

RAPPORT DE STAGE

Développeur WEB et WEB Mobile

2020 / 2021



GRETA Lorraine Nord
Jean-Michaël SALICE

Table des matières

1.	Liste des compétences	3
2.	Résumé du projet.....	4
3.	Cahier des Charges.....	5
3.1.	Analyse clientèle.....	5
3.2.	Description du besoin numérique	6
3.2.1.	Situation actuelle	6
3.2.2.	Situation future souhaitée	6
3.2.3.	Planning prévisionnel.....	8
3.3.	Annexes et options	9
3.3.1.	Maintenance	9
3.3.2.	Propriété intellectuelle.....	9
3.3.3.	Paiement	9
4.	Spécifications techniques	9
4.1.1.	Maquettage.....	9
4.1.2.	Frontend dynamique.....	9
4.1.3.	Créer une base de données	10
4.1.4.	Développer les composants d'accès aux données	10
4.1.5.	Back end et gestion de contenu	10
4.1.6.	Sécurité	11
4.1.7.	Adaptabilité	11
5.	Réalisation du projet.....	12
5.1.	Organisation en amont du stage	12
5.1.1.	Analyse comparative	12
5.1.2.	Réalisation du maquettage.....	14
5.2.	Organisation pendant le stage	17
5.2.1.	Récupération des médias.....	17
5.2.1.	Directions artistiques	18
5.2.1.	Réalisation du Front end.....	19
5.2.2.	Mise en place de la base de données.....	20
5.2.3.	Réalisation du Back-end	22
5.2.4.	Réalisation du Back-office	22
5.2.4.1.	Technologies.....	23
5.2.4.2.	Structure	24
5.2.4.3.	Fonctionnalités réalisées	25
6.	Création du site (Extrait de codes significatifs).....	29

6.1.	Extrait n°1 : Back Office – Manipulation des membres de l'équipe.....	29
6.1.1.	Formulaire	29
6.1.2.	Création.....	30
6.1.3.	Obtention	31
6.1.4.	Mise à jour	31
6.1.5.	Suppression.....	32
6.2.	Extrait n°2 : Site Back Office – téléversement d'une image dans la galerie via le panneau d'administration.....	33
6.2.1.	Formulaire	33
6.2.2.	Création.....	34
6.2.3.	Obtention	35
6.2.4.	Update.....	36
6.2.5.	Suppression.....	37
7.	Présentation du jeu d'essais sur la fonctionnalité la plus représentative.....	38
7.1.1.	Exemple n°1 : Page de connexion.....	38
7.1.1.1.	Test de tentative de connexion	38
7.1.2.	Exemple n°2 : Test du formulaire pour les actualités.....	41
7.1.1.	Exemple n°3 : Test sur les différents navigateurs.....	43
8.	Veille technologique, sécurité, web mobile.....	44
8.1.	Faille XSS (Cross-Site Scripting)	44
9.	Description d'une situation de travail ayant nécessité une recherche sur site en Anglais.....	46
10.	Extrait du site anglophone.....	47
11.	Conclusion	48
12.	Annexes.....	49
13.	Références	49
14.	Remerciements.....	50

1. Liste des compétences

Certificat de Compétences Professionnelle 1

- Compétence Professionnelle 1 : Maquetter une application
- Compétence Professionnelle 2 : Réaliser une interface utilisateur web statique et adaptable
- Compétence Professionnelle 3 : Développer une interface utilisateur web dynamique

Certificat de Compétence Professionnelle 2

- Compétence Professionnelle 5 : Créer une base de données
- Compétence Professionnelle 6 : Développer les composants d'accès aux données
- Compétence Professionnelle 7 : Développer la partie Backend d'une application web ou web mobile.
- Compétence Professionnelle 8 : Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce

2. Résumé du projet

Ce stage a été effectué dans le cadre de la formation Développeur WEB et Web Mobile (DWWM) du 26/10/2020 au 12/07/2021 sur une durée de trois semaines allant du 22 Février 2021 au 12 Mars 2021.

L'entreprise d'accueil « DERMOPHRAGME CORPORATION » est un salon de tatouage situé à Terville où collaborent quatre tatoueurs/perceurs. Ils ont accepté de me prendre en stage pour créer et mettre en ligne un site internet à leur image, leur permettant de promouvoir le travail réalisé par chacun pour établir des liens vers les réseaux sociaux et inciter les visiteurs intéressés à entrer en contact directement ou par téléphone. Le besoin initial du client reflète le besoin d'un site « vitrine », agrémenté d'un Back Office.

Le site a été entièrement développé en HTML5, CSS3, j'ai utilisé le framework Bootstrap pour le frontal. Je me suis orienté vers PHP pour la partie Back Office et MySQL pour la gestion de la base de données.

Pour l'affichage et la manipulation dynamique des photos, j'ai utilisé javascript accompagné de bibliothèque telles que Masonry et Lighbox2 de Lokesh Dhakar (lokeshdhakar.com/projects/lightbox2/).

Ce projet m'a permis de confirmer mes connaissances acquises au cours de la formation, de les améliorer par la recherche et de confirmer mon envie de pratiquer ce métier.

La partie suivante du rapport explicite les besoins du client à travers un cahier des charges.

3. Cahier des Charges

Rédacteur : Jean-Michaël SALICE, Développeur WEB, GRETA Lorraine Nord. Dénommé ci-après le stagiaire.

Client : DERMOGRAPHE CORPORATION SARL, dénommé ci-après le client.

Interlocuteurs : Davy CHARISSOU (Gérant de l'établissement et tatoueur), Jean-Michaël SALICE (Apprenti Développeur WEB).

3.1. Analyse clientèle

DERMOGRAPHE CORPORATION, est un salon de tatouage et piercing, vente de produit de soins, de bijoux et de boissons artisanales.

Siège social domicilié à :

37 Boucle du Ferronnier

57180 TERVILLE

Téléphone : +33 9 67 72 86 69

- Le salon comprend une grande salle d'environ 120m² dédié à l'accueil des clients et au tatouage, il comprend également une petite salle de piercing, une salle de stérilisation et un coin boutique.
- Le salon est composé de quatres artistes : Davy, Anton, Meddy et Remy.
- Le client est ouvert du Mardi au Samedi de 9h à 18h et de 9h à 17h en période de COVID-19.
- Le client possède un accès WIFI sécurisé partagé à la clientèle et membres de l'équipe ainsi qu'au stagiaire.
- Le processus de prise de rendez-vous s'effectue par téléphone ou en direct
- Les rendez-vous sont consignés dans un cahier dédié.
- La gestion et la comptabilité sont réalisé à l'aide d'un logiciel spécifique.

3.2. Description du besoin numérique

3.2.1. Situation actuelle

Le site internet est en attente de développement.

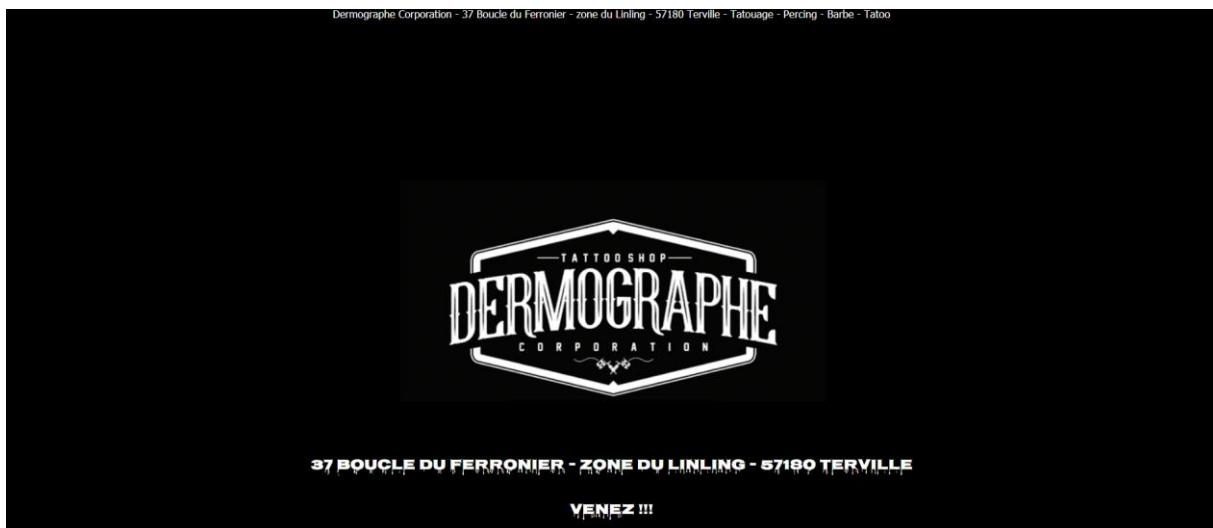


FIG1. CAPTURE D'ECRAN DU SITE EN LIGNE AVANT-PROJET

La communication du client sur internet passe principalement par l'utilisation des réseaux sociaux. Facebook et Instagram sont utilisés pour le salon et pour les tatoueurs individuellement pour partager des photos de leurs réalisations du moment ou des posts d'actualités (artistes tatoueurs invités, participation à des conventions de tatouage, présentation de nouveaux produits).

- <https://www.facebook.com/dermographecorporation>

3.2.2. Situation future souhaitée

Les besoins principaux du client étaient d'avoir un site vitrine afin de mettre en avant les artistes et leurs arts.

Le client souhaite un back-office lui permettant de gérer leurs informations personnelles, l'actualité du salon et les photos sur le site.

Attentes particulières du client :

- Un site internet vitrine dont l'objectif est d'améliorer sa visibilité sur Internet.
- Que le site soit à l'image du salon de tatouage.
- Que des liens de redirections vers les réseaux sociaux soient visibles et accessibles depuis la page principale.

Le client ne souhaite pas encore :

- Être contacté via le site via un formulaire de contact ou email.
- Permettre la prise de rdv en ligne.

Charte graphique :

Le client souhaite que la charte graphique soit en cohérence avec les couleurs du salon et la typographie du logo.

- Couleurs dominantes : gris, noir
- Couleurs secondaires : rouge, blanc

Logo : Le client possède déjà son propre logo.

