

# **RAPPORT DE STAGE**

***De Terence RENARD (Numéro INE 071022787DB)***

***Stage effectué du 15 Janvier 2024 au 23 Février 2024***

## **Développeur Web**

**Au sein du cabinet dentaire Dumitrascu Roxana**



**Coordonnées : Cabinet Dentaire Dumitrascu Roxana, 26 Rue Rudolph Serkin, 84000 Avignon, Vaucluse**

**Maître de stage : Mme Justine GARCIN**

**Réalisé dans le cadre de la seconde année de formation au BTS Services Informatiques aux Organisations (Option Solutions Logicielles et Applications Métiers) proposé par le lycée Théodore Aubanel (Aubanel For Business)**



# SOMMAIRE

<b>I - REMERCIEMENTS</b>	<b>3</b>
<b>II - PRÉSENTATION DE L'ENTREPRISE</b>	<b>4</b>
1 - Présentation du cabinet dentaire	4
2 - L'informatique au sein du cabinet dentaire	4
<b>III - DÉROULEMENT DU STAGE</b>	<b>5</b>
1 - Organisation du stage	5
2 - Langages informatiques et logiciels mis en œuvre	7
A - Langages informatiques	7
B - Logiciels	7
3 - Missions	8
4 - Suivi hebdomadaire	9
A - Semaine 1 (du 15 janvier au 19 janvier)	9
B - Semaine 2 (du 22 janvier au 26 janvier)	14
C - Semaine 3 (du 29 janvier au 2 février)	21
D - Semaine 4 (du 5 février au 9 février)	31
E - Semaine 5 (du 12 février au 16 février)	38
F - Semaine 6 (du 19 février au 23 février)	44
5 - Apports du stage	48
6 - Difficultés rencontrées	49
<b>IV - CONCLUSION</b>	<b>50</b>
<b>V - LEXIQUE</b>	<b>51</b>
<b>VI - ANNEXE</b>	<b>52</b>

## I - REMERCIEMENTS

La recherche d'un stage n'est, pour ainsi dire, pas une tâche agréable. Quand on demande à un individu encore inexpérimenté dans ce genre de démarche de trouver un stage avant une date butoire, au risque de devoir recommencer son année (ou d'abandonner sa formation), cela peut ajouter une bonne dose de stress non souhaitée.

L'année dernière, à cause d'un malheureux incident, j'ai dû trouver un autre stage d'urgence car celui que je voulais faire de base était grossièrement annulé. Cette année, dû fait de mon précédent stage en télétravail pour l'association [FarInWest](#), j'ai pu être pistonné par un enseignant et donc en trouver un quasiment directement au sein du cabinet dentaire Dumitrascu Roxana. Je fus très bien accueilli par celui-ci et je remercie profondément le personnel de m'avoir choisi pour réaliser leur site web. Qui plus est, m'adaptant très bien au contexte du télétravail (je travaillais sur le site depuis chez moi, et de temps en temps, je donnais rendez-vous aux personnes se chargeant de mon stage afin de valider, modifier ou refuser ce que je créais), ces six semaines de stage ont été très confortables pour moi, sans sacrifier ma productivité.

Je remercie Roxana DUMITRASCU et Justine GARCIN de m'avoir accordé leur confiance au sujet de la tâche qui m'a été confiée et de m'avoir laissé une grande liberté en ce qui concerne la création de leur site web. C'est un stage qui a permis de me faire travailler mon autonomie et ma confiance en moi en ce qui s'agit de mener un projet de bout en bout par mes propres moyens.

Une nouvelle fois encore, je tiens à remercier tous les enseignants du lycée Théodore Aubanel de nous préparer depuis dorénavant presque deux ans à la future épreuve de BTS qui nous attend. Ayant obtenu le baccalauréat sans vraiment savoir quoi faire ensuite, la première année à préparer une licence radicalement différente de ce que je fais maintenant m'a quelque peu rendu perplexe. « Qu'est-ce que je vais faire, maintenant ? ». C'est finalement avec la formation au BTS SIO que j'ai trouvé ma voie et je ne regrette en rien d'être dorénavant dans ce cursus qui me passionne.

## II - PRÉSENTATION DE L'ENTREPRISE

### 1 - Présentation du cabinet dentaire

Le Dr. Roxana DUMITRASCU exerce en tant que chirurgien-dentiste depuis onze ans en France, ayant succédé au Dr. Henri ALLOUCHE suite à sa retraite. Elle est titulaire de son diplôme depuis 1993.

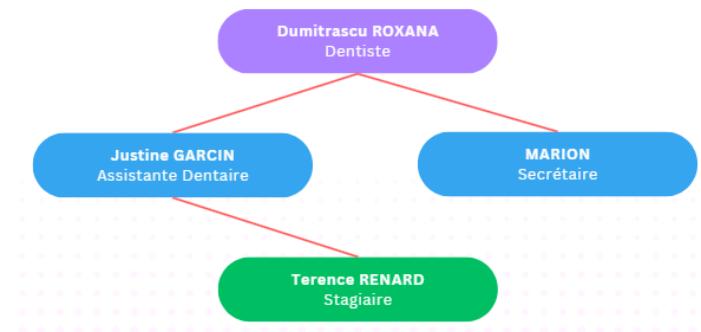
Le Dr. Dumitrascu est conventionnée et pratique des honoraires conformes aux régulations de l'assurance maladie. Elle assure des soins dentaires courants ainsi que des interventions spécialisées en parodontologie.

Pour mener à bien ses interventions, le Dr. Roxana DUMITRASCU est épaulée par son assistante, Justine GARCIN, qui travaille à ses côtés depuis 2012. Justine bénéficie également d'une formation en parodontie et en orthodontie.

### 2 - L'informatique au sein du cabinet dentaire

Le cabinet dentaire Dumitrascu Roxana ne possédait aucun site web au début du stage. Qui plus est, tout ce qui est relatif à l'hébergement web (qui se fera chez OVHcloud) géré par une autre personne ne sera fonctionnel qu'à partir de ma dernière semaine de stage. Cela en résulte un développement qui s'est déroulé majoritairement en localhost le temps que l'hébergement finisse par être utilisable.

De ce que j'ai pu comprendre, une autre personne avait été demandée pour réaliser un site web pour ledit cabinet, mais sans que cela soit concluant. J'interviens alors pour concevoir un site Internet de A à Z en l'espace de six semaines. L'objectif principal est d'abord d'avoir une partie "Vitrine" attrayante pour que l'utilisateur puisse se renseigner et que le cabinet ait une présence en ligne via un site web (ce qui peut potentiellement en augmenter sa visibilité). Ensuite, une partie d'administration permettant d'ajouter de nouveaux profils et de gérer un calendrier afin d'afficher les horaires.

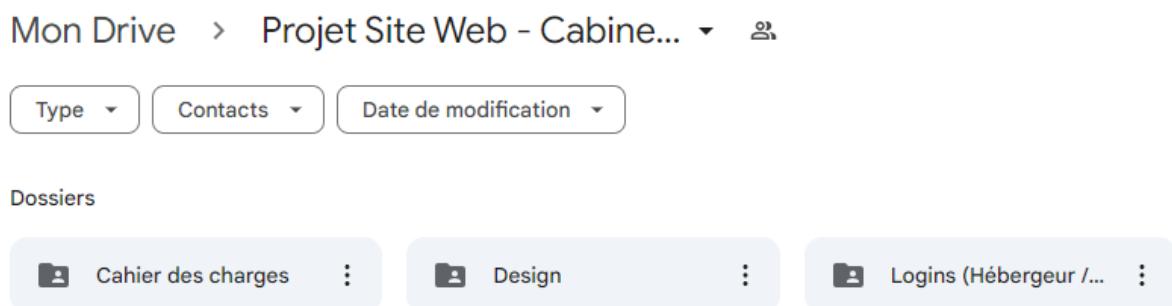


## III - DÉROULEMENT DU STAGE

### 1 - Organisation du stage

Mon stage s'est déroulé pendant six semaines, du 15 Janvier 2024 au 23 Février 2023, en télétravail. Mme GARCIN, à la fois responsable hiérarchique et cliente, avait en contact une personne s'y connaissant dans le développement web. Celle-ci me donna quelques semaines plus tard toutes les informations relatives à l'hébergement du futur site. Une fois tout en ordre, l'URL suivante était <https://dentiste-dumitrascu.fr/> et je pouvais transférer tout mon travail de mon ordinateur jusqu'au serveur hébergeant le site (OVHcloud) via un client FTP (FileZilla). En ce qui concerne la base de données, la transférer jusqu'à l'hébergeur fut simple, ne demandant simplement qu'à exporter celle de mon localhost (un serveur local de test afin de pouvoir développer en attendant que l'hébergement soit activé) pour l'importer directement vers OVHcloud. Qui plus est, un email de la personne gérant l'hébergement m'a fourni la procédure avec les différentes étapes, ce qui m'a mâché le travail.

En ce qui concerne tout ce qui est relatif à la concertation entre moi et ma cliente, nous échangions par SMS afin de nous informer mutuellement sur ce que nous ajoutions (moi à propos du code informatique, et Mme GARCIN pour ce qui est du contenu). Pour l'occasion, j'ai créé un dossier Google Drive afin de répertorier et organiser tout ce dont nous avions besoin.



*Un exemple des sous-dossiers qu'on peut retrouver dans le dossier partagé en question.*

De temps en temps, nous faisions le point en présentiel en ce qui concerne les nouvelles pages et fonctionnalités implémentées, généralement le Mercredi. Ainsi, j'obtenais ou non l'aval de ma cliente et ce processus se répétait environ une fois toutes les semaines. Cela me permettait de progresser à un rythme suffisamment optimal pour que les six semaines à ma disposition soient bien rentabilisées.

Le cahier des charges était structuré de façon simple, avec une partie concernant les rubriques statiques (tout le contenu visible, en somme) et une autre pour celles qui sont dynamiques (ce qui ne se voit pas, les interactions utilisateurs aboutissant à un résultat,

comme un formulaire de contact par exemple). Je profitais aussi du cahier des charges pour prendre des captures d'écran des différentes pages visibles composant le site afin qu'en attendant que l'hébergement soit actif, que ma cliente (ou mes clientes, si on compte Mme DUMITRASCU dans le lot) puisse voir ma progression. Je fais un petit aparté pour dire que je possédais une grande liberté dans la réalisation du site, étant le seul développeur à travailler dessus.

The top screenshot displays the homepage of the website. The header includes the logo 'Cabinet Dentaire Dumitrascu Roxana'. Below the header, a main title 'Une équipe du tonnerre qui vous redonnera le sourire !' is followed by the name 'Dumitrascu Roxana' and the title 'Chirurgien Dentiste'. A placeholder profile picture is shown. Below this, a bio states: 'Le docteur Dumitrascu Roxana exerce depuis 11 ans en France en tant que chirurgien dentiste. Elle est diplômée depuis 1993.' A large amount of placeholder text follows. The bottom screenshot shows a calendar interface for February 4-10, 2024. The days are labeled from Monday to Sunday. Blue and yellow rectangular boxes represent scheduled events or appointments throughout the day. The Windows taskbar at the bottom shows various open applications, including Microsoft Word, Google Sheets, and Google Docs.

*En attendant de pouvoir mettre le site web à la vue de tous, les différentes captures d'écran dans le Google Docs permettaient de constater mon avancement dans mes tâches et d'avoir un aperçu des pages qui seront disponibles.*

## 2 - Langages informatiques et logiciels mis en œuvre

### A - Langages informatiques

Pour la réalisation du site web, les langages HTML et CSS ont bien évidemment été sollicités (ces langages sont primordiaux pour au moins fournir un site web statique attrayant). J'ai aussi utilisé le JavaScript (avec jQuery) pour améliorer l'apparence esthétique du site afin de proposer des pages plus dynamiques. Enfin, le langage PHP m'a permis d'élaborer toute l'architecture dite « back-end » afin de communiquer avec le serveur (dont la base de données) pour proposer une interface d'administration ou encore un formulaire de contact qui envoie un mail une fois rempli.

En plus de ces langages précédemment cités, j'ai utilisé diverses bibliothèques<sup>1</sup> afin de développer le site dans les meilleures conditions. Tout d'abord, j'ai utilisé de [Bootstrap](#), offrant beaucoup de codes CSS et JavaScript pour aller plus vite. Ensuite, dans le cadre du formulaire de contact, j'ai réutilisé [PHPMailer](#). Etant donné que j'avais déjà codé un formulaire complet pour [FarInWest](#) (mon précédent stage), je n'avais qu'à le reprendre et l'adapter sur le nouveau site. [Animate On Screen \(AOS\)](#) pour les animations d'apparition des images lorsque l'on scrolle. Enfin, citons [FullCalendar](#), une bibliothèque JavaScript permettant d'intégrer des calendriers et de les utiliser de façon à en faire ce que bon nous semble.

### B - Logiciels

Pour mener à bien le projet de réalisation du site, voici les différents logiciels dont j'ai eu recours pour arriver à mes fins :

- [WampServer](#), puis [XAMPP](#) : Des ensembles de logiciels de type « plates-formes de développement Web » permettant la mise en place d'un serveur web local pour faire des tests. La raison m'ayant poussé à passer de WampServer à XAMPP est qu'à cause d'un bug inexplicable, WampServer ne répondait plus du tout (et ce, malgré de multiples réinstallations, dont certaines avec des versions différentes), ce qui m'a forcé à migrer pour XAMPP.
- [Visual Studio Code](#) : Un éditeur de code développé par Microsoft permettant de rédiger et corriger du code de la façon la plus optimale et confortable possible.
- [Screaming Frog](#) : Un logiciel faisant ce que l'on appelle du « web scraping », c'est-à-dire récupérer les données d'un site afin de les réutiliser. Dans le cadre ici présent, c'était dans le but d'avoir un sitemap complet et efficace pour que [dentiste-dumitrascu.fr](#) soit référencé par Google.

---

<sup>1</sup> Je me suis renseigné si le mot exact que je devais utiliser était « librairies » ou « bibliothèques », il s'avère que c'est ce dernier comme en témoigne l'article suivant :

<https://talks.freelancerepublik.com/framework-bibliotheque-differences/>, « librairie » est un anglicisme.

- [\*\*FileZilla\*\*](#) : Un logiciel client FTP (File Transfer Protocol) open-source utilisé pour transférer des fichiers entre un ordinateur local et un serveur distant via Internet. Il sert principalement à faciliter le téléchargement et le téléversement de fichiers vers et depuis un serveur web, pour qu'ainsi les pages web subissent diverses modifications.
- [\*\*Google Drive\*\*](#) : Un système de stockage et de partage de fichiers, avec mon cahier des charges, le code source à récupérer, etc. Tout ce qui est relatif aux informations pour réaliser le projet.
- [\*\*ChatGPT\*\*](#) : un agent conversationnel basé sur l'IA permettant de générer du texte à l'aide d'un modèle alimenté par une très grande quantité de données. Il m'a permis par exemple de complexifier certaines parties de mon code afin d'en augmenter sa sécurité ou de rédiger quelques mentions obligatoires.

### **3 - Missions**

Mes différentes missions, bien que non clairement énoncées, étaient très simples à récapituler :

- Réaliser un site vitrine qui permet de présenter le cabinet dentaire dans les grandes lignes.
- Inclure tout le contenu (textuel et multimédia) nécessaire qui me sera envoyé.
- Mettre en place un espace d'administration où on peut gérer les administrateurs en mode CRUD (Create, Read, Update, Delete). Cela permet à terme de pouvoir aisément modifier les informations des administrateurs (pseudonymes, mots de passe, etc.) aussi ergonomicement et facilement que possible.
- Pour les administrateurs, inclure la possibilité de modifier un calendrier qui pourra être ainsi visualisé sur la page d'accueil pour savoir les horaires proposés par le cabinet.
- Bien évidemment, utiliser l'hébergement web pour transférer tout le travail que j'aurais réalisé avec en plus la base de données en veillant à inclure les sécurités les plus basiques pour les formulaires (afin d'éviter notamment les attaques XSS ou les injections SQL, par exemple).
- Faire en sorte que le site web finisse par être référencé sur le moteur de recherche Google.

Au fil de la lecture de ce rapport de stage, vous verrez comment je me suis débrouillé pour remplir les tâches énoncées ci-dessus.

