

1. Présentation générale

Le projet que nous avons entrepris consiste à développer une application permettant la gestion du Fichier National d'Identification des Carnivores Domestiques (I-CAD) en ligne. Cette initiative fait partie d'une mission confiée à la société Ingenium Animalis par le Ministère de l'Agriculture et de la Souveraineté Alimentaire, dans le but de faciliter la gestion et l'actualisation des informations relatives aux animaux domestiques identifiés par puce électronique ou tatouage. Le projet répond à la nécessité d'un outil de gestion centralisé et sécurisé pour les informations des animaux, accessible par des professionnels (vétérinaires, policiers, refuges, etc.), tout en offrant une interface simplifiée pour les utilisateurs non authentifiés pour déclarer un animal trouvé.

Le client, Ingenium Animalis, demande le développement d'une plateforme permettant aux professionnels d'enregistrer et de gérer les informations sur les animaux, avec des fonctionnalités d'authentification et de gestion des données. Ils exigent également la possibilité de déclarer des animaux retrouvés ou perdus, de fournir des informations détaillées sur chaque animal, et de maintenir un historique complet de toutes les actions relatives à un animal (création, perte, vol, retrouvée, etc.).

Nous, en tant que groupe de trois développeurs, avons pour rôle de répondre à ces exigences en concevant une solution robuste et scalable. Nous mettrons en œuvre les fonctionnalités étape par étape, conformément au cahier des charges, en intégrant des technologies modernes et une interface utilisateur intuitive. Les étapes initiales incluront la création d'un système de gestion d'animaux, la mise en place de la base de données et la gestion des utilisateurs authentifiés. Les fonctionnalités avancées seront développées progressivement, notamment la possibilité de télécharger des photos d'animaux et la gestion des déclarations d'animaux trouvés par des non-authentifiés.

