

NOM	ICARE
Prénom	STEPHANE
Date de naissance	26/12/1982

Copie à rendre

TP – Développeur Web et Web Mobile

Documents à compléter et à rendre

Lien du git : <https://github.com/STARK2612/STUDI.git>

Lien de l'outil de gestion de projet :

<https://trello.com/invite/b/NPJn3k15/ATTI9297981a8f324b821c4c631afc3ace9cD9CF74BB/promo-novembre-decembre-2024>

Lien du déploiement : <https://zarca.alwaysdata.net/>

Login et mot de passe administrateur : « admin@arcadiazoo.com » et « G3nt1IJ0sé »

SANS CES ELEMENTS, VOTRE COPIE SERA REJETEE

Partie 1 : Analyse des besoins

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

Le projet consiste à développer une application web pour un zoo, avec pour objectif de refléter ses valeurs écologiques et son indépendance énergétique. La conception comprend plusieurs fonctionnalités pour différents types d'utilisateurs :

1. **Page d'accueil (US 1)** : Pour les visiteurs, présentant le zoo, ses habitats, services et avis.
2. **Menu de l'application (US 2)** : Offrant une navigation fluide avec accès à diverses sections et à la connexion pour le personnel autorisé.
3. **Vue globale des services (US 3)** : Récapitulant tous les services du zoo, modifiables par l'administrateur.
4. **Vue globale des habitats (US 4)** : Affichant les habitats avec leurs animaux associés, incluant les détails sur l'état de chaque animal.
5. **Avis (US 5)** : Permettant aux visiteurs de laisser des commentaires, soumis à validation par l'employé.
6. **Espace Administrateur (US 6)** : Pour gérer les comptes, les services, les habitats, les animaux, et accéder aux comptes rendus vétérinaires et à un tableau de bord des consultations.
7. **Espace Employé (US 7)** : Pour valider les avis, modifier les services et enregistrer les nourritures données aux animaux.
8. **Espace Vétérinaire (US 8)** : Pour rédiger des comptes rendus par animal, commenter les habitats et consulter l'alimentation des animaux.
9. **Connexion (US 9)** : Uniquement pour le personnel autorisé.
10. **Contact (US 10)** : Permettant aux visiteurs de contacter le zoo via un formulaire.
11. **Statistique de consultation (US 11)** : Enregistrant les consultations d'animaux pour des statistiques visibles dans le tableau de bord administrateur.

Le déploiement de l'application web est obligatoire, et un schéma de données est fourni pour guider l'implémentation.

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

Cahier des Charges - Application Web pour un Zoo Écologique

1. Introduction

L'application web est destinée à un zoo écologique. L'objectif est de créer une plateforme permettant aux visiteurs d'explorer les habitats, les services et les animaux du zoo, tout en reflétant les valeurs écologiques et durables de l'établissement.

2. Besoins Utilisateur

L'application doit répondre aux besoins des différents types d'utilisateurs suivants :

- **Visiteur** : Personne visitant le site web sans connexion.
- **Employé** : Personnel autorisé à gérer les avis, les services du zoo et à enregistrer les nourritures données aux animaux.
- **Vétérinaire** : Personne responsable de la santé des animaux, devant rédiger des comptes rendus et commenter les habitats.
- **Administrateur** : Responsable de la gestion globale de l'application et des utilisateurs.

3. Fonctionnalités Requises

Les fonctionnalités principales de l'application comprennent :

- **Page d'Accueil** : Présentation du zoo, de ses habitats, services et avis.
- **Menu de Navigation** : Faciliter l'accès aux différentes sections de l'application.
- **Gestion des Services** : Vue globale des services proposés, avec possibilité de modification par l'administrateur.
- **Gestion des Habitats et des Animaux** : Affichage des habitats et des animaux associés, avec détails sur l'état de chaque animal.
- **Gestion des Avis** : Soumission et validation des avis par l'employé.
- **Espaces Utilisateurs** : Connexion et gestion des comptes pour les employés, vétérinaires et administrateurs.
- **Contact** : Formulaire de contact pour les visiteurs.
- **Statistiques de Consultation** : Suivi du nombre de consultations par animal pour le tableau de bord administrateur.

