**VERGIER Guillaume**

***Dossier Projet***

Formation Gaming Campus

Titre

**Développeur Web – Web Mobile**

08/07/2023

Page 1/30

**Listes des compétences du référentiel étant couvertes par le projet**

**Présentation d'un projet réalisé en amont de la session**

1. Développer la partie front-end d’une application web ou web mobile en intégrant les recommandations de sécurité

* Maquetter une application
* Réaliser une interface utilisateur web statique et adaptable
* Développer une interface utilisateur web dynamique
* Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

1. Développer la partie back-end d’une application web ou web mobile en intégrant les recommandations de sécurité

* Créer une base de données
* Développer les composants d’accès aux données
* Développer la partie back-end d’une application web ou web mobile
* Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce
* Utiliser l’anglais dans son activité professionnelle en développement web et web mobile
* Actualiser et partager ses compétences en développement web et web mobile

Page 2/30

**Sommaire**

**Introduction :…………………………………………………………………………………………………………………….4**

**Remerciements : ………………………………………………………………………………………………………………4**

**Résumé du projet : …………………………………………………………………………………………..………………5**

**Informations complémentaires :……………………………………………………………………………………….6**

**Cahier des charges : …………………………………………………………………………………..…………………….7**

**Brief…………………………………………………………………………………………………………..…………………….7**

**Spécifications techniques du projet……………………………………………………………..…………………..8**

**Le maquettage (résumé)…………………………………………………………………………………..……………..9**

**Le zoning…………………………………………………………………………………………..…………………10**

**Le wireframe………………………………………………………………………………………………………..11**

**Le wireframe 2…………………………………………………………………………………………..………..12**

**Le mockup……………………………………………………………………………………………………………13**

**Le prototype………………………………………………………………………………………………………..14**

**Le front-end Formulation………………………………………………………………………………………..………15**

**HTML…………………………………………………………………………………………………………………..16**

**HTML CSS…………………………………………………………………………………………………………….17**

**HTML CSS JavaScript…………………………………………………………………………..……………….18**

**Le back-end Formulation………………………………………………………………………………….…………….19**

**Création de la base de données………………………………………………………..…………….…..20**

**Rendre le site dynamique……………………………………………………………………………………21**

**Rendre le site dynamique 2…………………………………………………………………………………22**

Page 3/30

**Introduction**

Passionné de jeux vidéo et de technologie dès mon plus jeune âge, j’ai vraiment fleuri et poursuivi ce projet sur le long terme en m’y intéressant sur mon temps libre.

Issue d’un baccalauréat des sciences de la technologie, du management et de la gestion, j’ai pu apprendre les premières approches de la technologie avec une matière sur le développement.

J’ai donc longtemps songé à m’orienter vers le domaine professionnel du développement, en suivant une filiale différente de celle de mon père, qui lui est dans le domaine de l’informatique et du dépannage réseau.

En parallèle, j’ai préparé un concours d’entrée en Gendarmerie où j’ai pu intégrer un peloton d’intervention. Je suis désormais à l’heure d’aujourd’hui réserviste.

Ma curiosité sur l’apprentissage de nouvelles choses me démange, et c’est ainsi que depuis février 2023 et jusqu’au 25 juin 2023, je suis les cours de Gaming Campus en formation, en vue du titre de Développeur Web et Web Mobile.

**Remerciements**

Je tiens tout d’abord à remercier Jean-Baptiste RACOUPEAU, ayant été mon supérieur pendant de nombreuses années dans une filière de Gaming Campus, et désormais mon directeur de campus pour l’accompagnement et les conseils qu’il a pu me fournir tout au long de cette formation.

Je tiens aussi à remercier Daniel TINEBRA pour son professionnalisme, son accompagnement, ses explications tout au long de la période de formation du front-end.

Je tiens également à remercier Jean-François BARON, pour sa démarche, ses conseils, sa pédagogie et son professionnalisme, les heures dédiés à l’accompagnement et son envie pour nous de réussir tout au long de la période de formation du back-end.

Pour terminer, je souhaite remercier mes collègues de promotion Stéphane, Alexis, Anaïs, Tristan pour cette épreuve qui semblait au départ compliqué, avec les heures de pair-coding que nous avons réalisé, et les heures conviées au développement.

Page 4/30

**Résumé du projet**

Nous avions comme tâche de trouver un projet sur lequel nous allions devoir travailler et développer au cours de notre formation en dehors des exercices donnés par le formateur, ou trouvés sur internet. Mon projet était de faire un site web pour mon père (étant informaticien administration réseaux de métier), qui ferait également passerelle pour moi, et permettant de faire ma propre publicité avec le référencement, et le respect de la sémantique.

Nous avons débuté sur le maquettage sur l’environnement du logiciel Figma, où nous avons pu avoir les bases de l’utilisation de celui-ci, ainsi que les facteurs qui le caractérisent : le Zoning, le Wireframe (mobile et desktop), le Mockup (mobile et desktop) et le Prototypage.

J’ai donc réalisé tout ce qui était nécessaire à encadrer le maquettage de mon projet de formation.

Le maquettage est terminé, et à l’issue nous découvrons l’utilisation de l’environnement de développement intégré Visual Studio Code, avec toutes ses fonctionnalités, extensions.

Par le biais de VSC, j’ai commencé à l’aide du formateur mais aussi de mes recherches à me pencher sur le développement HTML et CSS (sans Framework) de mon projet, grâce à l’aide visuelle de ma maquette réalisée au préalable.

Le développement front est terminé, et voilà que je découvre l’utilité des Framework, dont Bootstrap pour le HTML et CSS. Grâce à Bootstrap, j’ai pu réaliser ma propre barre de navigation comme je le souhaitais, et gagner un temps imbattable sur les langages front-end pour le côté responsive, et intégrer les classes CSS directement sur l’extension HTML.

Nous avons utilisé également GitHUB de façon à pouvoir commit et push notre travail réalisé, et pouvoir le visionner sur un site dédié de GitHUB en interne.

Voilà que le back-end prend également sa place avec la programmation PHP, la protection contre les scripts JavaScript, les attaques CSRF, les failles SQL. Une partie formulaire est aussi développée et la réalisation d’un back-office pour les articles.

Nous utiliserons également le Framework Symfony pour faciliter l’utilisation de mise en place d’un Modèle Vue Controlleur.

Ainsi, nous mettons en place le développement d’une base de données via Laragon et HeidiSQL, et utiliseront certaines requêtes SQL utiles au projet.

Page 5/30

**Informations complémentaires concernant la présentation d'un projet réalisé en amont de la session**

Pour commencer, cette formation fut très enrichissante pour nous préparer sur le métier de développeur web. Etant en voie de reconversion professionnelle, j’ai toujours été attiré par la programmation, le codage, l’algorithmie et développer une certaine logique.

Le début de l’immersion fut complètement vague pour moi.

Un cahier des charges nous a été donné pour le front-end et le back-end, ainsi que ce que l’on attend de nous via le passage devant le jury, les compétences nécessaires puis le développement d’un projet personnel.

