

Rapport De Projet: PAWLY

Application de Gestion de Clinique Vétérinaire

Pawly

Projet:Application web de gestion vétérinaire

Réalisé par: Brahimi Douae & El Achhab Khadija

Encadré par: Mr. Esbai Redouane

Établissement : L'école Supérieure De Technologie Nador

Remerciement:

Je tiens à vous remercier sincèrement pour votre précieuse contribution à la réussite de notre projet **Pawly**.

Votre expertise, votre engagement et votre professionnalisme ont joué un rôle essentiel dans l'atteinte de nos objectifs.

Grâce à votre implication constante et à vos conseils toujours pertinents, nous avons pu avancer sereinement et obtenir des résultats concrets.

Encore un grand merci pour votre présence et votre soutien tout au long de ce parcours .

Table des matières:

1. Résumé Exécutif
2. Introduction
 - Contexte du Projet
 - Objectifs
 - Portée du Projet
3. Architecture Technique
 - Stack MERN
 - Technologies Utilisées
 - Schéma d'Architecture
4. Fonctionnalités Implémentées
 - Gestion des Propriétaires
 - Gestion des Animaux
 - Gestion des Visites
 - Liste des Vétérinaires
5. Interface Utilisateur
 - Structure de Navigation
 - Flux Utilisateurs
 - Pages Principales
6. Base de Données
 - Modèle de données
 - Relations entre Collections
7. Méthodologie de développement
 - Gestion de Versions
 - Organisation du Travail
8. Difficultés Rencontrés et Solutions
9. Conclusion et Perspectives d'évolution

Dans un contexte où la digitalisation devient un enjeu majeur pour tous les secteurs, le domaine vétérinaire n'échappe pas à cette transformation. Les cliniques vétérinaires, tout comme les hôpitaux ou les cabinets médicaux, ont besoin d'outils modernes pour assurer un suivi précis et structuré des animaux, améliorer la qualité de service, et gagner du temps dans la gestion quotidienne.

C'est dans cette optique que nous avons développé PAWLY, une application web dédiée à la gestion d'une clinique vétérinaire. Le projet est né d'une réflexion simple : comment faciliter le travail des vétérinaires et du personnel administratif tout en assurant un meilleur suivi des animaux et une meilleure organisation des données ?

Pawly permet de centraliser l'ensemble des informations liées aux patients animaux : leurs fiches personnelles, leurs antécédents médicaux, les propriétaires, les vétérinaires responsables et les différentes visites effectuées. Elle offre une interface claire, intuitive et accessible, même pour les utilisateurs peu familiers avec les outils numériques.

En plus d'être un projet académique, ce travail est pour nous une opportunité concrète de mettre en œuvre les compétences techniques acquises durant notre formation, tout en répondant à un besoin réel du terrain. Grâce à une architecture moderne basée sur la stack MERN, nous avons pu construire une application robuste, modulaire et facilement évolutive.

Dans ce rapport, nous présentons l'ensemble du processus de réalisation de l'application Pawly : depuis l'analyse du besoin initial jusqu'au déploiement, en passant par la conception technique, le développement, les tests et les perspectives d'évolution. Ce travail reflète notre implication, notre esprit d'équipe et notre volonté de proposer une solution utile, fonctionnelle et durable.

Contexte du Projet, Objectifs et Portée

Le domaine vétérinaire connaît aujourd'hui de nombreuses évolutions, tant sur le plan médical que technologique. Avec la diversification des espèces prises en charge, l'apparition de nouveaux traitements, et l'augmentation des attentes des propriétaires d'animaux, les cliniques vétérinaires doivent s'adapter à un environnement en constante mutation. Cette évolution ne se limite pas à l'aspect clinique : elle implique également une transformation profonde des outils et méthodes de gestion utilisés au quotidien.

Dans ce contexte, la rigueur administrative, la traçabilité des soins, et la disponibilité instantanée des données médicales deviennent des exigences incontournables. Les cliniques doivent assurer un suivi précis de chaque animal, organiser efficacement les rendez-vous, maintenir un historique fiable des visites, et coordonner les différents acteurs de l'établissement (vétérinaires, assistants, réceptionnistes). Cela nécessite des outils informatiques adaptés, capables de centraliser l'information, de fluidifier les échanges, et d'optimiser la gestion globale.

