

NOM	DE CARVALHO TEIXEIRA
Prénom	Manuel
Date de naissance	21/08/1990

## Copie à rendre

### TP – Développeur Web et Web Mobile

**Lien du git :**

<https://github.com/ManuDC7/ECF-ZOO-ARCADIA>

**Lien de l'outil de gestion de projet :**

<https://trello.com/invite/b/RO02reLs/>  
<ATTI01eeb726de02850dc083d3c1ae69460b2A62457E/ecf-arcadia>

**Lien du déploiement :**

<https://arcadia.lamphie.moe/index.php>



**Login et mot de passe administrateur :**

eMail: [josearcadia1@hotmail.fr](mailto:josearcadia1@hotmail.fr)

Mot de passe : [arcadia](#)

**Login et mot de passe vétérinaire :**

eMail: [michellerodrig@gmail.fr](mailto:michellerodrig@gmail.fr)

Mot de passe : [arcadia](#)

**Login et mot de passe employé :**

eMail: [timoudini@gmail.fr](mailto:timoudini@gmail.fr)

Mot de passe : [arcadia](#)

**Requêtes SQL à la main :**

Présent dans [src/requete.sql](#)

## Sommaire

Remerciement... page 04

Résumé... page 05

### Front-End

#### Développer la partie front-end d'une application web ou web mobile sécurisée

Compréhension des exigences du projet...page 06

Conception visuelle et prototypage...page 10

Infrastructure et outils de développement...page 12

Fondements du développement web avec HTML et CSS...page 14

Implémentation d'un Carousel interactif...page 17

### Back-End

#### Développer la partie back-end d'une application web ou web mobile sécurisée

Gestion sécurisée des données avec des bases de données... page 19

Administration avancée avec un tableau de bord personnalisé...page 25

Tests et déploiement sur serveur en ligne...page 27

Simplification des tâches avec des alias personnalisés...page 29

Sécurisation des échanges avec un certificat SSL...page 31

Conclusion...page 35

Annexe...page 36

## Remerciement

Je tiens à remercier Philippe Lam, Admininstrateur Système Linux chez ClaraNet Paris et diplômé de l'EPITA, qui m'a fait découvrir les fondamentaux nécessaires pour mener à bien mes projets. Sa patience et son enseignement m'ont été d'une grande aide.

Grâce à lui, j'ai eu l'opportunité de participer à un projet de bot Discord, ce qui a suscité en moi un réel intérêt pour le métier de développeur. Cette expérience m'a permis de mettre en pratique les connaissances acquises auprès de lui et durant ma formation, tout en en acquérant de nouvelles.

La collaboration avec Philippe a été enrichissante, car elle m'a permis de concrétiser mes compétences théoriques en développement dans un projet concret. Son accompagnement attentif m'a également motivé à explorer davantage le domaine du développement informatique.

En outre, Philippe m'a généreusement partagé un nom de domaine pour faciliter la mise en œuvre de mes projets. Cette ressource a été essentielle pour configurer correctement les certificats SSL et garantir la sécurité de mes services en ligne.

Je lui suis reconnaissant pour son soutien et sa contribution à mon évolution professionnelle.

## Résumé

José est directeur du zoo Arcadia, situé près de la foret de Brocéliande, en Bretagne depuis les années 1960. Le zoo se porte très bien, le personnel et les animaux sont heureux et José souhaite moderniser son établissement en développant une application web. C'est par le biais de son assistante que je suis contacté.

Il souhaite que l'application respire l'écologie et que chaque visiteur ressente les valeurs du zoo dont il est fier. Il nous explique ensuite comment il entrevoit son application: une page d'accueil de présentation qui mentionne également les avis laissés par les visiteurs, un menu pour accéder aux différentes pages, la possibilité pour le personnel de se connecter. Le visiteurs doit également être en mesure de voir les services proposés par le zoo, les habitats reconstitués ainsi que les animaux présent dans chaque habitat. Un formulaire de contact doit également lui être mis à disposition.

La solution technologique idéale serait une application web responsive, développée avec des technologies telles que HTML5, CSS3, JavaScript et PHP pour assurer une expérience utilisateur fluide et intuitive. Pour la gestion des données, une base de données SQLite et une base de donnée MongoDB sont adaptées.

En travaillant 8 heures par jour, ce projet devrait prendre environ 30 jours. Je vais commencer par définir clairement les besoins, puis concevoir l'architecture de l'application avant de passer au développement. Enfin, des phases de tests et de corrections seront nécessaires pour garantir la qualité du produit final.

## Front-End

### Activité – Type 1 : Développer la partie front-end d'une application web ou web mobile sécurisée

#### Compréhension des exigences du projet

Une application web sera développée pour le zoo Arcadia, permettant aux visiteurs de découvrir les valeurs écologiques du directeur via la page d'accueil. Ils auront également accès aux avis post-visite des clients, aux services, et à la liste des animaux présents dans chaque habitat. Les visiteurs auront la possibilité de contacter le zoo via un formulaire de contact, qui transmettra directement leurs demandes aux employés.

Le directeur aura un accès complet au site une fois connecté. Il pourra créer, supprimer et modifier des utilisateurs (vétérinaires ou employés), des animaux, des services, des habitats et les horaires d'ouverture du zoo. Il pourra consulter les comptes rendus des vétérinaires, avec des filtres pour trier et filtrer les comptes rendus par animal ou date. Un dashboard lui montrera le nombre de consultations par animal.

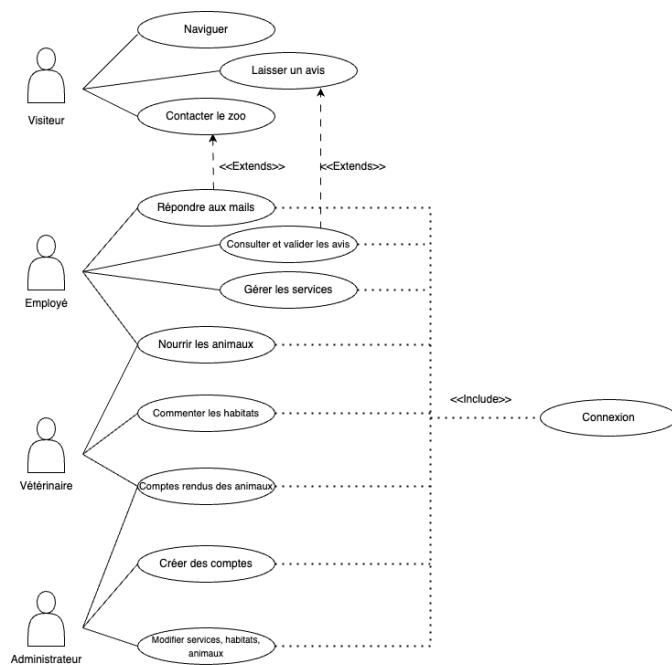
Les employés pourront approuver les avis des visiteurs pour qu'ils apparaissent sur la page d'accueil de l'application web et modifier les services proposés par le zoo. Ils auront également la possibilité d'ajouter des consommations de nourriture pour les animaux à partir de leur espace, en spécifiant la date, l'heure, la nourriture et la quantité.

Les vétérinaires auront un accès administratif limité. Ils pourront rédiger des comptes rendus sur les habitats et saisir des informations sur les animaux, telles que leur état, la nourriture proposée, le grammage et la date de leur intervention.

