

# DOSSIER PROFESSIONNEL (DP)

## 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

<b>Développer la partie front-end d'une application web ou web mobile</b>	<b>p.</b>	<b>5</b>
▶ Création d'un memory game (JS, HTML5 & CSS3) .....	p.	5 - 6
▶ Création d'un site web pour l'agence de construction et de rénovation de maisons : NorImmo-master (JS, DOM, HTML5/CSS3).....	p.	6 - 7
▶ Création d'un site statique de matériel informatique en responsive design : Terre de Geek (HTML5/CSS3) .....	p.	7 - 8

<b>Développer la partie back-end d'une application web ou web mobile</b>	<b>p.</b>	<b>9</b>
▶ Manage Your Project (HTML5, CSS3, JavaScript, Bootstrap, Symfony, PHP, SQL, DQL)	p.	9-10
▶ Projet fil rouge, Bank-root et banque-symfony (HTML5, CSS3, JavaScript, DOM, Bootstrap, Symfony, PHP, SQL, DQL) .....	p.	11-12-13
▶ Création d'une extension dynamique pour le site vitrine Event Tex (HTML5, CSS3, SVG, JavaScript, Bootstrap, Symfony, PHP, SQL, DQL).....	p.	13-14

<b>Titres, diplômes, CQP, attestations de formation</b> <i>(facultatif)</i>	<b>p.</b>	
<b>Déclaration sur l'honneur</b>	<b>p.</b>	
<b>Documents illustrant la pratique professionnelle</b> <i>(facultatif)</i>	<b>p.</b>	
<b>Annexes</b> <i>(Si le RC le prévoit)</i>	<b>p.</b>	

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 1 Développer la partie front-end d'une application web ou web mobile

Exemple n°1 ► Création d'un memory game en JavaScript, HTML5 & CSS3

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

Sur ce projet, je devais réaliser une application web dynamique avec les technologies HTML5/CSS3 et JavaScript. J'ai commencé ce projet par la création de mon Repository sur GitHub, et de la création d'un Kanban pour gérer les étapes de développement de ce jeu. Afin d'avoir une vision plus claire de l'ensemble du projet, j'ai effectué les Wireframes de la page de jeu. Puis pour faciliter l'écriture et le développement de mon code, j'ai créé l'arborescence utilisateur, cela m'a permis de suivre une ligne directrice dans l'élaboration de ce projet (la construction de mon code). J'ai ensuite édité la page HTML avec les outils Bootstrap, puis affiné le design en CSS. Grâce à l'arborescence créée en amont, j'ai pu développer mon code en JavaScript, suivre les étapes pas à pas. Pour finir, j'ai vérifié mon code sur W3C pour être sûre qu'aucune erreur ne m'ait échappée. J'ai aussi, tout au long du processus de développement, fait des sauvegardes via l'outil de gestion de source Git et envoyé celles-ci en ligne sur GitHub.

### 2. Précisez les moyens utilisés :

Pour réaliser l'ensemble de ce projet, je me suis servi d'un ordinateur. Une connexion internet a été indispensable pour pouvoir résoudre certains problèmes algorithmiques sur JavaScript, j'ai ainsi pu me servir de la documentation, essentiellement sur MDN et W3School, télécharger mes images et icônes via Flaticon et Pixabay. J'ai écrit et développé mon code sur l'IDE Visual Studio Code, fait mes sauvegardes via GitHub, réalisé mes Wireframes et mon arborescence utilisateur via Pencil.

### 3. Avec qui avez-vous travaillé ?

Bien que j'aie réalisé ce projet en solo, j'ai pu m'appuyer sur mes camarades de formation.