2. Le plan et la maquette du site

Dans le dossier en format balsamiq

3. Dictionnaire de données/MCD

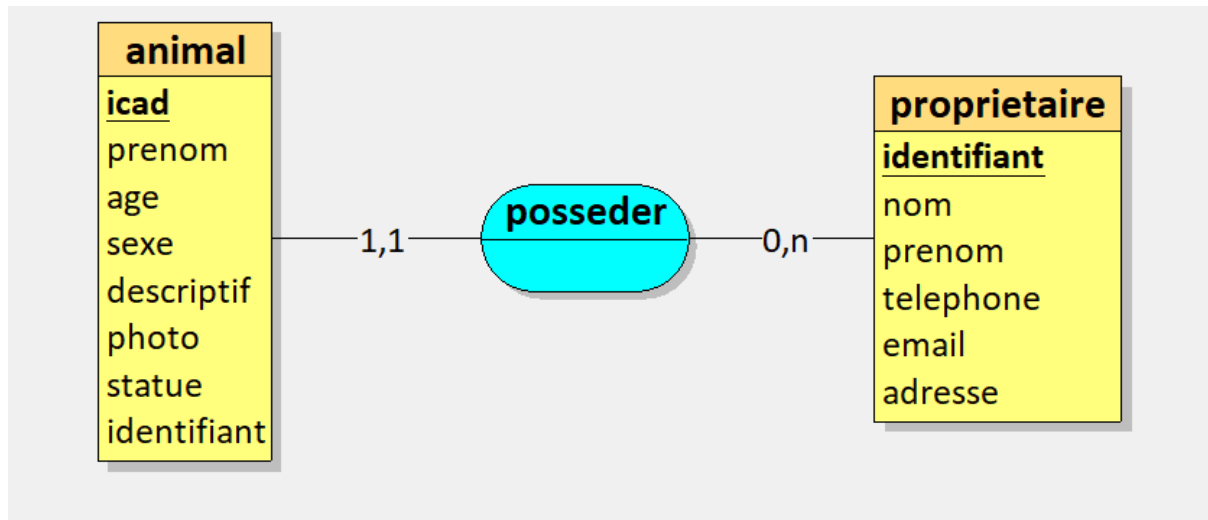
1) Animal

Attribut	Description	Type	Contraintes
ICAD	Identifiant unique de l'animal (puce ou tatouage)	INT(10)	PRIMARY KEY, NOT NULL
Nom	Prénom de l'animal	VARCHAR(50)	NULL
Age	Âge de l'animal	INT(11)	NULL
Sexe	Sexe de l'animal (Mâle/Femelle)	VARCHAR(10)	NULL
Descriptif	Descriptions particulières pour identifier l'animal (ex : signes distinctifs)	TEXT	NULL
Photo	Chemin vers la photo de l'animal (si disponible)	VARCHAR(255)	NULL
Statut	Statut de l'animal (par exemple, déclaré perdu, retrouvé, etc.)	INT	NOT NULL
Identifiant	L'ID de l'utilisateur auquel appartient l'animal	INT(11)	NOT NULL, FOREIGN KEY

2) Propriétaire

Attribut	Description	Type	Contraintes
Identifiant	Identifiant unique de l'utilisateur	INT(11)	PRIMARY KEY, AUTO_INCREMENT, NOT NULL
Nom	Nom du propriétaire	VARCHAR(50)	NULL
Prénom	Prénom du propriétaire	VARCHAR(50)	NULL
Email	Adresse email du propriétaire	VARCHAR(50)	NULL, UNIQUE
Téléphone	Numéro de téléphone du propriétaire	VARCHAR(15)	NULL
Adresse	Adresse complète du propriétaire	VARCHAR(150)	NULL

3) MCD



4. Descriptif Technique de l'Application

1) Technologies et Langages Utilisés

Pour le développement de l'application de gestion du Fichier National d'Identification des Carnivores Domestiques (I-CAD), nous avons choisi des technologies et des outils qui garantissent une efficacité maximale, une évolutivité et une sécurité optimale.

- **Langages de programmation :**
 - PHP : PHP sera utilisé côté serveur pour la gestion des requêtes, de l'authentification des utilisateurs et des interactions avec la base de données. Il est idéal pour le développement d'applications web dynamiques.
 - JavaScript : Utilisé pour ajouter de l'interactivité sur le côté client, notamment pour la gestion des formulaires, l'affichage dynamique des informations et la validation côté client.
 - HTML/CSS : HTML sera utilisé pour structurer les pages, tandis que CSS sera utilisé pour la mise en forme et l'adaptation de la page aux différents appareils (responsive design).
- **Base de données :**
 - MySQL : MySQL est utilisé pour stocker les informations sur les utilisateurs et les animaux, en raison de sa fiabilité, de sa rapidité, et de sa compatibilité avec les autres technologies utilisées.
 - PHPMyAdmin : Utilisé pour la gestion de la base de données et la manipulation des tables de manière graphique.
- **Outils de versioning :**

- Git : Le contrôle de version est géré via Git, permettant une collaboration fluide entre les membres du groupe et une gestion optimale des différentes versions du code. Nous utilisons GitHub pour l'hébergement du code source.
- **Environnement de développement :**
 - XAMPP : Un environnement de développement local basé sur Apache, MySQL et PHP, permettant de tester l'application en local avant de la déployer en production.

2) Organisation du Code

L'organisation du code suit une architecture MVC (Model-View-Controller), permettant une séparation claire des responsabilités :

- **Model (Modèle) :** La couche modèle contient la logique de gestion des données et la communication avec la base de données. Elle contient également les règles de validation des données, telles que la vérification de l'unicité des emails et de l'ICAD.
 - Exemple : Le modèle **propriétaire** interagit avec la base de données pour insérer, mettre à jour, ou supprimer des utilisateurs, et vérifiera si l'email est unique avant l'ajout.
 -
- **View (Vue) :** La vue est responsable de l'affichage des données à l'utilisateur. Elle est composée de fichiers HTML et CSS et contient le code pour afficher les formulaires, les listes d'animaux, et les pages d'administration. Bootstrap est utilisé pour faciliter la création de designs responsives.
 - Exemple : La vue de la page d'accueil affichera la liste des animaux d'un utilisateur authentifié, avec des options pour modifier ou supprimer un animal.
 -
- **Controller (Contrôleur) :** Le contrôleur fait le lien entre le modèle et la vue. Il gère les requêtes HTTP, appelle les modèles pour récupérer ou manipuler les données, et envoie les réponses appropriées à la vue.
 - Exemple : Le contrôleur **AnimalController** récupérera la liste des animaux d'un utilisateur et la passera à la vue correspondante.

