

.IPSSI



L'impact de WordPress par rapport à une approche PHP

Comment la refonte ou la création d'un site en utilisant
WordPress se compare-t-elle à l'utilisation de PHP, HTML et
JavaScript ?

Aurélien MONPIERRE

IPSSI- 2023-2024 - ASI – Bachelor Développeur Fullstack & DevOps

Supervisé par le Référent : Clément DOUGE

Certificat d'authenticité :



CERTIFICAT D'AUTHENTICITE DE MON MEMOIRE

Ce document rempli et signé par l'étudiant doit être inséré dans tous les mémoires après la page de garde

Je, soussigné(e) : NOM et Prénom

MONPIERRE Aurélien

Etudiant(e) de : Formation et année

2023-2024

Etablissement :

Ecole IPSSI Paris

Certifie que le texte soumis ne comporte aucun passage ou schéma copié sans qu'il soit explicitement fait référence aux sources.

Certifie ne pas avoir dépassé le taux de plagiat autorisé (15% maximum)

Fait à Paris le 12/10/2023

Signature :

1

École IPSSI : INSTITUT PRIVÉ SUPÉRIEUR DES SYSTÈMES D'INFORMATION
SARL au capital social de 105 000 € - RCS PARIS : 420 793 705 - CODE NAF 8559A - n° d'activité 11753362175 - TVA FR 41 420 793 705 - UAI 075724P
Paris : 25, rue Claude Tillier - BP 20024 - 75560 Paris Cedex 12 - Tél : 01.55.43.26.65
Marne-la Vallée : 14, avenue de l'Europe - 77144 Montévrain - Tél : 01.81.80.77.10
SQY : 8, rue Germain Soufflot 78180 Montigny-le Bretonneux - Tél : 01.81.80.38.50 - RCS Versailles : 852 544 923 - n° d'activité 11788427678

Remerciements

Tout d'abord, un grand merci à Clément DOUGE. Son management m'a vraiment aidé à m'intégrer au sein de l'entreprise. Grâce à lui, j'ai rapidement trouvé mes marques, et sa bonne humeur a créé un environnement de travail à la fois stimulant et agréable.

Je souhaite également exprimer ma gratitude à Audrey PONCEAU, ma collègue. Son soutien et son professionnalisme m'ont non seulement facilité l'intégration dans l'équipe. Sa bonne humeur a été une vraie source de motivation pour moi.

Je n'oublie pas Alexandre DUPAS, qui m'a beaucoup apporté, notamment sur le travail des sites web. Même si notre collaboration a été relativement courte, j'ai appris énormément grâce à son expertise.

Un grand merci aussi à Émilie BOUCART pour son aide précieuse concernant ma poursuite d'études, ainsi que pour son soutien constant lors de mon intégration dans l'entreprise.

Enfin, je tiens à exprimer ma reconnaissance envers mes enseignants. Nathan Laury, malgré un emploi du temps chargé avec de nombreux élèves à suivre, a toujours trouvé du temps pour m'accompagner dans la rédaction de ce mémoire.

Je remercie aussi Florence Baudry pour son implication et son accompagnement au sein de l'IPSSI, ainsi que pour son soutien constant tout au long de mon parcours scolaire.

À toutes ces personnes, je dis un grand merci.

Résumé

Dans ce mémoire, j'ai voulu explorer les différences entre l'utilisation de WordPress et le développement en open-source, à travers une étude comparative basée sur un projet concret que j'ai réalisé chez Partenor Digital.

Ce sujet m'a particulièrement intéressé en raison de mon expérience chez Partenor Digital, une filiale du Groupe Partenor spécialisée dans la transformation numérique. En tant que développeur web, j'ai eu l'occasion de travailler sur la refonte du site "Campus", qui est hébergé sur WordPress. Cela m'a permis de me familiariser avec les points forts et les limites de cette plateforme. En parallèle, j'ai aussi participé à des projets nécessitant un développement sur mesure en PHP. Ces différentes expériences m'ont amené à me poser des questions sur les critères à prendre en compte lorsqu'on doit choisir entre une solution clé en main comme WordPress et une approche de développement plus traditionnelle.

Ce mémoire cherche à répondre à deux questions principales : comment la refonte ou la création d'un site avec WordPress se compare-t-elle à un développement en PHP, HTML, et JavaScript ? Et dans quel cas choisir l'une ou l'autre de ces méthodes pour créer des formations en ligne ? Pour y répondre, j'ai fait une analyse détaillée des deux approches, en prenant en compte des aspects comme le coût, la flexibilité, la sécurité, la performance, et la maintenance.

Abstract

In this thesis, I aimed to explore the differences between using WordPress and open-source development through a comparative study based on a concrete project I carried out at Partenor Digital.

This topic particularly interested me due to my experience at Partenor Digital, a subsidiary of the Partenor Group specializing in digital transformation. As a web developer, I had the opportunity to work on the redesign of the "Campus" website, which is hosted on WordPress. This allowed me to become familiar with the strengths and limitations of this platform. At the same time, I also participated in projects requiring custom development in PHP. These different experiences led me to question the criteria that should be considered when choosing between a turnkey solution like WordPress and a more traditional development approach.

This thesis seeks to answer two main questions: how does redesigning or creating a site with WordPress compare to development using PHP, HTML, and JavaScript? And in which cases should one choose either method to create online training? To answer these questions, I conducted a detailed analysis of the two approaches, taking into account aspects such as cost, flexibility, security, performance, and maintenance.

Table des matières

Certificat d'authenticité :	2
Remerciements	3
Résumé	4
Abstract	5
Introduction	9
Explication du Sujet	9
Problématique	10
Annonce du Plan et Objectif	10
Partie 1 : Le Sujet	11
Introduction à WordPress et les Systèmes de Gestion de Contenu (CMS)	11
Qu'est-ce qu'un CMS ?	11
Fonctionnalités Clés d'un CMS	11
Avantages et Limites d'un CMS	12
Exemples de CMS Populaires	12
Introduction à Kinsta : Hébergement WordPress de Nouvelle Génération	13
Caractéristiques de Kinsta	13
Exemples de Thèmes et Constructeurs de Pages WordPress : Divi et Elementor	13
Rôle des Langages PHP, JavaScript, et HTML dans le Développement Web	14
Benchmark des Solutions et Analyse des Acteurs du Marché	14
Partie 2 : Cas Réel - Présentation de l'Entreprise : Partenor Group, Filiale : Partenor Digital, Site Campus	15
Introduction à Partenor Group	15
Structuration du Groupe	15
Direction et Leadership	16
Partie 3 - Comparaison des Approches PHP et WordPress : Un Cas Pratique	19
Accessibilité et Compatibilité Multiplateforme	35
SEO et Référencement	36
Gestion des Utilisateurs et Permissions	37
Lancement et Maintenance	38

Bilan	40
Amélioration des Performances et de l'Expérience Utilisateur	40
Gestion de Contenu Simplifiée	40
Défis Techniques et Solutions	40
Sécurité et Maintenance	41
Mon Expérience Personnelle	41
Évolutivité et Adaptabilité	41
Collaboration et Gestion de Projet	42
Retours des Utilisateurs et Itérations Futures	42
Conclusion	42
Annexes	44

Liste des abréviations :

CMS (Système de Gestion de Contenu) :

Un CMS est un logiciel en ligne qui permet de créer, gérer et modifier un site web de manière simple, sans nécessiter de compétences techniques en programmation.

PHP (Hypertext Preprocessor) :

PHP est un langage de script côté serveur utilisé pour développer des pages web dynamiques et interactives. Il est largement utilisé pour la gestion de contenu et la communication avec les bases de données.

HTML (HyperText Markup Language) :

HTML est le langage de balisage standard utilisé pour structurer et afficher le contenu d'une page web dans un navigateur.

JS (JavaScript) :

JavaScript est un langage de programmation qui permet d'ajouter des fonctionnalités interactives aux pages web, comme des animations, des formulaires dynamiques, ou la gestion des événements.

SEO (Search Engine Optimization) :

Le SEO regroupe l'ensemble des techniques visant à améliorer la visibilité d'un site web sur les moteurs de recherche, en optimisant le contenu et les aspects techniques du site.

API (Interface de Programmation d'Applications) :

Une API est un ensemble de règles qui permet à différentes applications de communiquer entre elles, facilitant l'intégration de services et de données.

SSL (Secure Sockets Layer) :

SSL est un protocole de sécurité qui chiffre les communications entre un site web et ses utilisateurs pour protéger les données échangées.

CDN (Réseau de Distribution de Contenu) :

Un CDN est un réseau de serveurs répartis dans le monde entier qui permet de distribuer le contenu d'un site web de manière rapide et efficace, en rapprochant les données des utilisateurs finaux.

ORM (Mapper Objet-Relationnel) :

Un ORM est un outil qui permet de convertir les données entre une base de données relationnelle et des objets dans un langage de programmation, simplifiant ainsi l'interaction avec la base de données.

CI/CD (Intégration Continue/Déploiement Continu) :

CI/CD est une méthode de développement qui automatise l'intégration et le déploiement des mises à jour logicielles, garantissant des livraisons rapides et fiables.

Introduction

Dans un monde où **43% des sites web sont propulsés par WordPress**¹, on ne peut plus se demander si cet outil est populaire, mais plutôt pourquoi autant d'entreprises continuent de coder manuellement en PHP. Ce chiffre montre clairement la domination de WordPress, mais il cache aussi une certaine complexité : malgré sa large adoption, les développeurs utilisent encore des langages comme PHP, HTML et JavaScript pour des projets bien spécifiques.

Explication du Sujet

J'ai choisi de me pencher sur l'impact de WordPress comparé à une approche PHP pour plusieurs raisons, à la fois professionnelles et techniques. En tant qu'alternant dans une entreprise qui utilise principalement WordPress pour gérer ses sites web, j'ai eu l'occasion d'explorer en profondeur les avantages et les inconvénients de cette plateforme. La refonte d'un de leurs sites est un exemple parfait pour illustrer ces comparaisons.

WordPress, en tant que CMS largement utilisé, propose une multitude d'outils et de plugins qui simplifient la création et la gestion de sites web, sans avoir besoin de compétences techniques poussées en programmation. C'est cette simplicité d'utilisation qui pousse beaucoup d'entreprises, y compris la mienne, à adopter cette solution. Mais cela soulève aussi des questions concernant la flexibilité et la performance par rapport à une méthode plus traditionnelle utilisant PHP, HTML, et JavaScript.

D'un autre côté, une approche personnalisée avec PHP permet d'avoir une flexibilité maximale et un contrôle total sur les fonctionnalités et les performances du site. PHP, en tant que langage de script côté serveur, offre des possibilités de personnalisation infinies, ce qui est particulièrement utile pour des projets complexes ou qui nécessitent des fonctionnalités spécifiques, absentes des plugins WordPress.

¹ Selon <https://w3techs.com/technologies/details/cm-wordpress>

Problématique

Comment la refonte ou la création d'un site en utilisant WordPress se compare-t-elle à l'utilisation de PHP, HTML et JavaScript ?

Annonce du Plan et Objectif

Dans ce mémoire, nous allons explorer et comparer ces deux approches en nous concentrant sur des aspects essentiels comme le coût de développement, le temps de déploiement, la sécurité, et la maintenabilité des sites web. Par exemple, nous verrons que WordPress, bien qu'il soit rapide à mettre en place, peut présenter des vulnérabilités si les plugins ne sont pas bien gérés, tandis qu'une solution personnalisée en PHP peut offrir une meilleure sécurité, mais avec un développement plus long et potentiellement plus coûteux.

L'objectif de ce mémoire est de fournir des recommandations basées sur des exemples concrets observés dans mon entreprise, afin d'évaluer l'efficacité de chaque approche et déterminer laquelle est la plus adaptée selon les besoins spécifiques de l'entreprise.