### 4. Contexte

Nom de l'entreprise, organisme ou association ► AFPA

Chantier, atelier, service ► Plateau technique DWWM

Période d'exercice ► Du : 31/08/2020 au : 06/09/2020

# DOSSIER PROFESSIONNEL (DP)

## 5. Informations complémentaires (facultatif)

Spécifications fonctionnelles :

- Affiche 12 cartes face cachée sur l'écran
- Quand le joueur clique sur une carte celle-ci se retourne,
- Quand le joueur clique sur une deuxième carte si elle est identique à la précédente les deux cartes restent face visible autrement les cartes sont à nouveau masquées.
- Quand toutes les paires sont trouvées et donc que toutes les cartes sont face visible le jeu est terminé
- Ajout d'un timer et d'un compteur d'essais à l'application qui donne un temps maximal et un nombre de coups impartis pour trouver les paires

## Activité-type 1 Développer la partie front-end d'une application web ou web mobile

**Exemple n°1 ► Création d'un site web pour l'agence de construction et de rénovation de maisons : NorImmo-master**

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

Dans ce projet, il nous a été demandé de travailler en équipe agile, nous devons réaliser un site d'agence immobilière, responsive sur tous les supports. Dans un premier temps nous avons dû nous répartir les rôles pour la gestion de ce projet. Mon camarade Baptiste Collignon a été nommé « chef de projet » et j'avais le rôle de « développeur ». Nous avons donc commencé par établir nos plans et la gestion de notre projet via le service web et d'hébergement GitHub (Kanban, assignation des tâches). Le chef de projet devait créer les formulaires et gérer la sécurité du site, quant à moi, mon travail a été de produire l'arborescence fonctionnelle de l'application, de créer un carrousel pour habiller la page d'accueil, ainsi qu'une page de FAQ. J'ai donc commencé par l'étape de la création de la page FAQ, en intégrant Bootstrap, mon fichier CSS ainsi que mon fichier Javascript. J'ai géré le HTML de ma page FAQ avec Bootstrap. La couleur et l'animation via mon fichier CSS. Et enfin l'écoute d'événement avec le DOM en JavaScript. Pour ce qui est du carrousel, j'ai travaillé de manière similaire à la création de la page FAQ. Pour la rédaction du JavaScript, surtout pour l'algorithme, je me suis appuyée sur le site web de questions/réponses anglais « Stack Overflow ».

### 2. Précisez les moyens utilisés :

Besoin fonctionnel : ordinateur, internet, GitHub, Visual Studio Code, Git Bash, Xampp

Organisation : Kanban

Arborescence utilisateur : Drawio

Faq & Carrousel : HTML5, CSS3, Bootstrap, JavaScript

# DOSSIER PROFESSIONNEL (DP)

Documentation : MDN, Stack Overflow  
Vérification : validator.w3.org

## 3. Avec qui avez-vous travaillé ?

Ce projet a été réalisé en collaboration avec Baptiste Collignon.

## 4. Contexte

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

Chantier, atelier, service ► Plateau technique DWWM

Période d'exercice ► Du : *17/08/2020* au : *23/08/2020*

## 5. Informations complémentaires (facultatif)

### Activité-type 1 Développer la partie front-end d'une application web ou web mobile

*Exemple n°1 ► Réalisation d'un site statique spécialisé dans le matériel informatique en Responsive design : Terre de Geek*

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

Cette réalisation fût la création de mon premier site web en entier. J'ai tout d'abord commencé par réaliser mes maquettes, comme le UseCase et les Wireframe. Puis j'ai défini une charte graphique, afin d'avoir une cohérence de structure et d'affichage de mes pages. Lors de la conception, j'ai suivi le concept « Mobile First », La structure du site est composée d'une page d'accueil affichant un header, une barre de navigation, un affichage des catégories des produits sous forme de cartes et un footer contenant les crédits (compte du client, mentions légales, informations sur la société, vidéo de présentation de l'entreprise), des liens réseaux sociaux. Le site contient aussi d'autres pages, « Qui sommes-nous ? » qui présente la société, trois pages dédiés (téléphonie, informatique, produits dérivés), une page mentions légales, un formulaire de contact. J'ai créé et intégré une animation en CSS sur mon titre, qui à l'écoute de l'événement du clic souris change de couleur. J'ai géré mon responsive design via les médias queries en CSS et constamment vérifié l'affichage de la structure de mes pages sur toutes les tailles d'écran. Et j'ai aussi fait attention au design, qu'il soit simple et moderne.

