

# RAPPORT DE PROJET DE FIN D'ÉTUDE

*Conception d'un site internet*



**Oberhauser Clément**

16/03/2021

DEUST IOSI – 2ème Année

# SOMMAIRE

<b>SOMMAIRE</b>	<b>1</b>
<b>INTRODUCTION</b>	<b>2</b>
<b>DÉVELOPPEMENT ET RÉALISATION</b>	<b>3</b>
1 - Identité graphique du Site: la Maquette	3
2 - Réalisation des différentes pages : HTML / CSS	15
3 - Ajout des animations et API : Javascript et GSAP	44
<b>CONCLUSION</b>	<b>55</b>

## INTRODUCTION

Pour choisir le sujet du projet en accord avec ma poursuite d'étude: le développement web, j'ai donc cherché un site internet à réhabiliter. De ce fait, et après quelques recherches, j'ai choisi de remettre au goût du jour le [site du Club Vosgien Lemberg](#). Le Club Vosgien Lemberg est une association de bénévoles dont le but est de créer et entretenir les sentiers ainsi que de faire découvrir les forêts et randonnées du "Bitcherland".

Ce projet a donc une double vocation, tout d'abord réaliser un travail sur un sujet qui me passionne avec des technologies très intéressantes et enfin, pouvoir mettre en ligne un nouveau site pour le club Vosgien. De plus, il est certain que les personnes utilisant ce site ne seront pas toutes adeptes de nouvelles technologies, les pages doivent être intuitives tout en étant agréables à visiter. Ce, en plus des différentes animations pouvant permettre une meilleure interactivité. Existant depuis environ 8 ans, le site actuel n'a pas eu de remise à jour depuis.

Le but final du projet est donc d'avoir un site internet, développé en front-end, pouvant être mis en ligne à l'aide d'un CMS ou d'un serveur, par exemple. La maquette sera réalisée à l'aide d'un logiciel adapté: ADOBE XD. Pour la réalisation du code, plusieurs langages de programmation sont utilisés: HTML pour le balisage du site, CSS pour la mise en forme, Javascript pour les différentes interactions et enfin GSAP (développé par GreenSock) comme librairie Javascript pour les animations. Le site est décomposé en 6 pages: L'index (page d'accueil), le Refuge du Chalet du Club Vosgien, Découvrir Lemberg, Lien Amis/Contact, Randonner à Lemberg et enfin Sorties du Club Vosgien. Nous verrons le développement et la réalisation du site suivant ces trois parties: "Identité graphique du site: la maquette", "Réalisation des différentes pages HTML/CSS" et enfin, "Ajout des animations et API: Javascript et GSAP".

# DÉVELOPPEMENT ET RÉALISATION

## 1 - Identité graphique du Site: la Maquette

La réalisation de la maquette, avant de commencer à coder le site, permet d'avoir une ligne directrice pour la réalisation du projet. Celle-ci n'étant pas un modèle strict à suivre, le résultat final pourra donc contenir quelques changements. Elle permet tout de même de définir les codes couleurs, les polices d'écriture, les différentes idées de placement des textes, images, etc.. Nous allons voir, page par page, les différentes créations graphiques. Aucune des animations n'a été réalisée sur Adobe Xd pour une meilleure optimisation du temps. Le code couleur utilisé pour le site est le suivant : pour le background `#EFEADF`, les fonts seront en noir ou blanc (de même couleur que le background), certaines parties `#0e5e26` en `#221914` pour ajouter du contraste ou de la couleur. La font sélectionnée pour ce projet est "[Poppins](#)" en épaisseur 300, 500, 700. Les différentes API (MAPS/ Google Calendar) sont illustrées sur la maquette par de grands rectangles roses.

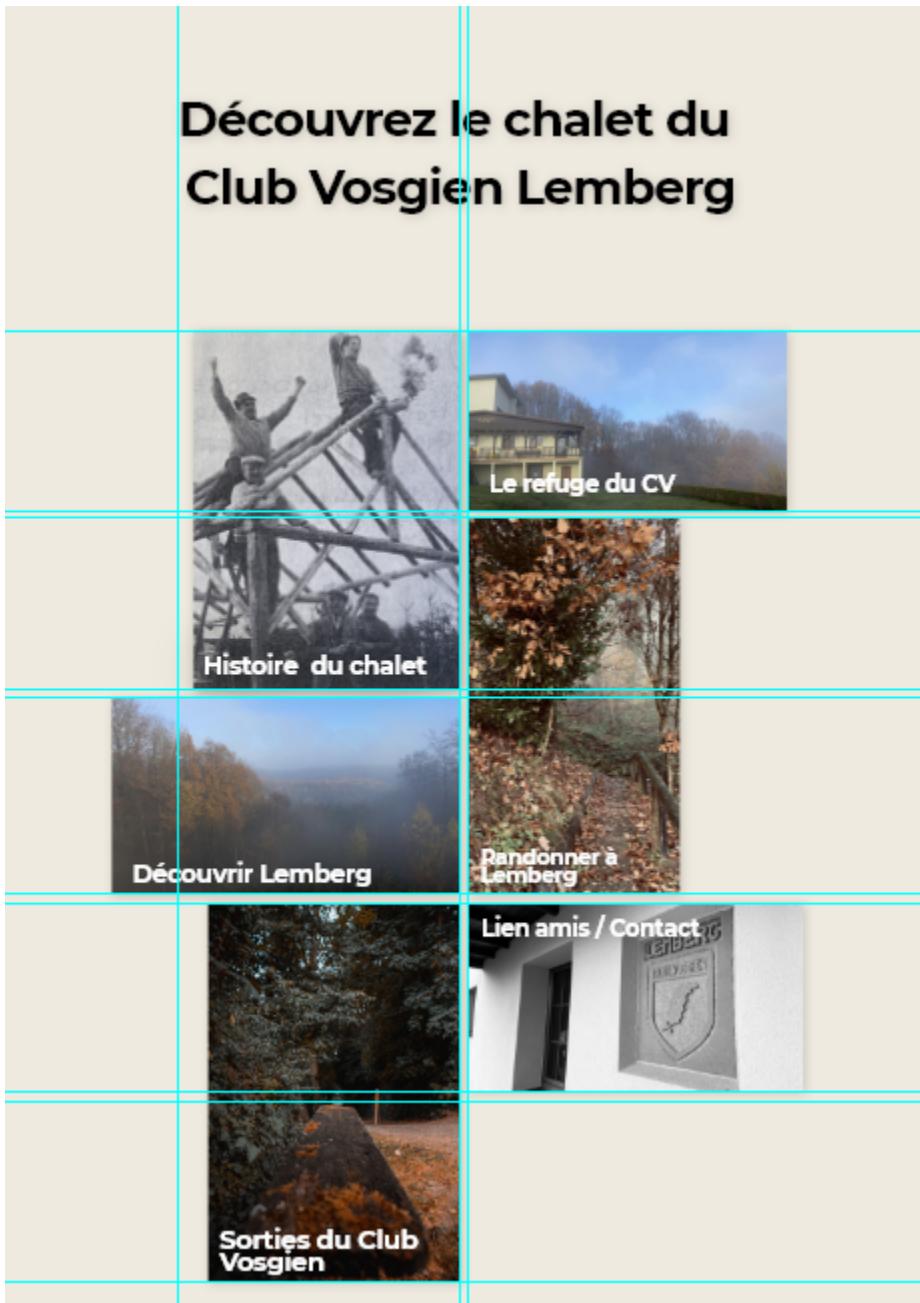
### L'index de la page :

La page d'accueil a une certaine importance: c'est la première impression que nous avons d'un site. J'ai donc décidé de réaliser celle-ci avec une image de couverture, contenant une belle image du Chalet, le logo, ainsi qu'un texte accrocheur.



La suite de la page doit

contenir les différents liens vers les autres pages du site, j'ai donc décidé, dans un souci d'ergonomie et de facilité d'utilisation, de créer ces différents liens sous forme de mosaïque. Pour lier la couverture avec la mosaïque, l'ajout d'un titre est nécessaire :



Le quadrillage permet, lors de l'intégration de cette maquette, d'organiser les images sur la page. Enfin, le footer permet d'indiquer la fin de page et de donner des informations sur les différents réseaux sociaux, on rappelle le logo sur la gauche:



Réseaux sociaux



© 2021  
Club Vosgien Lemberg

Ce dernier sera commun à toutes les pages.

### Refuge du Chalet du Club Vosgien :

Toute page commençant par un titre, j'ai décidé de l'entourer de deux rectangles, c'est un habillage qui n'est pas trop lourd mais est un peu plus original.

# Le Refuge du Chalet du Club Vosgien Lemberg

On y applique à chaque fois une ombre pour ajouter du contraste. Le reste de la page est classique avec des images ainsi que du texte :

Le Club Vosgien de Lemberg possède un gîte de randonnée.  
D'une capacité de 30 lits, il vous permettra de passer un excellent séjour  
dans les Vosges du Nord.



