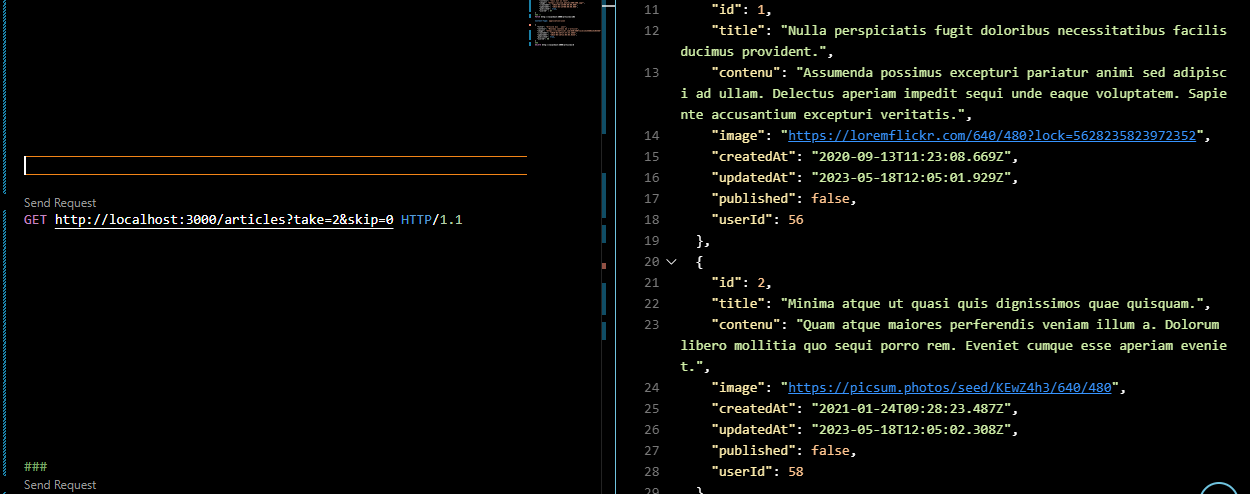
1. Dans le fichier **‘’web project\routes\seeds\seed.js’’** on exécute un code utilisant la bibliothèque Faker qui va nous permettre de créer :
   * 10 utilisateurs ayant le rôle “AUTHOR”
   * 1 utilisateur ayant le rôle “ADMIN”
   * 10 catégories
   * 100 articles appartenant à (de 1 à 4 catégories aléatoires) et écrit par l’un des 10 utilisateurs (AUTHOR)
   * Pour chaque article, créer de 0 à 20 commentaires)

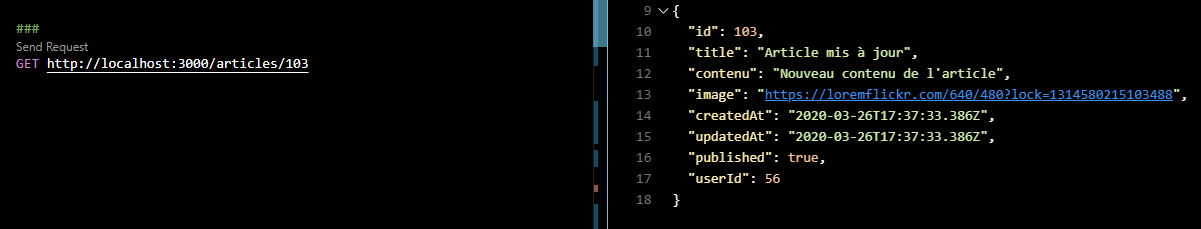
Voilà un exemple de l’exécution :



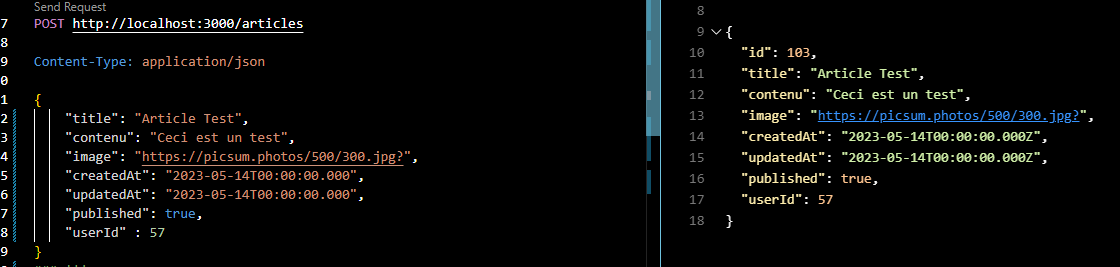
## Article :

## GET

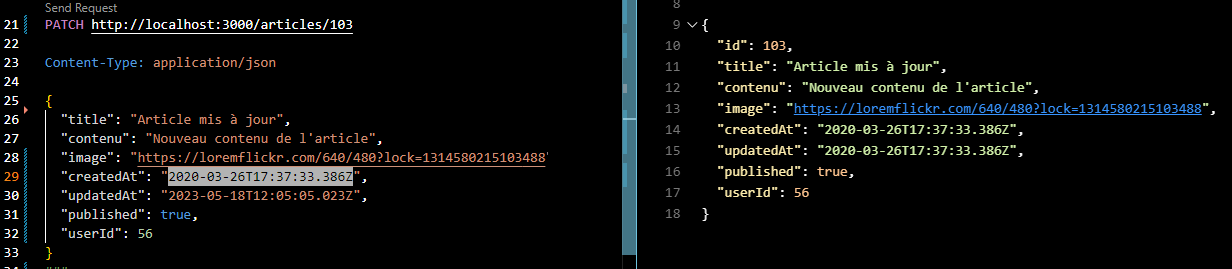
:



## POST



## PATCH

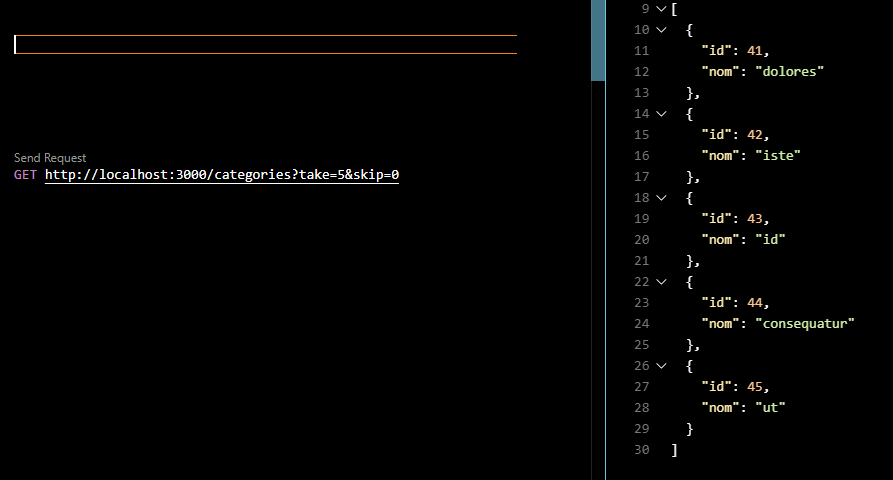


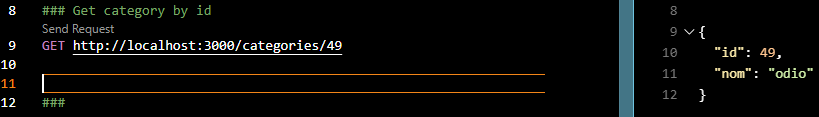
## DELETE



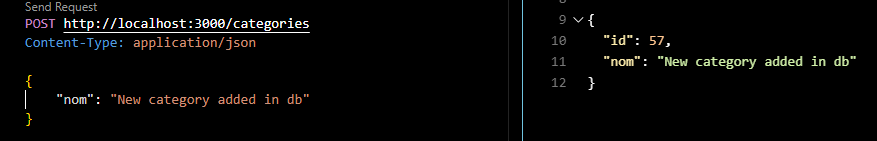
## Catégorie :

