



INFLUACTIVE
l'influence positive

Société Influactive
5 Quai Hippolyte Jaÿr
69009 Lyon - Vaise



IUT Lyon 1

l'excellence technologique

Université Claude Bernard - Lyon I
Institut Universitaire de Technologie
43 Boulevard du 11 Novembre 1918
69622 VILLEURBANNE Cedex

Rapport de stage

2^{ème} année

DUT informatique

Réalisé par **Julien Giraud**

Intitulés du stage : **Développeur web / Intégrateur Typo3**

Période du stage : **du 7 avril au 21 juin 2019 (11 semaines)**

Maître de stage : **Hervé Bigal**

Tutrice enseignante : **Mme Olivesi**

Fiche technique

La société

Influactive est une société du digitale composée d'une dizaine de personnes. Elle s'occupe de la création, du contenu, du référencement et de la maintenance de sites web personnalisés pour les professionnels.

Mes missions

Intitulés	Développeur web / Intégrateur Typo3.
Besoins	Les clients de la société ont besoin de faire réaliser ou de mettre à jour leur site web.
Objectifs	Satisfaction du client : effectuer le travail permettant au client d'avoir le résultat (site) souhaité. Maintenance : une fois réalisé, continuer à s'occuper du site.
Contraintes	Le visuel du site doit correspondre à la maquette validée. S'il y en a, les parties fonctionnelles doivent fonctionner. Le site doit être adapté à tout type d'affichage (téléphone, tablette ou grand écran). Le site doit être facilement modifiable par le client non informaticien (dans une mesure raisonnable). Temps : il y a normalement des contraintes de temps mais en pratique il y a juste des éléments à traiter en priorité.

Environnement de travail

Lieu	Un bureau pour six où sont les deux autres développeurs (et informaticiens) d' Influactive ainsi que le stagiaire commercial. Tout le monde a un poste fixe avec deux écrans.
Organisation	Les projets ou tâches sont répartis entre les développeurs, nous travaillons seuls dessus mais l'entraide est de rigueur si besoin.
Technologies	Typo3, TypoScript, WordPress, HTML, CSS ¹ , JavaScript ¹ , jQuery ² , Bootstrap ¹ , Bulma ² , PHP, SQL, shell (Linux).

Rapport de stage

Temps laissé pour la rédaction : mon temps libre

¹ Vu ou découvert à l'IUT mais beaucoup approfondi lors du stage

² Appris sur place

Glossaire

Les définitions suivantes sont celles que nous utiliserons pour ce rapport. Leur sens peut cependant varier selon les lieux, les entreprises ou les domaines d'application.

CMS : Content Management System, un outil qui permet de gérer un site web ainsi que son contenu depuis une interface en ligne.

Maquette : montage PhotoShop du résultat visuel attendu pour un site, indépendamment de son contenu. Une maquette contient généralement la home du site, le header, le footer et une page interne.

Template : thème d'un site dans une structure de fichiers organisée. Elle contient notamment les polices d'écriture, parfois des images, et surtout le code qui permet de structurer les pages, de les mettre en forme, de les dynamiser et de les animer.

Intégration : phase de création d'un template à partir de la maquette.

Front-end : ce que va voir l'utilisateur final. Autrement dit, c'est ce qu'il voit lorsqu'il consulte le site web.

Back-end : partie que le client voit lorsqu'il se connecte au CMS pour gérer son site. C'est donc l'interface qui lui permet d'en modifier le contenu. Cette partie est invisible pour l'utilisateur final.

Home : nom que l'on donne à la page d'accueil d'un site.

Header : partie haute d'un site, répétée sur toute ses pages. On y trouve généralement le menu.

Footer : partie basse d'un site, répétée sur toute ses pages.

Natif : terme qui qualifie un outil informatique (comme un CMS). Cela veut dire que l'outil permet de gérer « ce dont on parle » sans besoin d'installer quoi que ce soit (module, extension, plugin, framework...).

Librairie : un ensemble de fonctionnalités faites dans un langage donné, avec une documentation. Les utiliser permet de gagner du temps et de ne pas tenir compte des subtilités des différentes versions du langage d'origine.

Remerciements

Quelques jours avant ma soutenance au moment où je termine de rédiger ce rapport, je tiens à remercier toutes les personnes qui ont contribué au bon déroulement de mon stage.

Je remercie d'abord mon maître de stage, Hervé, de m'avoir choisi et fait confiance pour ce stage.

Je remercie Frédérique qui en plus d'avoir de très bon goûts en matière de café, m'a suivi depuis le début, toujours dans la joie.

Je remercie François et Sébastien de faire partie des rares personnes à apprécier l'humour de développeur, de m'avoir accordé leur temps et de m'avoir fait découvrir toutes ces technologies.

Enfin je remercie Christine, Marion, Angèle, Camille et Axel, ainsi que tous les membres d'Influactive que j'ai déjà cité ; merci pour votre accueil et pour votre bonne humeur.

Sommaire

Introduction	7
Présentation de la société	8
Environnement humain	9
Environnement de travail	10
Environnement technologique	12
Mes projets	15
Brochot	15
La mise en place du projet	15
Des début compliqués	17
La prise en main de Typo3	20
Le système multi-langues	23
Advineo, le fameux projet WordPress	25
Une mission qui semble plutôt simple	25
Le désenchantement de WordPress	25
La modification des éléments de page	26
Des solutions viables	27
Eye Tech Care	30
La récupération du projet	30
La première phase de l'intégration	31
La deuxième phase de l'intégration	34
Point technique : la gestion du slider	36

Rapport de stage

Le choix du point technique	36
La problématique du défilement automatique	36
L'exclusion mutuelle, une solution efficace	37
Bilans du stage	39
Bilan technique	39
Bilan professionnel	40
Bilan personnel	41
Conclusion	42
Sommaire des annexes	44

Introduction

Pour valider le DUT nous devons effectuer un stage en fin de deuxième année. J'ai donc commencé ma recherche de stage un peu avant le milieu du 3^{ème} semestre. J'étais intéressé par plein de domaines, dont le développement logiciel, et plus récemment le développement PHP. Je venais en effet de finir le projet web du semestre, en PHP, et j'ai adoré travailler dessus.

Peu de temps après, nous avons assisté à la conférence sur les projets web présentée par Hervé Bigal. J'ai trouvé son discours très intéressant et les réalisations qu'il nous a montré m'ont beaucoup plu. Quand il nous a annoncé qu'il recherchait des stagiaires en PHP j'ai voulu sauter sur l'occasion. Je suis donc allé le voir après la conférence pour demander de quelle façon candidater. Nous étions une vingtaine à avoir pris ses coordonnées, mais c'est finalement moi qui a été pris.

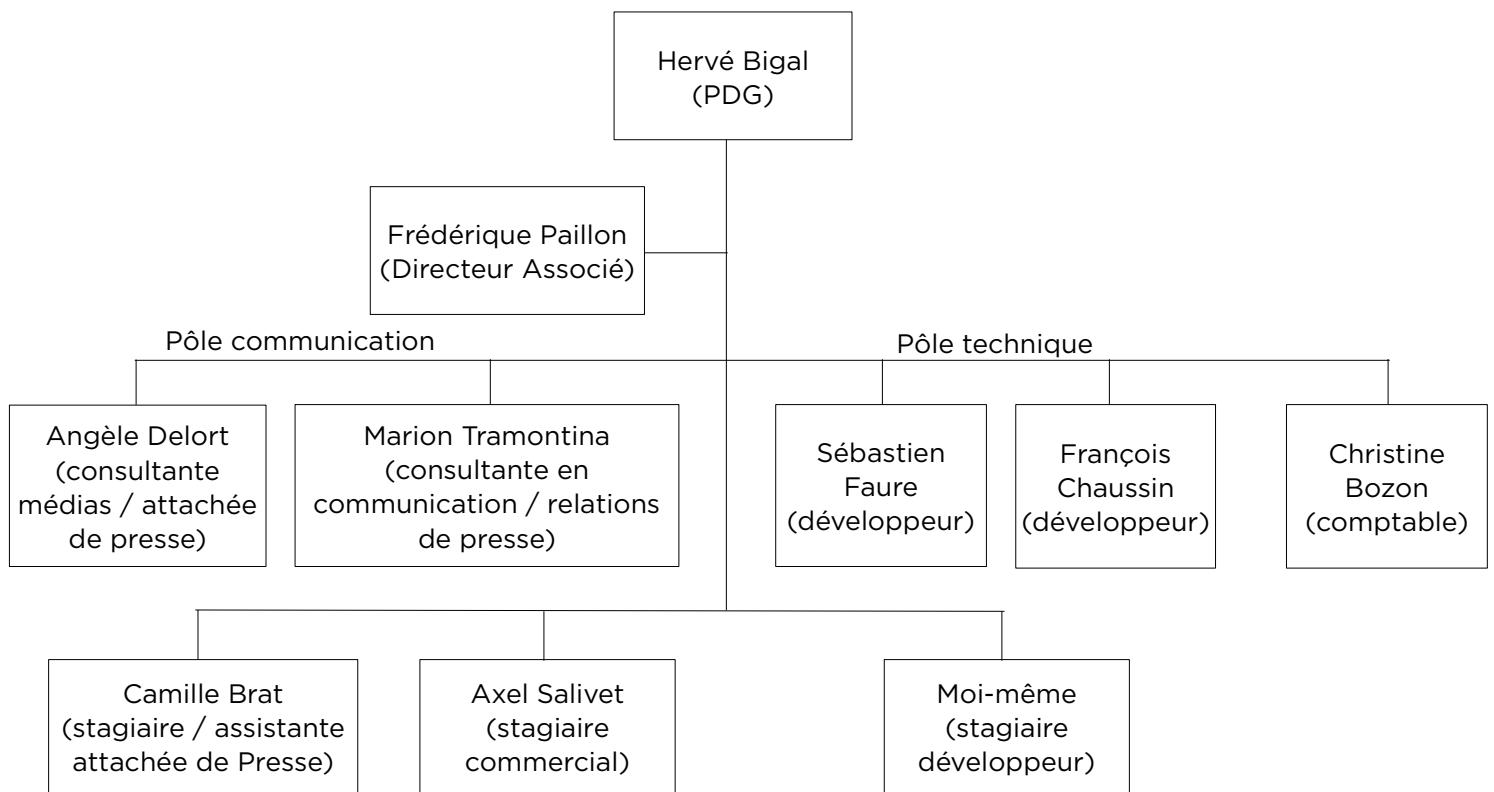
Avant le stage j'ai eu quelques échanges avec les développeurs, ils m'ont parlé des technologies qu'ils utilisent et m'ont demandé si je pouvais me former sur les bases du langage JavaScript. Par chance je m'en étais déjà servi, et j'ai pu approfondir mes connaissances au 4^{ème} semestre car c'était l'objet de l'un des modules de mon parcours.

Dans la suite de ce rapport je vais présenter la société Influactive ainsi que mon environnement de travail, puis montrerai le travail que j'ai réalisé au travers des différents projets que l'on m'a confié. Je continuerai sur l'étude du problème le plus technique que j'ai résolu, avant de finir par un bilan sur les apports de ce stage et la pertinence des enseignements de l'IUT.

Présentation de la société

Influactive est une société située au 5 quai Hippolyte Jaÿr qui compte une dizaine d'employés. Les avantages d'une structure de cette taille sont qu'il faut être polyvalent et que la hiérarchie est horizontale. Très pratique d'accès pour moi qui habite à Saint-Genis-les-Ollières, je prends juste le bus pour me rendre au métro puis j'ai deux stations à faire et 5 minutes de marche. Pour aller à l'IUT c'est presque deux fois plus long.

Organigramme



Environnement humain

Influactive cohabite avec la société IDSA qui a un bureau et peut utiliser la salle de réunion. Les locaux comportent un hall d'entrée où travaille Christine, un bureau pour six personnes où travaillent François et Sébastien ainsi qu'Axel, stagiaire, et moi-même. Il y a un bureau de la même taille où travaillent Angèle et Marion avec Camille, stagiaire aussi. Hervé, le PDG d'Influactive a son propre bureau, de même pour Frédérique, son associé. Enfin il y a la salle de réunion partagée avec IDSA.

Christine s'occupe de l'accueil des deux sociétés, tant physique que téléphonique ; et de tout ce qui est factures, relances ou bulletins de paie. Globalement c'est elle qui assure la gestion administrative.

Angèle, Marion et Camille s'occupent du pôle communication. Le contenu des sites, c'est elles.

Frédérique est le profil créatif de la société. C'est lui le graphiste, à l'origine des maquettes des sites. Globalement il est mon référent pour tout ce qui est extérieur au développement.

François et Sébastien sont les développeurs et informaticiens de la société, le pôle technique. Ce sont mes référents pour ce qui est développement. Ils assurent la maintenance des sites, ainsi que tout ce qui touche aux hébergements, aux certificats, aux problèmes de serveurs, aux adresses mails de la société ou encore aux problèmes informatiques des employés quand c'est de leur ressort.

Enfin il y a Hervé qui joue un rôle très important en plus d'être le dirigeant. C'est lui qui veille à l'avancé des différents projets en cours, il organise les réunions (et s'occupe de leur compte rendu) pour coordonner le pôle communication avec le pôle technique.

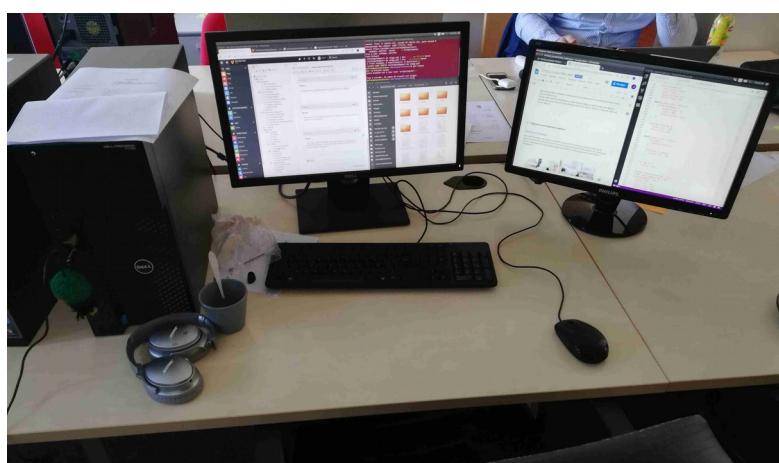
C'est aussi le principal commercial. Il assure comme Frédérique le lien entre l'intérieur et l'extérieur de la société. Il s'occupe de démarcher les nouveaux clients, de faire des propositions et de gérer leurs dossiers. En ce moment il est aidé par Axel, stagiaire commercial.

Hervé s'occupe en plus du recrutement.

Environnement de travail

Pour commencer nous avons à notre disposition deux machines à café (Senseo, Nespresso), un frigo, un micro-ondes et un peu de vaisselle (tasses, couverts, assiettes, verres) si besoin. La pause de midi est normalement de 12h30 à 13h30. On peut manger sur place ou dehors, il y a plein de petits restaurants à proximité. Généralement je mange dehors une fois par semaine et j'amène mon repas le reste du temps. J'ai parfois une pensée pour les machines de Daltys¹, je mets alors à profit mes tickets restaurants dans la boulangerie en bas de la rue.