## 4 - Suivi hebdomadaire

### A - Semaine 1 (du 15 janvier au 19 janvier)

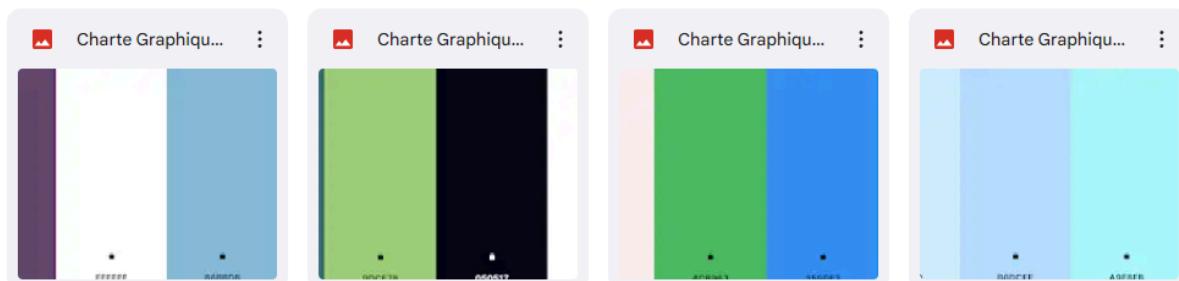
Pour le point de départ du projet, j'ai commencé par imaginer le squelette global du site. Je souhaitais partir sur le modèle MVC (Modèle-Vue-Contrôleur), mais afin d'éviter une structure de code trop alambiquée qui risquerait plus de me ralentir qu'autre chose, j'ai préféré rester sur quelque chose de simple.

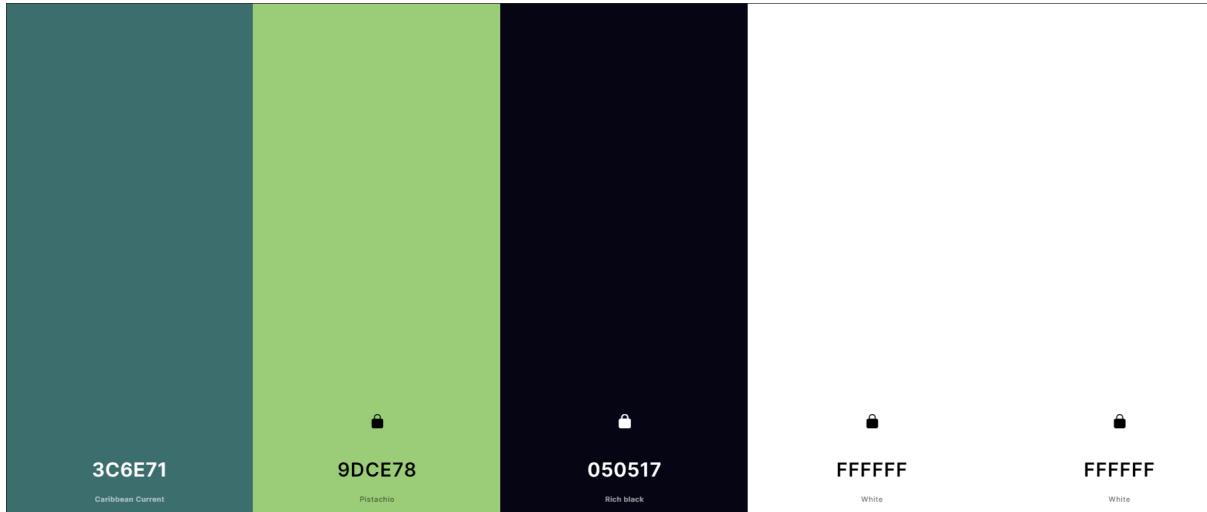
En premier lieu, j'ai pensé le site web pour être composé d'un petit nombre de pages, tout comme l'était [FarInWest](#). Pour ainsi dire, nous pouvons même considérer le site web du cabinet dentaire Dumitrascu Roxana comme une version grandement améliorée de celui de mon premier stage. C'est dans cette optique-là que j'ai décidé de concevoir un menu qui permettrait d'accéder à des pages pour obtenir davantage d'informations.



*La capture d'écran parle d'elle-même. A gauche, le nom du site avec le logo (cliquer dessus permet de revenir à la page d'accueil). A droite, d'autres pages accessibles pour en savoir plus sur le cabinet dentaire ou contacter celui-ci.*

Pour ma maître de stage, j'ai généré différentes palettes de couleurs en demandant de choisir laquelle elle préférait (sachant que pour évoquer le côté « médical », avoir du vert serait fortement appréciable), c'est la seconde qui fut choisie.





*Voici la charte graphique qui a été retenue. Personnellement, ce n'est pas celle que je préfère, mais qu'importe. Ce n'est pas un site web pour moi, mais pour d'autres personnes.*

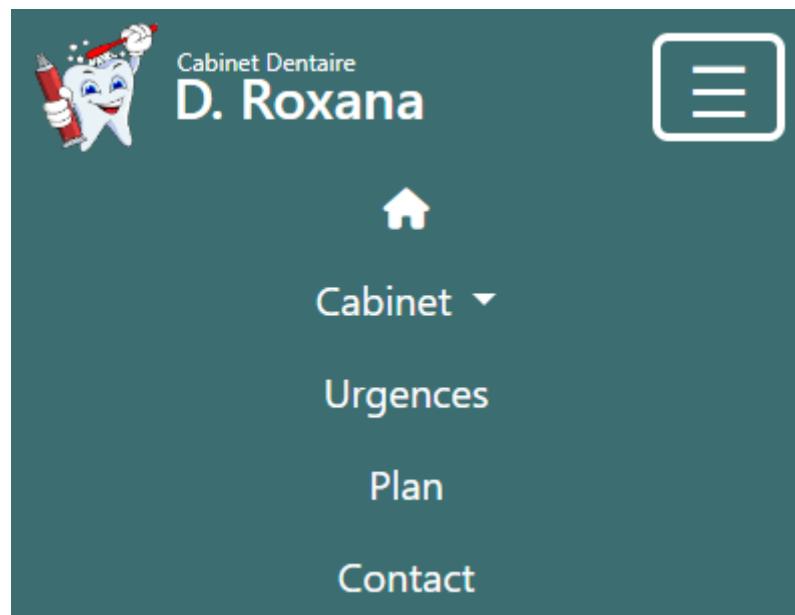
*En tout cas, vous retrouverez ces quatre couleurs sur le site web.*

A noter que cette fois, je ne suis parti d'aucun template. Je voulais coder l'apparence du site de A à Z sans avoir un modèle qui risquerait potentiellement de me brider si je devais faire des modifications. Tout comme mon premier stage, j'ai réutilisé Bootstrap qui est une boîte à outils sacrément utile en ce qui s'agit de concevoir un site esthétique sans écrire et gérer tout un code CSS.

```
<button class="navbar-toggler custom-toggler custom-toggler-text" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDropdown" aria-controls="navbarNavDropdown">
<div class="collapse navbar-collapse" id="navbarNavDropdown">
  <ul style="font-size:20px;" class="navbar-nav ms-auto text-center custom-nav">
    <li class="nav-item">
      <a class="nav-link text-white" href="index.php"><i class="fa-solid fa-house"></i></a>
    </li>
    <li class="nav-item d-none d-lg-inline">
      <span class="nav-bar-separator"></span>
    </li>
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle text-white" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">
        Cabinet
      </a>
      <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
        <li><a class="dropdown-item text-white haut-sous-menu" href="cabinet#conseils">Conseils</a></li>
        <li><a class="dropdown-item text-white haut-sous-menu" href="cabinet#traitements">Traitements</a></li>
        <li><a class="dropdown-item text-white bas-sous-menu" href="cabinet#equipements">Equipements</a></li>
      </ul>
    </li>
  </ul>
</div>
```

*Voici une petite partie du code du menu qu'on retrouve sur le site final. Le bouton (button) de classe "navbar-toggler" permet dans le cas où l'utilisateur utilise un petit appareil (les téléphones et tablettes, notamment) de pouvoir afficher le menu et le fermer en s'adapter aux dimensions de l'écran.*

Affichage en cas d'une grande résolution d'écran.



```
<script>
$(document).ready(function() {
    // Fonction pour vérifier la taille de l'écran
    function checkScreenWidth() {
        var smallScreen = window.matchMedia("(max-width: 460.02px)").matches;

        if (smallScreen) {
            $("#roxanaText").text("D. Roxana");
        } else {
            $("#roxanaText").text("Dumitrascu Roxana");
        }

        // Afficher l'élément une fois que le texte est mis à jour
        $("#roxanaText").removeClass("hidden-title");
    }

    // Vérifier la taille de l'écran lors du chargement de la page et lors du redimensionnement
    checkScreenWidth();
    $(window).resize(checkScreenWidth);
});
</script>
```

Affichage en cas d'une petite résolution d'écran. Le menu déroulant permet d'afficher ou masquer le menu pour le confort de l'utilisateur.

Le code en jQuery présenté permet de « raccourcir » le nom du site afin d'éviter qu'il ne force le menu déroulant à passer en-dessous, ce qui produirait un résultat très inesthétique.

Je n'avais pas encore l'intégralité du contenu à implémenter dans le site. Néanmoins, je pouvais déjà aisément imaginer ce qu'il contiendrait (et aussi car j'échange fréquemment avec ma maître de stage, ce qui évite les imprévus). Ainsi, j'ai préparé un certain espace

pour introduire succinctement le cabinet, le personnel gérant celui-ci et le futur tableau pour les horaires affichés.

# Bienvenue au cabinet dentaire Dumitrascu Roxana

## SITE EN CONSTRUCTION !

**Le cabinet dentaire** est situé à proximité du lycée René. **Le plateau technique** est composé de :

- 2 salles de soins
- 1 stérilisation centrale (thermodésinfecteur / ensachage / autoclaves)
- 1 radio panoramique numérique 2D
- 1 Caméra d'empreinte numérique

Le Docteur Dumitrascu soigne les enfants à partir de 8 ans quand si possible dans le cas contraire nous vous adresserons chez des pédodontistes.

Se rendre chez un chirurgien dentiste pour des soins dentaires peut être source d'appréhension chez de nombreuses personnes, inquiètes des actes à effectuer en elles-mêmes ou encore du prix.

Un bilan complet effectué afin de vous proposer un programme de soins sur mesure. Cette approche globale nous permet non seulement de vous proposer des soins adaptés qui répondront à vos attentes, mais aussi de vous rassurer et de vous proposer, au besoin, une entente financière de type paiement en plusieurs fois pour que vous puissiez accéder aux soins dont vous avez besoin. Surtout, nous vous proposons un partenariat sur le long terme, avec un suivi régulier pour une santé buccodentaire durable. Loin des soins ponctuels réalisés au coup par coup, nous vous informons dès le début pour mieux vous conseiller et vous soigner. Votre capital santé représente un investissement pour l'avenir, et nous souhaitons vous accompagner pour que vous puissiez en devenir acteur.



Une équipe du tonnerre qui vous redonnera le sourire !

### Dumitrascu Roxana

Chirurgienne Dentiste

Le docteur Dumitrascu Roxana exerce depuis 11 ans en France en tant que chirurgien dentiste. Elle est diplômée depuis 1993.  
Le Docteur Dumitrascu est chirurgien dentiste conventionné et applique les honoraires réglementés par l'assurance maladie.



### Justine Garcin

Assistante Dentaire

Depuis 2012, Justine Garcin est assistante dentaire au sein du cabinet dentaire. Elle l'aide à ce que vos dents soient les plus saines possibles.



## Horaire

18 – 22 mars 2024

Semaine actuelle



	lun. 18/03	mar. 19/03	mer. 20/03	jeu. 21/03	ven. 22/03
07 h					
08 h					
09 h	9:00 - 18:00	9:00 - 16:30	9:00 - 18:00	9:00 - 16:30	
10 h					
11 h					
12 h					
13 h	Ouverture	Ouverture	Ouverture	Ouverture	
14 h					
15 h					
16 h					
17 h					
18 h					

Pour finir avec de bonnes bases, j'ai donc mis en place un footer qui permettra d'accéder à des informations plus secondaires (tout ce qui est en rapport avec le législatif, les crédits ou encore le plan du site).

### LOCALISATION

26 Rue Rudolph Serkin  
84000 Avignon

### PLAN DU SITE

[Voir le plan du site](#)

### DIVERS

[Mentions légales](#)  
[Politique de confidentialité](#)  
[Cookies](#)  
[Crédits](#)

Copyright © 2024 Cabinet Dentaire Dumitrascu Roxana Tous droits réservés. Site réalisé par [Terence Renard](#)

Afin d'afficher le menu (header) et le bas de page (footer) à chaque fois, j'ai séparé ces parties en un fichier chacun qu'on n'a plus qu'à appeler sur chaque page afin de les retrouver sans devoir copier-coller de gros morceaux de code.

```

<?php
$metaRobots = '<meta name="robots" content="index, follow">';
$title = "Contact";
$metaDescription = "Le formulaire de contact du site web Cabinet Dentaire Dumitrascu Roxana";
require_once ('Include/Menu/header.php');
?>

<?php
require_once ('Include/Menu/footer.php');
?>

```

Tout simplement, des lignes de code à mettre respectivement tout au début et tout à la fin. A noter que j'initialise les variables \$metaRobots, \$title et \$metaDescription avant même d'appeler le header. Cela permet de définir le titre, la description et si la page doit être référencée ou non.

```

<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,initial-scale=1,shrink-to-fit=no">
    <?php echo $metaRobots . "\n" ?>
    <meta name="description" content="<?php echo $metaDescription ?>" />
    <link rel="stylesheet" href="https://unpkg.com/aos@next/dist-aos.css" />
    <link rel="icon" type="image/x-icon" href="images/logo.png" />
    <?php $site = " - Cabinet Dentaire Dumitrascu Roxana" ?>
    <title><?php echo $title . $site; ?></title>
    <link href="css/ajout.css" rel="stylesheet" />
    <link href="css/bootstrap.css" rel="stylesheet" />
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/css/bootstrap.min.css" rel="stylesheet" >
    <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>

```

Par exemple, pour la page contact, on demande explicitement de l'indexer et la suivre ("index, follow"), que son titre soit "Contact" suivi du nom du site web (ce qui donne pour la page en question "Contact - Cabinet Dentaire Dumitrascu Roxana") ainsi que la description de la page.

À la fin de cette première semaine, nous avons mis en place une architecture solide pour le site web. Cela nous permettra d'éviter les tracas futurs et de passer rapidement aux prochaines étapes du projet.

## B - Semaine 2 (du 22 janvier au 26 janvier)

C'est à partir de cette seconde semaine que j'ai décidé de commencer à m'atteler au back-end du site. Le back-end peut être divisé globalement en trois tâches majeures :

- La réalisation du formulaire de contact qui, s'il est correctement rempli, envoie un mail directement dans le service de messagerie (Gmail) du cabinet dentaire.
- Un espace pour gérer les administrateurs qui auront un compte. On pourra voir leurs informations (sauf les mots de passe, bien entendu), en ajouter des nouveaux, modifier les informations ou les supprimer (CRUD).
- Une page pour tout ce qui est relatif à la gestion du calendrier, ajouter une ou plusieurs horaires d'un coup (le cabinet dentaire possède des horaires fixes dont il

serait utile de pouvoir en mettre plusieurs d'un coup pour tel ou tel jour selon telle ou telle date), etc.

Commençons par créer une base de données avec deux tables. Une table pour stocker les informations des administrateurs, et une autre qui sera intégrée en clé étrangère pour choisir son rôle (bien que "Administrateur" soit le seul choix et qu'il ne soit pas prévu d'en ajouter d'autres).

<a href="#">id</a>	<a href="#">email</a>	<a href="#">mdp</a>	<a href="#">pseudo</a>	<a href="#">nom</a>	<a href="#">prenom</a>	<a href="#">idRole</a>
1				Renard	Terence	1

Cette table contient diverses informations sur l'administrateur. Son email et son pseudo (on peut utiliser l'un ou l'autre pour se connecter), un mot de passe (haché, cela va de soi avec la fonction PHP [password\\_hash\(\)](#)), l'identifiant de son rôle (1 = Administrateur) ainsi que son nom et prénom.

Créons enfin un formulaire pour se connecter, celui-ci n'est pas explicitement marqué sur le site, mais on peut y accéder en ajoutant "login(.php)" à l'URL du site, ou bien cliquer sur le lien en question sur le plan du site.

