



Développeur Web et Web Mobile

Prénom : Michel

Nom : Almont

Nom du projet : Charles-Cantin

Lien Github du projet : <https://github.com/malmont/front-charles-cantin>

Lien Github backend: <https://github.com/malmont/backen-charles-cantin>

Documentation github : <https://github.com/malmont/front-charles-cantin/tree/main/Documentation/>

URL du site (front-end) : <https://malmont.github.io/front-charles-cantin/>

URL du site (back-end) : <https://backend-strapi.online/admin/auth/login>

Description du projet

1. Liste des compétences du référentiel qui sont couvertes par le projet

- Maquetter une application.
- Développer une interface utilisateur web dynamique.
- Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

2. Résumé du projet en français d'une longueur d'environ 20 lignes soit 200 à 250 mots, ou environ 1200 caractères espaces non compris

Charles Cantin est un photographe de plus en plus reconnu qui souhaite acquérir un site web portfolio qui présente son travail et permette à ces futurs clients de communiquer rapidement avec lui.
Ce site aura pour but de développer son activité pour en vivre à terme.
Il souhaite qu'il y ait plusieurs sections obligatoires :

- Une section Page d'accueil
- Une section Galerie
- Une section Tarif
- Une section Contact

Tout le contenu du site devra être modifiable facilement par Charles à l'aide d'un CMS.

Il souhaite être autonome, il veut pouvoir modifier son copywriting, ces offres ou encore les images de sa galerie à volonté.

Pour faciliter la communication client, Charles-Cantin souhaite que le client puisse communiquer avec lui par Email, à l'aide d'un formulaire de contact.

Le client souhaite un site qui reflète sa créativité, il souhaite notamment qu'on puisse exploiter un logo qu'un ami lui aurait réalisé et que le site puisse proposer certaines animations visuelles originales.

Il y aura aussi un onglet galerie avec toutes ces anciennes réalisations rangées par section.

Enfin le site aura une section tarif, qui donnera rapidement aux clients une fourchette de prix selon les prestations qu'il propose.

3. Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet

Fonctionnalités de l'application

Une page d'accueil :

Charles aimerait qu'une image prenne toute la taille de l'écran.

Le texte "Charles Cantin - Photographe" devra être centré verticalement et horizontalement.

Une page galerie :

Ici, nous retrouverons les photographies que Charles souhaiterait mettre en valeur.

Chaque image fera partie d'une catégorie (voir annexes en fin de document) Horizontalement.

Un système de filtre dynamique doit permettre d'afficher les photos d'une catégorie en particulier sans avoir à recharger la page.

Une page "tarifs et prestations" :

Chaque prestation possède un titre, une petite description et un tarif (voir annexes en fin de document).

Une page "Contact" :

Un formulaire fonctionnel doit permettre de le contacter. Vous pouvez utiliser un service gratuit comme le plan starter de <https://formspring.io/>, ou profiter de netlify forms si vous choisissez cet hébergeur.

Fonctionnalités générales :

Depuis chacune des pages, une barre de navigation devra être disponible avec le menu ainsi que les icônes Facebook et Instagram (pour un lien vers son profil sur les réseaux sociaux à l'avenir).

Tout contenu devra être modifiable facilement par Charles à l'aide d'un CMS. Pour qu'il puisse apprécier le rendu du site avant de pouvoir le personnaliser, vous pouvez récupérer les images sur Pixabay (<https://pixabay.com/fr/>).

4. Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile

Spécifications techniques du projet :

Maquette/Illustration/Logo/Image

Pour la partie maquettage j'ai utilisé **Figma**.

Canva.com est l'outil que j'ai utilisé pour la documentation.

Environnement de travail

L'environnement de développement tournait sous **MacOs**

Vscode est mon éditeur de texte préféré.

Technologie front-end

Pour coder rapidement une interface responsive et efficace j'ai choisi l'outil **Bootstrap**.

J'ai utilisé aussi la bibliothèque de JavaScripts React js, pour optimiser mon code et générer des pages sans rechargement.