Pour ce qui est du matériel je dispose d'un ordinateur fixe avec deux écrans que je suis autorisé à gérer comme je veux. Lorsque je suis arrivé, le poste n'avait pas servi depuis très longtemps. Il était encore sous Windows 7 et était très lent alors qu'il possède de très bonnes caractéristiques. J'avais fait une grosse maintenance du poste (suppression de 300 Go de fichiers temporaires², désinstallation d'applications, suppression des programmes qui se lancent au démarrage...). L'ordinateur fonctionnait beaucoup mieux que le premier jour mais il était toujours très lent. Après quelques jours d'utilisation j'ai finalement installé Linux sur le poste (Ubuntu 18). J'ai enfin pu profiter d'un poste très performant.



Mon poste de travail

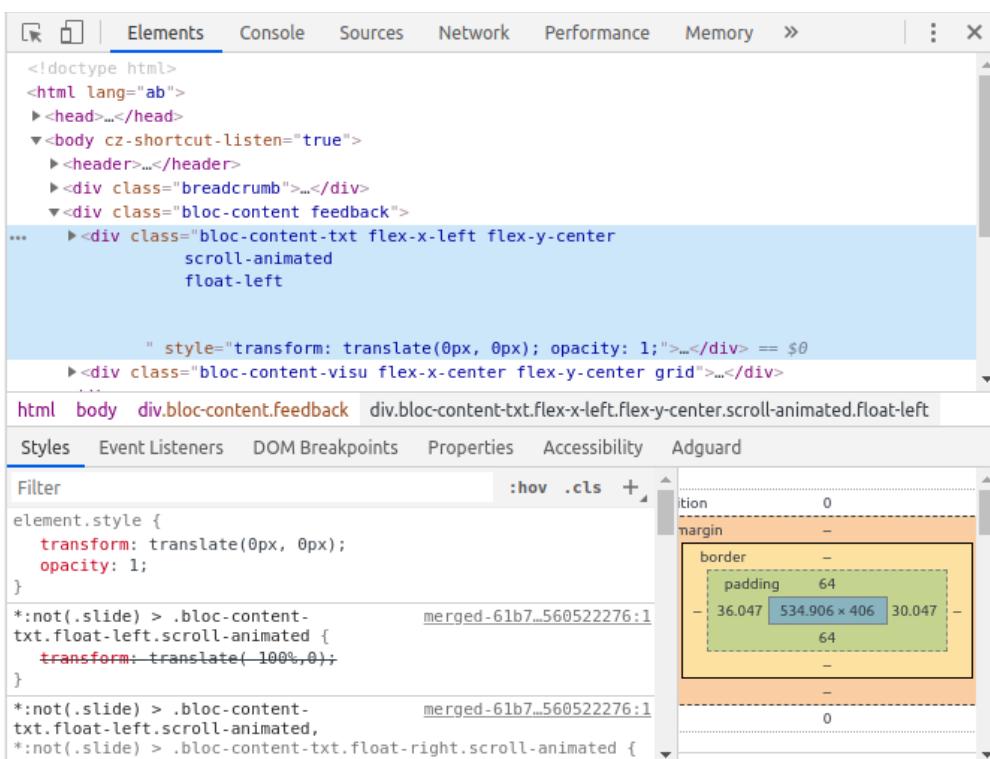
¹ La société qui possède les distributeurs et les machines à café à l'université

² Des fichiers de cache qui s'accumulent avec le temps mais ne sont plus utiles

Rapport de stage

Mes outils/logiciels de travail sont VS Code³, Google Chrome, son interface de développement, le navigateur de fichier et le terminal⁴ Linux.

L'avantage de Linux, en plus de son interface que j'apprécie, est que le gestionnaire de fichiers intègre nativement les connexions en (S)FTP⁵. Cela permet de naviguer dans les fichiers des sites web⁶ depuis le gestionnaire de fichiers comme s'ils étaient sur l'ordinateur. Sur Windows il faut installer un logiciel pour se connecter en FTP et c'est beaucoup moins pratique. Il est aussi très facile d'écrire un script⁷ qui lance des applications de façon personnalisée. J'en ai donc écrit un qui au démarrage me lance Google Chrome, ouvre chacun des répertoires que j'utilise pour travailler et ouvre avec VS Code tous les fichiers dans lesquels je développe.



Interface de développement de Chrome

3 Un éditeur de code gratuit très utilisé par les développeurs

4 Console dans laquelle on peut entrer des instructions en ligne de commande

5 Technologie de connexion à un serveur de fichiers distant, SFTP pour FTP sécurisé

6 Accessibles uniquement en FTP ou SFTP

7 Programme exécutable

Environnement technologique

Vous trouverez ci-dessous les technologies que j'ai principalement utilisées.



TYPO3 : Il s'agit d'un CMS libre extrêmement performant en terme de sécurité et de vitesse de chargement. C'est le CMS de prédilection de la société. Il permet de gérer nativement le multi-langues, les utilisateurs front-end/backend (les comptes abonnés/administrateurs dont les droits sont très différents). J'ai eu beaucoup de mal à m'en servir au début, il y a toute une arborescence de fichiers à comprendre, et des fichiers font le lien entre les fichiers que l'on développe et le backend. Ce sont mes collègues qui m'ont formé. Ensuite j'ai de plus en plus utilisé la documentation mais elle manque de contenu et il faut de bonnes bases sur le CMS pour comprendre.



TypoScript : Il s'agit du langage utilisé par Typo3. Il fait le lien entre le back-end du CMS, le back-end développeur (le template) et le front-end. Ce langage peut s'écrire sous forme XML (un langage de balisage) ou sous une forme qui ressemble au JSON (un format d'écriture de données). On peut en mettre de partout, dans le backend, dans les fichiers de configuration, dans les fichiers HTML, les fichiers PHP... Toute la difficulté est de trouver quelle forme utiliser et où. Il n'y a pas qu'une solution.



WordPress : WP est le CMS le plus utilisé au monde, à l'origine destiné aux blogs. Très pratique lorsqu'on veut réaliser un site vitrine rapidement avec un petit budget. Cependant c'est une contrainte car il faut le mettre à jour TRÈS souvent, et plus le site est complexe plus il est lent et compliqué à modifier.



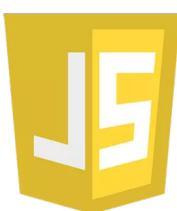
HTML / CSS : Ce sont les langages web fondamentaux. Le HTML permet de structurer le contenu d'une page avec des balises. Il y a les balises ouvrantes et les

balises fermantes. Une balise est composée d'une ou plusieurs lettres entourées de chevrons.

On ajoute un antislash après le chevron ouvrant pour les balises fermantes.

La balise ouvrante `p` donne `<p>` et sa balise fermante est `</p>`.

Le CSS permet de modifier l'affichage ou le comportement du contenu des balises HTML. J'avais déjà vu ces langages à l'IUT, mais le stage m'a vraiment fait découvrir les possibilités du CSS, dont les animations. J'ai écrit environ 4000 lignes de CSS au total.



JavaScript : JS pour les intimes, c'est le langage qui donne vie aux pages web. Je l'avais déjà vu tout seul avant de l'approfondir à l'IUT. Il permet de modifier le contenu d'une page, que ce soit sa structuration ou les attributs⁸ que possèdent ses éléments. Le plus compliqué c'est de savoir précisément ce que l'on veut faire, ensuite il faut trouver l'algorithmie correspondante puis suivre la documentation.



jQuery : Une librairie JS qui est très souvent utilisée. Son principal avantage est de prendre beaucoup moins de place que le vanilla JS⁹. Cela permet de rendre le code plus lisible. J'ai utilisé jQuery quand l'équivalent JS faisait perdre trop de temps à développer. Je me suis servi de la documentation.



Bootstrap : Une librairie HTML / CSS / JS très répandue. J'ai découvert Bootstrap à l'IUT mais j'ai complètement appris à l'utiliser lors du stage, grâce à la documentation. Les thèmes d'origine de Brochot et Eye Tech Care utilisaient Bootstrap, je m'en suis donc servi pour l'intégration.

⁸ Propriétés

⁹ JavaScript de base sans aucun ajout



PHP : Un langage de programmation web, exécuté côté serveur. J'ai appris ce langage à l'IUT, c'est même l'objet de mon stage à l'origine (« stage en PHP »). En soit, Typo3 et WordPress sont des CMS basés sur PHP mais il n'est pas nécessaire de savoir développer pour les utiliser. J'ai utilisé ce langage sur les deux CMS mais surtout sur WordPress.



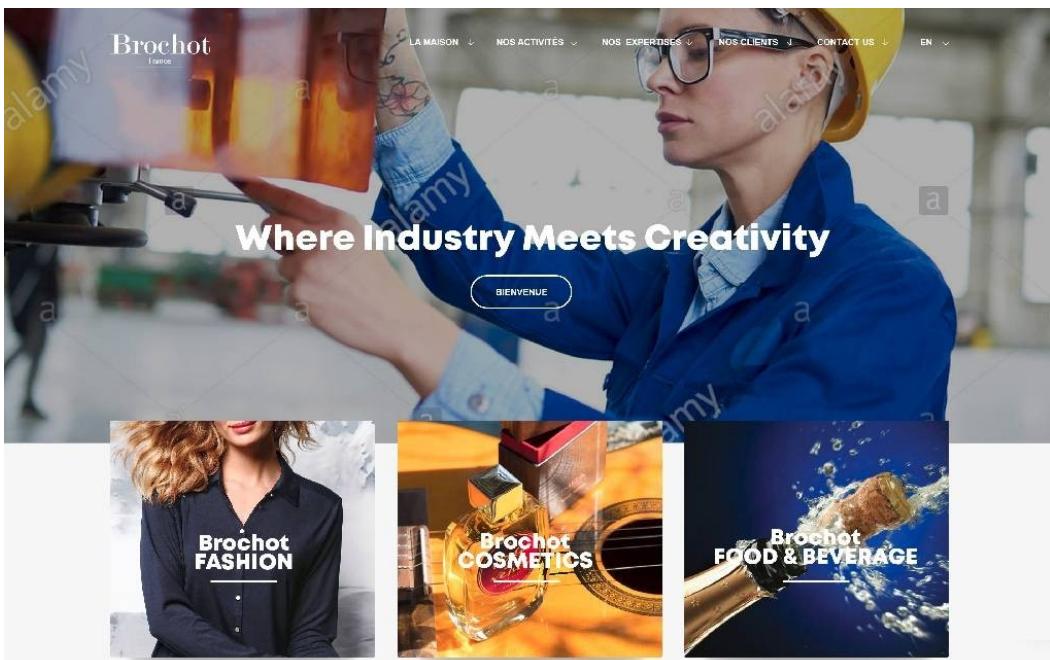
Shell (terminal Linux) : Le terminal Linux permet de faire plein de choses qu'on fait normalement avec la souris en plusieurs étapes. Je l'ai utilisé pour tout ce qui est installation de logiciel, connexion aux serveurs web pour installer Typo3 ou le mettre à jour, lancer des applications et rechercher du texte dans tous les fichiers d'un dossier et de ses sous-dossiers.

Mes projets

Brochot

Le site de Brochot est un site vitrine, ce sont des fabricants de boutons. Leur site actuelle a environ 10 ans. Afin de moderniser leur image, ils en souhaitent un nouveau qui fasse effet startup. C'est un projet Typo3 avec un budget de 19 000 €.

C'était mon premier site sur Typo3, je devais le gérer de l'installation jusqu'à sa publication.



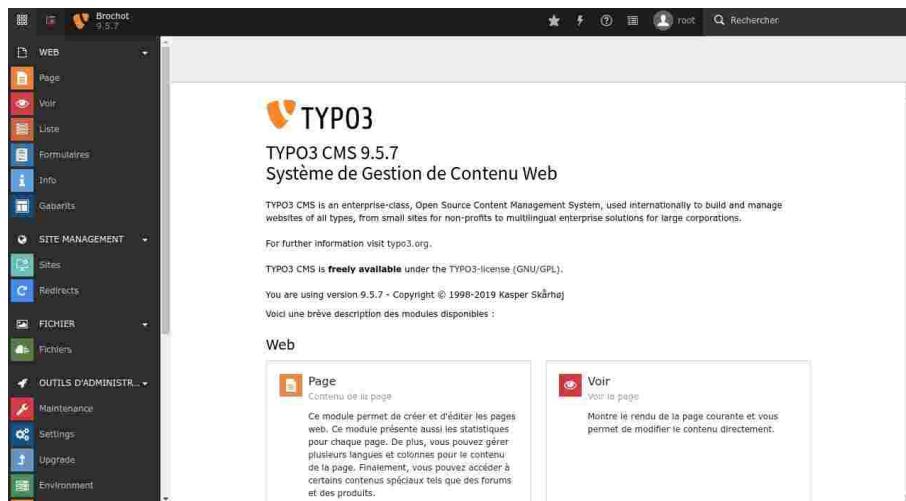
Maquette de brochot.com, haut de la page d'accueil

La mise en place du projet

Pour commencer j'avais une copie de la maquette du site. Mes collègues m'ont donné le lien d'un site générateur de template par défaut pour Typo3. J'ai rentré quelques informations puis j'ai téléchargé le template. Ensuite j'ai récupéré les accès du serveur web pour accéder aux fichiers mais aussi pour me connecter avec un terminal (les accès sont différents car le terminal peut exécuter des commandes en plus d'accéder aux fichiers, il lui faut donc plus de permissions). Ensuite on m'a redirigé vers la

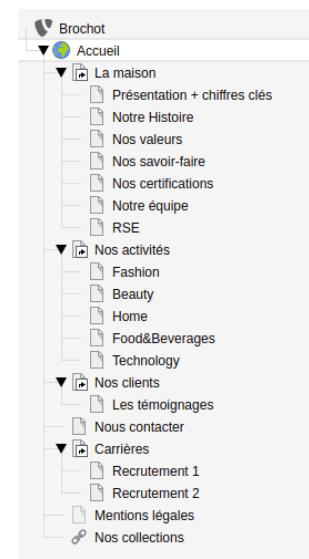
Rapport de stage

documentation de l'installation de Typo3 qui se fait depuis le terminal. Pas de difficulté pour cette partie, juste quelques commandes à copier dans la console. Juste après je suis entré dans l'installateur Typo3 en allant sur le site. J'ai entré quelques informations et j'ai envoyé le template puis je me suis retrouvé dans le back-end de Typo3.



Back-end de Typo3

Il y a eu une première phase de prise en main du CMS. J'ai fait le tour des options et j'ai essayé de comprendre le lien entre les fichiers du template et le site mais il était trop tôt pour ça. J'ai ajouté un nouveau site dans le back-end (Brochot, évidemment). Dans les options je lui ai assigné le template que j'ai généré. J'ai aussi ajouté une seconde langue au site car il sera minimum en français et anglais. L'ajout de la langue a été très compliqué ! Il fallait faire du ping pong entre les différents onglets de Typo et j'étais perdu. Ce sont mes collègues qui m'ont aidé. Ensuite j'ai ajouté l'arborescence des pages du site, ça c'était facile. Je pouvais passer à l'intégration.



Arborescence du site

Des début compliqués

Une fois le site installé, il faut s'occuper de son intégration. Cette phase se fait en deux étapes. D'abord il faut faire l'intégration HTML/CSS/JS des pages de la maquette. Puis ensuite on intègre les éléments/sections que l'on a créée (le code HTML/CSS/JS) au back-end de Typo3. C'est la première phase que je traite dans cette partie.

