المدرسة العليا للتكنولوجيا الناظور
École Supérieure de Technologie de Nador
ⵜⵙⴷⵔⵉⵜ ⵜⴰⵣⵣⵉⵣⵜ ⵏ ⵜⴰⵜⵣⵔⵉⵙⵜ ⵏⵏⴰⴹⵓⵔ

جامعة محمد الأول بوجدة
UNIVERSITÉ MOHAMMED PREMIER OUJDA
ⵜⴰⵙⵙⵓⵏⵜ ⵎⵓⵃⴰⵎⵎⴻⴷ ⴰⵣⵔⴰⵡ ⵏ ⵡⵓⵊⴷⴰ

Projet de Fin de semestre

Diplôme Universitaire de Technologie

Filière : Ingénierie Logicielle et Cybersecurity
(ILCS)

Intitulé du projet :

Une application web pour la gestion d'une clinique vétérinaire Vetecare 360 « Version Mongodb »

Réalisé par:

Alae Majjati
&
Oumaima Marzak

Professeur :
Mr. Esbai Redouane

Année universitaire : 2024/2025

Table des matières

Introduction

VetCare 360 est une application web de gestion pour clinique vétérinaire basée sur l'architecture MERN (MongoDB, Express, React, Node.js). Ce système permettra de gérer efficacement les dossiers des propriétaires d'animaux, les informations sur les animaux de compagnie, les visites médicales et la liste des vétérinaires.

Ce compte rendu présente la méthodologie détaillée pour développer cette application en respectant les fonctionnalités demandées et en suivant les meilleures pratiques de développement web.

Analyse des besoins

Fonctionnalités principales

1. **Gestion des propriétaires d'animaux**
   - Ajouter de nouveaux propriétaires
   - Modifier les informations des propriétaires existants
   - Supprimer des propriétaires
   - Rechercher des propriétaires par nom de famille
2. **Gestion des animaux de compagnie**
   - Ajouter des animaux et les associer à leurs propriétaires
   - Modifier les informations des animaux
   - Consulter les détails des animaux
3. **Gestion des visites médicales**
   - Enregistrer les nouvelles visites
   - Consulter l'historique des visites par animal
   - Afficher les détails des visites
4. **Liste des vétérinaires**
   - Afficher la liste des vétérinaires disponibles

Conception de la base de données

Structure des collections MongoDB

Collection Proprietaires

```
json
{
  "_id": "ObjectId(‘’)",
  " firstName ": "String",
  " lastName ": "String",
  " email ": "String",
  " phone ": "String",
  " address ": "String",
  " createdAt ": "Date",
  " updatedAt ": "Date",
  " __v ": 0
}
```

Collection Animaux

```
json
{
  "_id": "ObjectId()",
  " name ": "String",
  " species ": "String",
  " breed ": "String",
  " birthDate ": "Date",
  " gender ": "String",
  " owner ": "ObjectId ()",
  " medicalHistory ": "String",
  "createdAt": "Date",
  "updatedAt": "Date",
  " __v ": 0
}
```

Collection Visites

json

```
{
  "_id": "ObjectId()",
  " pet ": "ObjectId (ref: Animaux)",
  " veterinarian ": "ObjectId (ref: Veterinaires)",
  "date": "Date",
  " reason ": "String",
  " diagnosis ": "String",
  " treatment ": "String",
  "notes": "String",
  "createdAt": "Date",
  "updatedAt": "Date",
  " __v " : 0
}
```

Collection Veterinaires

json

```
{
  "_id": "ObjectId()",
  " firstName ": "String",
  " lastName ": "String",
  " specialization ": "String",
  " email ": "String",
  " phone ": "String",
  " licenseNumber ": "String",
  "createdAt": "Date",
  "updatedAt": "Date",
  " __v " : 0
}
```

Architecture technique

Stack MERN

- **MongoDB(MongoDB Atlas)** : Base de données NoSQL orientée documents
- **Express.js** : Framework backend pour Node.js
- **React** : Bibliothèque JavaScript pour l'interface utilisateur
- **Node.js** : Environnement d'exécution JavaScript côté serveur

Outils supplémentaires

- **Bootstrap** : Framework CSS pour un design responsive
- **Mongoose** : ODM (Object Data Modeling) pour MongoDB
- **Git/GitHub** : Gestion de versions et collaboration

Installation et configuration

Prérequis

- Node.js (version 16.x ou supérieure)
- MongoDB (version 5.x ou supérieure)
- npm (installé avec Node.js)
- Git

Configuration initiale

Création de la structure du projet

```
mkdir vetcare360

cd vetcare360

mkdir backend frontend
```

