

Documentation technique – Zoo Arcadia

I. Choix Technologiques

PHP

- **Large communauté** : PHP est utilisé par environ 76.2% de tous les sites web dont le langage côté serveur est connu.
- Permet de faire **des sites dynamiques**.
- **Communauté active** : Une large communauté active assure un bon support et de nombreuses bibliothèques et frameworks, facilitant ainsi le développement.
- **Documentation disponible** : La vaste documentation disponible en ligne permet de trouver rapidement des solutions aux problèmes, ce qui est important puisque je suis en apprentissage du code.
- **Compatibilité avec Symfony**
 - **Framework Symfony** : Symfony, l'un des frameworks PHP les plus populaires et puissants, a été choisi pour ce projet. Symfony fournit une architecture robuste, des composants réutilisables, et des outils de développement avancés.
 - **Intégration fluide** : PHP est naturellement compatible avec Symfony, permettant une intégration fluide et des performances optimales.
- **Mises à jour régulières** : PHP bénéficie de mises à jour régulières qui corrigent les vulnérabilités et améliorent la sécurité globale de l'application.
- **Productivité** : Grâce à des outils comme Composer et des IDE optimisés pour PHP, cela permet une meilleure productivité.
- **Déploiement et hébergement**
 - **Disponibilité des hébergements** : PHP est supporté par la plupart des hébergeurs web, ce qui simplifie le déploiement de l'application.
 - **Outils de déploiement** : Des outils de déploiement comme Docker et des services comme Heroku supportent PHP, permettant une mise en production rapide et fiable.

Conclusion : Le choix de PHP pour le projet Zoo Arcadia a été motivé par la combinaison de ses performances, sa sécurité, sa compatibilité avec Symfony, et sa facilité d'utilisation. Ces facteurs assurent une base solide pour le développement, la maintenance et l'évolution de l'application, garantissant ainsi un projet réussi et durable.

Symfony

- **Architecture Robuste et Modulaire**

- **Modularité** : Symfony est conçu avec une architecture modulaire qui permet d'utiliser uniquement les composants dont on a besoin.
- **Intégration facile** : Symfony s'intègre facilement avec des outils et des technologies populaires comme Doctrine pour l'ORM, Twig pour le templating, et Composer pour la gestion des dépendances, ce qui facilite le développement et la maintenance de l'application.
- **Modèle MVC**
 - **Flexibilité et Réutilisabilité**: Grâce au modèle MVC, on peut réutiliser des composants, améliorer la flexibilité et rendre le code plus propre et organisé.
- **Compatibilité avec MySQL et MongoDB** : L'ORM Doctrine utilisé par symfony est compatible avec MySQL pour la gestion des bases de données relationnelles et avec MongoDB pour les bases de données NoSQL, ce qui correspond parfaitement aux besoins du projet Zoo Arcadia.
- **Sécurité**
 - **Fonctionnalités de sécurité robustes** : notamment la gestion des utilisateurs et des rôles, la protection contre les attaques CSRF (Cross-Site Request Forgery) et la gestion sécurisée des sessions. En plus, Symfony propose des mesures de sécurité telles que la validation des entrées, la protection contre les attaques XSS (Cross-Site Scripting) et la prévention des injections SQL.
 - **Mises à jour régulières** : Symfony bénéficie de mises à jour régulières qui incluent des correctifs de sécurité, assurant que l'application reste sécurisée contre les nouvelles menaces.
- **Scalabilité** : Symfony est capable de gérer des applications web de grande envergure, ce qui est essentiel pour un projet comme Zoo Arcadia qui pourrait voir une augmentation du trafic utilisateur.
- **Documentation et Communauté Active**
 - **Documentation** : Symfony possède une documentation complète et bien organisée, ce qui facilite son l'apprentissage et son l'utilisation.
 - **Communauté active** : Une large communauté de développeurs utilise et contribue à Symfony, ce qui signifie qu'il y a un support abondant et de nombreuses ressources disponibles en ligne.

Conclusion : Symfony est un choix idéal pour le projet Zoo Arcadia en raison de sa robustesse, de sa flexibilité, de ses fonctionnalités de sécurité avancées, de ses performances optimisées, et de sa large adoption dans la communauté de développement web. Ces caractéristiques assurent que l'application sera bien conçue, maintenable, sécurisée et capable de répondre aux besoins évolutifs du zoo.

MySQL

- **Performance et Simplicité**
 - **Performance en lecture** : MySQL est souvent considéré comme offrant de meilleures performances en lecture que PostgreSQL, ce qui est essentiel pour les applications web qui nécessitent des accès rapides aux données, comme la visualisation des animaux et des services du zoo.