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

Besoin fonctionnel : ordinateur, internet, GitHub, Visual Studio Code, Git Bash, Xampp

Organisation : Kanban

Maquettes et charte graphique : Drawio, Pencil, Colors, Google Fonts

Images, icônes : Pixabay, Vector Icons

Documentation : Openclassrooms, FreeCodeCamp, CSS tricks

Vérification : validator.w3.org

## 3. Avec qui avez-vous travaillé ?

J'ai réalisé ce travail seule, mais j'ai pu m'appuyer sur mes camarades de promotions et sur le formateur pour la validation de mes maquettes.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► Cliquez ici pour taper du texte.

Chantier, atelier, service ►

Période d'exercice ► Du : 10/07/2020 au : 24/07/2020

## 5. Informations complémentaires (facultatif)



## Activité-type 2 Développer la partie back-end d'une application web ou web mobile

### Exemple n° 1 ► Manage Your Project

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

Pour cette évaluation finale, je devais réaliser une application métier de gestion de projet pour un client travaillant dans le bâtiment. En première étape, j'ai préparé mon environnement de travail sur GitHub, mon Kanban et une branche dev en supplément de ma branche main, afin d'avoir une sécurité supplémentaire si je devais rencontrer des problèmes sur mon code. En deuxième étape, j'ai réalisé tous mes plans (wireframes mobile/tablette/PC, l'arborescence utilisateur, le UseCase, le schéma de base de données et le diagramme de classe). En troisième étape, j'ai réalisé la base de données sous le Framework Symfony 5, créé mes Entities User, Project et Task. Puis j'ai créé les classes Controller pour la gestion des Controller sur mes entités Project et Task via le CRUD pour gagner du temps sur le développement de ce projet. En quatrième étape, j'ai développé toutes les fonctionnalités de cette application sous le Framework Symfony 5 (Inscription et connexion. Afficher, modifier, supprimer les projets et les tâches liées à ce projet), coder et améliorer la logique des différentes méthodes des classes Controller pour le CRUD. En cinquième étape, j'ai géré la sécurité sur mon entité User ainsi que dans les classes. En sixième étape, pour finir, j'ai réalisé un layer en javascript et ajouter un peu de design en CSS.

#### 2. Précisez les moyens utilisés :

Besoin fonctionnel : ordinateur, internet, GitHub, Visual Studio Code, Git Bash, Xampp  
Maquettage : drawio  
Gestion : GitHub, PowerShell, PC, Internet, phpmyadmin  
Organisation : Kanban  
Front-end : HTML5, CSS3, JavaScript, Bootstrap  
Back-end : Symfony, PHP, CRUD, SQL  
Documentation : MDN, Symfony doc, Stack Overflow  
Vérification : validator.w3.org

#### 3. Avec qui avez-vous travaillé ?

Bien qu'ayant travaillée en seule, j'ai eu besoin d'avoir la validation sur la construction de mon plan d'étapes du projet de la part de Thomas Gossart, notre formateur, ayant encore quelques doutes en tant qu'étudiante en développement web. Je voulais être sûre de n'omettre aucune étape.

# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

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

Chantier, atelier, service ► Hébergement AFPA (FOAD)

Période d'exercice ► Du : *23/11/2020* au : *29/11/2020*

## 5. Informations complémentaires (facultatif)

Spécifications fonctionnelles :

- Se connecter avec son compte personnel ou de s'inscrire s'il n'a pas de compte
- Voir tous les projets de cet utilisateur sur une page
- Créer un nouveau projet via un formulaire
- Voir le détail d'un projet, c'est-à-dire le projet avec ses tâches quand il clique dessus
- Créer des tâches liées à un projet particulier via un formulaire
- Supprimer les projets et les tâches comme il le désire
- Modifier les projets et les tâches comme il le désire
- Indiquer si une tâche est terminée
- Distinguer visuellement les tâches finies des tâches en cours
- Utiliser l'application sur les chantiers via une tablette ou un smartphone L'application est également visuellement enrichie afin d'offrir à l'utilisateur une expérience la plus intuitive possible.
- Un utilisateur se connecte avec un email et un mot de passe
- Un projet est composé à minima d'un nom, d'une description, d'une date de création, d'un deadline et d'un statut
- Une tâche est composée à minima d'un nom, d'une description, d'une date de création, d'un deadline et d'un statut
- Un schéma de type UseCase de l'application reprenant les fonctionnalités
- Un schéma de base de données listant les tables, leur contenu, leurs relations et les cardinalités
- Un diagramme de classes qui liste les entités de votre application, leur contenu et leurs relations
- Les wireframes de l'application qui spécifient à minima le template et la page principale pour les versions mobiles, tablettes et PC
- Le projet est géré à l'aide d'un outil de type Kanban

