

Formation Développeur Web et Web Mobile

Du 15 mars 2021 au 10 février 2022 à Sarreguemines

DOSSIER DE PROJET

Réalisation d'un site web pour le

Cabinet Médical du Docteur OLONA

Situé 1, rue Marcel Pierron à SARREGUEMINES



Dossier de :

KRAETHER François

Formation Développeur Web et Web Mobile

Année 2021 / 2022

Maître de stage :

Docteur Michel OLONA

Médecin généraliste

57200 SARREGUEMINES

Tuteur de stage :

BENAICHA Jérémie

Formateur de front-end

C.C.I. Moselle

Remerciements :

Dans un premier temps, je tenais à remercier Stanayka et Laurent de la C.C.I. Metz qui ont retenu ma candidature pour pouvoir intégrer cette formation, ainsi que leur suivi tout au long de sa durée.

Je remercie également le Docteur Michel OLONA, qui m'a fait confiance en acceptant que je réalise le site web de son cabinet médical à l'occasion de mon stage en entreprise.

Médecin généraliste à Sarreguemines, le Docteur OLONA est aussi un passionné de musique et d'informatique. Il a d'ailleurs réalisé lui-même un site web consacré à son groupe musical, à partir du CMS Joomla : <https://www.arsenger.fr/>

Il m'a également sollicité, une fois ma formation terminée, pour adapter son site aux différentes tailles d'écran, plus précisément pour les tablettes et téléphones portables, son site n'étant actuellement pas responsive et ciblant essentiellement des personnes plutôt jeunes, donc à l'aise avec l'utilisation des smartphones sur internet.

Je remercie également l'ensemble de mes formateurs, grâce auxquels j'ai énormément appris durant cette formation. J'ai également apprécié leur disponibilité, leurs conseils et cours tout au long de cette formation. Une petite mention particulière pour mon formateur de front-end, Jérémie, pour sa pédagogie et pour avoir su adapter ses cours en distanciel ; le fait de donner des cours à l'ensemble du groupe dans un salon sur discord, suivi d'un exercice et de s'isoler dans le salon « aide » dans lequel on pouvait aller le trouver en cas de problème est, pour moi, le seul point positif à la pratique du distanciel, chose qui n'aurait même pas été possible en présentiel. J'ai franchement plus qu'apprécié ses cours, qu'ils soient en distanciel ou présentiel.

Table des matières

1 - Liste des compétences du référentiel qui sont couvertes par le projet	1
2 - Résumé du projet.....	4
3 - Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet ..	7
3.1 Présentation d'ensemble du projet	7
3.1.1 <i>Présentation de l'entreprise</i>	7
3.1.2 <i>Les objectifs du site</i>	8
3.1.3 <i>La cible adressée par le site</i>	8
3.1.4 <i>Objectifs quantitatifs</i>	8
3.1.5 <i>Périmètre du projet</i>	9
3.1.6 <i>Description de l'existant</i>	9
3.2 Description graphique et ergonomique	9
3.2.1 <i>Charte graphique</i>	9
3.2.2 <i>Design</i>	10
3.2.3 <i>Maquettes</i>	10
3.3 Description fonctionnelle et technique	11
3.3.1 <i>Arborescence du site</i>	11
3.3.2 <i>Description fonctionnelle</i>	12
3.3.3 <i>Informations relatives aux contenus</i>	13
3.3.4 <i>Contraintes techniques</i>	13
3.4 Prestations attendues et modalités de sélection des prestataires	13
3.4.1 <i>Planning</i>	13
3.4.2 <i>Méthodologie de suivi</i>	13
4 - Spécifications techniques du projet, y compris pour la sécurité et le web mobile	14
5 - Réalisations comportant les extraits de code les plus significatifs et en les argumentant, y compris pour la sécurité et le web mobile	15
5.1 Maquettage du site	15
5.1.1 <i>La page ACCUEIL</i>	15
5.1.2 <i>La page A PROPOS</i>	16
5.1.3 <i>La page CABINET MEDICAL</i>	16
5.1.4 <i>La page PATIENTS INFOS</i>	16
5.1.5 <i>La page ARTICLES</i>	17
5.1.6 <i>La page CONTACT</i>	17
5.1.7 <i>Les pages ARTICLES SANTE</i>	17

5.1.8 Modifications de la maquette en cours de projet	18
5.2 Réalisation du site web	19
5.2.1 La page d'accueil	19
5.2.2 Page à-propos	28
5.2.3 Page Cabinet médical	28
5.2.4 Page informations	30
5.2.5 Page articles santé (pdf)	33
5.2.6 Page ARTICLES SANTE	36
5.2.7 Page ARTICLES SANTE détaillés	36
5.2.8 Page CONTACT	37
5.3 Partie back-office	38
5.3.1 La page inscription	38
5.3.2 La page connexion	40
5.3.3 Page d'accueil (gestion des messages)	42
5.3.4 Page modifier_info.php	44
5.3.5 Fichier supprimer.php	44
5.3.6 Page d'accueil (gestion des fichiers pdf)	45
5.3.7 Page d'accueil (gestion des catégories)	48
6 - Présentation du jeu d'essai de la fonctionnalité la plus représentative (données en entrée, données attendues, données obtenues)	50
7 - Description de la veille, durant le projet, sur les vulnérabilités de sécurité	51
7.1 password_hash()	51
7.2 htmlspecialchars	51
7.3 Requêtes préparées	51
8 - Description d'une situation de travail ayant nécessité une recherche, à partir de site anglophone	52
9 - Extrait du site anglophone, utilisé dans le cadre de la recherche décrite précédemment, accompagné de la traduction en français	52

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

1.1 Maquetter une application :

Cette opération intervient en phase de conception avec une forte implication de l'utilisateur finale et dans une démarche projet. J'ai utilisé Adobe XD pour réaliser le mappage, le zoning et le maquetage.

1.2 Réaliser une interface utilisateur web statique et adaptable :

À partir de la maquette, j'ai réalisé une intégration HTML/CSS adaptée aux différents dispositifs en respectant les bonnes pratiques SEO. Pour cette étape, j'ai utilisé Visual Studio Code pour créer mes fichiers HTML et CSS.

1.3 Développer une interface utilisateur web dynamique :

À partir du cahier des charges et d'une conception fonctionnelle, j'ai intégré à mon site une API Google Maps afin de géolocaliser le cabinet médical.

1.4 Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce :

J'ai intégré une solution de gestion de contenu en utilisant un carrousel bootstrap que j'ai adapté à mon site, en y apportant les photos appropriées, ainsi qu'en modifiant leur taille pour pouvoir les adapter aux différentes tailles d'écran

1.5 Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité :

J'ai développé un moteur de recherche, avec du code PHP et des requêtes MySQL, consacré à mes fiches santé PDF, qui, à partir d'un mot se trouvant dans le titre de la fiche PDF, permet de trouver et d'afficher cette dernière.

1.6 Créer une base de données :

A partir du cahier des charges, j'ai conçu une base de données dans phpMyAdmin qui permet le stockage de mes données ainsi que leur organisation.

1.7 Développer les composants d'accès aux données :

Avec des requêtes MySQL et des codes PHP, l'utilisateur a la possibilité, selon sa demande, d'afficher les données se trouvant dans une table de la base de données.

1.8 Développer la partie back-end d'une application web ou web mobile :

J'ai également développé une partie back-office permettant à l'administrateur du site de pouvoir organiser ces données ; il peut créer, afficher, modifier ou supprimer ces données (CRUD)

2 - Résumé du projet

Le projet consiste à créer un site vitrine pour le cabinet médical du Docteur OLONA, le rendant ainsi visible sur internet.

Le site contiendra essentiellement des informations destinées à renseigner ses patients, comme les actes pratiqués à son cabinet, les dates de congés, l'emplacement du cabinet, etc...

A ce jour, les informations concernant le cabinet médical se trouvent essentiellement sur les plateformes doctolib.fr et docvitae.fr.

On peut y trouver essentiellement des informations sur le cabinet médical. Je me suis d'ailleurs appuyé sur ces deux plateformes pour réaliser et renseigner le site web avec les informations existantes.

Le site contiendra également une bibliothèque regroupant plus de 600 fiches santé au format PDF, que le Docteur OLONA a collecté au fil du temps, et qu'il aimerait afficher sur le futur site.

Le site aura aussi une partie back-office, permettant ainsi au médecin de pouvoir gérer avec facilité et en totale autonomie les informations les plus importantes.

Ces informations seront affichées en haut de la page d'accueil afin qu'elles soient visibles directement à l'arrivée sur le site, telles que la mise à jour de ses congés passés et à venir, etc.

Il pourra également à partir de cette interface gérer les fiches santé citées précédemment.

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

3.1 Présentation d'ensemble du projet

3.1.1 Présentation de l'entreprise :

Entreprise : cabinet médical du Dr Michel OLONA, médecin généraliste. L'entreprise est située au 1, rue Marcel Pierron à Sarreguemines (57200).

Après une recherche Google afin de pouvoir identifier la concurrence, la plupart des médecins généralistes utilisent la plateforme [doctolib.fr](https://www.doctolib.fr).

Seul le Docteur Léraud Dominique se différencie des autres en proposant un site conçu avec Google My Business

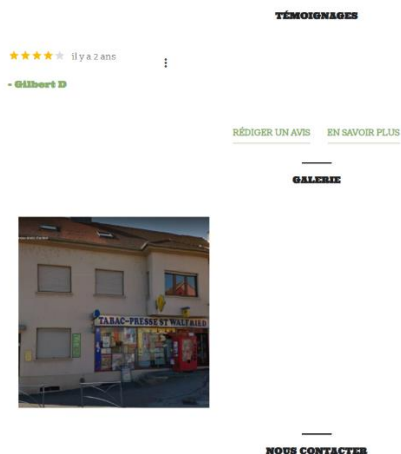
Léraud Dominique
Médecin à Sarreguemines
Ouvert aujourd'hui jusqu'à 12:00

[OBTENIR UN PROFS](#)

Coordonnées
[APPELER MAINTENANT](#)
03 87 95 64 29

Adresse
[ITINÉRAIRE](#)
1 Rue d'Ippling Welfering
57200 Sarreguemines
France

Horaires d'ouverture
lun.: 09:00 - 12:00
mar.: 14:30 - 18:00
mer.: Fermé
jeu.: 09:00 - 11:00, 14:30 - 18:00
ven.: Fermé
sam.: Fermé
dim.: Fermé



× Léraud Dominique

Témoignages

Galerie

Coordonnées

Le site est assez simple et peu développé ; il contient le minimum d'informations comme l'adresse, les horaires, un lien renvoyant sur les avis google, un lien « contact » permettant d'appeler directement le cabinet, et un lien devis... peut-être inutile concernant un médecin généraliste.

Le menu renvoie sur les éléments déjà affichés sur la page d'accueil et le bloc des témoignages bug.

Le rôle de ce projet web est de créer un site pour le cabinet médical, qui a pour principal but de se différencier de la concurrence, qui utilise principalement les plateformes [doctolib.fr](https://www.doctolib.fr) et [docvitae.fr](https://www.docvitae.fr), ce dernier étant un outil de création de sites internet destiné aux cabinets médicaux.

Il s'agit d'une refonte d'un site internet existant, <https://docteur-olona.docvitae.fr/> ; n'ayant pour le moment pas accès à l'administration du site, je ne connais ni son trafic mensuel ou autres statistiques de performance ; le site est bien alimenté en informations qui me seront utiles par la suite.

3.1.2 Les objectifs du site :

Il s'agira d'un site vitrine pour compléter l'activité du cabinet médical, l'objectif étant de donner un maximum d'information aux patients, comme les horaires de consultations, l'emplacement, les dates de congés, informations sur les actes pratiqués au cabinet, etc...

3.1.3 La cible adressée par le site :

La principale cible est la patientèle existante, le docteur Olona ne souhaitant plus avoir de nouveaux patients.

La patientèle existante pourra être informée directement sur le site, de l'actualité du cabinet, comme les jours de congés, etc...

3.1.4 Objectifs quantitatifs :

Etant donné le type d'activité, le trafic visé sera principalement local, optimisé par un référencement sur la section carte des moteurs de recherche.

3.1.5 Périmètre du projet :

Le français sera la langue utilisée.

Le site sera également adapté pour les supports mobiles et tablettes.

3.1.6 Description de l'existant :

Il existe une précédente version du site en ligne : docteur-olona.docvitae.fr, Docvitae étant une solution digitale dédiée aux professionnels de santé afin de leurs permettre de créer automatiquement un site internet adapté à leur pratique médicale.

Il serait préférable d'utiliser un nom de domaine plus ciblé comme docteur-olona.fr, par exemple. Ce nom de domaine est disponible à ce jour.

Le docteur Olona possède néanmoins un autre site concernant son activité musicale, qui pourrait éventuellement héberger le site web ; le cas échéant, je possède un nom de domaine qui pourra héberger le futur site web.

Les principales ressources disponibles sur le site existant sont des contenus texte ; aucun logo et éléments graphiques ne sont présents.

3.2 Description graphique et ergonomique

3.2.1 Charte graphique :

Les couleurs dominantes du site seront des couleurs bleues, car elles se prêtent très bien à l'activité pratiquée :

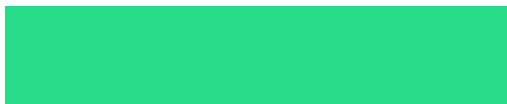
- La couleur `rgb(69, 167, 231)`



- La couleur `#064063`.



- La couleur `#27db88` sera utilisée légèrement pour les bordures des blocs.



Il n'y aura pas de variations du code couleur selon les sections ou les pages du site. Cependant, il serait intéressant qu'il y ait une déclinaison du logo en fonction du code couleur des pages.

Les polices d'écritures utilisées seront des polices simples, d'aspect sérieux et facilement lisibles.

- La police Roboto

Almost before we knew it

- La police Montserrat

Almost before we knew it

3.2.2 Design :

Un design sans exigence particulière sera privilégié, étant donné l'activité pratiquée.

Cependant, un design de type « flat design » pourra être envisagé pour la version mobile du site.

