

DOSSIER PROFESSIONNEL (DP)

Nom de naissance

▶ Regnier

Nom d'usage

▶ Entrez votre nom d'usage ici.

Prénom

▶ Sylvain

Adresse

▶ 6 avenue du marechal ney 13011 Marseille

Titre professionnel visé

Développeur web et web mobile

MODALITE D'ACCES :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.
Ce titre est délivré par le Ministère chargé de l'emploi.

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.



<http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

Développer la partie front-end d'une application web ou web mobile sécurisée	p.	5
▶ Installer et configurer son environnement de travail en fonction du projet web ou web mobile	p.	5
▶ Maquetter des interfaces utilisateur web ou web mobile	p.	6
▶ Réaliser des interfaces utilisateur statiques web ou web mobile	p.	14
▶ Développer la partie dynamique des interfaces utilisateur web ou web mobile	p.	18
Développer la partie back-end d'une application web ou web mobile sécurisée	p.	24
▶ Mettre en place une base de données relationnelle	p.	24
▶ Développer des composants d'accès aux données SQL et NoSQL	p.	29
▶ Développer des composants métier coté serveur	p.	38
▶ Documenter le déploiement d'une application dynamique web ou web mobile	p.	41
Titres, diplômes, CQP, attestations de formation <i>(facultatif)</i>	p.	43
Déclaration sur l'honneur	p.	46
Documents illustrant la pratique professionnelle <i>(facultatif)</i>	p.	48
	p.	49

EXEMPLES DE PRATIQUE PROFESSIONNELLE

DOSSIER PROFESSIONNEL (DP)

Activité-type 1

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

Exemple n°1 ► Installer et configurer son environnement de travail en fonction du projet web ou web mobile

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Installation de :


Visual Studio Code




Bootstrap


Symfony



-Composer
-Figma

2. Précisez les moyens utilisés :

Un ordinateur et une connexion internet

3. Avec qui avez-vous travaillé ?

seul

4. Contexte

Nom de l'entreprise, organisme ou association ► *Afpa*

Chantier, atelier, service ► Formation

Période d'exercice ► Du : 25/09/2023 au : 01/02/2024

5. Informations complémentaires (facultatif)

Activité-type 1

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

Exemple n° 1 ► Maquetter des interfaces utilisateur web ou web mobile

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Dans le cadre de la formation « Développeur web et web mobile » j'ai créé une maquette pour mon projet personnel VETOTOIL en fonction du cahier des charges.

Pour la réalisation de cette étape il m'a fallu plusieurs étapes :

- Bien prendre en compte les demandes du cahier des charges sur la partie visuelle
- Choix de l'application pour la création des maquettes
- Réalisation de la maquette

I-Analyse du cahier des charges :

VETOTOIL a pour objectif de faciliter la prise de rendez-vous entre professionnels (vétérinaires et toiletteurs) et les particuliers.

Dans une société où tout va vite, attendre au téléphone pour prendre un rendez-vous est totalement dépassé.

Code couleur :

Navbar et Footer :

-en dégradé de pink->purple

Navbar et Footer écriture :

-white

Mot important :

-Purple

Icones :

-Purple

Couleur Générale d'écriture :

-Black

Police d'écriture :

-Roboto

II- Choix de l'application de création des maquettes

Plusieurs applications sont disponibles sur le marché mais je me suis arrêté sur l'application Figma (<https://www.figma.com/fr/>)

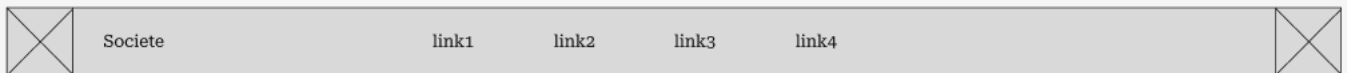
III Réalisation des maquettes.

DOSSIER PROFESSIONNEL (DP)

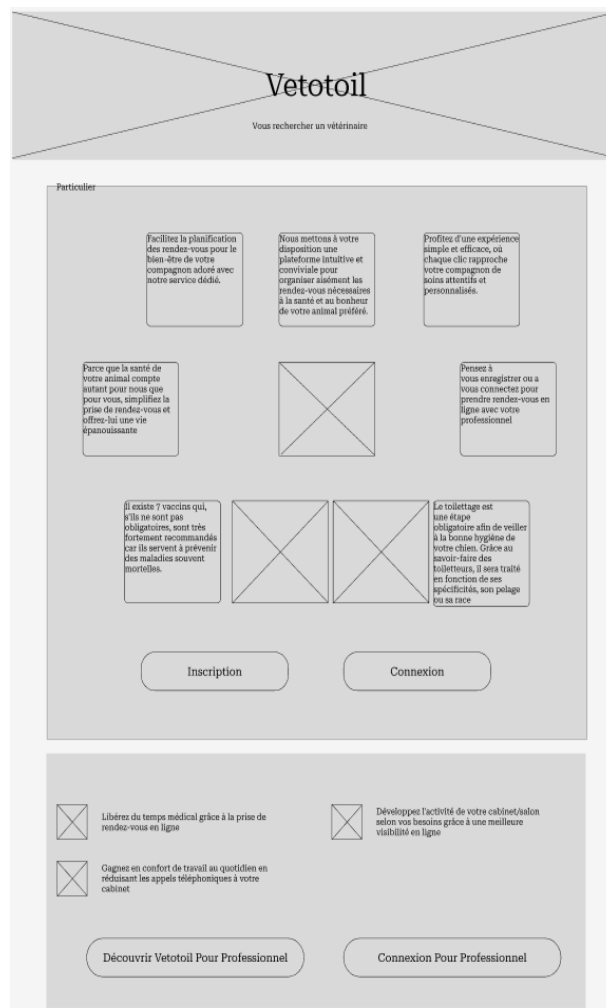
Avant de commencer, j'ai réalisé un croquis à la main pour me rendre compte du rendu de la page souhaitée.

Ensuite je suis passé à la création de la Wireframe en commençant par le mobile first

Dans un premier temps j'ai fait un header avec une navbar qui restera positionner au top durant la navigation.

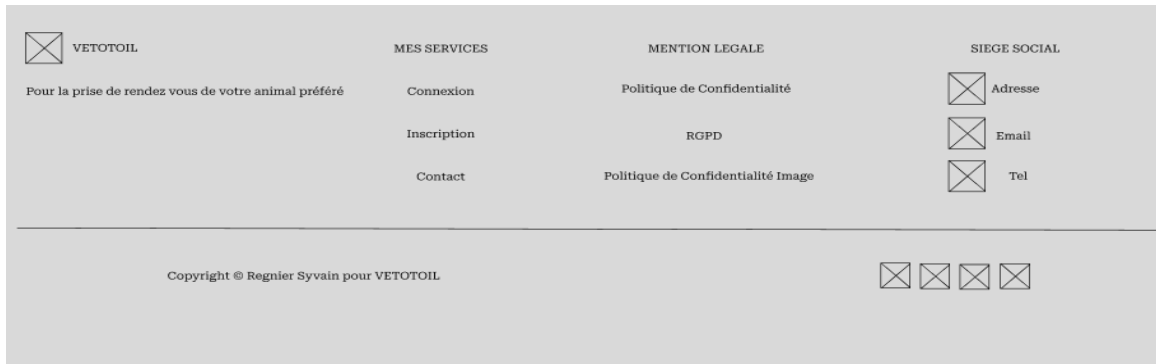


Ensuite j'ai élaboré la partie home de la page d'accueil, cela correspond à la page que les utilisateurs verront toujours en premier, donc j'ai pris soins que la page ne soit pas surchargée d'information, mais que des informations importantes y figure.

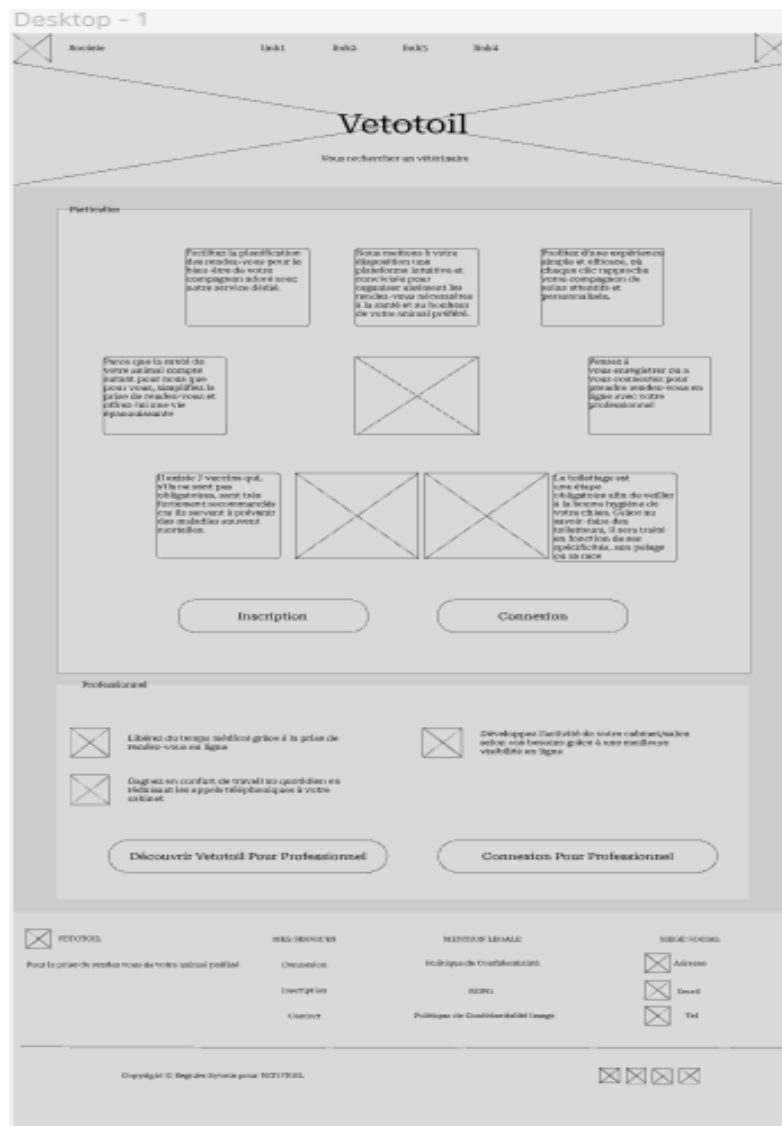


Et pour finir le footer qui est le bas de page qui est une partie importante pour retrouver certaine information importante

DOSSIER PROFESSIONNEL (DP)



Voici la wireframe complète de la page principale :



Etant donné que le responsive est tout aussi important voici le wireframe responsive avec le bouton « hamburger »



Voici le maquetage visuel de rendu


VETOTOIL

[Accueil](#)
[Connexion](#)
[Inscription](#)
[Contact](#)





VETOTOIL
 Vous recherchez un...