J'ai commencé par faire la partie HTML de la page puis je voulais m'occuper du CSS/JS après. J'ai mis un certain temps à terminer cette première partie car J'ai eu un peu de mal à récupérer le contenu du site (photo/texte) depuis la maquette. Je n'étais vraiment pas habitué à utiliser Photoshop. Ensuite je me perdais dans les fichiers du projet (voir annexe 2).

Une fois cette partie faite j'ai voulu m'occuper du CSS. J'avais très peu de souvenirs des cours de CSS, j'en avais refait pour les cours de JavaScript mais j'étais très loin de maîtriser l'étendu de ce qu'on peut faire avec ce langage. La première difficulté est que pour ajouter du CSS sur Typo3 il faut utiliser le fichier de configuration en TypoScript. En soit ce n'est pas très difficile mais c'est encore une nouvelle chose à comprendre. On utilise le même système pour ajouter du JS au site.

```
146     includeCSS {
147         bi_template_layout = EXT:bi_template/Resources/Public/Css/layout.min.css
148         montserrat = EXT:bi_template/Resources/Public/Css/montserrat.css
149         open_sans = EXT:bi_template/Resources/Public/Css/open_sans.css
150         styles = EXT:bi_template/Resources/Public/Css/styles.css
151         theme-energy = EXT:bi_template/Resources/Public/Css/theme-energy.css
152         # Attention à bien garder cet ordre
153         monCss = EXT:bi_template/Resources/Public/Css/main.css
154     }
155
156
157     includeJS {
158         monJs = EXT:bi_template/Resources/Public/JavaScript/Src/main.js
159         jquery = EXT:bi_template/Resources/Public/JavaScript/Src/jquery-1.12.3.min.js
160     }
161
162     includeJSFooter {
163         bi_template_scripts = EXT:bi_template/Resources/Public/JavaScript/Dist/scripts.js
164         # Scripts du thème
165         plugins = EXT:bi_template/Resources/Public/JavaScript/Src/plugins.js
166         core = EXT:bi_template/Resources/Public/JavaScript/Src/core.js
167     }
168 }
```

Ajout des fichiers CSS et JS en TypoScript, fichier de configuration du template

Rapport de stage

Bon, le temps passe et je me sens un peu bloqué sur l'affichage du menu. François m'a conseillé d'utiliser Bulma, une librairie HTML/CSS très efficace pour placer les éléments sur la page, et beaucoup plus légère que Bootstrap. Je me suis donc lancé, j'ai passé un petit moment à regarder la documentation puis j'ai réussi à faire tout ce que je voulais pour le menu. Il y avait même un système automatique pour le menu en mode téléphone. Je continue à utiliser Bulma pour la suite, le temps passe (trois jours après le début du stage) et j'apprends que la maquette du site était basée sur un thème qui venait d'être acheté ! J'ai alors pu récupérer les sources HTML/CSS/JS des pages avec tous les éléments de la maquette. Je suis donc revenu sur tout mon travail pour me servir du thème. Finalement il utilisait Bootstrap donc j'ai laissé Bulma de côté.

La suite de l'intégration a duré environ deux semaines. J'ai passé une partie du temps à comprendre le système de structuration des éléments des pages du thème, à comprendre comment utiliser les animations et autres effets du thème. Ensuite il y a le temps d'intégrer les éléments du thème au contenu du site de Brochot, pour obtenir les bonnes dispositions et les bonnes animations. Puis il y a eu l'ajout du CSS pour gérer les cas de redondance d'utilisation de Bootstrap et modifier tout ce qui n'était pas encore comme sur la maquette. Cela comprend tout ce qui est couleurs, polices d'écriture et ajustements sur le positionnement des éléments. Enfin il y a eu ma partie préférée : rendre le tout responsive (adapté pour mobile et tablettes). Je m'étais déjà bien remis à niveau en CSS et sur Bootstrap mais c'est vraiment sur cette partie que j'ai le plus appris. La démarche était simple : pour chaque élément du site je partais d'une largeur de 1500px et je réduisais jusqu'à voir un problème. Dès que c'était le cas, j'essayais de voir quel réglage CSS pouvait le corriger. Puis si je pouvais le gérer avec Bootstrap je le faisais, sinon j'utilisais mon fichier CSS. Il faut savoir qu'il y a quelques normes sur les tailles d'écran. À partir de 1200px de largeur on parle de très grand écran, de 992px à

Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
-----------------------	-----------------	------------------	-----------------	------------------------

Rapport de stage

1199px on est sur un grand écran, de 768 à 991px c'est un écran moyen (on commence à être sur un format tablette), de 576 à 767px c'est un petit écran et en-dessous c'est le format téléphone. On considère que la largeur minimum est entre 300 et 320px pour les vieux téléphones. Face à un problème de responsive il faut donc voir dans quelle tranche d'écran on se situe, et trouver un réglage qui fonctionne pour toute les largeurs de la tranche. Bootstrap permet de régler beaucoup de choses sur ces tailles mais parfois ce n'est pas suffisant. J'ai donc appris à gérer moi-même tous les cas de figure avec le système de CSS qui varie selon la taille de l'écran.

The image shows two side-by-side screenshots of the Brochot website. The left screenshot is a mobile version, showing a large landscape image at the top, followed by the brand name 'Brochot' and 'le détail essentiel'. Below this is a short description and a small circular icon with the number '88'. The right screenshot is a desktop version, showing a similar layout but with a more complex navigation bar at the top with multiple dropdown menus like 'LA MAISON', 'NOS ACTIVITÉS', 'NOS CLIENTS', 'NOUS CONTACTER', 'CARRIÈRES', 'FRA', and 'NOS COLLECTIONS'. It also features a larger description and a larger circular icon with '88'.

Comparatif version mobile / version PC

J'ai rencontré un nouveau problème dans cette partie de l'intégration. Le menu du site devait apparaître en haut de l'écran lorsqu'on se déplaçait sur la page mais il fallait descendre très bas pour qu'il apparaisse. Il s'agissait d'un bug du thème dans un fichier JavaScript. Il a été difficile de trouver où était la partie de JS responsable de cette fonctionnalité. Je me suis servi de l'interface de développement de Firefox pour la trouver. Ensuite, j'ai dû refaire le système de déclenchement du menu, c'était complexe car il s'agit de programmation asynchrone (plusieurs choses qui s'exécutent en même temps). Le soucis est qu'il y a une animation lorsque le menu apparaît et

Rapport de stage

pour ça il faut d'abord dire au menu qu'il est actif mais invisible puis après un petit délai on peut lui dire qu'il est visible. Vu comme ça on peut se dire que c'est vite réglé mais il faut aussi prendre en compte les cas où l'utilisateur descend puis remonte rapidement en haut de la page : les instructions de cacher et d'afficher le menu interfèrent l'une sur l'autre. La solution que j'ai utilisé prend tout ça en compte, il y a une quarantaine de lignes avec tout un tas de vérifications pour gérer les problèmes asynchrones.

Au final la première partie de l'intégration m'a pris une dizaine de jours de travail. J'ai rajouté un peu plus de 400 lignes de CSS et j'ai fait environ 600 lignes de HTML avec du Bootstrap de partout.

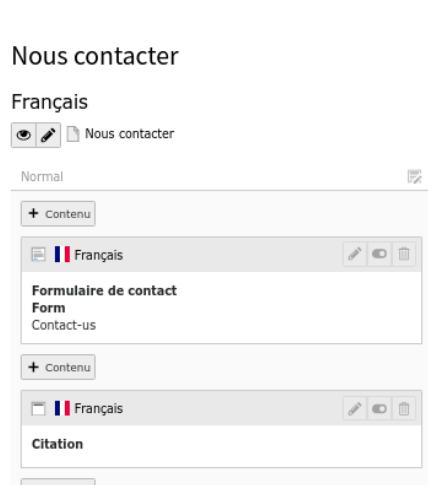
La prise en main de Typo3

Il s'agit là de la deuxième phase de l'intégration, l'objectif est le suivant :

- dans le back-end de chaque page on peut ajouter différents blocs qui possèdent un ou plusieurs champs,
- chaque bloc correspond à un élément du thème qui sera affiché sur la page, le contenu de ces éléments est saisi dans les champs,
- c'est de cette façon que le client peut modifier le contenu de son site.

Nous contacter

French



Back-end d'une page avec deux blocs

Modifier Contenu de la page "Citation" sur la page "Nous contacter"

Général

Type : Citation

Langue : Français

Colonnes : Normal

Visible

Contents:

Nom du bloc : Citation...

Texte : "Is loco et oppido Methymnaeus, terra atque insula omni Lesbius fuit. Eum Arionem rex Corinthi Periander amatumque habuit artis gratia."

Auteur : Arion

Contenu présent dans les champs du bloc « Citation »

Rapport de stage

Pour que tout ça soit possible, il faut créer les différents blocs qui composent le site, il faut leur ajouter les champs correspondant à l'élément et il faut expliquer à Typo3 ce qu'il doit faire avec le contenu de ces champs (comment il doit les insérer dans le code).

Pour ça nous utilisons l'extension DCE (Dynamic Content Element). Cette extension permet de créer des blocs personnalisés. Lorsqu'on ajoute ou modifie un DCE (depuis l'extension), il y a un onglet où on peut ajouter de nouveaux champs avec des paramètres TypoScript au format XML. Puis dans un autre onglet on peut écrire le code HTML qui intègre les différents champs. Il est possible d'y insérer les différents champs du bloc, et on peut y utiliser une autre forme de TypoScript avec des balises spéciales. Elles sont de la forme <f:nomDeLaFonctionnalité> contenu qui doit être traité par cette fonctionnalité </f:nomDeLaFonctionnalité>. Ces balises spéciales permettent de faire les mêmes traitements basiques que les langages de programmation, et des traitements spécifiques au web ou à Typo3.

DCEs	Status	Identifier	UID	Usage (CEs)	DCE infos	Field infos
Lien(s)	ENABLED	dce_liens	14	5	7 fields	<button>Edit</button>
Section Posts	ENABLED	dce_section_posts	18	1	4 fields	<button>Edit</button>
Citation	ENABLED	dce_citation	17	2	8 fields	<button>Edit</button>

Back-end de l'extension DCE

En soit, la deuxième phase de l'intégration se résume à effectuer la transition entre le code intégré et DCE, puis de recréer la page de la maquette en utilisant les DCE. Il faut tout de même réfléchir à la façon dont on va créer nos DCE. En effet, plus un bloc est réutilisable, mieux c'est. Il faut donc trouver le juste milieu entre le nombre de DCE et leur généralité. Pour ce premier site cette étape a vraiment été longue. Au début j'avais fait un DCE par grande section de page. Il y avait donc très peu de DCE mais ils étaient impossibles à réutiliser ailleurs. Le problème est que chaque section

Rapport de stage

du site commence par le même genre de code HTML (une ouverture de section¹⁰ puis une ouverture de conteneur¹¹), chacune de ces sections termine par le exactement le même code (fermeture du conteneur puis fermeture de section) mais les éléments à l'intérieur de ces sections ne sont jamais exactement les mêmes, et surtout ils sont coincés entre ces ouvertures/fermetures qui sont indispensables. On retrouve pourtant toujours le même genre d'éléments au final : des titres, du texte et des images.

Ma démarche a donc été la suivante. J'ai crée un bloc qui permet d'ouvrir une section et de la paramétriser (car elles ont des paramètres différents). Ensuite j'ai parcouru tous les DCE existants, je leur ai ajouté une option pour fermer un conteneur et j'ai enlevé leur ouverture de conteneur (j'ai copié les paramètres de chacun d'entre eux). Puis j'ai ajouté un bloc d'ouverture de conteneur devant chacun des blocs de ma page et j'ai mis leurs paramètres dans les champs des nouveaux blocs, et j'ai activé la fermeture de conteneur des anciens blocs. Cette étape a permis d'enlever la première et la dernière ligne de tous les DCE, ce qui a mis en évidence le premier élément qui peut venir après une ouverture de section : l'ajout d'une image de fond (avec ou non certains effets) pour la section.

J'ai donc recommencé. J'ai crée un DCE pour ajouter cette image de fond. J'ai supprimé de tous les DCE le code d'image de fond (tout en le sauvegardant de mon côté). Ensuite j'ai ajouté ce nouveau bloc à chaque endroit de la page où il y avait une image de fond. Cela a mis en évidence le prochain élément bloquant des DCE : l'ouverture de conteneur. J'ai réitéré le même processus que pour l'ouverture de section : j'ai ajouté un DCE pour l'ouverture de conteneur, j'ai ajouté dans chaque DCE une option de fermeture de conteneur et j'ai ajouté/mis à jour les blocs de la page.

Après cette dernière étape, je me suis retrouvé face à une situation très intéressante. Je venais de débloquer chacun des éléments présents dans les

10 Balise HTML

11 Élément Bootstrap

différents DCE (les titres, textes, images...). Pour être plus précis, jusqu'à présent ils étaient entourés par du code HTML qui rendait impossible de créer un DCE pour chacun de ces éléments. Mais maintenant que les ouvertures/fermetures de section/conteneur ne sont plus là, il est devenu possible de créer un DCE pour tout ce qui est titres, un pour les textes, un pour les boutons, pour les liens...

J'ai donc continué dans ce sens. Chaque fois que j'ai ajouté un DCE pour un type d'élément, j'ai pu enlever le code de cet élément dans plusieurs des anciens DCE. Et petit à petit, il y avait de moins en moins d'éléments dans les anciens DCE jusqu'à ce qu'ils n'en contiennent plus qu'un seul.

La dernière étape a été de renommer mes anciens blocs afin qu'ils aient un nom explicite. Je venais de finir la deuxième phase de l'intégration du site. Normalement j'aurais dû entrer le contenu final du site dans les pages mais la validation du contenu par ce client a pris beaucoup plus de temps que prévu.

Le système multi-langues

Je ne pense pas pouvoir parler de Typo3 si je ne parle pas de son système de gestion du multi-langues. Dans les paramètres du CMS on peut ajouter une ou plusieurs nouvelles langues à un site. Cela se traduit par une nouvelle colonne dans le back-end des pages. On gère donc les différentes langues des pages dans ces colonnes. Ça c'est pour la création des pages et de leurs traductions. Pour le moment l'utilisateur final n'a pas la possibilité de changer sa langue.

Ensuite il faut savoir que chaque langue a un identifiant. Pour Brochot par exemple c'est 0 pour l'anglais et 2 pour le français.

Maintenant je dois parler du système de lien de Typo3. Chaque page possède un identifiant unique dans chacune de ses langues. Chacun de ces identifiants correspond donc à une URL différente. Il existe une balise HTML spécifique à Typo3 qui permet de faire un lien vers une page du site à partir

de l'identifiant de cette page. Il est cependant possible de passer une option à ce lien. Hors la langue d'une page est également une option. Cela veut dire que si on fait un lien avec l'identifiant d'une page anglaise, mais qu'on lui précise que la langue a le numéro 2 (français), alors le lien sera redirigé vers la version française de la page anglaise. De plus, si on ne précise pas vers quelle page on veut aller, on est redirigé automatiquement vers la page actuelle. C'est très important.