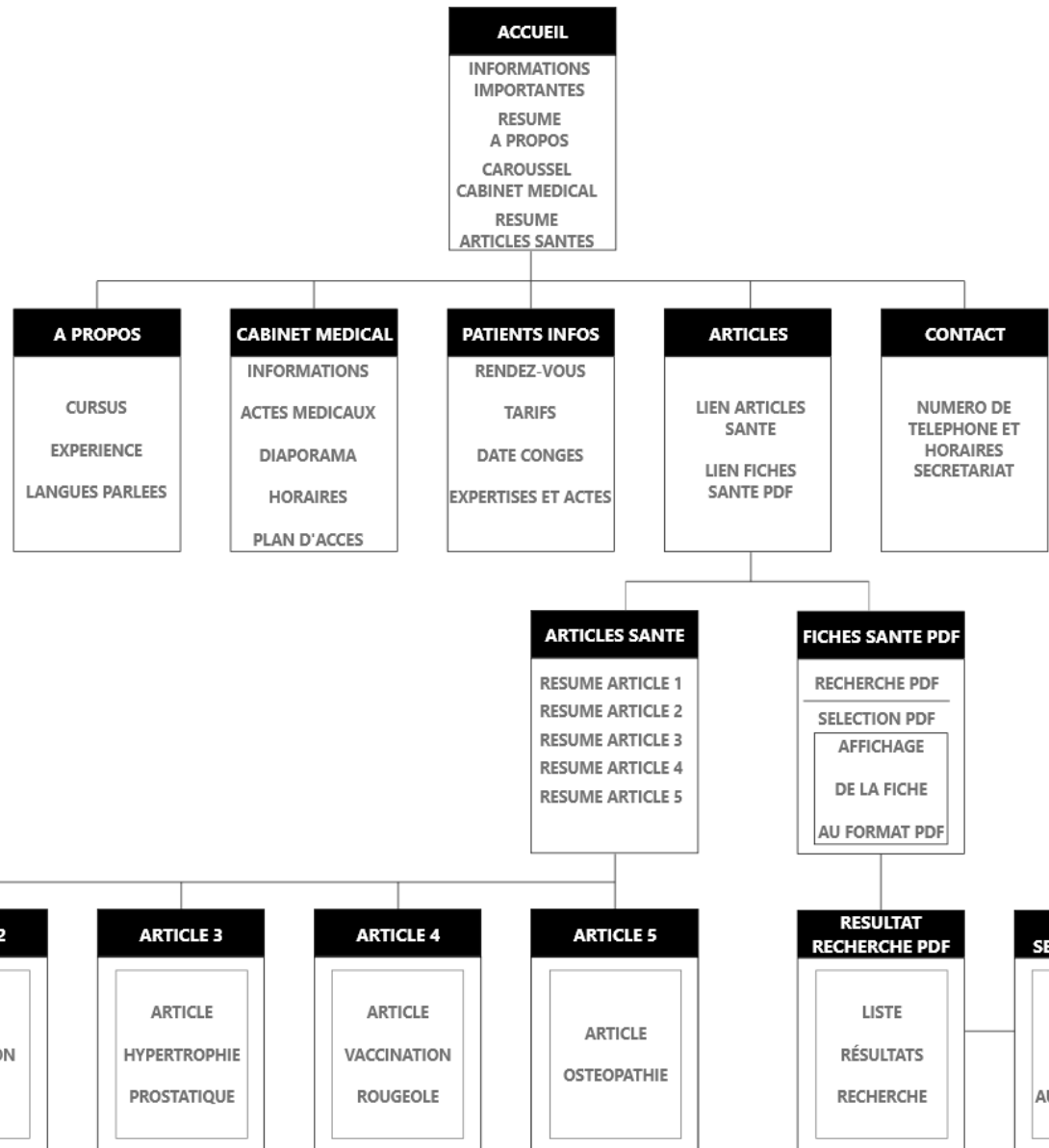
3.2.3 Maquettes :

Un maquettage complet du site sera réalisé. Une fois terminé, il sera présenté au docteur Olona. Le codage du site commencera après approbation du projet maqueté.

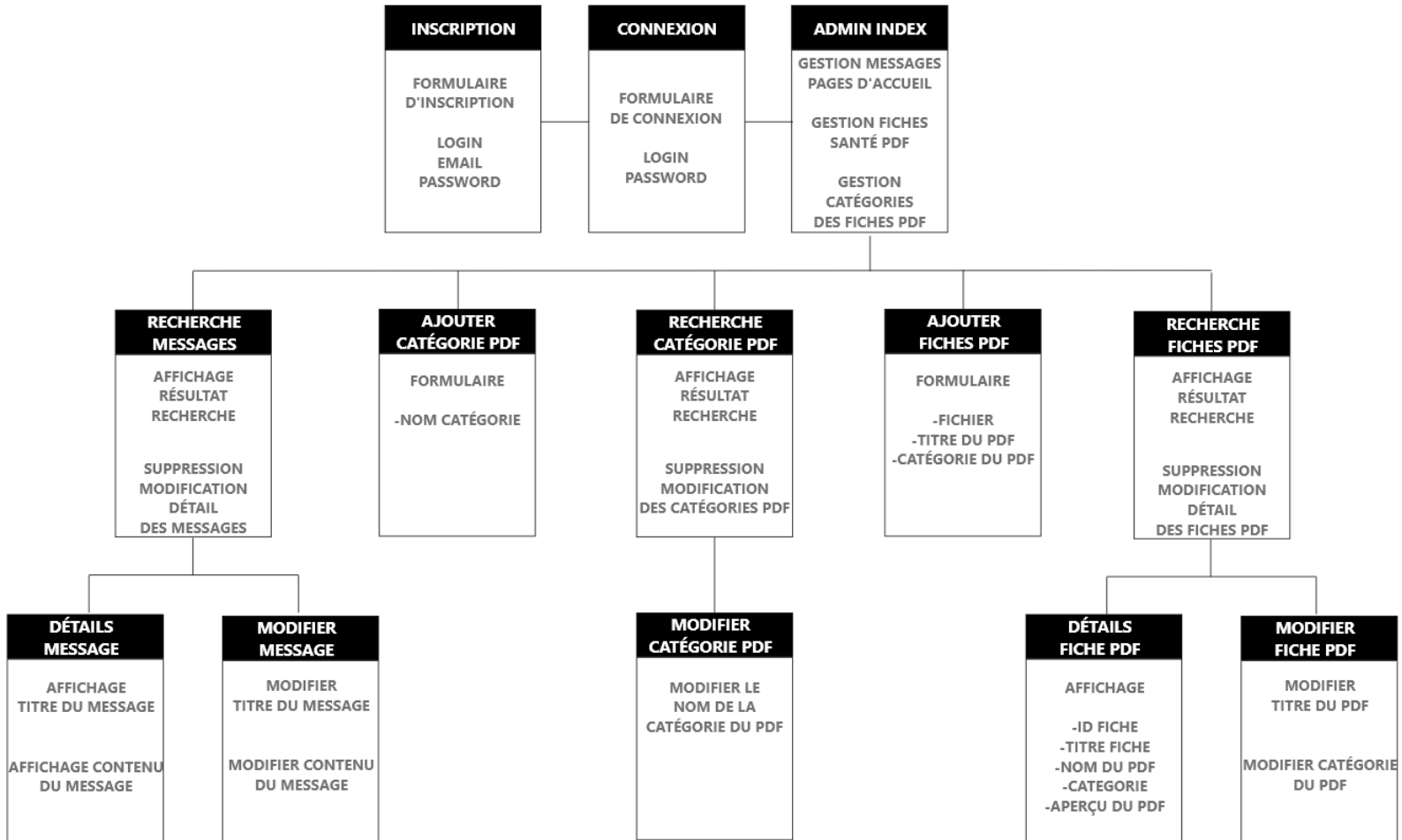
3.3 Description fonctionnelle et technique

3.3.1 Arborescence du site :

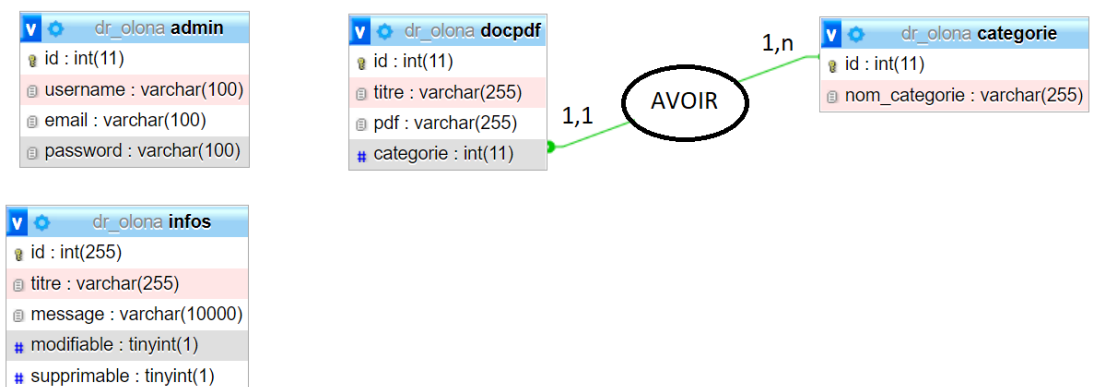
Partie site web



Partie Administration



3.3.2 Description fonctionnelle :



Le site aura une interface administration sécurisée par mot de passe, intégrera un moteur de recherche interne et un module de gestion des principales informations ainsi que des fiches santé au format PDF.

3.3.3 Informations relatives aux contenus :

Les différents types de contenus qui seront présents sur le site seront essentiellement des blocs de type articles, pouvant contenir des images, vidéos ou ressources téléchargeables (comme les fiches santé au format PDF).

Un back-office sera mis en place, afin que le docteur Olona puisse modifier, créer ou supprimer des articles.

Le code HTML respectera les standards SEO notamment en matière de balises titres et de méta-informations.

3.3.4 Contraintes techniques :

Le site ne sera pas créé à partir d'un CMS.

Les technologies utilisées seront le HTML/CSS, Javascript, Bootstrap, PHP et MySQL.

Le site devra être compatible avec les principaux navigateurs :

- Chrome,
- Safari,
- Mozilla Firefox,
- Edge.

3.4 Prestations attendues et modalités de sélection des prestataires

3.4.1 Planning :

La période de stage étant de 7 semaines, il faudra terminer au plus tard les parties cahier des charges et maquettage pour la fin de la première semaine.

Ensuite, le projet sera montré au docteur Olona pour approbation et le codage du site pourra commencer.

La livraison finale du projet aura lieu au plus tard courant semaine 42 de 2021.

3.4.2 Méthodologie de suivi :

Le suivi du projet peut être organisé soit par échanges téléphoniques, soit par email ou soit par présentation physique régulière des avancées... à la convenance du docteur Olona.

4 - Spécifications techniques du projet, y compris pour la sécurité et le web mobile

Lors de la création de mon projet, j'ai eu l'occasion de développer les compétences du référentiel suivantes :

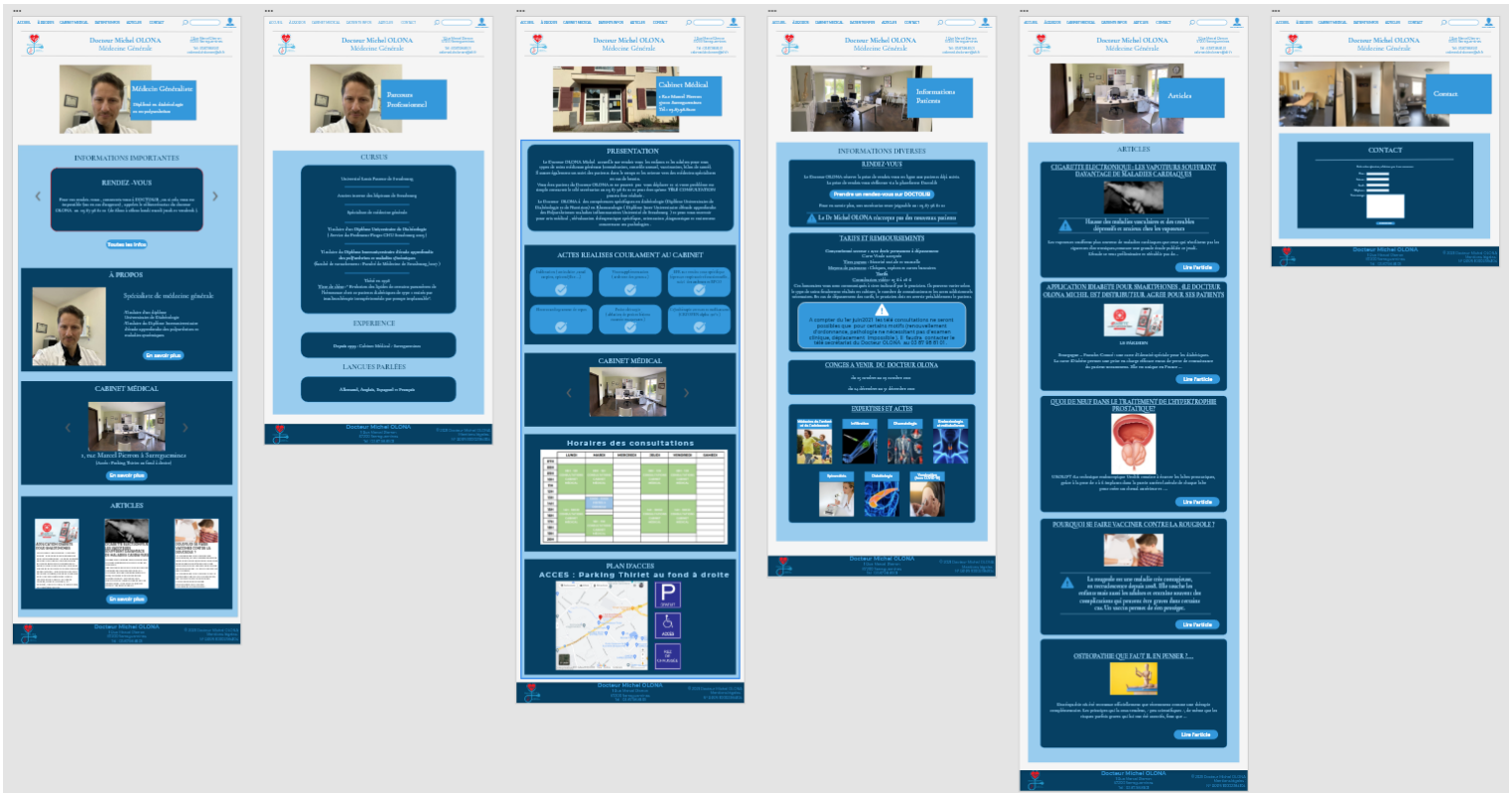
- J'ai déjà effectué un maquetage de l'application avec le logiciel Adobe XD (correspondant à la compétence 1 de l'activité type 1)
- J'ai adapté la totalité des pages du site aux différentes tailles d'écrans en utilisant Bootstrap (correspondant à la compétence 2 de l'activité type 1)
- J'ai intégré plusieurs carrousels sur la page d'accueil, en utilisant Bootstrap (correspondant à la compétence 3 de l'activité type 1)
- J'ai créé des vignettes à l'aide des cards bootstrap pour illustrer les actes médicaux pratiqués au cabinet (correspondant à la compétence 4 de l'activité type 1)
- J'ai créé une base de données (correspondant à la compétence 1 de l'activité type 2)
- J'ai créé un fichier PDO permettant la connexion à ma base de données (correspondant à la compétence 2 de l'activité type 2)
- J'affiche des données récupérées dans ma base de données, essentiellement sur la page « détail » de mes articles santé PDF (correspondant à la compétence 3 de l'activité type 2)
- J'ai créé une partie back office, permettant la gestion des messages sur la page d'accueil ainsi que la gestion des fiches santé PDF (correspondant à la compétence 3 de l'activité type 2)

5 - Réalisations comportant les extraits de code les plus significatifs et en les argumentant, y compris pour la sécurité et le web mobile

5.1 Maquettage du site

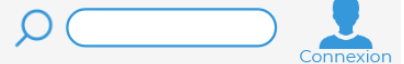
J'ai réalisé le maquettage du site avec le logiciel Adobe XD, à partir du cahier des charges, et des données existantes que j'ai pu trouver sur les pages du médecin de doctolib.fr et de Docvita.fr.