- **Facilité d'installation et de configuration** : MySQL est réputé pour sa simplicité d'installation et de configuration, ce qui permet une mise en œuvre rapide et efficace, facilitant ainsi le développement initial et la maintenance.
- **Large Adoption et Support**
 - **Popularité et communauté** : MySQL est l'une des bases de données les plus populaires et bénéficie d'une large communauté active. Cela signifie que l'on peut trouver facilement des ressources et des solutions.
- **Compatibilité et Intégration**
 - **Compatibilité avec PHP** : MySQL est souvent utilisé en conjonction avec PHP et Symfony, ce qui assure une intégration fluide et optimisée pour le développement web.
- **Excellent outil pour les relations**

Conclusion : Le choix de MySQL pour le projet Zoo Arcadia est principalement motivé par sa performance, sa simplicité, sa compatibilité avec les technologies utilisées, et son large support communautaire. Ces facteurs combinés assurent une base solide pour le développement, la maintenance et l'évolution de l'application.

MongoDB

- **Simplicité de Développement**
 - **Document-oriented storage** : Le modèle orienté document de MongoDB est naturel pour stocker des enregistrements de vues, chaque document pouvant représenter une vue avec des champs comme l'ID de l'animal, l'heure de la vue, et d'autres métadonnées pertinentes. En terme d'apprentissage c'est assez similaire au MVC
- **Opérations en temps réel**
 - **Analyse en temps réel** : MongoDB est bien adapté pour l'analyse en temps réel des données, ce qui permet de générer des statistiques sur les vues des animaux.
- **Communauté**
 - **Support et documentation** : MongoDB bénéficie d'un support actif de la communauté et d'une documentation exhaustive, ce qui facilite le développement et la résolution des problèmes.
- **Stockage d'information** : Excellent outil pour stocker de grand nombre d'informations sans relations

Conclusion : MongoDB est un choix idéal pour comptabiliser les vues sur les animaux dans le projet Zoo Arcadia en raison de sa flexibilité, sa performance, sa simplicité de développement, et ses capacités de traitement en temps réel. Ces caractéristiques permettent de gérer efficacement les données de vues.

