



Université Mohammed 1<sup>er</sup>  
École Supérieure de Technologie de Nador  
ESTN

## **MINI-PROJET**

# Application WEB de Gestion d'une Clinique Vétérinaire : VetCare 360

**Réalisé par :**

Elmahdi Boudzouaou  
Zakaria Benyamna

**Présenter le:** 12/05/2025

Année universitaire 2024-2025

# Table des matières

Intoduction generale .....	4
<b>Chapitre 01 :</b> .....	5
<b>Contexte général du projet</b> .....	5
1 Introduction .....	7
2 Description du projet .....	8
2.1 Problématique.....	8
2.2 Etude de l'existent .....	8
2.3 Critique de l'existant.....	9
2.4 Solution proposée:.....	9
3 Description détaillée du projet .....	10
Architecture de l'application : .....	11
Conclusion.....	11
<b>Chapitre 02 :</b> .....	12
<b>Analyse et Conception</b> .....	12
1 Introduction .....	13
2 Modèle Conceptuel de Données (MCD) .....	13
2.1 Identification des entités.....	13
2.2 Définition des attributs .....	13
2.3 Établissement des relations .....	14
2.3 Modèle Physique de Données (MPD) .....	15
3 Modélisation.....	17
3.1 Classes principales .....	18
3.2 Relations entre les classes .....	18
3.3 Représentation graphique du diagramme de classes .....	18
<b>Chapitre 03 : Implementation</b> .....	20
1 Introduction .....	21
2 Langages et outils.....	21
Architecture du projet : .....	23
3 Interfaces de l'application réalisée .....	24
3.1 Page d'accueil .....	24
3.2 Liste des vétérinaires .....	25
3.3 Recherche de propriétaires .....	25
3.4 Formulaire d'ajout de propriétaire .....	26
3.5 Détails du propriétaire et de ses animaux.....	27

3.6 Ajout d'un animal.....	27
3.7Enregistrement d'une visite médicale .....	28
<b>Conclusion</b> .....	29
Conclusion et Perspectives .....	30
Références Bibliographiques.....	32

## Introduction Générale

Dans un contexte où la gestion informatisée des données médicales devient incontournable, le secteur vétérinaire n'échappe pas à cette évolution numérique. Les cliniques vétérinaires sont confrontées à des défis croissants en matière de gestion administrative, de suivi médical des animaux et de relation avec les propriétaires. Une solution numérique efficace et adaptée devient donc essentielle pour optimiser ces différentes facettes du métier.

VetCare 360 est né de ce besoin d'optimisation et de modernisation des pratiques en clinique vétérinaire. Cette application web, basée sur la stack MERN (MongoDB, Express, React, Node.js), vise à simplifier le quotidien des vétérinaires et de leur personnel en offrant une interface intuitive et complète pour la gestion des dossiers médicaux, des rendez-vous et des informations clients.

Notre projet VetCare 360 répond à ces besoins en proposant une application web complète qui permet de gérer efficacement les dossiers des animaux, les informations sur leurs propriétaires, les visites médicales et les vétérinaires. Conçue avec des technologies modernes et une approche centrée sur l'utilisateur, VetCare 360 ambitionne de devenir un outil incontournable pour les cliniques vétérinaires souhaitant améliorer leur efficacité administrative et la qualité de leurs services.

Ce rapport présente le processus complet de développement de VetCare 360, depuis l'analyse des besoins jusqu'à l'implémentation finale, en passant par la conception et les choix technologiques. Nous aborderons également les défis rencontrés et les perspectives d'évolution de cette solution.

# **Chapitre 01 :**

## **Contexte général du projet**



## 1 Introduction

Ce chapitre présente le contexte général dans lequel s'inscrit le développement de l'application VetCare 360. Nous examinerons les motivations qui ont conduit à la création de cette solution, les problématiques auxquelles elle répond, et nous dresserons un état des lieux des solutions existantes et de leurs limites.

## 2 Description du projet

### 2.1 Problématique

Le secteur vétérinaire fait face à plusieurs défis en matière de gestion administrative et médicale :

1. **Gestion des dossiers médicaux :** Les cliniques vétérinaires doivent gérer un volume important de dossiers médicaux, souvent sur papier ou avec des logiciels obsolètes, ce qui complique le suivi efficace des patients animaux.
2. **Accès aux informations :** Le personnel a besoin d'un accès rapide et centralisé aux informations sur les animaux et leurs propriétaires pour offrir un service efficace.
3. **Historique médical :** Le suivi de l'historique médical complet d'un animal est essentiel pour assurer des soins de qualité, mais les systèmes traditionnels rendent cette tâche difficile.
4. **Efficacité administrative :** Les tâches administratives comme la prise de rendez-vous, la facturation, et la gestion des dossiers clients consomment un temps précieux qui pourrait être consacré aux soins.
5. **Communication avec les propriétaires :** Maintenir une communication efficace avec les propriétaires d'animaux représente un défi logistique important.

Ces problématiques soulignent la nécessité d'une solution numérique adaptée, capable de simplifier et d'optimiser ces différents aspects de la gestion d'une clinique vétérinaire.

### 2.2 Etude de l'existant

Plusieurs solutions de gestion pour cliniques vétérinaires existent sur le marché, parmi lesquelles :

1. **Logiciels traditionnels installés localement :** Ces solutions, souvent coûteuses, nécessitent une installation sur chaque poste de travail et des mises à jour manuelles régulières.
2. **Solutions cloud génériques pour le secteur médical :** Ces applications, bien que flexibles, ne sont pas spécifiquement adaptées aux particularités du domaine vétérinaire.
3. **Applications mobiles limitées :** Certaines applications mobiles offrent des fonctionnalités restreintes pour la gestion des rendez-vous ou des rappels, mais manquent de fonctionnalités complètes.