Voici une vue d'ensemble de mon maquettage



Ces pages correspondent au menu qui se trouve dans mon header.

ACCUEIL À PROPOS CABINET MEDICAL PATIENTS INFOS ARTICLES CONTACT



5.1.1 La page ACCUEIL :

Cette page sera un condensé du site. On y trouvera quatre blocs alignés verticalement :

- Un premier bloc intitulé « **Informations Importantes** » de type carousel bootstrap, dans lequel on pourra trouver l'actualité du cabinet médical. Les informations les plus importantes que j'ai pu trouver sur les données existantes sont : les modalités de prise de rendez-vous, les dates de congés à venir, et l'emplacement du cabinet médical. Un bouton contenant un lien renverra l'utilisateur sur la page **PATIENTS INFOS**.

- Le second bloc intitulé « **A propos** » concernera le parcours professionnel du dr OLONA, dans lequel on pourra trouver un petit résumé de son parcours.
Un bouton contenant un lien renverra l'utilisateur sur la page **A PROPOS**.
- Le troisième bloc intitulé « **Cabinet Médical** » contiendra un carrousel avec des photos du cabinet, ainsi que l'adresse.
Un bouton contenant un lien renverra l'utilisateur sur la page **CABINET MEDICAL**.
- Le quatrième bloc intitulé « **Articles** » contiendra un résumé de trois articles santé sur les cinq réalisés par le Docteur Olona.
Un bouton contenant un lien renverra l'utilisateur sur la page **ARTICLES**.

5.1.2 La page A PROPOS :

Cette page sera consacrée au parcours professionnel du Docteur Olona.

On y trouvera trois blocs alignés verticalement :

- Le cursus
- L'expérience professionnelle
- Les langues parlées.

5.1.3 La page CABINET MEDICAL :

Cette page contiendra cinq blocs alignés verticalement contenant des informations sur le cabinet médical :

- Le bloc « **Informations** » (type de soins, téléconsultation, etc.)
- Le bloc « **Actes réalisés couramment au cabinet médical** » dans lequel on trouvera des cards bootstrap représentant les actes médicaux (infiltration, viscosupplémentation, électrocardiogramme de repos, etc.)
- Un bloc « **Diaporama** » contenant des photos du cabinet médical, réalisé avec un carrousel bootstrap.
- Un bloc « **Horaires des consultations** » : un tableau indiquera les horaires des consultations au cabinet médical ainsi que les horaires des visites à domicile.
- Un bloc « **Plan d'accès** » contenant une interface de programmation d'applications Google Map (Application Programming Interface) pour géolocaliser l'emplacement du cabinet médical, ainsi que quelques informations (parking gratuit, etc.).

5.1.4 La page PATIENTS INFOS :

Quatre blocs alignés verticalement constitueront cette page destinée à informer la patientèle du Docteur Olona :

1. Le bloc « **Rendez-vous** », avec des informations concernant les modalités de prise de rendez-vous, ainsi qu'un bouton avec un lien renvoyant l'utilisateur sur la page Doctolib du Docteur Olona.

2. Le bloc « **Tarifs et consultations** » contenant des informations comme le tiers payant, tarifs, moyens de paiement, etc.
3. Le bloc « **Congés à venir** » signalant les absences du Docteur Olona pour l'année en cours.
4. Le bloc « **Expertises et actes** » informant la patientèle de ce qui est pratiqué au cabinet (spirométrie, diabétologie, vaccination, etc.), représentés par des cards bootstrap.

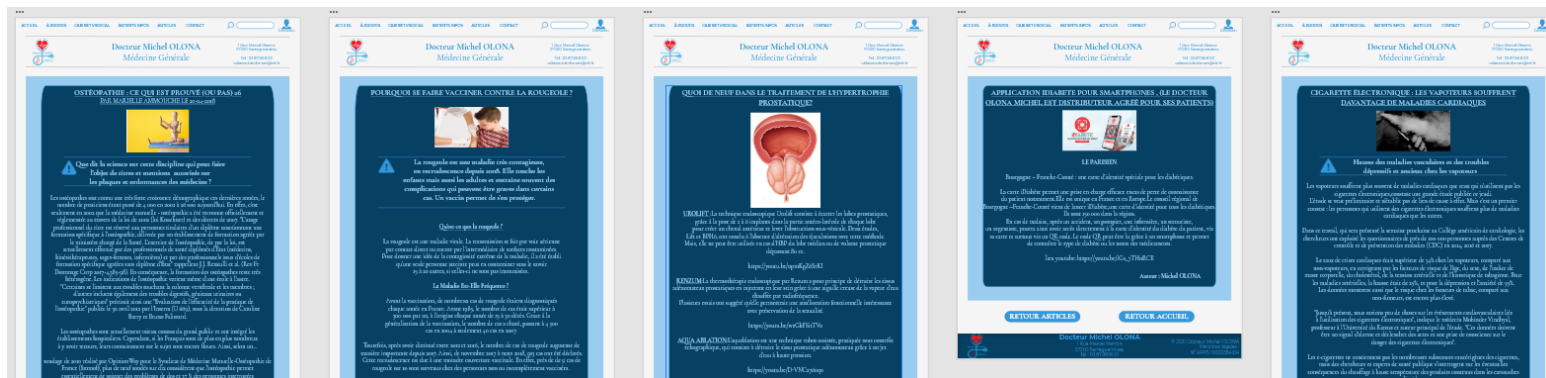
5.1.5 La page **ARTICLES :**

Cette page contiendra cinq blocs alignés verticalement contenant chacun un résumé des cinq articles santé rédigés par le Docteur Olona ; chaque bloc intègrera un bouton « Lire la suite » qui dirigera l'utilisateur sur la page contenant l'intégralité de l'article.

5.1.6 La page **CONTACT :**

Cette page contiendra un formulaire de contact (nom, prénom, email, téléphone, message) afin de pouvoir contacter le Docteur Olona.

5.1.7 Les pages **ARTICLES SANTE :**

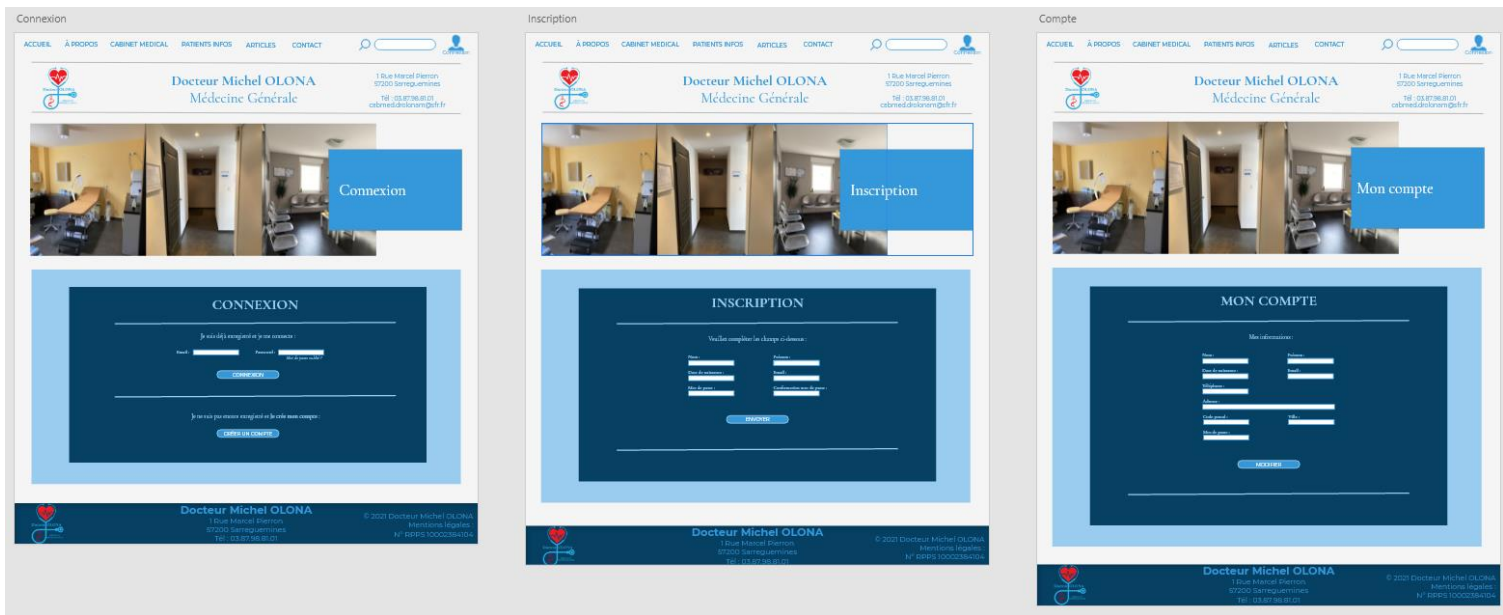


Les liens menant à ces pages se trouvent sur la page « **Articles santé** ».

Ces pages contiennent le titre de l'article santé, une photo d'illustration ainsi que l'article santé dans sa totalité. Certains articles contenant une partie texte très longue, j'utiliserai le composant « **accordéon** » de bootstrap pour adapter le texte aux écrans de téléphones portables.

Deux boutons se situeront en bas de page, l'un avec un lien renvoyant vers la page « **Articles** », l'autre vers la page « **Accueil** » du site.

Enfin pour terminer j'ai maqueté la partie destinée à la patientèle.



La page « **Inscription** » avec un formulaire contenant les champs *nom*, *prénom*, *date de naissance*, *email*, *mot de passe* et *confirmation du mot de passe*.

La page « **Connexion** » avec un formulaire contenant les champs *email* et *mot de passe*, ainsi qu'un lien renvoyant vers la page « **Inscription** » pour les personnes qui ne sont pas enregistrées.

La page « **Mon compte** » sur laquelle l'utilisateur pourra retrouver ses informations, avec un bouton lui permettant de les modifier.

5.1.8 Modifications de la maquette en cours de projet :

Une fois le maquetage terminé, je l'ai présentée au Docteur Olona qui a voulu y apporter la modification de la partie consacrée à la patientèle, car il ne désirait pas que ces patients puissent s'enregistrer sur le site.

J'ai donc modifié cette partie du site pour la transformer en back-office pour le médecin, qui pourra dans cette partie apporter des modifications aux messages importants situés sur la page d'accueil, ainsi que des modifications sur les fiches santé au format PDF.

Ces fiches santé au format PDF ne se trouvent pas dans le maquetage, car c'est à 15 jours de la fin de ma période en entreprise que le Docteur Olona m'a informé de son souhait de voir apparaître sur le site quelque 600 fiches santé au format PDF qu'il avait récupérées au fur et à mesure du temps.

5.2 Réalisation du site web

5.2.1 La page d'accueil

◆ Création de mon header

Pour l'ensemble du site web, j'ai privilégié Bootstrap, auquel j'ai apporté des modifications à mon fichier CSS afin de l'adapter à mon site web

Conformément à la maquette, mon header contiendra un menu horizontal ainsi que la bannière.

◆ Création du menu

• ACCUEIL • A PROPOS • CABINET MEDICAL • PATIENTS INFOS • ARTICLES • CONTACT

J'ai réalisé ce menu avec flexbox.

Pour ce faire, j'ai dans un premier temps créé une liste avec les balises HTML `` et `` pour obtenir une liste non ordonnée.

J'ai mis en forme mon menu avec ces fonctions Bootstrap pour obtenir le résultat voulu

```
<ul class="menuText d-flex justify-content-around w-75 mx-auto">
```

- **D-flex** pour aligner mon menu horizontalement
- **Justify-content-around** pour étirer mon menu en laissant un espace à chaque extrémité
- **W-75** pour utiliser 75% de ma largeur
- **Mx-auto** pour centrer mon menu dans la largeur

J'ai ensuite créé un effet au survol des mots par la souris

• CABINET MEDICAL • PATIENTS INFOS • ARTICLES

J'ai déjà utilisé la pseudo-classe **:hover** pour créer un background bleu sur le mot au passage de la souris ; j'ai voulu améliorer mon effet, j'ai donc fait des recherches sur internet et vu qu'on pouvait agrandir ou rétrécir le background en changeant le **scale()**. Dans mon cas, juste horizontalement avec **scaleX()**. Puis j'ai appliqué une transition avec la valeur **ease**, qui permet à la vitesse de l'animation d'augmenter au milieu de celle-ci puis de ralentir à la fin. Enfin, j'ai utilisé la valeur **left** appliquée à **transform-origin** pour que l'animation se fasse en partant de la gauche.

◆ Création de la bannière

J'ai reproduit le même principe que celui du menu pour la création de ma bannière.



Une fois mes 3 parties créées (logo, profession et nom, coordonnées), je les ai alignées avec les **flexbox** Bootstrap.

```
<div class="bandeau border-top border-bottom border-2 border-dark d-flex justify-content-between align-items-center mx-auto">
```

- **Border top** et **bottom**, pour ajouter une bordure au-dessus et en dessous de mon bloc

- **Border-2** pour la taille de ma bordure
- **Border-dark** pour la couleur de ma bordure
- **Justify-content-between** pour étirer mes 3 blocs dans la largeur
- **Align-items-center** pour aligner verticalement mes 3 blocs
- **Mx-auto** pour centrer l'ensemble dans la largeur

J'ai ensuite englobé la **div** contenant mon menu et celle de ma bannière dans une autre div qui constituera mon header.

