

# RAPPORT DE STAGE EN MILIEU PROFESSIONNEL

Stage effectué du 27/05/2024 au 05/07/2024 au sein de la société

**Web-Konseil** – 132 Rue Bossuet, 69006 Lyon



## Table des matières

I.	Présentation de l'entreprise .....	4
II.	Présentation du projet .....	5
1)	Maquette .....	6
2)	Base de données .....	9
III.	Développement du projet.....	10
1)	Environnement de développement .....	10
3)	Symfony .....	10
4)	Entity.....	11
5)	Repository .....	13
6)	Controller .....	15
7)	Les vues .....	18
8)	Connexion / Déconnexion .....	20
9)	Mise en production .....	23
IV.	Arborescence du projet.....	23
V.	Notions apprises .....	24
VI.	Conclusion.....	25
	Annexe 1 - Définitions .....	26
	Annexe 2 – Organigramme.....	27

## REMERCIEMENTS

Je tiens à remercier ici toute l'équipe de Web-Konseil pour son accueil chaleureux et bienveillant. En particulier M. Salim KORRICHI, mon maître de stage, qui m'a permis de découvrir des outils professionnels et m'a fait confiance pour réaliser un projet utile pour l'entreprise.

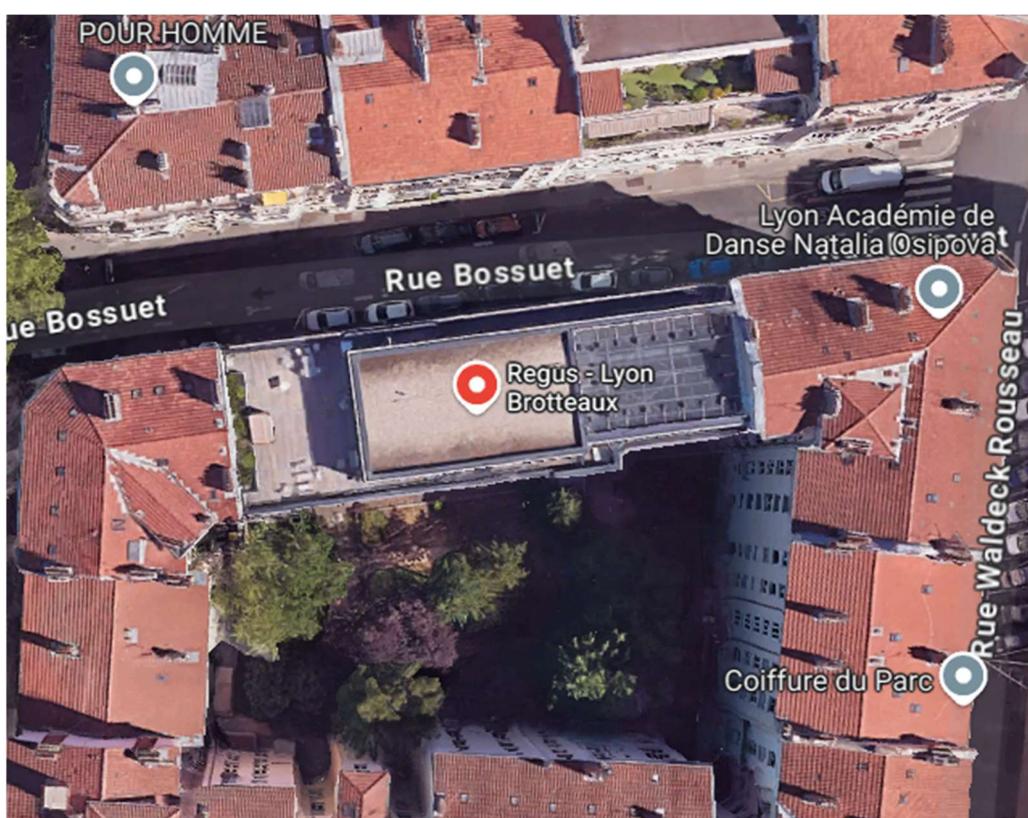
L'ensemble de l'équipe comprenant Morgane L., Kévin B., Eloy C. et Cédric P. m'a très bien accueilli, sans eux l'expérience aurait été différente. Dès les premiers jours, je me suis senti à l'aise grâce à leur bonne humeur et leur cohésion d'équipe, qu'ils en soient ici remerciés.

Ce stage a été enrichissant car, au-delà de l'aspect technique, cela m'a montré la vie en entreprise, avec les liens entre les professionnels et avec la hiérarchie.

## I. Présentation de l'entreprise

Web-Konseil est une agence web et digitale spécialisée dans la création et la maintenance de sites e-commerce Prestashop, dans le référencement naturel SEO, SEA, SMO (voir Annexe 1 - Définitions), le webmarketing et l'hébergement web. Elle est certifiée par Google Analytics, Google Ads, Oracle ou encore Prestashop Expert.

L'agence a ses bureaux dans le centre d'affaire « Regus » situé au 132 Rue Bossuet, 69006 Lyon.



Mais un déménagement est prévu fin août 2024 dans un nouveau bureau, cette fois indépendant, situé au 7 Rue Amédée Bonnet, 69006 Lyon. J'ai eu l'occasion de pouvoir le visiter lors de mon stage.



La société est constituée de 6 employés, travaillant tous à Lyon sauf un qui travaille à Paris et que je n'ai jamais rencontré (*voir Annexe 2 – Organigramme*).

Web-Konseil a accompagné beaucoup d'entreprises à créer leur site internet, héberge des sites web, corrige des bugs, réalise des migrations de sites e-commerce vers Prestashop et des migrations d'une ancienne version de Prestashop vers une plus récente.

Web-Konseil est très bien noté sur Google avec 5/5 étoiles sur 34 avis.

Tous les clients contributeurs sont très satisfaits de la prestation fournie par l'agence. L'entreprise se base sur quatre valeurs importantes : la réactivité, la proximité, l'innovation et l'expertise.

## II. Présentation du projet

Pour assurer la sécurité des sites web dont elle a la charge, la société Web-Konseil doit renouveler régulièrement les certificats SSL (*voir Annexe 1 - Définitions*). La plupart des certificats SSL sont renouvelés automatiquement, mais certains doivent l'être manuellement. Il est primordial de ne pas oublier cette tâche qui compromettrait l'intégrité et la sécurité des sites internet. Les professionnels doivent donc se mettre des alertes régulièrement pour effectuer ces renouvellements.