## GET

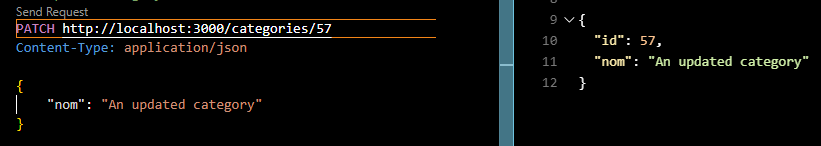




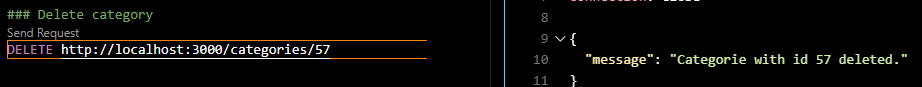
## POST



## PATCH

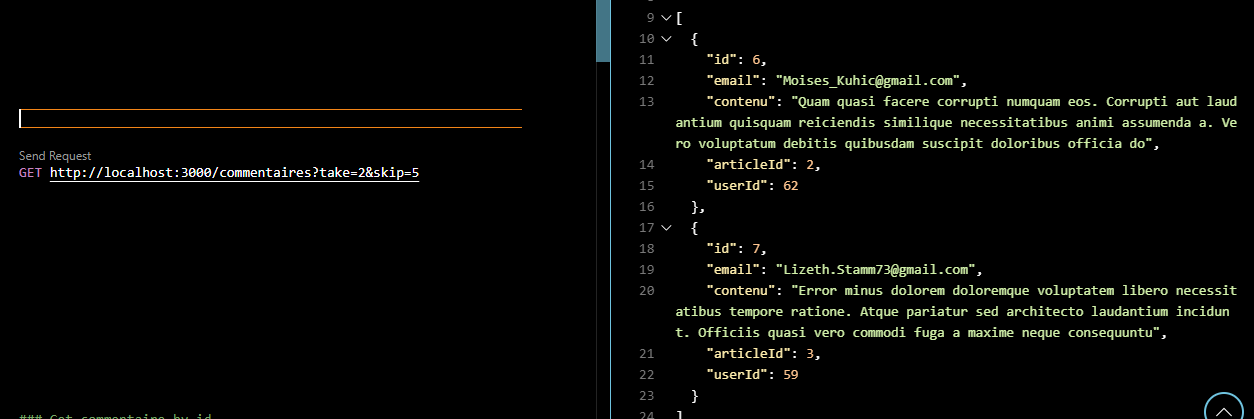


## DELETE



## Commentaire

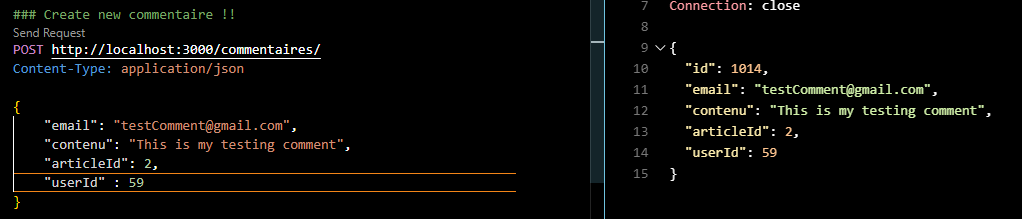
## GET



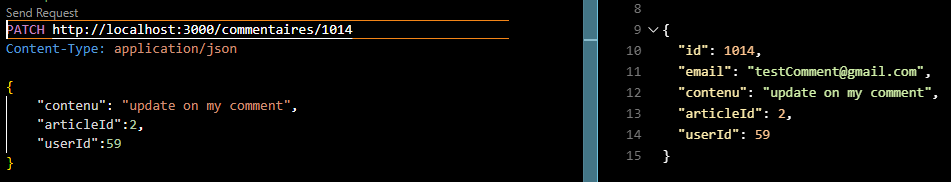


## 

## POST



## PATCH

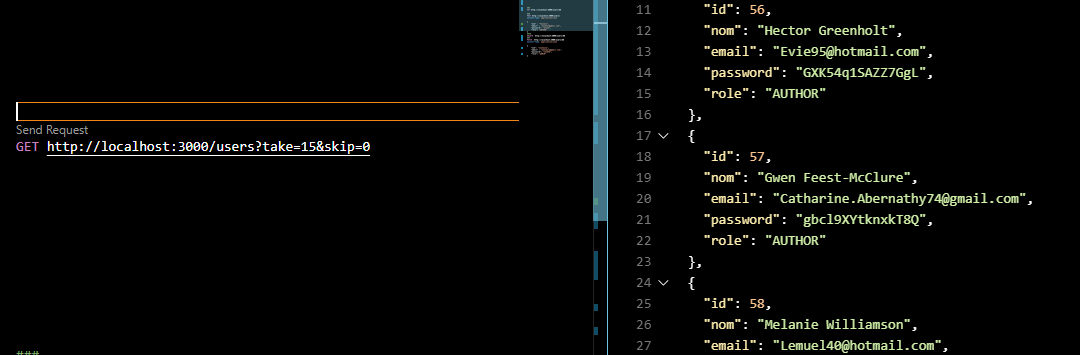


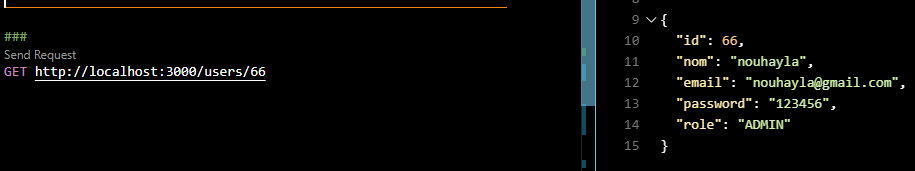
## DELETE



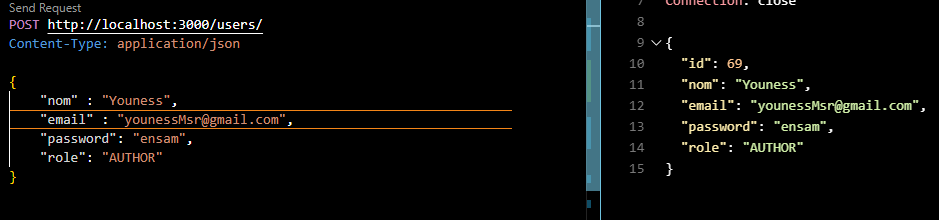
## Utilisateur

## GET

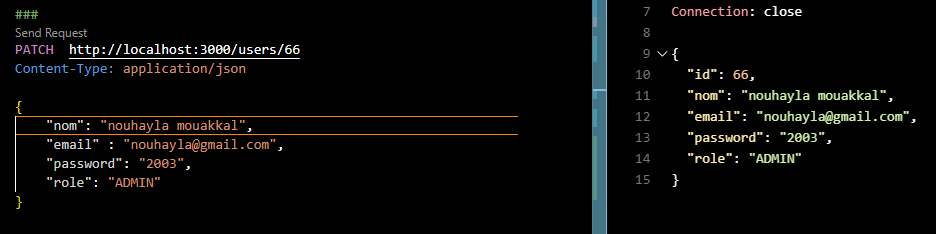