Pourtant, dans la pratique, beaucoup d'établissements vétérinaires continuent d'utiliser des méthodes traditionnelles, comme les registres papier, les tableurs Excel, ou des logiciels généralistes peu adaptés aux spécificités du métier. Ces solutions, bien que familières, révèlent rapidement leurs limites : perte de temps dans la recherche d'informations, duplication des saisies, difficultés à extraire des données pertinentes, et erreurs humaines fréquentes. Ce manque de modernisation impacte directement l'efficacité des équipes, la qualité du service fourni aux clients, et la qualité du suivi médical des patients animaux. Il peut également freiner la croissance de la clinique et nuire à la prise de décision stratégique.

Face à ces constats, il est apparu indispensable de proposer une solution numérique spécifique, pensée pour les besoins concrets du secteur vétérinaire. C'est dans ce cadre que nous avons imaginé **Pawly**, une application web de gestion vétérinaire simple, intuitive et évolutive. Le projet repose sur une analyse fine des attentes des utilisateurs, une structuration claire des fonctionnalités, et une implémentation technique solide reposant sur la stack MERN.

Le développement de Pawly s'inscrit également dans une logique pédagogique : il a été mené dans le cadre de notre formation à l'École Supérieure de Technologie de Nador, sous la supervision de M. Esbai Redouane. Il représente ainsi à la fois une réponse à un besoin

terrain réel, et une opportunité concrète de mettre en pratique nos compétences en développement web, en conception d'architecture logicielle et en gestion de projet collaboratif.

Objectifs

Le principal objectif de **Pawly** est de centraliser, structurer et moderniser la gestion des données au sein d'une clinique vétérinaire, tout en offrant un outil numérique accessible, performant et adapté aux besoins quotidiens des professionnels du secteur. À travers cette solution web, notre ambition est de proposer une réponse concrète aux difficultés rencontrées dans la gestion traditionnelle, en mettant la technologie au service de la simplicité et de l'efficacité.

Plus précisément, l'application vise à :

- **Assurer un suivi médical complet et personnalisé de chaque animal**, en centralisant des informations essentielles telles que le nom, l'espèce, la race, l'âge, le sexe, et surtout l'historique médical détaillé (visites, traitements, pathologies). Cette base de données permet une meilleure prise en charge des animaux et une continuité des soins sur le long terme.
- **Structurer la gestion des propriétaires**, en créant un lien explicite entre chaque animal et son détenteur. L'objectif est d'offrir une vue complète des relations entre les patients et leurs propriétaires, facilitant la communication, la facturation, et le suivi clientèle.
- **Faciliter l'enregistrement et la consultation des visites médicales**, avec un système simple d'ajout, de modification et d'historique des consultations, comprenant les dates, les motifs, les vétérinaires impliqués et les traitements administrés. Cela améliore la traçabilité des soins et permet aux vétérinaires de prendre des décisions éclairées.
- **Affiner la coordination interne de la clinique**, grâce à l'affichage et à la gestion de la liste des vétérinaires, de leurs spécialités et de leur disponibilité. Ce module permet, à terme, d'introduire une gestion de planning et de rendez-vous plus avancée.
- **Fournir une interface utilisateur intuitive et ergonomique**, pensée pour des profils variés, y compris les membres du personnel non techniques. La simplicité

d'utilisation est une priorité, afin de garantir une adoption rapide sans nécessiter de formation approfondie.

- **Proposer une architecture moderne, scalable et modulaire**, basée sur la stack MERN (MongoDB, Express.js, React.js, Node.js). Ce choix technologique permet non seulement de garantir de bonnes performances, mais aussi de faciliter les évolutions futures, telles que l'ajout d'un système d'authentification sécurisé, l'intégration d'un calendrier de rendez-vous, ou encore l'implémentation d'un tableau de bord statistique.

