

**DOSSIER PROJET**

**AMBOKA Limay**

**Développeur Web et Web mobile**

## Remerciements

**Je tiens à exprimer mes plus sincères remerciements à Mr Demba SOUMARE, mon formateur, pour son soutien constant, sa motivation et sa capacité à me faire apprécier et maîtriser des nouveaux langages de programmation, tels que JavaScript, ainsi que les framework React JS et Symfony .**

**Je tiens également à remercier chaleureusement l'association ASSOFAAC pour m'avoir offert une chance de poursuivre mes études en informatique .**

**Enfin, un grand merci à mes camarades de classe qui ont apprécié mes différentes implications pour nos projets communs.**

**Ce projet n'aurait pas pu se concrétiser sans leur présence et leur encouragement.**

**Merci à tous pour votre soutien et votre confiance.**

## Sommaire

<b>Remerciements.....</b>	<b>1</b>
<b>Sommaire.....</b>	<b>2</b>
<b>Introduction.....</b>	<b>3</b>
<b>1. Compétences du référentiel couvertes par le projet.....</b>	<b>3</b>
<b>A - Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité.....</b>	<b>3</b>
<b>B - Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité.....</b>	<b>3</b>
<b>2.Résumé du projet.....</b>	<b>4</b>
<b>Cahier des charges.....</b>	<b>5</b>
<b>1.Contexte du projet.....</b>	<b>5.2</b>
<b>Documentation technique.....</b>	<b>6</b>
<b>A - Spécificité technique.....</b>	<b>6</b>
<b>B-Fonctionnalité du site Médi'Avenir.....</b>	<b>7</b>
<b>C-Outils et technologies utilisés.....</b>	<b>9</b>
<b>D-Charte Graphique.....</b>	<b>12</b>
<b>Gestion de Projet.....</b>	<b>13</b>
<b>1.Organisation.....</b>	<b>13</b>
<b>A-Méthode Agile.....</b>	<b>13</b>
<b>B-Sauvegarde.....</b>	<b>14</b>
<b>C-Maquette.....</b>	<b>15</b>

D-Diagramme.....	15
------------------	----

E-ModèleVueContrôleur(MVC).....	17
---------------------------------	----

## 2.

Conception.....	18
-----------------	----

A-Front-End.....	18
------------------	----

B-Back-End.....	19
-----------------	----

C-Sécurité.....	25
-----------------	----

D-Test.....	27
-------------	----

Projet.....	28
-------------	----

Conclusion.....	29
-----------------	----

Annexes.....	30
--------------	----

## **Introduction**

### **1. Compétences du référentiel couvertes par le projet**

#### **A - Développer la partie front-end d'une application web ou web mobile sécurisée**

Toutes les compétences du titre professionnel Développeur Web et Web Mobile seront analysées grâce à ce projet.

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

#### **B - Développer la partie back-end d'une application web ou web mobile sécurisée**

- 1. Créer une base de données**
- 2. Développer les composants d'accès aux données**
- 3. Développer la partie back-end d'une application web ou web mobile**
- 4. Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce**

## 2. Résumé du projet

Le projet "Hôtel Hypnose" se concentre sur le développement du back-end pour une réservation d'hôtel, basé sur mon ECF. L'objectif principal était de mettre en place un système CRUD (Create , Read ,Update, Delete) complet, couvrant la création, la lecture, la mise à jour et la suppression de données, avec des interfaces adaptées pour les utilisateurs et les administrateurs.

Le défi consistait à garantir une expérience utilisateur optimale tout en respectant les exigences du cahier des charges, assurant ainsi le bon fonctionnement et l'efficacité du système de gestion de l'hôtel.

Dans ce contexte, j'ai travaillé sur la conception et la mise en œuvre des fonctionnalités back-end, en veillant à ce que chaque action effectuée par les utilisateurs soit traitée de manière sécurisée et efficace. La gestion des données, la sécurité et la performance ont été des aspects clés du développement, avec une attention particulière portée à l'optimisation des requêtes et à la protection des informations sensibles.

