

Département Génie des Communications et Réseaux

Rapport de Mini-Projet

Blogify : Plateforme de blogging Collaborative

ENIG

Elaboré par :

[Redacted Name]

Génie des communications et Réseaux

[Redacted Name]

[Redacted Name]

Encadré par :

[Redacted Name]

Maitre Assistance

Période: du 13/11/2024 au 15/12/2024

Année Universitaire: 2023/2024

Remerciements

Remerciements

On souhaite exprimer nos sincère reconnaissance à toutes les personnes qui ont facilité et enrichi notre projet. En premier lieu, nous remercions sincèrement **Mr.** [REDACTED] notre encadreur de ce projet, pour son accueil chaleureux, ses conseils précieux et son soutien tout au long de cette expérience. Son expertise et sa disponibilité ont été essentiels à l'aboutissement. Nous souhaitons aussi remercier mes enseignants de l'École Nationale d'Ingénieurs de Gabès pour m'avoir donné les bases théoriques et pratiques nécessaires pour aborder ce projet avec confiance. Enfin, nous tenons à exprimer nos gratitude à mes famille et à mes amis pour leur soutien constant tout au long de cette période d'apprentissage.

Sommaire

Introduction Générale	1
Contexte général.....	2
1.1 Introduction.....	2
1.2 Cadre du projet	2
1.3 Problématique	2
1.4 Analyse et critique.....	3
1.4.1 Analyse des besoins	3
1.4.2 Critique de l'existant.....	3
1.5 Solution Proposée.....	4
1.6 Conclusion	4
Analyse et Conception	5
2.1 Introduction.....	5
2.2 Analyse du système	5
2.2.1 Identification des acteurs	5
2.2.2 Spécification des besoins fonctionnels	5
2.3 Conception du système	6
2.3.1 Diagramme de cas d'utilisation	6
2.3.2 Diagramme de classe.....	6
2.4 Conclusion	8
Réalisation	9
3.1 Introduction.....	9
3.2 Architecture technique.....	9
3.2.1 Interaction entre les couches	9
3.2.2 Couche Backend.....	10
3.2.3 Couche Frontend.....	10
3.2.4 Couche Base de Données	10
3.3 Configuration du système.....	10
3.3.1 Configuration de Node.js et des dépendances	10
3.3.2 Sécurité	11
3.3.3 Gestion des environnements.....	11
3.4 Environnement et outils de développement.....	12

3.4.1 Environnement matériel.....	12
3.4.2 Outils de développement	12
3.5 Mise en Place des fonctionnalités.....	15
3.5.1 Développement des Modules	15
3.5.2 Implémentation des fonctionnalités clés	16
3.5.3 Tests et Validation	17
Resultat : Commentaire ajouté à la base de données.....	20
3.6 Conclusion	20
Conclusion générale et perspectives	21
Netographie	22

Table des figures

Figure 1.1 : Medium et Stack OverFlow.....	3
Figure 2.1 : Diagramme de cas d'utilisation.....	6
Figure 2.2 : Diagramme de classe.....	8
Figure 2.3 : .env.....	9
Figure 3.1 : Architecture 3-Tiers.....	11
Figure 3.1 : VSCode.....	12
Figure 3.2 : Logo Git.....	13
Figure 3.3 : Logo MongoDB.....	13
Figure 3.4 : Logo Next.js.....	14
Figure 3.5 : Logo ArcJet.....	14
Figure 3.6 : Logo Tailwind CSS.....	15
Figure 3.7 : Logo UploadThing.....	15
Figure 3.8 : Test d'enregistrement utilisateur.....	17
Figure 3.9 : Test de Connexion (Login)	18
Figure 3.10 : Test de Création de Blog.....	19
Figure 3.11 : Test de Recherche de Blog.....	19
Figure 3.12 : Test d'ajout de Commentaire.....	20

Introduction Générale

Les blogs sont devenus des outils essentiels pour partager des idées, des connaissances et des expériences à travers le web. Ils permettent aux utilisateurs de publier du contenu structuré, d'interagir avec une communauté et de créer des discussions autour de divers sujets. Dans ce contexte, notre projet s'inscrit dans une démarche d'apprentissage et de mise en pratique des concepts du développement logiciel. Le projet **Blogify** s'inscrit dans le cadre de notre formation à l'École Nationale d'Ingénieurs de Gabès (ENIG). L'objectif est de concevoir une plateforme fonctionnelle et intuitive répondant à un besoin spécifique en matière de partage et de collaboration en ligne.

Blogify est une plateforme de blogging collaborative, inspirée des fonctionnalités de sites populaires tels que **Medium** et **Stack Overflow**, mais qui propose une fusion unique entre le blogging classique et la collaboration communautaire. Cette plateforme permet aux utilisateurs de :

- Publier des articles enrichis.
- Partager des fichiers tels que des documents, des images ou des ressources.
- Intégrer des fonctionnalités spécifiques comme la recherche avancée, la catégorisation des blogs.
- Poser des questions et recevoir des commentaires, suggestions, ou solutions de la communauté.

En combinant ces fonctionnalités, Blogify vise à créer un espace interactif pour le **partage de connaissances et d'expériences**, tout en étant particulièrement adapté aux étudiants et professionnels cherchant un outil flexible et collaboratif.

Contexte général

1.1 Introduction

Ce chapitre présente le contexte dans lequel le projet a été réalisé, ainsi que les objectifs du mini-projet et les problématiques rencontrées. Il introduit également l'intérêt de ce projet, et décrit ses activités principales, permettant ainsi de mieux comprendre l'environnement dans lequel la plateforme a été développée.

1.2 Cadre du projet

Ce projet a été réalisé dans le cadre académique de l'ENIG et se repose sur plusieurs objectifs :

- **Approche pédagogique** : Permettre aux étudiants de développer des compétences en gestion de projet, en programmation avancée, et en collaboration d'équipe.
- **Objectifs techniques** : Développer une application web intégrant les dernières technologies en matière de développement front-end et back-end, de sécurité, et de gestion des données.
- **Finalité pratique** : Proposer une solution innovante aux professionnels pour partager leurs idées et interagir de manière efficace.