Typographies :

- Titre principal (h1) : Le client souhaite la même police que celle du logo correspondant à la police de caractères *YOUNG HEART*.
- Titres secondaire (h2) : Open Sans Condensed

Paragraphes (p) : la typographie du titre principal de chaque page devra reprendre la typographie du Logo. Celle-ci correspond à la police de caractère, téléchargeable gratuitement sur internet.

Nom de domaine et hébergement :

Le client possède déjà son propre nom de domaine et un hébergeur :

- Nom de domaine : <http://dermographe-corporation.com/> ;
- Hébergeur : OVHCloud ;

Adresse mail :

- Le client possède deux adresses mails.
- Le client ne souhaite pas d'adresses supplémentaires.
- Le client ne souhaite pas être contacté par mail depuis la page du site.

Images / Vidéos :

- Le client fournira une partie des photos réalisées par un professionnel.
- Le tri et les retouches des photos sont laissés à la charge du stagiaire.
- Le stagiaire pourra réaliser des photos ou vidéos selon le besoin. L'accord de la clientèle sera préalablement demandé pour les questions de droits à l'image.

Mise en ligne :

La mise en ligne sera effectuée par le stagiaire une fois le site terminé.

Textes :

Les textes sont laissés à la charge du stagiaire. Ils seront validés par le client avant la fin du stage.

3.2.3. Planning prévisionnel

Le stage se déroule sur une période de 3 semaines allant du 22 Février 2021 au 12 Mars 2021.

Cette contrainte impose un découpage prévisionnel des tâches suivant :



Nota : La date de livraison n'est pas définie volontairement pour laisser libre le temps de conception du site par le stagiaire.

3.3. Annexes et options

3.3.1. Maintenance

La maintenance du site après mise en production est laissée à accord entre le client et la société EXOSITE.

3.3.2. Propriété intellectuelle

L'intégralité du code appartiendra à DERMOGRAPHE CORPORATION dès la mise en production du site.

Tout contenu visuel (identité graphique, photos, vidéos) appartiendra à DERMOGRAPHE CORPORATION dès la mise en production du site.

3.3.3. Paiement

Aucun paiement par la société DERMOGRAPHE CORPORATION n'est requis. Ce projet étant intégralement financé par la région Grand-EST.

De ce fait :

- Aucune contribution financière ne sera acceptée par le stagiaire.
- Aucune pénalité de retard de livraison et de mise en service ne peut être demandé par quelconques parties.

4. Spécifications techniques

4.1.1. Maquettage

Un maquettage a été réalisé pour les versions mobile et PC pour la partie Front-end et Back-office à l'aide du logiciel Adobe XD.

4.1.2. Frontend dynamique

Le projet a orienté mon choix de technologie sur les langages HTML5, CSS3, Javascript ainsi que PHP pour l'affichage des données provenant du Back-Office.

La bibliothèque Bootstrap 5 et la collection d'icônes Font-Awesome ont été utilisés.

Je me suis appuyé sur les différentes bibliothèques Javascript suivantes :

- Masonry pour la galerie d'images
- Lightbox2 pour le zoom des images

J'ai également utilisé l'API Google Maps pour afficher la localisation géographique du salon de tatouage sur une carte dynamique.

4.1.3. Créeer une base de données

La base de données a été créé à l'aide de l'interface de Phpmyadmin depuis WAMP Server.

4.1.4. Développer les composants d'accès aux données

Pour accéder et récupérer les données stockées en base de données, le langage PHP a été utilisé. Ce langage m'a permis d'afficher les différentes informations sur le site Front end et le site Back office d'administration.

4.1.5. Back end et gestion de contenu

Les données stockées dans la base de données seront modifiables par l'administrateur du site de la façon suivante :

- Création, affichage et suppression de photos en fonction de l'artiste concerné.
- Créer, modifier, afficher ou supprimer l'actualité du salon
- Créer un nouveau membre de l'équipe, insérer des informations, modifier les informations, supprimer un membre, afficher le membre de l'équipe dans le tableau de bord de la partie administrateur.

4.1.6. Sécurité

Pour la sécurité, j'ai orienté mon choix vers l'utilisation du framework PHP Data Object (PDO) afin de sécuriser les données envoyées par les formulaires et sécuriser les requêtes vers la base de données.

4.1.7. Adaptabilité

Pour obtenir un site adapté et multi-support j'ai utilisé la bibliothèque Bootstrap 4 et orienté mon choix vers un site One Page pour faciliter la navigation sur appareil mobile.

Le site one page à plusieurs avantages :

- Navigation fluide, rapide, intuitive par scrolling sur PC, tablette ou mobile.
- Format adapté au marketing. Ici le but est de vendre un service.
- Oblige à se focaliser sur la qualité plutôt que sur une masse d'information
- Rapidité de développement comparé à un site multi-page.

Les risques :

- Risque de temps de chargement plus long en fonction du contenu
- Risque de handicap pour le référencement lié au faible contenu.

5. Réalisation du projet

5.1. Organisation en amont du stage

Avant le début du stage, j'ai souhaité analyser l'environnement numérique du client en effectuant un benchmarking et réaliser un premier maquettage à proposer dès le début du stage afin de lancer au plus vite la réalisation du site.

5.1.1. Analyse comparative

Etant donné que c'est un premier projet professionnel pour un domaine d'activité précis, j'ai jugé préférable de réaliser un benchmarking avant le lancement du stage. Plusieurs intérêts à cela :

1. Evaluer la présence, la qualité graphique et les fonctionnalités web proposé par la concurrence.
2. Comprendre les habitudes de communication de cette activité et les informations diffusées.
3. Proposer au client un site de meilleure qualité comparé à la concurrence locale.

Données récoltées :

Nombre de salon de tatouage dans un rayon de 10km (client inclus) : 20

Nb de salon possédant un lien internet : 13

	Nombre
Page Facebook	7
Sites développés	5
Sites autres (Wix, ...)	1

Observations sur les sites internet des salons locaux :

Les sites sont simples. Sites one page sans barre de navigation, du texte de présentation, la localisation et quelques photos.

Les deux sites sur quatre sont un peu plus aboutis en termes de contenu. On y retrouve du texte de présentation mais également une boutique en ligne ainsi qu'un carrousel de photos de réalisations. Un site bénéficie d'un effet de parallax. On comprend tout de suite le thème lié à l'histoire sidérurgique de la ville de Hayange. Les couleurs dans des tons noirs/bruns s'y rapportent. La présentation des artistes à l'aide de photos et de texte y est faite et les photos sont quasi de style professionnel.

Conclusion : La majorité des salons de tatouages utilisent les réseaux sociaux pour communiquer. Cela s'explique par la facilité de poster des photos pour montrer leurs réalisations.

Les points positifs :

- Facilité de mise à jour du contenu texte et photos.
- Coût inexistant.

Les Inconvénients :

- Le référencement les entreprises possédant leurs propres pages apparaissent en premier.
- La perte de l'identité visuelle et graphique.

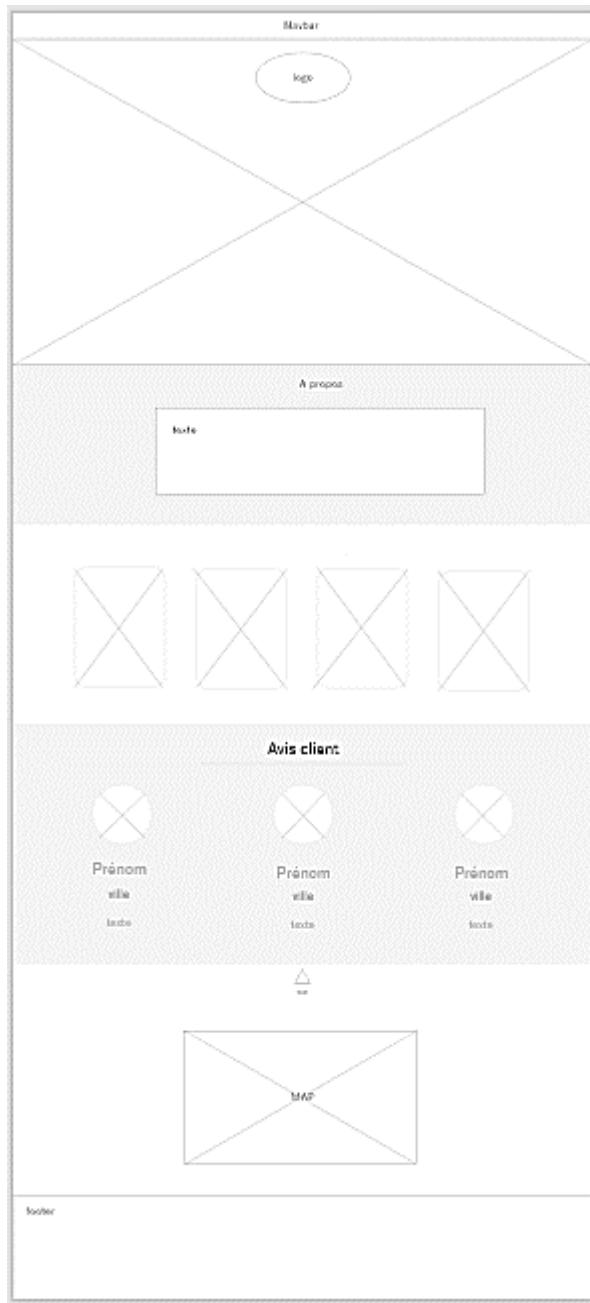
Inspirations :

Compte tenu du cahier des charges et du benchmarking, j'ai redirigé mes recherches vers les USA sur la côte Est et Ouest pour évaluer la qualité des sites et voir ce qu'il se faisait au niveau web outre-atlantique. Un site à particulièrement retenu mon attention : Le tatoueur New-Yorkais Bang Bang Forever. (<https://www.bangbangforever.com>). Ce site a été réalisé par une équipe de professionnel. Il est composé de vidéos et de photos bien réalisées visuellement et artistiquement parlant. Cela a donc représenté un challenge majeur en termes de réalisation artistiques. Défi que j'ai souhaité relever.

Pour la charte graphique du back-office, j'ai utilisé celle du template Bootstrap : Material Design Dark Edition réalisé par Creative Tim.