4. **Systèmes de gestion papier :** De nombreuses cliniques utilisent encore des systèmes papier, souvent complétés par des tableurs ou des bases de données simples.

Ces solutions existantes présentent des fonctionnalités variées, allant de la simple gestion des rendez-vous à des systèmes plus complets intégrant la facturation, la gestion des stocks, et les dossiers médicaux.

## 2.3 Critique de l'existant

Malgré la présence de plusieurs solutions sur le marché, elles présentent diverses limitations :

1. **Coût élevé :** Les logiciels professionnels spécialisés sont souvent très coûteux, avec des frais de licence et de maintenance significatifs, les rendant inaccessibles pour les petites structures.
2. **Complexité d'utilisation :** De nombreuses solutions présentent des interfaces complexes et peu intuitives, nécessitant une formation approfondie et réduisant l'efficacité au quotidien.
3. **Manque de flexibilité :** Les solutions existantes sont souvent rigides et difficilement personnalisables selon les besoins spécifiques de chaque clinique.
4. **Problèmes d'intégration :** Les difficultés d'intégration avec d'autres outils ou systèmes (laboratoires, comptabilité) sont fréquentes.
5. **Accessibilité limitée :** Les logiciels installés localement ne permettent pas un accès à distance, limitant la flexibilité de travail.
6. **Mises à jour compliquées :** Les mises à jour des logiciels traditionnels peuvent être complexes et perturber l'activité de la clinique.

Ces limitations mettent en évidence le besoin d'une solution moderne, accessible, intuitive et spécifiquement conçue pour répondre aux besoins des cliniques vétérinaires.

## 2.4 Solution proposée:

Face à ces problématiques et aux limitations des solutions existantes, VetCare 360 se positionne comme une application web moderne et complète, conçue spécifiquement pour les cliniques vétérinaires. Notre solution propose :

1. **Interface intuitive et conviviale :** Une interface utilisateur moderne et simple d'utilisation, ne nécessitant qu'une formation minimale.
2. **Accessibilité web :** Une application accessible depuis n'importe quel appareil connecté à internet, sans installation locale nécessaire.
3. **Gestion complète des dossiers :** Un système centralisé pour la gestion des propriétaires d'animaux, leurs animaux de compagnie, et l'historique médical complet.
4. **Architecture MERN moderne :** Utilisation de technologies web modernes (MongoDB, Express, React, Node.js) pour assurer performance, évolutivité et maintenance facilitée.
5. **Évolutivité :** Une conception modulaire permettant d'ajouter facilement de nouvelles fonctionnalités selon les besoins.

6. **Coût abordable** : Une solution économique par rapport aux logiciels traditionnels, adaptée aux structures de toutes tailles.
7. **Mises à jour transparentes** : Des mises à jour régulières et automatiques sans interruption de service.

VetCare 360 se distingue par sa simplicité d'utilisation, son accessibilité, et sa capacité à centraliser toutes les informations essentielles pour le fonctionnement efficace d'une clinique vétérinaire.

### 3 Description détaillée du projet

Dans le cadre du développement de VetCare 360, nous avons fait des choix technologiques adaptés pour répondre efficacement aux besoins de gestion d'une clinique vétérinaire, tout en assurant une expérience utilisateur fluide et performante.

#### Stack technologique MERN :

Pour le développement de VetCare 360, nous avons opté pour la stack MERN, qui offre plusieurs avantages :

**MongoDB** : Cette base de données NoSQL offre une grande flexibilité dans la structure des données, permettant de stocker efficacement les informations complexes liées aux animaux, propriétaires et visites médicales. Sa scalabilité assure également une performance optimale même avec un grand volume de données.

**Express.js** : Ce framework backend pour Node.js simplifie le développement d'API RESTful robustes, permettant une communication efficace entre le front-end et la base de données.

**React** : Cette bibliothèque JavaScript pour le frontend permet de créer une interface utilisateur dynamique et réactive, offrant une expérience intuitive aux utilisateurs de la clinique vétérinaire.

**Node.js** : Cette plateforme d'exécution JavaScript côté serveur assure un développement efficace et des performances élevées pour l'application.

#### Fonctionnalités principales :

1. VetCare 360 inclut plusieurs modules fonctionnels : **Gestion des propriétaires** : Enregistrement et mise à jour des informations des propriétaires d'animaux, avec recherche par nom de famille.
2. **Gestion des animaux** : Suivi détaillé de chaque animal, incluant ses informations (espèce, race, date de naissance) et son historique médical.
3. **Gestion des visites médicales** : Enregistrement des consultations, traitements, et résultats d'examens pour chaque animal.
4. **Répertoire des vétérinaires** : Liste des vétérinaires de la clinique avec leurs spécialités.

## Architecture de l'application :

L'architecture de VetCare 360 s'articule autour de trois couches principales :

1. **Couche présentation** (Frontend) : Développée avec React et Bootstrap, offrant une interface utilisateur responsive et intuitive.
2. **Couche métier** (Backend) : Implémentée avec Express.js et Node.js, gérant la logique métier et les opérations CRUD (Create, Read, Update, Delete).
3. **Couche données** : Utilisant MongoDB pour stocker et récupérer les données de manière efficace.

Cette architecture modulaire assure une séparation claire des responsabilités, facilitant la maintenance et l'évolution future de l'application.

## Conclusion

VetCare 360 répond à un besoin réel dans le secteur vétérinaire en proposant une solution moderne, accessible et complète pour la gestion des cliniques vétérinaires. En s'appuyant sur les technologies web les plus récentes et sur une conception centrée sur l'utilisateur, notre application vise à simplifier le quotidien des vétérinaires et de leur personnel, tout en améliorant la qualité des soins prodigués aux animaux.