Mon maître de stage m'a donc proposé de créer une solution web, que nous avons nommé « SSL Manager », que l'entreprise utilisera en interne et qui permettra aux équipes réseaux de Web-Konseil d'être alertés par mail sur les certificats SSL qu'ils doivent renouveler manuellement. Cet outil leur permettra d'ajouter, de modifier et de supprimer les certificats SSL dont ils devront s'occuper. Les alertes et les relances se feront par mail, dans les délais programmés depuis les interfaces d'administration de la solution.

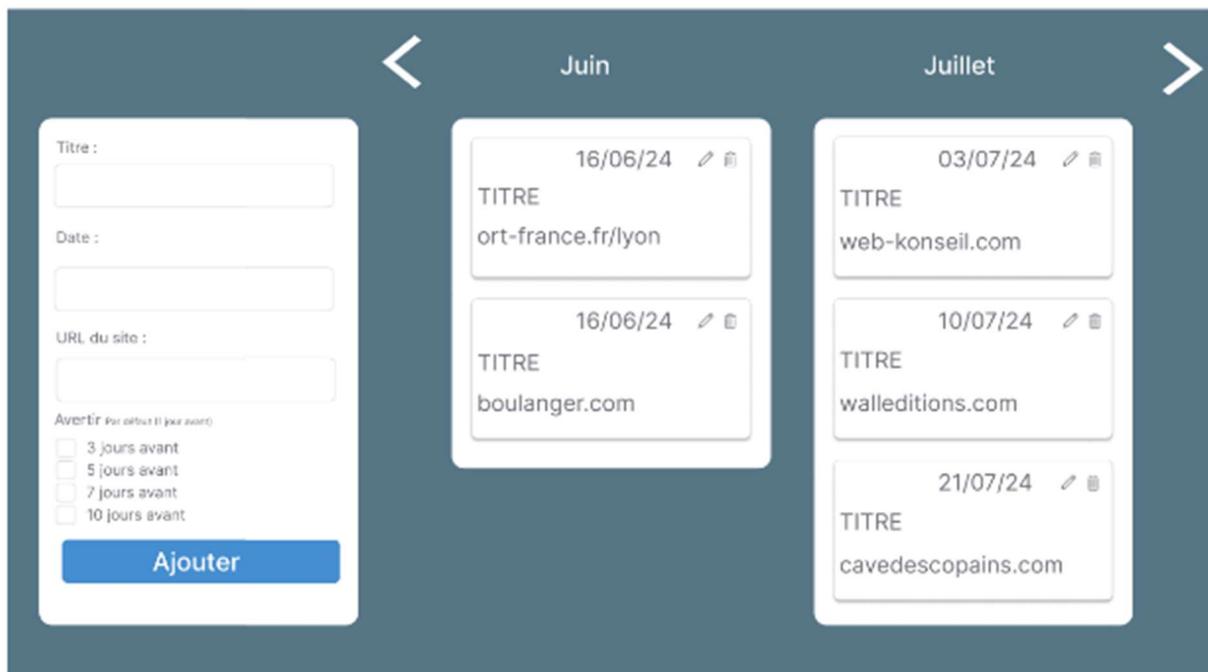
## 1) Maquette

Avant de me lancer dans le code de mon projet, j'ai d'abord créé des maquettes. Mon maître de stage m'a laissé deux jours pour penser à l'ergonomie et à l'expérience utilisateur. Les maquettes m'ont servi de fil rouge, me permettant de garder une vision claire de l'objectif final.

Pour faire mes maquettes j'ai utilisé le logiciel Figma (*voir Annexe 1 - Définitions*).

J'ai créé deux versions.

Voici la version 1 :



< Juin Juillet >

Titre :  
[ ]

Date :  
[ ]

URL du site :  
[ ]

Avertir Par défaut (1 jour avant)

- 3 jours avant
- 5 jours avant
- 7 jours avant
- 10 jours avant

Ajouter

< Juin Juillet >

Titre :  
[ ]

Date :  
[ ]

URL du site :  
[ ]

Avertir Par défaut (1 jour avant)

- 3 jours avant
- 5 jours avant
- 7 jours avant
- 10 jours avant

Ajouter

16/06/24 [ ] [ ]

03/07/24 [ ] [ ]

web-konseil.com

10/07/24 [ ] [ ]

ditions.com

21/07/24 [ ] [ ]