Configuration du backend

```
cd backend
npm install

// Crier un fichier .env dans le repertoire backend avec se
contenue :
PORT=5000
MONGODB_URI=mongodb+srv://(nom d'utilisateur):(mot de passe)
@mernapp.3pdrd.mongodb.net/vetcare?retryWrites=true&w=majority&appNa
me=MERNapp
```

Configuration du frontend

```
cd ../frontend
npm install
```

Développement du backend et du frontend

Structure des dossiers

```
vetcare360/
├── backend/
│   ├── src/
│   │   ├── config/
│   │   ├── controllers/
│   │   ├── models/
│   │   └── routes/
│   ├── .env
│   ├── package.json
│   └── server.js
│
├── frontend/
│   ├── public/
│   ├── src/
│   │   ├── components/
│   │   ├── pages/
│   │   ├── services/
│   └── package.json
│
├── start.js
└── README.md
```

➢ Détails de chaque dossier backend :

a) config/ (Configuration)

- Contient les paramètres de configuration
- Gestion de la connexion MongoDB
- Configuration des variables d'environnement
- Middleware de sécurité

b) controllers/ (Contrôleurs)

- Gestion de la logique métier
- Fonctions pour :
    - Gestion des propriétaires d'animaux
    - Gestion des animaux
    - Gestion des visites
    - Gestion des vétérinaires

c) models/ (Modèles)

Schémas MongoDB

d) routes/ (Routes API)

Points d'entrée API :

```
/api/owners      # Gestion des propriétaires
/api/pets        # Gestion des animaux
/api/visits      # Gestion des visites
/api/vets        # Gestion des vétérinaires
```

➢ Détails de chaque dossier frontend :

a) components/ (Composants)

Composants réutilisables :

- `Header.js` : Barre de navigation
- `Footer.js` : Pied de page
- `Forms/` :
  - `OwnerForm.js`
  - `PetForm.js`
  - `VisitForm.js`
- `Tables/` :
  - `OwnersTable.js`
  - `PetsTable.js`
  - `VisitsTable.js`
- `Cards/` :
  - `PetCard.js`
  - `OwnerCard.js`
  - `VetCard.js`

b) pages/ (Pages)

Pages principales :

- `Dashboard.js` : Vue d'ensemble
- `Owners/` :
  - `OwnersList.js`
  - `OwnerDetails.js`
  - `AddOwner.js`

- `Pets/` :
    - `PetsList.js`
    - `PetDetails.js`
    - `AddPet.js`
- `Visits/` :
    - `VisitsList.js`
    - `ScheduleVisit.js`
    - `VisitDetails.js`

c) services/ (Services)

Services API :

```
- ownerService.js     // CRUD propriétaires
- petService.js       // CRUD animaux
- visitService.js     // CRUD visites
- vetService.js       // CRUD vétérinaires
```

Flux de Données

1. L'utilisateur interagit avec l'interface (frontend)
2. Le frontend envoie une requête API
3. Le backend traite la requête via les routes
4. Les contrôleurs gèrent la logique métier
5. Les modèles interagissent avec MongoDB
6. La réponse remonte jusqu'au frontend
7. L'interface se met à jour

Démarrage de l'application

Nous avons deux options pour démarrer cette application :

Option1 : -Démarrer le Backend et le Frontend séparément :

```
cd backend
npm start
// Le backend va se démarrer sur http://localhost:5000

cd frontend
npm start
// Le frontend va se démarrer sur http://localhost:3000
```

Option2 : -Utiliser la combinaison de Start à travers le fichier `start.js` :

```
node start.js
```

Conclusion :

VetCare 360 représente une solution complète et innovante pour la gestion des cliniques vétérinaires, développée avec la stack MERN (MongoDB, Express, React, Node.js). Cette application combine une interface utilisateur moderne et intuitive avec une architecture backend robuste, offrant ainsi une expérience utilisateur optimale pour les vétérinaires et leur personnel. Grâce à ses fonctionnalités complètes de gestion des propriétaires d'animaux, de suivi médical, et de planification des rendez-vous, l'application répond efficacement aux besoins quotidiens d'une clinique vétérinaire. L'architecture modulaire et évolutive du projet permet non seulement une maintenance facile mais aussi une expansion future des fonctionnalités. La sécurité des données, la validation des entrées, et la gestion des erreurs ont été particulièrement soignées pour garantir la fiabilité du système.

Si vous voulez voir et découvrir notre application voici le liens vers le compte GitHub pour une meilleure expérience :

https://github.com/Alae06/vetcareMongo

Merci.