Partie 1 : Le Sujet

Introduction à WordPress et les Systèmes de Gestion de Contenu (CMS)

WordPress, lancé en 2003, est aujourd'hui le système de gestion de contenu (CMS) le plus utilisé au monde, propulsant environ 43% des sites web. Sa popularité s'explique par sa facilité d'utilisation, sa flexibilité, et la richesse de son écosystème de thèmes et de plugins, qui permettent de personnaliser les sites web en fonction des besoins spécifiques des utilisateurs.

Qu'est-ce qu'un CMS ?

Un CMS, ou système de gestion de contenu, est un logiciel qui permet de créer, gérer et modifier du contenu numérique de manière intuitive. Ces systèmes séparent la gestion du contenu de la présentation et de la logique applicative, rendant ainsi la création et la maintenance de sites web accessibles même à ceux qui ne sont pas développeurs. Un CMS comme WordPress propose une interface utilisateur simple à prendre en main, une gestion centralisée de différents types de contenu (articles, images, vidéos), et des fonctionnalités supplémentaires grâce aux plugins et extensions.

Fonctionnalités Clés d'un CMS

Interface Utilisateur Intuitive : Les CMS comme WordPress permettent de gérer du contenu sans avoir besoin d'écrire de code, grâce à des éditeurs WYSIWYG ("What You See Is What You Get"). Cela simplifie la mise en forme et l'organisation du contenu, rendant ces tâches accessibles à tous.

Gestion de Contenu : Les CMS facilitent l'organisation et la gestion de différents types de contenu, avec des options pour catégoriser, taguer et rechercher des informations rapidement, ce qui est pratique pour garder un site bien structuré.

Modèles et Thèmes : Les utilisateurs peuvent personnaliser l'apparence de leur site en utilisant des thèmes, qui offrent une structure prédéfinie mais adaptable aux besoins spécifiques de l'entreprise ou du projet.

Plugins et Extensions : WordPress propose un large éventail de plugins (voir Annexe 1) qui ajoutent des fonctionnalités variées, comme le SEO, les formulaires de contact, ou encore les galeries de photos, le tout sans nécessiter de compétences en développement.

Gestion des Utilisateurs et des Permissions : WordPress permet de créer plusieurs comptes utilisateurs avec des rôles spécifiques (administrateur, éditeur, auteur), ce qui est très utile pour les sites collaboratifs où plusieurs personnes travaillent sur le contenu.

Publication et Workflow : Les outils de planification de publication, de gestion des brouillons, et de révision offrent un contrôle précis sur ce qui est publié, ce qui est particulièrement important pour les équipes éditoriales qui doivent s'assurer de la qualité du contenu.

Avantages et Limites d'un CMS

Un CMS comme WordPress présente plusieurs avantages, notamment sa facilité d'utilisation, sa flexibilité, et sa capacité à s'adapter aux besoins changeants de l'entreprise. Ces atouts en font un outil idéal pour une grande variété de projets. Cependant, WordPress peut montrer ses limites lorsqu'il s'agit de performance, de sécurité, ou de personnalisation pour des sites plus complexes ou très spécifiques. Dans ces cas-là, une approche plus technique, utilisant des langages comme PHP, peut s'avérer plus adaptée pour répondre aux exigences particulières du projet.

Exemples de CMS Populaires

- **WordPress** : Idéal pour des blogs, des sites vitrines, et des sites e-commerce, grâce à son vaste écosystème de thèmes et de plugins qui facilitent la personnalisation et l'ajout de fonctionnalités.
- **Joomla** : Bien adapté pour des sites multilingues et complexes, offrant une gestion robuste des utilisateurs et des fonctionnalités avancées pour les sites nécessitant une structure plus sophistiquée.
- **Drupal** : Souvent préféré pour des sites à grande échelle qui nécessitent une personnalisation poussée et une gestion avancée des permissions, idéal pour les projets qui demandent une grande flexibilité.

- **Magento** : Spécialisé dans le commerce électronique, Magento propose des capacités avancées pour la gestion de catalogues de produits, ce qui en fait un choix privilégié pour les entreprises cherchant à développer des boutiques en ligne performantes.

-

Introduction à Kinsta : Hébergement WordPress de Nouvelle Génération

Fondé en 2013, Kinsta est un service d'hébergement WordPress haut de gamme qui utilise l'infrastructure de Google Cloud Platform. Conçu pour offrir des performances optimales aux sites WordPress, Kinsta se distingue par sa sécurité renforcée, sa rapidité, et son évolutivité. C'est une solution d'hébergement idéale pour les entreprises de toutes tailles qui recherchent une plateforme fiable et performante pour leurs sites WordPress.

Caractéristiques de Kinsta

- **Infrastructure Google Cloud Platform** : Kinsta s'appuie sur l'infrastructure de Google Cloud, ce qui assure une haute disponibilité, une faible latence, et des vitesses de chargement optimisées pour les sites WordPress.
- **Technologies Modernes** : Kinsta utilise des technologies avancées comme Nginx, PHP 8.0+, et MariaDB, garantissant des performances fiables et une compatibilité avec les dernières normes web.
- **Sécurité Renforcée** : Kinsta intègre des pare-feux, propose des certificats SSL gratuits, et effectue des sauvegardes automatiques quotidiennes, offrant ainsi une protection solide pour les sites hébergés.
- **Support Client et Interface Intuitive** : Kinsta offre un support client disponible 24/7 et met à disposition une interface utilisateur conviviale, facilitant la gestion des sites pour les utilisateurs.

Exemples de Thèmes et Constructeurs de Pages WordPress : Divi et Elementor

- **Divi** : Développé par Elegant Themes, Divi est un thème WordPress très populaire qui propose un constructeur de pages visuel intuitif. Il permet de créer des mises en page sophistiquées sans nécessiter de compétences en

codage, ce qui le rend accessible à un large public. **Divi** n'est plus seulement un thème comme les autres, mais un véritable « **Theme Builder** ». Il permet donc aux utilisateurs de créer un site ayant un webdesign sur mesure sans grandes connaissances du codage en HTML et CSS.²

- **Elementor** : Lancé en 2016, Elementor est un constructeur de pages pour WordPress qui permet de créer des designs avancés grâce à une interface glisser-déposer. Il offre une personnalisation approfondie et des intégrations avec d'autres plugins WordPress, ce qui en fait un outil puissant pour les utilisateurs cherchant à créer des sites sur mesure.

Rôle des Langages PHP, JavaScript, et HTML dans le Développement Web

Pour les projets nécessitant une personnalisation complète et un contrôle total, le développement avec PHP, JavaScript (JS) et HTML reste incontournable. Chaque langage a un rôle précis : HTML sert à structurer le contenu, JavaScript ajoute de l'interactivité, et PHP gère la logique côté serveur. Ensemble, ces langages permettent de créer des sites web dynamiques et sur mesure, capables de répondre à des besoins complexes que les CMS comme WordPress ne peuvent pas toujours satisfaire.

Benchmark des Solutions et Analyse des Acteurs du Marché

Le choix entre WordPress, d'autres CMS, et des solutions de développement libre comme PHP dépend de nombreux facteurs, comme les besoins spécifiques du projet, les compétences de l'équipe, et les exigences en matière de personnalisation, de sécurité, et de performance. WordPress offre une solution rapide et flexible pour la plupart des projets, tandis que des solutions comme Drupal, Joomla, ou des frameworks PHP permettent une personnalisation avancée, idéale pour des projets plus complexes.

² D'après : <https://www.mister-wp.com/divi/>

Partie 2 : Cas Réel - Présentation de l'Entreprise : Partenor Group, Filiale : Partenor Digital, Site Campus

Introduction à Partenor Group

Fondé en 1991, le Groupe Partenor est un acteur clé du conseil en innovation technologique et managériale en France. Au départ, le Groupe s'est concentré sur des secteurs stratégiques comme l'énergie, les utilities, et la banque, où il a développé une expertise solide. Au fil du temps, Partenor Group a élargi son champ d'action pour inclure d'autres domaines tels que le secteur public, l'industrie, et les services. Cette évolution reflète la capacité du Groupe à s'adapter aux besoins changeants de ses clients, tout en renforçant ses compétences pour devenir un cabinet de conseil innovant, capable de générer une réelle valeur ajoutée.

Aujourd'hui, Partenor Group rassemble 460 consultants et a réalisé un chiffre d'affaires de 66 millions d'euros en 2023. Avec une présence à travers plusieurs régions françaises, notamment à Paris, Lille, Lyon, et Nantes, le Groupe est en mesure de répondre aux besoins spécifiques de ses clients locaux tout en consolidant sa position sur le marché national. Cette implantation régionale permet à Partenor de rester proche de ses clients, en offrant des solutions personnalisées qui répondent aux défis particuliers de chaque secteur.

Structuration du Groupe

Partenor Group est structuré autour de quatre branches principales, chacune spécialisée pour répondre aux divers besoins de ses clients, offrant ainsi une gamme complète de services allant du conseil stratégique à la mise en œuvre technique.

- **SpinPart** : Cette branche se concentre sur le conseil en management et la conduite du changement. SpinPart accompagne les entreprises dans la transformation de leurs modèles organisationnels, en les aidant à s'adapter aux nouvelles réalités économiques et sociales. Cette approche vise à optimiser les performances des entreprises en les préparant aux défis futurs.
- **HitPart** : Spécialisée dans l'expertise IT et la réalisation technique, HitPart aide les entreprises à relever les défis posés par les technologies émergentes.

Elle se charge également de sécuriser les systèmes d'information (SI) et de développer des solutions sur mesure, adaptées aux besoins spécifiques des clients. Cette branche joue un rôle crucial dans la modernisation et la sécurisation des infrastructures IT.

- **StarClay** : Cette branche se concentre sur le Big Data, le Machine Learning, le Traitement du langage naturel (NLP) et l'Intelligence Artificielle. StarClay exploite les données des entreprises pour en faire des leviers de succès, en utilisant des techniques avancées de traitement et d'analyse des données. Cette expertise permet aux entreprises de prendre des décisions basées sur des données fiables et de maximiser leur performance.
- **Partenor Digital** : Dédiée aux conseils opérationnels et aux projets digitaux, Partenor Digital joue un rôle central dans la transformation numérique des entreprises. En transformant les systèmes d'information en atouts stratégiques, cette branche aide les entreprises à se positionner avantageusement dans un monde de plus en plus digitalisé.

Direction et Leadership

Le Groupe Partenor est dirigé par une équipe de leaders expérimentés qui pilotent les différentes branches et opérations du Groupe :

- **Ghislaine Palloix** : Présidente du Groupe Partenor, elle assure la direction stratégique et le leadership global du Groupe.
- **Ludivine Mairesse** : Directrice Générale, elle est en charge de la gestion quotidienne des opérations et de la mise en œuvre des stratégies du Groupe.
- **Fatima Moraux / Olivier Quéméneur** : Directeurs des opérations, ils supervisent l'exécution des projets et l'efficacité opérationnelle.
- **Thibault Palloix** : Appui à la direction des opérations, il apporte un soutien stratégique et opérationnel à la direction.
- **Cécilia Morais** : Directrice Administrative et Financière, responsable de la gestion financière et administrative du Groupe.

- **Nicolas Leclère** : Responsable du recrutement et des relations avec le CSE, il gère le capital humain et les relations sociales au sein du Groupe.
- **Clément Dougé** : Responsable du marketing et de la communication, il pilote la stratégie de communication et de marketing du Groupe.

Direction Commerciale

La direction commerciale du Groupe est organisée par Business Units (BU), chacune sous la responsabilité de leaders expérimentés :

- **Michael Kalifa** : Responsable de la Direction BU Industries, Services et Transport.
- **Bertrand Buissères** : Directeur de la BU Énergie et autres utilities.
- **Régis Garnung** : Responsable de la Direction BU Banque et Assurance.

(Voir Annexe 2)

Focus sur Partenor Digital

Partenor Digital, la filiale spécialisée dans les projets digitaux, est un pilier de la transformation numérique des entreprises au sein du Groupe Partenor. Avec une équipe de 310 consultants, Partenor Digital se consacre à aider les entreprises à naviguer dans le paysage numérique en constante évolution. Cette branche intervient principalement dans des secteurs variés tels que l'énergie, l'eau, le transport, la banque, l'assurance, l'industrie, le retail, et les télécommunications. L'objectif principal est de permettre à ces entreprises de mettre en œuvre des stratégies digitales efficaces et innovantes, leur donnant ainsi un avantage compétitif.