On nous a donc montré chaque étape à développer, de façon à ce que tout se déroule bien. Un énorme travail personnel est donc demandé, beaucoup de recherche et d’investissement ont pris place.

M’étant concentré sur les compétences nécessaires à l’obtention de ce titre, j’ai réalisé quelques recherches en autonomie de façon à ce que mon site soit le plus optimisé possible.

Ayant développé quelques « bases » en HTML avant cette formation, lors de mon Baccalauréat, je me suis retrouvé un peu dans mon élément.

Je nommerai le projet « **VERGIER Consulting** ».

Page 6/30

**Cahier des charges**

**Le brief**

**Stratégie** : être visible sur le web, avoir la possibilité de rédiger des articles dans un back-office, un front-office sera à venir de façon à développer une partie prise de rendez-vous.

**Front-office** :

* 5 pages comportant la navigation possible avec les liens actifs (Accueil, Métiers, Projet, Contact, A propos)

La page « **Accueil** » : Est présente sur les pages « Métiers », « Projet », « Contact », « A propos », comportera une phrase « d’accroche » pour le site.

La page « **Métiers** » : Comportera deux bulles avec deux images indicatives sur les métiers de Guillaume VERGIER et ALAIN VERGIER, avec une description rapide.

La page « **Projet** » : Comportera le logo de Gaming Campus (lien re dirigeable), le nom du titre et le but final de la formation

La page « **Contact** » : Comportera un simple formulaire pour connaître la demande du client. Une partie RGPD est en développement de façon à ce que le client accepte obligatoirement les conditions générales d’utilisation avant de pouvoir envoyer sa demande.

La page « **A propos** » : Présentera mon client, en tant que gestionnaire et traitement d’informations, et ma personne en tant qu’éditeur du site et développeur web.

**Fonctionnalités du front-office** :

* Formulaire de contact
* Création d’un calendrier de prise de rendez-vous (idée à venir)

**Back-office** :

* Mettre en place un back-office pour administrer et avoir la possibilité de mettre à jour les contenus (articles) ainsi que la base de données.

**Fonctionnalités du back-office** :

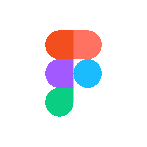
* Gestion des articles du blog (les créer, éditer ou supprimer)
* Avoir accès à la partie administrable par le biais d’un modal de connexion utilisateur (login/password)
* Déconnexion possible
* Interface dédiée pour la gestion des médias visibles (images)

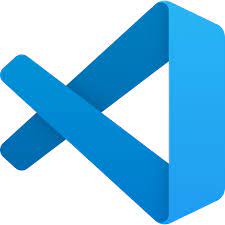
Page 7/30

**Spécifications techniques du projet élaborées par le candidat**

**Google Chrome pour l’utilisation de visualisation du site**

 **Bootstrap pour le framework HTML CSS**

** Figma pour le maquettage du site**

** VSC comme environnement de développement**

** HTML5 pour l’infrastructure statique du site**

**CSS3 pour la personnalisation de l’infrastructure du site**

** JavaScript pour la partie dynamique (front-end)**

**PHP pour le développement du back-end**



**Symfony comme framework PHP**

** HeidiSQL pour la création de base de données**

**GitHUB pour la création d’un site fictif + sauvegarde du code** Page 8/30

**Le Maquettage**

Le maquettage fût globalement une certaine expérience.

N’ayant jamais utilisé de logiciels de design, de maquettage, ou de création, j’ai réussi à optimiser complètement les idées données par mon client. A l’aide des cours, de divers exercices, de l’utilisation de plateformes proposant du contenu (YouTube) pour des démonstrations, j’ai pu réaliser ma maquette comme je le souhaitais.

J’ai découvert tout type de fonctionnalités, utilisé des frames, des incrustations d’images…

J’ai donc pu démontrer tout ce qui était principalement demandé :

* Montrer l’aspect visuel de toutes les pages et/ou interfaces du site
* Montrer les fonctionnalités du site
* Formaliser les enchaînements des interfaces
* Réaliser de façon obligatoire le responsive design
* Avoir la possibilité de réaliser le Mobile First en cas de demande du client

Pour commencer, j’ai réalisé un « brief » de façon à connaître complètement les intentions et envie du client, ce qui me permet aussi de réaliser un cahier des charges pour :

* Connaître les besoins du client
* Développer l’arborescence du site
* Zoning essentiel pour chaque page
* Définir le rôle de chaque page
* Atouts de communication
* Cas d’utilisation (front-office) ?
* Cas d’utilisation (back-office) ?
* Appréhender l’esthétique et l’ergonomie du site
* Gagner du temps en évitant de coder sans validation
* Définir le contexte du site
* Connaître le cas d’utilisation du client

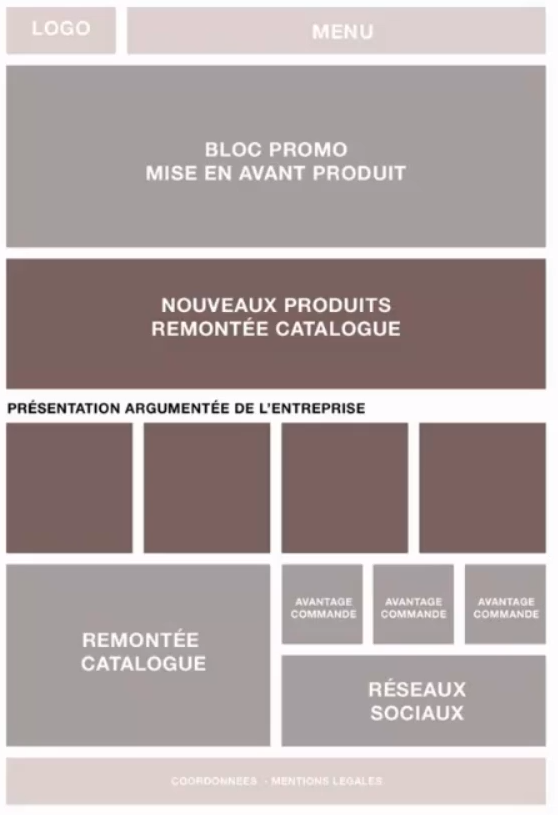
Ainsi, une fois le brief réalisé, la suite du maquettage peut prendre place, et les facteurs principaux peuvent prendre place.

Annexe : Brief Maquettage Figma

Page 9/30

**Le Maquettage**

**Le zoning**



Je commence par effectuer le zoning des pages avec le client, pour identifier les principales zones, lister les contenus efficacement pour avoir une vue globale sur le projet.

Cette étape permet de vérifier qu’aucune fonctionnalité ou information n’ont été oubliées et que l’ergonomie répond bien aux objectifs fixés, pour chaque page.

Toutes les informations sont donc recueillies pour l’aspect design, le wireframe prend désormais place de façon à montrer l’agencement les parties structurées composant une page web.

