**Présentation du rapport**

# Remerciement

Ce stage perfectionnement était une grande expérience. Ce projet est un apport très bénéfique quant au perfectionnement des connaissances de l’étudiant dans le domaine informatique et pour avoir l’opportunité d’appliquer ses connaissances dans le cadre professionnel.

Mes remerciements vont tout d’abord au directeur de la société Dope Vision Mr. Mohamed Taher Lazrak de m’accueillir à cette société, de sa confiance pour m'avoir accordé un tel projet et de son assistance et son encadrement tout au long de la durée de ce stage.

Je remercie également tous les membres de l’équipe de Dope Vision pour leur accueil chaleureux, pour les bons conseils et pour les bons moments passés ensemble et la bonne ambiance qu’ils entretiennent au sein de cette société.

J’espère que ce travail soit de qualité et plait au jury

Sommaire

[Remerciement 2](#_Toc96420241)

[Liste des figures 4](#_Toc96420242)

[Liste des tableaux 5](#_Toc96420243)

[I. Introduction générale 6](#_Toc96420244)

[Contexte et problématique du stage : 6](#_Toc96420245)

[Plan du rapport : 6](#_Toc96420246)

[II. Contexte du travail et présentation de l’entreprise 7](#_Toc96420247)

[Présentation de l’organisme d’accueil : 7](#_Toc96420248)

[Cadre et objectif de stage du projet : 7](#_Toc96420249)

[Etude de l’existant : 7](#_Toc96420250)

[Solutions Proposées : 8](#_Toc96420251)

[Conclusion : 8](#_Toc96420252)

[III. Développement de L’application 9](#_Toc96420253)

[1- Etude Conceptuelle 9](#_Toc96420254)

[Introduction : 9](#_Toc96420255)

[Analyse : 9](#_Toc96420256)

[a. Les fonctionnalités du système 9](#_Toc96420257)

[b. Identification des acteurs 9](#_Toc96420258)

[Conception : 10](#_Toc96420259)

[a. Conception des interfaces : 10](#_Toc96420260)

[b. Modélisation : 11](#_Toc96420261)

[c. Base de données 14](#_Toc96420262)

[Conclusion : 14](#_Toc96420263)

[2- Réalisation 15](#_Toc96420264)

[Introduction : 15](#_Toc96420265)

[Les choix Techniques : 15](#_Toc96420266)

[Mise en œuvre de l’application : 17](#_Toc96420267)

[a. Environnement de travail 17](#_Toc96420268)

[b. Etapes de réalisation 19](#_Toc96420269)

[Conclusion : 24](#_Toc96420270)

[3 Conclusion générale 25](#_Toc96420271)

[Bibliography 26](#_Toc96420272)

# Liste des figures

[Figure 1: Capture du site actuel 9](#_Toc96420276)

[Figure 2: Maquette Interface web 11](#_Toc96420277)

[Figure 3: Maquette Interface mobile 11](#_Toc96420278)

[Figure 4: Diagramme de cas d’utilisation 12](#_Toc96420279)

[Figure 5: Diagramme de Classe 13](#_Toc96420280)

[Figure 6: Diagramme seq lire article 14](#_Toc96420281)

[Figure 7: Diagramme seq authentification 14](#_Toc96420282)

[Figure 8: Architecture MERN Stack 16](#_Toc96420283)

[Figure 9: Mongoose object mapping 17](#_Toc96420284)

[Figure 10: méthodes de protocole http 18](#_Toc96420285)

[Figure 11:Création package.json 19](#_Toc96420286)

[Figure 12: créer "server.js" 20](#_Toc96420287)

[Figure 13: ajout connectDB à server.js 21](#_Toc96420288)

[Figure 14: connecte data base 21](#_Toc96420289)

[Figure 15: modèles des articles 22](#_Toc96420290)

[Figure 16: tester le routage 22](#_Toc96420291)

[Figure 17: serveur route 22](#_Toc96420292)

[Figure 18: resultat routage 23](#_Toc96420293)

[Figure 19: methode HTTPS 23](#_Toc96420294)

[Figure 20: POST methode 23](#_Toc96420295)

[Figure 21: Postman test API 24](#_Toc96420296)

[Figure 22: resultat test terminal 25](#_Toc96420297)

[Figure 23: resultat test dans DataBase 25](#_Toc96420298)

# Liste des tableaux

[Table 1: logiciels conception 11](#_Toc96420299)

[Table 2: Ressources matériels 18](#_Toc96420300)

[Table 3: Ressources Logiciels 18](#_Toc96420301)

[Table 4: dépendances de production 19](#_Toc96420302)

[Table 5: HTTP methode errors 24](#_Toc96420303)

# Introduction générale

## Contexte et problématique du stage :

Cadre du stage 2eme DSI…

Questions

## Plan du rapport :

Mon travail sera présenté sous la forme de trois chapitres. En premier lieu, on s’intéresse à la présentation générale du cadre de stage et à l’étude de l’existant et la solution proposé. Au troisième chapitre, on passe à l’étude conceptuelle du site web. Ensuite à la réalisation du site. Et, on finira par une conclusion générale pour conclure ce rapport

# Contexte du travail et présentation de l’entreprise

## Présentation de l’organisme d’accueil :

Dope vision a été créé sous la forme d’une boite de communication et de marketing digitale en 2020. Dope Vison travaille à l’amélioration de la présence digitale des sociétés. Elle fait l’accompagnement des stratégies de communication a l’objective d’avoir une meilleure image de marque.

Les activités principales de cette société est la production photographique et audiovisuelle, la création des logos, des chartes graphiques et des designs graphiques, la présentation des sociétés sur les réseaux sociaux et la création des site web pour les sociétés.

La société aspire créer une magasine pour ses clients afin de les guider et les cultiver selon leurs intérêts et leurs secteurs d’activité. Dope Vision Magasine couvre les nouveautés et les actualités nationales et internationales qui concerne leurs clients.

La majorité de presse papier ont démarré leur transition vers le numérique. Dope Vision Magasine vise une présence digitale ainsi qu’une présence sur papier.

## Cadre et objectif de stage du projet :

Le sujet de ce stage a été proposé dans le cadre digitale de cette magasine. La mission principale est d’apporter des améliorations sur le site web actuel de la magasine toute en gardant les fonctionnalités principales du site : publier, modifier et supprimer des articles du site.

## Etude de l’existant :

Dans cette partie, en premier temps, Avant de commencer l’étape d’étude et de réalisation du projet il faut analyser l’existant pour mieux comprendre les problèmes afin de les corriger.

Actuellement, le site de Dope Vision Magasine se base sur le CMS WordPress comme site provisoire. La société n’est certainement pas satisfaite avec ce site WordPress comme il est trop lent. Après installer beaucoup de plugins, chacun intègre son propre style et ses propres scripts. Ces derniers se retrouveront alors dans le code source. Après publier quelque article le site est devenue de plus en plus lent et difficile à maintenir ou a faire des modification sur le site.