En tant qu'administrateur système d'information dans le service communication et marketing de Partenor Digital, je fais partie d'une petite équipe de trois personnes (voir Annexe 3) dédiée à la création et à la maintenance des sites web. Ce service joue un rôle clé en étant le lien entre la direction et les besoins en communication digitale, s'occupant notamment de la gestion des événements sur LinkedIn, que je recrée ensuite sous forme de pages web. Notre travail est crucial pour garantir que les sites de nos clients non seulement fonctionnent bien, mais qu'ils offrent également une expérience utilisateur optimale.

Rôle et Missions

Mon rôle chez Partenor Digital s'inscrit dans le cadre de projets digitaux variés où j'interviens principalement sur les aspects techniques. Mes missions incluent le développement de sites web, l'optimisation des performances, ainsi que la gestion de la maintenance technique des plateformes digitales. Cette position me permet d'apporter une contribution significative au bon déroulement des projets, en veillant à ce que les sites web soient à la fois performants et sécurisés.

En Résumé

Partenor Group, avec ses quatre branches spécialisées, offre une expertise complète qui couvre tous les aspects du conseil, du stratégique au technique. La diversification de ses activités permet de proposer une offre globale, répondant aux défis de transformation des entreprises dans un environnement économique en constante évolution. En particulier, Partenor Digital incarne l'engagement du Groupe dans la transition numérique, en fournissant des solutions innovantes et personnalisées qui répondent aux besoins spécifiques de chaque client.

L'expansion régionale du Groupe, avec des bureaux à Lille, Lyon, et Nantes, témoigne de sa volonté de rester proche de ses clients et de répondre efficacement à leurs besoins locaux. Depuis sa création, Partenor a su s'imposer comme un leader dans le domaine du conseil en innovation technologique et managériale, en se positionnant comme un créateur de valeur pour ses clients grâce à une maîtrise des expertises et une approche axée sur l'efficacité et l'innovation.

Partie 3 - Comparaison des Approches PHP et WordPress : Un Cas Pratique

Cas Pratique : Site Vitrine et Site de Connexion/Inscription avec Accès Admin

Dans cette section, je vais vous présenter un cas pratique qui permet de mieux comprendre les différences entre l'utilisation de WordPress et une approche de développement plus traditionnelle basée sur PHP. Le cas en question concerne le site "Partenor Campus", qui combine un site vitrine avec des fonctionnalités de connexion, d'inscription, et des accès administratifs. (Voir Annexe 4,5,6)

Récemment, le site Partenor Campus a rencontré plusieurs problèmes techniques, comme des bugs récurrents, des temps de chargement trop longs, et une utilisation excessive de l'espace serveur. Ces problèmes ont été accentués par la complexité de la gestion multi-sites.

Ce cas pratique servira d'exemple concret pour comparer les solutions apportées par WordPress et celles issues d'une approche de développement libre en PHP. Nous allons voir ensemble les différentes étapes de la migration, les solutions qui ont été mises en place pour corriger ces problèmes, et comment chacune des approches (WordPress vs PHP) peut répondre aux besoins spécifiques d'un site complexe comme Partenor Campus.

Pour un site vitrine qui nécessite en plus des fonctionnalités de connexion et d'inscription avec des accès administratifs, voici comment on pourrait s'y prendre.

À chaque étape, je mettrai en avant les avantages et les inconvénients de PHP et de WordPress, tout en soulignant les différences majeures et les éléments importants à considérer pour choisir entre ces deux options.

Enfin, pour créer un site web, il est indispensable de passer par plusieurs étapes clés. Même si la conception et la planification initiales impliquent des processus similaires, comme la création de wireframes (maquettes) et la définition des objectifs

du site, je me concentrerai ici sur les aspects techniques du codage et de la configuration du site.

Choix du Langage de Programmation

Quand on commence à coder, le choix du langage de programmation est vraiment important. Pour un site construit avec WordPress, les langages de base utilisés sont HTML, CSS, et JavaScript (voir Annexe 7). On peut aussi ajouter du PHP, notamment grâce à des plugins comme "PHP Snippets". Ce plugin est super pratique car il permet d'ajouter et d'exécuter du code PHP personnalisé directement sur votre site, sans avoir à toucher aux fichiers du thème ou des plugins. Ça permet de garder une gestion centralisée du code, ce qui est bien plus simple à gérer.

Programmation en Open-Source

Si on choisit une solution de programmation en open-source, le langage utilisé dépend vraiment des besoins spécifiques du projet. Cette approche donne plus de liberté, en utilisant des langages comme HTML, CSS, et JavaScript pour le front-end, et PHP, Python, ou Node.js pour le back-end. Selon les compétences de l'équipe, il peut être judicieux de choisir un framework ou une bibliothèque pour accélérer le développement. Le choix de la base de données, que ce soit MySQL, MariaDB, ou MongoDB, est aussi un aspect important à prendre en compte pour s'assurer que tout fonctionne bien ensemble.

Hébergement et Installation de WordPress

Pour un site WordPress, le choix du serveur et de l'hébergement est vraiment fondamental. Prenons l'exemple de la migration du site "Campus" vers Kinsta. Le site a été déplacé vers un centre de données en Belgique (BE) (voir Annexe 8), ce qui a permis d'optimiser les performances pour un public européen. Même si je n'ai pas été impliqué dans la création initiale du site, j'ai joué un rôle crucial en tant qu'administrateur pour gérer cette migration. On a utilisé l'environnement "Live" pour le site principal, tandis qu'un environnement de staging a été mis en place pour tester les modifications avant de les déployer en production. Cette approche aurait été similaire lors de la création d'un nouveau site, surtout avec Kinsta qui offre une

installation automatique de WordPress et la possibilité de configurer les paramètres de base facilement.

Fonctionnalités Offertes par Kinsta

Kinsta offre des fonctionnalités avancées, comme un serveur de staging qui permet de tester les modifications avant de les mettre en production, des sauvegardes journalières et manuelles pour assurer la sécurité des données, et un support technique super réactif, avec des temps de réponse généralement en moins d'une minute.

De plus, Kinsta envoie régulièrement des notifications par e-mail pour informer les utilisateurs des failles de sécurité potentielles dans les plugins ou les thèmes utilisés sur leurs sites. Par exemple, un plugin vulnérable pourrait permettre à des visiteurs non authentifiés de découvrir la page de connexion, normalement cachée par le plugin. Ces alertes permettent aux administrateurs de réagir rapidement en mettant à jour les composants concernés, renforçant ainsi la sécurité du site. (Voir annexe 9)

Choix du Serveur

Développement Libre

Quand on parle de codage, le choix du serveur est un aspect à prendre en compte à différentes étapes du projet :

- **Début du projet** : Il faut évaluer les besoins en termes de performance et de sécurité.
- **Développement** : Utiliser un environnement de développement local pour travailler plus efficacement.
- **Tests** : Configurer un serveur de staging pour effectuer des tests en conditions réelles, afin d'identifier et corriger les éventuels problèmes.
- **Production** : Finaliser le choix de l'hébergement et configurer les aspects techniques comme le type de serveur, la gestion de la base de données, les certificats SSL, etc.

Même après le déploiement, il est essentiel de surveiller et de maintenir l'hébergement régulièrement pour garantir que les performances restent optimales, que la sécurité soit à jour, et que toutes les mises à jour nécessaires soient effectuées.

Wordpress

Infrastructure Kinsta : Kinsta s'appuie sur l'infrastructure de Google Cloud Platform, offrant une solution d'hébergement rapide, sécurisée, et évolutive. Kinsta utilise des conteneurs LXD pour isoler chaque site web, ce qui garantit un environnement de serveur dédié pour chaque projet, assurant ainsi une stabilité et une sécurité accrues.

Nginx : Kinsta utilise Nginx comme serveur web, reconnu pour sa capacité à gérer un grand nombre de connexions simultanées tout en maintenant une performance élevée. Nginx est souvent privilégié pour sa rapidité et son efficacité dans la gestion de la charge.

PHP : Sur Kinsta, les utilisateurs peuvent choisir parmi différentes versions de PHP, ce qui leur permet de rester à jour avec les dernières fonctionnalités et améliorations en termes de performance.

Développement Front-End et Back-End

Le développement peut commencer par le front-end ou le back-end, selon l'organisation de l'équipe.

Front-End

Wordpress

Le choix du thème est une étape cruciale quand on travaille avec WordPress. On peut en sélectionner un directement depuis le WordPress Theme Directory ou sur des marketplaces comme ThemeForest. Les thèmes ont un impact majeur sur

l'apparence visuelle du site, l'utilisation de templates, les fonctionnalités intégrées, le design responsive, et même l'optimisation SEO.

Pour le site "Campus", par exemple, on a utilisé un thème populaire comme Master Study Child. Ce thème est très flexible et compatible avec de nombreux plugins populaires. Il a été installé et activé via le tableau de bord WordPress (voir Annexe 10), puis personnalisé pour répondre aux besoins spécifiques du site. Avec des plugins comme Elementor et ses add-ons, il a été possible de personnaliser l'interface utilisateur de manière avancée, sans avoir besoin de compétences en codage complexes (voir Annexe 11).

Développement Libre

Dans un développement libre, la création d'un site web suit une approche méthodique où chaque élément est construit manuellement pour permettre une personnalisation totale. Le processus commence par la création de la structure HTML pour toutes les pages du site. Cette étape est vraiment cruciale, car le HTML sert de base pour structurer le contenu, en définissant les sections, les titres, les paragraphes, et tous les éléments nécessaires pour organiser le contenu.

Une fois la structure en place, on passe à l'ajout du CSS pour styliser le site. Le CSS est ce qui permet de définir l'apparence visuelle des pages en jouant sur les couleurs, les polices, les espacements, et d'autres aspects esthétiques. Cette étape est essentielle pour s'assurer que le site reflète bien l'identité visuelle souhaitée, en garantissant une cohérence dans le design et une expérience utilisateur agréable.

Enfin, la dernière étape consiste à intégrer de l'interactivité en utilisant JavaScript. C'est grâce à JavaScript qu'on peut ajouter des fonctionnalités dynamiques comme des menus déroulants, des sliders, des animations, et des formulaires interactifs. Avec JavaScript, le site devient non seulement fonctionnel, mais aussi beaucoup plus engageant pour les utilisateurs, en leur offrant des interactions en temps réel.

Création et Organisation du Contenu

La première étape consiste à créer les pages principales du site, comme l'Accueil, À propos, Contact, et Blog. Ces pages forment la base du site et sont faciles à configurer grâce à l'interface de WordPress.

Pour les blogs ou les sites axés sur le contenu, il est essentiel de créer et de publier des articles. Ces articles peuvent être organisés de manière logique avec des catégories et des étiquettes, ce qui facilite la navigation pour les utilisateurs et améliore la structure générale du site.

En parallèle, il faut aussi ajouter des médias comme des images, des vidéos, et d'autres fichiers pour enrichir le contenu des pages et des articles. WordPress simplifie cette tâche en offrant une gestion centralisée de tous ces médias via la bibliothèque de médias.

Installation des Plugins

L'une des grandes forces de WordPress, c'est sa capacité à être étendu grâce aux plugins. Le processus commence par choisir les plugins essentiels, ceux qui ajoutent des fonctionnalités spécifiques au site, comme le SEO, la sécurité, ou encore les formulaires de contact. (voir Annexe 12)

Pour le site "Campus", des plugins comme Elementor, WP Activity Log, Contact Form 7, et GDPR Compliance & Cookie Consent ont été installés et activés pour améliorer la personnalisation, la sécurité, et la conformité du site. Une fois activés, chaque plugin a été configuré en fonction des besoins du site pour en tirer le meilleur parti.