3) Architecture de la Base de Données

La base de données est conçue pour être simple mais efficace, en reliant les propriétaires à leurs animaux via des clés étrangères. La table **propriétaire** contient des informations sur les utilisateurs, et la table **animal** contient des informations sur les animaux. Les tables sont liées par l'ID utilisateur, ce qui permet de facilement lister tous les animaux d'un propriétaire spécifique.

La base de données sera organisée de manière à faciliter les recherches rapides et les mises à jour des informations sur les animaux. Des index seront ajoutés sur les champs fréquemment utilisés, tels que l'email du propriétaire et l'ICAD de l'animal.

4) Fonctionnalités Clés

Les principales fonctionnalités de l'application sont les suivantes :

- **Gestion des utilisateurs** : Les professionnels peuvent se créer un compte et s'authentifier.
- **Gestion des animaux** : Les propriétaires authentifiés peuvent déclarer de nouveaux animaux, les modifier ou les supprimer.
- **Déclarations supplémentaires** : Les utilisateurs peuvent déclarer la perte, le vol, ou la récupération d'un animal. Cette fonctionnalité sera également accessible à tout internaute pour déclarer un animal trouvé.

5) Sécurité

- **Protection des données personnelles** : L'application respecte les principes de sécurité des données personnelles.
- **Validation des entrées** : Toutes les entrées de l'utilisateur (par exemple, les informations sur l'animal) sont validées côté serveur pour éviter les injections SQL et garantir que les données sont correctes.

5. Déploiement et Installation

- Préparer l'environnement :

Ce document guide l'installation et la configuration d'un serveur pour héberger une application CodeIgniter. Les étapes incluent la configuration de base du serveur, l'installation de la pile LAMP, et la mise en place de l'application web.

- Étape 1 : Configuration de Base du Serveur Debian 12

- **Accès à l'interface Proxmox :**
 - URL : <https://10.0.112.200:8006/>
 - Modifier le mot de passe Proxmox.
- **Connexion au conteneur :**
 - Connectez-vous en tant que `root`.
 - Changez le mot de passe Linux : `passwd`
- **Création d'un utilisateur personnel :**
 - Pour une gestion sécurisée via SSH et Composer :
`adduser mboyer`
- **Mise à jour du système :**
`apt update && apt upgrade`

- Étape 2 : Installation et Configuration de LAMP

- **Installation d'Apache2 :**
 - Installer Apache : `apt install apache2`
 - Vérifier le statut : `systemctl status apache2`
 - Créer un dossier pour le projet :
`mkdir /var/www/projets_CI`
`chown mboyer /var/www/projets_CI`
`chmod 774 /var/www/projets_CI`
- **Installation de PHP**
 - Installer PHP : `apt install php`
 - Vérifier la version : `php -v`
- **Installation de MariaDB**
 - Installer MariaDB : `apt install mariadb-server`
 - Autoriser les connexions distantes :
 - Modifier `/etc/mysql/mariadb.conf.d/50-server.cnf`
 - Remplacer `bind-address` par `0.0.0.0`.
 - Créer un utilisateur :
 - `CREATE USER 'ICAD_MANAGER'@'%' IDENTIFIED BY 'ICAD_PASSWORD';`

- GRANT ALL PRIVILEGES ON *.* TO 'ICAD_MANAGER'@'%';
- FLUSH PRIVILEGES;