Le Chalet est fait sur trois étages :

- La grande salle
- Les dortoirs
- La buvette

Un abri pour faire des grillades  
et des pizzas est à disposition  
des locataires



Vous y trouverez tout l'équipement nécessaire:

- Une cuisine professionnelle.  
Équipée d'un lave vaisselle, d'une petite chambre froide et d'un fourneau,  
elle permet de préparer des plats pour plus de 75 personnes.
- 4 dortoirs. Ils sont équipés de lits superposés.
- Deux chambres équipées de douches et lavabos.



Le chalet refuge du Club Vosgien de Lemberg possède l'agrément Jeunesse et sport Numéro 1108 du 22.02.2011

Le gite est disponible à la location par semaine ou par week-end.  
Pour connaitre les disponibilités et réserver en ligne vous pouvez vous rendre sur la page  
[Réserver en ligne](#)

Lemberg est un petit village de Moselle.  
Il se situe dans l'arrondissement de Sarreguemines et le canton de Bitche.

Le quadrillage sert à nouveau à mieux organiser la page lors de son intégration.

Suite à cela, “l'accès au CVL” contient une API Maps permettant à l'utilisateur de se situer par rapport à l'emplacement du Club Vosgien:

# Accès au CVL



Le village est rapidement accessible depuis Wingen sur Moder (67) via Goetzenbruck, Bitche (57) via Reyerswiller, Rohrbach les Bitche (57) via Enchenberg.

Dans le village, le chalet est accessible en remontant la rue du stade ou la rue de la montagne :



Après avoir passé la partie “Heures d’ouverture” qui est identique aux début et ne contient que du texte, passons maintenant à la partie “Réservez en ligne” :

# Réservez en ligne

Avec le formulaire ci-dessous, vous pouvez réserver le refuge du Club Vosgien de Lemberg.

La réservation ne sera effective qu'après acceptation du contrat qui vous sera envoyé.

Sélectionnez vos jours puis complétez les champs. Un mail de confirmation vous sera envoyé.

**Pour des demandes de renseignements (tarifs, équipements...) utilisez le menu « Contact »**

Envoyez votre demande en fonction des disponibilités :

Nom \_\_\_\_\_

Prénom \_\_\_\_\_

Téléphone \_\_\_\_\_

E-mail \_\_\_\_\_

Message

Celle-ci contient l'API google Calendar permettant de consulter les différentes réservations qui seront déjà sur le calendrier. Ensuite, nous avons la partie formulaire,

qui permet à l'utilisateur d'envoyer un message à l'administrateur du site pour faire une demande de réservation. Dans le cadre de ce projet, je n'ai pas ajouté de bouton car cette page n'aura pas à vocation d'être mise en ligne directement mais dans les quelques mois suivants. Ainsi, un entretien avec l'administrateur du site sera fait pour avoir un bouton conforme à leurs attentes.

## Découvrir Lemberg :

Le but de cette page est de permettre à l'utilisateur de consulter les randonnées remarquables de la forêt avoisinante de Lemberg.



Après avoir réalisé le titre ainsi que le texte d'introduction, j'ai ajouté une API Maps permettant d'avoir un visuel direct sur le point remarquable de chaque randonnée.

La suite de la page contient les 11 articles sur chaque randonnée, elles se présentent à peu de choses près comme la première:

## 01 | La source Saint Hubert

Accès : Depuis le chalet du Club Vosgien. Descendre les escaliers et suivre le chevalet jaune.  
Traverser la route de Bitche et continuer sur le large chemin.  
Suivez le chevalet jaune jusqu'à la source Saint Hubert. Revenez sur vos pas pour le retour.  
Compter une bonne heure pour l'aller retour.

Descriptif: Le rocher sculpté date de l'époque gallo-romaine.  
Les historiens s'accordent à dire qu'il remonterait au IIIème siècle de notre ère.  
Initialement appelé « Bomphosen Brunnen », soit « Fontaine aux culottes bouffantes »  
le nom a été francisé quelque tardivement. Saint Hubert, patron des chasseurs,  
a donné son nom à tort à cette sculpture en l'interprétant comme une scène de chasse.

On y identifie malheureusement plus que la partie basse de deux divinités romaines :  
Sylvain, dieux de la forêt avec un bâton et Diane, déesse de la chasse, avec son arc.  
On y identifie également assez facilement un combat de cerfs,  
des sangliers ou un auroch, boeuf sauvage disparu au moyen âge.

A quelques mètres,  
un lavoir récemment restauré servait aux villageoises pour laver leur linge.  
Un personnage se baignant les pieds sur la sculpture pourrait attribuer  
des vertus curatives à cette source.



En premier lieu, nous avons le sous titre de chaque randonnée avec son numéro, un texte, alterné, pour bien les séparer. Enfin, une image d'illustration pour représenter le point important.

## Randonner à Lemberg :

Cette page permet de présenter à l'utilisateur les différents balisages et guides qui existent pour se promener au mieux dans les forêts à proximité.

# Randonner à Lemberg

Le Club Vosgien de Lemberg balise  
et entretient près de 180 kilomètres de sentiers. Afin de les rendre accessibles au plus grand nombre  
d'amoureux de la nature,  
tous les chemins sont balisés avec les signes  
conventionnels de la Fédération du Club Vosgien:

	Les signes du Club Vosgien®			
	Automne	Hiver	Printemps	Été
Randonnée (T)	■	■	■	■
Entierge (H)	◆	◆	◆	◆
Naturiste (vert) (N)	□	□	□	□
Vélo (E)	▲	▲	▲	▲
Cochonement (M)	+	+	+	+
Chasse (S)	✗	✗	✗	✗
Équitation (D)	●	●	●	●
Arrivée (O)	○	○	○	○

Le Club Vosgien a établi des règles permettant de choisir dans le tableau le « bon » signe en fonction du type de projet et du niveau atteint sur son chemin de randonnée, en fonction de l'heure et de la saison. C'est pourquoi il existe plusieurs types de balisage et il est donc possible d'en faire plusieurs sur un même itinéraire.

Après le titre, nous avons des petits textes explicatifs ainsi que des images d'illustrations.

Vous pouvez découvrir en toute sérénité le fabuleux patrimoine naturel et historique qui berce le village de Lemberg et ses environs. Que ce soit en sous-bois, dans les vallées où coulent de nombreux ruisseaux, près des sculptures gallo-romaines ou près des multiples lavoirs, toute la famille pourra venir se ressourcer dans ce beau village des Vosges du Nord.

En tout état de cause, pour préparer votre randonnée, favorisez toujours les ouvrages portant le logo du club vosgien :



## Liens Amis / Contact :

Cette page permet à l'utilisateur d'avoir les liens vers les différents sites utiles à tout randonneur ou voyageur et permet aussi de contacter un administrateur pour tout autre question qui ne soit pas en lien avec les réservations. Les 4 liens sont présentés comme suit:

**Lien Amis**  
**Contact**

---

## **Liens Amis**

Vous trouverez ci-dessous des liens vers des sites amis du CVL:

[L'office du tourisme du Pays de Bitche](#)

[Le Club Vosgien de Bitche](#)

[Le Club Vosgien de Soucht](#)

[Le ski club vosgien de Masevau](#)

Ceux-ci seront à animer en css.

Pour ce qui est de la partie contact, nous ajoutons un formulaire, celui-ci est composé des informations “classiques” (Nom, Prénom, adresse e-mail, message) ainsi que d'une sélection pour cibler les différentes questions (hébergement, infos diverses, webmaster), elle est représentée ici par un rectangle rose :

**Contact**

Quelle que soit votre question (location du chalet, horaires d'ouverture, état des sentiers, programme des sorties...), vous pouvez nous contacter en remplissant le formulaire ci-dessous :

Vos commentaires ou questions sont bienvenues.

Sélectionnez un contact:

Nom

Prénom

Courriel

Confirmez votre Courriel

Message

ou par téléphone:

• Le refuge aux heures d'ouverture: 03 87 06 81 26  
• Le président de l'association, Mlle Brunner: 06 77 23 29 15  
• Pour des questions concernant nos randonnées, M. André Neiter: 06 31 09 50 86

Pour des questions concernant la location du refuge utilisez le formulaire « contact »

La page sorties du Club Vosgien, contenant les dernières actualités de celui-ci, n'ont pas été réalisées sur une page lors de la création de la maquette. J'avais ajouté une petite partie contenant chaque article à la suite de la mosaïque dans l'index. Cependant, sachant qu'il y a beaucoup d'actualités qui sont ajoutées, j'ai décidé, une fois avoir créé la majorité des pages, d'en créer une dédiée.

Pour conclure, la création de la maquette est une partie très chronophage car, elle a nécessité une analyse des différentes pages du site actuel du Club Vosgien. Ceci pour en faire ressortir le plus pertinent et permettre d'aller à l'essentiel. C'est une partie qui n'est pas propre au développement mais elle est très prise au sérieux en entreprise car nécessaire pour organiser et structurer un projet comme celui-ci.

## 2 - Réalisation des différentes pages : HTML / CSS

Une fois avoir créé la maquette, passons à la réalisation du projet dans son code. Tout d'abord, html et css.

Nous ne verrons pas ici tout le code en détail tant il est dense, cependant, le code est commenté et accessible dans le dossier.