Configuration des Fonctionnalités Avancées

Pour des fonctionnalités spécifiques, on peut utiliser des plugins comme Contact Form 7 ou WPForms pour créer des formulaires personnalisés. D'autres plugins

comme Yoast SEO ou Rank Math sont parfaits pour optimiser la visibilité du site sur les moteurs de recherche. Pour renforcer la sécurité, des plugins comme Wordfence ou Sucuri sont très efficaces, et pour améliorer les performances, des outils comme WP Rocket ou Smush font un excellent travail.

Développement Back-End

Le développement back-end est essentiel pour gérer les données, la logique de l'application, et la communication entre le serveur et l'interface utilisateur. Ce processus inclut plusieurs étapes clés :

Configuration du Serveur

La première étape du développement back-end consiste à configurer un serveur web performant et sécurisé. Les serveurs les plus couramment utilisés sont Apache et Nginx. Ils sont responsables de gérer les requêtes entrantes, de distribuer les ressources, et de permettre la communication entre le client (navigateur) et le serveur.

WordPress

Pour le site "Campus" hébergé sur Kinsta, la configuration du serveur est en grande partie gérée par la plateforme elle-même. Kinsta utilise Nginx comme serveur web, associé à PHP (version 8.3), ce qui garantit des performances élevées et une gestion efficace des requêtes. Cette configuration est spécialement optimisée pour WordPress, assurant une gestion fluide et sécurisée du site, sans que l'utilisateur ait besoin de se plonger dans des configurations manuelles complexes.

Développement Libre

En développement libre, la configuration du serveur est entièrement manuelle. Les développeurs doivent choisir entre des serveurs comme Apache ou Nginx, configurer PHP (ou un autre langage back-end), et optimiser les paramètres du serveur en fonction des besoins du site. Cela permet une personnalisation complète, mais demande aussi une expertise technique pour bien configurer et sécuriser le tout.

Création de l'API

WordPress

Avec WordPress, l'API REST est déjà intégrée, ce qui permet aux développeurs de rendre les données du site accessibles sous forme de JSON. Pour le site "Campus", cette API gère les opérations CRUD (Create, Read, Update, Delete) pour des entités comme les utilisateurs et les articles. L'utilisation de l'API REST de WordPress est facilitée par une documentation abondante et une compatibilité native avec de nombreux outils front-end, ce qui rend le processus assez fluide.

Développement Libre

En développement libre, l'intégration front-end/back-end est entièrement personnalisée. Cela peut inclure l'utilisation d'API REST, GraphQL, ou même des protocoles WebSocket pour des communications en temps réel. Cette flexibilité permet de choisir l'approche la plus adaptée aux besoins du projet, mais elle demande aussi une expertise technique pour gérer ces intégrations efficacement.

Gestion de la Base de Données

WordPress

WordPress utilise MySQL pour stocker toutes les informations du site, que ce soit les articles, les pages, les utilisateurs, ou encore les réglages du site. Pour le site "Campus", MySQL gère efficacement l'organisation et la récupération des données. WordPress automatise les interactions avec la base de données grâce à des requêtes SQL exécutées par le code PHP intégré dans le noyau du CMS. Cette approche rend la gestion des données accessible, même pour les utilisateurs qui ne sont pas techniques, grâce aux outils d'administration intégrés directement dans le tableau de bord.

Développement Libre

En développement libre, les développeurs ont la liberté de choisir la base de données la mieux adaptée à leur projet, que ce soit MySQL, PostgreSQL, MongoDB, ou d'autres systèmes de gestion de bases de données (SGBD). La gestion des

données implique souvent de créer des modèles de données spécifiques et de mettre en place des ORM (Object-Relational Mappers) pour simplifier les interactions entre le code et la base de données. Cette flexibilité permet de répondre à des besoins très spécifiques en matière de gestion des données, mais cela nécessite aussi une gestion plus complexe de la performance et de la sécurité des données.

Authentification et Autorisation

WordPress

WordPress intègre directement un système de gestion des utilisateurs qui prend en charge l'inscription, la connexion, la récupération de mot de passe, et la gestion des rôles et des permissions. Pour le site "Campus", ces fonctionnalités sont très faciles à configurer via le tableau de bord WordPress. Les rôles prédéfinis (administrateur, éditeur, auteur, contributeur, abonné) permettent de gérer efficacement qui a accès à quoi sur le site. Si des besoins spécifiques se présentent, on peut toujours installer des plugins supplémentaires pour étendre ou personnaliser ces fonctionnalités.

Développement Libre

En développement libre, l'authentification et l'autorisation sont développées sur mesure. Les développeurs doivent créer ou intégrer des systèmes de gestion des utilisateurs, incluant la gestion des rôles et des permissions, souvent en utilisant des frameworks ou des bibliothèques spécifiques (comme Passport pour Node.js ou Devise pour Ruby on Rails). Cette approche permet de concevoir un système parfaitement adapté aux besoins spécifiques du projet, mais cela demande aussi une bonne compréhension des enjeux de sécurité liés à l'authentification des utilisateurs.

Fonctionnement du Back-End

WordPress

Le back-end de WordPress est accessible via le tableau de bord admin (wp-admin), qui propose une interface conviviale pour gérer tout le contenu du site, comme les

articles, les pages, les médias, les menus, et les commentaires. Les utilisateurs peuvent facilement personnaliser l'apparence du site grâce à l'éditeur de thème, aux widgets, et aux plugins, sans avoir besoin de compétences en codage.

Les thèmes définissent l'apparence globale du site, tandis que les plugins ajoutent des fonctionnalités spécifiques. Le back-end permet aussi de gérer les réglages essentiels du site, tels que le titre, la description, les permaliens, et les options de lecture et de discussion. La gestion des utilisateurs et des permissions est également très simple, permettant de contrôler facilement l'accès aux différentes fonctionnalités du site.

Développement Libre

En développement libre, le back-end est conçu et développé sur mesure pour répondre aux besoins spécifiques du projet. Contrairement à WordPress, qui propose une interface pré-construite (wp-admin), le développement libre nécessite de construire chaque composant du back-end, souvent en utilisant des langages comme PHP, Python, ou Node.js.

Le back-end d'un site développé librement est principalement structuré autour de frameworks ou de codes personnalisés. La gestion du contenu, telle que les articles, les pages, les médias, et les commentaires, est souvent réalisée via des bases de données relationnelles comme MySQL ou PostgreSQL, où chaque type de contenu a sa propre table et structure de données.

La personnalisation de l'apparence et des fonctionnalités du site se fait via du code écrit spécifiquement pour le projet. Par exemple, l'apparence du site est définie dans des fichiers de templates HTML/CSS, tandis que les fonctionnalités interactives sont ajoutées via du JavaScript côté client et des scripts côté serveur. Les développeurs doivent créer manuellement les fonctionnalités spécifiques au site, comme les systèmes de gestion de contenu, les formulaires dynamiques, et les API pour l'interaction avec d'autres services.

La gestion des réglages du site, tels que les paramètres de configuration globaux (comme le titre du site, les configurations de base de données, et les chemins

d'accès), se fait généralement à travers des fichiers de configuration (par exemple, config.php ou .env), qui sont directement manipulés par les développeurs.

Enfin, la gestion des utilisateurs et des permissions dans un environnement de développement libre est généralement implémentée à l'aide de bibliothèques ou de frameworks spécifiques (comme Bouncer pour PHP ou Passport pour Node.js). Cela permet de créer des rôles et des niveaux d'accès personnalisés, adaptés aux besoins spécifiques de l'application, mais nécessite une configuration et une maintenance plus complexe par rapport à une solution pré-configurée comme WordPress.

Interactions avec la Base de Données

WordPress

Les interactions entre WordPress et la base de données sont principalement gérées par des requêtes SQL exécutées via le code PHP. WordPress simplifie ces interactions grâce à ses fonctionnalités intégrées et à l'utilisation d'un ORM interne, ce qui réduit la complexité des interactions avec MySQL.

Développement Libre

En développement libre, les interactions avec la base de données nécessitent la rédaction de requêtes SQL personnalisées ou l'utilisation de bibliothèques d'ORM, comme Doctrine pour PHP ou Eloquent pour Laravel. Cette approche offre une flexibilité maximale, mais demande une bonne compréhension des structures de données et des performances des requêtes.

Sécurité et Performances

WordPress

Kinsta met en place des mesures de sécurité robustes, comme des pare-feux, une surveillance active des menaces, l'installation de certificats SSL, et des sauvegardes automatiques pour protéger les sites. En termes de performance, Kinsta propose des outils d'optimisation, tels que la mise en

cache côté serveur (Voir Annexe 13) et l'intégration d'un CDN (Content Delivery Network), pour garantir des performances élevées, même sous forte charge.

Développement Libre

En développement libre, la sécurité et les performances doivent être gérées manuellement. Cela comprend l'installation et la configuration de pare-feux, la gestion des certificats SSL, et la mise en place de stratégies de sauvegarde. L'optimisation des performances nécessite aussi l'implémentation de solutions de mise en cache, la compression des ressources, et la configuration d'un CDN, sans bénéficier de l'infrastructure préconfigurée comme celle offerte par Kinsta.

Développement et Personnalisation

WordPress

- **Thèmes Enfant (Child Themes)** : Les thèmes enfant sont un excellent moyen de personnaliser l'apparence et les fonctionnalités du site sans toucher au thème principal. Cela permet de conserver toutes les modifications même après les mises à jour du thème parent, évitant ainsi de perdre les personnalisations effectuées.
- **Plugins Personnalisés** : Le développement de plugins personnalisés permet d'ajouter des fonctionnalités spécifiques qui ne sont pas disponibles via les plugins existants. WordPress utilise un système de hooks (actions et filtres) qui permet aux développeurs d'interagir avec le noyau de WordPress et les autres plugins pour modifier ou étendre leurs fonctionnalités, offrant ainsi une grande flexibilité.

Développement Libre

- **Personnalisation du Code** : En développement libre, la personnalisation est totale. Les développeurs peuvent créer ou modifier chaque aspect du site en fonction des besoins spécifiques. Cela inclut la création de nouvelles fonctionnalités, l'intégration de services tiers, et la modification

de la logique du site. Cette flexibilité permet de créer des solutions uniques, mais elle s'accompagne d'une complexité accrue.

- **Extensions et Bibliothèques** : Au lieu de plugins, le développement libre repose sur des extensions et des bibliothèques spécifiques, qui doivent être installées et configurées manuellement. Cela permet d'ajouter des fonctionnalités complexes, mais nécessite une gestion rigoureuse des dépendances et des mises à jour pour assurer la stabilité du site.

Intégration Front-End/Back-End

Wordpress

L'intégration entre le front-end et le back-end sur WordPress se fait souvent via des appels API intégrés, notamment grâce à l'API REST de WordPress. Cette API permet d'exposer les données du site sous forme de JSON, ce qui facilite les interactions avec des applications externes ou des clients mobiles. Pour le site "Campus", l'API REST a été utilisée pour échanger des données de manière fluide entre le front-end et le back-end.

En plus de cela, WordPress permet la création de custom endpoints (points de terminaison personnalisés) pour répondre à des besoins spécifiques. Cela offre une flexibilité supplémentaire pour l'intégration avec d'autres systèmes ou pour développer des fonctionnalités sur mesure, tout en maintenant une gestion cohérente des données. Pour des applications plus complexes, des bibliothèques de gestion d'état comme Redux ou Vuex peuvent également être intégrées pour assurer que les données restent cohérentes entre le front-end et le back-end.

Développement Libre :

En développement libre, l'intégration entre le front-end et le back-end est entièrement personnalisée. Cela peut inclure l'utilisation d'API REST, de GraphQL, ou même de protocoles WebSocket pour des communications en temps réel. Cette flexibilité permet de choisir l'approche la plus adaptée aux besoins spécifiques du projet, mais elle nécessite également une expertise technique pour gérer efficacement ces intégrations. En développement libre, l'intégration front-end/back-end est entièrement

personnalisée. Cela peut inclure l'utilisation d'API REST, GraphQL, ou même des protocoles WebSocket pour des communications en temps réel. Cette flexibilité permet de choisir l'approche la plus adaptée aux besoins du projet, mais exige également une expertise technique pour gérer efficacement ces intégrations.