5.1.2. Réalisation du maquettage

Suite à mes recherches et après réflexion sur le planning prévisionnel, j'ai choisi de réaliser la maquette d'une page d'accroche à réaliser sous WordPress à mettre rapidement en ligne en attendant le site final.



J'ai ensuite réalisé une première maquette du site final à montrer au client dès le début du stage pour gagner du temps et confirmer l'orientation du projet.

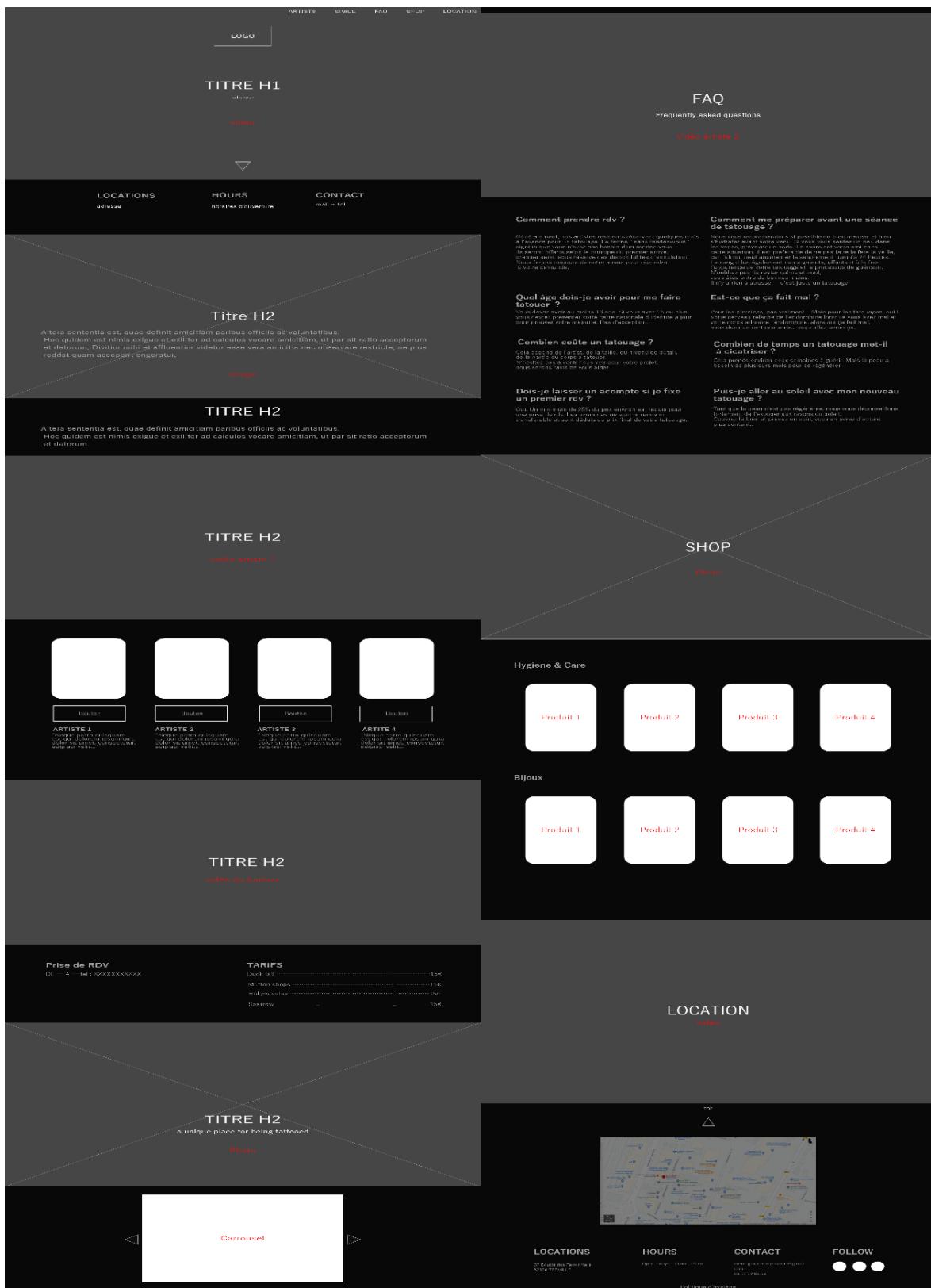


FIG. WIREFRAME

MAQUETTE VERSION DESKTOP

MAQUETTE VERSION MOBILE

5.2. Organisation pendant le stage

J'ai pris place aux côtés des tatoueurs dans l'open-space du 22 Février au 12 Mars 2021. Mes horaires étaient 9h 12h – 13h 17h.

J'avais mis à disposition le matériel adéquat pour travailler dans de bonnes conditions. Malgré la faible disponibilité des artistes, la proximité a néanmoins permis l'implication de ces derniers ainsi que du maître de stage pour les étapes de vérifications des photos et des textes. Ce qui a grandement facilité la conception du site mais également de bénéficier d'une excellente ambiance de travail.

Dès le début du stage et conformément au Cahier des Charges, j'ai tout d'abord présenté la première maquette à l'ensemble de l'équipe afin de recueillir leurs impressions, avis et conseil. Cette première étape fut fortement appréciée par l'équipe et valider dans la forme du site. De nouvelles informations sont venues modifier la maquette, ce qui donna lieu à une seconde version de la maquette du site.

5.2.1. Récupération des médias

J'ai œuvré à la récupération et au travail des médias dès le premier jour conformément au planning prévisionnel.

Le client m'a remis un ensemble d'environ 275 photos et une ancienne vidéo de présentation de Dermographe Corporation à modifier.

Cela a eu pour effet de :

- Consacrer une journée entière au tri des photos.
- Sélectionner les photos en accord avec le client.
- Evaluer la possibilité d'intégration des photos.
- Lister les photos manquantes pour le site.
- Planifier la réalisation des vidéos en fonction des emplois du temps des artistes et des clients du salon.

Les vidéos ont été réalisées avec l'accord des clients. Un stabilisateur fut nécessaire pour compenser les déplacements durant les travellings.

5.2.1. Directions artistiques

Les réalisations des photos et vidéos ont dû s'effectuer durant la période d'affluence au salon en pleine journée.

L'environnement sombre souhaité sera donc réalisé sur le site internet avec l'utilisation de calques noirs légèrement opaques (30%) sur les photos et vidéos, ainsi qu'un arrière-plan sombre pour les parties textes.

- Les photos ont été réalisé à l'aide :
 - o D'un appareil photo Canon EOS 1300D,
 - o D'un éclairage LED annulaire,
 - o Un kit de projecteurs à lampe
 - o Du logiciel Adobe Photoshop.
- Les vidéos ont été réalisé à l'aide d'un :
 - o Smartphone Huawei Honor 9 en résolution 4K,
 - o D'un stabilisateur DJI Mobile
 - o Du logiciel de montage SHOTCUT.

Les principales difficultés de la réalisation de vidéos ont été :

- Trouver un travelling artistique intéressant.
- Gérer les travellings en fonction de l'idée du résultat que l'on veut obtenir.
- Gérer l'environnement (passages, éclairage naturel, ombres...).
- Gérer les déplacements du corps.
- Gérer la vitesse des travellings.

Une dizaine de prises ont dû être réalisées au total.

Les montages ont été réalisés avec le logiciel Shotcut que j'ai découvert pendant le stage.

5.2.1. Réalisation du Front end

La réalisation du Front end s'est effectuée à partir de la 2^{ème} semaine de stage conformément au planning prévisionnel. Je me suis appuyé sur la 2^{ème} version de la maquette suite aux remarques de l'équipe. Seules les artistes ont une page dédiée pour chacun d'eux pour partager leurs réalisations.

Structure :

Le site étant sous forme de one page, la structure du site se divise en un ensemble de sections, avec deux sections par parties.

Pour le SEO :

J'ai suivi les règles de bases pour préparer le référencement du site en :

- Ajoutant dans le <head> une description
- Dans le <body> en utilisant des balises title différentes sur chaque page,
- Dans le <body> en utilisant les balises <h1>, <h2>,
- En utilisant des liens internes,
- Attributs title et alt sur les images

Version PC :

Après une seconde maquette, j'ai décidé de faire apparaître les parties suivantes :

- Présentation : Vidéo de présentation générale.
- Informations : Un premier rappel des horaires.
- Actualité : l'administrateur pourra faire apparaître et les actualités du shop.
Information modifiable depuis le back-office.
- Presse : Article de presse pour promouvoir le shop.
- Artistes : Présentation des artistes et liens vers leur Facebook et/ou Instagram.
- FAQ : Une foire aux questions/réponses les plus courantes pour les piercings et tatouages.
- Studio : Présentation du salon par une vidéo réalisée et mise en ligne sur YouTube.
- Contact : localisation google-map interactive, adresse, horaires, liens Facebook et Instagram du shop.

Pour les pages des artistes, la structure est organisée en section également sur le même principe. Le texte est lui remplacé par des photos gérables depuis le back-office.

Pour faciliter la mise en page des photos et rendre la navigation dynamique, j'ai effectué une recherche des différentes bibliothèques JavaScript disponible. Je me suis orienté vers l'utilisation de « Masonry » pour la mise en page dynamique et « Lightbox2 » pour la visualisation dynamique des photos.

Version mobile (largeur < 767px de large) :

La structure est identique. Les vidéos sont remplacées par des images afin de limiter la consommation de données sur les réseaux mobiles.