4. Technologie Utilisée

La stack technique recommandée comprend HTML5, CSS avec Bootstrap, JavaScript pour le front-end, PHP avec PDO pour le back-end, MySQL/MariaDB/PostgreSQL pour la base de données relationnelle, MongoDB pour la base de données NoSQL, et des services de déploiement tels que fly.io, Heroku, Azure ou Vercel.

5. Contraintes de Développement

- Respecter les valeurs écologiques et durables du zoo dans le design et les fonctionnalités de l'application.
- Utiliser une base de données relationnelle et non relationnelle pour stocker les données.
- Assurer la sécurité des informations et des transactions utilisateur.

6. Livrables Attendus

- Application web fonctionnelle conforme aux spécifications.
- Documentation détaillée du code source et des fonctionnalités.
- Formation pour l'administration et la maintenance de l'application.

7. Contrôle Qualité et Validation

Les fonctionnalités de l'application seront testées pour assurer leur bon fonctionnement et leur conformité aux spécifications. Les utilisateurs finaux seront également impliqués dans le processus de validation.

8. Gestion de Projet

Un plan de gestion de projet détaillé sera établi, incluant la répartition des tâches, le suivi du calendrier et la communication régulière avec les parties prenantes.

Ce cahier des charges définit les exigences et les attentes pour le développement de l'application web du zoo écologique, visant à offrir une expérience utilisateur immersive tout en promouvant les valeurs environnementales de l'établissement.

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 :

Pour l'ensemble de l'application web, j'ai choisi d'utiliser plusieurs technologies différentes pour le développement, le fonctionnement et l'expérience utilisateur. Voici une analyse des principales technologies que j'ai utilisées et pourquoi je les ai choisies :

1. PHP (Hypertext Preprocessor) :

- Justification : J'ai opté pour PHP car c'est un langage de script côté serveur très répandu et robuste. Il offre une grande flexibilité et dispose de nombreuses fonctionnalités pour interagir avec les bases de données, gérer les sessions utilisateur et générer du contenu dynamique. Dans l'application, j'utilise PHP pour la logique métier, l'interaction avec la base de données MySQL et la génération de pages web dynamiques.

2. HTML (HyperText Markup Language) :

- Justification : J'utilise HTML car c'est le langage standard pour créer des documents web. Il fournit la structure de base des pages web en utilisant des balises pour définir différents éléments comme les titres, les paragraphes, les liens, etc. Dans l'application, j'utilise HTML pour structurer le contenu des pages web et définir leur mise en page.

3. CSS (Cascading Style Sheets) :

- Justification : J'utilise CSS pour styliser et mettre en forme les pages HTML. Il me permet de contrôler l'apparence des éléments HTML en définissant des règles de style telles que la couleur, la police, la disposition, etc. CSS est essentiel pour créer une interface utilisateur attrayante et cohérente. Dans l'application, j'utilise CSS pour la mise en page, la mise en forme et la conception visuelle.

4. JavaScript :

- Justification : J'ai intégré JavaScript car c'est un langage de script côté client qui ajoute des fonctionnalités interactives et dynamiques aux pages web. Je l'utilise pour manipuler le contenu HTML, répondre aux actions de l'utilisateur, valider les formulaires et créer des effets visuels. JavaScript est crucial pour des fonctionnalités telles que la manipulation du DOM, les requêtes AJAX, la validation des formulaires et la gestion des événements utilisateur.

5. MySQL (ou une autre base de données) :

- Justification : J'ai choisi MySQL car c'est un système de gestion de base de données relationnelle largement utilisé dans le développement web. Il offre une solution robuste et évolutive pour stocker et gérer les données de mon application, comme les informations des utilisateurs, les données sur les habitats, les animaux, les services, etc.

6. Bootstrap (ou un autre framework CSS) :

- Justification : J'ai opté pour Bootstrap, un framework CSS open-source, car il facilite le développement web en fournissant une collection de composants et de styles prédéfinis. Il me permet de créer rapidement des interfaces utilisateur réactives et attrayantes, tout en assurant la compatibilité avec les différents navigateurs. Dans l'application, j'utilise Bootstrap pour la mise en page, le design responsive et la création d'éléments d'interface utilisateur cohérents.