Pseudo ou adresse email\* :

Mot de passe\* :

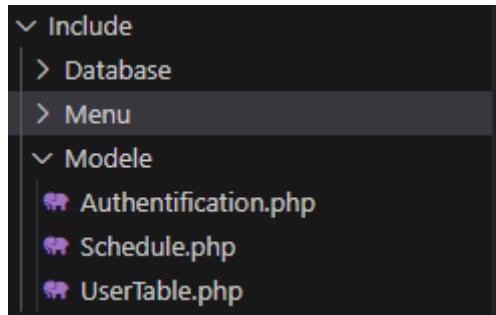
Envoyer

protection par reCAPTCHA  
Confidentialité - Conditions

Voici le formulaire de connexion, très basique et fonctionnel, à noter la présence très discrète du [reCAPTCHA v3 de Google](#) qui permet de vérifier si celui qui remplit le formulaire est un bot ou un humain selon les "actions" qu'il réalise sur le site. Il se base sur un score de confiance oscillant entre 0 (est considéré comme un bot) et 1 (est considéré comme un humain). Si le score est inférieur à 0.5, il est refusé.

On peut noter que j'ai inclus un petit bouton à côté du mot de passe permettant de vérifier son mot de passe au lieu qu'il soit masqué. C'est pas grand-chose, mais c'est un certain confort (surtout que la création d'un administrateur impose une certaine exigence dans le mot de passe).

L'authentification de l'utilisateur est traité dans un fichier se trouvant dans le dossier "Modele" se situant lui-même dans un dossier "Include" comme en témoigne la capture d'écran ci-dessous :



*Dans le cas du login, c'est le fichier "Authentification.php" qui se charge de traiter les données remplies.*

[reCAPTCHA v3](#) fonctionne sur un système de clé publique et de clé privée afin d'éviter un potentiel trucage lors du processus de validation du captcha. La partie « clé publique » est gérée avec du code JavaScript, tandis que celle « clé privée » avec du PHP (d'où le fait qu'elle soit privée). Si la vérification entre les deux ne peut pas se faire, cela échouera quoiqu'il arrive.

Une fois cette première étape passée, on vérifie si les informations envoyées via le formulaire sont valides avec la base de données (dont on aura préalablement connecté le site avec celle-ci dans "Include/Database"). Si tout est en ordre, une session se crée pour l'utilisateur connecté et il peut accéder à de nouvelles fonctionnalités, sinon, il devra recommencer.

```

private function authenticateUser($pseudo, $mdp) {
    // Préparer et exécuter la requête SQL sécurisée
    $query = "SELECT id, email, pseudo, mdp, idRole FROM user WHERE pseudo = ? OR email = ?";
    $statement = mysqli_prepare($this->connexion, $query);

    // Vérifier si la préparation de la requête a réussi
    if (!$statement) {
        exit("Erreur de préparation de la requête : " . mysqli_error($this->connexion));
    }

    mysqli_stmt_bind_param($statement, "ss", $pseudo, $pseudo);
    mysqli_stmt_execute($statement);
    mysqli_stmt_store_result($statement);

    // Vérifier le résultat de la requête
    if (mysqli_stmt_num_rows($statement) > 0) {
        // Associer les colonnes du résultat à des variables
        mysqli_stmt_bind_result($statement, $id, $pseudo, $pseudo, $hashedPassword, $idRole);

        // Récupérer les valeurs
        mysqli_stmt_fetch($statement);

        // Vérifier le mot de passe haché
        if (password_verify($mdp, $hashedPassword)) {
            // Mot de passe correct, connecter l'utilisateur
            $_SESSION['id'] = $id;
            $_SESSION['pseudo'] = $pseudo;
            $_SESSION['idRole'] = $idRole;

            // Récupérer l'action de l'administrateur ainsi que son IP
            $action = "Connexion réussie";
            $this->ipRecup($_SERVER['REMOTE_ADDR'], $action);

            // Rediriger l'utilisateur vers la page d'accueil
            header('Location: index.php');
            exit();
        } else {
            // Mot de passe incorrect
            $authResult = "<p class='text-danger display-6 text-center'>Votre identifiant ou votre mot de passe sont erronés.</p>";
        }
    }
}

```

*Note : Si on peut voir écrit deux fois "\$pseudo" côté à côté, c'est car le nom du champ dans le formulaire est "pseudo". Or, on peut autant utiliser son pseudonyme que son adresse email pour se connecter.*

L'étape de la connexion passée, un petit changement dans le menu du site devrait se manifester. Effectivement, un espace d'administration se débloque permettant de nouvelles actions au sein du site.



*Un nouveau menu déroulant "ADMIN" en jaune apparaît, une fois qu'on clique dessus, trois options nous sont proposées : Gestion Administrateur (pour gérer les administrateurs du site en CRUD), Gestion du calendrier (pour gérer les horaires du cabinet dentaire) et simplement pouvoir se déconnecter.*

```

<?php if (isset($_SESSION['idRole']) == 1) {
    // Vérifier si l'utilisateur est connecté et admin ?
<li class="nav-item d-none d-lg-inline">
    <span class="nav-bar-separator"></span>
</li>
<li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle text-warning fw-bold" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">
        ADMIN
    </a>
    <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
        <li><a class="dropdown-item text-white haut-sous-menu" href="admin">Gestion Administrateur</a></li>
        <li><a class="dropdown-item text-white bas-sous-menu" href="calendrier-modif">Gestion du calendrier</a></li>
        <li><a class="dropdown-item text-white bas-sous-menu" href="logout">Se déconnecter</a></li>
    </ul>
</li>
</?php } ?>

```

*Il faut que l'idRole soit égal à 1 (Administrateur) pour voir ce changement, ce qui est le cas quand l'utilisateur est bien connecté.*

Rapidement, j'ai implémenté la fonctionnalité de pouvoir simplement se déconnecter, ce qui est facile à faire. Cliquer sur "Se déconnecter" renvoie l'administrateur vers une page "logout.php" qui détruit sa session, et qui renvoie immédiatement vers "deconnexion.php" afin de lui confirmer qu'il est bien déconnecté.

```

logout.php
1   <head>
2     <meta name="robots" content="noindex, nofollow">
3   </head>
4   <?php
5   // Démarrage de la session (sait-on jamais)
6   session_start();
7
8   // Détruire toutes les données de la session
9   session_destroy();
10
11  // Redirigez l'utilisateur vers la page de confirmation de déconnexion
12  header('Location: deconnexion.php');
13  exit();
14 ?>

<?php
$metaRobots = '<meta name="robots" content="noindex, nofollow">';
$title = "Déconnexion";
$metaDescription = "";
require_once ('Include/Menu/header.php');
?>

<header class="pistache-background pt-4 pb-1">
    <div class="container text-center">
        <h1 class="display-4 fw-bold">Vous êtes bien déconnecté.</h1>
    </div>
</header>

<?php
require_once ('Include/Menu/footer.php');
?>

```

*Rien qu'avec ces deux blocs de code dans deux fichiers distincts, on gère la déconnexion autant d'un point de vue ergonomique que technique.*

La semaine se termine doucement avec la préparation des pages "Vitrine" du site. Tout au long du rapport de stage, je ne détaillerai que relativement peu ce que j'ai pu faire d'un point de vue "Front-End". Car je n'estime pas que ce soit la chose la plus intéressante ou la plus complexe du projet. Cela ferait même assez « redondant » avec la réalisation de mon premier stage. Néanmoins, je parlerai de temps en temps de mes décisions en ce qui concerne l'apparence du site ou des choix de design.

Sur la page d'accueil (index.php), je pensais ajouter un carrousel (ces espèces de diaporama où on clique sur des flèches à gauche et à droite pour les faire défiler) pour les différents menus comme ce fut le cas pour FarInWest. Je suis néanmoins assez mitigé d'une telle intégration car je ne suis pas fan des carrousels pour deux raisons :

- Les informations proposées arrivent les unes après les autres, ce qui peut faire cliquer inutilement un utilisateur. Surtout si ce qui vient après ne l'intéresse toujours pas.
- Même si Bootstrap propose des fonctionnalités pour qu'on ait pas à toucher une seule seconde au JavaScript, son implémentation peut rester plutôt lourde. Je préfère me focaliser sur quelque chose de simple et léger qui ne risque pas de surcharger les ressources de l'appareil.

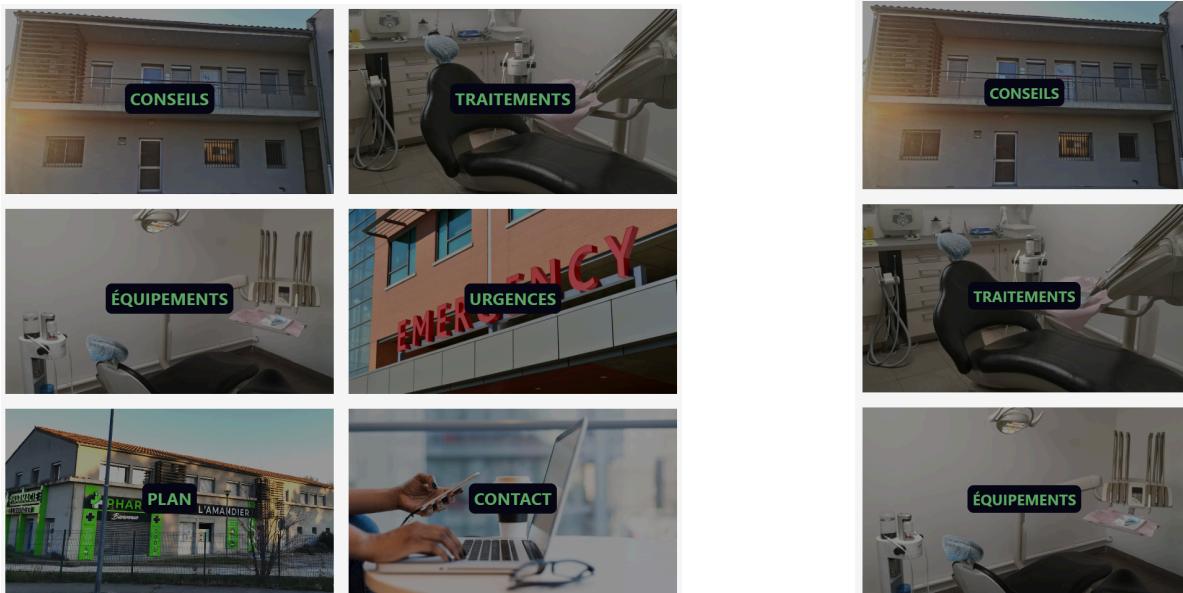
Donc à la place, j'ai choisi de mettre un petit ensemble d'images côté à côté avec du texte au centre de celle-ci pour indiquer immédiatement à l'utilisateur dans quelle page il se rendra.



*Lorsqu'on met le curseur sur une des images, le texte au centre disparaît, l'image zoomé légèrement tandis que l'overlay sombre se dissout. Cliquer dessus amène bien évidemment à une nouvelle page.*

Le choix de mettre très exactement six images n'est pas anodin. Afin de rendre la consultation des pages aussi agréables que possible qu'il importe la taille de l'écran (le Responsive, en somme). Il fallait afficher trois images par ligne pour un grand écran, deux images si celui-ci est moyen, et enfin, qu'une seule dans le cadre d'un smartphone. D'un

point de vue mathématique, on n'a qu'à dire qu'il faut un nombre qui soit un multiple de 1, de 2 et de 3. Et ce sont les nombres 6, 12, 18, etc. qui y répondent.



*Voici ce qu'on verra en regardant le site respectivement sur une tablette, ou sur un smartphone.*

```
<div class="container-fluid mt-5 data-aos="zoom-in">
<div class="row">

    <div class="col-xl-4 col-md-6 col-12 mb-4">
        <div class="image-container">
            <a href="#">
                
                <div class="overlay">
                    <h3 class="pistache-text fw-bold text-uppercase richblack-background-presentation">conseils</h3>
                </div>
            </a>
        </div>
    </div>

    <div class="col-xl-4 col-md-6 col-12 mb-4">
        <div class="image-container">
            <a href="#">
                
                <div class="overlay">
                    <h3 class="pistache-text fw-bold text-uppercase richblack-background-presentation">traitements</h3>
                </div>
            </a>
        </div>
    </div>

    <div class="col-xl-4 col-md-6 col-12 mb-4">
        <div class="image-container">
            <a href="#">
                
                <div class="overlay">
                    <h3 class="pistache-text fw-bold text-uppercase richblack-background-presentation">équipements</h3>
                </div>
            </a>
        </div>
    </div>

```

*Bootstrap simplifie ce qui est relatif à l'affichage des "div" selon la taille d'écran. Il faut imaginer que celui-ci divise une ligne en 12 colonnes. Ainsi, col-xl-4 pour les grands écrans ( $4*3 = 12$ ). col-md-6 pour les écrans moyens ( $6*2 = 12$ ). Et enfin, col-12 pour les petits.*

Nous pouvons achever cette seconde semaine en concluant que l'authentification fonctionne parfaitement et que le contenu du site se développe petit à petit.

### C - Semaine 3 (du 29 janvier au 2 février)

Si je devais faire un reproche récurrent avec les deux stages que j'ai pu faire. C'est le fait que je ne reçois pas le contenu à l'avance. Cela m'offre une liberté très prononcée en ce qui concerne l'apparence finale que prendra le site, mais cela s'accompagne aussi d'imprévus. Cela aboutit à des pages esthétiquement efficaces, mais franchement simples.

Pour contrebalancer ce défaut, j'ai trouvé une bibliothèque intéressante : [Animate On Scroll \(AOS\)](#) qui permet d'inclure de petites animations. Pour l'implémenter, il suffit simplement d'intégrer un lien externe d'un fichier CSS au début de la page, puis à la fin de celle-ci pour le JavaScript en plus de l'initialiser.

```
<link rel="stylesheet" href="https://unpkg.com/aos@next/dist-aos.css" />
<script src="https://unpkg.com/aos@next/dist-aos.js"></script>
<script>AOS.init();</script>
<div class="container-fluid mt-5" data-aos="zoom-in">
```

*Une fois cela fait, on a plus qu'à mettre dans les différentes balises dont on veut une animation spécifique "data-aos=" suivi de l'effet qu'on souhaite avoir, la page de présentation du projet montre divers exemples.*

Je n'en ai fait qu'un usage relativement restreint afin d'éviter un effet inverse où il y a tellement d'effets à l'écran que cela devient désorientant.

Maintenant, retournons sur le développement de la partie Back-End avec un gestionnaire des administrateurs. D'abord, commençons par réaliser un nouveau modèle qu'on appellera "UserTable.php". On connecte celle-ci avec la base de données associée.

```
class UserTable {
    private $connexion;

    public function __construct($connexion) {
        $this->connexion = $connexion;
    }
}
```

```

<?php
require_once("config.php");

// Créer une connexion
$connexion = mysqli_connect($server, $user, $password, $dbname);

// Vérifier la connexion
if (!$connexion) {
    die("La connexion à la base de données a échoué. Veuillez contacter l'administrateur du site.");
    header('Location: index.php');
}

// Définir l'encodage en UTF-8
mysqli_set_charset($connexion, "utf8");
?>

```

*\$connexion (une variable que le fichier peut utiliser grâce à l'inclusion du fichier "database.php" dans "header.php" qui, on le rappelle, est intégré à toutes les pages consultables) gère tout ce qui est relatif à la connexion avec la base de données, on n'a plus qu'à s'en servir dans les constructeurs de nos classes et méthodes.*

```

public function checkAdmin() {
    // Vérifier si l'utilisateur a le rôle nécessaire
    if (!isset($_SESSION['idRole']) || $_SESSION['idRole'] != 1) {
        header('Location: index.php');
        exit();
    }
}

```

*Logiquement, certaines pages doivent être protégées, la méthode "checkAdmin" qu'on retrouve sur tous les fichiers devant être protégés vérifie si l'utilisateur a le rôle nécessaire et si ce n'est pas le cas, il est ramené immédiatement à la page d'accueil.*

```

class UserTable {
    public function displayAdminUsers() {
        // Préparer et exécuter la requête SQL sécurisée
        $query = "SELECT id, email, pseudo, nom, prenom, idRole FROM user WHERE idRole = '1'";
        $statement = mysqli_prepare($this->connexion, $query);

        // Vérifier si la préparation de la requête a réussi
        if (!$statement) {
            die("Erreur de préparation de la requête : " . mysqli_error($this->connexion));
        }

        mysqli_stmt_execute($statement);
        mysqli_stmt_store_result($statement);

        // Lié les résultats à des variables
        mysqli_stmt_bind_result($statement, $id, $email, $pseudo, $nom, $prenom, $idRole);

        // Afficher le tableau
        echo "<section class='grey-background pt-2 pb-3'>
            <div class='container mt-3'>
                <div class='table-responsive'>
                    <table class='table text-center table-bordered bg-white'>
                        <thead class='thead-dark'>
                            <tr>
                                <th scope='col'>Pseudo</th>
                                <th scope='col'>Email</th>
                                <th scope='col'>Nom</th>
                                <th scope='col'>Prénom</th>
                                <th scope='col'>ID du Rôle</th>
                                <th scope='col'>Actions</th>
                            </tr>
                        </thead>
                        <tbody>";
        // Parcourir les résultats et afficher chaque ligne dans le tableau
        while (mysqli_stmt_fetch($statement)) {
            echo "<tr>
                <td>$pseudo</td>
                <td>$email</td>
                <td>$nom</td>
                <td>$prenom</td>
                <td>$idRole</td>
                <td>
                    <a href='admin-modifier.php?id=$id' class='btn btn-primary'>Modifier</a>
                    <a href='?action=delete&id=$id' class='btn btn-danger'>Supprimer</a>
                </td>
            </tr>";
        }
    }
}

```

La méthode `displayAdminUsers` affiche les administrateurs du site dans un tableau comme ci-dessous.