◆ **Application du responsive :**

J'ai adapté mon **header** à la taille de plusieurs écrans avec les fonctions **d-flex** et **d-none** de bootstrap, car je les trouve plus pratiques à appliquer que les **media queries** dans mon **CSS**.

Extrait des différents **breakpoint** du site Bootstrap :

	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container max-width	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-
# of columns	12					

- Voici donc mon code pour les écrans d'une taille supérieure ou égale à 1400 pixels :

```
<div class="header flex-column justify-content-between d-none d-xxl-flex">
XXL correspond à la taille d'écran supérieure ou égale à 1400 pixels
```

d-none -> d-xxl-flex sera invisible de 0 à 1399 px

d-xxl-flex : sera visible à partir de 1400 px

et le résultat



- Le code pour les écrans d'une taille d'écran allant de 1200 px à 1399 px

```
<div class="header flex-column justify-content-between d-none d-xl-flex d-xxl-
none">
```

XL correspond à la taille d'écran supérieure ou égale à 1200 pixels

d-none -> d-xl-flex : sera invisible de 0 à 1199 px

d-xl-flex -> d-xxl-none : sera visible de 1200 px à 1399 px

d-xxl-none : sera invisible au-dessus de 1400 px

Et le résultat



Pour éviter que les mots du menu se chevauchent, j'ai passé la largeur de ma div contenant mon menu de 75% (bootstrap : w-75) à 100%

```
<ul class="menuText d-flex justify-content-around w-75 mx-auto">
```

```
<ul class="menuText d-flex justify-content-around mx-auto">
```

➤ Le code pour les écrans d'une taille d'écran inférieure à 576 px

```
<div class="headerSM d-flex d-xl-none justify-content-around align-items-center border-bottom border-2 border-dark mx-auto">
```

d-flex -> d-xl-none : sera visible de 0 à 1199 px

d-xl-none : sera invisible au-dessus de 1200 px

et le résultat



J'ai ensuite appliqué le même principe, uniquement à ma classe **msBurgerCoord** qui contient le bloc central et le block de droite, pour adapter mon **header** aux écrans de taille inférieure ou égale à 768 px. J'ai également utilisé les **flexbox** pour restructurer complètement mon header.

```
<div class="msBurgerCoord d-none d-md-flex d-xxl-none">  
<div class="msBurgerCoord d-flex d-md-none">
```

Et le résultat (ici pour un smartphone ayant une largeur d'écran de 360 pixels)



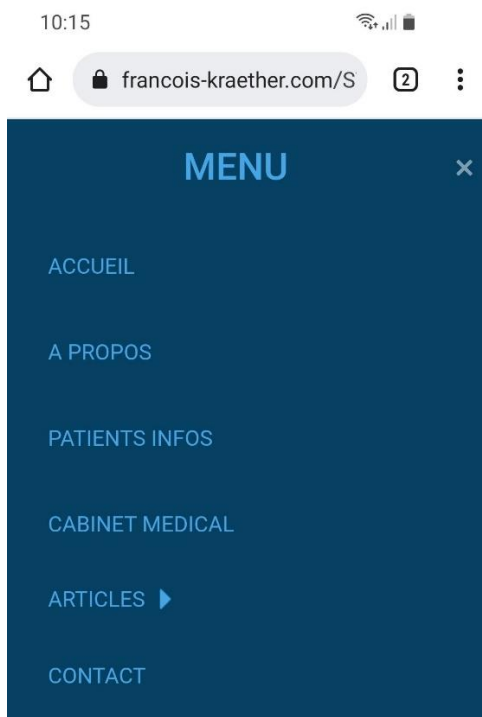
◆ **Création du menu burger :**

Pour créer mon menu burger, j'ai utilisé le composant « **offcanvas** » de bootstrap.

J'ai modifié les propriétés **CSS** afin d'adapter le **background** de mon site ainsi que sa largeur pour qu'une fois déployé, le menu remplit à 100% la largeur de l'écran.

```
.offcanvas
{
  background-color: #064063!important;
}
.offcanvas-start {
  width: 100%!important;
}
```

Voici le résultat de mon menu une fois déployé :



◆ Création de mon footer

Pour créer mon **footer**, j'ai procédé exactement de la même façon que pour la réalisation de mon **header**, c'est-à-dire avec les **flexbox** et bootstrap. Pour le responsive, j'ai procédé aussi de la même façon :

Code:

```
<div class="footer d-none d-xxl-flex">
<div class="footer d-none d-lg-flex d-xl-none">
<div class="footer d-none d-sm-flex d-md-none">
<div class="footer mt-5 d-flex d-sm-none">
```

Et les résultats



A chaque étape, j'ai modifié le texte et la taille de la police pour adapter le texte à mon **footer**.

Ces éléments étant présents sur la totalité des pages, je les ai isolés dans deux fichiers, « **header.php** » et « **footer.php** ».

Je les intégrerai dans toutes mes pages avec l'expression de langage **include** qui exécutera le fichier spécifié en argument.

Exemple du code avec mon header :

```
<?php
include("header.php");
?>
```

Exemple du code avec mon **PDO**:

J'ai privilégié "**require**" à "**include**" pour mon fichier **PDO**, car il provoque une erreur bloquante s'il est indisponible.

```
<?php
    require("pdo.php");
?>
```

J'ai également créé un lien sur « **Mentions légales** » de mon **footer** qui dirige sur page **mentionsLegales.php**, sur laquelle on peut trouver le détail des mentions légales.

```
<p class="copyrights me-2"> © 2021<br>Dr Michel OLONA<br><a
href="/STAGE/site_olona/mentionsLegales.php" class="text-reset text-
decoration-none">Mentions légales</a> :<br>N° RPPS 10002384104</p>
```

Une fois ces deux étapes terminées, j'ai développé le contenu de ma page d'accueil, toujours avec bootstrap et les **flexbox**.

Pour le design, j'ai respecté la charte graphique de mon cahier des charges, c'est-à-dire des nuances entre le bleu clair et le bleu foncé au niveau de mes blocs ainsi que pour la couleur de la police de caractères, et une bordure fine vert clair qui délimitera bien mes différents blocs ; j'ai gardé ce format car je trouve qu'il se prêtait très bien à l'activité pratiquée et surtout qu'elle convenait au Docteur Olona.

Pour le premier bloc « **informations importantes** », j'ai utilisé un carrousel bootstrap que j'ai modifié en y intégrant des cards bootstrap à la place des photos.

```
<div id="carouselExampleControls" class="carousel slide pt-4" data-bs-
ride="carousel">
    <div class="carousel-inner w-75 mx-auto">
        <div class="carousel-item active">
            <div class="card">
```

Le contenu des cards sera du texte, et afin que le Docteur Olona puisse gérer ses messages, je les ai placés dans une base de données nommée « **infos** ».

id	titre	message	modifiable	supprimable	← T →
1	RENDEZ-VOUS	Pour vos rendez -vous , connectez-vous à <a href="..."	0	0	Supprimer Copier Éditer
2	CONGÉS À VENIR DU DOCTEUR OLONA	Les dates de congés pour l'année 2022 n'ont pas en...	1	1	Supprimer Copier Éditer
3	NOUVELLE ADRESSE	À compter du 1er novembre 2020, le cabinet médical...	1	1	Supprimer Copier Éditer

Ensuite, j'ai créé un fichier **pdo.php** qui permettra la connexion entre mes pages web et ma base de données.

Exemple de fichier PDO (en local) :

```
<?php
    $dbh = 'mysql:host=localhost;dbname=dr_olona;charset=UTF8';
    $user = 'root';
    $pass = '';

    try {
        $pdo = new PDO($dbh, $user, $pass);
```



```

    } catch (PDOException $e) {
        print "Erreur !: " . $e->getMessage();
        die();
    }
?>

```

Explications :

1 – Chaîne de connexion à la base de données

```

$dbh = 'mysql:host=localhost;dbname=dr_olona;charset=UTF8';
$user = 'root';
$pass = '';

```

\$pdo correspond à l'instance PDO créée en passant au constructeur les paramètres qui constituent la chaîne de connexion.

\$dbh, **\$user** et **\$pass** sont des déclarations de variables.

Trois paramètres sont renseignés :

Le premier paramètre correspond à la chaîne de caractères

```

$dbh = 'mysql:host=localhost;dbname=dr_olona;charset=UTF8';

```

Le mot clé **mysql** indique qu'on a affaire à une base de données MySQL.

On trouve ensuite les mots clés **host** et **dbname** auxquels on affecte respectivement le nom du serveur de base de données (localhost dans mon cas car je suis en local) et le nom de ma base de données (dr_olona).

charset spécifie l'encodage (ou jeux de caractères).

Le deuxième paramètre correspond au nom d'utilisateur qui doit être préalablement créé sur le serveur de base de données.

```

$user = 'root';

```

Le troisième paramètre correspond au mot de passe de l'utilisateur.

```

$pass = '';

```

2 - Gestion d'erreurs éventuelles lors de l'instanciation (Exception PDOException)

Si les paramètres sont corrects, l'objet **\$pdo** sera créé sans encombre.

Si les paramètres ne sont pas bien déclarés, alors une exception de type **PDOException** est lancée automatiquement. Il faut donc la rattraper comme ceci :

```

try {
    $pdo = new PDO($dbh, $user, $pass);
} catch (PDOException $e) {

    print "Erreur !: " . $e->getMessage();
    die();
}

```

PHP essaie d'exécuter les instructions à l'intérieur du bloc **try** :

S'il y a une erreur, il rentre dans le bloc **catch** et fait ce qu'on lui demande (ici, on arrête l'exécution de la page en affichant un message décrivant l'erreur).

Si au contraire tout se passe bien, PHP poursuit l'exécution du code et ne lit pas ce qu'il y a dans le

bloc **catch**. Alors, la page PHP n'affichera rien.

De la même façon que pour mon **header** et mon **footer**, j'ai isolé ce fichier et l'appellerai dans mes pages web avec l'expression de langage **include** :

```
<?php
    include("pdo.php");
?>
```

Intégration du contenu de ma base de données dans les cards de mon carousel :

Voici le code en php que j'ai créé pour récupérer les contenus de ma base de données :

```
<?php
    $requete = "SELECT * FROM infos WHERE id=1";
    $req = $pdo->query($requete);
    $user = $req->fetch();
?>
```

Explications du code :

```
$requete = "SELECT * FROM infos WHERE id=1";
```

Je crée une requête qui sélectionne tous les champs (*) de ma table **infos**

```
$req = $pdo->query($requete);
```

J'effectue une requête sur la base de données (query en anglais signifie « requête »).

Je récupère ce que la base de données nous a renvoyé dans un autre objet que l'on a appelé ici **\$requete**.

```
$user = $req->fetch();
```

\$user est un **array** qui contient champ par champ les valeurs de la première entrée.

Il faut faire une boucle pour parcourir les entrées une à une.

\$req->fetch(), passe à l'entrée suivante.

La boucle est donc répétée autant de fois qu'il y a d'entrées dans la table.

Affichage des données dans ma div :

```
<div class="card-body d-flex flex-column align-items-center borBlocCar">
    <p class="rdvTitre mt-3"><?php echo $user['titre']; ?></p>
    <p class="rdvTxt text-center"><?php echo $user['message']; ?></p>
</div>
```

Explications :

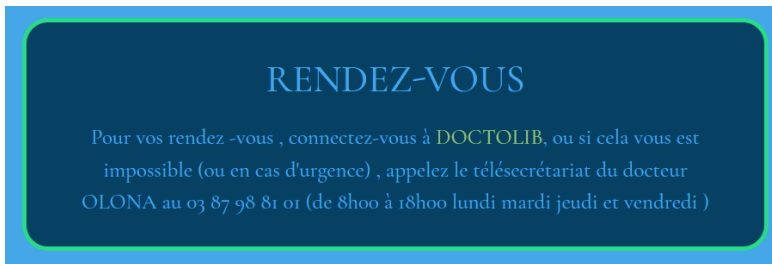
```
<?php echo $user['titre']; ?>
```

J'assigne le champs titre à **\$user** qui récupèrera la valeur de la première entrée qui s'affichera avec **echo**.

```
<?php echo $user['message']; ?>
```

J'assigne le champs message à \$user qui récupèrera la valeur de la première entrée qui s'affichera avec **echo**.

Voici le résultat obtenu :



Je récupère bien les valeurs des champs **titre** et **message** de la première entrée de ma base de données **infos**

id	titre	message	modifiable	supprimable	
1	RENDEZ-VOUS	Pour vos rendez -vous , connectez-vous à <a href="..."	0	0	Supprimer Copier Éditer

J'ai répété cette opération pour le 2ème et le 3ème message.

5.2.1.1 Page d'accueil – bloc à propos

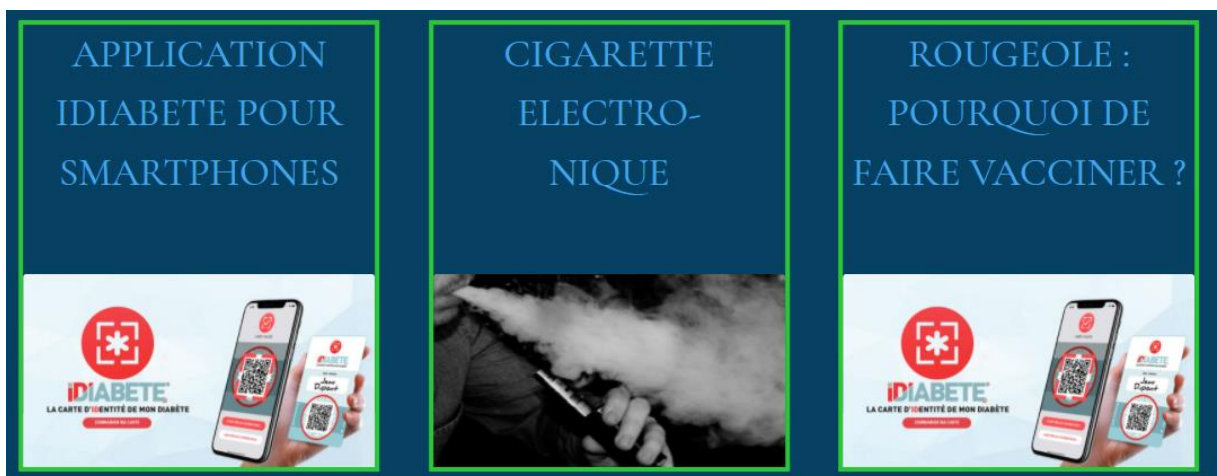
J'ai réalisé ce bloc avec bootstrap et les **flexbox** pour positionner la photo et les textes.