TITRE

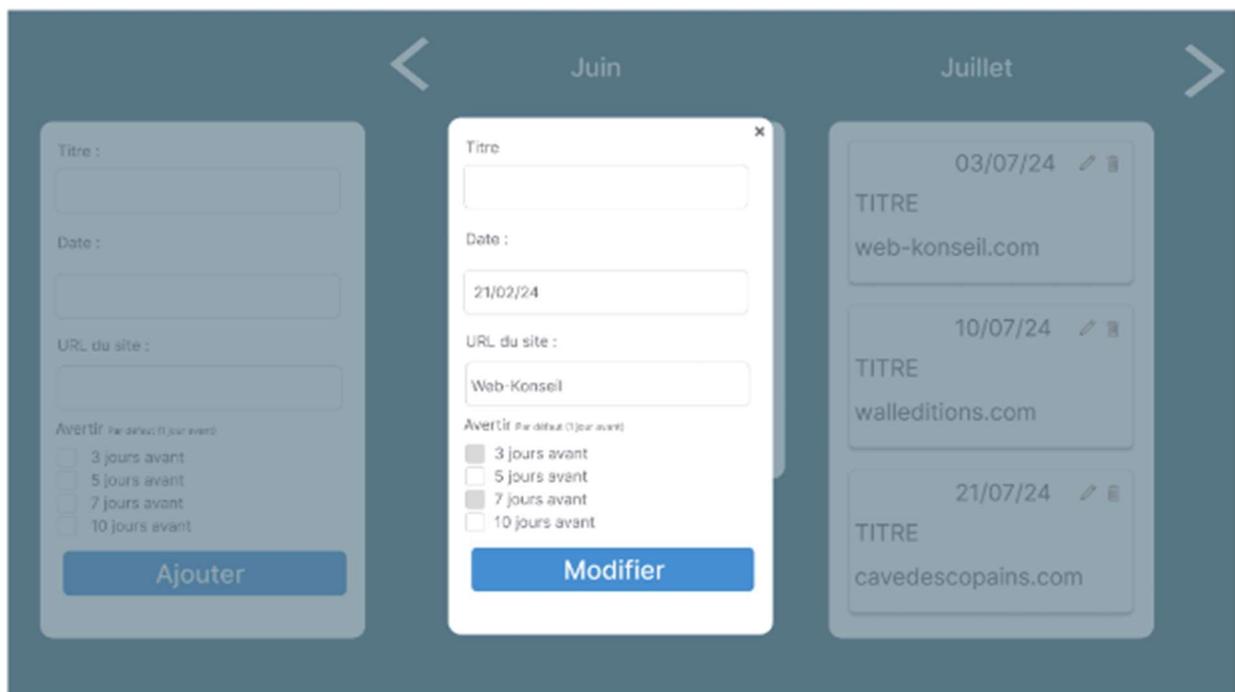
cavedescopains.com

Etes vous sur de vouloir supprimer définitivement cet événement ?

TITRE

web-konseil.com 03/02/24

Non Oui



Et la version 2 :





Plus simple à concevoir dans le délai du stage, nous avons choisi avec mon maître de stage la 2<sup>nd</sup> version.

## 2) Base de données

Une fois les maquettes réalisées et validées par mon maître de stage, j'ai établi une

structure de base de données qui reprend les champs importants selon le cahier des charges. On m'a conseillé aussi d'ajouter deux champs de dates pour enregistrer la date de création et la date de mise à jour de chaque événement.

#### **Base de donnée : TABLE EVENT**

**ID** : INT AUTO\_INCREMENT PRIMARY KEY

**title\_event** : STRING-VARCHAR(255)

**date\_event** : DATETIME

**url\_event** : STRING-VARCHAR(70)

**create\_date\_event** : DATETIME

**update\_date\_event** : DATETIME

Plus tard dans le projet, j'ai enrichi la base de données avec une nouvelle table d'utilisateurs.

## III. Développement du projet

### 1) Environnement de développement

J'ai choisi de travailler avec PhpStorm qui pour moi est le meilleur environnement de développement pour PHP. Ce qui me permet d'affirmer que PhpStorm est le meilleur choix d'IDE (*voir Annexe 1 - Définitions*) ce n'est pas sa puissance, ou le fait qu'il y ait la fonctionnalité de complétion de code et l'historique de versions, mais c'est le fait que j'ai essayé différents IDE (VisualStudioCode, Replit) et un éditeur de texte (Sublime Text) qui pour moi avaient une qualité inférieure à PhpStorm.

Je l'ai utilisé aussi car j'avais la chance d'avoir un accès gratuit grâce à la licence éducative.

### 3) Symfony

Pour débiter mon projet, j'ai dû apprendre à utiliser Symfony. Symfony est un Framework PHP qui permet de grandement accélérer les développements. Un Framework fournit une base de code pour aider les développeurs à démarrer

rapidement leur projet sans qu'ils aient besoin de devoir tout créer depuis zéro.

Tout d'abord, je devais installer Composer qui est un outil de gestion de dépendance PHP et créer le squelette du projet grâce à Symfony qui le génère. Pour cela, j'ai dû utiliser les commandes suivantes dans mon terminal qui me permettent d'installer Composer :

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"  
php -r "if (hash_file('sha384', 'composer-setup.php') ===  
'93b54496392c062774670ac18b134c3b3a95e5a5e5c8f1a9f115f203b75bf9a129d5da  
a8ba6a13e2cc8a1da0806388a8') { echo 'Installer verified'; } else { echo 'Installer  
corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"  
php composer-setup.php  
php -r "unlink('composer-setup.php');"
```

Et celle-ci pour créer le squelette du projet :

```
composer create-project symfony/website-skeleton Projet-Stage
```

Ensuite dans PhpMyAdmin j'ai créé la base de données mais sans ajouter des tables. J'ai appelé ma base de données « projet-stage ».

J'ai ajouté mes tables avec Symfony. Pour cela j'ai ouvert le terminal de mon IDE et tapé les commandes suivantes qui permettent de créer les tables mais aussi de créer le fichier Repository et Entity.:

```
Symfony console make:entity
```

#### 4) Entity

Une entité est un type d'objets qui sera manipulé par l'application Symfony. Les entités sont converties en tables de la base de données, à partir du schéma.

Voici ma base de données générée :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id	int			Non	Aucun(e)		AUTO_INCREMENT	Modifier  Supprimer Plus
2	url_event	varchar(70)	utf8mb4_unicode_ci		Non	Aucun(e)			Modifier  Supprimer Plus
3	date_event	datetime			Non	Aucun(e)			Modifier  Supprimer Plus
4	warning	smallint			Non	Aucun(e)			Modifier  Supprimer Plus
5	create_date_event	datetime			Non	Aucun(e)			Modifier  Supprimer Plus
6	update_date_event	datetime			Non	Aucun(e)			Modifier  Supprimer Plus
7	location_serveur	varchar(255)	utf8mb4_unicode_ci		Oui	NULL			Modifier  Supprimer Plus

Voici une partie de mon Entity final « Event »

```
1 <?php
2
3 namespace App\Entity;
4
5 use App\Repository\EventRepository;
6 use Doctrine\DBAL\Types\Types;
7 use Doctrine\ORM\Mapping as ORM;
8
9
10 #[ORM\Entity(repositoryClass: EventRepository::class)]
11 class Event
12 {
13     1 usage
14     #[ORM\Id]
15     #[ORM\GeneratedValue]
16     #[ORM\Column]
17     private ?int $id = null;
18
19     2 usages
20     #[ORM\Column(length: 70)]
21     private $urlEvent = '';
22
23     2 usages
24     #[ORM\Column(type: Types::DATETIME_MUTABLE)]
25     private ?\DateTimeInterface $date_event = null;
26
27     2 usages
28     #[ORM\Column(length: 255)]
29     private $locationServeur = '';
```