Le projet **Blogify** représente une opportunité unique d'appliquer les méthodologies modernes comme Scrum, en favorisant une approche agile et collaborative pour le développement.

1.3 Problématique

Les plateformes existantes telles que **Medium** ou **Stack Overflow** présentent des limitations :

- **Fragmentation des fonctionnalités** : Les plateformes actuelles manquent d'une intégration fluide entre le partage de contenu et la collaboration communautaire.
- **Absence de personnalisation** : Les utilisateurs, notamment les étudiants, ne trouvent pas des environnements adaptés à leurs besoins académiques et professionnels.
- **Accessibilité restreinte** : Les options de recherche et de catégorisation sont souvent limitées, rendant difficile la découverte rapide d'informations pertinentes.

Ces lacunes se traduisent par un manque d'outils adaptés pour répondre à des besoins spécifiques, notamment :

- Une plateforme où les utilisateurs peuvent combiner la **création de contenu riche** avec une **interaction communautaire active**.

Contexte Général

- Un espace qui favorise le **partage de ressources** dans un cadre académique ou professionnel.

La problématique centrale est donc :

Comment concevoir une plateforme qui fusionne blogging et collaboration communautaire, tout en répondant aux besoins d'une audience variée ?

1.4 Analyse et critique

1.4.1 Analyse des besoins

L'analyse des besoins utilisateurs pour Blogify a permis d'identifier plusieurs fonctionnalités essentielles :

- **Besoins fonctionnels :**
 - Permettre aux utilisateurs de publier des articles, partager des fichiers, et poser des questions.
 - Offrir des outils simples pour interagir avec les autres membres via des commentaires, des suggestions, et des solutions.
 - Mettre en place un système de recherche avancée et de filtrage par catégorie.
- **Besoins non fonctionnels :**
 - Garantir la sécurité des données des utilisateurs grâce à des technologies modernes comme ArcJet.
 - Proposer une interface intuitive et esthétiquement attrayante grâce à Tailwind CSS et shadcn/ui.
 - Assurer une validation fiable des formulaires via Zod Conform.

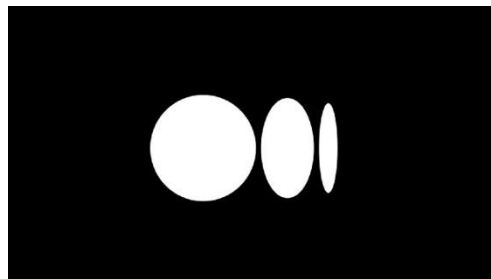


Figure 1.1 : Medium et Stack OverFlow

1.4.2 Critique de l'existant

Les solutions actuelles comme **Medium** et **Stack Overflow** offrent des fonctionnalités spécifiques mais :

Contexte Général

Les plateformes existantes offrent des fonctionnalités limitées par leur spécialisation :

- Elles ne permettent pas une intégration fluide entre le blogging et la collaboration communautaire.
- Elles ne permettent pas une intégration fluide entre le blogging et la collaboration communautaire.
- Elles manquent de personnalisation pour les étudiants et les professionnels ayant des besoins spécifiques.
- Les outils de recherche et de filtrage sont parfois limités ou peu intuitifs.

Aucune des deux plateformes n'intègre une **solution hybride** qui combine blogging riche et interaction communautaire, ce qui crée un manque que Blogify cherche à combler.

1.5 Solution Proposée

Blogify se distingue par sa structure et ses fonctionnalités innovantes :

- Une interface moderne développée avec **Next.js** pour la fluidité et la performance.
- Une base de données **MongoDB** pour une gestion flexible et efficace des données.
- Un système de sécurité renforcé grâce à **ArcJet**.
- Des outils avancés pour créer, rechercher, et interagir avec le contenu publié.
- Une approche modulaire permettant une évolutivité future.

L'approche méthodologique utilisée (**Scrum**) garantit une gestion agile du projet, assurant des livraisons fréquentes et une adaptation continue aux besoins.

1.6 Conclusion

Ce chapitre a présenté le contexte global de notre mini-projet. À travers une analyse des besoins et une critique des solutions existantes, nous avons justifié la pertinence de Blogify comme réponse innovante à la problématique identifiée. Cette plateforme se distingue par son approche hybride et son orientation vers l'apprentissage collaboratif, en proposant des outils adaptés aux étudiants et aux professionnels.

Analyse et Conception

2.1 Introduction

Cette partie présente les étapes de la conception et de la configuration du projet **Blogify**. Elle détaille l'analyse du système, les modèles conceptuels, ainsi que les outils et technologies utilisés pour sa configuration. Ces étapes préliminaires ont permis de poser les bases solides nécessaires au développement des fonctionnalités et à la mise en œuvre du projet.

2.2 Analyse du système

L'analyse du système Blogify vise à identifier les utilisateurs, leurs interactions avec la plateforme et les fonctionnalités essentielles du projet.

2.2.1 Identification des acteurs

Dans cette section, nous définissons les différents utilisateurs du système ainsi que leurs rôles et interactions avec la plateforme :

A. L'acteur utilisateur

L'utilisateur est l'acteur principal de Blogify. Il peut :

- **Créer un compte** : S'inscrire avec un nom, un e-mail et un mot de passe.
- **Se connecter** : Accéder à son compte pour interagir sur la plateforme.
- **Gérer son profil** : Modifier ses informations personnelles.
- **Créer et gérer des articles** : Rédiger, modifier ou supprimer ses propres articles.
- **Interagir avec les articles** : Lire, commenter et voter pour les articles d'autres utilisateurs.

B. L'acteur visiteur

Le visiteur est un utilisateur non inscrit. Il peut :

- **Consulter les articles** : Lire les articles publiés.

2.2.2 Spécification des besoins fonctionnels

A. Gestion des utilisateurs

- **Inscription** : Permettre aux nouveaux utilisateurs de créer un compte avec un email et un mot de passe.
- **Connexion** : Authentifier un utilisateur pour accéder à son espace personnel.