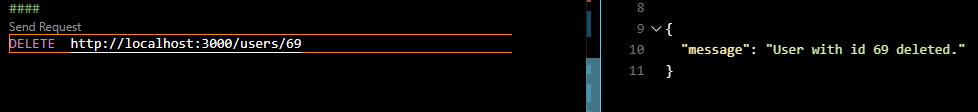
## POST



## PATCH



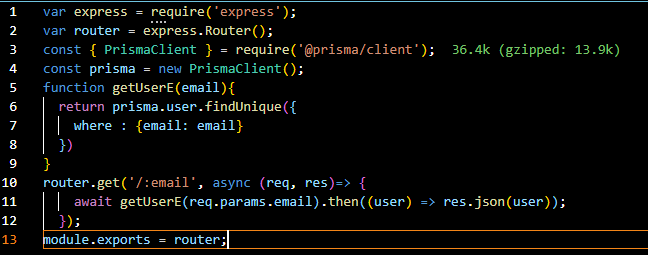
## DELETE



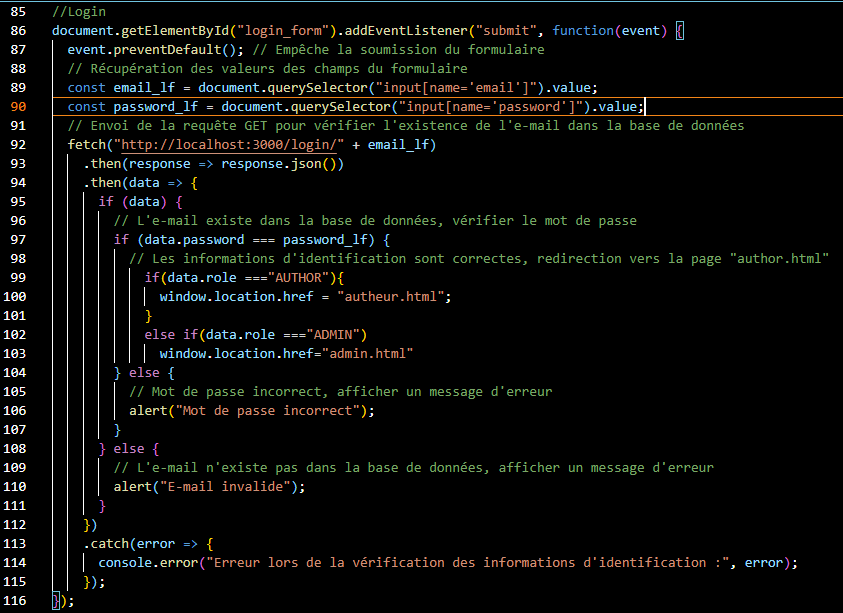
## Authentification

## Login :

Dans mon fichier que j’ai l’appellé Checkin.js :



On va rechercher l'utilisateur ayant l'adresse e-mail saisie, car elle est unique. Ensuite, On va vérifier le mot de passe et le rôle de l'utilisateur en utilisant le code JavaScript suivant :

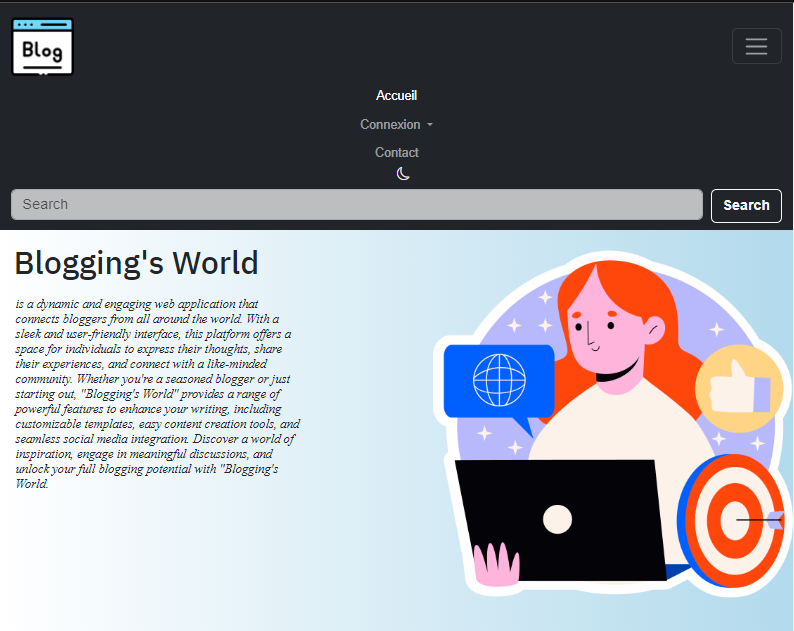


## Sign up :

Voici comment un nouvel utilisateur peut s'inscrire :