Page 10/30

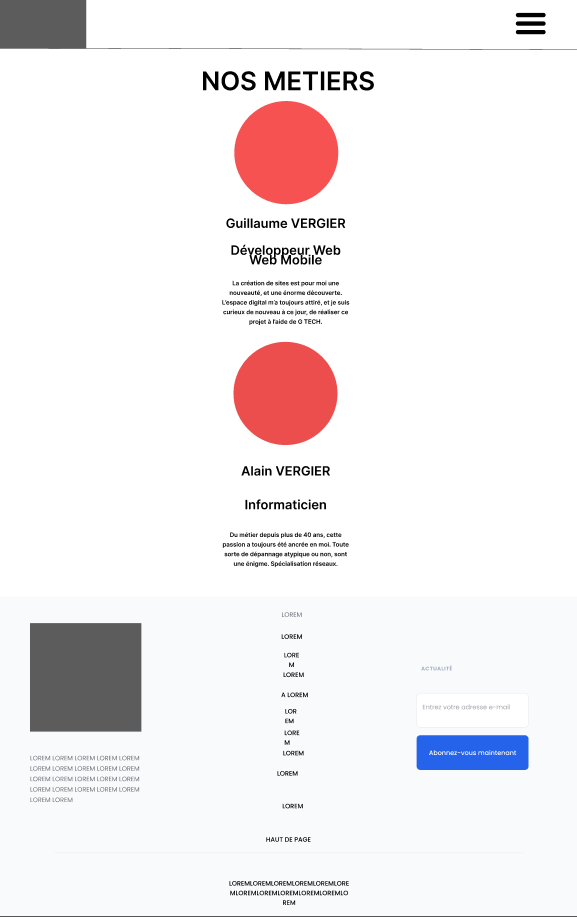
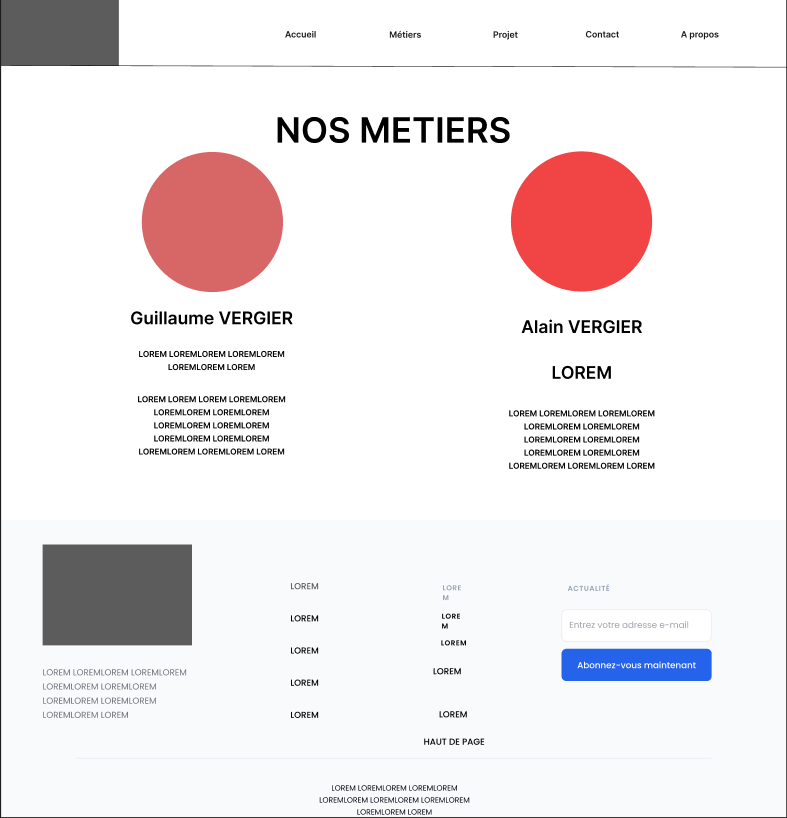
**Le Maquettage**

**Le wireframe**

Le wireframe m’a permis globalement de rajouter des zones de texte, des emplacements d’images, vidéos ou liens et de rajouter des éléments graphiques.

J’ai donc réalisé un wireframe mobile, et un wireframe desktop.

Voici un exemple avec la page de mon site web : NOS METIERS.



DESKTOP MOBILE

J’ai donc pu réaliser mon wireframe pour les 5 pages demandées lors du zoning avec le client.

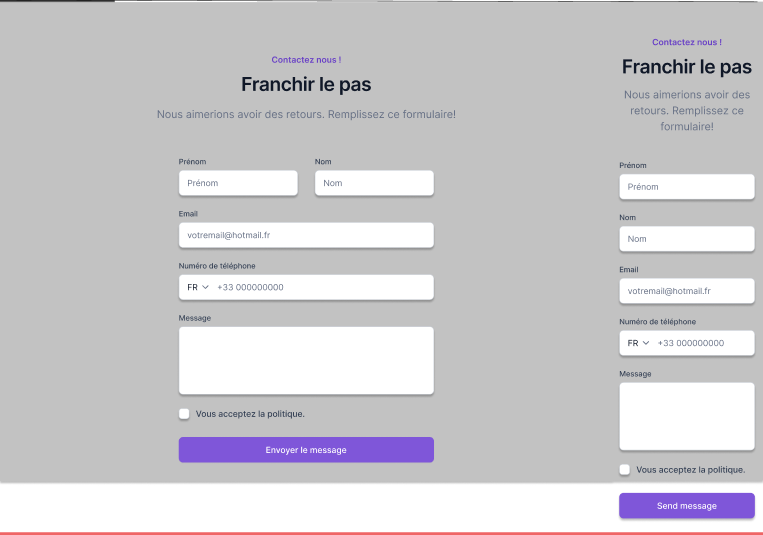
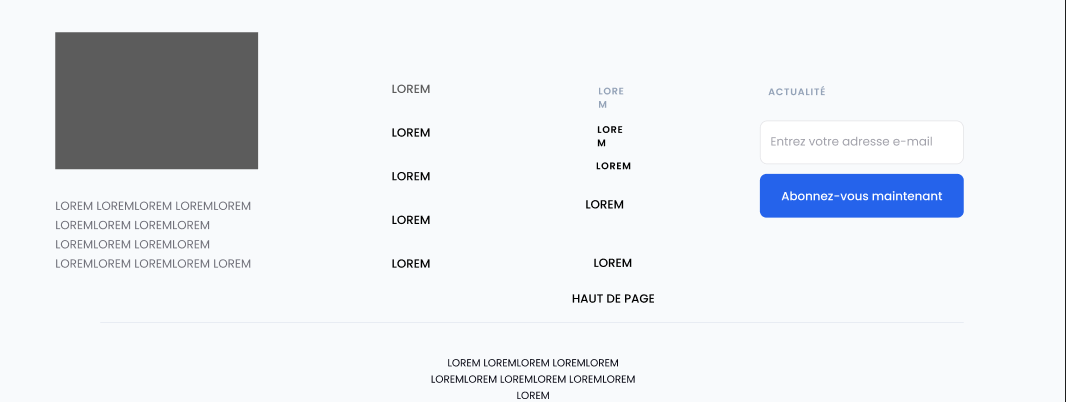
Page 11/30

**Le Maquettage**

**Le wireframe**

J’ai également pu découvrir toutes les fonctionnalités techniques de Figma, tel que les grids (colonnes) pour que les pages soient construites sur de bonnes bases et sur une bonne position, des contraintes pour la position ne soit pas décalée, une police et une taille de texte adaptée, les frames pour réaliser un bloc important…

J’ai également découvert l’utilité de Figcomponents, permettant d’utiliser des composants en tout liberté, permettant de construire sa page comme nous le souhaitons.