B. Gestion des articles

Analyse et Conception

- **Création d'un article** : Ajouter un nouvel article avec un titre, un contenu, une catégorie et une image.
- **Affichage** : Lire les articles publiés par d'autres utilisateurs.

C. Interaction avec les articles

- **Ajout de commentaires** : Permettre aux utilisateurs d'exprimer leurs avis sous un article.

2.3 Conception du système

La conception du système repose sur une architecture modulaire garantissant l'évolutivité et la sécurité de la plateforme. Elle comprend la modélisation des interactions entre les utilisateurs et la plateforme à travers des diagrammes.

2.3.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation permet de représenter graphiquement les interactions entre les acteurs et le système.

Voici le **diagramme de cas d'utilisation** représentant les interactions entre les utilisateurs et Blogify :

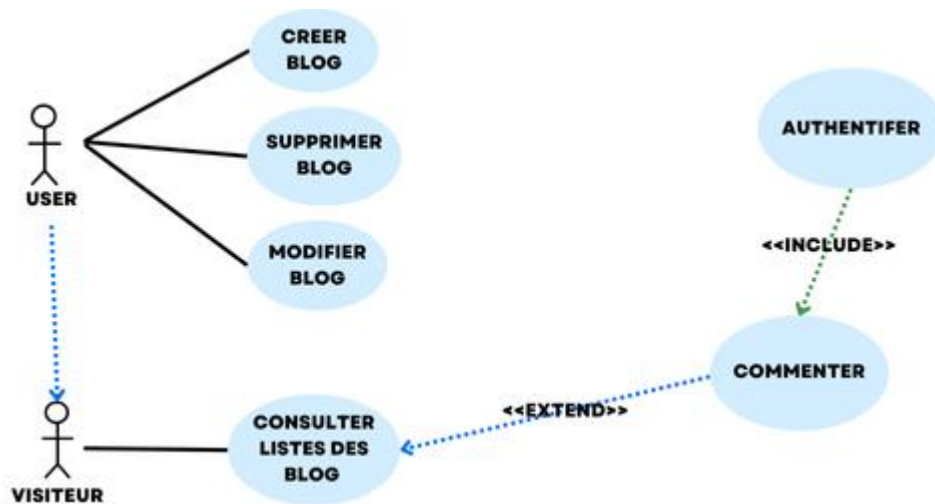


Figure 2.1 : Diagramme de cas d'utilisation

2.3.2 Diagramme de classe

Le diagramme de classes modélise les relations entre les entités principales de Blogify :

- **Utilisateur:**

Représente les données des utilisateurs. Chaque utilisateur peut créer plusieurs articles.

Analyse et Conception

- Attributs :
 - name : Nom de l'utilisateur (string).
 - email : Adresse e-mail (string).
 - password : Mot de passe haché (string).
 - createdAt : Date d'inscription (date).
- Méthodes : authentifier, mettre à jour le profil.

- **Article**

Représente le contenu principal publié par les utilisateurs.

- Attributs : titre, contenu, catégorie, date de création, auteur.
 - title : Titre du blog (string).
 - content : Contenu du blog (string).
 - coverImage : Image de couverture (string).
 - author : Référence à l'utilisateur qui a créé le blog (User).
 - createdAt : Date de création (date).
 - category : Catégorie du blog (string).
 - comments : Liste des commentaires associés (List<Comment>).
 - upvotes : Nombre de votes positifs (integer).
- Méthodes : publier, modifier, supprimer.

- **Comment**

- Attributs :
 - content : Contenu du commentaire (string).
 - author : Référence à l'utilisateur qui a écrit le commentaire (User).
 - authorName : Nom de l'auteur (string, dérivé de User).
 - createdAt : Date de création (date).
- Méthodes : ajouter, supprimer.

Relations :

- **Article** → **Comment** : Un blog peut avoir plusieurs commentaires (1..*).
- **Comment** → **Utilisateur** : Chaque commentaire est écrit par un utilisateur(1..1).
- **Article** → **Utilisateur** : Chaque blog est écrit par un utilisateur (1..1).