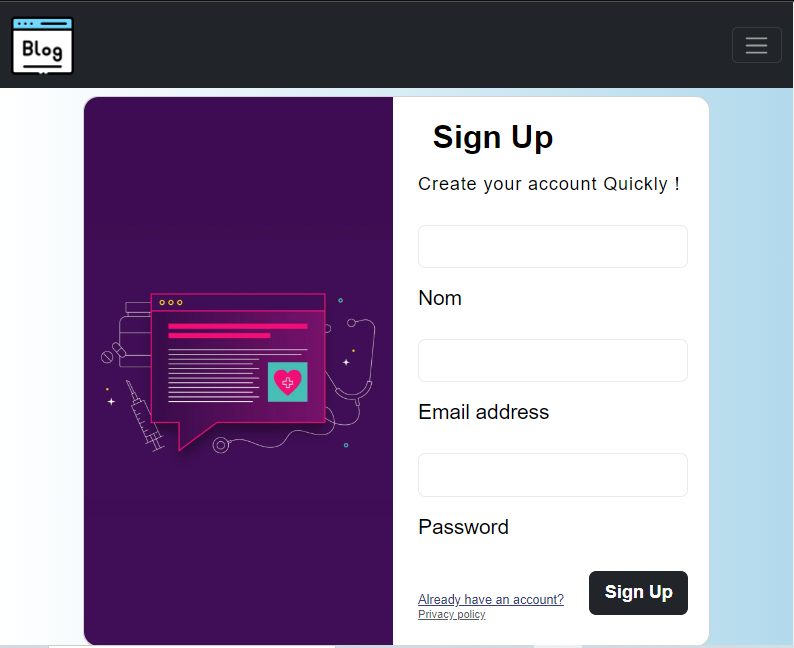
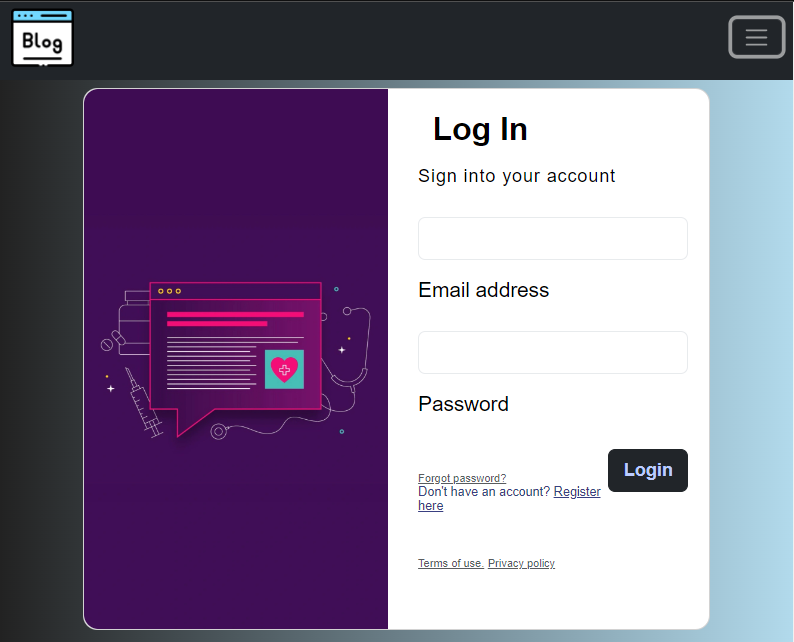