Voici un exemple avec le footer de chaque page, et la partie formulaire de ma page contact :

Page 12/30

**Le Maquettage**

**Le mockup**

Ici, le mockup graphique est très important. Cela m’a permis d’améliorer l’ergonomie de l’interface, et l’expérience avant de créer le prototype des pages.

Ma dernière partie a donc été réalisé en rajoutant au wireframe des touches finales, comme les plaquettes de couleur, l’architecture de l’information, l’interaction avec l’utilisateur, les icônes et images, la typographie et la hiérarchie visuelle de tous les éléments.

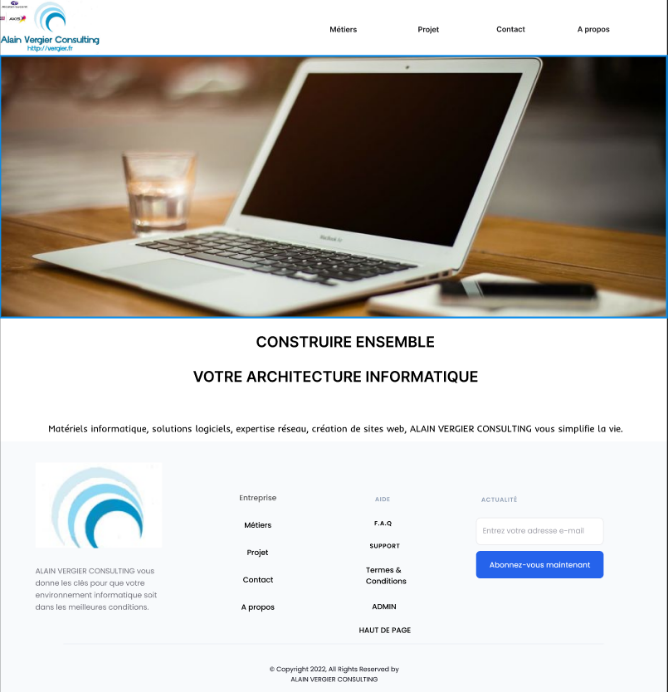
A l’issue, j’ai donc modifié les composants que j’ai utilisé de façon à ce que je puisse l’utiliser à ma convenance.

J’ai donc pu mettre le logo prévu à cet effet, modifier du texte et de la police, et rajouter des images prédéfinies.

En travaillant toujours sur le mobile first, je précise également que le responsive design était essentiel sur ce projet.

Le but était de réaliser les pages complètes et remplies d’informations, de façon à ce que le client puisse visualiser en phase finale les 5 pages du site web.

Je suis donc resté sur un design simple, mais efficace.



PJ : Site web

Page 13/30

**Le Maquettage**

**Prototype**

Une fois le mockup complètement terminé, je me suis occupé du prototypage. L’objectif de celui-ci était de représenter l’interface avec son aspect graphique, son design, la mise en page précise, et ses interactions principales.

J’ai enfin pu représenter l’interface générale, étant la plus proche du rendu final.

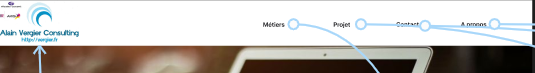
L’avantage du prototype m’a permis de créer une interface qui ressemble le plus possible aux résultats voulus, ce qui permet également de réaliser des tests utilisateurs, et d’avoir des retours rapidement sur le produit global.

Voici l’exemple de prototype complet réalisé sur Figma pour ce projet :



Ici, j’ai pu donner une logique à chaque contenu utilisé sur chaque page, par exemple le fait de rendre des liens et des images cliquables.

Chaque fonctionnalité du header ou du footer a donc son chemin pour accéder à une page, ou à une autre.



La maquette est validée, et l’orientation vers le code est donc désormais réalisée.

Page 14/30

**Le Front-end**

**Formulation**

J’ai donc réalisé tout ce qui était nécessaire à encadrer le maquettage de mon projet de formation.

Le maquettage est terminé, et à l’issue nous découvrons l’utilisation de l’environnement de développement intégré Visual Studio Code, avec toutes ses fonctionnalités.

Nous avons alors travaillé sur Visual Studio Code sur les extensions que nous a donné le formateur pour que notre travail soit le plus saint possible, tel que « HTML Preview », « Live Server », « open in browser », et j’ai rajouté moi-même un thème de couleur étant nommé « Tokyo Night » avec mes recherches personnelles.

Par le biais de VSC, j’ai commencé à l’aide du formateur mais aussi de mes recherches à me pencher sur le développement HTML et CSS (sans Framework) de mon projet, grâce à l’aide visuelle de ma maquette réalisée au préalable.

Tout le développement se faisait à la main, à l’aide également de la technologie EMMET de façon à travailler plus rapidement.

J’ai donc commencé à coder ma première page index à la main, et reproduis la partie maquette sans oublier des notions très importantes lors de ma progression :

* Le langage de syntaxe
* L’utilisation de balises
* Apprentissage de la technologie EMMET
* Organiser les fichiers de son projet
* Travailler l’ossature des pages

J’ai découvert également le critère de performance : la sémantique, permettant de donner du sens aux choses, de façon à ce que chaque code ait une raison d’exister.

Avec la sémantique, j’essaie d’être le plus propre possible sur mon code de façon à pouvoir optimiser le référencement que je peux être amené à mettre en avant.

Une fois ma session de code terminé, j’utilise le site du « W3C Validator » permettant de valider et corriger complètement mes erreurs, et si nécessaire de modifier une partie de code ciblée.

Je me suis énormément aidé de la doc officielle HTML via l’espace développeur Mozilla.

J’utilise aussi hors projet des sites d’entraînement pour quelques sessions d’exercices comme Codacademy, OpenClassroom, CodinGame, suivi quelques tutoriels en ligne de GrafiKart (pour certaines explications plus poussées), et quelques tours sur la Flexbox Froggy.

