**Ministère de l’Enseignement Supérieur, de la Recherche Scientifique et des Technologies de l'Information et de la Communication**

**DIRECTION GÉNÉRALE DES ÉTUDES TECHNOLOGIQUES**

**Institut Supérieur des Études Technologiques de Béja**

**Département Technologies de l’informatique**

**Rapport de Stage d’Initiation**

Siteweb Blog full stack

**Effectué à :**

**Entreprise : SILICOD**

**Elaboré par** : Hamza Benarfa

**Encadré Par** : MR Brahim Dahmani

**Année universitaire : 2022/2023**

**Remerciements**

Je souhaiterais exprimer ma profonde reconnaissance à tous ceux qui m'ont soutenu durant la réalisation de ce projet, ainsi qu'à ceux qui ont contribué à sa conception, son élaboration et sa réalisation.

Tout d'abord, je souhaiterais remercier mon encadrant professionnel au sein de l'entreprise SILICOD, Monsieur Brahim Dahmani, pour ses conseils avisés, son soutien tout au long du déroulement du projet et pour son bienveillant engagement à mon égard.

De plus, je tiens à remercier le Professeur Madame Samia Ghanmi enseignant universitaire du département informatique pour son aide précieuse, sa disponibilité constante et sa patience.

Je tiens également à remercier chaleureusement tous mes professeurs et tous les enseignants de ISET pour leur détermination à nous fournir une formation de qualité.

**Sommaire**

[Introduction generale 6](#_Toc128602939)

[Chapitre 1 : Présentation de la société d'accueil 7](#_Toc128602940)

[1. Présentation de la société 7](#_Toc128602941)

[2. Les services de la société 7](#_Toc128602942)

[3. L'organigramme de la société 7](#_Toc128602943)

[4. Contexte de stage et objectifs 7](#_Toc128602944)

[Chapitre 2 : les notions théoriques 8](#_Toc128602945)

[1. Environnement matériel 8](#_Toc128602946)

[5. Environnement logiciel 8](#_Toc128602947)

[6. Technologie et Langage des programation utilise: 10](#_Toc128602948)

[Chapitre 3 : les tâches effectuées 14](#_Toc128602949)

[Conclusion générale 20](#_Toc128602950)

[Bibliographie 21](#_Toc128602951)

[Nétographie 22](#_Toc128602952)

Liste des figures

[Figure 1: MS Word logo 3](#_Toc128605221)

[Figure 2:POSTMAN logo 3](#_Toc128605222)

[Figure 3: Google Chrome logo 4](#_Toc128605223)

[Figure 4: Microsoft PowerPoint logo 4](#_Toc128605224)

[Figure 5: VsCode logo 4](#_Toc128605225)

[Figure 6:Architecture de mern 5](#_Toc128605226)

[Figure 7: Expressjs logo 6](#_Toc128605227)

[Figure 8:ecmascript logo 6](#_Toc128605228)

[Figure 9: reactjs logo 6](#_Toc128605229)

[Figure 10: JSON logo 6](#_Toc128605230)

[Figure 11:Nodejs logo 7](#_Toc128605231)

[Figure 12: MongoDB logo 7](#_Toc128605232)

[Figure 13:Architecture de REST webservice 8](#_Toc128605233)

[Figure 14: Architecture de MVC 8](#_Toc128605234)

[Figure 15: page posts 9](#_Toc128605235)

[Figure 16: page d'une certaine post 9](#_Toc128605236)

[Figure 17: page de connection 10](#_Toc128605237)

[Figure 18: creaction du compte 10](#_Toc128605238)

[Figure 19: la page de profile de utilisateur 11](#_Toc128605239)

[Figure 20: formulaire d'ajout un poste 11](#_Toc128605240)

[Figure 21: le bouton de modifier de poste 12](#_Toc128605241)

[Figure 22: la formulaire de modification du poste 12](#_Toc128605242)

[Figure 23:formulaire est rempli 12](#_Toc128605243)

[Figure 24: le bouton qui supprimer le post 13](#_Toc128605244)

[Figure 25: la page d'accueil 13](#_Toc128605245)

[Figure 26 :la page d'à propos 14](#_Toc128605246)

[Figure 27: la page de contact 14](#_Toc128605247)

Liste des tableaux

[Table 1 : Environnement matériel 3](#_Toc128605535)

[Table 2 : Environnement logiciel 4](#_Toc128605536)

[Table 3 : Technologie et Langage des programation utilise 7](#_Toc128605537)

# Introduction generale

Au cours de mon stage d'une durée de 30 jour au sein du département informatique de l'entreprise Silicod, j'ai eu l'opportunité de me former à l'utilisation du MERN Stack. Cette pile technologique regroupe MongoDB, Express, React et Node.js, des outils indispensables pour créer des applications web modernes et complètes.

Dans ce rapport, nous présentons les résultats de notre travail qui se compose de trois chapitres distincts. Dans le premier chapitre, nous décrivons le contexte de notre stage ainsi que les objectifs fixés. Dans le deuxième chapitre, nous exposons les notions théoriques nécessaires à la compréhension de notre travail. Enfin, dans le troisième chapitre, nous présentons les tâches que nous avons effectuées et les résultats obtenus en utilisant le MERN Stack. Ce stage a été une opportunité exceptionnelle pour améliorer mes compétences en développement web et j'espère que ce rapport sera utile pour ceux qui souhaitent en apprendre davantage sur cette technologie.

# Chapitre 1 : Présentation de la société d'accueil

Introduction :

Dans ce chapitre, nous présentons l'entreprise dans laquelle nous avons effectué notre stage de formation sur les technologies MERN. Nous exposons les objectifs et les tâches à accomplir au sein de cette société.

## Présentation de la société

La société d'accueil de notre stage est Silicod, une entreprise de conseil en développement de logiciels basée à Siliana. Fondée en 2019, Silicod crée des produits et services numériques open source pour les start-ups et les entreprises établies. L'entreprise dispose d'une équipe technique centrale qui offre des services de développement backend et frontend. Silicod utilise un large éventail de technologies, telles que Ruby on Rails, GraphQL, PostgreSQL, Redis, Elasticsearch, React et Redux, pour créer des applications performantes et fiables avec une culture d'ingénierie saine qui suit les principes du déploiement continu.

## Les services de la société

Silicod offre une gamme de services de développement de logiciels, notamment des services de développement backend et frontend, des services de développement mobile, des services de développement de sites web et des services de conseil en technologie. L'entreprise propose également des solutions d'hébergement et de maintenance pour les applications développées.

## L'organigramme de la société

Silicod est dirigé par une équipe de direction composée de membres expérimentés du secteur de la technologie. L'équipe technique est dirigée par un directeur technique et se compose de développeurs web et mobile, de concepteurs UX/UI et d'ingénieurs DevOps.

## Contexte de stage et objectifs

Dans le cadre de notre stage de formation, Silicod nous a accueillis dans ses locaux pour nous former sur les technologies MERN. Notre objectif est de renforcer et d'approfondir nos compétences en matière de développement web et d'hébergement. Nous sommes ravis de travailler avec Silicod et d'acquérir des compétences précieuses pour notre avenir professionnel.

Conclusion

Ce chapitre a présenté la société d'accueil de notre stage, Silicod. Nous avons exposé les services offerts par l'entreprise et présenté son organigramme. Enfin, nous avons expliqué le contexte de notre stage et les objectifs à atteindre. Le chapitre suivant détaillera les tâches à accomplir pour réaliser ces objectifs.