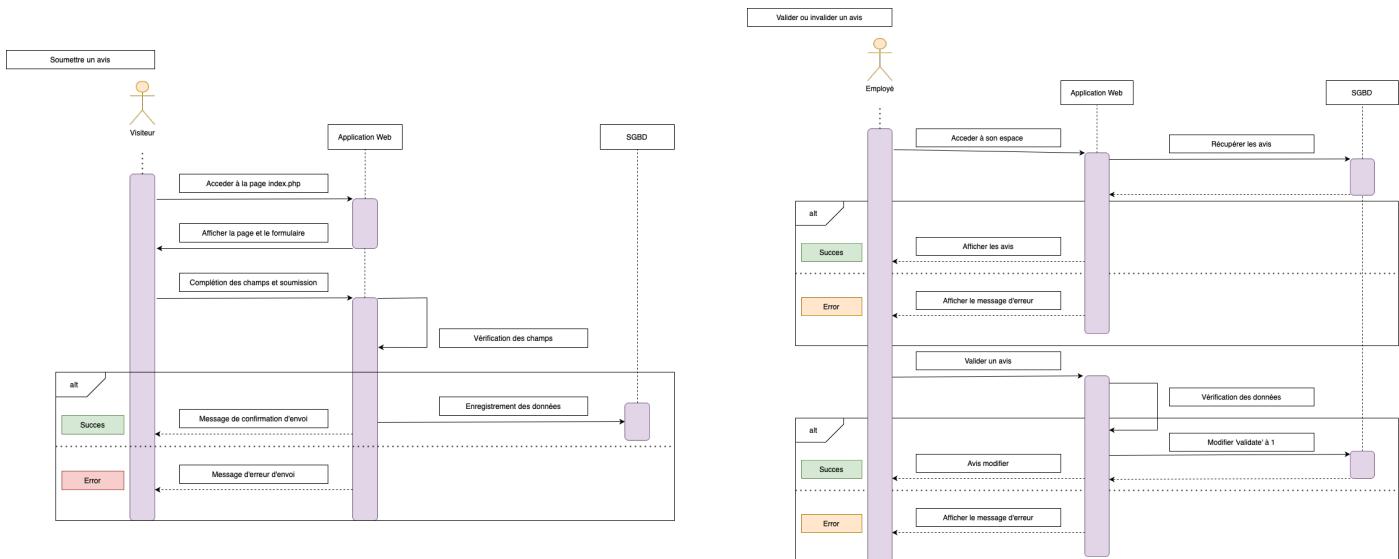
La conception d'une application web commence par une étape cruciale : l'analyse approfondie des besoins des utilisateurs. Cette phase essentielle permet de comprendre les attentes, les exigences et les comportements des utilisateurs finaux. En démarrant par cette analyse méticuleuse, nous sommes en mesure de cerner les divers rôles et actions envisageables pour chaque profil d'utilisateur.

La première étape de cette analyse consiste à réaliser une enquête approfondie pour identifier les besoins spécifiques et les fonctionnalités que les utilisateurs attendent de l'application. Cette démarche va bien au-delà de la simple identification des fonctionnalités de base ; elle vise à comprendre les motivations sous-jacentes de chaque action, les obstacles potentiels et les points de friction.

En résulte la création d'un diagramme de cas d'utilisation complet et précis. Ce diagramme offre une représentation visuelle des interactions entre les différents acteurs et les actions qu'ils peuvent entreprendre au sein de l'application. Il sert de fondation solide pour orienter le développement et garantir une expérience utilisateur optimale.



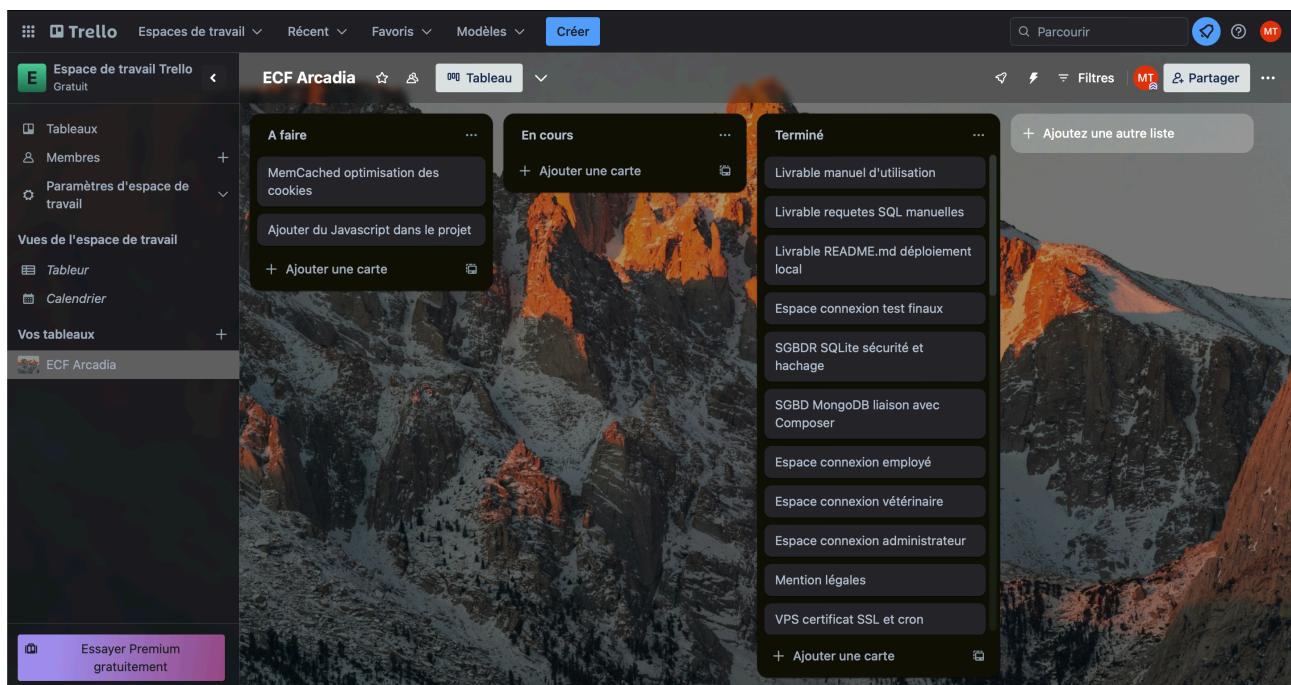
Le diagramme de séquence permet de visualiser de manière séquentielle comment chaque action des acteurs interagit avec le système, clarifiant ainsi le déroulement des processus et des échanges d'informations. Cette représentation temporelle des interactions renforce la conception de l'application en identifiant les scénarios clés et en assurant une meilleure cohérence dans le développement des fonctionnalités.



En comprenant en profondeur les attentes et les exigences des utilisateurs, nous sommes en mesure de concevoir une architecture adaptée qui répond efficacement à leurs besoins spécifiques. Cette approche centrée sur l'utilisateur permet de créer une application web qui offre une navigation intuitive et des fonctionnalités pertinentes pour chaque type d'utilisateur.

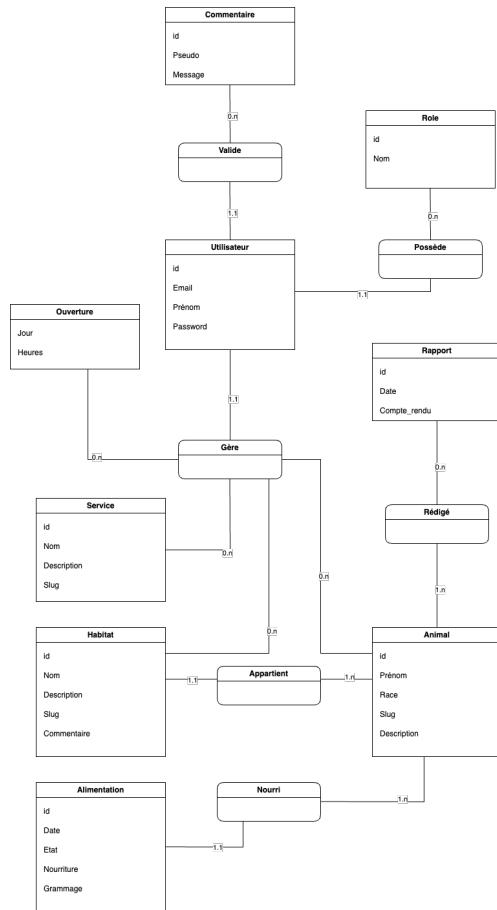
Parallèlement à cette analyse utilisateur, la mise en place d'une gestion de projet rigoureuse est essentielle pour assurer le succès du développement. Une liste détaillée des tâches à réaliser, en cours et terminés est établie, permettant ainsi une planification minutieuse du processus de développement. Cela m'offre une clarté et une visibilité accrues sur l'état d'avancement du projet. Cette approche optimise la gestion des tâches en permettant un suivi efficace de leur progression, tout en facilitant la collaboration grâce à une communication transparente et en temps réel.

En utilisant cette méthode, je favorise une gestion fluide et organisée, tout en assurant une vue d'ensemble des tâches déjà accomplies, ce qui permet de maintenir une dynamique positive.