En résumé, le projet "Hôtel Hypnose" a permis de mettre en pratique mes compétences en développement back-end, en fournissant une solution robuste et fonctionnelle pour la gestion des ressources d'une réservation en ligne, tout en répondant aux attentes des utilisateurs et en assurant une expérience utilisateur fluide et satisfaisante.

**Cahier des charges 1. Contexte du projet**

## **2. Documentation technique**

### **A - Spécificité technique**

#### **Serveur :**

- **Apache (Version 2.4.56)**
- **PHP (Version 8.4)**
- **Extension PHP : PDO**

#### **Pour le Front-End :**

- **HTML5**
- **CSS3**
- **Javascript**

#### **Pour le Back-End :**

- **Symfony (version 7)**
- **Mysql (version 8.4)**
- **PhpMyAdmin**
- **Doctrine / ORM**
- **PHP 8.2**
- **EasyAdmin**
- **Postman**



## **B- Fonctionnalité du site Hôtel hypnose**

### **US1 : Créer son compte**

**Utilisateurs concernés : Tout public voyageur**

-Le client désirant s'inscrire devra renseigner : un nom, un prénom, un email valide, une date de naissance et un mot de passe sécurisé résistant aux injections SQL.

### **US2 : Se Connecter**

**Utilisateurs concernés : inscrit , employé (administrateur)**

- Les utilisateurs concernés se connectent sur leur plateforme respective : l'inscrit se connecte sur l'interface inscrit et l'employé l'interface administrateur .

### **US3 : Découvrir le catalogue**

**Utilisateurs concernés : Client, inscrit , employé (administrateur)**

Les utilisateurs concernés auront accès à l'intégralité du catalogue des destinations ainsi que des hôtels.

Si besoin, il peut rechercher un livre précis par son titre grâce à une barre de recherche ou filtrer par genre (ex : romance, science-fiction, fantastique, etc.)

### **US4 : Réserver un hôtel**

**Utilisateurs concernés : inscrit , employé (administrateur)**

-Si le livre est disponible, un bouton associé permettra alors de l'emprunter

- Si l'inscrit clique sur le bouton, le livre ne sera plus accessible pour d'autres utilisateurs.

-Lorsque l'inscrit vient pour récupérer son livre à la médiathèque, un employé confirmera l'emprunt.

- Si, au bout de 3 jours, le livre n'a toujours pas été récupéré par l'inscrit, l'emprunt sera considéré comme annulé et le livre sera de nouveau disponible dans le catalogue.

### **US5 : Ajouter un livre dans le catalogue**

**Utilisateurs concernés : employé (administrateur)** L'employé peut ajouter un livre qui a pour caractéristique : - Un titre,

-Une image de la première de couverture,

- Une date de parution,

-Une description,

-Un auteur,

-Un genre.

## **US6 : Voir les disponibilités**

**Utilisateurs concernés : inscrit , employé (administrateur)**

**-Les employés de la médiathèque voient la liste des emprunts en cours - De son côté, l'inscrit voit aussi la liste des livres qu'il a empruntés.**

## **US7 :Payer sa location**

**Utilisateurs concernés : inscrit , employé (administrateur)**

**-Lorsqu'un inscrit vient rendre un livre sur place, un employé confirme la remise. Automatiquement, le livre devient à nouveau accessible dans le catalogue.**

## **US8 : Modifier son profil**

**Utilisateurs concernés : inscrit , employé (administrateur)**

**- Les utilisateurs concernés peuvent consulter et modifier leur profil**

## **C- Outils et technologies utilisés**

- **Pour l'organisation et la gestion du projet :**

**J'ai utilisé GitHub pour la gestion de versions du code source .**

**Pour la conception et la création de maquettes d'interfaces utilisateur, j'ai fait appel à Figma , qui m'a permis de visualiser et de prototyper les différents aspects visuels du projet.**