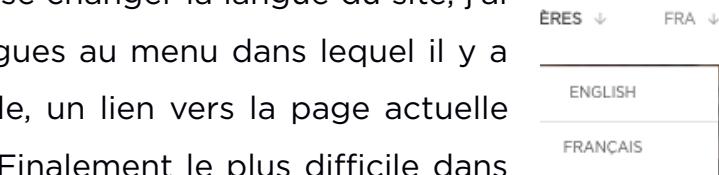
Pour que l'utilisateur final puisse changer la langue du site, j'ai donc ajouté un onglet de langues au menu dans lequel il y a pour chaque langue disponible, un lien vers la page actuelle dans la langue de son choix. Finalement le plus difficile dans le menu de changement de langue est d'afficher la langue actuelle de la page. En effet, vu que la langue est un paramètre Typo3, il faut d'abord

```
# Récupération de la langue
pageLanguage = TEXT
pageLanguage.data = TSFE:sys_language_uid

Extrait du fichier de
configuration TypoScript

<f:if condition="{pageLanguage} == 0">
| <f:then>Fra</f:then>
| <f:else>Eng</f:else>
</f:if>
```

*Extrait du menu de langue
HTML / TypoScript*



créer une variable TypoScript dans la configuration du site, lui donner pour valeur celle de la langue actuelle, et enfin appeler cette variable dans le fichier HTML.

Advineo, le fameux projet WordPress

Une mission qui semble plutôt simple

Trois semaines et demi après le début du stage, Frédérique est venu me voir pour me demander de faire une mise à jour d'affichage sur le site d'Advinéo. Il s'agissait de refaire l'affichage du menu, de faire quelques ajustements dans le footer et d'aérer un peu les pages en modifiant les titres et les blocs image / vidéo.

Ancienne version

Nouvelle version

Cette petite mission, qui aurait dû prendre maximum deux jours, a finalement nécessité cinq jours de travail.

Le désenchantement de WordPress

Le soucis est que le projet Advineo est un chantier total, il y a des morceaux du site de partout (texte seul ou mélangé au code HTML) : depuis le back-

end dans l'éditeur de pages, dans l'extension ACF¹², dans l'éditeur de thème (pour le menu) mais aussi dans des fichiers PHP sources du

thème. Sachant qu'une partie des textes à modifier et une partie du code HTML qui va avec ne sont visibles nulle part, notamment les textes anglais et le texte sous le menu du header (texte que je devais modifier). J'ai sondé

12 Extension WP qui fonctionne comme DCE sur Typo3



Un bloc « titre » avec un lien comme on peut en trouver sur toutes les pages

les tables de la base de données avec des requêtes SQL à la recherche de ce texte, il n'y était pas. J'ai aussi fait un script en shell pour chercher dans les fichiers du projet, pas de résultat. Cependant le script m'a servi pour la suite.

La modification des éléments de page

Comme demandé, j'ai fait la mise à jour de l'affichage des titres et des blocs image/vidéo avec leur description. J'ai cependant dû modifier la structure HTML qui entoure les blocs image/vidéo et leur description. Le gros soucis est qu'une partie du code HTML qui entoure les titres et les blocs image/vidéo ne se trouve pas écrite une fois dans les fichiers PHP du thème, mais se trouve écrite dans chacun des blocs du back-end, pour chaque page qui utilise cet élément.

L'autre partie du code HTML que j'ai modifié se trouve dans les fichiers sources du thème, en PHP. Le soucis est le même que dans le back-end : on retrouve les mêmes structures PHP/HTML dans plusieurs fichiers. Il fallait donc modifier tous ces fichiers.

Ensuite le titre des vidéos était écrit dans un bloc titre différent du bloc vidéo/image. Hors en PHP nous avons besoin de pouvoir récupérer les deux dans la même partie du code.

De plus, le code d'intégration d'une image se met dans la même section que celui d'une vidéo. En PHP il est donc impossible de savoir directement si on traite une image ou une vidéo. Hors le nouvel affichage de la vidéo ne peut pas s'adapter sur l'image. En effet, la vidéo se trouve dans un élément (iFrame) dont la taille varie de façon très particulière. L'image est beaucoup plus libre dans sa façon de changer de taille lorsqu'on redimensionne l'écran, elle est beaucoup plus malléable.

Toute la difficulté de cette partie réside dans le fond coloré qui est à droite de la vidéo ou de l'image. Moi je contrôle la taille du fond mais pas celle de la vidéo. Hors les deux doivent être égaux pour que le fond ne déborde pas.

Il faut donc connaître précisément la taille de la vidéo. L'image n'est pas un problème, je peux contrôler sa taille.

Des solutions viables

Il n'existe pas de solution parfaite mais j'ai fait de mon mieux pour résoudre ces problèmes avec des solutions facilement modifiables.

J'ai commencé par supprimer l'appel du texte (dont la source est introuvable dans le back-end) dans le fichier PHP du header, puis j'ai mis le nouveau texte directement dans le fichier. Normalement ce n'est pas la façon de faire mais il y avait déjà du texte dans le fichier ; nous ne sommes pas à une ligne près.

Avant de changer l'affichage, je devais modifier certains éléments du menu dans le back-end. J'avais déjà eu le temps de me familiariser sur ce CMS au Semestre 4 pour le projet du module de communication. J'ai donc rapidement enlevé les éléments du menu qui devaient aller dans le footer. Ensuite j'ai voulu modifier le footer pour y remettre ces éléments. Impossible de réaliser cette action dans le back-end. Je suis donc allé dans les fichiers PHP et j'ai fait comme pour le header. Il y avait aussi certains textes dans le fichier.

Ensuite dans la partie affichage où j'ai modifié la structure HTML des blocs, il a fallu modifier le code dans chacun des blocs de même type, pour toutes les pages du site. Beaucoup de temps qui aurait été gagné si le code était entièrement écrit dans les fichiers du thème.

Toujours pour modifier les structures HTML, je devais en modifier une partie dans le PHP. C'est là

que j'ai réutilisé le script shell de recherche dans les

```
1  for fichier in `ls *.php`  
2      do if [`cat $fichier | grep "Le texte recherché"` != ""]  
3          then echo Pas ici  
4          else echo Ici : $fichier  
5      fi  
6  done
```

Le script shell de recherche

fichiers. Il m'a permis d'avoir la liste des fichiers à modifier pour chacune des modifications que je devais faire.

Le problème suivant était de pouvoir récupérer en même temps le titre et la partie vidéo en PHP. Pour ça j'ai modifié le bloc image/vidéo dans l'extension ACF du back-end. J'ai simplement ajouté une section titre au bloc puis j'ai modifié toutes les pages où était le bloc pour mettre le titre dans le bloc image/vidéo et supprimer l'ancien bloc de titre.

J'ai aussi crée des sections distinctes pour le lien de la vidéo et pour le lien de l'image, car ils étaient tous les deux dans la même section du bloc. Hors il est impossible d'adapter le nouvel affichage de la vidéo sur l'image. En effet, la vidéo se trouve dans un iFrame1 dont la taille varie de façon très particulière. L'image est beaucoup plus libre dans sa façon de changer de taille lorsqu'on redimensionne l'écran.

Enfin il y a cette histoire de fond coloré qui déborde autour de la vidéo. Comme dit précédemment, elle se redimensionne de façon particulière en fonction de la largeur de l'écran : sa longueur et sa largeur ont une valeur précise au-dessus de 1200px, de 992 à 1199px et de 768 à 991px ; mais elles ont des dimensions variables de 0 à 767px. Il fallait donc utiliser une formule. J'ai mis beaucoup de temps à trouver celle de la hauteur. En même temps, une hauteur de [la largeur de l'écran moins 66px] fois 507 et divisé par 902 ça ne se trouve pas comme ça ! En fait après des tonnes de tests et de recherches j'ai téléchargé la vidéo pour connaître ses dimensions précises, puis j'ai calculé la largeur de l'iFrame qui est de [la largeur de l'écran moins 66px]. Ensuite pour trouver la hauteur j'ai utilisé un produit en croix, mais là encore ce n'était pas suffisant ! Il a fallu que je fasse plein de tests en variant légèrement les dimensions de la vidéo dans le calcul, c'est incompréhensible mais ça marche parfaitement.

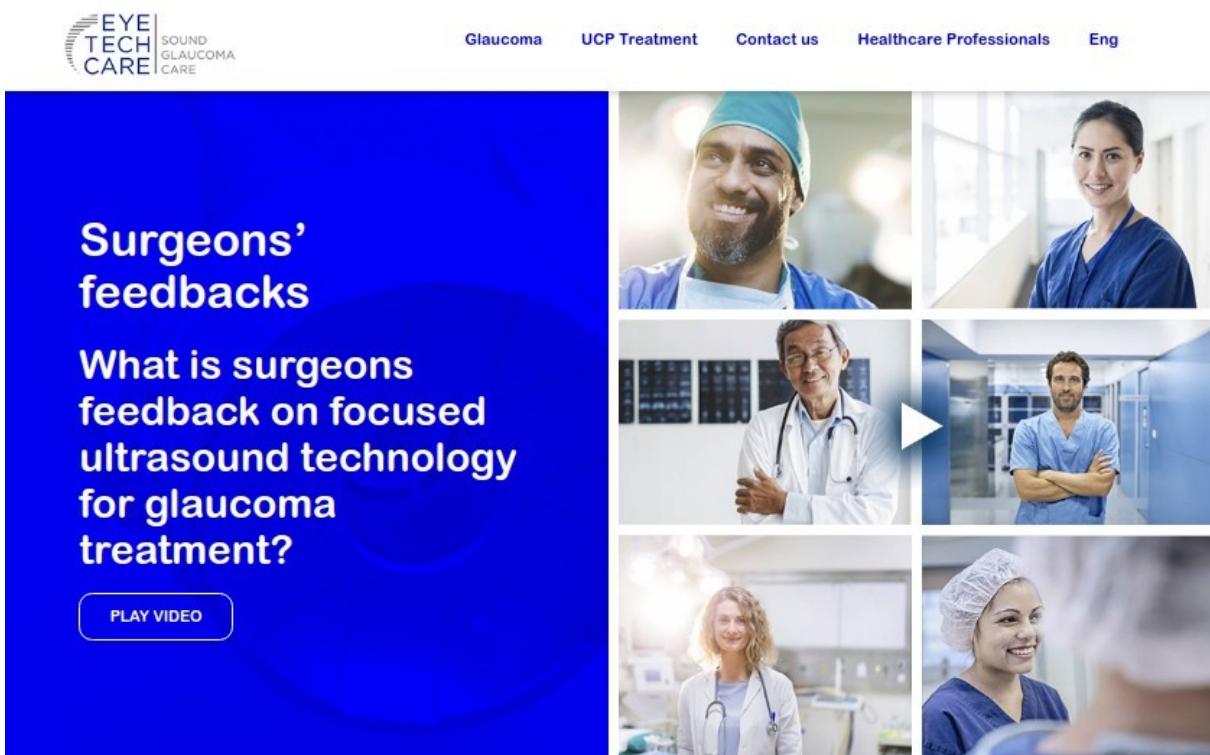
Pour résumer : beaucoup de recherches, séparation du problème en quatre intervalles de largeur (qui redimensionnent différemment la vidéo), puis résolution du problème de dimension de chacun des quatre intervalles (avec des maths et du vaudou pour le plus compliqué).

Rapport de stage

Pour finir sur le projet Advinéo, je dois parler de la résolution du problème des textes anglais qui n'étaient visibles nulle part. Après quatre jours et demi de travail sur le site, par le plus grand des hasards (une erreur de manipulation) j'ai modifié la langue d'une page française : j'ai dit à WP que cette page appartenait aux pages anglaises **puis** j'ai publié la page. Tous les blocs du back-end ont disparu ! J'ai fait l'inverse, tous les blocs ont réapparu. Mes collègues m'ont affirmé que cela ne pouvait être qu'un bug. J'ai donc réitéré le procédé pour faire les mises à jour des quelques pages du site anglais.

Eye Tech Care

Le site de Eye Tech Care est un site vitrine sous Typo3. Les contraintes sont les mêmes que pour le site de Brochot sauf que le client doit pouvoir faire plus que modifier le contenu de ses pages. Il doit pouvoir construire ses propres pages en y ajoutant différentes sections qui se trouvent dans la maquette. Ce point est important car ça n'a rien à voir avec le site de Brochot, il faut vraiment un système simple pour ajouter du contenu sur les pages.



Haut de la page d'accueil

La récupération du projet

Le site de Eye Tech Care avait déjà été commencé par mon collègue Sébastien. Je n'ai donc pas eu besoin de m'occuper de l'installation et il y avait déjà une partie du contenu à moitié intégrée pour la première phase d'intégration. Il y avait également les fichiers d'un thème tout fait, comme pour Brochot, dans les fichiers du site web. J'ai commencé par faire un tour de l'existant sur le site.

Il y avait déjà une base de CSS pour le template, ainsi que quelques animations. Les 3/4 du code HTML de la home étaient déjà présents. Cependant la page n'était pas du tout responsive. Il y avait aussi un compte à rebours seconde par seconde de 24h. Il fallait le modifier pour qu'il puisse compter de 75 millions à 112 millions. Dans une optique de réutilisation du code, il fallait en fait modifier ce compte à rebours pour qu'il aille d'un nombre A à un nombre B en affichant les chiffres des centaines, milliers et millions. Sur le site il y avait aussi un menu temporaire, qui n'était pas fait pour les mobiles.

Enfin, dans la maquette du site il y avait un système de slider à réaliser. Il s'agit d'une sorte de diaporama mais au lieu de fonctionner avec seulement des images, cela doit fonctionner avec toute sorte de contenu qu'on peut avoir sur la page. Il y avait aussi une liste déroulante qui devait être reliée à quatre blocs de statistiques. On devait pouvoir choisir un pays dans la liste déroulante, et les blocs de statistiques devaient afficher la valeur correspondant au pays.

Pour résumer le travail à faire sur ce site, il y avait beaucoup de gestion de blocs avec des images à rendre responsive, quelques systèmes d'animation à gérer, le thème à ajouter sur le site pour utiliser son système de menu / menu téléphone ainsi que certaines animations si besoin, et pour finir beaucoup de contenu à rentrer.

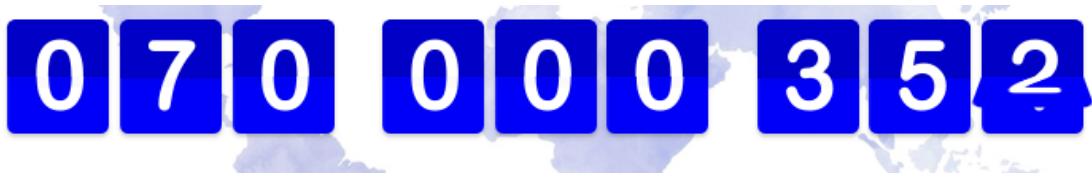
La première phase de l'intégration

Le travail réalisé sur Eye Tech Care est vraiment énorme. Déjà j'ai voulu rendre responsive le contenu existant sur le site. Il y avait beaucoup d'images de taille très variable à gérer, j'ai dû apprendre à utiliser de nouvelles propriétés CSS sur la gestion des images pour pouvoir tout faire. Ce travail peut sembler simple mais il y avait vraiment beaucoup de dispositions possibles entre les images des sections pleine largeur, les images des sections de contenu de page, les images des sections galeries (plusieurs images de taille différentes qui forment un bloc de galerie), les

images de fond de certaines sections et enfin les images qui servent d'icône.