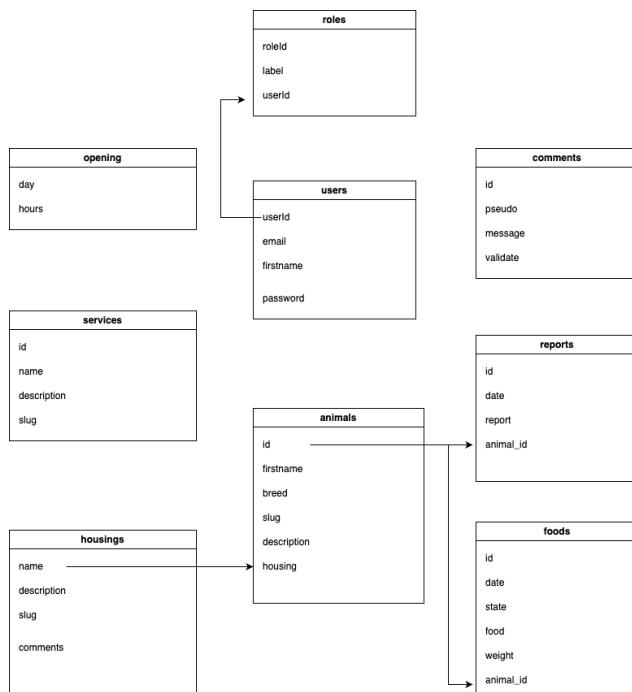


## Conception visuelle et prototypage

Suite à l'analyse des besoins des utilisateurs, j'ai entrepris la création d'un modèle conceptuel de données, dérivé des informations extraites du diagramme de cas d'utilisation.



Ce modèle a été minutieusement affiné au fil du temps pour aboutir à un modèle logique de données robuste, garantissant la cohérence des tables et des relations. Cette approche a été déployée de manière à répondre de manière exhaustive à toutes les fonctionnalités requises par l'application. Grâce à cette méthodologie, la structure de données est non seulement pertinente mais également efficace, permettant une gestion optimale des informations tout au long du processus de développement de l'application.



Une étape décisive du processus de conception a été la création des wireframes et des maquettes de l'application (voir Annexe). Ces représentations visuelles ont été élaborées en tenant compte des formats bureau et mobile, fournissant ainsi une vision précise et exhaustive du produit final. Cette approche a permis une visualisation concrète de toutes les fonctionnalités et interactions prévues, facilitant la communication avec les parties prenantes et garantissant une implémentation cohérente et efficace de l'application.

## Infrastructure et outils de développement

Après avoir élaboré les maquettes de l'application, j'ai entamé la phase de développement. Conscient de ma formation en cours et désirant mettre à profit mes compétences actuelles, j'ai délibérément choisi d'exclure l'utilisation de frameworks et de préprocesseurs. À la place, j'ai opté pour l'utilisation de technologies que je maîtrise déjà, notamment HTML pour la structure, CSS pour le style visuel, JavaScript pour l'interactivité et PHP pour la gestion de la base de données. jQuery assurera la liaison entre la base de données et l'interface utilisateur principalement grâce au protocole AJAX.

En ce qui concerne les bases de données, j'ai décidé d'adopter SQLite, une base de données SQL légère, ainsi que MongoDB, une base de données NoSQL. Cette décision découle de mon désir d'explorer et de me familiariser avec différents types de bases de données. De plus, je suis déjà familier avec MySQL, ce qui m'aide à apprécier rapidement les concepts fondamentaux des bases de données relationnelles.

J'ai choisi ces solutions en fonction des besoins spécifiques du projet, cherchant ainsi à acquérir une expérience variée dans ce domaine.

Pour commencer le développement, sur mon environnement MacOS, j'ai créé un nouveau dossier sur mon bureau en utilisant la commande `mkdir`, que j'ai nommé « ECF-ZOO-ARCADIA ». J'ai ensuite initialisé le projet en lançant la commande `git init` et créé immédiatement un fichier « `.gitignore` » pour exclure les fichiers temporaires indésirables générés par MacOS, tels que « `.DS_Store` ». Cette approche me permettra de développer le projet de manière efficace en utilisant des technologies familières et en organisant proprement le code source.

Par la suite, j'ai utilisé Homebrew pour installer tous les outils nécessaires au développement, notamment SQLite, MongoDB, PHP, PDO (pour interagir avec la base de données de manière sécurisée) et Composer (pour manipuler les données de manière efficace et flexible avec MongoDB). Cette démarche garantit la disponibilité de tous les composants indispensables au développement de l'application web, tout en simplifiant le processus d'installation et de gestion des dépendances.

Cependant, il aurait été pertinent d'envisager l'utilisation d'un moyen de conteneurisation tel que Docker, Podman ou Kubernetes. Cette approche aurait permis de maintenir un environnement de travail propre et isolé, tout en facilitant l'installation et la configuration de certains outils grâce à des images prédéfinies, optimisant ainsi le processus de développement.

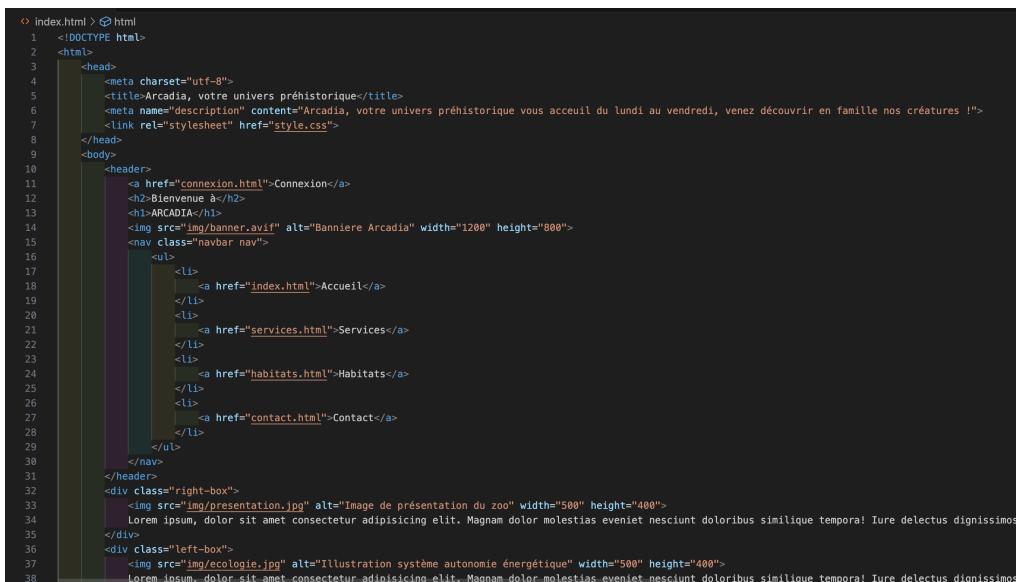
```
Manu@Air-de-Brunella ~ % brew services list
Name          Status  User  File
mongodb-community@4.4  error  3584 root ~/Library/LaunchAgents/homebrew.mxcl.mongodb-community@4.4.plist
mysql          none
php            none
php@8.2        started   Manu ~/Library/LaunchAgents/homebrew.mxcl.php@8.2.plist
podman         none
unbound        none
```

En parallèle, j'ai configuré un serveur Apache pour tester l'application localement, privilégiant XAMPP pour sa facilité d'utilisation et sa compatibilité avec les environnements de développement. Après l'installation, j'ai pris soin de configurer le fichier « php.ini » de XAMPP pour activer les extensions requises, assurant ainsi le bon fonctionnement de l'environnement de développement.

```
/opt/homebrew/etc/php/8.2
Manu@Air-de-Brunella ~ % sudo vim php.ini
```

## Fondements du développement web avec HTML et CSS

Après avoir entamé le processus de développement, j'ouvre mon environnement de développement intégré (IDE), Visual Studio Code, pour commencer à créer ma première page HTML. Avec méthode, j'incorpore progressivement les différents éléments, tels que des images libres de droits et les hyperliens, en veillant à respecter la sémantique des balises et les bonnes pratiques à adopter comme maintenir une bonne indentation pour assurer une bonne lisibilité du code.



