

Dossier de Projet

Stage en entreprise

20/11/2023 au 19/01/2024



Préparation au titre de Développeur·se Web et Web Mobile

Charles VANZETTA



Remerciements

Je tiens à débiter ce dossier en exprimant ma gratitude profonde envers ceux qui ont été essentiels dans mon parcours et dans l'aboutissement de ce projet.

Un immense merci va à mes professeurs pour leur enseignement précieux, leur patience et leur partage des connaissances.

Je suis également très reconnaissant envers Pauline pour sa confiance et son soutien. Son rôle dans ce projet a été central et inestimable.

Une mention spéciale à Albus, mon chat mon « rubber duck » personnel, qui, par sa présence silencieuse, m'a aidé à démêler les défis de la programmation. Merci d'avoir été là pour écouter mes raisonnements à voix haute.

Enfin, je ne saurais terminer sans un profond remerciement à ma compagne, dont le soutien inébranlable, la patience et la compréhension face à mes moments d'« eureka » suivis de sprints impromptus vers le PC ont été une source constante d'encouragement.

À chacun d'entre vous, je vous dois une part de la réussite de ce projet. Votre soutien et vos encouragements ont été les piliers de cette aventure.

Table des matières

Remerciements	2
Table des matières	3
Introduction	5
1 Présentation de l'entreprise	6
Origines de l'entreprise.....	6
Les services offerts.....	6
Structure opérationnelle.....	6
Objectif et évolution	6
2 Présentation du poste et de l'environnement technique	7
A. Environnement technique initial	7
Contexte.....	7
Solution WordPress initiale.....	7
Limite de WordPress.....	7
B. Choix technologique pour la refonte.	7
Critères de sélection	7
Développement avec Symfony et Angular.....	8
Rôle de WAMP	8
C. Procédure de développement	8
D. Travail en mode freelance	10
Influence de la littérature sur la gestion du projet	10
3 Liste des compétences du référentiel qui sont couvertes par le projet	11
4 Résumé du projet.....	12
5 Développement de l'application – Détails du Projet	13
Présentation d'une journée-type.....	13
Conception de l'application	13
Spécificités techniques.....	13
Organisation du développement	14
Présentation des pages principales l'application.....	16
Défis techniques et solutions.....	17
Maquettage de l'application	17
Création du diagramme de base de données	19
Mise à jour vers la version LTS de Symfony	20
Création des entités avec doctrine	21

Développement des controllers Symfony.....	22
Explication d'une partie d'un Controller	23
Comparaison avec des requêtes SQL directes	23
Création des API.....	24
Intégration du calendrier Angular.....	25
Tous les tests.....	26
Tests d'intégration et test fonctionnels	28
Le Défis Techniques : calendrier Angular.....	29
Apprentissage du développement frontend et backend séparé.....	29
API Symfony ne renvoyant aucune donnée.....	29
Problème spécifique au calendrier Angular.....	30
Et au final ?.....	33
La Sécurité.....	34
Authentification et autorisation	34
Validation des données.....	35
Protection contre les attaques CSRF et XSS.....	35
Communication sécurisée.....	35
Gestion des sessions	36
Stockage sécurisé des mots de passe	36
Évitez les spams	37
Perspectives et améliorations possibles.....	38
Améliorations techniques du projet	38
Impact du projet	38
Bilan personnel et compétences développées	38
Autonomie et souhait d'avoir un tuteur	38
Développement de compétences techniques	39
Importance de la veille technologique dans le développement web.....	39
Remerciements au Jury de l'Examen DWWM	39
Conclusion.....	40
Synthèse et accomplissement	40
Apprentissage et croissance	40
Impact et perspectives d'avenir.....	40

Introduction

Dans le paysage numérique contemporain, une présence en ligne est essentielle pour les professionnels souhaitant étendre leur portée et améliorer leur interaction client. Ce dossier de projet illustre cette transition, celle d'un site vitrine conçu initialement sous WordPress par moi-même vers une version avancée développée avec le framework PHP Symfony, pour le compte d'une auto-entrepreneuse en thérapie somato-sensorielle.

Le projet est né de la volonté de doter l'entreprise d'outils numériques à la hauteur de ses ambitions de services personnalisés et de bien-être client. En tant que concepteur initial du site WordPress, j'ai entrepris de le transformer, enrichissant ainsi l'expérience utilisateur par des fonctionnalités innovantes telles que la gestion de comptes et un module de réservation en ligne, tout en améliorant la communication entre la thérapeute et ses clients.

L'objectif de ce dossier est de mettre en lumière la manière dont le projet embrasse un large éventail de compétences techniques en développement web et web mobile, conformément au référentiel de certification professionnelle. Il atteste également de ma capacité à appliquer mes compétences acquises dans un contexte réel, en évaluant l'impact direct de ces dernières sur l'amélioration des services numériques de l'entreprise.

Ce document révèle non seulement les défis relevés et les solutions techniques adoptées mais sert aussi ma réflexion sur la pratique du développement web et sur les opportunités d'évolution du projet. Il constitue une vitrine de mes compétences et une évaluation de l'apport technologique au service de la profession du thérapeutique.

1 Présentation de l'entreprise

Origines de l'entreprise

Lancée en 2023 l'entreprise individuelle de Pauline Le Du Boucard émerge après des années dédiées à la formation en PNL en Thérapie manuelle et psycho corporelle ou encore en hypnose.

C'est pour elle l'aboutissement d'un parcours de formation spécialisé et la concrétisation d'un engagement envers le bien-être.

Elle a décidé d'exercer avant tout à « l'espace santé et bien-être » un cabinet de soins infirmiers qui l'accueille maintenant en tant que thérapeute psychocorporelle.

Les services offerts

Pauline Le Du Boucard propose des séances de thérapie somato sensorielle, de PNL ou encore d'hypnose, en s'appuyant sur ses connaissances et l'écoute de ses patients pour une pratique attentive et adaptée.

Structure opérationnelle

En tant qu'auto-entrepreneuse, Pauline gère toutes les facettes de son activité, depuis la prise de rendez-vous jusqu'au suivi des séances, avec une attention particulière à la qualité du service client.

Elle s'est greffée à un cabinet de soins infirmiers déjà existant.

L'espace santé et bien-être à Fontaine-Le-Comte.

Objectif et évolution

Pauline envisage de développer son entreprise par l'intégration de nouvelles techniques de thérapie et par une présence en ligne renforcée visant à étendre sa clientèle et à simplifier et moderniser ses services de réservation et de contact.

2 Présentation du poste et de l'environnement technique

A. Environnement technique initial

Contexte

En 2023, Pauline Le DU Boucard m'a confié la tâche de développer un site web pour son projet. Ce projet était axé sur une mise en marché rapide, nécessitant une plateforme facile à gérer et visuellement attrayante.

Solution WordPress initiale

Pour répondre à ces besoins, j'ai opté pour WordPress. Un choix stratégique en raison de sa facilité d'utilisation et de sa flexibilité.

WordPress s'est avéré idéal pour un déploiement rapide tout en offrant une interface intuitive permettant à Pauline de modifier elle-même le contenu de façon simple.

Limite de WordPress

Cependant, les besoins de Pauline, notamment en termes de fonctionnalités personnalisées et de performance ont mis en lumière les limites de WordPress. Les défis les plus notables étaient liés à la gestion des réservations et l'intégration de fonctionnalités avancées sur la manipulation des données utilisateurs.

B. Choix technologique pour la refonte.

Critères de sélection

Face aux limites de WordPress, il était impératif de trouver une solution plus apte à répondre aux besoins de Pauline. Mon choix s'est porté sur le framework PHP Symfony, accompagné d'Angular. Cette décision fut motivée par plusieurs facteurs : la performance accrue, la sécurité renforcée et la facilité de maintenance. Symfony, en particulier, se démarque par son architecture solide et sa grande flexibilité tandis qu'Angular offre une expérience utilisateur réactive et moderne.

Développement avec Symfony et Angular

L'intégration d'Angular dans le projet Symfony a marqué une étape cruciale. En tant que développeur plus à l'aise avec Symfony et les vues twig, j'ai pu harmoniser efficacement le HTML et le CMS, développer avec Symfony tout en intégrant un calendrier de réservation interactif conçu avec Angular. Cette combinaison a non seulement amélioré l'interface utilisateur mais également simplifié le processus de réservation : un élément clé pour la gestion efficace des rendez-vous de Pauline.

Rôle de WAMP

L'utilisation de WAMP (Windows, Apache, MySQL, PHP) dans notre environnement de développement a grandement facilité la phase de développement, de tests et de développement de l'application. Cette plateforme a permis une configuration simplifiée et une compatibilité accrue entre les différentes technologies utilisées, garantissant ainsi un flux de travail fluide et efficace.

C. Procédure de développement

1. Maquettage

Le processus de maquettage a débuté par une étroite collaboration avec Pauline pour déterminer ses exigences visuelles et fonctionnelles du site. Cette phase a été essentielle pour aligner l'aspect esthétique avec les fonctionnalités prévues, en mettant l'accent sur l'expérience utilisateur intuitive et engageante.

2. Diagramme de la base de données

La conception de la base de données était une étape cruciale nécessitant une attention particulière pour assurer une structuration efficace des données. Le diagramme de la base de données a servi de guide tout au long du processus de développement garantissant l'intégrité et la performance des données.

3. Création des entités Symfony et de la BDD avec doctrine

L'implémentation des entités Symfony et la configuration de la base de données via doctrine ont constitué la fondation du back end. Cette étape a mis en évidence la maîtrise de Symfony et de doctrine en assurant une gestion optimale des données et une intégration fluide avec le front end.

4. Développement des controllers et des fonctionnalités

La phase de développement a permis la création des controllers et des fonctionnalités clés du site, y compris des API pour la communication entre le frontend et le backend. Cette étape a non seulement démontré ma capacité à construire une architecture logicielle robuste, mais aussi à implémenter des fonctionnalités répondant précieusement aux besoins du projet.

5. Vues twig et tests de fonctionnalité

La création des vues twig a été un aspect important du développement frontend. Chaque vue a été soigneusement conçue pour offrir une expérience utilisateur fluide et esthétique. Les tests de fonctionnalités ont assuré que chaque aspect du site fonctionnait comme prévu, garantissant une qualité, une fiabilité, permettant aussi à Pauline de voir les premières avancées.

6. Responsives du site

L'adaptabilité du site sur différents appareils est une priorité. Assurant une expérience utilisateur cohérente sur mobile, tablette et ordinateur. Cette étape a impliqué un travail minutieux sur le CSS pour garantir que le site soit effectivement agréable et fonctionnel sur tous les appareils.