Bootstrap

- **Design Responsive**

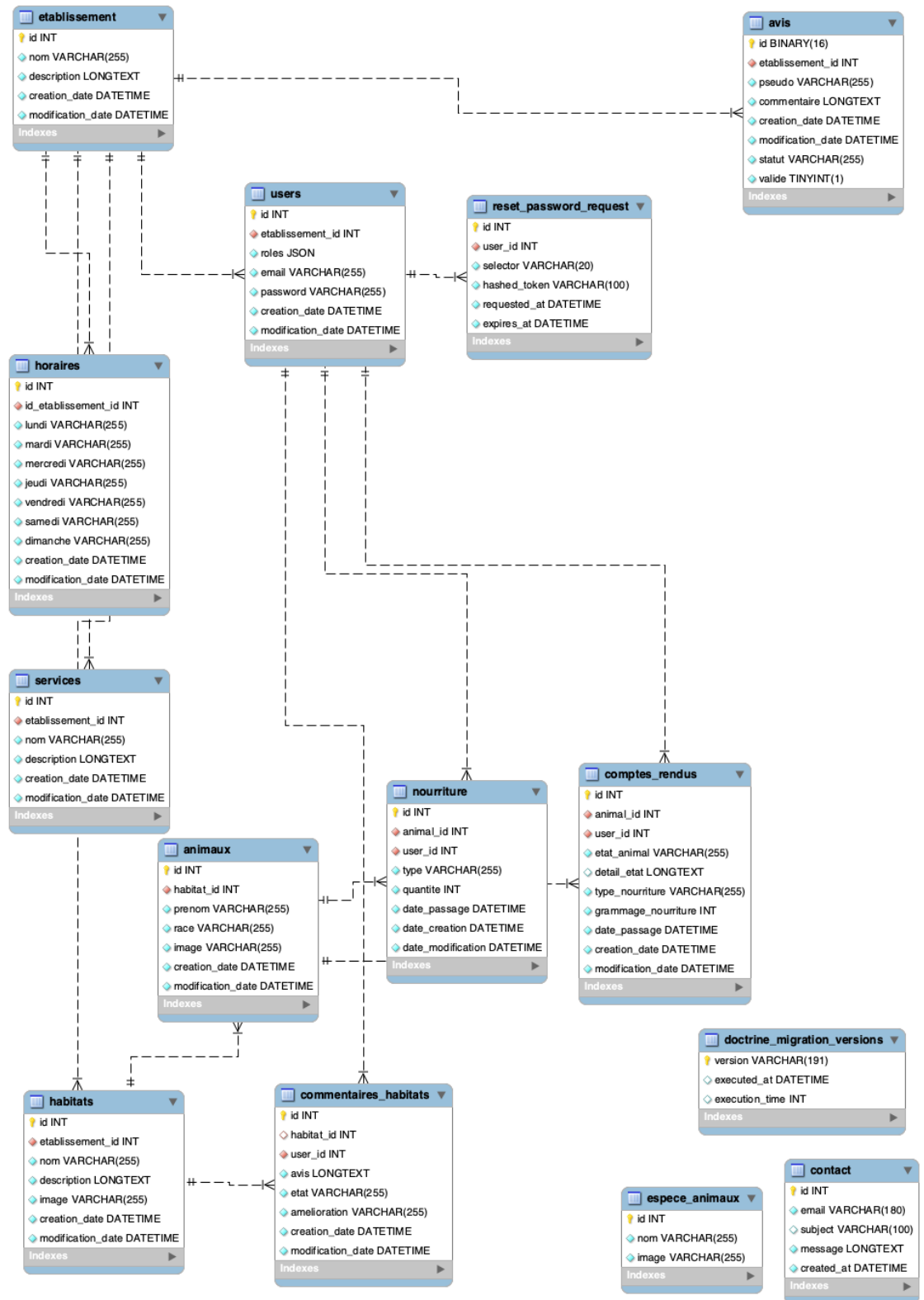
- **Grille flexible** : Le système de grille flexible de Bootstrap permet de créer des mises en page adaptatives qui s'ajustent automatiquement aux différentes tailles d'écran.
- **Large Bibliothèque de Composants UI**
 - **Composants préconstruits** : Bootstrap fournit une vaste bibliothèque de composants UI prêts à l'emploi, tels que des boutons, des formulaires, des cartes, des modals, des carrousels, etc. Cela permet de développer rapidement des interfaces utilisateur attrayantes et fonctionnelles sans avoir à écrire beaucoup de CSS ou de JavaScript personnalisé.
 - **Cohérence visuelle** : Utiliser Bootstrap assure une cohérence visuelle dans toute l'application, ce qui améliore l'expérience utilisateur en offrant une apparence et un comportement uniformes.
- **Personnalisation et Extensibilité**
 - **Facilité de personnalisation** : Bien que Bootstrap offre des styles par défaut, il est facilement personnalisable.
- **Documentation et Support Communautaire**
 - **Documentation exhaustive** : Bootstrap dispose d'une documentation complète et bien organisée, ce qui facilite l'apprentissage et l'utilisation du framework.
 - **Support communautaire** : Une large communauté de développeurs utilise Bootstrap, ce qui signifie qu'il y a un support abondant et de nombreuses ressources disponibles en ligne pour résoudre les problèmes.
- **Intégration avec Symfony**
 - **Compatibilité avec Symfony** : Bootstrap s'intègre parfaitement avec Symfony. Cette compatibilité permet de développer rapidement des interfaces utilisateur tout en tirant parti des puissantes fonctionnalités backend de Symfony.
 - **Facilité d'utilisation avec Twig** : Utiliser Bootstrap avec Twig simplifie la création de vues dynamiques et réactives.
- **Meilleures Pratiques et Accessibilité**
 - **Conformité aux normes** : Bootstrap suit les meilleures pratiques.