# Chapitre 2 : les notions théoriques

Introduction :

Environnement matériel

|  |  |
| --- | --- |
| Ordinateur |  |
| CPU | I5 |
| RAM | 8Go |
| Disq Dur | 1To |
| System d’exploitation | Windows 10 |

Table 1 : Environnement matériel

Environnement logiciel

|  |  |
| --- | --- |
| Redaction du rapport:    Figure 1: MS Word logo | MS Word est un logiciel de traitement de texte populaire qui permet aux utilisateurs de créer, modifier et formater des documents texte. |
| Test Api :  Postman : le guide complet de l'outil indispensable au développement web  Figure 2:POSTMAN logo | Postman est un outil de développement d'API qui permet aux développeurs de tester et de déboguer facilement des API. |
| Test frontend  Fichier:Logo-Google Chrome.jpg — Wikimini, l'encyclopédie pour enfants  Figure 3: Google Chrome logo | Google Chrome est un navigateur web populaire développé par Google. Il est utilisé pour accéder à des sites web et des applications en ligne. |
| Presentation  Using Custom PowerPoint Animations and Transitions  Figure 4: Microsoft PowerPoint logo | PowerPoint est un logiciel de présentation développé par Microsoft. Il permet aux utilisateurs de créer des diaporamas pour présenter des informations de manière visuelle et attrayante. |
| IDE:  Visual Studio Code png images | PNGWing  Figure 5: VsCode logo | Visual Studio Code (VS Code) est un éditeur de code source développé par Microsoft. Il est utilisé par les développeurs pour écrire et déboguer du code dans une variété de langages de programmation. |

Table 2 : Environnement logiciel

Technologie et Langage des programation utilise:

Introduction :

Le MERN Stack est un ensemble de technologies de développement web qui se compose de MongoDB, Express, React et Node.js. Chaque technologie joue un rôle important dans la construction d'applications web modernes et performantes.

On choisi MERN pour plusieurs raisons :

* Performance: MERN utilise JavaScript pour les composants côté client et côté serveur, ce qui permet de créer des applications web hautement performantes.
* Réutilisabilité : La structure MERN est très modulaire, ce qui signifie que les composants peuvent être réutilisés dans différentes parties de l'application, ce qui permet de gagner du temps et des efforts.
* Flexibilité : MERN peut être utilisé pour créer une grande variété d'applications web, des petites applications monopages aux grandes applications complexes.

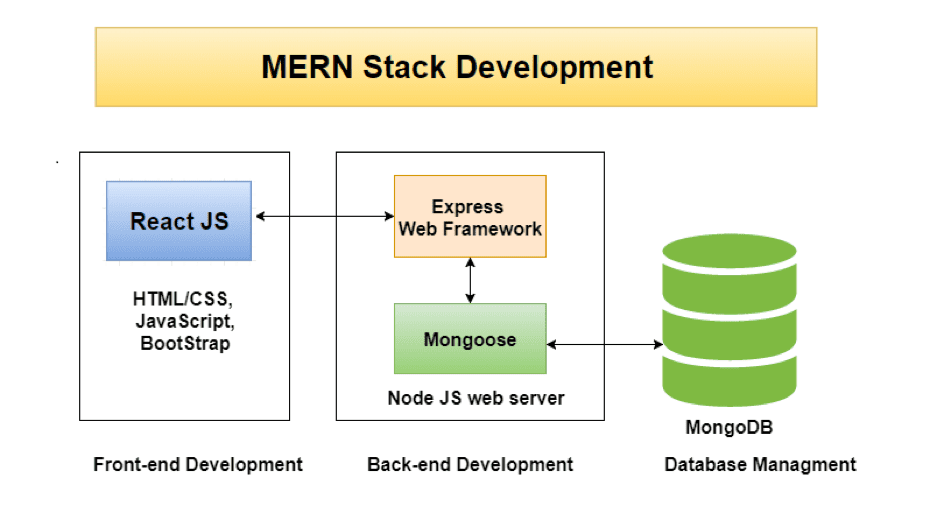


Figure 6:Architecture de mern

|  |  |
| --- | --- |
| Express - Node.js web application framework  Figure 7: Expressjs logo | Express.js est un framework open-source pour Node.js, conçu pour faciliter la création d'applications web et d'API. Il fournit un ensemble d'outils et de fonctionnalités pour créer rapidement des applications web avec Node.js, notamment des routes, des gestionnaires de middleware, des modèles et des vues. |
| C:\Users\hamza\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\98D41C80.tmp  Figure 8:ecmascript logo | ES6, également connu sous le nom d'ECMAScript 2015, est la sixième version de la norme ECMAScript pour le langage de programmation JavaScript. Elle apporte un grand nombre de nouvelles fonctionnalités et améliorations, notamment de nouveaux mots-clés, des classes, des modules, des fonctions fléchées, des itérateurs et des générateurs. |
| Your custom development solution with React JS| Ubidreams  Figure 9: reactjs logo | React.js est une bibliothèque JavaScript open-source développée par Facebook. Elle est utilisée pour la création d'interfaces utilisateur pour des applications web interactives et dynamiques. React.js utilise une approche basée sur les composants pour la création d'interfaces utilisateur réutilisables et maintenables. |
| What is JSON? The most important questions explained simply  Figure 10: JSON logo | JSON (JavaScript Object Notation) est un format de données léger et facile à lire utilisé pour l'échange de données entre les applications. Il est basé sur la syntaxe JavaScript et est souvent utilisé pour transmettre des données structurées sur le Web. |
| Interpreteurs de commandes  C:\Users\hamza\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\855C89F4.tmp  Figure 11:Nodejs logo | Node.js est une plateforme logicielle open-source basée sur le moteur JavaScript V8 de Google. Elle permet aux développeurs de créer des applications en utilisant JavaScript pour le côté serveur, au lieu de l'utiliser uniquement pour le côté client. |
| Base des donnees :  How to get started with MongoDB in 10 minutes | by Navindu Jayatilake |  We've moved to freeCodeCamp.org/news | Medium  Figure 12: MongoDB logo | MongoDB est une base de données NoSQL populaire qui permet de stocker des données sous forme de documents JSON flexibles. Elle est conçue pour être évolutive, performante et facile à utiliser, offrant une grande flexibilité dans la gestion des données. |

Table 3 : Technologie et Langage des programation utilise

Architecture du system :

L'architecture système utilisée pour le projet est le style REST, qui permet de créer des services web performants, évolutifs, fiables et faciles à entretenir. Ce style architectural permet une communication simple et flexible entre les différentes parties d'un système distribué, en utilisant des méthodes standardisées telles que GET, POST, PUT et DELETE pour accéder et modifier les ressources.

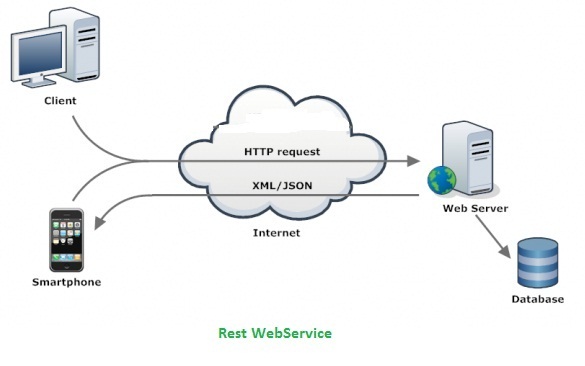


Figure 13:Architecture de REST webservice

Le système a été conçu en utilisant le modèle de conception MVC (Modèle-Vue-Contrôleur). Le modèle représente les données et l'état de l'application web. Habituellement, ces données sont stockées dans une base de données ou des fichiers. La vue est l'interface graphique de l'application, composée de code HTML et CSS pour afficher les données sans les modifier. Le contrôleur relie la vue et le modèle. Il gère les interactions avec l'utilisateur et détermine les actions à effectuer pour chaque commande. Cette architecture permet de séparer les différents composants de l'application pour une meilleure maintenance et évolutivité.