7. Passage sur Angular et création du calendrier

L'introduction d'Angular a marqué un tournant dans le développement frontend. La création d'un calendrier de réservation interactif avec Angular a amélioré l'engagement utilisateur, en offrant une interface dynamique pour la planification des rendez-vous.

8. Tests du Calendrier et compilation Angular dans Symfony

Les tests rigoureux du calendrier ont été essentiels pour assurer son intégration sans faille dans le système. La compilation d'Angular dans Symfony a démontré une fusion efficace des technologies, en harmonisant le dynamisme d'Angular avec la stabilité de Symfony.

9. Test de l'Intégration du calendrier dans la vue Twig

La phase finale a consisté à tester l'intégration du calendrier dans les vues Twig, s'assurant que tout fonctionnait de manière transparente. Cette étape a été cruciale pour offrir une expérience utilisateur sans interruption et de haute qualité.

D. Travail en mode freelance

Travailler sur ce projet en tant que développeur indépendant a été à la fois un défi et une opportunité enrichissante. Opérant dans un mode similaire à celui d'un freelance, j'ai pris en charge toutes les phases du projet, depuis la conceptualisation jusqu'à la mise en œuvre finale. Cette expérience a non seulement renforcé ma capacité à travailler de manière autonome, mais a également affiné mes compétences en gestion de projet. Gérer les exigences de Pauline, tout en coordonnant le développement technique et la mise en œuvre, a démontré une autonomie complète et une capacité à mener un projet complexe sans supervision directe.

Influence de la littérature sur la gestion du projet

Le livre "Deep Work" de Cal Newport a joué un rôle crucial dans ma méthode de travail. En adoptant les stratégies de travail, j'ai pu maximiser ma concentration et mon efficacité, en me focalisant intensément sur les tâches complexes du développement web. Cette approche a été essentielle pour gérer efficacement les multiples facettes de ce projet en mode freelance.

Parallèlement, "Atomic Habits" de James Clear a eu une influence significative sur ma routine quotidienne. En mettant en pratique les principes de Clear, j'ai développé des habitudes de travail régulières et productives. Cela m'a permis de maintenir un rythme de travail soutenu sans empiéter sur mon temps personnel, garantissant ainsi un équilibre sain entre vie professionnelle et personnelle.

3 Liste des compétences du référentiel qui sont couvertes par le projet

Nom et prénom du candidat : VANZETTA Charles

Document complété d'un commun accord entre le stagiaire et le responsable du stage en entreprise à joindre au rapport d'activité.

Compétences Voir le détail dans le référentiel d'emploi, d'activités et de compétences	Cocher les compétences mises en œuvre lors du projet en entreprise (en totalité ou partiellement)
C1 - Maquetter une application	<input checked="" type="checkbox"/>
C2 - Réaliser une interface utilisateur web statique et adaptable	<input checked="" type="checkbox"/>
C3 - Développer une interface utilisateur web dynamique	<input checked="" type="checkbox"/>
C4 - Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce	<input checked="" type="checkbox"/>
C5 - Créer une base de données	<input checked="" type="checkbox"/>
C6 - Développer les composants d'accès aux données	<input checked="" type="checkbox"/>
C7 - Développer la partie back-end d'une application web ou web mobile	<input checked="" type="checkbox"/>
C8 - Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce	<input checked="" type="checkbox"/>

Observations éventuelles :

Du stagiaire :

.....

.....

De l'entreprise :

.....

.....

Nom et Prénom du stagiaire :

VANZETTA Charles

Signature :



Entreprise : Pauline Le Du Boucard

Nom du responsable de stage : Pauline Le Du Boucard

Signature :



4 Résumé du projet

Le projet consistait à développer un site web dynamique pour Pauline Le DU Boucard, lancée en 2023. Initialement conçu avec WordPress pour sa rapidité de mise en marché et sa facilité de gestion, le site a rapidement montré ses limites, notamment en matière de fonctionnalités personnalisées et de gestion des réservations.

Face à ces défis, j'ai opté pour une refonte complète en utilisant le framework PHP Symfony et Angular pour le frontend, permettant une personnalisation accrue et une meilleure performance. En tant que développeur indépendant, j'ai géré toutes les phases du projet : de la conception initiale, en passant par le développement de fonctionnalités complexes comme un calendrier de réservation interactif, jusqu'aux tests finaux et la mise en production.

Ce projet a mis en évidence mes compétences en développement web full-stack, ma capacité à travailler de manière autonome et mon engagement à résoudre des problèmes techniques complexes, tout en équilibrant les exigences fonctionnelles et esthétiques.

L'adoption de principes de "Deep Work" et "Atomic Habits" a optimisé ma productivité et mon efficacité, assurant une gestion équilibrée entre le travail et la vie personnelle.

5 Développement de l'application – Détails du Projet

Présentation d'une journée-type

Une journée type sur ce projet commençait à 8h par une revue du code déjà fait et des problèmes rencontrés et la mise en place des objectifs de la journée autour d'un café. Cela me prenait généralement de 15 à 30 minutes.

Ensuite, je passais à des séances de développement intensives. Ces périodes étaient dédiées à la programmation, au débogage et aux tests. Je m'arrêtais vers 13h00.

Après une pause repas de 30 minutes à 1h, je me replongeais dans mon code pour me relire, continuer ou finir les phases de débogage. C'est souvent l'après-midi que j'effectuais mes recherches sur les problèmes rencontrés lors de la matinée ou que je risquais de rencontrer à la suite du projet.

Mes journées se terminaient généralement aux alentours de 16h30. Il pouvait aussi y avoir du travail de recherche en dehors de ces horaires.

Conception de l'application

Spécificités techniques

Langages de Programmation : Le projet utilise Symfony 6.4.2 (PHP) pour le backend et Angular 17 (TypeScript/JavaScript) pour le frontend.

Vue côté serveur : Les vues côté serveur sont générées en utilisant Twig (un moteur de modèles pour PHP) pour certaines parties de l'application, tandis qu'Angular est utilisé pour le calendrier de réservation.

Base de données : Une base de données est utilisée pour stocker les informations liées aux activités thérapeutiques, aux utilisateurs, aux réservations, etc...

API : L'interaction entre les vues Twig côté serveur et le front-end Angular se fait via des API RESTful pour récupérer et envoyer des données.

Services : Des services Angular sont utilisés pour interagir avec les API du back-end.

Tests : Le projet comprend des tests unitaires et d'intégration pour assurer la qualité du code.

Affichage des données : Les données du backend sont affichées dans les vues Twig côté serveur et dans le composant Angular du calendrier de réservation. Elles sont mises à jour dynamiquement.

Gestion des erreurs : Une gestion des erreurs est implémentée des deux côtés (serveur et client) pour assurer une expérience utilisateur robuste.

CSS/SCSS : Les styles sont gérés avec CSS ou SCSS pour la présentation de l'interface utilisateur, tant du côté des vues Twig que du côté du composant Angular.

Git et versionnement : Le contrôle de version est effectué avec Git et le projet est hébergé sur GitHub.

Sécurité : Symfony offre des fonctionnalités pour gérer l'authentification, l'autorisation et la sécurité globale de l'application.

Hébergement : le projet sera déployé sur la plateforme d'hébergement Hostinger une fois terminé.

Organisation du développement

Le projet s'est étendu sur une période de 2 mois durant laquelle plusieurs étapes clés avec des objectifs spécifiques et des étapes de test rigoureuses, ont été suivies pour assurer un développement structuré et efficace.

Phase de planification et de conception (1ère semaine)

Maquettage : Cette étape a consisté à créer des maquettes en étroite collaboration avec Pauline. J'ai utilisé les couleurs de sa marque et conçu l'interface sur papier pour faciliter la communication et les ajustements rapides avec elle.

Diagramme de la base de données : A la suite du maquettage, j'ai élaboré un diagramme de base de données définissant la structure nécessaire pour stocker et gérer les données de manière efficace.

Développement initial (2ème à 4ème Semaine)

Création de la base de données et des entités avec doctrine : J'utilisais doctrine pour définir les entités et les relations dans la base de données en me basant sur le diagramme établi.

Migration vers la nouvelle version LTD de Symfony : Cela permet de s'assurer de la pérennité du site web.

Développement des interfaces utilisateur statiques : Avec HTML et CSS, j'ai développé les interfaces utilisateur, en m'assurant qu'elles soient responsives pour ordinateur tablette et téléphone. Des tests sur différentes tailles d'appareils ont été effectués pour garantir une expérience utilisateur cohérente.

Intégration et développement des fonctionnalités (5ème à 6ème Semaine)

Développement des controllers Symfony : J'ai implémenté la logique de l'application en créant des controllers Symfony qui géraient la dynamique des données et des interactions utilisateurs. Après chaque développement de controllers, des tests ont été réalisés pour s'assurer de leur bon fonctionnement et de leur interaction efficace avec les vues.

Création des API Symfony pour communiquer avec Angular : L'objectif de la création des API Symfony est qu'elle permette à Angular d'interagir avec notre base de données. Ces API Symfony serviront à récupérer des données de la BDD pour ensuite les transmettre à Angular afin de simplifier le processus de réservation pour les utilisateurs.

Développement d'une interface dynamique : J'ai commencé à travailler sur l'intégration d'Angular, pour le développement du calendrier de réservation.

Phase finale et tests (7ème à 8ème Semaine)

Finalisation du calendrier Angular : Les dernières semaines ont été consacrées à finaliser le calendrier Angular bien que cette fonctionnalité n'ait pas été entièrement opérationnelle à la fin du stage.

Test continu : Etape du développement. Les composants réalisés ont été soumis à des tests. Cette approche itérative a permis de détecter et de corriger rapidement les problèmes, garantissant ainsi une progression stable du projet.

Pas de déploiement : N'étant pas finie pour le moment, elle ne sera pas déployée pendant le stage.

Présentation des pages principales l'application

L'application comporte plusieurs pages clés : une page d'accueil, une page « à propos », une page qui présente les prestations, une page de réservation, une page de contact... Chaque page a été conçue pour être à la fois intuitive et fonctionnelle en respectant les couleurs et le style de la marque de l'entreprise. Toutes les pages sont aussi responsives. Pour cela, j'ai intégré des break points et un menu hamburger. Toutes ces pages ont en commun la Nav bar et le Footer. La Nav bar est différente en fonction de si l'utilisateur est connecté, déconnecté ou si c'est l'administrateur. Toutes les pages du site sont responsives d'après une étude en juillet 2022, 6 demandes de page web sur 10. Proviennent d'un téléphone mobile, mobile, et ces chiffres sont en constante hausse.