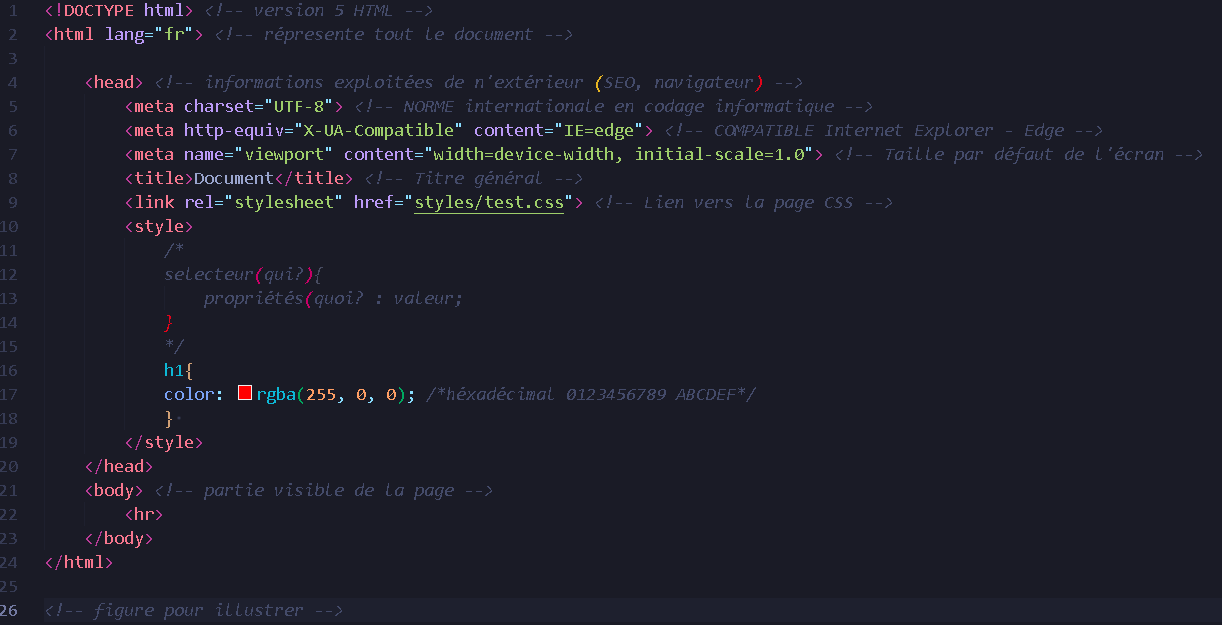
Page 15/30

**Le Front-end**

**HTML**

J’ai commencé par développer et comprendre le DOCTYPE HTML, ainsi que l’ensemble du contenu présent dans la balise <head>.

J’ai également appris l’existence de la balise <link> permettant de lier toute sorte de fonctionnalités, tel que les fonts pour la police précise, font-awesome pour les icones, et le lien permettant de relier un fichier HTML au CSS.



J’ai énormément utilisé de commentaires sur un brouillon pour comprendre ce que j’écrivais, et pouvoir utiliser au propre les balises ainsi que leurs utilisations.

Page 16/30

**Le Front-end**

**HTML CSS**

Je suis resté sur du HTML CSS brut pendant approximativement 3 semaines sur mon développement (dans le projet, et hors projet avec exercices). Cela m’a permis d’enrichir mes connaissances sur ces deux langages et de maîtriser les balises principales, pour développer des automatismes lors de développement d’autres sites web, d’autres clients.

J’ai désormais découvert l’utilisation du framework Bootstrap, qui m’a permis de gagner un temps record sur le développement de mon front-end.

J’ai pu utiliser la bibliothèque bootstrap qui met à disposition la navbar.

La navbar est donc inclue dans mon front-end, et j’ai pu la modifier comme je le souhaitais.

Celle-ci est donc également responsive et adaptable à tout type de version et taille d’écrans depuis lequel l’utilisateur se connectera au site :

* Mobile
* Tablette
* Ordinateur



Page 17/30

**Le Front-end**

**HTML CSS JavaScript**

L’utilisation de scripts est également présente pour pouvoir mettre en place Bootstrap.



Tout cela permet de gagner du temps sur la Flexbox, qui peut être inclus sur les modifications de classes dans les balises.

La partie blog (articles) a été mis en place via Bootstrap.



Celle-ci est également responsive, c’est-à-dire qu’il y a 3 articles pré-créés et visibles pour les écrans d’ordinateur de gauche à droite, et 3 articles de haut en bas pour les versions mobiles.



Desktop Mobile

Page 18/30

**Le Back-end**

**Formulation**

La partie front-end statique est donc terminée, et il est désormais temps de laisser place au côté programmation du projet.

Toujours sur l’environnement de développement Visual Studio Code, nous avons réalisé multiples exercices sur les besoins importants qu’aurait besoin notre site web, sur le langage PHP.

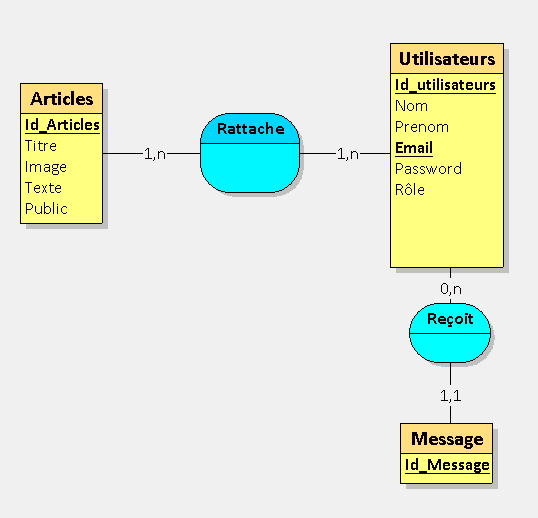
Comme une partie back-office est officiellement prévue de façon à pouvoir administrer les formulaires (avec système de mailing), mais aussi une gestion indépendante avec session administrative du blog (articles), j’ai donc pu réaliser ce que je souhaitais intégrer.

Nous avons alors appris à gérer une base de données à l’aide du logiciel HeidiSQL. L’avantage est que nous avons pu créer des requêtes SQL pour créer des tables, et leurs rajouter des propriétés.

Je précise également que la modélisation la base de données a été réalisée en utilisant la méthode MERISE. La réalisation du schéma MCD (Modèle Conceptuel de Données) a été réalisée par le biais du logiciel Looping.

**Le système de gestion de base de données relationnelle (SGBDR)**

Pour commencer, j’ai utilisé Looping pour modéliser visuellement la base de données sous la méthode MERISE.

La base de données de VERGIER Consulting contient donc un total de 3 tables (Utilisateurs, Articles, Message). Prenons en compte que la table « Utilisateurs » et la table « Articles » sont en utilisation directe, mais la table « Message » sera prévue en développement pour plus tard, à l’issue de la création d’un front-office. Voici son schéma MCD : Page 19/30

**Le Back-end**

**Création de la base de données**

Pour continuer sur mon projet, j’ai installé le framework **SYMFONY**. J’ai donc à l’issue installé **Composer**.

Avant de créer la base, il m’a fallu alors configurer le fichier .env de SYMFONY avec les informations de ma database.