Particulier

Des soins à votre disposition sans plateformes tierces et connexions pour garantir l'authenticité des rendez-vous, répondre à la demande et assurer la qualité de nos services.

Faciliter la prise de rendez-vous pour le bien-être de nos compagnons grâce à notre service dédié.

Profiter d'une expérience simple et efficace, à chaque clic, répondre votre demande de soins, trouver un vétérinaire et prendre rendez-vous.

Faciliter la prise de rendez-vous pour vous, simplifier la prise de rendez-vous et offrir la meilleure expérience possible.

Permettre à nos vétérinaires de mieux connaître leur clientèle et de mieux répondre à leurs besoins.

Il existe 2 versions qui s'adaptent à votre besoin, soit une version pour les professionnels et une version pour les particuliers.



La version pour les professionnels est conçue pour les vétérinaires afin de faciliter la prise de rendez-vous, la gestion des consultations, la gestion des dossiers médicaux et la gestion des comptes.


Frame 20

[Inscription](#)
[Connexion](#)

Professionnel

Professionnel

 **Libérez** du temps médical grâce à la prise de rendez-vous en ligne.

 **Développez** l'activité de votre cabinet, selon vos besoins grâce à une meilleure visibilité en ligne.

 **Gagnez** en confort de travail au quotidien en réduisant les appels téléphoniques à votre cabinet.

Découvrez Vetotoil pour Professionnel


VETOTOIL

Pour la prise de rendez-vous de votre cabinet vétérinaire.

LES SERVICES :

- Connexion
- Inscription
- Contact

MENTION LEGALE :

- Politique de Confidentialité
- RGPD
- Contact de suppression des données

SIÈGE SOCIAL :

- 14 Avenue Marceline
- 13011 Marseille
- 04 91 31 31 31
- 06 20 20 11 31

Copyright © 2020 Vetotoil pour VETOTOIL.





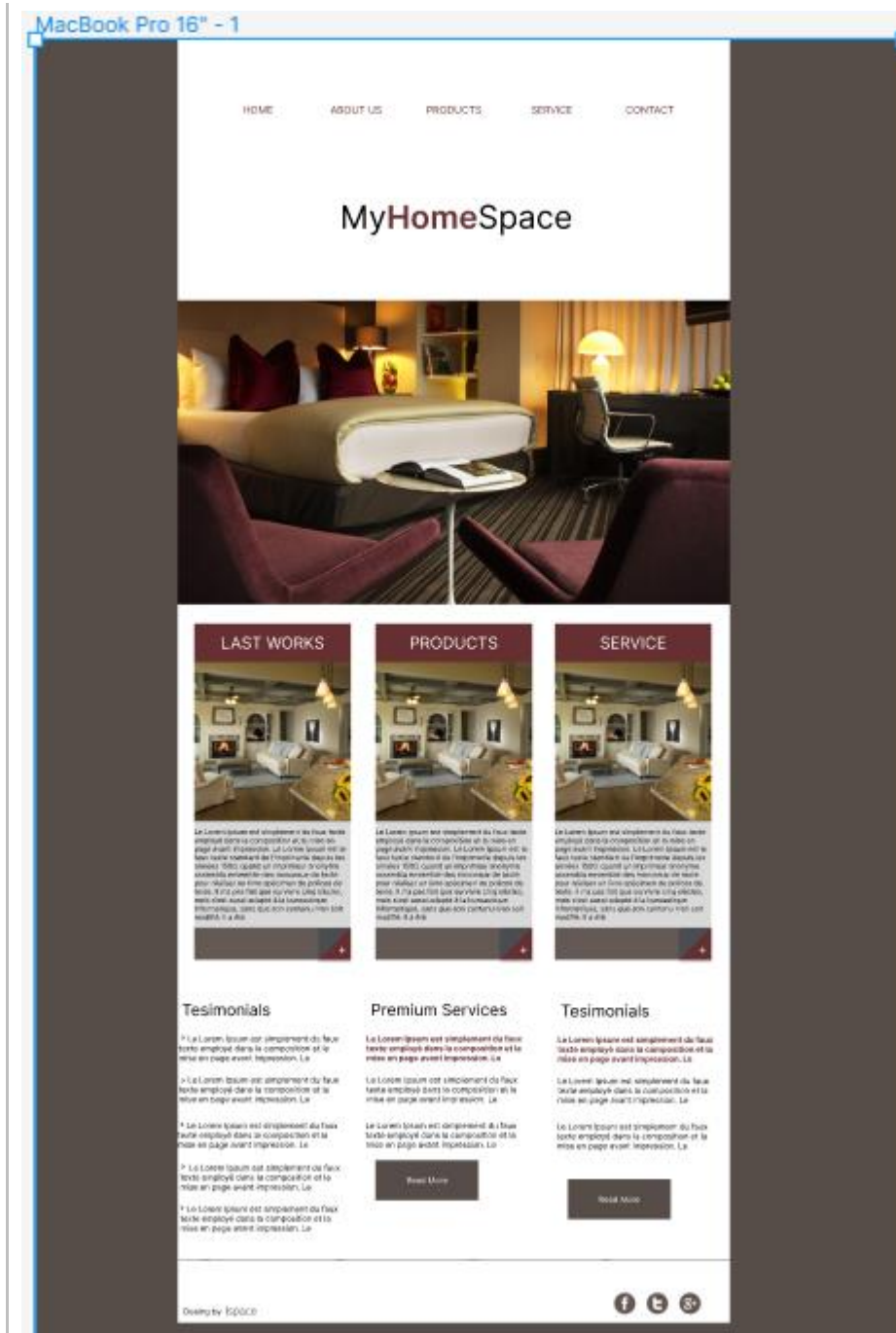

Page 10
11/09/2017

DOSSIER PROFESSIONNEL (DP)



Pour un autre projet du nom MyHomeSpace j'ai réalisé le maquettage ci-joint

DOSSIER PROFESSIONNEL (DP)



2. Précisez les moyens utilisés :

Pour faire le design des maquettes j'ai utilisé Figma
Croquis papier
Générateur d'image Dall-e libre de droit
Cahier des charges pour Vetotoil et l'exercice pour MyhomeSpace
Les polices d'écritures ont été trouvées sur le Google Font

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

seul

4. Contexte

Nom de l'entreprise, organisme ou association ► *Afpa*

Chantier, atelier, service ► formation

Période d'exercice ► Du : 15/11/2023 au : 22/11/2023

5. Informations complémentaires (facultatif)

Activité-type 1

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

Exemple n° 1 ► Réaliser des interfaces utilisateur statiques web ou web mobile

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Pour mon projet personnel j'ai dû faire des pages statiques

J'ai d'abord créé un nouveau projet Symfony puis j'ai créé une vue home qui est étendue à base.html.twig (ce qui me permet de ne pas renseigner par exemple les liens bootstrap dans toutes les pages), dans celle-ci, il y a la structure du HTML standard en modifiant la définition de la langue utilisée pour mon cas le fr pour le français puis j'y ai ajouté une balise meta pour avoir le responsive

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

J'y ai ajouté les liens link ainsi que les scripts du Framework Bootstrap

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/5.1.3/js/bootstrap.bundle.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.2/font/bootstrap-icons.min.css">
```

Et j'ai inclus dans mon body le header et footer ainsi que le block body qui sera le positionnement de toutes mes pages

```
<body>
    {% include "header.html.twig" %}

    {% block body %}
    {% endblock %}

    {% include "footer.html.twig" %}
    {% block javascripts %}
        {% block importmap %}{{ importmap('app') }}{% endblock %}

        <script src="{{ asset('js/home.js') }}" type="module" defer></script>
```

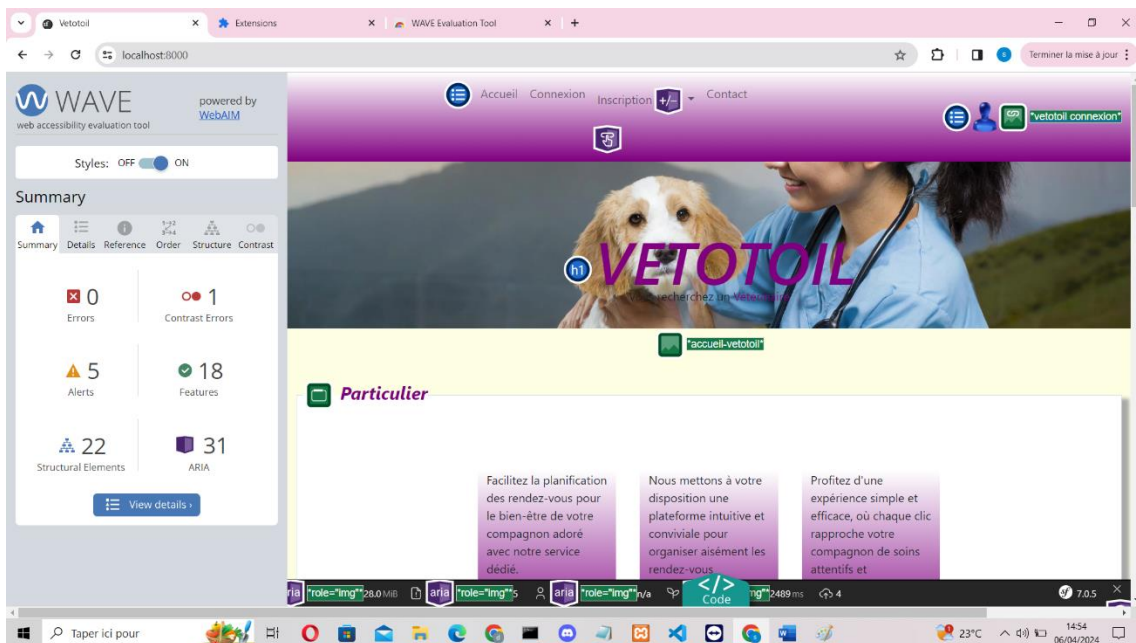
```
{% endblock %}  
</body>
```

Voici une partie de ma page home