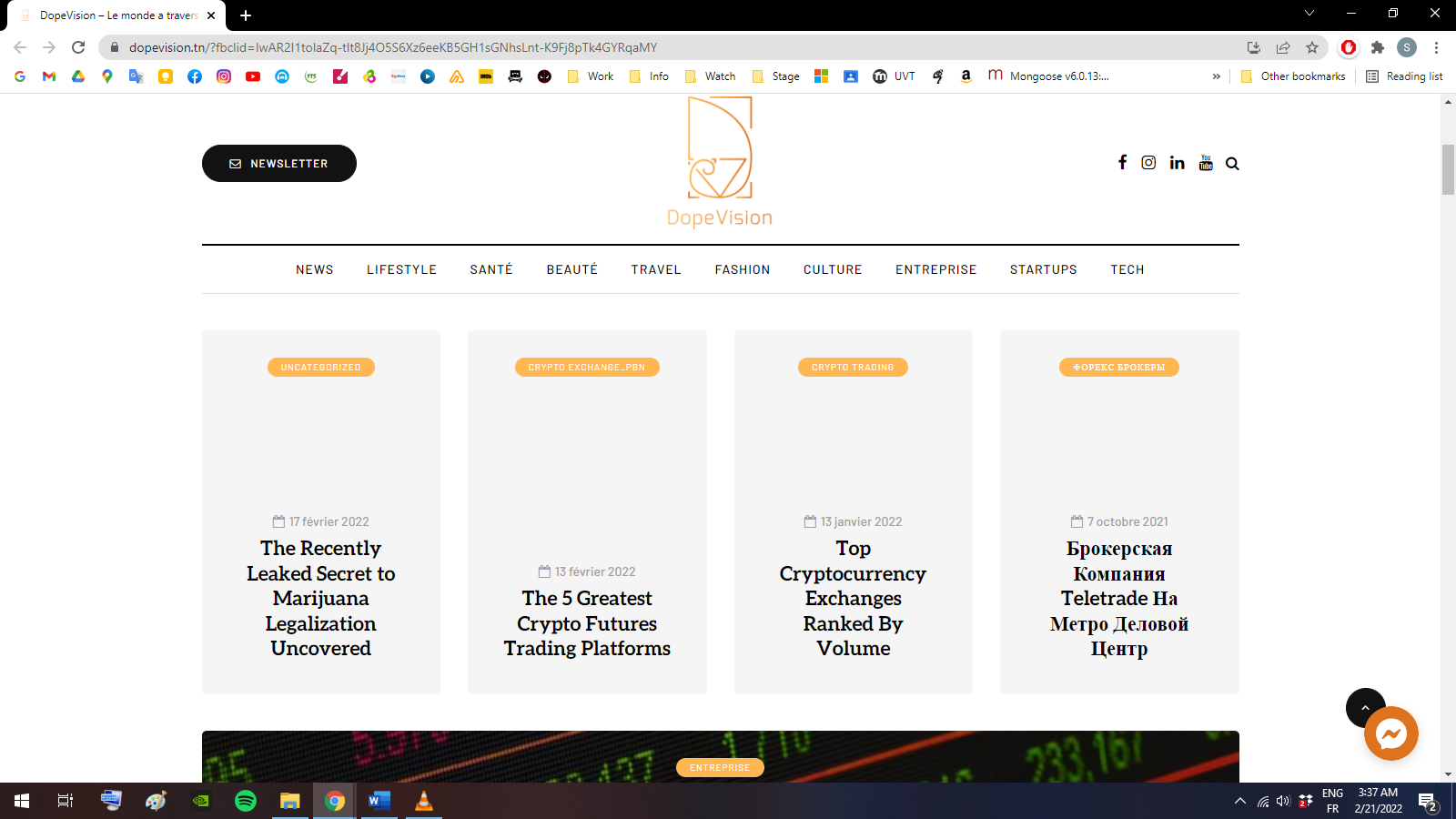


Figure 1: Capture du site actuel

## Solutions Proposées :

Comme Solution, j’ai proposé de redévelopper ce site autrement. D’où on peut avoir une partie Front-End dynamique et interactive pour une meilleure expérience pour le client. A la foi une partie Back-End avec un temps d’exécution rapide ainsi qu’une base donnée flexible autonome d’une structure riche et adaptive au diffèrent article et puis une base donnée orientée documents.

Pour la réalisation de cette solution j’ai choisi l’architecture MERN Stack. Qui est la plus compatible avec la solution proposée à la repense au besoin de la société.

MERN Stack est l’ensemble de ses quatre technologies :

* MongoDB
* Express JS
* React JS
* Node JS

## Conclusion :

L’étude de l’existant a permis de poser la problématique du sujet ensuite on est arrivé a proposé une solution, à choisir des outils de développement et d’adopter la méthodologie du travail et de sélectionner les données essentielles pour la modélisation et la conception du site web.

Dans le chapitre suivant je vais définir ses outils de développement et décrire les différentes étapes de la conception et la réalisation de ce projet selon les choix technique de la solution proposé.

# Développement de L’application

## Etude Conceptuelle

## Introduction :

Pour une meilleure maitrise d’un projet informatique, il est important de suivre une démarche. La démarche adoptée dans ce projet consiste en un processus unifié de développement construit autour d'UML (Unified Modeling Language).

Nous entamerons le processus de développement de notre application par l’étape de l’analyse qui permet de mettre en évidence les différentes interactions des acteurs avec le système projeté. Et, nous poursuivrons avec l’étape de conception de L’interface puis la modélisation.

## Analyse :

Cette étape recense les besoins des utilisateurs, met en évidence les différents acteurs intervenants dans le système projeté.

### Les fonctionnalités du système

Nous commençons le processus d’analyse par la mise en évidence d’une liste des différentes fonctionnalités du système d’information en prenant en compte les différents besoins des utilisateurs :

* Lister les articles ;
* Ajouter les articles ;
* Modifier les articles ;
* Supprimer les articles ;
* Consulter magazine ;

### Identification des acteurs

Un acteur est une [entité](https://fr.wikipedia.org/wiki/Entit%C3%A9#En_informatique) qui définit le rôle joué par un utilisateur ou par un système qui interagit avec le système modélisé.

* Utilisateur ;
* Auteur/journaliste ;

## Conception :

La conception d'application web se distingue de la définition des interfaces utilisateurs sous forme de pages web et de la modélisation.