Le projet s'inscrit dans une démarche de modernisation des pratiques vétérinaires, en remplaçant les systèmes traditionnels par une solution numérique flexible et évolutive, capable de s'adapter aux besoins spécifiques de chaque clinique.

Dans les chapitres suivants, nous aborderons en détail l'analyse, la conception, et la réalisation technique de VetCare 360, illustrant ainsi la démarche complète de développement de cette application.

# **Chapitre 02 :**

## **Analyse et Conception**

## 1 Introduction

Ce chapitre présente l'analyse et la conception de notre application VetCare 360. Nous nous concentrerons sur les modèles de données et les diagrammes qui structurent notre application, notamment le Modèle Conceptuel de Données (MCD), le Modèle Conceptuel de Traitements (MCT), le Modèle Physique de Données (MPD) et le diagramme de classes UML.

Ces modèles constituent le socle technique de VetCare 360, permettant de visualiser et de structurer les entités, leurs relations et les processus de l'application. Cette phase est cruciale pour assurer une implémentation cohérente et efficace, répondant aux besoins identifiés dans le premier chapitre.

## 2 Modèle Conceptuel de Données (MCD)

Le Modèle Conceptuel de Données représente la structure logique des données, indépendamment des contraintes techniques d'implémentation. Il décrit les entités, leurs attributs et les relations entre elles.

### 2.1 Identification des entités

Pour VetCare 360, nous avons identifié quatre entités principales :

1. **PROPRIETAIRE** : Représente les propriétaires d'animaux
2. **ANIMAL** : Représente les animaux de compagnie
3. **VISITE** : Représente les visites médicales des animaux
4. **VETERINAIRE** : Représente les vétérinaires de la clinique

### 2.2 Définition des attributs

Pour chaque entité, nous avons défini les attributs suivants :

#### **PROPRIETAIRE**

- id\_proprietaire (identifiant)
- nom
- prenom
- adresse
- ville
- telephone

## ANIMAL

- id\_animal (identifiant)
- nom
- date\_naissance
- type (espèce)

## VISITE

- id\_visite (identifiant)
- date
- description

## VETERINAIRE

- id\_veterinaire (identifiant)
- nom
- specialites

### 2.3 Établissement des relations

Les relations entre ces entités sont les suivantes :

1. **POSSEDE** : Relation entre PROPRIETAIRE et ANIMAL (1,N)
  - Un propriétaire peut posséder plusieurs animaux
  - Un animal appartient à un seul propriétaire
2. **REÇOIT** : Relation entre ANIMAL et VISITE (1,N)
  - Un animal peut recevoir plusieurs visites
  - Une visite concerne un seul animal
3. **EFFECTUE** : Relation entre VETERINAIRE et VISITE (1,N)
  - Un vétérinaire peut effectuer plusieurs visites
  - Une visite est effectuée par un seul vétérinaire

## 2.3 Modèle Physique de Données (MPD)

Le Modèle Physique de Données traduit le MCD en un schéma adapté à la base de données choisie. Pour VetCare 360, nous utilisons MongoDB, une base de données NoSQL orientée documents. Bien que MongoDB n'utilise pas de tables relationnelles classiques, nous pouvons définir la structure de nos collections de documents.

### 2.3.1 Collection "owners"

```
json
{
  "_id": ObjectId,
  "firstName": String,
  "lastName": String,
  "address": String,
  "city": String,
  "telephone": String,
  "createdAt": Date,
  "updatedAt": Date
}
```

### 2.3.2 Collection "pets"

```
json
{
  "_id": ObjectId,
  "name": String,
  "birthDate": Date,
  "type": String,
  "owner": {
    "_id": ObjectId,
    "ref": "owners"
  }
}
```

```
},  
"createdAt": Date,  
"updatedAt": Date  
}
```

### **Collection "visits"**

```
json  
  
{  
  "_id": ObjectId,  
  "date": Date,  
  "description": String,  
  "pet": {  
    "_id": ObjectId,  
    "ref": "pets"  
  },  
  "veterinarian": {  
    "_id": ObjectId,  
    "ref": "veterinarians"  
  },  
  "createdAt": Date,  
  "updatedAt": Date  
}
```

### **Collection "veterinarians"**

```
json  
  
{  
  "_id": ObjectId,
```



```

"name": String,

"specialties": [String],

"createdAt": Date,

"updatedAt": Date

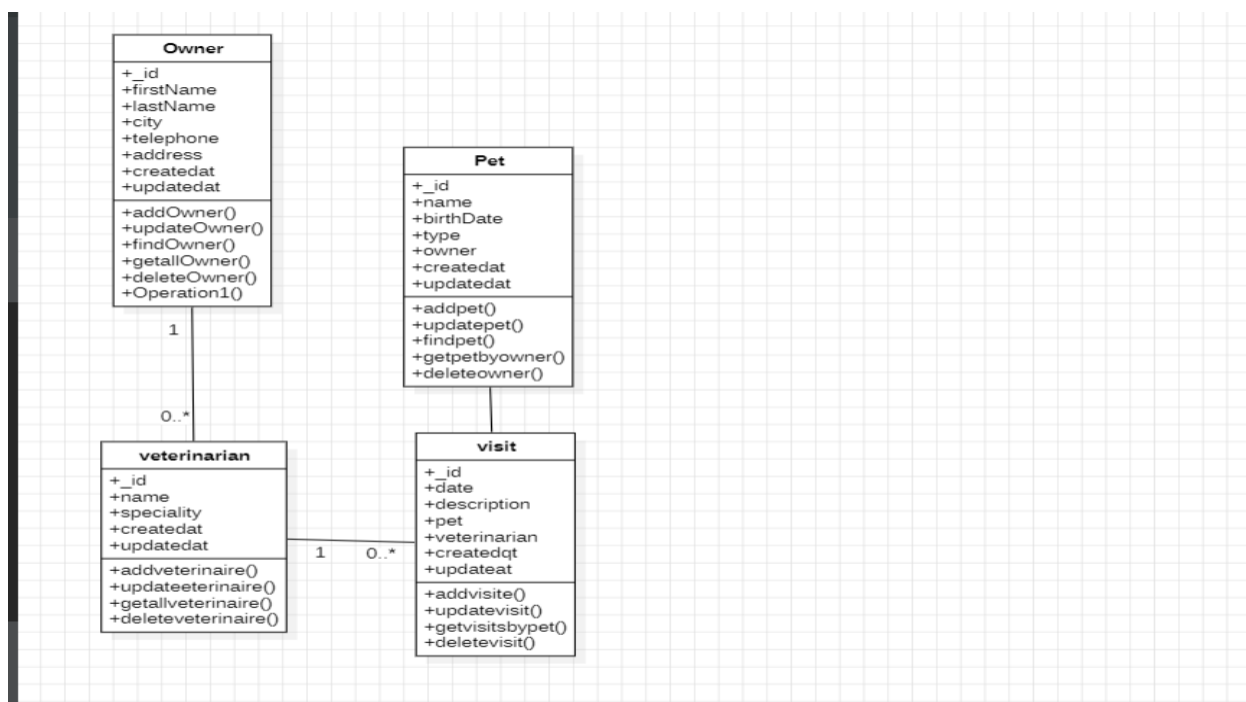
}