```
{% extends 'base.html.twig' %}  
  
{% block title %}Vetotoil  
{% endblock %}  
  
{% block body %}  
  
    <div class="row justify-content-center">  
        <div class="col-12 text-center position-relative">  
              
            <h1 class="position-absolute top-50 start-50 translate-middle grand" style="transform: translate(-50%, -50%); z-index: 1;">VETOTOIL</h1>  
            <p class="position-absolute top-50 start-50 translate-middle mt-5" style="transform: translate(-50%, 50%); z-index: 1;">Vous recherchez un  
                <span id="choix">Vétérinaire</span>  
            </p>  
        </div>  
    </div>  
    <div class="container" id="home">  
  
    {% for message in app.flashes('info') %}  
        <div class="alert alert-info">  
            {{ message }}  
        </div>
```

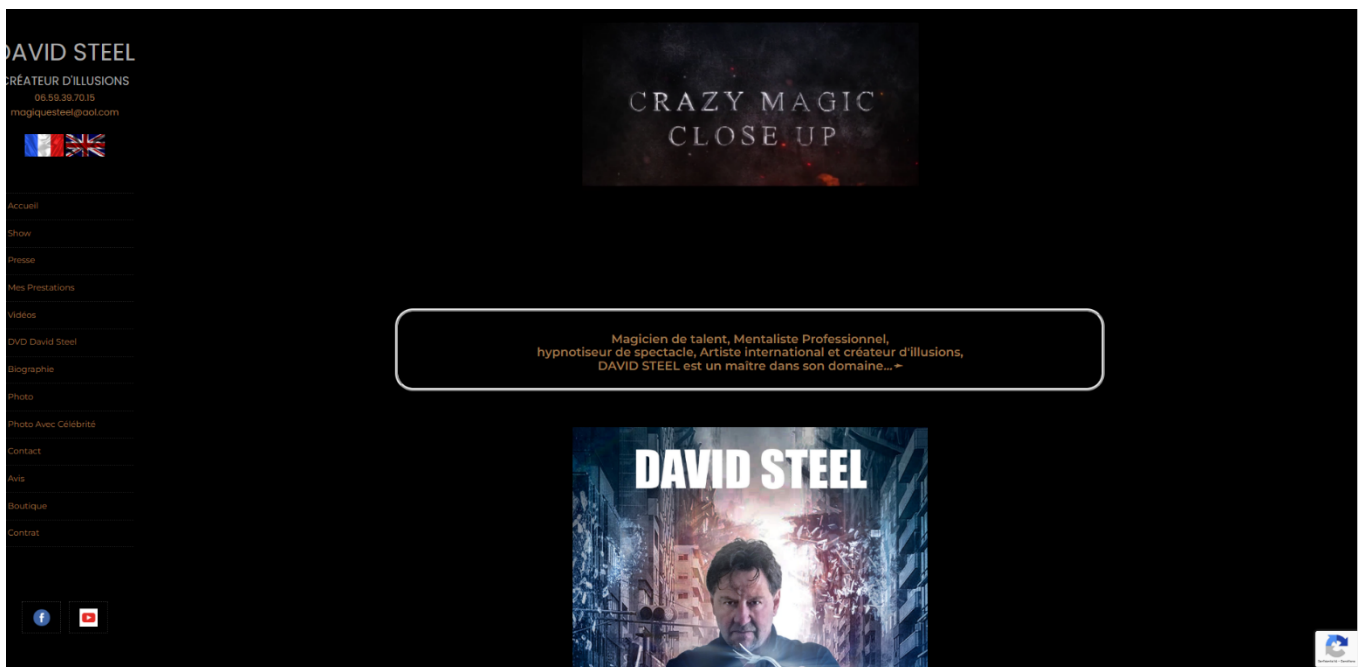
Bootstrap me permet de travailler directement avec des class prédéfini par son Framework ce qui permet aussi de gérer la version responsive plus rapidement qu'avec un site fait uniquement en css. Chaque page nécessite certaine information et balise indispensable pour le SEO (comme par exemple qu'elle est un titre et au moins un <h1> mais aussi pour l'accessibilité pour les personnes en situation de handicap), j'y ai intégré des images ,j'ai dû utilisé du css pour les media query pour les boutons et aussi pour les dégrader de couleur car Bootstrap ne me le permettait pas ,ensuite pour avoir une validation que la page est bien structuré et sans erreurs je l'ai passé au validateur W3C Validator(étant données que c'est du twig j'ai copié le code dans la console google pour faire le test html) qui m'a confirmé qu'il n'y avait pas d'erreur puis j'ai installé le plugin pour google chrome Wave Evaluation tool pour vérifier que l'accessibilité soit correct.

DOSSIER PROFESSIONNEL (DP)



Par le passé j'ai aussi fait un site internet statique pour un magicien en utilisant Bootstrap

<http://www.magicien-davidsteel.fr/>



2. Précisez les moyens utilisés :

VsCode

Page 16
11/09/2017

DOSSIER PROFESSIONNEL-Version du

DOSSIER PROFESSIONNEL (DP)

Symfony
Twig
Html
Css
Bootstrap
Dall-E ->générer des images
Compresseur d'image-> <https://www.iloveimg.com/fr/compresser-image>
Google Chrome pour voir le responsive ainsi que la version pc
W3C Validator
Wave Evaluation Tool

3. Avec qui avez-vous travaillé ?

seul

4. Contexte

Nom de l'entreprise, organisme ou association ► *AFPA*

Chantier, atelier, service ► Formation

Période d'exercice ► Du : *12/02/2024* au : *13/02/2024*

5. Informations complémentaires (facultatif)

Activité-type 1

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

Exemple n° 1 ► Développer la partie dynamique des interfaces utilisateur web ou web mobile

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Pour la partie dynamique j'ai créé un fichier javascript qui s'appelle adresse.js que j'ai mis dans un dossier où il y a tous les fichiers javascripts.

L'objectif de ce script est de dynamiser la saisie de l'adresse par l'utilisateur. À mesure qu'il remplit le champ d'adresse, une requête est envoyée à l'API Adresse de l'État pour proposer des adresses correspondantes. Plus l'utilisateur précise son entrée, plus les suggestions se raffinent, lui permettant de sélectionner l'adresse exacte, le code postal, et la ville d'un simple clic. Cela simplifie le processus de saisie, car l'utilisateur n'a pas à remplir manuellement tous ces champs. Pour contourner les limitations de l'API, qui rejette les requêtes de moins de quatre caractères, j'ai implémenté une fonction vérifiant la longueur de la saisie avant de procéder à la consultation de l'API, évitant ainsi les erreurs CORS.

Dans la page html j'ai mis un ul pour faire mes listes en invisible et en javascript je le fais apparaître et disparaître suivant les interactions des utilisateurs

```
document.addEventListener("DOMContentLoaded", function () {
  //#####
  #####
  //se lance sur page inscription
  //#####
  #####

  let adresse = document.getElementById("patient_adresse_patient");
  let nbrLetter = 0;
  let ulListe = document.getElementById("list");

  adresse.addEventListener("input", async (eventInput) => {
    nbrLetter = eventInput.target.value.length;
    const rue = eventInput.target.value.split(" ").join("+");

    await findAdress(rue, nbrLetter);
  });

  //appel de l'api et affiche le resultat

  async function findAdress(rue, nbrLetter) {
    if (nbrLetter > 4) {
      console.log("ok");
      const response = await fetch(
```

```
        "https://api-adresse.data.gouv.fr/search/?q=" + rue + "&limit=50"
    );
    const adressResult = await response.json();
    console.log(adressResult);
    ulliste.style.display = "block";
    findData(adressResult);
}
}

function findData(adressResult) {
    const labels = [];
    const city = [];
    const cityCode = [];
    const cityName = [];

    for (const recherche of adressResult.features) {
        labels.push(recherche.properties.label);
        city.push(recherche.properties.city);
        cityCode.push(recherche.properties.postcode);
        cityName.push(recherche.properties.name);
    }

    listeLi(labels, city, cityCode, cityName);
}

function listeLi(labels, city, cityCode, cityName) {
    ulliste.innerHTML = "";
    for (let i = 0; i < labels.length; i++) {
        const liListe = document.createElement("li");
        liListe.innerHTML = labels[i];
        liListe.setAttribute("id", i);
        liListe.classList.add("form-list-item");

        ulliste.appendChild(liListe);

        clickLi(liListe, city, cityCode, cityName);
    }
}

function clickLi(liListe, city, cityCode, cityName) {
    liListe.addEventListener("click", (event) => {
        const li = event.target;
        document.getElementById("patient_code_postal_patient").value =
            cityCode[li.id];
        document.getElementById("patient_ville_patient").value = city[li.id];
    });
}
```

```
    adresse.value = cityName[li.id];

    ulliste.innerHTML = "";
    ulliste.style.display = "none";
  });
}
//#####
#####
//verifie le click en dehors du champs de la liste
//#####
#####

document.addEventListener("click", function (event) {
  if (!ulliste.contains(event.target) && event.target !== adresse) {
    ulliste.style.display = "none";
  }
});
});
```

J'ai aussi fait le même procédé pour une inscription d'une entreprise avec le numéro Siret cette fois ci en utilisant l'api Siren de l'Insee