**Afin d'organiser les tâches, suivre les progrès, j'ai opté pour Trello comme outil de gestion de projet visuel, créant des tableaux, des listes et des cartes pour une organisation claire.**

**Pour la création de diagrammes, j'ai utilisé Whimsical pour visualiser les idées et les processus de manière structurée.**

**En tant qu'éditeur de code principal, Visual Studio Code a été essentiel pour le développement du projet, offrant des fonctionnalités avancées d'édition, de débogage et d'intégration avec d'autres outils de développement.**

- **Pour le développement de l'environnement de projet :**

**J'ai choisi Symfony comme framework PHP pour le développement de l'application, offrant une structure robuste et des fonctionnalités avancées pour la création d'applications web. Ainsi que l'extension Easy Admin de Symfony qui permet de générer l'interface administrateur.**

**Pour tester et valider les API développées, j'ai utilisé Postman comme outil de test d'API, permettant de vérifier les requêtes HTTP, les réponses et les performances des API.**

**Pour la gestion de la base de données et des entités, j'ai intégré Doctrine, un ORM (Object-Relational Mapping) pour Symfony, facilitant la manipulation des données et la communication avec la base de données.**

**Pour l'administration de la base de données, j'ai utilisé phpMyAdmin , une interface web permettant de gérer les bases de données MySQL de manière visuelle et conviviale.**

**Enfin, pour la gestion du système de gestion de base de données relationnelle, j'ai opté pour PhpMyAdmin pour stocker et gérer les données du projet.**

**D - Charte Graphique**

**Logo de Hôtel Hypnose**

## **Gestion de Projet**

### **1. Organisation**

#### **A - Méthode Agile**

**Pour organiser mon projet , j'ai utilisé Trello en faisant la méthode Agile :**

**J'ai créé des blocs de cartes permettant de définir le statut des événements (A FAIRE , EN COURS , FAIT, A REVOIR ) et de créer des étiquettes en fonction des catégories (front-end, back-end, organisation, bonus , backlog (idée) et à revoir).**

#### **B - Sauvegarde**

**J'ai réalisé mes sauvegarde avec les commandes github qui sont généralement :**

**git add . : pour ajouter les fichier modifiés**

**git commit -m 'nom des fichier modifiés ' : pour donner un nom à la version des les fichier modifier**

**git status : regarder le statut des fichier qui ont été modifier**

**git push : pour envoyer les fichier modifier au repository**

**git pull : pour récupérer les modification récentes**

#### **C - Maquette**

**Pour ma maquette, j'ai élaboré trois interfaces distinctes adaptées aux utilisateurs, avec des versions conçues pour les appareils ordinateur et mobile. En plus de ces interfaces, j'ai également créé des wireframes pour planifier la structure et l'agencement des éléments visuels de l'application.**

**(Voir Annexe 1 - Maquette Desktop et Mobile) D - Diagramme**

#### **• Diagramme de classe**

**- La table User représente les utilisateurs du système, avec des attributs tels que l'identifiant, le rôle, le nom et le prénom.**

**(Voir Annexe 2 - Diagramme de classe)**

- **Diagramme de séquence**

**Pour le diagramme de séquence détaille le flux d'actions et de messages**

**entre les différents acteurs du système, mettant en lumière les étapes clés telles que la demande d'emprunt, la validation de l'emprunt, la notification de retour, et la mise à jour de la disponibilité du livre. Il permet de visualiser de manière séquentielle comment les utilisateurs interagissent avec le système pour réaliser ces opérations, offrant ainsi une vue dynamique du processus d'emprunt et de retour de livres. (Voir Annexe 3 - Diagramme de Séquence)**

- **Diagramme de cas d'utilisations**

**Le diagramme de cas d'utilisation schématise les fonctionnalités du site ,**

**c'est-à-dire les utilisateurs concernés et les actions qui leurs ont été attribuées**

## **E - Modèle Vue Contrôleur (MVC)**

**Le modèle Vue-Contrôleur (MVC) est un concept architectural qui permet de structurer une application en trois composants principaux : la Vue, le Contrôleur et le Modèle.**