### L'index de la page :

Pour commencer, en tête de page, il est nécessaire de déclarer l'icône de la page, le titre, les polices d'écriture ainsi que la feuille de style css:

```
<head>
  <meta charset="utf-8">
  <title>Chalet du Club Vosgien Lemberg</title>
  <link rel="icon" type="image/png" href="img/icon.png">
  <link rel="stylesheet" href="css/style.css">
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link
    href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;700&display=swap" rel="stylesheet">
</head>
```

La page d'accueil est intégrée comme suit. L'image principale de la couverture est placée à une hauteur de 100vh (pour permettre de faire la taille de l'écran de l'utilisateur) et une largeur de 100%. Le texte est, quant à lui, placé en position absolute pour être placée par dessus la cover:

```
<div class="texttopleft">

    <div class="aventure">
        <div class="av1">
            <span>Vivez </span>
        </div>
        <div class="av2">
            <span>L'Aventure</span>
        </div>
    </div>

    <div class="txtclubvosg">
        <span>Club Vosgien Lemberg</span>
    </div>
</div>
```

On définit des propriétés communes à une grande majorité des textes des pages :

```
font-family: 'Poppins', sans-serif;
font-weight: 500;
color: #EFEADF;
font-size: 160px;
line-height: 0.8em;
text-align: center;
text-shadow: rgba(0, 0, 0, 0.30) 0px 0px 99px;
```

Comme sur toutes les autres pages, le logo est placé en haut à gauche de la page en **position: fixed** car on l'utilisera dans la dernière partie: Animation (il servira comme bouton pour retourner à l'accueil).

```
<div class="logo">
    
</div>

.logo {
```

```

height: 140px;
width: 140px;
position: fixed;
top: 0px;
left: 0px;
display: flex;
align-items: center;
justify-content: center;
}

.icon {
height: 110px;
width: 110px;
user-select: none;
}

```

On définit la taille du logo ainsi que son div englobant, ce dernier doit être un peu plus grand pour permettre une intégration plus jolie des animations. On ajoute `display: flex`, `align-items: center` et `justify-content: center` pour que l'icône soit au centre du div (horizontalement et verticalement).

La partie la plus intéressante de la page est la mosaïque. En effet, pour la réaliser, on utilise une propriété de `display : grid`. Celle-ci permet de diviser la page en plusieurs lignes et colonnes pour placer des divs (enfants) par rapport à celles-ci.

La div parents est “.menu-accueil”, on lui applique donc :

```

.menu-accueil {
display: grid;
grid-template-columns: [centre-gauche] 49.5% [centre-droit] 1% [fin]
49.5%;
grid-template-rows: [unimg] 368px 17px [deuximg] 353px 17px [troisimg]
403px 17px [quatreimg] 389px 17px [cinqimg] 369px;
margin-bottom: 5%;
}

```

Les lignes sont réalisées en pourcentage pour permettre à la page de se réduire correctement quand un utilisateur à un écran plus petit (bien que le rendu final ne sera pas responsive). Pour ce qui est des lignes, on les fait en sorte qu'elles aient la taille de chaque image, avec un écart entre de 17px.

Les divs enfants auront donc (exemple pour la première image) :

```
height: 735.82px;  
width: 548.9px;  
grid-column-start: centre-gauche;  
justify-self: end;
```

Chaque image se présente comme suit :

```
<div class="img-refuge-cv">  
      
    <div class="txtRefugeCV">  
        <div class="fleche-2">  
            <svg xmlns="http://www.w3.org/2000/svg" width="30.411"  
height="50.561" viewBox="0 0 30.411 50.561">  
                <line id="Ligne_1" data-name="Ligne 1" x2="24.685" y2="25.302"  
transform="translate(2.863 2.793)" fill="none" stroke="#fff"  
stroke-width="8" />  
                <line id="Ligne_2" data-name="Ligne 2" x1="24.685" y1="25.302"  
transform="translate(2.863 22.466)" fill="none" stroke="#fff"  
stroke-width="8" />  
            </svg>  
        </div>  
        <span>Le Refuge du CV</span>  
    </div>  
</div>
```

Une div englobante, une image, le texte qui est contenu dans cette image ainsi qu'une flèche en svg (cette dernière sera utile pour l'animation, nous le verrons lors de la dernière partie). On place donc notre texte par rapport au div parent: img-refuge-cv (dans ce cas ci).

```
.txtRefugeCV {  
    position: relative;  
    bottom: 22%;  
    left: -40px;  
    user-select: none;  
    display: flex;  
    ...  
}
```

Le display flex sert à placer le texte et la flèche comme on le souhaite et éviter un retour à la ligne.

On obtient donc ceci:



Enfin, nous plaçons le footer, qui est commun à toutes les pages (nous le verrons donc qu'ici) :

```
<div class="footer">
    <div class="text-gauche">
        <span>Chalet du Club <br> Vosgien </span> <span> Lemberg.</span>
        <span>Tous droits réservés</span>
    </div>
    <div class="reseau-sociaux">
        <p>Réseaux Sociaux</p>

        <div class="logo-reseaux">
            <svg id="Capa_1" enable-background="new 0 0 512 512" height="50"
viewBox="0 0 512 512" width="50" xmlns="http://www.w3.org/2000/svg">
                ...
            </svg>
        </div>
    </div>
</div>
```

```

<svg height="50" viewBox="-21 -81 681.33464 681" width="50"
xmlns="http://www.w3.org/2000/svg">
    ...
</svg>
<svg height="50" viewBox="0 0 511 511.9" width="50"
xmlns="http://www.w3.org/2000/svg">
    ...
</svg>
</div>
</div>
<div class="text-droit">
    <p>Ecrit par Philippe</p>
    <p>Réalisé par son fils, Clément</p>
</div>
</div>

```

On divise notre footer en trois parties, pour contenir, Le texte du Club Vosgien (finalement le remplacement du logo), les réseaux sociaux, et le texte de gauche. Pour organiser ceci en css on ajoute à nouveau `display: flex` :

```

.footer {
    height: 200px;
    width: 100%;
    background-color: #221914;
    display: flex;
    justify-content: space-around;
}

```

On ajoute `justify-content: space-around` pour disposer les div enfants horizontalement. On les définit avec une largeur de 33.3% pour remplir la largeur de la page.

```

.text-gauche {
    color: #EFEADF;
    height: 100%;
    width: 33.3%;
    font-family: 'Poppins', sans-serif;
    font-weight: 500;
    font-size: 25px;
    padding-left: 5%;
}

```

```
padding-top: 3.5%;  
border-right: 2px #2B2019 solid;  
}
```

On obtient donc le résultat suivant (qui diffère par rapport à la maquette en terme de couleur mais pas de placement):



## Refuge du Chalet du Club Vosgien :

Sur cette page comme sur toutes les autres, on commence par intégrer le titre, qu'on place dans une div

```
<div class="titre">  
    <span>Le Refuge du Chalet du <br> Club Vosgien Lemberg</span>  
</div>
```

Comme les autres titres, on lui applique un rectangle double, un ombrage et quelques propriétés de placement horizontal et vertical:

```
.titre {  
    border-style: double;  
    border-width: 15px;  
    width: 90%;  
    height: 100%;  
    margin-top: 5%;  
    margin-bottom: 12%;  
    padding: 2%;  
    text-align: center;  
    align-self: center;  
    box-shadow: rgba(0, 0, 0, 0.10) 0px 0px 99px;  
}
```

Viens ensuite un texte d'introduction (que nous ne verrons pas ici étant donnée sa simplicité d'intégration).

Sur la page Refuge du Chalet du Club Vosgien, il y a un placement particulier des images et du texte, le balisage est le suivant: on définit une div principale puis des divs enfants qui vont être: img-gauche, texte-droit, img-droit, texte-gauche :

```
<div class="img-gauche-droite">
    <div class="img-gauche">
        
    </div>

    <div class="texte-droit">
        <svg xmlns="http://www.w3.org/2000/svg" width="361.282" height="6"
viewBox="0 0 361.282 6">
            <path id="Tracé_1" data-name="Tracé 1" d="M0,0H361.282V6H0Z" />
        </svg>
        <p class="texte-droit-p-1">Le Chalet est fait sur trois étages
:</p>

        <p class="texte-droit-p-2">grande salle</p>
        <p class="texte-droit-p-3">Les dortoirs</p>
        <p class="texte-droit-p-4">La buvette</p>

        <svg xmlns="http://www.w3.org/2000/svg" width="361.282" height="6"
viewBox="0 0 361.282 6">
            <path id="Tracé_2" data-name="Tracé 1" d="M0,0H361.282V6H0Z" />
        </svg>
    </div>

    <div class="texte-gauche">
        <div class="ligne-droite-1">
            <svg xmlns="http://www.w3.org/2000/svg" width="361.282"
height="6" viewBox="0 0 361.282 6">
                <path id="Tracé_3" data-name="Tracé 3" d="M0,0H361.282V6H0Z" />
            </svg>
        </div>
        <div class="span-gauche">
            <span>Un abri pour faire des grillades <br> et des pizzas </span>
            <span>est à disposition</span>
            <p>des locataires</p>
        </div>
    </div>

```

```

<div class="ligne-droite-2">
    <svg xmlns="http://www.w3.org/2000/svg" width="361.282"
height="6" viewBox="0 0 361.282 6">
        <path id="Tracé_4" data-name="Tracé 3" d="M0,0H361.282V6H0Z" />
    </svg>
</div>

</div>

<div class="img-droite">
    
</div>
</div>

```

Puis on place les images et le texte à l'aide de `display: flex`. On définit les largeurs de 50%, ainsi, il peut y avoir deux divs par ligne (img-gauche/texte-droit pour la première et img-droit/texte-gauche pour la deuxième). Pour obliger un retour à la ligne on utilise la propriété `flex-wrap: wrap`

```

.img-gauche-droite {
    display: flex;
    flex-direction: row;
    flex-wrap: wrap;
    align-items: center;
    margin-left: 5%;
    margin-bottom: 3%;
}

```

Nous obtenons:



Suite à cela, nous avons une suite classique, avec du texte et des images alignés au centre. Il y a deux API sur cette page, Maps et Google Calendar, nous les verrons au dernier chapitre. Pour les intégrer il suffit de créer une div vide, de la largeur et hauteur que nous voulons.