La bibliothèque React propose des outils ou des extensions comme par exemple JSX, qui permet de fusionner la logique interface utilisateur avec le langage JavaScripts.

Les avantages de react :

- **Facile à apprendre** : React est facile à apprendre et à comprendre tant que vous avez une bonne maîtrise des pré-requis. React dispose d'une solide documentation et de nombreuses ressources gratuites créées par d'autres développeurs en ligne via la communauté très active de React.
- **Composants réutilisables** : Chaque composant de React possède sa propre logique qui peut être réutilisée partout dans l'application. Cela réduit le besoin de réécrire le même morceau de code plusieurs fois.
- **Opportunités d'emploi** : Un plus grand pourcentage d'opportunités de développement web frontend en ce moment ont React comme l'une des compétences requises. Le fait de comprendre

le fonctionnement de React et de pouvoir travailler avec lui augmente donc vos chances de décrocher un emploi.

- **Performances améliorées** : Grâce au DOM virtuel de React, le rendu des pages peut se faire plus rapidement. En utilisant une bibliothèque de routage comme React Router, vous aurez différentes pages rendues sans aucun rechargement de page.
- **Largement extensible** : React est une bibliothèque qui assure uniquement le rendu de l'interface utilisateur de notre application. C'est au développeur de choisir les [outils avec lesquels il souhaite travailler](#), comme les bibliothèques de rendu de différentes pages, les bibliothèques de conception, etc.

Composant React-Js utilisés :

- styled-components (pour styliser mes composants)
- react-spring(bibliothèque d'animation)
- react-router-dom (bibliothèque de routage)
- react-icons (bibliothèque d'icone)
- gh-pages(Package pour déployer votre application sur GitHub-Page).

Technologie Backend

Comme CMS, nous avons utilisé l'outil STRAPI.

C'est un CMS (Content Management System) headless qui est Open Source et créé par une startup française. Le CMS a été développé sous NodeJs.

Créer une API avec Strapi n'est pas très compliqué, cela permet surtout de gagner du temps de développement sur vos projets.

MAIL

Formspree est le provider que nous avons choisi pour relayer les envois des formulaires par email.

Github

Pour la gestion du travail collaboratif, du versionning et de l'hébergement des dépôts du projet le tandem GIT/GITHUB a été utilisé.

Déploiement

Le déploiement a été effectué sur un serveur VPS Ubuntu.

Le serveur web utilisé est l'outil NGINX

Un certificat SSL chez le provider ZEROSL a été créé et téléchargé pour passer le serveur NGINX en HTTPS.

Sécurité

Sécurisation de l'environnement

Concernant la sécurité du serveur, j'ai installé le tandem Fail2ban le firewall UFW pour filtrer les paquets et limiter l'ouverture des ports du serveur.

Un pare-feu est essentiel lors de la configuration du VPS pour limiter le trafic indésirable sortant ou entrant dans votre VPS.

L'outil fail2ban permet de surveiller l'activité des logs de certains services, tel que SSH ou Apache. Lors d'un trop grand nombre d'authentifications ratées fail2ban va générer une règle IPTables, cette règle aura pour

but d'interdire pendant une durée déterminée les connexions depuis l'adresse IP susceptible d'être un attaquant.

5. Description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité

Passer son site web en HTTPS

Qu'est-ce que le HTTPS ?

HTTPS (l'acronyme pour Hypertext Transfer Protocol Secure) est la version sécurisée du HTTP, le protocole par lequel les données sont échangées entre votre navigateur et le site web auquel vous êtes connectés.

Le «S» à la fin de HTTPS signifie «sécurisé». Cela signifie que toutes les communications entre votre navigateur et le site web sont cryptées, c'est à dire que les données informatiques ne sont pas lisibles par un humain ou une machine.

Le protocole HTTPS est utilisé pour protéger les données de vos utilisateurs et des données sensibles (de paiement par exemple) et pour éviter les intrusions de hackers. Nous allons voir à la suite de cet article les objectifs de ces intrusions malveillantes.

Pourquoi le HTTPS devient une urgence ?