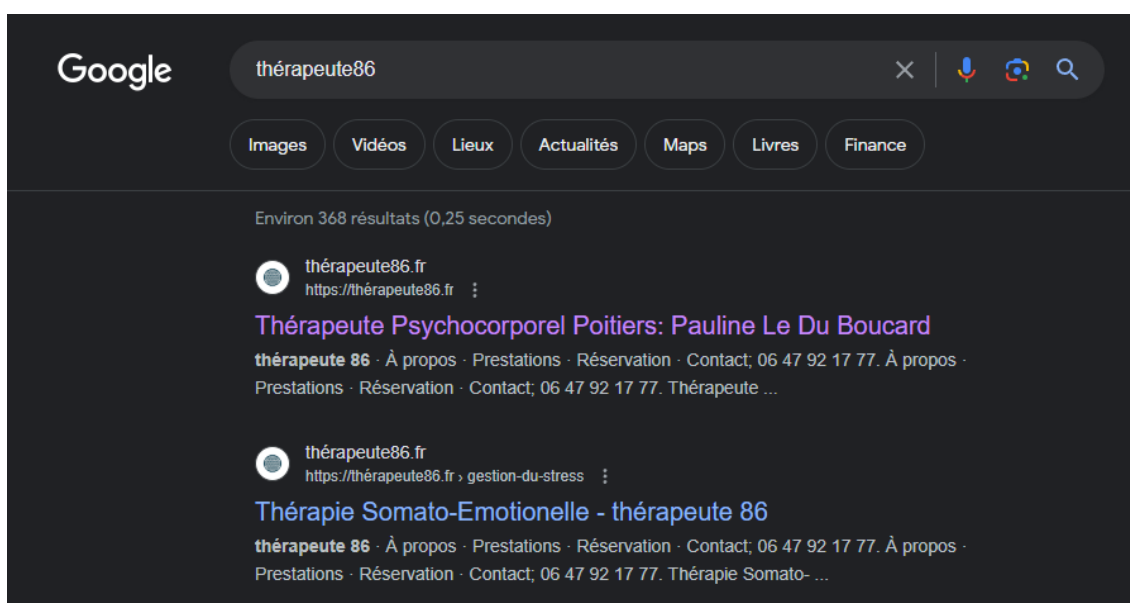
La page d'accueil, souvent la première interaction qu'un utilisateur a avec le site, a été conçue avec un soin particulier en tenant compte de son importance, tant pour l'expérience utilisateur que pour le référencement (SEO). Je me suis assuré que la page d'accueil reflète fidèlement l'identité et les valeurs de la marque par l'intégration des couleurs de la marque, un design épuré et une navigation intuitive grâce à des couleurs spécifiques au Call to action.

Des éléments visuels accrocheurs, un contenu informatif et des effets de mouvement ont été placés stratégiquement pour capter l'attention des visiteurs dès leur arrivée sur le site.

La page d'accueil a été optimisée pour le SEO par l'utilisation de balises méta pertinentes afin d'augmenter la visibilité du site sur les moteurs de recherche : deux titres et des descriptions optimisées avec intégration de mots-clés stratégiques.

Cela fonctionne car le site apparaît en tant que 6ème recherche quand on tape « thérapeute 86 ».

De plus, si l'utilisateur tape « thérapeute86 » (tout attaché), les deux premiers liens renvoient vers le site de Pauline.



La page de connexion, création de compte et modification est essentielle pour permettre aux clients de pouvoir réserver tout en s'assurant de l'existence d'une identité réelle pour éviter toute réservation intempestive par des concurrents ou des bots. Ces informations peuvent aussi être utiles à Pauline pour les fiches-client qui l'aident dans la pratique de son métier.

La page de réservation est un élément clé du site, conçu pour offrir une expérience utilisateur fluide et intuitive tout en répondant aux besoins de Pauline. La page a été élaborée en mettant l'accent sur la facilité d'utilisation. Elle permet au client de visualiser rapidement les services disponibles et les créneaux horaires libres pour leurs rendez-vous. Je voulais un calendrier interactif avec une réservation dite « on click ». Je l'ai donc développé avec Angular et intégré pour permettre une sélection de créneaux (date et heure) de manière visuelle. Cette fonctionnalité est au cœur de l'interface utilisateur de la page de réservation. La page de réservation interagit avec le back End Symfony via les API. Ces dernières assurent la récupération en temps réelle des disponibilités et de la gestion des réservations.

Défis techniques et solutions

Maquettage de l'application

Le maquettage avait été fait avant le stage lors de la réalisation de son site en WordPress. J'ai repris la même méthode pour les pages qui n'étaient pas présentes à cette époque.

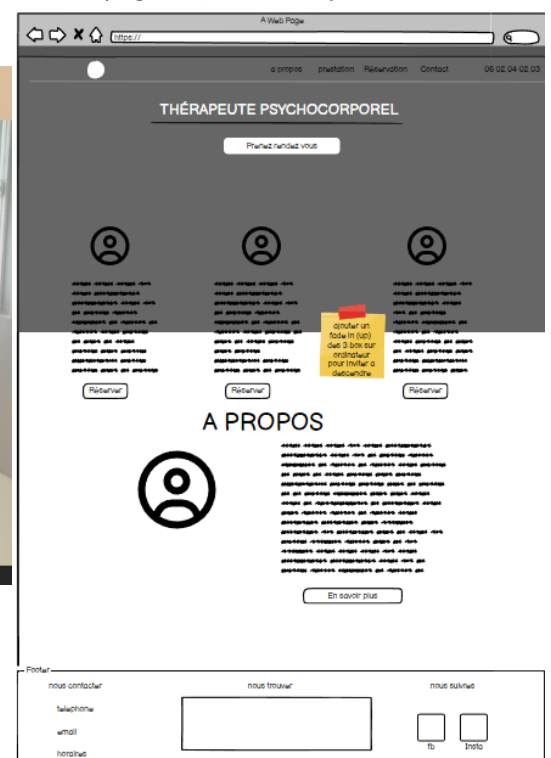
Défi : Le principal défi du maquettage résidait dans le fait de créer un design visuellement attrayant et fonctionnel, tout en étant dans mes capacités de code.

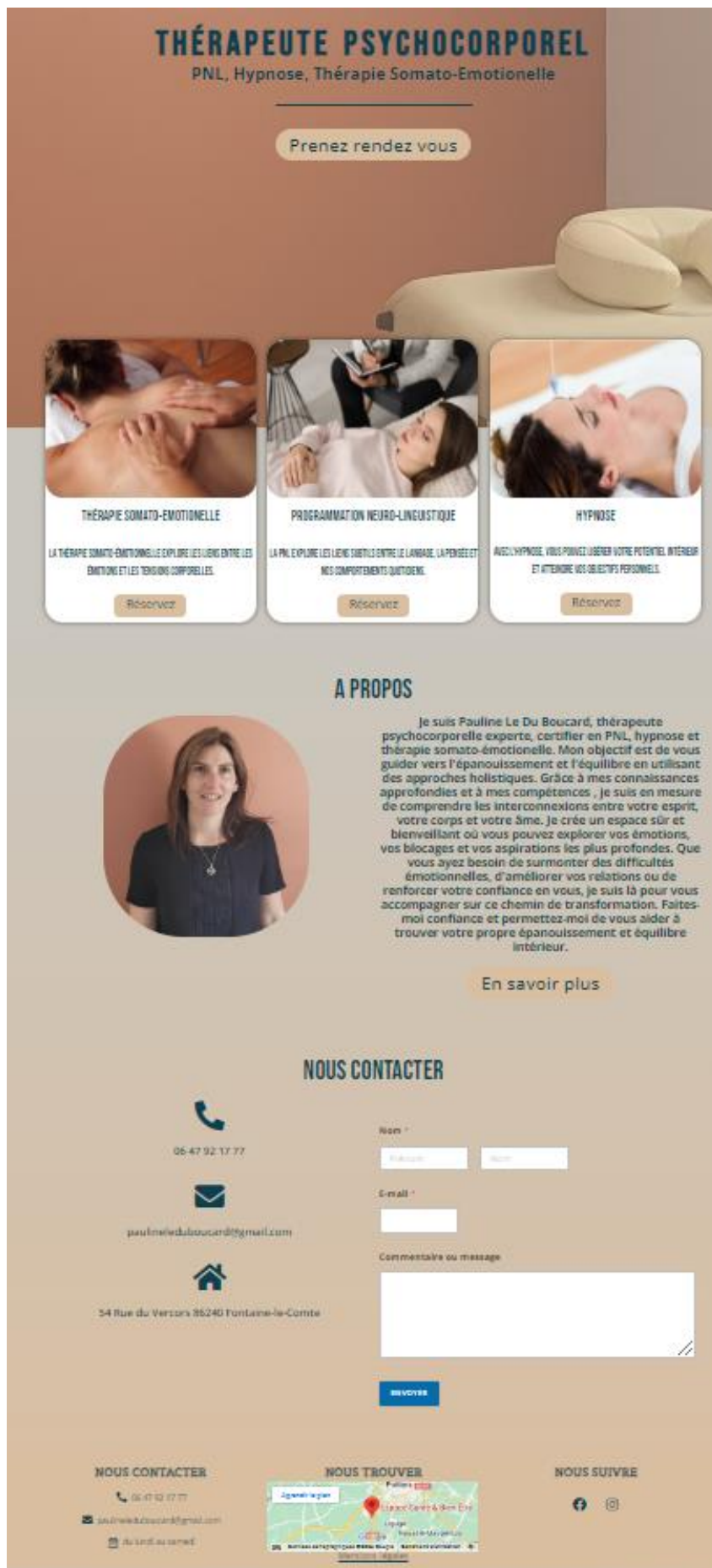
Solution : Pour relever ce défi, j'ai adopté une approche itérative en commençant par les ébauches papier de chaque composant séparé (footer, nav-bar, composants de page, ...) une fois présenté, retravaillé et accepté, nous les avons validées.