## Gestion des administrateurs

Pseudo	Email	Nom	Prénom	ID du Rôle	Actions	
		Renard	Terence	1	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
				1	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
		Dentiste	Dentiste	1	<a href="#">Modifier</a>	<a href="#">Supprimer</a>

```

public function deleteUser($id) {
    // Vérifier si l'utilisateur a le rôle nécessaire
    if (!isset($_SESSION['idRole']) || $_SESSION['idRole'] != 1) {
        header('Location: index.php');
        exit();
    }

    // Préparer et exécuter la requête SQL sécurisée pour supprimer l'utilisateur
    $query = "DELETE FROM user WHERE id = ?";
    $statement = $this->connexion->prepare($query);

    // Vérifier si la préparation de la requête a réussi
    if ($statement) {
        // Lier le paramètre de l'ID de l'utilisateur
        $statement->bind_param('i', $id);

        // Exécuter la requête
        $statement->execute();

        // Vérifier si l'exécution a réussi
        if ($statement->affected_rows > 0) {
            echo "<p class='text-success display-6 text-center'>Utilisateur supprimé avec succès !</p>";
        } else {
            echo "Aucun utilisateur trouvé avec cet ID.";
        }
    }

    // Récupérer l'action et l'IP de l'utilisateur qui fait une "action d'administrateur"
    $action = "Suppression réussie d'un administrateur";
    $this->ipRecup($_SERVER['REMOTE_ADDR'], $action);

    // Fermer le statement
    $statement->close();
} else {
    echo "Erreur de préparation de la requête : " . $this->connexion->error;
}
}

```

*La méthode deleteUser récupère l'identifiant du compte ciblé afin de le supprimer.*

```

class UserTable {
    public function addUser($email, $mdp, $pseudo, $prenom, $nom, $idRole) {
        // Vérifier si l'utilisateur a un mot de passe suffisamment puissant (min. 12 caractères + caractères spéciaux)
        if (strlen($mdp) < 12 || !preg_match('/^(?=.*[A-Z])(?=.*[\d])(?=.*[^\w\d])[\w]{12,}$/', $mdp)) {
            echo "<p class='text-danger display-6 text-center'>Votre mot de passe doit contenir au moins 12 caractères et des caractères spéciaux (exemple : _-/ etc.), un chiffre";
            return;
        }

        // Insérer un nouveau utilisateur
        $query = "INSERT INTO user (email, mdp, pseudo, nom, prenom, idRole) VALUES (?, ?, ?, ?, ?, ?)";

        // Me demandez pas pourquoi avec mysqli_prepare ça ne fonctionne pas, j'en ai aucune idée, donc j'ai fait un truc différent.
        $statement = $this->connexion->prepare($query);

        // Vérifie si la préparation de la requête a réussi
        if ($statement) {

            // Hasher le mot de passe avec la fonction password_hash de PHP pour plus de sécurité
            $hashedPassword = password_hash($mdp, PASSWORD_DEFAULT);

            // Lie les paramètres
            $statement->bind_param("sssssi", $email, $hashedPassword, $pseudo, $nom, $prenom, $idRole);

            // Exécute la requête
            $statement->execute();

            // Vérifie si l'exécution a réussi
            if ($statement->affected_rows > 0) {
                echo "<p class='text-success display-6 text-center'>Enregistrement réussi du nouvel utilisateur.</p>";
            } else {
                echo "<p class='text-danger display-6 text-center'>Échec de l'enregistrement du nouvel utilisateur.</p>";
            }
        }

        // Ferme le statement
        $statement->close();

        // Récupérer l'action et l'IP de l'utilisateur qui fait une "action d'administrateur"
        $action = "Ajout réussi d'un administrateur";
        $this->ipRecup($_SERVER['REMOTE_ADDR'], $action);
    } else {
        echo "Erreur de préparation de la requête : " . $this->connexion->error;
    }
}

```

*La méthode addUser permet d'ajouter un utilisateur dans la base de données. Comme on le constate, une certaine exigence est demandée pour le mot de passe.*

Email :	<input type="text"/>
Mot de passe :	<input type="password"/>
Pseudo :	<input type="text"/>
Prénom :	<input type="text"/>
Nom :	<input type="text"/>
Rôle :	<input type="text"/> Administrateur (1) <input checked="" type="checkbox"/>
<a href="#">Ajouter administrateur</a>	

*C'est ce formulaire qu'on trouve juste en-dessous du tableau qui permet de rajouter de nouveaux utilisateurs, et qui est traité par la méthode addUser.*

```
class UserTable {
    public function updateUser($id, $nouveauPseudo, $nouvelEmail, $nouveauNom, $nouveauPrenom, $nouveauIdRole, $nouveauMdp, $confirmerMdp) {
        // Vérifiez si le formulaire a été soumis
        if ($_SERVER['REQUEST_METHOD'] === 'POST') {
            // Vérifiez si l'utilisateur souhaite modifier le mot de passe
            if (!empty($nouveauMdp) || !empty($confirmerMdp)) {
                // Vérifiez la confirmation du mot de passe
                if ($nouveauMdp !== $confirmerMdp) {
                    $errors_update['verifmdtpropcourt'] = "<span style='color:red;'>Vos mots de passe sont différents.</span>";
                    return;
                } elseif (strlen($nouveauMdp) < 12 || !preg_match('/^(?=.*[A-Z])(?=.*[0-9])(?=.*[^A-Za-z\d])[\S]{12,}$/', $nouveauMdp)) {
                    $errors_update['verifmdpdifferent'] = "<span style='color:red;'>Votre mot de passe est très court.</span>";
                    return;
                } else {
                    // Hacher le mot de passe s'il est fourni
                    $hashedPassword = password_hash($nouveauMdp, PASSWORD_DEFAULT);

                    // Préparez et exécutez la requête de mise à jour avec changement de mot de passe
                    $query = "UPDATE user SET pseudo=?, email=?, nom=?, prenom=?, idRole=?, mdp=? WHERE id=?";
                }
            } else {
                // Préparez et exécutez la requête de mise à jour sans changer le mot de passe
                $query = "UPDATE user SET pseudo=?, email=?, nom=?, prenom=?, idRole=? WHERE id=?";
            }

            $statement = $this->connexion->prepare($query);

            // Vérifie si la préparation de la requête a réussi
            if ($statement) {
                // Lier les paramètres
                if (!empty($nouveauMdp)) {
                    $statement->bind_param("ssssisi", $nouveauPseudo, $nouvelEmail, $nouveauNom, $nouveauPrenom, $nouveauIdRole, $hashedPassword, $id);
                } else {
                    $statement->bind_param("ssssii", $nouveauPseudo, $nouvelEmail, $nouveauNom, $nouveauPrenom, $nouveauIdRole, $id);
                }

                // Exécute la requête
                $statement->execute();

                // Vérifie si l'exécution a réussi
                if ($statement->affected_rows > 0) {
                    if ($_SESSION['id'] == $id) {
                        // Mettez à jour les informations dans la session si l'utilisateur modifie son propre profil
                        $_SESSION['pseudo'] = $nouveauPseudo;
                        $_SESSION['idRole'] = $nouveauIdRole;
                    }
                }
            }
        }
    }
}
```

*Enfin, updateUser est la dernière méthode qui permet comme son nom l'indique, de modifier les informations de l'utilisateur, à noter que celle-ci gère si l'utilisateur veut changer de mot de passe (dans ce cas-là, on procède à une seconde vérification en lui demandant de le répéter) ou non (dans ce second cas, aucune mise à jour sur le mot de passe n'est réalisée).*

Cabinet Dentaire  
**Dumitrascu Roxana**

[Home](#) | Cabinet | Urgences | Plan | Contact | **ADMIN**

## Modification de l'admin

**Modifier l'utilisateur**

Pseudo :

Email :

Nom :  Renard

Prénom :  Terence

ID du Rôle :  1

Nouveau mot de passe :  Laissez vide pour ne pas changer

Confirmer le mot de passe :  Confirmez le mot de passe

**Enregistrer les modifications**

Tout cela se passe sur une autre page qui est "admin-modifier.php". On passe en paramètre "GET" l'identifiant de l'administrateur et on peut effectuer les changements qu'on veut faire. Une fois que tout est en ordre, on clique sur "Enregistrer les modifications" et le tableau montre les modifications effectuées.

Pseudo	Email	Nom	Prénom	ID du Rôle	Actions
<input type="text"/>		Renard	Terence Rapport de Stage	1	<a href="#">Modifier</a> <a href="#">Supprimer</a>

Juste avant de passer à autre chose, je souhaiterai m'exprimer sur un point particulier. Il y a globalement deux API qui sont utilisables sur PHP permettant d'interagir avec les bases de données : [PDO \(PHP Data Objects\)](#) et [MySQLi \(MySQL Improved\)](#). Étant donné qu'au départ du projet j'étais relativement peu expérimenté sur tout ce qui est à propos de l'interaction avec une base de données via le langage PHP, le hasard a fait que je me suis tourné vers MySQLi. Celle-ci a ses petites particularités que je vais essayer d'expliquer.

```

// Insérer un nouveau utilisateur
$query = "INSERT INTO user (email, mdp, pseudo, nom, prenom, idRole) VALUES ?, ?, ?, ?, ?, ?";
// Me demandez pas pourquoi avec mysqli_prepare ça ne fonctionne pas, j'en ai dû faire des tests, donc j'ai fait un truc différent
$statement = $this->connexion->prepare($query);

// Vérifie si la préparation de la requête a réussi
if ($statement) {

    // Hasher le mot de passe avec la fonction password_hash de PHP pour plus de sécurité
    $hashedPassword = password_hash($mdp, PASSWORD_DEFAULT);

    // Lie les paramètres
    $statement->bind_param("sssssi", $email, $hashedPassword, $pseudo, $nom, $prenom, $idRole);
}

```

Lorsque l'on utilise MySQLi en PHP pour insérer des valeurs dans une base de données, on utilise des points d'interrogation ("?") pour indiquer les emplacements des valeurs à insérer. Ensuite, pour spécifier quel champ correspond à quelle valeur, on utilise "bind\_param".

Avant d'utiliser bind\_param, il est nécessaire de préciser si la valeur à insérer est une chaîne de caractères (string) ou un nombre (integer). Par exemple, si l'on voit "sssssi", cela signifie simplement que les cinq premières variables à insérer, de \$email à \$prenom, sont des chaînes de caractères (d'où les "s" pour string), tandis que le dernier, l'identifiant du rôle, est un nombre (d'où le "i" pour integer).

Pour faire fonctionner toutes ces méthodes, rien de plus simple. Il suffit de se rendre dans la bonne page (admin.php ou admin-modifier.php) et d'appeler les méthodes selon les interactions effectuées :

```

<?php
// Créer une instance de la classe userTable avec la connexion à la base de données
$userTable = new UserTable($connexion);

// Vérifier si l'utilisateur est bien un administrateur
$userTable->checkAdmin();

// Vérifier si l'ID de l'admin à supprimer est présent dans l'URL
if (isset($_GET['action']) && $_GET['action'] == 'delete' && isset($_GET['id'])) {
    $id = $_GET['id'];

    // Appeler la méthode deleteUser pour supprimer un admin précis
    $userTable->deleteUser($id);
}

// Vérifier si le formulaire a été soumis
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Récupérer les données du formulaire
    $email = htmlspecialchars($_POST['email']);
    $mdp = htmlspecialchars($_POST['mdp']);
    $pseudo = htmlspecialchars($_POST['pseudo']);
    $prenom = htmlspecialchars($_POST['prenom']);
    $nom = htmlspecialchars($_POST['nom']);
    $idRole = htmlspecialchars($_POST['idRole']);

    // Appeler la méthode addUser pour ajouter un administrateur
    $userTable->addUser($email, $mdp, $pseudo, $prenom, $nom, $idRole);
}

// Appeler la méthode displayAdminUsers pour afficher les administrateurs
$userTable->displayAdminUsers();
?>

```

Les commentaires parlent d'eux-mêmes, on initialise une instance pour la classe avec une connexion avec la base de données, puis on appelle la bonne méthode selon ce qui se passe.

Maintenant, tout est nickel, le gestionnaire des administrateurs est fonctionnel. Passons au formulaire de contact. Pour celui-ci, j'ai récupéré ce que j'avais déjà fait avec [FarInWest](#) en utilisant la bibliothèque [PHPMailer](#).

```
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require 'ressources/PHPMailer-6.6.5/src/Exception.php';
require 'ressources/PHPMailer-6.6.5/src/PHPMailer.php';
require 'ressources/PHPMailer-6.6.5/src/SMTP.php';
```

*Une fois le dossier récupéré, on fait appel à ce dont on a besoin.*

En premier lieu, l'email du cabinet dentaire a pour service de messagerie Gmail. Celui-ci propose une fonctionnalité permettant de recevoir des mails via des applications... étonnamment bien dissimulée !

The screenshot shows the Google Account settings page. The search bar at the top contains the text "mots de passe des applications". Below it, a sidebar lists "Accueil" and "Sécurité". The main content area is titled "Mots de passe des applications" under the heading "28 RÉSULTATS". It displays two entries: "Cabinet Dentaire" (created on Jan 18, last used on Feb 22) and "Test" (created on Jan 18). Below these, there is a form to "Pour créer un mot de passe spécifique à une appli, indiquez son nom ci-dessous." with a "Nom de l'appli" input field and a "Créer" button.