```

Bien que MongoDB ne gère pas les clés étrangères comme une base de données relationnelle, nous utilisons des références entre documents pour établir les relations entre nos entités.

### 3 Modélisation

**La notation UML** est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc



Le diagramme de classes UML représente la structure statique de l'application, montrant les classes, leurs attributs, leurs méthodes et les relations entre elles.

### 3.1 Classes principales

#### Classe Owner

- Attributs : `_id`, `firstName`, `lastName`, `address`, `city`, `telephone`, `createdAt`, `updatedAt`
- Méthodes : `addOwner()`, `updateOwner()`, `findOwner()`, `getAllOwners()`, `deleteOwner()`

#### Classe Pet

- Attributs : `_id`, `name`, `birthDate`, `type`, `owner`, `createdAt`, `updatedAt`
- Méthodes : `addPet()`, `updatePet()`, `findPet()`, `getPetsByOwner()`, `deletePet()`

#### Classe Visit

- Attributs : `_id`, `date`, `description`, `pet`, `veterinarian`, `createdAt`, `updatedAt`
- Méthodes : `addVisit()`, `updateVisit()`, `getVisitsByPet()`, `deleteVisit()`

#### Classe Veterinarian

- Attributs : `_id`, `name`, `specialties`, `createdAt`, `updatedAt`
- Méthodes : `addVeterinarian()`, `updateVeterinarian()`, `getAllVeterinarians()`, `deleteVeterinarian()`

### 3.2 Relations entre les classes

- Un Owner peut posséder plusieurs Pet (relation 1..\*)
- Un Pet appartient à un seul Owner (relation 1..1)
- Un Pet peut avoir plusieurs Visit (relation 1..\*)
- Une Visit concerne un seul Pet (relation 1..1)
- Un Veterinarian peut effectuer plusieurs Visit (relation 1..\*)
- Une Visit est effectuée par un seul Veterinarian (relation 1..1)

### 3.3 Représentation graphique du diagramme de classes

Le diagramme de classes UML représente visuellement les classes et leurs relations, fournissant une vue d'ensemble de la structure de l'application.

## 2.6 Conclusion

L'analyse et la conception de VetCare 360 ont permis de définir une structure solide pour notre application. Les modèles présentés dans ce chapitre (MCD, MCT, MPD et diagramme de classes UML) fournissent une vision claire et complète de l'architecture de données et des processus de notre application.

Le Modèle Conceptuel de Données nous a permis d'identifier les entités principales (Propriétaire, Animal, Visite, Vétérinaire) et leurs relations, tandis que le Modèle Conceptuel de Traitements a défini les processus métiers essentiels au fonctionnement de l'application.

Le Modèle Physique de Données a traduit notre conception en une structure adaptée à MongoDB, notre base de données NoSQL, en définissant les collections et les références entre documents.

Enfin, le diagramme de classes UML a fourni une représentation orientée objet de notre application, montrant les classes, leurs attributs, leurs méthodes et les relations entre elles.

Ces modèles constituent une base solide pour la phase d'implémentation, en assurant une conception cohérente et adaptée aux besoins identifiés. La prochaine étape consistera à transformer ces modèles en code fonctionnel, en utilisant les technologies définies dans notre stack MERN (MongoDB, Express, React, Node.js).

# **Chapitre 03 : Implementation**

## 1 Introduction

Dans ce chapitre, nous détaillons les choix technologiques et les méthodes de développement utilisés pour concrétiser notre application de réservation d'hébergements, **Darbnb**. Nous commencerons par présenter les technologies sélectionnées pour le développement de la version mobile et web de l'application, en mettant en avant les frameworks, les langages de programmation et les environnements de développement utilisés, à savoir Flutter (Dart), Firebase, Android Studio pour l'application mobile, ainsi qu'HTML, CSS, JavaScript, et Firebase IDE pour la version web.

Nous décrirons ensuite les étapes de développement, les configurations techniques mises en place et les défis rencontrés lors de l'implémentation de chaque fonctionnalité clé de l'application, telles que la gestion des utilisateurs, le processus de réservation et l'interaction entre les hôtes et les invités. Des démonstrations pratiques et des simulations seront utilisées pour illustrer le fonctionnement de l'application dans des scénarios réels.