Exemple du maquettage de la page d'accueil fait avec balsamiq

Et le résultat codé





Résultat de la page d'accueil en HTML CSS avec une vue Twig

Exemple d'effets over sur les boutons

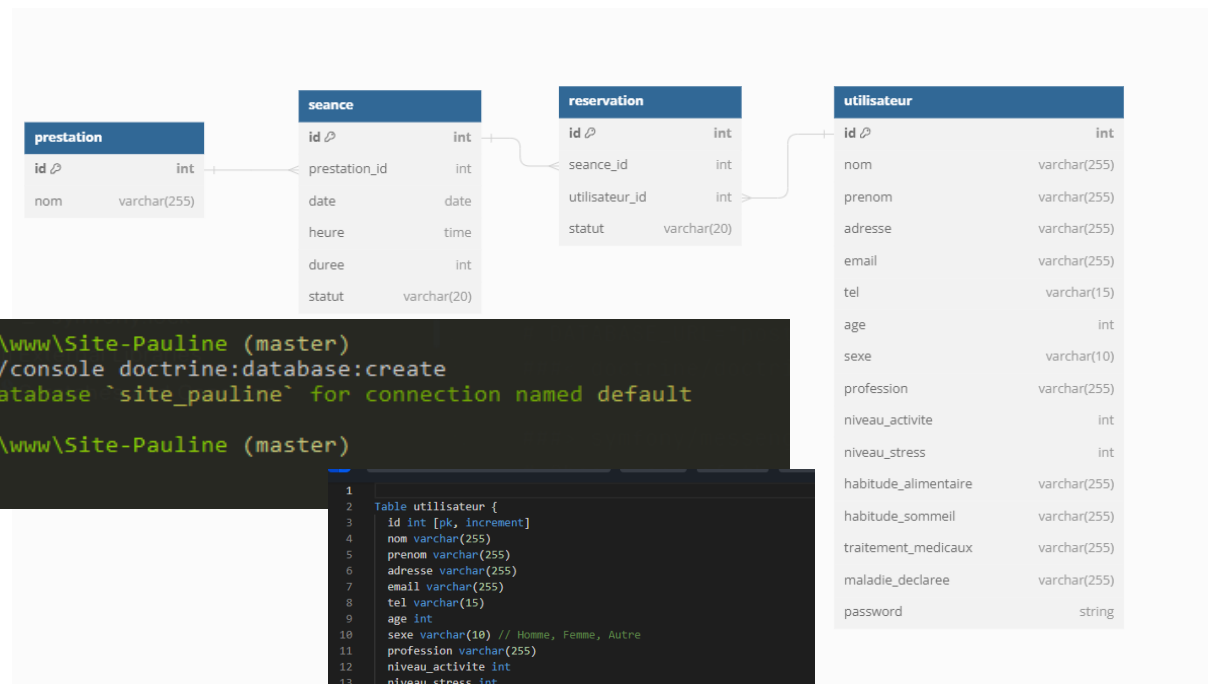
renforcer votre confiance en vous, je suis là pour vous accompagner sur ce chemin de transformation. Faites-moi confiance et permettez-moi de vous aider à trouver votre propre épanouissement et équilibre intérieur.

En savoir plus

Création du diagramme de base de données

Défi : Le diagramme de la base de données représentait un défi en termes de structuration efficace et logique des données. Il fallait aussi être sûr des données présentes et de leurs relations avec les tables.

Solution : Pour créer un diagramme de base de données efficace j'ai d'abord analysé en détail les besoins en données de l'application en identifiant les entités clés et leurs relations. J'ai ensuite utilisé des diagrammes pour visualiser la structure de la base de données, ce qui m'a permis de mieux comprendre et d'ajuster certains détails que j'avais omis.



```

D:\wamp64\www\Site-Pauline (master)
λ php bin/console doctrine:database:create
Created database `site_pauline` for connection named default

D:\wamp64\www\Site-Pauline (master)
λ |
  
```

```

Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.31 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| therapeute86 |
+-----+
5 rows in set (0.09 sec)

mysql> use therapeute86
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_therapeute86 |
+-----+
| contact |
| doctrine_migration_versions |
| messenger_messages |
| prestation |
| reservation |
| seance |
| user |
+-----+
7 rows in set (0.02 sec)

mysql> SELECT * FROM user
-> ;
  
```

```

1
2 Table utilisateur {
3   id int [pk, increment]
4   nom varchar(255)
5   prenom varchar(255)
6   adresse varchar(255)
7   email varchar(255)
8   tel varchar(15)
9   age int
10  sexe varchar(10) // Homme, Femme, Autre
11  profession varchar(255)
12  niveau_activite int
13  niveau_stress int
14  habitude_alimentaire varchar(255)
15  habitude_sommeil varchar(255)
16  traitement_medicaux varchar(255)
17  maladie_declaree varchar(255)
18  password string
19 }
20
21 Table prestation {
22   id int [pk, increment]
23   nom varchar(255)
24 }
25
26 Table seance {
27   id int [pk, increment]
28   prestation_id int [ref: > prestation.id]
29   date date
30   heure time
31   duree int
32   statut varchar(20) // Libre, Réservé, Confirmé, Annulé, etc.
33 }
34
35 Table reservation {
36   id int [pk, increment]
37   seance_id int [ref: > seance.id]
38   utilisateur_id int [ref: > utilisateur.id]
39   statut varchar(20) // En attente, Confirmée, Annulée, etc.
40 }
41
  
```

Vue de la base de données,
des tables et de la table user
sur MySQL

```

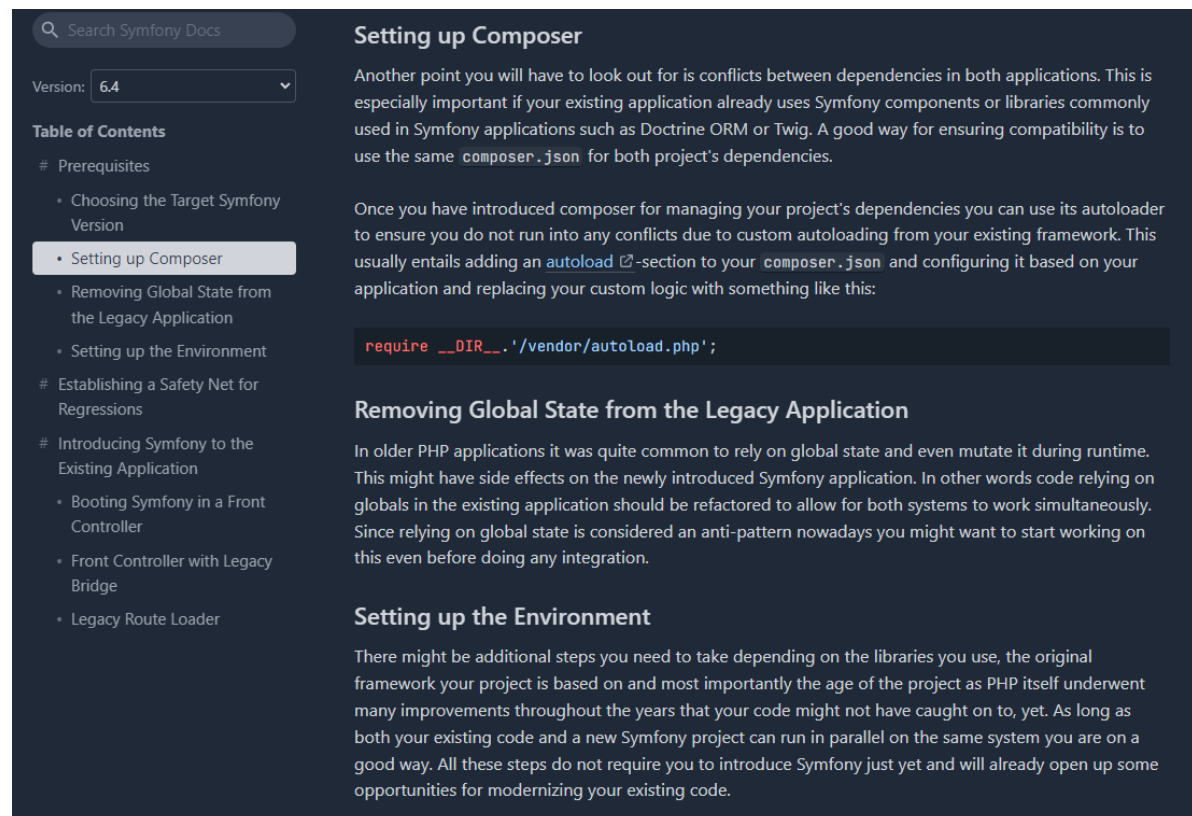
mysql> SELECT * FROM user
-> ;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | email | roles | password | nom | prenom | adresse | tel | age | sexe | profession | is_verified |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 4 | mixocoder@gmail.com | [] | $2y$13$GG3uA.o9h8/iqzsm6Ifi.js522bq1ZnCcmT/0PaJAgRwo2ak01Qq | mixo | coder | 6 boulevard du code | NULL | 23 | Homme | barman | 0 |
| 5 | Dossier@projet.com | [] | $2y$13$E4zoYC59MvCfPh/eZOKu/.bSson18KBA623pRQEBFYIZjQXH7A1r1 | Dépoint | Marc | 6 boulevard pas loins | 07 80 53 62 94 | 49 | Homme | barman | 0 |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.02 sec)

mysql>
  
```

Mise à jour vers la version LTS de Symfony

Défi : La dernière version LTS de Symfony été mise à disposition en novembre 2023. Après 2 semaines de stage, j'ai donc dû effectuer la migration.

Solution : J'ai utilisé un « composer update » après l'installation de cette version et j'ai suivi, des Guides (officiels) de migration pour apporter les modifications nécessaires.



Search Symfony Docs

Version: 6.4

Table of Contents

- # Prerequisites
 - Choosing the Target Symfony Version
 - Setting up Composer
 - Removing Global State from the Legacy Application
 - Setting up the Environment
- # Establishing a Safety Net for Regressions
- # Introducing Symfony to the Existing Application
 - Booting Symfony in a Front Controller
 - Front Controller with Legacy Bridge
 - Legacy Route Loader

Setting up Composer

Another point you will have to look out for is conflicts between dependencies in both applications. This is especially important if your existing application already uses Symfony components or libraries commonly used in Symfony applications such as Doctrine ORM or Twig. A good way for ensuring compatibility is to use the same `composer.json` for both project's dependencies.

Once you have introduced composer for managing your project's dependencies you can use its autoloader to ensure you do not run into any conflicts due to custom autoloading from your existing framework. This usually entails adding an `autoload` section to your `composer.json` and configuring it based on your application and replacing your custom logic with something like this:

```
require __DIR__.'vendor/autoload.php';
```

Removing Global State from the Legacy Application

In older PHP applications it was quite common to rely on global state and even mutate it during runtime. This might have side effects on the newly introduced Symfony application. In other words code relying on globals in the existing application should be refactored to allow for both systems to work simultaneously. Since relying on global state is considered an anti-pattern nowadays you might want to start working on this even before doing any integration.

Setting up the Environment

There might be additional steps you need to take depending on the libraries you use, the original framework your project is based on and most importantly the age of the project as PHP itself underwent many improvements throughout the years that your code might not have caught on to, yet. As long as both your existing code and a new Symfony project can run in parallel on the same system you are on a good way. All these steps do not require you to introduce Symfony just yet and will already open up some opportunities for modernizing your existing code.