```
document.addEventListener("DOMContentLoaded", function () {
  const url = "https://api.insee.fr/entreprises/sirene/V3/siret/";
  const accessToken = "75c7b058-1c1c-3cdd-8ce2-dba0a65362df";
  const button = document.getElementById("validerSiret");
  button.addEventListener("click", async function () {
    const siret = document.getElementById("societe_siret");
    // console.log("test : ", siret.value);

    siret
      .addEventListener("keydown", function (event) {
        let keyCode = event.which || event.keyCode;

        //#####
        //autoriser que les chiffres et certaines touche comme supprimer
        //#####
        if (
          !(
            (keyCode >= 48 && keyCode <= 57) ||
            (keyCode >= 96 && keyCode <= 105) ||
            [8, 9, 13, 27, 46].includes(keyCode)
          )
        ) {
        }
      }) {
```

```
        event.preventDefault();
    }
    //#####
    //limite a 14chiffres et supprime les lettres
    //#####
    let inputValue = event.target.value.replace(/\D/g, "");
    if (inputValue.length >= 14 && ![8, 46].includes(keyCode)) {
        event.preventDefault();
    }
});
//#####
//recherche ds l api siret de l insee les info
//#####
const headers = {
    "Content-Type": "application/x-www-form-urlencoded",
    Accept: "application/json",
    Authorization: `Bearer ${accessToken}`,
};
try {
    const response = await fetch(url + siret.value, {
        method: "GET",
        headers: headers,
    });
    if (!response.ok) {
        throw new Error("Réponse réseau non OK");
    }
    const resultat = await response.json();
    // console.log("resultat : ", resultat);
    searchInfo(resultat);
} catch (error) {
    console.error("Erreur :", error);
}
});
function searchInfo(resultat) {
    const numeroVoie =
        resultat.etablissement.adresseEtablissement.numeroVoieEtablissement;
    const typeVoie =
        resultat.etablissement.adresseEtablissement.typeVoieEtablissement;
    const libelleVoie =
        resultat.etablissement.adresseEtablissement.libelleVoieEtablissement;
    const complementAdresse =
        resultat.etablissement.adresseEtablissement
            .complementAdresseEtablissement;
    const libelleCommune =
        resultat.etablissement.adresseEtablissement.libelleCommuneEtablissement;
```

DOSSIER PROFESSIONNEL (DP)

```
const codePostal =
    resultat.etablissement.adresseEtablissement.codePostalEtablissement;
const denominationUsuelleEtablissement =
    resultat.etablissement.periodesEtablissement[0]
        .denominationUsuelleEtablissement;
// console.log('denominationUsuelleEtablissement :
',denominationUsuelleEtablissement);
const nomUniteLegale = resultat.etablissement.uniteLegale.nomUniteLegale;
document.getElementById("societe_adresse_societe").value =
    numeroVoie + " " + typeVoie + " " + libelleVoie;
document.getElementById("societe_complement_adresse_societe").value =
complementAdresse;
document.getElementById("societe_code_postal_societe").value = codePostal;
document.getElementById("societe_ville_societe").value = libelleCommune;
document.getElementById("societe_nom_societe").value =
    denominationUsuelleEtablissement;