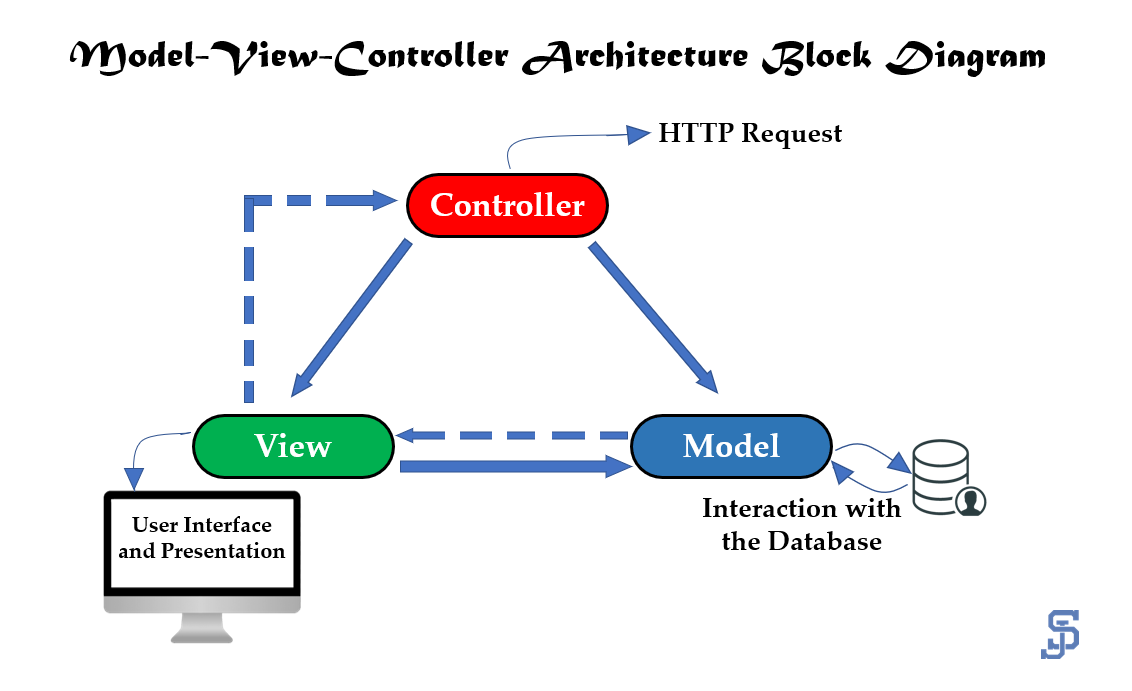


Figure 14: Architecture de MVC

# Chapitre 3 : les tâches effectuées

Afficher les post

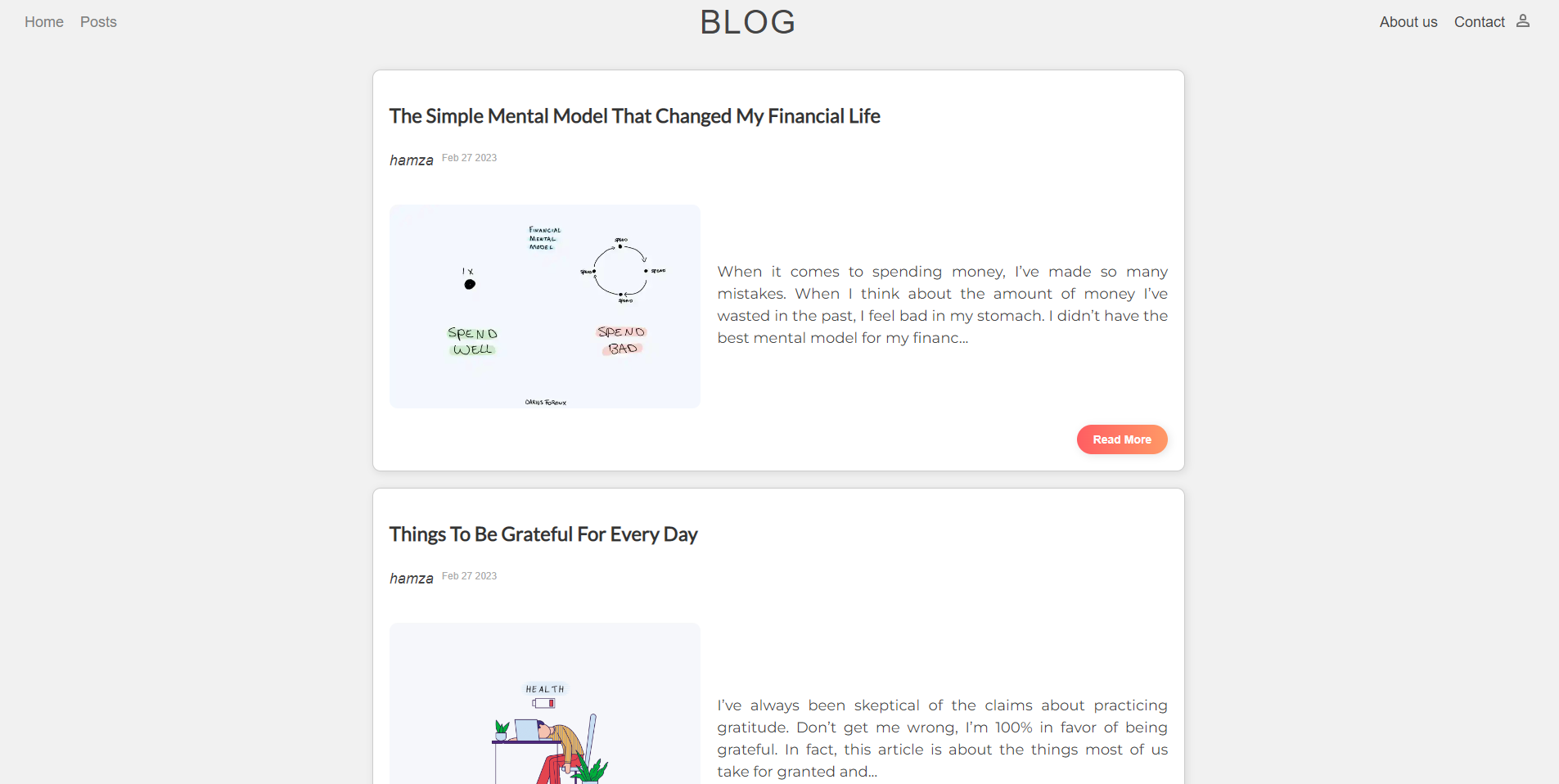


Figure 15: page posts

Afficher un post

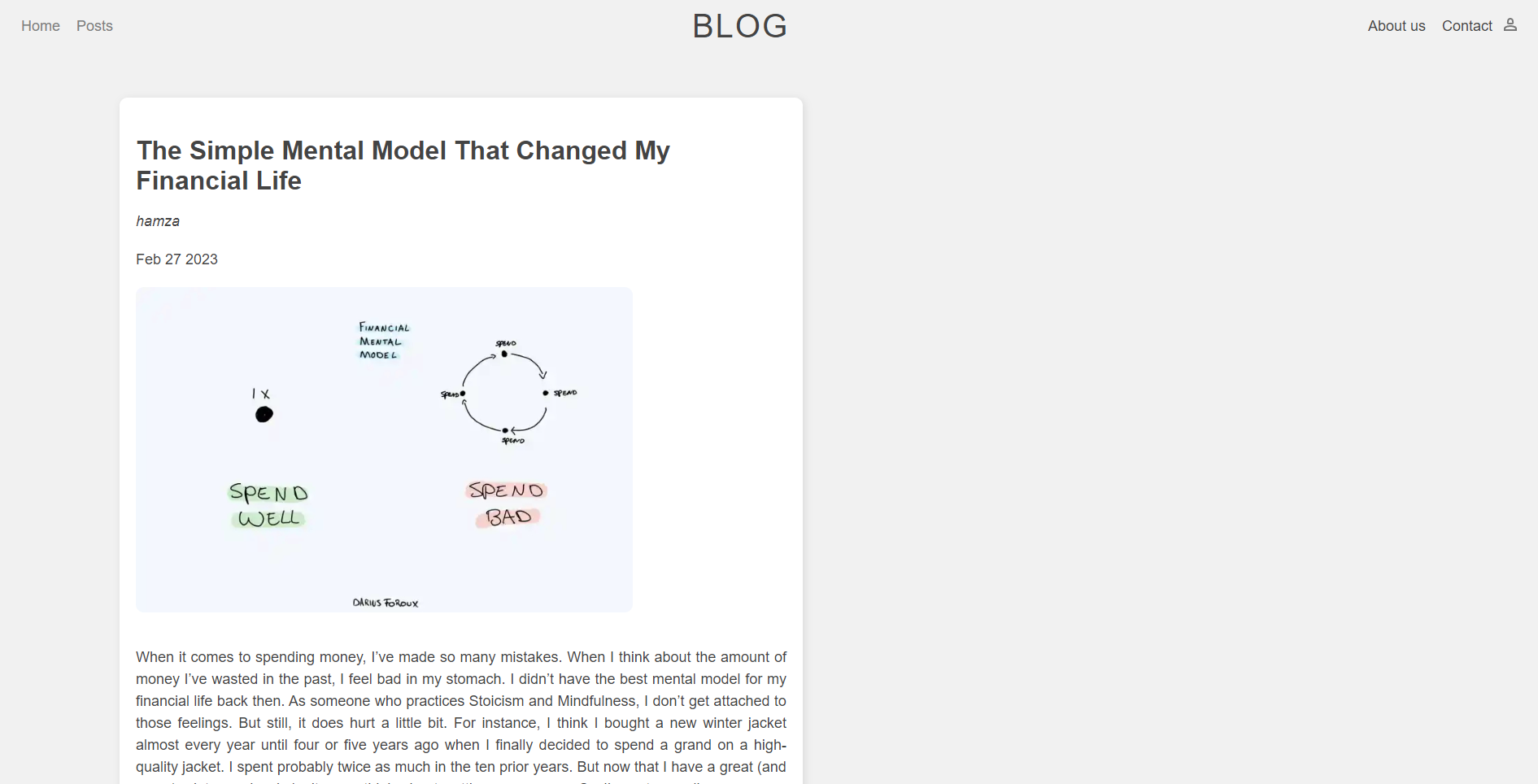


Figure 16: page d'une certaine post

Page connexion

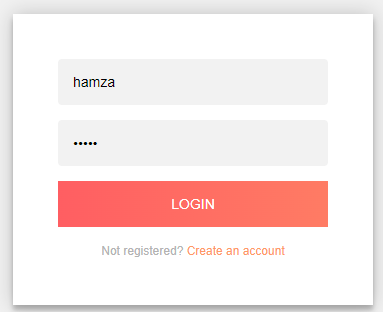


Figure 17: page de connection

Page de création un compte

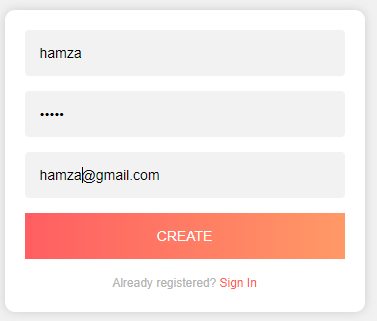


Figure 18: creaction du compte

Page de profil

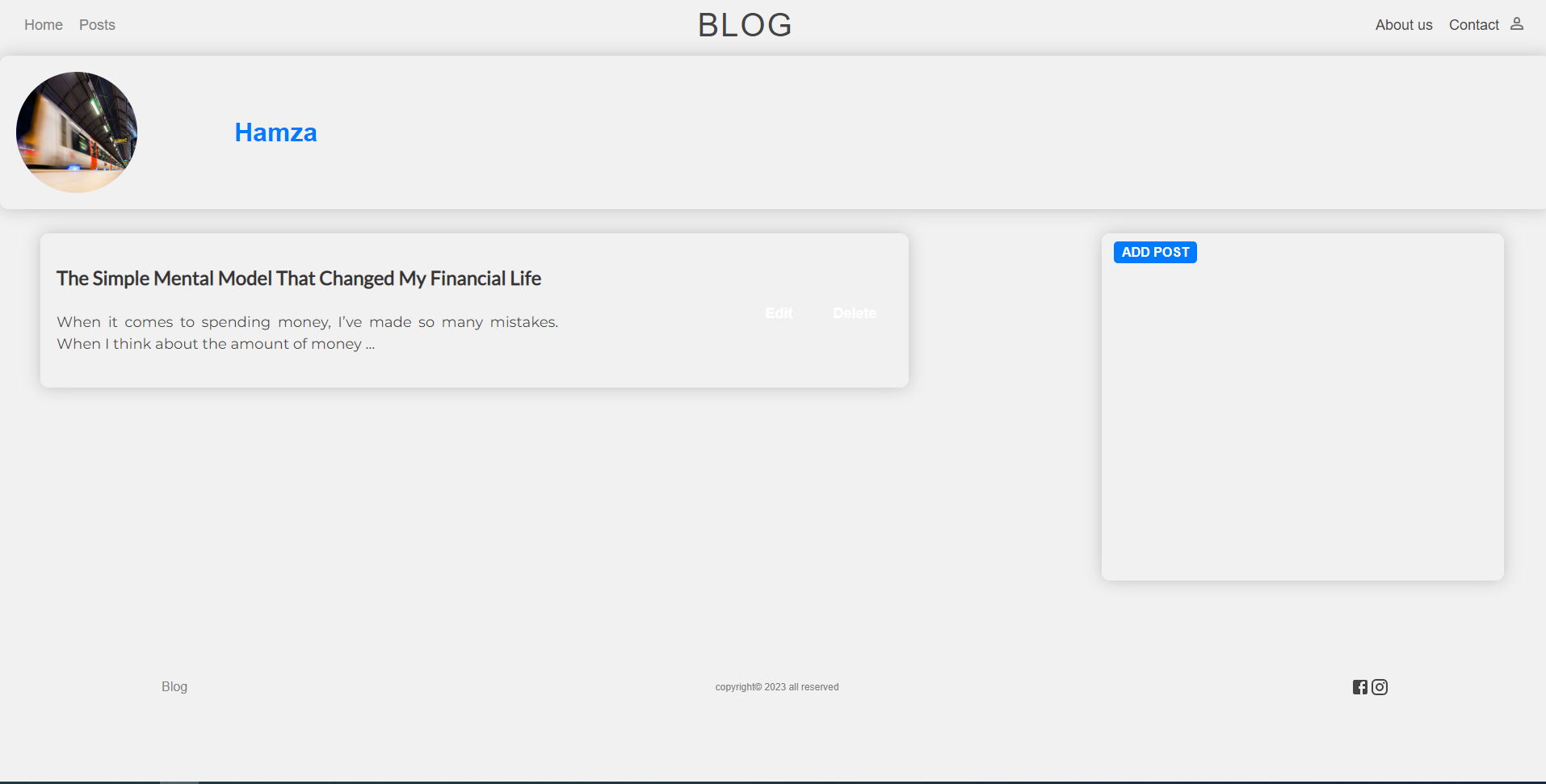


Figure 19: la page de profile de utilisateur

Formulaire d’ajout un poste

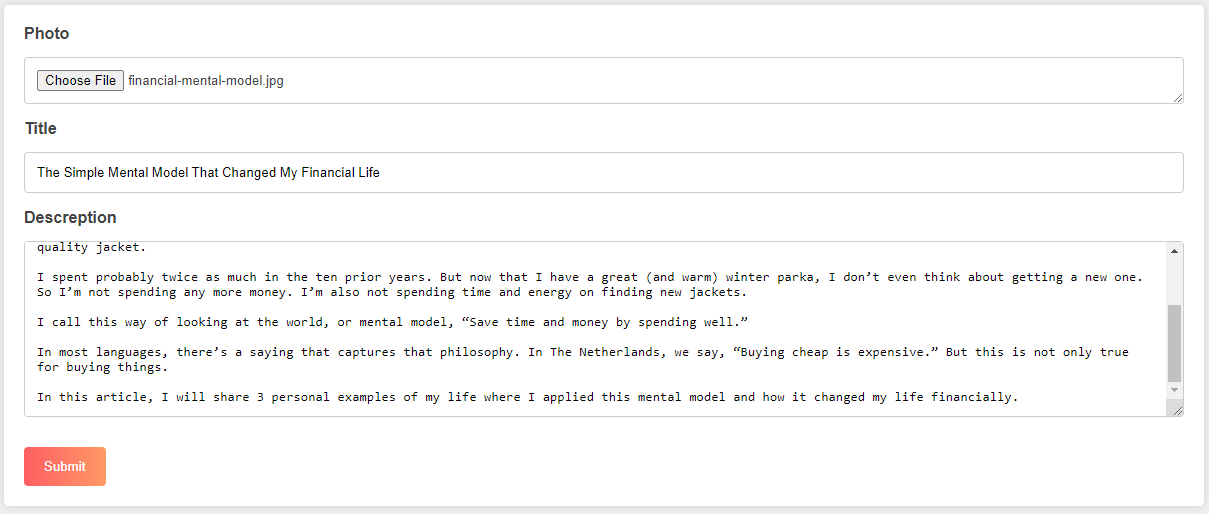


Figure 20: formulaire d'ajout un poste

Formulaire de mise à jour de poste

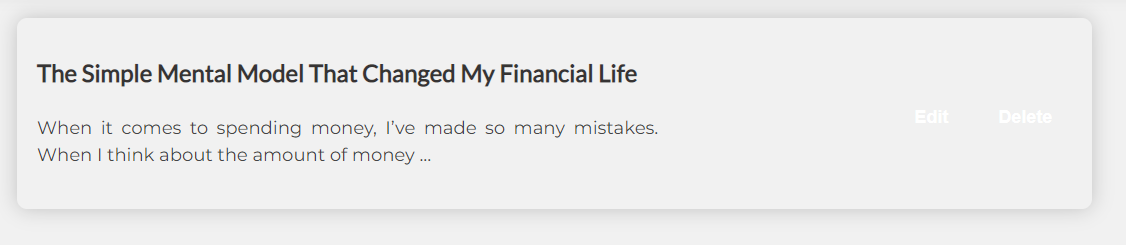


Figure 21: le bouton de modifier de poste

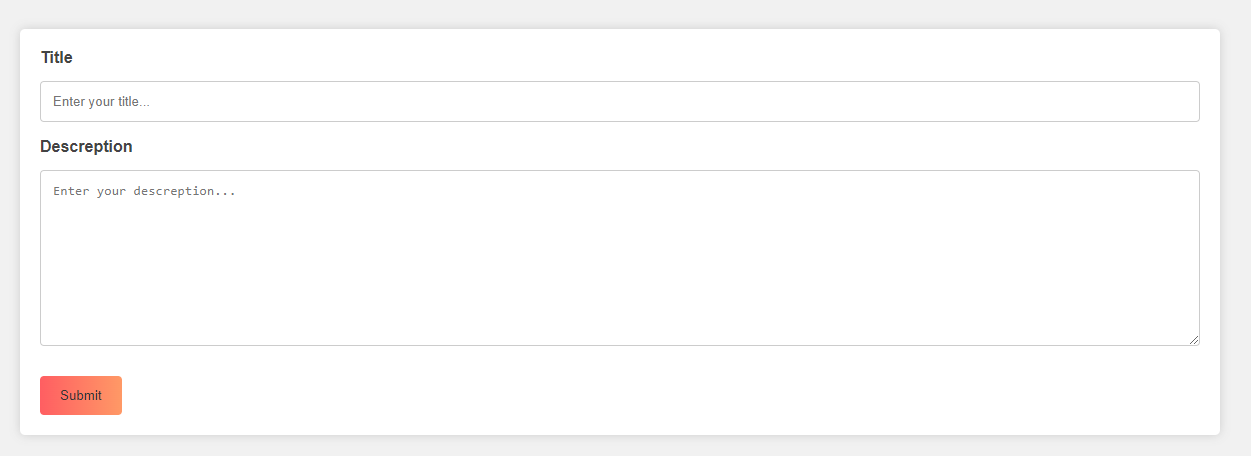


Figure 22: la formulaire de modification du poste

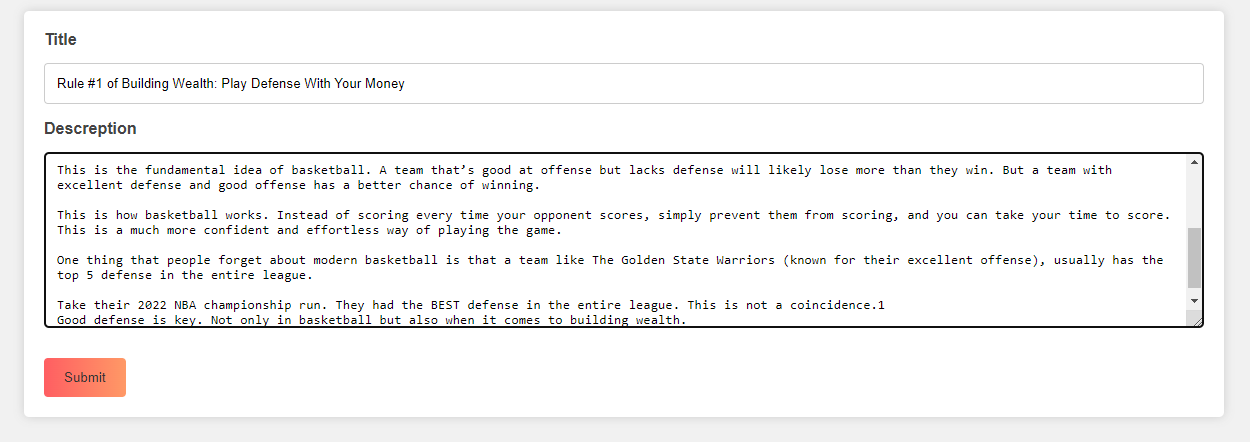


Figure 23:formulaire est rempli

Suppression de post

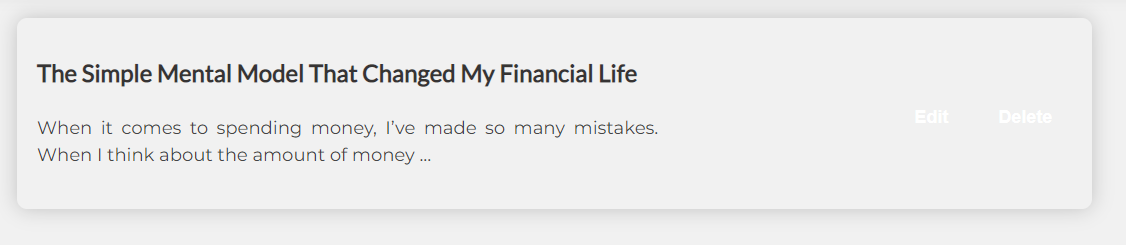


Figure 24: le bouton qui supprimer le post