Sécurité

Wordpress

- **SSL/TLS** : L'installation des certificats SSL est gérée automatiquement par Kinsta pour les sites WordPress, y compris le site "Campus" (Voir Annexe 14). Cela garantit que toutes les communications entre le serveur et les utilisateurs sont chiffrées. Kinsta gère aussi le renouvellement automatique des certificats SSL, ce qui simplifie la gestion de la sécurité. De plus, il est important de **forcer le HTTPS** en redirigeant tout le trafic de HTTP vers HTTPS. Cette redirection renforce non seulement la sécurité, mais elle a également un impact positif sur le SEO, car les moteurs de recherche préfèrent les sites sécurisés. (Voir Annexe 15)
- **Protection contre les Attaques** : WordPress propose une gamme de plugins comme Wordfence et Sucuri pour protéger le site contre les attaques courantes telles que les injections SQL, les attaques XSS, et les CSRF. Ces outils offrent une protection en temps réel, des scans de sécurité, et des pare-feux spécifiques à WordPress. L'activation de la **géolocalisation** permet également d'ajouter des règles de redirection qui adaptent le contenu en fonction de l'emplacement IP des visiteurs du site web, offrant ainsi une couche supplémentaire de sécurité et de personnalisation.
- **Surveillance et Mises à Jour** : Les mises à jour de sécurité pour le noyau WordPress et les plugins sont automatisées, avec des notifications et des options de gestion manuelle via le tableau de bord. Kinsta offre également une surveillance continue, alertant immédiatement en cas de détection de menaces ou de failles de sécurité.

Développement Libre :

- **SSL/TLS** : L'installation des certificats SSL est gérée automatiquement par Kinsta pour les sites WordPress, y compris le site "Campus". Cela garantit que toutes les communications entre le serveur et les utilisateurs sont chiffrées, protégeant ainsi les données sensibles échangées. Kinsta prend également en charge le renouvellement automatique des certificats SSL, ce qui simplifie la gestion de la sécurité pour les administrateurs du site.

En ce qui concerne le site "Campus", plusieurs domaines et sous-domaines sont utilisés : `campus.partenordigital.com` et `*.campus.partenordigital.com`, ainsi que `campus.kinsta.cloud` et `*.campus.kinsta.cloud`. Ces domaines sont configurés pour répondre aux différents besoins d'accès au site.

- **Domaines et Sous-domaines** : Le domaine principal (`campus.partenordigital.com`) est utilisé pour l'accès principal au site. Les sous-domaines, représentés par le wildcard `*.campus.partenordigital.com`, permettent de sécuriser plusieurs sous-domaines sous le même domaine principal, comme `blog.campus.partenordigital.com` ou `shop.campus.partenordigital.com`. Cela est particulièrement utile pour les grandes plateformes qui divisent leur contenu ou leurs services sur plusieurs sous-domaines. (Voir Annexe 16)
- **Certificat Wildcard SSL** : Le certificat wildcard SSL (`*.campus.partenordigital.com`) permet de sécuriser tous les sous-domaines sous un seul certificat. Cela simplifie la gestion des certificats et garantit que toutes les connexions vers les sous-domaines sont également chiffrées, renforçant ainsi la sécurité globale du site.
- **Domaine Kinsta (`campus.kinsta.cloud`)** : Ce domaine est utilisé par Kinsta pour des besoins internes, tels que la gestion et le staging des sites avant leur déploiement en production. Il est également sécurisé par un certificat SSL pour garantir que toutes les connexions sont protégées, même lors des phases de développement et de test.

- **Protection contre les Attaques** : WordPress propose une gamme de plugins comme Wordfence et Sucuri pour protéger le site contre les attaques courantes telles que les injections SQL, les attaques XSS (Cross-Site Scripting), et les CSRF (Cross-Site Request Forgery). Ces outils offrent une protection en temps réel, des scans de sécurité réguliers, et des pare-feux spécifiques à WordPress pour renforcer la sécurité.
- **Surveillance et Mises à Jour** : Les mises à jour de sécurité pour le noyau WordPress et les plugins sont automatisées, ce qui réduit le risque de vulnérabilités dues à des versions obsolètes. Les administrateurs reçoivent des notifications et ont des options de gestion manuelle via le tableau de bord de WordPress. Kinsta offre également une surveillance continue, alertant immédiatement en cas de détection de menaces ou de failles de sécurité, permettant une réaction rapide pour préserver l'intégrité du site.

Tests et Validation

WordPress :

- **Tests Unitaires** : Les tests unitaires pour WordPress peuvent être réalisés avec des frameworks comme PHPUnit, souvent utilisés par les développeurs de thèmes et de plugins pour s'assurer que chaque fonction individuelle fonctionne correctement.
- **Tests d'Intégration** : Les tests d'intégration peuvent être effectués via des plugins spécifiques ou des outils d'intégration continue (CI) compatibles avec WordPress. Ces tests vérifient que les différentes parties du site fonctionnent bien ensemble après l'ajout de nouvelles fonctionnalités ou modifications.
- **Tests de Performance** : Sur Kinsta, les tests de performance sont facilités par des outils intégrés comme Google PageSpeed Insights via le plugin Google Site Kit. Cela permet de tester la réactivité du site sous différentes charges et d'optimiser les temps de chargement.
- **Tests de Sécurité** : Les scans de sécurité sont réalisés automatiquement via des plugins comme Wordfence, qui identifient les vulnérabilités potentielles et proposent des correctifs avant la mise en production.

Développement Libre

- **Tests Unitaires** : En développement libre, les tests unitaires sont réalisés avec des frameworks comme JUnit pour Java, Mocha pour JavaScript, ou PHPUnit pour PHP. Ces tests doivent être écrits et exécutés manuellement pour vérifier que chaque fonction individuelle du code fonctionne comme prévu.
- **Tests d'Intégration** : Les tests d'intégration nécessitent des configurations personnalisées, souvent via des pipelines CI/CD configurés pour vérifier que les différentes parties de l'application interagissent correctement après chaque modification.
- **Tests de Performance** : Les tests de performance doivent être configurés manuellement avec des outils comme Apache JMeter ou Gatling, qui permettent de simuler des charges sur le serveur et d'identifier les goulots d'étranglement.
- **Tests de Sécurité** : Les tests de sécurité doivent être effectués avec des outils spécialisés comme OWASP ZAP ou Burp Suite, qui permettent de scanner les applications pour détecter des vulnérabilités connues et des erreurs de configuration.

Accessibilité et Compatibilité Multiplateforme

WordPress

- **Compatibilité** : WordPress et ses thèmes sont généralement conçus pour être compatibles avec la majorité des navigateurs et appareils. Par exemple, le thème "Master Study Child" utilisé sur le site "Campus" est responsive par défaut, ce qui garantit une compatibilité multiplateforme sans nécessiter de modifications supplémentaires.
- **Accessibilité** : WordPress propose des plugins et des pratiques intégrées pour assurer l'accessibilité, en respectant notamment les standards WCAG (Web Content Accessibility Guidelines). Des outils comme WP Accessibility permettent de tester et d'améliorer l'accessibilité directement depuis le tableau de bord.

Développement Libre

- **Compatibilité** : En développement libre, la compatibilité multiplateforme doit être testée et ajustée manuellement. Les développeurs doivent utiliser des frameworks comme Bootstrap ou Foundation pour créer des interfaces responsives, et effectuer des tests sur différents navigateurs et appareils pour assurer une expérience utilisateur cohérente.
- **Accessibilité** : L'accessibilité doit être intégrée dès la phase de conception, en suivant les standards WCAG. Des outils comme axe-core peuvent être utilisés pour vérifier l'accessibilité, mais les implémentations spécifiques nécessitent un effort supplémentaire en termes de développement.

SEO et Référencement

- **Optimisation SEO** : Il est crucial d'optimiser les pages pour le SEO dès le début du développement pour améliorer la visibilité du site sur les moteurs de recherche.

Dans notre cas, nous utilisons les plugins **Rank Math SEO** et **Google Site Kit** pour gérer et optimiser le référencement du site.

- **Rank Math SEO** : Rank Math est un plugin WordPress complet pour l'optimisation SEO. Il permet de gérer facilement les paramètres de référencement, comme les méta-titres, les descriptions, les balises ALT pour les images, et bien d'autres aspects techniques du SEO (voir Annexe 17). Il propose aussi des outils pour analyser les mots-clés, suivre les performances SEO des pages, et optimiser le contenu pour les moteurs de recherche. (voir Annexe 18)
- **Google Site Kit** : Google Site Kit est un plugin officiel de Google qui intègre plusieurs outils Google directement dans WordPress. Il fournit des données analytiques sur le trafic du site, les performances, et le SEO, en connectant le site à des services comme Google Analytics, Google Search Console, et Google PageSpeed Insights (voir Annexe 19). Cela permet de surveiller et d'améliorer le référencement et les performances du site en continu. (voir Annexe 20)

Ces outils sont essentiels pour garantir que le site reste visible et performant dans les résultats des moteurs de recherche, en fournissant des insights précieux et des recommandations pour une optimisation continue.

Développement Libre

Optimisation SEO : En développement libre, l'optimisation SEO doit être mise en place manuellement. Cela inclut la création de sitemaps, l'optimisation des balises méta, et la configuration du fichier robots.txt. L'intégration avec des outils comme Google Analytics et Google Search Console doit être configurée manuellement, et le suivi des performances SEO nécessite des scripts ou des outils personnalisés.

Gestion des Utilisateurs et Permissions

Wordpress

- **Rôles et Permissions :** La configuration des rôles et des permissions des utilisateurs est cruciale pour contrôler l'accès aux différentes fonctionnalités du site, assurant ainsi la sécurité et l'intégrité des données.

Dans notre cas, je suis administrateur sur Kinsta, ce qui me permet de gérer les sites existants. Cela inclut la possibilité de faire des modifications, des sauvegardes, et même de supprimer des sites si nécessaire. Ces permissions me donnent un contrôle étendu sur la gestion quotidienne du site.

Cependant, Alexandre Dupas, qui était mon supérieur lors de cette migration, détient le rôle de super administrateur. En tant que super administrateur, il dispose de permissions supplémentaires qui vont au-delà de celles d'un administrateur standard. Il peut notamment créer de nouveaux sites, configurer les environnements initiaux (comme le choix du centre de données, la version de PHP, etc.), et gérer les utilisateurs à un niveau plus élevé, y compris modifier les rôles et permissions des autres administrateurs. Cela lui permet également d'accéder à des fonctionnalités de

configuration avancée, comme l'ajustement des ressources serveur (par exemple, le nombre de PHP workers), la gestion des domaines au niveau de l'organisation, et la configuration des politiques de sécurité globales.

Développement Libre

- **Rôles et Permissions** : En développement libre, les rôles et permissions doivent être créés et gérés manuellement, souvent via des bibliothèques spécifiques comme Bouncer pour PHP ou Passport pour Node.js. Cette flexibilité permet de définir des rôles très spécifiques, mais nécessite une gestion complexe des autorisations et des permissions au niveau du code.