}

});
```

2. Précisez les moyens utilisés :

VsCode
Fichier avec l'extension .js
Api adresse de l'état
Api Siren de l'insee

3. Avec qui avez-vous travaillé ?

seul

4. Contexte

Nom de l'entreprise, organisme ou association ► AFPA

Chantier, atelier, service ► Formation

Période d'exercice ► Du : 19/02/2024 au : 24/12/2023

DOSSIER PROFESSIONNEL (DP)

5. Informations complémentaires (*facultatif*)

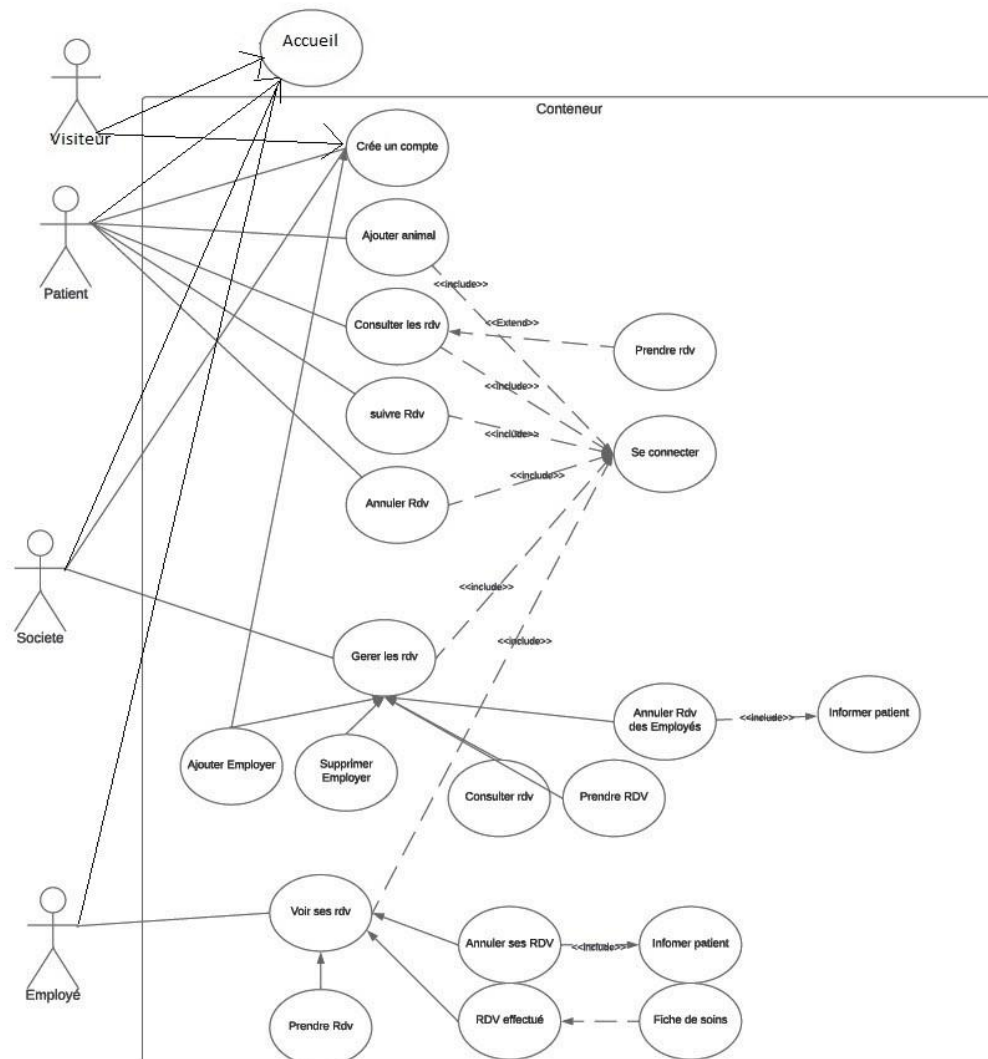
Activité-type 2

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

Exemple n° 1 ► Mettre en place une base de données relationnelle

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

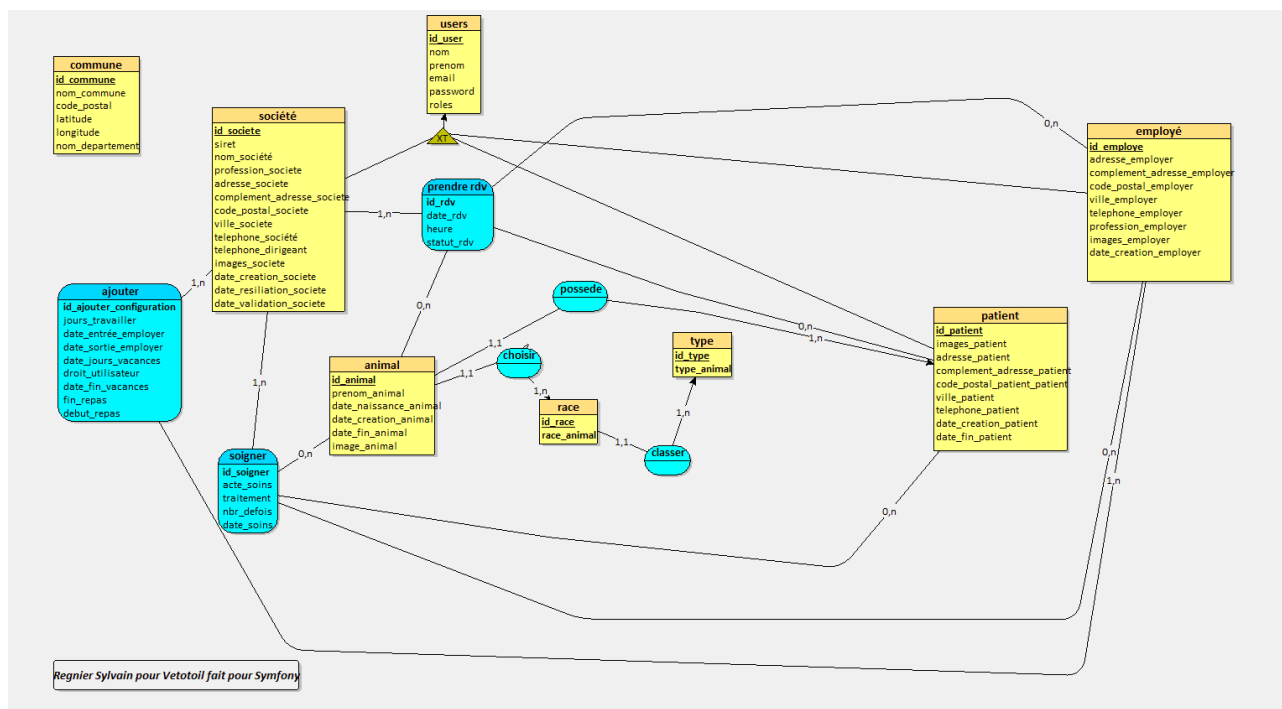
Pour cette tâche je vais me servir de mon projet personnel, dans un premier temps j'ai fait le cahier des charges avec les principales fonctions nécessaires pour le fonctionnement du projet, ensuite j'ai fait un UML de class afin de représenter ses fonctionnalités pour passer de la théorie à une vue plus structurée et voir les incohérences.



DOSSIER PROFESSIONNEL (DP)

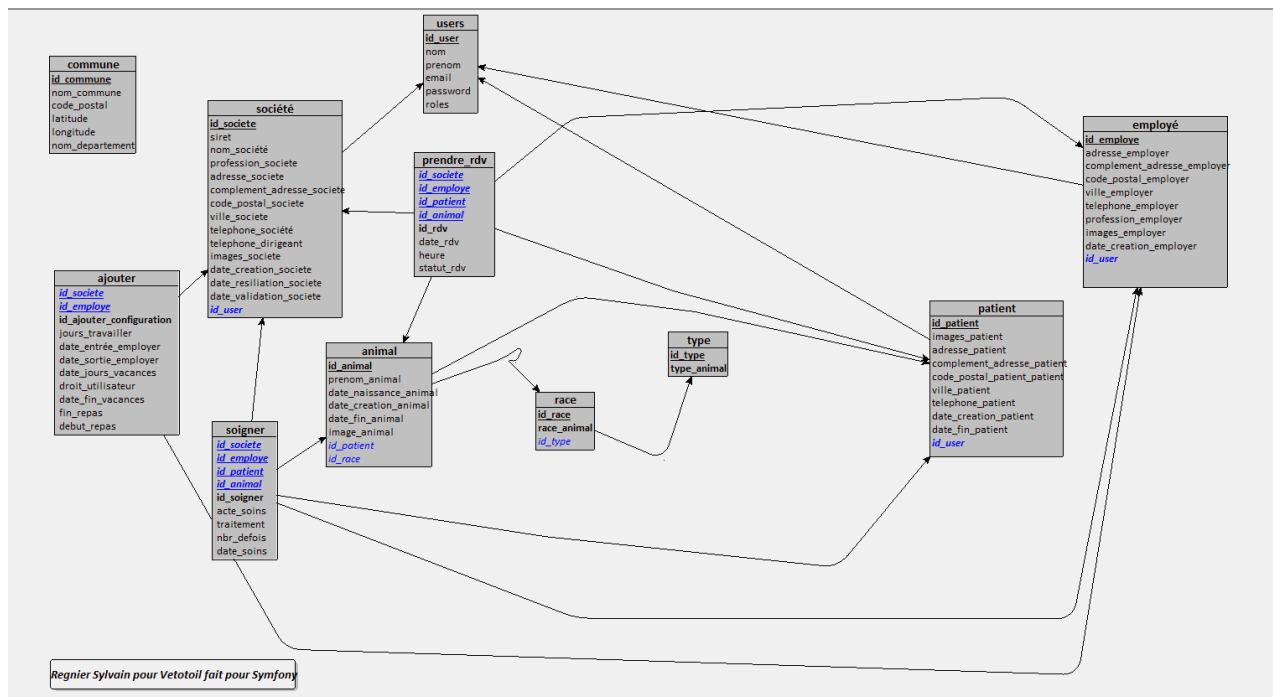
Puis j'ai commencé à élaborer sur une feuille le MCD pour établir les entités avec les informations nécessaires pour chaque entité en évitant les redondances de nom et d'effectuer les associations entre chaque entité mettant une attention particulière aux cardinalités entre les entités et les associations. Une fois effectué et validé j'ai effectué le MLD qui là me permet de voir mes tables à créer et de voir les relations avec les clés étrangères.

MCD :



MLD :

DOSSIER PROFESSIONNEL (DP)



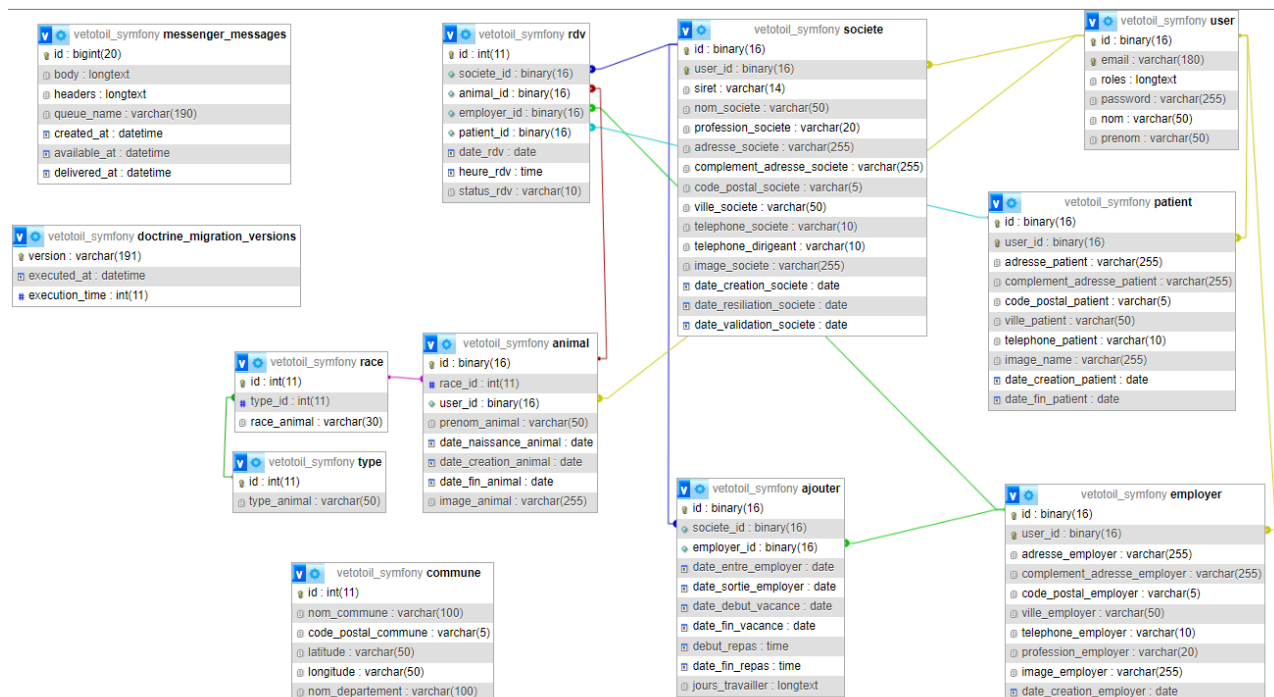
J'ai choisi d'utiliser MySQL avec le client xampp et son module apache pour le serveur web. Je suis ensuite partie sur la création de la base de donnée puis la création des différentes tables et dans chaque table ses différents champs et attribue si c'est du texte(varchar) des nombres(int)... et en fonction des champs les plus utilisés de les indexer pour effectuer des recherches plus rapidement, j'ai aussi spécifié les l'id des tables en auto-incrémente et non Null, pour chaque type de champs j'ai aussi mis les attribut et si il peut être Null ou non.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id_animal	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
2	prenom_animal	varchar(50)	utf8_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
3	date_naissance_animal	date			Non	Aucun(e)			Modifier Supprimer Plus
4	date_creation_animal	date			Oui	NULL			Modifier Supprimer Plus
5	date_fin_animal	date			Oui	NULL			Modifier Supprimer Plus
6	id_race	int(11)			Non	Aucun(e)			Modifier Supprimer Plus
7	id_patient	int(11)			Non	Aucun(e)			Modifier Supprimer Plus
8	image_animal	varchar(255)	utf8_general_ci		Oui	NULL			Modifier Supprimer Plus

Ensuite j'ai effectué les relations entre les tables en fonction du MLD

Une fois tout ce travail effectué j'ai commencé à intégrer via l'interface visuelle de phpMyAdmin, les données, pour m'assurer que les relations étaient bien faites mais aussi pour voir s'il y avait des erreurs, j'ai aussi procédé à des requêtes SQL simple puis plus complexe avec les jointures pour m'assurer du bon fonctionnement de celle-ci, une fois tous cela validé j'ai pu passer au déploiement de ma base de données pour commencer à coder.

DOSSIER PROFESSIONNEL (DP)



Il est aussi tout à fait possible de créer une base de données en SQL, voici un exemple sur la création simple d'une base de données créée en java pour une application Android que j'ai créée

```
private static final String DATABASE_NAME = "mydatabase.db";
private static final int DATABASE_VERSION = 1;
```

```
// Définir les noms de table et les colonnes
public static final String TABLE_NAME = "users";
private static final String COLUMN_ID = "_id";
public static final String COLUMN_NOM = "nom";
public static final String COLUMN_TRIGRAMME = "trigramme";
public static final String COLUMN_STABLECOIN = "stablecoin";
public static final String COLUMN_EXCHANGE = "exchange";
```

Définir la requête SQL pour créer la table

```
private static final String SQL_CREATE_TABLE =
    "CREATE TABLE " + TABLE_NAME + " (" +
        COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
        COLUMN_NOM + " TEXT, " +
        COLUMN_TRIGRAMME + " TEXT, " +
        COLUMN_STABLECOIN + " TEXT, " +
        COLUMN_EXCHANGE + " TEXT);";
```

DOSSIER PROFESSIONNEL (DP)

<https://github.com/sly08051980/androidcrypto/blob/main/app/src/main/java/com/example/crypto/DBHelper.java>

2. Précisez les moyens utilisés :

Word

LuciChart pour faire l'uml

Loopin pour faire le mcd et mld

Xampp pour le server apache et mysql

PhpMyAdmin pour avoir une interface graphique pour créer la base de données et les tables et gérer tous les paramètres de celle-ci ainsi que d'utiliser le SQL pour tester les requêtes

3. Avec qui avez-vous travaillé ?

seul

4. Contexte

Nom de l'entreprise, organisme ou association ► AFPA

Chantier, atelier, service ► Formation

Période d'exercice ► Du : 24/02/2024 au : 31/12/2023

5. Informations complémentaires (facultatif)

J'ai effectué la même chose sous Symfony mais, étant donné qu'avec doctrine, la partie de phpMyAdmin ne me sert plus sur ce type de projet vu que l'on crée les tables via Symfony avec les entités puis on fait les relations de type ManyToMany ManyToOne

Activité-type 2

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

Exemple n° 1 ► Développer des composants d'accès aux données SQL et NoSQL

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai pris le langage PHP pour accéder aux données SQL, aussi bien pour les requêtes SQL qu'en requête DQL de doctrine.

Pour le SQL il faut au départ établir la connexion au serveur