Home Page

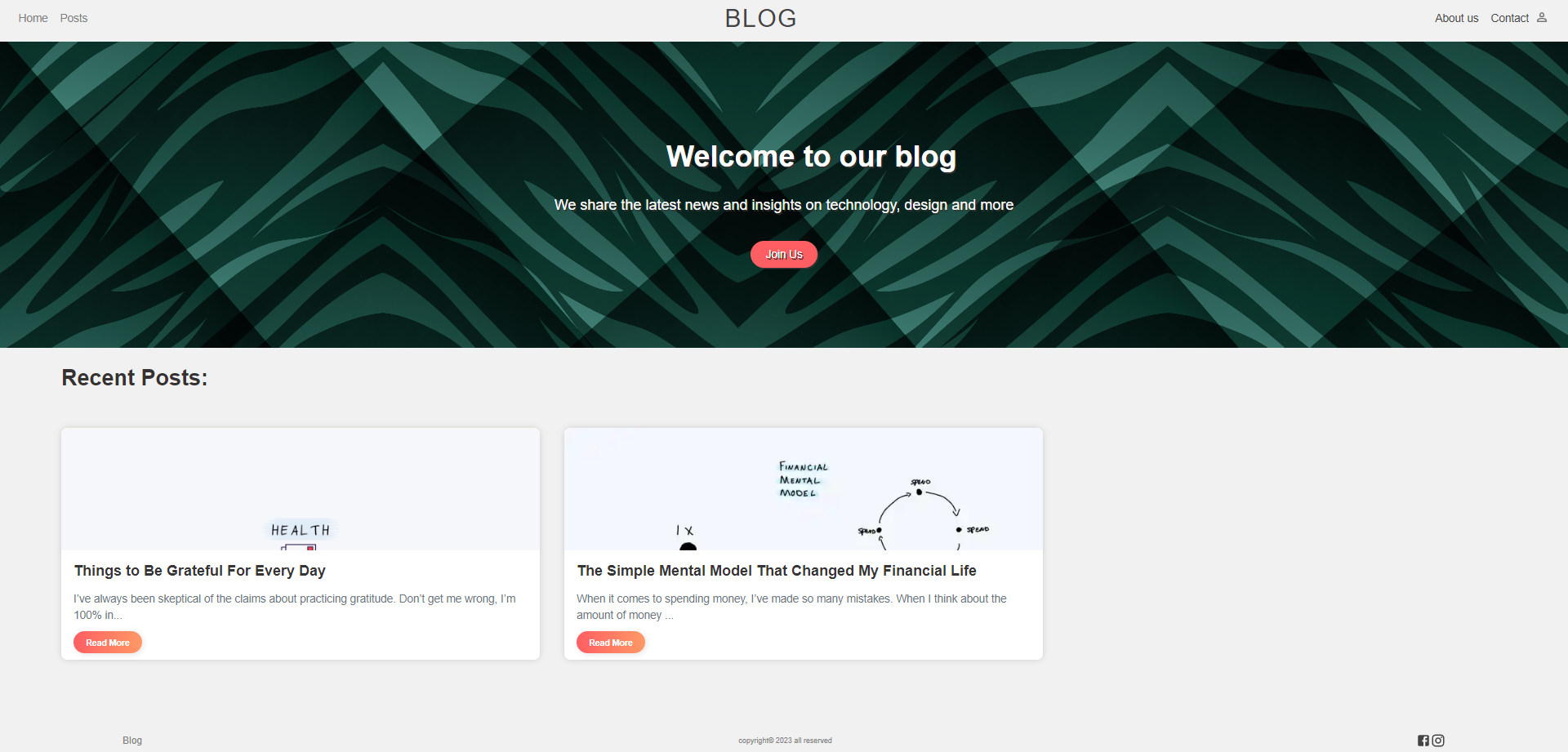


Figure 25: la page d'accueil

About Page

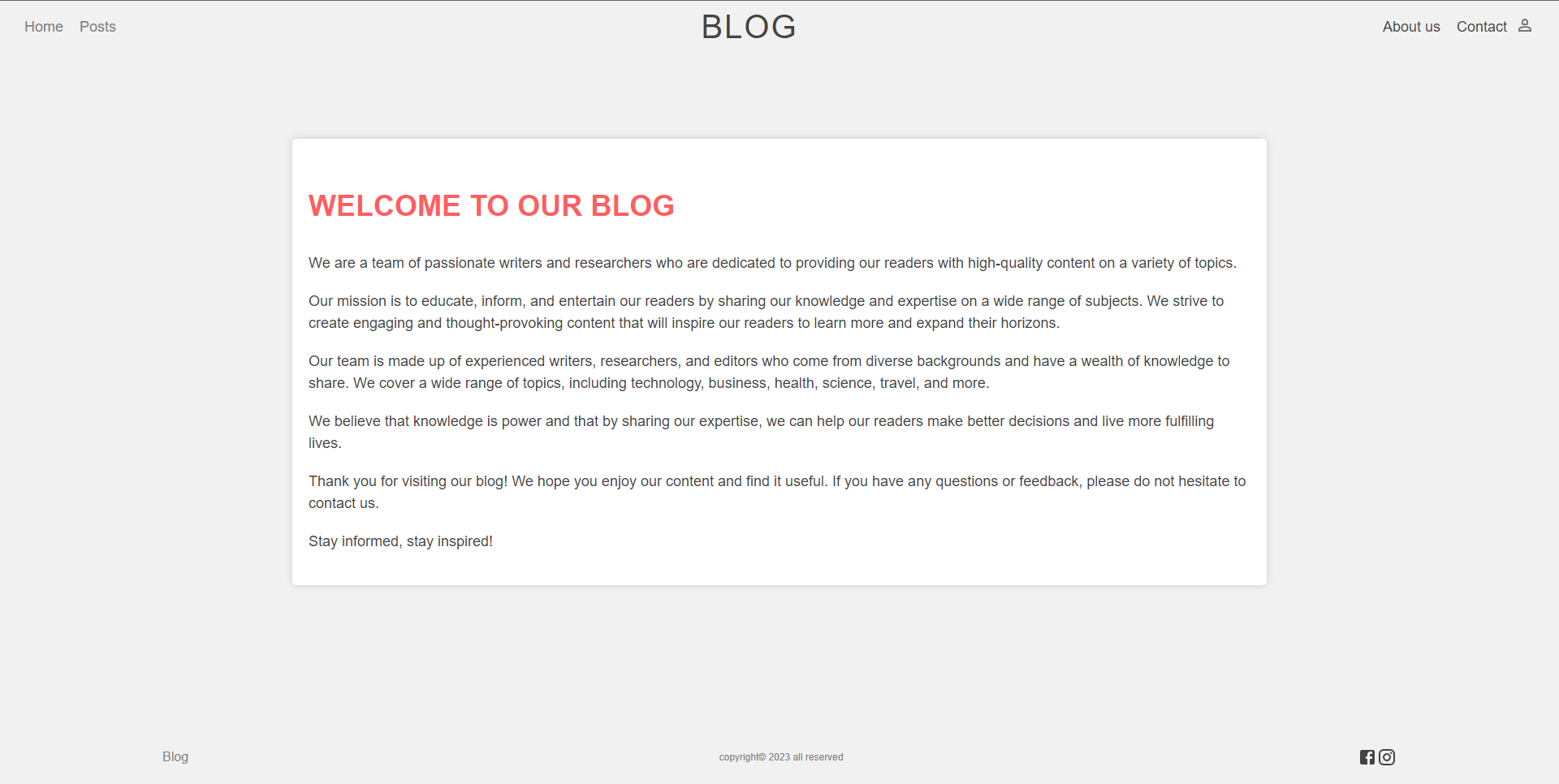


Figure 26 :la page d'à propos

Contact Page

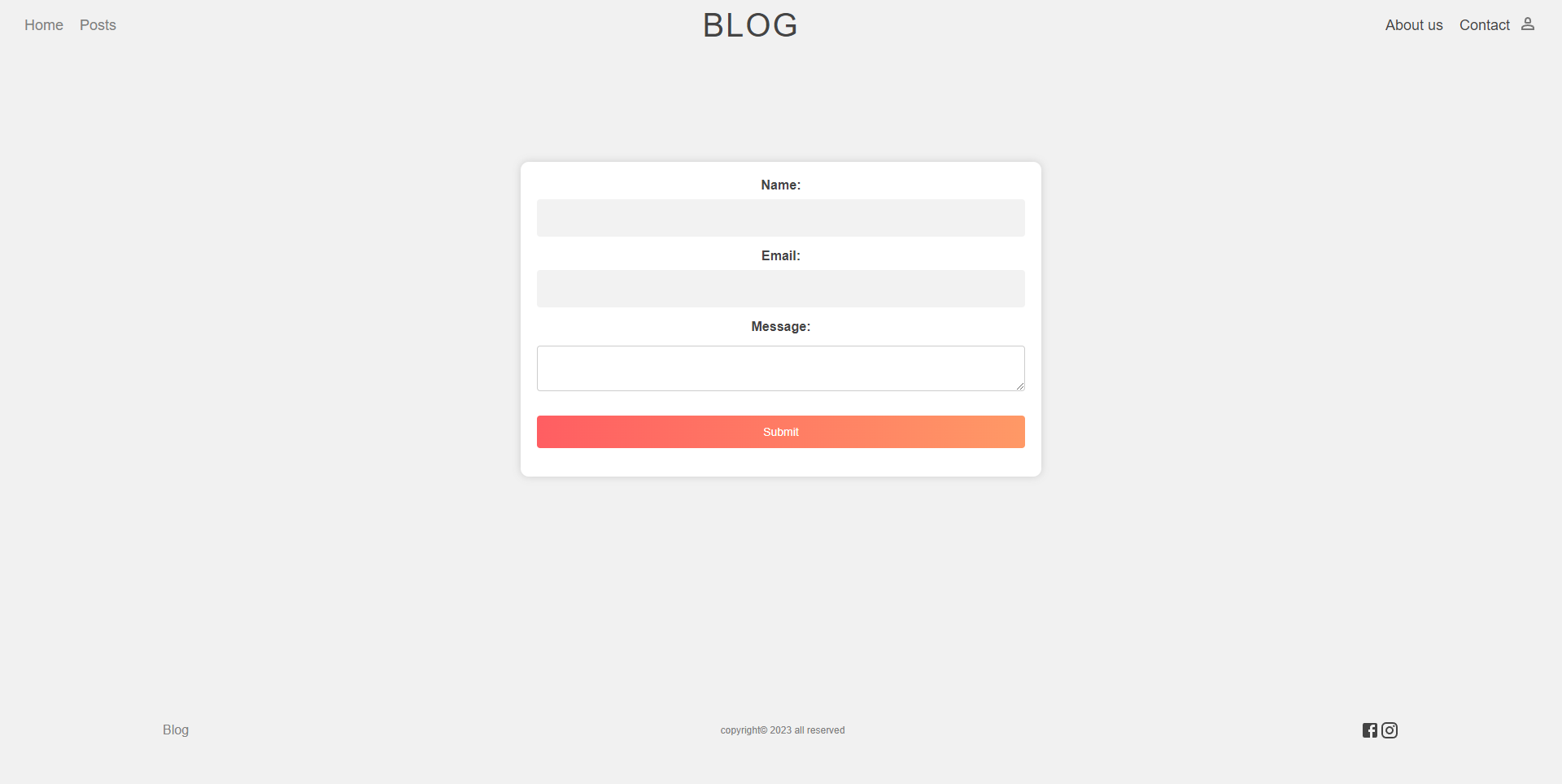


Figure 27: la page de contact

# Conclusion générale

En conclusion, mon stage m'a offert une opportunité unique d'apprendre et de mettre en pratique les compétences nécessaires pour créer des applications web modernes en utilisant le MERN Stack. J'ai acquis des connaissances approfondies sur les différentes technologies du MERN Stack, telles que MongoDB, Express, React et Node.js, et j'ai appris à les utiliser de manière efficace pour construire des applications web complètes et performantes.

Je suis reconnaissant de l'expérience que j'ai acquise lors de ce stage et je suis convaincu que ces compétences me seront utiles tout au long de ma carrière professionnelle. Je suis également heureux d'avoir travaillé avec une équipe de professionnels expérimentés qui m'ont soutenu tout au long de mon stage.