Cette recherche en anglais, issue du guide officiel de migration de Symfony, explique que l'on doit être attentif à la configuration de Composer, car il peut y avoir des conflits de dépendance dans les deux applications, surtout si l'on utilise des bibliothèques couramment utilisées telles que Doctrine, ORM ou Twig. Le guide nous conseille donc d'utiliser l'autoloader de Composer pour ne pas rencontrer de conflits liés à un chargement automatique modifié. Cela implique généralement d'ajouter la section autoloader au fichier `composer.json`. Un exemple illustre le propos.

La suite parle de la suppression de l'état global de l'application qui était présent dans les anciennes applications PHP et de la configuration de l'environnement.

Il peut y avoir des étapes supplémentaires à suivre en fonction des bibliothèques utilisées.

Création des entités avec doctrine

Défi : La création des entités pour structurer la base de données a nécessité une compréhension approfondie des relations entre les tables.

Solution : J'ai utilisé les capacités de mappage Objet-Relationnel de doctrine pour créer des entités en garantissant l'intégrité et la sécurité des opérations de la base de données.

Exemple de création d'entités

```

7
8 #[ORM\Entity(repositoryClass: ReservationRepository::class)]
9 #[ORM\HasLifecycleCallbacks]
10 class Reservation
11 {
12     #[ORM\Id]
13     #[ORM\GeneratedValue]
14     #[ORM\Column(type: "integer")]
15     #[Groups("reservation")]
16     private ?int $id = null;
17
18     #[ORM\ManyToOne(cascade: ['persist'], inversedBy:
19     #[Groups("reservation")]
20     private ?Seance $seance = null;
21
22     #[ORM\ManyToOne(inversedBy: "reservations")]
23     #[Groups("reservation")]
24     private ?User $utilisateur = null;
25
26     #[ORM\Column(length: 20)]
27     #[Groups("reservation")]
28     private ?string $statut = null;
29
30     public function getId(): ?int
31     {
32         return $this->id;
33     }
34
35     public function getSeance(): ?Seance
36     {
37         return $this->seance;
38     }
39
40     public function setSeance(?Seance $seance): static
41     {
42         $this->seance = $seance;
43
44         return $this;
45     }
46
47     public function getUtilisateur(): ?User
48     {
49         return $this->utilisateur;
50     }
51

```

```

D:\wamp64\www\Site-Pauline (master)
λ php bin/console make:entity

Class name of the entity to create or update (e.g. VictoriousPizza):
> Prestation
Prestation

created: src/Entity/Prestation.php
created: src/Repository/PrestationRepository.php

Entity generated! Now let's add some fields!
You can always add more fields later manually or by re-running this command.

New property name (press <return> to stop adding fields):
> nom

Field type (enter ? to see all types) [string]:
>

Field length [255]:
>

Can this field be null in the database (nullable) (yes/no) [no]:
>

updated: src/Entity/Prestation.php

```

Exemple de migration pour persister les données en BDD

```

c:\wamp64\www\Sortir.com (main -> origin)
λ php bin/console doctrine:migrations:migrate

WARNING! You are about to execute a migration in database "sortir" that could result in schema changes
and data loss. Are you sure you wish to continue? (yes/no) [yes]:
>

```

Développement des controllers Symfony

Défi : Le développement de controllers robustes pour gérer la logique métier et les interactions utilisateurs a été un aspect complexe du projet.

Solution : En adoptant une approche Modèle-Vue-Controller, j'ai pu isoler la logique métier dans les controllers, en rendant le code plus clair, modulaire et facile à maintenir.

```
D:\wamp64\www\therapeute86 (main -> origin)
λ php bin/console make:controller ProfileController
created: src/Controller/ProfileController.php
created: templates/profile/accueil.html.twig
```

Success!

Next: Open your new controller class and add some pages!

```
D:\wamp64\www\therapeute86 (main -> origin)
λ
```

Exemple de contrôleur dans mon code

```
<?php

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
use Symfony\Component\Security\Http\Authentication\AuthenticationUtils;

class SecurityController extends AbstractController
{
    #[Route('/security', name: 'app_security')]
    public function index(): Response
    {
        return $this->render('security/accueil.html.twig', [
            'controller_name' => 'SecurityController',
        ]);
    }

    #[Route('/login', name: 'app_login')]
    public function login(AuthenticationUtils $authenticationUtils): Response
    {
        // Obtenir l'erreur de connexion s'il y en a une
        $error = $authenticationUtils->getLastAuthenticationError();
        // Dernier nom d'utilisateur saisi par l'utilisateur
        $lastUsername = $authenticationUtils->getLastUsername();

        return $this->render('security/login.html.twig', [
            'last_username' => $lastUsername,
            'error'         => $error,
        ]);
    }

    #[Route('/logout', name: 'app_logout')]
    public function logout(): void
    {
        // La logique de déconnexion est gérée automatiquement par Symfony.
        throw new LogicException('This method can be blank - it will be intercepted by the logout key on your firewall.');
```


Explication d'une partie d'un Controller

```
class ReservationController extends AbstractController
{
    private EntityManagerInterface $entityManager;

    public function __construct(EntityManagerInterface $entityManager)
    {
        $this->entityManager = $entityManager;
    }

    #[Route('/planning/{seanceId}/reserve', name: 'planning_reserve')]
    public function reserveSlot(Seance $seance): JsonResponse
    {
        if (!$this->getUser()) {
            return new JsonResponse(['error' => 'Authentification requise.'], Response::HTTP_UNAUTHORIZED);
        }

        if (!$this->isSeanceFree($seance)) {
            return new JsonResponse(['error' => 'La séance est déjà réservée.'], Response::HTTP_BAD_REQUEST);
        }

        $reservation = new Reservation();
        $reservation->setSeance($seance);
        $reservation->setUtilisateur($this->getUser());
        $reservation->setStatut('confirmed');

        $this->entityManager->persist($reservation);
        $this->entityManager->flush();

        if ($seance->getReservations()->count() === 1) {
            $seance->setStatut('reserve');
            $this->entityManager->flush();
        }

        return new JsonResponse(['id' => $reservation->getId()]);
    }
}
```

Vérifie si l'utilisateur est actuellement connecté.

Vérifie si la séance est disponible.

Création d'une nouvelle réservation.

Persistance et sauvegarde.

Mise à jour de la séance.

Réponse, JSON.

Comparaison avec des requêtes SQL directes

Vérification d'authentification :

SELECT * FROM users **WHERE** session_id = ?;

Vérification de disponibilité :

SELECT * FROM seances **WHERE** id = ? **AND NOT EXISTS** (SELECT 1 FROM reservations **WHERE** seance_id = ?)

Création d'une nouvelle réservation:

INSERT INTO reservations (seance_id, user_id, status) **VALUES** (?, ?, 'confirmed');

Mise à jour de la séance :

UPDATE seances **SET** status = 'reserve' **WHERE** id = ?;

Création des API

Défi : Concevoir des API fiables et sécurisées pour la communication entre le frontend et le backend.

Solution : J'ai développé des API RESTful en utilisant Symfony, en veillant à leur sécurité et à leur efficacité. Les tests effectués avec Postman ont permis de valider leur fonctionnalité.

Exemples de mon code pour les API

Résultat avec Postman

```
class ReservationApiController extends AbstractController
{
    private EntityManagerInterface $entityManager;

    public function __construct(EntityManagerInterface $entityManager)
    {
        $this->entityManager = $entityManager;
    }

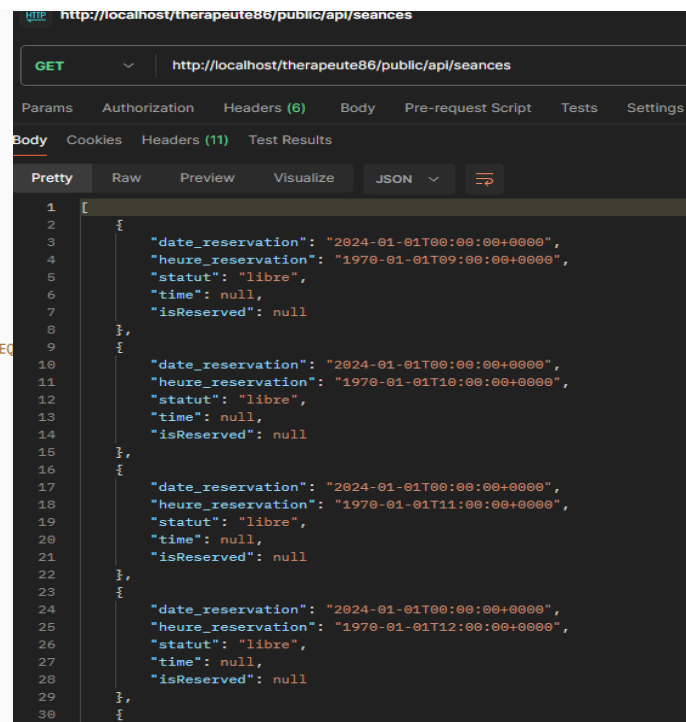
    #[Route('/api/reservation/{seanceId}/reserve', name: 'api_reservation_reserve')]
    public function reserveSlot(Seance $seance): JsonResponse
    {
        // Vérifier si La séance est libre avant de permettre la réservation
        if (!$seance->isFree()) {
            // Séance déjà réservée, retourner une réponse JSON avec une erreur
            return new JsonResponse(['error' => 'La séance est déjà réservée.'], Response::HTTP_BAD_REQUEST);
        }

        // Logique pour réserver un créneau (ajouter une nouvelle réservation)
        $reservation = new Reservation();
        $reservation->setSeance($seance);
        $reservation->setUtilisateur($this->getUser());
        $reservation->setStatut('confirmed');

        // Enregistrez la nouvelle réservation dans la base de données
        $this->entityManager->persist($reservation);
        $this->entityManager->flush();

        // Mettre à jour le statut de la séance à "réservé" si c'est la première réservation
        if ($seance->getReservations()->count() === 1) {
            $seance->setStatut('reserve');
            $this->entityManager->flush();
        }

        return new JsonResponse(['message' => 'Reservation successful']);
    }
}
```



```
1 [
2   {
3     "date_reservation": "2024-01-01T00:00:00+0000",
4     "heure_reservation": "1970-01-01T09:00:00+0000",
5     "statut": "libre",
6     "time": null,
7     "isReserved": null
8   },
9   {
10    "date_reservation": "2024-01-01T00:00:00+0000",
11    "heure_reservation": "1970-01-01T10:00:00+0000",
12    "statut": "libre",
13    "time": null,
14    "isReserved": null
15  },
16  {
17    "date_reservation": "2024-01-01T00:00:00+0000",
18    "heure_reservation": "1970-01-01T11:00:00+0000",
19    "statut": "libre",
20    "time": null,
21    "isReserved": null
22  },
23  {
24    "date_reservation": "2024-01-01T00:00:00+0000",
25    "heure_reservation": "1970-01-01T12:00:00+0000",
26    "statut": "libre",
27    "time": null,
28    "isReserved": null
29  },
30 ]
```

```
#[Route('/api/reservation/{id}/cancel', name: 'api_reservation_cancel', methods: ['POST'])]
public function cancelReservation(Reservation $reservation): Response
{
    $now = new \DateTime();
    $diff = $now->diff($reservation->getSeance()->getCreatedAt());

    if ($diff->days < 2) {
        $reservation->setStatut('cancelled');
        $this->entityManager->flush();
        return $this->json(['message' => 'Reservation annuler']);
    } else {
        return $this->json(['error' => 'La réservation ne peut pas être annulée car plus de 48 heures se sont écoulées.'], Response::HTTP_BAD_REQUEST);
    }
}

#[Route('/api/seances', name: 'api_seances_list', methods: ['GET'])]
public function listSeances(SerializerInterface $serializer): Response
{
    $seances = $this->entityManager->getRepository(Seance::class)->findAll();
    if (!$seances) {
        return $this->json(['error' => 'Aucune séance trouvée.'], Response::HTTP_NOT_FOUND);
    }

    $context = ['datetime_format' => \DateTimeInterface::ISO8601, 'groups' => 'seance'];
    $responseData = $serializer->normalize($seances, null, $context);
    return $this->json($responseData);
}
```