La création de la base de données s’est alors faite avec **Doctrine**, étant un ORM (Object-Relational Mapping). Symfony intègre alors Doctrine, qui facilite la gestion de la persistance des données, et de préparer les requêtes SQL avant injection dans la base de données. Il permet de travailler avec des objets plutôt qu’avec des requêtes SQL brutes, simplifiant ainsi l’interaction avec la base de données.

****

Pour continuer la structuration de la base de données, je créé alors les entités, et leurs propriétés.



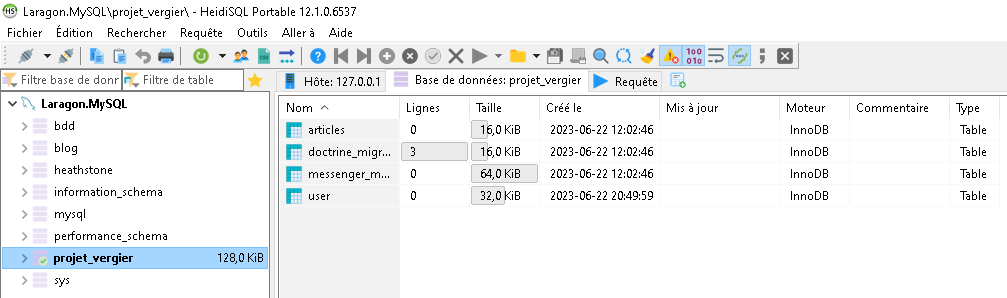
A l’issue, je créé le script SQL.



Dernière étape : injection dans la base de données via Doctrine (ORM).



Ma base de données est désormais créée, je peux alors naviguer et regarder si celle-ci est en place grâce à l’une des fonctionnalités de **Laragon** : **HeidiSQL**.



Page 20/30

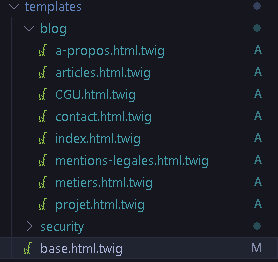
**Le Back-end**

**Rendre le site dynamique**

Au cours de la progression back-end, j’ai pu exploiter le moteur de template moderne puissant qui n’est d’autre que **Twig**.

Il m’a permit d’offrir une syntaxe simple et expressive pour générer les vues de mes pages. Twig m’a facilité la gestion des templates, et à améliorer grandement la lisibilité du code.

Un autre avantage : il intègre des mécanismes de sécurité pour prévenir les attaques (XSS par exemple, Cross-site Scripting) et d’autres failles de sécurité liées aux templates. Il échappe automatiquement les données insérées dans les templates pour s’assurer qu’elles sont affichées de manière sécurisée.



J’ai donc mis à jour l’ensemble de mes pages pour les rendre dynamiques, et à l’issue pouvoir créer un MVC (Modèle Vue Controlleur) à l’aide de Symfony.

J’ai à l’issue personnalisé le fichier base.html.twig pour importer le CSS de mes vues, ajouté le header ainsi que le footer.

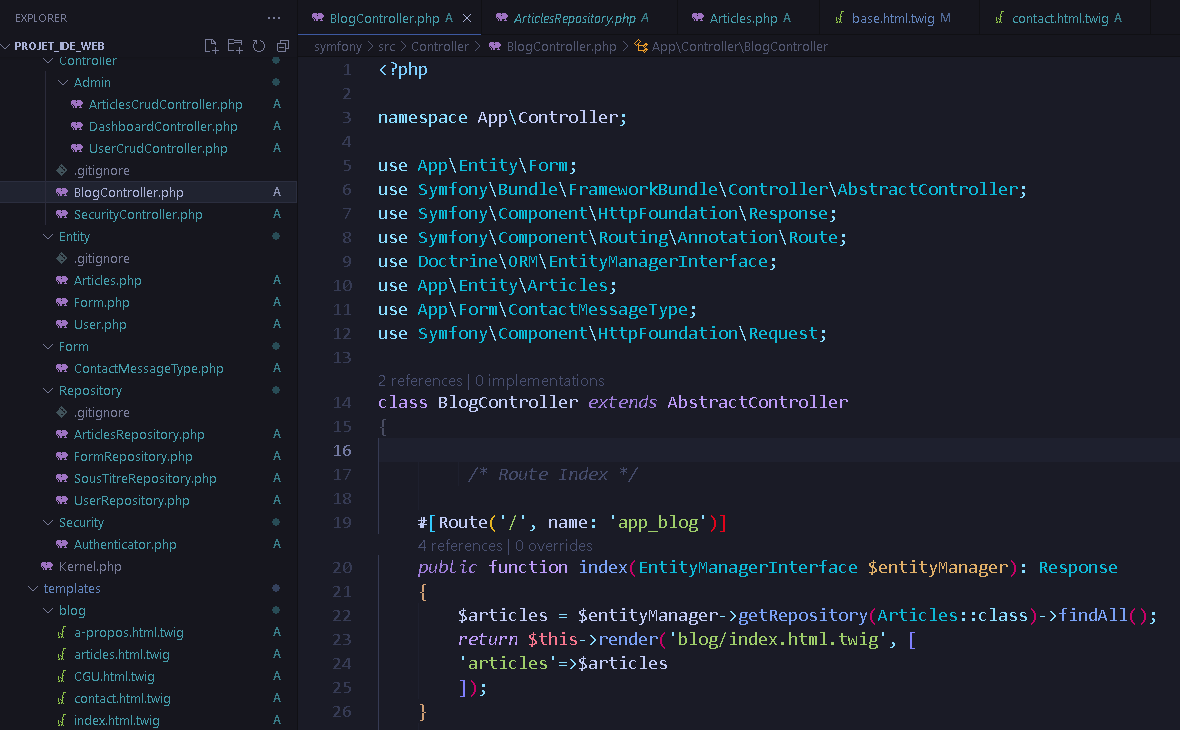
J’ai pu créer pour ma partie article le « BlogController » permettant de créer des « routes » pour mes vues.

Page 21/30

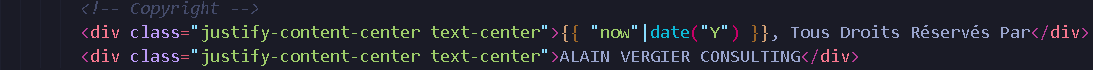
**Le Back-end**

**Rendre le site dynamique**





Grâce à la syntaxe Twig, j’ai pu personnaliser certaines « dates » qui se règleront automatiquement.



Page 22/30

**Veille effectuée sur la sécurité des sites web**

**Formulation**

Dans ma formation, j’ai fait des recherches sur la sécurité générales sur les sites et / ou une application web sur internet, et les réflexes à adopter de façon à pouvoir se protéger au maximum.

Je suis tombé sur la page web developper.mozilla.org qui expliquait tout le rayon sécurité, les risques, mais aussi la protection à adopter pour éviter tous problèmes.

J’ai pu noter 3 grandes menaces principales qui peuvent être un danger de sécurité :

* **Cross-Site Scripting (XSS)**

Cette attaque permet à l’attaquant d’injecter des scripts directement exécutés par le navigateur.