Ensuite il y avait tout le système de titre/texte du site à réaliser. Il faut penser pour chaque taille de titre à vérifier qu'elle n'est pas trop grosse pour certaines dimensions d'écran, qu'elle ne déborde pas hors de l'écran, que leur couleur est bien visible sur le fond de la page (car d'un bloc à l'autre la couleur de fond du bloc change et on peut vite se retrouver à écrire blanc sur fond blanc, pas super niveau lisibilité...). Il faut aussi prévoir que pour des raisons esthétiques, que les titres de chaque niveau peuvent avoir besoin de s'afficher comme s'ils étaient d'un autre niveau. Par exemple un titre de niveau 1 peut avoir la taille d'un titre de niveau 2 si ça semble plus cohérent d'un point de vue visuel. Le but est ainsi de gérer les tailles indépendamment du niveau de titre pour toujours avoir un système de titre cohérent, ce qui est très important pour le référencement.

Il y a eu beaucoup de développement JavaScript / jQuery. En effet, déjà il y a eu le système de compte à rebours à modifier, beaucoup modifier même. J'ai commencé par comprendre comment fonctionnait le compte à rebours, puis j'ai modifié le nom de ces éléments (heure, minute, seconde) en millions, milliers, centaines et je leur ai ajouté un troisième chiffre. J'ai modifié le sens de compte à rebours pour qu'il compte au lieu de décompter, j'ai recodé la façon dont les chiffres interagissent entre eux et pour finir j'ai ajouté un système pour paramétriser le point de départ et le point d'arrivée du compteur.

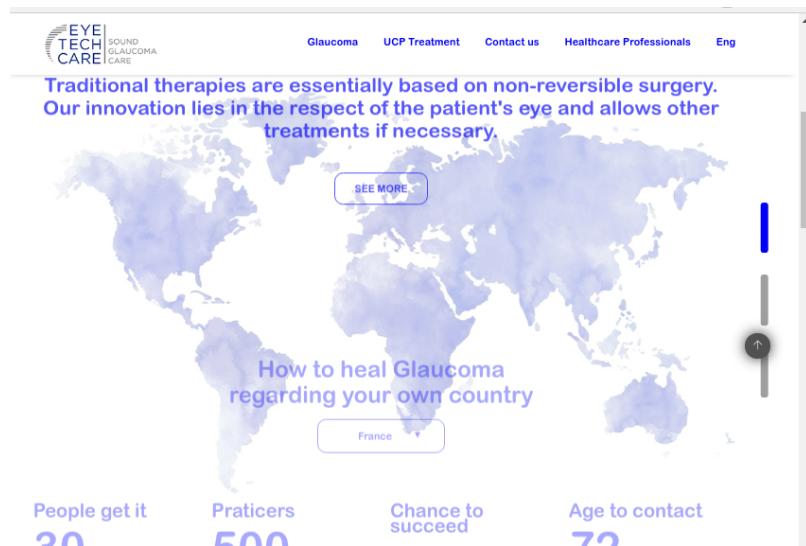


Compteur en action

Aussi j'ai entièrement réalisé le système de slider. J'ai crée toutes les animations nécessaires en CSS. J'ai utilisé du JavaScript pour lancer chacune des animations de la bonne façon, pour faire fonctionner le

Rapport de stage

changement de slide et pour gérer le défilement automatique des slides (c'est justement l'objet de mon point technique, c'est pourquoi je n'en parle pas trop ici). Pour être plus précis sur les animations des slides, il y a d'abord la détection du nombre de slides qui va insérer les boutons de changement de slide. Ensuite il y a un système qui détecte lorsqu'on clique sur l'un de ces boutons. Il va alors détecter quelle est la slide active, puis sur quelle slide l'utilisateur veut aller, puis il va lancer soit la première animation si la nouvelle slide se trouve au-dessus de la slide active, sinon (si la prochaine slide se trouve en-dessous de la slide active) il va lancer la deuxième animation. Bon d'un point de vue technique il n'y a qu'une seule animation qui a un sens, mais concrètement il y a deux animations qui en résultent. Une animation (disons dans le sens 1) se déroule de la façon suivante : on déplace la nouvelle slide (invisible) au-dessus de la slide active, puis on va rendre visible les animation sur les deux slides, et on va en même temps déplacer les deux slides vers le bas jusqu'à ce que la slide du haut ai pris la place de celle du bas. En même temps on va rendre la slide du bas invisible et celle du haut visible. Une fois fini on désactive la visibilité des animations sur les slides.



Transition entre la slide 3 et la slide 1

Pour continuer sur le JavaScript, j'ai écrit un programme qui permette de faire le lien entre la liste déroulante des pays et les blocs de statistiques qui sont affichés après. En fait pour chaque pays ajouté à la liste, on ajoute les valeurs statistiques du pays. Ensuite je détecte lorsque la valeur du pays change et je vais récupérer chacune des quatre valeurs pour la mettre dans

le bloc correspondant. En pratique il faut aussi faire une sauvegarde de l'unité de la valeur de chaque statistique pour ne pas les supprimer.



Pays sélectionné : France

Pays sélectionné : Chine

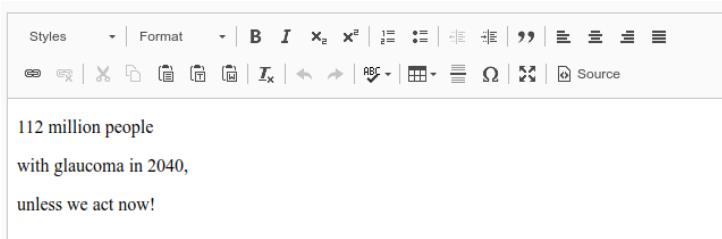
Enfin il y a eu l'intégration du thème qui a complètement planté le site. Le thème utilise de très nombreux fichiers CSS et encore plus de JavaScript. J'ai dû faire beaucoup de tests pour désactiver le chargement d'un maximum de fichiers et il restait encore des dizaines d'erreurs JS. J'ai été dans plusieurs scripts pour désactiver de force un certain nombre de fonctionnalités et résoudre tous ces problèmes. De plus, le site prenait plusieurs secondes pour charger à cause de tous ces fichiers. Heureusement j'ai trouvé un réglage dans la configuration du template, il permet de fusionner tous les fichiers entre eux et de les compresser. Ça a multiplié par dix la vitesse de chargement.

La deuxième phase de l'intégration

La deuxième phase d'intégration a vraiment été super intéressante pour moi. J'ai utilisé la même extension que sur le site de Brochot mais je ne m'en suis pas du tout servi de la même façon. Il n'y avait pas de système de conteneur ou d'ouverture/fermeture de section. Seulement des blocs de contenu et plein d'options. Vu que le client doit pouvoir facilement insérer des nouveaux blocs de contenu lui-même j'ai utilisé beaucoup plus de TypoScript dans le back-end des DCE afin que client n'ai que des cases à cocher, des zones de texte à remplir et des niveaux de texte à sélectionner dans des listes déroulantes.

Du côté développement j'ai vraiment utilisé beaucoup d'options TypoScript. Déjà j'ai beaucoup de balises de conditions et de balises de boucle mais j'ai aussi utilisé des options avancées avec des calculs sur les index des éléments et je me suis beaucoup servi des balises de formatage. Il s'agit

d'une sorte de filtre à balises HTML, ça permet de transformer du texte en code HTML, de supprimer certaines balises ou encore de remplacer une balise par une autre. En fait il faut utiliser ce type de filtres lorsqu'on utilise un champs TRE dans un DCE. Il s'agit d'un champs d'édition de texte qui permet notamment de gérer l'alignement du texte, le gras, l'italique et surtout cela permet d'ajouter un lien sur une partie du texte. Le soucis c'est qu'il y a d'autres réglages qui peuvent faire planter l'affichage du bloc. C'est pour filtrer ces réglages que j'ai utilisé les options de formatage.



Un champ RTE

Le défi de cette partie a tout de même été de trouver de quelle façon ajouter un slider sur une page, car dans le slider il faut pouvoir mettre n'importe quel bloc ! En fait la solution n'était pas très compliquée. J'ai créé un élément de séparation de slide qui permet (au choix) de commencer un slider, de séparer deux slides ou de finir un slider. Il est aussi possible d'ajouter un fond d'écran pour tout le slider lorsqu'on est en mode fin de slider.

Tout comme pour Brochot, j'ai ajouté un menu de sélection de langue. Il fonctionne exactement de la même façon, je ne vais donc pas en dire plus sur cet élément.

Enfin j'ai passé beaucoup de temps à ajouter le contenu du site lorsque j'ai refait les pages de la maquette avec les blocs du back-end. En fait il y avait un certain nombre de changements dans le code de la page et j'ai dû reprendre une partie du code CSS.

Point technique : la gestion du slider

Le choix du point technique

Choisir le point technique n'a pas été facile. J'ai fait beaucoup de choses compliquées tout au long du stage mais en même temps il n'y avait rien qui me semblait vraiment mériter un point technique. Finalement après avoir réfléchi sur la question je me suis dit que le point technique n'est pas forcément le problème le plus long à résoudre, ni celui qui fait le plus de ligne. Le point technique que j'ai choisi est le problème qui m'a demandé le plus de connaissances techniques en algorithmie. Il s'agit d'un problème de programmation asynchrone que j'ai rencontré lorsque j'ai développé le système de défilement automatique des slides du site de Eye Tech Care. Le langage utilisé est JavaScript avec jQuery.

La problématique du défilement automatique

Pour expliquer ce problème je dois expliquer ce qui influe sur le défilement des slides. Tout d'abord il y a le défilement automatique qui doit changer de slide (passer à la suivante) toutes les 3,5 secondes. Ensuite un clique sur le bouton d'une slide doit faire défiler le slider sur cette slide, sauf si elle est déjà active. Il faut également que le délai de 3,5 secondes reparte de 0 après un clique de l'utilisateur pour changer de slide. Enfin il faut stopper le défilement automatique lorsque la souris est sur une slide. Le défilement automatique doit reprendre dès que l'utilisateur enlève sa souris de la slide. Il faut alors repartir de 0 pour le délai.

Le fonctionnement algorithmique de base est le suivant (chaque point est une partie indépendante du programme) :

- au chargement du programme on lance la première slide sans animation puis on fait appel au *lanceur de la slide suivante* (un programme)

- le *lanceur de la slide suivante* détecte quelle est la prochaine slide puis il envoie dans le futur (3,5 secondes plus tard) l'ordre de charger la slide qu'il avait détecté comme la suivante
- 3,5 secondes après l'exécution du *lanceur de slide suivante*, l'ordre de changement de slide modifie la slide active avec l'animation
- modifier la slide active avec l'animation déclenche de nouveau le *lanceur de slide suivante*
- cliquer sur un bouton de changement de slide (s'il ne s'agit pas de la slide déjà active) va modifier la slide active pour charger la nouvelle slide, avec l'animation (donc relancer le *lanceur automatique de slide*).

Le gros problème est le suivant : chaque appel au lanceur automatique de slide forcera le changement de slide 3,5 secondes plus tard. Cela veut dire que plusieurs ordres de changement de slide peuvent arriver presque en même temps.

Concrètement pour l'utilisateur final, les slides font n'importe quoi.

Pour être plus précis sur le problème, il y a des interférences, dues au fonctionnement asynchrone, entre les ordres de la souris et ceux du lanceur automatique.

L'exclusion mutuelle, une solution efficace

Je me suis rendu compte que ce problème fonctionne exactement comme les problèmes d'accès concurrentiels (différentes parties du programme veulent accéder à une ressource partagée en même temps). Nous avons justement étudié ce type de problème en cours de Système d'Exploitation. La solution est d'utiliser des variables d'exclusion mutuelle (des mutex). En fait chaque ressource est associé à une variable (quelque chose qui contient une valeur). La valeur par défaut est 0 pour dire que la ressource est libre. Lorsqu'on veut l'utiliser, si elle est libre, alors on change la valeur de son mutex (à 1 normalement) puis lorsqu'on a fini de se servir de cette ressource, on remet la valeur de son mutex à 0.

Bon en pratique il fallait plus qu'un simple mutex pour résoudre le problème mais toutes les solutions sont basées là dessus.

J'ai utilisé un premier mutex qui indique qu'une action due à la souris est en cours. Il permet notamment d'empêcher l'utilisateur de changer de slide en plein pendant une transition. De cette façon il ne peut pas faire dysfonctionner le slider en faisant n'importe quoi avec la souris.

Ensute pour le défilement automatique j'ai dû pousser plus loin l'utilisation des cours de système¹³. Il faut bien un mutex (différent du premier, on l'appellera mutex2), mais pas la version normal qui prend pour valeur 0 ou 1. Le soucis du slider est que seul le dernier ordre de changement de slide doit être pris en compte. Pour ça, le mutex2 doit agir sous forme de tickets : chaque ordre prend un ticket avant de patienter les 3,5 secondes. Chaque fois qu'on prend un ticket, la valeur du mutex2 augmente. Une fois passé ce délai, il regarde si la valeur affichée sur le mutex2 est celle de son ticket. Si c'est le cas, alors c'est que c'est à lui d'effectuer le changement de slide. Si la valeur de son ticket n'est pas bonne, alors c'est qu'un autre ordre a la priorité sur lui. Il doit donc annuler son changement de slide.

Pour finir j'ai utilisé du jQuery pour détecter lorsque la souris arrive sur la slide et lorsqu'elle en repart. Le fait d'entrer sur la slide fait augmenter le mutex2. Ainsi plus aucun ticket n'est valide et les slides ne défilent plus, sauf en cas de clique sur les boutons. Le fait de sortir la souris de la slide relance le défilement automatique.

En résumé, toutes les contraintes sont respectées, il est impossible de faire planter le slider ou le défilement automatique, et il n'y a pas d'attente active (programme qui consomme des ressources alors qu'il ne fait rien).

13 Heureusement pour moi, j'avais appris par cœur cette partie !

Bilans du stage

Bilan technique

D'un point de vue technique c'est une évidence que ce stage a été très enrichissant. J'ai pu mettre en pratique et renforcer plein de connaissances que j'ai acquise à l'IUT, pour une fois tous les domaines se rejoignent et c'est génial de voir tout ça interagir. J'ai toujours beaucoup mieux compris les cours grâce à la pratique et avec ce stage je sens que j'ai vraiment beaucoup progressé en web. En plus le fait de travailler sur des vrais projets m'a beaucoup motivé pour perfectionner mon travail au maximum.

J'ai appris bien plus que je n'avais prévu en CSS, j'ai découvert le CMS Typo3 que j'ai vraiment adoré et je suis sûr de m'en resservir pour créer mon site personnel. J'ai géré toute la partie responsive de plusieurs sites, c'est un travail monstre. Il faut penser à tous les cas de figure pour que ça fonctionne sur toutes les pages. C'est cependant un travail qui m'a poussé à apprendre le système de taille/de grilles utilisé par Bootstrap ou encore Bulma, je suis sûr de les réutiliser. J'ai pas mal amélioré mes compétences en JavaScript (l'un de mes langages préférés d'ailleurs). J'ai appris à me servir de jQuery dont on parle temps.

Il y a des choses dont je n'ai pas pu parler dans le reste du rapport mais j'ai aidé à utiliser les clés ssh, à faire des migrations de sites web, de bases de données, des redirections de DNS, j'ai découvert comment automatiser des sauvegardes de bases de données et j'ai découvert l'étendu des problèmes de piratage de site.

En plus de tout ça j'ai appris plein de choses sur les normes de programmation, les normes de responsive design et les normes d'ergonomie. J'ai plusieurs références de documentation sur ces sujets que je garde précieusement de côté. J'ai aussi compris comment choisir entre Wordpress, Typo3, Drupal ou Prestashop. De même, j'ai découvert plein de technologies que je n'ai pas pu mettre en pratique mais que j'ai noté et qui seront le sujet de mes prochaines auto-formations.