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Arcadia, votre univers préhistorique</title>
    <meta name="description" content="Arcadia, votre univers préhistorique vous accueille du lundi au vendredi, venez découvrir en famille nos créatures !">
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <header>
      <a href="connexion.html">Connexion</a>
      <h2>Bienvenue à</h2>
      <h1>ARCADIA</h1>
      
      <nav class="navbar nav">
        <ul>
          <li>
            <a href="index.html">Accueil</a>
          </li>
          <li>
            <a href="services.html">Services</a>
          </li>
          <li>
            <a href="habits.html">Habits</a>
          </li>
          <li>
            <a href="contact.html">Contact</a>
          </li>
        </ul>
      </nav>
    </header>
    <div class="right-box">
      
      Lorem ipsum, dolor sit amet consectetur adipisicing elit. Magnam dolor molestias eveniet nesciunt doloribus similitique tempora! Iure delectus dignissimos
    </div>
    <div class="left-box">
      
      Lorem ipsum, dolor sit amet consectetur adipisicing elit. Magnam dolor molestias eveniet nesciunt doloribus similitique tempora! Iure delectus dignissimos
    </div>
  </body>
</html>
```

Une fois cette étape achevée, je m'assure que mon HTML est conforme aux normes de référencement en me rendant sur le site « <https://validator.w3.org/> ». Cette vérification minutieuse garantit la qualité et la validité de mon code HTML, ce qui est crucial pour le référencement naturel (SEO).

Le SEO (Search Engine Optimization) est un ensemble de pratiques visant à améliorer la visibilité d'un site web dans les résultats des moteurs de recherche comme Google, Bing, ou Yahoo. Lorsque le code HTML est valide et bien structuré, cela facilite la tâche aux moteurs de recherche pour explorer et indexer le contenu du site.

En garantissant la conformité de mon code HTML, je m'assure que les balises sont correctement utilisées, les liens fonctionnent comme prévu, les images sont bien décrites, et les métadonnées sont correctement renseignées. Tout cela contribue à une meilleure indexation de mon site web, ce qui peut potentiellement augmenter sa position dans les résultats de recherche.

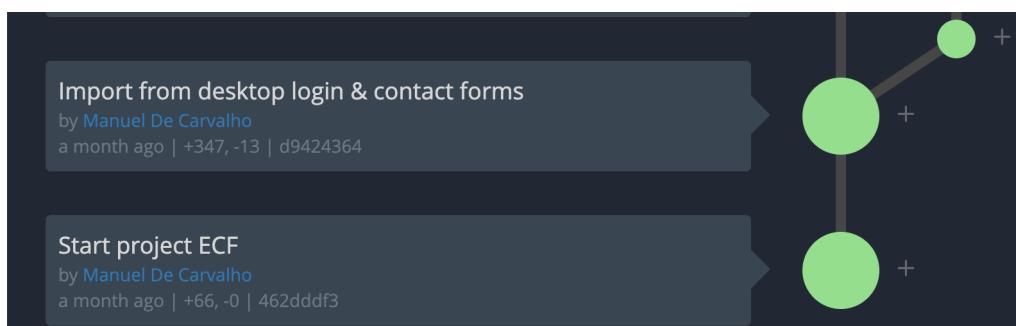
En somme, l'optimisation du code HTML pour le SEO est une étape essentielle dans le processus de développement d'un site web. Cela permet non seulement d'améliorer sa visibilité en ligne, mais aussi d'offrir une expérience utilisateur optimale, car un site bien optimisé est plus facile à naviguer et à trouver pour les utilisateurs et les moteurs de recherche.

Après avoir finalisé ma première page HTML, j'ai entamé le développement de son style en créant une feuille de style CSS. Cependant, je dois admettre que je n'ai pas suivi la méthodologie du "mobile-first", ce qui s'est avéré être une erreur. Cette approche aurait permis de concevoir l'interface en priorisant les dispositifs mobiles, assurant ainsi une expérience utilisateur optimale sur toutes les plateformes. De plus, avant de commencer à styliser ma page, j'ai utilisé un normalizer CSS afin que tous les prérglages des navigateurs soient réinitialisés, assurant ainsi une base uniforme pour le développement du style.

Malgré cette lacune, j'ai consacré un effort considérable à rendre mon application web responsive. J'ai mis en place pas moins de huit media queries dans mes feuilles de style CSS pour adapter l'apparence et le comportement de l'application en fonction de différentes résolutions d'écran. Ces media queries ont été soigneusement ajustées pour garantir que l'application reste fonctionnelle et esthétique, quel que soit le dispositif utilisé par l'utilisateur.

Cette approche itérative m'a permis de développer progressivement l'aspect visuel de chaque page tout en assurant sa fonctionnalité de base. Bien que le choix de ne pas suivre le principe du "mobile-first" ait été une erreur, cette expérience m'a tout de même permis de comprendre l'importance de la conception responsive et de la planification minutieuse de l'interface utilisateur. Suite à la conception de mon CSS, j'ai utilisé un préfixeur CSS, tel que Autoprefixer, qui ajoute automatiquement les préfixes vendeurs nécessaires (-webkit, -moz, -ms, etc.) aux propriétés CSS, assurant ainsi une compatibilité maximale avec différents navigateurs. Cela permet d'éviter les problèmes de compatibilité et de garantir une expérience utilisateur homogène sur tous les appareils et navigateurs.

Pendant le développement de l'interface, j'adopte une approche de versionnement structurée en créant une branche intitulée « dev » sur Git. Cette pratique me permet d'effectuer de nombreux commits de manière organisée, de suivre l'évolution du projet et de revenir à des versions antérieures si nécessaire. En isolant le développement sur une branche dédiée, je peux travailler sur de nouvelles fonctionnalités ou des corrections de bugs sans affecter la branche principale du projet. Cette stratégie de versionnement contribue à maintenir la stabilité du code tout au long du processus de développement.



## Implémentation d'un Carousel interactif

Dans l'optique d'améliorer l'expérience utilisateur sur notre plateforme, l'ajout de Javascript représente une étape significative. Le choix du carousel découle de la volonté de proposer une expérience de consultation des services plus fluide et immersive pour nos utilisateurs.

Pour illustrer le processus de développement de cette fonctionnalité, voici une explication détaillée :

- Définition de la fonctionnalité : Tout d'abord, j'ai identifié la nécessité d'introduire un carrousel sur notre plateforme pour améliorer la navigation et la consultation des services proposés.
- Planification et conception : Après avoir défini le besoin, j'ai planifié les différentes étapes de développement du carrousel. J'ai réfléchi à son emplacement sur la page, à son apparence visuelle et à son fonctionnement interactif.
- Implémentation technique : Pour concrétiser cette fonctionnalité, j'ai commencé par définir une variable pour stocker la position des diapositives dans le carrousel. Ensuite, j'ai mis en place la fonction SlideShow(slidePosition) qui permet la navigation entre les différents éléments identifiés comme des services. Cette fonctionnalité clé offre aux utilisateurs la possibilité de parcourir aisément les divers services proposés, créant ainsi une expérience de consultation plus intuitive.
- Tests et ajustements : Après avoir implémenté le carrousel, j'ai effectué des tests approfondis pour m'assurer de son bon fonctionnement sur différentes plates-formes et navigateurs. J'ai également recueilli des retours d'utilisateurs pour identifier d'éventuels problèmes ou améliorations à apporter.