II. Configuration de l'environnement de travail

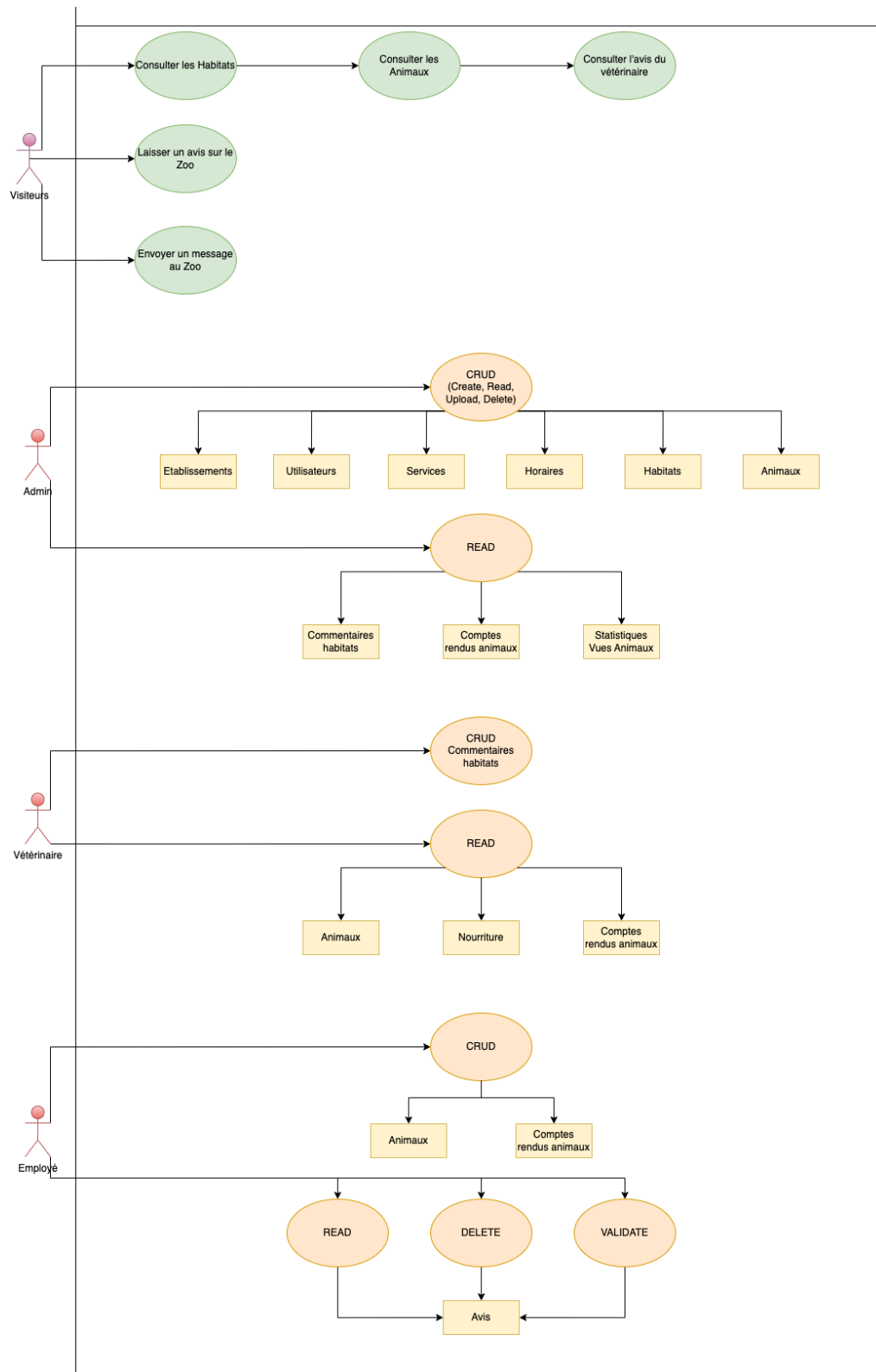
1. Installation de l'IDE
 - Utilisation de VS Code
2. Création d'un dossier projet sur la machine
3. Contrôle des versions
 - **Git** : Installation de Git pour le contrôle de version et la gestion des branches depuis le dossier
 - **GitHub** : Utilisation de GitHub pour héberger le code source et faciliter la collaboration.

4. Installation de php
5. Installation de composer
6. Installation de symfony CLI
7. Création d'un projet symfony depuis le dossier projet
8. Installation de MySQL et Téléchargement de SequelAce comme interface MySQL
9. Installation de MongoDB et téléchargement de l'interface MongoDB Compass
10. Configuration de bootstrap et importation des fichiers bootstrap.min.css et bootstrap.min.js dans le dossier public
11. Configuration du fichier .env.local