Après l'intégration de l'API Google Calendar, nous avons le formulaire. Pour ajouter celui-ci, on commence par le créer en html avec la balise form:

```
<div class="input-text">
    <input type="text" name="" value="" placeholder="Nom">
    <input type="text" name="" value="" placeholder="Prénom">
    <input type="text" name="" value="" placeholder="E-Mail">
    <input type="text" name="" value="" placeholder="Confirmation de
l'E-mail">
    <textarea name="" rows="8" cols="80" placeholder="Message"></textarea>
</div>
```

On ne définit pas de name pour le moment car ce n'est pas nécessaire. Cela viendra lors de la mise en ligne et de l'utilisation d'un langage de programmation backend pour récupérer les valeurs. On utilise placeholder pour que l'utilisateur puisse voir où placer les informations, ce texte grisé disparaît lorsqu'on écrit dans une case.

Pour mettre en forme ce formulaire, on commence par mettre tous les input en colonne pour enfin les aligner au centre.

```
.input-text {
    display: flex;
    flex-direction: column;
    align-items: center;
}
```

Enfin, on met du style sur chaque input ou textarea à l'aide de la ligne suivante :

```
.input-text>input, .input-text>textarea {
    font-family: 'Poppins', sans-serif;
    font-weight: 300;
    width: 40%;
    border: 0;
    border-bottom: 1.5px solid black;
    outline: 0;
    font-size: 20px;
```

```
padding: 7px 0;  
background: transparent;  
margin-bottom: 2%  
}
```

On ne met une bordure qu'en dessous, puis on définit la taille de la police, l'écart de la bordure de gauche, ...

Enfin, pour conclure la page, on ajoute le footer vu sur la première page.

## Découvrir Lemberg :

Comme les autres pages, il faut tout d'abord intégrer le titre. Le css ne change pas de la partie précédente.

```
<div class="titre">  
    <span>Découvrir Lemberg</span>  
</div>
```

Le but de la page est de proposer les différentes randonnées à l'utilisateur, pour ce faire, il a d'abord fallu créer la carte Maps, qui contient les différents points faisant référence aux articles de la page. On déclare une div vide qu'on utilisera dans le javascript:

```
<div id="MapsOpen">  
</div>
```

Tout d'abord, il est nécessaire de créer un div parent nommé "text" permettant d'organiser tous les articles entre eux. On utilise à nouveau la propriété `display: grid` avec des valeurs en pourcentage pour être plus responsives.

```
.text {  
    height: auto;  
    width: 100%;  
    display: grid;  
    grid-template-columns: 10% 40% 40% 10%;  
}
```

Chaque article se présente de la manière suivante :

```

<div class="source-SH">
    <div class="titre-SH">
        <p id="source-lien">01
            <svg xmlns="http://www.w3.org/2000/svg" width="5.52"
height="47.82" viewBox="0 0 5.52 47.82">
                <rect id="Rectangle_23" data-name="Rectangle 23"
width="47.82" height="5.52" transform="translate(5.52) rotate(90)" />
            </svg>
            La source Saint Hubert
        </p>
    </div>
    <p>...</p>
    <p>Descriptif: ...</p>
    <p>....</p>
    <p>...</p>
    <div class="img-SH">
        
    </div>
</div>

```

On crée une div parent du nom de l'article que l'on veut faire, puis trois divs sous-jacentes contenant respectivement, le titre, la description et l'image (s'il y en a une). On organise l'article au niveau du css de la manière suivante:

```

.source-SH {
    grid-column-start: 2;
    grid-column-end: 4;
    height: auto;
    width: auto;
}

```

On définit où commence l'article et où il se finit.

```

.titre-SH {
    font-family: 'Poppins', sans-serif;
    font-weight: 700;
    font-size: 55px;
    text-shadow: 0px 0px 9px rgba(0, 0, 0, 0.20);
    text-align: left;
    margin-bottom: 2%;
    transform: translateY(20px);
}

```

```
    opacity: 0;  
}
```

On donne les spécificités pour le titre, sa taille, son alignement, etc... (`opacity: 0` et `transform: translateY(20px)` serviront dans la dernière partie: les animations).

```
.source-SH>p:nth-of-type(0n + 1),  
.source-SH>p:nth-of-type(0n + 3) {  
    font-family: 'Poppins', sans-serif;  
    font-weight: 300;  
    font-size: 20px;  
    margin-bottom: 3%;  
    text-shadow: 0px 0px 9px rgba(0, 0, 0, 0.20);  
    text-align: left;  
}
```

Dans cette partie, on alterne droite et gauche pour la position du texte (ci-dessus à gauche), la partie suivante permet de faire l'inverse (ci-dessous à droite):

```
.source-SH>p:nth-of-type(0n + 2),  
.source-SH>p:nth-of-type(0n + 4){  
    font-family: 'Poppins', sans-serif;  
    font-weight: 300;  
    font-size: 20px;  
    margin-bottom: 3%;  
    text-shadow: 0px 0px 9px rgba(0, 0, 0, 0.20);  
    text-align: right;  
}
```

Un équivalent moins long pour sélectionner les balises p auraient été  
`.source-SH>p:nth-of-type(2n + 0)` pour les textes pairs et  
`.source-SH>p:nth-of-type(2n + 1)` pour les textes impairs

Les articles continuent, à peu de chose près, jusqu'à la fin de la page et du footer.

On obtient donc ceci :

## 01 | La source Saint Hubert

Accès : Depuis le chalet du Club Vosgien. Descendre les escaliers et suivre le chevalet jaune.  
Traverser la route de Bitche et continuer sur le large chemin.  
Suivez le chevalet jaune jusqu'à la source Saint Hubert. Revenez sur vos pas pour le retour..  
Compter une bonne heure pour l'aller retour..

Descriptif: Le rocher sculpté date de l'époque gallo-romaine.  
Les historiens s'accordent à dire qu'il remonterait au IIIème siècle de notre ère.  
Initialement appelé « Bomphosen Brunnen », soit « Fontaine aux culottes bouffantes »  
le nom a été francisé que tardivement. Saint Hubert, patron des chasseurs,  
a donné son nom à tort à cette sculpture en l'interprétant comme une scène de chasse.

On y identifie malheureusement plus que la partie basse de deux divinités romaine :  
Sylvain, dieux de la forêt avec un bâton et Diane, déesse de la chasse, avec son arc.  
On y identifie également assez facilement un combat de cerfs,  
des sangliers ou un auroch, bœuf sauvage disparut au moyen âge.

A quelques mètres,  
un lavoir récemment restauré servait aux villageoises pour laver leur linge.  
Un personnage se baignant les pieds sur la sculpture pourrait attribuer  
des vertus curatives à cette source.



## Randonner à Lemberg :

Sur cette page, on retrouve, comme la page précédente, un schéma qui se répète: du texte avec des images. Notre page est donc divisée en plusieurs divs: certains contenu de texte, d'autre, d'image. La div principale, est la div “page”, nous permettant à nouveau d'organiser au mieux notre page à l'aide de la propriété **display: flex**.

```
.page {  
  display: flex;  
  flex-direction: column;  
  text-align: center;}
```

On continue en définissant le titre, comme les pages précédentes.

Puis chaque image et texte sont développés comme suit :

```
<div class="premier-texte">
    <p>Le Club Vosgien de Lemberg balise</p>
    <p>et entretient près de 180 kilomètres de sentiers. Afin de les
rendre accessibles au plus grand nombre</p>
    <p>d'amoureux de la nature,</p>
    <p>tous les chemins sont balisés avec les signes</p>
    <p>conventionnels de la Fédération du Club Vosgien:</p>
</div>

<div class="img-signe-cv">
    
</div>
```

Ainsi, on utilise des balises `<p>` pour que le texte soit suivi d'un retour à la ligne.