## Activité-type 2 Développer la partie back-end d'une application web ou web mobile

Exemple n° 1 ► *Bank-Root / banque-Symfony (projet fil rouge)*

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

Dans ce projet fil rouge, je devais créer une application de gestion de comptes bancaires pour un public d'un âge avancé, utilisant peu internet. Cette application a évolué au fur et à mesure de mon apprentissage et de ma découverte de nouvelles technologies.

1<sup>ère</sup> partie (du 24/08 au 30/08) HTML5/CSS3, JavaScript, PHP - Dans cette partie, afin de gagner en maintenabilité, mon header, ma nav et mon footer ne sont pas dupliqués sur chaque page, mais chargé là où j'en ai besoin. Toujours dans cette optique de ne plus dupliquer le code, j'appelle toutes mes données pour l'instant enregistré en dur dans des fichiers Json ou PHP afin d'effectuer des tests unitaires pour vérifier mes fonctionnalités. J'utilise la transmission de données par l'URL, pour spécifier l'affichage d'une seule donnée d'une table de ma BDD. Création d'une page d'accueil permettant la redirection vers un formulaire de connexion ou un formulaire d'inscription. Une page d'affichage de l'ensemble des comptes de l'utilisateur. Une page d'affichage d'un seul compte avec tous les détails le concernant. Ajout de la sécurité sur les formulaires : vérifications grâce à des conditions des informations requises, comme le type de compte qu'il est possible d'ouvrir, qu'une somme minimum est bien enregistrée à la création d'un compte, que le mot de passe contient bien le nombre de caractères nécessaires).

2<sup>ème</sup> partie (du 21/09 au 27/09) SQL - Dans cette partie, j'ai créé mon schéma de base de données. Il m'a fallu prendre un temps de réflexion pour n'omettre aucune information qui pourrait m'être utile lors de la création de mes tables et de leurs champs en base de données (Customer, Account, Operation) et les relations entre elles. Par sécurité, l'application est désormais réservée aux utilisateurs connectés via leur email et un mot de passe. Création des requêtes SQL et intégrations de fixtures pour désormais gérer mes tests depuis la base de données.

3<sup>ème</sup> partie (du 28/09 au 04/10 ou 11/10) CRUD - Dans cette partie, je réalise les opérations du CRUD sur mes entités Customer, Account et Opération, ce qui permet à l'utilisateur de créer, voir, modifier et supprimer son compte client, ses comptes bancaires. J'ai rajouté des messages d'erreurs flash sur mes formulaires afin d'améliorer l'expérience de l'utilisateur lors de la création de son compte, la création de ses comptes bancaires et sa connexion.

4<sup>ème</sup> partie (du 19/10 au 25/10) MVC, POO - Dans cette partie, j'ai donc passé mon application procédurale en application objet organisée avec le pattern MVC (1 model « 1 entity », 1 view, 1 controller). Avant de passer mon application en orienté objet, j'ai commencé par créer un diagramme de classe. J'ai aussi ajouté de la sécurité en y intégrant un système de rôles et une hiérarchie pour les gérer. J'ai aussi optimisé mes requêtes SQL avec des jointures pour l'appel et l'affichage des données sur les tables stockées en BDD.