# Frontend :

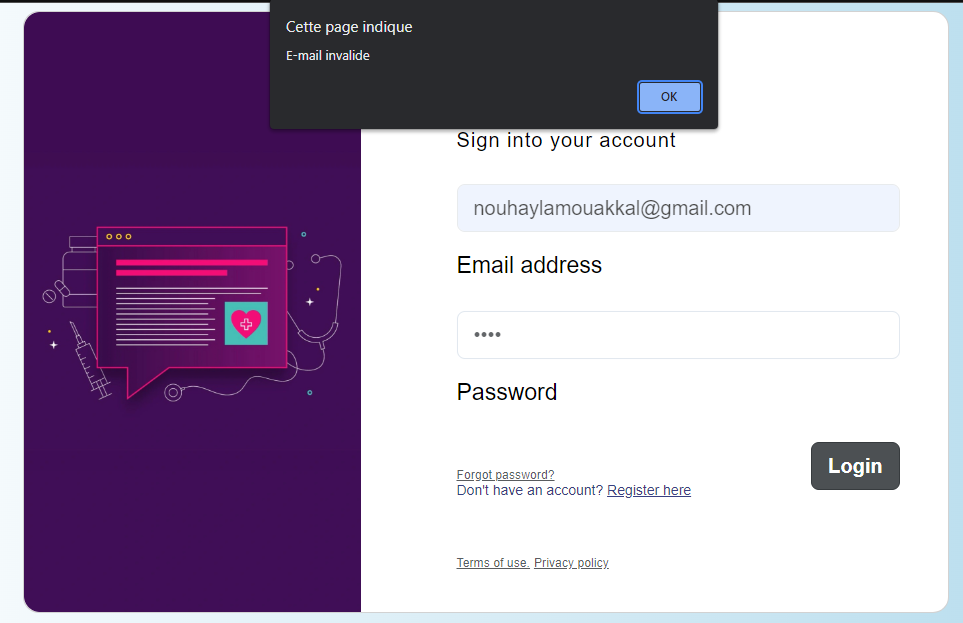
🡺La première interface

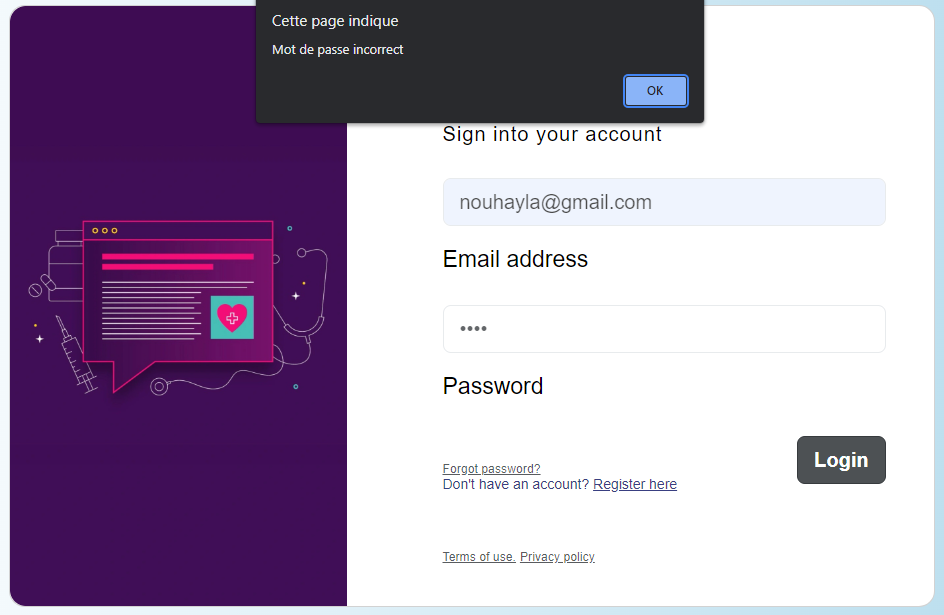


🡺Les deux interfaces d’authentification :

