# Introduction

A cartoon raccoon with text

Description automatically generated

Problématique : Matchmaking pour Animaux de Compagnie : Une plateforme de rencontre pour animaux de compagnie où les propriétaires peuvent trouver des amis de jeu ou des partenaires pour leurs animaux. Cela pourrait inclure des événements locaux et des recommandations basées sur la compatibilité des animaux.

# Sommaire

[Introduction 1](#_Toc169210251)

[Sommaire 1](#_Toc169210252)

[Charte de projet 1](#_Toc169210253)

[Liste des fonctionnalitées 3](#_Toc169210254)

[Description des technologies choisies 3](#_Toc169210255)

[Choix palettes avec Colors 4](#_Toc169210256)

[Design MCD de la Database 5](#_Toc169210257)

[Design figma du front-end 5](#_Toc169210258)

[Organisation du Trello ( Rendu final ) 6](#_Toc169210259)

[Journal de bord 6](#_Toc169210260)

# Charte de projet

Nom de projet

Pedro’s Pals

A cartoon raccoon with text

Description automatically generated

Chef de Projet

FRIKICH Yassine

Equipe

DABO Liliane

But

Un site web, une API et une infrastructure qui réponde aux compétences visées du projet. Afin d'avoir un projet aussi complet que possible, qui a déjà été réalisé auparavant pour certains points.

Les fonctionnalités minimums pour que le projet soit viable sont donc :

- Une page d'accueil

- Un système de like et de commentaires reposant sur une database appelée par une API

- Un style correspondant au thème choisi

- L'utilisation d'une infrastructure en arrière-plan ( soit avec VMWare soit avec Docker ) qui permettent aux différents services utilisés de communiqués entre eux

Contexte

Le projet est inscrit dans un contexte de fin de notre deuxième année de Bachelor. Il représente donc une évaluation des compétences acquises tout au long de notre parcours, ainsi que l'opportunité de rajouter des compétences en plus à notre portfolio.

Rôles

Yassine et Liliane travailleront à la fois sur la partie infrastructure que sur la partie développement. Le but étant de se répartir les tâches et non les domaines du projet entre nous deux.

Responsabilités

Chaque activité est donc sous la responsabilisée de nos deux rôles, sauf indication contraire.

Dates de déroulement

02/05 au 18/05 : Recherche d'idée de développement ( + Nom du projet ); Redaction de la charte de projet ; Création du Git

18/05 au 01/06 : Mise en place de l'infrastructure

01/06 au 13/06 : Développement de l'application web, et préparation à la soutenance

14/06 : Soutenance

# Liste des fonctionnalitées

1. Flux de profil d’animaux de companie
2. Likes et commentaire
3. Matching entre profil
4. Messages envoyés / reçus

# Description des technologies choisies

1. ARCHITECTURE : Docker
2. BACK-END : ( API ) Python, ( Framework web ) Django, ( Database ) PostgreSQL
3. FRONT-END : HTML, ( Framework CSS ) Bootstrap

# Choix palettes avec Colors

A screenshot of a website

Description automatically generated

A screenshot of a computer

Description automatically generated

# Design MCD de la Database

A screenshot of a computer

Description automatically generated

# Design figma du front-end

A screenshot of a chat

Description automatically generated

( Voir figma pdf )

# Organisation du Trello ( Rendu final )

A screenshot of a computer

Description automatically generated

Lien : <https://trello.com/invite/b/GMqZnc92/ATTIe36fbc10d9587fa86694458689289e6bBAB7D431/pedros-pals>

# Journal de bord

Journal de Bord - Développement du Projet Pedro's Pals

Introduction

Ce journal de bord détaille toutes les étapes, les erreurs rencontrées, et les fonctionnalités ajoutées lors du développement du projet Pedro's Pals. Le projet a évolué à travers différentes phases, chaque phase apportant son lot de défis et de solutions.

1. Initial Setup and Configuration ( Yassine et Liliane )

Date: 02/05 – 01/06

Action:

Le projet a débuté par la configuration initiale des environnements de développement et de production. Nous avons choisi d'utiliser Docker pour garantir une configuration homogène entre les différentes machines de développement et de production. Le fichier docker-compose.yml a été configuré pour orchestrer les différents conteneurs nécessaires au projet, incluant la base de données PostgreSQL, l'application Django, et le frontend React.

Error:

Nous avons rencontré plusieurs problèmes avec les versions des images Docker et la configuration des volumes. En particulier, le message "version is obsolete" a indiqué que nous utilisions une ancienne syntaxe de version dans notre fichier Docker Compose.

Resolution:

La mise à jour du fichier docker-compose.yml pour utiliser la syntaxe de version actuelle a été nécessaire. Nous avons également corrigé les erreurs de montage des volumes pour garantir que les changements de code soient reflétés en temps réel dans les conteneurs.

2. Création de la Base de Données et des Modèles

Date: 02/05 – 16/05

Action:

Une fois l'environnement configuré, nous avons créé le modèle PetProfile pour stocker les informations sur les animaux de compagnie. Ce modèle incluait des champs pour le propriétaire, le nom, l'âge, l'espèce, la race, la description et une photo.