5<sup>ème</sup> partie (du 09/11 au 22/11) Symfony – Sur la dernière partie de ce projet, j'ai travaillé en équipe Agile avec Adrien Girin (étudiant DWWM). Nous avons donc réuni nos travaux réalisés chacun de notre côté depuis le début de ce projet. Dans un premier temps, nous nous sommes assignés des rôles pour l'évolution du projet, mon camarade a pris le rôle de développeur et j'ai donc pris le rôle de chef de projet. J'ai géré l'organisation des tâches via Kanban et les sauvegardes via GitHub. Bien entendu,

# DOSSIER PROFESSIONNEL (DP)

nous avons établi un planning et fait un point chaque matin et chaque soir sur l'avancée du projet, les problèmes rencontrés (surtout dû à la découverte de Symfony et au manque de maîtrise de la documentation officielle) et ainsi, avons pu résoudre un grand nombre d'entre eux en travaillant en collaboration sur leurs résolutions. La tâche principale était donc de migrer cette application sur le framework Symfony. Nous nous sommes partagé les tâches afin de pouvoir avoir une vue d'ensemble sur le framework Symfony (page statique et dynamique). Pour ce qui est des maquettes, Adrien Girin développeur a réalisé le UseCase et de mon côté j'ai créé le schéma de base de données. Puis j'ai créé la base de données ainsi que des fixtures pour que l'on puisse rapidement effectuer des test par la suite et réalisée les migrations en base de données. J'ai aussi intégré des requêtes jointes pour optimiser l'affichage des données. J'ai aussi géré le formulaire de connexion, la validité des données rentrées grâce au bundle Validator, sa sécurité et de l'ajout de messages informatifs pour l'utilisateur. J'ai aussi géré l'affichage et les fonctionnalités pour créer un compte, supprimer un compte bancaire, afficher tous les comptes de l'utilisateur grâce à l'ID et afficher une page détails d'un compte sur lequel mon camarade Adrien Girin à intégrer les opérations (transferts, retraits, virement). Il a aussi dû gérer le formulaire d'inscription et sa sécurité et le design de la page d'accueil présentant des liens vers la connexion et l'inscription.

## 2. Précisez les moyens utilisés :

Besoin fonctionnel : ordinateur, internet, GitHub, Visual Studio Code, Git Bash, Xampp  
Maquettes et charte graphique : Drawio, Pencil, Colors, Google Fonts  
Gestion : GitHub, PowerShell, PC, phpmyadmin  
Organisation : Kanban, Diagramme de Gantt  
Front-end : HTML5, CSS3, JavaScript, Bootstrap, SVG  
Back-end : Symfony, PHP, SQL, DQL  
Documentation : OpenClassrooms, w3schools, phpunit, Symfony doc, Stack Overflow

## 3. Avec qui avez-vous travaillé ?

En solo sauf sur la 5<sup>ème</sup> partie, où j'ai travaillé en équipe agile avec XXXXXXXX.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► AFPA  
Chantier, atelier, service ► Plateau technique DWWM ou Hébergement AFPA (FOAD)  
Période d'exercice ► Du : 24/08/2020 au : 22/11/2020

## 5. Informations complémentaires (facultatif)

### Activité-type 2 Développer la partie back-end d'une application web ou web mobile

Exemple n° 1 ► STAGE : Extension dynamique du site vitrine XXXXXX

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

En amont du projet, nous avons fait une réunion avec le Manager et le Webmaster afin de décider du fond de mon projet et des fonctionnalités à y intégrer.

Puis j'ai réalisé mon cahier des charges listant tous les spécifications fonctionnelles à minima, au fur et à mesure, des spécifications fonctionnelles optionnelles se sont ajoutés.

J'ai établi ma charte graphique en reprenant les couleurs du site statique originel.

J'ai ensuite réalisé toutes mes maquettes, l'arborescence utilisateur, qui m'a permis d'avoir un visuel sur le nombre de page à créer. Le schéma de base de données, m'a permis de bien établir les tables dont j'aurais besoin pour gérer les informations stockées en base de données, ainsi que les relation entre elles. Sachant que je devais aussi gérer au minimum deux rôles, celui des clients et celui des employés de l'Agence Event Tex. Et enfin j'ai établi un diagramme de Gantt afin d'avoir une ligne conductrice dans l'élaboration des fonctionnalités et le temps définit pour les réaliser. Et créer un point de sauvegarde via la plateforme GitHub, par sécurité pour pouvoir faire un back-up de mon projet en cas de problème.