On associe ces divs au css suivant:

```
.premier-texte {
    font-family: 'Poppins', sans-serif;
    font-weight: 500;
    font-size: 20px;
    margin-bottom: 2%;
    text-shadow: 0px 0px 9px rgba(0, 0, 0, 0.20);
}

.img-signe-cv {
    margin-bottom: 2%;
```

On définit un `text-shadow`, la police ainsi que toutes les propriétés d'un texte. Lorsqu'il est nécessaire, on redéfinit la taille des images. On obtient le résultat suivant :



# Randonner à Lemberg

Le Club Vosgien de Lemberg balise et entretient près de 180 kilomètres de sentiers. Afin de les rendre accessibles au plus grand nombre d'amoureux de la nature, tous les chemins sont balisés avec les signes conventionnels de la Fédération du Club Vosgien:

Nomenclature	Les signes du Club Vosgien® Le signe officiellement utilisé par le Club Vosgien			
	rouge	bleu	jaune	vert
Nordique (1)	■	■	■	■
Lorraine (2)	◆	◆	◆	◆
Alsacien (3)	□	□	□	□
Triangulaire (4)	▲	▲	▲	▲
Croix (5)	+	+	+	+
Croix brisée (6)	✗	✗	✗	✗
Drapeau (7)	●	●	●	●
Américain (8)	○	○	○	○



Le Club Vosgien de Lemberg balise et entretient près de 180 kilomètres de sentiers. Afin de les rendre accessibles au plus grand nombre d'amoureux de la nature, tous les chemins sont balisés avec les signes conventionnels de la Fédération du Club Vosgien:

Nomenclature	Les signes du Club Vosgien® Le signe officiellement utilisé par le Club Vosgien			
	rouge	bleu	jaune	vert
Nordique (1)	■	■	■	■
Lorraine (2)	◆	◆	◆	◆
Alsacien (3)	□	□	□	□
Triangulaire (4)	▲	▲	▲	▲
Croix (5)	+	+	+	+
Croix brisée (6)	✗	✗	✗	✗
Drapeau (7)	●	●	●	●
Américain (8)	○	○	○	○

Le Club Vosgien a délibérément choisi de donner à la croix un signe en forme de croix brisée. Cela nous permet de distinguer nos signes de ceux utilisés par d'autres clubs ou associations. Ces signes doivent être facilement reconnaissables et éviter de donner l'impression que le Club Vosgien est une organisation militaire.

Vous pouvez découvrir en toute sérénité le fabuleux patrimoine naturel et historique qui berce le village de Lemberg et ses environs. Que ce soit en sous-bois, dans les vallées où coulent de nombreux ruisseaux, près des sculptures gallo-romaines ou près des multiples lavoirs, toute la famille pourra venir se ressourcer dans ce beau village des Vosges du Nord.

En tout état de cause, pour préparer votre randonnée, favorisez toujours les ouvrages portant le logo du club vosgien :



On finit à nouveau la page avec le footer.

## Liens Amis / Contact :

Comme indiqué dans la partie Identité Graphique, cette page nous sert à pouvoir faire le pont entre ce site et les différents sites “amis” du Club Vosgien. De plus, elle comprend aussi la partie Contact permettant de contacter l’administrateur du site.

Comme les autres pages, on définit le titre, puis un sous titre “Liens Amis” et enfin une ligne d’introduction. Ceci fait, il faut intégrer les différents liens :

```
<div class="liens-amis">
    <p>Vous trouverez ci-dessous des liens vers des sites amis du Club
Vosgien de Lemberg:</p>
    <br>
    <br>
    <a href="https://www.tourisme-paysdebitche.fr/" class="lien">L'office
du tourisme du Pays de Bitche</a>
    <br>
    <br>
    <a href="http://www.clubvosgienbitche.fr/" class="lien">Le Club
Vosgien de Bitche</a>
    <br>
    <br>
    <a href="https://souchtclubvosgien.wordpress.com/" class="lien">Le
Club Vosgien de Soucht</a>
    <br>
    <br>
    <a href="http://www.skiclubmasevaux.fr/ski-club-masevaux-historique/"
class="lien">Le ski club vosgien de Masevaux</a>
</div>
```

On utilise simplement des balises  qui permettent de créer un lien cliquable en html.

On définit quelques propriétés css à ces liens :

```
.liens-amis>a {
    font-family: 'Poppins', sans-serif;
    font-weight: 300;
    font-size: 20px;
    text-shadow: rgba(0, 0, 0, 0.1) 0px 0px 99px;
    outline: none;
    text-decoration: none;
    padding: 2px 1px 0;
    color: #0E5E26;}
```

A l'aide de text-decoration: none, on évite de laisser la barre horizontale en dessous, car celle-ci sera créée avec une animation

Nous avons donc ceci comme résultat (le premier lien est souligné car c'est le résultat d'effet d'animation qui va être appliqué) :



Suite à cela, on intègre le sous-titre suivant : “Contact”. Cette partie contient un formulaire pour permettre à l'utilisateur de transmettre un message.

# Contact

Quelle que soit votre question (location du chalet,

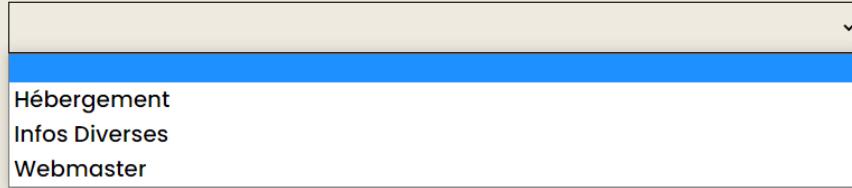
horaires d'ouverture, état des sentiers, programme des sorties...),

vous pouvez nous contacter en remplissant le formulaire ci-dessous :

Vos commentaires ou questions sont bienvenues.

\* indique un champ requis

Sélectionnez un contact:\*



A screenshot of a dropdown menu. The menu has a white background and a black border. It contains three items: "Hébergement", "Infos Diverses", and "Webmaster". The first item, "Hébergement", is highlighted with a blue background and white text. There is a small downward arrow icon at the top right corner of the menu.

On commence par laisser choisir à l'utilisateur quel domaine cibler. Pour ce faire on utilise une balise select avec des option :

```
<div class="input-contact">
  <select name="contact" id="contact-select">
    <option value=""></option>
    <option value="hebergement">Hébergement</option>
    <option value="infosdiv">Infos Diverses</option>
    <option value="webmasterr">Webmaster</option>
  </select>
</div>
```

Dans la partie css, on redéfinit les bordures, la taille du texte, la police, ...

```

.input-contact > select {
    font-family: 'Poppins', sans-serif;
    font-weight: 300;
    width: 40%;
    border: 1px solid black;
    outline: 0;
    font-size: 20px;
    color: black;
    padding: 7px 0;
    background: transparent;
    margin-bottom: 2%;
}

```

On ajoute un background transparent pour éviter qu'il soit blanc et n'aille pas avec la couleur de la page. `outline: 0;` sert à éviter de faire une bordure lorsque l'utilisateur sélectionne une valeur.

Il faut maintenant réaliser le form, qui va servir à pouvoir laisser entrer des valeurs par l'utilisateur.

```

<div class="input-text">
    <input type="text" name="" value="" placeholder="Nom">
    <input type="text" name="" value="" placeholder="Prénom">
    <input type="text" name="" value="" placeholder="E-Mail">
    <input type="text" name="" value="" placeholder="Confirmation de
    l'E-mail">
        <textarea name="name" rows="8" cols="80"
placeholder="Message"></textarea>
</div>

```

(On remplira les names quand on utilisera un langage back-end, type PHP). On place les éléments de input-text :

```

.input-text {
    display: flex;
    flex-direction: column;
    align-items: center;
}

```

Puis enfin , il faut styliser les 4 inputs ainsi que le textarea :

```
.input-text > input, .input-text > textarea {  
    font-family: 'Poppins', sans-serif;  
    font-weight: 300;  
    width: 40%;  
    border-bottom: 1.5px solid black;  
    outline: 0;  
    font-size: 20px;  
    color: black;  
    padding: 7px 0;  
    background: transparent;  
    margin-bottom: 2%  
}
```

Comme pour la balise select, on remet des bordures plus petites, on augmente la taille, on met un background transparent, ceci après avoir fixé la hauteur et la largeur. On obtient alors le résultat suivant :

The image shows a web form with a light beige background. At the top right, there is a note: "\* indique un champ requis". Below it, a dropdown menu is set to "Infos Diverses". The form contains five input fields: "Nom" (Name), "Prénom" (First Name), "E-Mail" (Email), "Confirmation de l'E-mail" (Email Confirmation), and "Message" (Message). Each input field has a horizontal line underneath it. The "Message" field is a large text area with a horizontal scroll bar at the bottom right.

Il nous reste à ajouter un dernier texte d'information et le footer pour clôturer cette page.

## Sorties du Club Vosgien:

“Sorties du Club Vosgien”, n'étant pas réalisées sur la maquette, nous la détaillerons ici. Cette page a été pensée comme étant une suite d'articles, comme pour la page originale. J'ai donc choisi d'ajouter un scroll vertical (que nous verrons dans la dernière partie) pour les articles. La page finale ressemble donc à ceci :

Sorties du Club Vosgien  
Lemberg

Retrouvez, ici, toutes les actualités et sorties du Club Vosgien.

**01 | Sortie du mercredi 7 Octobre**

Le mercredi 7 octobre le CV Lemberg vous propose une sortie chez nos voisins allemands, du coté de Pirmasens. (n'oubliez pas le masque !)