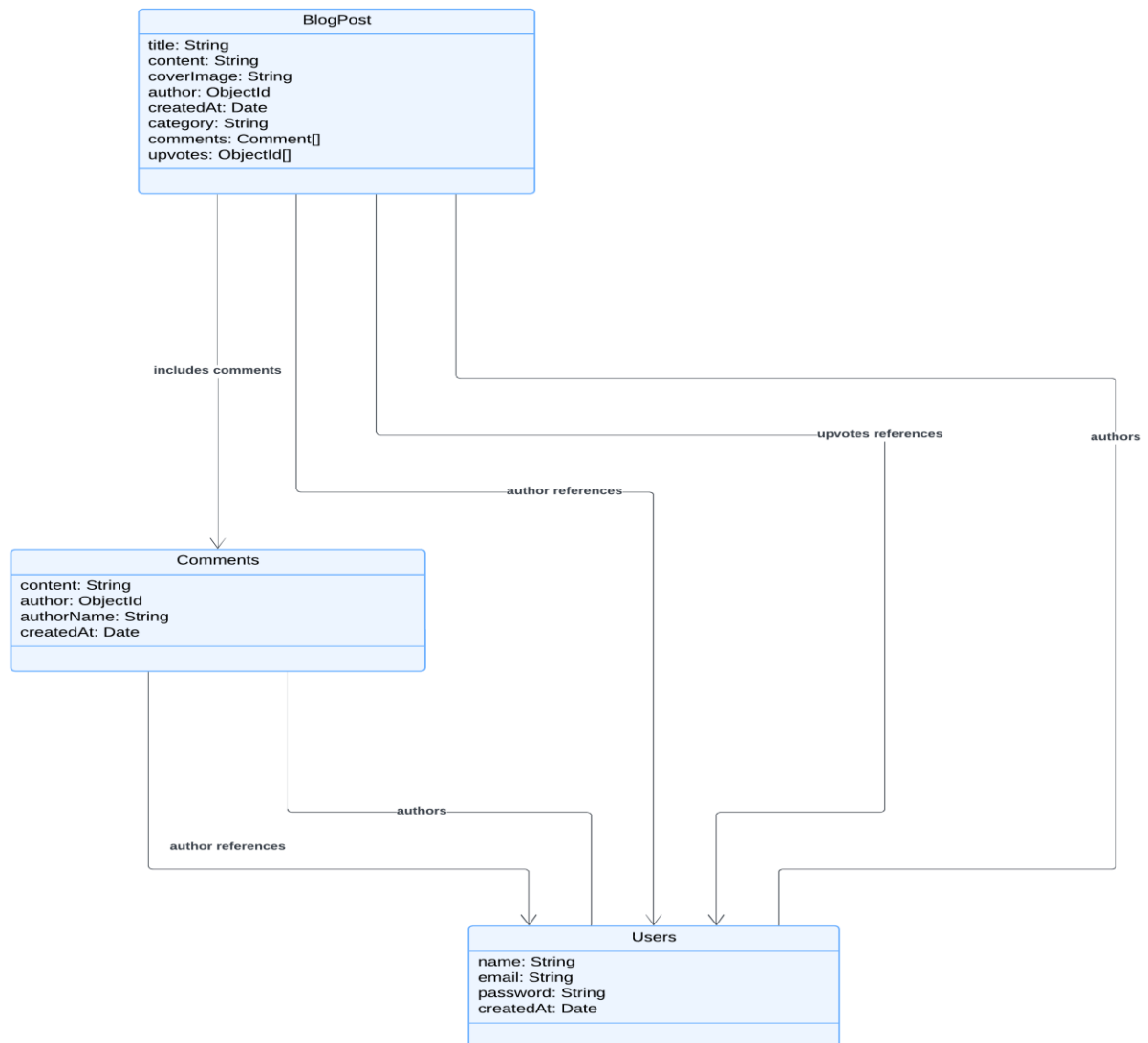


Figure 2.2 : Diagramme de classe

2.4 Conclusion

Cette phase de conception et de configuration a permis de définir une architecture solide pour **Blogify**, adaptée aux besoins du projet. La mise en place de mécanismes de sécurité avancés et d'une gestion efficace des environnements garantit la fiabilité et la scalabilité du système. Cette base technique constitue une étape clé pour la réalisation des fonctionnalités de l'application.

Réalisation

3.1 Introduction

Ce chapitre décrit les étapes pratiques réalisées pour la mise en œuvre du projet **Blogify**. Nous détaillerons l'architecture technique adoptée, les outils et environnements utilisés, les différentes fonctionnalités développées, ainsi que les tests effectués pour valider le bon fonctionnement du système.

3.2 Architecture technique

Le projet **Blogify** repose sur une architecture technique organisée en trois couches principales :

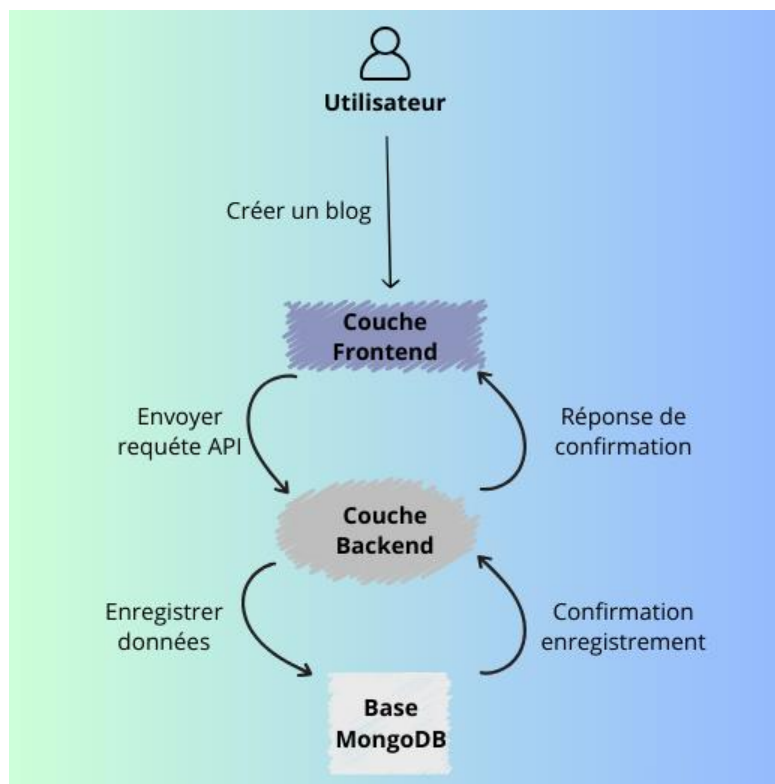


Figure 3.1 : Architecture 3-Tiers

3.2.1 Interaction entre les couches

Les couches communiquent de manière fluide grâce à des API sécurisées. Voici un exemple de flux :

1. L'**utilisateur** crée un blog via l'interface **frontend**.
2. Une **requête API** est envoyée au couche **backend**.
3. Le **backend** traite les données et les enregistre dans la base **MongoDB**.

Réalisation

4. Une **réponse** est renvoyée au **frontend** pour confirmer l'action.

Cette architecture garantit la modularité, facilite la maintenance, et permet d'ajouter des fonctionnalités futures sans compromettre la structure existante.

3.2.2 Couche Backend

- Framework : **Next.js 15**, pour la création de routes API.
- Sécurité : **ArcJet** pour la protection des données utilisateur et des API.
- Gestion des utilisateurs : **JWT Tokens** et **Jose** pour l'authentification et l'autorisation.
- Gestion des données : Communication avec la base de données via le module **Mongoose**.

3.2.3 Couche Frontend

- Framework UI : **shadcn/ui** et **Tailwind CSS** pour la conception d'une interface utilisateur moderne et réactive.
- Fonctionnalités frontales :
 - Recherche de blogs via une barre de recherche.
 - Navigation intuitive via des icônes et des catégories.
 - Éditeur de texte enrichi pour la création de blogs.