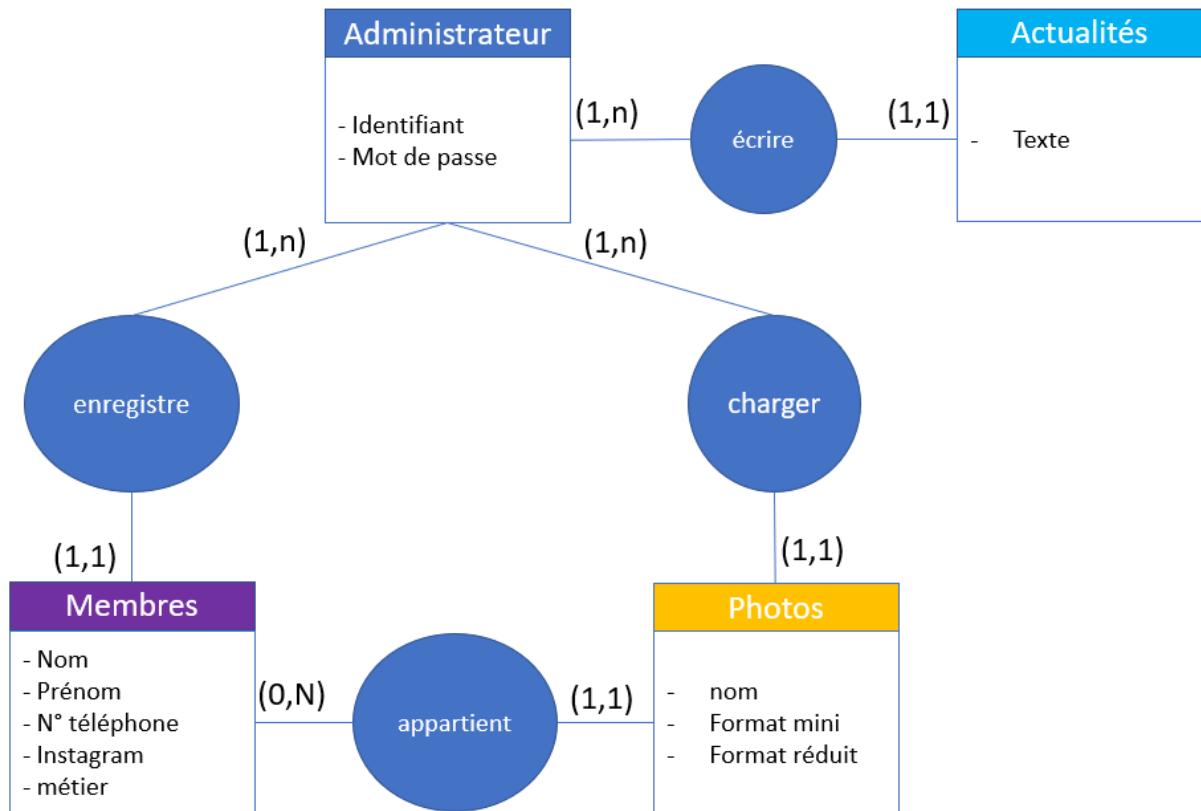
L'utilisation de Bootstrap a permis l'alignement en colonne de chaque élément de section.

Esthétisme :

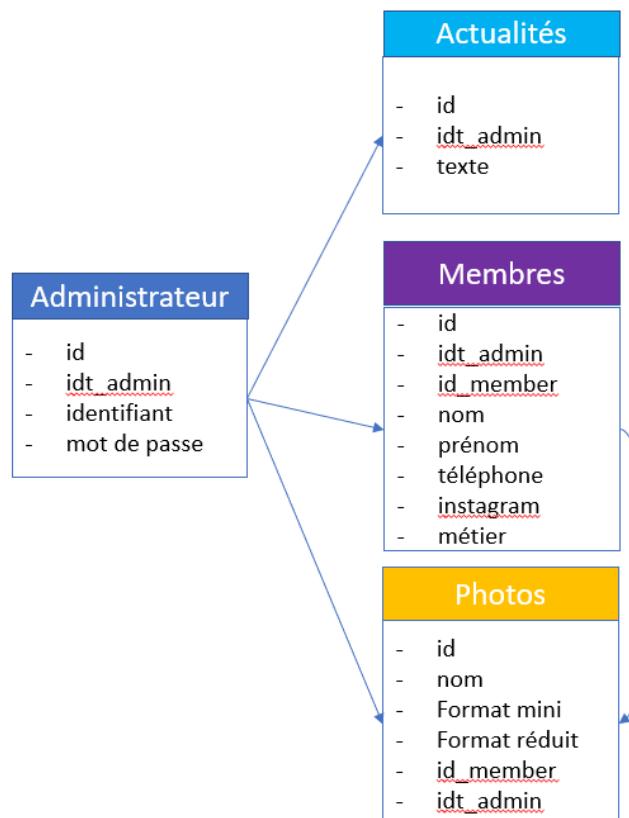
- La première section contient la vidéo ou l'image de présentation.
 - o Positionnement en background en fixe pour la vidéo et pour les images.
 - o Un overlay par-dessus positionné en absolue pour noircir à 30% les images ou la vidéo.
 - o Un titre h1 pour le nom du salon et h2 pour les autres parties au-dessus de l'overlay.
- La seconde section contient le texte. Ecriture grise sur fond noir. Type de police : Open Sans Condensed - sans serif.

5.2.2. Mise en place de la base de données

J'ai utilisé la méthode MERISE pour conceptualiser la base de données avant de la réaliser. Ceci m'a permis de réfléchir aux tables et aux champs nécessaires au bon fonctionnement du projet.



MCD DE LA BASE DE DONNEES



MLD DE LA BASE DE DONNEES

actualite

id	int	text	int
texte	text		
idt_admin	int		
Add field			

users
Contain user informations for login

id	int		
idt_admin	int		
username	varchar(24)		
password	varchar(128)		
Add field			

photos

id	int		
id_member	int		
idt_admin	bigint		
photo_name	varchar(128)		
path_small_img	varchar(128)		
path_resize_img	varchar(128)		
Add field			

MPD DE LA BASE DE DONNEES DB_DERMOGRAPH_Corp

5.2.3. *Réalisation du Back-end*

Le développement du back end s'est effectué en local depuis mon PC avec la suite WAMP.

La base de données, les tables et les champs ont été créé à partir de l'interface de PhpMyAdmin.

Une fois la base de données créée, j'ai réalisé la connexion à la base de données dans un fichier php intitulé db_connect.php.

J'ai ensuite créé les pages du Back Office avec les requêtes nécessaires à la réalisation des fonctionnalités que je souhaitais réaliser pour le site Front End et le site Back Office.

5.2.4. *Réalisation du Back-office*

Le back-office est la partie qui s'est révélée être la plus chronophage du projet. Afin de gagner du temps de développement, j'ai choisi d'appuyer le rendu du Back Office sur un template Bootstrap Material Design Dark Edition réalisé par CREATIVE TIM (www.creative-tim.com).

La page de connexion, elle, est indépendante et prend son inspiration d'une réalisation de Louis-Nicolas LEUILLET du site www.believemy.com et www.udemy.com.

5.2.4.1. Technologies

Les technologies employées pour le développement du Back Office ont été :

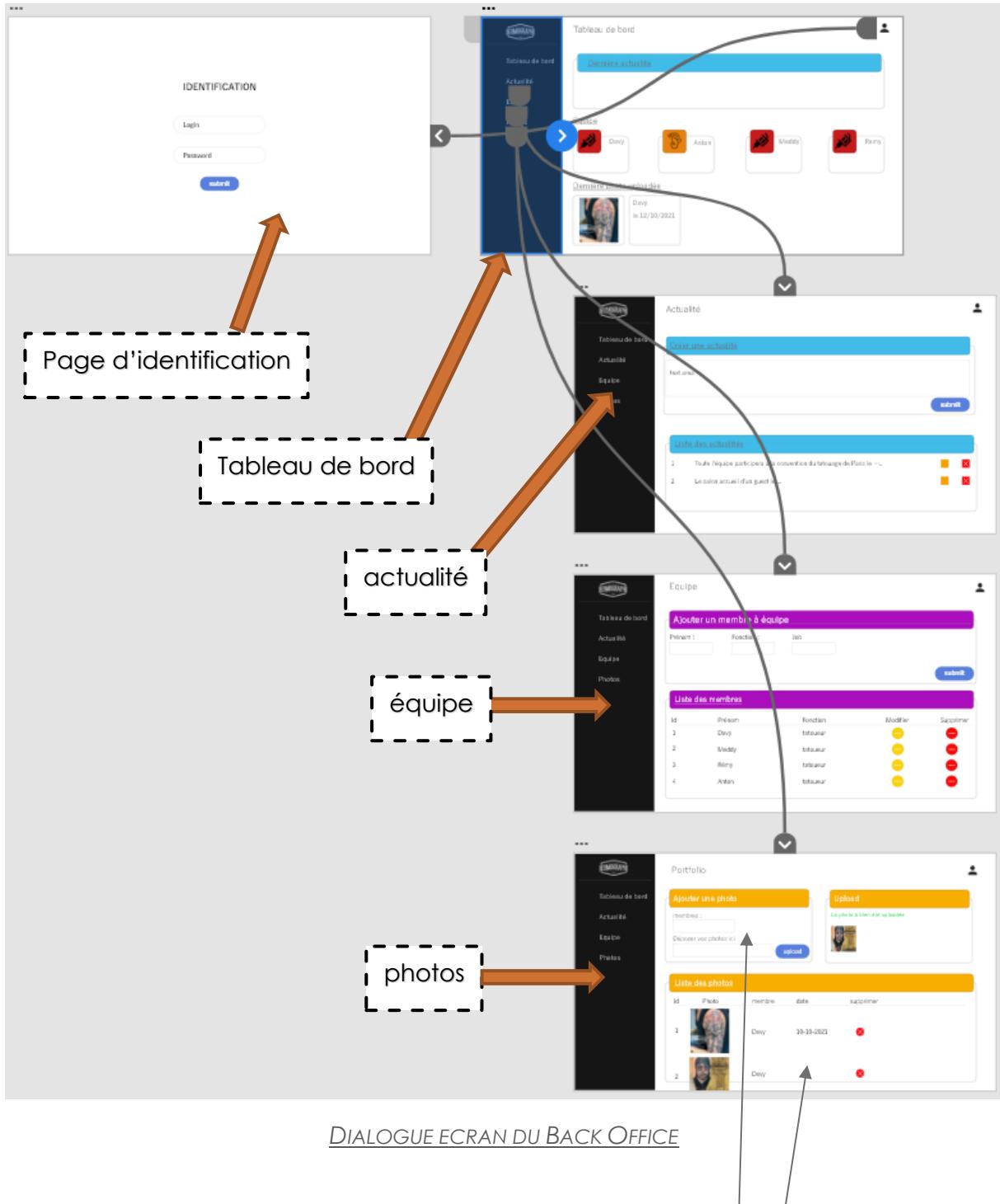
- HTML5 (modification du thème pour l'adapter à mon besoin).
- CSS3 (inclus dans le thème. Pas de modification).
- Javascript (inclus pour les fonctionnalités du thème).
- PHP PDO.