Intégration du calendrier Angular

Défi : L'intégration du calendrier interactif Angular dans l'environnement Symfony a été particulièrement complexe, notamment en ce qui concerne la synchronisation des données et la communication entre les deux technologies.

Solution : Après plusieurs tentatives, j'ai réussi à intégrer le calendrier Angular avec l'API Symfony. Bien que le calendrier n'ait pas été complètement opérationnel à la fin du stage, des progrès significatifs ont été réalisés en termes de fonctionnalité et d'intégration. Une fois le calendrier testé, j'utilise la commande ng Build ce qui crée dans le fichier DIST 3 fichiers, que je devrai mettre dans le dossier public, de mon application Symfony.

Commande exécuter et les dossiers qui sont impactés

```
D:\wamp64\www\therapeute86\Angular (main)
λ ng build
```

Initial Chunk Files	Names	Raw Size	Estimated Transfer Size
main-OY5RFOMZ.js	main	504.68 kB	124.18 kB
styles-B2AQ5SLV.css	styles	94.04 kB	9.88 kB
polyfills-LZBJRJJE.js	polyfills	32.69 kB	10.59 kB
Initial Total		631.42 kB	144.65 kB

```
Application bundle generation complete. [14.863 seconds]
```

- dist
 - front-end
 - browser
 - favicon.ico
 - index.html
 - main-PYEXFX3C.js
 - polyfills-LZBJRJJE.js
 - styles-5R6CU2RA.css

- public
 - angular
 - main-OY5RFOMZ.js
 - polyfills-LZBJRJJE.js
 - styles-B2AQ5SLV.css

LE DU BOUCARD Pauline

25 sur 40

ENI - École Informatique

Tous les tests

Test de responsive

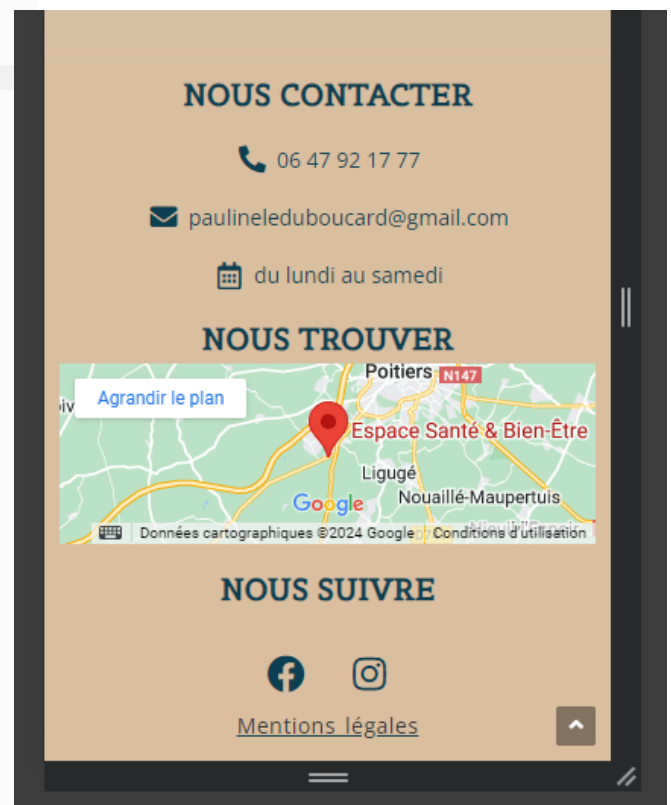
J'ai vérifié sur mon navigateur web (que le site web était utilisable par les utilisateurs, ordinateurs, tablette et téléphone en utilisant la touche F 12). Il aurait sûrement été plus pertinent de coder en mobile first. J'ai utilisé les Media Queries.

```
@media (max-width: 768px) {  
  footer div {  
    width: 100%;  
    display: block;  
    margin-right: 0;  
    margin-left: 0;  
  }  
  
  footer h3 {  
    font-size: 1.4em;  
  }  
  
  footer i.follow-icon {  
    font-size: 25px;  
    padding: 10px;  
  }  
}  
  
@media (max-width: 480px) {  
  footer div {  
    width: 100%;  
    display: block;  
    margin-right: 0;  
    margin-left: 0;  
    padding: 10px 0;  
  }  
  
  footer h3 {  
    font-size: 1.2em; /* Taille de titre réduite */  
  }  
  
  footer p, footer a {  
    font-size: 0.9em; /* Taille de texte réduite */  
  }  
  
  footer i {  
    font-size: 20px; /* Taille des icônes réduite */  
    margin-right: 5px;  
  }  
  
  iframe {  
    height: 100px;  
  }  
}
```

Footer en mode mobile

et

le CSS qui le gère



Test unitaire

J'ai isolé chaque fonctionnalité et les ai testées indépendamment pour valider la logique et leur comportement. Pour le backend Symfony j'ai utilisé PHP unit. Pour le frontend, les tests ont été mis en place avec Jasmine et Karma.

```
import { TestBed } from '@angular/core/testing';
import { AppComponent } from './app.component';

describe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [AppComponent],
    }).compileComponents();
  });

  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have the 'Front-end' title`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('Front-end');
  });

  it('should render title', () => {
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as HTMLElement;
    expect(compiled.querySelector('h1')?.textContent).toContain('Hello, Front-end');
  });
});
```

Configuration du Testbed

Test de création de l'application et test de titre du composant

Karma v 6.4.2 - connected;

Chrome 120.0.0.0 (Windows 10) is idle

Jasmine 4.6.0
.....

7 specs, 0 failures, randomized with seed 3072

```
ContactFormComponent
  • should create
ContactService
  • should be created
CalendarComponent
  • should create
ReservationService
  • should be created
AppComponent
  • should create the app
  • should render title
  • should have the 'Front-end' title
```

Jasmine 4.6.0
...xx

7 specs, 3 failures, randomized with seed 19484

Spec List | [Failures](#)

```
CalendarComponent
  • should create
ReservationService
  x should be created
AppComponent
  • should render title
  • should have the 'Front-end' title
  • should create the app
ContactFormComponent
  x should create
ContactService
  x should be created
```

Exemples de tests unitaires

Tests d'intégration et test fonctionnels

J'ai vérifié que les différents composants du système (frontend, backend, base de données) interagissent correctement entre eux.

J'ai évalué le système dans son ensemble pour m'assurer qu'il réponde aux exigences spécifiques du projet.

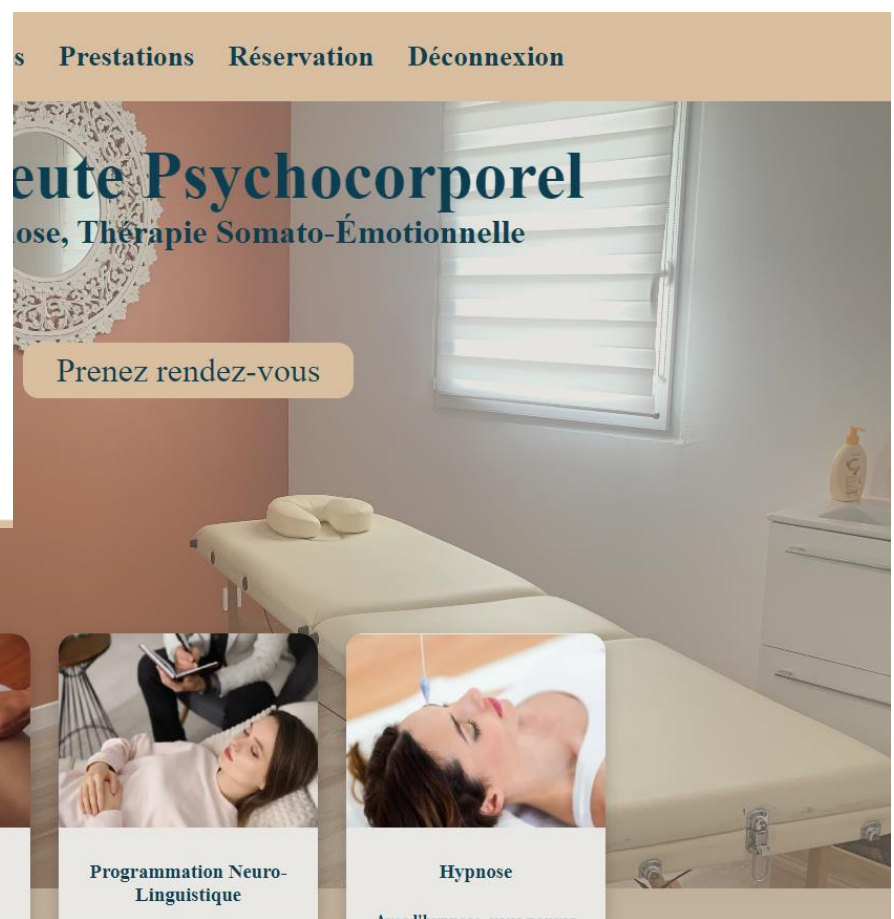
Pour faire ces tests, j'ai utilisé le site web en testant les fonctionnalités, création d'utilisateur, connexion, déconnexion, suppression de comptes, modification de compte et réservation.