```
1 let slidePosition = 1;
2 slideShow(slidePosition);
3
4 function plusSlides(n) {
5     slideShow(slidePosition += n);
6 }
7
8 function slideShow(n) {
9     const slides = document.getElementsByClassName("service");
10    if (n > slides.length) {
11        slidePosition = 1;
12    }
13    if (n < 1) {
14        slidePosition = slides.length;
15    }
16    for (let i = 0; i < slides.length; i++) {
17        slides[i].style.display = "none";
18    }
19    slides[slidePosition - 1].style.display = "block";
20 }
```

En résumé, l'ajout du carrousel va au-delà de simplement faciliter la navigation. Il vise également à rendre l'expérience de consultation des services plus interactive et engageante. En permettant aux utilisateurs de visualiser rapidement une variété de services, le carrousel les encourage à explorer davantage notre offre. De plus, l'aspect dynamique du carrousel capte l'attention de l'utilisateur et suscite son intérêt, favorisant ainsi une interaction plus prolongée avec notre plateforme.

## Back-End

### Activité – Type 2 : Développer la partie back-end d'une application web ou web mobile sécurisée

#### Gestion sécurisée des données avec des bases de données

Après avoir établi mon modèle logique de données, j'ai entrepris la création initiale de mes premières tables et collections. Cette phase cruciale a été réalisée en utilisant le terminal, où j'ai exécuté les commandes nécessaires pour mettre en place la structure de base de ma base de données. En m'appuyant sur le modèle logique préalablement établi, cette approche a permis d'assurer la cohérence entre la conception théorique et sa mise en œuvre pratique.

L'utilisation du terminal m'a offert un contrôle direct sur le processus de création des tables et des collections. Cette flexibilité m'a permis d'ajuster et de modifier la structure au besoin, garantissant ainsi son adéquation avec les exigences fonctionnelles évolutives de l'application.

En plus de l'aspect fonctionnel, la sécurité SQL a été au cœur de ma démarche lors de la création des tables et des collections. Chaque commande SQL a été minutieusement examinée pour garantir le respect des principes de sécurité établis. Par exemple, lors de la définition des champs, j'ai veillé à utiliser des types de données appropriés pour éviter les risques de débordement ou de troncature. De plus, j'ai défini des contraintes d'intégrité pour assurer la cohérence des données et prévenir les vulnérabilités potentielles.

```

CREATE TABLE users (
    userId INTEGER PRIMARY KEY AUTOINCREMENT,
    email VARCHAR(50) NOT NULL,
    firstname VARCHAR(50) NOT NULL,
    password_hash VARCHAR(20) NOT NULL);

CREATE TABLE roles (
    roleId INTEGER PRIMARY KEY AUTOINCREMENT,
    label VARCHAR(20) NOT NULL,
    userId INTEGER NOT NULL,
    FOREIGN KEY(userId) REFERENCES users(userId));

```

```

INSERT INTO users (userId, email, firstname, password_hash)
VALUES
    ('', 'josearcadia@hotmail.fr', 'José', 'arcadmin'),
    ('', 'samheldib@gmail.fr', 'Samir', 'test'),
    ('', 'nadegeletelier@outlook.fr', 'Nadège', 'test');

INSERT INTO roles (roleId, label, userId)
VALUES
    ('', 'Administrator', 1),
    ('', 'Employee', 2),
    ('', 'Veterinarian', 3);

```

Les noms des tables et des champs ont également été choisis avec soin pour éviter les conflits et les attaques par injection de code. En prenant ces mesures dès la création des tables et des collections, j'ai établi une base solide pour assurer la sécurité des données tout au long du cycle de vie de l'application. Cette approche proactive en matière de sécurité garantit la confidentialité, l'intégrité et la disponibilité des données, renforçant ainsi la confiance des utilisateurs dans notre plateforme.

```

> use animal_click
switched to db animal_click
> db.animal_click.find()
{ "_id" : ObjectId("65e305ea9026bd5cff94a996"), "id" : 1, "click" : 1 }
> db.animal_click.insertOne({id: 2, click: 1})
{
    "acknowledged" : true,
    "insertedId" : ObjectId("65ec27bb0fba3a785e805129")
}
> db.animal_click.find()
{ "_id" : ObjectId("65e305ea9026bd5cff94a996"), "id" : 1, "click" : 1 }
{ "_id" : ObjectId("65ec27bb0fba3a785e805129"), "id" : 2, "click" : 1 }
>

```

Le processus de création de collection implique l'intégration d'une sécurité dans la base de données MongoDB en ajoutant un utilisateur avec un mot de passe. Cependant, après avoir effectué cette opération, je rencontre une erreur lors du démarrage du service MongoDB, identifiée comme « l'erreur 3584 ». Cette erreur bloque ma progression dans le travail durant quelques jours.

En parcourant différents forums de discussion, j'ai découvert une solution sur un forum coréen (Source : <https://kimbarbie.tistory.com/49>) qui redirige vers un fil de discussion sur Stack Overflow (Source : <https://stackoverflow.com/questions/63562177/mongod-aborts-on-mac>). Ce dernier explique que macOS génère des fichiers temporaires qui perturbent le fonctionnement de MongoDB. Ces fichiers temporaires peuvent causer des conflits ou des problèmes de permissions avec les fichiers nécessaires au bon fonctionnement de MongoDB, tels que le fichier de socket.

Pour résoudre ce problème, je dois exécuter la commande `sudo rm -rf /tmp/mongodb-27017.sock` pour supprimer le fichier de socket MongoDB temporaire.

Ensuite, je redémarre le service MongoDB en utilisant la commande `sudo brew services restart mongod`. Cette séquence d'opérations me permet de réinitialiser correctement les fichiers temporaires et de redémarrer le service MongoDB avec succès, me permettant ainsi de reprendre le travail sans rencontrer l'erreur 3584.

2 Answers

Sorted by: Highest score (default) ▾



Please use one of the following solutions.



18

for Linux



```
$ sudo rm -rf /tmp/mongodb-27017.sock  
$ sudo service mongod start
```



for Mac

```
$ sudo rm -rf /tmp/mongodb-27017.sock  
$ sudo brew services restart mongod
```

```
Manu@Air-de-Brunella ~ % brew services list
Name          Status   User  File
mongodb-community@4.4 error   3584 root ~/Library/LaunchAgents/homebrew.mxcl.mongodb-community@4.4.plist
mysql         none
php           none
php@8.2       started Manu ~/Library/LaunchAgents/homebrew.mxcl.php@8.2.plist
podman        none
unbound       none
Manu@Air-de-Brunella ~ %
```

Après avoir établi la structure de ma base de données, j'ai entrepris l'implémentation de fonctionnalités PHP sécurisées visant à relier cette base de données à mon application web. Cette étape revêt une importance capitale dans le processus de développement, car elle garantit la sécurité et l'intégrité des données manipulées par l'application.

La sécurité SQL étant au cœur de ma démarche, j'ai réalisé une veille technologique sur les vulnérabilités de sécurité, je commence généralement par regarder des vidéastes spécialisés, les forums de sécurité en ligne et les sites web dédiés à la cybersécurité.

Je garde également un œil sur les bases de données de vulnérabilités publiques telles que le CVE (Common Vulnerabilities and Exposures) ainsi que les bases de données de vulnérabilités spécifiques à certains logiciels ou systèmes. Ces sources fournissent des informations détaillées sur les vulnérabilités découvertes, y compris leur gravité, les systèmes affectés et les correctifs disponibles.

En plus de cela, je suis attentif aux avis de sécurité émis par les organismes gouvernementaux, les groupes de recherche en sécurité et les entreprises spécialisées dans la cybersécurité. Ces avis peuvent fournir des informations précieuses sur les menaces émergentes, les attaques en cours et les meilleures pratiques en matière de sécurité.

Pour assurer la sécurité des données, j'ai mis en place plusieurs mesures de protection. Tout d'abord, j'ai intégré des mécanismes de validation des données d'entrée. Cela m'a permis de filtrer et de valider les données fournies par les utilisateurs, afin d'éviter les attaques d'injection SQL. En validant les données dès leur saisie, j'ai pu empêcher l'exécution de requêtes malveillantes qui pourraient compromettre la sécurité de ma base de données.

Parallèlement, j'ai opté pour l'utilisation systématique de requêtes préparées. Cette technique de programmation sécurisée consiste à séparer les données des instructions SQL, ce qui permet à nouveau d'éliminer le risque d'injection SQL. En préparant les requêtes SQL à l'avance et en liant les paramètres de manière sécurisée, j'ai pu garantir que seules les données légitimes seraient utilisées dans mes requêtes, renforçant ainsi la sécurité de mon application contre les attaques potentielles.

Pour renforcer la sécurité de l'application contre les attaques XSS, j'ai pris des mesures pour échapper les caractères spéciaux dans les données utilisateur. En effectuant une échappement approprié des caractères spéciaux tels que « <, >, & » avant d'afficher les données sur les pages web, je m'assure qu'ils ne sont pas interprétés comme des balises HTML ou des entités HTML potentiellement dangereuses. Cette pratique empêche efficacement l'exécution de code JavaScript non autorisé sur le navigateur des utilisateurs et réduit ainsi le risque d'attaques XSS.

En outre, j'ai mis en place des mécanismes de chiffrement pour les données sensibles stockées dans ma base de données. En chiffrant ces données, j'ai assuré leur confidentialité et leur protection contre tout accès non autorisé. Cette couche supplémentaire de sécurité garantit que même en cas de violation de la base de données, les données sensibles restent inaccessibles et illisibles pour les tiers.

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = filter_input(INPUT_POST, 'name', FILTER_SANITIZE_STRING);
    $message = filter_input(INPUT_POST, 'message', FILTER_SANITIZE_STRING);

    $name = htmlspecialchars($name);
    $message = htmlspecialchars($message);
```

En combinant ces différentes mesures de sécurité, j'ai pu garantir que mon application web était robuste et résistante aux attaques potentielles. Cette approche proactive en matière de sécurité témoigne de mon engagement à fournir une expérience utilisateur fiable et sécurisée, tout en préservant l'intégrité et la confidentialité des données manipulées par l'application.