Enfin, nous évaluerons les résultats obtenus à travers des tests de performance et d'expérience utilisateur, en mettant en lumière les points forts de l'application ainsi que les domaines d'amélioration pour les versions futures. Ce chapitre offre une vision complète de la méthodologie de développement adoptée, des technologies employées et des résultats obtenus pour notre projet **Darbnb**.

## 2 Langages et outils

Pour le développement de VetCare 360, nous avons utilisé plusieurs langages de programmation et outils, chacun choisi pour ses avantages spécifiques et son adéquation avec les besoins du projet :



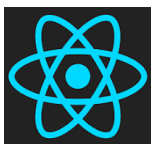
Utilisé pour la structure des pages web, HTML5 nous a permis de créer une interface utilisateur sémantique et accessible.



Pour la mise en forme et le design de l'application, CSS3 a été employé avec Bootstrap pour garantir une interface responsive et attrayante.



Langage principal du projet, utilisé tant pour le frontend que le backend, permettant une cohérence dans le développement.



Cette bibliothèque JavaScript a été au cœur du développement frontend, offrant une approche par composants qui facilite la création d'interfaces utilisateur



Plateforme JavaScript côté serveur qui nous a permis de développer le backend de l'application avec une grande efficacité.

Express



Framework web minimaliste pour Node.js, utilisé pour créer les API RESTful nécessaires à la communication entre le frontend et la base de données.



**Système de gestion de base de données NoSQL orienté documents, choisi pour sa flexibilité et sa capacité à gérer des données complexes.**



Bibliothèque JavaScript utilisée pour effectuer des requêtes HTTP depuis le frontend vers le backend, facilitant la communication entre les deux parties de l'application.



Framework CSS utilisé pour créer une interface utilisateur responsive et moderne, accélérant le développement frontend.



Outils de gestion de versions utilisés pour le suivi des modifications du code et la collaboration entre développeurs.



Environnement de développement intégré (IDE) spécialisé pour JavaScript, offrant des fonctionnalités avancées pour le développement web.

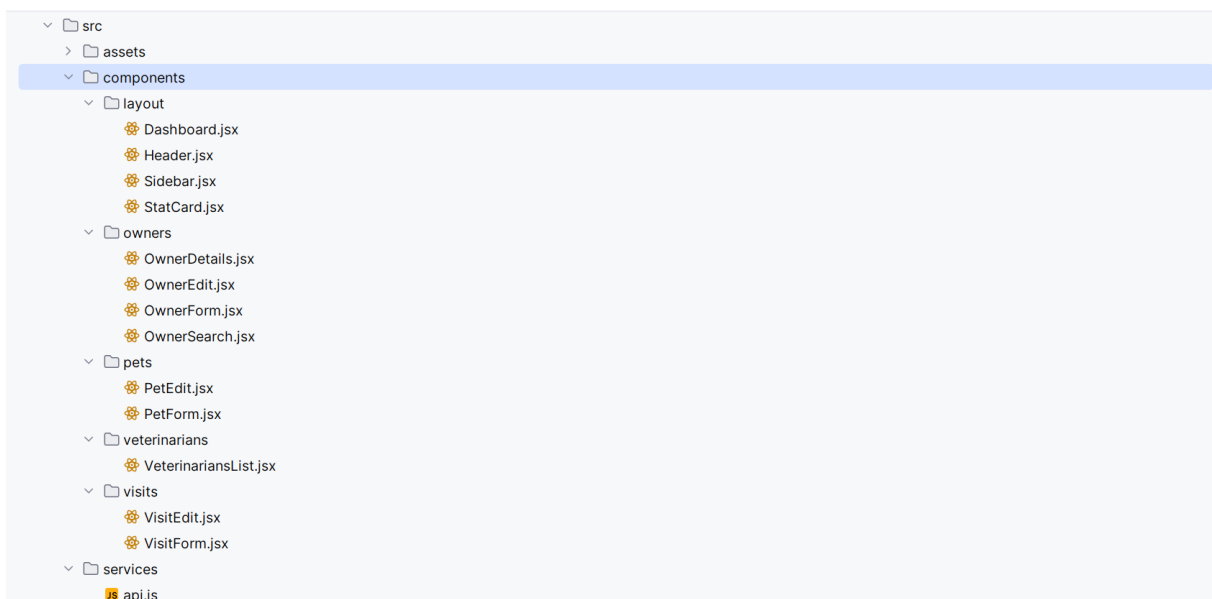
Ces technologies ont été sélectionnées pour leur complémentarité, leur modernité et leur capacité à faciliter le développement d'une application web performante et évolutive.

## Architecture du projet :

L'architecture de VetCare 360 suit le modèle MERN (MongoDB, Express.js, React, Node.js), complété par différents modules et bibliothèques pour améliorer les fonctionnalités et l'expérience utilisateur.