3.2.4 Couche Base de Données

- Type : **MongoDB**, pour une gestion flexible des données non relationnelles.
- Collections principales :
 - Users : Stocke les informations des utilisateurs.
 - Blogs : Gère les articles publiés.
 - Comments : Gère les commentaires des utilisateurs sur les blogs.

3.3 Configuration du système

3.3.1 Configuration de Node.js et des dépendances

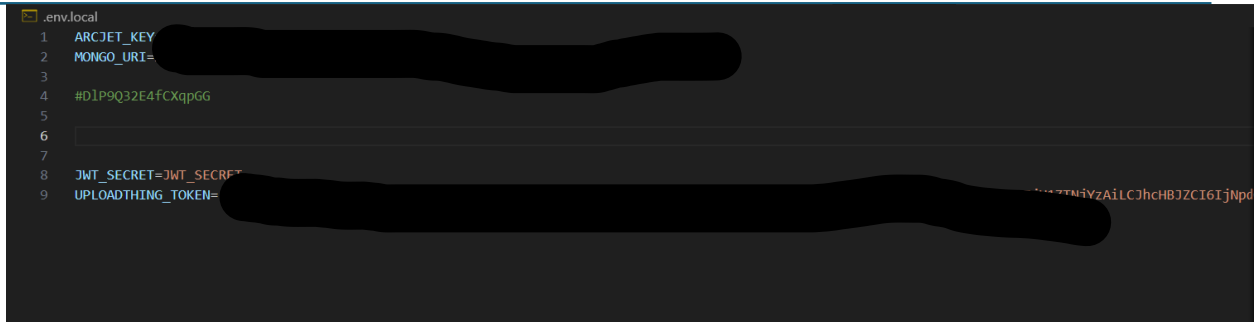
- Initialisation du projet Node.js

Un projet **Next.js** a été configuré à l'aide de la commande :

npx create-next-app@latest blogify

- Les dépendances nécessaires ont été installées avec npm :
 - Tailwind CSS et shadcn/ui pour la conception des interfaces utilisateur.

Réalisation



```
.env.local
1 ARCJET_KEY=
2 MONGO_URI=
3
4 #DlP9Q32E4fCXqpGG
5
6
7
8 JWT_SECRET=JWT_SECRET
9 UPLOADTHING_TOKEN=
```

Figure 2.3 : .env

3.3.2 Sécurité

La sécurité est un aspect crucial de Blogify. Voici les principales mesures mises en place :

- **Authentification :**

L'authentification utilise des **JWT Tokens** générés lors de la connexion de l'utilisateur.

- **Autorisation :**

- Mise en place de rôles utilisateurs : chaque utilisateur a accès uniquement aux sites et articles qu'il possède.
- Protection des routes sensibles via des middlewares qui vérifient les permissions avant de permettre l'accès.

- **Stockage sécurisé des clés:**

- Les clés API sensibles, comme les clés JWT, sont stockées dans des variables d'environnement protégées via **.env**.

2.3.3 Gestion des environnements

Le projet est configuré pour plusieurs environnements :

- **Développement :**

- Utilisation de données factices pour tester les fonctionnalités.

- **Production :**

- Déploiement sur Vercel, offrant des performances optimales et une scalabilité automatique.

Les configurations spécifiques à chaque environnement sont définies dans des fichiers séparés, garantissant une flexibilité maximale.

3.4 Environnement et outils de développement

3.4.1 Environnement matériel

Le développement du projet s'est fait sur un ordinateur Lenovo, équipé des spécifications suivantes :

- **Os:** Windows 11
- **Processeur:** Intel Core i5
- **RAM:** 12 Go
- **Disque dur:** 1024 Go SSD

3.4.2 Outils de développement

Plusieurs outils ont été utilisés pour mener à bien le développement du projet :

- **Visual Studio Code (VSCode) :**

Un éditeur de code puissant et léger, développé par Microsoft, utilisé pour écrire, déboguer et tester le code. Il est extensible via des plugins, prend en charge plusieurs langages et intègre des outils comme Git pour le contrôle de version [1].

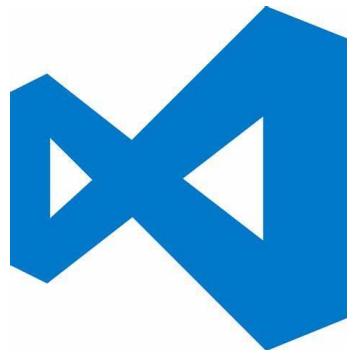


Figure 3.2 : VSCode

- **Git :**

Un outil de contrôle de version distribué qui permet de gérer les modifications de code, de travailler en collaboration sur des projets et de maintenir l'historique des versions. Il est essentiel pour organiser le travail d'équipe et assurer la traçabilité des changements [2].

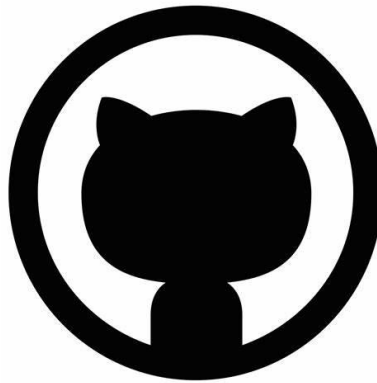


Figure 3.3 : Logo Git

- **MongoDB Compass :**

Une interface graphique pour MongoDB qui facilite la gestion et la visualisation des bases de données. Elle permet d'exécuter des requêtes, d'explorer les collections et d'analyser les données de manière intuitive [3].



Figure 3.4 : Logo MongoDB

- **Next.js**

Next.js est un framework React polyvalent permettant de créer des applications web modernes, rapides et optimisées pour le référencement. Il supporte le rendu côté serveur (SSR) et la génération statique (SSG), offrant une flexibilité pour diverses applications [4].



Figure 3.5 : Logo Next.js

- **ArcJet**

ArcJet offre une solution de sécurité avancée, incluant des mécanismes robustes pour l'authentification et l'autorisation. Il garantit une gestion fiable des accès utilisateurs et une protection renforcée des données [5].



Figure 3.6 : Logo ArcJet

- **Tailwind CSS**

Tailwind CSS est un framework CSS utilitaire qui permet un design rapide et entièrement personnalisable. Il facilite la création d'interfaces utilisateur modernes en utilisant des classes préconfigurées pour le style [6].



