

DOCUMENTATION TECHNIQUE

1. Réflexions Initiales Technologiques sur le Sujet

Le projet consiste à développer une application web pour le zoo Arcadia, qui permettra aux visiteurs de visualiser les animaux, leur état de santé, les services proposés par le zoo, et de consulter les horaires d'ouverture. Le système doit également gérer des rôles utilisateurs (administrateur, vétérinaire, employé) avec des fonctionnalités spécifiques pour chacun.

Choix des technologies :

- **Backend** : Utilisation de **Symfony** et **API Platform** en PHP. Symfony est un framework robuste et bien structuré qui facilite le développement d'applications web avec une architecture MVC. API Platform, basé sur Symfony, est idéal pour construire rapidement des API RESTful.
- **Base de données relationnelle** : **MySQL** pour stocker les données structurées telles que les informations sur les animaux, utilisateurs, et services.
- **Base de données NoSQL** : MongoDB pour stocker des données non structurées comme les statistiques de consultation des animaux.
- **Frontend** : Utilisation de **HTML**, **CSS**, **JavaScript** et **Bootstrap** pour développer une interface utilisateur responsive et intuitive. Bootstrap simplifie le développement d'interfaces modernes et responsives.
- **Routing en JS** : Utilisation de JavaScript pour gérer le routage client-side, permettant une navigation fluide et une meilleure expérience utilisateur.

2. Configuration de Votre Environnement de Travail

Pour configurer votre environnement de travail, suivez ces étapes :

1. Installer PHP et Composer :

- Téléchargez et installez PHP (version 8.0 ou supérieure recommandée).
- Installez Composer, le gestionnaire de dépendances PHP.

2. Configurer Symfony :

- Installez le Symfony CLI en utilisant Composer : `composer global require symfony/cli``.
- Créez un nouveau projet Symfony : `symfony new arcadia_zoo --webapp``.

3. Installer API Platform :

- Ajoutez API Platform à votre projet Symfony : `composer require api``.

4. Configurer la Base de Données MySQL :

- Installez MySQL et créez une base de données pour le projet.
- Configurez le fichier `.env` de Symfony pour la connexion à la base de données :

```
```env
DATABASE_URL="mysql://db_user:db_password@127.0.0.1:3306/arcadia_zoo"
```
```

5. Installer MongoDB :

- Téléchargez et installez MongoDB.
- Ajoutez le bundle Doctrine MongoDB ODM à Symfony : ``composer require doctrine/mongodb-odm-bundle``.

6. Frontend :

- Assurez-vous d'avoir `**Node.js**` et `**npm**` installés pour gérer les dépendances front-end.
- Installez Bootstrap et d'autres dépendances via npm :

```
```bash
npm install bootstrap
```
```

7. Configurer l'Environnement de Développement :

- Utilisez un IDE comme `**PhpStorm**` ou `**VS Code**` avec les plugins appropriés pour Symfony, PHP, et JavaScript.
- Configurez Git pour le contrôle de version et créez un dépôt GitHub public pour héberger le code source.

3. Modèle Conceptuel de Données (Diagramme de Classe)

Le modèle conceptuel de données doit capturer les principales entités et leurs relations. Voici une description simplifiée :

- `**Utilisateur**` : ``id``, ``email``, ``password``, ``role``
- `**Animal**` : ``id``, ``prenom``, ``race``, ``images``, ``habitat_id``
- `**Habitat**` : ``id``, ``nom``, ``description``, ``images``
- `**Service**` : ``id``, ``nom``, ``description``
- `**Avis**` : ``id``, ``pseudo``, ``avis``, ``validation_status``
- `**CompteRenduVeterinaire**` : ``id``, ``animal_id``, ``etat``, ``nourriture``, ``grammage``, ``date_passage``, ``details_etat``

Ces entités et relations peuvent être modélisées à l'aide de Doctrine ORM pour MySQL et Doctrine ODM pour MongoDB.

4. Diagramme d'Utilisation et Diagramme de Séquence

Diagramme d'Utilisation

Les diagrammes d'utilisation décrivent les principales fonctionnalités de l'application :

- `**Visiteur**` :
 - Voir les habitats et animaux

- Laisser un avis
- Consulter les services et horaires
- Contacter le zoo
- ****Administrateur**** :
 - Gérer les utilisateurs (employés et vétérinaires)
 - Gérer les services, habitats, et animaux
 - Visualiser les statistiques des consultations
- ****Vétérinaire**** :
 - Remplir les comptes rendus des animaux
 - Commenter sur les habitats
- ****Employé**** :
 - Valider ou invalider les avis des visiteurs
 - Gérer la nourriture des animaux

Diagramme de Séquence

Les diagrammes de séquence illustrent les interactions entre les utilisateurs et le système pour différentes fonctionnalités. Par exemple, pour "Voir les habitats et animaux", le diagramme de séquence pourrait ressembler à ceci :

1. ****Visiteur**** envoie une requête pour voir les habitats.
2. ****Application**** utilise le contrôleur pour récupérer les informations des habitats depuis la base de données.
3. ****Application**** renvoie la réponse au visiteur avec la liste des habitats.
4. ****Visiteur**** clique sur un habitat pour voir les détails.
5. ****Application**** récupère les détails de l'habitat et les animaux associés depuis la base de données.
6. ****Application**** renvoie la réponse avec les détails.

5. Documentation du Déploiement de l'Application

Pour déployer votre application, suivez ces étapes :

1. Préparer le Code pour le Déploiement :
 - Assurez-vous que toutes les dépendances sont installées.
 - Exécutez les migrations de base de données pour MySQL et MongoDB.
2. Configurer MongoDB Atlas pour la Base de Données NoSQL :
 - Créez une nouvelle base de données MongoDB sur ****MongoDB Atlas****.