

NOM	Ouattara
Prénom	Nanwokan
Date de naissance	22/05/2003

Copie à rendre

TP – Développeur Web et Web Mobile

Documents à compléter et à rendre

Lien du git : <https://github.com/Nanwokan/Zoo-Arcadia.git>

Lien de l'outil de gestion de projet : <https://app.clickup.com/9015082414/v/s/90152702178>

Lien du déploiement : <https://zoo-arcadia-2hj34516a-nanwokans-projects.vercel.app>

Login et mot de passe administrateur : j.legrand@gmail.com / Legrand2024.

SANS CES ELEMENTS, VOTRE COPIE SERA REJETEE

Partie 1 : Analyse des besoins

1. Effectuez un résumé du projet en français d'une longueur d'environ 20 lignes soit 200 à 250 mots

Le projet "Zoo Arcadia" vise à développer une application web complète pour la gestion d'un zoo. Cette application comprend plusieurs interfaces dédiées à différents types d'utilisateurs : administrateurs, employés, vétérinaires et visiteurs. Les administrateurs peuvent gérer les comptes utilisateurs, les services, les habitats et les animaux du zoo. Ils ont la possibilité d'ajouter, de modifier et de supprimer des informations et des images concernant les animaux et les habitats. Les employés ont accès à des fonctionnalités spécifiques, telles que la validation des avis des visiteurs et la gestion des services offerts par le zoo. Les vétérinaires peuvent rédiger et consulter des rapports vétérinaires, assurant ainsi le suivi de la santé des animaux. Les visiteurs, quant à eux, peuvent consulter les informations détaillées sur les animaux, les habitats et les services via une interface utilisateur réactive et intuitive. Le projet comprend également des mécanismes de sécurité pour protéger les données, comme l'authentification par JWT et le chiffrement des mots de passe. L'application est développée en utilisant des technologies modernes telles que Node.js pour le backend, MySQL pour la base de données, et HTML, CSS, JavaScript pour le frontend, garantissant ainsi une application performante, réactive et sécurisée. L'objectif final est de fournir une solution robuste et facile à utiliser pour la gestion quotidienne du zoo, tout en offrant une expérience utilisateur enrichissante et sécurisée pour les différents intervenants.

2. Exprimez le cahier des charges, l'expression du besoin ou les spécifications fonctionnelles du projet

Cahier des charges

- **Gestion des utilisateurs et des rôles**

- Création, modification et suppression des comptes utilisateurs.
- Assignation de rôles spécifiques (administrateur, employé, vétérinaire, visiteur).

- **Gestion des animaux**

- Ajout, modification et suppression des informations sur les animaux, y compris les images et les états de santé.
- Fonctionnalité de recherche et de filtrage des animaux par habitat et par état de santé.
- **Gestion des habitats**
 - Ajout, modification et suppression des informations sur les habitats du zoo.
 - Association des habitats avec les animaux et les images correspondantes.
- **Gestion des services**
 - Ajout, modification et suppression des services proposés par le zoo (par exemple, visites guidées, spectacles d'animaux).
- **Gestion des avis**
 - Validation et consultation des avis laissés par les visiteurs.
 - Possibilité de répondre aux avis et de les classer par pertinence.
- **Fonctionnalités pour les vétérinaires**
 - Rédaction et consultation des rapports vétérinaires.
 - Suivi des traitements et des soins des animaux.
- **Interface utilisateur réactive**
 - Adaptation de l'interface utilisateur pour différents appareils (ordinateurs, tablettes, smartphones).
 - Utilisation d'un design cohérent et intuitif pour améliorer l'expérience utilisateur.

Partie 2 : Spécifications technique

1. Spécifiez les technologies que vous avez utilisé en justifiant les conditions d'utilisation et pourquoi le choix de ses éléments

- **Frontend**
 - **HTML, CSS, JavaScript** : pour la structure, le style et l'interactivité de l'application.
 - **Pourquoi** : Ces technologies sont standards pour le développement web et permettent de créer des interfaces réactives et interactives.
- **Backend**

- **Node.js, Express** : pour la création de l'API RESTful et la gestion des requêtes serveur.
 - **Pourquoi** : Node.js offre une performance élevée et une gestion efficace des connexions multiples, tandis qu'Express facilite la création d'API robustes.
- **Base de données**
 - **MySQL** : pour le stockage des données relationnelles.
 - **Pourquoi** : MySQL est un SGBD relationnel fiable et performant, idéal pour gérer les relations complexes entre les différentes entités du zoo (utilisateurs, animaux, habitats, etc.).

2. Comment avez-vous mis en place votre environnement de travail ? Justifiez vos choix. (README.md)

README.md

- **Contenu** : Instructions pour configurer l'environnement de développement, installer les dépendances (via npm), et exécuter le projet localement.
 - **Justification** : Un fichier README détaillé est essentiel pour garantir que tous les développeurs impliqués peuvent rapidement comprendre et configurer l'environnement de développement.
- ## 3. Énumérez les mécanismes de sécurité que vous avez mis en place, aussi bien sur vos formulaires que sur les composants front-end ainsi que back-end.
- **Formulaires**
 - Validation côté client avec JavaScript pour des entrées correctes et formatées.
 - Validation côté serveur avec Express Validator pour prévenir les injections SQL et les entrées malveillantes.
 - **Backend**
 - Utilisation de tokens JWT pour l'authentification sécurisée des utilisateurs.
 - Chiffrement des mots de passe avec bcrypt pour protéger les données sensibles des utilisateurs.
 - **Front-end**
 - Utilisation de CSP (Content Security Policy) pour prévenir les attaques XSS (Cross-Site Scripting).
 - Utilisation de HTTPS pour sécuriser les communications entre le client et le serveur.

4. Décrivez une veille technologique que vous avez effectuée, sur les vulnérabilités de sécurité.

Vulnérabilités de sécurité

- Utilisation d'outils comme Snyk et npm audit pour identifier et corriger les vulnérabilités des dépendances.
- Suivi des mises à jour de sécurité via les blogs de sécurité, les forums et les newsletters (par exemple, OWASP, Hacker News).

Partie 3 : Recherche

1. Décrivez une situation de travail ayant nécessité une recherche durant le projet à partir de site anglophone. N'oubliez pas de citer la source

- **Problème** : Intégration de la gestion des images pour les habitats et les animaux. J'ai aussi rencontré un problème sur le Dashboard de l'administrateur pour les données des graphique et statistiques ainsi que les compteur de vue des animaux demandé. Ce sont des problèmes qui m'ont empêcher d'avancer sur beaucoup d'élément dans le back-end et sur la page animaux des visiteurs.
- **Source** : Documentation officielle de Multer (<https://github.com/expressjs/multer>)

2. Mentionnez l'extrait du site anglophone qui vous a aidé dans la question précédente en effectuant une traduction en français.

Extrait : "Multer is a node.js middleware for handling multipart/form-data, which is primarily used for uploading files."

Traduction : "Multer est un middleware pour node.js qui gère les données multipart/form-data, principalement utilisé pour le téléchargement de fichiers."

Partie 4 : Informations complémentaire

1. Autres ressources
2. Informations complémentaires