Lancement et Maintenance

Wordpress

- **Lancement du Site** : Une fois le développement terminé, le site est mis en ligne et des vérifications sont effectuées pour s'assurer que tout fonctionne correctement.
- **Surveillance Continue** : Des outils de surveillance sont utilisés pour suivre les performances du site et détecter rapidement toute anomalie ou problème de sécurité. De plus, WordPress permet de **visualiser les logs** (Voir Annexe 20) via l'interface d'administration ou des plugins spécifiques. Cette fonctionnalité est essentielle pour identifier rapidement les erreurs et les problèmes potentiels, facilitant ainsi le dépannage et la correction des bugs en temps réel.
- **Maintenance Régulière** : Effectuer des sauvegardes régulières, mettre à jour les logiciels, et résoudre les bugs au fur et à mesure sont des pratiques essentielles pour assurer la pérennité du site.
-

Développement Libre

- **Lancement du Site** : Le lancement d'un site développé librement nécessite plusieurs étapes manuelles, telles que la configuration du serveur, le déploiement du code, et la mise en place des bases de données. Des pipelines CI/CD peuvent automatiser certaines de ces étapes, mais une configuration manuelle reste souvent nécessaire.
- **Surveillance Continue** : La surveillance continue doit être configurée à l'aide d'outils comme Nagios ou Prometheus, avec des alertes personnalisées pour suivre les performances et la sécurité. Cette approche offre une flexibilité totale, mais elle demande une configuration initiale complexe.
- **Maintenance Régulière** : La maintenance d'un site développé librement nécessite une gestion manuelle plus approfondie. Les sauvegardes doivent être programmées via des scripts ou des outils tiers comme rsync ou cron jobs, et les mises à jour de toutes les dépendances doivent être suivies de près pour éviter les conflits ou les régressions. Cela inclut les mises à jour de sécurité pour le système d'exploitation du serveur, les frameworks, les bibliothèques, et les bases de données. Cette flexibilité permet un contrôle total sur l'infrastructure, mais elle exige également une expertise technique avancée et un suivi rigoureux.

Bilan

L'objectif principal de ce projet était de comparer l'efficacité de WordPress et d'une approche sur mesure en PHP/JS pour la refonte du site Partenor Campus, dans le but d'améliorer les performances et la gestion de contenu. Le projet a permis de réaliser plusieurs avancées significatives tout en mettant en lumière certains défis techniques.

Amélioration des Performances et de l'Expérience Utilisateur

La refonte du site a conduit à une réduction du temps de chargement, ce qui a considérablement amélioré l'expérience utilisateur. Cette optimisation des performances s'est traduite par une navigation plus fluide, une meilleure rétention des visiteurs, et potentiellement un meilleur référencement sur les moteurs de recherche.

Gestion de Contenu Simplifiée

L'adoption de WordPress a simplifié la gestion du contenu grâce à son interface intuitive et à ses outils intégrés pour l'édition et la publication. Cela a non seulement facilité les opérations quotidiennes pour les administrateurs du site, mais a également permis des mises à jour plus rapides et plus régulières du contenu, ce qui est essentiel pour maintenir l'engagement des utilisateurs.

Défis Techniques et Solutions

Plusieurs défis techniques ont été rencontrés tout au long du projet. Par exemple, lors de la migration du site, des problèmes ont émergé, tels que des difficultés de connexion, des temps de chargement infinis, et des images qui ne s'affichaient pas correctement. Ces problèmes ont nécessité un changement de certains plugins, ainsi que la reconfiguration des paramètres pour corriger les bugs restants. La migration à Kinsta a permis de résoudre la majorité de ces problèmes, améliorant ainsi la stabilité et la performance du site.

De plus, un autre défi a été rencontré lors de l'import/export de contenu sur le site SpinPart, où certaines images n'étaient pas disponibles en raison de problèmes d'accès et de droits administratifs. Grâce à l'accès SFTP, nous avons pu résoudre ce problème en important et exportant les plugins avec les paramètres corrects, ce qui a permis de maintenir l'intégrité des données lors de la refonte graphique du site.

Sécurité et Maintenance

Bien que la sécurité du site ait été renforcée grâce à des mesures de base, telles que l'installation de certificats SSL et l'utilisation de plugins de sécurité, des problèmes comme le blocage de l'environnement staging lors du passage en live ont nécessité l'intervention du support technique de Kinsta. Ce soutien, ainsi que l'aide d'Alexandre Dupas pour résoudre des problèmes complexes, a été crucial pour garantir la sécurité et la stabilité du site.

Mon Expérience Personnelle

Ce projet a également été ma première expérience avec WordPress, ce qui a été un défi en soi. Le début a été lent, notamment pour prendre en main le thème DIVI, mais cette expérience m'a permis de découvrir une nouvelle manière de développer des sites web. Bien que je préfère coder "à la dure" pour la liberté qu'offre un développement personnalisé, j'ai reconnu les avantages non négligeables de WordPress, notamment sa facilité d'accès, son usage intuitif, et la rapidité de mise en place des sites. Cependant, WordPress impose certaines limites, en particulier avec des thèmes ou des Template préétablis que l'on retrouve souvent sur de nombreux sites.

Évolutivité et Adaptabilité

Le projet a également permis d'évaluer l'évolutivité des deux approches. WordPress s'est révélé être une solution flexible, particulièrement adaptée pour les besoins de gestion de contenu en constante évolution. Cependant, l'approche PHP/JS personnalisée offre un contrôle plus fin sur les fonctionnalités spécifiques, ce qui pourrait être exploité pour des développements futurs nécessitant une grande adaptabilité.

Collaboration et Gestion de Projet

Ce projet m'a donné l'occasion de renforcer mes compétences en gestion de projets numériques, notamment en ce qui concerne la coordination d'une équipe multidisciplinaire et la gestion des délais et des ressources. L'importance de la communication entre les développeurs front-end et back-end, ainsi qu'avec les parties prenantes, a été un facteur clé de succès.

Retours des Utilisateurs et Itérations Futures

Un autre aspect important du projet a été la collecte des retours des utilisateurs après la mise en ligne du site. Les commentaires de mes collègues ont permis d'identifier certaines améliorations à apporter, comme l'optimisation des formulaires de contact et l'ajout de nouvelles fonctionnalités pour une meilleure interaction utilisateur. Des itérations futures seront nécessaires pour affiner ces aspects et répondre aux attentes évolutives des utilisateurs.

Conclusion

Ce mémoire a exploré en profondeur les différences entre l'utilisation de WordPress et une approche sur mesure en PHP/JS pour la refonte du site Partenor Campus. Les résultats ont montré que pour ce projet spécifique, une solution hybride offre un équilibre optimal entre performance et gestion de contenu. Cette approche a permis de combiner la simplicité et la rapidité de déploiement de WordPress avec la flexibilité et les capacités de personnalisation offertes par PHP et JavaScript.

En fin de compte, bien que WordPress soit idéal pour des déploiements rapides et une gestion simplifiée des contenus, l'utilisation de PHP s'avère indispensable pour des fonctionnalités sur mesure. Cette combinaison des deux technologies s'est révélée être la plus avantageuse, permettant de répondre aux exigences spécifiques tout en maintenant une interface utilisateur conviviale et facile à gérer.

Il est recommandé aux entreprises de considérer l'intégration de CMS avec des scripts personnalisés pour maximiser la flexibilité tout en réduisant les coûts de

développement. Cette approche permet de bénéficier du meilleur des deux mondes : l'efficacité opérationnelle des systèmes préconstruits et la souplesse des développements sur mesure.

À l'avenir, l'intégration de technologies émergentes, telles que l'intelligence artificielle, pourrait transformer notre manière d'aborder les projets de refonte web, offrant de nouvelles opportunités d'innovation. Ces technologies pourraient notamment améliorer la personnalisation, automatiser des tâches complexes, et offrir une analyse prédictive pour optimiser l'expérience utilisateur.

Ce projet m'a permis de concrétiser mes compétences théoriques en développement web, de renforcer ma compréhension des systèmes de gestion de contenu et des scripts sur mesure, et m'a ouvert de nouvelles perspectives dans le domaine du conseil digital. En combinant les connaissances acquises avec une vision stratégique des nouvelles technologies, je suis désormais mieux préparé pour aborder les défis futurs et contribuer à l'innovation dans le développement de projets web.

Annexes











Annexe 1 :

Ajouter des extensions [Téléverser une extension](#) Aide

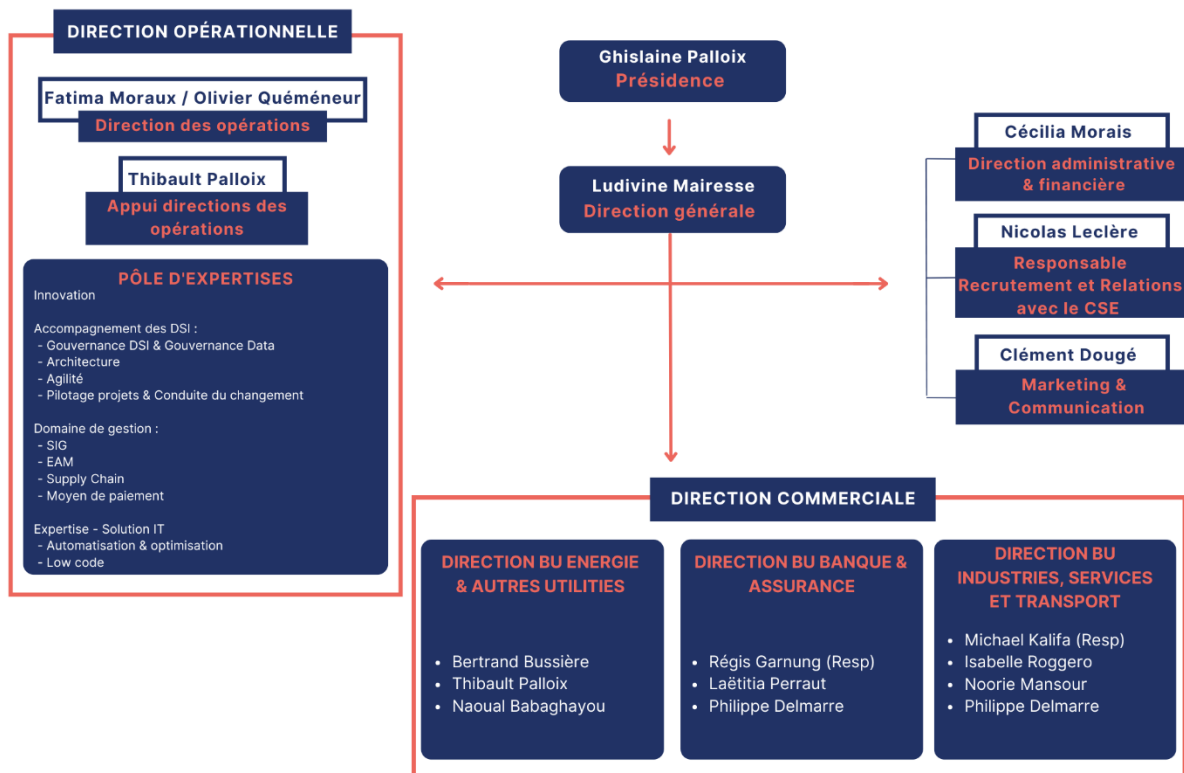
Mises en avant Populaires Recommandées Favorites

Rechercher des extensions Mot-clé

Les extensions élargissent les fonctionnalités de WordPress, et en ajoutant de nouvelles. Vous pouvez installer des extensions automatiquement en provenance du [répertoire des extensions WordPress](#), ou téléverser une extension au format .zip en cliquant sur le bouton en haut de cette page.