Départ de L'EICHELBACHERMUHLE – KALBERKLAMM – VINNINGEN Repas à la

**02 | Sortie du dimanche 6 septembre**

Le dimanche 6 septembre le CV Lemberg vous propose une sortie au Grand Wintersberg

DÉPART au chalet de Lemberg à 9h, à bitche à 9h15 et à 10h à la maison forestière du Riesthal.

Guide: Antoine Piccin

Pour réaliser cette page en html et css, il faut diviser la page en plusieurs div. La première contenant le titre et le sous-titre, la suivante contenant les articles et enfin une pour chaque article.

On commence donc à définir notre première div :

```
<div class="introduction">  
  
<div class="titre-sortie">  
  <div class="titre-parallax-1">  
    <span>Sorties du Club Vosgien</span>
```

```

</div>
<br>
<div class="titre-parallax-2">
    <span>Lemberg</span>
</div>

</div>
<div class="text-explication">
    <div class="parallax-1">
        <span>Retrouvez, ici, toutes les</span>
<span>actualités</span><span> et </span>
    </div>
    <br>
    <div class="parallax-2">
        <span>sorties</span><span> du Club Vosgien.</span>
    </div>

    </div>
</div>

```

On divise en `<span>` pour pouvoir diviser les phrases en mot et au final pour les styliser, certains seront en vert, d'autre en noir.

```

.introduction {
    height: 50vh;
    width: 100vw;
    display: flex;
    align-items: center;
    background-color: #EFEADF;
}

```

On définit une hauteur de 50 vh pour que le titre et le sous-titre fassent un demi écran de haut. Un `display: flex` permet d'aligner correctement les divs enfant dans le div parent.

```

.titre-sortie {
    text-align: right;
    height: auto;
    width: 50vw;
    padding-right: 5%;
}

```

On définit la largeur de la div titre-sortie à 50vw pour qu'il prenne la moitié de la largeur d'un écran. Ensuite, on aligne à droite le texte et enfin, on applique un petit décalage sur la gauche avec `padding-right: 5%` pour que le texte ne soit pas attaché à la bordure.

Il faut maintenant définir la taille du texte, la couleur, la police, ...

```
.titre-parallax-1>span {  
    color: #221914;  
    font-family: 'Poppins', sans-serif;  
    font-weight: 500;  
    font-size: 60px;  
    text-align: center;  
}  
  
.titre-parallax-2>span {  
    color: #0E5E26;  
    font-family: 'Poppins', sans-serif;  
    font-weight: 500;  
    font-size: 60px;  
    text-align: center;  
    line-height: 0.5;  
}
```

De même pour la le texte d'explication :

```
.parallax-1>span, .parallax-2>span {  
    color: #221914;  
    font-family: 'Poppins', sans-serif;  
    font-weight: 300;  
    font-size: 30px;  
    text-align: center;  
    line-height: 1.2;  
}
```

Enfin, on change la couleur verte de certains mots

```
.parallax-1>span:nth-of-type(2), .parallax-2>span:nth-of-type(1) {  
    color: #0E5E26;  
}
```

Passons maintenant à la div contenant les articles:

Celle-ci se nommera horizontal-box. Nous ajoutons les articles dedans par ordre décroissant (l'article le plus ancien est nommé page-1). Nous faisons ceci de sorte qu'une personne voulant reprendre le code plus tard pour ajouter des articles, n'ai pas à changer tous les numéros de page. Il suffira d'ajouter 1 au numéro du précédent article. Pour ressembler plus à un article, le style de police a été changé par "[Old Standard TT](#)". Nous obtenons donc ceci :

```
<div class="horizontal-box">

<div class="page-5">

    <div class="title-5">
        <p>01 | Sortie du </p>
        <p>mercredi 7 Octobre</p>
    </div>

    <div class="sujet-5">
        <span>Le mercredi 7 octobre le CV Lemberg vous propose une sortie chez nos voisins allemands, du coté de Pirmasens. (n'oubliez pas le masque !)</span><br><br>
        <span>Départ de L'EICHELBACHERMUHLE - KALBERKLAMM - VINNINGEN Repas à la STOCKWALDHUTTE WINZEL .
        retour par HEXENKLAMM - GERSBACH. </span> <br> <span>DURÉE DE LA RANDONNÉE </span><span> : 2h30 le matin et 1h30 l'après midi.</span>
        <br><br>
        <span>DÉPART</span><span> du Chalet du club vosgien de Lemberg à 9H où 9H15 PARKING DU SUPERMARCHE MATCH A BITCHE.</span>
    </div>

</div>
...
```

Dans la div page-5, on ajoute deux div, une pour le titre, une autre pour le sujet. Cela permet de compartimenter les articles et de pouvoir les organiser par rapport à la div parent: page-5.

On commence par styliser la div parent horizontal-box:

```
.horizontal-box {  
    height: 100vh;  
    width: 250vw;  
    background-color: #221914;  
    display: flex;  
    z-index: 0;  
}
```

Sachant qu'un article fait un demi écran, on définit une largeur pouvant contenir les 5 (il faudra donc augmenter la largeur quand on ajoutera des articles). On applique une largeur de 250vw, c'est-à-dire 5 demi-écran. Nous définissons ici largeur et non la hauteur car c'est dans ce sens que nous irons pour scroller les articles. On définit une hauteur de 100vh pour que celle-ci fasse un écran de haut.

Le style d'un article sera donc :

```
.page-5, .page-4, .page-3, .page-2, .page-1 {  
    height: 100vh;  
    width: 50vw;  
    border-right: 2px #2B2019 solid;  
    color: #EFEADF;  
    text-align: center;  
}
```

Pour la page entière. On aligne le texte au centre, on définit une bordure sur la droite pour délimiter les articles puis on définit une hauteur de 100vh et une largeur de 50vw (comme dit précédemment, un demi écran de large).

Le style du titre et du texte sera donc :

```
.title-5{  
    padding-top: 4%;  
    font-family: 'Old Standard TT', serif;  
    font-weight: 700;  
    font-size: 60px;  
    text-align: center;  
    text-shadow: 0px 0px 20px rgba(0, 0, 0, 0.10);}
```

Avec une plus grande taille de texte, on augmente aussi le font-weight à 700. Pour ce qui est du texte “.sujet” nous avons :

```
.sujet-5{  
    height: 90%;  
    font-family: 'Old Standard TT', serif;  
    font-weight: 400;  
    font-size: 20px;  
    text-align: left;  
    padding: 5%;  
    line-height: 3;  
    text-shadow: 0px 0px 20px rgba(0, 0, 0, 0.10);  
}
```

On augmente l'écart entre les lignes pour que cela soit plus aéré. On définit une hauteur de 90% qui est le restant de la hauteur du div page, sans le titre.

Enfin, on change quelques mots ou parties de phrases comme suit:

```
.title-5>p:last-of-type{  
    color: #0E5E26;  
}  
.sujet-5>span:nth-of-type(3), .sujet-5>span:nth-of-type(4){  
    font-style: italic;  
}  
  
.sujet-5>span:nth-of-type(3), .sujet-5>span:nth-of-type(5){  
    color: #0E5E26;  
}
```

On ajoute le logo comme pour les autres pages (cette fois-ci en bas à gauche de la page) :

```
.logo {  
    height: 140px;  
    width: 140px;  
  
    position: fixed;  
    top: 85vh;  
    left: 0%;  
  
    display: flex;  
    align-items: center;  
    justify-content: center;
```

```

cursor: pointer;
opacity: 0;

z-index: 1;
}

.icon {
  height: 110px;
  width: 110px;
  user-select: none;
}

```

Voici ce que nous obtenons (pour l'article 3 et 4 par exemple) :



Nous en avons donc finis avec le html et le css du site, pour accorder toutes les pages entre elles et ajouter de l'interactivité, passons à la partie Animation.

### 3 - Ajout des animations et API : Javascript et GSAP

Pour traiter des différentes animations réalisées sur ce site, nous allons les présenter par ordre de page. Nous suivrons le même schéma que les parties précédentes. Pour voir les différentes animations, une vidéo du site est jointe à ce fichier dans le dossier.

