

Cahier des charges

Projet Arcadia

2024/2025

1. Contexte du Projet

Le Zoo Arcadia, situé près de la forêt de Brocéliande, souhaite développer une application web pour améliorer son image de marque, refléter ses valeurs écologiques et proposer des fonctionnalités interactives aux visiteurs et aux employés.

L'application devra répondre aux besoins des visiteurs, du personnel (employés, vétérinaires) et de l'administrateur, en assurant une gestion optimisée des données et en mettant en avant les habitats et animaux du zoo.

2. Objectifs

1. Créer une plateforme intuitive et accessible.
2. Mettre en avant les valeurs écologiques du zoo dans le design et le thème.
3. Fournir une interface responsive adaptée aux différents rôles des utilisateurs :
 - a. Visiteur : Consultation des animaux, habitats, horaires et services.
 - b. Employé : Validation des avis, gestion des services, alimentation des animaux.
 - c. Vétérinaire : Suivi de l'état des animaux et des habitats.
 - d. Administrateur : Gestion globale de l'application et des utilisateurs.

3. Fonctionnalités

3.1. Fonctionnalités pour les Visiteurs

- **Page d'accueil** : Présentation du zoo, des habitats, des animaux, des services, et des avis validés.
- **Menu interactif** : Navigation vers les habitats, services, page de connexion, et formulaire de contact.
- **Vue des habitats** : Liste des habitats avec détails des animaux et rapport des vétérinaires.
- **Avis** : Possibilité de soumettre un commentaire (pseudo et avis texte).

3.2. Fonctionnalités pour les Employés

- **Validation des avis** : Gestion des avis soumis par les visiteurs.
- **Gestion des services** : Ajout, modification ou suppression de services.
- **Suivi alimentaire** : Enregistrement de l'alimentation donnée aux animaux (type, quantité, date, heure).

3.3. Fonctionnalités pour les Vétérinaires

- **Gestion des rapports** : Suivi de l'état des animaux avec possibilité d'ajouter des observations.
- **Avis sur les habitats** : Évaluation des conditions et recommandations d'amélioration.
- **Consultation alimentaire** : Visualisation de l'historique alimentaire des animaux.

3.4. Fonctionnalités pour l'Administrateur

- **Gestion des utilisateurs** : Création de comptes pour les employés et vétérinaires.
- **Gestion globale** : Modification des données sur les services, horaires, habitats et animaux.
- **Tableau de bord** : Consultation des statistiques d'interactions (ex. consultation des animaux).

Contraintes Techniques

- **Technologies obligatoires** :
 - Base de données relationnelle (MySQL, PostgreSQL, etc.) et non relationnelle (MongoDB).
 - Déploiement via Fly.io, Heroku, ou équivalent.
- **Stack recommandée** :
 - **Frontend** : HTML5, CSS (Bootstrap), JavaScript.
 - **Backend** : PHP (avec PDO ou framework Symfony).
- **Responsive design** pour les appareils mobiles et desktop.

5. Sécurité

- Gestion sécurisée des mots de passe (hashing).
- Authentification stricte pour les rôles (employé, vétérinaire, administrateur).
- Validation des données soumises (ex : avis des visiteurs) pour éviter les abus.

6. Livrables

1. **Code source :**
 - a. Hébergé sur un dépôt GitHub public avec fichier README.md.
 - b. Documentation sur le déploiement et la configuration locale.
2. **Maquettes :**
 - a. Wireframes et mockups (3 versions desktop et 3 versions mobiles).
3. **Charte graphique :**
 - a. Palette de couleurs et police d'écriture.
4. **Manuel d'utilisation :**
 - a. Présentation des fonctionnalités et comptes tests.
5. **Documentation technique :**
 - a. Diagrammes de données, de classes, et de séquence.
 - b. Justifications des choix techniques.

7. Gestion de Projet

- Utilisation d'un outil Kanban (Trello, Jira, etc.) avec les colonnes suivantes :
 - Fonctionnalités prévues.
 - Fonctionnalités en cours.
 - Fonctionnalités terminées sur branche de développement.
 - Fonctionnalités fusionnées sur branche principale.

8. Délais

- **Durée indicative :** 70 heures.

- **Livraison finale** : Respecter le format demandé et soumettre les fichiers dans l'espace de dépôt prévu.