Comme ressources logiciels de la partie conception :

Table 1: logiciels conception

|  |  |
| --- | --- |
| Adobe XD | vector-based experience design platform |
| Star UML | modeling tool that supports the UML |

### Conception des interfaces :



Figure 2: Maquette Interface web

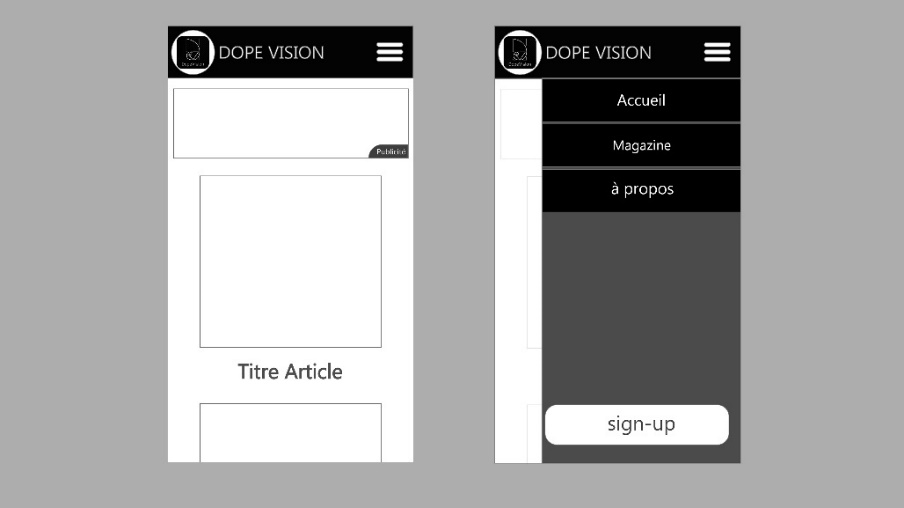


Figure 3: Maquette Interface mobile

### Modélisation :

Cette étape présente trois diagrammes. Le diagramme de cas d’utilisation, le diagramme de séquences et le diagramme de classe.

* **Diagramme de cas d’utilisation**

Les diagrammes de cas d'utilisation (DCU) sont des [diagrammes UML](https://fr.wikipedia.org/wiki/Unified_Modeling_Language#Les_diagrammes) utilisés pour une représentation du comportement fonctionnel d'un système [logiciel](https://fr.wikipedia.org/wiki/Logiciel).

(fr.wikipedia.org, Les diagrammes de cas d'utilisation, n.d.)

Le diagramme de cas d’utilisation représente les relations fonctionnelles entre les acteurs et le système étudié. Il donne une description cohérente de toutes les vues que l’on peut avoir du système.

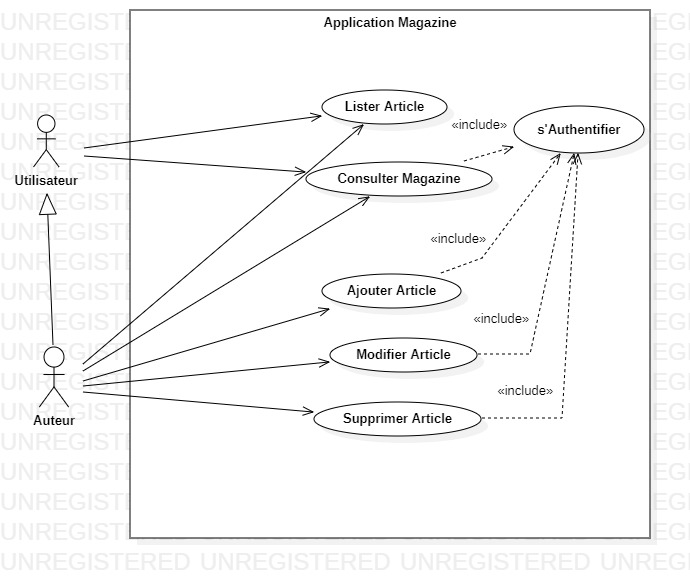
****

Figure 4: Diagramme de cas d’utilisation

* **Le diagramme de classe**

Le diagramme de classes est un schéma utilisé en [génie logiciel](https://fr.wikipedia.org/wiki/G%C3%A9nie_logiciel) pour présenter les [classes](https://fr.wikipedia.org/wiki/Classe_(informatique)) et les [interfaces](https://fr.wikipedia.org/wiki/Interface_(informatique)) des systèmes ainsi que leurs relations.

(fr.wikipedia.org, Diagramme de classes, n.d.)

Un diagramme de classe est la description de tout ou d'une partie du système d'une manière abstraite, en termes de classes, de structure et d'associations.

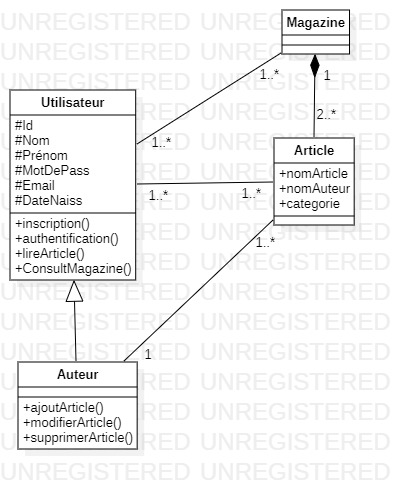


Figure 5: Diagramme de Classe

* **Le diagramme de séquence**

Les diagrammes de séquences sont la représentation graphique des [interactions](https://fr.wikipedia.org/wiki/Unified_Modeling_Language) entre les [acteurs](https://fr.wikipedia.org/wiki/Acteur_(UML)) et le système selon un ordre chronologique dans la formulation [Unified Modeling Language](https://fr.wikipedia.org/wiki/Unified_Modeling_Language" \o "Unified Modeling Language).

(fr.wikipedia.org, Diagramme de séquance, n.d.)

Le diagramme de séquence représente l’enchainement des échanges de messages entre objets. Il permet de représenter un processus de façon simplifiée, en se centrant sur les échanges entre acteurs ou avec le système d’information.

Lire Article :

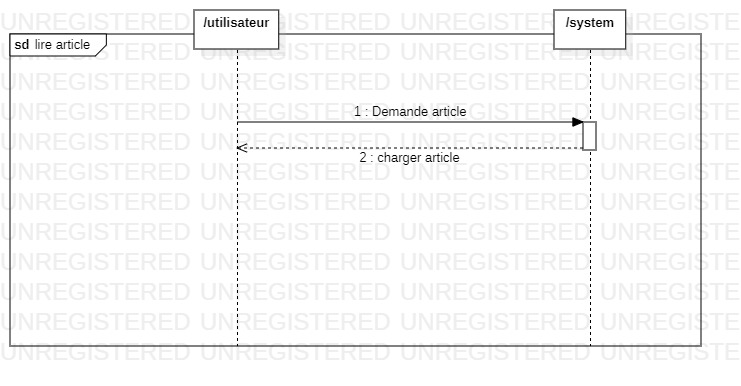


Figure 6: Diagramme seq lire article

Authentification :

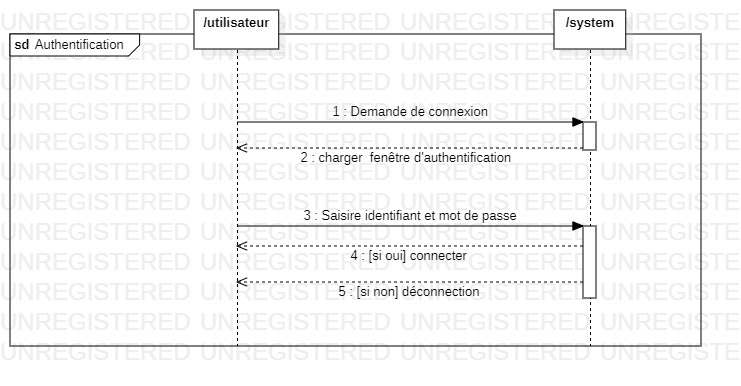


Figure 7: Diagramme seq authentification

### Base de données

Pour la conception de base de données, les bases de données NoSQL (not only SQL) sont des bases de données non tabulaires et stockent les données différemment des tables relationnelles. Les bases de données NoSQL sont disponibles dans une variété de types en fonction de leur modèle de données.