5.2.1.2 Page d'accueil – Cabinet médical

Pour ce bloc, j'ai utilisé un modèle de carrousel bootstrap que j'ai adapté à mon site.

5.2.1.3 Page d'accueil – Bloc articles santé

Pour ce bloc, j'ai utilisé les cards Bootstrap.



5.2.2 Page à propos

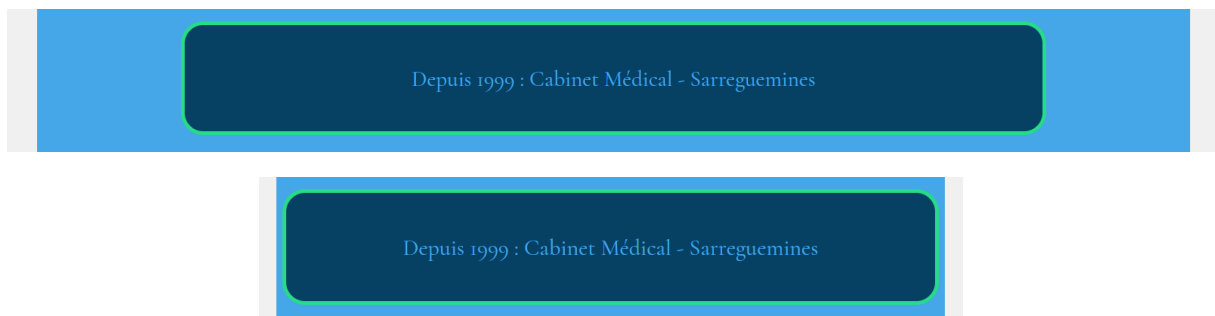
J'ai utilisé bootstrap et les **flexbox** pour le positionnement des différents éléments de cette page.

J'ai modifié les tailles de polices de caractère pour le rendre responsive.

J'ai également étiré mes blocs par rapport aux marges pour les écrans de taille inférieure à 992px afin de rendre le texte plus lisible. Pour ce faire, je suis passé d'une largeur de 75% à 100%, et, pour néanmoins garder une marge, j'ai utilisé les codes bootstrap « **margin start** » et « **margin end** »

```
ms-2 me-2
```

Résultat :



5.2.3 Page Cabinet médical

5.2.3.1 Page Cabinet médical – Bloc « Actes réalisés au cabinet »

J'ai utilisé les **cards** de bootstrap pour décrire les actes médicaux exercés au sein du cabinet, dans lesquels j'ai placé des spinners de bootstrap pour avoir une petite animation



J'ai modifié les **cards** et les spinners afin de les adapter à ce que je voulais via mon fichier **CSS**, par exemple pour les **cards** un **border-radius** de 30% pour avoir un aspect légèrement arrondi, ou la vitesse de l'animation et la couleur des spinners, etc.

5.2.3.2 Page Cabinet médical – Bloc « Diaporama »

Pour réaliser ce diaporama, j'ai utilisé un **carrousel** de type bootstrap. Contrairement à celui qui se trouve sur la page d'accueil, celui-ci contient des légendes sur les photos.

J'ai vu en cours avec mon formateur de front-end les carrousels **SLICK**, mais une fois ma période de stage terminée. Si j'avais vu ce carrousel avant mon stage, je l'aurais utilisé à la place de celui de bootstrap, car je le trouve très complet au niveau des options de configuration.

J'ai utilisé ce type de carrousel sur ma [page personnelle](#) après l'avoir vu en cours pour m'entraîner à le manipuler.

5.2.3.3 Page Cabinet médical – Bloc « Horaires des consultations »

J'ai réalisé un tableau pour indiquer les horaires des consultations et les visites à domicile.

APRES-MIDI									
	LUNDI	MARDI	MERCREDI	JEUDI	VENDREDI	SAMEDI	DIMANCHE		
12H30									
13H									
13H30		de 13H30 à 15H30							
14H	de 14H à 18H30 CONSULTATIONS CABINET MÉDICAL	VISITES A DOMICILE		de 14H à 18H30 CONSULTATIONS CABINET MÉDICAL	de 14H à 18H30 CONSULTATIONS CABINET MÉDICAL				
14H30									
15H									
15H30									
16H			de 16H à 19H CONSULTATIONS CABINET MÉDICAL						
16H30									
17H									
17H30									
18H									
18H30									
19H									
20H									

J'ai entièrement codé ce tableau avec les éléments HTML `<tr>`, `<td>` et `<th>`.

J'ai utilisé des codes bootstrap pour les couleurs du **background** et du texte (**bg-success** et **text-white**).

```
<td rowspan="9" class="openCab bg-success text-white">de 08H<br><br>à 12H<br><br><br>CONSUL-<br><br>TATIONS<br><br><br>CABINET<br><br>MÉDICAL</td>
```

J'ai repris ce tableau dans ma page « **contact** », dans laquelle j'explique comment j'ai adapté ce tableau aux différentes tailles d'écran.

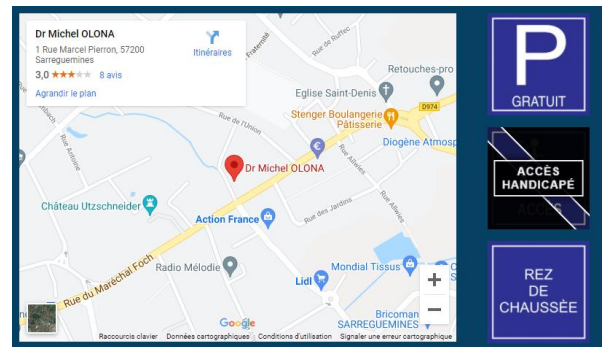
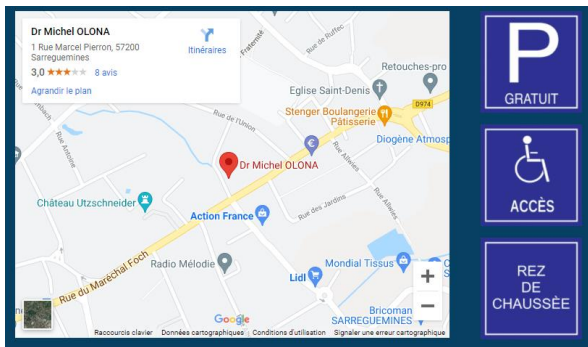
5.2.3.4 Page Cabinet médical – Bloc « Plan d'accès »

Pour le plan d'accès, j'ai utilisé l'**api google map** que j'ai placée dans une `<div>`.

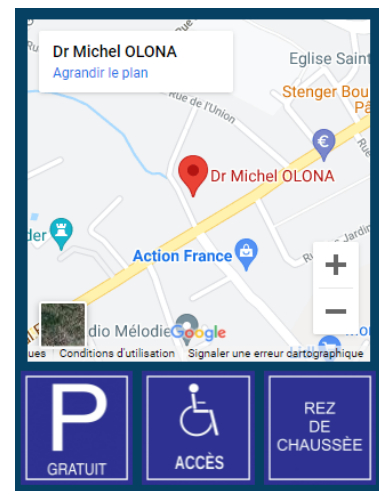
J'ai également créé des petits panneaux avec le logiciel **paint.net** pour indiquer les particularités de l'accès au cabinet.

J'ai également appliqué une petite animation sur les panneaux avec un code que j'ai trouvé à cette adresse : <https://www.littlesnippets.net/snip1573>

J'ai modifié les propriétés CSS de ce code pour l'adapter aux panneaux.



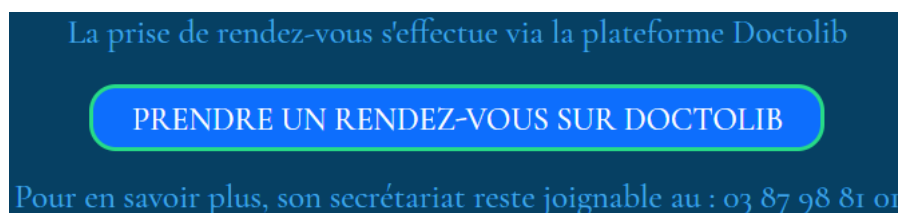
J'ai également appliqué les propriétés des flexbox pour adapter ce bloc aux différentes tailles d'écran (capture à droite pour les écrans de smartphones).



5.2.4 Page informations

5.2.4.1 Page informations – bloc rendez-vous

Dans ce bloc, je voulais placer une API Doctolib car j'ai vu que cela existait sur google. J'ai demandé au médecin de se renseigner comme il est professionnel de santé et a un compte sur doctolib, mais il n'y a pas eu de suite. Alors, j'ai créé un lien dans le bloc qui dirige l'utilisateur directement sur la page doctolib du Docteur Olona ; j'ai utilisé l'attribut HTML `target="_blank"` afin que la page doctolib s'ouvre dans un nouvel onglet, ceci pour garder l'utilisateur sur le site.



```
<a class="btn lienInterne btn-primary mx-auto mt-1 fs-4"
href="https://www.doctolib.fr/medecin-generaliste/sarreguemines/michel-olona"
target="_blank" role="button" style="width: 550px;height:50px">PRENDRE UN
RENDEZ-VOUS SUR DOCTOLIB</a>
```

Le docteur Olona n'acceptant plus de nouveau patient, j'ai fait ressortir cette information en appliquant un effet **hover** au texte, qui passe du bleu au rouge au survol du texte par la souris. J'ai également appliqué une transition de 2 secondes afin de rendre l'effet plus fluide.

```
.mouhovred {  
  -webkit-transition: color 2s;  
  transition: color 2s;  
}  
.mouhovred:hover {  
  color: rgb(206, 58, 13);  
}
```

5.2.4.2 Page informations – bloc « Expertises et actes »

Pour ce bloc, j'ai utilisé les cards bootstrap. J'ai placé le texte dans la partie supérieure et une image d'illustration dans la partie inférieure.

J'ai encore utilisé la pseudo-classe **hover** pour animer mes cards ; l'image passe d'une forme carrée à une forme circulaire au survol de l'image par la souris avec la propriété **border-radius** à laquelle j'ai affecté une valeur de 50%.



CSS :

```
.circle:hover  
{  
  border-radius:50%;  
  transition: 0.8s;  
}
```

5.2.4.3 Page articles santé (fiches PDF) – Partie rechercher une fiche

Pour rechercher une des fiches PDF stockée dans ma base de données, j'ai créé un formulaire avec un **input**.

J'ai ensuite fait une **requête préparée** qui sélectionne le champs « **titre** » de ma table « **docpdf** » dans lequel sont stockés les titres de mes fiches PDF.

```
<?php
    $requete = 'SELECT * FROM docpdf WHERE 1 ' ;
    if(!empty($_POST['docpdf']))
    {
        $requete .= ' AND ( titre like "%'.$_POST['docpdf'].'%" )';
    }
    echo $requete;
    $req = $pdo->query($requete);
    $Nom = $req->fetchAll();
?>
```

Explications:

```
$requete = 'SELECT * FROM docpdf WHERE 1 ' ;
```

Je crée une variable **\$requete** qui sélectionne tous les champs (*) de ma table **docpdf**

```
if(!empty($_POST['docpdf']))
```

je vérifie si les variables ne sont pas vides

```
$requete .= ' AND ( titre like "%'.$_POST['docpdf'].'%" )';
```

J'utilise l'opérateur **LIKE** qui me permet d'effectuer une recherche. Le caractère **"%"** est un caractère joker qui remplace tous les autres caractères.

Le point après **\$requete** sert à la concaténation des deux **\$requete** dans le cas où les champs ne sont pas vides.

```
$req = $pdo->query($requete);
```

Je déclare une variable **\$req** qui exécutera ma requête

```
$Nom = $req->fetchAll();
```

Je récupère toutes mes lignes avec **fetchAll**

Exemple de recherche avec le mot « pour » :

- Recherche



Rechercher un mot clé dans les titres des PDF


```
<?php
foreach ($Nom as $res)
{
?>
```

J'utilise la boucle **foreach** qui passe en revue chaque ligne du tableau. A chaque passage, elle met la valeur de cette ligne dans une variable temporaire **\$res**


```
<p class="lh-lg my-auto align-middle"><?php echo $res ['titre']?></p>
```

Ensuite, j'affiche les valeurs dans ma boucle avec **echo**

- Résultat

RESULTAT DE VOTRE RECHERCHE
APPLICATION IDIABETE pour smartphones Voir cette fiche
7 bons conseils pour préserver le sommeil de vos proches souffrant d'Alzheimer Voir cette fiche
5 infos indispensables pour protéger les enfants des dangers du soleil Voir cette fiche
Exemples de remboursement pour une consultation chez le médecin. Voir cette fiche
Arrêter de fumer peut m'apporter de vrais bénéfices pour ma santé Voir cette fiche

A partir des résultats retournés, je peux cliquer sur « Voir cette fiche » pour voir le détail de la fiche.

```
<a class="boutCol" href="/STAGE/site_olona/details_pdf.php?id=<?php echo $res['id']; ?>">Voir cette fiche</a>
```

Pour ce faire, j'ai placé un lien sur le texte « Voir cette fiche » vers la page **details.pdf**, avec ma variable **\$res** qui récupère l'**id** correspondant au titre de ma fiche pdf.

5.2.5 Page articles santé (pdf)

5.2.5.1 Page articles santé (pdf) – Partie afficher le détail d'une fiche

```
<?php  
$requete = 'SELECT * FROM docpdf WHERE id = '.$_GET['id'];  
$req = $pdo->query($requete);  
$user = $req->fetch();  
?>
```

Je récupère ma fiche santé pdf avec

```
id = '.$_GET['id']
```

qui affichera l'**id** sélectionnée dans ma barre d'adresse

```
localhost/STAGE/site_olona/details_pdf.php?id=56
```

J'affiche ensuite les données dans ma page web avec **echo** :

```
<h1 class="mb-5 text-center">FICHE SANTE N° <?php echo htmlspecialchars($user['id']); ?></h1>
```

Affichage du numéro de ma fiche (L'**id** étant une **clé primaire unique**, elle correspondra au numéro de ma fiche).

Pour des raisons de sécurité, j'utilise la fonction `htmlspecialchars` dans mon code php, ceci pour coder l'entrée d'un utilisateur sur le site afin que les utilisateurs ne puissent pas insérer des codes HTML nuisibles dans la page web.

```
<h2 class="pb-5"><?php echo htmlspecialchars( $user['titre']); ?></h2>
```

Affichage du titre de ma fiche

```
<iframe id="recup" src="<?php echo "/STAGE/site_olona/PDF/$user[pdf]"; ?>"  
class="docPDF mt-3" width="100%" height="1100px">
```

Affichage de mon fichier PDF, que j'ai choisi d'afficher dans un élément HTML `<iframe>` qui permet en fait d'obtenir une page HTML intégrée dans la page courante.