*Il faut se rendre dans les paramètres de son compte Gmail, chercher « Mots de passe des applications » et entrer un nom d'application, ce qui va faire apparaître un mot de passe de seize caractères qu'on intégrera dans les paramètres de notre formulaire de contact.*

```

function myMail($dest, $subject, $body) {
    $mail = new PHPMailer(true);

    try {
        $mail->isSMTP();
        $mail->Host = 'smtp.gmail.com'; // Remplacez par l'adresse de votre serveur SMTP
        $mail->SMTPAuth = true;
        $mail->Username = '██████████'; // Remplacez par votre nom d'utilisateur SMTP
        $mail->Password = '██████████'; // Remplacez par votre mot de passe SMTP
        $mail->SMTPSecure = 'ssl'; // Ne surtout pas mettre en majuscule... genre vraiment.
        $mail->Port = 465; // Généralement 587, mais ça peut varier

        // Récupérer les informations du formulaire
        $nom = htmlspecialchars($_POST['nom']);
        $prenom = htmlspecialchars($_POST['prenom']);
        $email = htmlspecialchars($_POST['email']);
        $telephone = htmlspecialchars($_POST['telephone']);
        $motif = htmlspecialchars($_POST['motif']);
        $message = htmlspecialchars($_POST['message']);
    }
}

```

*Pour Username, mettez votre adresse email complète, et pour Password, le mot de passe qui vous a été généré (s'il y a des espaces, retirez-les).*

```

// Inclure les informations du formulaire dans l'objet `full_message`
$full_message = "<b>Nom :</b> " . $nom . "<br>" .
    "<b>Prénom :</b> " . $prenom . "<br>" .
    "<b>Email :</b> " . $email . "<br>";
// Ajouter l'âge s'il est renseigné
if (!empty($age)) {
    $full_message .= "<b>Âge :</b> " . $age . " ans<br>";
}
$full_message .= "<b>Téléphone :</b> " . $telephone . "<br>" .
    "<b>Motif :</b> " . $motif . "<br>" .
    "<b>Message :</b> " . $message;

$mail->CharSet = 'UTF-8'; // Définir l'encodage des caractères

// Ajouter le destinataire par défaut
$mail->addAddress('██████████');

// Ajouter le destinataire saisi par l'utilisateur
$mail->addAddress($dest);

$mail->addReplyTo('██████████', $nom . ' ' . $prenom);
$mail->Subject = $subject;
$mail->isHTML(true); // Définir le contenu du message comme HTML
$mail->MsgHTML($full_message); // Utiliser la méthode MsgHTML pour conserver les caractères spéciaux et les sauts de ligne

$mail->send();
// echo 'Le message a été envoyé avec succès.';
} catch (Exception $e) {
    // echo 'Une erreur est survenue lors de l'envoi du message : ' . $mail->ErrorInfo;
}
}

```

*On configure tout ce qui est relatif à l'envoi du mail : L'adresse email qui la recevra (le Gmail du cabinet dentaire, donc), la mise en forme de l'email (très basique, je ne me suis pas pris la tête sur ce point), etc.*

# Formulaire de contact

Pour toute information, veuillez compléter ce formulaire afin de prendre contact avec le cabinet dentaire Dumitrascu Roxana !

Nom\* :

Prénom\* :

Email :

Téléphone :

Âge :  25 ans  
 Ne pas renseigner

Sujet\* :

Motif :   
(Non applicable)

Message\* :  
Maximum de 1500 caractères.  
 En envoyant ce formulaire, vous acceptez qu'on recueille les données envoyées afin de vous identifier et vous répondre.

Envoyer

\* = Champ obligatoire à remplir

*Le formulaire de contact du cabinet dentaire est bien plus complet que celui de [FarInWest](#). Beaucoup de champs sont néanmoins optionnels.*

C'est pour le rapport de stage ➔ Boîte de réception ✉



*Voici le genre de résultat qu'on reçoit par mail. Ah, et j'ai aussi implémenté de nouveau [reCAPTCHA v3](#), cela permet d'éviter les spams de bots.*

Pour « récolter » le consentement, rien de plus simple que de créer une nouvelle table "consentement" où on récupère l'email (si renseigné), le prénom, le nom et le moment exact où la personne remplit le formulaire.

	<input type="button" value="←"/>	<input type="button" value="→"/>		id	consent	email	prenom	nom	date_enregistrement
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="Supprimer"/>	14	Oui	<input type="text" value=""/>	Terence	Renard	2024-01-24 01:57:29
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="Supprimer"/>	15	Oui	<input type="text" value=""/>	Terence	Renard	2024-01-24 02:14:19

```
$consentement = "INSERT INTO consentement (consent, email, prenom, nom, date_enregistrement) VALUES ('Oui', ?, ?, ?, NOW());"
```

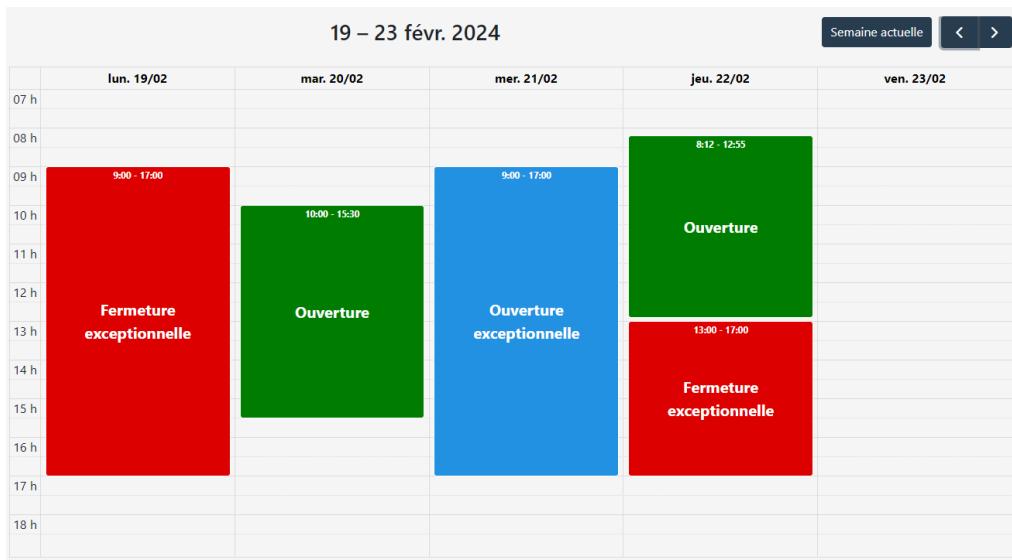
Mine de rien, pour cette troisième semaine, j'avançais sacrément rapidement, de quoi me rassurer concernant l'éventualité regrettable que je ne puisse pas intégrer les principales fonctionnalités avant la deadline (la fin de la 6ème semaine). Il ne me restait globalement plus qu'un gros morceau à attaquer : Le calendrier.

## D - Semaine 4 (du 5 février au 9 février)

J'avais fait quelques recherches d'avance sur comment je pourrais aborder une telle tâche. Il faut dire que je reste malgré tout un développeur débutant et que la réalisation d'un calendrier était quelque peu... intimidante.

Une solution ressortait régulièrement, utiliser la bibliothèque [FullCalendar](#) qui propose ce dont j'ai besoin et qui possède une certaine flexibilité sur les possibilités et paramètres qu'on peut choisir.

Une fois la bibliothèque "importée" via une ligne de code, j'ai pu commencer à m'en servir, nous allons voir étape par étape ce qui a permis d'aboutir à un calendrier comme celui en-dessous.



```
<script src='https://cdn.jsdelivr.net/npm/fullcalendar@6.1.10/index.global.min.js'></script>
```

On va d'abord configurer FullCalendar pour que le tableau avec les horaires soient fidèles à ce qu'on souhaite retrouver comme résultat final.

```
document.addEventListener('DOMContentLoaded', function() {
var calendarEl = document.getElementById('calendar');
var calendar = new FullCalendar.Calendar(calendarEl, {
    locale: 'fr',
    initialView: 'timeGridWeek',
    headerToolbar: {
        start: '',
        center: 'title',
        end: 'today prev,next'
    },
    views: {
        timeGridWeek: {
            type: 'timeGrid',
            firstDay: 1 // Définir le premier jour de la semaine comme Lundi
        }
    },
    buttonText: {
        today: 'Semaine actuelle' // Changer le texte du bouton "today" par "Semaine actuelle"
    },
    weekends: false, // Permettre de cacher le Samedi & Dimanche
    editable: false, // Permettre l'édition des événements sur le calendrier
    selectable: true, // Permettre de sélectionner des plages horaires pour ajouter des événements
    allDaySlot: false, // Désactiver le all-day qui ne sert à rien
    nowIndicator: true, // Montre la ligne rouge afin de montrer le moment actuel
    slotMinTime: "07:00:00", // Visibilité du calendrier qui commence à 7 heures
    slotMaxTime: "19:00:00", // Visibilité du calendrier qui termine à 19 heures
});
```

*Notre calendrier sera configuré en français (locale), on veut que celui-ci soit hebdomadaire (initialView) et que le premier jour soit un Lundi (firstDay: 1, sinon cela commence à 0, et c'est un Dimanche).*

*Ensuite, on veut que le bouton "today" (qui remet le calendrier à la semaine en cours) change pour "Semaine actuelle". On ne veut pas le week-end (weekends: false), que les événements ne soient pas éditables (beaucoup trop complexe pour un rien à implémenter) mais sélectionnables (ce sera utile pour les supprimer si on le souhaite). Enfin, les horaires s'affichent de 7 heures à 19 heures (le cabinet dentaire ne travaille pas la nuit).*

Pour gérer l'intégralité des horaires qui seront ajoutés et supprimés au fil du temps, une nouvelle table s'impose qu'on va appeler "horaires". Elle sera constituée d'une clé primaire pour identifier chaque horaire indépendamment afin de les gérer, la date, l'horaire de début et l'horaire de fin.

	<b>id</b>	<b>nom</b>	<b>date_complet</b>	<b>heure_debut</b>	<b>heure_fin</b>	
	65	Ouverture	2024-02-05	09:00:00	18:00:00	
	73	Ouverture exceptionnelle	2024-02-06	09:00:00	13:00:00	
	77	Ouverture	2024-02-07	08:00:00	12:00:00	
	83	Ouverture exceptionnelle	2024-02-09	13:00:00	17:00:00	
	84	Fermeture exceptionnelle	2024-02-08	10:00:00	15:00:00	*

Pour exploiter les données, on va intégrer celles-ci à FullCalendar. Pour cela, on crée un nouveau modèle qu'on nommera "Schedule.php" avec diverses méthodes permettant d'arriver à nos fins.

```
class Schedule {
    /**
     * Récupère tous les horaires depuis la base de données
     */
    public function getAllSchedules() {
        $query = "SELECT * FROM horaires";
        $result = $this->connexion->query($query);

        $schedules = array();

        if ($result->num_rows > 0) {
            while ($row = $result->fetch_assoc()) {
                $schedules[] = $row;
            }
        }

        return $schedules;
    }
}
```

Après avoir géré la connexion avec la base de données (une fois encore avec la variable \$connexion), on récupère l'intégralité des horaires et on les exploite ainsi dans le calendrier.

```
// Créer une instance de la classe Schedule avec la connexion à la base de données
$schedule = new Schedule($connexion);

// Vérifier si l'utilisateur est bien un administrateur
$schedule->checkAdmin();
?>

events: [
    <?php
    // Récupérer tous les événements depuis la base de données
    $events = $schedule->getAllSchedules();

    // Générer les événements pour le calendrier à partir des données de la base de données
    foreach ($events as $event) {
        echo '[';
        echo 'title: "' . $event['nom'] . '",'; // Nom de l'événement
        echo 'start: "' . $event['date_complet'] . 'T' . $event['heure_debut'] . '",'; // Heure de début
        echo 'end: "' . $event['date_complet'] . 'T' . $event['heure_fin'] . '",'; // Heure de fin
        echo 'id: "' . $event['id'] . '"'; // ID de l'événement
    }
]
```

FullCalendar appelle chaque horaire des "événements" (events). On a d'abord le titre de celui-ci (qui se limite à « Ouverture », « Ouverture exceptionnelle »), « Fermeture exceptionnelle » et « Congés »), le début (start), la fin (end) puis enfin l'identifiant (id).

Ces informations seront traitées sous le format .json et peuvent ainsi être « lues » par le calendrier.

```
events: [
    {title: "Ouverture", start: "2024-02-05T09:00:00", end: "2024-02-05T18:00:00", id: "65", color: "green",}
```

*Cette horaire a comme titre "Ouverture", elle commence le 5 Février à 9 heures et se termine le même jour à 18 heures, son identifiant est "65" et sa couleur est verte (on en parle un peu plus tard).*



*Et voilà le résultat.*

Pour ajouter facilement de nouveaux horaires, un formulaire s'impose dans la page "calendrier-modif.php".

```

<!-- Calendrier -->
<section class="mb-2 pb-3 grey-background" id="calendrier-vue">
  <div class="container-fluid">
    <div class="row">
      <!-- Aperçu du calendrier -->
      <div class="col-xl-8 col-12 mt-4">
        <div id="calendar"></div>
      </div>

      <!-- Formulaire d'ajout d'horaires dans le calendrier -->
      <div class="col-xl-4 col-12 mt-4 border pb-2 align-items-center">
        <h2 class="mb-3 text-center text-uppercase fw-bold">Ajouter une horaire</h2>
        <form action="calendrier-modif.php?action=ajouter" class="fw-bold" method="POST">
          <div class="form-group">
            <label style="font-size:20px;" for="nom">Nom :</label>
            <select class="form-select fw-bold text-white" style="font-size:20px;" id="nom" name="nom" required onchange="changeCouleur(this)">
              <option style="background-color:green; font-size:20px;" class="fw-bold" value="Ouverture">Ouverture</option>
              <option style="background-color:#2794e1; font-size:20px;" class="fw-bold" value="Ouverture exceptionnelle">Ouverture exceptionnelle</option>
              <option style="background-color:#dc0000; font-size:20px;" class="fw-bold" value="Fermeture exceptionnelle">Fermeture exceptionnelle</option>
              <option style="background-color:purple; font-size:20px;" class="fw-bold" value="Congés">Congés</option>
            </select>
          </div>
          <div class="form-group">
            <label style="font-size:20px;" for="date_complet">Date :</label>
            <input type="date" name="date_complet" id="date" class="form-control" required>
          </div>
          <div class="row">
            <div class="form-group col-6">
              <label style="font-size:20px;" for="heure_debut">Heure de début :</label>
              <input type="time" name="heure_debut" id="heure_debut" class="form-control" required>
            </div>
            <div class="form-group col-6">
              <label style="font-size:20px;" for="heure_fin">Heure de fin :</label>
              <input type="time" name="heure_fin" id="heure_fin" class="form-control" required>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>
</section>

```

Voici l'interface d'ajout des horaires, simple et efficace, avec une vue immédiate sur le calendrier pour un maximum de confort.

```

class Schedule {
    /**
     * Ajoute un horaire à la base de données
     */
    public function addSchedule($nom, $date_complet, $heure_debut, $heure_fin) {
        // Vérifier si l'utilisateur a le rôle nécessaire
        if (!isset($_SESSION['idRole']) || $_SESSION['idRole'] != 1) {
            header('Location: index.php');
            exit();
        }

        $query = "INSERT INTO horaires (nom, date_complet, heure_debut, heure_fin) VALUES (?, ?, ?, ?)";
        if ($heure_debut >= $heure_fin) {
            return false;
        } else {
            $statement = $this->connexion->prepare($query);
            $statement->bind_param("ssss", $nom, $date_complet, $heure_debut, $heure_fin);
            if ($statement->execute()) {
                // Récupérer l'action de l'administrateur ainsi que son IP
                $action = "Ajout d'une horaire réussie";
                $this->ipRecup($_SERVER['REMOTE_ADDR'], $action);

                return true;
            } else {
                return false;
            }
        }
    }
}