(mongodb.com, n.d.)

Les bases de données documentaires stockent les données dans des documents similaires aux objets JSON (JavaScript Object Notation). Chaque document contient des paires de champs et de valeurs. Les valeurs peuvent généralement être de divers types, y compris des éléments tels que des chaînes, des nombres, des booléens, des tableaux ou des objets.

(mongodb.com, Types of NoSQL Databases, n.d.)

## Conclusion :

Ce chapitre est consacré à l’étude de l’application en utilisant le langage UML. Il met en évidence et donne l'exemple de quelques composants de l’application. Le diagramme de cas d’utilisation, les diagrammes de séquences et le diagramme de classe

## Réalisation

## Introduction :

Cette partie présente les étapes de développement de ce projet.

## Les choix Techniques :

**Le MERN Stack** est un ensemble de quatre technologies : MongoDB, framework Express JS, React JS et Node JS. Cette combinaison permet aux développeurs de créer des sites Web complets (back-end et front-end). Avec le MERN Stack, nous utilisons JavaScript côté client et Node.js côté serveur.

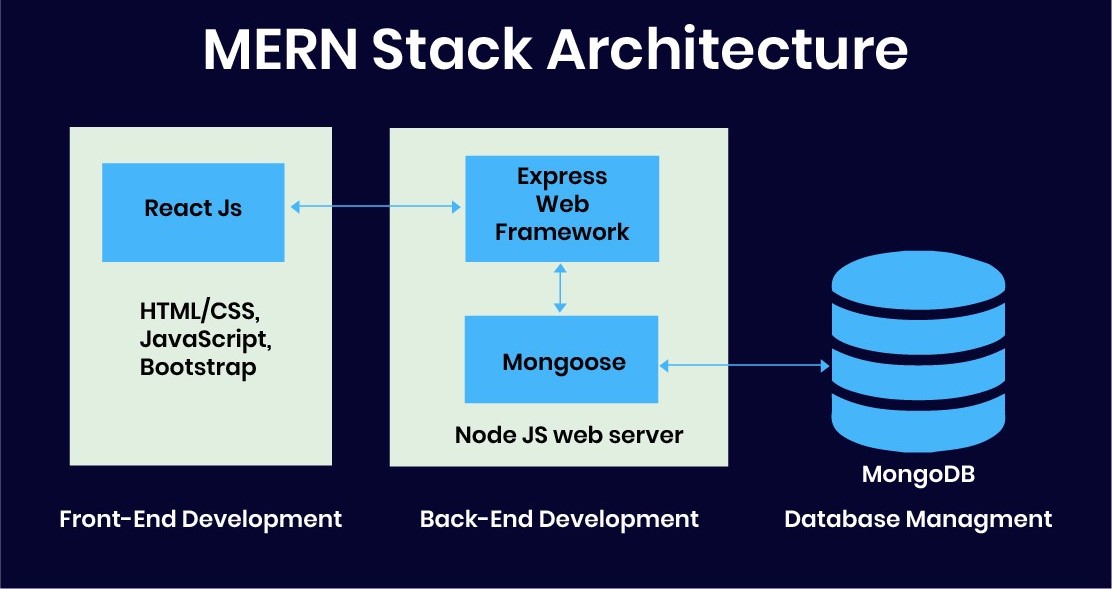
****

Figure 8: Architecture MERN Stack

**MongoDB** vous permet de gérer des bases de données. Il diffère de SQL. MongoDB est utilisé au MERN car les données sont manipulées au format JSON. Il est également très simple de transformer des données JavaScript en MongoDB et vice versa, grâce à des bibliothèques telles que **mongoose**

**Express** est un langage de modélisation de données qui spécifie formellement les données. Il offre la possibilité de développer une application Web complexe. Express utilise le système middleware, dédié à la gestion d'applications Web très complexes.

Express est utilisé pour la création de **l'API REST**.

**React.js** développe la partie front-end du site Web et connu pour accélérer les vitesses de chargement des sites Web en général. Les animations, les téléchargements et toute autre transition sont effectués beaucoup plus rapidement.

**Node.js** fonctionne avec JavaScript. Ces deux langues sont fortement liées et sont capables d'interagir l'une avec l'autre très facilement. Node.JS a la particularité d'être un langage extrêmement rapide, l'un des plus rapides en programmation informatique, car c'est aussi un langage asynchrone

(bocasay.com, n.d.)

**Mongoose** est une bibliothèque ODM (Object Data Modeling) pour MongoDB distribuée sous forme de package npm. Mongoose équivaut à utiliser le pilote natif MongoDB Node.js avec MongoDB Schema Validation.

(mongodb.com, mongoose, n.d.)

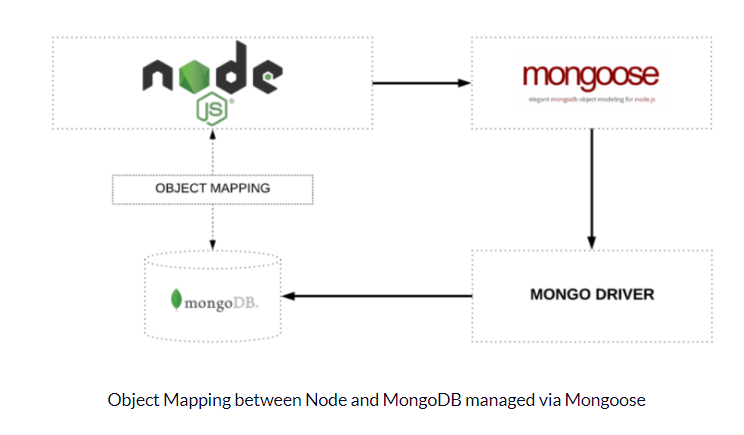
****

Figure 9: Mongoose object mapping

**L'API (Application Program Interface)** est un moyen convenu d'envoyer et de recevoir des données entre ordinateurs. Par exemple, si vous souhaitez afficher Google Maps sur votre site, mais que les cartes sont sur les serveurs de Google, vous avez besoin d'un moyen de demander à Google de vous fournir les cartes. La façon de demander à Google de vous envoyer les cartes demandées est via une API fournie par Google qui vous indique à quelles adresses Web devez-vous envoyer les demandes pour obtenir les données. Dans un langage plus formel, vous devez envoyer une requête au serveur distant pour obtenir une réponse.

**REST (Representational State Transfer)** est une API qui définit un ensemble de fonctions que les programmeurs peuvent utiliser pour envoyer des requêtes et recevoir des réponses à l'aide **des méthodes de protocole HTTP** telles que GET et POST.

L'API REST peut être utilisée par n'importe quel site ou application, quel que soit le langage dans lequel il est écrit, car les requêtes sont basées sur le protocole HTTP universel et les informations sont généralement renvoyées au format JSON que presque tous les langages de programmation peuvent lire.