- **Installation de PHPMyAdmin**

- Installer PHPMyAdmin et extensions PHP : `apt install phpmyadmin php-zip php-gd php-json php-curl libapache2-mod-php`
- Activer la configuration Apache :
`In -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-available/phpmyadmin.conf a2enconf phpmyadmin.conf systemctl reload apache2`

- **Étape 3 : Installation de l'Application Web**

- **Installer Git et Composer**

- Installer Git : `apt install git-all`
- Installer Composer dans le dossier du projet :
 - `cd /var/www/projets_CI`
 - `php -r "eval('?'>'.file_get_contents('https://getcomposer.org/installer'))';"`
 - `mv composer.phar /usr/local/bin/composer`
 - `composer --version`

- **Cloner le dépôt GitHub**

- Dans `/var/www/projets_CI` :
 - `git clone https://github.com/temigame2/icad.git`

- **Installer Codelgniter**

- Installer `php-intl` :
 - `apt install php-intl`
- Activer les extensions nécessaires dans `php.ini` et redémarrer Apache :
 - `systemctl restart apache2`
- Installer les composants manquants :
 - `composer update`
 - `chmod 777 -R writable`

- **Étape 4 : Configuration du VirtualHost :**

- **Créer un fichier VirtualHost :**

- `nano /etc/apache2/sites-available/monappli.conf`

Contenu :

```
<VirtualHost *:80>
    ServerName icad.web
    DocumentRoot "/var/www/projets_CI/icad/public"
    <Directory "/var/www/projets_CI/icad/public">
        AllowOverride All
```

</Directory>

</VirtualHost>

- Activer le VirtualHost :
 - a2ensite monappli.conf
 - a2enmod rewrite
 - systemctl restart apache2
- Ajouter une entrée dans le fichier hosts local :
 - <adresse_ip> www.monappli.web

6. Guide d'utilisation

Liste des animaux

Ajouter un animal

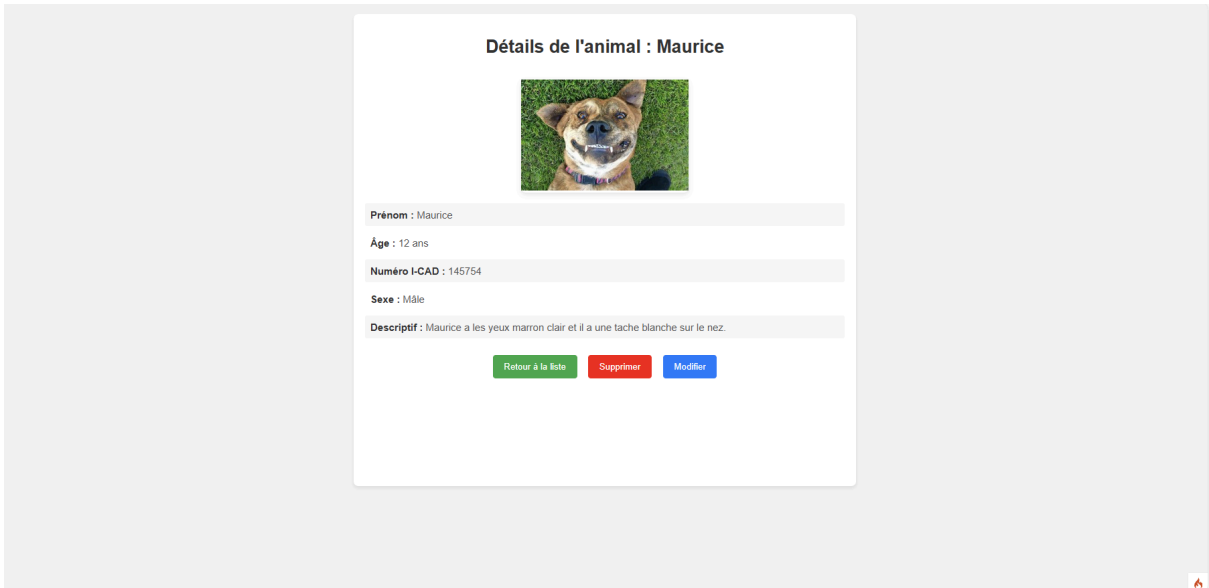
Ajouter un propriétaire

Rechercher un animal...