Figure 3.7 : Logo Tailwind CSS

- **UploadThing**

UploadThing simplifie la gestion et le traitement des fichiers téléchargés. C'est une solution robuste et facile à intégrer pour les développeurs travaillant avec des applications nécessitant des uploads fréquents [7].



Figure 3.8 : Logo UploadThing

3.5 Mise en Place des fonctionnalités

3.5.1 Développement des Modules

Le projet **Blogify** a été structuré en modules spécifiques, chacun conçu pour une fonction particulière. Le développement a respecté les bonnes pratiques de programmation pour garantir maintenabilité, sécurité et performance.

1) Authentification et Autorisation

- **Login** : Vérification des identifiants utilisateur avec JWT pour sécuriser les sessions.
- **Register** : Validation des champs utilisateur avec **Zod Conform**.

Réalisation

- **Protection des routes** : Middleware pour restreindre l'accès aux pages selon l'état connecté.

2) Gestion des Blogs

- Création de blogs :
 - Formulaire d'ajout avec titre, catégorie, et contenu enrichi.
 - Upload sécurisé des images via **UploadThing**.
- Recherche et filtrage :
 - Barre de recherche pour trouver des blogs par titre ou catégorie.
 - Filtres prédéfinis pour les catégories.
- Mise à jour et suppression des blogs :
 - Possibilité de modifier ou supprimer un blog pour son auteur.

3) Interaction Communautaire

- Ajout de commentaires aux blogs.
- Affichage du nombre de votes positifs (upvotes).

3.5.2 Implémentation des fonctionnalités clés

L'implémentation des fonctionnalités clés du système a suivi une approche modulaire, où chaque module est développé et testé individuellement avant d'être intégré au système global. Voici les principales fonctionnalités:

- **Authentification et Autorisation** :
 - **Enregistrement utilisateur** : Cette fonctionnalité permet aux nouveaux utilisateurs de créer un compte en fournissant un nom, une adresse e-mail, et un mot de passe. Les informations sont stockées de manière sécurisée dans la base de données après vérification et chiffrement du mot de passe.
 - **Connexion utilisateur (Login)** : Les utilisateurs enregistrés peuvent se connecter à leur compte en utilisant leurs identifiants. Une fois validés, un jeton d'authentification sécurisé (JWT) est généré pour les sessions futures.
- **Gestion des Blogs**:
 - **Création de blogs** : Les utilisateurs connectés peuvent créer un article en fournissant un titre, une catégorie, et du contenu. Ils ont également la possibilité d'ajouter une image de couverture pour illustrer leur blog.

Réalisation

- **Mise à jour et suppression des blogs** : Les auteurs peuvent modifier ou supprimer leurs articles existants, en gardant le contrôle total sur leurs contenus.
- **Interaction Communautaire**
 - **Ajout de commentaires** : Les utilisateurs peuvent commenter les blogs pour exprimer leurs avis ou poser des questions. Chaque commentaire est associé à un utilisateur et horodaté pour un meilleur suivi.
 - **Recherche et filtrage** : Une barre de recherche permet aux utilisateurs de trouver des blogs par titre ou catégorie. Des filtres spécifiques offrent une navigation ciblée et rapide.

3.5.3 Tests et Validation

Les tests ont été réalisés pour vérifier que chaque fonctionnalité fonctionne correctement et répond aux spécifications. Les tests incluent :

A. Enregistrement Utilisateur

Cette étape teste la fonctionnalité d'inscription, où un utilisateur saisit ses informations et crée un compte. Après validation des champs, les données sont ajoutées à la base de données.

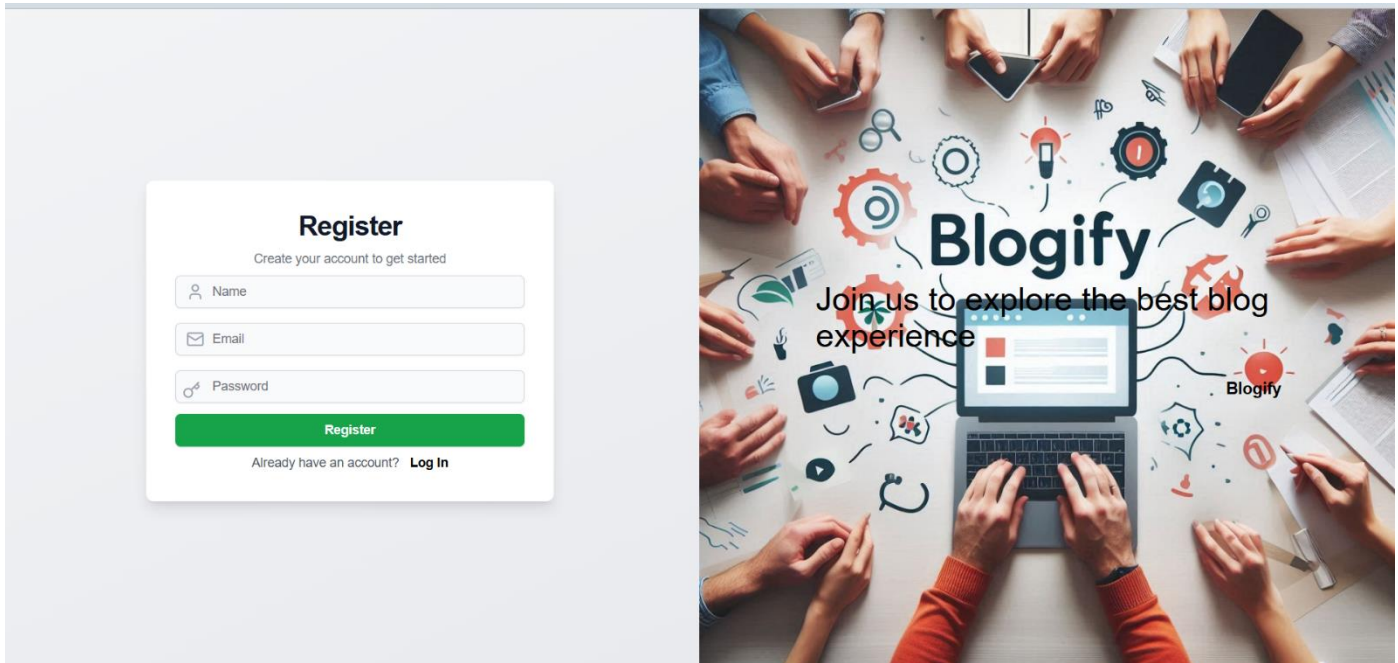


Figure 3.9 : Test d'enregistrement utilisateur

- **Resultat** : Validation des champs et hash du mot de passe

Réalisation

B. Connexion Utilisateur (Login)

Test de la connexion d'un utilisateur avec des identifiants valides. Une réponse contenant un jeton d'authentification est attendue en retour.