Bilan professionnel

Pour le site de Brochot pour le moment mon travail est fini. Le site fonctionne, il y a énormément de possibilités de personnalisation des pages. J'ai testé tous les éléments qu'on peut mettre sur le site, en soit tout fonctionne. Il ne manque plus que les retours du client sur le contenu et je vais pouvoir m'occuper de le mettre sur le site car je reste travailler chez Influactive cet été.

Le site d'Advinéo est fini, toutes les pages utilisent le nouvel affichage. Nous avons eu un retour du client il y a quelques jours. Globalement ils aiment le nouvel affichage mais ils ont trouvé plusieurs bugs et ils ne sont pas tout à fait convaincu par le menu. Il est donc probable que je revienne dessus. Pour le moment rien n'a été décidé.

Enfin, pour Eye Tech Care toute la partie intégration est terminée. J'ai testé tous les blocs du back-end, pas de problème visible. Les animations marchent, le système de slider fonctionne bien, l'affichage des statistiques en fonction des pays fonctionne aussi, de même pour le compteur. Le site est optimisé pour le référencement.

Nous venons de recevoir les textes anglais définitifs, je suis entrain de les ajouter aux différentes pages du site. Le contenu anglais est ajouté à environ 30 %.

Bilan personnel

Sur le plan personnel ce stage m'a beaucoup apporté. Déjà il m'a permis de confirmer que je ne souhaite pas faire une carrière en web mais que le développement fera partie de mon quotidien. Même si j'ai beaucoup aimé ce stage et que pour le moment je me plaît dans ce domaine, au travers des échanges que j'ai eu avec mes collègues, je trouve que le web évolue vraiment trop vite. C'est un domaine où il faut tout le temps se mettre à jour sur les dernières technologies qui évoluent constamment. Pour le moment je suis dans le bain mais à long terme je recherche un domaine plus stable.

J'ai eu la chance de faire mon stage dans une petite structure. Avant je pensais vouloir aller dans une moyenne ou grande entreprise mais je suis complètement revenu sur cet avis. Tout l'avantage d'une petite structure est qu'on peut mettre à profit notre polyvalence. Moi qui adore toucher un peu à tous les domaines, aujourd'hui je suis sûr de vouloir continuer à apprendre un maximum de choses dans les domaines informatiques et j'ai envie de trouver ma place dans une petite ou moyenne structure où je puisse mettre à profit cette polyvalence.

Enfin, ce stage m'a permis de découvrir réellement le milieu professionnel. Les contraintes professionnelles, la satisfaction du client, les réunions de productions ; tout ça change complètement la donne. Ce sont pour moi des sources de motivation. J'ai aussi dû faire face à de nombreux problèmes que j'ai résolu. Ça m'a beaucoup fait gagner en autonomie. J'ai pu voir toute la chaîne de création d'un site web, et pas seulement du point de vue technique. Je dois dire que c'est très satisfaisant de voir la finalité de ses projets.

Conclusion

Ce stage vient finaliser mes deux premières années d'étude universitaire. D'un côté c'est l'aboutissement de centaines d'heures de travail, mais c'est aussi le début de ma vie professionnelle. Il n'y a pas de doute à avoir, les enseignements que nous avons reçu à l'IUT, qu'ils soient techniques ou généraux, correspondent parfaitement aux besoins du monde de l'entreprise.

Ce stage m'a beaucoup apporté sur tous les plans. D'un point de vue technique il m'a permis de renforcer mes compétences informatiques acquises à l'IUT, il m'a fait apprendre de nouvelles technologies et il m'a fait comprendre une multitude de liens entre les différents domaines informatiques et les différentes technologies. D'un point de vue humain il m'a fait découvrir le monde de la petite entreprise (enfin plutôt de la petite société si on joue sur les mots), j'ai pu voir comment on travaille au sein d'une équipe d'informaticiens et quelle part on représente sur la réalisation d'un projet. Enfin ce stage m'a beaucoup conforté sur mon projet d'étude et professionnel. J'ai pu vérifier que j'aime faire du développement comme activité principale et j'ai pu écarter un domaine informatique pour la suite de mes études. Pour moi qui ne sait pas exactement dans quel domaine je vais me spécialiser, savoir vers quoi je ne veux pas aller est très important.

Pour finir je dirais que je suis très satisfait de ce stage. J'ai eu la chance de travailler avec beaucoup de technologies différentes. et je vais encore travailler pendant deux mois pour Influactive. Je vais découvrir comment réaliser un site de vente, ce qui est une nouvelle pour moi.

Cette expérience professionnelle a été très positive pour moi et je compte bien la mettre en avant !

Sources

Documentation Bootstrap : <https://getbootstrap.com/docs/4.0/layout/grid/>

Thème source pour Brochot : <http://themes.suelo.pl/okno>

Documentations Typo3 :

- <https://docs.typo3.org/>
- <https://wiki.typo3.org/Fluid>
- <https://fluidtypo3.org/>
- <https://typo3buddy.com/home/>
- <https://www.nitsan.in/blog/post/30-tips-tricks-for-optimum-typo3-website-speed-and-performance/>

Gestion des images :

- <https://developer.mozilla.org/fr/docs/Web/CSS/object-fit>
- <https://la-cascade.io/utiliser-css-object-fit/>
- https://www.w3schools.com/css/css3_object-fit.asp
- <https://www.alsacreations.com/xmedia/tuto/html5/srcset/srcset-image-responsive.html>

Animations staggering : <https://css-tricks.com/staggering-animations/>

Normes Google pour l'affichage et les animations d'un site :
<https://material.io/design/shape/about-shape.html>

Sommaire des annexes

Annexe 1 : Lanceur automatique des applications au démarrage_____	45
Annexe 2 : Arborescence du template d'un site Typo3_____	46
Annexe 3 : Fonctions JS additionnelle de Eye Tech Care _____	46
Annexe 4 : Fonction JS de gestion des animations Eye Tech Care_____	47
Annexe 5 : Fonction JS de mise à jour des statistiques Eye Tech Care____	48
Annexe 6 : Fonction JS de gestion du compte à rebours Eye Tech Care_	49
Annexe 7 : Fonction JS de gestion du slider Eye Tech Care_____	51
Annexe 8 : Blocs avancé du back-end d'un DCE_____	53
Annexe 9 : Template HTML / TypoScript d'un DCE avancé_____	54
Annexe 10 : Header-Footer HTML / TypoScript de Eye Tech Care_____	55

Rapport de stage

Annexe 1 : Lanceur automatique des applications au démarrage

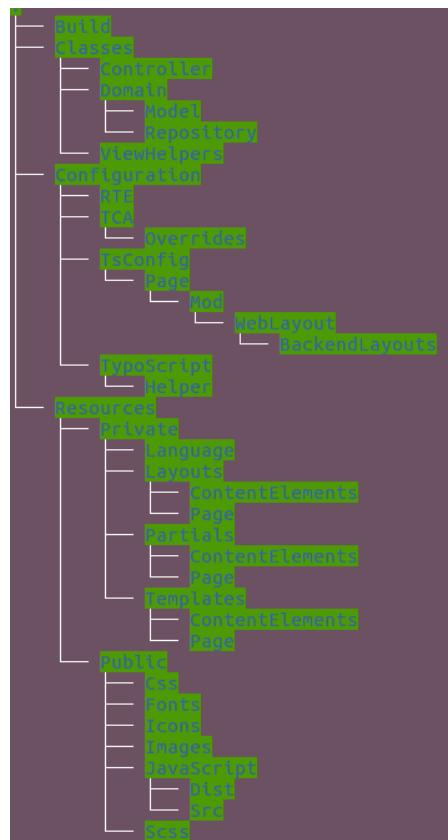
```
# layout.min.css      JS simple_animations.js      JS slider.js      JS countdown.js      Default.html      demarrage.sh •
home julien Bureau demarrage.sh
1 #nautilus sftp://51.38.178.77/var/www/html/brochot.com/ &
2 #nautilus sftp://51.38.178.77/var/www/html/brochot.com/typo3conf/ext/bi_template/Resources/Private/Templates/Page/ &
3 #nautilus sftp://51.38.178.77/var/www/html/brochot.com/typo3conf/ext/bi_template/Resources/Public/Css/ &
4 #nautilus sftp://51.38.178.77/var/www/html/brochot.com/typo3conf/ext/bi_template/Resources/Public/JavaScript/Src/ &
5 nautilus sftp://51.38.178.77/var/www/html/eyetechcare.com/typo3conf/ext/bi_template/Resources/Private/Layouts/Page/ &
6 nautilus sftp://51.38.178.77/var/www/html/eyetechcare.com/typo3conf/ext/bi_template/Resources/Public/Css/ &
7 nautilus sftp://51.38.178.77/var/www/html/eyetechcare.com/typo3conf/ext/bi_template/Resources/Public/JavaScript/Dist/ &
8 #nautilus sftp://149.202.195.149/var/www/html/preprod/advineo/wp-content/themes/advineo/ &
9 chromium-browser &
10 sleep 5;
11 code /run/user/1000/gvfs/sftp:host=51.38.178.77/var/www/html/eyetechcare.com/typo3conf/ext/bi_template/Resources/
Private/Layouts/Page/Default.html &
12 sleep 30;
13 code /run/user/1000/gvfs/sftp:host=51.38.178.77/var/www/html/eyetechcare.com/typo3conf/ext/bi_template/Resources/Public/
Css/layout.min.css &
14 sleep 0.2;
15 code /run/user/1000/gvfs/sftp:host=51.38.178.77/var/www/html/eyetechcare.com/typo3conf/ext/bi_template/Resources/Public/
JavaScript/Dist/slider.js &
16 code /run/user/1000/gvfs/sftp:host=51.38.178.77/var/www/html/eyetechcare.com/typo3conf/ext/bi_template/Resources/Public/
JavaScript/Dist/simple_animations.js &
17 code /run/user/1000/gvfs/sftp:host=51.38.178.77/var/www/html/eyetechcare.com/typo3conf/ext/bi_template/Resources/Public/
JavaScript/Dist/countdown.js &
18 exit
19 code /run/user/1000/gvfs/sftp:host=51.38.178.77/var/www/html/brochot.com/typo3conf/ext/bi_template/Resources/Private/
Templates/Page/Default.html &
20 sleep 20;
21 code /run/user/1000/gvfs/sftp:host=51.38.178.77/var/www/html/brochot.com/typo3conf/ext/bi_template/Resources/Private/
Layouts/Page/Default.html &
22 sleep 0.2;
23 code /run/user/1000/gvfs/sftp:host=51.38.178.77/var/www/html/brochot.com/typo3conf/ext/bi_template/Resources/Private/
Templates/Page/sauvegarde\ site\ statique.html &
24 sleep 0.2;
25 code /run/user/1000/gvfs/sftp:host=51.38.178.77/var/www/html/brochot.com/typo3conf/ext/bi_template/Resources/Public/Css/
main.css &
26 sleep 0.2;
27 code /run/user/1000/gvfs/sftp:host=51.38.178.77/var/www/html/brochot.com/typo3conf/ext/bi_template/Resources/Public/
JavaScript/Src/main.js &
28 sleep 0.2;
29 exit
30 code /run/user/1000/gvfs/sftp:host=149.202.195.149/var/www/html/preprod/advineo/wp-content/themes/advineo/style.css &
31 sleep 0.2;
32 code /run/user/1000/gvfs/sftp:host=149.202.195.149/var/www/html/preprod/advineo/wp-content/themes/advineo/header.php &
33 sleep 0.2;
34 code /run/user/1000/gvfs/sftp:host=149.202.195.149/var/www/html/preprod/advineo/wp-content/themes/advineo/footer.php &
35 sleep 0.2;
36 code /run/user/1000/gvfs/sftp:host=149.202.195.149/var/www/html/preprod/advineo/wp-content/themes/advineo/
page-produit.php &
```

Rapport de stage

Annexe 2 : Arborescence du template d'un site Typo3

Arborescence du template elle-même dans :

racine_du_site/typo3_conf/ext/nom_du_template/...



Annexe 3 : Fonctions JS additionnelles de Eye Tech Care

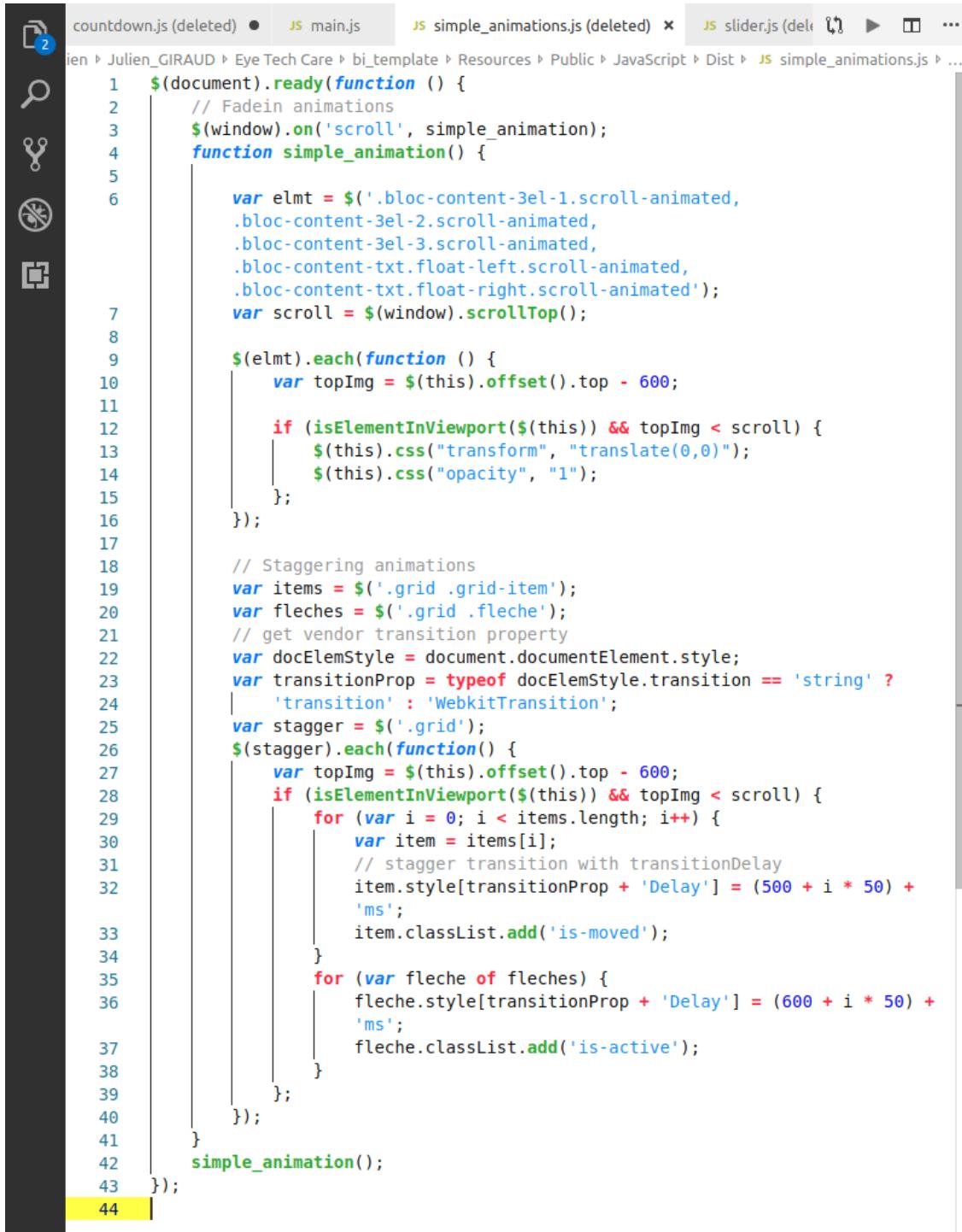
Le code editor affiche une fonction `sleep(ms)` qui utilise la méthode `setTimeout` pour faire attendre l'exécution d'une promesse. Ensuite, il définit une fonction `isElementInViewport(c)` qui vérifie si un élément jQuery est visible dans le viewport. La fonction utilise la méthode `getBoundingClientRect` pour obtenir les coordonnées de l'élément et compare ses positions avec celles de la fenêtre.

```
function sleep(ms) {
    return new Promise(resolve => setTimeout(resolve, ms));
}

function isElementInViewport(c) {
    if (typeof jQuery === "function" && c instanceof jQuery) {
        c = c[0];
    }
    if (!c) return;
    var rect = c.getBoundingClientRect();
    return rect.top <= (window.innerHeight || document.documentElement.clientHeight) && rect.bottom >= 0;
}
```