PHOTO	PRÉNOM DE L'ANIMAL	ÂGE	NUMÉRO I-CAD	SEXE	DESSCRIPTIF	TÉLÉPHONE	EMAIL	ACTIONS
	Maurice	12 ans	145754	Mâle	Maurice a les yeux marron clair et il a une tache blanche sur le nez.	0761832007	illan-pro@outlook.fr	Voir les détails
	Bella	6 ans	145760	Femelle	Bella a un pelage doré brillant et porte un collier rouge avec une médaille en forme d'os.	0761832007	dylanbalmigere@gmail.com	Voir les détails
	Urus	9 ans	145761	Mâle	Urus est un chat au pelage marron tigré et a une petite cicatrice sur l'oreille droite permet de l'identifier facilement.	0761832007	dylanbalmigere@gmail.com	Voir les détails

Nous voici sur la page d'accueil sur laquelle vous pouvez retrouver :

- un bouton "Ajouter un animal" pour ajouter un animal,
- un bouton "Ajouter un propriétaire" pour ajouter un propriétaire,
- une barre de recherche lié avec le prénom de l'animal,
- la liste des animaux présent sur notre site,
- une photo de l'animal, son prénom, son âge, son numéro ICAD, son sexe, une description détaillé,
- le numéro de téléphone du propriétaire, son email,
- bouton "Voir les détails" qui permet d'afficher les détails de l'animal sélectionné.



Nous voici sur la page des détails d'un animal sur laquelle vous pouvez retrouver :

- une photo de l'animal, son prénom, son âge, son numéro ICAD, son sexe, une description détaillé,
- un bouton "Retour à la liste" pour revenir à la page d'accueil
- un bouton "Supprimer" pour supprimer l'animal de la liste
- un bouton "Modifier" pour modifier les informations de l'animal

Modifier l'animal : Maurice

Prénom

Maurice

Âge

12

Sexe

Mâle

Statut

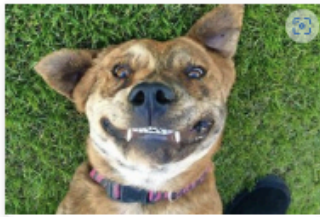
Retrouvé

Descriptif

Maurice a les yeux marrons clairs et il a une tache blanche sur le nez

Photo (optionnelle)

Choisir un fichier Aucun fichier n'a été sélectionné



Sauvegarder les modifications

Annuler

On se retrouve sur la page de la modification d'un animal sur laquelle vous pouvez retrouver :

- une photo de l'animal, son prénom, son âge, son numéro ICAD, son sexe, une description détaillée que l'on peut modifier ainsi que son statut
- un bouton "Sauvegarder les modifications" pour enregistrer les modifications faites sur un animal
- un bouton "Annuler" pour revenir sur la page des détails de l'animal

Ajouter un animal

Photo de l'animal:

Aucun fichier n'a été sélectionné

Prénom:

Âge:

Sexe:

Descriptif:

Statut:

Identifiant propriétaire:

On enchaîne avec la page d'ajout d'un animal sur laquelle vous pouvez retrouver :

- des champs pour ajouter une photo de l'animal, son prénom, son âge,, son sexe, une description détaillé et son statue
- un bouton "Ajouter l'animal" pour ajouter l'animal à la liste

Ajouter un propriétaire

Identifiant :

Nom :

Prénom :

Téléphone :

Email :

Adresse :

Et on finit avec la page d'ajout d'un propriétaire sur laquelle vous pouvez retrouver :

- des champs pour ajouter le nom du propriétaire, son prénom, son numéro de téléphone, son adresse email, son adresse de localisation ainsi que l'identifiant qui lui est attribué par défaut
- un bouton "Ajouter le propriétaire" pour ajouter le propriétaire a la base de données