Puis j'ai géré ma connexion avec sa sécurité (token, password haché), puis j'ai rajouté des champs en base de données afin que l'Agence Event Tex puisse accéder à toutes les informations requises afin de gérer au mieux les demandes des clients. Pour finir, j'ai retravaillé le design et l'affichage des formulaires avec le framework Bootstrap.

J'ai créé un template de base afin d'y intégrer le framework Bootstrap et pouvoir intégrer à mon projet des script JavaScript, CSS et des librairies supplémentaires.

Une page d'accueil générale, qui propose des boutons permettant la connexion et l'inscription.

J'ai créé un Template header, nav, footer pour l'affichage de la page d'accueil des admins et un second pour l'affichage de la page d'accueil des clients. J'ai intégré un design simple dès la connexion, réalisé via le framework Bootstrap, mais aussi réalisé des animations en CSS sur le titre de la page et sur le footer et une animation SVG dès la connexion d'un client. J'ai aussi intégré dans mon footer des boutons permettant de naviguer vers le site statique d'Event Tex, afin que les clients puissent y retrouver les informations générale sur l'Agence, ses prestations, les liens vers les réseaux sociaux, etc.

Pour que les clients puissent rapidement avertir l'équipe d'Event Tex d'un problème sur leur site, d'une amélioration à apporter, du contenu supplémentaire à ajouter, etc., un système de Tickets géré sur plusieurs page a été intégré au projet. Le client peut accéder à une page lui permettant de créer un ticket, une page listant tous les tickets qu'il a envoyé, et voir le détail d'un ticket sur une page dans

# DOSSIER PROFESSIONNEL (DP)

laquelle il a la possibilité d'ouvrir une Chatbox avec les employés de l'Agence. Les admins ont accès à une liste des tickets envoyés par tous les clients et les trier suivant divers critères, comme du plus récent au plus ancien, par client et par statuts grâce à des requêtes en SQL ou DQL écrites soit dans mes Repository, soit directement dans mes Controller. Ils peuvent aussi modifier via formulaire un ticket et les supprimer.

Pour ce qui est des pages Chatbox, la navigation sur celles-ci et fonctionnalités dédiées à chaque rôle sont construites de la même manière que pour celles des tickets. La seule différence est qu'une Chatbox peut être créée par un client sans que forcément celle-ci soit liée à un ticket.

En ce qui concerne les pages des comptes clients, le client n'a accès qu'à deux fonctionnalités, celle de pouvoir modifier ses informations personnelles et celle, par sécurité, gérée sur une autre table en base de données de pouvoir modifier son mot de passe. Les admins peuvent accéder rapidement à un compte client, grâce à une page dédiée qui liste tous les clients ayant créé un compte.

Ayant réalisé une hiérarchie dans mes rôles, les admins ont donc accès à toutes les fonctionnalités des clients, pouvant même interagir directement sur leurs compte afin de les accompagner au mieux quand les démarches leur semblent trop techniques.

Les pages FAQ, qui permettent d'éditer les demandes les plus récurrentes des clients, seul les admins peuvent éditer de façon dynamique sur ces pages. Ils ont aussi la possibilité de modifier celles déjà existante et de supprimer celles qui n'auraient plus d'intérêt informatif pour les clients.

J'ai par la suite amélioré mes rendus visuels en front-end avec des animations et des transitions en CSS et SVG.

## 2. Précisez les moyens utilisés :

Besoin fonctionnel : ordinateur, internet, GitHub, Power Shell, Git Bash, Xampp

Gestion : GitHub, PowerShell, phpmyadmin

Organisation : Kanban, Diagramme de Gantt

Maquettes et charte graphique : Drawio, Pencil, Colors, Google Fonts

Front-end : HTML5, CSS3, JavaScript, Bootstrap, SVG

Back-end : Symfony, PHP, SQL, DQL

Documentation : Symfony doc, Stack Overflow

## 3. Avec qui avez-vous travaillé ?

J'ai créé ce projet seule mais j'ai pu m'appuyer sur le Manager de l'entreprise Event Tex, qui est le seul dans l'Agence sachant travailler sous le framework Symfony.

# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

Nom de l'entreprise, organisme ou association ► Cliquez ici pour taper du texte.

Chantier, atelier, service ► Cliquez ici pour taper du texte.

Période d'exercice ► Du : 04/01/2021 au : 05/03/2021

## 5. Informations complémentaires (facultatif)

### Résumé des besoins fonctionnels

- ✓ L'utilisateur peut s'inscrire
- ✓ L'utilisateur doit recevoir un mail de vérification pour valider son adresse mail
- ✓ L'utilisateur peut se connecter
- ✓ L'utilisateur doit être connecté pour accéder aux fonctionnalités du site
- ✓ La page d'accueil du site permet affiche des liens de navigation vers le site principal d'Event Tex (flocage, site web, etc.)
- ✓ L'utilisateur peut se déconnecter
- ✓ La déconnexion se fait automatiquement si l'utilisateur ferme la page du site
- ✓ La déconnexion se fait automatiquement au bout d'un laps de temps si l'utilisateur n'effectue aucune action
- ✓ L'utilisateur peut accéder à la page de gestion de son compte une fois connecté
- ✓ L'utilisateur peut envoyer une requête pour réinitialiser son mot de passe
- ✓ L'utilisateur peut personnaliser sa fiche client
- ✓ L'utilisateur peut accéder à son site professionnel avec un lien depuis son espace de gestion
- ✓ Une fois connecté, l'utilisateur est directement redirigé vers sa page de gestion de compte
- ✓ Une fois connecté, l'admin est directement redirigé vers sa page de gestion de compte ou via un lien disponible uniquement à l'admin
- ✓ L'admin peut naviguer sur son espace de gestion de compte via un menu latéral
- ✓ Les pages de gestion de compte sont différentes suivant le rôle de la personne connecté
- ✓ L'utilisateur peut modifier ses informations personnelles
- ✓ L'utilisateur peut envoyer une requête de suppression de son espace de gestion (qui sera validé par l'admin)
- ✓ L'admin doit être connecté pour pouvoir accéder à son l'espace de gestion
- ✓ L'admin peut modifier un compte utilisateur
- ✓ L'admin peut supprimer un compte utilisateur
- ✓ L'admin a accès à tous les comptes utilisateur
- ✓ L'admin peut créer un compte utilisateur
- ✓ L'admin peut accéder à une page ou se trouvent en liste tous les comptes des utilisateurs
- L'admin peut trier tous les comptes suivant différents critères (nom, email, etc.)
- ✓ L'admin peut spécifier le type de contrat du client (rajouter colonne sur la table en base de données)
- ✓ L'utilisateur peut accéder à la page des tickets
- ✓ L'utilisateur peut éditer un ticket

## DOSSIER PROFESSIONNEL (DP)

- ✓ L'utilisateur peut envoyer un ticket
- L'utilisateur reçoit un mail à chaque étape de prise en charge du ticket
- ✓ L'utilisateur peut accéder à une page d'historique des tickets
- ✓ L'utilisateur peut voir le détail d'un ticket depuis la page d'historique des tickets
- L'utilisateur peut trier ses tickets suivant différents critères (date, statuts, etc.)
- L'admin peut trier ses tickets suivant différents critères (date, statuts, etc.)
- ✓ L'admin peut accéder à une page d'historique des tickets.
- ✓ L'admin peut supprimer un ticket
- L'admin voit les statistiques de ses gestions des tickets
- L'admin voit une barre de progression de la résolution du problème du client
- L'admin voit visuellement (code couleur) si un ticket est résolu ou non
- ✓ L'admin à un visuel sur les derniers tickets ouvert sur sa page de gestion de compte
- ✓ L'admin peut accéder à des fonctions pour automatiser le système de réponses aux tickets sur des thèmes récurrent.
- ✓ L'utilisateur peut envoyer un message dans la chatbox
- ✓ L'utilisateur peut répondre à un message dans la chatbox
- ✓ L'utilisateur peut voir l'ensemble des messages dans une page dédiée
- L'utilisateur reçoit un mail/sms quand une réponse lui est apporté dans la chatbox
- ✓ L'admin peut voir tous les messages de la chatbox
- ✓ L'admin peut écrire un message dans la chatbox
- ✓ L'admin peut répondre à un message dans la chatbox
- ✓ L'admin peut supprimer une conversation de la chatbox
- ✓ L'admin peut supprimer un message dans la chatbox
- Optimisation, la création & la vue des messages sont jointes sur une même page.
- Sécurité reCAPTCHA\*
- ✓ Sécurité failles XSS\*
- ✓ Sécurité injection SQL\*
- ✓ Arborescence fonctionnelle de l'utilisateur
- UseCase\*
- ✓ Schéma de base de données
- Diagramme de classe
- ✓ Wireframe\* PC page de connexion et d'inscription
- ✓ Wireframe PC page accueil gestion côté client
- ✓ Wireframe PC page accueil gestion côté admin
- Wireframe tablette
- Wireframe mobile
- ✓ Interface responsive design\*
- ✓ Cahier des charges
- ✓ Charte graphique
- ✓ Gestion des sauvegardes sur GitHub\*
- ✓ Diagramme de Gantt
- ✓ Les requêtes sont optimisées avec des jointures
- ✓ Gestion des rôles (ADMIN & USER)
- ✓ Traduction en français
- ✓ Nettoyer le code