Résultat :



FICHE SANTE N° 56

Exemples de remboursement pour une consultation chez le médecin.

[RETOUR A LA LISTE](#)

201702MG4008TExemple...RemboursementPRE.pdf 1 / 1 | 100% + |

Exemples de remboursement pour une consultation chez le médecin.

Exemple pour une consultation chez un généraliste de secteur 2, c'est-à-dire pratiquant des honoraires libres :

Base de remboursement de la sécurité sociale = 23 €			
Taux de remboursement de la mutuelle	Remboursement de la sécurité sociale (taux = 70%)	Remboursement de la mutuelle	Total maximum de votre remboursement (en fonction du prix de la consultation)
100 %	15,10 €	6,90 €	22 €
150 %	15,10 €	18,40 €	33,50 €

5.2.5.2 Page articles santé (fiches PDF) – Partie « sélectionner une fiche »

J'ai également mis sur ma page contenant les articles santé au format pdf une liste déroulante avec l'objet de formulaire « `select` ».

Pour ce faire, je prépare et exécute ma requête, de la même manière qu'expliquée précédemment.

```
<?php  
$query = "SELECT titre, pdf FROM docpdf";  
try {  
    $pdo_select = $pdo->prepare($query);  
    $pdo_select->execute();  
    $listePdfs=$pdo_select->fetchAll();  
  
} catch (PDOException $e){
```

```

    echo 'Erreur SQL : '. $e->getMessage(). '<br/>';
    die();
}
?>

```

J'utilise la fonction intégrée en PHP `fetch_all()`, qui va récupérer toutes les lignes de résultat du champ « **titres** » de ma table « **docpdf** » et retourne l'ensemble de résultats sous la forme d'un tableau associatif.

```

<select class="listeFichesPdf w-75" style="height: 36px;"
onchange="selectFichePdf(this.value)" id="Sel2Fiches">

```

Dans mon `select`, j'utilise l'évènement javascript « **onchange** » pour que l'utilisateur modifie l'option sélectionnée.

Fichier javascript *main1.js*

```

function selectFichePdf(pdf){
    document.getElementById("iframePdf").src = pdf;
}

```

Je définis une fonction et définis les paramètres que celle-ci utilise.

J'utilise la méthode `getElementById()` qui renvoie un objet représentant l'élément dont la propriété `id` « **iframePDF** » correspond à la chaîne de caractères spécifiée.

`Src` correspond à mon `iframe` dans laquelle j'afficherai mon fichier pdf (code html ci-dessous).

```

<iframe src="/STAGE/site_olona/PDF/_accueil.pdf" class="mx-auto docPDF mt-1
mb-5" width="100%" height="1195px" id="iframePdf"></iframe>

```

```

<?php
    foreach($listePdfs as $pdf) {

        $url = $pdf['pdf'];
        $titre = $pdf['titre'];
        echo '<option
value="/STAGE/site_olona/PDF/.'.$url.'">'.$titre.'</option>';
    }
?>

```

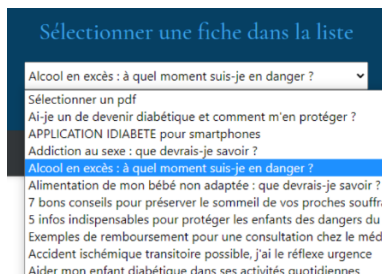
J'utilise la boucle `foreach` qui passe en revue chaque ligne du tableau. A chaque passage, elle met la valeur de cette ligne dans une variable temporaire `$pdf`.

Je déclare la variable `$url` qui correspondra au champ « **pdf** », qui contient les noms de mes fichiers pdf, récupéré par ma variable temporaire `$pdf`.

Je déclare la variable `$titre` qui correspondra au champs « **titre** », qui contient les titres de mes fichiers pdf, récupérés par ma variable temporaire `$pdf`.

J'affiche les titres de mes fiches pdf dans ma liste déroulante avec `echo`.

Résultat :



5.2.6 Page ARTICLES SANTE

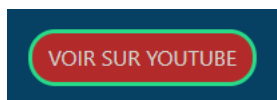
Cette page est destinée aux articles santé rédigés par le Docteur Olona, que j'ai récupérés sur le site docvitaefr. Il y a un total de 5 articles.

Sur cette page, on trouve 5 blocs contenant chacun un résumé de chaque article, ainsi qu'un bouton « **LIRE LA SUITE** » qui redirige vers la page dans laquelle se trouve la totalité de l'article.

5.2.7 Page ARTICLES SANTE détaillés

Ces pages aux nombres de 5 contiennent chacune la totalité d'un article.

Pour les liens « youtube » qui se trouvent dans certaines pages, j'ai créé des boutons renvoyant sur la vidéo youtube.



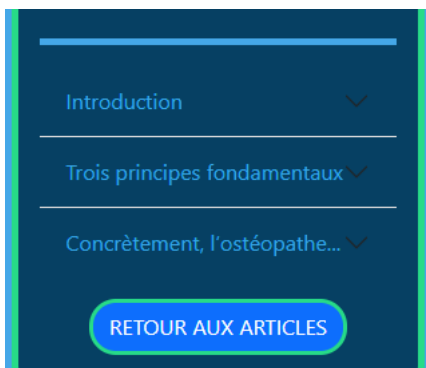
J'ai appliqué aux liens l'attribut **target="_blank"** afin que la page s'ouvre dans un nouvel onglet, dans le but de garder l'utilisateur sur le site du Docteur Olona.

J'ai placé en bas de chaque page un bouton avec un lien permettant le retour à la page « **Articles** ».

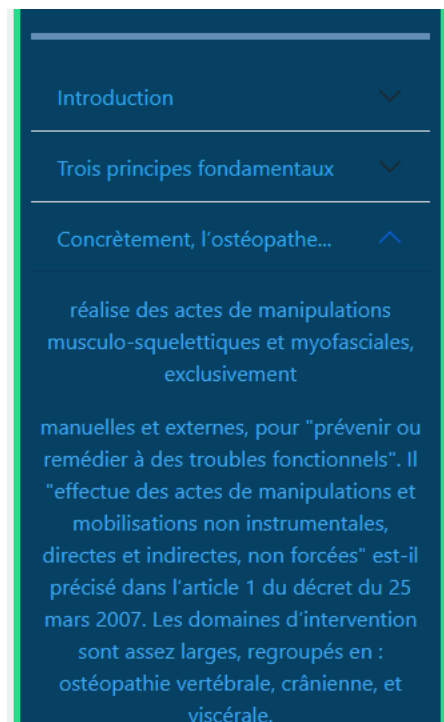
Il y a également des articles qui contiennent beaucoup de texte qui sont donc très longs, comme la page de l'article sur l'ostéopathie.

Pour cette page, en appliquant mon **responsive**, pour les écrans de tailles inférieures à 992 pixels, j'ai scindé l'article en 3 parties, que j'ai ensuite placées dans les **accordéons de Bootstrap**, afin que l'utilisateur n'ait pas à scroller longtemps pour arriver en bas de page.

ACCORDEON FERME



ACCORDEON OUVERT



5.2.8 Page CONTACT

Cette page devait contenir un formulaire de contact ; le Docteur Olona m'a demandé de remplacer le formulaire par les coordonnées téléphoniques du secrétariat, car il trouve qu'il est déjà trop sollicité au téléphone ; son adresse email n'apparaîtra pas sur le site pour la même raison.

J'ai donc remplacé le formulaire de contact par un bloc contenant les coordonnées téléphoniques. J'y ai placé une icône « téléphone » sur laquelle on peut joindre le cabinet directement à partir d'un smartphone en cliquant dessus avec ce code :

```
<a href="tel:+33387988101">
```

J'ai également rappelé les horaires du cabinet en intégrant dans un second bloc le tableau de la page « **Cabinet médical** ». Pour adapter le tableau à différentes tailles d'écran, j'ai dû supprimer des colonnes et des lignes, ainsi que la taille des polices de caractère.

Ecrans d'une taille supérieure ou égale à 992 pixels

	LUNDI	MARDI	MERCREDI	JEUDI	VENDREDI	SAMEDI	DIMANCHE
12H30							
13H							
13H30		de 13H30 à 15H30					
14H		VISITES A DOMICILE					
14H30							
15H							
15H30	de 14H à 18H30			de 14H à 18H30	de 14H à 18H30		

Taille allant de 576 à 991 pixels

	LUNDI	MARDI	JEUDI	VENDREDI
12H30				
13H				
13H30				
14H		de 13H30 à 15H30		
14H30		VISITES		
15H		A		
15H30	de 14H	DOMICILE	de 14H	de 14H

Taille inférieure à 576 pixels

LUNDI	MARDI	JEUDI	VENDREDI
	de 13H30 à 15H30		
de 14H à 18H30	VISITES	de 14H à 18H30	de 14H à 18H30
CONSUL-	A DOMICILE	CONSUL-	CONSUL-

5.3 Partie back-office

Cette partie sera destinée à la gestion des messages d'information de la page d'accueil ainsi que la gestion des fiches santé au format PDF.

Pour ce faire, j'ai utilisé le **CRUD**, acronyme anglais pour **create, read, update, delete**, qui désigne les quatre opérations de base pour la persistance des données, en particulier le stockage d'informations en base de données.

Soit en français : **créer, lire, mettre à jour et supprimer.**

5.3.1 La page inscription :

Cette page sera destinée uniquement au Docteur Olona. Il pourra rentrer un login, une adresse mail et un mot de passe pour s'enregistrer dans la table admin de la base de données et ensuite pouvoir se connecter au back-office via la page de connexion.

Une fois enregistré, le Docteur Olona pourra supprimer le fichier inscription.php via son client FTP.

Code php du fichier inscription_req.php

```
<?php
include "../pdo.php";
if (isset($_POST['username']) && isset($_POST['email']) &&
isset($_POST['password'])){
    $sql="SELECT * FROM admin WHERE username = '".$_POST['username']."' ;
    $stmt = $pdo->prepare($sql);
    $stmt->execute();
    $result = $stmt->fetch();
    if (empty($result))
    $req = $pdo->prepare("
    INSERT INTO admin (username, email, password)
VALUES(:username,:email,:password)",array(PDO::ATTR_CURSOR =>
PDO::CURSOR_FWDONLY)
);
    $req->execute(
    array(
        ':username'=>$_POST['username'],
        ':email'=>$_POST['email'],
        ':password'=> password_hash($_POST["password"], PASSWORD_DEFAULT),
```

```

)
);
$req->closeCursor();
header("location:connexion.php");
else{
header("location:inscription.php?etat=doublon");
}
}
else
{
header("location:inscription.php?etat=vide");
}
?>

```

Explications :

```
include "../pdo.php";
```

J'effectue une connexion entre PHP et le serveur de ma base de données avec mon fichier **PDO**

```

if (isset($_POST['username']) &&isset($_POST['email']) &&
isset($_POST['password'])){
$sql="SELECT * FROM admin WHERE username = '".$_POST['username']."' ;
$stmt = $pdo->prepare($sql);
$stmt->execute();
$result = $stmt->fetch();

```

Je récupère toutes les informations de la table admin de ma base de données et je teste si le username entré existe déjà dans ma base de données

```

if (empty($result))
{

```

Si aucun doublon n'est trouvé dans ma base de données, la procédure continue

```

$req = $pdo->prepare("INSERT INTO admin (username, email, password)
VALUES(:username,:email,:password)",array(PDO::ATTR_CURSOR =>
PDO::CURSOR_FWDONLY)
);
$req->execute(
array(
':username'=>$_POST['username'],
':email'=>$_POST['email'],
':password'=> password_hash($_POST["password"], PASSWORD_DEFAULT),
)
);
$req->closeCursor();
header("location:connexion.php");
}

```

Je prépare puis exécute ma requête pour ajouter avec **INSERT** le login, email et mot de passe dans les champs username, email et password de la table **admin** de ma base de données.

Je crée une clé de hachage pour le mot de passe avec la fonction **password_hash**.

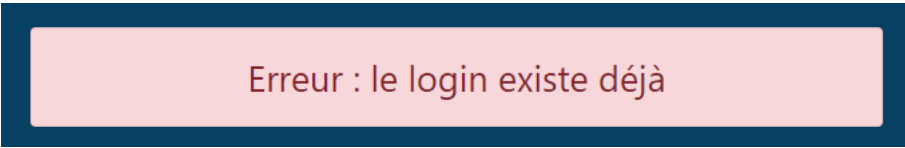
Je libère la connexion à ma base de données avec la fonction `closeCursor`.

Je redirige l'utilisateur vers la page `connexion.php` avec la fonction `header`.

```
else{
header("location:inscription.php?etat=doublon");
}
```

Si un doublon existe, je redirige l'utilisateur sur la page `inscription.php` avec la méthode `GET` du code se trouvant sur ma page `inscription.php` en affichant un message pour l'informer que le pseudo existe déjà.

```
if (isset($_GET["etat"]) && $_GET["etat"]=="doublon")
{
?>
<p class="mt-5">Erreur : le pseudo existe déjà :</p>
```




Erreur : le login existe déjà

```
else{
header("location:inscription.php?etat=vide");
}
```

Et la même opération que ci-dessus dans le cas où un champ ne serait pas rempli

```
if (isset($_GET["etat"]) && $_GET["etat"]=="vide")
{
```



Erreur : veuillez compléter tous les champs

Dans `phpMyAdmin`, je constate que je retrouve les données entrées dans la table `admin` de ma base de données et que le mot de passe a bien été haché.

+ Options				
	id	username	email	password
<input type="checkbox"/>		docteur	fgfgf@gcgff	\$2y\$10\$rOIPQDRfKvWfKQEJd2ReyuhhK3szbRe8dijyef6BEFX...

5.3.2 La page connexion :

Cette page est destinée à la connexion du Docteur Olona afin d'accéder au **back-office**.

Elle contient un formulaire de connexion avec 2 champs à compléter : login et password.