Nous allons, majoritairement, utiliser la librairie [GSAP](#) par GreenSock pour les animations. Cette librairie est présente sur de nombreux sites et est très utilisée en entreprise. Pour intégrer cette librairie à notre projet, nous utiliserons un CDN. Pour travailler sur un projet en entreprise, l'utilisation de NodeJs avec NPM est bien plus courante, cependant, le dossier devient vite lourd avec de nombreux fichiers. Il convient donc mieux, dans notre cas, d'utiliser un CDN. Ce dernier est un lien que l'on ajoute à la fin ou au début de notre page html et qui va charger la librairie via internet:

```
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.6.0/gsap.min.js"></script>
```

Nous utiliserons une méthode de GSAP qui est ScrollTrigger, pour ce faire nous devons aussi ajouter le CDN correspondant :

```
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.6.0/ScrollTrigger.min.js  
></script>
```

#### Animation commune à toutes les pages :

Sur les différentes pages, plusieurs animations sont intégrées. Pour réaliser ces dernières, nous utiliserons gsap.to. Cette méthode permet de créer des animations via des attributs css. On utilise gsap.to de la manière suivante:

```
gsap.to(".maClasse", {mes différents attributs css});
```

Ainsi, la méthode suivante sert à faire déplacer la div ayant la classe retour horizontalement (sur l'axe x) à -100%.

```
gsap.to(".retour", {  
    xPercent: -100,  
    duration: 1  
});
```

## Transition entre les pages:

Pour une meilleure expérience utilisateur, on réalise une transition entre les différentes pages. Pour ce faire, on crée une div avec la classe “changement”. A chaque passage d'une page à l'autre on va ajouter un petit délai et déplacer cette div au milieu de l'écran (celle-ci étant à droite hors de l'écran). On utilise une propriété de gsap.to qui est onComplete. On commence donc par déplacer la div au centre, lorsque c'est fait, on affiche le logo et enfin, on change de page :

```
function chgtPage(location) {
  gsap.to(".changement", {
    xPercent: -100,
    duration: 1,
    ease: "slow(0.7, 0.7, false)",
    onComplete: () => {
      gsap.to(".logo-pat", {
        opacity: 1,
        duration: 2,
        onComplete: () => {
          document.location.href = location;
          gsap.to(".logo-pat", {opacity: 0});
          gsap.to(".changement", {x: 0, delay: 2});
          gsap.to(".explication-accueil>span", {opacity: 0});
        }
      });
    }
  });
}
```

On réalise la même chose lors de l'arrivée sur une page. Un div est au centre de l'écran, lorsqu'on arrive sur la page, on déplace cette div sur la gauche :

```
gsap.to(".changement-retour", {
  xPercent: -100,
  duration: 1,
  ease: "slow(0.7, 0.7, false)"
});
```

### Animation du logo:

Une autre animation commune aux différentes pages est l'animation du logo pour le retour à la page d'accueil. Comme pour les éléments de la mosaïque de la page Index.html, on récupère la div parent du logo puis on ajoute un événement mouseover et mouseout. Dans ces deux évènements, on fait vriller le logo de 360° sur la droite avec un ease “elastic.out(1.2, 0.4)” puis on fait apparaître le texte en dessous du logo :

```
var logo = document.querySelector('.logo');

logo.addEventListener('mouseover', ev => {
  gsap.to(".logo", {
    rotation: 360,
    ease: "elastic.out(1.2, 0.4)",
    duration: 3
});

  gsap.to(".retour-accueil", {
    y: 30,
    ease: "power1.in",
    duration: 0.2,
    opacity: 1
});
});

});
```

Lors de la sortie de la souris du logo on remet les valeurs à 0, qui est la valeur de base:

```
logo.addEventListener('mouseout', ev => {
  gsap.to(".logo", {
    rotation: 0,
    ease: "elastic.out(1.2, 0.4)",
    duration: 3
});

  gsap.to(".retour-accueil", {
    y: 0,
    ease: "power1.in",
    duration: 0.2,
    opacity: 0
});
});
```

## Animation des titres :

Cette animation sert à faire apparaître un titre lorsque l'utilisateur arrive au niveau celui-ci. Nous allons prendre le titre de la page d'accueil comme exemple. Pour ce faire, il faut utiliser la propriété ScrollTrigger (que l'on ajoute au javascript avec `gsap.registerPlugin(ScrollTrigger);`). Celle-ci sert à délimiter un début et une fin d'animation en fonction de l'endroit où est l'utilisateur sur la page (en fonction du scroll horizontal). On ajoute donc scrollTrigger à notre gsap.to avec, comme attributs possibles:

-Trigger: sélectionne la div où on veut "attacher" une animation (pour le début)

-endTrigger: sélectionne la div où on veut "attacher" une animation (pour la fin)

-start: qui va définir à quel moment commence l'animation (un emplacement sur l'écran et un autre par rapport à la div)

-end: comme pour start, mais pour mettre fin à l'animation

-scrub: qui permet de lier l'animation au scroll (si on remonte, l'animation va en arrière)

-pin: permet de fixer une div à l'écran lorsque l'utilisateur scroll comme utiliser position: fixed juste le temps de l'animation.

-toggleActions: permet de définir comment doit réagir l'animation quand on la quitte, puis quand on repasse dessus par le bas...(restart sera utilisé dans ce cas car cela permet de relancer l'animation lorsque l'on revient par le dessus).

On veut donc que, lorsque l'utilisateur scroll, l'animation se lance au moment d'arriver sur le titre. Puis restart lorsqu'il revient par le dessus. A l'aide de ScrollTrigger on a :

```
gsap.to(".titre-menu", {  
    y: -10,  
    duration: 2,  
    opacity: 1,  
    scrollTrigger: {  
        trigger: ".titre-menu",  
        toggleActions: "restart"  
    }  
});
```

On définit donc dans notre css une opacité de 0 et on descend un peu la position du titre. Quand on lance l'animation, le titre apparaît en remontant un peu (de 10px exactement).

Cette animation est utilisée sur une grande partie des titres du site.

## L'index de la page :

La première animation a pour but de déplacer le titre en fonction de l'emplacement de la souris sur l'écran. Il faut donc commencer par récupérer l'emplacement de la souris par rapport à la div. On commence donc par la récupérer en javascript:

```
var doc = document.querySelector('.accueil');
```

Ensuite, il faut ajouter un addEventListener pour récupérer l'emplacement x et y:

```
doc.addEventListener('mousemove', (ev) => {  
});
```

La variable ev nous donne la valeur x et y de la souris sur la div à l'aide de l'attribut mousemove. On met ces x et y dans des variables, qu'on divise par un nombre plus ou moins grand pour ne pas avoir une trop grande amplitude :

```
var xpos = -(ev.pageX / 50);  
var ypos = -(ev.pageY / 50);
```

Enfin, on déplace les différentes div du titre avec gsap.to() :

```
gsap.to(".av1", {  
    x: xpos,  
    y: ypos  
});
```

On répète la même opération pour les différentes div du titre.

L'animation suivante sert à déplacer le texte sur les différents éléments de la mosaïque pour faire apparaître les flèches. Lorsque l'utilisateur a la souris sur la div d'une mosaïque, il faut que le texte se déplace sur la droite pour faire entrer la flèche. Ainsi, quand la souris est sur la div: mouseover, on utilise gsap.to sur le texte pour qu'il se déplace de 40 pixels sur la droite. On utilise ease: "circ.out". On réalise la même opération pour la flèche, qui elle, se déplacera de 20 pixels.

```
var divRefugeCV = document.querySelector('.img-refuge-cv');

divRefugeCV.addEventListener('mouseover', ev => {
    gsap.to(".txtRefugeCV", {
        duration: 1,
        x: 40,
        ease: "circ.out"
    });
    gsap.to(".fleche-2", {
        duration: 1,
        x: 20,
        ease: "circ.out"
    });
});
```

Le problème est que, lorsque l'utilisateur sort la souris de la div, il faut que l'animation revienne à la normale. Nous avons donc le même addEventListener que précédemment avec mouseout comme paramètre, puis on redéfinit x et y à 0 :

```
divRefugeCV.addEventListener('mouseout', ev => {
    gsap.to(".txtRefugeCV", {
        duration: 1,
        x: 0,
        ease: "circ.out"
    });
    gsap.to(".fleche-2", {
        duration: 1,
        x: 0,
        ease: "circ.out"
    });
});
```

Lors du clique, on dirige l'utilisateur vers la page correspondante (ici refuge-chalet.html)

```
divRefugeCV.addEventListener('click', ev => {
    chgtPage("refuge-chalet.html");
});
```

Une dernière animation pour l'index est le logo. En effet, celui-ci n'est pas le même que les autres pages. Ici, on souhaite le déplacer au centre de l'écran après avoir passé la page de couverture. On utilise alors la propriété Scroll Trigger, vu plus haut.

Ainsi, on a:

```
gsap.to(".logo", {
    y: 300,
    scrollTrigger: {
        trigger: ".logo",
        start: "top top",
        end: "top 33%",
        endTrigger: ".titre-menu",
        scrub: 1
    }
});
```

Puis on fait apparaître le texte en dessous de la même manière:

```
gsap.to(".explication-accueil>span", {
    duration: 1,
    opacity: 0.2,
    scrollTrigger: {
        trigger: ".titre-menu",
        start: "center bottom",
        scrub: 1
    }
});
```

## Liens Amis / Contact :

Sur cette page, une animation est en plus des autres pages: les liens. Ces différents liens sont réalisés en css et non en Javascript car ceux-ci sont liés à la propriété de la balise <a>. Pour réaliser ces animations, il faut commencer par créer la barre horizontale sous le lien (on ajoute donc `outline: none; text-decoration: none;` aux balises <a>) On utilise un attribut de classe css: `::after`, qui va ajouter des attributs à la suite des liens.

```
.lien::after{  
    content: '';  
    display: block;  
    height: 2px;  
    background: #0E5E26;  
    transition: 0.3s;  
}
```

Il nous suffit d'ajouter `content: '';` et `display: block` pour que la barre soit créée, on ajoute une hauteur de 2px, une couleur et enfin une transition pour permettre de faire l'animation. On ajoute une largeur de 0 `width: 0;` pour pouvoir l'agrandir lors du passage de la souris: `:hover`

```
.lien:hover::after{  
    width: 100%;  
}
```

On utilise `:hover::after` pour mettre cette animation (hover) à la fin de chaque lien. Ainsi, quand l'utilisateur passe la souris sur le lien, la barre prend 100% de largeur avec une transition.