```
try {  
    $this->bd = new PDO('mysql:host=localhost;dbname=vetotoil', 'root', '');  
    $this->bd->query("SET NAMES 'utf8'");  
    $this->bd->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
} catch (PDOException $e) {  
    die('<p>Echec connexion. Erreur [' . $e->getCode() . '] : ' . $e->getMessage() . '</p>');  
}
```

J'utilise les objet pour créer ma connexion avec l'extensions de PHP qui se nome PDO ,je renseigne le type de base de donnée 'mysql' le chemin d'accès qui est host étant donné que je suis en local ,ensuite il y a le nom de la base de donnée puis le root est le nom d'utilisateur pour se connecté a la base de donnée et le dernier champs est pour le mot de passe et surtout, il y a la gestion d'erreur avec le catch (try étant le code qui peut potentiellement générer une erreur),donc si il y a un échec de connexion à la base de donnée l'utilisateur sera informé avec l'exception de l'erreur

Une fois la connexion établie il nous ai possible de faire le CRUD create read update delete ,

Voici un exemple de de create :

Exemple complet est dans l annexe

```
$requete = $this->bd->prepare('INSERT INTO patient (id_patient,  
nom_patient, prenom_patient, adresse_patient, complement_adresse_patient,  
code_postal_patient, ville_patient, telephone_patient, mdp_patient  
,email_patient, date_creation_patient, date_fin_patient,droit_utilisateur_patient)  
VALUES (NULL, :nom, :prenom, :adresse, :complementAdresse, :codePostal,  
:ville, :telephone,  
:pass, :email, :datejour, NULL,:droit)');  
  
$requete->execute([
```

```
        ':nom' => $nom,
        ':prenom' => $prenom,
        ':adresse' => $adresse,
        ':complementAdresse' => $complementAdresse,
        ':codePostal' => $codePostal,
        ':ville' => $ville,
        ':telephone' => $telephone,
        ':pass' => $password,
        ':email' => $email,
        ':datejour' => $date,
        ':droit' => $droit
    ]);

    return [
        'email' => $email,
        'nom' => $nom,
        'prenom' => $prenom,
        'adresse' => $adresse,
        'ville' => $ville,
        'telephone' => $telephone,
        'codePostal' => $codePostal,
        'droit' => $droit
    ];
} catch (PDOException $e) {
    die('Erreur [' . $e->getCode() . '] : ' . $e->getMessage() . '</p>');
}
}
```

Cela est une fonction où je récupère les données saisies par l'utilisateur via un formulaire avec une méthode POST, avant d'insérer les données je vérifie que les champs ne soient pas vides puis je fais un traitement via une fonction pour vérifier qu'il n'y ait pas de tentative d'injection de code voici la fonction

```
function validate_formulaire($data)
{
    $data = trim($data);
    $data = htmlspecialchars($data);
    $data = stripslashes($data);
    $data = strtolower($data);
    return ($data);
}
```

Je prépare ensuite une requête en utilisant la variable définie pour la connexion à la base de données. Je configure la requête pour insérer des données dans la table Patient, spécifiant les champs où les données doivent être insérées. Pour une sécurité accrue, je n'insère pas directement les valeurs dans la

requête, mais j'utilise des paramètres nommés pour chaque champ. Au moment de l'exécution de la requête, je fournis les valeurs correspondantes à ces paramètres à partir de mes variables préalablement traitées. Cette approche me permet non seulement d'insérer les données de manière sécurisée, mais également de retourner certaines informations à mon contrôleur, en fonction de mon code spécifique. Voici une requête read (select pour récupérer des valeurs)

```
try{
    $requete=$this->bd->prepare("SELECT *
    FROM rdv
    INNER JOIN animal ON rdv.id_animal =animal.id_animal
    INNER JOIN ajouter ON rdv.id_employer=ajouter.id_employer
    INNER JOIN race ON animal.id_animal=race.id_race
    INNER JOIN patient ON rdv.id_patient=patient.id_patient
    WHERE rdv.id_employer = :idEmployer
    AND date_rdv BETWEEN CURDATE() AND DATE_ADD(CURDATE(), INTERVAL 31 DAY) AND
    ajouter.date_sortie_employer IS NULL" );
    $requete->execute([':idEmployer'=>$idEmployer]);
    return $requete->fetchAll(PDO::FETCH_ASSOC);
}catch (PDOException $e) {

    die('Erreur [' . $e->getCode() . '] : ' . $e->getMessage());
}
```

Comme précédemment, je prépare une requête dans laquelle je réalise une recherche dans la table RDV, en utilisant des jointures pour accéder aux champs d'autres tables en fonction des identifiants de jointure. Je spécifie ensuite les colonnes à interroger et les valeurs correspondantes. Pour terminer, j'intègre certaines conditions à la requête. Une fois exécutée, je récupère tous les résultats à l'aide de FetchAll pour les envoyer à mon contrôleur, permettant ainsi leur utilisation dans ma vue. J'inclus également une gestion des erreurs pour traiter les cas où une exception serait levée. En ce qui concerne le dql sous symfony

La connexion a la base de donnée se fait à partir du .env

```
DATABASE_URL="mysql://root:@127.0.0.1:3306/vetotoil_symfony?charset=utf8"
```

C'est assez similaire avec le SQL on définit le type de base le chemin d'accès ,le port ,le nom de la base de donnée ainsi que le type de caractère spéciaux que la base de donnée peut reconnaître dans mon cas utf8

Symfony utilise Doctrine qui simplifie la gestion des données et qui permet de formuler des requêtes avec des objets. Puis en créant des formulaires symfony nous permet un gain de temps pour faire des actions simples avec la base de donnée.

Je crée d'abord une entity user dans Symfony avec la commande symfony console make :user , cela est une class qui représente un objet , pour le faire je vais utiliser l'ORM (Mapping objet-relationnel), ainsi

Doctrine va permettre de lier les données de la base de données aux entity et va automatiquement me crée un repository associé à l'entity ,c'est ici que les requêtes sur la table de la base de donnée sera renseigner.

```
#[ORM\Entity(repositoryClass: UserRepository::class)]
#[UniqueEntity(fields: ['email'], message: 'There is already an account with this email')]
class User implements UserInterface, PasswordAuthenticatedUserInterface
{
    #[ORM\Id]
    #[ORM\GeneratedValue('CUSTOM')]
    #[ORM\Column(type: 'uuid',unique:true)]
    #[ORM\CustomIdGenerator('doctrine.uuid_generator')]
    private ?UuidV7 $id = null;

    #[ORM\Column(length: 180, unique: true)]
    private ?string $email = null;

    /**
     * @var list<string> The user roles
     */
    #[ORM\Column]
    private array $roles = [];

    /**
     * @var string The hashed password
     */
    #[ORM\Column]
    private ?string $password = null;

    #[ORM\Column(length: 50)]
    private ?string $nom = null;

    #[ORM\Column(length: 50)]
    private ?string $prenom = null;

    #[ORM\OneToMany(targetEntity: Animal::class, mappedBy: 'user')]
    private Collection $animals;
    #[ORM\OneToOne(mappedBy: "user", cascade: ['persist', 'remove'])]
```

Dans l'entity c'est ici les champ de la table sont renseigné, pour des raisons de sécurité j'ai utilisé l'uuidv7 générer des id aléatoire et unique. Le champ email est renseigné en unique ce qui me permet de ne pas avoir deux adresses email identique. En créant l'entity user le mot de passe sera automatiquement crypté par symfony .

Il y a aussi les annotations qui commencent par **#[ORM** qui sont utilisées pour définir la structure de la table lorsque l'on rajoute **Column**, ensuite on peut ajouter certains paramètres comme la longueur du champ, si il est unique ou nom qui sera renseigné dans la base de données au moment de la migration. Nous pouvons aussi constater qu'il y a aussi des annotations pour les relations pour une ou plusieurs entités **#[ORM\OneToMany]**

Pour créer et gérer des formulaires dans Symfony, la commande `symfony console make:form` est utilisée. Elle génère un formulaire basé sur une entité, qui est une représentation d'une table de la base de données facilitant la manipulation des données. Par exemple, pour insérer ou mettre à jour des données via un formulaire, Symfony simplifie le code :

```
$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {

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

    $entityManager->persist($user);
    $entityManager->flush();

    return $this->redirectToRoute('_profiler_home');
}
```

Bien sûr il faut toujours vérifier les données insérées par l'utilisateur et comme pour le SQL, j'ai établi une fonction de vérification pour la sécurité.