Les bibliothèques :

- JQUERY (inclus dans le thème).
- Bootstrap (inclus dans le thème).

5.2.4.2. Structure

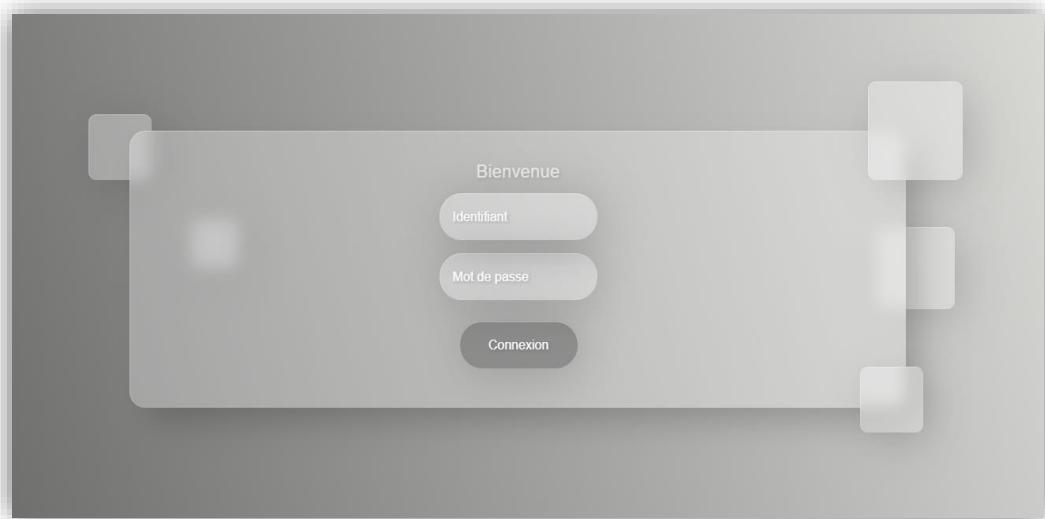
J'ai choisi de structurer le back-office de la manière suivante :



Chaque section de page contient des cartes Bootstrap, renferment soit un formulaire de création ou modification, soit un tableau de visualisation des données avec possibilité de déclenchement de modification ou suppression définitive de la donnée concernée.

5.2.4.3. Fonctionnalités réalisées

Page de connexion :



La page de connexion permet à l'administrateur de se connecter au Back Office. Etant donné qu'il n'y a qu'un seul administrateur, j'ai choisi d'enregistrer directement dans la base de données son identifiant et mot de passe haché en BCrypt. J'ai fait le choix de ne pas permettre d'inscription ni de pouvoir récupérer le mot de passe perdu car il n'y a qu'un seul administrateur, le gérant.

TABLEAU DE BORD :

A screenshot of the dashboard titled "Tableau de bord". On the left is a sidebar with icons for "Tableau De bord", "Actualité", "Équipe", and "Portfolio". The main area shows "Dernière actualité en ligne" with four items: Davy (informations), Anton (informations), Meddy (informations), and Remy (informations). Below that is "Dernière photo postée" showing a photo of a tattooed arm posted by Remy on 2021-04-16.

Le tableau de bord est une page récapitulative :

- De la dernière actualité postée,
- Des membres de l'équipe,
- De la dernière photo postée

Pas d'interaction possible avec les données dans cette page.

ACTUALITE :

The screenshot shows a web application interface for managing news. On the left, there's a sidebar with a logo for 'DERMOGRAPHIC' and navigation links: 'Tableau De Bord', 'Actualité' (which is highlighted in purple), 'Équipe', and 'Portfolio'. The main content area has a dark header 'Actualité'. Below it, a teal bar says 'Poster une nouvelle actualité sur le site'. A rich text editor toolbar is above a large text input field. In the bottom right corner of this bar is a red 'ENVOYER' button. Below this is another teal bar labeled 'Liste des actualités publiées'. A table with columns 'ID', 'Actualité', 'Modifier', and 'Supprimer' follows. The background of the page features a hexagonal pattern.

La page d'actualité est divisée en deux parties :

- Une carte contenant un formulaire pour créer une nouvelle actualité.
- Une carte contenant un tableau qui affiche toutes les actualités enregistrées dans la base de données.

EQUIPE :

The screenshot shows the 'Equipe' (Team) page. On the left, there's a sidebar with icons for Tableau De Bord, Actualité, Equipe (highlighted in purple), and Portfolio. The main content area has a header 'Equipe' and a sub-header 'Ajouter un membre'. It contains fields for Prénom*, Nom, Téléphone, Instagram, and Sélectionnez une fonction: (with a dropdown menu). A large 'AJOUTER UN MEMBRE' button is at the bottom right. Below this is a section titled 'Liste des membres enregistrés' with a table:

ID	Prénom	Nom	Fonction	Modifier	Supprimer
9	Davy		tatoueur		
10	Anton		perceur		
11	Meddy		tatoueur		
12	Remy		tatoueur		

La page d'équipe est divisée en deux parties :

- Une carte contenant un formulaire pour créer un nouveau membre.
- Une carte contenant un tableau qui affiche tous les membres enregistrés dans la base de données.

PORTFOLIO :

The screenshot shows the 'Portfolio' page. The sidebar includes icons for Tableau De Bord, Actualité, Equipe, and Portfolio (highlighted in purple). The main area has a header 'Portfolio' and two sections: 'Upload' and 'Prévisualisation'. The 'Upload' section has a dropdown 'Sélectionnez un membre de l'équipe:' with 'membre' selected, a file input 'Déposez votre photo ici:', and a preview area 'Prévisualisation' showing 'Vous n\'avez pas encore saisi de photo'. Below this is a section titled 'Liste des photos publiées:' with a table:

ID	Photo	Posté par	le :	Supprimer
56		Davy	2021-04-16	
60		Davy	2021-04-16	

La page Portfolio est divisée en trois parties :

- Une carte contenant un formulaire pour charger et enregistrer une photo. Il faut ici sélectionner au préalable le membre de l'équipe concerné. Il n'apparaîtra dans la liste que s'il a été créé avant dans la page « Equipe »
- Une carte de confirmation de l'upload et de l'affichage d'une miniature de la photo.
- Une carte contenant un tableau qui affiche toutes les photos enregistrées dans le fichier photos ainsi que les créateurs. Les opérations de modifications des informations ou suppression s'effectue également ici. Il est possible de trier les photos par membre de l'équipe en sélectionnant le membre souhaité dans la ligne de titre du tableau.

6. Création du site (Extrait de codes significatifs)

6.1. Extrait n°1: Back Office – Manipulation des membres de l'équipe.

6.1.1. Formulaire

Création d'un formulaire avec méthode « POST » et envoi des données sur la même page (le code de traitement php est au-dessus du html).

```
<form method="post" action="accueil.php?page=3">
    <div class="row">
        <div class="col-md-6">
            <div class="form-group">
                <label class="bmd-label-floating">Prénom *</label>
                <input type="text" name="first_name" class="form-control" value =
                    <?php if(isset($_GET['id'])) {echo $result['first_name'];} ?>" required>
            </div>
        </div>
        <div class="col-md-6">
            <div class="form-group">
                <label class="bmd-label-floating">Nom</label>
                <input type="text" name="last_name" class="form-control" value =
                    <?php if(isset($_GET['id'])) {echo $result['last_name'];} ?>">
            </div>
        </div>
    </div>
```

FIG.1 - EXTRAIT DU FORMULAIRE POUR LA CREATION DES MEMBRES.

Création d'un champs fantôme prenant la valeur 0 de base. Lorsque le bouton « edit » est cliqué, il envoie dans l'URL l'id du membre sélectionné. Le code de traitement est composé d'une condition récupérant la valeur de la variable dans l'URL. Cette stratégie permettra lors de la soumission du formulaire d'indiquer si l'on est dans un cas de création (value = 0) ou de modification des données (value = id) et ainsi aiguiller le type d'action à effectuer dans la base de données.

```
<input type="hidden" value="=php if(isset($_GET['id'])) {echo $_GET['id'];} else {echo "0";} ?&gt;" name="id"&gt;
&lt;button type="submit" name="submit" class="btn btn-primary pull-right"&gt;
Ajouter un membre&lt;/button&gt;
&lt;div class="clearfix"&gt;&lt;/div&gt;
&lt;/form&gt;</pre
```

FIG. 2 EXTRAIT DU CHAMPS PHANTOME

6.1.2. *Création*

```

//CREATE NEW MEMBER
//Verify click on submit
if(isset($_POST['submit'])) {
    //Verify first_name field is not empty
    if(!empty($_POST['first_name'])) {
        //Variables
        $first_name = htmlspecialchars($_POST['first_name']);
        $last_name = htmlspecialchars($_POST['last_name']);
        $phone = htmlspecialchars($_POST['phone']);
        $instagram = htmlspecialchars($_POST['instagram']);
        $job = $_POST['job'];
        $id = $_POST['id'];
        //CASE IF NEW ENTRY
        if($id == '0') {
            //Prepare the query
            $query = $bdd->prepare('INSERT INTO
members(first_name, last_name, phone, instagram, job) VALUES
(:first_name, :last_name, :phone, :instagram, :job)');
            $query->bindParam(':first_name', $first_name, PDO::PARAM_STR);
            $query->bindParam(':last_name', $last_name, PDO::PARAM_STR);
            $query->bindParam(':phone', $phone, PDO::PARAM_STR);
            $query->bindParam(':instagram', $instagram, PDO::PARAM_STR);
            $query->bindParam(':job', $job, PDO::PARAM_STR);
            //Execute the query
            $result = $query->execute();
            //Enable message according to the result
            if($result){
                $message = "Un nouveau membre à bien été ajouté";
            } else {
                $message = "Un problème à été rencontré";
            }
            $query->closeCursor();
        } else {

```

- On crée les variables contenant les informations envoyées et protégées
 - On prépare la requête d'insertion des données dans la table en spécifiant les champs et les valeurs.
 - On lie les valeurs à des paramètres de types souhaité pour prévenir des injections SQL.
 - On exécute la requête.
 - On vérifie que la requête à bien fonctionnée en affichant de confirmation à l'écran.