En parallèle de ces objectifs purement fonctionnels, **Pawly** est également un projet d'apprentissage ambitieux. Il a pour but de renforcer nos compétences professionnelles dans plusieurs domaines clés :

- La gestion et la planification de projet,
- Le travail collaboratif dans un environnement agile,
- La conception de bases de données relationnelles et documentaires,
- Le développement frontend et backend avec des technologies modernes,
- L'intégration d'API RESTful, et la réalisation de tests pour garantir la qualité du produit final.

Ce projet nous offre ainsi une expérience complète, mêlant rigueur technique, créativité, et sens de l'organisation.

Portée du Projet

Le périmètre fonctionnel de **Pawly** a été soigneusement défini afin de répondre de manière ciblée aux besoins essentiels d'une clinique vétérinaire de taille moyenne. Il s'agit de proposer une solution efficace sans complexité inutile, capable de soutenir les opérations quotidiennes tout en offrant une base technique extensible pour des évolutions futures.

La portée actuelle du projet inclut les fonctionnalités principales suivantes :

- **La gestion des propriétaires d'animaux**, avec la possibilité de créer, modifier, consulter et supprimer les fiches clients. Chaque fiche inclut des informations personnelles (nom, prénom, téléphone, email, adresse) et est reliée aux animaux

- **Mongoose** : Un ODM (Object Data Modeling) pour MongoDB qui facilite l'interaction avec la base de données en fournissant des modèles pour la validation des données et l'exécution de requêtes complexes.
- **Bootstrap** : Framework CSS utilisé pour rendre l'interface responsive et conviviale, s'adaptant à la fois aux écrans de bureau et mobiles.

? Schéma d'Architecture

L'architecture de Pawly repose sur une séparation claire des responsabilités entre le frontend, le backend et la base de données. Le frontend React interagit avec le backend Node.js via une API RESTful, qui, à son tour, se connecte à la base de données MongoDB pour récupérer ou manipuler les informations.

Architecture Simplifiée de Pawly

1. Frontend (React.js)

- a. **Composants React** : Interface utilisateur, gestion des vues et des interactions avec l'utilisateur.
- b. **Axios** : Effectue des requêtes HTTP vers l'API backend.

2. Backend (Node.js + Express.js)

- a. **Routes Express** : Point de contact pour les requêtes envoyées par le frontend.
- b. **Controllers** : Logique métier pour gérer les données.
- c. **Mongoose** : Permet l'interaction avec MongoDB.

3. Base de données (MongoDB)

- a. **Collections** : Stocke les données des propriétaires, animaux, vétérinaires, visites médicales.
- b. **Mongoose Models** : Définit les structures des données et les règles de validation.

4. Fonctionnalités Implémentées

L'application **Pawly** a été conçue pour faciliter la gestion quotidienne d'une clinique vétérinaire, en assurant le suivi des propriétaires, de leurs animaux et des visites médicales. Chaque fonctionnalité a été pensée pour offrir une utilisation simple, rapide et efficace.

Gestion des Propriétaires

Cette fonctionnalité permet d'ajouter, modifier, consulter et supprimer les informations des propriétaires d'animaux.

- **Ajout d'un propriétaire** via un formulaire dédié (nom, prénom, téléphone, email, adresse).
- **Affichage** d'une liste complète des propriétaires avec leurs coordonnées.
- **Consultation détaillée** des informations d'un propriétaire, incluant ses animaux.
- **Modification et suppression** des données en un clic.

Cette section facilite le suivi des clients de la clinique et assure un accès rapide aux informations nécessaires.

Gestion des Animaux

Les animaux sont associés à un propriétaire spécifique, ce qui permet de structurer les données de manière logique et efficace.

- **Ajout d'un animal** : nom, espèce (chien, chat, etc.), âge, race, sexe.
- **Association automatique** de l'animal à un propriétaire existant.
- **Consultation et modification** des détails de l'animal.
- **Suppression** possible si l'animal n'est plus suivi à la clinique.

 Chaque fiche propriétaire affiche dynamiquement les animaux qui lui sont rattachés.

Gestion des Visites

Un volet essentiel de l'application concerne le **suivi médical des animaux**.