## Sorties du Club Vosgien:

Sur cette page, deux animations ont été réalisées, la première est quasiment identique à celle qui agit sur le titre de l'accueil. C'est une animation qui permet de déplacer le texte sur les mouvements de la souris seulement sur l'axe horizontal (x). Comme pour l'animation du titre :

```
var divIntro = document.querySelector('.introduction');

divIntro.addEventListener('mousemove', (ev) => {

    var xpos = (Math.abs((screen.width / 2) - ev.pageX));

    xpos = xpos / 70;

    gsap.to(".parallax-1", {
        x: xpos
    });

    xpos = -(ev.pageX / 100);

    gsap.to(".parallax-2", {
        x: xpos
    });

    xpos = (ev.pageX / 50);

    gsap.to(".titre-parallax-1", {
        x: xpos
    });

    xpos = (ev.pageX / 30);

    gsap.to(".titre-parallax-2", {
        x: xpos
    });
});
```

On récupère la valeur horizontale avec ev.pageX qui retourne la position de la souris en pixel. Ensuite il suffit de diviser ce nombre par un grand nombre pour diminuer

l'amplitude des déplacements. Enfin, on affecte cette valeur à x dans gsap.to().

La deuxième animation est le scroll horizontal de la page sur la partie article. Pour réaliser ceci, on utilise la propriété ScrollTrigger. Avant tout, on déplace la div à -100%, car c'est l'état de scroll final que l'on veut atteindre. On affecte la largeur de l'écran à x pour que, lorsqu'on arrive à la fin de la page, la div ne sorte complètement de l'écran, mais qu'elle reste à une certaine largeur de la gauche.

```
gsap.to(".horizontal-box", {  
    x: innerWidth,  
    xPercent: -100,  
    ease: "none"  
});
```

A ceci, il faut ajouter la propriété ScrollTrigger, pour connecter la div au scroll de l'utilisateur. On utilise scrub: true et pin: true pour que la div soit liée au scroll. On définit la fin de l'animation à innerWidth qui correspond à la taille de l'écran, pour que le scroll horizontal s'arrête lorsqu'on arrive en fin de page. On a donc:

```
gsap.to(".horizontal-box", {  
    x: innerWidth,  
    xPercent: -100,  
    ease: "none",  
    scrollTrigger: {  
        trigger: ".horizontal-box",  
        start: "top top",  
        end: () => innerWidth,  
        scrub: true,  
        pin: true  
    }  
});
```

Pour finir, nous allons voir comment a été intégré l'API Maps.

## API OpenMaps :

Pour afficher une carte map notre site, plusieurs solutions s'offrent à nous, voyons deux d'entre elles. La première est l'utilisation de l'API Google Maps, qui une API nécessitant la création de clés propre à chaque carte. De plus, l'utilisation de Google Maps est devenue payante depuis peu. L'utilisation de la carte "Developper" aurait été possible mais elle n'offre que peu de possibilités. C'est donc pour cette raison que lors de ce projet j'ai choisi d'utiliser la deuxième possibilité qui est d'utiliser une API de carte Open Source.

[OpenStreetMaps](#) est une carte open source qui peut être utilisée gratuitement dans n'importe quel projet. Pour l'utiliser il faut passer par une librairie Javascript : [Leaflet](#). Pour créer une carte, il faut commencer par l'initialiser (on le fait via une fonction que l'on nomme InitMap). On déclare donc trois variables, la première prenant la longitude de l'endroit que l'on veut pointer, la deuxième la latitude et enfin la troisième une valeur nulle, qui sera notre carte à la fin.

```
var lat = 49.00568710008271;
var lon = 7.384707013780214;
var macarte = null;
```

Dans la fonction initMap, on commence par initialiser la variable macarte. Celle-ci prend comme valeurs:

- la div où l'on veut afficher notre carte,
- la latitude,
- la longitude,
- le zoom auquel on se place lors du chargement de la carte

On met donc:

```
macarte = L.map('MapsOpen').setView([lat, lon], 15);
```

Nous récupérons la carte avec ces variables sur le serveur de OpenStreetMap ([tile.openstreetmap.fr/osmfr](http://tile.openstreetmap.fr/osmfr)). On ajoute les copyright avec attribution. On définit le zoom max et zoom min à 1 et 20. On fait donc appel à titleLayer comme suit:

```
L.tileLayer('https://s.tile.openstreetmap.fr/osmfr/{z}/{x}/{y}.png', {
    attribution: 'données (c) <a
    href="//osm.org/copyright">OpenStreetMap</a>/ODbL - rendu <a
    href="//openstreetmap.fr">OSM France</a>',
    minZoom: 1,
    maxZoom: 20
}).addTo(macarte);
```

Enfin, pour ajouter des marqueurs il suffit de le déclarer et de l'ajouter à la carte de la manière suivante :

```
var marqueur = L.marker([lat, lon]).addTo(macarte);
```

Si on veut styliser le popup du marqueur il nous suffit d'ajouter bindPopup('Notre texte html')

Nous avons donc la fonction initMap, qu'on appelle juste avant la fonction main:

```
function initMap() {
    var map = L.map('MapsOpen')
    macarte = L.map('MapsOpen').setView([lat, lon], 15);
    L.tileLayer('https://s.tile.openstreetmap.fr/osmfr/{z}/{x}/{y}.png', {
        attribution: 'données (c) <a
        href="//osm.org/copyright">OpenStreetMap</a>/ODbL - rendu <a
        href="//openstreetmap.fr">OSM France</a>',
        minZoom: 1,
        maxZoom: 20
    }).addTo(macarte);
    var marqueur = L.marker([lat, lon])
        .bindPopup('<p style="font-family: Poppins, sans-serif; font-weight: 500; font-size: 20px;">Chalet du club Vosgien</p>')
        .addTo(macarte);
}
```

Nous avons l'intégration de toutes les pages ainsi que leurs animations. Une vidéo présentant l'ensemble du site est disponible dans le dossier contenant ce rapport. Voyons maintenant ce que m'a apporté ce projet dans la conclusion.

## CONCLUSION

Lorsque j'ai commencé ce projet, j'ai évalué le nombre d'heures que je devrais faire pour la réalisation complète de celui-ci. J'ai donc pensé qu'il serait possible de le réaliser à l'aide d'un framework. Cependant, je me suis rendu compte que le temps d'apprentissage d'un tel langage serait trop long. J'ai donc dû redéfinir mon cahier des charges pour l'adapter à ce projet. Aujourd'hui, j'ai passé environ 50 heures pour la maquette, une centaine d'heures sur la réalisation du site internet, avec donc environ 40 heures pour l'apprentissage des librairies, des API, de certains attributs css et enfin du Javascript. Pour ce qui est de la réalisation de ce présent rapport, il a fallu une vingtaine d'heures pour arriver à sa version finale. Un temps total de plus de 200 heures a donc été nécessaire pour la réalisation totale du projet, de sa première version en fin 2020 jusqu'à aujourd'hui.

Plusieurs problématiques se sont posées à moi lors de cette réalisation. Tout d'abord, cela m'a appris que déterminer le temps de travail sur un devoir de programmation est très compliqué. Il arrive de passer une dizaine d'heures sur des parties, ou sur la recherche de solution, où l'on pensait passer la moitié. Les deux parties les plus compliquées pour moi ont été, tout d'abord, la réalisation de la maquette. Prendre en main un nouveau logiciel comme Adobe XD a été une expérience nouvelle pour moi. J'ai pu découvrir une infime partie du travail de webdesigner, qui est nécessaire pour la réalisation d'un site. Cependant, malgré le passage obligé par cette partie, cela commençait à devenir long. Cela m'a pris le double du temps que j'avais prévu d'accorder au développement de la maquette. Le besoin de travailler sur le code se faisait ressentir. La deuxième partie la plus compliquée, a été l'apprentissage. Avant de commencer à utiliser une librairie, par exemple, il faut la comprendre, savoir comment s'utilisent les différentes fonctions, comment les mettre en œuvre, dans quel contexte, dans quel but. Mieux connaître le langage que l'on utilise permet de mieux corriger et plus rapidement, les différentes erreurs que l'on a faites.

Pour conclure, la réalisation de ce projet était une expérience très enrichissante. Dans le cadre de ma poursuite d'étude dans le domaine du web, j'ai aujourd'hui, des bases très solides. J'ai un bien meilleur niveau que lorsque j'ai commencé à travailler dessus. L'utilisation d'un logiciel pour la réalisation de maquettes, l'apprentissage des langages, de librairies, enfin la réalisation d'un rapport technique sont des compétences qu'il faut posséder pour évoluer en entreprise, ce projet m'a permis de me familiariser

avec. La réalisation de celui-ci m'a permis d'apprendre à travailler sur un projet de plusieurs mois. J'ai appris comment se déroule la création d'un site internet du stade de maquette, jusqu'à la fin de la programmation.