```
$builder->addEventListener(FormEvents::SUBMIT, function (FormEvent $event) {
    $user = $event->getData();

    $user->setNom(validate_form($user->getNom()));
    $user->setPrenom(validate_form($user->getPrenom()));
    $user->setEmail(email_form($user->getEmail()));
    $event->setData($user);
});
```

Symfony permet aussi de créer le CRUD par entité et ensuite nous avons juste à faire pour des requêtes simples des appels comme celui-ci

DOSSIER PROFESSIONNEL (DP)

```
$patient = $entityManager->getRepository(Patient::class)->findOneBy(['user' => $userId]);
```

Ou je demande de rechercher un patient par un id déjà récupéré dans le champ user

Le NOSQL

Mise en service d'une application de notation de bière avec login qui est inséré dans un document et pour les bières on insère les données dans une collection puis dans celle-ci il y a le document attribué à la collection.

- Voici le cahier des charges
Fournir une application web recueillant les avis de consommateurs de bières.
- Exposition de bières permettant aux utilisateurs d'ajouter leurs bières préférées et de liker ou disliker les bières que d'autres partagent.
- Service réservé aux utilisateurs enregistrés

Le front-end de l'application a été développé à l'aide d'Angular et a été précompilé après des tests internes, mais Beer Paradise a besoin d'un développeur back-end pour construire l'API

	Point d'accès	Request body	Type de réponse attendue	Fonction
POST	/api/auth/signup	{ email: string, password: string }	{message: string}	Hachage du mot de passe utilisateur, et ajout de l'utilisateur à la base de données.
POST	/api/auth/login	{ email: string, password: string }	{ userId: string, token: string }	Vérifiera les informations d'identification de l'utilisateur, renvoie l'_id de l'utilisateur depuis la base de données et un token web JSON signé.
GET	/api/beers	-	Array of beers	Renvoie un tableau de toutes les bières présentes en base de données.
GET	/api/beers/:id	-	Single beer	Renvoie la bière ayant l'_id demandé
POST	/api/beers	{ beer: string, image: file }	{message: string}	Capture et enregistre l'image, analyse la bière transformée en chaîne de caractères et l'enregistre dans la base de données en définissant correctement son champ "imageUrl". Initialise les likes et dislikes de la bière à 0 et les "usersLiked" et "usersDisliked" avec des tableaux vides.

DOSSIER PROFESSIONNEL (DP)

	Point d'accès	Request body	Type de réponse attendue	Fonction
PUT	/api/beers/:id	Beer en JSON ou { beer: string, image: file }	{message: string}	Met à jour la bière avec l'_id fourni. Si une image est téléchargée, elle est capturée et l'imageUrl de la bière est mise à jour. Si aucun fichier n'est fourni, les informations sur la bière se trouvent directement dans le corps de la requête (req.body.name, req.body.degree, etc.).
DELETE	/api/beers/:id	-	{message: string}	Supprime la bière avec l'_id fourni.
POST	/api/beers/:id/like	{ userId: string, like: Number }	{message: string}	Définit le statut « Like » pour l'userId fourni. Si like = 1, l'utilisateur aime (= like) la bière. Si like = 0, l'utilisateur annule son like ou son dislike. Si like = -1, l'utilisateur n'aime pas (= dislike) la bière. L'ID de l'utilisateur doit être ajouté ou retiré du tableau approprié. Cela permet de garder une trace de leurs préférences et les empêche de liker ou de ne pas disliker la même bière plusieurs fois : un utilisateur ne peut avoir qu'une seule valeur pour chaque bière. Le nombre total de « Like » et de « Dislike » est mis à jour à chaque nouvelle notation.

Exemple de connexion a mongo db à distance mais il est aussi possible de l'utilisé en local

Dans app.js

```
mongoose.connect('mongodb+srv://mongoose:Sxxxxxxr@cluster0.0n18yb6.mongodb.net/db_test?retryWrites=true&w=majority&appName=Cluster0')
```

Voici l'exemple pour insérer un utilisateur

```
const User = require('../models/User');
const bcrypt = require('bcrypt');
const jwt = require('jsonwebtoken');

exports.signup = (req, res, next) => {
  bcrypt.hash(req.body.password, 10)
    .then(hash => {
      const user = new User({
        email: req.body.email,
        password: hash
      });
      user.save()
        .then(() => res.status(201).json({ message: 'Utilisateur créé avec succès !' }))
        .catch(error => res.status(400).json({ error }));
    })
    .catch(error => res.status(500).json({ error }));
};
```

```
exports.login = (req, res, next) => {
  User.findOne({ email: req.body.email })
    .then(user => {
      if(!user) {
        return res.status(401).json({ error: "L'utilisateur n'existe pas ! "});
      }
      bcrypt.compare(req.body.password, user.password)
        .then(valid => {
          if(!valid) {
            return res.status(401).json({ error: 'Mot de passe incorrect !' });
          }
          res.status(200).json({
            userId: user._id,
            token: jwt.sign(
              { userId: user._id },
              'RANDOM_TOKEN_SECRET',
              { expiresIn: '24h' }
            )
          });
        })
        .catch(error => res.status(500).json({ error }));
    })
    .catch(error => res.status(500).json({ error }));
};
```

2. Précisez les moyens utilisés :

Vscode
Xampp
Symfony
Doctrine

Mongodb
Postman
Express
Node JS
bcrypt
Jsonwebtoken
angular

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

seul

4. Contexte

Nom de l'entreprise, organisme ou association ► *AFPA*

Chantier, atelier, service ► Formation

Période d'exercice ► Du : *01/01/2024* au : *07/01/2024*

5. Informations complémentaires (facultatif)

Activité-type 2

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

Exemple n° 1 ► Développer des composants métier coté serveur

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Dans le cadre du développement de mon projet personnel je présente le PatientController de mon application.

Ce composant a été développé en utilisant le framework Symfony qui grâce à sa robustesse j'ai pu sécurisé et simplifier la mise à jours des données des patients, tout en m'assurant que seuls les utilisateurs connectés puissent accéder à leur profil.

```
<?php

namespace App\Controller;

use App\Entity\Patient;
use App\Form\PatientType;

use Doctrine\ORM\EntityManagerInterface;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Bundle\SecurityBundle\Security;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Attribute\Route;

class PatientController extends AbstractController
{
    private $security;

    public function __construct(Security $security)
    {
        $this->security = $security;
    }
    #[Route('/patient/profil', name: 'app_patient_profil')]

    public function update(Request $request, EntityManagerInterface $entityManager,
Security $security): Response
    {

```

DOSSIER PROFESSIONNEL (DP)

```
$user = $this->security->getUser();
$userId = $user->getId();

$edit = $entityManager->getRepository(Patient::class)->findOneBy(['user' =>
$userId]);

if (!$edit) {
    throw $this->createNotFoundException('Personne trouvée.');
```

```
}

$form = $this->createForm(PatientType::class, $edit);
$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {
    $entityManager->flush();

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

return $this->render('patient/index.html.twig', [
    'form' => $form->createView(),
]);
}
```

J'ai donc utiliser le composant de symfony Security pour authentifier les utilisateurs et leurs permettre d'accéder à leur propre information .

J'ai utilisé l'orm de Doctrine pour me permettre de manipuler l'entity Patient de manière efficace et de pouvoir interagir avec la base de données

La mise a jours ce fait donc par un formulaire symfony (PatientType) ce qui permet de facilité la validation des données coté serveur qui sont saisie par l'utilisateur et qui est traiter par (\$form->isSubmitted() && \$form->isValid())

Ici nous voyons l'annotation de route qui définit le chemin d'accès à l'action du Controller