Rapport de stage

Annexe 4 : Fonction JS de gestion des animations Eye Tech Care



```
countdown.js (deleted) ● JS main.js JS simple_animations.js (deleted) ✘ JS slider.js (dele ⌂ ► └ ...  
ien > Julien_GIRAUD > Eye Tech Care > bi_template > Resources > Public > JavaScript > Dist > JS simple_animations.js > ...  
1  $(document).ready(function () {  
2      // Fadein animations  
3      $(window).on('scroll', simple_animation);  
4      function simple_animation() {  
5  
6          var elmt = $('.bloc-content-3el-1.scroll-animated,  
7              .bloc-content-3el-2.scroll-animated,  
8              .bloc-content-3el-3.scroll-animated,  
9              .bloc-content-txt.float-left.scroll-animated,  
10             .bloc-content-txt.float-right.scroll-animated');  
11             var scroll = $(window).scrollTop();  
12  
13             $(elmt).each(function () {  
14                 var topImg = $(this).offset().top - 600;  
15  
16                 if (isElementInViewport($(this)) && topImg < scroll) {  
17                     $(this).css("transform", "translate(0,0)");  
18                     $(this).css("opacity", "1");  
19                 };  
20             });  
21  
22             // Staggering animations  
23             var items = $('.grid .grid-item');  
24             var fleches = $('.grid .fleche');  
25             // get vendor transition property  
26             var docElemStyle = document.documentElement.style;  
27             var transitionProp = typeof docElemStyle.transition == 'string' ?  
28                 'transition' : 'WebkitTransition';  
29             var stagger = $('.grid');  
30             $(stagger).each(function(){  
31                 var topImg = $(this).offset().top - 600;  
32                 if (isElementInViewport($(this)) && topImg < scroll) {  
33                     for (var i = 0; i < items.length; i++) {  
34                         var item = items[i];  
35                         // stagger transition with transitionDelay  
36                         item.style[transitionProp + 'Delay'] = (500 + i * 50) +  
37                         'ms';  
38                         item.classList.add('is-moved');  
39                     };  
40                 };  
41             });  
42             simple_animation();  
43         });  
44     
```

Annexe 5 : Fonction JS de mise à jour des statistiques

```
45 function start_update_select() {
46     var select_parent, select_childs, select_i, sauvegarde_unite = []; // Variables pour le script
47     // Récupération du sélecteur (s'il existe)
48     select_parent = document.getElementById("select_parent");
49     // Si le sélecteur n'existe pas on ferme le programme
50     if (!select_parent) return;
51     // Récupération des blocs d'affichage
52     select_childs = document.querySelectorAll(".select_child");
53     // Sauvegarde du contenu initial de ces blocs
54     for (select_i = 0; select_i < 4; select_i++)
55         sauvegarde_unite.push(select_childs[select_i].innerHTML);
56     // Fonction principale
57     function update_select() {
58         for (select_i = 1; select_i < 5; select_i++) { // Pour chacun des 4 attributs modifiables
59             // Chaque bloc se voit attribuer la valeur de l'attribut correspondant dans le pays sélectionné, plus le contenu initial du bloc
60             select_childs[select_i-1].innerHTML = select_parent.options[select_parent.selectedIndex].getAttribute("att"+select_i) +
61             sauvegarde_unite[select_i-1];
62         }
63     update_select(); // Lancement de la fonction au chargement de la page
64     $(window).on('change', update_select); // Lancement de la fonction à chaque changement du pays sélectionné
65 }
66 // Lancement de tout le programme une fois le chargement de la page terminé
67 window.addEventListener("load", start_update_select);
68 }
```

Annexe 6 : Fonction JS de gestion du compte à rebours



```
JS countdown.js (deleted) ● JS main.js JS simple_animations.js JS sliderjs ⏪ ⏴ ⏵ ⏹ ⏷ ...  
dia > julien > Julien_GIRAUD > Eye Tech Care > bi_template > Resources > Public > JavaScript > Dist > JS countdown.js > 🗑  
8 var Countdown = {  
9     // Backbone-like structure  
10    $el: $('.countdown'),  
11    // Params  
12    countdown_interval: null,  
13    total_centaines: $('.countdown').attr('data-final-value'),  
14    // Initialize the countdown  
15    init: function () {  
16        // DOM  
17        this.$ = {  
18            millions: this.$el.find('.bloc-number.millions .figure'),  
19            milliers: this.$el.find('.bloc-number.milliers .figure'),  
20            centaines: this.$el.find('.bloc-number.centaines .figure')  
21        };  
22        // Init countdown values  
23        this.values = {  
24            millions: this.$millions.parent().attr('data-init-value'),  
25            milliers: this.$milliers.parent().attr('data-init-value'),  
26            centaines: this.$centaines.parent().attr('data-init-value'),  
27        };  
28        // Initialize total centaines  
29        this.total_centaines = this.values.millions * 1000 * 1000 +  
30        this.values.milliers * 1000 + this.values.centaines;  
31        // Animate countdown to the end  
32        this.count();  
33    },  
34    count: function () {  
35        var that = this,  
36            $million_1 = this.$millions.eq(0),  
37            $million_2 = this.$millions.eq(1),  
38            $million_3 = this.$millions.eq(2),  
39            $millier_1 = this.$milliers.eq(0),  
40            $millier_2 = this.$milliers.eq(1),  
41            $millier_3 = this.$milliers.eq(2),  
42            $centaine_1 = this.$centaines.eq(0),  
43            $centaine_2 = this.$centaines.eq(1),  
44            $centaine_3 = this.$centaines.eq(2);  
45        this.countdown_interval = setInterval(function () {  
46            if (112000000 != that.values.millions * 1000 * 1000 +  
47            that.values.milliers * 1000 + that.values.centaines) {  
48                ++that.values.centaines;  
49                if (that.values.centaines == 1000) {  
50                    that.values.centaines = 0;  
51                    ++that.values.milliers;  
52                }  
53                if (that.values.milliers == 1000) {  
54                    that.values.milliers = 0;  
55                    ++that.values.millions;  
56                }  
57                if (that.values.millions == 1000) {  
58                    that.values.millions = 0;  
59                }  
60                // Update DOM values  
61                // Millions  
62                that.checkMillion(that.values.millions, $million_1,  
63                $million_2, $million_3);  
64                // Milliers  
65                that.checkMillion(that.values.milliers, $millier_1,  
66                $millier_2, $millier_3);  
67                // Centaines  
68                that.checkMillion(that.values.centaines, $centaine_1,  
69                $centaine_2, $centaine_3);  
70            }  
71        },  
72    },  
73    checkMillion: function (value, $millier_1, $millier_2, $millier_3) {  
74        if (value < 1000) {  
75            $millier_1.html(value);  
76        } else if (value < 1000000) {  
77            $millier_1.html(Math.floor(value / 1000));  
78            $millier_2.html((value % 1000) / 1000);  
79        } else {  
80            $millier_1.html(Math.floor(value / 1000000));  
81            $millier_2.html((value % 1000000) / 1000);  
82            $millier_3.html((value % 1000000) % 1000 / 1000);  
83        }  
84    }  
85}
```

Rapport de stage

The screenshot shows a code editor interface with a dark theme. On the left is a vertical toolbar with icons for file operations, search, and other developer tools. The main area displays a JavaScript file named 'countdown.js'. The code is a class implementation for a digital clock. It includes methods for clearing intervals, animating figures (top and bottom segments), and checking values for millions, thousands, and hundreds. It uses jQuery and TweenMax for DOM manipulation and animation. The code is well-structured with comments explaining its logic.

```
JS countdown.js (deleted) ● JS main.js JS simple_animations.js JS slider.js ⏪ ▶ ⏴ ...  
dia julien > Julien_GIRAUD > Eye Tech Care > bi_template > Resources > Public > JavaScript > Dist > JS countdown.js >   
66     else {  
67         |   clearInterval(that.countdown_interval);  
68     }  
69     }, 1000);  
70 },  
71 animateFigure: function ($el, value) {  
72     var that = this,  
73         $top = $el.find('.top'),  
74         $bottom = $el.find('.bottom'),  
75         $back_top = $el.find('.top-back'),  
76         $back_bottom = $el.find('.bottom-back');  
77     // Before we begin, change the back value  
78     $back_top.find('span').html(value);  
79     // Also change the back bottom value  
80     $back_bottom.find('span').html(value);  
81     // Then animate  
82     TweenMax.to($top, 0.8, {  
83         rotationX: '-180deg',  
84         transformPerspective: 300,  
85         ease: Quart.easeOut,  
86         onComplete: function () {  
87             $top.html(value);  
88             $bottom.html(value);  
89             TweenMax.set($top, { rotationX: 0 });  
90         }  
91     });  
92     TweenMax.to($back_top, 0.8, {  
93         rotationX: 0,  
94         transformPerspective: 300,  
95         ease: Quart.easeOut,  
96         clearProps: 'all'  
97     });  
98 },  
99 checkMillion: function (value, $el_1, $el_2, $el_3) {  
100    var val_1 = value.toString().charAt(0),  
101        val_2 = value.toString().charAt(1),  
102        val_3 = value.toString().charAt(2),  
103        fig_1_value = $el_1.find('.top').html(),  
104        fig_2_value = $el_2.find('.top').html(),  
105        fig_3_value = $el_3.find('.top').html();  
106    if (value < 10) {  
107        // If we are under 10, replace first and second figure with 0  
108        if (fig_1_value !== '0') this.animateFigure($el_1, 0);  
109        if (fig_2_value !== '0') this.animateFigure($el_2, 0);  
110        if (fig_3_value !== val_1) this.animateFigure($el_3, val_1);  
111    }  
112    else if (value < 100) {  
113        // If we are under 100, replace first figure with 0  
114        if (fig_1_value !== '0') this.animateFigure($el_1, 0);  
115        if (fig_2_value !== val_1) this.animateFigure($el_2, val_1);  
116        if (fig_3_value !== val_2) this.animateFigure($el_3, val_2);  
117    }  
118    else {  
119        // Animate only if the figure has changed  
120        if (fig_1_value !== val_1) this.animateFigure($el_1, val_1);  
121        if (fig_2_value !== val_2) this.animateFigure($el_2, val_2);  
122        if (fig_3_value !== val_3) this.animateFigure($el_3, val_3);  
123    }  
124 };  
125 $(document).ready(function() {  
126     if (Countdown.total_centaines) {  
127         |   Countdown.init();  
128     }  
129 }  
Ln 63, Col 1  Spaces: 4  UTF-8  LF  JavaScript  ☺  🔔
```

Rapport de stage

Annexe 7 : Fonction JS de gestion du slider Eye Tech Care



The screenshot shows a code editor with a dark theme. On the left is a sidebar with various icons for file operations. The main area displays a block of JavaScript code. The code defines a function named `slider_start`. It initializes variables for slides, buttons, mutex click, mutex autoNextSlide, and newId. It then queries the DOM for the slider element and its slides. It creates buttons for the slider and gets elements by class name. The function then handles mouse events on each slide. It uses `clearTimeout`, `setInterval`, and `fadeOut` methods. It also handles the activation of the first slide, deactivation of other slides, and adds click listeners to buttons. It includes functions for switching slides based on newId and lastId, and for getting the last active button's id. Finally, it defines a `newActive` function to handle the transition between slides.

```
function slider_start() {
    var slides, buttons, mutex_click = 0, mutex_autoNextSlide = 0, newId = 0,
        lastId = 0, sens1, sens2, slider;
    slider = document.querySelector(".slider");
    slides = document.getElementsByClassName("slide");
    createButtonsSlider();
    buttons = document.getElementsByTagName("button-slider");
    // Si pas de slides on ferme
    if (slides.length < 1) return;
    // Ajout du listener de la souris
    $(".slide").each(function(){
        $(this).mouseenter(function(){
            clearTimeout($(this).data('timeoutId'));
            mutex_autoNextSlide++;
        }).mouseleave(function(){
            var someElement = $(this),
                timeoutId = setTimeout(function(){
                    | someElement.find(".tooltip").fadeOut("slow");
                }, 650);
            //set the timeoutId, allowing us to clear this trigger if the
            mouse comes back over
            someElement.data('timeoutId', timeoutId);
            switchNextSlide();
        });
    });
    // Activation de la première slide
    newActive(0, 0);
    switchNextSlide();
    // Désactivation des autres slides
    for (var slide of slides) {
        if (slide != slides[0])
            | slide.classList.add("inactive");
    }
    for (var button of buttons) {
        button.addEventListener('click', clickAction, {passive: true});
    }
    function clickAction() {
        switchNextSlide();
        if (mutex_click != 0) {
            | return;
        }
        mutex_click = 1;
        newId = this.getAttribute("data-id");
        lastId = getLastId();
        switchSlide();
    }
    function switchSlide() {
        if (newId == lastId) {
            mutex_click = 0;
            return;
        }
        sens1 = "transform-" + (newId > lastId ? "down" : "up");
        sens2 = "transform-" + (newId < lastId ? "down" : "up");
        transition();
    }
    function getLastId() {
        for (var button of buttons) {
            if (button.classList.contains("active")) {
                | return parseInt(button.getAttribute("data-id"));
            }
        }
    }
    function newActive(before, after) {