Google et les autres moteurs de recherche (Firefox et Safari) militent pour la sécurisation du web sur l'ensemble des sites internet pour protéger les internautes.

Impact du HTTPS dans votre référencement naturel

C'est un travail de fond que Google a entrepris dès 2014, avec l'annonce de la mise à jour de leur algorithme. Le HTTPS devient alors un critère de référencement naturel pour être mieux positionné dans les résultats de recherche.

Alerte "Non sécurisé" pour les sites web non HTTPS qui collectent des informations liées aux utilisateurs

En 2016 et 2017, Google confirme son envie de passer le web en HTTPS avec l’affichage d’une alerte, un label « Non sécurisé » pour l’ensemble des sites internet HTTP qui collectent des informations sur l’utilisateur (email, mot de passe, nom, prénom...) à travers un formulaire.

Alerte “Non sécurisé” pour l’ensemble des sites web HTTP

Sur sa lancée, Google a annoncé qu’il mettrait une alerte sur tous les sites internet non HTTPS d’ici 2018. « Finalement, nous prévoyons de marquer toutes les pages HTTP comme non sécurisées, et de remplacer l’indicateur de sécurité HTTP par le triangle rouge que nous utilisons pour HTTPS. » Google.

Dans la même logique que l’envie de pousser les sites internet à s’adapter aux tablettes et aux mobiles avec le responsive design, Google a clairement pris position sur le HTTPS avec des intentions matérialisés par des sites internet dédiés. Pour avoir plus de chiffres sur le sujet, Google met à jour sa vision du HTTPS dans le web avec un site internet qui s’intitule: [Chiffrement HTTPS sur le web](#) avec le pourcentage de site web HTTP présents dans chaque pays.

4 raisons pour passer votre site internet HTTP en HTTPS

1. HTTPS protège l’intégrité d’un site web des intrusions

Le HTTPS à travers un certificat SSL protège votre site internet des intrusions malveillantes.

C’est en quelque sorte comme une porte blindée que vous installez chez vous.

Les intrus, des hackers, peuvent exploiter chaque ressource non protégée qui se déplace entre votre site Web et vos utilisateurs (Images, cookies, scripts, HTML) et peuvent agir à n’importe quel endroit de votre réseau internet (Wifi, Fournisseur d’accès à Internet ...)

Le HTTPS vous protège de ces intrusions qui tentent d’installer des outils, logiciels malveillants ou insérer des publicités sur votre site internet par exemple.

2. HTTPS protège la confidentialité et la sécurité de vos utilisateurs

Le HTTPS empêche des intrus d’écouter les communications entre vos sites Web et vos utilisateurs. Chaque requête HTTP non protégée peut potentiellement révéler des informations sur les comportements et les identités de vos utilisateurs.

3. HTTPS: un impact en référencement naturel (SEO)

Comme évoqué précédemment, le HTTPS est devenu depuis 2014 un critère pris en compte dans l’algorithme de référencement de Google. Depuis, 50% des sites internet présents sur la première page de Google sont en HTTPS (+60 % en 9 mois).

Google référence mieux les sites internet qui sont en https et va automatiquement scanner les pages en HTTPS avant celles en http

4. Let’s Encrypt: Des certificats enfin gratuits

Let’s Encrypt est une autorité de certification gratuite, automatisée et ouverte, soutenue par Mozilla, Google ou Facebook, qui permet d’avoir des certificats SSL gratuits et donc de passer en HTTPS gratuitement.

La seule contrainte imposée par la certification Let's Encrypt est de devoir la renouveler tous les 3 mois au lieu de tous les ans.

6. Description d'une situation de travail ayant nécessité une recherche, effectuée par le candidat durant le projet, à partir de site anglophone

Lors du déploiement de l'application STRAPI sur un serveur VPS, je me suis appuyer sur cette documentation Technique officiel :

https://strapi.io/blog/deploying-strapi-to-your-own-vps?utm_campaign=Strapi%20Blog&utm_source=devto&utm_medium=blog&utm_content=strapiw

7. Extrait du site anglophone, utilisé dans le cadre de la recherche décrite précédemment, accompagné de la traduction en français effectuée par le candidat sans traducteur automatique (environ 750 signes).