```
#[Route('/patient/profil', name: 'app_patient_profil')]
```

J'ai aussi utilisé le service sécurité pour gérer les informations d'authentification de l'utilisateur.

2. Précisez les moyens utilisés :

Vscode
Xampp
Symfony

DOSSIER PROFESSIONNEL (DP)

doctrine

3. Avec qui avez-vous travaillé ?

seul

4. Contexte

Nom de l'entreprise, organisme ou association ► AFPA

Chantier, atelier, service ► Formation

Période d'exercice ► Du : 29/02/2024 au : 10/03/2024

5. Informations complémentaires (facultatif)

Activité-type 2

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

Exemple n° 1 ► Documenter le déploiement d'une application dynamique web ou web mobile

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Pour le site <http://davidsteel.fr>, après avoir refait complètement le site internet j'ai donc procéder au déploiement, dans un premier temps après l'avoir tester en local ,je l'ai mis sur mon serveur ionos pour tester tout le fonctionnement et m'assurer définitivement que je pouvais le mettre en production sur l'hébergeur de celui ou son précédant site internet était actif , malheureusement il a souhaité que je conserve sur son hébergeur son ancienne version du site ,donc j'ai intégré mes fichiers dans son hébergeur et j'ai modifié uniquement index.html.

Pour ce faire j'ai utilisé Filezilla pour faire le transfert de donnée, j'ai dû contacter son hébergeur pour avoir les informations d'accès ftp.

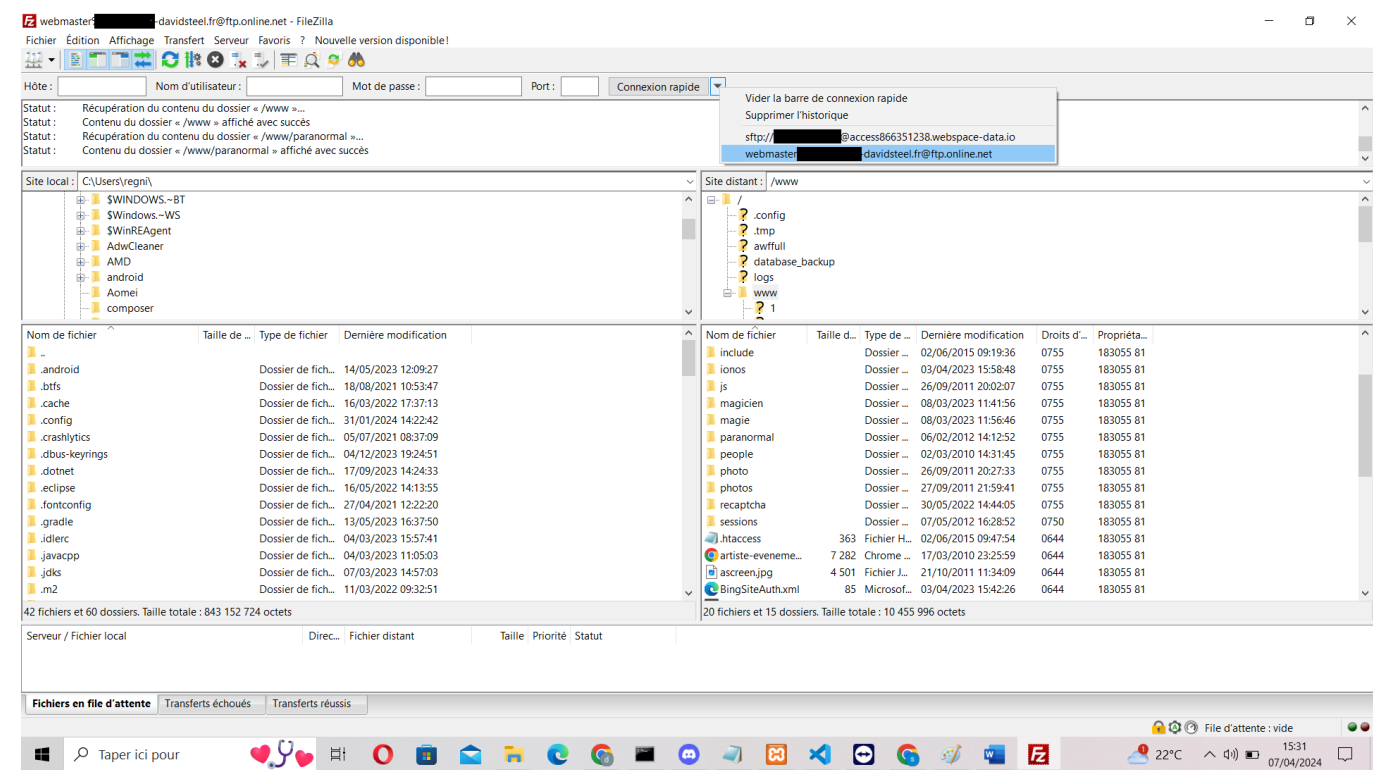
Pour cet exemple j'utilise ma connexion rapide que j'ai configuré avec Filezilla(car il est toujours enregistré pour certaine maintenance) sinon il faut le nom d hôte, le nom utilisateur, le mot de passe ainsi que le port du serveur toutes ses informations sont à retrouver chez l'hébergeur.

Dans le premier rectangle nous avons les informations si l'on est bien connecté au serveur ou s'il y a une erreur puis en dessous, il y a une partie locale à gauche (pc) et une partie serveur à droite.

Suivant la configuration du serveur l'index doit être mis à la racine puis ensuite on transfère les autres dossiers en fonction de son squelette de développement.

Tout en bas de Filezilla lorsque nous effectuons le transfère nous voyons le transfert en cours avec réussit ou échec de noté.

DOSSIER PROFESSIONNEL (DP)



2. Précisez les moyens utilisés :

Filezilla

3. Avec qui avez-vous travaillé ?

seul

4. Contexte

Nom de l'entreprise, organisme ou association ▶

David Steel

Chantier, atelier, service ▶

Refaire le site

Période d'exercice ▶

Du : 01/03/2021

au : 05/05/2024

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Bac pro commerce	Education national	30/06/2000
Développement Android N 2017	Udemy	18/03/2023

DOSSIER PROFESSIONNEL (DP)

The screenshot shows a web browser displaying a quiz page on the OpenClassrooms platform. The page title is "Utiliser une base de données - OpenClassrooms". The URL in the address bar is "https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/exercices/4206". The page header includes the OpenClassrooms logo and navigation links: "Formations", "Alternance", "Financements", and "Pour les entreprises". The main content area has a purple header with the course title "Concevez votre site web avec PHP et MySQL" and a sub-header "Accueil > Cours > Concevez votre site web avec PHP et MySQL > Quiz : Utiliser une base de données". Below this, there is a progress bar and a message "Bravo ! Vous avez réussi cet exercice !". The section "Compétences évaluées" lists "Écrire des fonctions SQL basiques". The "Description" section states: "Pour répondre aux questions de ce quiz, vous devez créer une base de données et [importer](#) les données." A sidebar on the right titled "Stocker des informations dans une base de données" lists five tasks: 1. Travaillez avec une base de données, 2. Mettez en place une base de données avec phpMyAdmin, 3. Accédez aux données en PHP avec PDO, 4. Ajoutez, modifiez et supprimez des recettes, 5. Ajoutez des données. The bottom of the page shows a Windows taskbar with various application icons and a system tray displaying "16°C Très ensoleillé" and the date "19/03/2023".

Utiliser une base de données - OpenClassrooms

https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/exercices/4206

OPENCCLASSROOMS Formations Alternance Financements Pour les entreprises

Accueil > Cours > Concevez votre site web avec PHP et MySQL > Quiz : Utiliser une base de données

Concevez votre site web avec PHP et MySQL

8 heures Moyenne

Licence CC BY-SA

Mis à jour le 26/01/2023

Utiliser une base de données

Bravo ! Vous avez réussi cet exercice !

Compétences évaluées

- ✓ Écrire des fonctions SQL basiques

Description

Pour répondre aux questions de ce quiz, vous devez créer une base de données et [importer](#) les données.

Stocker des informations dans une base de données

- ✓ 1. Travaillez avec une base de données
- ✓ 2. Mettez en place une base de données avec phpMyAdmin
- ✓ 3. Accédez aux données en PHP avec PDO
- ✓ 4. Ajoutez, modifiez et supprimez des recettes
- ✓ 5. Ajoutez des données

Taper ici pour

16°C Très ensoleillé 15:29 19/03/2023

DOSSIER PROFESSIONNEL (DP)

The screenshot shows a web browser window displaying the OpenClassrooms website. The page title is "Utiliser des fonctionnalités avancées de HTML et CSS - OpenClassrooms". The URL is "https://openclassrooms.com/fr/courses/1603881-creez-votre-site-web-avec-html5-et-css3/exercices/4743". The page content includes a progress bar, a message "Bravo ! Vous avez réussi cet exercice !", and a list of competencies evaluated: "Utiliser des fonctionnalités avancées de HTML et CSS". A sidebar on the right shows a checklist of tasks completed, including "Ajoutez des tableaux", "Créez des formulaires", "Finalisez un formulaire et ajoutez un bouton d'envoi", "Utilisez le responsive design avec les Media Queries", and "Allez plus loin". The bottom of the browser window shows the Windows taskbar with various application icons and the system clock indicating 14:51 on 19/03/2023.

The screenshot shows a Udemy certificate of completion. The certificate is for the course "Android N Développement 2017 - Le Cours Complet" by Benjamin Lize. The student's name is Regnier Sylvain. The date of completion is 18 mars 2023, and the total length of the course is 13.5 heures au total. The certificate includes the Udemy logo, the course title, the instructor's name, and the student's name. It also displays the certificate number, the URL of the certificate, and the reference number. The certificate is framed by a black border.

DOSSIER PROFESSIONNEL (DP)



**RÉPUBLIQUE
FRANÇAISE**

*Liberté
Égalité
Fraternité*

diplome.gouv.fr

Paris, le 7 avril 2024

ATTESTATION DE RÉUSSITE

BACCALAUREAT PROFESSIONNEL

M. REGNIER SYLVAIN

Né le 08/05/1980 à MONTREUIL SOUS BOIS (93)

A OBTENU LE RÉSULTAT ADMIS MENTION ASSEZ BIEN

Spécialité COMMERCE

Session 2000, ACADEMIE DE CRETEIL

Clé de contrôle : i4J3cVf3

Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] Sylvain Regnier ,

déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis

l'auteur(e) des réalisations jointes.

DOSSIER PROFESSIONNEL (DP)

Fait à Marseille le 07/04/2024

pour faire valoir ce que de droit.

Signature : 

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé
Documentation d'utilisation pour une application Android gérant les rendez-vous des taxis, ceux-ci étaient pris sur le site internet que j'avais fait(en passant par firestore de google pour gérer la partie web et l'application mobile) pour un ami ainsi que l'application Android

ANNEXES

(Si le RC le prévoit)

Exemple complet de requete sql

```
public function get_inscription_valider(array $data)
{

    try {

        $nom = isset($_POST['nom']) ? $_POST['nom'] : "";
        $nom = validate_formulaire($nom);

        $prenom = isset($_POST['prenom']) ? $_POST['prenom'] : "";
        $prenom = validate_formulaire($prenom);

        $adresse = isset($_POST['adresse']) ? $_POST['adresse'] : "";
        $adresse = validate_formulaire($adresse);

        $complementAdresse = isset($_POST['complementAdresse']) ?
$_POST['complementAdresse'] : "";

        $complementAdresse= validate_formulaire($complementAdresse);

        $codePostal = isset($_POST['codePostal']) ? $_POST['codePostal'] : "";
        $codePostal = validate_formulaire($codePostal);

        $ville = isset($_POST['ville']) ? $_POST['ville'] : "";
        $ville = validate_formulaire($ville);

        $telephone = isset($_POST['telephone']) ? $_POST['telephone'] : "";
        $telephone = validate_formulaire($telephone);

        $password = isset($_POST['password']) ? $_POST['password'] : "";
```

DOSSIER PROFESSIONNEL (DP)

```
$password = mdp($password);

$email = isset($_POST['email']) ? $_POST['email'] : "";
$email = validate_formulaire($email);

$date = date("Y-m-d");
$droit = "Patient";

$requete = $this->bd->prepare('INSERT INTO patient (id_patient,
nom_patient, prenom_patient, adresse_patient, complement_adresse_patient,
code_postal_patient, ville_patient, telephone_patient, mdp_patient
,email_patient, date_creation_patient, date_fin_patient,droit_utilisateur_patient)
VALUES (NULL, :nom, :prenom, :adresse, :complementAdresse,
:codePostal, :ville, :telephone,
:pass, :email, :datejour, NULL,:droit)');

$requete->execute([

    ':nom' => $nom,

    ':prenom' => $prenom,

    ':adresse' => $adresse,

    ':complementAdresse' => $complementAdresse,

    ':codePostal' => $codePostal,

    ':ville' => $ville,

    ':telephone' => $telephone,

    ':pass' => $password,

    ':email' => $email,

    ':datejour' => $date,

    ':droit' => $droit

]);
```

```
return [  
    'email' => $email,  
    'nom' => $nom,  
    'prenom' => $prenom,  
    'adresse' => $adresse,  
    'ville' => $ville,  
    'telephone' => $telephone,  
    'codePostal' => $codePostal,  
    'droit' => $droit  
];  
} catch (PDOException $e) {  
    die('Erreur [' . $e->getCode() . '] : ' . $e->getMessage() . '</p>');  
}  
}
```