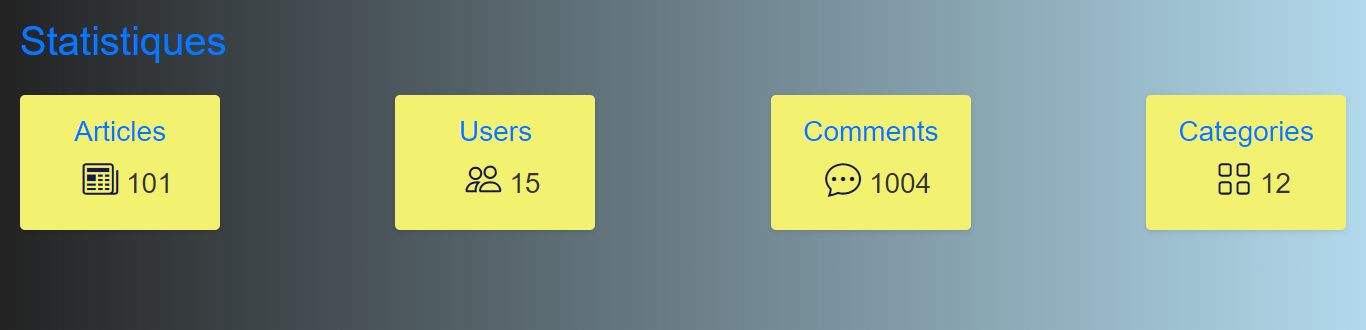


🡺Validation des données

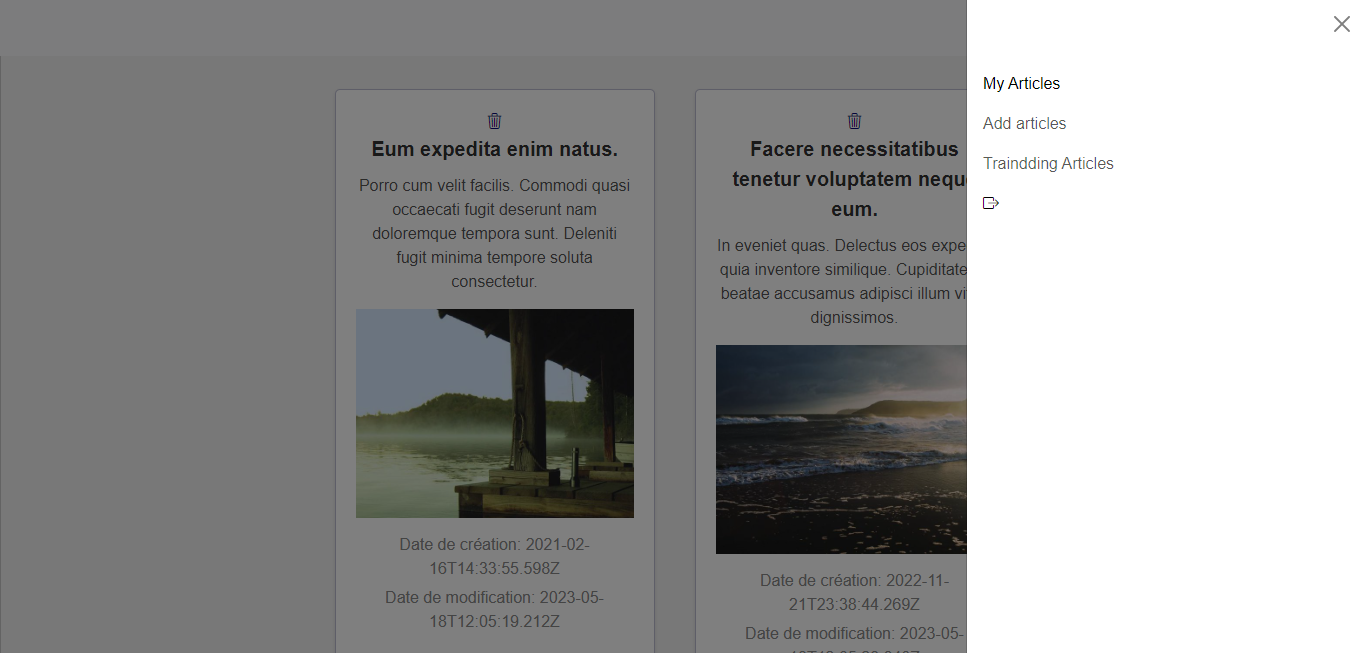


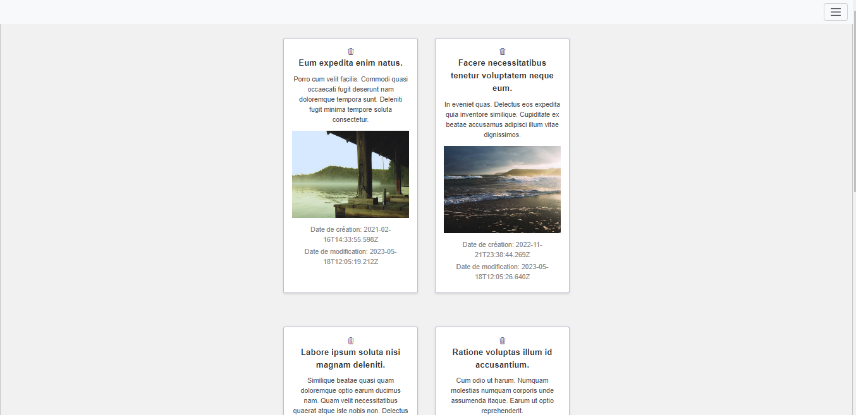


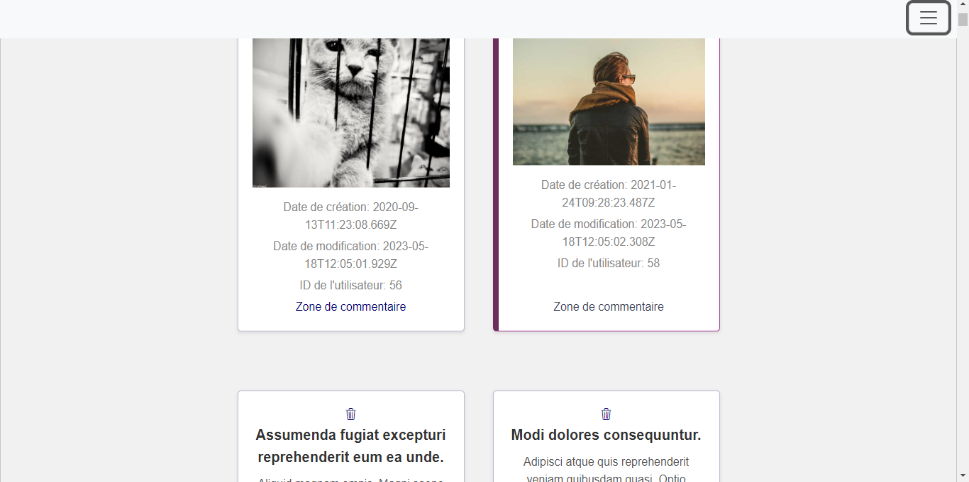
🡺 Interface de l’admin



**Les interfaces de l’auteur**

🡺Première on a cette navbar qui contient les articles de cet auteur + zone ou il peut ajouter un nouvel article