```

11 class Event
35     public function getId(): ?int
36     {
37         return $this->id;
38     }
39
40     1 usage
41     public function getUrlEvent(): ?string
42     {
43         return $this->urlEvent;
44     }
45
46     2 usages
47     public function setUrlEvent(?string $urlEvent): self
48     {
49         $this->urlEvent = $urlEvent ?? '';
50         return $this;
51     }
52
53     3 usages
54     public function getDateEvent(): ?\DateTimeInterface
55     {
56         return $this->date_event;
57     }
58
59     2 usages
60     public function setDateEvent(\DateTimeInterface $date_event): static
61     {
62         $this->date_event = $date_event;
63
64         return $this;

```

## 5) Repository

Un Repository est une classe qui permet de faire des requêtes sur une table de la base de données.

Ceci est mon Repository final « EventRepository »

```
1 <?php
2
3 namespace App\Repository;
4
5 use App\Entity\Event;
6 use App\Entity\Users;
7 use Doctrine\Bundle\DoctrineBundle\Repository\ServiceEntityRepository;
8 use Doctrine\Persistence\ManagerRegistry;
9 use Doctrine\DBAL\Connection;
10
11 /**
12  * @extends ServiceEntityRepository<Event>
13  *
14  * @method Event|null find($id, $lockMode = null, $lockVersion = null)
15  * @method Event|null findOneBy(array $criteria, array $orderBy = null)
16  * @method Event[] findAll()
17  * @method Event[] findBy(array $criteria, array $orderBy = null, $limit = null, $offset = null)
18  */
19 class EventRepository extends ServiceEntityRepository
20 {
21     2 usages
22     private $connection;
23
24     public function __construct(ManagerRegistry $registry)
25     {
26         parent::__construct($registry, Event::class);
27         $this->connection = $this->getEntityManager()->getConnection();
28     }
29 }
```

```
29
30 /**
31  * @return array Returns an array of Event objects expiring soon
32  */
33 1 usage
34 public function sendEmail(): array
35 {
36     $sql = 'SELECT *
37           FROM event
38           WHERE DATE_SUB(date_event, INTERVAL 2 DAY) = CURDATE()
39           OR DATE_SUB(date_event, INTERVAL warning DAY) = CURDATE()';
40
41     $result = $this->connection->executeQuery($sql);
42     return $result->fetchAllAssociative();
43 }
44 }
```

## 6) Controller

Après avoir généré la base de données, le fichier Entity et le fichier Repository, je vais ensuite générer le fichier Controller grâce à Symfony.

Un Controller est un élément clé de l'architecture MVC (*voir Annexe 1 - Définitions*), c'est le cerveau de l'architecture, c'est lui qui va gérer la relation entre les modèles, entre autre la base de donnée et les vues.

Pour générer le Controller il suffit de taper cette commande dans le terminal de l'IDE **Symfony console make :controller**

Une fois le Controller, l'Entity et le Repository générés, j'étais maintenant prêt à commencer à coder mon projet.

J'ai donc choisis de commencer par coder la page « Ajouter un évènement » ce qui m'a demandé un certain temps car j'ai dû apprendre le fonctionnement du Controller et sa logique.

Pour commencer, j'ai défini une route pour la méthode d'ajout d'événements : la route "/add".

J'ai ensuite rajouté une vérification pour m'assurer que le champ « URL » contenait bien une url et que le champ « date de renouvellement » ne soit pas une date passée. S'il y a une erreur, un message s'affiche en fonction de l'erreur et garde les données saisies pour qu'elles puissent être modifiées afin de les rendre valides.

The image displays two side-by-side screenshots of a web form titled "Ajouter un évènement".

**Left Screenshot:** Shows a validation error "URL invalide" in a red banner. The "URL du site:" field contains "site1" and is highlighted with a red border. The "Date de renouvellement:" field contains "15/07/2024". The "Localisation du serveur" field contains "Paris". The "Alerte J-:" field contains "3". A green "+ Ajouter" button and a blue "Retour à la liste" button are visible at the bottom.

**Right Screenshot:** Shows a validation error "Date invalide" in a red banner. The "URL du site:" field contains "https://site1". The "Date de renouvellement:" field contains "01/01/2000" and is highlighted with a red border. The "Localisation du serveur" field contains "Paris". The "Alerte J-:" field contains "3". A green "+ Ajouter" button and a blue "Retour à la liste" button are visible at the bottom.

Si toutes les données sont remplies et valides, cela crée un nouvel événement qui est ensuite sauvegardé dans la base de données.

	id	url_event	date_event	warning	create_date_event	update_date_event	location_serveur
  	137	https://site1	2024-07-15 00:00:00	3	2024-07-09 09:13:27	2024-07-09 09:13:27	Paris

Après avoir créé la page « Ajouter un événement », je m'attaque maintenant à la page principale de mon site : la page « Liste des événements ». Cette page recense tous les événements ajoutés.

La méthode récupère tous les événements depuis la base de données en utilisant l'entité « Event ». Une fois les événements récupérés, ils sont triés dans le tableau pour que les événements les plus récents en fonction de la date de renouvellement apparaissent en haut du tableau.

URL Cible	Date de renouvellement	Localisation du serveur	Relance	Actions
https://site1	15-07-2024	Paris	J-3	 

Puis j'ai réalisé la page « Modifier l'événement », sensiblement similaire à la page « Ajouter un événement ».

Quoique semblable à ce que j'avais déjà fait, j'ai rencontré quelques problèmes lors de la réalisation de cette page. Il m'a fallu récupérer les données de la base de données et modifier les données dans la base de données. Cela m'a pris du temps mais j'ai réussi à résoudre seul ces difficultés.

### Modifier l'évènement

URL du site:

Date de renouvellement:

Localisation du serveur:

Alerte J- :

URL Cible	Date de renouvellement	Localisation du serveur	Relance	Actions
https://site1	15-07-2024	Londres	J-3	 

	id	url_event	date_event	warning	create_date_event	update_date_event	location_serveur
<input type="checkbox"/>   	137	https://site1	2024-07-15 00:00:00	3	2024-07-09 09:13:27	2024-07-09 15:32:23	Londres

Après avoir établi la partie « Modifier l'événement », je me suis occupé de la partie « Suppression », pour que les événements plus utiles puissent être supprimés manuellement. Pour cela j'avais besoin de récupérer l'id de l'événement qui me permet de supprimer la bonne ligne dans la base de données. Je devais donc modifier la route pour y ajouter l'id. Voici la route que j'ai établie : /delete/{id}

Voici la liste une fois les événements supprimés :

URL Cible	Date de renouvellement	Localisation du serveur	Relance	Actions
-----------	------------------------	-------------------------	---------	---------

Et la base de données :

id	url_event	date_event	warning	create_date_event	update_date_event	location_serveur
----	-----------	------------	---------	-------------------	-------------------	------------------

Et pour finir mon Controller « Event », je me suis occupé de la partie envoi de mail. Pour cela mon maître de stage m'a fourni l'accès SMTP pour pouvoir envoyer le mail. Le SMTP est un protocole qui permet d'envoyer un mail d'un serveur à un autre. Sans ce protocole, il serait impossible d'envoyer des emails d'une messagerie à une autre et il n'y aurait ni protections ni vérifications de l'identité de l'expéditeur, ce qui augmenterait grandement l'usurpation d'identité.

Pour faire les tests j'ai utilisé le site MailTrap qui fournit un SMTP.

Pour envoyer les mails à la date renseignée lors de la création de l'événement (en fonction de la date de relance qui a été demandée), tout se passe avec des requêtes SQL.

Voici celui que j'utilise dans mon fichier Repository :