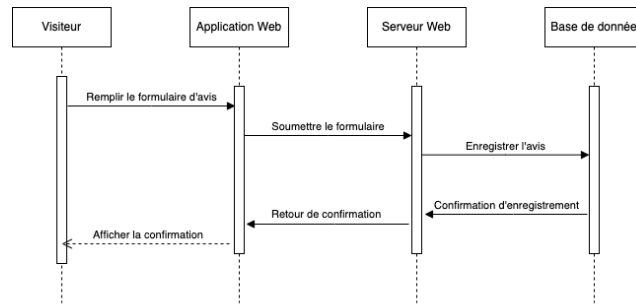
III. Modèle Conceptuel de Données



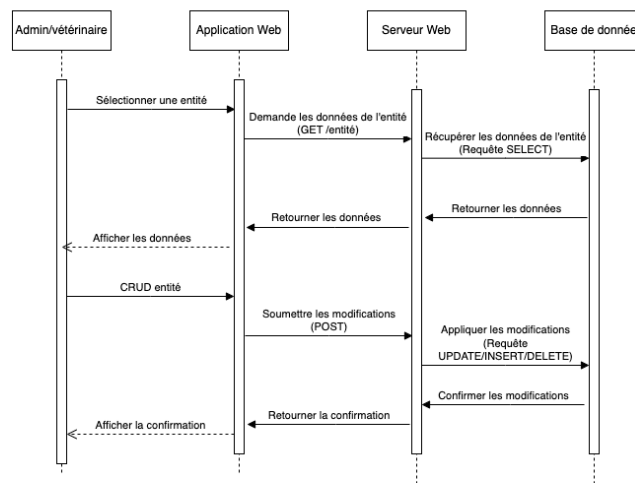
IV. Diagramme d'utilisation et Diagramme de séquence



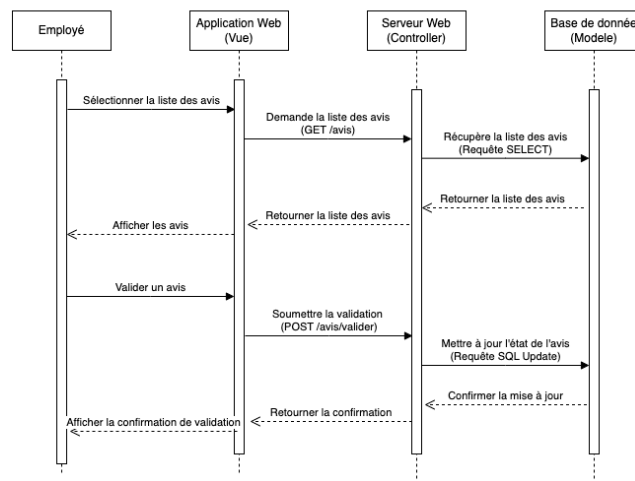
Laisser un avis au zoo



Gérer les entités



Valider les avis



V. Déploiement

1. Mise en place du serveur de production

- Installation de Heroku CLI
- Création d'un compte Heroku
 - heroku login
- Création d'un Projet Heroku *zoo-arcadia-studi*
 - heroku create zoo-arcadia
- Ajouter l'add-on JawsDB pour MySql
 - heroku addons:create jawsdb:kitefin

2. Configurer MongoDB Atlas

- Configurer les variables d'environnement
 - Obtenez l'URL de connexion MongoDB Atlas
 - Connectez-vous
 - Section Cluster, cliquez sur connect
 - Sélectionnez connect your application et copiez l'URL de connexion. Elle ressemblera à ceci
mongodb+srv://<username>:<password>@cluster0.mongodb.net/<dbname>?retryWrites=true&w=majority.

3. Configurer les variables d'environnement sur Heroku

- Configurer les variables d'environnement nécessaires
 - APP_ENV
 - APP_SECRET
 - DATABASE_URL
 - MONGODB
 - MONGODB_DB
 - MONGODB_URL
 - BASE_URL
 - MAILER_DSN
 - RECAPTCHA3_KEY
 - RECAPTCHA3_SECRET

4. Créer le Procfile

- Créez le fichier à la base du projet et ajoutez-y
 - web: heroku-php-apache2 public/

5. Ajoutez les dépendances

- Depuis votre console
 - `composer install --no-dev --optimize-autoloader`

6. Configurer la base de données relationnelle

- Migration de la base de données local vers JawsDB
 - Exporter la base de données locale en créant un « backup.sql »
 - `mysqldump -u local_user -p local_db_name > local_db_backup.sql`
 - Importer le dump backup dans JawsDB (les informations sont dans le Dashboard de JawsDB)
 - `mysql -h <jawsdb_host> -u <jawsdb_user> -p <jawsdb_name> < local_db_backup.sql`
 - Configurer les migrations Doctrine
 - `heroku run php bin/console doctrine:database:create --env=prod`
 - `heroku run php bin/console doctrine:migrations:migrate --env=prod`

7. Configurer la base de données non relationnelle

- Créer le schéma dans MongoDB Atlas
 - Depuis la console
 - `php bin/console doctrine:mongodb:schema:create`
- Exporter les données locales
 - Depuis la console
 - `mongodump --uri="mongodb://localhost:27017/zoo_db" --out=/path/to/dump`
- Importer les données dans MongoDB Atlas
 - Depuis la console
 - `mongorestore --uri="mongodb+srv://<username>:<password>@<cluster-url>/<dbname>" --drop /path/to/dump`

8. Déployer l'application sur Heroku

- Stagez vos modifications
 - `git add .`
- Validez-les
 - `git commit -m "Prepare for Heroku deployment"`
- Mettez à jour le dépôt distant
 - `git push heroku main`