J'ai aussi commis des erreurs pour voir si elles remontaient et étaient refusées.

Register

Email Dossier@projet.com
 Password
 Nom Dépoint
 Prenom Marc
 Adresse 6 boulevard pas loins
 Tel 07 80 53 62 94
 Age 49
 Sexe Homme ▾
 Profession barman
 Agree terms ☒
 Register
 Already have an account? [Sign in.](#)

On peut voir que le « client » créé est bien connecté et présent en BDD avec un mot de passe hashé.



	id	email	roles	password	nom	prenom	adresse	tel	age	sexe	profession	is_verified
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	4	mixocoder@gmail.com	[]	\$2y\$13\$GG3uA.o9h8/foqzsm6lfi.jSs22bq1ZnCcmT/0PajAg...	mixo	codér	6 boulevard du code	NULL	23	Homme	barman	0
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	5	Dossier@projet.com	[]	\$2y\$13\$E4zoYC59MvCfpH/eZOKu/bSsoNl8KBA62JpFQEBFYI...	Dépoint	Marc	6 boulevard pas loins	07 80 53 62 94	49	Homme	barman	0

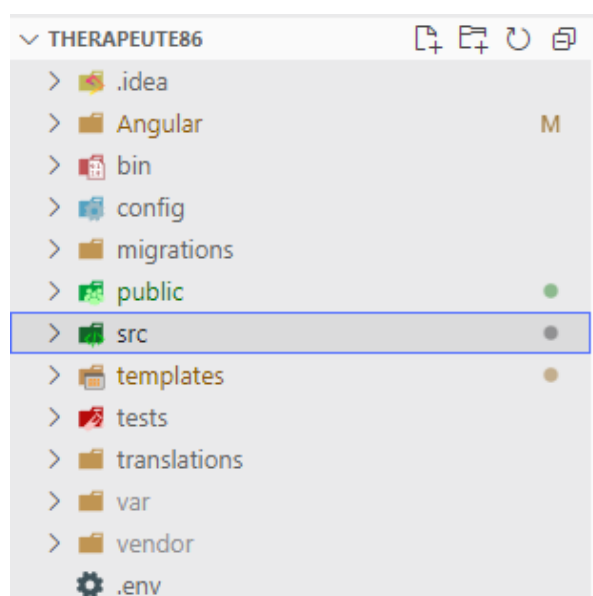
Le Défis Techniques : calendrier Angular

Apprentissage du développement frontend et backend séparé

Problème : Le défi majeur a été d'apprendre à développer le frontend (Angular) et le backend (Symfony) de manière séparée, une compétence essentielle pour ce projet.

Solution : J'ai consacré du temps à étudier les meilleures pratiques pour le développement frontend et backend séparés, en m'adaptant rapidement pour intégrer ces nouvelles compétences dans le projet (j'ai changé 2 à 3 fois l'architecture du projet).

J'ai fini par opter pour une architecture avec Angular présent dans Symfony.



API Symfony ne renvoyant aucune donnée

Problème : Initialement, mon API Symfony ne renvoyait aucune donnée, un problème identifié en utilisant Postman.

Solution : J'ai examiné et corrigé le backend Symfony pour résoudre ce problème, en assurant que l'API renvoie correctement les données. (Voir page 19)

Problème spécifique au calendrier Angular

Problème avec la Classe Seance (Symfony)

Problème : Absence des champs time et isReserved dans l'entité Seance, nécessaires pour le calendrier Angular.

Solution : Ajout de ces champs à l'entité Seance et mise à jour de la base de données avec une migration Symfony.

```
#[ORM\Column(type: Types::BOOLEAN, nullable: true)]  
#[Groups('seance')]  
private ?bool $isReserved = false;
```

Erreur de routage dans Angular

Problème : Erreurs de routage dues à une mauvaise utilisation de RouterModule.forRoot(routes) dans Angular.

Solution : Correction de la configuration du routage en déplaçant RouterModule.forRoot(routes) dans le fichier main.ts pour une configuration globale.

Import de RouterModule et de routes dans main.ts

```
import { bootstrapApplication } from '@angular/platform-browser';  
import { RouterModule } from '@angular/router';  
import { AppComponent } from './app/app.component';  
import { routes } from './app/app.routes';  
import { importProvidersFrom } from '@angular/core';  
import { HttpClientModule } from '@angular/common/http';  
import { CalendarModule, DateAdapter } from 'angular-calendar';  
import { adapterFactory } from 'angular-calendar/date-adapters/date-fns';  
  
bootstrapApplication(AppComponent, {  
  providers: [  
    importProvidersFrom(RouterModule.forRoot(routes)),  
    importProvidersFrom(HttpClientModule),  
    importProvidersFrom(CalendarModule.forRoot({  
      provide: DateAdapter,  
      useFactory: adapterFactory,  
    })),  
  ]  
}).catch(err => console.error(err));
```

Erreur CORS entre Angular et Symfony

Problème : Problèmes CORS empêchant la communication entre Angular et l'API Symfony.

Solution : Configuration du `nelmio/cors-bundle` dans Symfony pour autoriser les requêtes inter-domaines depuis Angular.

Exemple d'une recherche faite avec
Chat GPT et son application

Configuration de CORS dans Symfony

Vous pouvez gérer CORS dans Symfony en utilisant le bundle `'nelmio/cors-bundle'`.
Voici les étapes pour l'installer et le configurer :

1. Installer le bundle :

Ouvrez un terminal et exécutez la commande suivante dans votre projet Symfony pour installer le bundle :

```
bash
```

[Copy code](#)

```
composer require nelmio/cors-bundle
```

```
paths:
    '^/api/':
        allow_origin: [ 'http://localhost:4200' ]
        allow_headers: [ 'X-Custom-Auth' ]
        allow_methods: [ 'POST', 'PUT', 'GET', 'DELETE' ]
        max_age: 3600
        hosts: [ '^api\.' ]
```

Invalid Date dans Angular

Problème : Objets Date invalides dans Angular à cause d'un format de date incorrect de l'API Symfony.

Solution : Vérification et validation du format des dates renvoyées par l'API Symfony et ajustement dans Angular pour la création d'objets Date.

```
getAvailableSeances(): Observable<Seance[]> {
  return this.http.get<any[]>(this.apiUrl).pipe(
    map(data => {
      console.log('Données reçues de l\'API:', data);
      return data.map(item => ({
        id: item.id,
        prestationId: item.prestationId,
        dateReservation: new Date(item.dateReservation),
        heureReservation: item.heureReservation,
        statut: item.statut,
        time: item.time || '',
        isReserved: item.isReserved || false
      }));
    })
  );
}
```

Code de mon service Angular

Problèmes de dépendance Angular

Problème : Erreur NullInjectorError : No provider for HttpClient! indiquant une dépendance manquante.

Solution : Importation du HttpClientModule dans le fichier main.ts pour fournir le service HttpClient nécessaire.

```
import { bootstrapApplication } from '@angular/platform-browser';
import { RouterModule } from '@angular/router';
import { AppComponent } from './app/app.component';
import { routes } from './app/app.routes';
import { importProvidersFrom } from '@angular/core';
import { HttpClientModule } from '@angular/common/http';
import { CalendarModule, DateAdapter } from 'angular-calendar';
import { adapterFactory } from 'angular-calendar/date-adapters/date-fns';

bootstrapApplication(AppComponent, {
  providers: [
    importProvidersFrom(RouterModule.forRoot(routes)),
    importProvidersFrom(HttpClientModule),
    importProvidersFrom(CalendarModule.forRoot({
      provide: DateAdapter,
      useFactory: adapterFactory,
    })),
  ],
}).catch(err => console.error(err));
```

Import de HttpClientModule

Et au final ?

J'ai bel et bien un calendrier Angular que j'ai pu ajouter à mon application Symfony, mais il ne renvoie pas les données voulues pour la réservation. Il y a encore des améliorations visuelles à effectuer.

	Sunday Jan 14	Monday Jan 15	Tuesday Jan 16	Wednesday Jan 17	Thursday Jan 18	Friday Jan 19	Saturday Jan 20
12 AM							
1 AM							
2 AM							
3 AM							
4 AM							
5 AM							
6 AM							
7 AM							
8 AM							
9 AM							
10 AM							
11 AM							
12 PM							
1 PM							

Vues mensuelles et hebdomadaires
du calendrier

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

La Sécurité

Authentification et autorisation

Importance : Assurer que seuls les utilisateurs autorisés puissent accéder à certaines fonctionnalités comme réserver, modifier ou annuler des séances.

Implémentation : Utiliser des mécanismes d'authentification et vérifier les autorisations pour chaque action.

Vérification de l'authentification de l'utilisateur `if (!$this->getUser())`

```
#[Route('/planning/{seanceId}/reserve', name: 'planning_reserve')]
public function reserveSlot(Seance $seance): JsonResponse
{
    if (!$this->getUser()) {
        return new JsonResponse(['error' => 'Authentification requise.'], Response::HTTP_UNAUTHORIZED);
    }

    if (!$this->isSeanceFree($seance)) {
        return new JsonResponse(['error' => 'La séance est déjà réservée.'], Response::HTTP_BAD_REQUEST);
    }

    $reservation = new Reservation();
    $reservation->setSeance($seance);
    $reservation->setUtilisateur($this->getUser());
    $reservation->setStatut('confirmed');

    $this->entityManager->persist($reservation);
    $this->entityManager->flush();

    if ($seance->getReservations()->count() === 1) {
        $seance->setStatut('reserve');
        $this->entityManager->flush();
    }

    return new JsonResponse(['id' => $reservation->getId()]);
}
```

Affichage conditionnel des liens de navigation

```
<div class="navbar-collapse" id="navbar-collapse">
  <ul class="navbar-nav">
    <li class="nav-item"><a href="{{ path('app_a_propos') }}" class="nav-link">À propos</a></li>
    <li class="nav-item"><a href="{{ path('app_prestations') }}" class="nav-link">Prestations</a></li>
    <li class="nav-item"><a href="{{ path('seances_disponibles') }}" class="nav-link">Réservation</a></li>

    {% if is_granted('IS_AUTHENTICATED_FULLY') %}
    <li class="nav-item"><a href="{{ path('app_logout') }}" class="nav-link">Déconnexion</a></li>
    {% else %}
    <li class="nav-item"><a href="{{ path('contact') }}" class="nav-link">Contact</a></li>
    <li class="nav-item"><a href="{{ path('app_register') }}" class="nav-link">Créer un compte/Se connecter</a></li>
    {% endif %}
  </ul>
</div>
```

Validation des données

Importance : S'assurer que les données reçues du client sont valides et ne contiennent pas de données malveillantes.