## Administration avancée avec un tableau de bord personnalisé

J'ai conçu un tableau de bord administratif qui offre une panoplie de fonctionnalités à l'administrateur, lui permettant de gérer efficacement divers aspects de l'application. Ces fonctionnalités incluent la possibilité d'ajouter, de modifier et de supprimer des services, des habitats, des animaux et les horaires d'ouverture. De plus, ils peuvent créer de nouveaux utilisateurs, avec un système d'e-mail automatisé intégré pour la notification.

Pour les employés, le tableau de bord offre un accès aux avis laissés par les visiteurs, avec la capacité de valider ces avis pour qu'ils soient affichés sur la page d'accueil de l'application. De plus, les employés peuvent enregistrer les repas et les quantités de nourriture attribués à chaque animal, assurant ainsi une gestion détaillée et précise de leur alimentation.

Pour les vétérinaires, le tableau de bord offre des fonctionnalités spécifiques, telles que la possibilité de laisser des commentaires sur chaque habitat, de soumettre des comptes rendus pour chaque animal, et d'accéder aux informations détaillées sur l'alimentation saisie pour chaque animal par les employés.

De manière à offrir une vue d'ensemble complète, le tableau de bord permet également aux administrateurs de consulter les comptes rendus rédigés par les vétérinaires, ainsi que des statistiques telles que le nombre de consultations par animal. Ces données sont tirées de la collection NoSQL de MongoDB, garantissant une gestion intégrée et complète de l'ensemble des activités du zoo.

Dans le cadre de ma solution pour récupérer l'identifiant de l'animal et le traiter dans ma base de données NoSQL, je me suis confronté à plusieurs étapes cruciales.

Tout d'abord, j'ai mis en place un mécanisme pour vérifier si l'identifiant de l'animal existait déjà dans la base de données MongoDB. Pour cela, j'ai effectué une requête vers la base de données afin de rechercher l'identifiant donné. Si celui-ci était trouvé, cela signifiait qu'il existait déjà dans la base de données et aucune action supplémentaire n'était requise pour sa création.

Ensuite, une fois confirmé que l'identifiant existait déjà, j'ai procédé à l'incrémantation de la valeur de la colonne "click" associée à cet identifiant. J'ai réalisé cette action en mettant à jour le document correspondant dans la collection MongoDB en utilisant l'opérateur d'incrémantation approprié.

En revanche, si l'identifiant n'existe pas dans la base de données, j'ai dû le créer et attribuer la valeur 1 à la colonne "click". Pour ce faire, j'ai inséré un nouveau document dans la collection MongoDB avec l'identifiant spécifié et la valeur 1 pour la colonne "click".

```
$database = $client->selectDatabase("animals_click");
$collection = $database->selectCollection("animals_click");

$animal = $collection->findOne(['id' => (int)$animal_id]);
if ($animal) {
    $collection->updateOne(
        ['id' => (int)$animal_id],
        ['$inc' => ['click' => 1]]
    );
} else {
    $collection->insertOne(['id' => (int)$animal_id, 'click' => 1]);
}
```

Pour implémenter ces actions, j'ai exploré l'utilisation de JQuery et du protocole AJAX. J'ai développé un code qui récupérait l'identifiant de l'animal sélectionné par l'utilisateur à partir de l'interface utilisateur. Ce code utilisait JQuery pour capturer l'identifiant et l'envoyer via une requête POST à la page suivante, où le traitement des données avait lieu ce qui a grandement facilité la manipulation des éléments HTML rendant l'expérience utilisateur plus fluide et réactive, en évitant le rechargeement complet de la page à chaque interaction.

## Tests et déploiement sur serveur en ligne

Une fois toutes les configurations effectuées et répondant aux exigences initiales, je procède à une dernière vérification avec XAMPP avant le déploiement pour m'assurer du bon fonctionnement de l'ensemble. Ensuite, je consolide mes modifications en effectuant un dernier commit que je fusionne avec ma branche principale, assurant ainsi l'intégrité et la traçabilité de mon code.

Passant au travail sur mon VPS via le terminal, je réinitialise l'environnement de développement en réinstallant les composants nécessaires, notamment SQLite, MongoDB, PHP, PDO et Composer, afin de garantir un environnement de production cohérent avec celui de développement. J'installe et configure également Postfix pour garantir l'envoi correct des e-mails, un élément crucial pour assurer la communication avec les utilisateurs de l'application. Cependant, du fait que je ne suis pas propriétaire du nom de domaine, je n'ai pu configurer convenablement les DNS, ce qui entraîne l'envoi d'emails de manière non sécurisée et provoque le blocage de la plupart des fournisseurs de boîtes mails traditionnelles. L'intégration de mécanismes tels que SPF (Sender Policy Framework) et DKIM (DomainKeys Identified Mail) serait nécessaire pour assurer l'envoi sécurisé des emails en authentifiant l'identité de l'expéditeur et en vérifiant l'intégrité des messages envoyés.

Profitant de cette session, je déploie une application de pré-production destiné à tester les fonctionnalités à venir.

Pour cela, je clone deux branches de mon dépôt en utilisant les commandes `git clone` et `git clone -b` dans mon répertoire « docroot » (/var/www). Cela me permet de disposer d'une copie exacte de mon code sur le serveur de pré-production, prêt à être testé.

```
manu@arcadia:/var/www/arcadia/ECF-ZOO-ARCADIA$ ls -la
total 5752
drwxr-xr-x 6 manu daemon 4096 Apr 19 17:31 .
drwxr-xr-x 3 manu daemon 4096 Feb 18 14:37 ..
-rw-r--r-- 1 manu manu 58188 Apr 19 17:31 adminPanel.php
-rw-r--r-- 1 manu manu 5015 Apr 19 17:31 animals.php
-rw-r--r-- 1 manu manu 474 Apr 19 17:31 carousel.js
-rw-r--r-- 1 manu manu 429 Apr 19 17:31 composer.json
-rw-r--r-- 1 manu manu 87694 Apr 19 17:31 composer.lock
-rw-r--r-- 1 manu manu 4752 Apr 19 17:31 contact.php
-rw-r--r-- 1 manu manu 77824 Apr 19 17:31 db.sqlite
-rw-r--r-- 1 manu manu 20125 Apr 19 17:31 employPanel.php
drwxrwxr-x 8 manu daemon 4096 Apr 19 17:31 .git
-rwxrwxr-x 1 manu daemon 9 Feb 18 14:37 .gitignore
-rw-r--r-- 1 manu manu 4761 Apr 19 17:31 housingNext.php
-rw-r--r-- 1 manu manu 4647 Apr 19 17:31 housing.php
drwxrwxr-x 3 manu daemon 4096 Apr 19 17:31 img
-rwxrwxr-x 1 manu daemon 903 Feb 18 14:37 incrementClick.php
-rw-r--r-- 1 manu manu 7252 Apr 19 17:31 index.php
-rw-r--r-- 1 manu manu 4601 Apr 19 17:31 login.php
-rw-r--r-- 1 manu manu 5483532 Apr 19 17:31 "Manuel d'utilisation.pdf"
-rw-r--r-- 1 manu manu 5941 Apr 19 17:31 mentions_legales.php
-rw-r--r-- 1 manu manu 5952 Apr 19 17:31 normalize.css
-rw-r--r-- 1 manu manu 7328 Apr 19 17:31 README.md
-rw-r--r-- 1 manu manu 4551 Apr 19 17:31 services.php
drwxr-xr-x 2 manu manu 4096 Apr 19 17:31 src
-rw-r--r-- 1 manu manu 13579 Apr 19 17:31 style.css
drwxr-xr-x 7 manu manu 4096 Apr 19 17:31 vendor
-rw-r--r-- 1 manu manu 14511 Apr 19 17:31 veterPanel.php
```

Je procède ensuite à l'ajustement de la configuration de mon fichier « `php-fpm.conf` » et de mon « `php.ini` » dans `/etc/php/8.3/fpm`, en veillant à ce que les permissions soient correctement configurées à l'aide des commandes `chmod` et `chown`. Cela garantit que les utilisateurs disposent des droits nécessaires pour lire et écrire dans la base de données SQL, assurant ainsi la sécurité et l'intégrité des données.