```

*La méthode addSchedule gère les insertions des nouveaux horaires.*

```

if (isset($_GET['action'])) {
    $action = $_GET['action'];

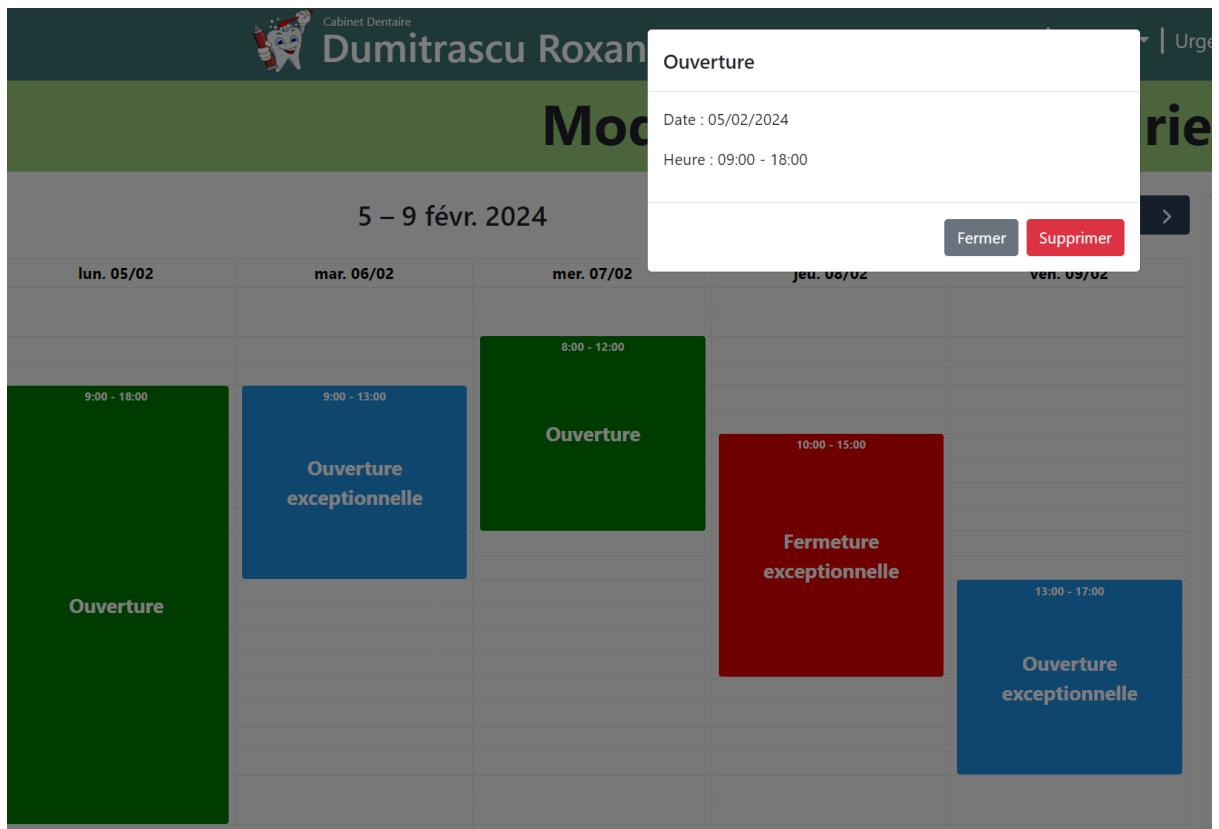
    // En fonction de l'action demandée, effectuer les opérations appropriées
    switch ($action) {
        case 'ajouter':
            // Traitement de l'ajout d'un nouvel événement
            if (isset($_POST['nom'], $_POST['date_complet'], $_POST['heure_debut'], $_POST['heure_fin'])) {
                $nom = htmlspecialchars($_POST['nom']);
                $date_complet = htmlspecialchars($_POST['date_complet']);
                $heure_debut = htmlspecialchars($_POST['heure_debut']);
                $heure_fin = htmlspecialchars($_POST['heure_fin']);

                $schedule->addSchedule($nom, $date_complet, $heure_debut, $heure_fin); // Appel de la méthode addSchedule
                header("Location: calendrier-modif.php");
            }
            break;
    }
}

```

*Le formulaire a comme action "ajouter", on récupère alors tous les champs pour les traiter dans la méthode juste au-dessus. Il y a un total de quatre actions dont je présenterai l'utilité de chacune au fil de ce rapport de stage.*

Pour supprimer un horaire, c'est très simple : Vous cliquez sur celle que vous souhaitez retirer, cela ouvre une fenêtre modale qui propose un bouton pour la supprimer.



```

case 'supprimer':
    // Traitement de la suppression d'un événement existant
    if (isset($_GET['id'])) {
        $id = $_GET['id'];

        $schedule->deleteSchedule($id); // Appel de la méthode deleteSchedule
        header("Location: calendrier-modif.php");
    }
    break;

```

```

class Schedule {
    public function deleteSchedule($id) {
        // Vérifier si l'utilisateur a le rôle nécessaire
        if (!isset($_SESSION['idRole']) || $_SESSION['idRole'] != 1) {
            header('Location: index.php');
            exit();
        }

        $query = "DELETE FROM horaires WHERE id=?";
        $statement = $this->connexion->prepare($query);
        $statement->bind_param("i", $id);
        if ($statement->execute()) {
            // Récupérer l'action de l'administrateur ainsi que son IP
            $action = "Suppression d'une horaire réussie";
            $this->ipRecup($_SERVER['REMOTE_ADDR'], $action);

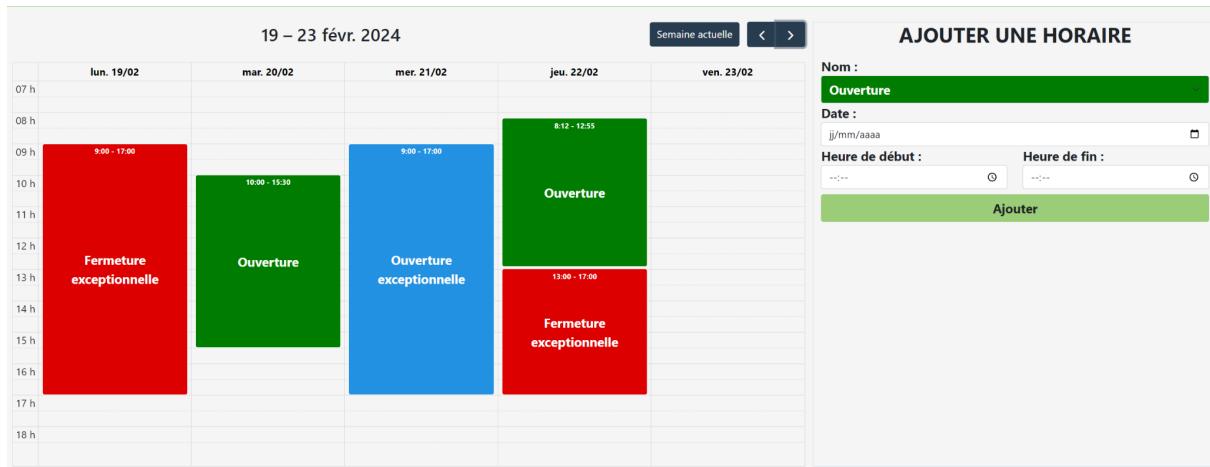
            return true;
        } else {
            return false;
        }
    }
}

```

Je me suis rapidement occupé du code pour gérer l'apparence que prendra les différents horaires du calendrier. Il s'agit simplement d'un peu de CSS et de JavaScript.

```
if ($event['nom'] === "Ouverture") { // Gestion des couleurs selon le nom de l'événement
    echo 'color: "green",';
} elseif ($event['nom'] === "Ouverture exceptionnelle") {
    echo 'color: "#2794e1",';
} elseif ($event['nom'] === "Fermeture exceptionnelle") {
    echo 'color: "#dc0000",';
} elseif ($event['nom'] === "Congés") {
    echo 'color: "purple",';
}
echo '\n';
```

*En JavaScript, selon le nom (titre) de l'horaire, la couleur diffère.*



*L'interface d'administration ressemble alors à ça, accueillante et facile d'utilisation, j'estime avoir fait du bon travail.*

Je finis alors cette quatrième semaine, fier d'avoir fait tout ce qui était essentiel pour le site. Les deux semaines restantes me permettront de peaufiner certains détails.

## E - Semaine 5 (du 12 février au 16 février)

La cabinet dentaire a des horaires fixes :

- Ouverture le Lundi de 9h à 18h.
- Ouverture le Mardi de 9h à 16h30.
- Ouverture le Mercredi de 9h à 18h.
- Ouverture le Jeudi de 9h à 16h30.
- Fermé le Vendredi.

Si je raconte cela, ce n'est pas anodin. Il fallait que je trouve une solution pour ajouter tous ces horaires disons jusqu'à... fin 2026 ? Terrible problème : Bon courage pour écrire une requête SQL qui puisse me défaire de ce travail répétitif si on devait mettre les horaires un

par un. Une idée m'est alors venue : Utiliser ChatGPT en lui décrivant mon problème afin qu'il me produise une requête SQL très poussée permettant d'arriver à un tel résultat.

```

INSERT INTO horaires (nom, date_complet, heure_debut, heure_fin)
SELECT 'Ouverture', dates.date_complet,
CASE
    WHEN DAYOFWEEK(dates.date_complet) = 2 THEN '09:00:00' -- Lundi
    WHEN DAYOFWEEK(dates.date_complet) = 3 THEN '09:00:00' -- Mardi
    WHEN DAYOFWEEK(dates.date_complet) = 4 THEN '09:00:00' -- Mercredi
    WHEN DAYOFWEEK(dates.date_complet) = 5 THEN '09:00:00' -- Jeudi
    ELSE NULL -- Autres jours
END AS heure_debut,
CASE
    WHEN DAYOFWEEK(dates.date_complet) IN (2, 4) THEN '18:00:00' -- Lundi et Mercredi
    WHEN DAYOFWEEK(dates.date_complet) IN (3, 5) THEN '16:30:00' -- Mardi et Jeudi
    ELSE NULL -- Autres jours
END AS heure_fin
FROM (
    SELECT DATE_ADD('2024-01-01', INTERVAL (t4*10000 + t3*1000 + t2*100 + t1*10 + t0) DAY) AS date_complet
    FROM
        (SELECT 0 t0 UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) t1
        (SELECT 0 t1 UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) t2
        (SELECT 0 t2 UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) t3
        (SELECT 0 t3 UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) t4
) dates
WHERE dates.date_complet BETWEEN '2024-01-01' AND '2025-12-31';

```

*Cette... horreur sortie tout droit des Enfers est littéralement parfaite. Une fois exécutée dans l'interface de phpMyAdmin, cela implémente des centaines de lignes qui font le résultat voulu.*

<b>id</b>	<b>nom</b>	<b>date_complet</b>	<b>heure_debut</b>	<b>heure_fin</b>
233	Ouverture	2024-04-05	00:00:00	00:00:00
234	Ouverture	2024-04-06	00:00:00	00:00:00
235	Ouverture	2024-04-07	00:00:00	00:00:00
236	Ouverture	2024-04-08	09:00:00	18:00:00
237	Ouverture	2024-04-09	09:00:00	16:30:00
238	Ouverture	2024-03-21	09:00:00	16:30:00
239	Ouverture	2024-03-22	00:00:00	00:00:00
240	Ouverture	2024-03-23	00:00:00	00:00:00

*Faisons un petit ménage en supprimant toutes les lignes avec des horaires "00:00:00" (ce sont les jours de Vendredi à Dimanche où le cabinet est évidemment fermé).*

✓ 672 lignes supprimées. (traitement en 0,0032 seconde(s).)

`DELETE FROM horaires WHERE heure_debut = "00:00:00";`

8 – 12 avr. 2024				
	lun. 08/04	mar. 09/04	mer. 10/04	jeu. 11/04
07 h				
08 h				
09 h	9:00 - 18:00	9:00 - 16:30	9:00 - 18:00	9:00 - 16:30
10 h				
11 h		Ouverture		
12 h				
13 h	Ouverture		Ouverture	
14 h				
15 h				
16 h				
17 h				
18 h				

Nickel.

Il m'est alors venu l'idée de pouvoir faire cela directement dans la page "calendrier-modif.php" avec d'autres formulaires, je me suis alors mis au travail pour cela.

**ATTENTION ! Les formulaires en-dessous doivent être remplies sérieusement car pouvant potentiellement affecter de nombreuses horaires !**

**AJOUTER PLUSIEURS HORAIRES**

Nom :

Heure de début (Lundi) :  Heure de début (Mardi) :  Heure de début (Mercredi) :   
Heure de fin (Lundi) :  Heure de fin (Mardi) :  Heure de fin (Mercredi) :   
Heure de début (Jeudi) :  Heure de début (Vendredi) :  Date de début :   
Heure de fin (Jeudi) :  Heure de fin (Vendredi) :  Date de fin :

Un sacré formulaire qui permet de définir les horaires (début et fin) de Lundi à Vendredi (on ne remplit rien si on ne veut pas inscrire « d'événement » selon le jour de la semaine).

```

class Schedule {
    function addMultipleSchedule($type, $heure_debut_lundi, $heure_debut_mardi, $heure_debut_mercredi, $heure_debut_jeudi, $heure_debut_vendredi, $heure_fin_lundi, $heure_fin_mardi, $heure_fin_mercredi, $heure_fin_jeudi, $heure_fin_vendredi) {
        // Requête SQL pour insérer les horaires d'ouverture
        $query = "
        INSERT INTO horaires (nom, date_complet, heure_debut, heure_fin)
        SELECT ?, dates.date_complet,
        CASE
            WHEN DAYOFWEEK(dates.date_complet) = 2 THEN ? -- Lundi
            WHEN DAYOFWEEK(dates.date_complet) = 3 THEN ? -- Mardi
            WHEN DAYOFWEEK(dates.date_complet) = 4 THEN ? -- Mercredi
            WHEN DAYOFWEEK(dates.date_complet) = 5 THEN ? -- Jeudi
            WHEN DAYOFWEEK(dates.date_complet) = 6 THEN ? -- Vendredi
        END AS heure_debut,
        CASE
            WHEN DAYOFWEEK(dates.date_complet) = 2 THEN ? -- Lundi
            WHEN DAYOFWEEK(dates.date_complet) = 3 THEN ? -- Mardi
            WHEN DAYOFWEEK(dates.date_complet) = 4 THEN ? -- Mercredi
            WHEN DAYOFWEEK(dates.date_complet) = 5 THEN ? -- Jeudi
            WHEN DAYOFWEEK(dates.date_complet) = 6 THEN ? -- Vendredi
        END AS heure_fin
        FROM (
            SELECT DATE_ADD(?, INTERVAL (t4*10000 + t3*1000 + t2*100 + t1*10 + t0) DAY) AS date_complet
            FROM
                (SELECT 0 t0 UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION
                (SELECT 0 t1 UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION
                (SELECT 0 t2 UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION
                (SELECT 0 t3 UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION
                (SELECT 0 t4 UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION
            ) dates
        WHERE dates.date_complet BETWEEN ? AND ?;

        // Préparer la requête
        $statement = mysqli_prepare($this->connexion, $query);

        // Vérifier si la préparation de la requête a réussi
        if (!$statement) {
            die("Erreur de préparation de la requête : " . mysqli_error($connexion));
        }

        // Binder les valeurs
        mysqli_stmt_bind_param($statement, "ssssssssssss", $type, $heure_debut_lundi, $heure_debut_mardi, $heure_debut_mercredi, $heure_debut_jeudi, $heure_debut_vendredi, $heure_fin_lundi, $heure_fin_mardi, $heure_fin_mercredi, $heure_fin_jeudi, $heure_fin_vendredi);
        // Exécuter la requête
        mysqli_stmt_execute($statement);
    }
}

case 'ajouter_plein':
    // Traitement de l'ajout de plusieurs horaires
    if (isset($_POST['type'], $_POST['heure_debut_lundi'], $_POST['heure_debut_mardi'], $_POST['heure_debut_mercredi'], $_POST['heure_debut_jeudi'], $_POST['heure_debut_vendredi'], $_POST['heure_fin_lundi'], $_POST['heure_fin_mardi'], $_POST['heure_fin_mercredi'], $_POST['heure_fin_jeudi'], $_POST['heure_fin_vendredi'], $_POST['date_debut'], $_POST['date_fin'])) {
        $schedule->addMultipleSchedule($type, $heure_debut_lundi, $heure_debut_mardi, $heure_debut_mercredi, $heure_debut_jeudi, $heure_debut_vendredi, $heure_fin_lundi, $heure_fin_mardi, $heure_fin_mercredi, $heure_fin_jeudi, $heure_fin_vendredi, $date_debut, $date_fin);
        header("Location: calendrier-modif.php");
    }
    break;
}

```

*Concours du monde de la plus grande requête SQL. 13 champs sont pris en compte pour ajouter de multiples horaires en un seul envoi de formulaire.*

En conséquence de la possibilité d'ajouter de nouveaux horaires en masse, il faudrait aussi pouvoir les supprimer en masse, créons alors une ultime méthode pour les supprimer entre deux dates :