```
$sql = 'SELECT *
FROM event
WHERE DATE_SUB(date_event, INTERVAL 2 DAY) = CURDATE()
OR DATE_SUB(date_event, INTERVAL warning DAY) = CURDATE()';
```

Cette requête SQL calcule l'intervalle entre la date de renouvellement et la date de relance pour savoir quel jour envoyer le mail.

Mon maître de stage m'a demandé de rajouter un email automatique, quelle que soit la date de relance, deux jours avant la date de renouvellement.

Voici l'exemple d'un mail :

Event Notification Boîte de réception x

À moi ▾

## Rappel : Renouvellement de certificat SSL

Le certificat SSL pour l'URL suivante arrive bientôt à expiration :

URL : <https://site2>

Localisation du serveur : Londres

Date d'expiration : 15-07-2024

## 7) Les vues

Maintenant que mon Controller est prêt, je débute le développement des vues qui permettent d'avoir un affichage sur le navigateur. Symfony utilise le template Twig (*voir Annexe 1 - Définitions*) pour l'affichage des vues sur le navigateur, ce qui permet de séparer la partie backend de la partie frontend (*voir Annexe 1 - Définitions*). J'ai trouvé ce principe très utile car les fichiers ont des noms d'extension de fichiers différents selon leur utilisation.

J'ai donc commencé par coder ma page « Ajouter un événement ». Pour cette page je n'ai rencontré aucune difficulté.

Puis j'ai continué avec la page « Liste des événements », et pour cette page j'ai eu un problème.

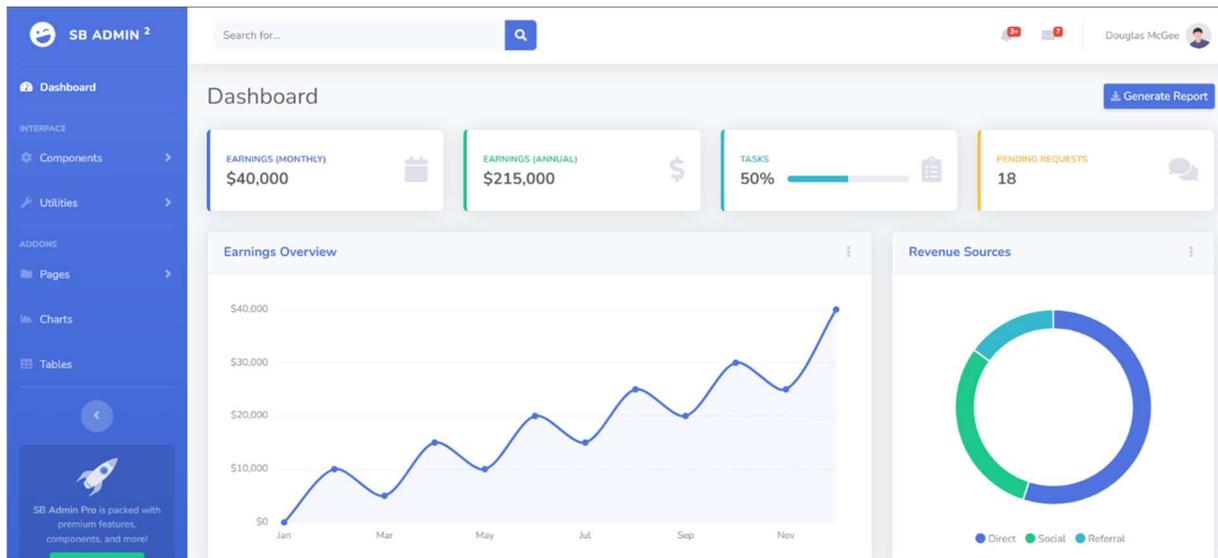
Notamment à cause d'une fonctionnalité visuelle qui permet à l'entreprise de voir si les dates sont au bord de l'échéance ou pas avec différentes couleurs dans le tableau (Vert : plus de 15 jours, Orange : entre 8 et 15 jours, Rouge : entre 0 et 7 jours et Gris : date passée).

Ma page « Liste des événements » contient des parties de code Bootstrap pour améliorer le visuel.

Bootstrap est un framework qui facilite la création de design d'un site web. J'ai utilisé le Bootstrap SB Admin2. L'avantage de Bootstrap est de faciliter la mise en forme de certains éléments des interfaces (tableaux, boutons...). J'ai uniquement récupéré les

bouts de code qui m'étaient utiles. L'autre avantage de Bootstrap est le fait que le rendu soit très esthétique.

Voici l'exemple de page sur laquelle j'ai basé mon interface « Liste des événements »

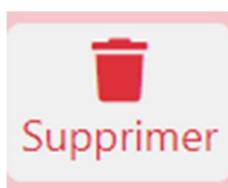


Pour faire les icônes de mes boutons, j'ai utilisé « Front Awesome » qui est une bibliothèque d'icônes. Beaucoup sont payantes mais certaines, gratuites, sont intéressantes. Au début, j'utilisais des icônes images mais mon maître de stage m'a conseillé d'utiliser Front Awesome car il propose des icônes sous forme de code.

Par exemple, j'ai utilisé une poubelle pour supprimer un événement.