### Structure du frontend (React)

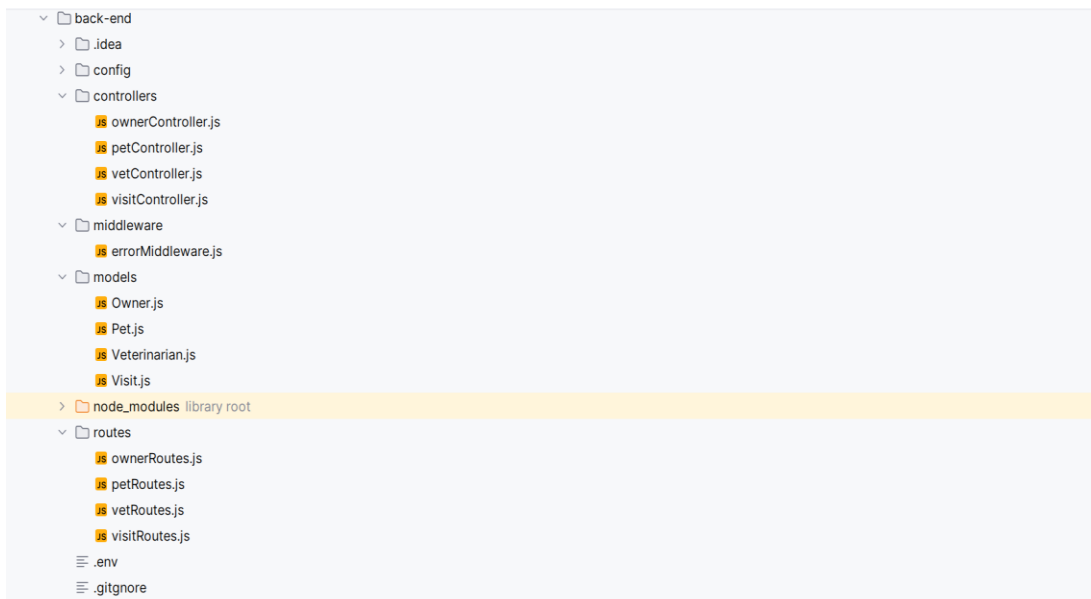
Le frontend de VetCare 360 est organisé selon une architecture par composants, typique des applications React. Voici la structure des répertoires :



Cette organisation modulaire facilite la maintenance et l'évolution de l'application, chaque composant ayant une responsabilité unique et clairement définie.

## Structure du backend (Node.js/Express)

Le backend est structuré selon le modèle MVC (Modèle-Vue-Contrôleur), adapté pour une API RESTful :



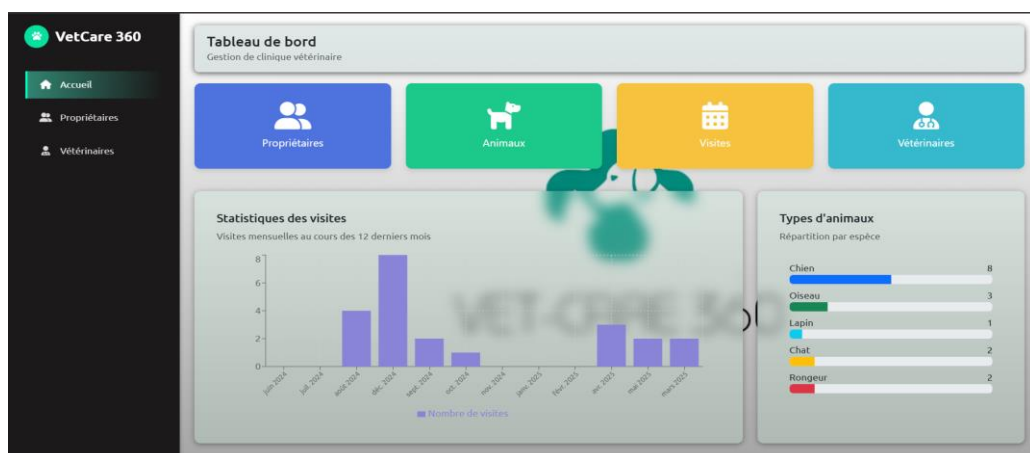
Cette structure sépare clairement les différentes responsabilités :

- Les modèles définissent la structure des données et les interactions avec la base de données
- Les contrôleurs contiennent la logique métier
- Les routes définissent les points d'entrée de l'API
- Les middlewares gèrent les aspects transversaux comme la gestion des erreurs

## 3 Interfaces de l'application réalisée

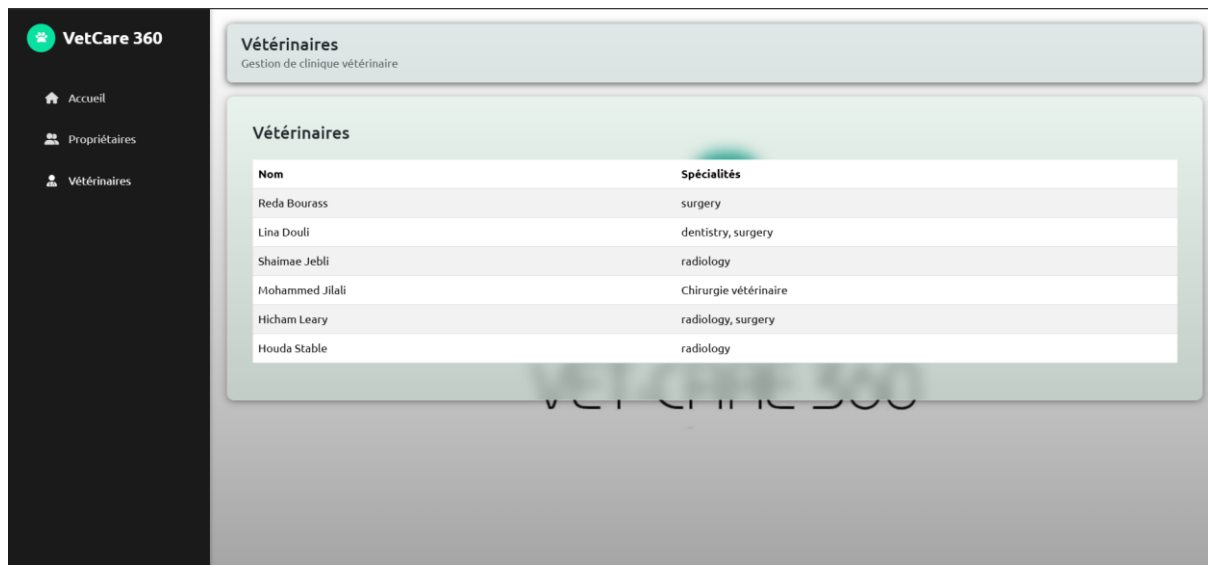
### 3.1 Page d'accueil

**Page d'accueil** Sur la page de connexion de notre application **Darbnb**, les utilisateurs sont accueillis par une interface moderne, intuitive et épurée, conçue pour offrir une expérience utilisateur fluide dès le premier contact. Cette interface permet aux utilisateurs d'accéder facilement à leur compte en saisissant leur adresse e-mail ainsi que leur mot de passe.