Code:


```

<?php
    include "../pdo.php";
    if (isset($_POST['username'])){
        $username = $_POST['username'];
        $password = $_POST['password'];
        $sql="SELECT * FROM admin";
        $stmt = $pdo->prepare($sql);
        $stmt->execute();
        $result = $stmt->fetchAll();
        $connexion = "";

        foreach ($result as $res) {
            if (password_verify($password, $res['password']) AND $username ==
$res['username'])
                $connexion = "Mot de passe correct";
        }

        if(!empty($connexion)){
            $_SESSION['username'] = $username;
            header("Location: index.php");
        }
        else{
            $message = "<p style='font-size: 27px'>Le nom d'utilisateur ou le mot de
passe est incorrect.";
        }
    }
?>

```

Explications :

```
include "../pdo.php";
```

J'effectue une connexion entre PHP et le serveur de ma base de données avec mon fichier **PDO**

```
if (isset($_POST['username'])){
```

Je vérifie si les champs du formulaire ont bien été remplis

```
$username = $_POST['username'];
```

```
$password = $_POST['password'];
```

Je stocke les données entrées par l'utilisateur dans des variables

```
$sql="SELECT * FROM admin";
```

```
$stmt = $pdo->prepare($sql);
```

```
$stmt->execute();
```

```
$result = $stmt->fetchAll();
```

```
$connexion = "";
```

Je récupère toutes les informations de la table « **admin** » de ma base de données.

```
foreach ($result as $res) {
```

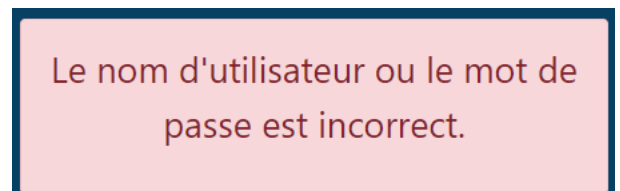
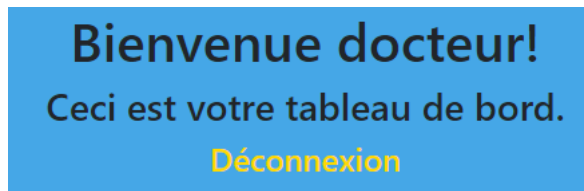
Je parcours mon tableau des résultats avec la boucle **foreach**

```
if (password_verify($password, $res['password']) AND $username == $res['username'])
```

Je vérifie si le mot de passe entré dans le champ correspond au mot de passe haché dans ma base de données avec `password_verify()`, et je vérifie si le login entré correspond bien à celui stocké dans le champ `username` de ma table `admin`

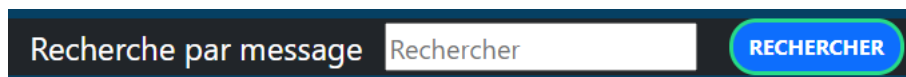
```
if(!empty($connexion)){
$_SESSION['username'] = $username;
header("Location: index.php");
}
Else
{
$message = "<p style='font-size: 27px'>Le nom d'utilisateur ou le mot de passe est incorrect.";
}
```

Si la variable connexion n'est pas vide, l'utilisateur se connecte puis est redirigé vers la page index du back-office, sinon la connexion ne s'établit pas et le message d'erreur « Le nom d'utilisateur ou le mot de passe est incorrect » s'affiche.



5.3.3 Page d'accueil (gestion des messages)

La page d'accueil du **back-office** contient un moteur de recherche interne permettant de retrouver des messages par leurs titres.



Le code est le même que celui utilisé précédemment pour la recherche des fiches santé pdf, seuls le champ et la table de la base de données changent.

Le médecin a la possibilité via cette partie du **back-office** de voir les détails des messages, les modifier ou les supprimer.

OPÉRATIONS* Les messages contenant du code HTML ne sont ni modifiables ni supprimables					
ID	Titre	Message	OPÉRATIONS *		
1	RENDEZ-VOUS	Pour vos rendez -vous , connectez-vous à DOCTOLIB, ou si cela vous est impossible (ou en cas d'urgence) , appelez le télésecrétariat du docteur OLONA au 03 87 98 81 01 (de 8h00 à 18h00 lundi mardi jeudi et vendredi)	Détails		
2	CONGÉS À VENIR DU DOCTEUR OLONA	Les dates de congés pour l'année 2022 n'ont pas encore été définies à ce jour ...	Détails	Modifier	Supprimer
3	NOUVELLE ADRESSE	À compter du 1er novembre 2020, le cabinet médical du Docteur OLONA Michel sera transféré au 1 rue Marcel Pierron à Sarreguemines(parking Thiriet)	Détails	Modifier	Supprimer

```
<td class="lh-lg my-auto align-middle"><?php echo $res ['id']?></td>
<td class="lh-lg my-auto align-middle" class="text-center"><?php echo $res
['titre']?></td>
<td class="lh-lg my-auto align-middle" class="text-center"><?php echo $res
['message']?></td>
```

Le même code est utilisé sur la page **details_info.php** pour afficher le détail des messages.

Le premier message contient un lien vers la plateforme doctolib pour la prise de rendez-vous, et, pour cette raison, j'ai empêché la possibilité de modifier ou supprimer ce message :

- Pour la modification, puisque le message contient du code, il est préférable de ne pas donner la possibilité à l'utilisateur, pour éviter des changements qui pourraient altérer le bon fonctionnement du code
- Pour la suppression, les rendez-vous se prenant le plus souvent via la plateforme doctolib, cette procédure est indispensable sur le site.

Pour empêcher la suppression et la modification des messages, j'ai ajouté 2 champs dans ma table « **infos** » : « **modifiable** » et « **supprimable** ».

Je leur ai donné un type **tinyint** ; ainsi, en définissant la valeur sur 0, le message ne sera pas modifiable, et en définissant la valeur sur 1, il pourra être modifié

id	titre	message	modifiable	supprimable
1	RENDEZ-VOUS	Pour vos rendez -vous , connectez-vous à <a href="...	0	0
2	CONGÉS À VENIR DU DOCTEUR OLONA	du 24 décembre au 31 décembre 2021	1	1
3	NOUVELLE ADRESSE	À compter du 1er novembre 2020, le cabinet médical...	1	1

<input type="checkbox"/>	3 message	varchar(10000)
<input type="checkbox"/>	4 modifiable	tinyint(1)
<input type="checkbox"/>	5 supprimable	tinyint(1)

Et mon code :

Les données sont récupérées avec une boucle **foreach**

```
foreach ($Nom as $res)
```

Et affichées dans le tableau avec **echo** en récupérant le champ « **id** » de ma table « **infos** ».

J'affiche uniquement la possibilité de modifier ou supprimer si la valeur des champs de ma table a pour valeur 1.

```
<td class="text_center"><a class="boutCol"
href="/STAGE/site_olona/admin/details_info.php?id=<?php echo $res['id'];
?>">Détails</a></td>

<td class="text_center"><?php if($res['modifiable']){ ?><a class="boutCol"
href="/STAGE/site_olona/admin/modifier_info.php?id=<?php echo $res['id'];
?>">Modifier</a><?php } ?></td>

<td class="text_center"><?php
if($res['supprimable']){ ?><a class="boutCol"
href="/STAGE/site_olona/admin/supprimer_info?id=<?php echo $res['id'];
?>">Supprimer</a><?php } ?></td>
```

5.3.4 Page modifier_info.php

Voici le code de ma requête permettant de modifier un message :

```
<?php
    include("../pdo.php");
?>
<?php
$req = $pdo->prepare("UPDATE infos
SET titre = :titre, message = :message
WHERE id = :id"
,array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY)
);
$req->execute(
array(
':id'=>$_POST['id'],
':titre'=>$_POST['titre'],
':message'=>$_POST['message'],
)
);
$req->closeCursor();
header('Location:./index.php');
?>
```

Tout d'abord, je prépare ma requête.

Le mot-clé **UPDATE** signifie que je vais modifier une entrée. Ensuite, le nom de la table (**infos**).

Le mot-clé **SET**, sépare le nom de la table de la liste des champs à modifier.

Viennent ensuite les champs qu'il faut modifier, séparés par des virgules. Ici, je modifie le champ « **titre** », je lui affecte la nouvelle valeur qui sera celle que j'aurai rentrée dans mon formulaire **input** de ma page **modifier_info.php**.

Et je fais de même pour le champ message.

Le mot-clé **WHERE** me permet de dire à MySQL quelle entrée il doit modifier.

J'exécute ensuite ma requête.

5.3.5 Fichier supprimer.php

Voici mon code pour supprimer une entrée

```
<?php
$req = $pdo->prepare("DELETE FROM infos WHERE id = :id"
,array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY)
);
$req->execute(
array(
':id'=>$_GET['id']
)
);
$req->closeCursor();
header('Location:index.php');
```

?>

DELETE FROM : pour dire « supprimer dans » ;

Infos : le nom de la table ;

WHERE est indispensable pour indiquer quelle entrée doit être supprimée.

5.3.6 Page d'accueil (gestion des fichiers pdf)

Dans cette partie, le Docteur Olona peut gérer les fiches santé au format PDF.

Tout comme les messages qui se trouvent sur la page d'accueil, il peut rechercher une fiche avec le moteur de recherche interne, afficher le détail d'une fiche, ainsi que les modifier ou les supprimer.

ID	Titre	Nom du fichier PDF	Catégorie	OPÉRATIONS		
2	Ai-je un de devenir diabétique et comment m'en protéger ?	ai-je-un-risque-de-d...omment-m-en-protéger.pdf	Diabète	Détails	Modifier	Supprimer
3	APPLICATION IDIABETE pour smartphones	application-idiabete...ee-pour-ses-patients.pdf	Diabète	Détails	Modifier	Supprimer
46	Addiction au sexe : que devrais-je savoir ?	addiction-au-sexe-que-devrais-je-savoir.pdf	Sexe	Détails	Modifier	Supprimer

Il peut aussi ajouter des fichiers PDF via un formulaire.



Code :

```
<form action="ajouter_pdf.php" name="add_pdf" method="post"
enctype="multipart/form-data">
```

- Dans mon formulaire, je déclare une option **enctype="multipart/form-data"** indiquant un échange de données, et j'utilise de la **method="post"** la seule permettant le transfert de données volumineuses

```
<?php
    $extension = "";
    if(!empty($_FILES))
    {
        switch ($_FILES['pdf']['type']) {
```

```

    case "PDF/pdf":
    $extension = '.pdf';
    break;
    default:
    $extension == 'erreur';
    break;
    }
    if($extension != 'erreur')
    {
    $nompdf = $_FILES["pdf"]["name"];
    move_uploaded_file($_FILES["pdf"]["tmp_name"],
$_SERVER['DOCUMENT_ROOT'].'/STAGE/site_olona/PDF/'.$nompdf);
    }
    else
    {
    $nompdf = '';
    echo 'Erreur de format';
    }
    }
    else
    {
    $nompdf = '';
    }
    if(!empty($_POST)){
    $req = $pdo->prepare("
INSERT INTO docpdf (pdf, titre, categorie)
VALUES(:pdf,:titre,:categorie)",array(PDO::ATTR_CURSOR =>
PDO::CURSOR_FWDONLY)
);
    $req->execute(
    array(
    ':pdf'=>$nompdf,
    ':titre'=>$_POST['titre'],
    ':categorie'=>$_POST['nom_categorie'],
    )
    );
    $req->closeCursor();
    }
?>

```

Explications :

```

$extension = "";
if(!empty($_FILES))
{
switch ($_FILES['pdf']['type']) {
case "PDF/pdf":
$extension = '.pdf';

```

```
break;
default:
$extension == 'erreun';
break;
```

switch() est une autre alternative à la structure **if()** / **elseif()** / **else** ou bien aux imbrications de blocs **if()**. Sa syntaxe est assez simple et repose sur l'utilisation de 3 mots clés : **switch**, **case** et **default**. Cette instruction conditionnelle permet de tester toutes les valeurs possibles que peut prendre une variable.

\$_FILES est la variable à tester.

Les différents '**case**' testent la valeur, et exécutent le code contenu entre le '**case**' en question et le '**break**'.

L'instruction contenue dans la clause **default** est l'instruction à exécuter par défaut lorsque la variable **\$_FILES** ne prend aucune des valeurs définies dans les différents '**case**'.

```
move_uploaded_file($_FILES["pdf"]["tmp_name"],
$_SERVER['DOCUMENT_ROOT'].'/STAGE/site_olona/PDF/'.$nompdf);
```

move_uploaded_file s'assure que le fichier **\$_FILES** est un fichier téléchargé par **HTTP POST**. Si le fichier est valide, il est déplacé jusqu'à destination.

\$_SERVER correspond à la destination du fichier.

```
if(!empty($_POST)){
$req = $pdo->prepare("
INSERT INTO docpdf (pdf, titre, categorie)
VALUES (:pdf, :titre, :categorie)", array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY)
```

Je vérifie que les champs ne sont pas vides.

Je prépare ma requête.

INSERT INTO va insérer les informations dans les champs **pdf**, **titre** et **catégorie** de ma table **docpdf**

```
$req->execute(
array(
':pdf'=>$nompdf,
':titre'=>$_POST['titre'],
':categorie'=>$_POST['nom_categorie'],
```

Et j'exécute ma requête.

Résultat :

AJOUTER UN ARTICLE PDF

Choisir un fichier

57	Accident ischémique transitoire possible, j'ai le ...	accident-ischémique-...i-le-reflexe-urgence.pdf	1
87	Aider mon enfant diabétique dans ses activités quo...	aider-mon-enfant-dia...our-une-vie-épanouie.pdf	1
100	Arrêter de fumer peut m'apporter de vrais bénéfice...	arreter-de-fumer-peu...efices-pour-ma-sante.pdf	1

Le fichier a bien été envoyé dans ma base de données.

5.3.7 Page d'accueil (gestion des catégories)

Le Docteur Olona peut aussi gérer les catégories liées aux fiches pdf.

GESTION DES CATEGORIES

Recherche par titre de catégorie OU

ID	Catégorie	OPÉRATIONS	
1	divers	Modifier	Supprimer
2	Diabète	Modifier	Supprimer
3	Cholestérol	Modifier	Supprimer
5	Enfant	Modifier	Supprimer

Le principe est le même que celui détaillé pour les fiches pdf.

Le Docteur Olona peut rechercher, ajouter, modifier ou supprimer une catégorie.

Le détail n'est pas important étant donné qu'il n'y aura que l'**id** et le nom dans la table catégorie.

Les fiches au format pdf sont liées à des catégories, dans le but de pouvoir les regrouper.

J'ai créé une **clé étrangère** dans ma table **docpdf** qui permet de garantir l'**intégrité référentielle** entre ma table **docpdf** et **categorie**.

La **clé étrangère** oblige ma table catégorie à être liée aux données de ma table **docpdf**.

Pour ce faire, mes 2 tables doivent avoir le type **InnoDB**

<input type="checkbox"/> categorie ★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	4 InnoDB
<input type="checkbox"/> docpdf ★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	11 InnoDB

Ainsi, je retrouve l'**id** de ma table **categorie** dans le champ **categorie** de ma table **docpdf**.
 Etant donné que les tables sont liées, le 1 qui se trouve dans le champ **categorie** de ma table **docpdf** correspondra à divers.