## SUPPRIMER TOUTES LES HORAIRES ENTRE DEUX DATES

Supprimer les horaires du... :  au :

```
class Schedule {
    public function deleteMultipleSchedule($date_debut, $date_fin) {
        // Vérifier si l'utilisateur a le rôle nécessaire
        if (!isset($_SESSION['idRole']) || $_SESSION['idRole'] != 1) {
            header('Location: index.php');
            exit();
        }

        $query = "DELETE FROM horaires WHERE date_complet BETWEEN ? AND ?";
        $statement = $this->connexion->prepare($query);
        $statement->bind_param("ss", $date_debut, $date_fin);

        if ($statement->execute()) {
            // Récupérer l'action de l'administrateur ainsi que son IP
            $action = "Suppression de plusieurs horaires réussies";
            $this->ipRecup($_SERVER['REMOTE_ADDR'], $action);

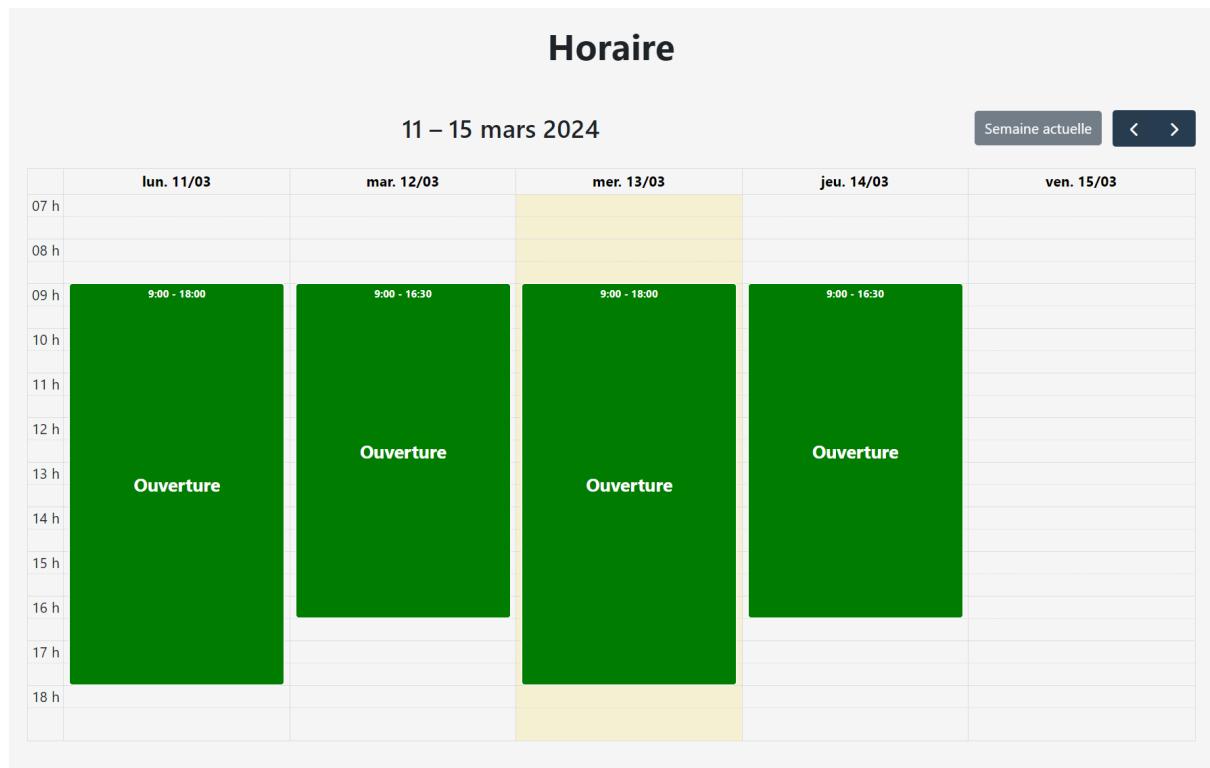
            return true;
        } else {
            return false;
        }
    }

    case 'explosion':
        // Traitement de la suppression de plusieurs horaires
        if (isset($_POST['date_debut'], $_POST['date_fin'])) {
            $date_debut = htmlspecialchars($_POST['date_debut']);
            $date_fin = htmlspecialchars($_POST['date_fin']);

            $schedule->deleteMultipleSchedule($date_debut, $date_fin);
            header("Location: calendrier-modif.php");
        }
    }
}
```

*Une simple requête SQL est exécutée après l'envoi du formulaire dont l'action est "explosion", supprimant tous les horaires entre deux dates.*

La gestion du calendrier est alors totalement terminée, il ne reste qu'à implémenter la vue sur la page d'accueil et le tour est joué.



L'utilisateur lambda peut donc visualiser quand le cabinet est ouvert, fermé, etc. en bas de la page d'accueil.

Pour récapituler, voici ce que peut faire un administrateur du site et un simple visiteur (non connecté) par rapport au calendrier :

	Visualiser les horaires	Ajouter un horaire	Supprimer un horaire	Ajouter plusieurs horaires	Supprimer plusieurs horaires
Visiteur	OUI	NON	NON	NON	NON
Administrateur	OUI	OUI	OUI	OUI	OUI

Si vous avez été attentif à la lecture de mes méthodes, vous pourrez remarquer qu'un élément récurrent se retrouve sur celles-ci :

```
// Récupérer l'action et l'IP de l'utilisateur qui fait une "action d'administrateur"
$action = "Modification réussie d'un administrateur";
$this->ipRecup($_SERVER['REMOTE_ADDR'], $action);

// Récupérer l'action de l'administrateur ainsi que son IP
$action = "Suppression de plusieurs horaires réussies";
$this->ipRecup($_SERVER['REMOTE_ADDR'], $action);

// Récupérer l'action de l'administrateur ainsi que son IP
$action = "Connexion réussie";
$this->ipRecup($_SERVER['REMOTE_ADDR'], $action);
```

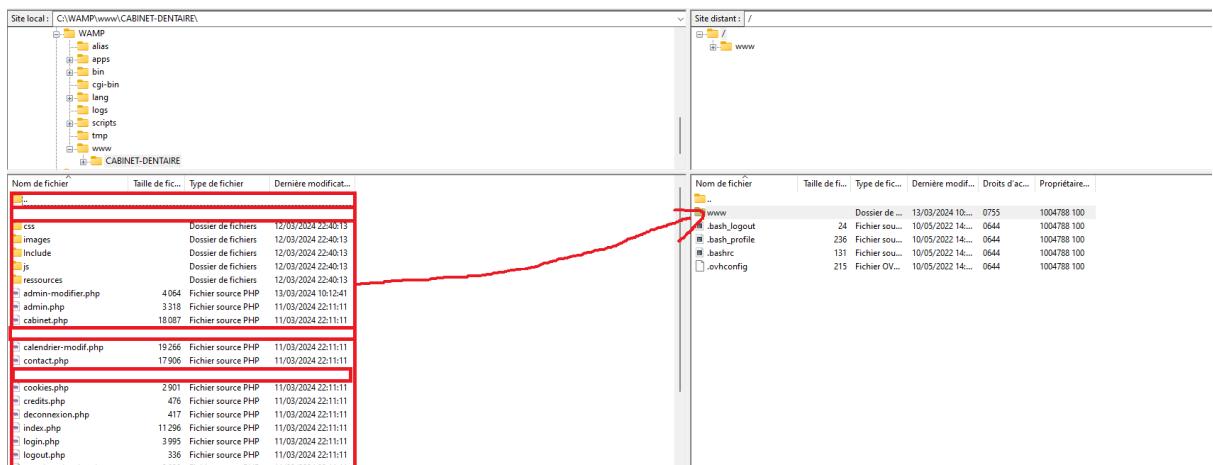
Effectivement, pour chaque modèle, une nouvelle méthode a été implémentée : ipRecup qui permet comme son nom le sous-entend, de récupérer l'IP d'une personne en cas d'action qu'uniquement un administrateur est supposé faire (pas d'inquiétude alors, l'IP n'est pas récupéré pour le formulaire de contact).

	←	→		id	action	date	ip	
<input type="checkbox"/>		<a href="#">Éditer</a>		<a href="#">Copier</a>		<a href="#">Supprimer</a>	12 Suppression réussie d'un administrateur	2024-02-18 04:15:26 127.0.0.1
<input type="checkbox"/>		<a href="#">Éditer</a>		<a href="#">Copier</a>		<a href="#">Supprimer</a>	13 Suppression réussie d'un administrateur	2024-02-18 04:26:11 127.0.0.1
<input type="checkbox"/>		<a href="#">Éditer</a>		<a href="#">Copier</a>		<a href="#">Supprimer</a>	14 Suppression réussie d'un administrateur	2024-02-18 04:26:12 127.0.0.1
<input type="checkbox"/>		<a href="#">Éditer</a>		<a href="#">Copier</a>		<a href="#">Supprimer</a>	15 Ajout d'une horaire réussie	2024-02-18 04:26:24 127.0.0.1
<input type="checkbox"/>		<a href="#">Éditer</a>		<a href="#">Copier</a>		<a href="#">Supprimer</a>	16 Connexion réussie	2024-02-18 04:44:46 127.0.0.1
<input type="checkbox"/>		<a href="#">Éditer</a>		<a href="#">Copier</a>		<a href="#">Supprimer</a>	17 Ajout d'une horaire réussie	2024-02-18 06:02:53 127.0.0.1
<input type="checkbox"/>		<a href="#">Éditer</a>		<a href="#">Copier</a>		<a href="#">Supprimer</a>	18 Suppression d'une horaire réussie	2024-02-18 06:02:59 127.0.0.1

*La cinquième et dernière table de notre base de données, "action", est un registre des actions d'administration qui se sont déroulées. Cela peut s'avérer très pratique pour surveiller ce qui se passe en cas de cyberattaque (si l'IP est 127.0.0.1, pas d'inquiétude, c'est normal lorsqu'on fait des tests en local).*

## **F - Semaine 6 (du 19 février au 23 février)**

La semaine la plus tranquille de mon stage. Le dernier objectif majeur est de tout implémenter vers l'hébergeur qui est enfin fonctionnel ! J'ai donc reçu un mail de la personne s'étant chargé d'activer l'hébergement via OVHcloud avec toutes les informations relatives pour mener à bien ma tâche (connexion via un client FTP (FileZilla), importer la base de données vers l'hébergeur, etc.).



*On met les pages .php et les dossiers à l'intérieur du dossier "www" de l'hébergeur.*

A l'intérieur du dossier "www", on peut trouver un fichier .htaccess qui permet la réécriture des liens, il est très simple et n'offre que la fonctionnalité de retirer les .php à la fin des liens. Je l'ai légèrement amélioré pour ainsi proposer une redirection vers une page 404 si

l'utilisateur se perd lors de sa visite sur le site, et interdire l'indexation sur les moteurs de recherche certaines pages.

```
# Interdire l'indexation (et l'accessibilité publique) des fichiers spécifiés
<Files "admin.php">
    <IfModule mod_headers.c>
        | Header set X-Robots-Tag "noindex, nofollow"
    </IfModule>
</Files>

<Files "admin-modifier.php">
    <IfModule mod_headers.c>
        | Header set X-Robots-Tag "noindex, nofollow"
    </IfModule>
</Files>

<Files "calendrier.php">
    <IfModule mod_headers.c>
        | Header set X-Robots-Tag "noindex, nofollow"
    </IfModule>
</Files>

<Files "login.php">
    <IfModule mod_headers.c>
        | Header set X-Robots-Tag "noindex, nofollow"
    </IfModule>
</Files>

<Files "logout.php">
    <IfModule mod_headers.c>
        | Header set X-Robots-Tag "noindex, nofollow"
    </IfModule>
</Files>

<Files "deconnexion.php">
    <IfModule mod_headers.c>
        | Header set X-Robots-Tag "noindex, nofollow"
    </IfModule>
</Files>
```

ErrorDocument 404 /page404.php

The screenshot shows a 404 error page. At the top, there's a navigation bar with a logo, the text 'Cabinet Dentaire Dumitrascu Roxana', and links for 'Accès', 'Cabinet', 'Urgences', 'Plan', and 'Contact'. Below the navigation is a large red header with the text 'ERREUR 404 :('. Underneath, a message in French says 'La page que vous recherchez n'existe pas, ou a été supprimée.' followed by 'Pas de panique, nous vous renvoyons à la page d'accueil dans 3 seconde(s) !'. At the bottom, it says 'La page 404 en question.'

On modifie le fichier "config.php" dans le dossier "Include/Database" pour remplacer les paramètres du localhost par ceux de l'hébergeur web. Et le site web est enfin consultable : <https://dentiste-dumitrascu.fr/>.

Ultime étape désormais : Créer un sitemap pour référencer les différentes pages du site avec [Google Search Console](#). Pour cela, j'ai utilisé [Screaming Frog](#) qui permet de "crawler" un site afin d'en détecter les pages et, par exemple, de générer un sitemap en format .xml, ce qu'on souhaite faire.

Fichier Vue Mode Configuration Exportation en bloc Rapports Sitemaps Visualisations Analyse du crawl Licence Aide

Screamingfrog https://dentiste-dumitrascu.fr/ Sous-domaine Commencer Vider

Interne Externe Sécurité Codes de réponse URL Title des pages Meta descr Vue d'ensemble Problèmes Structure du site Segments Temps de réponse AF

Tous Exporter

Adresse	
1 https://dentiste-dumitrascu.fr/	
2 https://dentiste-dumitrascu.fr/politique	
3 https://dentiste-dumitrascu.fr/mentions-legales	
4 https://dentiste-dumitrascu.fr/images/contact_image.jpg	
5 https://dentiste-dumitrascu.fr/css/bootstrap.css	
6 https://dentiste-dumitrascu.fr/plan	
7 https://dentiste-dumitrascu.fr/images/logo.png	
8 https://dentiste-dumitrascu.fr/urgences	

Cellules sélectionnées : 0 Filtrer total : 28

Tous	URL	% du total
HTML	12	42,86%
JavaScript	0	0%
CSS	2	7,14%
Images	14	50%
PDF	0	0%
Flash	0	0%
Autre	0	0%
Inconnu	0	0%

Exporter Rechercher... Interne

Nom Valeur

Aucune URL sélectionnée

Cellules sélectionnées : 0 Total : 0

Détails de l'URL Liens entrants Liens sortants Détails de l'image Ressources

Mode Spider: Inactif Moyenne : 31,97 URL Actuel : 31,97 URL 40 sur 40 terminé (100 %) 0 restant

Pour crawler un site et en récupérer un sitemap en .xml, rentrez d'URL du site web en question et cliquez sur "Commencer". Il ne reste plus qu'à cliquer sur l'onglet "Sitemaps" -> Sitemap XML -> Exporter, et on obtient ce que l'on souhaite.



Nom de fichier	Taille de fi...	Type de fic...	Dernière modif...	Droits d'ac...	Propriétaire...
js	1220	Fichier HT...	20/02/2024 18...	0604	1004788 100
www	3 961	Fichier sou...	13/03/2024 10...	0604	1004788 100
css	3 231	Fichier sou...	04/03/2024 18...	0604	1004788 100
images	17 904	Fichier sou...	04/03/2024 18...	0604	1004788 100
include	18 896	Fichier sou...	04/03/2024 18...	0604	1004788 100
js	17 500	Fichier sou...	04/03/2024 18...	0604	1004788 100
ressources	2 851	Fichier sou...	04/03/2024 18...	0604	1004788 100
.htaccess	461	Fichier sou...	04/03/2024 18...	0604	1004788 100
admin-modifier.php	402	Fichier sou...	04/03/2024 18...	0604	1004788 100
admin.php	11 067	Fichier sou...	04/03/2024 18...	0604	1004788 100
cabinet.php	3 897	Fichier sou...	04/03/2024 18...	0604	1004788 100
calendrier-modif.php	323	Fichier sou...	04/03/2024 18...	0604	1004788 100
contact.php	2 851	Fichier sou...	04/03/2024 18...	0604	1004788 100
cookies.php	461	Fichier sou...	04/03/2024 18...	0604	1004788 100
credits.php	402	Fichier sou...	04/03/2024 18...	0604	1004788 100
deconnection.php	1 293	Fichier sou...	04/03/2024 18...	0604	1004788 100
index.php	2 896	Fichier sou...	04/03/2024 18...	0604	1004788 100
login.php	2 990	Fichier sou...	04/03/2024 18...	0604	1004788 100
logout.php	1 293	Fichier sou...	04/03/2024 18...	0604	1004788 100
mentions-legales.php	1 293	Fichier sou...	04/03/2024 18...	0604	1004788 100
page404.php	2 896	Fichier sou...	04/03/2024 18...	0604	1004788 100
plan.php	4 925	Fichier sou...	04/03/2024 18...	0604	1004788 100
politique.php	1 849	Fichier XML	04/03/2024 15...	0604	1004788 100
sitemap.xml	1 197	Fichier sou...	04/03/2024 18...	0604	1004788 100
urgences.php					

On place le sitemap.xml à la racine du site.