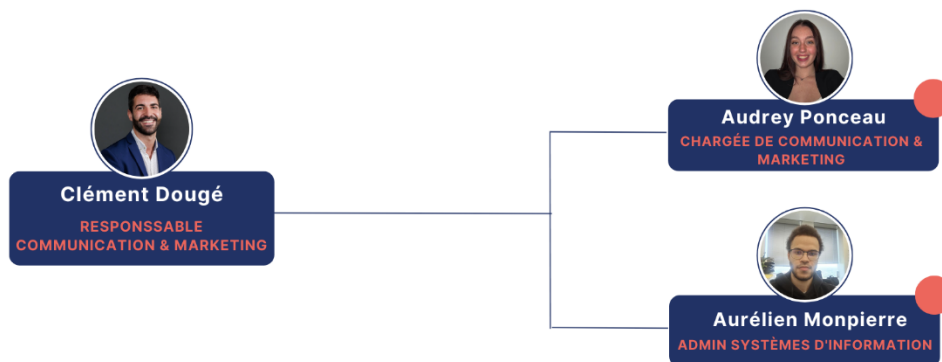
 Classic Editor Enables the previous "classic" editor and the old-style Edit-Post screen with TinyMCE, Meta Boxes, etc. Supports all plugins that extend this screen. <i>Par WordPress Contributors</i>	 Akismet Anti-spam: Spam Protection La meilleure protection anti-spam pour bloquer les notifications des commentaires et des formulaires de contact. La solution anti-spam la plus fiable pour WordPress et WooCommerce. <i>Par Automattic - Anti-spam Team</i>	 Jetpack – WP Security, Backup, Speed, & Growth Améliorez votre sécurité WP avec des outils puissants et rapides, comme le sauvegarde, le pare-feu d'application Web et la recherche de logiciels malveillants. Comment des outils gratuits tels que les outils statistiques, le niveau de diffusion de contenu et le partage sur les réseaux sociaux. <i>Par Automattic</i>	 Classic Widgets Machive l'écran de widgets classique étudié dans Appearance – Widgets ainsi que l'outil de personnalisation. Désactive la gestion des widgets via l'éditeur de blocs. <i>Par WordPress Contributors</i>
★★★★★ (1 181) 10 millions et + installations actives Dernière mise à jour : il y a 5 semaines ✓ Compatible avec votre version de WordPress	★★★★★ (1 022) 8 millions et + installations actives Dernière mise à jour : il y a 6 jours ✓ Compatible avec votre version de WordPress	★★★★★ (2 122) 4 millions et + installations actives Dernière mise à jour : il y a 5 jours ✓ Compatible avec votre version de WordPress	★★★★★ (250) 2 millions et + installations actives Dernière mise à jour : il y a 4 semaines ✓ Compatible avec votre version de WordPress
 Health Check & Troubleshooting Health Check identifies common problems, and helps you troubleshoot plugin and theme conflicts. <i>Par The WordPress.org community</i>	 Gutenberg The Gutenberg plugin adds editing, customization, and site building to WordPress. Use it to test beta features before their official release. <i>Par Gutenberg Team</i>	 bbPress bbPress est un forum pour WordPress <i>Par The bbPress Contributors</i>	 Performance Lab Performance plugin from the WordPress Performance Team, which is a collection of standalone performance features. <i>Par WordPress Performance Team</i>
★★★★☆ (164) 300 000+ installations actives Dernière mise à jour : il y a 2 semaines ✓ Compatible avec votre version de WordPress	★★★★☆ (3 792) 300 000+ installations actives Dernière mise à jour : il y a 2 semaines Non testé avec votre version de WordPress	★★★★☆ (336) 100 000+ installations actives Dernière mise à jour : il y a 1 mois Non testé avec votre version de WordPress	★★★★★ (18) 100 000+ installations actives Dernière mise à jour : il y a 2 semaines ✓ Compatible avec votre version de WordPress
 BuddyPress Rassemblez vous en toute sécurité, et selon vos règles, dans WordPress. <i>Par The BuddyPress Community</i>	 Rollback Auto Update A feature plugin now only for testing. Rollback Auto Update, plus Rollback pane 3. Manual Rollback of update failures has been committed in WordPress 6. <i>Par VIP Core Contributors</i>		

Annexe 2 :




Annexe 3 :


Service communication & marketing



● Alternant(e)

Annexe 4,5,6 :

**CAMPUS**
partenariat digital

FORMATIONS
FILIÈRES
SUPPORT
MON COMPTE


MON COMPTE

amonpierre
Instructor





SETTINGS

☆☆☆☆☆ 0
0 Reviews.

Courses
All
Published
In draft

+ Add New course

**CAMPUS**
partenariat digital

FORMATIONS
FILIÈRES
SUPPORT
MON COMPTE


MON COMPTE

amonpierre
Instructor



SETTINGS

☆☆☆☆☆ 0
0 Reviews.

Courses
All
Published
In draft

+ Add New course

Mendix : Intermediate Developer

Instructor
Lucas Taranne

100
Students enrolled

★★★★★ 0
0 reviews

Add to wishlist Share

START COURSE

Course details

Duration	48h
Lectures	15
Level	Intermédiaire

Popular courses

- 

Découvrez le framework Django

Free ★★★★★
By Lucas Taranne
- 

Apprenez à apprendre

Free ★★★★★
By Lucas Taranne
- 

Initiez-vous au machine

Description Curriculum Reviews



The next level of Mendix! You've learned the basics with Rapid Developer and have built a first app. Now learn best practice around collaborating in larger teams and add your next level of knowledge

Annexe 7 :

Code Paramètres





Préréglé: Défaut

Contenu

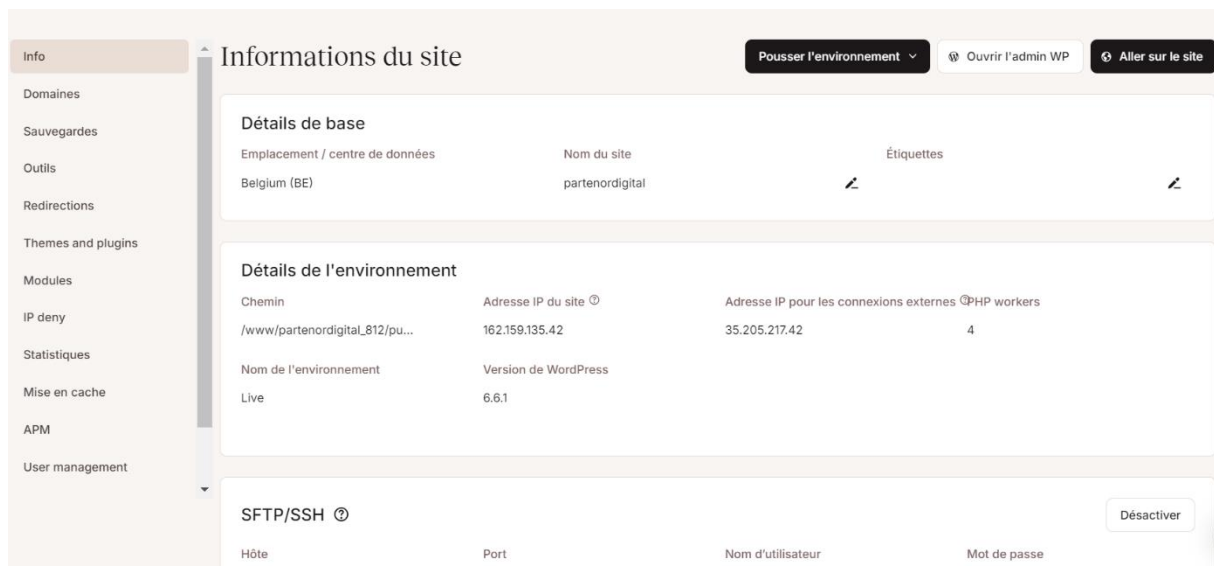
Style

Avancé

```
1 [pgaf_post_filter design="design-1" exclude_cat="29" show_date="false" show_author="false"]
2
3 <script>
4 let filtreList = document.getElementsByClassName('pgafu-filter')[0];
5 let filtreButton = document.getElementsByClassName('boutonFiltre')[0];
6 filtreList.style.display = "none";
7 let isButtonFiltreEnable = false;
8 filtreButton.addEventListener("click", () =>{
9   if(isButtonFiltreEnable){
10     filtreList.style.display = "none";
11     filtreButton.innerHTML = "Afficher les filtres";
12     isButtonFiltreEnable = false;
13   }
14   else{
15     filtreList.style.display = "block";
16     filtreButton.innerHTML = "Masquer les filtres";
17     isButtonFiltreEnable = true;
18   }
19 });
20 </script>
```



Annexe 8 :



Annexe 9 :

Hello Aurel,

We are writing to you today to notify you about a security vulnerability discovered in the **WPS Hide Login** plugin. We detected the plugin on one or more of your websites:

- oldspinpart
- partenordigital
- starclay

You can ignore this message if you have already updated WPS Hide Login to the latest version or have it set to update automatically.

The plugin has a vulnerability that allows **unauthenticated visitors** to discover the login page that should be hidden by the plugin.

This vulnerability affects versions before 1.9.16. We recommend updating this plugin immediately to the latest version. To update, log in to MyKinsta and go to WordPress Sites > *sitename* > Themes and Plugins > Installed Plugins > Wps hide login, and click the "Update" button.

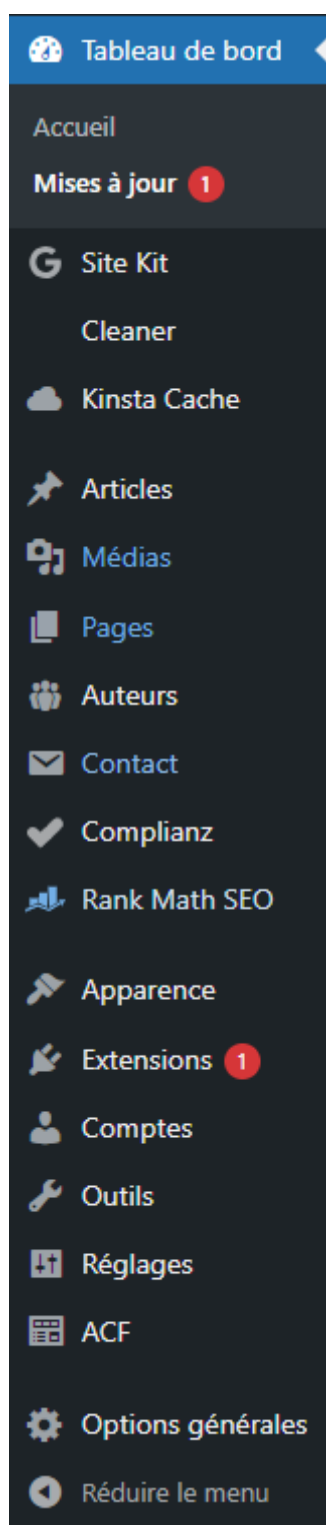
This vulnerability may exist in both live and staging environments. We recommend that both be checked and updated.

We detected this vulnerability on June 11th, 2024, at 9:05 PM UTC, but due to the time difference between scanning for vulnerabilities and sending notifications, you may have already updated the plugins.

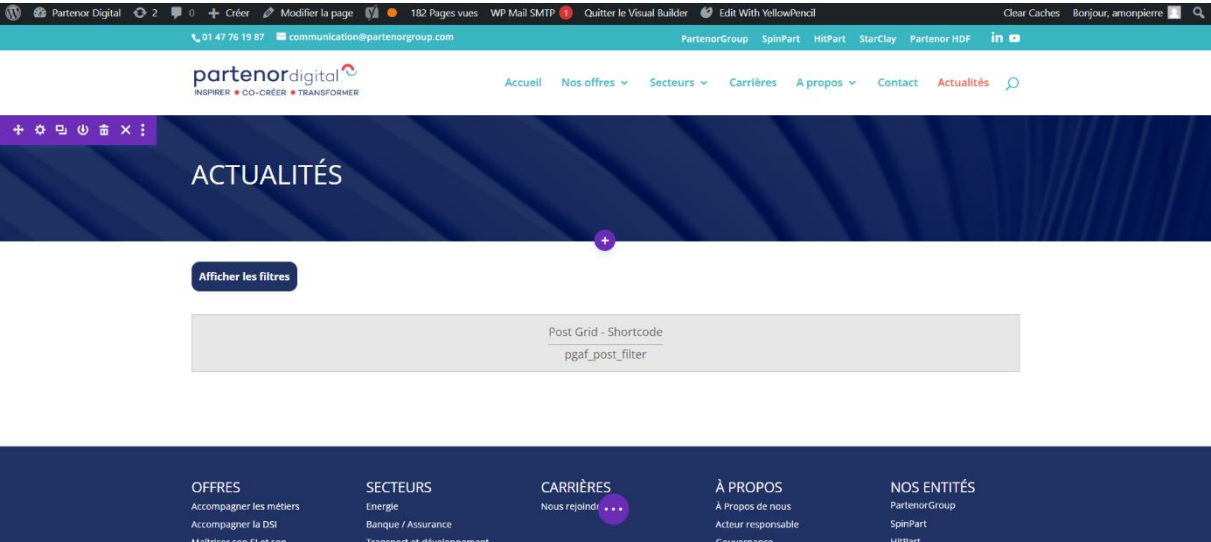
If you have any questions, don't hesitate to contact our support team.