- **Ajout d'une visite médicale** pour un animal donné : date, motif, traitement, vétérinaire responsable.
- **Historique complet** des visites disponible pour chaque animal.
- **Modification** des informations en cas de mise à jour.
- **Affichage organisé** par animal, facilitant la consultation chronologique.

Cette fonctionnalité joue un rôle crucial dans le suivi vétérinaire de chaque animal.

Liste des Vétérinaires

Cette partie permet d'afficher la liste des vétérinaires de la clinique.

- **Ajout manuel** dans la base de données ou via formulaire selon l'évolution future.
- **Affichage simple** : nom, spécialité, disponibilité (statique dans la version actuelle).

Ce module est une base pour un futur système de planification des consultations par vétérinaire..

5.Interface Utilisateur

L'interface utilisateur de l'application **Pawly** a été conçue avec un souci de clarté, de simplicité et d'efficacité. L'objectif était de garantir une prise en main rapide, même pour des utilisateurs non techniques, tout en offrant une expérience fluide et agréable.

Structure de Navigation

La navigation est organisée autour d'un **menu principal** accessible depuis toutes les pages de l'application. Ce menu permet de passer rapidement d'un module à l'autre :

- Accueil
- Liste des Propriétaires
- Liste des Animaux
- Visites Médicales
- Vétérinaires

Chaque page possède un en-tête clair et un bouton de retour ou d'ajout selon le contexte.

L'interface est **responsive** grâce à Bootstrap, ce qui la rend utilisable sur ordinateur, tablette et smartphone.

Flux Utilisateurs

Le parcours type d'un utilisateur dans l'application est pensé pour être logique et sans complexité :

1. **Ajout d'un propriétaire** depuis la section dédiée.
2. **Ajout d'un ou plusieurs animaux** liés à ce propriétaire.
3. **Ajout de visites médicales** pour chacun des animaux.
4. **Consultation ou modification** à tout moment des données saisies.

Le tout se fait via des formulaires simples, avec des champs bien identifiés et des boutons d'action clairs (Ajouter, Modifier, Supprimer).

Pages Principales

Voici un aperçu des principales pages présentes dans l'application :

Collections Principales

owners (Propriétaires)

Contient les informations des propriétaires: prénom, nom, téléphone, email, adresse.

🔗 Chaque propriétaire est lié à plusieurs animaux.

animals (Animaux)

Représente les animaux de compagnie : nom, type (chien/chat), âge, et ownerId (référence au propriétaire).

🔗 Un animal appartient à un seul propriétaire.

visits (Visites)

Historique médical : date, motif, traitement, vétérinaire, et animalId.

🔗 Une visite est liée à un seul animal.

veterinarians (Vétérinaires)

Informations de base sur les vétérinaires : nom et spécialité.

Actuellement utilisée pour l’affichage uniquement.

Relations

- Un **propriétaire** (owners) peut avoir plusieurs **animaux** (animals) grâce au champ ownerId.
- Un **animal** peut avoir plusieurs **visites médicales** (visits) via le champ animalId.
- Un **vétérinaire** est actuellement une entité indépendante, non directement reliée aux autres collections (mais mentionné dans les visites).

Solution : Utilisation des hooks React (useEffect et useState) pour recharger les données automatiquement après chaque opération.

Organisation du Code

Problème : L'augmentation des fichiers rendait la structure du projet difficile à suivre.

Solution : Réorganisation en modules clairs (pages, composants, services, etc.), ce qui a permis une meilleure lisibilité et maintenance.

9.Conclusion

Le projet **Pawly** a été une expérience riche sur les plans technique, organisationnel et personnel. Il nous a permis d'appliquer concrètement les compétences acquises durant notre formation, tout en découvrant les réalités du développement d'une application web complète.

À travers l'élaboration de cette solution de gestion pour une clinique vétérinaire, nous avons mis en œuvre les principes de l'architecture MERN (MongoDB, Express, React, Node.js), en assurant l'interaction fluide entre le frontend et le backend, ainsi qu'une modélisation claire des données.