Je suis impatient de continuer à explorer et à développer mes compétences en développement web à l'avenir, en utilisant les connaissances que j'ai acquises lors de ce stage et en explorant de nouvelles technologies et outils.

# Nétographie

# Bibliographie

# ANNEXE : Création de API à travers Nodejs et Expressjs

NodeJs/Express

Node.js est une plateforme de développement JavaScript côté serveur, et Express est un framework minimaliste qui simplifie la création d'applications web et d'API RESTful.

Le choix de Node.js et Express pour développer une API RESTful s'est avéré judicieux pour plusieurs raisons. Tout d'abord, ces technologies offrent une grande simplicité, flexibilité et rapidité. Express permet notamment de gérer efficacement le routage, la gestion des erreurs et des middlewares, ce qui facilite grandement le développement de l'API.

De plus, Node.js est connu pour sa capacité à gérer des connexions simultanées avec des temps de réponse rapides, ce qui en fait un choix efficace pour les applications nécessitant des performances élevées et une scalabilité facile.

En somme, l'utilisation de Node.js et Express pour le développement de notre API RESTful s'est avérée être une solution efficace et performante, en raison de leur simplicité, de leur flexibilité, de leur rapidité et de leur capacité à gérer des connexions simultanées avec des temps de réponse rapides.

Model

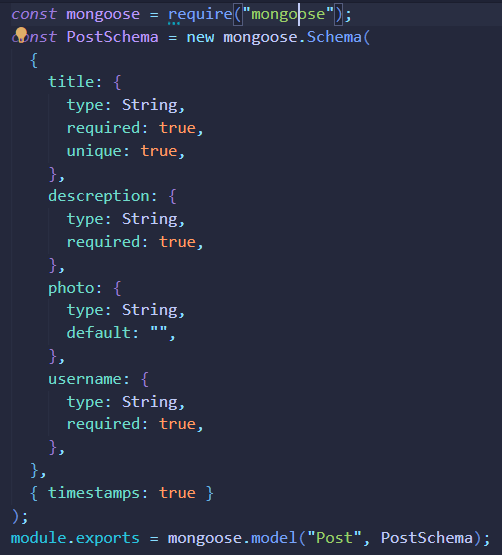
En Express.js, un modèle est un schéma de données qui définit la structure et les propriétés des données à stocker dans une base de données. Les modèles sont souvent utilisés pour effectuer des opérations de CRUD (Create, Read, Update, Delete) sur les données dans une base de données.

Les modèles sont généralement créés à l'aide d'une bibliothèque appelée Mongoose, qui permet de définir des schémas de données pour les bases de données MongoDB. Les schémas définis par Mongoose comprennent des champs, des types de données, des contraintes de validation et d'autres propriétés qui déterminent comment les données sont stockées et manipulées dans la base de données.

En résumé, un modèle dans Express.js est un schéma de données qui définit la structure et les propriétés des données stockées dans une base de données. Il est souvent utilisé pour effectuer des opérations de CRUD sur les données, et est généralement créé à l'aide de la bibliothèque Mongoose.

Le modèle que j'ai créé se compose des schémas suivants :

* Le modèle de post :



On a définit un schéma de modèle de données pour les publications ("Post") d'une application utilisant le framework Express.js avec le module Mongoose qui permet de travailler avec MongoDB.

Le schéma Post a quatre propriétés:

* "title" : une chaîne de caractères qui représente le titre de la publication.
* "description": une chaîne de caractères qui représente la description de la publication.
* "photo": une chaîne de caractères qui représente le chemin d'accès à une image associée à la publication.
* "username": une chaîne de caractères qui représente le nom d'utilisateur de l'auteur de la publication.

Le schéma définit également une option "timestamps" qui ajoute automatiquement deux propriétés supplémentaires à chaque enregistrement de modèle : "createdAt" et "updatedAt", qui enregistrent respectivement la date de création et la date de mise à jour de la publication.