**La Vue représente l'interface utilisateur à travers laquelle les utilisateurs interagissent avec l'application, (template) le Contrôleur gère les requêtes des utilisateurs et orchestre les actions à effectuer, (controller) Tandis que le Modèle représente les données et la logique métier de l'application.(base de données) .(Voir Annexe 4 - MVC Medi'Avenir : Exemple Page Contact )**

## **2. Conception**

### **A - Front-End**

- **Responsive**

Pour garantir à l'utilisateur un accès optimal sur toutes les plateformes, j'ai implémenté le design responsive pour le site Hôtel hypnose. Celui-ci est désormais disponible en versions adaptées aux mobiles, aux ordinateurs de bureau et aux tablettes . (Voir Annexe 5 - Responsive )

- **Arborescence**

Pour structurer mon arborescence, j'ai mis en place plusieurs interfaces, notamment tel que :

- **Interface Générale :**

Cette partie de l'application constitue le point d'entrée pour tous les visiteurs. Elle offre une vue d'ensemble de la plateforme, mettant en avant les services et les contenus de manière accessible à tous.

- **Interface Inscrits :**

Les utilisateurs inscrits bénéficient d'une interface personnalisée qui leur permet d'accéder à des fonctionnalités supplémentaires. Ils peuvent personnaliser leur expérience, interagir avec le contenu et profiter de services exclusifs.

- **Interface Administrateurs (Employés) :**

Les administrateurs, qui sont également des employés, disposent d'une interface spécifique avec des outils avancés de gestion et de supervision. Cette interface leur permet de gérer efficacement les différentes facettes de la plateforme pour assurer un fonctionnement optimal et répondre aux besoins des utilisateurs.  
(Voir Annexe 6 - Arborescence )

- **SEO**

J'ai mis en œuvre plusieurs pratiques de référencement naturel (SEO) pour améliorer la visibilité du site web .Voici un aperçu des actions que j'ai entreprises :

- **Optimisation des balises META :** J'ai soigneusement rédigé des balises <title> et <meta description> uniques et pertinentes pour chaque page du site, en incluant des mots-clés stratégiques.

- **Optimisation des images :** J'ai optimisé les images en utilisant des balises ALT descriptives et en les compressant pour améliorer le temps de chargement des pages et l'expérience utilisateur.

- **Structure du site :** J'ai veillé à ce que la structure du site soit claire et organisée, en utilisant des balises HTML appropriées pour structurer le contenu et faciliter l'indexation par les moteurs de recherche.

## **(Voir Annexe 7 - SEO ) B - Back-End**

- **Création projet en Symfony**

- **Installation du projet Symfony** : J'ai installé Symfony en utilisant Composer, un gestionnaire de dépendances pour PHP. J'ai exécuté la commande `"composer create-project symfony/skeleton nom_du_projet"` pour créer un nouveau projet Symfony.

- **Configuration initiale** : Après l'installation de Symfony, j'ai configuré les paramètres de base tels que la base de données, l'environnement, et d'autres configurations spécifiques au projet dans le fichier `.env` et `config/packages`.

- **Gestion des packages avec Composer** : J'ai utilisé Composer pour gérer les dépendances et les packages nécessaires à mon projet Symfony. J'ai ajouté des packages supplémentaires en utilisant la commande `"composer require package_nom"` pour installer et mettre à jour les dépendances.

- **Utilisation des bundles Symfony** : J'ai intégré des bundles Symfony pour ajouter des fonctionnalités spécifiques à mon projet, tels que MakerBundle pour générer du code rapidement, DoctrineBundle pour la gestion de la base de données, et d'autres bundles selon les besoins du projet.

- **Configuration des routes et des contrôleurs** : J'ai défini les routes et les contrôleurs pour gérer les requêtes HTTP et afficher les vues correspondantes en suivant les bonnes pratiques de Symfony.