```

Rapport de stage

```
61   function newActive(before, after) {
62     slides[before].classList.remove("active");
63     buttons[before].classList.remove("active");
64     slides[after].classList.add("active");
65     buttons[after].classList.add("active");
66   }
67   async function transition() {
68     slides[newId].classList.remove("inactive");
69     slides[newId].classList.add(sens1);
70     await sleep(100);
71     slides[newId].classList.add("slide-transition");
72     slides[lastId].classList.add("slide-transition");
73     newActive(lastId, newId);
74     slides[lastId].classList.add(sens2);
75     slides[newId].classList.remove(sens1);
76     await sleep(1000);
77     slides[newId].classList.remove("slide-transition");
78     slides[lastId].classList.remove("slide-transition");
79     slides[lastId].classList.remove(sens2);
80     slides[lastId].classList.add("inactive");
81     if (mutex_click == 1) {
82       mutex_click = 0;
83     }
84     switchNextSlide();
85   }
86   async function switchNextSlide() {
87     let id = mutex_autoNextSlide + 1;
88     mutex_autoNextSlide++;
89     await sleep(4500);
90     if (mutex_autoNextSlide == id && mutex_click == 0) {
91       mutex_click = 1;
92       lastId = getLastId();
93       newId = (lastId + 1)%slides.length;
94       switchSlide();
95     }
96   }
97   function createButtonsSlider() {
98     var newLi, newSpan, ul;
99     ul = document.querySelector('.slider .buttons-slider');
100    for (i = 0; i < slides.length; i++) {
101      newLi = document.createElement('li');
102      newSpan = document.createElement('span');
103      newLi.appendChild(newSpan);
104      newSpan.setAttribute("data-id", i);
105      newSpan.setAttribute("class", "button-slider");
106      ul.appendChild(newLi);
107    }
108  }
109 }
110 }
111 window.addEventListener("load", slider_start);
112 
```



master C ✘ 0 ▲ 0

Ln 112, Col 1 Spaces: 4 UTF-8 LF JavaScript 😊 📣

Rapport de stage

Annexe 8 : Blocs avancé du back-end d'un DCE

Edit DCE "Section image d'un côté / texte de l'autre côté" on root level

The screenshot displays the configuration interface for a DCE (Dynamic Content Element) with five main sections:

- Niveau du texte principal - {field.texteLevel}**: Shows a configuration panel for an "Element" type with a title "Texte principal". The configuration XML is as follows:

```

1 <config>
2   <type>select</type>
3   <items type="array">
4     <numIndex index="0" type="array">
5       <numIndex index="0">Titre 1</numIndex>
6       <numIndex index="1">h1</numIndex>
7     </numIndex>
8     <numIndex index="1" type="array">
9       <numIndex index="0">Titre 2</numIndex>
10      <numIndex index="1">h2</numIndex>
11    </numIndex>
12    <numIndex index="2" type="array">
13      <numIndex index="0">Titre 3</numIndex>
14      <numIndex index="1">h3</numIndex>
15    </numIndex>
16    <numIndex index="3" type="array">
17      <numIndex index="0">Titre 4</numIndex>
18      <numIndex index="1">h4</numIndex>
19    </numIndex>
20    <numIndex index="4" type="array">
21      <numIndex index="0">Texte</numIndex>
22      <numIndex index="1">p</numIndex>
23    </numIndex>
24  </items>
25  <size>1</size>
26  <minitems>1</minitems>
27  <maxitems>1</maxitems>
28 </config>

```

- Texte principal - {field.texte}**: Shows a configuration panel for an "Element" type with a title "Texte principal". The configuration XML is as follows:

```

1 <config>
2   <type>text</type>
3   <enableRichText>1</enableRichText>
4   <richTextConfiguration>default</richTextConfiguration>
5 </config>
6

```

- Positionnement de l'image - {field.images.n.position}**: Shows a configuration panel for an "Element" type with a title "Positionnement de l'image". The configuration XML is as follows:

```

1 <config>
2   <type>select</type>
3   <items type="array">
4     <numIndex index="0" type="array">
5       <numIndex index="0">Éléments importants au centre</numIndex>
6       <numIndex index="1">obj-pos-center</numIndex>
7     </numIndex>
8     <numIndex index="1" type="array">
9       <numIndex index="0">Éléments importants à gauche</numIndex>
10      <numIndex index="1">obj-pos-left</numIndex>
11    </numIndex>
12    <numIndex index="2" type="array">
13      <numIndex index="0">Éléments importants à droite</numIndex>
14      <numIndex index="1">obj-pos-right</numIndex>
15    </numIndex>
16  </items>
17  <size>1</size>
18  <minitems>1</minitems>
19  <maxitems>1</maxitems>
20 </config>
21

```

- Ordre des blocs - {field.ordre}**: Shows a configuration panel for an "Element" type with a title "Ordre des blocs". The configuration XML is as follows:

```

1 <config>
2   <type>select</type>
3   <items type="array">
4     <numIndex index="0" type="array">
5       <numIndex index="0">Texte à gauche, image(s) à droite</numIndex>
6       <numIndex index="1">0</numIndex>
7     </numIndex>
8     <numIndex index="1" type="array">
9       <numIndex index="0">Texte à droite, image(s) à gauche</numIndex>
10      <numIndex index="1">1</numIndex>
11    </numIndex>
12  </items>
13  <size>1</size>
14  <minitems>1</minitems>
15  <maxitems>1</maxitems>
16 </config>

```

- Mode image - {field.mode}**: Shows a configuration panel for an "Element" type with a title "Mode image". The configuration XML is as follows:

```

1 <config>
2   <type>select</type>
3   <items type="array">
4     <numIndex index="0" type="array">
5       <numIndex index="0">Mode 1 : image simple</numIndex>
6       <numIndex index="1">0</numIndex>
7     </numIndex>
8     <numIndex index="1" type="array">
9       <numIndex index="0">Mode 2 : galerie de 6 images</numIndex>
10      <numIndex index="1">1</numIndex>
11    </numIndex>
12  </items>
13  <size>1</size>
14  <minitems>1</minitems>
15  <maxitems>1</maxitems>
16 </config>

```

Rapport de stage

Annexe 9 : Template HTML / TypoScript d'un DCE avancé

```
1  <f:if condition="{field.page} == 0"><div class="bloc-content bg-white one-page bloc-content-txt bloc-content-txt-wide"></f:if>
2  <div class="bloc-content {field.options}">
3      <f:if condition="{field.ordre} == 1">
4          <f:if condition="{field.mode} == 0">
5              <f:then>
6                  <div class="bloc-content-img {field.cssBlocImage}">
7                      <f:if condition="{field.image.0}"><f:image class="full-img {field.cssImage}" image="{field.image.0}" /></f:if>
8                  </div>
9              </f:then>
10             <f:else>
11                 <div class="bloc-content-visu flex-x-center flex-y-center <f:if condition="{field.animationGalerie} == 1" grid></f:if>">
12                     <f:if condition="{field.galerieLien}">
13                         <f:link.typolink parameter="{field.galerieLien}" target="_blank" class="flex-x-center flex-y-center">
14                             <f:then>
15                                 <f:for each="{field.images}" as="image">
16                                     <div class="grid-item <f:if condition="{field.animationGalerie} == 1" left></f:if>">
17                                         <f:image class="image.positionnement" image="{image.image.0}" />
18                                     </div>
19                                 </f:for>
20                                 <f:if condition="{field.centre}">
21                                     <div class="fleche-container"><f:image image="{field.centre.0}" class="fleche" /></div>
22                                 </f:if>
23                             </f:then>
24                             <f:else>
25                                 <f:for each="{field.images}" as="image">
26                                     <div class="grid-item <f:if condition="{field.animationGalerie} == 1" left></f:if>">
27                                         <f:image class="image.positionnement" image="{image.image.0}" />
28                                     </div>
29                                 </f:for>
30                                 <f:if condition="{field.centre}">
31                                     <div class="fleche-container"><f:image image="{field.centre.0}" class="fleche" /></div>
32                                 </f:if>
33                         </f:link.typolink></f:if></div></f:else></f:if></f:if>
34                         <div class="bloc-content-txt flex-x-left flex-y-center">
35                             <f:if condition="{field.animation} != 0">scroll-animated</f:if>
36                             <f:if condition="{field.animation} == 1 && {field.ordre} == 0">float-left</f:if>
37                             <f:if condition="{field.animation} == 1 && {field.ordre} == 1">float-right</f:if>
38                             <f:if condition="{field.cssBlocTexte}">{field.cssBlocTexte}</f:if>
39                         ">
40                         <f:if condition="{field.titre}">
41                             <{field.titleLevel} class="field.cssTitre">
42                                 <f:format.stripTags allowedTags="<a><strong><em><br>">
43                                     <f:format.nl2br><f:format.html>{field.titre}</f:format.html></f:format.nl2br>
44                                 </f:format.stripTags>
45                             </{field.titleLevel}>
46                         </f:if>
47                         <f:if condition="{field.texte}">
48                             <{field.texteLevel} class="field.cssTexte">
49                                 <f:format.stripTags allowedTags="<a><strong><em><br>">
50                                     <f:format.nl2br><f:format.html>{field.texte}</f:format.html></f:format.nl2br>
51                                 </f:format.stripTags>
52                             </{field.texteLevel}>
53                         </f:if>
54                         <f:if condition="{field.lien}">
55                             <f:then>
56                                 <f:link.typolink parameter="{field.lien}">
57                                     <f:if condition="{field.texteBouton}">
58                                         <button class="bloc-content-lire">{field.texteBouton}</button>
59                                     </f:if>
60                                 </f:link.typolink>
61                             </f:then>
62                             <f:else>
63                                 <f:if condition="{field.texteBouton}">
64                                     <button class="bloc-content-lire">{field.texteBouton}</button>
65                                 </f:if>
66                             </f:else>
67                         </f:if>
68                     </div>
69                     <f:if condition="{field.ordre} == 0">
70                         <f:if condition="{field.mode} == 0">
71                             <f:then>
72                                 <div class="bloc-content-img {field.cssBlocImage}">
73                                     <f:if condition="{field.image.0}"><f:image class="full-img {field.cssImage}" image="{field.image.0}" /></f:if>
74                                 </div>
75                         </f:then><f:else>
76                             <div class="bloc-content-visu flex-x-center flex-y-center <f:if condition="{field.animationGalerie} == 1" grid></f:if>">
77                                 <f:if condition="{field.galerieLien}">
78                                     <f:then>
79                                         <f:link.typolink parameter="{field.galerieLien}" target="_blank" class="flex-x-center flex-y-center">
80                                             <f:for each="{field.images}" as="image">
81                                                 <div class="grid-item <f:if condition="{field.animationGalerie} == 1" right></f:if>">
82                                                     <f:image class="image.positionnement" image="{image.image.0}" />
83                                                 </div>
84                                             </f:for>
85                                             <f:if condition="{field.centre}">
86                                                 <div class="fleche-container"><f:image image="{field.centre.0}" class="fleche" /></div>
87                                             </f:if>
88                                         </f:link.typolink>
89                                     </f:then><f:else>
90                                         <f:for each="{field.images}" as="image">
91                                             <div class="grid-item <f:if condition="{field.animationGalerie} == 1" right></f:if>">
92                                                 <f:image class="image.positionnement" image="{image.image.0}" />
93                                             </div>
94                                         </f:for>
95                                         <f:if condition="{field.centre}">
96                                             <div class="fleche-container"><f:image image="{field.centre.0}" class="fleche" /></div>
97                                         </f:if>
98                                     </f:else></f:if></div></f:else><f:if></f:if>
99                                     <f:if condition="{field.legend}"><div class="legend"><f:format.nl2br>{field.legend}</f:format.nl2br></div></f:if>
100                                </div>
101                            <f:if condition="{field.page} == 0"></div></f:if>
```

Rapport de stage

Annexe 10 : Header-Footer HTML / TypoScript de Eye Tech Care

```
④ Default.html ▶ ❸ header ▶ ❸ nav.navbar.navbar-default.bootsnav.navbar-top.header-light.background-white.navbar-expand-lg ▶ ❸ div.container.nav-header-container ▶ ❸ div.col-auto.j
1  <header>
2    <nav class="navbar navbar-default bootsnav navbar-top header-light background-white navbar-expand-lg">
3      <div class="container nav-header-container">
4        <!-- start logo -->
5        <div class="col-auto pl-lg-0"><f:link.page pageUid="67" class="logo-haut"></f:link.page></div>
6        <!-- end logo -->
7        <div class="col accordion-menu pr-0 pr-md-3">
8          <!-- start burger -->
9          <button type="button" class="navbar-toggler collapsed" data-toggle="collapse" data-target="#navbar-collapse-toggle-1">
10            <span class="sr-only">toggle navigation</span>
11            <span class="icon-bar"></span>
12            <span class="icon-bar"></span>
13            <span class="icon-bar"></span>
14          </button>
15          <!-- end burger -->
16          <div class="navbar-collapse collapse justify-content-end" id="navbar-collapse-toggle-1">
17            <!-- start menu item -->
18            <f:for each="{mainnavigation}" as="mainnavigationItem" iteration="infos">
19              <li class="{f:if(condition: mainnavigationItem.active, then:'')} {f:if(condition: mainnavigationItem.children, then:'dropdown simple-dropdown')}">
20                <a href="{mainnavigationItem.link}" target="{mainnavigationItem.target}" title="{mainnavigationItem.title}">{mainnavigationItem.title}</a>
21                <f:if condition="{mainnavigationItem.children}">
22                  <i class="fas fa-angle-down darkgrey dropdown-toggle" data-toggle="dropdown" aria-hidden="true"></i>
23                  <ul class="dropdown-menu bg-white" role="menu">
24                    <!-- start submenu item -->
25                    <f:for each="{mainnavigationItem.children}" as="child">
26                      <li class="{f:if(condition: child.active, then:'active')}"><a href="{child.link}" target="{child.target}" title="{child.title}">{child.title}</a></li>
27                    </f:for>
28                    <!-- end submenu item -->
29                  </ul></f:if></li>
30                  <!-- end menu item -->
31                <f:if condition="{infos.isLast}">
32                  <li class="dropdown simple-dropdown lang">
33                    <a href="#"><f:spaceless>
34                      <f:if condition="{pageLanguage} == 0">Eng</f:if>
35                      <f:if condition="{pageLanguage} == 2">Fra</f:if>
36                    </f:spaceless></a>
37                    <i class="fas fa-angle-down darkgrey dropdown-toggle" data-toggle="dropdown" aria-hidden="true"></i>
38                    <ul class="dropdown-menu bg-white" role="menu">
39                      <li><f:link.page additionalParams="{L: '0'}">English</f:link.page></li>
40                      <li><f:link.page additionalParams="{L: '2'}">Français</f:link.page></li>
41                    </ul>
42                  </li>
43                </f:if></f:for></ul></div></div></div></nav></header>
44
45 <div class="breadcrumb">
46   <!--<f:format.html>{BREADCRUMB}</f:format.html>-->
47 </div>
48
49 <f:render section="Main" />
50
51 <a class="scroll-top-arrow" href="javascript:void(0);"><i class="ti-arrow-up"></i></a>
52
53 <footer class="flex-x-center">
54   <div class="footer-col footer-col1">
55     <f:link.page pageUid="67" class="logo-bas"></f:link.page>
56   </div>
57   <f:for each="{menuBas}" as="a">
58     <div class="footer-col">
59       <div class="h4 text-center">{a.title}</div>
60       <f:for each="{a.children}" as="b">
61         <f:link.page pageUid="{b.data.uid}" class="footer-pages">{b.title}</f:link.page>
62       </f:for>
63     </div>
64   </f:for>
65   <div class="footer-col footer-col6">
66     <p class="footer-p">+33 (0)4 78 88 09 00</p>
67     <p>
68       <f:link.email email="contact@eyetechcare.com" />
69     </p>
70     <p>
71       2871, Av. de l'Europe - F
72       <br>69140 Rillieux-la-Pape
73       <br>France
74     </p>
75     <a href="https://www.facebook.com/pages/EYE-TECH-CARE/850565101621466" class="socials facebook" target="_blank"></a>
76     <a href="https://fr.linkedin.com/company/eyetechcare" class="socials linkedin" target="_blank"></a>
77     <a href="https://twitter.com/EyetechCare?lang=fr" class="socials twitter" target="_blank"></a>
78   </div>
79 </footer>
```