Figure 10: méthodes de protocole http

## Mise en œuvre de l’application :

Pour la mise en œuvre de l’application

### Environnement de travail

Pour la réalisation de cette application, J'ai utilisé comme recours plusieurs moyens matériels et logiciels :

Table 2: Ressources matériels

|  |  |  |
| --- | --- | --- |
| Processeur | RAM | Disque Dur |
| Intel i7 - 3770 | 16.0 Gb | Ssd 240gb |

Puis installer les programmes suivants comme ressources logiciels :

Table 3: Ressources Logiciels

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Git Bash** | **Visual Studio Code** | **MongoDB** | **Node.js** | **Yarn** | **React** | **Postman** |
|  |  |  |  |  |  |  |

**Postman** est une application utilisée pour les tests d'API. Il s'agit d'un client HTTP qui teste les requêtes HTTP, en utilisant une interface utilisateur graphique, à travers laquelle nous obtenons différents types de réponses qui doivent ensuite être validées.

(postman, n.d.)

Ensuite en passe à la création du package JSON en utilisant Git Bash avec la commande « npm init »

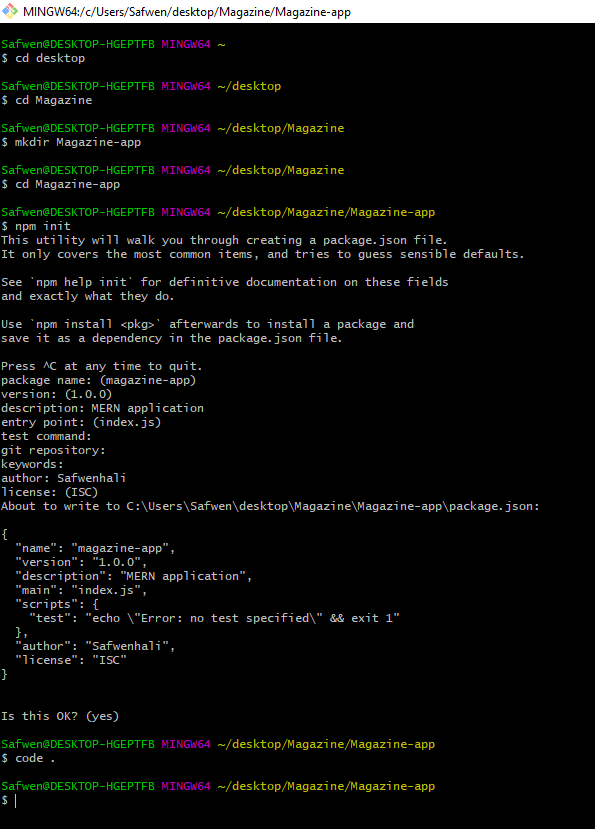


Figure 11:Création package.json

En passe à l’installation des dépendances de production

Table 4: dépendances de production

|  |  |  |
| --- | --- | --- |
| **Express** | **Mongoose** | **Nodemon** |
|  |  |  |

Lors du développement d’une application Node et de faire des modifications, pour les voir effectives, vous devez redémarrer le serveur. Lorsque vous lancez votre application node.js avec **Nodemon**, elle surveille tout changement et redémarre automatiquement le serveur, améliorant ainsi votre productivité.

L’installation se fait dans le terminal de VS Code avec les commandes suivantes :

* « npm i express »
* « npm i mongoose »
* «npm install -g nodemon »

### Etapes de réalisation

1. Démarrer le serveur de node en créant le fichier « server.js », require Express et spécifier le port. Puis démarrer le serveur avec la commande « npm run dev »

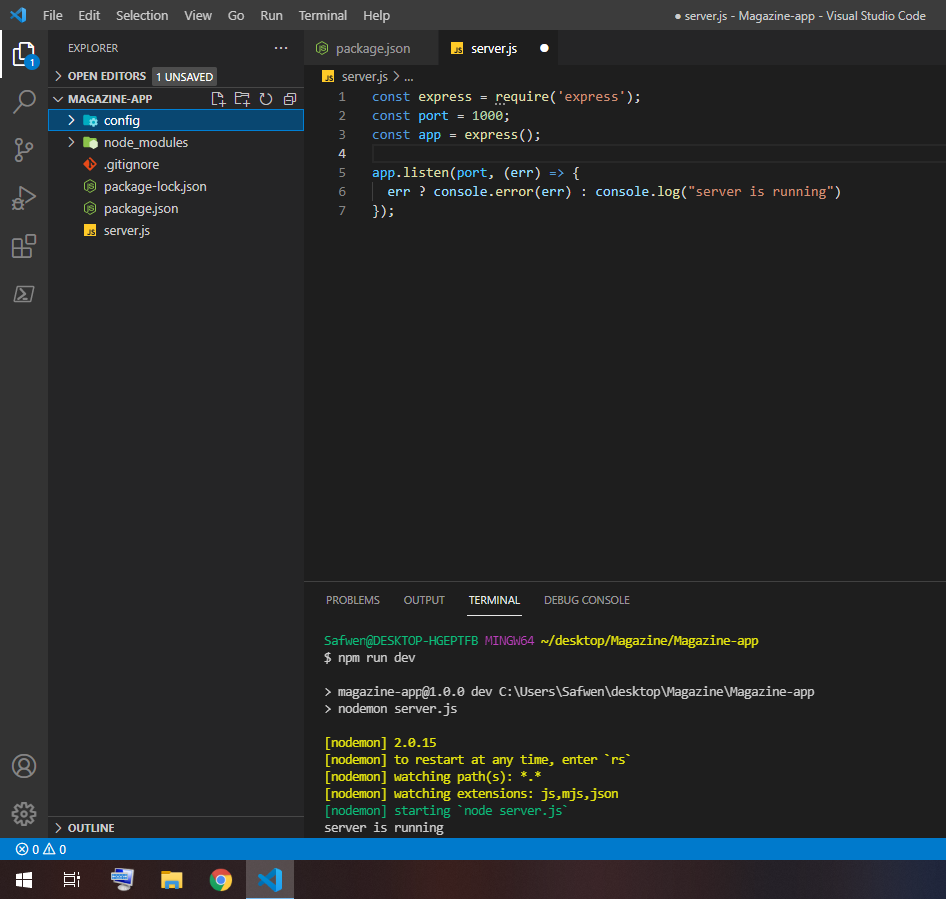


Figure 12: créer "server.js"

1. On se connecte à la base donné MongoDB, dans le dossier config on crée le fichier « connectDB.js » Dans le fichier « connectDB.js » premièrement require mongoose puis se connecter sur le serveur local. Et ajouter la méthode connectDB to serveur.js