Error:

Les migrations initiales ont rencontré des erreurs dues à des problèmes de configuration de la base de données PostgreSQL. Les erreurs incluaient des problèmes de connexion à la base de données et des conflits de migration.

Resolution:

Nous avons corrigé ces problèmes en réinitialisant la base de données et en vérifiant les configurations dans le fichier settings.py. Nous avons également recréé les migrations pour garantir qu'elles soient correctement appliquées.

3. Mise en Place des Vues et des URLS

Date: 16/05 – 01/06

Action:

Nous avons mis en place plusieurs vues pour gérer les profils d'animaux. Ces vues incluaient:

Index View: Liste tous les profils d'animaux et les affiche dans une vue de liste.

Detail View: Affiche les détails d'un animal spécifique, incluant toutes les informations du profil.

Create View: Permet aux utilisateurs connectés de créer un nouveau profil d'animal via un formulaire.

Search View: Fournit une fonctionnalité de recherche pour trouver des animaux en fonction de critères comme le nom, l'espèce, ou la race.

Error:

Nous avons rencontré des problèmes avec les URL et les noms de routes, causant des erreurs 404 et des conflits de route.

Resolution:

La structure des URLs a été revue et corrigée pour garantir que chaque vue ait une route unique et accessible. Des tests ont été effectués pour s'assurer que toutes les routes fonctionnent correctement.

4. Gestion de l'Authentification

Date: 16/05 – 01/06

Action:

Nous avons implémenté le système d'authentification des utilisateurs. Les vues de connexion, d'inscription et de déconnexion ont été créées pour permettre aux utilisateurs de gérer leur accès.

Error:

Les principaux problèmes rencontrés incluaient des erreurs de validation de formulaire et des redirections incorrectes après l'inscription ou la connexion.

Resolution:

Les formulaires ont été améliorés pour fournir des messages d'erreur plus clairs aux utilisateurs. Les redirections après l'authentification ont été corrigées pour diriger les utilisateurs vers la page d'accueil ou la page précédente.

5. Ajout de l'API REST

Date: 01/06 – 05/06

Action:

Pour offrir une interface programmatique, nous avons ajouté une API REST en utilisant Django REST Framework. Cette API permet de gérer les profils d'animaux via des endpoints sécurisés.

Details:

Endpoints:

PetProfileListCreate: Pour lister et créer des profils d'animaux.

PetProfileDetail: Pour récupérer, mettre à jour, et supprimer un profil spécifique.

Permissions:

L'API est sécurisée et nécessite une authentification pour effectuer des opérations de modification.

Error:

Des problèmes de configuration de CORS et de permissions d'API ont été rencontrés, empêchant l'accès correct à l'API depuis le frontend.

Resolution:

Nous avons configuré les en-têtes CORS et ajusté les permissions pour permettre l'accès sécurisé à l'API. Des tests ont été effectués pour s'assurer que l'API fonctionne correctement avec le frontend.

6. Intégration du Frontend

Date: 05/06 – 09/06

Action:

Le frontend a été développé en utilisant React. create-react-app a été utilisé pour configurer l'application frontend initiale. Des composants React ont été créés pour afficher les profils d'animaux et interagir avec l'API.

Error:

Nous avons rencontré des erreurs de compilation dues à des incompatibilités de versions des paquets npm et des dépendances manquantes.

Resolution:

Les versions des paquets ont été ajustées et les dépendances nécessaires ont été ajoutées. Des configurations supplémentaires ont été mises en place pour garantir une compilation sans erreur.

7. Mise en Place de la Fonctionnalité de Swipe

Date: 09/06 – 12/06

Action:

Une fonctionnalité de swipe inspirée de Tinder a été ajoutée pour permettre aux utilisateurs de parcourir les profils d'animaux de manière interactive.

Details:

Vue Swipe: Création d'une nouvelle vue pour afficher les profils avec la fonctionnalité de swipe.

Template Swipe: Un nouveau template a été conçu pour intégrer l'interface de swipe.

Bibliothèque Utilisée: swiper.js a été choisi pour gérer l'interaction de swipe.

Error:

Des problèmes d'intégration de la bibliothèque de swipe ont été rencontrés, notamment avec le chargement des images et l'affichage des profils.

Resolution:

Les chemins d'accès aux images ont été corrigés et la bibliothèque Swiper a été correctement intégrée dans le projet. Des tests utilisateurs ont été effectués pour s'assurer de la fluidité de l'expérience de swipe.

Conclusion

Le projet Pedro's Pals a progressé de manière significative grâce à une série de développements itératifs et de résolutions de problèmes. Chaque étape a été marquée par des défis techniques et des corrections qui ont amélioré la robustesse et l'expérience utilisateur de l'application. Les fonctionnalités clés comme l'authentification, la gestion des profils d'animaux, l'API REST, et la fonctionnalité de swipe offrent une plateforme complète et interactive pour les utilisateurs.

Si vous avez besoin de plus de détails sur une section particulière ou souhaitez ajouter d'autres informations, n'hésitez pas à me le faire savoir.