6.1.3. *Obtention*

```
<?php
$query = $bdd->prepare('SELECT * FROM members');
$query->execute();

while ($result = $query->fetch()) {
echo '
<tr>
<td>'.$result['id'].'</td>
<td>'.$result['first_name'].'</td>
<td>'.$result['last_name'].'</td>
<td>'.$result['job'].'</td>
<td>
    <a href="accueil.php?page=3&id='.$result['id'].'">
        <i class="material-icons" name="edit">edit</i>
    </a>
</td>
<td>
    <a href="accueil.php?page=3&delete='.$result['id'].'
        onclick="return(confirm(\'
        Etes-vous sûr de vouloir supprimer cette entrée?\'))">
        <i class="material-icons" name="delete">delete_forever</i>
    </a>
</td>
</tr>';
}
?>
```

L'affichage des données contenues dans la table ‘member’ s’effectue par une requête ‘SELECT’ dans le corps du tableau en bas de page.

6.1.4. *Mise à jour*

L’administrateur clique sur le bouton d’édition qui envoie la valeur de l’identifiant du membre que l’on souhaite modifier.

La valeur de l’id de l’champs fantôme récupère cette information et prend la même valeur.

Dans ce cas la requête devient un UPDATE de la table.

```

//UPDATE DATA
$query = $bdd->prepare("UPDATE members SET first_name =:first_name, last_name= :last_name, phone=:phone, instagram=:instagram, job=:job WHERE id = '". $id ."' ");
$query->bindParam(':first_name', $first_name, PDO::PARAM_STR);
$query->bindParam(':last_name', $last_name, PDO::PARAM_STR);
$query->bindParam(':phone', $phone, PDO::PARAM_STR);
$query->bindParam(':instagram', $instagram, PDO::PARAM_STR);
$query->bindParam(':job', $job, PDO::PARAM_STR);
$result = $query->execute();
$message = "La modification à bien été effectuée";
}

```

Fig.3 – Extrait du code de l'update

Les champs du formulaire récupèrent également la valeur de l'id contenu dans le champ fantôme et déclenchent une requête 'SELECT' pour récupérer les valeurs des champs correspondants au membre de l'équipe sélectionné :

```

//CALL DATA TO BE DISPLAYED INSIDE TEXTAREA
if(isset($_GET['id'])) {
    $query = $bdd->prepare("SELECT * FROM members WHERE id = '". $_GET['id'] ."' ");
    $query->execute();
    $result = $query->fetch();
}

```

6.1.5. *Suppression*

Si l'administrateur clique sur le bouton corbeille d'un membre, le bouton envoi la valeur de l'identifiant.

On récupère l'information via un GET puis :

- On crée une variable '\$delete' qui stock l'identifiant de l'utilisateur que l'on souhaite supprimer.
- On prépare une requête DELETE dans laquelle on fait correspondre la valeur du champ 'id' de la table à la valeur stockée dans la variable '\$delete'.
- On exécute la requête de suppression.

```

//DELETE
if(isset($_GET['delete'])) {
    $delete = $_GET['delete'];
    $query = $bdd->prepare("DELETE FROM members WHERE id = '". $delete ."' ");
    $result = $query->execute();
}

```

6.2. Extrait n°2 : Site Back Office – téléversement d'une image dans la galerie via le panneau d'administration.

6.2.1. Formulaire

Création d'un formulaire dans le fichier « photos.php » avec méthode « POST » et envoi des données dans le fichier « upload_photo.php » contenant le code de traitement des données.

Le fichier « resize_photo.php » contient le code permettant le redimensionnement de la photo en deux tailles plus petites. Ces deux derniers fichiers sont liés au fichier « photos.php » par des includes.

```
<?php include('upload_photo.php'); ?>
<?php include('resize_photo.php'); ?>

<form method="post" action="accueil.php?page=4" enctype=
'multipart/form-data'>
    <div class="row">
        <div class="col-md-6">
            <label for="member">
                Sélectionnez un membre de l'équipe :</label>
                <select name="member" id="member" class=
"form-control" style="width: 100px;" required>
                    <option value="0">-- membre --</option>
                    <!-- add a member as an option -->
                    <?php
                        $query = $bdd->prepare('SELECT * FROM members');
                        $query->execute();
                        while ($result = $query->fetch()){
                            $first_name = $result['first_name'];
                            $id = $result['id'];
                        }
                    ><option value="<?php echo $id; ?>"><?php echo
                        $first_name; ?></option>
                    <?php } ?>
                </select>
            </div>
        </div>
        <div class="row pt-3">
            <div class="col-md-12">
                <label>Déposez votre photo ici :</label>
                <div class="input-group no-border">
                    <input type="file" name="photo" value="" class
="form-control" placeholder="Sélectionnez une photo...">
                    <button type="submit" name="submit" class=
"btn btn-default btn-round btn-just-icon">
                        <i class="material-icons">upload</i>
                        <div class="ripple-container"></div>
                    </button>
                </div>
            </div>
            <div class="clearfix"></div>
        </div>
    </div>
</form>
```

6.2.2. *Création*

a. Création des variables

```
<?php
//Variables
$message= "";
$photo_name = time().' .jpg';
$date = date("Y-m-d");
$dossier = '.../pages/photos/';
$original = $dossier.$photo_name;
$vignette = $dossier.'vignette_'. $photo_name;
$mini = $dossier.'mini'. $photo_name;
```

b. Création de la requête d'insertion des données

```
//INSERT PHOTOS INSIDE THE DB
if(isset($_POST['submit'])) {
    //Variables
    if(!empty($_POST['member']) && !empty($_FILES['photo'])) {
        $member = $_POST['member']; //Post the value of the select, here the id of the member

        //INSERT query
        $query = $bdd->prepare('INSERT INTO photos VALUES (NULL
, :photo_name, :photo_date, :id_member, :path_original_img, :path_small_img, :path_resize_img)');
        $query->bindParam(':photo_name', $photo_name, PDO::PARAM_STR);
        $query->bindParam(':id_member', $member, PDO::PARAM_INT);
        $query->bindParam(':photo_date', $date, PDO::PARAM_STR);
        $query->bindParam(':path_original_img', $original, PDO::PARAM_STR);
        $query->bindParam(':path_small_img', $vignette, PDO::PARAM_STR);
        $query->bindParam(':path_resize_img', $mini, PDO::PARAM_STR);
        $result = $query->execute();

        if(!$result) {
            $message = "L'envoi de la photo n'a pas été effectué";
        } else {
            $message = "L'envoi est réussi !";
        }
        $query->closeCursor();
    }
}
```

- On vérifie que le membre a été sélectionné ainsi que la présence du fichier dans le formulaire.
- On prépare la requête dans la table en spécifiant les valeurs.
- On lie les valeurs à des paramètres. On enregistrera dans cette table le chemin de la photo originale, de la miniature et de la photo redimensionnée. On

enregistre également l'id du membre sélectionné dans le formulaire afin de lier la photo à un utilisateur.

- On exécute la requête.
- On vérifie si la requête est bien effectuée en affichant un message de confirmation.

6.2.3. *Obtention*

Même principe que pour l'affichage des membres.

On utilise une jointure interne pour afficher le nom du membre contenu dans la table 'member'. On affiche la miniature en récupérant le chemin complet.

```
<tbody id="photoBody">
    <?php
        $query = $bdd->prepare("SELECT
            photos.id, photos.photo_date, photos.path_small_img, photos.id_member, members.first_name
        FROM photos INNER JOIN members ON photos.id_member = members.id ");
        $result = $query->execute();
        while($result = $query->fetch()) {
    ?>
    <tr>
        <td><?php echo $result['id']; ?></td>
        <td></td>
        <td><?php echo $result['first_name']; ?></td>
        <td><?php echo $result['photo_date']; ?></td>
        <td><a href='accueil.php?page=4&delete=<?php echo $result['id']; ?>'>
            onclick='return(confirm
('Etes-vous sûr de vouloir supprimer cette entrée?'))'<i
                class="material-icons" name="delete">delete_forever</i></a></td>
    </tr>
    <?php
    }
    $query->closeCursor();
?>
</tbody>
```

Une fonction de tri selon le membre de l'équipe a été implémenter. Pour cela le langage JavaScript a été utilisé avec la bibliothèque AXIOS afin d'éviter un rafraîchissement de la page.

```

<script>
  window.addEventListener('load', _ => {
    let selector = document.getElementById("select"); //on chope le select
    let tbody = document.getElementById("photoBody"); //on chope l'emplacement de destination
    selector.addEventListener('input', e => { // à chaque changement d'input dans le select
      let formData = new FormData();
      // on crée un formulaire virtuel (php n'aime que les formulaires...)
      let id = parseInt(selector.options[selector.selectedIndex]
        .value);
      //selectedIndex donne l'index de l'option qui est choisie. Value récupère la valeur de l'option
      formData.append('select', id)
      // On rattache la valeur du select dans le formulaire contenu dans l'id.
      axios.post("trt.php", formData, {
        //on demande à Axios de poster la valeur dans le fichier trt.php
        headers: {
          'Content-Type': 'multipart/form-data'
        }, //en disant que c'est un formulaire
      }).then(resp => { //quand la réponse est reçue
        tbody.innerHTML = resp.data; //écrire la réponse
      }).catch(err => console.error(err));
    });
  });
</script>

```

Une requête parallèle est effectuée dans le fichier « trt.php » dans laquelle Axios enverra les données du formulaire virtuel.

6.2.4. *Update*

Je n'ai pas souhaité réaliser de fonction UPDATE dans ce cas particulier. Il n'est pas utile de modifier le membre ni la photo. En cas d'erreur, l'administrateur pourra simplement supprimer la photo et les informations liées et recommencer l'opération de création avec les bonnes informations.