**(Voir Annexe 8 -Création Projet Symfony )**

- **Conception base de donnée**

Avant de démarrer le projet, j'ai procédé à la création d'une base de

données sur PhpMyadmin en utilisant la commande

`"create database nomBaseDeDonnees;"`. Ensuite, j'ai créé un utilisateur administrateur en utilisant la commande `"create user nomUtilisateur identified by motDePasse"` et je lui ai accordé les privilèges nécessaires avec la commande `"GRANT ALL PRIVILEGES ON . TO *.*"`.

Par la suite, j'ai configuré la connexion entre la base de données et mon projet Hôtel hypnose en modifiant le fichier `.env`. Ce fichier, étant caché, assure la sécurité des informations de connexion à la base de données.

Par la suite, j'ai utilisé la commande `"php bin/console make:entity"` pour créer une entité dans mon projet. Chaque entité représente une table dans la base de données, et j'ai configuré les propriétés en définissant leur nom, leur type de données, la longueur maximale (pour les champs texte), ainsi que leur caractère nullable ou non. Ensuite, toujours avec cette commande, j'ai établi les relations entre les tables Reservation et User.

Enfin, j'ai effectué une sauvegarde de la base de données en utilisant la commande "php bin/console doctrine:migrations:migrate" et mis à jour avec la commande "php bin/console doctrine:schema:update".  
(Voir Annexe 9 - Conception base de donnée)

#### **Conception formulaire Formulaire Inscription (Register) :**

Pour créer mon formulaire d'inscription, j'ai utilisé la commande `php bin/console make:registration-form` dans Symfony. Après avoir exécuté cette commande, j'ai suivi les étapes pour configurer le formulaire d'inscription, définir les champs requis et générer automatiquement le code associé à ce formulaire. Ensuite, j'ai personnalisé le formulaire d'inscription en ajoutant des champs supplémentaires, en modifiant les validations et en adaptant les vues générées pour correspondre aux besoins spécifiques de mon application.

(Voir Annexe 10 - Formulaire Inscription (Register))

#### **- Formulaire Authentification (Login) :**

Pour créer mon formulaire d'authentification, j'ai utilisé la commande php bin/console make:auth dans Symfony.

Après avoir lancé cette commande, j'ai suivi les étapes pour configurer le type d'authentification souhaité, définir les options nécessaires et générer automatiquement le code associé à l'authentification. Ensuite, j'ai personnalisé le formulaire d'authentification en adaptant les routes, les contrôleurs et les vues générées pour répondre aux besoins spécifiques de mon application.

(Voir Annexe 11 - Formulaire Authentification (Login )

#### **- Formulaire Contact :**

Pour le formulaire de contact, j'ai créé une entité Contact puis créé un CRUD avec la commande "php bin/console make:crud " et j'ai apporté des modifications manuelles pour personnaliser le formulaire selon les besoins spécifiques de mon application .



CREATE	READ	UPDATE	DELETE
Autorise le visiteur à s'inscrire sur le site en fournissant les informations nécessaires pour créer un compte.	Permet au visiteur de consulter le catalogue et de visualiser les produits, services ou informations disponibles.	_____	_____

(Voir Annexe 12 - Formulaire Contact )

- **Conception CRUD**

Le CRUD permet aux utilisateurs d'effectuer des actions de création, de lecture, de mise à jour et de suppression (Create, Read, Update, Delete) sur les données de l'application. Grâce à ces fonctionnalités, les utilisateurs peuvent interagir de manière complète avec les informations présentes dans le système.

- **CRUD Visiteur :**

- **CRUD Admin:**

CREATE	READ	UPDATE	DELETE
L'administrateur peut ajouter des lieux de visites, des hôtels...	L'administrateur peut consulter les détails des comptes utilisateurs créés, ainsi que les informations sur les réservations en cours.	L'administrateur peut mettre à jour les informations des comptes utilisateurs si nécessaire, modifier les détails des réservations et mettre à jour son propre profil administrateur	L'administrateur peut supprimer des comptes utilisateurs, des lieux ou d'autres données si besoin, en respectant les politiques de protection des données et de confidentialité.