- ✓ Optimiser le poids des images
- ✓ Présenter un design propre, élégant et moderne

## Fonctionnalités optionnelles ou futures :

- ✓ L'utilisateur peut voir la page où sont affichées ses factures
- ✓ L'utilisateur peut voir le détail d'une facture
- L'utilisateur peut télécharger une facture quand il est sur la page de voir le détail d'une facture
- L'utilisateur voit chaque étape du paiement d'une facture lorsque celle-ci est payée en plusieurs fois
- ✓ L'admin peut éditer une facture
- ✓ L'admin peut voir l'ensemble des factures
- ✓ L'admin peut modifier une facture
- ✓ L'admin peut supprimer une facture
- L'admin peut trier les factures suivant différents critères (client, payé ou impayé, etc.)
- ✓ L'utilisateur peut accéder à une page FAQ (aide pour la gestion de son compte et de l'utilisation des outils proposés)
- ✓ L'admin peut créer une carte FAQ
- ✓ L'admin peut modifier une carte FAQ
- ✓ L'admin peut supprimer une carte FAQ
- L'utilisateur peut accéder à une plateforme de gestion des tâches (kanban)
- L'utilisateur peut créer un projet
- L'utilisateur peut accéder à une page du projet avec les tâches dédiées à celui-ci
- L'utilisateur peut modifier le projet (nom + description)
- L'utilisateur peut éditer des tâches
- L'utilisateur peut modifier une tâche
- L'utilisateur peut supprimer une tâche
- L'utilisateur voit visuellement (code couleur) si une tâche est terminée ou non
- Si l'utilisateur supprime un projet, toutes les tâches associées à celui-ci sont automatiquement supprimées
- L'utilisateur a accès à une page pour y déposer des fichiers texte ou des images/photos
- L'utilisateur peut gérer la compression de ses fichiers, images/photos avant l'envoi (voir à automatiser le processus)
- L'utilisateur peut planifier ses RDV via un calendrier
- L'utilisateur reçoit un mail/sms pour rappel de RDV
- ✓ L'utilisateur peut effectuer des demandes de devis suivant les services proposés par Event Tex.

## DOSSIER PROFESSIONNEL (DP)

### Titres, diplômes, CQP, attestations de formation

*(facultatif)*

Intitulé	Autorité ou organisme	Date
DAEU	Centre Universitaire d'Évreux	2013 - 2014

### Déclaration sur l'honneur

---

Je soussigné(e) [prénom et nom] .....,  
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis  
l'auteur(e) des réalisations jointes.

Fait à Saint Étienne du Rouvray ..... le 05/03/2021 .....

pour faire valoir ce que de droit.

Signature :

## Documents illustrant la pratique professionnelle

*(facultatif)*

Intitulé
Cliquez ici pour taper du texte.

---

## DOSSIER PROFESSIONNEL (DP)

---

---

### ANNEXES

---

*(Si le RC le prévoit)*