7. Répartition des Tâches dans le Projet

- **Dylan :**
 - MCD

- Organisation des pages
- Création des pages (HTML / CSS / PHP)
- Liaison Base de données
- Contrôleur et modèle

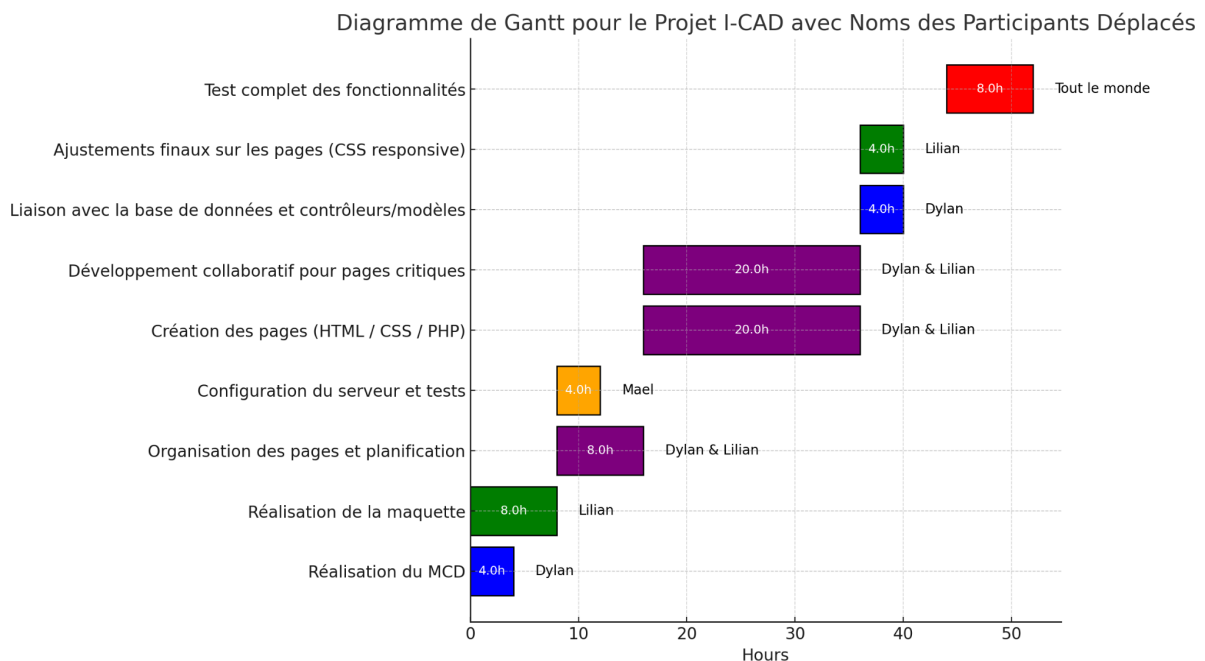
- **Lilian :**

- Maquette
- Organisation des pages
- Création des pages (HTML / CSS / PHP)

- **Mael :**

- Serveur web
- Mise en place des environnements (Github)

8. Diagramme de Gantt



9. Les difficultés rencontrées, les approches envisagées, les solutions choisies

1. Gestion des relations en base de données

- **Difficulté rencontrée :**
La structuration de la base de données, avec des relations entre utilisateurs et animaux, a nécessité une attention particulière, notamment pour gérer correctement les clés primaires et étrangères. Le lien entre chaque animal et son propriétaire (via l'ID utilisateur) devait être géré de manière fluide pour permettre une utilisation cohérente.
- **Approche envisagée :**
Étudier plusieurs architectures de base de données possibles et tester les relations entre les tables **Propriétaire** et **Animal** avec des clés étrangères et des contraintes d'intégrité. Utiliser des jointures pour récupérer facilement les données liées.
- **Solution choisie :**
Utilisation des clés primaires et étrangères pour relier les utilisateurs aux animaux. Chaque animal étant associé à un seul utilisateur via l'ID, cela a permis une gestion simple et claire des données. La structure relationnelle a été optimisée pour assurer une bonne performance des requêtes SQL.