```
<button type="button" class="btnAll btn-delete" data-toggle="modal" data-target="#deleteModal" data-id="{{ event.id }}">  
  <i class="fa-solid fa-trash fa-xl"></i>  
  <div>Supprimer</div>  
</button>
```

Et au final, le bouton ressemble à cela :



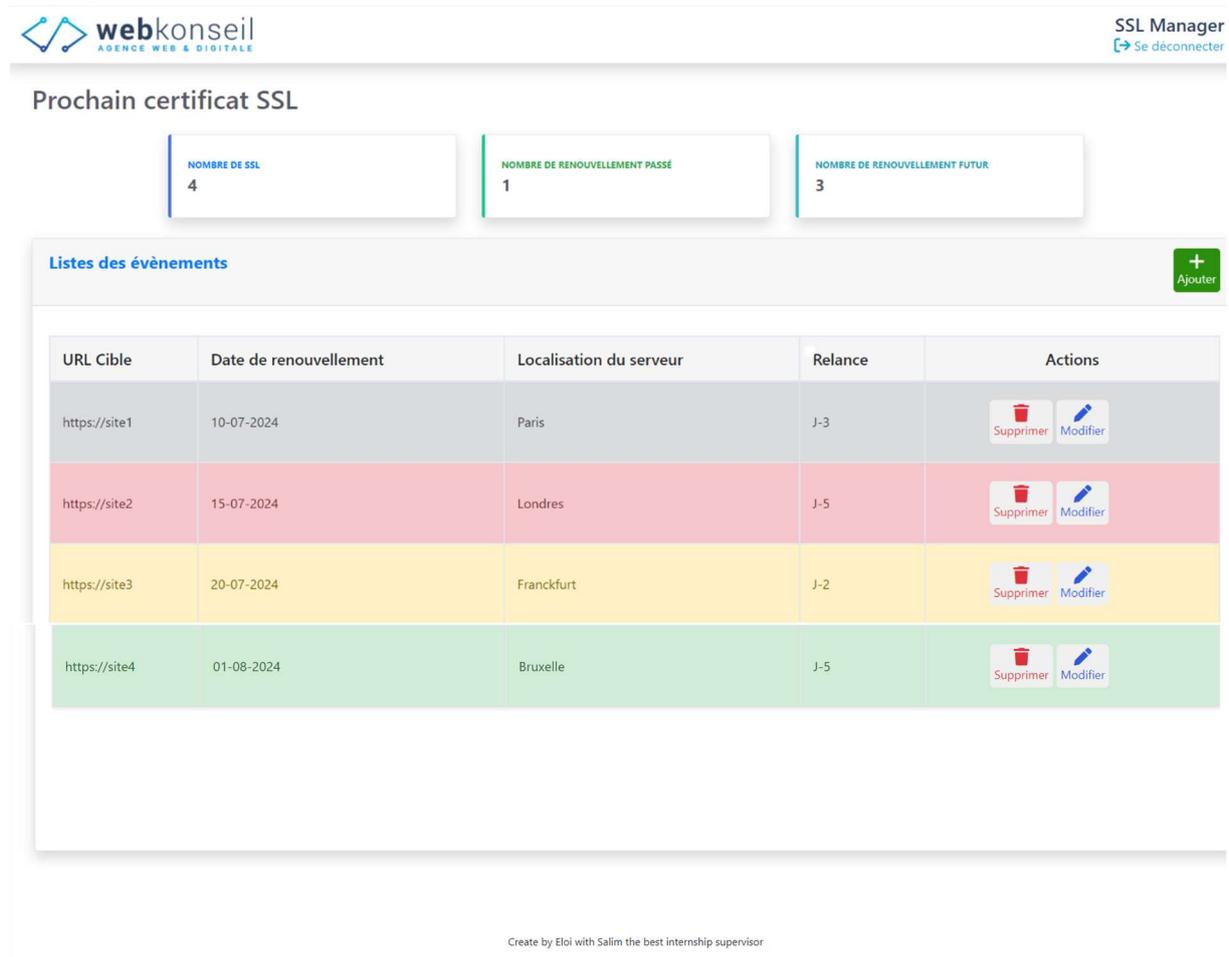
L'utilisation des templates Twig permet également de centraliser les éléments communs tels que le « header et le footer », simplifiant ainsi la gestion et l'uniformité de l'interface.

Ainsi en ajoutant une ligne de code dans mes templates twig « Liste des événements »,

« Ajouter un événement » et « Modifier un événement » le « header et le footer » s'affichent.

```
1 {% extends 'base.html.twig' %}
```

La page « Liste des événements » finale ressemble à cela :



## 8) Connexion / Déconnexion

Après avoir terminé le projet initial établi avec mon tuteur, il me restait encore une semaine de stage. Il m'a demandé de m'occuper de la page « connexion/déconnexion ». La partie « inscription » sera faite directement dans la base de données pour éviter qu'une personne extérieure à l'entreprise ne puisse se connecter et modifier des informations, voire les supprimer.

Je ne savais pas que j'entrais dans la période la plus difficile de mon projet.

Pour commencer, il m'a fallu créer une nouvelle table dans ma base de données, ce qui signifiait aussi générer un second Repository, un second Entity et un second Controller. Une fois ces trois fichiers générés, j'ai commencé par créer un utilisateur test dans ma base de données. Deux champs sont nécessaires, un email et un mot de passe qui sera

crypté en SHA256 (voir Annexe 1 - Définitions).

Voilà à quoi ressemble ma base de données :

	id	email	password
2		eloi.poly@gmail.com	31f7a65e315586ac198bd798b6629ce4903d0899476d5741a9...

J'ajoute ensuite cette requête SQL dans mon Repository, ce qui permet de vérifier si l'utilisateur existe :