Enfin, le modèle est exporté en tant que module "Post" qui peut être utilisé dans d'autres parties de l'application pour créer, lire, mettre à jour ou supprimer des publications dans la base de données MongoDB.

* Le modèle de post :



On a définit un schéma de modèle de données pour les utilisateurs ("User") d'une application utilisant le framework Express.js avec le module Mongoose qui permet de travailler avec MongoDB.

Le schéma User a trois propriétés:

* "username" : une chaîne de caractères qui représente le nom d'utilisateur de l'utilisateur.
* "email": une chaîne de caractères qui représente l'adresse e-mail de l'utilisateur.
* "password": une chaîne de caractères qui représente le mot de passe de l'utilisateur.

Le schéma définit également une option "timestamps" qui ajoute automatiquement deux propriétés supplémentaires à chaque enregistrement de modèle : "createdAt" et "updatedAt", qui enregistrent respectivement la date de création et la date de mise à jour de l'utilisateur.

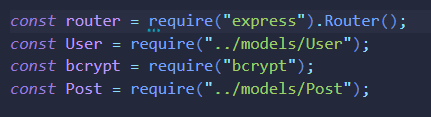
Enfin, le modèle est exporté en tant que module "User" qui peut être utilisé dans d'autres parties de l'application pour créer, lire, mettre à jour ou supprimer des utilisateurs dans la base de données MongoDB.

Route

Dans le framework Express.js, les routes sont utilisées pour définir les points d'entrée pour les requêtes HTTP. Lorsqu'une requête est reçue, Express recherche la route correspondante et exécute la fonction de rappel associée.

Les routes sont définies en utilisant l'objet "Router" d'Express. Les routes sont associées à une méthode HTTP (par exemple, GET, POST, PUT, DELETE) et un chemin (par exemple, "/users", "/posts/:id"). Le chemin peut inclure des paramètres dynamiques, marqués par des deux-points (:), qui peuvent être récupérés dans la fonction de rappel de la route.

Nous avons importé la bibliothèque suivante :



Index.js

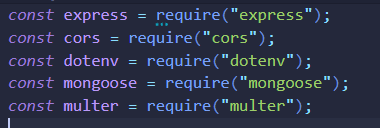
* Definition :

index.js est un fichier important dans les applications Node.js et Express.js, notamment dans les projets de pile MERN. Ce fichier sert de point d'entrée pour l'application et est responsable de la configuration du serveur et de la définition des routes et des middleware.

En général, le fichier index.js est situé dans le répertoire racine du projet et contient des instructions pour créer une instance de l'application Express, configurer le middleware, définir les routes et initialiser la connexion à la base de données MongoDB.

Le contenu de ce fichier peut varier en fonction des besoins et des exigences spécifiques du projet. Par exemple, il peut inclure des déclarations d'importation pour des bibliothèques tierces telles que Express, MongoDB ou React, des instructions pour gérer les erreurs, des définitions de fonctions personnalisées ou des configurations spécifiques pour l'environnement de développement ou de production.

Dans l'ensemble, le fichier index.js est un élément clé de l'infrastructure de l'application et doit être soigneusement conçu et testé pour assurer la stabilité et les performances de l'application.



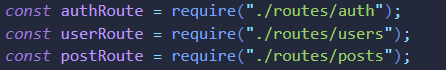
**Express** : Framework web pour Node.js.

**Cors** : Middleware utilisé pour permettre à une application Express de communiquer avec des ressources provenant d'autres domaines.

**dotenv** : Module utilisé pour charger les variables d'environnement à partir d'un fichier .env.

**mongoose** : Bibliothèque utilisée pour interagir avec MongoDB à partir d'une application Node.js.

**multer** : Middleware utilisé pour la gestion des fichiers téléchargés à partir d'une application Express.



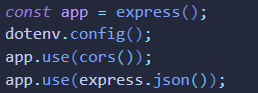
authRoute : Fichier de route pour l'authentification utilisateur.

userRoute : Fichier de route pour les utilisateurs.

postRoute : Fichier de route pour les publications.



Cette ligne de code définit le port sur lequel le serveur doit écouter. Elle utilise une variable d'environnement appelée PORT si elle est définie, sinon elle utilise le port 4000 par défaut. Cela permet de rendre l'application plus flexible, car elle peut être facilement déployée sur différents environnements sans avoir besoin de modifier le code source.

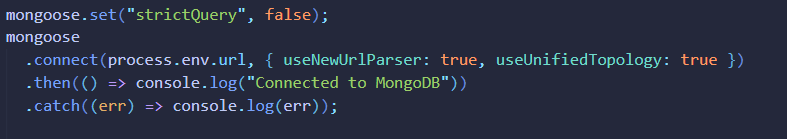


app : Instance de l'application Express.

dotenv.config() : Chargement des variables d'environnement.

app.use(cors()) : Activation de CORS pour permettre les requêtes croisées entre différents domaines.

app.use(express.json()) : Utilisation de l'analyseur de corps JSON pour traiter les demandes HTTP contenant des données JSON.



Ces lignes de code initialisent la connexion de l'application à une base de données MongoDB :

mongoose.set("strictQuery", false); : Configuration de Mongoose pour désactiver le mode strict de requête.

mongoose.connect(process.env.url, { useNewUrlParser: true, useUnifiedTopology: true }) : Connexion à la base de données MongoDB en utilisant une URL définie dans une variable d'environnement appelée "url".

.then(() => console.log("Connected to MongoDB")).catch((err) => console.log(err)); : Gestion de la réussite ou de l'échec de la connexion.

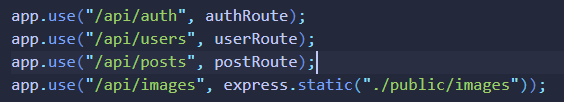


Ces lignes de code implémentent un middleware Multer pour gérer les téléchargements de fichiers dans l'application :

const upload = multer({ ... }) : Initialisation de Multer pour gérer les téléchargements de fichiers. La configuration de Multer est définie à l'intérieur de l'objet passé en paramètre.

storage: multer.diskStorage({ ... }) : Configuration de Multer pour stocker les fichiers sur le disque. Les options destination et filename sont utilisées pour définir le répertoire de destination et le nom de fichier respectivement.

app.post("/api/upload", upload.single("file"), (req, res) => { ... }) : Définition d'une route pour gérer les téléchargements de fichiers. La route attend une requête POST avec un fichier unique dans le champ "file". La fonction middleware upload.single("file") est utilisée pour traiter la requête et stocker le fichier. Une réponse JSON est renvoyée pour indiquer que le fichier a été téléchargé avec succès.



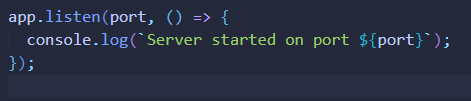
Ces lignes de code définissent les routes de l'application :

app.use("/api/auth", authRoute); : Utilisation de la route authRoute pour gérer les opérations d'authentification de l'API.

app.use("/api/users", userRoute); : Utilisation de la route userRoute pour gérer les opérations d'utilisateur de l'API.

app.use("/api/posts", postRoute); : Utilisation de la route postRoute pour gérer les opérations de publication de l'API.

app.use("/api/images", express.static("./public/images")); : Définition d'une route statique pour servir les fichiers d'images téléchargés à partir du répertoire "./public/images". Lorsqu'un client effectue une requête pour cette route, le serveur renvoie simplement le fichier image correspondant.



Cette ligne de code lance l'application en écoutant les demandes du client sur le port spécifié :

app.listen(port, () => { console.log(Server started on port ${port}); }); : Cette instruction appelle la méthode listen d'Express pour démarrer l'application sur le port spécifié. Elle affiche également un message dans la console pour indiquer que le serveur a démarré avec succès.