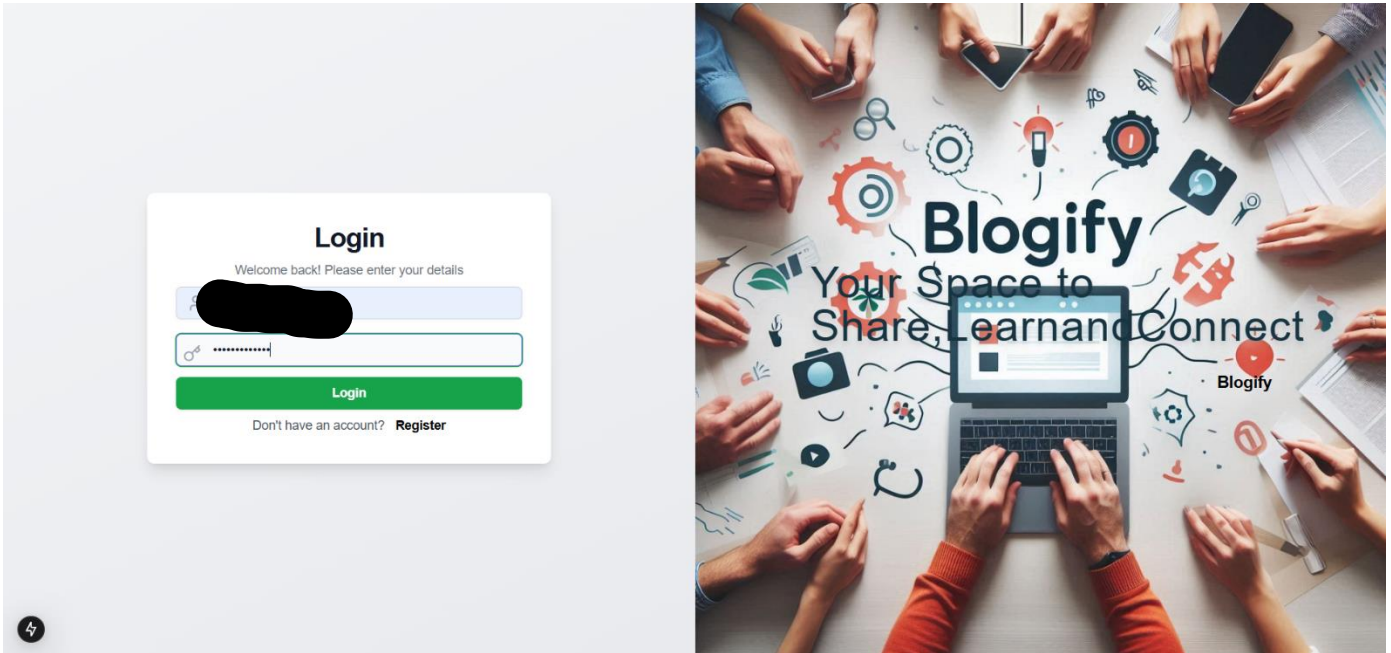


Figure 3.10 : Test de Connexion (Login)

Resultat : Retour d'un token valide

C. Création de Blog

Vérification de l'ajout d'un blog avec des informations valides (titre, contenu, image). Le blog est sauvegardé dans la base de données et affiché dans la liste des blogs.

Réalisation



Figure 3.11 : Test de Création de Blog

Resultat : Blog ajouté à la base de données

D. Recherche de Blog

Test de la recherche d'un blog par mot-clé ou catégorie. Les résultats doivent afficher uniquement les blogs correspondants au filtre.