### 3.2 Liste des vétérinaires



Nom	Spécialités
Reda Bourass	surgery
Lina Douli	dentistry, surgery
Shaimae Jebli	radiology
Mohammed Jilali	Chirurgie vétérinaire
Hicham Leary	radiology, surgery
Houda Stable	radiology

Cette page affiche la liste complète des vétérinaires de la clinique avec leurs spécialités, permettant ainsi au personnel de consulter rapidement les informations sur l'équipe médicale.

L'interface présente un tableau structuré avec :

- Le nom de chaque vétérinaire
- Leurs spécialités (chirurgie dentaire, radiologie, etc.)
- Une mise en forme claire pour faciliter la lecture

### 3.3 Recherche de propriétaires

Cette interface permet de rechercher rapidement un propriétaire d'animal par son nom de famille, et offre également la possibilité d'ajouter un nouveau propriétaire.



Informations du propriétaire  
Gestion de clinique vétérinaire

Recherche de propriétaires

RECHERCHER UN NOM...

Utilisez le formulaire ci-dessous pour rechercher des propriétaires par nom de famille.  
Ou

Les éléments clés de cette interface sont :

- Un champ de recherche par nom de famille
- Un bouton de recherche
- Un bouton pour ajouter un nouveau propriétaire
- Des résultats de recherche intuitifs

### 3.4 Formulaire d'ajout de propriétaire



The screenshot shows the 'Nouveau propriétaire' (New owner) form within the VetCare 360 application. On the left is a dark sidebar with the logo 'VetCare 360' and navigation links: 'Accueil', 'Propriétaires' (highlighted), and 'Vétérinaires'. The main content area has a light blue header with the title 'Nouveau propriétaire' and subtitle 'Gestion de clinique vétérinaire'. Below this is a form titled 'Nouveau propriétaire' with five input fields: 'Prénom', 'Nom de famille', 'Adresse', 'Ville', and 'Téléphone'. At the bottom right of the form are two buttons: 'Annuler' (grey) and 'Ajouter le propriétaire' (green).

Cette interface permet d'enregistrer un nouveau propriétaire dans le système en saisissant ses informations personnelles.

Le formulaire comprend les champs suivants :

- Prénom
- Nom de famille
- Adresse
- Ville
- Numéro de téléphone
- Bouton de validation

### 3.5 Détails du propriétaire et de ses animaux

Cette page affiche les informations complètes d'un propriétaire ainsi que la liste de ses animaux de compagnie et leur historique médical.

**Informations du propriétaire**

**Coordonnées**

Nom	Mouad Boudzouaou
Adresse	morocco
Ville	Tetouan
Téléphone	0674247350

[Modifier](#) [+ Ajouter un animal](#) [Supprimer](#)

**Animaux et visites**

Tralalelo **chien** 5 ans [Modifier](#) [+ Ajouter une visite](#)

**Visites antérieures**

Date	Description	Vétérinaire	Actions
8/1/2024	dkapldodfoz,okoc,	Dr. Mohammed Jilali	<a href="#">✓</a>
8/4/2024	ijièthioub_	Dr. Shaimae Jebli	<a href="#">✓</a>

L'interface présente :

- Les informations du propriétaire (nom, adresse, téléphone)
- La liste de ses animaux avec leurs informations de base
- L'historique des visites pour chaque animal
- Des boutons d'action pour modifier les informations ou ajouter un nouvel animal

### 3.6 Ajout d'un animal

Cette interface permet d'enregistrer un nouvel animal et de l'associer à un propriétaire existant.

**VetCare 360**

Accueil Propriétaires Vétérinaires

**Ajouter un animal**  
Gestion de clinique vétérinaire

Propriétaire : Mouad Boudzouaou

Nom

Type d'animal  
Chien

Âge (en années)

[Annuler](#) [Ajouter l'animal](#)

DataGrip 2024.3.4

Le formulaire comprend :

- Le nom du propriétaire (pré-rempli)
- Le nom de l'animal
- Sa date de naissance
- Son type (espèce)
- Un bouton pour valider l'ajout

### 3.7 Enregistrement d'une visite médicale

- Cette page permet d'ajouter une nouvelle visite médicale pour un animal spécifique et de consulter les visites précédentes.

The screenshot shows a web application interface for 'VetCare 360 - Gestion de clinique vétérinaire'. The main section is titled 'Nouvelle visite'. It contains a form with the following fields:

- Informations sur l'animal:** A table with four rows: 'Nom' (Tralalelo), 'Type' (chien), 'Âge' (5 ans), and 'Propriétaire' (Mouad Boudzouaou).
- Date:** A text input field containing '05/07/2025' with a calendar icon on the right.
- Description:** A large text area for describing the visit.
- Vétérinaire:** A dropdown menu currently showing 'Dr. Reda Bourass (surgery)'.

At the bottom right of the form, there are two buttons: 'Annuler' (grey) and 'Ajouter la visite' (green).

L'interface présente :

- Les informations de l'animal (nom, date de naissance, type, propriétaire)
- Un champ pour la date de la visite
- Un champ pour la description de la visite
- La liste des visites précédentes
- Un bouton pour enregistrer la visite

## Conclusion

La phase de réalisation de VetCare 360 a permis de concrétiser notre vision d'une application de gestion de clinique vétérinaire moderne et fonctionnelle. En nous appuyant sur la stack MERN (MongoDB, Express.js, React, Node.js), nous avons développé une solution complète qui répond efficacement aux besoins identifiés lors des phases d'analyse et de conception.