Thank you for being a Kinsta client!

Annexe 10 :



Annexe 11 :



Annexe 12 :

<input type="checkbox"/> Classic Editor Désactiver Réglages	Active l'éditeur classique de WordPress et l'ancienne disposition de l'écran de modification d'article (TinyMCE, boîtes à méta, etc.). Prend en charge les anciennes extensions qui dépendent cet écran. Version 1.6.4 Par Contributeurs et contributeurs WordPress Afficher les détails	Activer les mises à jour auto
<input type="checkbox"/> Compliance GDPR/CCPA Cookie Consent Mettre à niveau vers la version premium Support Réglages Désactiver	Compliance Privacy Suite pour le RGPD, CCPA, DSGVO et AVG, avec un avertissement conditionnel de cookies et une politique de cookies personnalisée. Version 7.1.0 Par Really Simple Plugins Afficher les détails	Activer les mises à jour auto
<input type="checkbox"/> Contact Form 7 Réglages Désactiver	Juste une autre extension de formulaire de contact. Simple mais souple d'utilisation. Version 5.9.8 Par Takayuki Miyoshi Afficher les détails	Activer les mises à jour auto
<input type="checkbox"/> Duplicate Page Réglages Fais un don Désactiver	Dupliquer des publications, des pages et des publications personnalisées à l'aide d'un simple clic. Version 4.5.4 Par rmdpsingh3037 Afficher les détails	Activer les mises à jour auto
<input type="checkbox"/> InfiniteWP - Client Activer Supprimer	This is the client plugin of InfiniteWP that communicates with the InfiniteWP Admin panel. Version 1.13.0 Par Rammatu Afficher les détails	Activer les mises à jour auto
<input type="checkbox"/> Rank Math SEO Désactiver Réglages Assistant de configuration	Rank Math SEO is the Best WordPress SEO plugin with the features of many SEO and AI SEO tools in a single package to help multiply your SEO traffic. Version 1.0.225 Par Rank Math Afficher les détails Commencer Documentation	Activer les mises à jour auto
<input type="checkbox"/> Shortcode Cleaner Lite Désactiver Dashboard Get Pro	Clean your WordPress content from unused broken shortcodes. Version 1.0.9 Par Joozer Afficher les détails	Activer les mises à jour auto
<input type="checkbox"/> Site Kit by Google Paramètres Désactiver	Site Kit est une solution complète destinée aux utilisateurs de WordPress. Elle regroupe tous les services Google utiles pour réussir sur le Web. Version 1.132.0 Par Google Afficher les détails Évaluer Site Kit Support	Désactiver les mises à jour auto
<input type="checkbox"/> WebP Express Désactiver Settings Provide coffee for the developer	Fournissez des images WebP générées automatiquement au lieu de jpeg/png pour les navigateurs qui prennent en charge WebP. Fonctionne sur n'importe quoi (images de bibliothèque de médias, galeries, images de thème etc.). Version 0.25.9 Par Egm Rosali Afficher les détails	Activer les mises à jour auto
<input type="checkbox"/> WPS Hide Login Réglages Désactiver	Protégez votre site Web en changeant l'URL de connexion et en empêchant l'accès à la page wp-login.php et au répertoire wp-admin aux personnes non-connectées. Version 1.8.16.7 Par WPShneur, NicolasLutka, wpformation Afficher les détails	Activer les mises à jour auto

Annexe 13 :

PARTENOR DIGITAL / WordPress sites / Campus ● Live

🔍 Aller à ou recherch... 🌐 🔔 ⓘ ⚙️ Aurel Monpierre ▾

Mise en cache

Cache edge

CDN

Mise en cache du serveur

Redis

Le cache edge stocke le cache de vos pages dans l'un des plus de 260 points de présence (PoP) de Cloudflare dans le monde, augmentant ainsi les performances et la vitesse des sites web jusqu'à 40 %.

Désactiver

✔ Le cache edge est activé

Cache mobile

Activer

Créez un cache de page supplémentaire spécifiquement pour les appareils mobiles. Le cache d'ordinateur de bureau continue à être servi normalement.

Vider le cache

Vider le cache

Cette opération permet de purger les données enregistrées dans le cache edge dans tous les emplacements. Le processus peut durer jusqu'à cinq minutes.

Vider le cache URL

Cette opération permet de purger les données de l'URL sélectionnée enregistrées dans le cache Edge à tous les emplacements. Le processus peut durer jusqu'à cinq minutes.

Commencez à saisir ou choisissez dans la liste ▾


Vider le cache URL

☐ Vider le cache de chaque sous-répertoire sous l'URL spécifié

Annexe 14 :

<input type="checkbox"/> Domaine ↑	État
<input type="checkbox"/> campus.partenordigital.com 🔗 Primaire	🔒 Vérification SSL en cours ⋮
*.campus.partenordigital.com	


Annexe 15 :



Surveillance New Relic

Obtenez des données détaillées sur les performances pour résoudre les problèmes. La surveillance a un impact sur les performances du site, n'oubliez donc pas de la désactiver.


Commencer la surveillance



Protection par mot de passe

Ajoute une simple protection .htpasswd à votre environnement.

Activer




Forcer le HTTPS

Redirigez le trafic de HTTP vers HTTPS pour une sécurité renforcée. Les moteurs de recherche préfèrent le HTTPS, la redirection a donc un impact positif sur le SEO.

Domaine principal

Modifier




Géolocalisation

Activez la géolocalisation pour avoir la possibilité d'ajouter des règles de redirection qui adaptent le contenu en fonction de l'emplacement IP des visiteurs du site web.

Activée pour le pays et la ville


Modifier



ionCube Loader

Activez ionCube Loader sur votre environnement si celui-ci possède une extension, un thème ou un script PHP qui utilise ionCube pour crypter son code.

Activer



Moteur PHP

Utilisez ces commandes pour basculer entre les différentes versions de PHP. Nous recommandons d'utiliser PHP 8.1 pour la sécurité et la stabilité.

PHP 8.3

Modifier

Annexe 16 :

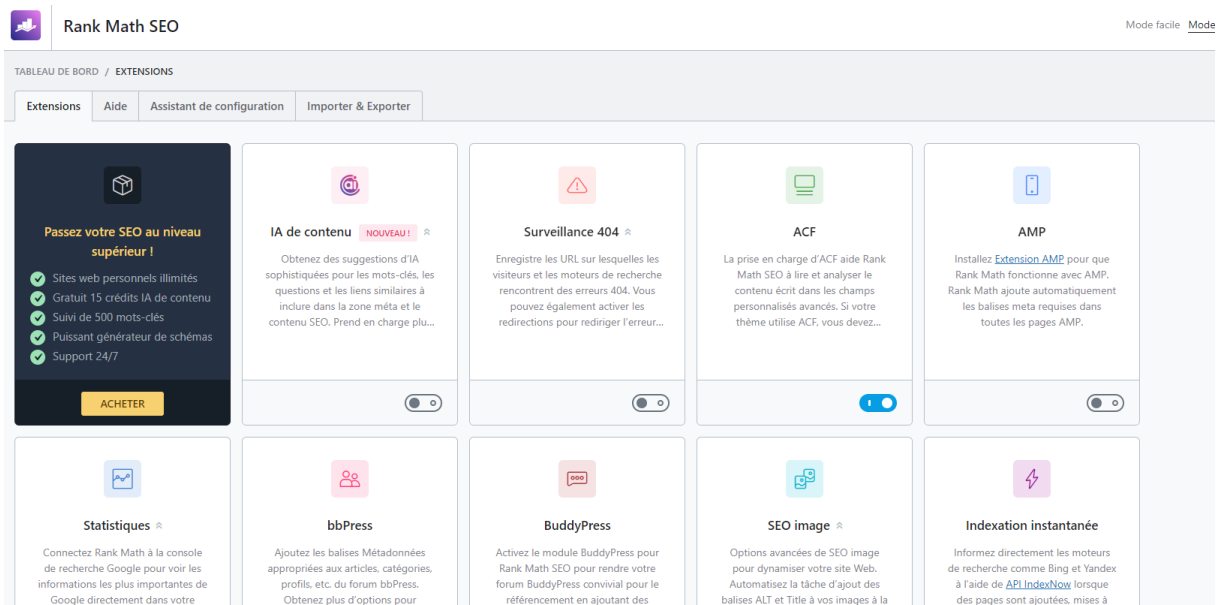
Liste des domaines

Ajouter un domaine

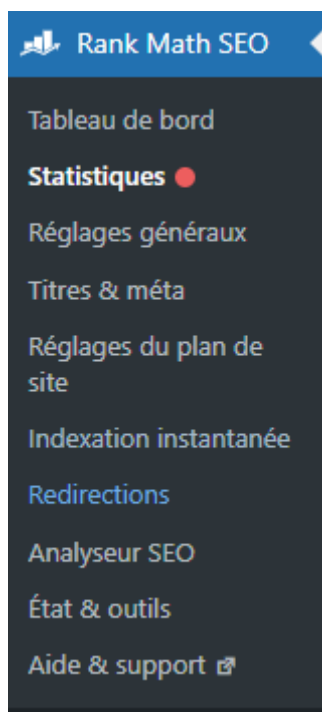
Les domaines listés ci-dessous avec un état de « Domaine en production » pointent vers ce site et sont couverts par l'intégration Cloudflare gratuite de Kinsta pour plus de rapidité et de sécurité. Les enregistrements DNS doivent être configurés correctement pour que vos domaines fonctionnent.

Rechercher des domaines	Tous les domaines
<input type="checkbox"/> Domaine ↑	État
<input type="checkbox"/> campus.partenordigital.com 🔗 Primaire	✓
<input type="checkbox"/> *.campus.partenordigital.com	
<input type="checkbox"/> campus.kinsta.cloud 🔗	✓
<input type="checkbox"/> *.campus.kinsta.cloud	

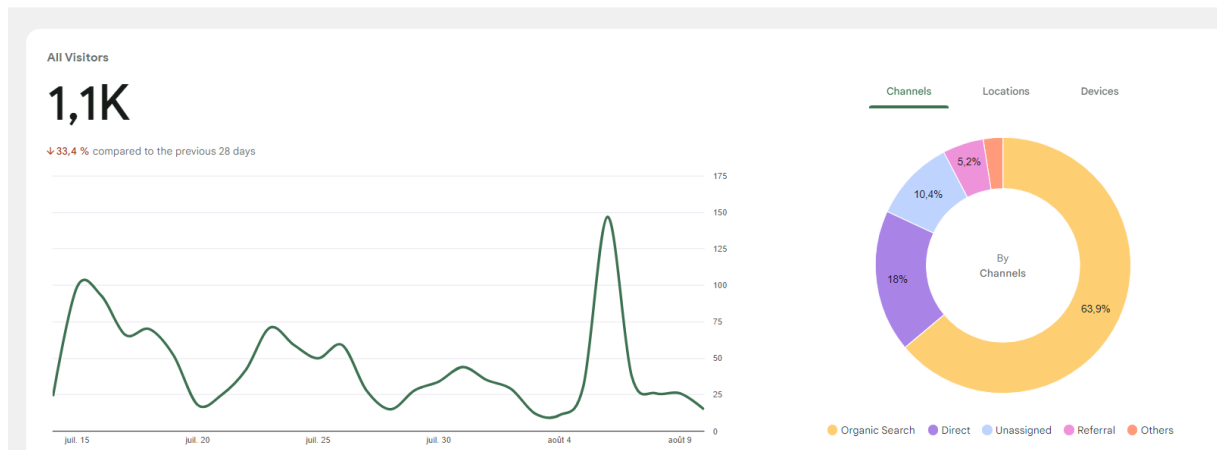
Annexe 17 :



Annexe 18 :



Annexe 19 :



Annexe 20 :