**(Voir Annexe 13 - CRUD Admin )**

## **C - Sécurité**

- **Routes**

**J'ai créé une interface pour les utilisateurs inscrits et une interface distincte**

**pour les administrateurs. J'ai également ajouté et nommé des routes pour accéder aux pages respectives, permettant à l'administrateur d'accéder uniquement à son interface d'administration et aux utilisateurs inscrits à leur interface dédiée.**

**(Voir Annexe 6 & 14)**

- **Privilèges**

J'ai mis en place des privilèges distincts pour les rôles 'admin' et 'user', ce qui

me permet de définir des autorisations spécifiques pour chaque type d'utilisateur en fonction de leur rôle attribué.

En effet, le visiteur n'a pas de rôle défini puisqu'il consulte simplement la page. J'ai attribué le rôle 'admin' à l'administrateur car c'est lui qui a des droits étendus de gestion et de supervision. De même, j'ai attribué le rôle 'user' aux utilisateurs inscrits car ils ont des autorisations limitées par rapport à l'administrateur.

(Voir Annexe 14)

- **RGPD & CNIL**

J'ai renforcé la sécurité en hachant les mots de passe des utilisateurs, une pratique de cryptage qui rend les données illisibles et sécurisées en cas d'accès non autorisé. Les administrateurs ne sont pas en mesure de visualiser ou de modifier les mots de passe des utilisateurs, ce qui garantit la confidentialité et la sécurité des informations sensibles. Ces mesures contribuent à assurer la confidentialité, l'intégrité et la sécurité des informations personnelles des utilisateurs, en conformité avec les exigences du RGPD en matière de protection des données.

(Voir Annexe 10 & Annexe 15 )

- **Base de donnée**

Dans le cadre de la sécurisation de la base de données de mon projet, j'ai mis en place plusieurs mesures pour garantir la protection des données sensibles des utilisateurs. Tout d'abord, j'ai utilisé des pratiques de cryptage pour sécuriser les informations stockées dans la base de données.

J'ai également configuré des niveaux d'accès et des permissions stricts pour limiter l'accès aux données uniquement aux utilisateurs autorisés.

De plus, j'ai mis en œuvre des techniques de sauvegarde régulières pour prévenir la perte de données et assurer la disponibilité des informations en cas d'incident.

(Voir Annexe 16 )

- **Formulaire**

De même, lors de l'authentification des utilisateurs, une clé token est

automatiquement générée pour renforcer la sécurité. Le formulaire d'authentification intègre des fonctionnalités de sécurité avancées pour garantir l'identification fiable et sécurisée des utilisateurs.

(Voir Annexe 11 )

- **Message de journalisation (Log)**

Lorsque je démarre mon projet Symfony, les messages de journalisation s'affichent dans le terminal. Ces messages fournissent des informations sur le fonctionnement de l'application, y compris les événements, les erreurs et les opérations en cours. (Voir Annexe 17 )

## **D -Test**

- **Test des routes avec Postman**

J'ai utilisé l'application Postman pour réaliser des tests sur les routes de l'application. Pour les routes accessibles aux visiteurs, j'ai effectué des tests en utilisant la méthode GET sur la route '/catalogue' afin de vérifier l'accès aux catalogues par les visiteurs (voir Annexe 18).

- **Test des fonctionnalités des utilisateurs avec Phpmyadmin**

En ce qui concerne les fonctionnalités des utilisateurs, j'ai également effectué des tests avec PhpMyAdmin .

Pour tester l'ajout de nouveaux éléments par l'administrateur, j'ai réalisé une requête SQL et vérifié les résultats dans PhpMyAdmin. (voir Annexe 19)

- **Test du formulaire de Contact**

Enfin, j'ai testé la fonctionnalité du formulaire de contact pour les visiteurs afin de vérifier s'ils peuvent envoyer un message avec succès .