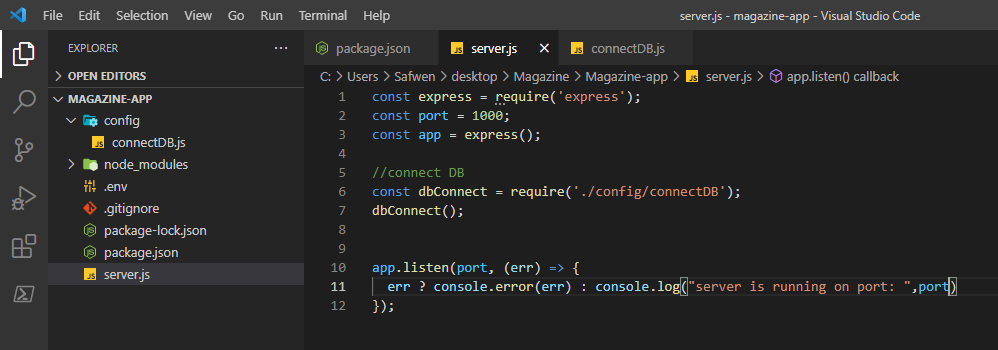


Figure 13: ajout connectDB à server.js

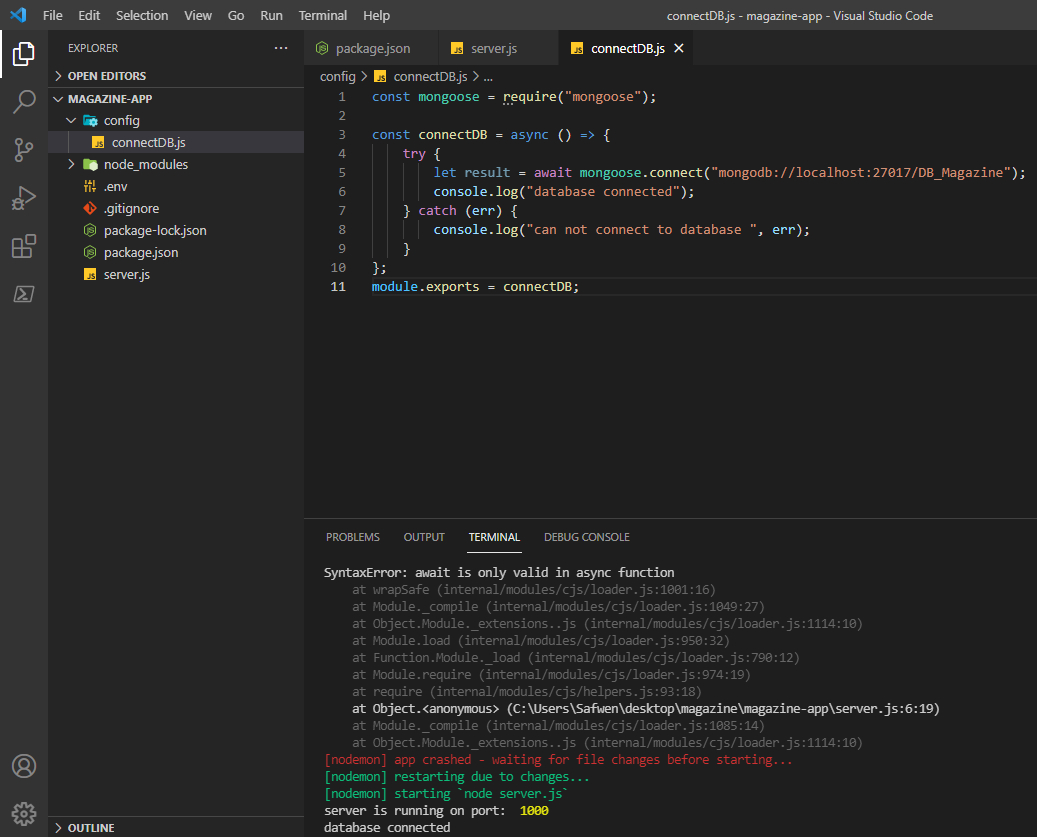


Figure 14: connecte data base

1. Créer les modèles des articles avec « mongoose.schema »

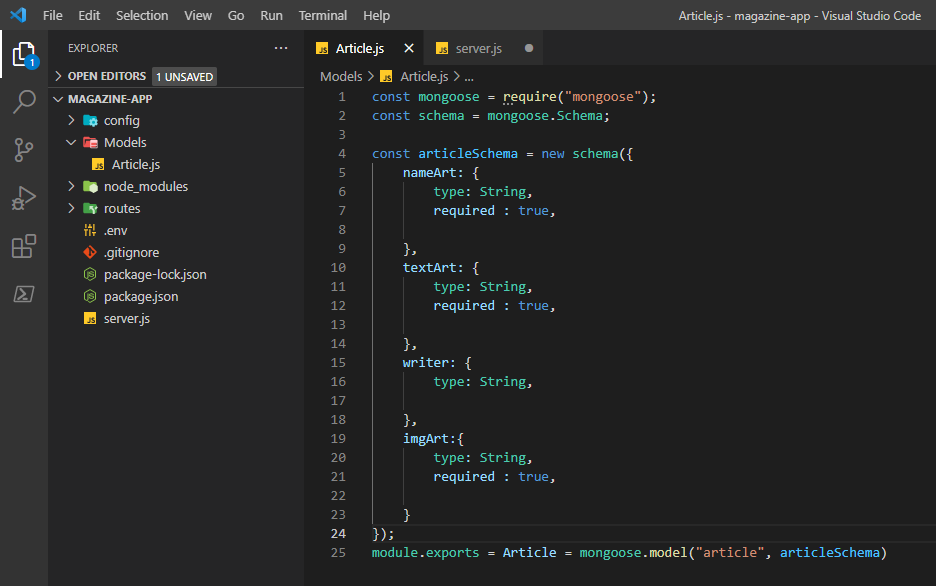


Figure 15: modèles des articles

1. Tester le routage des articles au serveurs selon les modèles avec **Postman** et ajouter le routeur a « serveur.js »

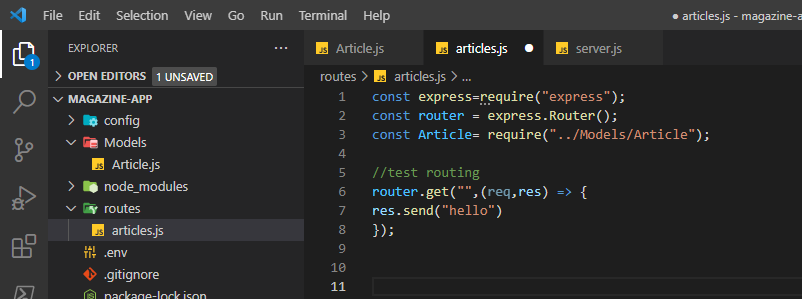


Figure 16: tester le routage