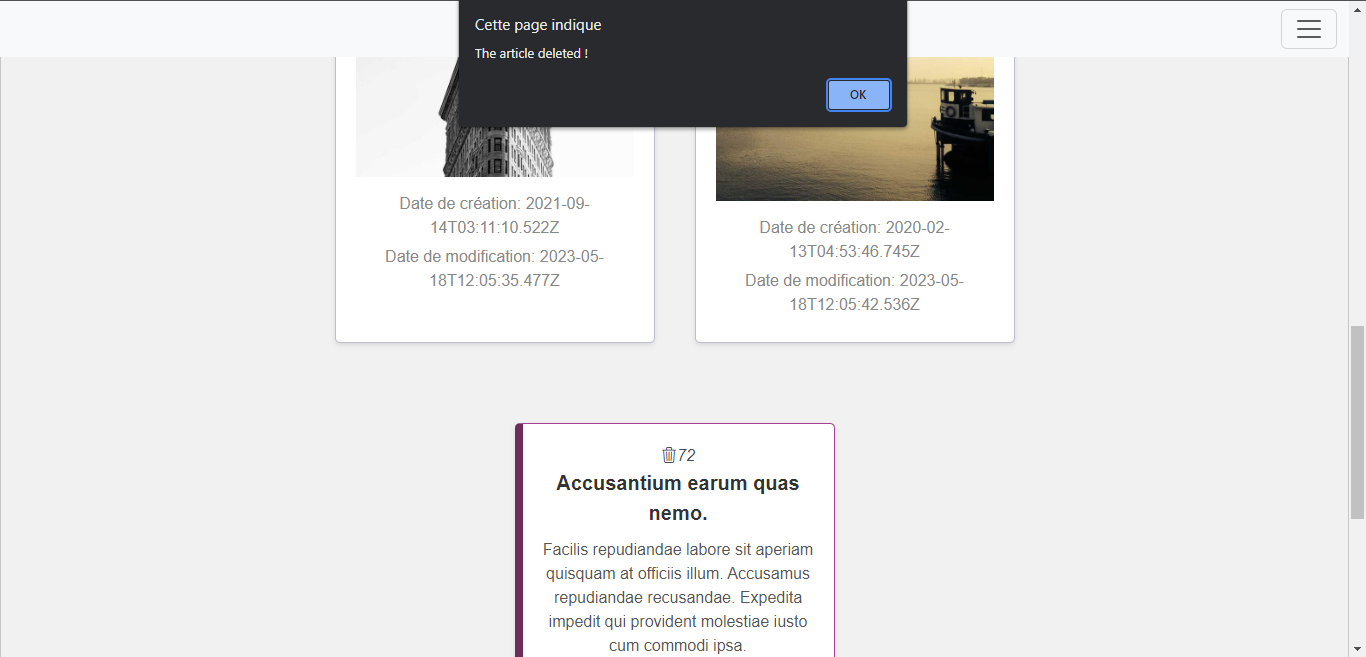
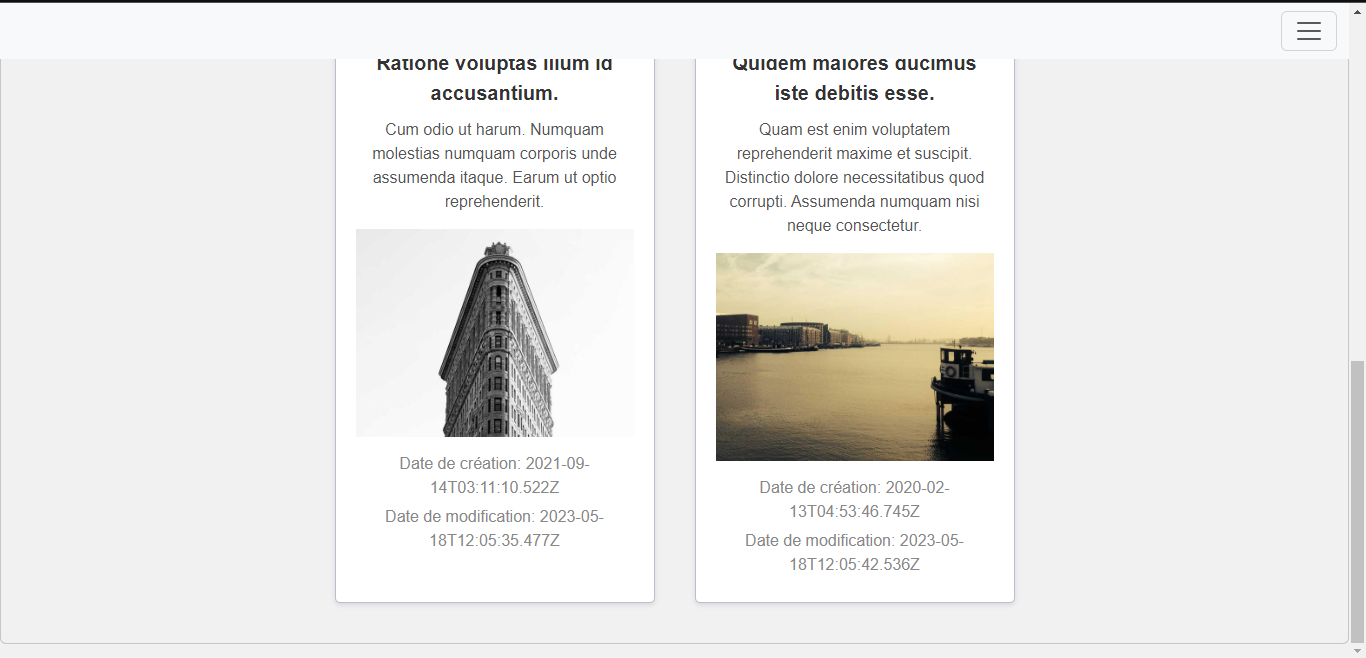
🡺Mes Articles 🡺Ajouter un nouvel article



🡺Différents Article à partir

ma base de données

🡺Suppression d’un article, il suffit juste de de cliquer sur l’Icone



# Conclusion :

Ce projet m'a permis de mettre en valeur mes atouts et connaissances tout en découvrant une approche de conception et d'analyse. J'ai respecté les délais pour les documents à rendre et j'ai acquis une nouvelle expérience dans le monde de l'informatique, approfondissant mes connaissances en développement web et découvrant de nouvelles technologies pour résoudre certains problèmes. Mon projet a pleinement répondu aux exigences du cahier des charges. En outre, il s'agit d'une première étape pour de futurs développements. À l'avenir, cette application pourrait être améliorée et enrichie en ajoutant des fonctionnalités avancées. Bien que le côté backend ait été respecté, il existe des interfaces côté frontend qui pourraient être ajoutées et améliorées, mais en raison de contraintes de temps, je n'ai pas pu tester tous les scénarios sur différentes interfaces.