Deploying Strapi to Your Own VPS

TRADUCTION

Déployer strapi sur votre serveur VPS.

There are multiple cloud options to deploying your Strapi app, but, in this article, I will walk you through how to install your Strapi instance on a Virtual Private Server for usage.

The article will be focused on introducing you to setting up an existing Strapi App and deploying on a VPS.

What is Strapi?

Strapi is an open-source headless Content Management System that offers you the chance to develop API with distinctive features. It's built on Node.js and works with any GraphQL enabled Frontend Frameworks. You are able to create custom content types, relationships and media library for images and audio files with Strapi.

What is a VPS?

A Virtual Private Server is a multi-tenant *infrastructure as a service* cloud-hosting provided for developers to host their website and application by hosting platform.

Goals

At the end of this guide, you should be able to set up a VPS, install your core requirements, and clone and deploy your Strapi project on VPS.

Prerequisites

Before you continue with this content, you require the following.

1. Basic knowledge of JavaScript
2. Basic understanding of Strapi - [get started here](#).
3. Github Account
4. Git installed on your Computer
5. Download and install [Node.js v14+](#)
6. VScode or any other text editor

Strapi Hardware Requirements

- At least 1 CPU Core (Recommended: 2 CPU Cores)
- Node Js Version 12 or 14 (Odd Number releases aren't supported)
- Minimum 2GBs RAM (Recommended: 4GBs)
- Disk Space Minimum of 32GBs

Supported Operating System

- Ubuntu >= 18.04 (LTS-Only)
- Debian >= 9.x
- CentOS/RHEL >= 8
- macOS Mojave or newer (ARM not supported)
- Windows 10
- Docker - [docker repo](#)

TRADUCTION

Il existe plusieurs options cloud pour déployer votre application Strapi, mais, dans cet article, je vais vous expliquer comment installer votre instance Strapi sur un serveur privé virtuel.

L'article se concentrera sur la présentation de la configuration d'une application Strapi existante et du déploiement sur un VPS.

Qu'est-ce que Strapi ?

Strapi est un système de gestion de contenu open source sans tête qui vous offre la possibilité de développer une API avec des fonctionnalités distinctives. Il est construit sur Node.js et fonctionne avec tous les frameworks frontaux activés par GraphQL.

Vous pouvez créer des types de contenu personnalisés, des relations et une bibliothèque multimédia pour les images et les fichiers audio avec Strapi.

Qu'est-ce qu'un VPS ?

Un serveur privé virtuel est une infrastructure multi-locataire en tant que service d'hébergement dans le cloud fourni aux développeurs pour héberger leur site Web et leur application par une plate-forme d'hébergement.

Buts

À la fin de ce guide, vous devriez être en mesure de configurer un VPS, d'installer les prérequis de base, de cloner et de déployer votre projet Strapi sur VPS.

Conditions préalables

Avant de continuer avec ce contenu, vous avez besoin des éléments suivants.

- Connaissance de base de JavaScript
- Compréhension de base de Strapi - commencez ici.
- Compte GitHub
- Git installé sur votre ordinateur
- Téléchargez et installez Node.js v14+
- VScode ou tout autre éditeur de texte
- Configuration matérielle requise pour Strapi
- Au moins 1 cœur de processeur (recommandé : 2 cœurs de processeur)
- Node Js Version 12 ou 14 (les versions de numéro impair ne sont pas prises en charge)
- Au moins 2 Go de RAM (recommandé : 4 Go)
- Espace disque Minimum de 32 Go

Système d'exploitation pris en charge

- Ubuntu >= 18.04 (LTS uniquement)
- Debian >= 9.x
- CentOS/RHEL >= 8
- macOS Mojave ou plus récent (ARM non pris en charge)
- Windows 10
- Docker - dépôt docker

8. Autres ressources

9. Informations complémentaires

--