Et lorsque le visiteur a envoyé le message celui ci est dans la base de donnée (voir Annexe 12 & Annexe 20)

## **Projet**

### **Conclusion**

- **Ce que ce projet m'a appris ?**

Ce projet m'a offert une expérience enrichissante dans le développement d'une application web avec Symfony. J'ai acquis des compétences précieuses en concevant une base de données, en créant des entités et des contrôleurs, en gérant les routes et les vues, et en implémentant des fonctionnalités essentielles pour répondre aux besoins des utilisateurs. Cette expérience m'a permis de développer une meilleure compréhension de la structure et du fonctionnement global d'une application Symfony.

- **Les points à améliorer ?**

En termes d'améliorations, je suis conscient des aspects sur lesquels je pourrais me concentrer pour renforcer mes compétences. Je souhaite approfondir mes connaissances en matière de gestion avancée des entités, d'optimisation des requêtes SQL, d'implémentation de tests unitaires et fonctionnels, ainsi que d'amélioration de l'interface utilisateur pour offrir une expérience utilisateur optimale.

- **Qu'est ce que je vais faire plus tard ?**

Pour l'avenir, je suis déterminé à poursuivre mes études dans le domaine de l'informatique avec une spécialisation en développement front-end au niveau Bac +3. Je souhaite approfondir mes compétences en HTML, CSS, JavaScript et les frameworks front-end modernes, tout en explorant les dernières technologies et tendances du développement front-end. Mon objectif est de me préparer pour une carrière réussie en tant que développeur front-end et de contribuer à la création d'expériences web innovantes et attrayantes.

## **Annexes**

### **Annexe 1 : Maquette version Desktop & Mobile**

- Maquette Desktop
- Maquette Mobile

### **Annexe 2 : Diagramme de classe**

### **Annexe 3 : Diagramme de Séquence**

### **Annexe 4 : MVC Hôtel hypnose**

- Exemple Page Contact
- Page Contact Twig (Template)
- Page Contrôleur Accueil (Controlleur)
- Page Repository Contact (Modèle)

### **Annexe 5 : Responsive**

- Version Mobile
- Version Desktop
- Version Tablette

### **Annexe 6 : Arborescence**

- Interface Général
- Interface Admin
- Interface Inscrit

### **Annexe 7 : SEO**

- Optimisation des balises META
- Optimisation des Images
- Structure du site

## **Annexe 8 : Création projet Symfony**

- Installation du projet Symfony :
- Gestion des packages avec Composer :
- Installation de Twig
- Utilisation des bundles Symfony :
- Configuration des routes et des contrôleurs :

## **Annexe 9 : Conception base de donnée**

- Création base de donnée & l'utilisateur
- Accorder les droits à l'utilisateur
- Liaison base de donnée au projet Medi'Avenir
- Création des entités
- Liaison des Entités Livre et User dans l'entité Emprunt
- Mise à jour de la base de donnée

## **Annexe 10 : Formulaire Inscription (Register)**

- Contrôleur Inscription
- Twig Inscription
- Form Inscription

## **Annexe 11 : Formulaire Authentification (Login)**

- Contrôleur Authentification
- Twig Authentification
- Sécurité et gestion de l'authentification



## **Annexe 12 : Formulaire Contact**

- Entité et Form Contact
- Route Contact

## **Annexe 13 : CRUD ADMIN avec EasyAdmin**

- Exemple CRUD Admin pour le livre

## **Annexe 14 : Privilège**

- Mise en place des rôles dans le fichier security.yaml
- Controller Admin
- Controller Inscrit

## **Annexe 15 : RGPD & CNIL**

- Liste des Utilisateurs
- Détails de l'utilisateur
- Table User dans PhpMyadmin

## **Annexe 16 : Sauvegarde de la base de donnéeAnnexe 17 : Journal (Log)**

## **Annexe 18 : Test des routes avec Postman**

- Test Route Visiteur

## **Annexe 19 : Test des fonctionnalités des utilisateurs avec Phpmyadmin**

## **Annexe 20 : Test Formulaire Contact**