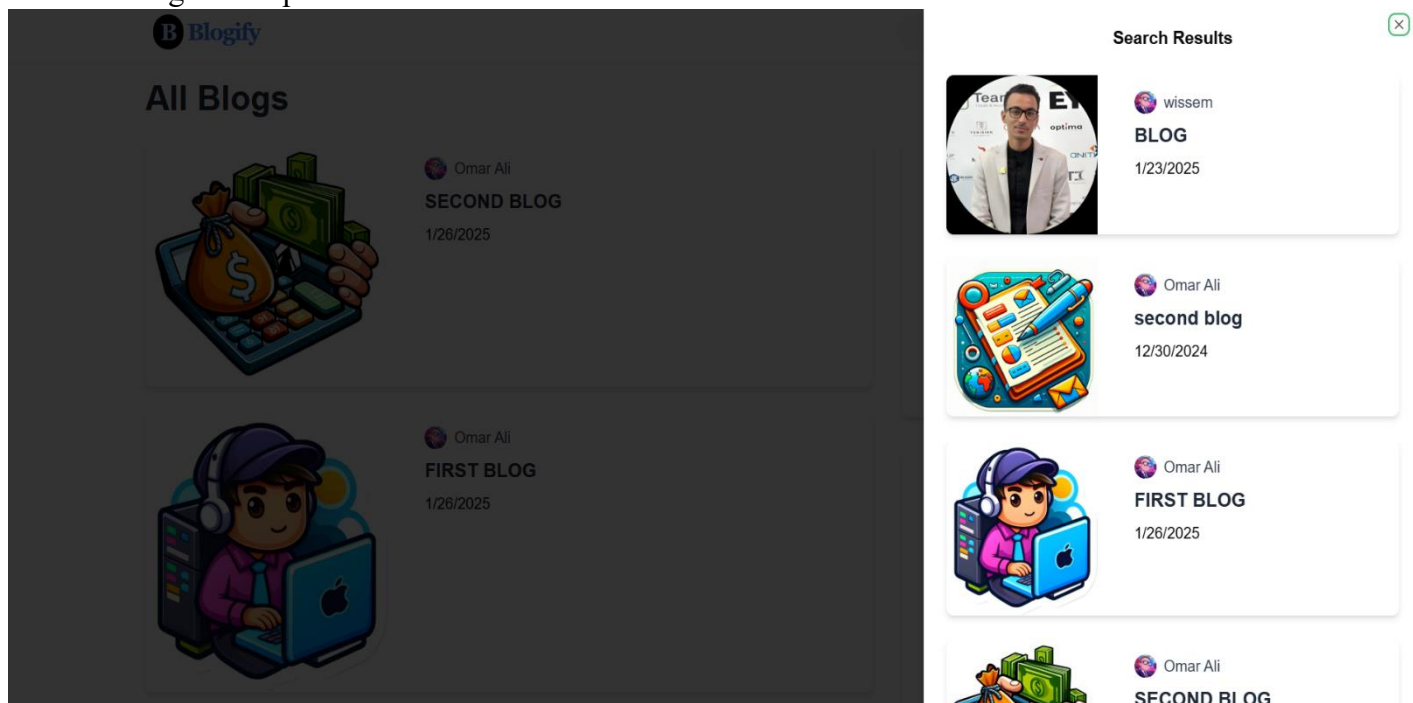


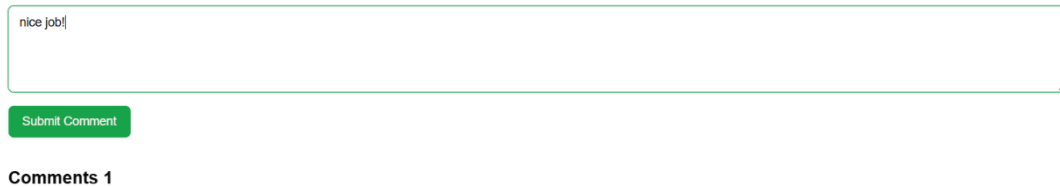
Figure 3.12 : Test de Recherche de Blog

Réalisation

Resultat : Affichage des résultats correspondants

E. Ajout de Commentaire

Validation de l'ajout d'un commentaire à un blog existant. Le commentaire est enregistré avec l'utilisateur et l'heure, puis affiché sous l'article.



The image shows a web form for adding a comment. It consists of a large rectangular text input field with a light gray border. Inside the field, the text "nice job!" is visible. Below the input field is a green button with the text "Submit Comment" in white. Underneath the button, the text "Comments 1" is displayed.

Figure 3.13 : Test d'ajout de Commentaire

Resultat : Commentaire ajouté à la base de données

3.6 Conclusion

Ce chapitre a détaillé les étapes concrètes de réalisation du projet **Blogify**, en insistant sur l'architecture technique, les outils utilisés, et l'implémentation des fonctionnalités principales. Les tests effectués ont validé la conformité du système avec les spécifications initiales, assurant ainsi une expérience utilisateur optimale.

Conclusion générale et perspectives

Ce projet, **Blogify**, a été une expérience enrichissante qui a permis de combiner plusieurs compétences techniques et méthodologies de développement. En intégrant des outils modernes comme **Next.js**, **MongoDB**, **shadcn/ui**, et **Git**, nous avons pu concevoir une plateforme collaborative innovante, axée sur le partage de connaissances et l'interaction communautaire.

Tout au long de ce projet, nous avons suivi une approche structurée, en commençant par une analyse approfondie des besoins, la conception d'une architecture robuste, et la mise en œuvre des fonctionnalités clés comme l'authentification sécurisée, la création et la gestion des blogs, ainsi que l'intégration de catégories et de commentaires interactifs.

Le projet a également permis de maîtriser des concepts avancés tels que :

- L'utilisation d'une architecture en couches (Frontend, Backend, et Base de données) pour une meilleure modularité.
- L'application de techniques de sécurité, y compris l'authentification via JWT et le stockage sécurisé des données utilisateur.
- La gestion efficace des versions grâce à **Git**, favorisant un développement collaboratif fluide.

D'un point de vue personnel et professionnel, ce projet a renforcé notre capacité à travailler en équipe, à résoudre des problèmes complexes, et à respecter les exigences d'un projet dans un cadre méthodologique agile comme Scrum.

En conclusion, **Blogify** n'est pas seulement une plateforme technique, mais aussi une réalisation pratique de nos compétences acquises, marquant une étape importante dans notre parcours académique et professionnel. Ce projet pourrait servir de base pour des améliorations futures, comme l'intégration de fonctionnalités avancées (par exemple, l'intelligence artificielle pour les recommandations ou l'analyse des contenus) et l'expansion vers d'autres domaines d'applications.

Netographie

- [1] « Visual Studio Code. » [Accès le 20-04-2023]. (), adresse : <https://www.hostinger.com/tutorials/what-is-vs-code>
- [2] « Git. » [Accès le 14-10-2020]. (), adresse : <https://medium.com/swlh/an-introduction-to-git-and-github-22ecb4cb1256>
- [3] « MongoDB Compass . » [Accès le 14-10-2021]. (), adresse : <https://www.tutorialsteacher.com/mongodb/mongodb-compass>
- [4] « Next.js. » [Accès le 31-03-2023]. (), adresse : <https://www.geeksforgeeks.org/nextjs/>
- [5] « ArcJet. » [Accès le 14-10-2020]. (), adresse : <https://docs.arcjet.com/get-started>
- [6] « TailwindCSS. » [Accès le 24-08-2023]. (), adresse : <https://www.geeksforgeeks.org/what-is-tailwind-css/>
- [7] « UploadThing. » [Accès le 11-07-2023]. (), adresse : <https://uploadthing.com>