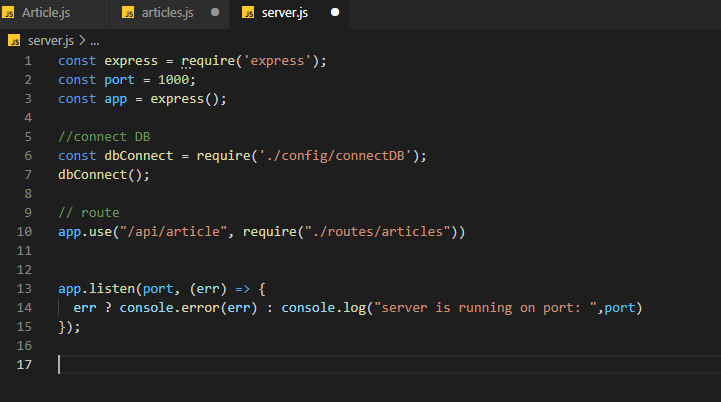


Figure 17: serveur route

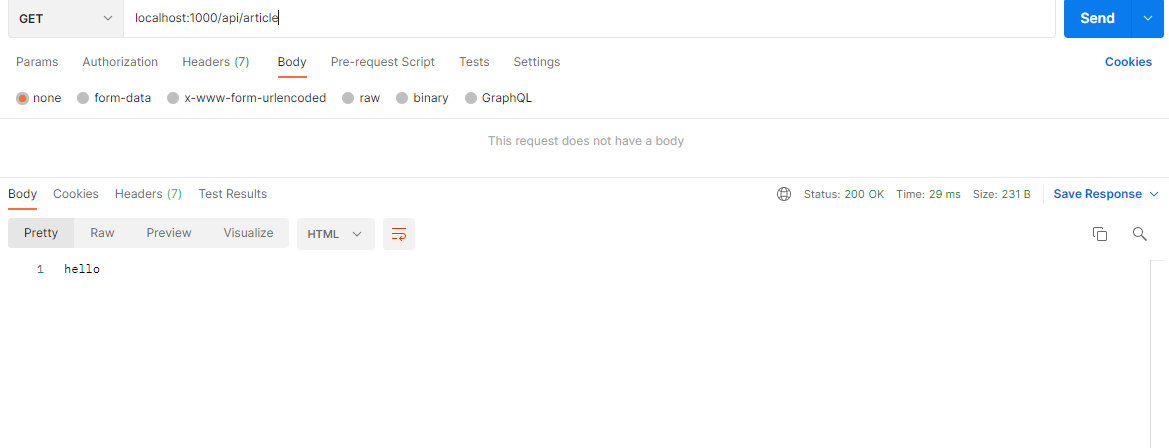


Figure 18: resultat routage

1. Préparer au méthode HTTPS

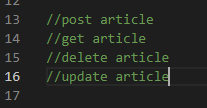


Figure 19: methode HTTPS

1. Créer et tester la méthode *POST* de REST API :

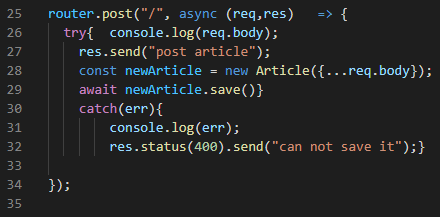
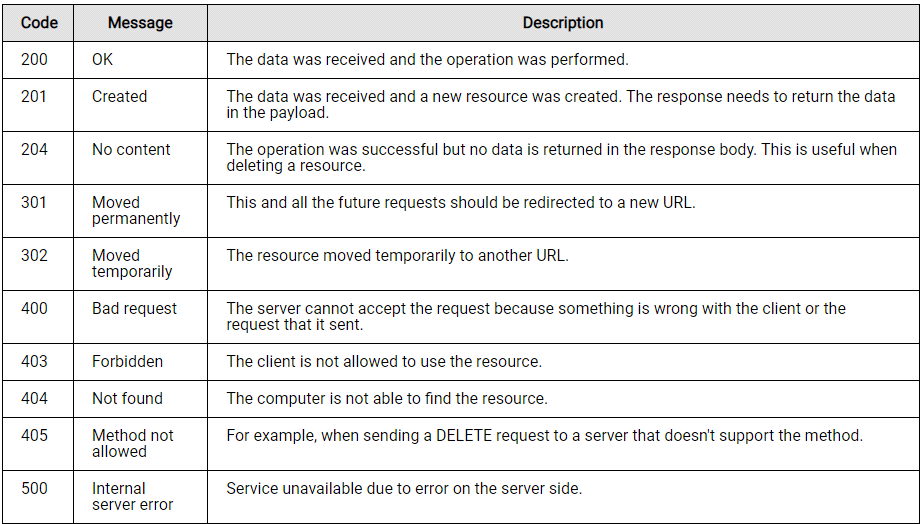


Figure 20: POST methode

Catch errors:

Table 5: HTTP methode errors



Test POST méthodes :