2. Sécurisation de l'accès et de la gestion des utilisateurs

- **Difficulté rencontrée :**
Assurer une authentification sécurisée pour les utilisateurs (professionnels) tout en permettant un accès limité à certaines pages pour la création d'utilisateurs en réseau local. De plus, le défi résidait dans la gestion de l'authentification tout en évitant les vulnérabilités liées à la gestion des sessions.
- **Approche envisagée :**
Implémentation d'un système d'authentification basé sur des sessions, avec des restrictions d'accès spécifiques selon le rôle de l'utilisateur. Prévoir une interface de gestion pour ajouter des utilisateurs uniquement en réseau local via une page protégée par des contrôles d'IP.
- **Solution choisie :**
Mise en place d'un **système d'authentification basé sur les sessions** dans Codelgniter. La création de nouveaux utilisateurs se fait via une **page sécurisée** en réseau local, avec un contrôle d'accès basé sur l'adresse IP. L'authentification des utilisateurs a été intégrée en toute sécurité grâce à l'utilisation de sessions cryptées et la gestion des rôles via des middleware.

3. Gestion des déclarations des animaux (perte, vol, retrouvé)

- **Difficulté rencontrée :**
Le défi résidait dans le **suivi et l'historisation** des déclarations faites sur

chaque animal (perte, vol, retrouvé), tout en permettant à différents types d'utilisateurs d'effectuer ces actions. Il fallait aussi s'assurer de l'intégrité des données et éviter toute confusion dans les différents types de déclarations.

- **Approche envisagée :**

Créer une **table historique** pour enregistrer toutes les actions effectuées sur chaque animal. Permettre aux utilisateurs de consulter l'historique des actions pour chaque animal dans un format clair et structuré. Ajouter un contrôle supplémentaire pour les déclarations sans authentification (animaux retrouvés par des utilisateurs non-authentifiés).

- **Solution choisie :**

Implémentation d'une **table historique** qui enregistre les différentes actions sur un animal avec des champs permettant de spécifier le type d'action (création, perte, vol, retrouvé), la date, et l'auteur de l'action. Cette table est associée à chaque animal, permettant une traçabilité complète. Les utilisateurs non authentifiés peuvent également déclarer un animal retrouvé, mais uniquement après avoir vérifié l'identifiant ICAD.

4. Gestion des images des animaux

- **Difficulté rencontrée :**

Permettre l'ajout d'**images** pour chaque animal tout en garantissant la **sécurisation des fichiers** et la **compatibilité mobile**. Il fallait également gérer les contraintes de taille des fichiers pour éviter la surcharge du serveur.

- **Approche envisagée :**

Utiliser une **bibliothèque d'upload sécurisée** pour les images. Limiter la taille des images et leur format (par exemple, accepter uniquement les formats JPEG et PNG). Tester l'affichage des images sur différents appareils pour garantir un affichage responsive.

- **Solution choisie :**

Mise en place d'un système d'upload d'images avec des restrictions sur la taille et le format (JPEG, PNG), via le **framework CodeIgniter**. Un **système de compression des images** a été intégré pour réduire l'impact sur les performances.

10. Conclusion

Ce projet I-CAD a été une expérience précieuse pour nous. Il nous a permis de renforcer nos compétences techniques, en particulier avec **CodeIgniter**, la gestion des bases de données et l'authentification des utilisateurs. Travailler ensemble nous a appris l'importance du **travail en équipe**, de la communication et de l'adaptabilité face aux imprévus. Nous avons développé une meilleure compréhension des défis liés à la création d'applications sécurisées et à la gestion de données sensibles. Ce

projet nous a préparés à relever de nouveaux défis et à aborder des projets plus complexes dans le futur.