Enfin, des sous-domaines sont configurés pour assurer une accessibilité appropriée du site, garantissant ainsi que les utilisateurs peuvent accéder à l'application de manière fiable et sécurisée.

## Simplification des tâches avec des alias personnalisés

L'utilisation d'alias dans le terminal simplifie grandement les tâches répétitives. En accédant au répertoire personnel via la commande `cd /home/manu/`, j'ai ouvert le fichier « `.bashrc` » dans un éditeur de texte tel que Vim. Ensuite, j'ai ajouté les lignes suivantes à mon fichier pour créer des alias :

```
alias ab='/usr/bin/bash /home/manu/repos/scriptsPokemonKingdom/arcadiabuild.sh'  
alias pab='/usr/bin/bash /home/manu/repos/scriptsPokemonKingdom/preprodarcadiabuild.sh'
```

Après avoir défini les alias dans le fichier « `.bashrc` », je me suis dirigé vers le répertoire spécifique où je souhaitais stocker mes scripts personnalisés. Dans ce cas, le répertoire choisi était `/home/manu/repos/scripts`. Une fois à l'intérieur de ce répertoire, j'ai créé deux nouveaux scripts pour automatiser des tâches récurrentes.

Les deux scripts ont pour objectif de simplifier le processus de mise à jour du code depuis GitHub et le redémarrage du serveur nginx. Ces scripts combinent essentiellement les commandes utilisées dans l'alias `gitpull`, mais ils offrent la flexibilité supplémentaire d'être exécutés en tant que scripts indépendants. « `systemctl restart php8.3-fpm.service` » est conçu spécifiquement pour redémarrer le serveur php-fpm. Enfin, « `systemctl restart nginx.service` » redémarre le serveur nginx.

Ces scripts offrent une méthode efficace et rapide pour effectuer des tâches récurrentes avec une seule commande, ce qui simplifie considérablement le flux de travail quotidien et améliore la productivité du développement.

```
#!/bin/bash

cd /var/www/preprod-arcadia.lamphie.moe/ECF-ZOO-ARCADIA;
/usr/bin/git pull;
sudo /usr/bin/systemctl restart php8.3-fpm.service;
sudo /usr/bin/systemctl restart nginx.service;
echo "Last 20 line in php logs"
tail -n 20 /var/log/php8.3-fpm.log
```

Après avoir enregistré les modifications dans le fichier « .bashrc », j'ai rechargé le fichier de configuration en exécutant la commande source `~/.bashrc` pour rendre les alias immédiatement disponibles dans le terminal. Désormais, il me suffit de saisir « ab » (pour Arcadia Build) ou `pab` (Pré-production Arcadia Build) dans le terminal pour effectuer rapidement ces trois actions en une seule commande.

## Sécurisation des échanges avec un certificat SSL

Avant de déployer un certificat SSL sur mon serveur web, je m'assure toujours de suivre une procédure rigoureuse pour garantir un déploiement correct et sécurisé. Toutefois, il convient de noter que le terme SSL est obsolète, et il est préférable de parler de certificat TLS (Transport Layer Security). Ainsi, je veille à vérifier minutieusement la configuration de Nginx pour m'assurer que le virtual host ou le fichier de configuration principal pointe vers le bon répertoire. Cette étape est cruciale car une mauvaise configuration peut entraîner des problèmes de sécurité ou des erreurs de redirection. En utilisant la commande `sudo nginx -t`, je m'assure que la syntaxe de mon fichier de configuration est correcte et qu'il n'y a pas d'erreurs de configuration.

```
manu@arcadia:/etc/nginx/sites-available$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Ensuite, je crée un lien symbolique (symlink) vers le répertoire `/etc/nginx/sites-enabled/`. Cela permet à Nginx de prendre en compte mon fichier de configuration pour le site en question. Ce lien symbolique assure une meilleure gestion des virtual hosts et facilite les mises à jour ou les modifications ultérieures. En utilisant la commande `sudo ln -s`, je lie mon fichier de configuration disponible dans `/etc/nginx/sites-available/` au répertoire `sites-enabled/`.

Une fois que le fichier de configuration est correctement configuré et activé, je vérifie attentivement les permissions des répertoires et des fichiers pour m'assurer qu'ils sont sécurisés et accessibles uniquement aux utilisateurs autorisés. Cela comprend la configuration des permissions de lecture, d'écriture et d'exécution ainsi que la propriété des fichiers et des répertoires. En utilisant des commandes comme `sudo chmod` et `sudo chown`, je m'assure que le serveur web a les droits nécessaires pour accéder aux fichiers du site sans compromettre la sécurité du système.

Une fois toutes ces vérifications effectuées, je redémarre le serveur nginx pour appliquer les modifications. Cela permet de charger le nouveau fichier de configuration et de prendre en compte les modifications apportées au serveur web. En utilisant la commande `sudo systemctl reload nginx`, je m'assure que le serveur redémarre correctement et que les changements sont effectifs.

En parallèle, j'installe Dehydrated, un outil de gestion de certificats SSL, qui facilite la génération et le renouvellement des certificats Let's Encrypt. Pour ce faire, j'exécute la commande `sudo dehydrated -c`, qui vérifie et renouvelle automatiquement les certificats SSL. Cette commande est également utilisée pour générer un nouveau certificat si aucun certificat correspondant à un domaine spécifique n'est trouvé. Je crée un répertoire dédié pour stocker les certificats et un script de déploiement pour automatiser le processus de déploiement des certificats sur mon serveur web.

En outre, je configure une redirection bien connue (well-known) vers mon fichier de configuration pour permettre la vérification de domaine par Let's Encrypt. Cette redirection est cruciale pour le processus de validation ACME (Automated Certificate Management Environment) utilisé par Let's Encrypt. Lorsque Let's Encrypt vérifie la demande de certificat, elle accède à un fichier spécifique dans un répertoire 'well-known' sur mon serveur web. La redirection vers ce répertoire garantit que Let's Encrypt peut accéder au fichier de validation nécessaire pour prouver que je contrôle le domaine en question. Cela assure ainsi que le processus de génération et de renouvellement des certificats SSL se déroule sans accroc, renforçant la sécurité de mon serveur web.

```
#!/bin/bash

# Emplacement des certificats Let's Encrypt
CERT_DIR="/etc/dehydrated/certs"

# Emplacement des fichiers de clé et de certificat pour Nginx
NGINX_SSL_DIR="/etc/nginx/ssl"
DOMAINS_TXT="/etc/dehydrated/domains.txt"

# Itération sur les domaines dans le fichier domains.txt
while IFS= read -r DOMAIN; do
    cp -f "${CERT_DIR}/$DOMAIN/fullchain.pem" "${NGINX_SSL_DIR}/$DOMAIN.crt"
    cp -f "${CERT_DIR}/$DOMAIN/privkey.pem" "${NGINX_SSL_DIR}/$DOMAIN.key"
done < "$DOMAINS_TXT"

# Rechargement de la configuration nginx
systemctl reload nginx
```

Je configure également un CRON pour planifier le renouvellement automatique des certificats chaque jour à 2 heures du matin, en utilisant la commande *sudo crontab -e*.

```
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
  
# Dump bdd  
0 3 * * * /usr/bin/bash /home/manu/repos/scriptsPokemonKingdom/backup.sh  
  
# Let's encrypt  
0 2 * * 0 /usr/bin/dehydrated -c -f /etc/dehydrated/config && systemctl reload nginx.service
```

Une fois que le certificat SSL est installé avec succès, je l'ajoute à mon virtual host en mettant à jour mon fichier de configuration. Dans ce fichier, je configure une redirection automatique de toutes les requêtes HTTP vers HTTPS pour garantir une connexion sécurisée. Je m'assure également que les paramètres de sécurité SSL sont correctement configurés pour garantir la confidentialité et l'intégrité des données échangées entre le serveur et le client.