Implémentation : Valider toutes les entrées côté serveur avant de les traiter ou de les stocker.

On peut utiliser le composant Validator de Symfony pour valider les données de la requête.

```
use Symfony\Component\Validator\Constraints as Assert;

$data = json_decode($request->getContent(), true);
$constraints = new Assert\Collection([
    'seanceId' => new Assert\NotNull(),
]);
$errors = $validator->validate($data, $constraints);

if (count($errors) > 0) {

}
```

Protection contre les attaques CSRF et XSS

Importance : Prévenir les attaques qui exploitent la confiance de l'utilisateur dans le navigateur.

Implémentation : Utiliser des jetons CSRF pour les formulaires et échapper/encoder les données affichées pour éviter le XSS.

```
{# Afficher les données en les échappant pour prévenir le XSS #}
<div>{{ data|e }}</div>
```

Communication sécurisée

Importance : Protéger les données transmises entre le client et le serveur.

Implémentation : Utiliser HTTPS pour chiffrer les données en transit.

Gestion des sessions

Importance : Assurer la sécurité des sessions utilisateur.

Implémentation : Utiliser des jetons de session sécurisés et mettre en place une expiration de session appropriée.

```
session:
  handler_id: null
  cookie_secure: auto
  cookie_samesite: lax
```

Cela est géré par Symfony et Angular.
Il faut s'assurer d'avoir correctement configuré les paramètres de session.

Stockage sécurisé des mots de passe

Importance : Stocker les mots de passe de manière sécurisée en utilisant un hashage.

Implémentation : Utiliser un PasswordHasher.

```
use Symfony\Component\PasswordHasher\Hasher\UserPasswordHasherInterface;
```

```
$user->setPassword(
    $userPasswordHasher->hashPassword(
        $user,
        $form->get('plainPassword')->getData()
    )
);
```

				id	email	roles	password
<input type="checkbox"/>	Éditer	Copier	Supprimer	4	mixocoder@gmail.com	[]	\$2y\$13\$GG3uA.o9h8/foqzsm6lfi.jSs22bq1ZnCcmT/0PajAg...
<input type="checkbox"/>	Éditer	Copier	Supprimer	5	Dossier@projet.com	[]	\$2y\$13\$E4zoYC59MvCfpH/eZOKu/.bSsoNI8KBA62JpFQEBFYI...

Évitez les spams

Importance : Évitez que Pauline loupe un message important d'un client à cause du nombre de spams.

Implémentation : On peut utiliser une clé reCAPTCHA ou encore utiliser un mailer.

Exemple de Captcha

```
<div class="g-recaptcha" data-sitekey="votre_clé_publique"></div>
  <button type="submit">Envoyer</button>
</form>
<script src="https://www.google.com/recaptcha/api.js" async defer></script>
```

Exemple de mailer

```
###> symfony/mailer ###
MAILER_DSN=null://null
#penser a le modifier MAILER_DSN=smtp://apikey:SG.xxxxxx@smtp.sendgrid.net
###< symfony/mailer ###
```

```
$this->emailVerifier->sendEmailConfirmation(
    'app_verify_email',
    $user,
    (new TemplatedEmail())
        ->from(new Address('noreply@therapeute86.fr', 'Thérapeute Mail Bot'))
        ->to($user->getEmail())
        ->subject('Please Confirm your Email')
        ->htmlTemplate('registration/confirmation_email.html.twig')
);
```

```
#[Route('/verify/email', name: self::VERIFY_EMAIL_ROUTE)]
public function verifyUserEmail(Request $request, TranslatorInterface $translator): Response
{
    $this->denyAccessUnlessGranted('IS_AUTHENTICATED_FULLY');

    // validate email confirmation link, sets User::isVerified=true and persists
    try {
        $this->emailVerifier->handleEmailConfirmation($request, $this->getUser());
    } catch (VerifyEmailExceptionInterface $exception) {
        $this->addFlash('verify_email_error', $translator->trans($exception->getReason(), [], 'VerifyEmailBundle'));

        return $this->redirectToRoute('app_register');
    }

    $this->addFlash('success', 'Your email address has been verified.');
```

```
framework:
  mailer:
    dsn: '%env(MAILER_DSN)%'
```

```
// Inform the user that an email has been sent for verification
$this->addFlash('success', 'A verification email has been sent to your email address. Please check your inbox.');
```

```
return $this->redirectToRoute('app_login');
} elseif ($form->isSubmitted()) {
    $this->addFlash('error', 'Registration failed. Please check your input and try again.');
```

```
return $this->render('registration/register.html.twig', [
    'registrationForm' => $form->createView(),
]);
```

Perspectives et améliorations possibles

Améliorations techniques du projet

Finalisation du Calendrier : Un objectif immédiat est de finaliser le calendrier de réservation pour qu'il enregistre les réservations en base de données. Cela rendra le système pleinement opérationnel.

Ajout d'un historique de réservations : Le développement d'une fonctionnalité d'historique permettrait aux utilisateurs et à l'administrateur de suivre les réservations passées.

Gestion multi-administrateurs : Actuellement, Pauline est le seul administrateur créé directement en base de données. L'ajout d'une fonctionnalité pour gérer plusieurs administrateurs permettra à différentes personnes du même cabinet d'avoir le même site de réservation.

Intégration avec le calendrier Google : Combiner le calendrier de réservation avec le calendrier Google que Pauline utilise pour une synchronisation et une gestion des rendez-vous plus efficaces.

Création de comptes : Possibilité de se connecter et de créer un compte grâce à son compte Google.

Impact du projet

Facilitation des réservations pour les clients : Le projet a considérablement simplifié le processus de réservation pour les clients de Pauline, leur permettant de réserver en ligne sans avoir à appeler. Cela a entraîné un gain de temps significatif pour les clients et a offert à Pauline une tranquillité d'esprit en réduisant la fréquence des appels téléphoniques.

Présence en ligne : Pauline a maintenant un site qui peut lui servir de site vitrine pour montrer les compétences acquises lors de ses formations.

Bilan personnel et compétences développées

Autonomie et souhait d'avoir un tuteur

Appréciation de l'autonomie : Travailler de manière autonome sur ce projet a été une expérience enrichissante, me permettant de mieux connaître ma capacité à me concentrer pleinement sur les objectifs et les défis du métier.

Souhait d'un mentorat : Bien que l'autonomie soit gratifiante, j'aurais aimé avoir un tuteur pour me guider, en particulier dans les phases plus complexes du projet. Un mentor aurait pu m'aider à affiner mes compétences, à approfondir ma compréhension des aspects techniques et sûrement à gagner du temps dans les moments d'incompréhension.

Développement de compétences techniques

Gestion de projet frontend et backend : J'ai appris à gérer des projets impliquant des développements frontend et backend séparés : une compétence essentielle dans le développement web moderne.

Adoption de nouvelles technologies : L'utilisation de nouveaux bundles et technologies a été un aspect clé de mon apprentissage, me permettant d'élargir mes connaissances et de rester à jour avec les tendances actuelles du développement web.

Importance de la veille technologique dans le développement web

Dans un domaine en constante évolution comme le développement web, la veille technologique est cruciale. Ce projet a renforcé ma compréhension de l'importance de rester informé des dernières tendances et technologies. En tant que développeur, je veillerais à continuer mon apprentissage et à explorer de nouvelles méthodologies et outils. Je planifie de participer à des webinaires et de rejoindre des communautés professionnelles (AFUP) pour rester à l'avant-garde de la technologie. Je vais aussi continuer à me renseigner comme je le faisais déjà avant le début de ma formation en regardant des vidéos et en écoutant des podcasts de façon régulière sur le domaine du développement et de la tech.

Remerciements au Jury de l'Examen DWWM

Avant de conclure ce rapport, je tiens à exprimer ma gratitude aux lecteurs de ce dossier. Je suis reconnaissant pour le temps et l'attention que vous consacrez à ce rapport et j'espère qu'il vous donnera un aperçu clair et complet de mon travail et des compétences développées tout au long de ce projet. L'opportunité de présenter mes réalisations et d'obtenir vos retours est une étape précieuse dans mon parcours professionnel.

Conclusion

Synthèse et accomplissement

Le développement de l'application web pour Pauline Le DU Boucard a été une aventure professionnelle enrichissante, jalonnée de défis techniques. L'immersion complète dans chaque aspect du développement web, du maquettage à l'implémentation de fonctionnalités avancées comme le calendrier de réservation Angular et l'intégration d'APIs Symfony, a significativement enrichi mes compétences. Ce travail marque un départ dans ma carrière de développeur.

La transition de WordPress à un cadre technologique plus complexe avec Symfony et Angular a été cruciale. Elle m'a permis de me prouver que je suis capable de faire de « vrais » site web, mais a aussi souligné l'importance de l'adaptabilité dans le secteur technologique. Cette étape a mis en évidence ma capacité à apprendre et à rechercher rapidement ainsi qu'à appliquer de nouvelles connaissances dans un projet concret.

Apprentissage et croissance

Ce projet m'a permis de comprendre des aspects techniques du développement web, mais aussi de l'importance de la gestion de projet et de la communication efficace avec les clients. Les échanges avec Pauline ont renforcé mes compétences en communication et en compréhension des besoins des clients, essentielles pour tout développeur.

La confrontation à divers défis, qu'ils soient techniques ou liés à la gestion du projet, a été une source d'apprentissage inestimable. Chaque obstacle surmonté a renforcé mes compétences et ma confiance de développeur.

Impact et perspectives d'avenir

Le projet a eu un impact notable, tant sur mon développement personnel et professionnel que sur l'apport à l'entreprise de Pauline. La mise en place d'un système de réservation en ligne va simplifier et améliorer l'expérience client.

Je suis désormais enthousiaste à l'idée de mettre en pratique mes compétences dans de nouveaux domaines du développement. Mon objectif à court terme est de terminer le calendrier Angular. Je prévois également de continuer mes études en alternance pour approfondir mes connaissances dans le domaine du développement et expérimenter le travail en entreprise. Je veux acquérir de nouvelles connaissances et compétences. Mon ambition est de continuer à évoluer en tant que développeur, d'explorer de nouvelles technologies et de contribuer à des projets innovants et stimulants car je reste avant tout un passionné.

« I WANT TO BROADEN MY HORIZON »