

# Déploiement de l'Application tout au long du projet

## Contexte

Le déploiement du site web du zoo Arcadia a été une étape cruciale. Passer d'un environnement local à un hébergeur en ligne a apporté son lot de défis et d'apprentissages, notamment avec Hostinger, l'hébergeur choisi.

## Défis rencontrés

Le transfert initial semblait relativement simple : le site fonctionnait correctement en local. Cependant, des problèmes inattendus ont surgi lors de la migration sur Hostinger. Par exemple, MongoDB, configuré localement, n'était pas pris en charge directement par Hostinger. J'ai dû opter pour une version en ligne de MongoDB, ce qui a simplifié la synchronisation et réduit le poids de l'application, évitant ainsi des configurations lourdes pour chaque installation locale.

La configuration SMTP pour l'envoi d'e-mails a été un autre défi. En local, j'avais ignoré cette partie, mais une fois le site déployé, l'envoi de messages aux utilisateurs est devenu essentiel. J'ai configuré une adresse e-mail liée à l'hébergeur et effectué des tests pour m'assurer du bon fonctionnement, rendant ainsi la fonctionnalité de contact pleinement opérationnelle.

## Enseignements

Ce déploiement m'a permis de mieux comprendre les différences entre le développement en local et en ligne. Il est crucial de prendre en compte les contraintes d'hébergement dès le début du projet, notamment en ce qui concerne la compatibilité des bases de données et la sécurité des communications (comme SMTP).

Pour l'avenir, je serai plus vigilant quant aux technologies choisies et à leur compatibilité avec l'environnement d'hébergement final. Cela m'a conforté dans l'idée qu'une approche plus modulaire et flexible est essentielle pour simplifier le déploiement et la maintenance.

## Choix technologiques et réflexions

Pour ce projet, j'ai décidé de ne pas utiliser de frameworks particuliers, afin de m'entraîner à tout développer moi-même. Cela a été très formateur, mais je reconnais que cela a entraîné une perte de temps sur certains aspects. À l'avenir, je pense utiliser des bibliothèques comme Bootstrap pour gagner du temps, notamment pour le design et l'interaction utilisateur.

J'ai également été attiré par des technologies comme Vue.js, mais je voulais avant tout maîtriser les bases. J'aime relever des défis et apprendre en faisant, même si ce n'est pas toujours la solution la plus efficace. Je prévois de développer une structure particulière dans un prochain projet, incluant un document d'affichage, un document tampon pour gérer les variables, et un serveur de calcul, afin d'expérimenter une approche plus modulaire.

Pour le backend, j'ai principalement utilisé PHP. C'est une technologie que je maîtrise bien et qui était adaptée aux besoins du projet, même si j'envisage d'explorer Node.js pour des projets plus dynamiques. J'ai utilisé PHPMyAdmin pour la gestion des bases de données par confort, mais je reconnais que MongoDB offre des avantages, notamment en termes de flexibilité.

Ce projet m'a conforté dans l'idée de bien réfléchir aux choix technologiques dès le départ, et de ne pas se laisser guider uniquement par ses habitudes. Je compte m'ouvrir à de nouvelles technologies pour optimiser la performance et la maintenabilité de mes futurs projets.

## Procédure d'installation

Pour installer le projet, commencez par accéder au dépôt GitHub contenant l'ensemble des fichiers nécessaires, y compris la base de données et les documents d'accès. Il est impératif de lire le fichier README.md qui contient toutes les informations pour l'installation.

Voici les étapes à suivre :

1. **Création de la base de données** : Créez une base de données dans PHPMyAdmin, en utilisant la version 7 ou plus. Suivez la nomenclature indiquée dans le README. Importez soit le fichier `all-in-one.sql`, soit les trois fichiers distincts (`1.sql`, `2.sql`, `3.sql`) si vous avez des restrictions d'importation.
2. **Configuration de l'hébergement** : Configurez votre hébergeur (local ou en ligne). Si vous utilisez un hébergeur local comme XAMPP, assurez-vous de lier le bon dossier au serveur Apache pour accéder au site. Pensez également à configurer le serveur SMTP pour permettre l'envoi d'e-mails, notamment sur les pages de contact et d'administrateur.
3. **Transfert des fichiers** : Mettez tous les fichiers du dossier du site à héberger dans l'emplacement désigné par votre hébergeur. N'oubliez pas de configurer les paramètres de connexion à la base de données (`localhost`, nom d'utilisateur, mot de passe) dans l'ensemble des pages qui requièrent une connexion (les pages navigantes PHP du site) en fonction des paramètres de votre hébergeur (sur une solution local, par défaut vous n'avez rien à configurer avec Xamp, l'utilisateur par défaut étant "root" sans mot de passe).
4. **Vérification** : Passez en revue toutes les pages du site pour vérifier la connexion à la base de données et le bon fonctionnement des fonctionnalités.

En suivant ces étapes, le site devrait être opérationnel. Pour un déploiement en ligne, configurez également le serveur SMTP pour garantir le fonctionnement des fonctionnalités de contact.