```
24     $sql = 'SELECT `u`.`id`, `u`.`email`, `u`.`password`
25           FROM `users` `u`
26           WHERE `u`.`email` = :email
27           AND `u`.`password` = :password';
```

Puis le Controller doit être configuré.

Malheureusement, j'ai sous-estimé cette étape de mon projet, impossible de me connecter avec les identifiants renseignés en base de données et j'ai passé plusieurs heures à tenter de résoudre ce problème.

Le problème n'était pas dans le Controller, mais dans l'authentification, plus précisément la vérification du hashage du mot de passe. Mon tuteur m'informe que le problème est le salage des mots de passe dont j'ignorais l'existence.

Le salage des mots de passe est une technique de sécurité utilisée pour renforcer les mots de passe en ajoutant des données supplémentaires avant d'être hashés. Ce qui signifie que le mot de passe que j'écrivais dans mon formulaire était forcément faux. J'ai donc demandé à mon maître de stage s'il pouvait m'aider à régler le problème, ce qu'il a fait en supprimant l'option de salage.

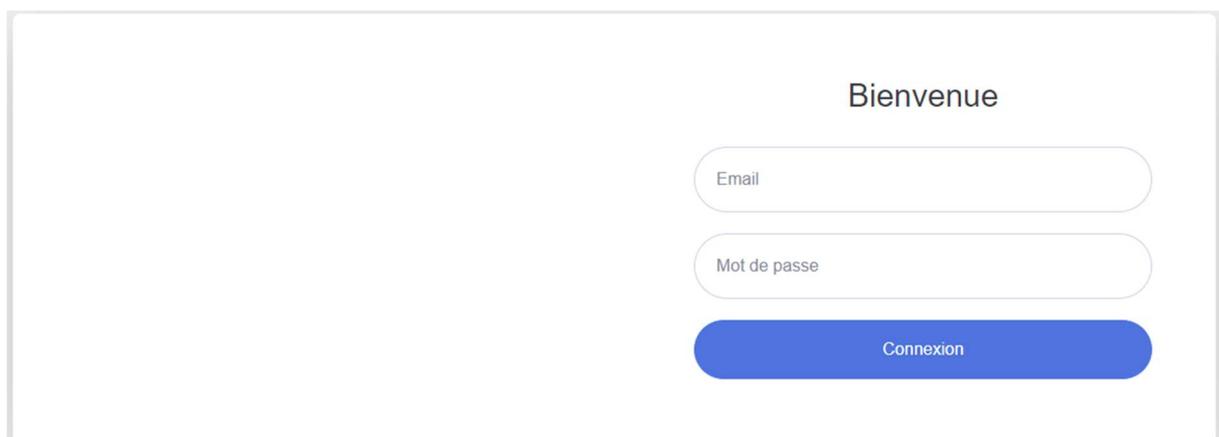
Ceci est le code qui permet la connexion au site.