En utilisant ces technologies ensemble, l'application web bénéficie d'une combinaison de puissance, de flexibilité et de convivialité pour offrir une expérience utilisateur optimale. PHP fournit la logique métier côté serveur, MySQL stocke les données, HTML structure le contenu, CSS stylise l'interface, JavaScript ajoute de l'interactivité, et Bootstrap facilite le développement de l'interface utilisateur.

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

Configuration de mon environnement de travail

1. J'utilise XAMPP • Justification : J'ai opté pour XAMPP, un package logiciel qui rassemble Apache, MySQL, PHP et Perl pour créer un environnement de développement local. Son installation et sa configuration sont simples en font un choix populaire pour le développement web en PHP. Avec XAMPP, je peux me concentrer pleinement sur le développement de mon application, car il simplifie la configuration du serveur web et de la base de données.
2. J'utilise Visual Studio Code • Justification : Mon éditeur de prédilection est Visual Studio Code (VSCode), un outil léger, rapide et extensible offrant de nombreuses fonctionnalités pour le développement web. Il supporte plusieurs langages de programmation, dont le PHP, HTML, CSS et JavaScript. En plus, VSCode intègre Git et propose des terminaux intégrés, facilitant ainsi la gestion du code source et l'exécution de commandes en ligne.
3. J'utilise Git et GitHub • Justification : Pour le contrôle de version de mon code, j'utilise Git, un système distribué qui suit les modifications apportées au fil du temps. GitHub, quant à lui, me permet de stocker, partager et collaborer sur mes projets Git. Cette combinaison simplifie la gestion des versions du code source, le travail en équipe et le déploiement de mon application sur différentes plateformes.
4. J'utilise PHPMyAdmin pour gérer ma base de données • Justification : Pour la gestion de ma base de données MySQL, j'ai choisi PHPMyAdmin, une application web open-source offrant une interface conviviale. Elle me permet d'exécuter des requêtes SQL, de créer des bases de données, des tables, des utilisateurs, etc. Grâce à PHPMyAdmin, la gestion de ma base de données est simplifiée, surtout lors de la création de nouvelles bases de données et de l'importation de fichiers SQL.

En combinant ces outils et technologies, mon environnement de travail est configuré pour faciliter le développement, la gestion et le déploiement de mon application web en PHP. Avec XAMPP, j'ai un serveur web local complet, VSCode m'offre un éditeur de code puissant, Git et GitHub me permettent de gérer le code source et PHPMyAdmin simplifie la gestion de ma base de données MySQL.

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 :

Sur les formulaires, j'ai pris plusieurs mesures de sécurité. Tout d'abord, j'utilise JavaScript pour valider les données saisies par l'utilisateur avant de les envoyer au serveur, cela réduit les requêtes invalides et améliore l'expérience utilisateur. Ensuite, j'ai mis en place une validation côté serveur pour toutes les données envoyées par les formulaires, assurant ainsi que seules les données valides et autorisées sont traitées. De plus, je filtre et j'échappe les données entrées par l'utilisateur pour prévenir les attaques XSS, et j'utilise des jetons CSRF dans les formulaires pour éviter les attaques CSRF. Ces jetons sont générés côté serveur à chaque soumission de formulaire et vérifiés lors du traitement de la demande pour garantir leur légitimité.

Pour ce qui est des composants front-end, j'ai configuré mon serveur web pour utiliser HTTPS, assurant ainsi le cryptage des données échangées entre le navigateur des utilisateurs et le serveur.

Du côté du back-end, je m'assure de valider et d'échapper rigoureusement les données entrées par les utilisateurs avant de les utiliser dans des requêtes SQL ou des opérations de traitement, afin de prévenir les attaques par injection SQL. Pour protéger les sessions utilisateur, j'utilise des techniques de gestion sécurisée des sessions telles que l'utilisation de cookies sécurisés, de jetons d'authentification et la déconnexion automatique après une période d'inactivité. Enfin, je veille à maintenir tous les composants logiciels à jour avec les derniers correctifs de sécurité pour combler les vulnérabilités connues.

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

1. Injection SQL :

- Je commence par identifier les points d'entrée de mon application où les données utilisateur sont traitées et transmises à la base de données.
- J'utilise des requêtes préparées avec des paramètres pour éviter les attaques par injection SQL.
- Je m'assure que mes requêtes SQL sont correctement échappées ou utilisent des méthodes de liaison de paramètres pour éviter les injections SQL.