Via ce script, l’attaquant peut donc récupérer en toute liberté les cookies d’authentification par exemple, et ainsi se connecter au site via une session existante.

En fonction du site sur lequel l’attaquant se trouve, les actions peuvent alors entraîner de très gros problèmes (dans un exemple typique : récupérer les accès bancaires d’un client).

**Pour contourner cette faille,** il faut supprimer ou désactiver les balises exécutables existantes, contenant potentiellement du code malveillant, comme par exemple la balise <script>.

Pour comprendre et augmenter la protection du site se trouvant principalement dans l’interaction avec les visiteurs (exemple : formulaire).

Il faudra donc rajouter une ligne de commande permettant à l’utilisateur de transformer tout utilisation d’une ligne de code en ligne de texte basique. Nous allons donc utiliser le htmlspecialchars() par exemple pour contourner le risque d’utilisation de scripts.



**Veille effectuée sur la sécurité des sites web**

**Formulation 2**

* **Injection SQL**

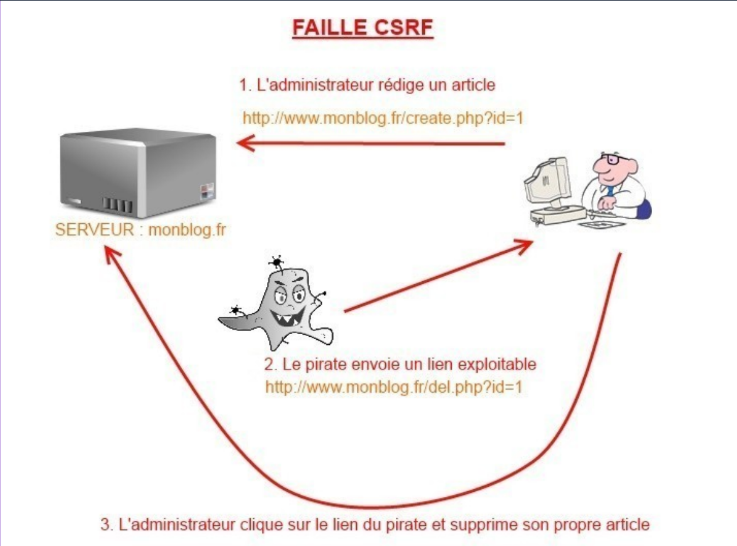
Elle consiste à injecter des requêtes par le biais des champs de formulaire et traitements mal protégés, afin de récupérer des données, de se connecter sans avoir les identifiants, de supprimer des données, de récupérer tout type d’informations…

Pour contrer **L’injection SQL** , l’utilisation de framework (Symfony) ou alors utiliser la classe PDO en PHP pour faire des requêtes préparées permettent d’éviter cela.

* **Falsification de requêtes inter-sites (CSRF)**

Il s’agit d’effectuer une action visant un site ou une page précise en utilisant l’utilisateur comme déclencheur, sans qu’il en ait conscience.

L’attaquant crée alors son propre formulaire en champ caché. Quand un utilisateur clique sur le bouton de validation alors le formulaire s’exécute.



**Comment contrer les requêtes CSRF :** grâce à un jeton (aussi appelé token en anglais) qui est un nombre ou une chaîne de caractère aléatoire qui va être testée avant toute modification ou édition d’un article.

**Jeu d’essai**

**Présentation de la navigation du site**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **En tant que…** | **Je veux pouvoir…** | **Afin de…** |
| **Visiteur 1** | Simple visiteur | Naviguer sur le site | Lire des articles du blog |
| **Visiteur 2** | Simple visiteur | Naviguer sur le site | Naviguer sur le site |
| **Administrateur 3** | Administrateur | Accéder à la partie administrateur | Pouvoir administrer, modifier les articles en fonction |
| **//////////////** | /////////////////// | A VENIR // AXES D’AMELIORATIONS | ///////////////////////////////////// |
| **Visiteur 4** | Client | Me connecter | Avoir accès à une messagerie en ligne |
| **Visiteur 5** | Client | Me connecter | Avoir accès à un calendrier de prise de rendez-vous |

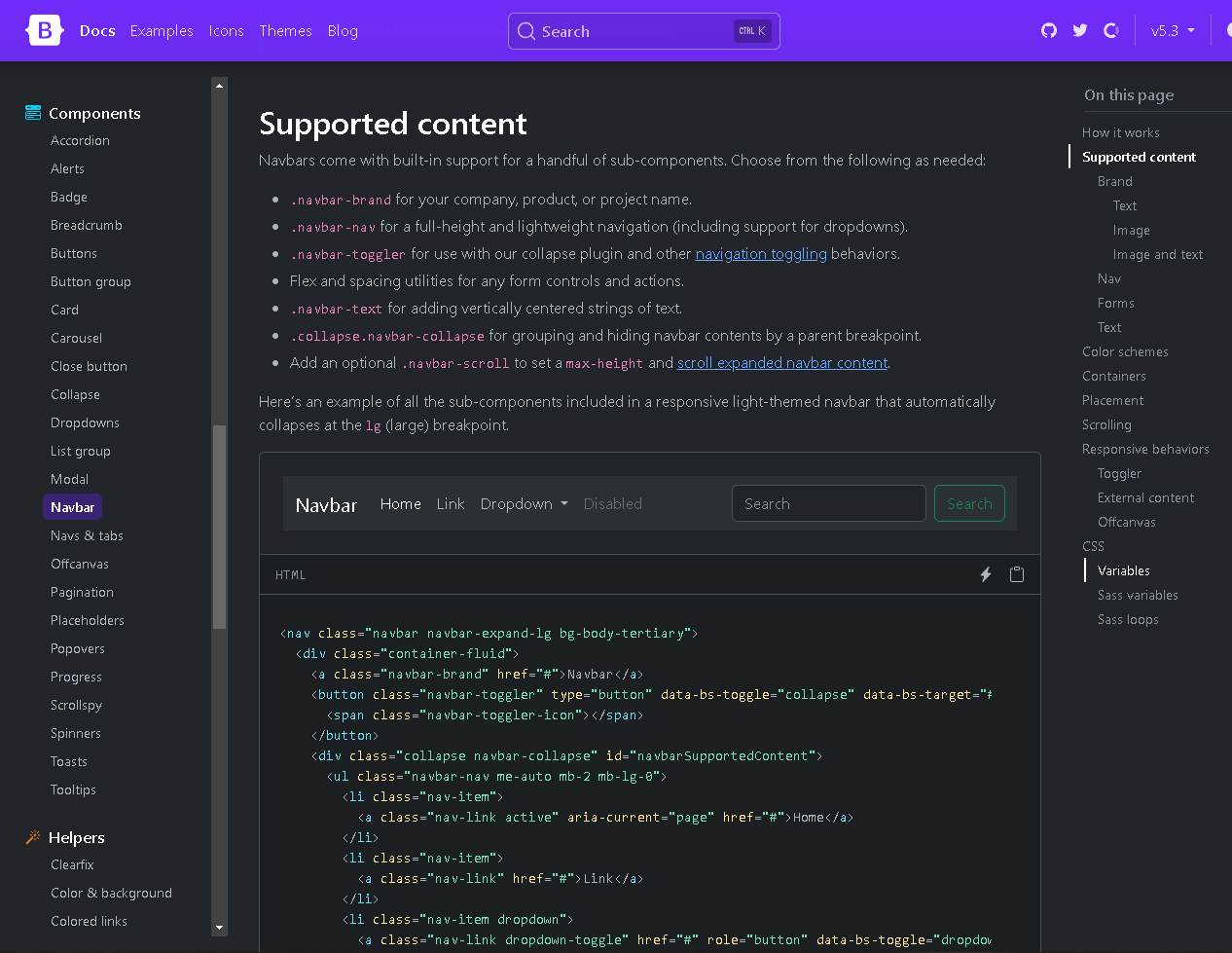
Il faut prendre en compte que le visiteur aura accès à l’ensemble du site dit « vitrine », et aura la possibilité de lire les articles édités par l’administrateur.