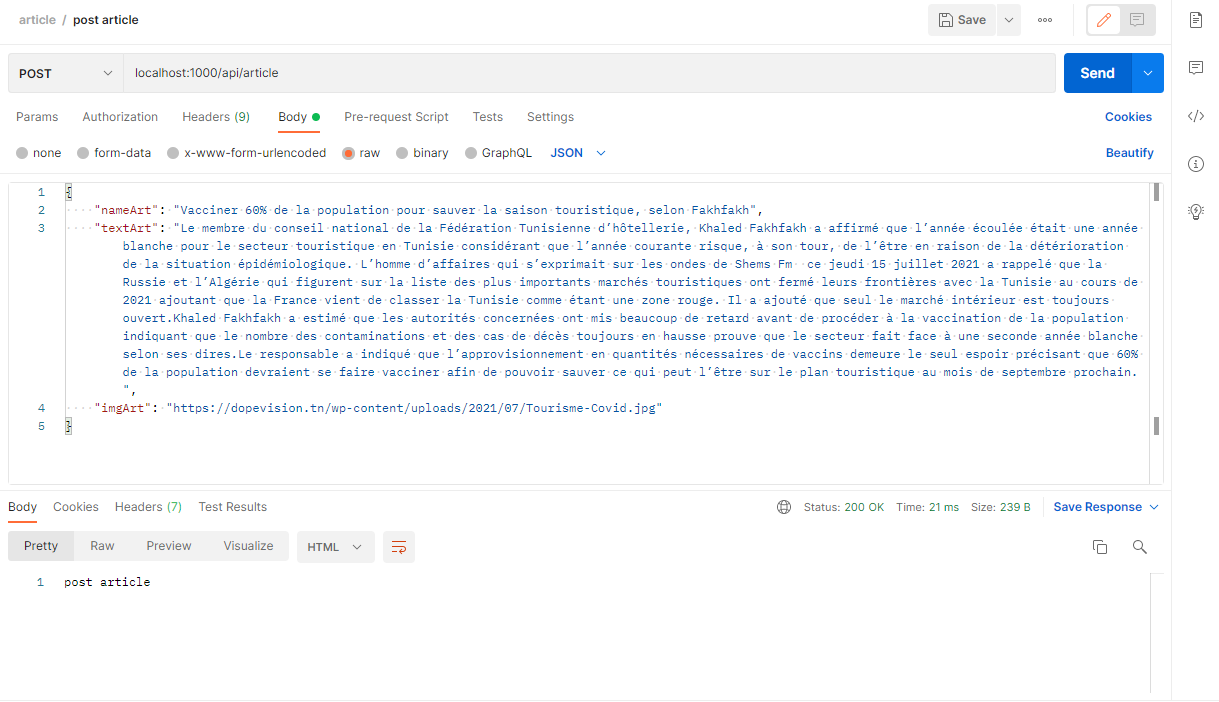


Figure 21: Postman test API

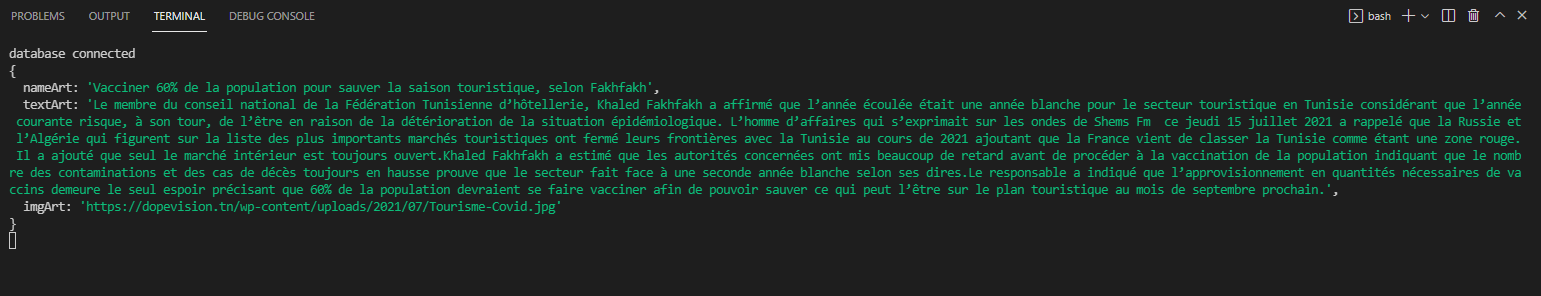


Figure 22: resultat test terminal

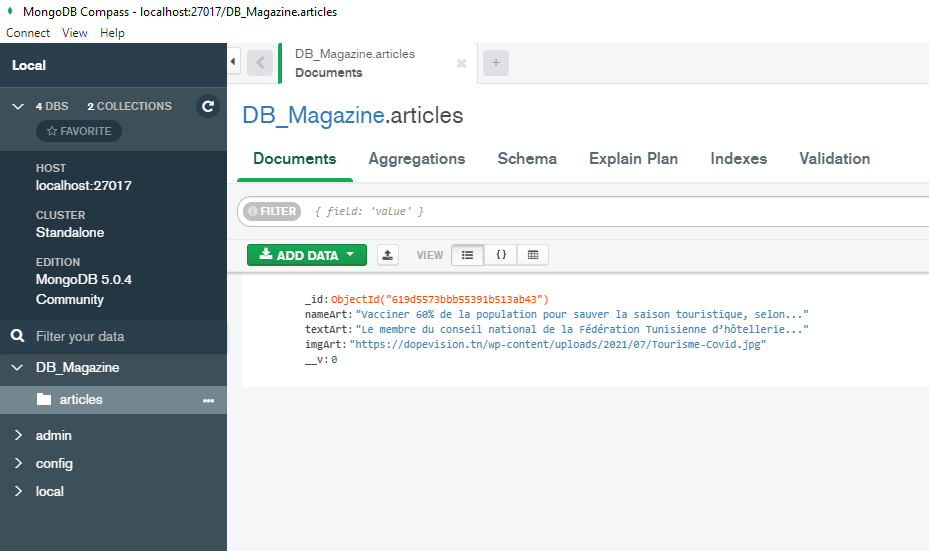


Figure 23: resultat test dans DataBase

## Conclusion :

Cette partie a été consacrée à la présentation de la méthode de développement qui a été menée dans ce projet, et les outils qui nous ont servi d’appui pour arriver aux solutions à la problématique qui a été posée au début du projet, afin de satisfaire les besoins des utilisateurs.

# Conclusion générale

La conclusion comporte une à deux pages. Elle donne une synthèse du travail effectué et présente des perspectives. Elle comporte généralement : - Un résumé des principales contributions apportées dans ce travail de stage.

- Des perspectives d'amélioration de l'application/ du site développé(e) dans le cadre du stage effectué,

- Des perspectives concernant la formation et le projet professionnel de l'étudiant.

Ce projet a été mené dans le cadre d'une demande spécifique du service « techniques culturales » d’eRcane en matière de gestion des données expérimentales, concernant le développement d'un système d’information. La première étape a été l’implémentation d’une base de données, et la deuxième étape a été consacrée au développement d’une application web. La base de données STC conçue lors de ce projet, est d’une valeur très importante pour le service « techniques culturales ». D’une part, elle permet de stocker, de centraliser et de pérenniser toutes leurs données reparties sur plusieurs années de recherche. D’autre part, elle permet d’exploiter et de rechercher toutes les informations dont les agronomes ont besoin pour réaliser des analyses, comparer et valider les résultats de leurs travaux de recherches. L’application développée, offre la possibilité aux utilisateurs de gérer leurs données à travers des interfaces web, qui sont accessibles depuis n’importe quel ordinateur et n'importe quel lieu. L'alimentation de la base de données est réalisée à travers des formulaires de saisie, ou par l'intermédiaire d'importations automatisées de fichiers Excel prédéfinis, avec un format étudié spécifiquement pour l'importation. Pour une meilleure gestion des données, ces interfaces web offrent d'autres fonctionnalités aux utilisateurs, comme par exemple, l’affichage des tableaux de données ordonnés, pour pouvoir consulter toutes les données stockées mais aussi, l'exportation des données à des fins d'analyse et de modélisation. Plusieurs perspectives sont envisagées pour ce projet tels que : • la création d’interfaces de géolocalisation des parcelles ; • l’ajout d'équations de calculs, pour gérer les traitements phytosanitaires. Durant cette formation j’ai été confrontée à quelques difficultés, dues à l’évolution des besoins des utilisateurs qui m’a obligée à réadapter la conduite du projet. Mais loin de me décourager, j’ai occupé tout mon temps à réunir les connaissances nécessaires qui m’ont permis d’être à l’interface entre le domaine informatique et agronome. De ce fait, j’ai pu également développer mes connaissances et acquérir de nouvelles compétences en base de données, en développement web et dans l’utilisation des Frameworks. Grâce à ce stage, je pense développer une passion dans le domaine des cultures agronomes qui est en sans cesse évolution, mais aussi enrichir mon expérience humaine à travers le travail en entreprise.

# Bibliography

fr.wikipedia.org. (n.d.). Retrieved from Les diagrammes de cas d'utilisation: https://fr.wikipedia.org/wiki/Diagramme\_de\_cas\_d%27utilisation#:~:text=Les%20diagrammes%20de%20cas%20d,d'utilisation%20sont%20plus%20appropri%C3%A9s.

fr.wikipedia.org. (n.d.). Retrieved from Diagramme de séquance: https://fr.wikipedia.org/wiki/Diagramme\_de\_s%C3%A9quence

fr.wikipedia.org. (n.d.). Retrieved from Diagramme de classes: https://fr.wikipedia.org/wiki/Diagramme\_de\_classes

mongodb.com. (n.d.). Retrieved from NoSQL Explained: https://www.mongodb.com/nosql-explained