2. Cross-Site Scripting (XSS) :

- Je vérifie attentivement le code HTML généré dynamiquement pour m'assurer que les données utilisateur sont correctement échappées, empêchant ainsi les attaques XSS.
- J'intègre des bibliothèques de validation des entrées pour filtrer et nettoyer les données utilisateur avant de les afficher sur le frontend.
- J'utilise des en-têtes HTTP comme Content-Security-Policy pour limiter l'exécution de scripts non autorisés.

3. Gestion des sessions et des cookies :

- Je veille à ce que mes cookies de session soient sécurisés en utilisant des attributs tels que Secure et HttpOnly.
- Je valide et nettoie toujours les données de session côté serveur pour éviter les attaques de session hijacking et de fixation de session.

4. Validation des entrées utilisateur :

- Je mets en place une validation côté serveur pour tous les formulaires et les données utilisateur afin de détecter et de rejeter les entrées malveillantes ou incorrectes.
- J'utilise des filtres d'entrée adaptés à chaque type de données (par exemple, validation des adresses email, des numéros de téléphone, etc.).

5. Gestion des fichiers téléchargés :

- Je restreins les types de fichiers autorisés à être téléchargés.
- Avant de stocker les fichiers téléchargés sur le serveur, je vérifie qu'ils ne contiennent pas de scripts malveillants ou de charges utiles dangereuses.

6. Mises à jour de sécurité :

- Je surveille régulièrement les avis de sécurité et les mises à jour des composants tiers utilisés dans mon application.
- J'applique systématiquement les correctifs de sécurité et les mises à jour recommandées pour tous les composants de mon application.

7. Sécurité du côté client :

- J'éduque mes utilisateurs sur les bonnes pratiques de sécurité, comme la création de mots de passe forts et la reconnaissance des URL malveillantes.
- Je mets en place des mécanismes de protection contre les attaques de force brute, telles que les limites de tentatives de connexion.

Il est essentiel de réaliser cette veille technologique de manière régulière pour rester au courant des dernières menaces et vulnérabilités, et pour maintenir mon application web sécurisée contre les attaques potentielles.

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 :

Pendant le développement du projet ZOOArcadia, j'ai rencontré un défi majeur lié à la gestion des sessions utilisateur et des autorisations d'accès. Je devais mettre en place un système d'authentification sécurisé pour différents types d'utilisateurs, comme les administrateurs, les vétérinaires et les employés.

Pour résoudre ce problème, j'ai entrepris une recherche sur des sites anglophones spécialisés dans le développement web, tels que Stack Overflow et MDN Web Docs. J'ai exploré des articles, des tutoriels et des discussions portant sur des sujets tels que les meilleures pratiques pour l'authentification et l'autorisation en PHP, la gestion des sessions et le contrôle d'accès basé sur les rôles.

Après avoir parcouru différentes ressources, j'ai trouvé des solutions et des exemples de code pertinents pour mettre en œuvre un système d'authentification robuste et gérer les sessions utilisateur de manière sécurisée. J'ai également découvert des techniques pour appliquer efficacement des autorisations basées sur les rôles dans notre application.

En appliquant ces connaissances au projet ZOOArcadia, j'ai pu développer un système d'authentification et de gestion des autorisations qui répondait à aux besoins spécifiques. Cette recherche sur les sites anglophones a été cruciale pour surmonter ce défi technique et assurer la sécurité et la fiabilité de l'application.

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

L'extrait pertinent du site anglophone qui m'a aidé dans la question précédente concerne la gestion des sessions utilisateur et des autorisations d'accès en PHP. Voici une traduction en français de l'extrait pertinent :

"Pour gérer les sessions utilisateur et les autorisations d'accès de manière sécurisée en PHP, vous pouvez utiliser des fonctions natives telles que `session_start()` pour démarrer une session et `$_SESSION` pour stocker des données utilisateur. Assurez-vous de valider les données d'entrée et de vérifier les autorisations avant de fournir un accès à certaines fonctionnalités de l'application."

Cette information, combinée à d'autres ressources, m'a permis de développer une solution adaptée aux besoins du projet ZOOArcadia.

Partie 4 : Informations complémentaires

- 1. Autres ressources**
- 2. Informations complémentaires**