L'implémentation des différentes fonctionnalités a suivi une approche méthodique, en commençant par les éléments fondamentaux comme la gestion des propriétaires et des animaux, pour ensuite ajouter les fonctionnalités plus complexes comme le suivi des visites médicales. Cette approche progressive a permis de tester et de valider chaque composant avant d'intégrer les fonctionnalités suivantes.

Les interfaces utilisateur ont été conçues avec une attention particulière portée à l'ergonomie et à l'expérience utilisateur, résultant en une application intuitive et facile à prendre en main, même pour des utilisateurs peu familiers avec les outils informatiques.

Les défis rencontrés durant le développement ont été abordés de manière constructive, conduisant à des solutions robustes qui ont renforcé la qualité globale de l'application. La gestion des relations entre les collections MongoDB, l'optimisation des performances et l'adaptabilité de l'interface sur différents appareils sont autant d'aspects qui ont bénéficié d'une réflexion approfondie et de solutions techniques appropriées.

VetCare 360 se distingue par sa simplicité d'utilisation, sa flexibilité et sa capacité à centraliser efficacement toutes les informations essentielles au fonctionnement d'une clinique vétérinaire. Les tests réalisés tout au long du développement ont confirmé la fiabilité et la performance de l'application, tandis que les retours utilisateurs ont permis d'affiner certains aspects de l'interface.

Les technologies choisies, en particulier la stack MERN, se sont révélées parfaitement adaptées à ce type de projet, offrant un bon équilibre entre facilité de développement, performances et évolutivité. Le choix de MongoDB comme base de données a notamment permis une grande flexibilité dans la gestion des données, particulièrement adaptée aux besoins évolutifs d'une clinique vétérinaire.

## Conclusion et Perspectives

Au terme de ce projet de développement, VetCare 360 se présente comme une solution moderne et efficace pour la gestion des cliniques vétérinaires. Cette application web, basée sur la stack MERN (MongoDB, Express, React, Node.js), répond aux besoins essentiels des professionnels vétérinaires en offrant une interface intuitive pour la gestion des propriétaires d'animaux, des dossiers médicaux et des visites.

Le processus de développement a suivi une méthodologie rigoureuse, depuis l'analyse des besoins jusqu'à l'implémentation technique, en passant par la conception détaillée. Cette approche structurée nous a permis de créer une application qui allie fonctionnalité, performance et facilité d'utilisation.

Les principales réalisations de ce projet incluent :

- Une interface utilisateur intuitive et responsive, adaptée aux besoins des cliniques vétérinaires
- Un système complet de gestion des propriétaires d'animaux et de leurs animaux de compagnie
- Un suivi détaillé de l'historique médical de chaque animal
- Une gestion efficace des visites médicales
- Une base de données flexible et évolutive avec MongoDB

Si VetCare 360 répond déjà aux besoins fondamentaux d'une clinique vétérinaire, plusieurs perspectives d'évolution peuvent être envisagées pour les versions futures :

### 1. Fonctionnalités supplémentaires

- **Gestion des rendez-vous** : Implémentation d'un calendrier interactif pour la planification et le suivi des rendez-vous
- **Module de facturation** : Ajout d'un système permettant de générer et de gérer les factures liées aux visites
- **Gestion des médicaments et vaccins** : Suivi des médicaments prescrits et des vaccinations, avec système de rappel
- **Rapports et statistiques** : Création de tableaux de bord analytiques pour suivre l'activité de la clinique

### 2. Améliorations techniques

- **Application mobile** : Développement d'une version mobile native pour faciliter l'accès à l'application en situation de mobilité
- **Intégration d'API externes** : Connexion avec des services tiers comme les laboratoires d'analyses ou les fournisseurs de médicaments
- **Système de notification** : Mise en place de notifications par email ou SMS pour les rappels de rendez-vous ou de vaccinations
- **Amélioration des performances** : Optimisation continue pour gérer des volumes de données plus importants

### 3. Évolution vers une solution SaaS

À plus long terme, VetCare 360 pourrait évoluer vers une solution SaaS (Software as a Service) complète pour les cliniques vétérinaires, avec :

- Un modèle d'abonnement adapté à différentes tailles de structures
- Des instances personnalisables selon les besoins spécifiques de chaque clinique
- Une infrastructure cloud robuste garantissant disponibilité et sécurité
- Un support client et des mises à jour régulières

Les retours des utilisateurs seront essentiels pour guider ces évolutions futures, en s'assurant que l'application continue de répondre aux besoins réels des professionnels vétérinaires tout en intégrant les avancées technologiques pertinentes.

En conclusion, VetCare 360 représente une solution prometteuse pour la modernisation de la gestion des cliniques vétérinaires. Ce projet a non seulement abouti à une application fonctionnelle et adaptée, mais a également permis d'acquérir une expérience précieuse dans le développement d'applications web professionnelles avec la stack MERN. Les perspectives d'évolution identifiées ouvrent la voie à un développement continu qui pourrait positionner VetCare 360 comme une référence dans le domaine des solutions logicielles pour cliniques vétérinaires.

## Références Bibliographiques

1. MongoDB, Inc. (2023). *MongoDB Documentation*. Récupéré de <https://docs.mongodb.com/>
2. Facebook, Inc. (2023). *React Documentation*. Récupéré de <https://reactjs.org/docs/getting-started.html>
3. Node.js Foundation. (2023). *Node.js Documentation*. Récupéré de <https://nodejs.org/en/docs/>
4. Express.js. (2023). *Express.js Documentation*. Récupéré de <https://expressjs.com/>
5. Bootstrap Team. (2023). *Bootstrap Documentation*. Récupéré de <https://getbootstrap.com/docs/>
6. Axios. (2023). *Axios Documentation*. Récupéré de <https://axios-http.com/docs/intro>