Sitemap	Type	URL envoyées	Dernière lecture	État	Pages découvertes	Vidéos découvertes
<a href="https://dentiste-dumitrascu.fr/sitemap.xml">https://dentiste-dumitrascu.fr/sitemap.xml</a>	Sitemap	2 mars 2024	13 mars 2024	Opération effectuée	10	0
<a href="https://dentiste-dumitrascu.fr/sitemap.xml">https://dentiste-dumitrascu.fr/sitemap.xml</a>	Sitemap	2 mars 2024	10 mars 2024	Opération effectuée	10	0

*Une fois qu'on a ajouté la propriété du site web dans Google Search Console, il ne reste plus qu'à aller sur l'onglet "Sitemaps" et mettre l'URL complet du chemin vers le bon sitemap (<https://dentiste-dumitrascu.fr/sitemap.xml>), vous constaterez d'ailleurs que les pages qui ne doivent pas être référencées ne sont effectivement pas inscrites).*

*Nous constatons qu'à l'heure où j'écris ces lignes, le site web a bien commencé à être référencé (il s'agit d'un processus qui peut demander jusqu'à plusieurs semaines, ce qui fait qu'il est recommandable de s'occuper du référencement le plus tôt possible).*

Enfin, j'implémente tout le contenu restant (il reste deux-trois petits points à régler, surtout de l'ordre des mentions légales, ce n'est donc pas la mer à boire en somme) et je procède à quelques vérifications de l'ordre de la cybersécurité. Nous pouvons alors considérer le stage comme terminé.

## 5 - Apports du stage

Mon stage s'est avéré être une période d'apprentissage relativement agréable et très diversifiée, propice à l'acquisition de compétences clés dans le domaine du développement web. Parmi les nombreux bénéfices tirés de cette expérience, plusieurs points méritent d'être mis en lumière.

Tout d'abord, ce stage m'a permis de développer un sens accru de l'autonomie. En étant confronté à des défis variés, j'ai été amené à résoudre des problèmes de manière autonome, renforçant ainsi ma confiance en mes capacités professionnelles. Cette liberté dans la gestion de mes tâches m'a également offert l'opportunité de développer ma capacité à prendre des décisions et à faire preuve d'initiative dans un environnement professionnel.

Par ailleurs, la diversité des missions qui m'ont été confiées m'a permis d'explorer différents aspects du développement web, du front-end au back-end, en passant par la gestion de bases de données. Cette polyvalence m'a permis d'aborder les projets avec une perspective plus « globale ».

Une autre dimension enrichissante de mon stage a été l'accent mis sur l'organisation du travail et la communication efficace avec les différentes parties prenantes. En travaillant en collaboration avec mes « clientes », j'ai appris à structurer mon travail de manière efficace et assurer une communication transparente tout au long du processus de développement.

En outre, l'utilisation de bibliothèques telles que FullCalendar ou Animate on Screen (AOS) m'a ouvert les yeux sur les nombreuses solutions disponibles en ligne pour faciliter mon travail. Ces outils m'ont grandement simplifié certaines tâches.

Enfin, l'utilisation de l'intelligence artificielle pour quelques cas particuliers très difficiles m'a été très utile. Cela m'a permis de proposer un "produit" final meilleur que cela aurait pu être si je m'étais refusé ChatGPT, par exemple.

En résumé, mon stage en télétravail pour le cabinet dentaire Dumitrascu Roxana a été une bonne expérience qui m'a permis de développer mes compétences techniques et organisationnelles. Les enseignements tirés de cette expérience continueront à me guider dans ma carrière professionnelle.

## 6 - Difficultés rencontrées

Le stage n'a pas été exempt de défis, m'ayant confronté à des situations qui ont mis à l'épreuve ma capacité d'adaptation. Voici quelques-unes des difficultés que j'ai rencontrées et surmontées au cours de mon stage :

Premièrement, le défi de travailler en autonomie totale s'avère toujours être une épreuve assez intimidante. Étant souvent seul face à des problèmes techniques, j'ai dû faire preuve de beaucoup d'initiative et d'une capacité d'auto-formation pour trouver des solutions. Cette responsabilité accrue m'a poussé à développer des compétences d'auto-apprentissage essentielles dans le domaine du développement web.

De plus, l'utilisation de ressources inédites, telles que la bibliothèque FullCalendar, a constitué un défi supplémentaire. Apprendre à maîtriser de nouveaux outils et technologies implique souvent la consultation de documentations détaillées et la navigation à travers des tutoriels complexes. Bien que cela ait demandé un investissement de temps et d'efforts supplémentaires, cette expérience m'a permis d'acquérir de nouvelles compétences et d'élargir mes connaissances dans des domaines spécifiques du développement web.

Par ailleurs, un obstacle imprévu est survenu lorsque WampServer, l'environnement de développement que j'utilisais initialement, a rencontré certains dysfonctionnements. Malgré plusieurs tentatives de réinstallation, le problème persistait, ce qui m'a finalement contraint à migrer vers XAMPP, une alternative que je n'avais jamais utilisée auparavant. Cette transition inattendue a nécessité une adaptation rapide à un nouvel environnement de travail et me familiariser avec celui-ci.

En somme, quelques difficultés qui m'ont permis de développer ma résilience, ma capacité d'adaptation et ma maîtrise de nouvelles compétences techniques. Chaque obstacle surmonté m'a permis de progresser et de devenir un développeur web plus compétent et plus confiant, prêt à relever les défis futurs avec détermination et assurance.

## IV - CONCLUSION

Mon expérience au cabinet dentaire Dumitrascu Roxana a été une belle expérience dans mon parcours professionnel. Ce stage m'a offert l'opportunité d'acquérir de précieuses compétences et de consolider mes connaissances dans le domaine du développement web.

La prise en charge de projets en autonomie m'a permis de développer ma capacité à résoudre des problèmes de manière indépendante et à prendre des décisions éclairées. Cette expérience m'a également sensibilisé à l'importance de la gestion efficace du temps et des ressources, ainsi qu'à l'organisation rigoureuse du travail.

La diversité des missions qui m'ont été confiées m'a exposé à une variété de technologies, renforçant ainsi ma polyvalence et ma capacité d'adaptation. J'ai pu approfondir mes compétences dans des domaines tels que le front-end, le back-end et la gestion de bases de données, tout en explorant de nouvelles technologies et en élargissant mes horizons professionnels.

Bien que j'aie été confronté à des difficultés, telles que la maîtrise de nouvelles technologies ou la résolution de problèmes techniques imprévus, j'ai su faire preuve de résilience et de détermination pour surmonter ces obstacles avec succès. Chaque défi rencontré m'a permis de progresser et de mûrir professionnellement, me préparant ainsi à affronter les défis futurs avec confiance.

Je tiens à exprimer ma gratitude envers toute l'équipe du cabinet dentaire Dumitrascu Roxana pour leur confiance à mon égard. Cela a été bénéfique pour la réalisation du site web de posséder beaucoup de liberté dans sa création.

En définitive, ce stage a été une expérience formatrice et stimulante qui a contribué de manière significative à mon développement en tant que développeur web. Cela m'aidera à poursuivre ma carrière dans le domaine du développement web.

## V - LEXIQUE

Dans le cadre de ce rapport de stage, un nombre assez conséquent de termes inconnus du grand public ont été expliqués. Il en reste néanmoins certains dont je n'ai pas encore abordé.

**HTML** : Acronyme de HyperText Markup Language, il s'agit du langage de balisage qui permet de structurer les pages web (les titres, les tableaux, les images, etc.).

**CSS** : Acronyme de Cascading Style Sheets, il s'agit d'un langage de description permettant la mise en forme d'un code HTML afin de le rendre plus attrayant (couleurs, disposition des éléments, police d'écriture, etc.).

**JavaScript** : Un langage de programmation interprété par le navigateur pour rendre les pages web dynamiques, c'est-à-dire qui changent selon la date de consultation ou avec les interactions de l'utilisateur. À l'inverse des pages dites « statiques » affichant un contenu fixe.

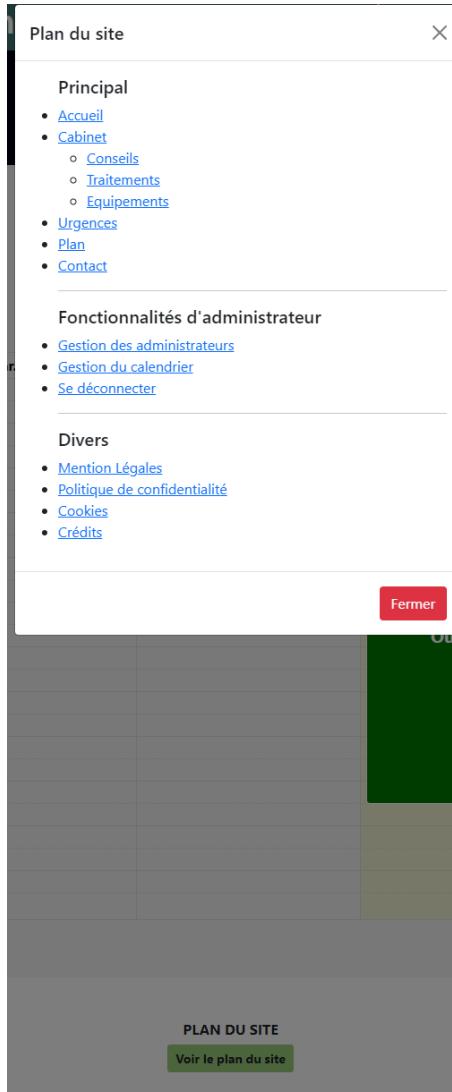
**PHP** : Un langage de programmation interprété par le serveur web (et non pas par le navigateur). Le code n'est pas affiché dans le code source de la page web consultée par l'utilisateur lambda, permettant ainsi de masquer certains détails (comme une clé privée utilisée dans le cadre d'un formulaire).

**Client FTP** : Il s'agit d'une application informatique (comme FileZilla) qui permet à un utilisateur de se connecter à un serveur distant via le protocole FTP (File Transfer Protocol) pour transférer des fichiers entre son propre ordinateur et le serveur (hébergeur web).

**Framework** : Un framework (« bibliothèque » en français) est un ensemble d'outils et de règles qui aident les développeurs à construire des applications plus rapidement en fournissant une structure préétablie et des fonctionnalités réutilisables (exemple : Bootstrap, FullCalendar, Animate on Screen (AOS)).

## VI - ANNEXE

Le rapport de stage étant déjà extrêmement long, je propose qu'on passe en revue vite fait d'autres petites fonctionnalités.



*Cliquer sur "Voir le plan du site" en bas permet de visualiser toutes les pages accessibles, celui-ci s'adapte si l'utilisateur est connecté ou non.*

▼ Sommaire

## CONSEILS

- Se brosser les dents après chaque repas
- Se rendre régulièrement chez son dentiste
- Utiliser la bonne technique de brossage
- Choisir une brosse à dents à poils souples
- Changer de brosse à dents tous les 3 mois

## TRAITEMENTS

- La carie dentaire
- Le plombage et les composites
- La dévitalisation dentaire
- L'extraction dentaire
- Les couronnes
- La couronne sur implant
- Les prothèses dentaires

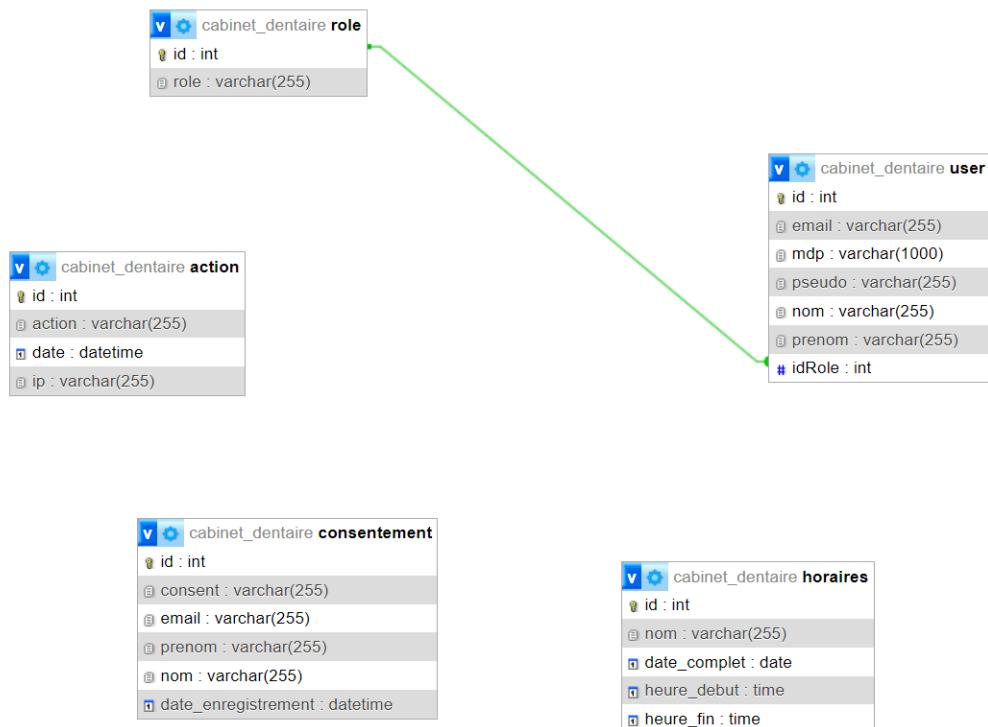
## ÉQUIPEMENTS

- Aéropolissage
- Caméra

Dans la page "cabinet.php", un sommaire est disponible en haut de la page, on peut cliquer sur un des éléments pour être emmené directement vers la bonne partie.

css	Dossier de ...	24/02/2024 01:...	0705	1004788 100	
images	Dossier de ...	04/03/2024 18:...	0705	1004788 100	
Include	Dossier de ...	20/02/2024 18:...	0705	1004788 100	
js	Dossier de ...	22/02/2024 05:...	0705	1004788 100	
ressources	Dossier de ...	13/03/2024 23:...	0705	1004788 100	
.htaccess	1 220	Fichier HT...	20/02/2024 18:...	0604	1004788 100
admin-modifier.php	3 961	Fichier sou...	13/03/2024 10:...	0604	1004788 100
admin.php	3 231	Fichier sou...	04/03/2024 18:...	0604	1004788 100
cabinet.php	17 904	Fichier sou...	04/03/2024 18:...	0604	1004788 100
calendrier-modif.php	18 896	Fichier sou...	04/03/2024 18:...	0604	1004788 100
contact.php	17 500	Fichier sou...	04/03/2024 18:...	0604	1004788 100
cookies.php	2 851	Fichier sou...	04/03/2024 18:...	0604	1004788 100
credits.php	461	Fichier sou...	04/03/2024 18:...	0604	1004788 100
deconnexion.php	402	Fichier sou...	04/03/2024 18:...	0604	1004788 100
index.php	11 067	Fichier sou...	04/03/2024 18:...	0604	1004788 100
login.php	3 897	Fichier sou...	04/03/2024 18:...	0604	1004788 100
logout.php	323	Fichier sou...	04/03/2024 18:...	0604	1004788 100
mentions-legales.php	2 990	Fichier sou...	04/03/2024 18:...	0604	1004788 100
page404.php	1 293	Fichier sou...	04/03/2024 18:...	0604	1004788 100
plan.php	2 896	Fichier sou...	04/03/2024 18:...	0604	1004788 100
politique.php	4 925	Fichier sou...	04/03/2024 18:...	0604	1004788 100
sitemap.xml	1 849	Fichier XML	04/03/2024 15:...	0604	1004788 100
urgences.php	1 197	Fichier sou...	04/03/2024 18:...	0604	1004788 100

Tous les fichiers et dossiers qui composent le site web.



Les cinq tables de la base de données.

Copyright © 2024 Cabinet Dentaire Dumitrascu Roxana Tous droits réservés. Site réalisé par **Terence Renard**

Tout en bas de la page, quand on clique sur le nom complet de l'unique créateur du site (moi) avec **sa couleur préférée**, cela mène vers son profil LinkedIn quand on clique dessus. La rumeur raconte qu'il cherche une alternance pour l'année prochaine.

Enfin, si l'envie vous prend, vous pouvez visiter le site web directement via l'adresse ci-dessous :

<https://dentiste-dumitrascu.fr/>