

NOM	MOULINDRE
Prénom	NOUFAIL
Date de naissance	07/12/1995

Copie à rendre

TP – Développeur Web et Web Mobile

Documents à compléter et à rendre

Lien du git :

Lein frontend :

<https://github.com/Neyliss/ZOOFRONTEND.git>

Lien backend :

<https://github.com/Neyliss/zoo.git>

Lien de l'outil de gestion de projet : J'avais un mois pour réaliser le projet donc j'ai travaillé uniquement via une organisation papier basé sur : Semaine A FRONT END / SEMAINE B BACKEND

Lien du déploiement : voir le lien backend

Login et mot de passe administrateur : admin@mail.com mot de passe : Adminlog456789

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

Après avoir obtenu mon diplôme de Développeur Web et Web Mobile, j'ai été embauché par DevSoft pour un projet passionnant : créer une application web pour le zoo Arcadia, situé en Bretagne. Le directeur du zoo, José, souhaite une application qui reflète les valeurs écologiques du zoo, tout en permettant aux visiteurs de découvrir les animaux, leurs états de santé, et les services offerts par le zoo.

Dans ce projet, je suis chargé de développer à la fois la partie front-end et back-end de l'application. Pour la partie front-end, je vais concevoir des interfaces utilisateur attrayantes et intuitives qui incluent une page d'accueil avec une présentation du zoo, un menu de navigation complet, et des vues globales des habitats et services. Chaque animal et habitat aura sa propre page détaillée, permettant aux visiteurs d'explorer le zoo virtuellement.

Du côté back-end, je mettrai en place une base de données relationnelle pour stocker les informations sur les animaux, les habitats, et les avis des visiteurs. J'utiliserai également une base de données NoSQL pour suivre les statistiques de consultation des animaux. En plus de cela, l'application comprendra des espaces dédiés pour les vétérinaires et les

©Studi - Reproduction interdite

employés, qui leur permettront de saisir des informations sur les animaux et de gérer les avis des visiteurs.

Je vais m'assurer que l'application est sécurisée et déployée correctement, et je fournirai une documentation complète pour garantir sa maintenance future. Mon objectif est de créer une application qui non seulement répond aux besoins de José et du zoo, mais qui offre également une expérience enrichissante aux visiteurs.

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

Voir les fichiers disponible via le lien GitHub frontend dans le dossier PDF

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

J'ai utilisé symfony /httpd (serveur apache)/ VisualStudio Code (IDE)/ BOOTSTRAP/ JS avec un système de router. Il était compliqué pour moi d'installer certaine fonctionnalité sur mon Mac et développé mon projet avec des framework comme node.js ou encore vue.js. En effet, après avoir fait des recherches, la réalisation de mon projet à nécessité énormément de configurations qui m'ont parfois forcé à formater mon pc et recommencer.

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

J'ai mis en place mon environnement de travail en fonction des compatibilités lié à mon Mac.

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.

Dans le cadre du développement de l'application web pour le zoo Arcadia, j'ai mis en place plusieurs mécanismes de sécurité pour protéger à la fois le front-end et le back-end de l'application. Voici les mesures que j'ai prises :

1. Protection contre les failles XSS (Cross-Site Scripting) : J'ai implémenté une validation stricte et un encodage approprié des données utilisateur sur tous les formulaires. Cela empêche l'injection de scripts malveillants qui pourraient compromettre la sécurité de l'application ou des utilisateurs.

3. Framework Symfony très sécurisé: J'ai choisi d'utiliser Symfony pour le développement back-end de l'application, car ce framework est reconnu pour sa robustesse et ses bonnes pratiques en matière de sécurité. Symfony offre des outils intégrés pour la gestion des sessions, la protection CSRF (Cross-Site Request Forgery), et d'autres vulnérabilités courantes.

4. Utilisation de JWT (JSON Web Tokens) : Pour sécuriser l'authentification et les sessions utilisateurs, j'ai mis en place un système de tokens JWT. Ces tokens permettent de vérifier l'identité des utilisateurs et de s'assurer que seules les personnes autorisées peuvent accéder aux ressources protégées de l'application.

5. Validation et filtrage des données: Tous les formulaires de l'application intègrent des mécanismes de validation côté client et côté serveur. Cela garantit que seules les données correctement formatées et attendues sont traitées par le système, réduisant ainsi le risque d'attaques par injection.

6. Hébergement sécurisé et HTTPS : L'application est déployée sur un serveur sécurisé avec HTTPS activé, garantissant que toutes les communications entre les utilisateurs et le serveur sont chiffrées et protégées contre les interceptions.

En mettant en place ces mécanismes de sécurité, je m'assure que l'application est robuste contre les attaques courantes et protège efficacement les données des utilisateurs.

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

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

La configuration de mon environnement de travail et la document de mon API.

le lien pour la configuration de mon environnement de travail :

<https://getgrav.org/blog/macos-sonoma-apache-multiple-php-versions>

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

Le dernier macOS 13.0 Sonoma est livré avec Apache 2.4 préinstallé, cependant, ce n'est plus une tâche simple d'utiliser cette version avec Homebrew car Apple a supprimé certains scripts requis dans cette version. Cependant, la solution est d'installer Apache 2.4 via Homebrew, puis de le configurer pour qu'il fonctionne sur les ports standard (80/443).

Si vous avez déjà l'Apache intégré en cours d'exécution, il devra d'abord être arrêté et tous les scripts de chargement automatique supprimés. Cela ne fait vraiment pas de mal d'exécuter toutes ces commandes dans l'ordre - même s'il s'agit d'une nouvelle installation :

```
Sudo apachectl -k stop
```

```
Sudo launchctl unload -w /System/Library/LaunchDaemons/org.apache.httpd.plist  
2>/dev/null
```

Maintenant, nous devons installer la nouvelle version fournie par Brew :

```
Brew install httpd
```

Sans options, httpd n'aura pas besoin d'être construit à partir de la source, il s'installe donc assez rapidement. Une fois terminé, vous devriez voir un message comme :

Partie 4 : Informations complémentaire

1. Autres ressources
2. Informations complémentaires