Table docpdf

id	titre	pdf	categorie
2	Ai-je un de devenir diabétique et comment m'en pro...	ai-je-un-risque-de-d...omment-m-en-protéger.pdf	1
3	APPLICATION IDIABETE pour smartphones	application-idiabete...ee-pour-ses-patients.pdf	1
46	Addiction au sexe : que devrais-je savoir ?	addiction-au-sexe-que-devrais-je-savoir.pdf	1
49	Alcool en excès : à quel moment suis-je en danger ...	alcool-en-exces-a-qu...nt-suis-je-en-danger.pdf	2
50	Alimentation de mon bébé non adaptée : que devrais...	alimentation-du-nour...entation-non-adaptee.pdf	1

Table catégorie

id	nom_categorie
1	divers
2	Diabète
3	Cholestérol
5	Enfant

6 - Présentation du jeu d'essai de la fonctionnalité la plus représentative

Un jeu d'essai est une séquence simple où l'on teste une partie d'une boucle de commande.

Le tableau ci-dessous est le jeu d'essai qui permet d'empêcher un upload de fichier dans un mauvais format de fichier. (Fichier **ajouter_pdf.php**)

<pre>switch (\$_FILES['pdf']['type']) { case "application/pdf": \$extension = '.pdf'; break; default: \$extension = 'erreur';</pre>	ENTREE	ATTENDU	OBTENU	ANALYSE
SANS	.docx	*.pdf	.docx	NOK
AVEC	.docx	*.pdf	Erreur de format	OK

La fonction **htmlspecialchars** dans PHP est utilisée pour convertir 5 caractères en entités HTML correspondantes. Cette fonction **htmlspecialchars** est utilisée pour coder l'entrée d'un utilisateur sur un site Web afin que les utilisateurs ne puissent pas insérer des codes HTML nuisibles dans une page web. (fichier **details_info.php**)

htmlspecialchars	ENTREE	ATTENDU	OBTENU	ANALYSE
SANS	<script>	< script >	<script>	NOK
AVEC	<script>	< script >	< script >	OK

7 - Description de la veille, durant le projet, sur les vulnérabilités de sécurité

7.1 password_hash()

J'ai utilisé la fonction **password_hash()** dans mon formulaire d'inscription, afin de créer un nouveau hachage en utilisant un algorithme de hachage fort et irréversible. La fonction **password_hash()** est compatible avec la fonction **crypt()**. Aussi, un mot de passe haché par la fonction **crypt()** peut être utilisé avec la fonction **password_hash()**.

Exemple avec le mot de passe 'Olona' que l'on retrouve haché dans la base de données :

id	username	email	password
5	docteur	fgfgf@gcgfi	\$2y\$10\$rOIPQDRfKvWfKQEJd2ReyuhhK3szbRe8dijyef6BEFX...

Dans le formulaire de connexion, j'ai utilisé la fonction **password_verify** pour vérifier que le mot de passe correspond à un hachage.

7.2 htmlspecialchars

J'ai utilisé la fonction **htmlspecialchars** pour convertir les caractères spéciaux en entités HTML.

htmlspecialchars() évite que des données fournies par les utilisateurs contiennent des balises HTML.

Les 5 caractères qui seront remplacés sont :

& (et commercial) devient "&" (guillemets doubles) devient """

" lorsque ENT_NOQUOTES n'est pas utilisée.

' (guillemet simple) devient "'" uniquement lorsque ENT_QUOTES est utilisée.

< (inférieur à) devient "<"

> (supérieur à) devient ">"

Certains caractères ont des significations spéciales en HTML, et doivent être remplacés par des entités HTML pour être affichés. **htmlspecialchars()** remplace tous ces caractères par leur équivalent dans la chaîne string.

7.3 Requêtes préparées

Pour prévenir les injections SQL, j'ai utilisé des requêtes préparées. Ce sont des requêtes dans lesquelles les paramètres sont interprétés indépendamment de la requête elle-même. De cette manière, il est impossible d'effectuer des injections. Pour ce faire, j'ai utilisé la méthode **prepare()** qui prépare la requête et **execute()** qui exécute la requête avec les paramètres.

8 - Description d'une situation de travail ayant nécessité une recherche, à partir de site anglophone

Dès que le site web est quasi terminé, j'ai voulu ajouter une animation sur les blocs.

En l'occurrence, j'aimerais qu'au survol du bloc par la souris, la bordure qui est à l'origine de couleur verte devienne rouge, dans le but d'attirer l'attention de l'utilisateur sur le bloc.

9 - Extrait du site anglophone, utilisé dans le cadre de la recherche décrite précédemment, accompagné de la traduction en français

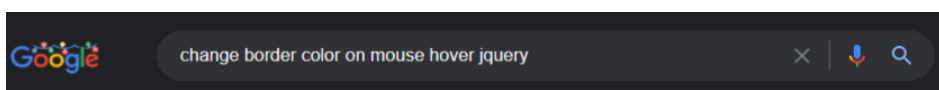
One of the problems I had for the conception of my website is that I wanted to change the color of my DiV border when the mouse flew over it with jQuery.

We have learned this code recently. For exemple, when I fly with the mouse over the frame « experience », I want the green border to change to a red border.



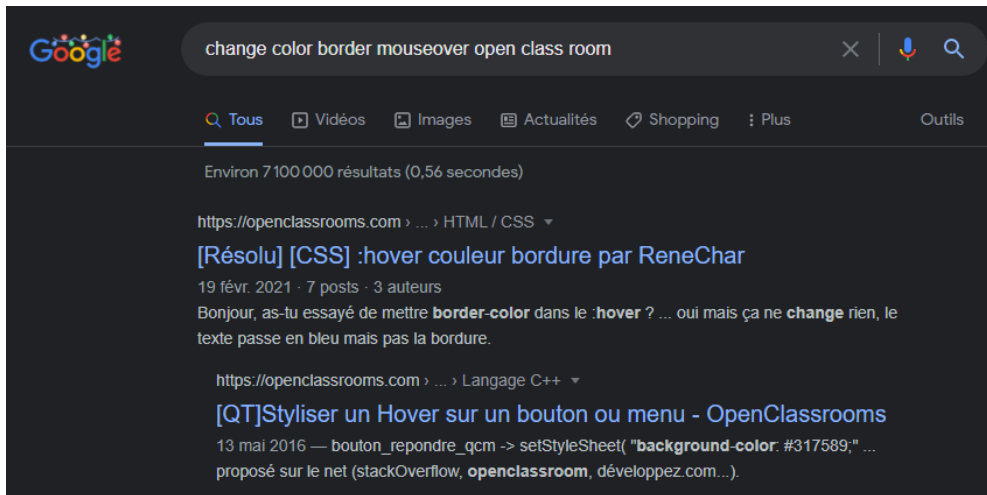
Bordure qui doit changer de couleur au survol de la souris

I searched on google *change border color on mouse hover jquery*



Recherche Google

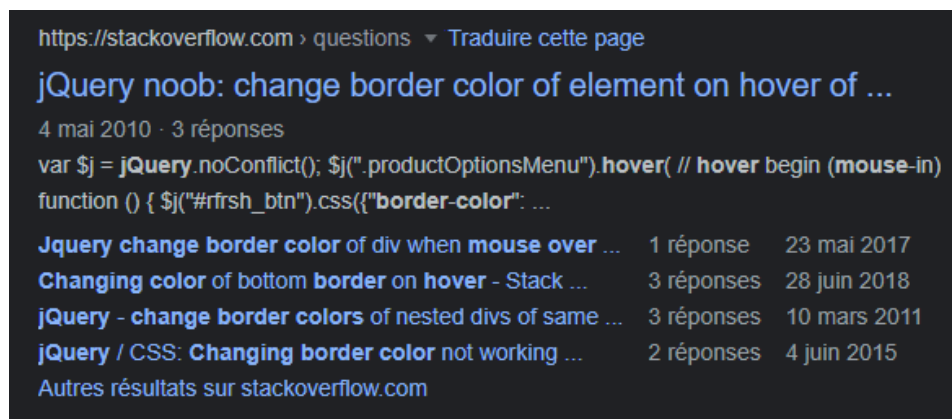
And I found a solution on the site OPENCLASSROOMS



Résultat ne correspondant pas au résultat voulu

But It didn't work it was only when you clicked on it.

So I looked for help on google, and I found a solution on the website STACKOVERFLOW which could solve the problem. It was exactly what I was looking for : changing the border color when the mouse flew over the selection.



Recherche avec le résultat voulu

I included the code, and I modified it to fit my website

```
<p class="txtPolPropos text-center fw-bold mt-3 pt-3">EXPERIENCE</p>
<div class="cadreCursus w-75 mx-auto mt-3 bor8loc">
  <div class="bioIndex d-flex justify-content-center">
    <div class="bioInfos w-75">
      <div class="cursus d-flex justify-content-center">
        <p class="dipCursus text-center mb-5">Depuis 1999 : Cabinet Médical - Sarreguemines</p>
      </div>
    </div>
  </div>
</div>
<script>
  var $j = jQuery.noConflict();

  $j(".productOptionsMenu").hover(
    // hover begin (mouse-in)
    function () {
      $j("#rfrsh_btn").css({"border-color": "#85c222"});
    },
    // hover end (mouse-out)
    function () {
      $j("#rfrsh_btn").css({"border-color": ""});
    }
  );
</script>
```

Code jQuery intégré à ma "DIV"

And finally I obtained the result that I wanted, and it now works perfectly.



Résultat final

Traduction en français :

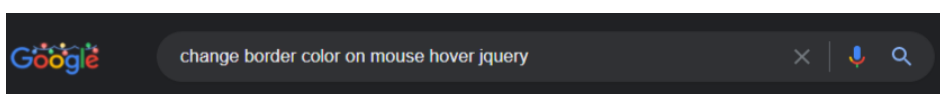
Un des problèmes que j'ai eu pour la conception de mon site web est que je voulais changer la couleur de ma bordure DIV lorsque la souris la survolait avec jQuery.

Nous avons appris ce code récemment. Par exemple, lorsque je survole avec la souris le cadre « expérience », je veux que la bordure verte se transforme en bordure rouge.



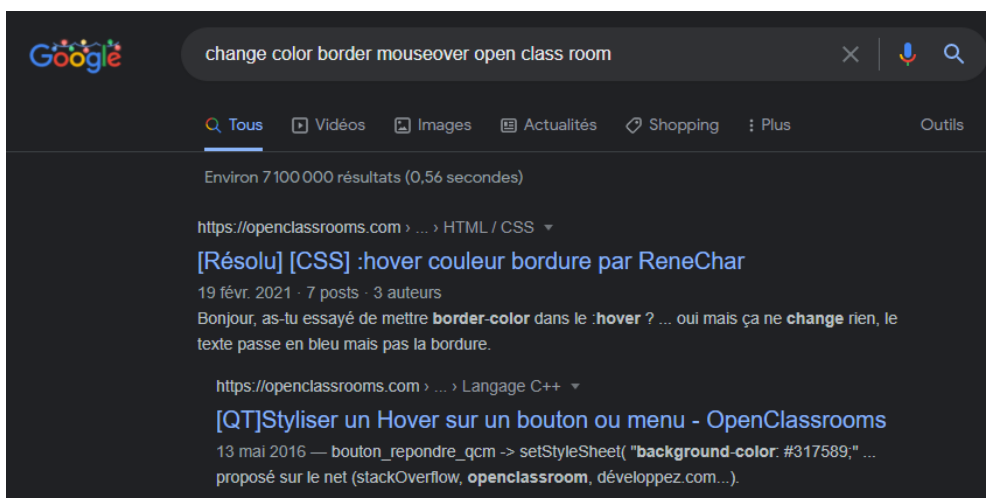
Bordure qui doit changer de couleur au survol de la souris

J'ai cherché sur google changer la couleur de la bordure au survol de la souris jquery



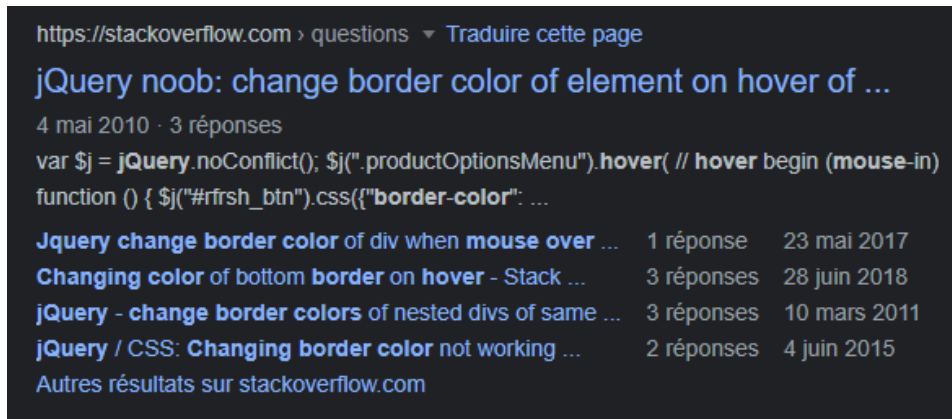
Recherche Google

Et j'ai trouvé une solution sur le site OPENCLASSROOMS



Résultat ne correspondant pas au résultat voulu

Mais cela n'a pas fonctionné, c'était seulement lorsque vous avez cliqué dessus. J'ai donc cherché de l'aide sur google, et j'ai trouvé une solution sur le site STACKOVERFLOW qui pourrait résoudre le problème. C'était exactement ce que je cherchais : changer la couleur de la bordure au passage de la souris sur la sélection.



Recherche avec le résultat voulu

J'ai inclus le code et je l'ai modifié pour l'adapter à mon site Web

```
<p class="txtPolPropos text-center fw-bold mt-3 pt-3">EXPERIENCE</p>
<div class="cadreCursus w-75 mx-auto mt-3 borBloc">
  <div class="bioIndex d-flex justify-content-center">
    <div class="bioInfos w-75">
      <div class="cursus d-flex justify-content-center">
        <p class="dipCursus text-center mb-5">Depuis 1999 : Cabinet Médical - Sarreguemines</p>
      </div>
    </div>
  </div>
</div>
</div>
<script>
  var $j = jQuery.noConflict();

  $j(".productOptionsMenu").hover(
    // hover begin (mouse-in)
    function () {
      $j("#frsh_btn").css({"border-color": "#85c222"});
    },
    // hover end (mouse-out)
    function () {
      $j("#frsh_btn").css({"border-color": ""});
    }
  );
</script>
```

Code jQuery intégré à ma "DiV"

Et finalement j'ai obtenu le résultat que je voulais, et ça marche maintenant parfaitement.



Résultat final