```
24 public function index(Request $obj_request): Response
25 {
26
27     if ($this->isUserLogged($obj_request)) {
28         return $this->redirectToRoute( route: 'list_event');
29     }
30
31     if ($obj_request->isMethod( method: 'POST')) {
32
33         $user_data = $this->entityManager->getRepository(User::class)->authenticate(
34             $obj_request->request->get( key: 'email'),
35             $obj_request->request->get( key: 'password')
36         );
37
38         if (!empty($user_data) && is_array($user_data)) {
39             $this->addSessionValues($obj_request, array(
40                 'user_id' => $user_data['id'],
41                 'email' => $user_data['email'],
42             ));
43
44             return $this->redirectToRoute( route: 'list_event');
45         } else {
46             $this->addFlash( type: 'error', message: 'Authentication failed. Please check your credentials and try again. ');
47             return $this->render( view: 'event/login.html.twig', [
48                 'email' => $obj_request->request->get( key: 'email'),
49                 'authentication_error' => true
50             ]);
51         }
52     }
53 }
```

Puis j'ai ajouté une sécurité qui vérifie :

- Si l'utilisateur est connecté alors il a accès au site
- Si l'utilisateur n'est pas connecté, il doit se connecter pour accéder au site
- Si l'utilisateur n'est pas connecté et qu'il recherche une page spécifique dans le site (comme par exemple « /add »), il sera renvoyé sur la page de connexion

Voici ma page de connexion :



The image shows a login form with the following elements:

- Header: "Bienvenue"
- Input field: "Email"
- Input field: "Mot de passe"
- Button: "Connexion" (blue)

## Page de connexion en cas d'erreur :

Email ou mot de passe incorrect

Bienvenue

azerty@gmail.com

Mot de passe

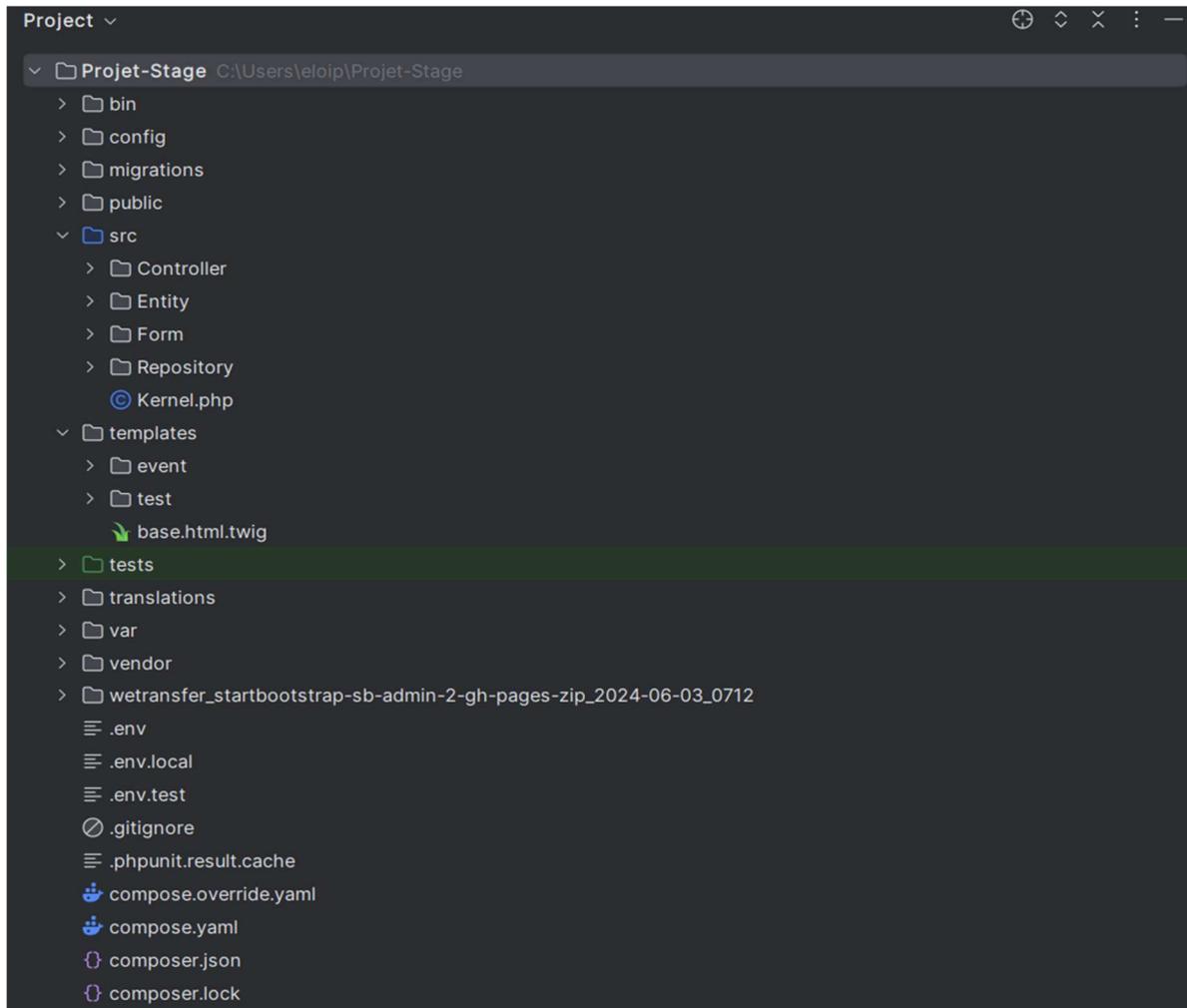
Connexion

### 9) Mise en production

Après avoir finalisé le code, je m'attèle à la mise en production de mon projet. J'ai utilisé Filezilla (*voir Annexe 1 - Définitions*) pour envoyer tous mes fichiers sur le serveur de l'entreprise. A ma grande surprise, mes vingt-neuf mille fichiers ont été envoyés sans problème. J'ai aussi importé ma base de données. J'ai bien évidemment testé toutes les fonctionnalités en ligne, que ce soit la connexion, l'ajout, la modification et la suppression d'événement, la déconnexion et l'envoi de mails. Mon maître de stage s'est occupé de finaliser la mise en production pour que le site apparaisse en ligne.

## IV. Arborescence du projet

Symfony génère lui-même une arborescence avec un classement des fichiers optimisé. J'ai utilisé les dossiers préexistant dans l'arborescence de Symfony pour y ajouter mes fichiers créés.



## V. Notions apprises

Ce stage a été très formateur car il m'a permis de découvrir de nouvelles notions, principalement liées à l'utilisation de Symfony. Sans ce Framework, la difficulté de mon projet aurait augmenté et sa faisabilité diminuée. De plus, il est particulièrement pertinent d'apprendre à manipuler des outils utilisés par les professionnels du web.

En premier lieu, j'ai appris à installer le Framework Symfony, ce qui était nouveau pour moi. J'ai été amené à faire des recherches très précises sur Internet et à trouver de l'aide auprès d'intelligences artificielles comme ChatGPT, Claude, Gemini et Llama 3 pour trouver les commandes adéquates pour l'installation. J'ai réalisé qu'il fallait être précis dans l'intitulé de mes recherches.

En formation, nous apprenons à créer et à remplir la base de données et des fichiers avec des « Getter et Setter » manuellement, alors que Symfony crée tout cela automatiquement.

Ensuite, je me suis heurté à des difficultés comme des erreurs de logique. Pour remédier à cela, j'ai appris à utiliser les fonctions PHP « var\_dump » et « die » qui sont

des outils de débogage. Mais j'ai également appris à repérer mes erreurs en relisant mon code.

J'ai également renforcé ma connaissance en PHP, HTML et CSS.

## VI. Conclusion

Mon stage chez Web-Konseil a été une expérience très enrichissante. J'ai eu l'opportunité de travailler sur un projet concret de développement d'une solution interne, ce qui m'a permis d'améliorer mes compétences en PHP, Symfony et base de données. Mon maître de stage m'a aidé à plusieurs reprises, mais il m'a plus souvent orienté pour que je puisse trouver la solution moi-même et cela a été très formateur.

Au-delà de l'aspect technique, ce stage m'a aussi permis de découvrir le monde du travail. L'accueil chaleureux de toute l'équipe de Web-Konseil a grandement contribué à cette expérience plus que positive, me permettant de m'intégrer rapidement.

Ce stage m'a non seulement permis de développer mes compétences techniques, mais il m'a aussi donné un aperçu de la vie en entreprise. Je suis convaincu que les connaissances et l'expérience acquises durant mon stage me seront très utiles pour la suite de mon parcours scolaire et professionnel. De plus, en accord avec l'entreprise, je vais poursuivre le développement du projet en dehors du stage afin de l'améliorer. Cela me permettra de continuer à progresser en tenant le site à jour puisque, plus qu'un exercice de stage, cet outil sera réellement utilisé par l'entreprise.

## Annexe 1 - Définitions

**Backend** : partie cachée de l'application qui gère la logique, les bases de données et les serveurs.

**Certificat SSL** : certificat électronique qui identifie et authentifie l'identité d'un site web et permet une connexion chiffrée et sécurisée.

**Figma** : éditeur graphique utile pour le design digital et la conception d'interfaces de sites web.

**Filezilla** : permet de transférer des fichiers entre l'ordinateur et un serveur distant

**Footer** : La partie inférieure d'une page web, où l'on trouve généralement des informations de contact, des mentions légales...

**Frontend** : partie visible de l'application, il est codé généralement en HTML CSS.

**Header** : Partie supérieure d'une page web, contenant souvent le logo, le menu de navigation, et d'autres éléments importants.

**IDE** : Integrated Development Environment, logiciel de développement de code efficace incluant un débogueur et un compilateur.

**MVC** : Model-View-Controller, méthodologie ayant pour objectif de faire le lien entre l'interface utilisateur et les modèles de données.

**SEA** : Search Engine Advertising, permet de diffuser des annonces ciblées pour gagner en visibilité. Référencement payant.

**SEO** : Search Engine Optimization, techniques qui permettent d'améliorer la position d'un site web dans un moteur de recherche. Référencement naturel gratuit.

**SHA256** : Secure Hash Algorithm 256 bits, algorithme de hashage généralement utiliser pour crypter des mots de passe dans une base de données.

**SMO** : Social Media Optimisation, utilise des médias et réseaux sociaux pour optimiser le référencement d'un site web.

**SSL** : Secure Sockets Layer, protocole de sécurité qui crée un lien chiffré entre un serveur et un navigateur.

**Template** : modèle préconçu utilisé pour créer des sites web.

## Annexe 2 – Organigramme