```
server {
    listen 80;

    root /var/www/preprod-arcadia.lamphie.moe/ECF-ZOO-ARCADIA;
    index index.html index.htm index.nginx-debian.html index.php;

    server_name preprod-arcadia.lamphie.moe;

    location ^~ /.well-known/acme-challenge {
        alias /var/www/dehydrated;
    }

    location / {
        return 301 https://$host$request_uri;
    }
}

server {
    listen 443 ssl http2;
    server_name preprod-arcadia.lamphie.moe;
    root /var/www/preprod-arcadia.lamphie.moe/ECF-ZOO-ARCADIA;
    index index.php;

    # Let's Encrypt SSL
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_certificate /var/lib/dehydrated/certs/preprod-arcadia.lamphie.moe/fullchain.pem;
    ssl_certificate_key /var/lib/dehydrated/certs/preprod-arcadia.lamphie.moe/privkey.pem;

    location / {
        try_files $uri $uri/ /index.php?q=$uri$args;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php-fpm.sock;
    }

    location ~ /\.ht {
        deny all;
    }
}
```

En suivant cette procédure étape par étape, je m'assure que le déploiement du certificat SSL est correctement effectué et que mon site web bénéficie d'une connexion sécurisée et fiable.

Bien que l'utilisation d'un playbook Ansible aurait été pertinente pour automatiser la génération d'alias, de fichiers de configuration (vhost), de certificats SSL et de redirection HTTPS, la décision de ne pas l'adopter est motivée par plusieurs raisons. Tout d'abord, bien que Ansible soit un outil puissant de gestion de configuration, il peut parfois masquer les détails des processus sous-jacents. En détaillant manuellement chaque étape du déploiement, on favorise la transparence et une compréhension plus approfondie des tâches impliquées.

Cette approche manuelle permet également d'acquérir une vision claire de chaque composant de l'infrastructure serveur, ce qui peut être précieux pour le dépannage et la maintenance à long terme. En comprenant en profondeur comment chaque élément est configuré et déployé, on est mieux préparé à faire face à des scénarios imprévus ou des exigences spécifiques du projet.

## Conclusion

Durant ce projet, j'ai été confronté à une multitude de défis, chacun offrant une opportunité d'apprentissage. Mon expérience m'a enseigné l'importance de l'adaptabilité, de la patience et de la résolution de problèmes. À plusieurs reprises, j'ai dû faire preuve de créativité pour trouver des solutions innovantes à des problèmes techniques complexes. Cette capacité à m'adapter aux circonstances changeantes et à persévérer face aux obstacles s'est avérée essentielle pour mener ce projet à bien.

Je reconnaiss également que ce projet n'a pas été exempt d'erreurs ou d'imperfections. Cependant, chaque erreur a été une opportunité d'apprentissage, me permettant de mieux comprendre les concepts sous-jacents et d'améliorer mes compétences techniques. Je suis également conscient de l'importance de la refactorisation du code pour améliorer sa maintenabilité et sa lisibilité, une pratique que je me dois d'intégrer dans mes processus de développement pour assurer la qualité et l'évolutivité du code.

En rétrospective, je réalise l'importance de la gestion du temps et des priorités. J'ai appris à hiérarchiser mes tâches de manière à maximiser mon efficacité et à optimiser mes efforts. Cette capacité à gérer efficacement mon temps et mes ressources a été cruciale pour maintenir la progression du projet, même face à des défis imprévus.

Je suis profondément motivé à poursuivre cette trajectoire d'apprentissage continu. Je suis déterminé à explorer de nouveaux langages de programmation, à maîtriser davantage les frameworks existants et à me tenir au courant des dernières avancées technologiques. Ce projet n'est que le début d'un voyage passionnant vers l'excellence technique et professionnelle.

## Annexes

### Charte graphique

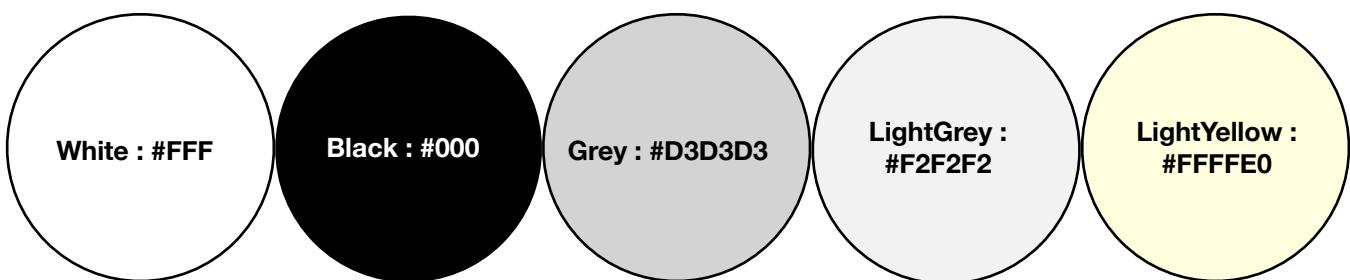
Font :

#### Source Sans Pro Font Family

 By Adobe



Colors :



## Wireframes de l'application web

### Bureau :

The screenshot shows a wireframe interface for a desktop application. The left sidebar lists components under categories like "Wireframe - animals", "Wireframe - housing\_Next", and "Wireframe - index". The main canvas displays six wireframe prototypes for different sections: "ARCADIA", "SERVICES", "HABITAT 01", "HABITAT 02", "HABITAT 03", "ANIMAL", "CONTACT", and "CONNEXION". The right sidebar contains design and prototype settings, local variables, local styles, and export options.

### Mobile :

The screenshot shows a wireframe interface for a mobile application. The left sidebar lists components under categories like "Wireframe - login", "Wireframe - contact", and "Wireframe - index". The main canvas displays six wireframe prototypes for different sections: "ARCADIA", "SERVICES", "HABITAT 01", "HABITAT 02", "HABITAT 03", "ANIMAL", "CONTACT", and "CONNEXION". The right sidebar contains design and prototype settings, local variables, local styles, and export options.

## Mockup de l'application web

### Bureau :

The screenshot shows a desktop application interface in a design tool. The top bar includes standard icons for file operations, a search bar, and a title 'Drafts / Mockup desktop ECF'. On the left, a sidebar lists components such as '# adminPanel\_delete', '# adminPanel\_add', '# adminPanel', '# Contact', '# Login\_role', '# Login', '# Animals', '# Housing Next', '# Housing', '# Services 02', '# Services 01', and '# Index'. The main area displays a grid of 12 screen prototypes. The first two columns contain 'Login' and 'Contact' screens. The third column contains 'Index' and 'Services 01' screens. The fourth column contains 'Services 02' and 'Housing' screens. The fifth column contains 'Housing N...' and 'Animals' screens. The sixth column contains 'adminPanel' and 'adminPan...' screens. The right side of the interface features a sidebar with tabs for 'Design' and 'Prototype', and sections for 'Page' (with a color palette), 'Local variables', 'Local styles', and 'Export'.

### Mobile :

The screenshot shows a mobile application interface in a design tool. The top bar includes standard icons for file operations, a search bar, and a title 'Drafts / Mockup mobile ECF'. On the left, a sidebar lists components such as '# iPhone 13 housing\_Next', '# iPhone 13 housing\_Next', '# iPhone 13 housing', '# iPhone 13 services 02', '# iPhone 13 login\_role', '# iPhone 13 login', '# iPhone 13 contact', '# iPhone 13 services', and '# iPhone 13 index'. The main area displays a grid of 12 screen prototypes, each labeled with an iPhone 13 placeholder. The screens include labels like 'ARCADA', 'SERVICES', 'HABITATS', 'MARAIS', 'JUNGLE', 'SAVANE', 'CONTACT', 'CONNEXION', and 'NAME'. The right side of the interface features a sidebar with tabs for 'Design' and 'Prototype', and sections for 'Page' (with a color palette), 'Local variables', 'Local styles', and 'Export'.

### Serveur

PHP 8.3.3

Extension PHP : PDO

Extension MongoDB : Composer

### Front-End

HTML 5

CSS 3

Javascript

### Back-End

PHP 8.3.3 avec PDO

SQLite

MongoDB

### Déploiement

VPS personnel