6.2.5. Suppression

```
//DELETE PHOTOS
if(isset($_GET['delete'])) {
    $delete = $_GET['delete'];

    $query = $bdd->prepare ("SELECT * FROM photos WHERE id = '".$delete."'");
    $query->execute();
    $data_file = $query->fetch();

    $file_pointer = realpath($data_file['path_original_img']);
    unlink($file_pointer);
    $file_pointer = realpath($data_file['path_small_img']);
    unlink($file_pointer);
    $file_pointer = realpath($data_file['path_resize_img']);
    unlink($file_pointer);

    $query = $bdd->prepare("DELETE FROM photos WHERE id = '".$delete."' ");
    $result = $query->execute();

    $message = "La photo a bien été supprimé";

}
?>
```

Si l'administrateur appui sur le bouton de la corbeille, le bouton envoi le numéro de l'identifiant de la ligne correspondante.

Cette information est récupérée via un GET puis :

- On stock l'information dans une variable '\$delete'.
- On prépare la requête de sélection des données dans la table 'photos' en faisant correspondre l'id de la table 'photos' avec la valeur de l'id contenu dans la variable '\$delete' ;
- On pointe les fichiers en utilisant la fonction native 'realpath()' qui contient le résultat de la requête, en particulier la valeur des champs des chemins des fichiers.
- On supprime le fichier désigné avec la fonction native 'unlink()' ;

7. Présentation du jeu d'essais sur la fonctionnalité la plus représentative.

Une fois le back-office terminé j'ai effectué une série de test de vérification de bon fonctionnement sur la connexion au Back Office, de la page de login et des diverses pages de modification des actualités.

7.1.1. Exemple n°1 : Page de connexion

Ce jeu de test porte sur la connexion de l'administrateur au Back Office. La base de données possède une table contenant l'identifiant et le mot de passe hachés avec l'algorithme BCRYPT. Le premier jeu de test se réalise sur la page afin de vérifier la sécurité de la connexion.

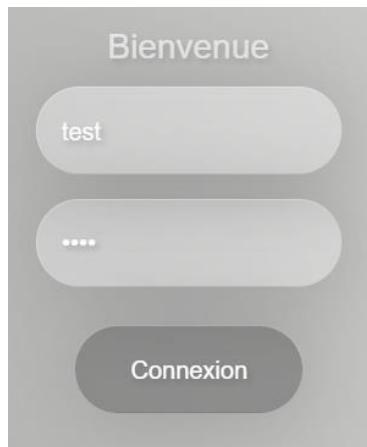
7.1.1.1. Test de tentative de connexion

Test 1 : Identifiant incorrect ou mot de passe incorrect

Si lors de la tentative de connexion, le pseudonyme existe dans la base de données, la fonction PHP « password_verify » va effectuer une comparaison du mot de passe haché et le mot de passe entré par l'utilisateur. Si les deux mots de passe correspondent, alors l'administrateur accède directement à la page du tableau de bord du Back Office.

Résultats attendus : Dans le cas d'une mauvaise authentification, un message d'erreur est attendu sur la page de Login. L'accès au Back Office est refusé.

Entrée :



Sortie :



Conclusion : le résultat attendu est confirmé.

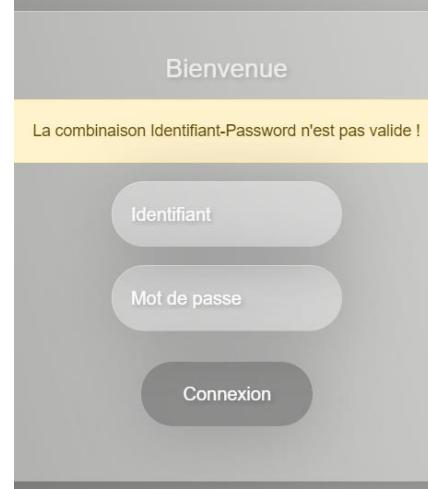
Test 2 : Absence de l'identifiant ou du mot de passe

Résultats attendus : Dans le cas de l'absence de l'identifiant et/ou mot de passe dans le formulaire, un message d'erreur doit s'afficher et la connexion au Back Office refusée.

Entrée :



Sortie :



Conclusion : le résultat attendu est confirmé.

Test 3 : Test de vérification de protection

- **Côté client :** Vérification de la protection contre les injections SQL

Pour cela, nous allons entrer dans le champ identifiant :

toto OR 1=1

Et dans le champ password :

toto OR 1=1

Résultats attendus :

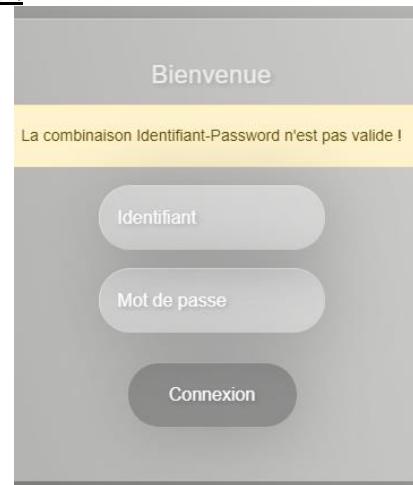
Dans le cas où le site est correctement protégé : affichage d'un message d'erreur « Vous avez entrez une mauvaise combinaison Identifiant/password ».

Dans le cas contraire, nous accèderons à la partie administrateur du site.

Entrée :



Sortie :



Conclusion : le site est correctement protégé.

7.1.2. Exemple n°2 : Test du formulaire pour les actualités

Cette phase de test va permettre de vérifier :

- **Côté client :**

- La bonne transmission des données vers la page Front End.
- La protection contre l'envoi d'un formulaire vide.
- La protection contre l'envoi de caractères spéciaux
- La bonne prise en compte des accents

- **Côté serveur :**

- La bonne transmission des données vers la base de données.
- La protection contre l'envoi de types de données autres que du texte (fichiers, codes etc.).

Test 1 : Envoi de la requête de création de l'actualité

Résultats attendus : Confirmation de l'envoi. Message envoyé sur la page Front end et dans la Back end.

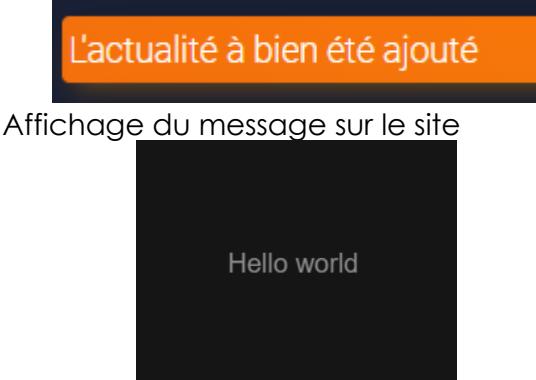
Entrée :

Entrée du message Hello world

The screenshot shows a rich text editor interface. At the top, a teal bar says "Poster une nouvelle actualité sur le site". Below it is a toolbar with icons for Paragraph, Bold (B), Italic (I), Underline (U), and other styling options. A text area contains the text "Hello world".

Sortie :

Affichage du message sur le site



Affichage du message sur le site
Enregistrement du message dans la base de données

+ Options	← T →	▼	id	texte
<input type="checkbox"/>			48	<p>Hello world</p>

Conclusion : Les résultats attendus sont confirmés.

Test 2 : Envoi d'un message vide

Résultats attendus : Affichage d'un message d'erreur dans le back office. Message par défaut sur le site. Aucune donnée dans la base de données.

Entrée :

Champ vide



Sortie :

Message d'erreur :

Le formulaire ne doit pas être vide !

Message sur le site :

Aucune nouveauté pour le moment. Restez à l'écoute !

Données :

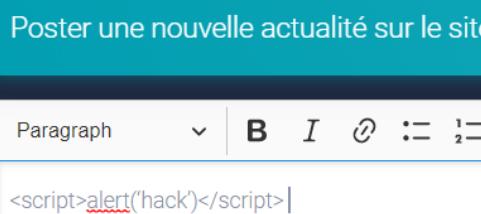
	id	texte
--	-----------	--------------

Conclusion : Les résultats attendus sont confirmés.

Test 3 : Protection du formulaire

Résultats attendus :

Entrée :



Poster une nouvelle actualité sur le site

Paragraph B I O := 1=

```
<script>alert('hack')</script>
```

Sortie :

L'actualité a bien été ajouté

<>alert('hack')</>

id	texte
-----------	--------------

50 <p><>alert('hack')</>&nbsp<

Conclusion : Les résultats attendus sont confirmés.

7.1.1. Exemple n°3 : Test sur les différents navigateurs

Pour cette phase de test, j'ai vérifié la bonne fonctionnalité du site sur les navigateurs suivants sur PC et smartphone :

- Chrome,
- Mozilla Firefox,
- Microsoft Edge

Résultats attendus :

- Pour le front : Pas décalage des images ou des icônes, couleurs respectées.
- Pour le back : Esthétisme respecté, couleurs respectées, fonctionnalités dynamiques fonctionnelles.
- Pas de changement majeur ou destructif

8. Veille technologique, sécurité, web mobile.

8.1. Faille XSS (Cross-Site Scripting)

[Extrait du site <https://www.web-eau.net/blog/10-outils-en-ligne-pour-tester-la-securite-de-votre-site>] :

Selon Cenzic, ce type d'attaque représente environ 25% des attaques. En quoi cela consiste ?

La faille XSS (plus officiellement appelée Cross-Site Scripting) est une faille permettant l'injection de code HTML ou JavaScript dans des variables mal ou non protégées.

[Extrait du site <https://www.bases-hacking.org/>] :

L'attaquant va être capable de modifier un aspect du site ou d'injecter des scripts dans ce que la victime va ensuite voir à l'écran. Il existe plusieurs types de failles XSS :

Le type connu en tant que **type 1**, ou vulnérabilité réfléchie résulte de l'utilisation de données fournies par l'utilisateur dans un script quelconque, sans les modifier. Typiquement, une simulation en ligne ou une page de statistiques. Ainsi, si ces données ne sont pas modifiées, on peut ajouter du "script dans le script" qui sera lui-même exécuté.

Ceci dit, en modifiant les données qui doivent être traitées, le résultat du XSS ne va modifier que la page que peut afficher l'utilisateur. Cela peut paraître bénin, mais ça l'est beaucoup moins quand l'attaquant utilise le Social Engineering et diffuse des pages piégées de cette façon. Ce genre de vulnérabilités est souvent utilisé pour lancer des campagnes de spam afin de ternir l'image d'un site (redirections, modifications d'apparence) ou de voler des informations (phishing).