Je souligne également la partie « Me connecter » en précisant que c’est un axe d’amélioration qui sera exploité pour le confort du client, à l’issue de la mise en place d’un front-office.

Le client pourra alors recevoir des messages de la part de l’administration, et pourra, si cela est nécessaire pour lui, convenir d’un rendez-vous pour la prestation attendue.

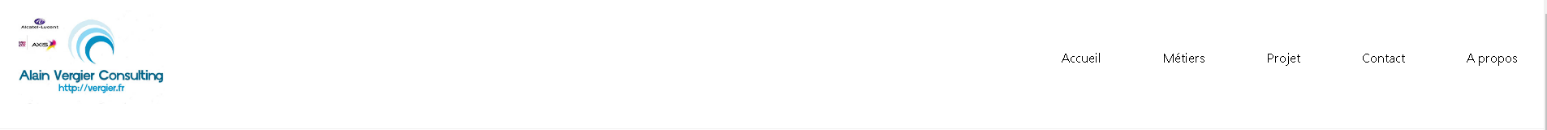
**Extrait de sites anglophones**

**Bootstrap**



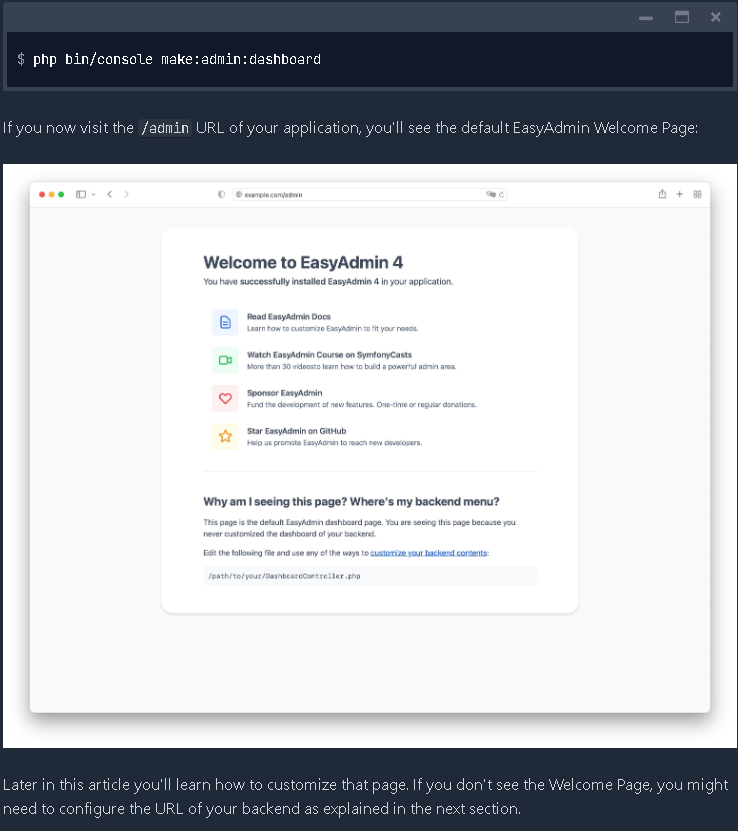
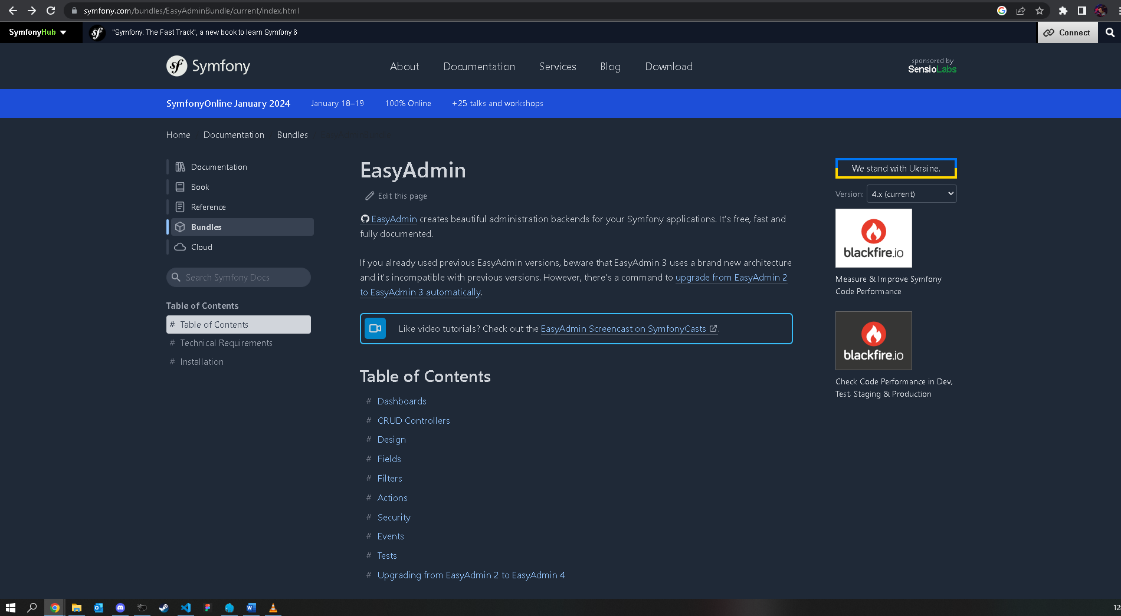
Pour mon extrait de site anglophone, je vais prendre l’exemple de Bootstrap, utilisé dans le cadre de la recherche décrite précédemment.

J’ai utilisé Bootstrap, étant un Framework CSS, de façon à mettre en place la NavBar qui est à l’heure d’aujourd’hui présente sur mon projet. La NavBar a été modifiée à la convenance du client, mais aussi de façon à reprendre le visuel de la maquette Figma.



**Extrait de sites anglophones**

**Le bundle EasyAdmin**



Le bundle EasyAdmin de Symfony m’a permis de créer la partie back-end administrable pour mes articles.