Le **type 2**, ou vulnérabilité persistante ou du second ordre, permet des exploitations plus en profondeur. C'est la faille des livres d'or, présente dans les forums, les formulaires d'inscription. La différence essentielle est que les données entrées sont stockées dans des bases de données et sont renvoyées dès qu'un utilisateur les demande. Par conséquent, on peut affecter n'importe qui sollicitera un certain sujet dans un forum ou la liste des pseudos enregistrés, etc. Cette faille peut permettre des exécutions côté client ou côté serveur selon les cas et peut permettre tout type d'exploitation, de la récupération de cookies à l'exécution de scripts malveillants. On a vu des XSS intégrés à des bases de données institutionnels rendant inaccessibles des dizaines de sites dépendants de ces contenus.

Enfin, le **type 0**, connu comme le DOM-based ou Local Cross Scripting Site est un problème directement sur le script côté client (en général, le javascript) de la page (variables passées en URL qui sont réutilisées dans le javascript, etc.). Cette vulnérabilité est soit exploitée à nouveau pour le Social Engineering, soit par liens interposés dans lesquels on injecte du code qui sera ensuite exécuté côté client. Celui-

ci est finalement très sensible au type I et est très répandu et facilement repérable, notamment par des scanners automatisés.

Actions réalisées :

J'ai protégé les variables des formulaires avec les fonctions natives de PHP « `htmlspecialchars` » pour les chaînes de caractères et « `intval` » lorsqu'il s'agissait de valeurs numériques.

Les tests précédents ont montré la bonne exécution de ces protections coté client et coté serveur.

9. Description d'une situation de travail ayant nécessité une recherche sur site en Anglais.

Pour la responsivité du site, j'ai cherché plusieurs solutions pour adapter la vidéo et les images de la version PC vers la version mobile et tablette.

Pour la vidéo, j'ai trouvé la solution sur www.redstapler.co après avoir tapé la recherche suivante dans google : « how to make a back-ground video responsive ». Bien entendu des solutions en CSS sont présentées mais un paragraphe a retenu mon attention :

“The final step is to replace the footage with a static image on mobile. We usually do not want to play the video on mobile due to high data cost. We'll add another media query for any devices that have 767px width and below. Simply add display: none to the video tag and the poster image as background to the body.”

Que j'interprète par :

L'étape finale est de remplacer les dimensions (littéralement le métrage) par une image statique sur mobile. Nous ne voulons pas généralement lire une vidéo sur mobile du fait de la forte consommation de données. Nous allons ajouter un autre media query pour les appareils ayant une largeur de 767 pixels et moins.

J'ai donc suivi ce conseil et rajouté ce media query tel qu'indiqué sur le site.

Pour les images, j'ai effectué cette même manipulation afin de remplacer les photos en mode paysage par des photos en mode portraits.

```
@media (max-width: 767px) {  
    #video-intro {  
        display: none;  
    }  
    body {  
        background: url('../images/DSCF0248.JPG');  
        background-size: cover;  
    }  
    .artistsImg{  
        background: url('../images/davy.jpg');  
        background-position: center center;  
        background-repeat: no-repeat;  
        background-size: cover;  
    }  
    .studioImg {  
        background: url('../images/customer1.jpg');  
        background-position: center center;  
        background-repeat: no-repeat;  
        background-size: cover;  
    }  
    .locationImg {  
        background: url('../images/customer2.jpg');  
        background-position: center center;  
        background-position: center center;  
        background-repeat: no-repeat;  
        background-size: cover;  
    }  
}
```

10. Extrait du site anglophone

Cet extrait provient du site de Masonry. Je cherchais un moyen d'implémenter cette fonctionnalité. J'ai donc suivi pas à pas le processus d'implémentation et pousser plus loin mes recherches pour plus de compréhension du fonctionnement de Masonry.

Extrait du site et traduction en français=> bibliothèque Masonry > Options

All options are optional, but `columnWidth` and `itemSelector` are recommended.

Traduction : Toutes les options sont optionnelles, mais columnWidth et itemSelector sont recommandés.

itemSelector

Specifies which child elements will be used as item elements in the layout.

We recommend always setting `itemSelector`. `itemSelector` is useful to exclude `sizing elements` or other elements that are not part of the layout.

Traduction : Spécifie quels éléments enfants seront utilisés en tant qu'item dans la mise en page.

Nous recommandons toujours de régler itemSelector. ItemSelector est utile pour exclure les éléments non redimensionnables ou qui ne font pas partie de la mise en page.

columnWidth

Aligns items to a horizontal grid.



We recommend setting `columnWidth`. If `columnWidth` is not set, Masonry will use the outer width of the first item.

Use `element sizing` for responsive layouts with percentage widths. Set `columnWidth` to an `Element` or `Selector String` to use the outer width of the element for the size of the column.

Traduction :

Aline les éléments sur une grille horizontale.

Utilisez le dimensionnement d'élément pour une mise en page responsive avec une largeur en pourcentage. Réglez columnWidth pour un élément ou une chaîne de sélection pour utiliser la largeur extérieure de l'élément pour la taille de la colonne colonne.

Element sizing

Sizing options `columnWidth` and `gutter` can be set with an element. The size of the element is then used as the value of the option.

The option can be set to a Selector String or an Element.

Element sizing options allow you to control the sizing of the Masonry layout within your CSS. This is useful for responsive layouts and media queries.

Traduction : Les options de redimensionnement `columnWidth` et `gutter` peuvent être réglées avec un élément. La taille de l'élément est alors utilisée comme valeur de l'option. Cette option peut être utilisée sur une chaîne de sélection ou un élément.

Les options de dimensionnement de l'élément vous permettent de contrôler le dimensionnement des éléments mis en page à travers votre CSS. Ceci est utile pour les mises en pages responsives et les media-queries.

11. Conclusion

Ce stage fut une expérience intéressante et enrichissante tant techniquement qu'humainement.

Il fut représentatif de la réalité de la profession puisque le stage est arrivé prématûrement par rapport au planning d'apprentissage. Il a donc fallu faire preuve de souplesse et de persévérance en allouant du temps supplémentaire à la recherche et l'apprentissage de PHP PDO, technologie choisie pour le développement.

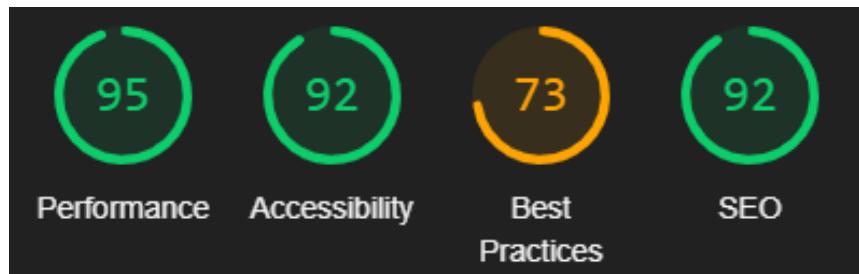
Cependant, la date de livraison n'ayant pas été définis par le client et par le formateur référent, ceci m'a permis de travailler encore dessus pour respecter le cahier des charges et les attentes du client.

Ce projet m'a permis de confirmer mes connaissances acquises au cours de la formation, d'en acquérir de nouvelles, de les améliorer par la recherche et de confirmer mon envie de pratiquer ce métier.

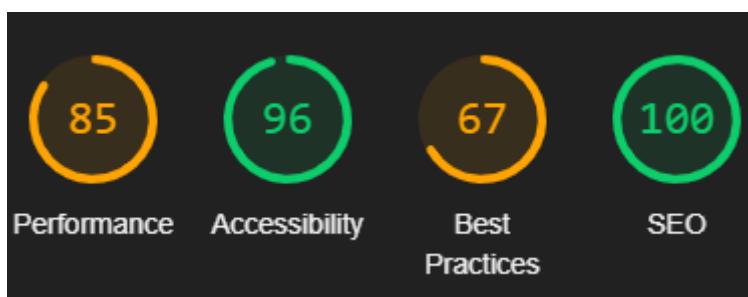
12. Annexes

J'ai souhaité mettre en ligne le site sur mon nom de domaine et mesurer sa performance avec Lighthouse avant la mise en production dont voici les résultats :

Version Desktop :



Version Mobile :



Les baisses de performances au niveau des « Best Practices » sont liées au fait que mon site www.jimprojects.com n'utilise pas de protocole HTTPS malgré un certificat SSL chez mon hébergeur.

13. Références

- ❖ <https://www.w3.org/>
- ❖ <https://www.php.net/>
- ❖ <https://frontendscript.com/video-parallax-background-css/>
- ❖ <https://redstapler.co/responsive-css-video-background/>
- ❖ <https://www.creative-tim.com/>
- ❖ <https://masonry.desandro.com/>
- ❖ <https://lokeshdhakar.com/projects/lightbox2/>
- ❖ <https://www.bases-hacking.org/faille-xss.html>
- ❖ <https://www.lije-creative.com/script-php-faille/>
- ❖ <https://www.web-eau.net/blog/10-outils-en-ligne-pour-tester-la-securite-de-votre-site>

14. Remerciements

- Mme Patricia SALUSTI, coordinatrice de formation continue du GRETA pour cette formation.
- M. Jordan STIEN, Formateur Référent pour son suivi et support technique.
- M. Laurent LAGONDET, Formateur pour son suivi et support technique.
- M. Ralph LEMASSON, Formateur pour son suivi et conseils.
- M. Christian HESCHNG, Formateur pour son suivi et conseils.
- M. Louis-Nicolas LEUILLET pour ses cours sur Udemy.
- M. Davy CHARISSOU de Dermographe Corporation pour m'avoir offert l'opportunité de ce stage.
- L'équipe Dermographe Corporation pour leurs disponibilités et de s'être prêtés au jeu pour les shootings photos et vidéos.
- L'équipe CREATIVE TIM pour leur template Bootstrap.
- Mes collègues de la formation pour leurs soutiens, appréciations et aides techniques.
- Ma compagne pour sa patience et son soutien durant mon incubation de 9 mois en formation.
- Mon fils de 9 ans pour son intérressement à mon projet et le développement.
- Mickey Mouse et Dingo qui ne liront probablement jamais ce rapport.